Polunadzirano učenje semantičke segmentacije slika primjenom konzistencijskih kriterija

Grubišić, Ivan

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivan Grubišić

POLUNADZIRANO UČENJE SEMANTIČKE SEGMENTACIJE SLIKA PRIMJENOM KONZISTENCIJSKIH KRITERIJA

DOKTORSKI RAD

Zagreb, 2023.

Sveučilište u Zagrebu

FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

Ivan Grubišić

POLUNADZIRANO UČENJE SEMANTIČKE SEGMENTACIJE SLIKA PRIMJENOM KONZISTENCIJSKIH KRITERIJA

DOKTORSKI RAD

Mentor: Prof. dr. sc. Siniša Šegvić

Zagreb, 2023.

University of Zagreb

FACULTY OF ELECTRICAL ENGINEERING AND COMPUTING

Ivan Grubišić

SEMI-SUPERVISED LEARNING OF SEMANTIC SEGMENTATION OF IMAGES BY APPLYING CONSISTENCY CRITERIA

DOCTORAL THESIS

Supervisor: Professor Siniša Šegvić, PhD

Zagreb, 2023

Doktorski rad izrađen je na Sveučilištu u Zagrebu Fakultetu elektrotehnike i računarstva, na Zavodu za elektroniku, mikroelektroniku, računalne i inteligentne sustave.

Mentor: prof. dr. sc. Siniša Šegvić

Doktorski rad ima: 121 stranicu

Doktorski rad br.: _____

O mentoru

Siniša Šegvić rođen je 1971. u Splitu. Osnovnu školu i gimnaziju završio je u Zadru osim osmog razreda osnovne škole, koji je pohađao u Milanu. Diplomirao je elektrotehniku na zagrebačkom ETF-u (1996.), gdje je i magistrirao (2000.) i doktorirao (2004.) te se zaposlio kao docent od 2006. godine.

Bio je postdoktorski istraživač na institutu IRISA u Rennesu (2006. – 2007.) te na TU Graz (2007. – 2008.). Vodio je tri istraživačka projekta Hrvatske zaklade za znanost (MultiCLOD, MASTIF, ADEPT) te više industrijskih istraživačkih projekata koje su financirale tvrtke Rimac automobili, RoMB, MicroBlink te Promet i prostor. Sudjelovao je u istraživačkom centru izvrsnosti DATACROSS, na nekoliko ERDF projekata (SafeTram, MAS, A-UNIT), kao i na jednom projektu iz programa FP7 (ACROSS).

Njegovi istraživački i profesionalni interesi uključuju računalni vid, strojno učenje, razumijevanje scena, i gustu predikciju dubokim konvolucijskim modelima. Objavio je 5 radova na vrhunskim konferencijama računalnog vida i umjetne inteligencije (3×CVPR, 2×ECCV, NeurIPS) te 12 radova u časopisima koje indeksira SCI. Recenzent je na vrhunskim konferencijama računalnog vida i umjetne inteligencije, kao i u znanstvenim časopisima u područjima računalnog vida, inteligentnih transportnih sustava i robotike. Sudjelovao je u industrijskom razvoju kao tehnički konzultant. Mentorira više doktoranada koje financiraju europski projekti, nacionalni projekti i privatne tvrtke. Njegova istraživačka grupa postigla je zapažene rezultate na nekoliko natjecanja u računalnom vidu (WildDash, Robust vision challenge, Cityscapes, Fishyscapes i SegmentMeIfYouCan).

Siniša Šegvić odlično govori engleski i talijanski i ima osnovne komunikacijske vještine na francuskom. Oženjen je i ima troje djece. Član je IEEE-a.

About the Supervisor

Siniša Šegvić was born in 1971 in Split, Croatia. He completed elementary school and high school in Zadar, Croatia, with one year abroad in Milano, Italy. He received the BS degree in electrical engineering (9 semesters) in 1996 as well as the MS and PhD degrees in 2000 and 2004. He has been employed at UniZg-FER as an assistant professor since 2006.

He was a postdoc researcher at IRISA, Rennes and at TU Graz. He led three research projects of the Croatian Science Foundation (MultiCLOD, MASTIF, ADEPT) as well as several industrial research projects funded by Rimac automobili, RoMB technologies, Microblink etc. He has participated in the research center of excellence DATACROSS, several ERDF projects (SafeTram, MAS, A-UNIT) as well as on one FP7 project (ACROSS).

His research and professional interests include computer vision, visual recognition, scene understanding, and dense prediction with deep convolutional models. He has published several papers at top conferences (3×CVPR, 2×ECCV, NeurIPS) and scientific journals. He has been a reviewer at top conferences as well as in scientific journals in the fields of computer vision, intelligent transportation systems and robotics. He participated in the industrial development as a technical consultant. He advises several PhD students funded by EU projects, national projects and private companies. His research group has achieved notable results while participating at computer vision challenges such as WildDash, Robust vision challenge, Cityscapes, Fishyscapes, and SegmentMeIfYouCan.

Siniša Šegvić speaks English and Italian very well, and has basic communication skills in French. He is married and has three children. He is a member of IEEE.

Zahvala

Hvala profesoru Siniši Šegviću na vođenju, savjetima, korisnim idejama i dobroj suradnji.

Hvala svim trenutnim i bivšim članovima istraživačke grupe na suradnji, ugodnom društvu i dobrom radnom okruženju.

Hvala mojoj obitelji na podršci i brizi.

Sažetak

Polunadzirano učenje je važno za praktičnu primjenu dubokih modela jer ublažava ovisnost o označenim podacima. Ono je posebno zanimljivo u kontekstu guste predikcije, gdje označavanje na razini piksela zahtijeva velik napor. Ovaj rad razmatra oblike polunadziranog učenja koji uz nadzirani gubitak potiču konzistenciju predikcija u perturbacijama ulaznog podatka. Istražili smo formulacije konzistencije s obzirom na smjer konzistencije i perturbiranje ulaza. Oblikovali smo učinkovit algoritam za polunadzirano učenje modela za semantičku segmentaciju. Konačno, predložili smo novi perturbacijski model koji je posebno prikladan za polunadzirano učenje modela za gustu predikciju. Teorijska analiza i empirijski uvidi pokazuju da najbolju generalizaciju postiže jednosmjerna konzistencija s čistim učiteljem. Ovaj oblik polunadziranog konzistencijskog učenja traži se da jedan primjerak modela, učenik, bude konzistentan s drugim, učiteljem, i samo se učenikov ulaz (jako) perturbira. Jednosmjerna konzistencija ima i prednost memorijske učinkovitosti u odnosu na dvosmjernu jer za određivanje gradijenta ne mora pamtiti međurezultate učitelja. Eksperimenti se usredotočuju na učinkovite modele koji su važni za primjene na štedljivom sklopovlju i u stvarnom vremenu. Još neki zaključci istraživanja su: (1) fotometrijske perturbacije su puno korisnije od geometrijskih za semantičku segmentaciju, (2) korištenje pokretnog prosjeka učenikovih parametara u učitelju ima prednost nad jednostavnim dijeljenjem učenikovih parametara samo kad je dostupno vrlo malo označenih podataka, (3) polunadzirano učenje može biti bolja alternativa dodatnim grubim oznakama. Generalizacijska izvedba predložene metode nadmašuje sve prethodne i konkurentne pristupe za polunadziranu semantičku segmentaciju.

Ključne riječi: polunadzirano učenje, semantička segmentacija, konzistencijsko učenje, jednosmjerna konzistencija

Semi-supervised learning of semantic segmentation of images by applying consistency criteria

Supervised learning is one of the most important mechanisms for realizing visual perception for autonomous driving, medical diagnostics, and numerous other applications. However, the scaling of such approaches is hampered by data labeling difficulties. For dense prediction tasks such as semantic segmentation, pixel-level annotation is particularly expensive.

Semi-supervised methods aim to solve the same task as supervised learning by exploiting unlabeled data in addition to labeled data. Among them, consistency training and pseudo-label training (self-training) are especially interesting for dense prediction. This thesis studies consistency training, which combines a supervised loss with a consistency loss that can encourage consistency of predictions over different perturbations of unlabeled inputs.

This thesis investigates two-branch consistency formulations concerning the consistency direction and input perturbations. Theoretical analysis and empirical insights show that one-way consistency with a clean teacher generalizes best. This form of consistency requires one model instance, the student, to be consistent with the other, the teacher, and only the teacher's input is (strongly) perturbed. One-way consistency also has the advantage of memory efficiency compared to two-way consistency because it does not have to store the teacher's intermediate results for gradient computation.

We present an efficient algorithm for semi-supervised learning for semantic segmentation. We also propose a new perturbation model that is suitable for semi-supervised learning of dense prediction models. Experiments evaluate the proposed method and validate components thereof on standard benchmarks, and mostly focus on efficient models that are important for inexpensive-hardware and real-time applications. The source code for the experiments is available at `https://github.com/Ivan1248/semisup-seg-efficient`.

The following text summarizes the chapters of this thesis.

Introduction

Machine learning models for complex computer vision tasks require large data sets. The standard approach for such tasks is supervised learning. However, for tasks such as dense prediction tasks, annotation can be especially demanding, while unlabeled, weakly labeled, and other kinds of data that also contain useful information are more readily available. Therefore, different approaches can be interesting depending on what additional data is available. Some methods may take advantage of incomplete (weak) labels. Supervised and unsupervised (self-supervised) pretraining can also use data that are dif-

ferent, but suitable for tasks that require solving similar subtasks. Domain adaptation mehods have aim to exploit data from a distribution that differs in some properties from the distribution of the test set. Semi-supervised learning can exploit unlabeled data. This paper considers semi-supervised learning for semantic segmentation. Pixel-level labels for semantic segmentation are expensive, and semi-supervised learning can make use of additional unlabeled images.

Dense prediction models need to process high-resolution features to recognize small objects. For the best generalization performance, learning with large mini-batches and large crops of input images is desirable. Hence, training approaches that do not have significant additional memory requirements are particularly interesting. However, some semi-supervised algorithms introduce additional components that require additional memory. Approaches based on adversarial models require an additional generator or discriminator. Some approaches use multiple similar model instances or accumulate predictions over the entire training set. Pseudo-label training (self-training) and some consistency training approaches have similar memory requirements to supervised training, and are more suitable for dense prediction in this regard.

This thesis studies semi-supervised consistency training. In addition to the standard supervised criterion, these approaches also require consistency of predictions over different perturbations of unlabeled images. Such algorithms usually benefit more from stronger perturbations than supervised learning. This might be influenced by the fact that consistency learning requires the perturbations to be strong enough to make the predictions sufficiently different for learning from unlabeled data. The disadvantage of strong perturbations is losing useful features and learning features that are useful for robustness to perturbations, but not for generalization on unperturbed data. In semi-supervised learning, additional information from unlabeled data can compensate for such losses, especially when the labeled training set is smaller.

In consistency training, not all model instances (branches) that participate in the consistency loss have to receive a perturbed input. We will refer to an input as being clean if it is perturbed by weak perturbations like those used in standard supervised training, while we will refer to it as being perturbed if it is additionally perturbed by stronger perturbations. We will refer to model instances to as a perturbed or clean branch depending on whether the input is (strongly) perturbed or clean.

One-way consistency is particularly interesting for dense prediction because it can be trained efficiently since the gradient is computed in only one of two model instances. In other words, the parameters are optimized as if the output of one branch is independent of them. We call this branch the teacher, while we call the other branch the student. The main approach considered in this work is one-way consistency with a clean teacher, where

only the student's input is perturbed.

Depending on the task and the perturbations, a fairly general consistency objective is equiariance to perturbations. For such input transformations T, there are corresponding output transformations T' which, when applied to the correct prediction in the clean example, yield the correct prediction in the transformed input. Examples of such transformations are geometric transformations such as translation and rotation in the case of semantic segmentation. A special case equivariance is invariance, when $T' = \mathrm{id}$. Let $\boldsymbol{x}$ be an input from the unlabeled training set, $h_{\boldsymbol{\theta}}$ the student, and $h_{\mathrm{sg}(\boldsymbol{\theta})}$ the teacher. Furthermore, the teacher parameters $\mathrm{sg}(\boldsymbol{\theta})$ are a frozen copy of the student parameters $\boldsymbol{\theta}$. Then the loss of one-way consistency with a clean teacher can be expressed by a divergence D between the student prediction and the transformed teacher prediction:

$$L^{\mathrm{ct}}_{\boldsymbol{\theta}}(\boldsymbol{x}, T, T') = D(T'(h_{\mathrm{sg}(\boldsymbol{\theta})}(\boldsymbol{x})), h_{\boldsymbol{\theta}}(T(\boldsymbol{x}))) \,. \tag{1}$$

We illustrate the results and investigate the properties of the most interesting variants of consistency training on low-dimensional classification problems. One-way consistency outperforms two-way consistency and both outperform supervised training, while one-way consistency with a clean student underperforms supervised training. Such a result is influenced by the fact that predictions in clean inputs are on average more correct than predictions in perturbed inputs. Another interesting observation is that one-way consistency training has different dynamics and converges to different results than two-way consistency. Both kinds of one-way consistency encourage the predictions of the student to be similar to the predictions of the teacher, but not the other way around.

In one-way consistency with a clean teacher, the predictions in the perturbed examples, the neighborhood of the clean example, are pulled towards the prediction in the clean example. The predictions spread from clean examples into their neighborhoods (including other clean examples), but parts of the neighborhoods outside of clusters do not influence the predictions in clean examples. This results in the same prediction spreading from labeled examples across their clusters and pushing decision boundaries outside of the clusters.

In one-way consistency with a clean student, the prediction in the clean example is pulled towards the predictions in its neighborhood. Here, only neighborhoods influence predictions in clean examples and wrong predictions can be pulled into the cluster from outside of it. It is also interesting that, in this case, predictions can converge even if not consistent within neighborhoods: a part of the neighborhood can pull the prediction in the clean example in one direction while another part of the neighborhood pulls the prediction in the other direction. This results in the prediction varying within clusters.

In the case of two-way consistency, the consistency loss is lower, but intra-cluster consistency is lower than in the case of one-way consistency with a clean teacher due to the influence of the environment outside of the cluster.

Fundamental concepts

This chapter presents a concise overview of some basic concepts that are useful for understanding and devising machine learning algorithms. It contains sections on probability theory, information theory, statistical inference, machine learning, and deep learning. The first section is about basic concepts from probability theory, which is important for machine learning because it describes ideal inference under incomplete knowledge (uncertainty). The information theory section defines terms such as information content, entropy, and relative entropy (KL divergence), which are important when evaluating predictions and comparing probability distributions. The section on statistical inference explains the basic concepts of inferring distributions from a finite set of observations, and explains simplifications and approximations used in practice. The next section covers the basic concepts of machine learning and briefly introduces forms of machine learning such as supervised, unsupervised, self-supervised, and semi-supervised learning. Finally, the deep learning section describes the basic components of deep models, the problems in learning deep models, and the algorithmic components that are generally useful for solving them. This section also includes a more detailed description of optimization procedures, other heuristics, and components of convolutional models for computer vision.

Literature overview

This chapter presents an overview of the literature in the narrower field of research. It is divided into 3 sections: dense prediction, semi-supervised learning and semi-supervised learning of semantic segmentation. The section on dense prediction discusses the main problems encountered in the development of dense prediction algorithms, listing the main solutions from the literature. Furthermore, it describes supervised learning of semantic segmentation, which is applied and extended to semi-supervised learning in this work. The section about semi-supervised learning presents the basic assumptions of semi supervised learning, and categorizes and describes previous algorithms with an emphasis on consistency training. The last section describes semi-supervised learning of semantic segmentation and particularly highlights the shortcomings that the proposed method overcomes and differences to similar approaches.

One-way consistency with a clean teacher

This chapter describes the proposed algorithm for semi-supervised semantic segmentation based on one-way consistency with a clean teacher and photometric and geometric perturbations. The proposed algorithm uses the mean divergence between the corresponding prediction pixels in the clean and perturbed image as a consistency loss for dense prediction. Images are perturbed by a composition of random photometric and geometric transformations. The consistency loss is minimal if the prediction is invariant to photometric transformations and equivariant to geometric transformations.

The proposed perturbation model is a composition of a photometric transformation $T^{\mathrm{P}}_{\boldsymbol{\varphi}}$ and a geometric transformation $T^{\mathrm{G}}_{\boldsymbol{\gamma}}$. We assume that geometric transformations should be applied equally to images and predictions, and that the photometric perturbation in the output space (in predictions and labels) is the identity. Thus, we can express the perturbation in image space as $T_{\boldsymbol{\tau}} = T^{\mathrm{G}}_{\boldsymbol{\gamma}} \circ T^{\mathrm{P}}_{\boldsymbol{\varphi}}$, and express the corresponding perturbation in the output space as $T'_{\boldsymbol{\tau}} = T^{\mathrm{G}}_{\boldsymbol{\gamma}}$.

The photometric perturbation is a composition of 5 simpler transformations with 5 parameters. It changes the brightness, saturation, and hue in the HSV space, changes the contrast by multiplying all the channels in the RGB space, and randomly permutes the channels. The geometric perturbation is based on the thin plate spline (TPS) warp. In experiments, we use 4 control points that correspond to the centers of the four quadrants of the image. The displacements of the control points are randomly sampled from a normal distribution. The polyharmonic displacement interpolation gives a smooth parametric distortion that we use as a perturbation.

We adapt one-way consistency with a clean teacher for the task of dense prediction with the proposed perturbation model. The consistency loss in an input example is calculated as follows. One branch perturbs the image and computes the student prediction is computed. The other branch computes the teacher prediction in the clean input and perturbs it with the corresponding geometric output transformation. Intermediate results in the teacher branch do not have to be stored because the gradient should only be computed in the student. After that, the consistency loss is computed as the mean relative entropy over valid pixels that the geometric transformation samples from within the image coordinates.

In simple one-way consistency, the teacher's parameters are a frozen copy of the student's parameters: $\mathrm{sg}(\boldsymbol{\theta})$. In the case of Mean Teacher, the parameters are an exponential moving average of the student's parameters. In simple two-way consistency, both branches use the same parameters $\boldsymbol{\theta}$, and the gradient is computed in both branches.

A general consistency training error function can be expressed as a weighted sum of the average supervised loss and the average expected (per perturbations) consistency loss.

In our experiments, the supervised loss is the usual mean cross-entropy loss per pixel with L^2 regularization. We estimate the expectation over perturbation parameters $\underline{\boldsymbol{\tau}}$ with a single sample per input example per training step.

Memory measurements of our implementation based on PyTorch with the model SwiftNet-RN34 with 768×768 crops, 8 labeled and 8 unlabeled examples per batch, and 19 classes show that one-way consistency has a slightly greater memory footprint (6.86 GiB) than supervised training (5.46 GiB) and almost half the memory footprint of two-way consistency (11.77 GiB).

Experiments

We perform semantic segmentation experiments on Cityscapes and PASCAL VOC 2012 with the augmented training subset, and image classification experiments on CIFAR-10. In most cases, we run training 5 times with different subsets of labels, and measure the generalization performance on the corresponding validation set. We use label subset sizes 371, 743, 1487 and 2975 (all labels) on Cityscapes, 662, 1323, 2645 and 10582 (all labels) on PASCAL VOC 2012, and 250, 1000, 4000 and 50000 (all labels) on CIFAR-10.

The first experiments compare different algorithms and algorithmic components, including previous and concurrent work, on half-resolution Cityscapes. This experiments include the standard model DeepLabv-RN101 and the efficient model SwiftNet-RN18. The two models behave similarly under the same semi-supervised algorithm, and SwiftNet-RN18, which is about $4\times$ to $12\times$ faster, has a slightly lower generalization performance. Our perturbations, PhTPS outperforms CutMix perturbations, and, under PhTPS, Mean Teacher outperforms simple one-way consistency when 743 or less labels are available.

Next, we perform ablation studies. Experiments with applying PhTPS and its components as additional data augmentation for supervised training show that supervised training only benefits from the photometric (Ph) component of PhTPS, while the geometric component (TPS) underperforms the baseline. However, both perturbation components are useful for consistency training: Ph is most useful, while TPS is useful in addition to TPS only in the case of 371 labels, and neutral otherwise. We also observe that supervised learning with photometric perturbations performs similarly to consistency training when 743 or more labels are used. Additional experiments show that consistency training does not outperform supervised learning without the unlabeled training subset unless more than 1487 labels are available. Experiments that validate the consistency loss weight show that weights $\alpha \in [0.25..1]$ perform similarly.

The next experiments are on full-resolution Cityscapes. Here, Mean Teacher (MT) outperforms simple one-way consistency only when 743 or fewer labels are used. Here, we had to use SwiftNet-RN34 to outperform a concurrent method that uses DeepLabV3$^+$-

RN50. SwiftNet-RN34 is still about 4× faster and our training has an about 3× smaller memory footprint. We also perform experiments with an additional large coarsely labeled Cityscapes subset ("train-extra"). In the case of SwiftNet-RN18, Mean Teacher (MT) does not benefit from the additional data without labels, while simple one-way consistency does come close to supervised training with the additional coarse labels. In the case of SwiftNet-RN34, both semi-supervised algorithms without the additional coarse labels outperform supervised training.

In PASCAL VOC experiments, one-way consistency with a clean teacher achieves the best generalization performance, while MT underperforms the supervised baseline. This was influenced by the fact that we used the same hyperparameters as for Cityscapes.

Another set of experiments validates consistency variants on Cityscapes semantic segmentation and CIFAR-10 classification. One-way consistency with a clean teacher outperformed other forms of consistency. Two-way consistency with one perturbed input is between two forms of one-way consistency. This favors the hypothesis that predictions in clean examples are better consistency targets. One-way consistency with both inputs perturbed in many cases outperformed one-way consistency with a clean student, but always underperformed one-way consistency with a clean teacher. A closer look suggests that the algorithm perturbing both inputs results in cheating the consistency loss by predicting similar outputs for the perturbed images. Similar cheating occurred in the case of the clean teacher on Cityscapes, but together with severe overfitting.

The final set of experiments quantifies the effect of disabling the updating of batch normalization statistics when the student receives perturbed inputs in one-way consistency with a clean teacher. In the case of the half-resolution Cityscapes and PASCAL VOC, disabling updating in the perturbed student increased the validation mIoU by between 0.3 pp and 1.4 pp, depending on the fraction of labels used. However, in the case of Cityscapes at full resolution, the opposite happened – mIoU dropped by between 0.1 pp and 1.1 pp.

Conclusion

The main contributions of the research relate to the exhaustive characterization and validation of forms of semi-supervised consistency training, the design of an effective training method, and the design of a perturbation model for semantic segmentation. The thesis shows that one-way consistency with a clean teacher generalizes better than other forms of consistency with respect to the selection of the branch in which perturbations are applied and where the gradient is computed. Better generalization performance with a clean teacher is influenced by the fact that predictions in clean images are more often less accurate than predictions in perturbed images. The disadvantage of perturbing both branches is

a greater tendency to learn trivial solutions that map all perturbed inputs to the same output. One-way consistency also has the advantage that a careful implementation can have memory requirements similar to supervised learning.

The paper proposes an efficient algorithm for semi-supervised learning of semantic segmentation based on one-way consistency with a clean teacher and the proposed perturbation model, which is a composition of photometric and geometric transformations. The consistency loss encourages the model instance to be invariant to photometric perturbations and equivariant to geometric perturbations. The proposed algorithm outperforms all previous and concurrent methods: in most semantic segmentation experiments. we noted improved generalization performance with varying amounts of labeled and unlabeled training data, for the standard models DeepLabv2-RN101 and DeepLabv3$^+$-RN50, and for the efficient models SwiftNet-RN18 and SwiftNet-RN34. The efficient model SwiftNet-RN18 generalizes similarly to DeepLabv2-RN101, while its inference is about $9\times$ faster at half resolution and about $15\times$ faster at full resolution on the RTX 2080Ti GPU. Semi-supervised learning with efficient models can be useful for algorithm development and real-time practical applications when large labeled sets are not available. Simplicity, competitive performance and computational efficiency make this approach an interesting basis for evaluating new semi-supervised procedures for dense prediction.

The paper also compares two kinds of teachers for one-way consistency. In simple consistency, the teacher has the same parameters as the student, while in Mean Teacher, the teacher parameters are a moving average of the students. Experiments on Cityscapes suggest that the averaged teacher generalizes better than simple consistency when the data is at a lower resolution or when the number of labels is smaller. However, when more data and more labels are used, simple consistency still outperforms supervised learning, while Mean Teacher generalizes worse in some cases. Experiments with an additional coarsely labeled Cityscapes subset show that semi-supervised learning can be a better alternative to adding a large number of coarse labels to supervised learning.

There are several directions for future work that can ensue from this research. It would be interesting to better understand the dynamics of optimization with one-way consistency. In addition, it is not clear what causes the differences in the behavior of simple consistency and Mean Teacher in different conditions. Finally, it could be useful to investigate consistency training that is more resistant to incorrect teacher predictions in early training iterations.

Keywords: semi-supervised learning, semantic segmentation, consistency training, one-way consistency

Sadržaj

Poglavlje 1

Uvod

Strojno učenje modela za složene zadatke računalnog vida zahtijeva velike podatkovne skupove. Standardni pristup kod takvih zadataka je nadzirano učenje. Međutim, za neke zadatke, kao što su zadaci guste predikcije, izrada oznaka može biti zahtjevna, dok su lakše dostupni neoznačeni, slabo označeni i drugi podaci koji isto sadrže korisne informacije. Zato mogu biti zanimljivi različiti pristupi ovisno o tome kakvi dodatni podaci su dostupni. Polunadzirano učenje može iskoristiti neoznačene podatke [1]. Neki postupci mogu iskoristiti nepotpune (slabe) oznake [2]. Nadzirano i nenadzirano (samonadzirano) predtreniranje mogu iskoristiti i podatke koji su drugačiji, ali su prikladni za zadatke koji zahtijevaju rješavanje sličnih podzadataka [3]. Predloženi su i različiti postupci prilagodbe na pomak domene, kojima je cilj iskoristiti označene ili neoznačene podatke čija se razdioba po nekim svojstvima razlikuje od razdiobe skupa za testiranje [4].

Ovaj rad bavi se polunadziranim učenjem [5, 6, 7] za zadatak semantičke segmentacije [8, 9, 10]. Oznake na razini piksela za semantičku segmentaciju su skupe, a polunadzirano učenje omogućuje iskorištavanje dodatnih neoznačenih slika do kojih je lako doći.

Modeli za gustu predikciju obrađuju slike na visokim rezolucijama kako bi mogli raspoznavati male objekte. Za najbolju generalizacijsku performansu poželjno je učenje s velikim mini-grupama i velikim isječcima ulaznih slika [11, 12, 13]. Zbog konačne količine memorije na grafičkoj procesnoj jedinici [14], posebno su zanimljivi pristupi koji nemaju značajne dodatne memorijske zahtjeve. Međutim, mnogi polunadzirani algoritmi iz literature uvode dodatne komponente koje u slučaju guste predikcije zahtijevaju dodatnu memoriju. Pristupi temeljeni na suparničkim modelima zahtijevaju dodatni generator [8, 15] ili diskriminator [9, 16, 17]. Neki pristupi koriste više sličnih primjeraka modela [18, 19, 20, 21] ili akumuliraju predikcije preko cijelog skupa za učenje [22]. Takvi su pristupi zbog velikih memorijskih zahtjeva često manje prikladni za gustu predikciju.

1.1 Konzistencijsko učenje i jednosmjerna konzistencija

Ovaj rad proučava polunadzirane algoritme **konzistencijskog učenja** [5, 6, 7, 19, 22, 23]. Pored standardnog nadziranog kriterija, ovi pristupi zahtijevaju i konzistenciju predikcija preko različitih perturbacija neoznačenih slika. Takvi algoritmi obično imaju više koristi od jačih perturbacija nego nadzirano učenje. Na to utječe to što konzistencijsko učenje zahtijeva da perturbacije budu dovoljno jake kako bi se se predikcije dovoljno razlikovale za učenje iz neoznačenih podataka. Nedostatak jakih perturbacija je gubljenje korisnih značajki i učenje značajki koje su korisne za **robusnost** na perturbacije, ali nisu za generalizaciju na neperturbiranim podacima [24]. Kod polunadziranog učenja dodatne informacije iz neoznačenih podataka mogu kompenzirati takve gubitke, pogotovo kad je označeni skup za učenje manji.

U konzistencijskom učenju ne moraju biti perturbirani ulazi svih primjeraka modela koji sudjeluju u konzistenciji. Govorit ćemo kraće da je podatak **čist** ako je perturbiran osnovnim perturbacijama (rastresanjem) koje se koriste kod nadziranog učenja, dok ćemo kraće reći da je **perturbiran** ako je dodatno perturbiran jačim perturbacijama (jako perturbiran). Primjerke modela kraće ćemo nazivati **perturbiranom** odnosno **čistom granom**, ovisno o tome je li ulaz (jako) perturbiran ili čist.

Za gustu predikciju je posebno zanimljivo što se konzistencija može učiti učinkovito tako da se gradijent računa samo u jednom od dvaju primjeraka modela. Drugim riječima, parametri se mogu optimirati kao da je izlaz jednog primjerka modela neovisan o njima i to nazivamo **jednosmjernom konzistencijom**. Kod jednosmjerne konzistencije gradijent se računa samo u jednoj grani. Tu granu nazivamo **učenikom**, a drugu granu **učiteljem**. Posebno je zanimljiva **jednosmjernom konzistencijom s čistim učiteljem**, kod koje samo učenik dobiva perturbiranu inačicu ulaza [5, 6, 7]. Ona je glavni pristup koji ovaj rad razmatra.

Ovisno o zadatku i perturbacijama, najjednostavniji konzistencijski cilj je invarijantnost primjerka modela na perturbacije ulaza. Neka je $\boldsymbol{x}$ ulazni primjer iz neoznačenog skupa za učenje, T perturbacija ulaza na koju bi idealni model trebao biti invarijantan, $h_{\boldsymbol{\theta}}$ učenik i $h_{\mathrm{sg}(\boldsymbol{\theta})}$ učitelj. Nadalje, parametri učitelja $\mathrm{sg}(\boldsymbol{\theta})$ su smrznuta kopija parametara učenika $\boldsymbol{\theta}$. Tada gubitak jednosmjerne konzistencije s čistim učiteljem možemo izraziti divergencijom D između predikcija učenika i učitelja:

$$L_{\boldsymbol{\theta}}^{\mathrm{ct}}(\boldsymbol{x}, T) = D(h_{\mathrm{sg}(\boldsymbol{\theta})}(\boldsymbol{x}), h_{\boldsymbol{\theta}}(T(\boldsymbol{x}))) \,. \tag{1.1}$$

Međutim, postoji i općenitija klasa transformacija s obzirom na koje bi idealni model h^* trebao biti **ekvivarijantan** [25]. Za takve transformacije ulaza postoje odgovarajuće transformacije izlaza T' koje, kad se primijene na točnu predikciju u čistom primjeru, daju

točnu predikciju u transformiranom ulazu. Primjeri takvih transformacija su geometrijske transformacije kao translacija i rotacija u slučaju zadatka semantičke segmentacije, gdje se izlaz moraju izmijeniti kao i ulaz. Ova rasprava vodi nas do općenitije formulacije konzistencijskog gubitka koji uključuje i perturbaciju ulaza T i odgovarajuću transformaciju izlaza T':

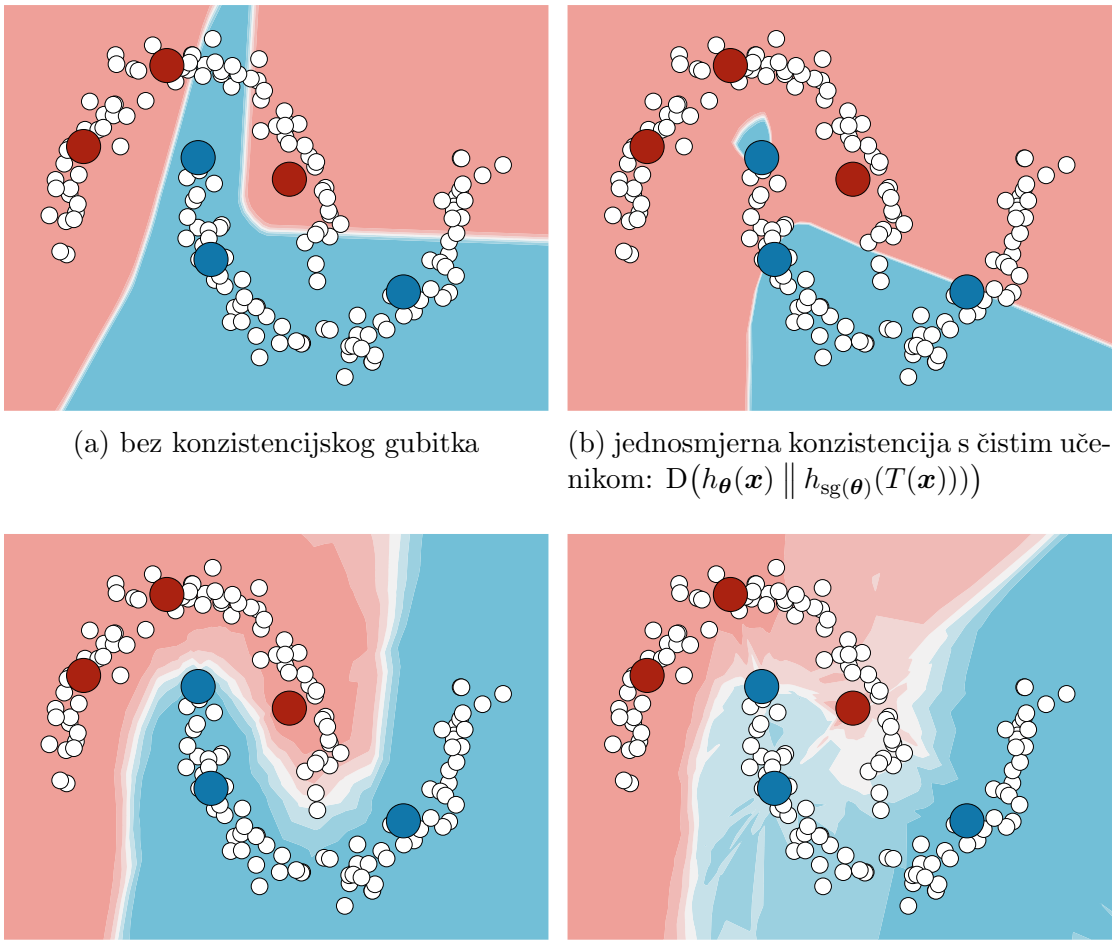
$$L_{\boldsymbol{\theta}}^{\mathrm{ct}}(\boldsymbol{x}, T, T') = D(T'(h_{\mathrm{sg}(\boldsymbol{\theta})}(\boldsymbol{x})), h_{\boldsymbol{\theta}}(T(\boldsymbol{x}))) \,. \tag{1.2}$$

1.2 Svojstva varijanti konzistencije na niskodimenzionalnim zadacima

Tvrdimo da je čisti učitelj najbolji odabir, posebno kad su perturbacije jake. Slika 1.1 ilustrira 4 oblika konzistencijskog učenja na dvodimenzionalnom klasifikacijskom zadatku "Dva mjeseca". Obojane točke su primjeri iz dvaju razreda, a bijele točke su neoznačene. U svim eksperimentima za konzistencijski gubitak koristimo relativnu entropiju (KL divergenciju).

Nadzirano učenje (a) vrlo slabo generalizira jer koristi samo označene podatke. Jednosmjerna konzistencija potiče pomicanje učenikove predikcije prema učiteljevoj, ali ne učiteljeve predikcije prema učenikovoj. Ako je jedna grana perturbirana, moguća su dva oblika jednosmjerne konzistencije. Jedan mogući oblik je jednosmjerna konzistencija s čistim učenikom (b), koja zahtijeva da se neoznačeni primjeri klasificiraju što sličnije svojoj okolini (perturbiranim inačicama). Jedan od nedostataka jednosmjerne konzistencije s čistim učiteljem je da su predikcije u jače perturbiranim primjerima manje pouzdane od predikcija u čistim primjerima. Osim toga, ako perturbirani primjeri koji padaju izvan područja u kojem podaci imaju visoku gustoću mogu konzistentno povlačiti predikciju u čistom primjeru prema krivoj predikciji. Učinak širenja krivih predikcija iz okoline prema unutrašnjosti područja gdje su podaci gusti najbolje se vidi u središtu lijevog kraka plavog polumjeseca, gdje su brojne bijele točke klasificirane u okolni crveni razred. Tog problema nema kod jednosmjerne konzistencije s čistim učiteljem (c). Zbog toga označeni i neoznačeni primjeri šire svoju predikciju u okolinu sve dok granica odlučivanja ne završi u području niske gustoće između polumjeseca. Dvosmjerna konzistencija [22] s jednim čistim ulazom (d) djeluje kao zbroj ovih dviju jednosmjernih konzistencija. Bolje generalizira od čistog učenika, ali lošije generalizira i sporije konvergira od čistog učitelja zbog utjecaja komponente koja odgovara čistom učeniku (perturbiranom učitelju).

Možemo pažljivije razmotriti ponašanje jednosmjerne konzistencije u usporedbi s dvosmjernom. Za ilustraciju definiramo jednostavan model koji binarno klasificira jednodimenzionalne ulaze i ima skup parametara $\Theta = \{(\xi_k, \theta_k)\}_{k=1}^{|\Theta|}$, gdje su θ_k parametri koji lokalno utječu na predikciju u okolini fiksnih točaka ξ_k. Predikcija je težinski zbroj Gaussovih



(a) bez konzistencijskog gubitka

(b) jednosmjerna konzistencija s čistim učenikom: $\mathrm{D}\big(h_{\boldsymbol{\theta}}(\boldsymbol{x}) \,\big\|\, h_{\mathrm{sg}(\boldsymbol{\theta})}(T(\boldsymbol{x}))\big)$

(c) jednosmjerna konzistencija s čistim učiteljem: $\mathrm{D}\big(h_{\mathrm{sg}(\boldsymbol{\theta})}(\boldsymbol{x}) \,\big\|\, h_{\boldsymbol{\theta}}(T(\boldsymbol{x}))\big)$

(d) dvosmjerna konzistencija s jednim čistim ulazom: $\mathrm{D}(h_{\boldsymbol{\theta}}(\boldsymbol{x}) \,\|\, h_{\boldsymbol{\theta}}(T(\boldsymbol{x})))$

Slika 1.1: Nadzirano i polunadzirano učenje binarne klasifikacije 2D točaka u crveni i plavi razred. Skup za učenje sastoji se od 6 označenih (crveno i plavo) i velikog broja neoznačenih točaka (bijelo). U prikazanim slučajevima učenje uglavnom konvergira nakon 20000 epoha, a optimizacijski algoritam je Adam s pretpostavljenim hiperparametrima. Nadzirana funkcija gubitka je negativna log-izglednost, a konzistencijska funkcija gubitka je relativna entropija, s obzirom na predikciju u čistom podatku. Slike prikazuju nadzirano učenje s isključenom konzistencijom (a) i oblike konzistencijskog učenja (b-d). Jednosmjerna konzistencija s čistim učiteljem nadmašuje ostale oblike konzistencije.

radijalnih baznih funkcija:

$$\mathrm{p}(y=1 \mid x,\Theta) = \frac{1}{|\Theta|} \sum_{(\xi_k,\theta_k)\in\Theta} \sigma(\theta_k) \frac{1}{\sqrt{2\pi}} \exp((\xi_i - x)^2). \tag{1.3}$$

Logistička sigmoida, σ, služi za preslikavanje parametara u raspon $(0..1)$.

Parametri modela u eksperimentima su $\Theta = \{(-10+1.5k, \theta_k)\}_{i=1}^{20}$. Parametri θ_k uče se minimizacijom zbroja nadziranog gubitka negativne log-izglednosti i konzistencijskog gubitka. Za konzistencijski gubitak koristimo Jensen-Shannonovu (JS) divergenciju radi

simetrije:

$$\mathrm{D}_{\mathrm{JS}}(p,q) \coloneqq \frac{1}{2}\mathrm{D}\left(p \,\middle\|\, \frac{p+q}{2}\right) + \frac{1}{2}\mathrm{D}\left(q \,\middle\|\, \frac{p+q}{2}\right). \tag{1.4}$$

U ovom slučaju slične rezultate daje i relativna entropija U svakoj iteraciji koriste se svi primjeri za učenje i konzistencijski gubitak se uprosječuje preko 50 perturbacija uzorkovanih iz okoline $\mathrm{U}([x-1, x+1])$ za svaki primjer x. Optimizacijski algoritam je gradijentni spust u smjeru predznaka gradijenta (Adam s $\beta_1 = 0$ i $\beta_2 = 0$) s korakom 0.02.

Eksperimenti s takvim modelom su ilustrirani na slici 1.2. Oni upućuju na to da jednosmjerna konzistencija (s čistim učiteljem ili učenikom) čini dinamički sustav koji ima drugačije atraktore od lokalnih minimuma gubitka (lokalni minimumi su otprilike atraktori optimizacije s pravim gradijentom). Razlike se najbolje vide u evoluciji predikcija oko neoznačenih primjera koji su blizu granice odlučivanja ili na rubovima nakupina primjera. Kod jednosmjerne konzistencija s čistim učenikom (i perturbiranim učiteljem) možemo primijetiti sljedeće pojave. Ažuriranje parametara prema gradijentu u učeniku ne povećava nužno konzistenciju u okolini čistog primjera. U slučaju perturbiranog učitelja, učitelj su predikcije u okolini čistog primjera. Ako dio okoline povlači predikciju u čistom primjeru u jednom smjeru, a drugi dio okoline ju povlači jednako u drugom smjeru, gradijent s obzirom na učenikove parametre je $\mathbf{0}$ iako je konzistencijski gubitak visok. Takva optimizacija može biti sklona povlačenju granice odlučivanja prema području veće gustoće. Ako je neoznačeni primjer na rubu nakupine primjera, dio okoline koji je izvan nakupine može imati različitiju predikciju ili po volumenu biti veći od dijela okoline koji je unutar nakupine, pa zbog toga više povlačiti predikciju tog primjera nego unutrašnjost nakupine. S druge strane, jednosmjerna konzistencija s čistim učiteljem granicu odlučivanja gura prema van i kao rezultat daje najveću konzistenciju unutar nakupina (među bliskim čistim primjerima). U slučaju dvosmjerne konzistencije, konzistencija unutar nakupina je manja nego u slučaju jednosmjerne konzistencija s čistim učiteljem zbog utjecaja okoline izvan nakupine na primjere na rubovima.

1.3 Pregled rada

Ovaj rad konsolidira rezultate autorovog istraživanja u području polunadzirane semantičke segmentacije [26, 27]. Rad predstavlja komparativnu analizu konzistencijskih pristupa pod pretpostavkom korištenja dvaju primjeraka modela. Predlažemo učinkovit polunadzirani algoritam utemeljen na jednosmjernoj konzistenciji s čistim učiteljem [6, 7, 28] s fotometrijskim i geometrijskim perturbacijama za semantičku segmentaciju. Eksperimenti ispituju svojstva različitih oblika konzistencijskih algoritama na zadacima semantičke seg-

mentacije i klasifikacije slika. Jednosmjerna konzistencija s čistim učiteljem najbolje generalizira i, za razliku od dvosmjerne konzistencije [21, 22], ne zahtijeva puno više memorije od nadziranog učenja. Međurezultati učitelja se ne moraju čuvati za računanje gradijenta, pa ukupni memorijski otisak ne ograničava kapacitet modela s obzirom na standardni nadzirani slučaj [14, 29].

Eksperimenti sa standardnom konvolucijskom arhitekturom DeepLabV2-RN101 [30] pokazuju da fotometrijske i geometrijske perturbacije postižu kompetitivnu generalizacijsku performansu i nadmašuju lijepljenje okana iz drugih slika [28]. Validacijski eksperimenti pokazuju da su fotometrijske perturbacije učinkovitije od geometrijskih. Nadalje, učinkovita arhitektura SwiftNet-RN18 [31] postiže sličnu polunadziranu generalizaciju kao i DeepLabV2 uz 9 do 15 puta brže zaključivanje i 2 do 5 puta brže učenje, ovisno o sklopovlju. Ovaj rezultat je značajan za praktične robotičke primjene koje karakteriziraju maleni skupovi označenih podataka te restriktivan računski budžet u fazi zaključivanja.

Glavni istraživački doprinosi ovog rada su:

- Vrednovanje generalizacijske performanse i učinkovitosti različitih konzistencijskih pristupa polunadziranog učenja.
- Učinkovit algoritam za polunadzirano učenje semantičke segmentacije primjenom optimalnog konzistencijskog kriterija s obzirom na smjer konzistencije i perturbiranje ulaza.
- Novi perturbacijski model utemeljen na kompoziciji geometrijskog izobličenja i fotometrijskog rastresanja.

Izvorni kod za eksperimente je dostupan na `https://github.com/Ivan1248/semisup-seg-efficient`.

Slijedi opis strukture rada. Poglavlje 2 predstavlja sažet pregled osnovnih pojmova koji su korisni za razumijevanje i razvijanje algoritama strojnog učenja. Započinjemo s osnovnim pojmovima iz teorije vjerojatnosti, teorije informacije i statističkog zaključivanja. U nastavku obrađujemo osnovne pojmove strojnog učenja i kratko predstavljamo pristupe strojnog učenja kao što su nadzirano, nenadzirano i polunadzirano učenje. Na kraju predstavljamo osnove dubokog učenja, probleme kod učenja dubokih modela i algoritamske komponente koje su općenito korisne za njihovo rješavanje. Ovaj odjeljak uključuje i detaljniji opis optimizacijskih postupaka, drugih heuristika i elemenata konvolucijskih modela za računalni vid.

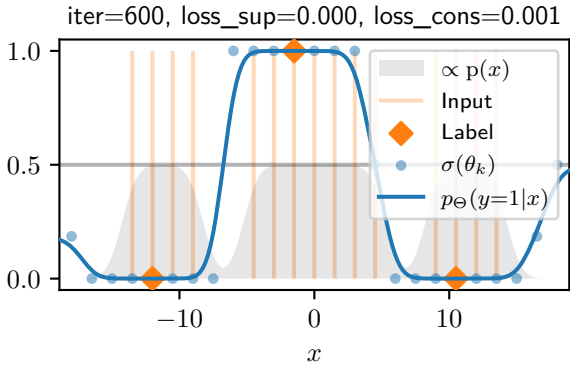
Poglavlje 3 daje pregled literature u užem području istraživanja. Podijeljeno je na 3 odjeljka: gusta predikcija, polunadzirano učenje i polunadzirano učenje semantičke segmentacije. Odjeljak o gustoj predikciji govori o glavnim problemima s kojima se susrećemo pri razvoju algoritama guste predikcije, navodi glavna rješenja iz literature. Nadalje, opisuje nadzirano učenje semantičke segmentacije, kakvo se primjenjuje i proširuje na po-

lunadzirano učenje u ovom radu. Odjeljak o polunadziranom učenju predstavlja osnovne pretpostavke te kategorizira i opisuje prethodne algoritme uz naglasak na konzistencijsko učenje. Posljednji odjeljak opisuje polunadzirano učenje semantičke segmentacije i posebno naglašava nedostatke koje predložena metoda otklanja.

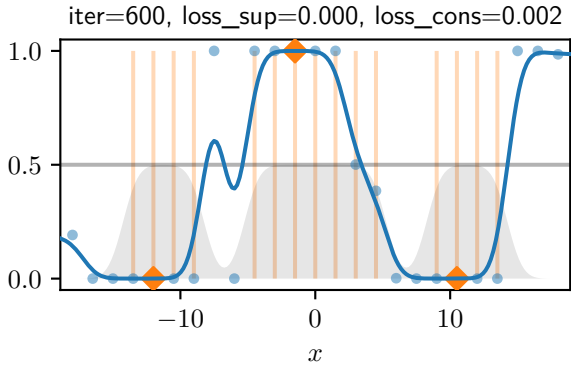
Poglavlje 4 detaljno opisuje predloženu metodu. Prvo su opisane fotometrijska i geometrijska komponenta predloženog perturbacijskog modela i navedeni korišteni hiperparametri. Iza toga slijedi opis predloženog algoritma učenja s jednosmjernom konzistencijom s čistim učiteljem za semantičku segmentaciju. Kraj poglavlja donosi algoritamski postupak memorijski učinkovitog učenja jednosmjerne konzistencije uz grafički prikaz memorijskog otiska u ovisnosti o liniji pseudokoda.

Poglavlje 5 opisuje eksperimente. Na početku opisani su skupovi podataka i eksperimentalne postavki kao što su hiperparametri algoritama i postavke evaluacije. Slijedi prikaz rezultata eksperimenata semantičke segmentacije na podatkovnom skupu Cityscapes na pola rezolucije. Prikazujemo usporedbe algoritama i algoritamskih komponenata (što uključuje usporedbu s literaturom), ablaciju perturbacijskog modela i validaciju hiperparametara. Nakon toga dolaze segmentacijski eksperimentu na Cityscapesu na punoj rezoluciji. Prikazujemo usporedbe algoritama i algoritamskih komponenata, eksperimente s velikim dodatnim skupom podataka i usporedbu s nadziranim učenjem na grubim oznakama. Nakon toga dolaze klasifikacijski eksperimenti na skupu CIFAR-10 i validacija oblika konzistencije na semantičkoj segmentaciji i klasifikaciji slika. Na kraju razmatramo utjecaj ažuriranja statistika normalizacije po grupi u perturbiranom učeniku na generalizacijsku performansu.

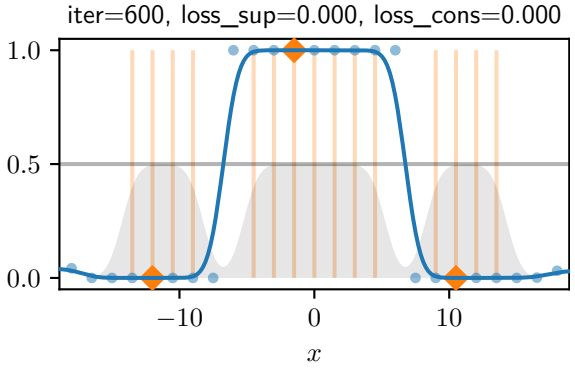
Na kraju rada poglavlje 6 donosi kratak pregled ostvarenih rezultata te predstavlja prikladne pravce za budući rad.



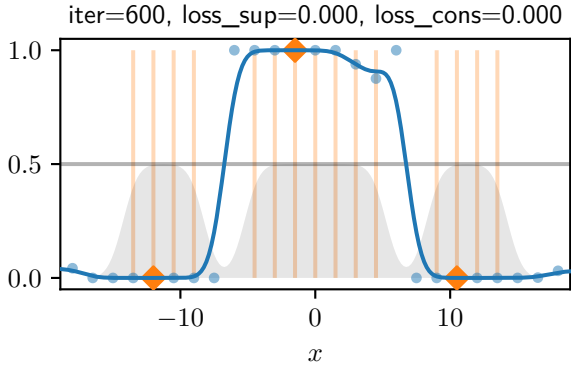
(a) bez konzistencijskog gubitka



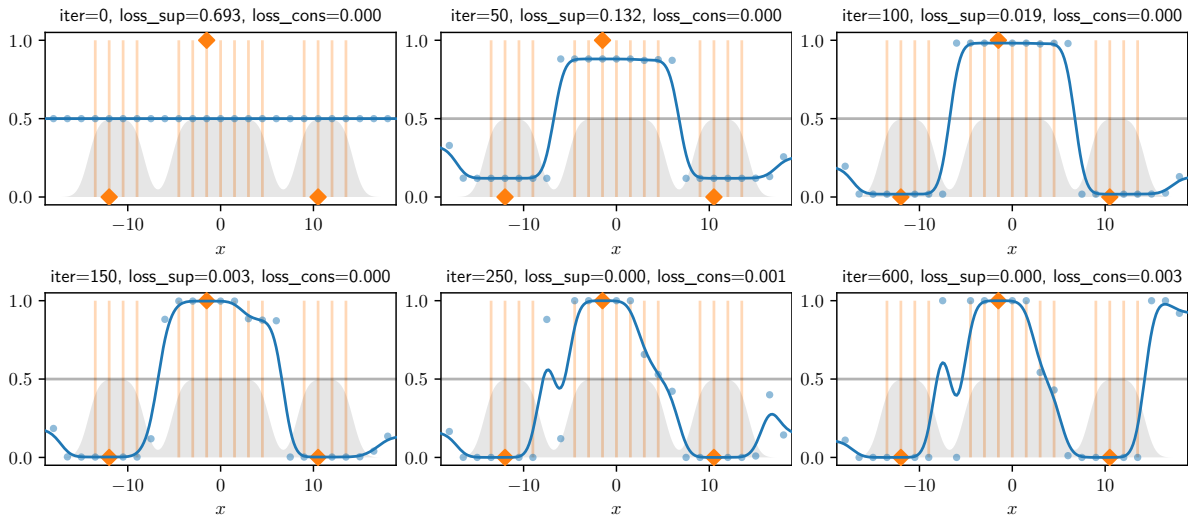
(b) jednosmjerna konzistencija s čistim učenikom



(c) jednosmjerna konzistencija s čistim učiteljem



(d) dvosmjerna konzistencija s 1 čistim ulazom



(e) Prvi redak: početak učenja – 100 iteracija jednosmjerne konzistencije s čistim učiteljem. Drugi redak: nastavak učenja – 500 iteracija jednosmjerne konzistencije s čistim učenikom.

Slika 1.2: Nadzirano i konzistencijsko učenje binarne klasifikacije 1D točaka. Skup za učenje sastoji se od 17 točaka (narančaste crte), od kojih 3 imaju oznake (narančasti dijamanti). Učenje ima 600 iteracija gradijentnog spusta u smjeru predznaka gradijenta s korakom 0.02. Konzistencijski gubitak koristi JS divergenciju. Plava crta je predikcija, a plavi krugovi predstavljaju parametre. Siva površina u pozadini je proporcionalna procjeni gustoće podataka Gaussovom jezgrom standardne devijacije 1. Slike a-d prikazuju rezultate različitih oblika učenja, a slika e međurezultate učenja jednosmjerne konzistencija s čistim učenikom uz inicijalizaciju učenjem s čistim učiteljem, što završava slično kao sa slučajnom inicijalizacijom.

Poglavlje 2

Temeljni pojmovi

Ovo poglavlje daje kratak pregled osnovnih pojmova iz teorije vjerojatnosti, teorije informacije i strojnog učenja.

2.1 Teorija vjerojatnosti

Teorija vjerojatnosti opisuje idealno zaključivanje pri nepotpunom znanju (neizvjesnosti). Ovaj odjeljak daje sažet pregled nekih osnovnih pojmova teorije vjerojatnosti i objašnjava značenje izraza koriste u njenim primjenama. Pri pisanju odjeljka korištena je stranica ProofWiki [32], koja nudi preglednu zbirku matematičkih definicija i teorema.

2.1.1 Vjerojatnosni prostori i slučajne varijable

Slijede definicije temeljnih pojmova teorije vjerojatnosti.

Definicija 1. **σ-algebra** na skupu $\mathbb{X}$ je skup podskupova od $\mathbb{X}$ koji je zatvoren pod komplementom i prebrojivim unijama i sadrži $\mathbb{X}$.

Najveća σ-algebra na skupu Ω je partitivni skup 2^{Ω}, a najmanja je $\{\{\}, \Omega\}$. Primjer σ-algebre na skupu $\Omega = \{1,2,3\}$ je $\{\{\},\{1\},\{2,3\},\{1,2,3\}\}$.

Definicija 2.Mjerljivi prostor je par (Ω, Σ), gdje je Ω skup, a Σ σ-algebra na Ω.

Definicija 3 (**Mjera**)**.** Neka je (Ω, Σ) mjerljivi prostor. Mjera na Σ je preslikavanje $\mu\colon \Sigma \to \mathbb{R} \cup \{+\infty, -\infty\}$ takvo da:

- $\forall \mathbb{E} \in \Sigma\colon \mu(\mathbb{E}) \geq 0$ (nenegativnost),
- $\mu\big(\bigcup_{\mathbb{E}\in\mathbb{S}} \mathbb{E}\big) = \sum_{\mathbb{E}\in\mathbb{S}} \mu(\mathbb{E})$ za svaki prebrojiv skup $\mathbb{S} \subseteq \Sigma$ takav da $\forall \mathbb{E}, \mathbb{F} \in \mathbb{S}\colon \mathbb{E} \neq \mathbb{F} \implies \mathbb{E} \cap \mathbb{F} = \{\}$ (prebrojiva aditivnost) i
- $\mu(\{\}) = 0$.

Definicija 4.Vjerojatnosni prostor je (Ω, Σ, P), gdje

- Ω (prostor uzorkovanja) je skup,
- Σ (prostor događaja) je σ-algebra na Ω i
- $P\colon \Sigma \to [0..1]$ (vjerojatnosna mjera) je mjera na Σ takva da $P(\Omega) = 1$.

Prostor uzorkovanja Ω možemo interpretirati kao skup svih mogućih **ishoda** slučajnog eksperimenta. Ishodi su međusobno isključivi i jedan se mora ostvariti. **Prostor događaja** Σ sadrži podskupove od Ω, za koje kažemo da su **događaji**. Za događaje koji sadrže samo jedan ishod kažemo da su **elementarni događaji**. **Vjerojatnosna mjera** P dodjeljuje **vjerojatnosti** događajima.

Ako je prostor uzorkovanja Ω prebrojiv (konačan ili prebrojivo beskonačan), prostor događaja može sadržavati sve njegove podskupove: $\Sigma = 2^{\Omega}$. Minimalni mogući prostor događaja, $\Sigma = \{\{\}, \Omega\}$, može jedino imati vjerojatnosnu mjeru $\mathbb{E} \mapsto [\![\mathbb{E} = \Omega]\!]$.

Prostor događaja prema navedenoj definiciji ne mora sadržavati sve podskupove prostora uzorkovanja jer to nekad omogućuje jednostavniju formalizaciju i definiranje mjera za neprebrojive skupove uzorkovanja [33, odjeljak 4.3]. Kad je prostor uzorkovanja kontinuiran, obično je prikladan odabir prostora događaja Borelova σ-algebra – najmanja σ-algebra koja uključuje sve otvorene podskupove prostora uzorkovanja [33, odjeljak 4.3].

Definicija 5.Inverzna slika skupa $\mathbb{X}'$ pod funkcijom $f\colon \Omega \to \mathbb{X}$ je skup

$$f^{\leftarrow}(\mathbb{X}') = \{\omega \in \Omega\colon f(\omega) \in \mathbb{X}'\}. \tag{2.1}$$

Definicija 6.Slučajna varijabla na vjerojatnosnom prostoru (Ω, Σ, P) koja poprima vrijednosti u mjerljivom prostoru $(\mathbb{X}, \Sigma')$ je preslikavanje $\underline{x}\colon \Omega \to \mathbb{X}$ takvo da

$$\forall \mathbb{X}' \in \Sigma'\colon \underline{x}^{\leftarrow}(\mathbb{X}') \in \Sigma. \tag{2.2}$$

Slučajna varijabla svakom ishodu iz prostora uzorkovanja Ω dodjeljuje vrijednost iz $\mathbb{X}$. Inverznu sliku $\underline{x}^{\leftarrow}(\mathbb{X}') \in \Sigma$ interpretiramo kao sve ishode za koje slučajna varijabla poprimi vrijednost iz $\mathbb{X}'$. Dakle, vjerojatnost da $\underline{x}$ poprimi neku vrijednost iz $\mathbb{X}' \in \Sigma'$ je $P(\underline{x}^{\leftarrow}(\mathbb{X}')) = P(\{\omega \in \Omega\colon \underline{x}(\omega) \in \mathbb{X}'\})$, što možemo izraziti i oznakom $\mathrm{P}(\underline{x} \in \mathbb{X}')$.

Ovakva formalizacija vjerojatnosnog prostora i slučajnih varijabli omogućuje predstavljanje zavisnosti među slučajnim varijablama. Ishod određuje vrijednosti (realizacije) svih slučajnih varijabli.

Neka su $\underline{x}_1, .., \underline{x}_n$ slučajne varijable na vjerojatnosnom prostoru (Ω, Σ, P), a R predikat. Koristit ćemo uobičajene izraze oblika $R(\underline{x}_1, .., \underline{x}_n)$ za označavanje događaja u kojem $\underline{x}_i$ poprimaju vrijednosti koje zadovoljavaju predikat: $\{\omega \in \Omega\colon R(\underline{x}_1(\omega), .., \underline{x}_n(\omega))\}$. Za vjerojatnost ćemo koristiti oznake oblika $\mathrm{P}(R(\underline{x}_1, .., \underline{x}_n))$. Primjer takve oznake je

$\mathrm{P}(\underline{x}_1 > \underline{x}_2, \underline{x}_3 \in \mathcal{A})$, gdje zarez predstavlja konjunkciju. Pretpostavljat ćemo da je vjerojatnosni prostor takav da postoji događaj za svaki nama zanimljiv skup vrijednosti slučajnih varijabli.

U sljedećim definicijama pretpostavljamo da je $\underline{x}\colon \Omega \to \mathbb{X}$ slučajna varijabla na vjerojatnosnom prostoru (Ω, Σ, P) koja poprima vrijednosti u mjerljivom prostoru $(\mathbb{X}, \Sigma')$.

Definicija 7.Razdioba slučajne varijable $\underline{x}\colon \Omega \to \mathbb{X}$ je funkcija $\Pi_{\underline{x}} : \Sigma' \to [0..1]$ takva da $\forall \mathbb{X}' \in \Sigma'\colon \Pi_{\underline{x}}(\mathbb{X}') = \mathrm{P}(\underline{x} \in \mathbb{X}')$.

Može se pokazati da je razdioba $\Pi_{\underline{x}}$ vjerojatnosna mjera na $(\mathbb{X}, \Sigma')$, pa je $(\mathbb{X}, \Sigma', \Pi_{\underline{x}})$ vjerojatnosni prostor. Svaka slučajna varijabla ima razdiobu.

Definicija 8.Potpora slučajne varijable $\underline{x}\colon \Omega \to \mathbb{X}$ je najmanji skup $\mathbb{X}' \subseteq \mathbb{X}$ takav da $\mathrm{P}(\underline{x} \in \mathbb{X}') = 1$.

Definicija 9.Funkcija vjerojatnosti je funkcija $P\colon \mathbb{X} \to [0..1]$, gdje je $\mathbb{X}$ prebrojiv skup i vrijedi $\sum_{x \in \mathbb{X}} P(x) = 1$.

Definicija 10. Funkcija vjerojatnosti slučajne varijable $\underline{x}\colon \Omega \to \mathbb{X}$ s prebrojivom potporom je funkcija vjerojatnosti $\mathrm{P}_{\underline{x}} : \mathbb{X} \to [0..1]$ za koju vrijedi $\mathrm{P}_{\underline{x}}(x) = \mathrm{P}(\underline{x} = x)$.

Za slučajnu varijablu s prebrojivom potporom $\mathbb{S}$ vrijedi: $\Pi_{\underline{x}}(\mathbb{X}') = \sum_{x \in \mathbb{X}' \cap \mathbb{S}} \mathrm{P}(\underline{x} = x)$.

Definicija 11.Funkcija gustoće vjerojatnosti je funkcija $p\colon \mathbb{X} \to [0..1]$ za koju vrijedi $\int_{x \in \mathbb{X}} p(x)\,\mathrm{d}x = 1$.

Definicija 12 (Funkcija gustoće vjerojatnosti slučajne varijable)**.** Neka je $\underline{x}\colon \Omega \to \mathbb{X}$ slučajna varijabla i $\mathbb{X}' \subseteq \mathbb{X}$. Za funkciju gustoće $\mathrm{p}_{\underline{x}} : \mathbb{X} \to [0..\infty)$ slučajne varijable $\underline{x}$ uz prikladnu definiciju integracije i pretpostavku postojanja integrala [34], vrijedi:

$$\int_{x \in \mathbb{X}'} \mathrm{p}_{\underline{x}}(x)\,\mathrm{d}x = \mathrm{P}(\underline{x} \in \mathbb{X}'). \tag{2.3}$$

Vrijednost funkcije gustoće u nekoj točki nazivamo **gustoćom** (vjerojatnosti). Gustoću u nekoj točki ćemo još označavati s $\mathrm{p}(\underline{x} = x)$. Za kontinuirane slučajne varijable, koje poprimaju vrijednost u $\mathbb{R}^n$, gustoća se može preciznije definirati kao Radon-Nikodymova derivacija $\frac{\mathrm{d}\Pi_x}{\mathrm{d}\lambda}$, gdje je λ Lebesgueova mjera [32].

Gustoća nije definirana kad je skup vrijednosti $\mathbb{X}$ prebrojiv. Kad je skup vrijednosti $\mathbb{X}$ kontinuiran, gustoća može biti nedefinirana u nekim vrijednostima. Npr. slučajna varijabla $\underline{x}\colon \Omega \to \mathbb{R}$ može imati gustoću definiranu svugdje osim u prebrojivom podskupu $\mathbb{S} = \{x \in \mathbb{R}\colon \mathrm{P}(\underline{x} = x) > 0\}$. Za takvu slučajnu varijablu vrijedi:

$$\mathrm{P}(\underline{x} \in \mathbb{X}') = \int_{x \in \mathbb{X}' \setminus \mathbb{S}} \mathrm{p}(\underline{x} = x)\,\mathrm{d}x + \sum_{x \in \mathbb{X}' \cap \mathbb{S}} \mathrm{P}(\underline{x} = x). \tag{2.4}$$

Ako koristimo isti simbol za slučajna varijablu i njenu vrijednost, npr. $\underline{x} = x$, predikate ćemo kraće pisati na način da navedemo samo vrijednost. Tako ćemo umjesto $\mathrm{P}(\underline{x} = x)$ i $\mathrm{p}(\underline{y} = y)$ kraće pisati $\mathrm{P}(x)$ i $\mathrm{p}(y)$. Funkcije $\Pi_{\underline{x}}$, $\mathrm{P}_{\underline{x}}$ i $\mathrm{p}_{\underline{x}}$ nekad ćemo radi preglednosti označavati s $\Pi[\underline{x}]$, $\mathrm{P}[\underline{x}]$ i $\mathrm{p}[\underline{x}]$. Izrazi kao "razdioba $\mathrm{P}[\underline{x}]$" i "razdioba $\mathrm{p}[\underline{x}]$" će se odnositi na razdiobu $\Pi[\underline{x}]$.

2.1.2 Uvjetna, združena i marginalna vjerojatnost

Definicija 13.Uvjetna vjerojatnost događaja $\mathbb{E}$ uz uvjet da se dogodio događaj $\mathbb{F}$ je

$$\mathrm{P}(\mathbb{E} \mid \mathbb{F}) \coloneqq \frac{\mathrm{P}(\mathbb{E} \cap \mathbb{F})}{\mathrm{P}(\mathbb{F})} \text{ kad } \mathrm{P}(\mathbb{F}) > 0. \tag{2.5}$$

Rubni slučajevi su $\mathrm{P}(\mathbb{E} \mid \mathbb{F}) = 0$ kad $\mathbb{E} \cap \mathbb{F} = \{\}$ i $\mathrm{P}(\mathbb{E} \mid \mathbb{F}) = 1$ kad $\mathbb{F} \subseteq \mathbb{E}$. Svaka vjerojatnost može se izraziti kao uvjetna vjerojatnost ako se u uvjet stavi prostor uzorkovanja: $\mathrm{P}(\mathbb{E}) = \mathrm{P}(\mathbb{E} \mid \Omega)$. $\mathrm{P}(\mathbb{E})$ se može interpretirati kao prethodna uvjerenost u $\mathbb{E}$, a $\mathrm{P}(\mathbb{E} \mid \mathbb{F})$ kao uvjerenost nakon opažanja uvjeta $\mathbb{F}$ – eliminiranja svih ishoda iz Ω koji nisu u uvjetu.

Definicija 14.Uvjetna gustoća vjerojatnosti za $\underline{x} = x$ uz uvjet $\underline{y} = y$ je

$$\mathrm{p}(x \mid y) \coloneqq \frac{\mathrm{p}(x, y)}{\mathrm{p}(y)} \text{ kad } \mathrm{p}(y) > 0. \tag{2.6}$$

Oznake funkcije uvjetne gustoće koje se mogu koristiti su i $\mathrm{p}_{\underline{x}\,|y}$ i $\mathrm{p}[\underline{x} \mid \underline{y} = y]$.

Uz poznate uvjetne gustoće ili vjerojatnosti, **pravilo umnoška** daje **združenu razdiobu**:

$$\mathrm{p}(x, y) = \mathrm{p}(x)\,\mathrm{p}(y \mid x). \tag{2.7}$$

Marginalizacija združenih gustoća (ili vjerojatnosti) daje **marginalnu razdiobu**:

$$\mathrm{p}(x) = \int_{y \in \mathbb{Y}} \mathrm{p}(x, y)\,\mathrm{d}y = \int_{y \in \mathbb{Y}} \mathrm{p}(x \mid y)\,\mathrm{p}(y)\,\mathrm{d}y. \tag{2.8}$$

Analogno se zbrojem može izraziti marginalna vjerojatnost diskretnih slučajnih varijabli. Izraz s integralom može vrijediti i za diskretne slučajne varijable ako bismo funkciju gustoće izrazili pomoću Diracove delte: $\mathrm{p}(x) = \sum_{x' \in \mathbb{X}} \mathrm{P}(x')\delta(x - x')$.

2.1.3 Odnosi slučajnih varijabli

Neki mogući odnosi slučajnih varijabli $\underline{x} \colon \Omega \to \mathbb{X}$, $\underline{y} \colon \Omega \to \mathbb{Y}$ i $\underline{z} \colon \Omega \to \mathbb{Z}$ su:

- nezavisnost: $\underline{x} \perp \underline{y}$, što znači da $\mathrm{P}(x, y) = \mathrm{P}(x)\,\mathrm{P}(y)$ za sve x, y,
- zavisnost: $\underline{x} \not\perp \underline{y}$, što znači da $\neg(\underline{x} \perp \underline{y})$,

- uvjetna nezavisnost: $\underline{x} \perp \underline{y} \mid \underline{z}$, što znači da $\mathrm{P}(x, y \mid z) = \mathrm{P}(x \mid z)\,\mathrm{P}(y \mid z)$ za sve x, y, z,
- uvjetna zavisnost: $\underline{x} \not\perp \underline{y} \mid \underline{z}$, što znači da $\neg(\underline{x} \perp \underline{y} \mid \underline{z})$.

Poseban slučaj zavisnosti je **funkcijska zavisnost**. Neka $f\colon \mathbb{X} \to \mathbb{Y}$. Izraz $\underline{y} \coloneqq f(\underline{x})$ definira slučajnu varijablu za koju vrijedi:

$$\mathrm{P}(\underline{y} \in \mathbb{Y}') = \mathrm{P}(\underline{x} \in f^{\leftarrow}(\mathbb{Y}')). \tag{2.9}$$

Ako su $\underline{x}$ i $\underline{y}$ diskretne slučajne varijable, onda $\mathrm{P}(x, y) = \mathrm{P}(x)\,\mathrm{P}(y \mid x) = \mathrm{P}(x)[\![y = f(x)]\!]$. Ako je f bijekcija, onda vrijedi i $\mathrm{P}(x) = \mathrm{P}(\underline{y} = f(x))$. Za kontinuirane slučajne varijable ne mora vrijediti $\mathrm{p}(\underline{x}) = \mathrm{p}(\underline{y} = f(x))$, tj. funkcija gustoće nije invarijantna na reparametrizaciju slučajne varijable. U točkama u kojima je f diferencijabilna vrijedi formula **zamjene varijabli**: $\mathrm{p}(x) = \left|\frac{\mathrm{d}f(x)}{\mathrm{d}x}\right| \mathrm{p}(\underline{y} = f(x))$. Općenitije, ako su $\boldsymbol{\underline{x}}$ i $\boldsymbol{\underline{y}} = f(\boldsymbol{\underline{x}})$ vektori, onda

$$\mathrm{p}(\boldsymbol{x}) = \left|\det \frac{\mathrm{d}f(\boldsymbol{x})}{\mathrm{d}\boldsymbol{x}}\right| \mathrm{p}(\boldsymbol{\underline{y}} = f(\boldsymbol{x})). \tag{2.10}$$

Primjer 1 (Zamjena varijabli)**.** *Neka je $\underline{x}$ slučajna varijabla s gustoćom $p(x) = [\![x > 0]\!] \exp(-x)$. Neka $\underline{y} = \ln(\underline{x})$. Onda $\underline{x} = \exp(\underline{y})$. Primjenom zamjene varijabli dobivamo njenu gustoću:*

$$\mathrm{p}(y) = \left|\frac{\partial x}{\partial y}\right| \mathrm{p}(x) = \left|\frac{\partial \exp(y)}{\partial y}\right| \mathrm{p}(\underline{x} = \exp(y)) = \exp(y - \exp(y)). \tag{2.11}$$

Ta gustoća je maksimalna u $y = 0$ ($x = 1$), dok je gustoća slučajne varijable $\underline{x}$ maksimalna u $x \to 0^+$ ($y \to -\infty$).

Neka su $\underline{x}_1, .., \underline{x}_n$ slučajne varijable na vjerojatnosnom prostoru (Ω, Σ, P). Neka $f\colon \bigtimes_{i=1..n} \mathbb{X}_i \to \mathbb{Y}$. Izraz $\underline{y} \coloneqq f(\underline{x}_1, .., \underline{x}_n)$ definira slučajnu varijablu na istom vjerojatnosnom prostoru za koju vrijedi:

$$\mathrm{P}(\underline{y} \in \mathbb{Y}') = \mathrm{P}((\underline{x}_1, .., \underline{x}_n) \in \{(x_1, .., x_n)\colon f(x_1, .., x_n) \in \mathbb{Y}'\}) \tag{2.12}$$

$$= \mathrm{P}(\{\omega\colon f(\underline{x}_1(\omega), .., \underline{x}_n(\omega)) \in \mathbb{Y}'\}). \tag{2.13}$$

Jednostavan takav primjer je, ako odaberemo $f(x, y) = (x, y)$, **združena slučajna varijabla**, za koju vrijedi: $\mathrm{P}((\underline{x}, \underline{y}) \in \mathbb{Z}') = \mathrm{P}(\{\omega\colon (\underline{x}(\omega), \underline{y}(\omega)) \in \mathbb{Z}'\})$.

2.1.4 Implicitno definiranje vjerojatnosnog prostora

U primjenama se vjerojatnosni prostor i slučajne varijable obično ne definiraju eksplicitno – definiraju se razdiobe i odnosi slučajnih varijabli, a događaji se izražavaju preko predikata koji uključuju slučajne varijable.

Za zaključivanje nam je dovoljno znati skupove $\mathbb{X}_1, .., \mathbb{X}_n$ u kojima razmatrane slučajne varijable $\underset{\sim}{x}_1, .., \underset{\sim}{x}_n$ poprimaju vrijednosti i znati kakva je njihova združena razdioba $\Pi[\underset{\sim}{x}_1, .., \underset{\sim}{x}_n]$. Ne moramo znati konkretan prostor uzorkovanja Ω ni kakve konkretne ishode sadrži događaj izražen predikatom $R(\underset{\sim}{x}_1, .., \underset{\sim}{x}_n)$. Za zaključivanje je dovoljno znati skup vrijednosti koje zadovoljavaju predikat: $\mathbb{X}' = \{(x_1, .., x_n) \in \mathbb{X}_1 \times \cdots \times \mathbb{X}_n \colon R(x_1, .., x_n)\}$. Na temelju skupa $\mathbb{X}'$ i poznate razdiobe vidi se da za vjerojatnosnu mjeru vrijedi $P(R(\underset{\sim}{x}_1, .., \underset{\sim}{x}_n)) = \Pi[\underset{\sim}{x}_1, .., \underset{\sim}{x}_n](\mathbb{X}')$.

Postoji beskonačno konkretnih kombinacija vjerojatnosnih prostora i slučajnih varijabli koje mogu predstavljati istu združenu razdiobu. Jedan slučaj uključuje prostor uzorkovanja Ω jednak skupu u kojem združena slučajna varijabla $(\underset{\sim}{x}_1, .., \underset{\sim}{x}_n)$ poprima vrijednosti: $\Omega = \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$. U tom slučaju vrijednost slučajne varijable $\underset{\sim}{x}_i$ jednaka je odgovarajućem elementu ishoda: $\forall \boldsymbol{\omega} \in \Omega \colon \underset{\sim}{x}_i(\boldsymbol{\omega}) = \boldsymbol{\omega}_{[i]}$. Onda je i vjerojatnosna mjera P jednaka razdiobi $\Pi[\underset{\sim}{x}_1, .., \underset{\sim}{x}_n]$ – obje funkcije kao domenu imaju isti prostor događaja $\Sigma \subseteq \mathbb{X}_1 \times \cdots \times \mathbb{X}_n$.

2.1.5 Očekivanje i zakoni velikih brojeva

Definicija 15.Očekivanje funkcije f slučajne varijable $\underset{\sim}{x} \colon \Omega \to \mathbb{X}$ je

$$\mathbf{E}[f(\underset{\sim}{x})] \coloneqq \sum_{x \in \mathbb{X}} f(x)\,\mathrm{P}(x) \tag{2.14}$$

ako je $\underset{\sim}{x}$ diskretna, a

$$\mathbf{E}[f(\underset{\sim}{x})] \coloneqq \int_{x \in \mathbb{X}} f(x)\,\mathrm{p}(x)\,\mathrm{d}x \tag{2.15}$$

ako je $\underset{\sim}{x}$ kontinuirana.

Alternativne oznake kao $\mathbf{E}_{\underset{\sim}{x}}\, f(x) = \mathbf{E}_{x \sim p}\, f(x)$, gdje su x vrijednosti slučajne varijable $\underset{\sim}{x}$ s razdiobom p, nekad mogu biti prikladnije. Tako se može izraziti očekivanje $\mathbf{E}_{\underset{\sim}{x}}\, f(x, \underset{\sim}{y})$, koje ide samo po slučajnoj varijabli $\underset{\sim}{x}$, i rezultat je slučajna varijabla koja ovisi o $\underset{\sim}{y}$.

Teorem 1 (Slabi i jaki zakon velikih brojeva)**.** *Neka je $(\underset{\sim}{x}_1, .., \underset{\sim}{x}_n)$ niz od n nezavisnih kontinuiranih slučajnih varijabli s razdiobom p s konačnim očekivanjem μ. Neka*

$$\hat{\underset{\sim}{x}}_n = \frac{1}{n} \sum_{1=1..n} \underset{\sim}{x}_i. \tag{2.16}$$

Slabi zakon velikih brojeva: Niz $(\hat{x}_n)_n$ konvergira u vjerojatnosti prema μ, tj. $\forall \epsilon > 0 \colon \lim_{n \to \infty} \mathrm{P}(|\underset{\sim}{x}_n - \mu| < \epsilon) = 1$.

Jaki zakon velikih brojeva: Niz $(\hat{x}_n)_n$ konvergira gotovo sigurno prema μ, tj. $\forall \epsilon > 0 \colon \mathrm{P}(\lim_{n \to \infty} |\underset{\sim}{x}_n - \mu| < \epsilon) = 1$

2.2 Teorija informacije

Ovaj odjeljak daje kratki pregled osnovnih pojmova iz teorije informacije. Koristit ćemo oznake slične su onima predloženima u [35, 36, 37]. Mjere teorije informacije koje kao argumente dobivaju događaje i slučajne varijable implicitno pretpostavljaju vjerojatnosni prostor s vjerojatnosnom mjerom P.

2.2.1 Informacijski sadržaj i entropija

Definicija 16.Informacijski sadržaj I zadovoljava sljedeća svojstva:

- Ima vrijednost 0 za siguran događaj: $\mathrm{P}(\mathbb{E}) = 1 \iff \mathrm{I}(\mathrm{P}(\mathbb{E})) = 0$.
- Veći je za događaj s manjom vjerojatnošću: $\mathrm{P}(\mathbb{E}) < \mathrm{P}(\mathbb{F}) \iff \mathrm{I}(\mathrm{P}(\mathbb{E})) > \mathrm{P}(\mathbb{F})$.
- Informacijski sadržaj konjunkcije nezavisnih događaja je zbroj njihovih informacijskih sadržaja: $\mathbb{E}$ i $\mathbb{F}$ su nezavisni događaji $\iff \mathrm{I}(\mathrm{P}(\mathbb{E} \cap \mathbb{F})) = \mathrm{I}(\mathrm{P}(\mathbb{E})) + \mathrm{I}(\mathrm{P}(\mathbb{F}))$.

Ta svojstva zadovoljavaju funkcije oblika $\mathrm{P}(\mathbb{E}) \mapsto -\log_b \mathrm{P}(\mathbb{E})$, gdje $b \in (1..\infty)$ [38], koje se razlikuju samo u faktoru skaliranja. Najčešće se odabiru baze $b = 2$ (jedinica bit) i $b = \mathrm{e}$ (jedinica nat $= \ln(2)$bit). Mi ćemo koristiti bazu e, koja je uobičajena u strojnom učenju, pa definirati nat $= 1$.

Definicija 17.Informacijski sadržaj (u bazi e) događaja $\mathbb{E}$ je

$$\mathrm{I}(\mathrm{P}(\mathbb{E})) \coloneqq -\ln(\mathrm{P}(\mathbb{E})). \tag{2.17}$$

Vrijednost informacijskog sadržaja je od 0 do ∞ za vjerojatnosti od 1 do 0.

Ako ništa ne znamo o ishodu, možemo saznati je li se ostvario događaj $\mathbb{E}$ otkrivanjem odgovora na broj pitanja jednak informacijskom sadržaju u bazi 2 ili najbližem većem cijelom broju: $\lceil \log_2(\mathrm{P}(\mathbb{E})) \rceil$. Informacijski sadržaj može biti koristan kao mjera neočekivanosti potencijalnog ishoda ili pogreške predviđanja nekog ishoda.

Definicija 18.Entropija slučajne varijable $\underline{x}$ je

$$\mathrm{H}[\underline{x}] \coloneqq \mathop{\mathbf{E}}_{\underline{x}} \mathrm{I}(\mathrm{P}(\underline{x} = x)) = -\mathop{\mathbf{E}}_{\underline{x}} \ln(\mathrm{P}(x)). \tag{2.18}$$

Entropija (funkcije vjerojatnosti) slučajne varijable $\underline{x} \colon \Omega \to \mathbb{X}$ može imati vrijednost od 0 (kad $\exists x \in \mathbb{X} \colon \mathrm{P}(x) = 1$) do $\ln|\mathbb{X}|$ (za uniformnu razdiobu: $\forall x \in \mathbb{X} \colon \mathrm{P}(x) = |\mathbb{X}|^{-1}$). Za slučaj vjerojatnosti 0, uzima se $\lim_{p \to 0^+} p \ln(p) = 0$.

Entropija je očekivanje informacijskog sadržaja i također donja granica očekivane duljine poruke pri optimalnom kodiranju b-arnim prefiksnim kodom [38].

Kod prefiksnog kodiranja svaki događaj je predstavljen nekom kodnom rječju, kodne riječi se mogu razlikovati po duljini i nijedna kodna riječ nije prefiks neke druge. Prednost

različitih duljina je da se događajima s većom vjerojatnošću mogu dodijeliti kraće kodne riječi.

Primjer 2 (Entropija i očekivanje duljine poruke)**.** *Za funkciju vjerojatnosti* $P(n) = (0.5, 0.25, 0.125, 0.125)_{[n]}$ *optimalno binarno kodiranje je* $e(n) = (\texttt{0}, \texttt{10}, \texttt{110}, \texttt{111})_{[n]}$*, a za funkciju vjerojatnosti* $P(n) = (0.25, 0.25, 0.25, 0.25)_{[n]}$ *optimalno je* $e(n) = (\texttt{00}, \texttt{01}, \texttt{10}, \texttt{11})_{[n]}$*. U oba slučaja očekivana duljina koda odgovara entropiji u bazi* 2 *(vrijednosti su* 1.75bit *i* 2bit*). U slučaju funkcije vjerojatnosti* $P(n) = (1/3, 1/3, 1/3)_{[n]}$ *optimalno binarno kodiranje je* $e(n) = (\texttt{0}, \texttt{10}, \texttt{11})_{[n]}$*. U ovom slučaju binarnim kodom ne možemo postići očekivanu duljinu poruke jednaku entropiji: entropija je* $-\log_2(1/3)\text{bit} \approx 1.58\text{bit}$*, dok je očekivana duljina koda* $1/3 \cdot 1 + 2/3 \cdot 2\text{bit} = 5/3\text{bit} \approx 1.67\text{bit}$*.*

Entropija se može interpretirati i kao nesigurnost ili očekivana količina informacija potrebna za saznavanje vrijednosti slučajne varijable.

Ako je umjesto slučajne varijable argument funkcija vjerojatnosti, koristit ćemo oznake s oblim zagradama, npr. $\mathrm{H}(\mathrm{P}[\underline{x}]) = \mathrm{H}[\underline{x}]$.

Definicija 19.Diferencijalna entropija kontinuirane slučajne varijable $\underline{x}$ je

$$\mathrm{h}[\underline{x}] \coloneqq -\mathop{\mathbf{E}}_{\underline{x}} \ln(\mathrm{p}(x)). \tag{2.19}$$

Za razliku od entropije, diferencijalna entropija može biti negativna i, kao i gustoća, nije invarijantna na reparametrizaciju slučajne varijable. Diferencijalna entropija nije pravo proširenje entropije na kontinuirane razdiobe [39].

2.2.2 Usporedba vjerojatnosnih razdioba

Sljedeće informacijsko-teorijske mogu se koristiti za usporedbu razdiobe koje imaju zajedničku domenu.

Definicija 20.Unakrsna entropija funkcije vjerojatnosti q s obzirom na funkciju vjerojatnosti p je

$$\mathrm{H}(p \parallel q) \coloneqq \mathop{\mathbf{E}}_{x\sim p} \mathrm{I}(q(x)) = -\mathop{\mathbf{E}}_{x\sim p} \ln(q(x)). \tag{2.20}$$

Uz fiksni p unakrsna entropija može imati vrijednost od $\mathrm{H}(p \parallel p) = \mathrm{H}(p)$ (kad $q = p$) do ∞ (kad $\exists x \colon p(x) > 0 \wedge q(x) = 0$). Ona odgovara očekivanju duljine poruke ako se koristi optimalan kod za q, a točna funkcija vjerojatnosti je p.

Definicija 21.Relativna entropija funkcije vjerojatnosti q s obzirom na funkciju vjerojat-

nosti p je

$$\mathrm{D}(p \parallel q) \coloneqq \mathop{\mathbf{E}}_{x\sim p}(\ln(p(x)) - \ln(q(x)) = \mathrm{H}(p \parallel q) - \mathrm{H}(p). \tag{2.21}$$

Još jedan naziv za ralativnu entropiju je **Kullback–Leiblerova divergencija** (**KL-divergencija**). Relativna entropija može imati vrijednost od 0 (kad $q = p$) do ∞ (kad $\exists x\colon\, p(x) > 0 \wedge q(x) = 0$). Ona odgovara očekivanju viška duljine poruke ako se koristi optimalan kod za razdiobu q, a stvarna razdioba je p.

Relativna entropija poopćava informacijski sadržaj. Ako ciljna razdioba ima samo element x^* u potpori, tj. njena funkcija vjerojatnosti je $x \mapsto [\![x = x^*]\!]$, onda je relativna entropija funkcije vjerojatnosti q s obzirom na nju

$$\mathrm{D}(x \mapsto [\![x = x^*]\!] \parallel q) = \mathrm{H}(x \mapsto [\![x = x^*]\!] \parallel q) = \mathrm{I}(q(x^*)) = -\ln(q(x^*)). \tag{2.22}$$

Slika 2.1 ilustrira odnose između entropije, unakrsne entropije i relativne entropije.

$\mathrm{H}(p)$	$\mathrm{D}(p \parallel q)$
$\mathrm{H}(p \parallel q)$	

Slika 2.1: Odnosi entropije, unakrsne entropije i relativne entropije, gdje su p i q funkcije vjerojatnosti s istom domenom. Slične ilustracije su prikazane u [35, 36].

Ove oznake mogu se zapamtiti po tome što očekivanje ide po prvom argumentu u zagradi.

Za razliku od diferencijalne entropije, relativna entropija se može definirati općenitije, da obuhvati i kontinuirane razdiobe uz zadržavanje svojstava nenegativnosti i invarijantnosti na reparametrizaciju [40].

Definicija 22 (Relativna entropija kontinuiranih razdioba)**.** Relativna entropija funkcije gustoće q s obzirom na funkciju gustoće p je

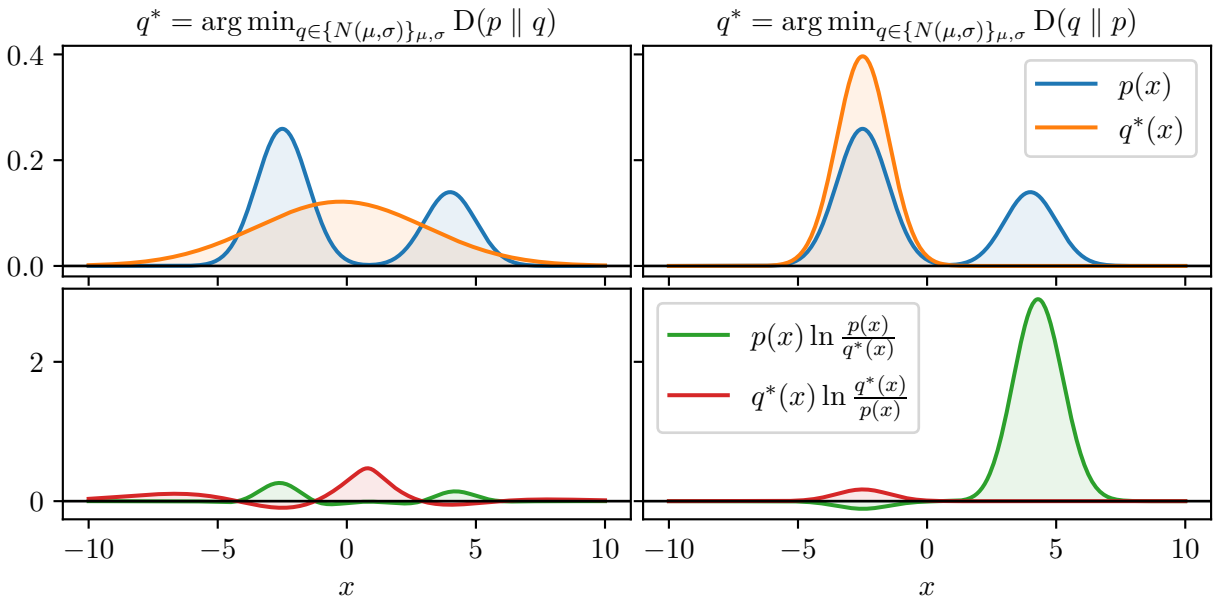
$$\mathrm{D}(p \parallel q) \coloneqq \mathop{\mathbf{E}}_{x\sim p} \ln\left(\frac{p(x)}{q(x)}\right). \tag{2.23}$$

Slično se mogu poopćiti druge mjere koje se mogu svesti na relativnu entropiju.

Slika 2.2 ilustrira vrijednosti podintegralne funkcije relativne entropije između dviju kontinuiranih razdioba i pokazuje asimetričnost relativne entropije.

2.2.3 Mjere odnosa slučajnih varijabli

Sljedeće informacijsko-teorijske mjere govore o odnosima između slučajnih varijabli.



Slika 2.2: Parovi funkcija gustoće i podintegralne funkcije relativne entropije. Plava razdioba, p, je fiksna mješavina normalnih razdioba. Narančaste razdiobe, q^*, su normalne razdiobe. Lijevo q^* minimizira relativnu entropiju normalne razdiobe s obzirom na p. Desno q^* minimizira relativnu entropiju p s obzirom na q^*. U donjem retku su prikazane podintegralne funkcije od $\mathrm{D}(p \parallel q^*)$ (zeleno) i $\mathrm{D}(q^* \parallel p)$ (crveno). Za dobivanje slike korišten je kod iz [41] uz izmjene.

Entropija združene slučajne varijable $(\underline{x}, \underline{y})$ se još naziva **združena entropija**:

$$\mathrm{H}[\underline{x}, \underline{y}] \coloneqq \mathrm{H}[(\underline{x}, \underline{y})] = -\mathop{\mathbf{E}}_{\underline{x},\underline{y}} \ln(\mathrm{P}(x, y)). \tag{2.24}$$

Vrijedi $\mathrm{H}[\underline{x}, \underline{y}] \leq \mathrm{H}[\underline{x}] + \mathrm{H}[\underline{y}]$, uz jednakost samo u slučaju nezavisnosti $\underline{x}$ i $\underline{y}$, kad opažanje jedne slučajne varijable ne govori ništa o drugoj.

Definicija 23.Uvjetna entropija slučajne varijable $\underline{x}$ uz uvjet slučajne varijable $\underline{y}$ je

$$\mathrm{H}[\underline{x} \mid \underline{y}] \coloneqq -\mathop{\mathbf{E}}_{\underline{x},\underline{y}} \ln(\mathrm{P}(x \mid y)) = \mathop{\mathbf{E}}_{\underline{y}} \mathrm{H}(\mathrm{P}[\underline{x} \mid y]) = \mathrm{H}[\underline{x}, \underline{y}] - \mathrm{H}[\underline{y}]. \tag{2.25}$$

Uvjetna entropija je očekivanje entropije slučajne varijable $\underline{x}$ uz znanje o $\underline{y}$. Uz fiksni $\underline{x}$, ona može imati vrijednost od 0 (kad $\exists f\colon \underline{x} = f(\underline{y})$) do $\mathrm{H}[\underline{x}]$ (kad $\underline{x} \perp \underline{y}$).

Definicija 24.Uzajamna informacija slučajnih varijabli $\underline{x}$ i $\underline{y}$ je

$$\mathrm{I}[\underline{x}; \underline{y}] \coloneqq \mathop{\mathbf{E}}_{\underline{x},\underline{y}} \ln\left(\frac{\mathrm{P}(x, y)}{\mathrm{P}(x)\,\mathrm{P}(y)}\right) \tag{2.26}$$

$$= \mathrm{H}[\underline{x}] - \mathrm{H}[\underline{x} \mid \underline{y}] = \mathrm{H}[\underline{y}] - \mathrm{H}[\underline{y} \mid \underline{x}] = \mathrm{H}[\underline{x}] + \mathrm{H}[\underline{y}] - \mathrm{H}[\underline{x}, \underline{y}]. \tag{2.27}$$

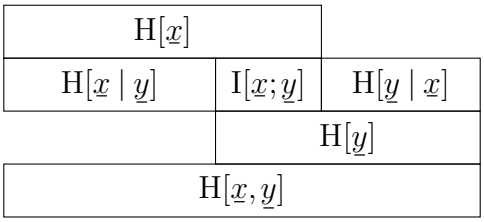
Uzajamna informacija je maksimalna kad su $\underline{x}$ i $\underline{y}$ u funkcijskoj zavisnosti. Ako postoji $\exists f\colon \underline{y} = f(\underline{x})$, onda $\mathrm{H}[\underline{y} \mid \underline{x}] = 0$, pa je uzajamna informacija $\mathrm{I}[\underline{x}; \underline{y}] = \mathrm{H}[\underline{y}] \leq \mathrm{H}[\underline{x}]$ (jednakost ako i $\exists f^{-1}\colon \underline{x} = f^{-1}(\underline{y})$). Uzajamna informacija se može izraziti i preko relativne

entropije:

$$\mathrm{I}[\underline{x};\underline{y}] = \mathrm{D}(\mathrm{P}[\underline{x},\underline{y}] \parallel \mathrm{P}[\underline{x}]\,\mathrm{P}[\underline{y}]) = \mathop{\mathbf{E}}_{\underline{y}} \mathrm{D}(\mathrm{P}[\underline{x} \mid y] \parallel \mathrm{P}[\underline{x}]), \tag{2.28}$$

gdje je $\mathrm{P}[\underline{x}]\,\mathrm{P}[\underline{y}]$ funkcija vjerojatnosti nezavisnih slučajnih varijabli s jednakim marginalnim razdiobama: $(\mathrm{P}[\underline{x}]\,\mathrm{P}[\underline{y}])(x,y) = \mathrm{P}(x)\,\mathrm{P}(y)$. Izraz (2.28) vrijedi i za gustoće kontinuiranih slučajnih varijabli.

Slika 2.3 ilustrira odnose informacijsko-teorijskih mjera dviju slučajnih varijabli.



Slika 2.3: Odnosi informacijsko-teorijskih mjera koje izražavaju zavisnost slučajnih varijabli. Slične ilustracije su prikazane u [35, 36].

Definicija 25.Smanjenje entropije slučajne varijable $\underline{x}$ od opažanja događaja $\mathbb{E}$ je

$$\mathrm{I}[\underline{x};\mathbb{E}] \coloneqq -\mathop{\mathbf{E}}_{\underline{x}} \ln(\mathrm{P}(x)) + \mathop{\mathbf{E}}_{\underline{x}\,|\,\mathbb{E}} \ln(\mathrm{P}(x \mid \mathbb{E})) \tag{2.29}$$

$$= \mathrm{H}[\underline{x}] - \mathrm{H}(\mathrm{P}[\underline{x} \mid \mathbb{E}]). \tag{2.30}$$

Smanjenje entropije* izražava koliko opažanje smanjuje nesigurnost o slučajnoj varijabli. Za razliku od uzajamne informacije, smanjenje entropije može biti i negativno jer aposterirna entropija $\mathrm{H}(\mathrm{P}[\underline{x} \mid \mathbb{E}])$ može biti veća od apriorne entropije $\mathrm{H}[\underline{x}]$.

Definicija 26.Iznena đenje o slučajnoj varijabli $\underline{x}$ od opažanja događaja $\mathbb{E}$ je

$$\mathrm{S}[\underline{x};\mathbb{E}] \coloneqq \mathop{\mathbf{E}}_{x\sim\mathrm{P}[\underline{x}\,|\,\mathbb{E}]} \ln\left(\frac{\mathrm{P}(\underline{x} \mid \mathbb{E})}{P(x)}\right) \tag{2.31}$$

$$= \mathrm{D}(\mathrm{P}[\underline{x} \mid \mathbb{E}] \parallel \mathrm{P}[\underline{x}]). \tag{2.32}$$

Iznenađenje je uvijek nenegativno – može se izraziti kao relativna entropija.

Uzajamna informacija se može izraziti i kao očekivanje smanjenja entropije i kao očekivanje iznenađenja [36, 37]:

$$\mathrm{I}[\underline{x};\underline{y}] = \mathop{\mathbf{E}}_{\underline{y}} \mathrm{I}[\underline{x};y] = \mathop{\mathbf{E}}_{\underline{y}} \mathrm{S}[\underline{x};y], \tag{2.33}$$

*Smanjenje entropije se u literaturi još naziva specifična informacija (engl. *specific information*) [37] i informacijski dobitak *information gain*) [36], ali izraz *information gain* se negdje koristi i za uzajamnu informaciju [42] i iznenađenje i relativnu entropiju [40].

gdje y označava događaj $\underline{y} = y$.

2.3 Statističko zaključivanje

Statističko zaključivanje na temelju poznatih opažanja (podataka) $\mathbb{D} = \{\boldsymbol{d}_1, .., \boldsymbol{d}_n\}$ donosi zaključak o nekom svojstvu razdiobe iz koje dolaze. To svojstvo može biti parametar $\boldsymbol{\theta}$ razdiobe podataka.

Dva glavna pristupa statističkom zaključivanju su frekventistička i bayesovska statistika. Frekventistička statistika parametar koji procjenjuje smatra fiksnim, a nesigurnost u njega mjeri u tome kako se procjena mijenja mijenjanjem opažanih podataka (uzorka) u ponavljanim eksperimentima. Bayesovska statistika parametar smatra nepoznatom vrijednošću o kojoj se može zaključivati kao o drugim slučajnim varijablama primjenom pravila teorije vjerojatnosti, a opažani podaci su fiksni. Više o usporedbi bayesovske i frekventističke statistike govori [43, poglavlje 3].

U strojnom učenju obično se za zaključivanje o parametrima modela na temelju opažanih podataka (skupa za učenje) kaže učenje, a za dobivanje predikcije na temelju parametara modela i ulaznog podatka kaže se zaključivanje.

2.3.1 Bayesovsko zaključivanje

Bayesovsko zaključivanje daje idealna vjerovanja o parametrima $\underline{\boldsymbol{\theta}}$ (hipotezama) nakon opažanja $\mathbb{D}$ kao **aposteriornu razdiobu**:

$$\mathrm{p}(\underline{\boldsymbol{\theta}} \mid \mathbb{D}) = \frac{\mathrm{p}(\mathbb{D}, \boldsymbol{\theta})}{\mathrm{p}(\mathbb{D})} = \frac{\mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta})}{\mathrm{p}(\mathbb{D})}. \tag{2.34}$$

$\mathrm{p}[\boldsymbol{\theta}]$ je **apriorna razdioba**, koja predstavlja prethodna vjerovanja (pretpostavke), $\mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta})$ je **izglednost** (engl. *likelihood*) hipoteze, koja ovisi samo o opažanjima, a

$$\mathrm{p}(\mathbb{D}) = \mathbf{E}[\mathrm{p}(\mathbb{D} \mid \underline{\boldsymbol{\theta}})] = \int_{\boldsymbol{\theta}} \mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta})\,\mathrm{d}\boldsymbol{\theta} \tag{2.35}$$

je **marginalna izglednost** (engl. *marginal likelihood*, *evidence*).

Ako potpora apriorne razdiobe sadrži točnu hipotezu, povećanje broja nezavisnih opažanja $n = |\underline{\mathbb{D}}_n|$ u uzorku $\underline{\mathbb{D}}_n$ povećava očekivano smanjenje entropije (uzajamnu informaciju) $\mathrm{I}[\underline{\boldsymbol{\theta}}; \underline{\mathbb{D}}_n]$ do neke vrijednosti unutar $[0\,..\,\mathrm{H}[\underline{\theta}]]$ i smanjuje nesigurnost o parametrima:

$$\mathrm{H}[\underline{\boldsymbol{\theta}} \mid \underline{\mathbb{D}}_n] = \mathrm{H}[\underline{\boldsymbol{\theta}}] - \mathrm{I}[\underline{\boldsymbol{\theta}}; \underline{\mathbb{D}}_n]. \tag{2.36}$$

Opažanjem se smanjuje **epistemička nesigurnost** – nesigurnost zbog neznanja. Ako ne

postoji veličina uzorka n koja eliminira sve osim jedne hipoteze, uvijek ostaje **aleatorna nesigurnost** – nesigurnost zbog nejednoznačnosti. Konkretno opažanje može i povećati nesigurnost $\mathrm{H}[\underline{\boldsymbol{\theta}} \mid \underline{\mathbb{D}}_n]$, tj. smanjenje entropije može biti negativno (za razliku od uzajamne informacije i iznenađenja).

Zaključivanje o neviđenim podacima

Na temelju aposteriorne razdiobe može se provoditi zaključivanje o neviđenim podacima. Vjerojatnost podatka $\boldsymbol{d}$ (predikciju) daje marginalizacija po svim mogućim parametrima:

$$\mathrm{p}(\boldsymbol{d} \mid \mathbb{D}) = \int \mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{\theta} \mid \mathbb{D})\, \mathrm{d}\boldsymbol{\theta} = \mathop{\mathbf{E}}_{\underline{\boldsymbol{\theta}} \mid \mathbb{D}} \mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta}), \tag{2.37}$$

gdje $\mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta}, \underline{\mathbb{D}}) = \mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta})$ zbog pretpostavke uvjetne nezavisnosti o opažanjima uz poznate parametre, $\underline{\boldsymbol{d}} \perp \underline{\mathbb{D}} \mid \underline{\boldsymbol{\theta}}$. Kod uvjetnih modela dio podatka $(\boldsymbol{x}, \boldsymbol{y})$ je u uvjetu:

$$\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \mathbb{D}) = \mathop{\mathbf{E}}_{\underline{\boldsymbol{\theta}} \mid \mathbb{D}} \mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}). \tag{2.38}$$

Izglednost kod generativnih i uvjetnih modela

Uz pretpostavku međusobne nezavisnosti primjera, izglednost **generativnog modela** može se izraziti umnoškom:

$$\mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta}) = \prod_{\boldsymbol{d} \in \mathbb{D}} \mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta}). \tag{2.39}$$

Uvjetni (diskriminativni) modeli modeliraju uvjetne razdiobe $\mathrm{p}[\boldsymbol{y} \mid \boldsymbol{x}, \mathbb{D}]$, pa kod njih razdioba ulaznih primjera ne ovisi o parametrima $\boldsymbol{\theta}$, tj. $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{x})$. Onda je izglednost

$$\mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta}) = \prod_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{D}} \mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})\, \mathrm{p}(\boldsymbol{x} \mid \boldsymbol{\theta}) = \mathrm{p}(\boldsymbol{x}) \prod_{(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{D}} \mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}). \tag{2.40}$$

Faktor $\mathrm{p}(\boldsymbol{x})$ ne ovisi o parametrima, pa se može zanemariti pri uspoređivanju hipoteza.

2.3.2 Praktični problemi kod zaključivanja

U praksi postoje određeni izazovi koji onemogućuju idealno probabilističko zaključivanje. Dolazak do točnog zaključka zahtijeva uzimanje u obzir svih mogućih hipoteza i dostupnih opažanja. To može zahtijevati neograničeno računanje ili neograničeno logičko i matematičko znanje jer idealno zaključivanje treba uzeti u obzir sve potrebne teoreme i sve moguće hipoteze [44, 45]. Velik problem bayesovskog zaključivanja u praksi je izračun

marginalne izglednosti, koja ovisi o svim mogućim hipotezama (izraz (2.35)). Još jedan problem bayesovskog zaključivanja (i induktivnog zaključivanja općenito) je određivanje apriorne razdiobe, dok funkcija izglednosti općenito sama po sebi nije dovoljna za dolazak do zaključka: hipoteza može biti savršeno prilagođena opažanjima, a da loše predviđa neviđena opažanja.

Zbog praktičnih ograničenja u vremenu i memoriji, složeniji zadaci zahtijevaju pojednostavljenja. To uključuje odabir pojednostavljenih modela i korištenje aproksimacija, statističkih procjena i heuristika u algoritmima pretraživanja hipoteza. Neki primjeri pojednostavljenja su točkasta aproksimacija razdiobe, aproksimiranje jednostavnijom razdiobom, procjene očekivanja na temelju manjeg broja podataka i uzorkovanje pomoću Markovljevog lanca (engl. *Markov Chain Monte Carlo*).

2.3.3 Procjenitelji i točkaste procjene parametara

Razmatramo razdiobu s nama nepoznatim parametrom $\boldsymbol{\theta}$. Slučajni skup $\underline{\mathbb{D}} = \{\underline{\boldsymbol{d}}_1, .., \underline{\boldsymbol{d}}_n\}$ koji se sastoji od međusobno nezavisnih slučajnih varijabli s tom istom razdiobom je **uzorak**. Možemo definirati funkciju f koja daje procjenu parametara na temelju skupa opažanja i slučajnu varijablu koja ovisi o uzorku:

$$\hat{\underline{\boldsymbol{\theta}}} \coloneqq f(\underline{\mathbb{D}}). \tag{2.41}$$

Takva slučajna varijabla naziva se **statistika** i ona je **procjenitelj** parametra $\boldsymbol{\theta}$, a njena vrijednost $\hat{\boldsymbol{\theta}}$ je **procjena** parametra $\boldsymbol{\theta}$.

Svojstva jednostavnog procjenitelja

Ako pretpostavimo da su nam poželjnije procjene s manjom kvadratnom pogreškom s obzirom na stvarni parametar $\theta \in \mathbb{R}$, očekivana pogreška se može rastaviti na kvadrat pristranosti i varijancu:

$$\mathbf{E}\big[(\hat{\underline{\theta}} - \theta)^2\big] = \mathbf{E}[\hat{\underline{\theta}} - \theta]^2 + \mathbf{E}\big[(\hat{\underline{\theta}} - \mathbf{E}[\hat{\underline{\theta}}])^2\big]. \tag{2.42}$$

Kažemo da je procjenitelj $\underline{\theta}$ **nepristran** ako $\mathbf{E}[\hat{\underline{\theta}}] = \theta$. Pristrani procjenitelj može biti poželjniji od nepristranog ako ima manju očekivanu pogrešku. Još jedno poželjno svojstvo je **konzistentnost**. Neka n označava veličinu uzorka $\underline{\mathbb{D}}_n$ procjenitelja $\hat{\underline{\theta}}_n = f(\underline{\mathbb{D}}_n)$. Niz procjenitelja $(\hat{\underline{\theta}}_n)_n$ je konzistentan ako konvergira u vjerojatnosti prema θ.

Ovakve definicije poželjnih svojstava procjenitelja nisu prikladne u nekim slučajevima, npr. kad postoje različite vrijednosti parametra koje daju istu točnu razdiobu. Onda i dalje može biti prikladno razmatrati pogrešku (gubitak) u predikcijama podataka umjesto

u parametrima.

Procjenitelji maksimalne aposteriorne vjerojatnosti i maksimalne izglednosti

Točkasta procjena parametara može se interpretirati kao aproksimacija aposteriorne razdiobe jednom vrijednošću: $\mathrm{p}(\boldsymbol{\theta} \mid \mathbb{D}) \approx \delta(\hat{\boldsymbol{\theta}} - \boldsymbol{\theta})$. Za zaključivanje o podacima onda vrijedi $\mathrm{p}(\boldsymbol{d} \mid \mathbb{D}) \approx \mathrm{p}(\boldsymbol{d} \mid \hat{\boldsymbol{\theta}})$. Time se zadatak uzorkovanja pretvara u optimizacijski zadatak i zaključivanje za složenije zadatke postaje računski ostvarivo.

Procjenitelj maksimalne aposteriorne vjerojatnosti (**MAP-procjenitelj**, engl. *maximum a posteriori estimator*) aproksimira aposteriornu razdiobu njenim modom:

$$\underline{\boldsymbol{\theta}}_{\mathrm{MAP}} \coloneqq \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\boldsymbol{\theta} \mid \underline{\mathbb{D}}) = \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\underline{\mathbb{D}} \mid \boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{\theta}). \quad \text{(neovisnost } \mathrm{p}(\mathbb{D}) \text{ o } \boldsymbol{\theta}) \tag{2.43}$$

Marginalna izglednost iz nazivnika izraza (2.34) ne treba se računati jer ne ovisi o parametrima. Jedan nedostatak MAP-procjene je što ne uzima u obzir nesigurnost o parametrima, što često dovodi do podcjenjivanja nesigurnosti predikcija [43]. Nadalje, mod ne mora biti reprezentativan za razdiobu i, kao ni apriorna i aposteriorna gustoća parametara, nije invarijantan na reparametrizaciju parametara $\underline{\theta}$ [43].

Odabirom uniformne apriorne razdiobe MAP-procjenitelj postaje **procjenitelj maksimalne izglednosti** (**ML-procjenitelj**, od engl. *maximum likelihood*), koji daje parametre koji maksimiziraju gustoću ili vjerojatnost uzorka, tj. imaju najveću **izglednost**:

$$\underline{\boldsymbol{\theta}}_{\mathrm{ML}} \coloneqq \arg\max_{\boldsymbol{\theta}} \mathrm{p}(\underline{\mathbb{D}} \mid \boldsymbol{\theta}). \tag{2.44}$$

ML-procjenitelj je invarijantan na reprezentaciju parametara jer ne koristi apriornu gustoću parametara. Nedostatak ovakvog procjenitelja je veća sklonost **prenaučenosti**: ako je model dovoljno složen ili količina podataka dovoljno mala, veliku izglednost na poznatim podacima mogu dati parametri koji daju malu izglednost na neviđenim podacima, tj. loše generaliziraju.

Procjena očekivanja postupcima Monte Carla

Kad je teško točno izračunati neku marginalizaciju ili očekivanje, mogu biti korisni postupci Monte Carla. Monte Carlo – procjenitelj daje nepristranu i konzistentnu procjenu integrala $I = \int u(x)\,\mathrm{d}x$ kod kojih se funkcija u može izraziti kao umnožak neke funkcije f i gustoće p s ograničenom varijancom.

Definicija 27 (**Monte Carlo – procjenitelj**)**.** Neka je $(\underline{x}_1, .., \underline{x}_n)$ niz nezavisnih slučajnih

varijabli s razdiobom p. Monte Carlo – procjenitelj očekivanja

$$I = \mathop{\mathbf{E}}_{x \sim p} f(x) = \int f(x) p(x) \, \mathrm{d}x \tag{2.45}$$

je

$$\hat{\underline{I}}_n = \frac{1}{n} \sum_{i=1..n} f(\underline{x}_i). \tag{2.46}$$

Neke razdiobe su teško uzorkovati, ali se njihova funkcija gustoće p može lako evaluirati. Algoritmi kao važnosno uzorkovanje i uzorkovanja pomoću Markovljevog lanca omogućuju procjenu očekivanja za takve razdiobe. Važnosno uzorkovanje (engl. *importance sampling*) [46] omogućuje procjenu očekivanja i smanjivanje varijance procjenitelja odabirom prikladne razdiobe q koja se može uzorkovati i oslanjanjem na to da je očekivanje u izrazu (2.45) jednako očekivanju $\mathbf{E}_{x \sim q} f(x) \frac{p(x)}{q(x)}$. U slučaju previše složene funkcije gustoće p ili kada je dostupna samo funkcija koja joj je proporcionalna, može biti moguće uzorkovanje pomoću konstruiranja Markovljevog lanca kojemu je p stacionarna razdioba (engl. *Markov chain Monte Carlo*, MCMC) [47].

2.4 Strojno učenje

Zadatak algoritama strojnog učenja je zaključivanje (predikcija) o neopažanim podatcima na temelju skupa opažanih podataka. Cilj strojnog učenja je ostvariti što bolju performansu na neopažanim podacima, za koju kažemo **generalizacijska performansa** ili **generalizacija**.

Rezultat algoritma strojnog učenja može biti razdioba nad funkcijama ili pojedina funkcija koju nazivamo **hipoteza** ili **primjerak modela**. Hipoteza je funkcija koja provodi zaključivanje o ulaznom podatku i daje predikciju ovisno o zadatku koji algoritam rješava. Izlaz hipoteze može biti gustoća vjerojatnosti, vjerojatnost, razdioba ili neka druga funkcija ulaznog podatka.

Konačan skup podataka nije dovoljan za zaključivanje o tome kako koja hipoteza generalizira. Većina hipoteza koje su konzistentne s opažanim podacima neće dobro generalizirati[†]. Za generalizaciju su potrebne dodatne pretpostavke koje nazivamo **induktivna pristranost**. Kod idealnog zaključivanja induktivna pristranost je određena apriornom razdiobom nad hipotezama i podrazumijeva odabrani **model**, tj. skup razmatranih hi-

[†]Slično, teoremi pod imenom "No free lunch" govore o tome kako nema algoritma koji na uniformno uzorkovanim zadacima u prosjeku radi bolje od svih drugih [48]. Ipak, nama zanimljivi zadaci dijele neke pravilnosti, kao to da se na temelju prošlih opažanja nešto može znati o budućima, pa postoje induktivne pristranosti koje su općenito korisne.

poteza. Česte induktivne pristranosti su davanje prednosti jednostavnijim hipotezama (Occamova oštrica), znanje dobiveno na temelju validacije i drugo znanje o zadatku i pravilnostima u podacima.

Uz ograničene računalne resurse nije moguće idealno probabilističko zaključivanje, koje zahtijeva uzimanje u obzir svih mogućih hipoteza i primjenu svog potrebnog logičkog znanja (odjeljak 2.3.2). Zato je kod složenijih zadataka potrebna je učinkovita strategija istraživanja, tj. odabira hipoteza koje će se uzeti u obzir. Ako su razmatrane hipoteze previše raznolike, algoritam će trošiti previše vremena na hipoteze s malom vjerojatnošću, a ako nisu dovoljno raznolike, skup hipoteza koje algoritam može uzeti u obzir je previše ograničen. Praktični algoritmi umjesto aposteriorne razdiobe često daju točkastu procjenu parametara, a mnogi algoritmi i nemaju jasnu probabilističku interpretaciju. Sva računska pojednostavljenja i heuristike implicitno ili eksplicitno utječu na induktivnu pristranost i rezultate učenja.

2.4.1 Komponente algoritma strojnog učenja

Osnovne komponente algoritma strojnog učenja su model i algoritam učenja.

Model je skup hipoteza (primjeraka modela) koje se uzimaju u obzir: $\mathbb{H} = \{h_{\boldsymbol{\theta}}\}_{\boldsymbol{\theta} \in \Theta}$, gdje je Θ skup svih mogućih parametara modela. Model se može definirati i kao funkcija s domenom $\Theta \times \mathbb{X}$ ako je $\mathbb{X}$ domena primjerka modela. Ako su svi parametri primjerka modela ujedno i opažani podaci, kažemo da je model neparametarski. Induktivna pristranost koja dolazi od modela je u ograničavanju skupa hipoteza. U model mogu biti ugrađene čvrste pretpostavke o nekim pravilnostima u podacima kao što su ekvivarijantnost na pomake u prostoru i vremenu i druge simetrije. Bitno ograničenje je i **kapacitet** (ili složnost) modela. Modeli s većim kapacitetom mogu se bolje prilagoditi većoj količini i složenosti podataka. Kad je dostupno manje podataka, prikladniji su jednostavniji modeli, koji ne sadrže previše složene hipoteze, dok prejednostavni modeli ne mogu imati dobru performansu ni na skupu za učenje.

Algoritam učenja određuje cilj i ostale induktivne pristranosti. Algoritam učenja općenito se može predstaviti funkcijom $(\mathbb{H}, \mathbb{D}) \mapsto \mathrm{p}[\underline{\boldsymbol{\theta}} \mid \mathbb{H}, \mathbb{D}]$, koja model $\mathbb{H}$ i skup podataka za učenje $\mathbb{D}$ preslikava u razdiobu nad hipotezama ili jednu hipotezu $h_{\boldsymbol{\theta}}$. Dvije komponente algoritma učenja koje se često ističu su funkcija pogreške i optimizacijski algoritam (analog kod bayesovskog zaključivanja algoritam uzorkovanja aposteriorne razdiobe).

Funkcija pogreške (**ciljna funkcija**) $E(\boldsymbol{\theta}, \mathbb{D})$ ocjenjuje dobrotu hipoteze na skupu podataka $\mathbb{D}$. Često se pretpostavlja da su primjeri $\boldsymbol{d} \in \mathbb{D}$ nezavisni uzorci iz neke stvarne razdiobe. Onda se funkcija pogreške može izraziti kao zbroj gubitaka koji nezavisno ocje-

njuju predviđanja na pojedinim primjerima:

$$E(\boldsymbol{\theta}, \mathbb{D}) = \frac{1}{|\mathbb{D}|} \sum_{\boldsymbol{d} \in \mathbb{D}} L(\boldsymbol{\theta}, \boldsymbol{d}) + \lambda R(\boldsymbol{\theta}, \mathbb{D}), \tag{2.47}$$

gdje je L funkcija gubitka, a $\lambda R(\boldsymbol{\theta}, \mathbb{D})$ općeniti član koji predstavlja **regularizaciju**, dodatnu induktivnu pristranost. Kod probabilističkog modela funkcija pogreške je proporcionalna logaritmu aposteriorne gustoće ili vjerojatnosti:

$$\begin{aligned} -\ln(\mathrm{p}(\boldsymbol{\theta} \mid \mathbb{D})) &= -\ln(\mathrm{p}(\mathbb{D} \mid \boldsymbol{\theta})) - \ln(\mathrm{p}(\boldsymbol{\theta})) + \ln(\mathrm{p}(\mathbb{D})) \quad \text{(logaritam izraza (2.34))} \qquad (2.48) \\ &= -\sum_{\boldsymbol{d} \in \mathbb{D}} \ln(\mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta})) - \ln(\mathrm{p}(\boldsymbol{\theta})) + \ln(\mathrm{p}(\mathbb{D})) \quad \text{(međusobna nezavisnost } \boldsymbol{d} \in \mathbb{D}) \\ &\propto \frac{1}{|\mathbb{D}|} \sum_{\boldsymbol{d} \in \mathbb{D}} \underbrace{-\ln(\mathrm{p}(\boldsymbol{d} \mid \boldsymbol{\theta}))}_{L(\boldsymbol{\theta}, d)} \underbrace{-\frac{1}{|\mathbb{D}|} \ln(\mathrm{p}(\boldsymbol{\theta}))}_{\lambda R(\boldsymbol{\theta}, \mathbb{D})} \quad \text{(neovisnost p}(\mathbb{D}) \text{ o } \boldsymbol{\theta}) \qquad (2.49) \end{aligned}$$

Optimizacijski algoritam traži hipotezu koja minimizira funkciju pogreške:

$$\boldsymbol{\theta}^* = \arg\min_{\boldsymbol{\theta}} E(\boldsymbol{\theta}, \mathbb{D}). \tag{2.50}$$

Kod složenijih zadataka i algoritama učenja, kad funkcija pogreške nije konveksna, vjerojatnost pronalaska globalnog optimuma je zanemariva. U takvim slučajevima inicijalizacija parametara i odabir optimizacijskog algoritma određuju bitan dio induktivne pristranosti. Često se koriste optimizacijski algoritmi koji se temelje na gradijentnom spustu i u iteracijama koriste stohastičku procjenu gradijenta na temelju manjeg broja primjera za učenje. Osim što smanjuje količinu računanja, ta stohastičnost može biti pogodna za izbjegavanje zapinjanja kod nekonveksnih funkcija pogreške.

2.4.2 Kategorije strojnog učenja

Jedna od glavnih podjela algoritama strojnog učenja je na nadzirano učenje i nenadzirano učenje. Ona ne obuhvaća sve algoritme, nije uvijek jasna i može ovisiti o tome za što se algoritam primjenjuje.

Kod **nadziranog učenja** skup podataka sastoji se od ulaza s pridruženim oznakama: $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{D} \subset \mathbb{X} \times \mathbb{Y}$. Domena primjerka modela je $\mathbb{X}$, a kodomena je $\mathbb{Y}$ ili skup razdioba nad $\mathbb{Y}$. Zadaci koje nadzirano učenje neposredno rješava su diskriminativni: cilj je na temelju skupa opažanih ulaza i izlaza naučiti predviđati izlaze za neopažane buduće ulaze. Zadatak je klasifikacijski ako je izlazni skup $\mathbb{Y}$ diskretan, a regresijski ako je kontinuiran.

Osnovni primjer funkcije gubitka probabilističkog modela je negativna log-izglednost:

$$L(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = -\ln(\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta})), \tag{2.51}$$

gdje p označava vjerojatnost ili gustoću vjerojatnosti.

Jednostavan primjer klasifikacijskog modela je višeklasna logistička regresija.

Primjer 3 (Višeklasna logistička regresija)**.** *Model višeklasne logistička regresije ima oblik*

$$h_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x}) = \mathrm{softmax}(\boldsymbol{W}\phi(\boldsymbol{x}) + \boldsymbol{b}), \tag{2.52}$$

gdje $\mathrm{softmax}(\boldsymbol{s}) \coloneqq \frac{1}{\mathbf{1}^\intercal \exp(\boldsymbol{s})} \exp(\boldsymbol{s})$*,* $\boldsymbol{W}$ *i* $\boldsymbol{b}$ *su afini parametri, a* ϕ *je nelinearno preslikavanje iz ulaznog prostora u prostor značajki. Gubitak je* $L(y, h(\boldsymbol{x})) = -\ln(\mathrm{P}(y \mid \boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b})$*, gdje* $\mathrm{P}(y \mid \boldsymbol{x}, \boldsymbol{W}, \boldsymbol{b}) = h_{\boldsymbol{W},\boldsymbol{b}}(\boldsymbol{x})_{[y]}$ *za* $y \in \{1..C\}$*.*

Za nadzirano učenje su obično prikladni **diskriminativni (uvjetni) modeli** jer oni na temelju ulaza predviđaju izlaz. Moguće je i učenje generativnih modela koji modeliraju cijelu razdiobu podataka $\mathrm{p}[\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}} \mid \boldsymbol{\theta}]$. Jedna moguća faktorizacija takvog generativnog modela je $\mathrm{p}(\boldsymbol{x} \mid \boldsymbol{y}, \boldsymbol{\theta})\,\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{\theta})$. Za rješavanje diskriminativnog zadatka može se izraziti uvjetna vjerojatnost: $\mathrm{p}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \frac{\mathrm{p}(\boldsymbol{x}, \boldsymbol{y} \mid \boldsymbol{\theta})}{\int_{\boldsymbol{y}} \mathrm{p}(\boldsymbol{x}, \boldsymbol{y} \mid \theta)}$.

Nenadzirano učenje obuhvaća algoritme koji traže pravilnosti (strukturu, uzorke, odnose) u podacima (ili skupu podataka). Mnogi algoritmi nenadziranog učenja mogu se izraziti tako da traže hipoteze koje, za razliku od hipoteza kod nadziranog učenje, ne moraju davati konkretne vrijednosti izlaza, nego su bitni samo odnosi između izlaza različitih ulaza.

Neke vrste algoritama koje spadaju pod nenadzirano učenje su algoritmi za procjenu gustoće, grupiranje, pronalaženje neobičnih podataka, smanjivanje dimenzionalnosti i učenje reprezentacije. Za procjenu gustoće su izravno primjenjivi generativni modeli. Nenadzirano učenje može biti korisno za analiziranje i razumijevanje podataka ili kao komponenta drugih algoritama strojnog učenja. Isto tako nadzirano učEnje može biti komponenta nenadziranog učenja. Primjer može biti modeliranje razdiobe $\mathrm{p}[\underline{\boldsymbol{x}}, \underline{\boldsymbol{y}}]$ faktorizacijom $\mathrm{p}(x)\,\mathrm{p}(y \mid x)$, gdje se za $\mathrm{p}[\underline{\boldsymbol{y}} \mid \underline{\boldsymbol{x}}]$ koristi nadzirano učenje.

Samonadzirano učenje je oblik nenadziranog učenja kod kojeg se rješava neki **samonadzirani zadatak** (engl. *pretext task*) – zadatak kojeg nam samog po sebi nije cilj riješiti, ali njegovo rješavanje rješava podzadatke koji su korisni i za rješavanje nekog ciljnog zadatka (engl. *downstream task*). Npr. rješenje samonadziranog zadatka može dati korisne reprezentacije za rješavanje ciljnog zadatka ili približiti stanje algoritma učenja boljim hipotezama. Općenit zadatak samonadziranog učenja može se definirati kao predviđanje jednog dijela značajki ulaznog primjera na temelju nekog drugog dijela značajki. Sa-

monadzirano učenje često se sastoji komponente nadziranog učenja i komponente koja iz izvornih podataka stvara podatke za komponentu nadziranog učenja. Neki primjeri samonadziranih zadataka su predviđanje sljedećih riječi ili slika u nizu, predviđanje maskiranog dijela slike, raspoznavanje odgovaraju li izmijenjene inačice primjera za učenje istom primjeru za učenje, bojanje zasivljene slike, uklanjanje šuma, učenje sažete reprezentacije.

Algoritmi strojnog učenja mogu kombinirati više nadziranih i nenadziranih komponenata. Ako se jedan zadatak (pomoćni ili izvorni) rješava kako bi se poboljšalo rješavanje nekog ciljnog zadatka, kažemo da je to **učenje s prijenosom znanja**. Pomoćni zadatak je obično nadzirano ili nenadzirano (samonadzirano) učenje na jednom skupu podataka, a za ciljni zadatak se koristi označeni skup za učenje.

Polunadzirano učenje je oblik strojnog učenja koji kombinira nadzirano učenje i nenadzirano učenje za rješavanje istog diskriminativnog zadatka kao njegova nadzirana komponenta. Cilj takvog učenja je pomoću nanadzirane komponente iskoristiti dodatne informacije iz neoznačenih podataka. Takvi algoritmi mogu biti korisni kad je dostupan velik neoznačeni skup podataka koji sadrži korisne informacije, a označeni skup podataka nije dovoljan za postizanje dovoljno dobre generalizacijske performanse. Polunadziranog učenje može se provoditi optimiranjem kombinacije nadziranog i nenadziranog gubitka. Moguće je i učenje s prijenosom znanja kod kojeg se prvo pronalaze parametri koji dobro rješavaju samonadzirani zadatak na neoznačenom skupu, a onda se koristi označeni skup podataka za nadzirani nastavak učenja. Odjeljak 3.3 predstavlja pregled polunadziranih algoritama.

2.5 Duboko učenje

Većina ovog odjeljka se temelji na [41].

Dok je kod klasičnih (plitkih) modela preslikavanje ϕ iz ulaznog prostora u prostor značajki obično ručno određeno ili se uči na temelju nekih jednostavnih pretpostavki (neparametarski modeli, jezgrene metode), kod **dubokog učenja** [49] ϕ se uči. Odabirom

$$\phi(\boldsymbol{x}) = f(\boldsymbol{W}_{\mathrm{h}}\boldsymbol{x} + \boldsymbol{b}_{\mathrm{h}}), \tag{2.53}$$

gdje je $\boldsymbol{W}_{\mathrm{h}}$ matrica težina, $\boldsymbol{b}_{\mathrm{h}}$ vektor pomaka, a f nelinearna prijenosna funkcija koja se primjenjuje na svaki element zasebno, dobivamo jednostavan duboki model s jednim skrivenim slojem:

$$h_{\boldsymbol{\theta}}(\boldsymbol{x}) = f(\boldsymbol{W}_{\mathrm{o}}^{\mathsf{T}} f(\boldsymbol{W}_{\mathrm{h}}\boldsymbol{x} + \boldsymbol{b}_{\mathrm{h}}) + \boldsymbol{b}_{\mathrm{o}}). \tag{2.54}$$

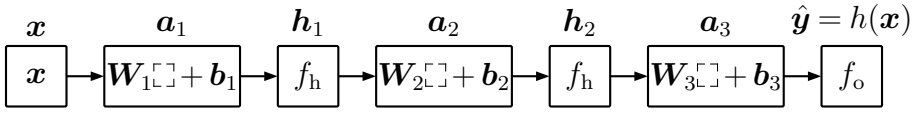
Svi parametri tog modela su $\boldsymbol{\theta} = (\boldsymbol{W}_{\mathrm{h}}, \boldsymbol{b}_{\mathrm{h}}, \boldsymbol{W}_{\mathrm{o}}, \boldsymbol{b}_{\mathrm{o}})$.

Za duboke modele funkcija pogreške nije konveksna, pa nije garantirano da će optimizacija biti uspješna. Empirijski rezultati ipak pokazuju da se neki složeni duboki modeli mogu uspješno optimirati stohastičkim algoritmima koji se temelje na gradijentnom spustu (odjeljak 2.5.2).

2.5.1 Duboki unaprijedni modeli

Prema **teoremu o univerzalnoj aproksimaciji** [50] modeli s (barem) jednim skrivenim slojem su univerzalni aproksimatori, tj. uz dovoljnu dimenziju (širinu) skrivenog sloja mogu proizvoljno dobro aproksimirati svaku neprekinutu funkciju kojoj je domena konveksni podskup od $\mathbb{R}$. Teorem o univerzalnoj aproksimaciji ne govori o tome hoće li takav model generalizirati.

Primjerak modela može se predstaviti računskim grafom, tj. usmjerenim acikličkim grafom čiji listovi predstavljaju ulaze, a drugi čvorovi operacije i međurezultate ili konačne rezultate. Dublji modeli, kao na slici 2.4, često mogu funkcije koje su nam zanimljive jednako dobro ostvariti s manje parametara jer se isti izlaz (značajka) ranijeg sloja može više puta upotrijebiti za računanje složenijih značajki kasnijih slojeva [49]. Duboki modeli prilagođeni određenim vrstama zadataka imaju ugrađene još neke pretpostavke o podacima, kao ekvivarijantnost na pomake u prostoru ili vremenu. Takvi su konvlucijski modeli opisani u odjeljku 2.5.4.



Slika 2.4: Prikaz jednostavnog dubokog modela s 3 sloja afine transformacije i prijenosnom funkcijom f_h. Sadržaj čvorova opisuje operaciju koju ostvaruju izrazom ili simbolom funkcije i prikazuje parametre koji se uče ili ulazni podatak $\boldsymbol{x}$. [ ] označava ulaz koji dolazi iz ranijeg čvora. Slika je preuzeta iz [41].

Kao prijenosna funkcija se u praksi često koristi zglobnica (ReLU, engl. *rectified linear unit*), $\mathrm{ReLU}(x) = \max(0, x)$. Ona je računski jednostavnija i pogodnija je za optimizaciju na temelju gradijenta od funkcija sa zasićenjima kao što su logistička funkcija ($\sigma(s) = \frac{\exp(s)}{1+\exp(s)}$) i tangens hiperbolni [51].

2.5.2 Optimizacija parametara

Zadaci dubokog učenja obično uključuju rješavanje visokodimenzionalnih nelinearnih optimizacijskih zadataka. Parametri dubokih modela obično se optimiraju postupcima koji se temelje na stohastičkoj procjeni gradijenta funkcije pogreške.

Problemi u optimizaciji

Neki od problema u nekonveksnoj optimizaciji u dubokom učenju navedeni u [49] su:

1. Loše kondicioniranje Hesseove matrice. Prevelike druge derivacije mogu biti razlog da funkcija pogreške raste i s jako malim korakom optimizacije.
2. Stacionarne točke i područja s niskim gradijentom. Zbog nelinearnosti i simetrija u strukturi dobokog modela moguć je velik broj lokalnih (i globalnih) ekstrema i sedlastih točaka. Zbog visoke dimenzionalnosti puno su češće sedlaste točke, koje su po nekim smjerovima lokalni minimumi, a po drugim maksimumi. Kod optimizacije dubukih modela lokalni minimumi i sedlaste točke empirijski nisu velik problem. Stohastički optimizacijski algoritmi pomažu izbjeći sedlaste točke i veće dimenzije slojeva daju veći broj prilika za optimizaciju.
3. Iščezavajući i eksplodirajući gradijenti. Kod dubokih modela s puno slojeva gubici kod nekih ulaza mogu imati previsoke vrijednosti gradijenta (litice). Osim inicijalizacije može pomoći podrezivanje gradijenta i manji korak učenja tijekom početnih iteracija učenja, dok su gradijenti obično veći. Eksplodirajući i iščezavajući gradijenti su češće problem kod povratnih modela kod kojih ista matrica više puta množi značajke.
4. Neslaganje globalne i lokalne strukture funkcije pogreške. Put do mimimuma može biti neizravan i zahtijevati puno iteracija optimizacije.
5. Neprecizne procjena gradijenta. Ograničenje broj primjera u mini-grupi može dati lošu procjenu gradijenta. Nekad može pomoći inercija kod optimzacijskog algoritma.

Izračun derivacija unatražnom automatskom diferencijacijom

Gradijent se može učinkovito računati algoritmom **unatražne automatske diferencijacije** (propagacije pogreške unatrag [52]), koji se temelji na pravilu deriviranja kompozicije funkcija.

Neka je f funkcija s ulazima $x_1, .., x_n$ i izlazom y. Neka se izračun te funkcije sastoji od elementarnih operacija koje računaju međurezultate i izlaz na temelju ulaza ili drugih međurezultata. Izračun izlaza može se izraziti usmjerenim acikličkim grafom G čiji su čvorovi varijable koje se javljaju u izračunu: ulazi su korijeni, međurezultati su unutarnji čvorovi, izlazi su listovi, a bridovi su parovi (a, b), gdje je roditelj a jedan od ulaza elementarne operacije, a dijete $b \in \mathrm{ch}_G(a)$ izlaz elementarne operacije (orijentacija bridova kao na slici 2.4).

Svakom bridu (a, b) odgovara parcijalna derivacija (jakobijan) $\frac{\partial b}{\partial a}$ koja je poznata za svaku elementarnu operaciju. Primjenom pravila deriviranja kompozicije funkcija, čvoru a može se pridružiti derivacija izlaza cijele funkcije f po međurezultatu a rekurzivnim

izrazom:

$$\frac{\partial y}{\partial a} = \frac{\partial y}{\partial \operatorname{ch}_G(a)} \frac{\partial \operatorname{ch}_G(a)}{\partial a} = \sum_{b \in \operatorname{ch}_G(a)} \frac{\partial y}{\partial b} \frac{\partial b}{\partial a}. \tag{2.55}$$

Unatražna automatska diferencijacija računa derivaciju s obzirom na odabrane čvorove $x_{i_1}, .., x_{i_k}$ tako da računa derivacije po svim putovima izlaza od izlaza prema $x_{i_1}, .., x_{i_k}$. U izrazu (2.55) treba biti izračunata derivacija $\frac{\partial y}{\partial b}$ da bi se izračunala derivacija $\frac{\partial y}{\partial a}$ i derivacije po drugim potencijalnim roditeljima čvora b. Ako je izlaz y skalar, a međurezultati su vektori, na desnoj strani izraza (2.55) uvijek se množi vektor redak s matricom.

Tablica 2.1 prikazuje parcijalne derivacije (jakobijane) nekih operacija s obzirom na njihove ulaze.

Operacija	**Derivacije**
$\boldsymbol{y} = \boldsymbol{W}\boldsymbol{x} + \boldsymbol{b}$,	$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \boldsymbol{W}$, $\frac{\partial \boldsymbol{y}_{[i]}}{\partial (\boldsymbol{W}_{[j,:]})^\mathsf{T}} = [\![i = j]\!] \boldsymbol{x}^\mathsf{T}$, $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{b}} = \mathbf{I}$
$\boldsymbol{y} = \boldsymbol{a} \odot \boldsymbol{b}$,	$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{a}} = \operatorname{diag}(\boldsymbol{b})$, $\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{b}} = \operatorname{diag}(\boldsymbol{a})$
$\boldsymbol{y} = \operatorname{ReLU}(\boldsymbol{x})$	$\frac{\partial \boldsymbol{y}_{[i]}}{\partial \boldsymbol{x}_{[j]}} = [\![i = j]\!] [\![\boldsymbol{x}_{[j]} > 0]\!]$
$\boldsymbol{y} = \sigma(\boldsymbol{x})$	$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \operatorname{diag}(\boldsymbol{y} \odot (\mathbf{1} - \boldsymbol{y}))$
$\boldsymbol{y} = \tanh(\boldsymbol{x})$	$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}} = \operatorname{diag}(1 - \boldsymbol{y} \odot \boldsymbol{y})$
$\boldsymbol{y} = \operatorname{softmax}(\boldsymbol{x})$	$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}_{[j]}} = \boldsymbol{y} \odot (\mathbf{e}_j - \boldsymbol{y})$
$y = -t \ln \sigma(x) - (1-t) \ln(1 - \sigma(x))$	$\frac{\partial y}{\partial x} = \sigma(x) - t$
$y = -\ln \operatorname{softmax}(\boldsymbol{x})_{[j]}$	$\frac{\partial y}{\partial \boldsymbol{x}} = (\operatorname{softmax}(\boldsymbol{x}) - \mathbf{e}_j)^\mathsf{T}$

Tablica 2.1: Parcijalne derivacije (jakobijani) nekih operacija po njihovim ulazima. $\mathbf{e}_t$ označava jednojedinični vektor s elementima $\mathbf{e}_{j\,[i]} \coloneqq [\![i = j]\!]$. Tablica je preuzeta iz [41].

Radni okviri kao PyTorch [53] i TensorFlow [54] definiraju elementarne operacije i funkcije za računanje gradijenta tako da djeluju nad grupama podataka umjesto nad pojedinim podacima. Dio memorije za čuvanje međurezultata potrebnih za računanje derivacija može se uštediti tako da se neki međurezultati obrišu i ponovo izračunavaju kad budu potrebni za računanje derivacija (engl. *gradient checkpointing*) [55].

Ako je broj ulaza manji od broja izlaza, može biti učinkovitiji algoritam unaprijedne automatske diferencijacije, koji kreće od ulaza prema izlazima [56].

Stohastička optimizacija

Stohastički gradijentni spust se od gradijentnog spusta razlikuje po tome što u pojedinoj iteraciji koristi gradijent procjene pogreške na temelju slučajnog podskupa skupa za

učenje. Stohastički gradijentni spust ima i-tu iteraciju oblika

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \eta_i \boldsymbol{g}_i, \tag{2.56}$$

gdje je $\boldsymbol{g}_i = \nabla_{\boldsymbol{\theta}_i} E(\boldsymbol{\theta}_i, \mathbb{D}_i)$ gradijent pogreške, η_i korak optimizacije, $\boldsymbol{\theta}_i$ vrijednost parametara, a $\mathbb{D}_i$ slučajno odabrani podskup skupa za učenje veličine B, koji se još naziva **mini-grupa**. Dovoljan uvjet za konvergenciju stohastičkog gradijentnog spusta je da se korak optimizacije η_i smanjuje, da $\sum_i \eta_i = \infty$ i da $\sum_i \eta_i^2 < \infty$ [49]. Učenje s manjim, ali ne premalim, mini-grupama može imati bolji regularizacijski učinak, ali zahtijeva više iteracija za konvergenciju [57].

Jednostavan način za dobivanje mini-grupa je slučajni odabir B primjera za učenje u svakoj iteraciji (s ili bez ponavljanja), ali učinkovitije je učenje kod kojeg se prije svakog prolaza skup za učenje slučajno permutira i podijeli na $\lfloor |\mathbb{D}|/B \rfloor$ mini-grupa [58]. Obično se jedan prolaz kroz skup za učenje naziva **epoha**.

Predložene su različite heuristike za poboljšavanje optimizacije i generalizacije. **Inercija s prigušenjem** može smanjiti šum procjene gradijenta i ublažiti oscilacije i zapinjanje u stacionarnim točkama [59]. Uz faktor prigušenja $\gamma \in (0..1)$ iteracija ima oblik

$$\boldsymbol{m}_i = \gamma \boldsymbol{m}_{i-1} + \boldsymbol{g}_i, \tag{2.57}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \eta_i \hat{\boldsymbol{m}}_i. \tag{2.58}$$

RMSProp [60] dijeli komponente gradijenta eksponencijalnim pokretnim prosjekom komponenata kroz iteracije kako bi pomaci $\boldsymbol{\theta}_i - \boldsymbol{\theta}_{i-1}$ manje ovisili o vrijednostima komponenata gradijenta. Uz faktor pokretnog prosjeka $\beta \in (0..1)$ i-ta iteracija ima oblik

$$\boldsymbol{v}_i = \beta \boldsymbol{v}_{i-1} + (1-\beta) \boldsymbol{g}_i^{\odot 2}, \tag{2.59}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \eta_i (\boldsymbol{v}_i + \epsilon \mathbf{1})^{\odot -0.5} \odot \boldsymbol{g}_i. \tag{2.60}$$

Adam [61] koristi slično skaliranje i uz to oblik inercije s otporom:

$$\boldsymbol{m}_i = \beta_1 \boldsymbol{m}_{i-1} + (1-\beta_1) \boldsymbol{g}_i, \tag{2.61}$$

$$\boldsymbol{v}_i = \beta_2 \boldsymbol{v}_{i-1} + (1-\beta_2) \boldsymbol{g}_i^{\odot 2}, \tag{2.62}$$

$$\hat{\boldsymbol{m}}_i = (1-\beta_1^i)^{-1} \boldsymbol{m}_i, \tag{2.63}$$

$$\hat{\boldsymbol{v}}_i = (1-\beta_2^i)^{-1} \boldsymbol{v}_i, \tag{2.64}$$

$$\boldsymbol{\theta}_i = \boldsymbol{\theta}_{i-1} - \eta_i (\hat{\boldsymbol{v}}_i^{\odot 0.5} + \epsilon \mathbf{1})^{\odot -1} \odot \hat{\boldsymbol{m}}_i. \tag{2.65}$$

$\boldsymbol{m}_i$ je pokretni prosjek gradijenta, koji ima ulogu inercije i smanjivanja varijance, a $\boldsymbol{v}_i$ je pokretni prosjek kvadrata gradijenta po komponentama. Dijeljenje s $\hat{\boldsymbol{v}}_i^{\odot 0.5}$ čini pomake

otprilike neovisnima o apsolutnim vrijednostima gradijenata. Mala konstanta ϵ ima ulogu sprječavanja dijeljenja s premalim brojem. Za $\epsilon = 0$ skaliranje svih gradijenata istom konstantom ne utječe na pomake i rezultat optimizacije. Kad su komponente gradijenta konzistentnije kroz iteracije, apsolutne vrijednosti komponenata pomaka su bliže η_i, a u suprotnom slučaju su bliže 0. Dijeljenje s $1 - \beta_1^i$ i $1 - \beta_2^i$ služi tome da se na početku optimizacije poništi utjecaj inicijalizacije $\boldsymbol{m}_0 = \boldsymbol{v}_0 = \mathbf{0}$ na $\hat{\boldsymbol{m}}$ i $\hat{\boldsymbol{v}}_i$.

Inicijalizacija parametara

Kod dubokih modela inicijalizacija parametara ima velik utjecaj na rezultat učenja. Zbog strukture dubokih modela, različiti parametri mogu dati jednak primjerak modela – npr. redoslijed jedinica u skrivenom sloju može se permutirati (reci i stupci matrica susjednih slojeva), a da primjerak modela ostvaruje istu funkciju.

Ako se parametri inicijaliziraju konstantom, izlaz o svim jedinicama slojeva koji nisu zadnji jednako ovisi i sve jedinice dobivaju isti gradijent. Takve simetrije se razbijaju inicijalizacijom nasumičnim vrijednostima. Heuristike korištene za inicijalizaciju temelje se na ostvarivanju da varijance izlaza slojeva i gradijenata po ulazima slojeva budu otprilike budu konstantne kroz slojeve [49].

Uz pretpostavku identiteta kao aktivacijske funkcije, otprilike konstantna varijanca izlaza slojeva može se ostvariti inicijalizacijom slučajnim vrijednostima iz normalne ili unifomne razdiobe s varijancom $\frac{1}{n}$, gdje je n broj ulaza. Kao kompromis između održavanja varijanci gradijenta i izlaza slojeva u [62] predlažu varijancu $\frac{2}{n+m}$, gdje je m broj izlaza.

2.5.3 Regularizacija i poboljšavanje učenja

Regularizacija u postupak učenja uključuje dodatne informacije s ciljem poboljšavanja generalizacije.

Poticanje manjih težina

Jedan od najjednostavnijih oblika regularizacije je L^2 regularizacija koja potiče manje težine kod linearnih operacija. Komponenta gubitka općenite L^p regularizacije onda ima oblik

$$R_{\mathrm{L}^p}(\boldsymbol{\theta}) = \lambda \|\boldsymbol{\theta}\|_p^p, \tag{2.66}$$

gdje λ određuje jačinu regularizacije (ili koncentraciju apriorne raziobe parametara). U bayesovskom okviru L^p regularizacija odgovara apriornoj razdiobi

$$\mathrm{p}(\boldsymbol{\theta}) \propto \exp(-R(\boldsymbol{\theta})) = \exp(-\lambda \|\boldsymbol{\theta}\|_p^p). \tag{2.67}$$

Rastresanje podataka

Rastresanje podataka mijenjanjem (perturbiranjem) podataka za učenje unosi dodatne informacije u postupak učenja. Ono se oslanja na pretpostavku da sličnim ulaznim podacima odgovaraju slični izlazi. Kod najjednostavnijih podataka ta se pretpostavka ostvaruje dodavanjem jednostavnog šuma ulaznim podacima. Kod složenijih podataka korisne perturbacije mogu zahtijevati više domenskog znanja. Kod slikovnih podataka neke korisne perturbacije mogu biti translacija, rotacija i promjene boja.

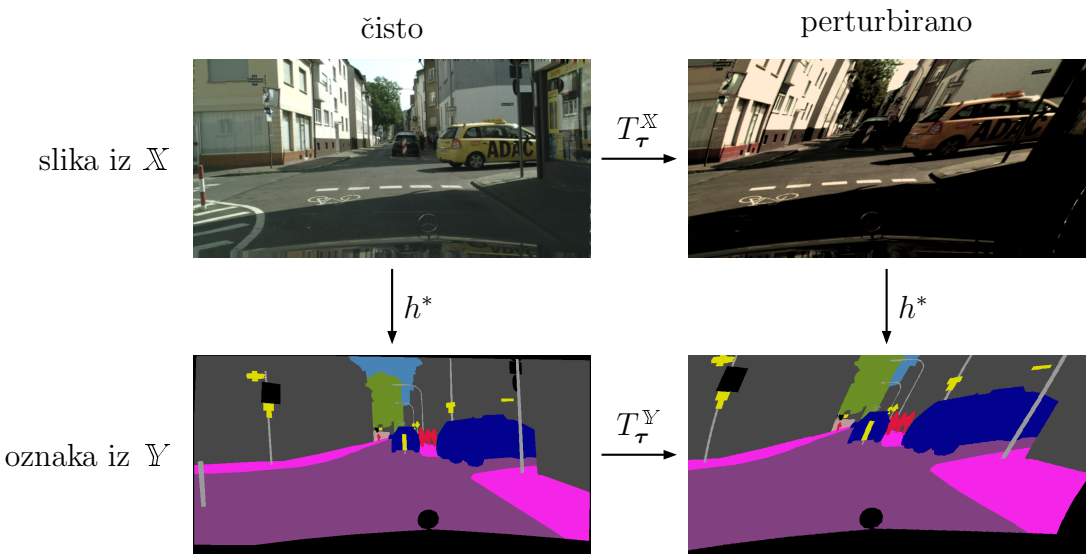
Takve perturbacije obično ne utječu na točnu klasu kod klasifikacije slika, tj. točna oznaka je **invarijantna** na perturbacije ulaza, ali mogu utjecati na točnu oznaku kod nekih drugih zadataka. Perturbacije mogu biti takve da se i oznaka mora mijenjati ako se mijenja ulaz. Npr. ako su podaci slikovni i oznake ovise o položajima objekata na slici, a slika se translatira, onda se i oznaka mora translatirati. Onda kažemo da je točna oznaka **ekvivarijantna** na perturbacije ulaza.

Općenito, podaci se mogu predstaviti u različitim **modalitetima** – načinima predstavljanja koji mogu obuhvaćati različite vrste informacija (značajke). Neki primjeri modaliteta kod slika su RGB-reprezentacije, klasifikacijske oznake na razini slike ili na razini piksela, okviri objekata, dubinske mape i tekstovi koji opisuju slike. Za neke parove modaliteta postoji funkcija koja jedne preslikava u druge, kao što je primjerak modela koji sliku preslikava u točnu klasu. Neke promjene u jednom modalitetu odgovaraju promjenama u drugom modalitetu.

Definicija 28.Perturbacijski model nad skupom modaliteta $\varDelta$ sa skupom mogućih parametara $\mathbb{T}$ je skup funkcija indeksiranih odgovarajućim modalitetima $\left\{T^{\mathbb{M}} \in \mathbb{M} \times \mathbb{T} \to \mathbb{M}\right\}_{\mathbb{M} \in \varDelta}$ s razdiobom parametara $\Pi[\boldsymbol{\tau}]$.

Perturbacija u modalitetu (s domenom) $\mathbb{M} \in \varDelta$ i parametrima $\boldsymbol{\tau} \in \mathbb{T}$ je funkcija $\boldsymbol{x} \mapsto T^{\mathbb{M}}(\boldsymbol{x}, \boldsymbol{\tau})\colon \mathbb{M} \to \mathbb{M}$, koju ćemo označavati s $T_{\boldsymbol{\tau}}^{\mathbb{M}}$. Odabir vrijednosti parametara $\boldsymbol{\tau} \in \mathbb{T}$ može dati skup indeksiran modalitetima $\left\{T_{\boldsymbol{\tau}}^{\mathbb{M}}\right\}_{\mathbb{M} \in \varDelta}$, koji predstavlja istu perturbaciju u različitim modalitetima. Slika 2.5 ilustrira perturbaciju u dvama modalitetima (slike $\mathbb{X}$ i semantičke segmentacije $\mathbb{Y}$) i ekvivarijantnost između perturbacije i idealnog primjerka modela, koji preslikava sliku u točnu segmentacijsku oznaku.

Funkcija gubitka s rastresanjem podataka za nadzirano učenje može izraziti kao očekivanje uobičajene funkcije gubitka L u primjeru $(\boldsymbol{x}, \boldsymbol{y}) \in \mathbb{D} \subset \mathbb{X} \times \mathbb{Y}$ peturbiranom po



Slika 2.5: Ilustracija primjene idealnog segmentacijskog primjerka modela h^* (odozgor prema dolje) i ekvivarijantnog perturbacijskog modela $\{T^{\mathbb{X}}, T^{\mathbb{Y}}\}$ s parametrima $\boldsymbol{\tau}$ (s lijeva na desno) na primjeru slike.

razdiobi parametara perturbacije $\mathrm{p}[\underline{\boldsymbol{\tau}}]$:

$$L_{\text{pert}}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = \mathop{\mathbf{E}}_{\underline{\boldsymbol{\tau}}} L(\boldsymbol{\theta}, T_{\boldsymbol{\tau}}^{\mathbb{X}}(\boldsymbol{x}), T_{\boldsymbol{\tau}}^{\mathbb{Y}}(\boldsymbol{y})). \tag{2.68}$$

U iteraciji učenja se očekivanje po parametrima $\underline{\boldsymbol{\tau}}$ obično procijeni na temelju jednog uzorka parametara za svaki primjer iz mini-grupe.

Normalizacija po grupi

Normalizacija po grupi (engl. *batch normalization*) [63] je postupak koji omogućuje brže učenje, smanuje osjetljivost na hiperparametre i ima regularizacijski učinak. Normalizacija po grupi tijekom učenja aktivacije normalizira sa srednjom vrijednošću i standardnom devijacijom po dimenziji mini-grupe i prostornim dimenzijama. Obično dolazi iza sloja afine transformacije, kao što je konvolucijski sloj.

Neka je $\boldsymbol{X} \in \mathbb{R}^{N \times H \times W \times C}$ reprezentacija neke mini-grupe veličine N iza nekog konvolucijskog sloja (opisanog u odjeljku 2.5.4) s C izlaznih kanala i prostornih dimenzija $H \times W$. Normalizacija po grupi $\mathrm{BN}_{\mathrm{2D}}$ djeluje kroz cijelu mini-grupu na svaki kanal zasebno:

$$\mathrm{BN}_{\mathrm{2D}}(\boldsymbol{X})_{[n,h,w,c]} \coloneqq \frac{\boldsymbol{X}_{[n,h,w,c]} - \mathrm{bm}(\boldsymbol{X})_{[c]}}{\left(\mathrm{bv}(\boldsymbol{X})_{[c]} + \epsilon\right)^{0.5}}, \text{ gdje} \tag{2.69}$$

$$\mathrm{bm}(\boldsymbol{X})_{[c]} \coloneqq \mathop{\mathbf{E}}_{n,h,w} \boldsymbol{X}_{[n,h,w,c]} \text{ i} \tag{2.70}$$

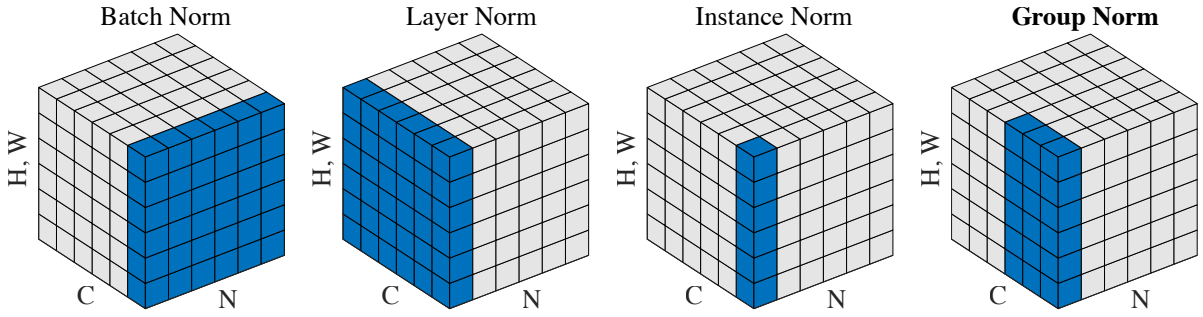
$$\mathrm{bv}(\boldsymbol{X})_{[c]} \coloneqq \mathop{\mathbf{E}}_{n,h,w} (\boldsymbol{X}_{[n,h,w,c]} - \mathrm{bm}(\boldsymbol{X}))^2, \tag{2.71}$$

gdje $\mathbf{E}_{n,h,w}$ označava aritmetičku sredinu po indeksima $(n,h,w) \in \{1..N\} \times \{1..H\} \times \{1..W\}$, a konstanta ϵ služi za sprječavanje dijeljenja s premalim brojem.

Kako se ne bi izgubila ekspresivnost, bitno je nakon sloja normalizacije po grupi dodati afini sloj skaliranja i pomaka po kanalima (čiji se parametri uče). Kod ispitivanja normalizacija po grupi koristi procjenu populacijskih statistika (sredine i varijance) umjesto statistika mini-grupe.

Nije jasno kako točno normalizacija po grupi pomaže. Jedan problem u optimizaciji dubokih modela je da se kasnije reprezentacija nekog kasnijeg sloja može puno promijeniti nakon promjena u parametrima ranijih slojeva. Normalizacija po grupi je predložena s ciljem ublažavanje tih promjena kako bi se parametri kasnijih slojeva morali manje prilagođavati – sredina i varijanca značajki ovise samo o parametrima pojedinog afinog sloja skaliranja i pomaka po kanalima iza normalizacije po grupi. Međutim, kasniji empirijski rezultati pokazuju da normalizacija po grupi djeluje i kad se namjerno dodaje nestabilnost u reprezentacije i da čini funkciju gubitka glađom [64]

Predloženi su i drugi oblici normalizacije koji imaju sličan učinak, ali kod njih nema nekad nepoželjnog međusobnog utjecaja primjera mini-grupe i jednako se ponašaju tijekom učenja i ispitivanja. Neki od njih su normalizacija težina (engl. *weight normalization*), normalizacija sloja (engl. *layer normalization*), normalizacija po kanalima (engl. *instance normalization*) i normalizacija po grupama kanala (engl. *group normalization*) [65]. Slika 2.6 ilustrira po kakvim isječcima slikovnih reprezentacija nezavisno djeluju različite operacije normalizacije.



Slika 2.6: Ilustracija toga po kakvim isječcima slikovnih reprezentacija nezavisno djeluju različite operacije normalizacije. N označava dimenziju mini-grupe, H, W označava prostorne dimenzije, a C označava kanale. Izvor: [65].

Normalizacija i L^2 regularizacija

Zbog neovisnosti izlaza normalizacijskog sloja o normi težina afinog sloja koji mu prethodi, L^2 regularizacija nema regularizacijski utjecaj na afine slojeve, ali ima na efektivni korak učenja. Skaliranje težina afinog sloja pozitivnim faktorom ne utječe na izlaz normalizacijskog sloja iza njega, ali ima na omjer norme gradijenta gubitka i norme težina:

efektivni korak učenja za glavni gubitak je obrnuto proporcionalan normi ili kvadratu norme težina, ovisno o optimizacijskom algoritmu [66, 67]. Smanjivanjem norme težina L^2 regularizacija održava visok efektivni korak učenja.

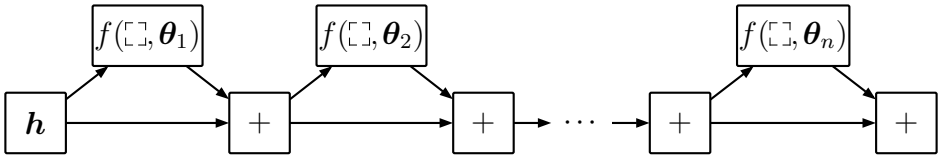
Preskočne veze

Preskočne veze [68, 69] su arhitekturna komponenta koja kasnijim slojevima daje izravniji pristup izlazima ranijih slojeva u modelu. One olakšavaju optimizaciju i poboljšavaju generalizaciju dubokih modela. Jedan učinak preskočnih veza je lakše učenje propuštanja informacija i mogućnost višestrukog korištenja značajki – propuštanje značajki se ne mora učiti ili je lakše za naučiti. Preskočne veze daju i veću stabilnost reprezentacija tijekom učenja. Zbog toga se parametri moraju manje prilagođavati promjenama ranijih reprezentacija. Preskočne veze mogu pomoći i u sprječavanju problema isčezavajućih ili eksplodirajućih gradijenata [70].

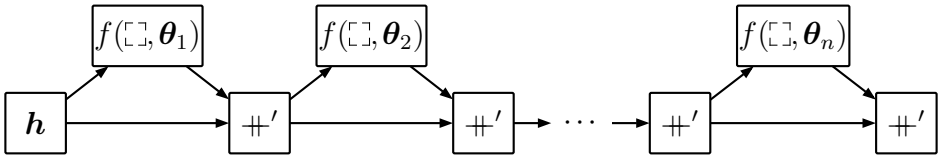
Najuspješniji modeli u računalnom vidu su **rezidualni modeli**, kao ResNet [68, 71], koji izlazu bloka slojeva pribrajaju njegov ulaz. Osnovna jedinica rezidualnog modela na temelju svog ulaza $\boldsymbol{h}_i$ i svojih parametara $\boldsymbol{\theta}_i$ računa izlaz:

$$\boldsymbol{h}_{i+1} = \boldsymbol{h}_i + f(\boldsymbol{h}_i, \boldsymbol{\theta}_i). \tag{2.72}$$

Funkciju f ćemo zvati **rezidualna funkcija**, a $f(\boldsymbol{h}_i, \boldsymbol{\theta}_i)$ rezidual. Kod ResNeta je f niz u kojem se više puta ponavlja normalizaciju po grupi, ReLU i konvolucijski sloj. Strukturu rezidualnog modela imaju i transformeri [72], ali uz drugačije reziduale. Gusto povezani modeli, kao DenseNet [69], umjesto zbrajanja koriste konkatenaciju. Slika 2.7 ilustrira rezidualne i gusto povezane modele.



(a) Osnovna struktura rezidualnog modela.



(b) Osnovna struktura gusto povezanog modela.

Slika 2.7: Osnovne strukture rezidualnih modela, koji koriste zbrajanje (+) i gusto povezanih modela, koji koriste konkatenaciju po kanalima ($+'$). Kod ResNeta i DenseNeta je rezidualne funkcija f niz koji sadrži slojeve normalizacije po grupi, ReLU i konvolucijske slojeve. Slike su preuzete iz [41].

Empirijski rezultati pokazuju da rezidualni modeli imaju glađu funkciju pogreške [73] i da se povećavanjem dubine rezidualnog modela poboljšava generalizacija [71]. Rezidualni modeli mogu se izraziti i kao ansambl plićih modela s dijeljenim parametrima [74].

DenseNet i ResNet su konvolucijski modeli koji se primjenjuju u računalnom vidu. Propuštanje ranijih informacija je korisno i u povratnim modelima kao što je LSTM [70], gdje se izlaz iz prethodnog koraka propušta uz množenje s faktorom iz $(0\,..1)$. Transformerski modeli [75], koji su trenutno stanje tehnike i u jezičnom modeliranju i u obradi slika, također koriste rezidualnu strukturu. Pri tome se u ulozi rezidualne funkcije obično izmjenjuju slojevi globalne pažnje (engl. *multi-head self-attention*) i uobičajeni blokovi s više afinih slojeva [72].

Preskočne veze od ranijih značajki na višoj rezoluciji prema kasnijim značajkama korisne su kod guste predikcije (odjeljak 3.1), gdje je bitna prostorna preciznost.

2.5.4 Konvolucijski modeli

Ovaj odjeljak se temelji na odjeljku o konvolucijskim modelima u [41]. **Konvolucijski modeli** su modeli koji koriste operaciju konvolucije. Koriste pretpostavku ekvivarijantnosti na translaciju po nekim dimenzijama ulaza i posebno se uspješno primjenjuju na zadacima u vezi slika.

Definicija 29.Konvolucija funkcija f i g iz $\mathbb{Z} \to \mathbb{R}$ je operacija $*$ takva da za svaki $t \in \mathbb{Z}$

$$(f * g)(t) \coloneqq \sum_{\tau \in \mathbb{Z}} f(\tau) g(t - \tau). \tag{2.73}$$

Neka svojstva konvolucije su

- komutativnost: $f * g = g * f$,
- distributivnost zbrajanja: $(f + g) * h = f * h + g * h$,
- translacijska ekvivarijantnost: $((u \mapsto f(u + d)) * g)(t) = (f * g)(t + d)$,

U frekvencijskoj domeni konvolucija odgovara umnošku: $\mathrm{F}[f * g] = \mathrm{F}[f]\mathrm{F}[g]$, gdje F označava odgovarajuću Fourierovu transformaciju.

Konvolucija se može na više načina poopćiti na funkcije iz $\mathbb{Z}^m \to \mathbb{R}^n$, kojima je kodomena vektorski prostor. Jedan način je konvolucija po svakoj komponenti zasebno:

$$(f *_{\mathrm{p}} g)(\boldsymbol{t}) \coloneqq \sum_{\boldsymbol{\tau} \in \mathbb{Z}^m} f(\boldsymbol{\tau}) \odot g(\boldsymbol{t} - \boldsymbol{\tau}). \tag{2.74}$$

Drugi način koristi skalarno množenje:

$$(f *_{\mathrm{s}} g)(\boldsymbol{t}) \coloneqq \sum_{\boldsymbol{\tau} \in \mathbb{Z}^m} \langle f(\boldsymbol{\tau}) | g(\boldsymbol{t} - \boldsymbol{\tau}) \rangle, \tag{2.75}$$

pri čemu $f *_{\mathrm{s}} g \colon \mathbb{Z}^m \to \mathbb{R}$. Ovakvu konvoluciju ćemo zvati m**-dimenzionalne konvolucija**.

m-dimenzionalna konvolucija može se definirati i za nizove $\boldsymbol{x}$ i $\boldsymbol{w}$ s elementima $\boldsymbol{x}_{[\boldsymbol{t},:]} = f(\boldsymbol{t})$ i $\boldsymbol{w}_{[\boldsymbol{t},:]} = g(\boldsymbol{t})$:

$$(\boldsymbol{x} *_{\mathrm{s}} \boldsymbol{w})_{[\boldsymbol{t}]} = \sum_{\boldsymbol{t}-\boldsymbol{\tau} \in \mathbb{I}} \left\langle \boldsymbol{x}_{[\boldsymbol{\tau},:]} \middle| \boldsymbol{w}_{[\boldsymbol{t}-\boldsymbol{\tau},:]} \right\rangle. \tag{2.76}$$

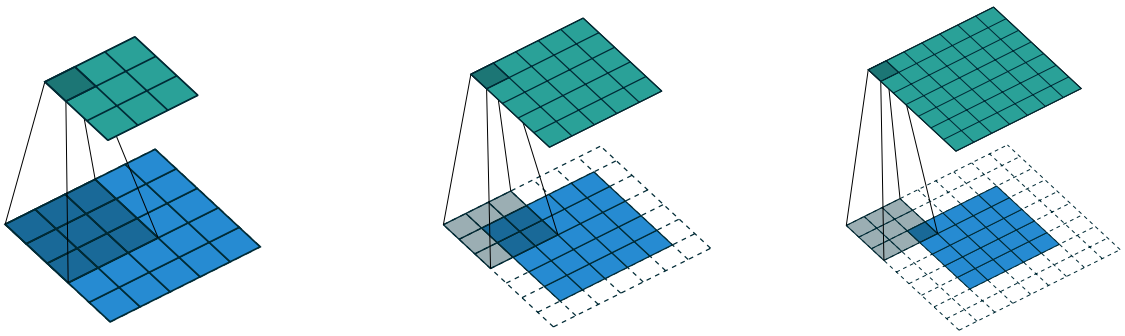
Oznaka $[\boldsymbol{t},:]$ u indeksu je kraći zapis za $\left[\boldsymbol{t}_{[1]}, .., \boldsymbol{t}_{[m]}, :\right]$. Kažemo da su sve osim zadnje dimenzije takvih nizova prostorne dimenzije. Ako $m = 2$, matrice oblika $\boldsymbol{x}_{[:,:,c]}$ se nazivaju mape značajki ili kanali, a vektori oblika $\boldsymbol{x}_{[\boldsymbol{t},:]}$ se nazivaju vektori značajki.

Općenita konvolucijska operacije u konvolucijskom modelu (konvolucijski sloj) obavlja K**-struku** m**-dimenzionalnu konvoluciju** između ulaznog niza $\boldsymbol{x} \in \mathbb{R}^{d_1 \times \cdots \times d_m \times C}$ i K manjih **konvolucijskih jezgri** $\boldsymbol{w}_k \in \mathbb{R}^{k_1 \times \cdots \times k_m \times C}$, koja se sastoji od K konvolucija kao u izrazu 2.76. Neka su radi jednostavnosti prostorne dimenzije jezgre $k_1, .., k_m$ neparne. Neka su vektori jezgre koji se trebaju množiti s vektorima značajki indeksirani skupom $\mathbb{I} = \left\{-\frac{k_1-1}{2} .. \frac{k_1-1}{2}\right\} \times \cdots \times \left\{-\frac{k_m-1}{2} .. \frac{k_m-1}{2}\right\}$, tako da sredina prostornih dimenzija jezgre bude $\mathbf{0}_m$. Izračun rezultata $\boldsymbol{z} \in \mathbb{R}^{d_1 \times \cdots \times d_m \times K}$ K-struke m-dimenzionalne konvolucija može se izraziti ovako:

$$\boldsymbol{z}_{[\boldsymbol{i},k]} = (\boldsymbol{x} *_{\mathrm{s}} \boldsymbol{w}_k)_{[\boldsymbol{i}]} = \sum_{\boldsymbol{i}-\boldsymbol{j} \in \mathbb{I}} \left\langle \boldsymbol{x}_{[\boldsymbol{j},:]} \middle| \boldsymbol{w}_{k[\boldsymbol{i}-\boldsymbol{j},:]} \right\rangle, \tag{2.77}$$

uz $\boldsymbol{i} \in \{0..d_1\} \times \cdots \times \{0..d_m\}$ i $k \in \{1..K\}$.

Kako bi izlaz imao iste prostorne dimenzije kao ulaz, i-ta dimenzija ulaza se obično nadopunjava s obje strane s $\frac{k_i-1}{2}$ uz pretpostavku neparnih prostornih dimenzija jezgre. Slika 2.8 ilustrira dopunjavanje na primjeru jednostruke dvodimenzionalne konvolucije.



(a) Bez dopunjavanja.

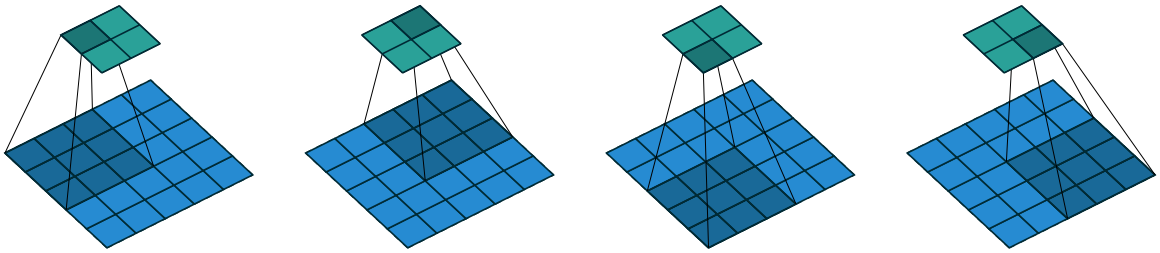
(b) Dopunjavanje takvo da izlaz ima iste dimenzije.

(c) Potpuno dopunjavanje.

Slika 2.8: Dopunjavanje kod dvodimenzionalne konvolucije. Slika se temelji na slikama iz [76].

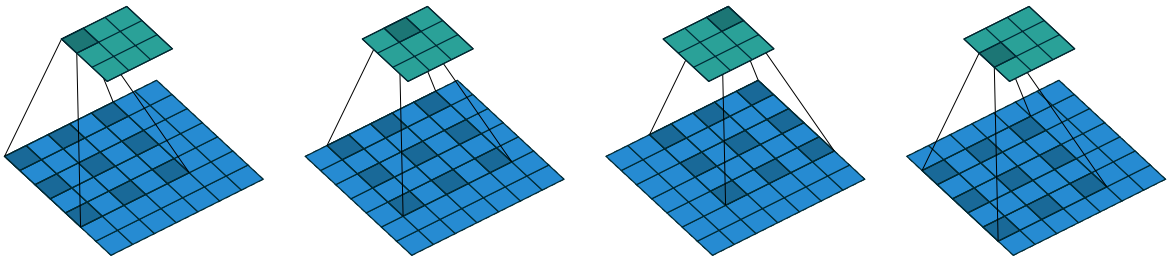
Povećavanje učinkovitosti i receptivnog polja konvolucijskih modela

Konvolucijski modeli često koriste još neke izmjene konvolucije za ostvarivanje veće učinkovitosti. Kod konvolucije s (izlaznim) korakom $s > 1$ jezgra preskače po s mjesta u prostornoj dimenziji. Time se oko s puta smanjuje svaka prostorna dimenzija izlaza i povećava **receptivno polje** sljedećih slojeva, tj. povećava se broj piksela ulazne slike o kojima ovisi pojedini piksel izlaza. Slika 2.9 ilustrira konvoluciju s korakom 2 po svim prostornim dimenzijama.



Slika 2.9: Konvolucija s korakom 2. Slike su preuzete iz [76].

Receptivno polje konvolucijskog sloja može se povećati bez povećavanja količine računanja dilatacijom (širenjem) jezgre kao na slici 2.10. Takva konvolucija je ekvivalentna konvoluciji s većom jezgrom kod koje se svaki drugi redak i stupac sastoje od nula.

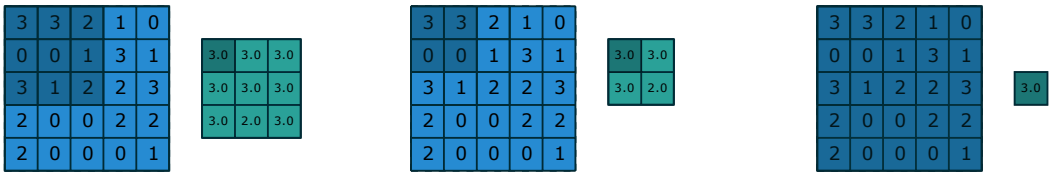


Slika 2.10: Konvolucija s dilacijom 1. Slike su preuzete iz [76].

Za smanjivanje prostornih dimenzija mogu se koristiti operacije sažimanja. Operacije sažimanja nemaju parametre koji se uče i obično djeluju po svakom kanalu zasebno. Slika 2.11 ilustrira česte operacije sažimanja. Sažimanje srednjom vrijednošću usrednjava vrijednosti unutar okna prostornih dimenzija $s \times s$, obično uz korak s po prostornim dimenzijama. Klasifikacijski modeli često na kraju agregiraju prostorne dimenzije globalnim sažimanjem srednjom vrijednošću.

Za smanjivanje ili povećavanje prostornih dimenzija koriste se i postupci interpolacije[‡] kao što su bilinearna interpolacija i interpolacija najbližim susjedom.

[‡] `https://en.wikipedia.org/wiki/Multivariate_interpolation`



(a) $k = 3$, $s = 1$. (b) $k = 2$, $s = 2$. (c) Globalno sažimanje.

Slika 2.11: Dvodimenzionalno sažimanje s jezgrama prostornih dimenzija $k \times k$ i korakom s po prostornim dimenzijama. Slike su preuzete iz [76] i prilagođene.

Poglavlje 3

Pregled literature

Glavna tema ovog rada nalazi se na presjeku polunadziranog učenja i guste predikcije. Rad ispituje polunadzirane algoritme prilagođene za semantičku segmentaciju i uključuje eksperimente s različitim modelima za gustu predikciju.

3.1 Gusta predikcija

Modeli za klasifikaciju cijelih slika obično ostvaruju učinkovitost, invarijantnost na lokalne pomake i uključivanje kontekstnih informacija postupnim prostornim smanjivanjem konvolucijskih reprezentacija i korištenjem operacija globalnog prostornog sažimanja. Kod guste predikcije potrebna je i prostorna preciznost, pa tražimo kompromis između učinkovitosti i kvalitete značajki na visokoj rezoluciji. Mnoge arhitekture koriste klasifikacijsku okosnicu i njoj nadodaju "dekoder" koji vraća prostornu rezoluciju uz korištenje značajki iz okosnice na različitim rezolucijama.

FCN-8s [77] je jedan od ranijih modela koji kombiniraju kasnije značajke niže rezolucije, koje sadrže više informacija o kontekstu, s ranijim značajkama više rezolucije, koje sadrže više prostornih detalja. Značajke koje se združuju održavaju broj kanala značajki niže rezolucije, koji je jednak broju klasa K. Značajke više rezolucije se prilagođavaju K-strukom konvolucijom 1×1 i zbrajaju s kasnijim značajkama kojima se rezolucija poveća K-strukom transponiranom konvolucijom. Rezultat se još dvaput tako kombinira sa značajkama viših rezolucija i logiti se dobiju još jednom transponiranom konvolucijom. UNet [78] poboljšava performansu korištenjem ekspresivnijeg, simetričnog dekodera, koji na početku svakog koraka povećava rezoluciju značajki transponiranom konvolucijom, konkatenira ih s odgovarajućim značajkama iz koderskog dijela i primjenjuje konvolucijske slojeve praćene ReLU-om.

Daljni radovi poboljšavaju učinkovitost računski manje zahtjevnim dekoderima [29, 79], modulima za agregaciju konteksta kao što je prostorno piramidalno sažimanje (engl.

spatial pyramid pooling) [80] i ansambliranjem predikcija po različitim rezolucijama (engl. *multi-scale inference*) [79, 81].

Jedna od popularnijih arhitektura, DeepLab [82], povećava receptivno polje pomoću dilatiranih konvolucija. Ranije verzije te arhitekture predlagale su poboljšanje prostornih detalja naknadnim usklađivanjem predikcija i ulazne slike primjenom uvjetnih slučajnih polja [83]. Arhitektura HRNet [84] održava značajke na punoj rezoluciji kroz cijeli model i postupno uvodi paralelne grane na nižim rezolucijama koje razmjenjuju informacije.

Semantička segmentacija ima puno koristi od predtreniranja okosnice na ImageNetu [29, 82] i od samonadziranog predtreniranja [85].

Probabilistički primjerak modela za semantičku segmentaciju bi idealno trebao dati razdiobu preko svih mogućih segmentacija zadane slike: $h_{\boldsymbol{\theta}}(\boldsymbol{x}) = \mathrm{P}[\underline{\boldsymbol{y}} \mid \boldsymbol{x}, \boldsymbol{\theta}]$. Međutim, u praksi je model često ostvaren tako da za svaki piksel nezavisno daje razdiobu preko mogućih razreda: $h_{\boldsymbol{\theta}}(\boldsymbol{x})_{[i,j]} = \mathrm{P}[\underline{\boldsymbol{y}}_{[i,j]} \mid \boldsymbol{x}, \boldsymbol{\theta}]$. U tom slučaju radi pojednostavljenja funkcija gubitka nezavisno ocjenjuje predikcije u pojedinim pikselima, što odgovara (krivoj) pretpostavci uvjetne nezavisnosti piksela:

$$\mathrm{P}(\underline{\boldsymbol{y}} = \boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = \prod_{i,j} \mathrm{P}(\underline{\boldsymbol{y}}_{[i,j]} = \boldsymbol{y}_{[i,j]} \mid \boldsymbol{x}, \boldsymbol{\theta}). \tag{3.1}$$

Uz tu pretpostavku gubitak negativne log-izglednosti ima ovakav oblik:

$$L_{\mathrm{seg}}(\boldsymbol{\theta}, \boldsymbol{x}, \boldsymbol{y}) = -\ln \mathrm{P}(\boldsymbol{y} \mid \boldsymbol{x}, \boldsymbol{\theta}) = -\sum_{i,j} \ln \mathrm{P}(\underline{\boldsymbol{y}}_{[i,j]} = \boldsymbol{y}_{[i,j]} \mid \boldsymbol{x}, \boldsymbol{\theta}). \tag{3.2}$$

Nedavno su predloženi pristupi koji razdvajaju raspoznavanje objekata i drugih semantički povezanih područja od njihove klasifikacije [86, 87, 88]. Predikcija takvih modela može se predstaviti skupom binarnih maski i njima pridruženih klasifikacija. Skup maski ovisnih o slici uči se da pokriva svaki segment i da se svaki segment točno klasificira. Takvi pristupi su prikladni i za segmentaciju primjeraka i panoptičku segmentaciju [89].

3.2 Perturbiranje podataka

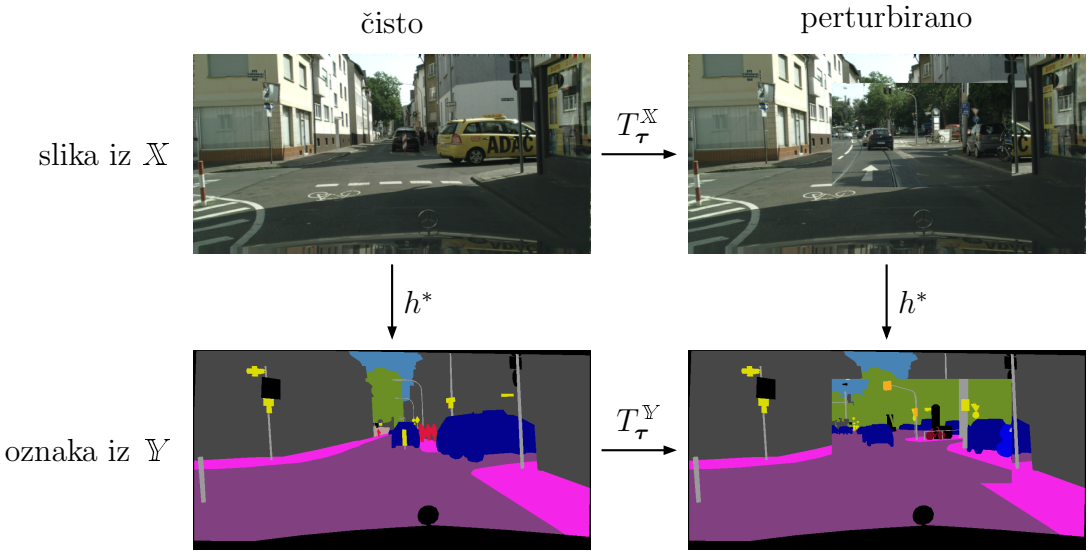
Nasumično perturbiranje (mijenjanje) primjera za učenje je oblik regularizacije koji daje dodatno znanje o tome kako izmjene ulaza trebaju utjecati na izlaz. Često je najlakše ostvariti pertubacije koje ne bi trebale utjecati na izlaz, tj. na koje bi izlaz trebao biti invarijantan. Takve perturacije su često korisne kod klasifikacije.

Metode AutoAugment [91] i RandAugment [92] predlažu složene perturbacije koje se sastoje od više komponenata i istražuju načine optimizacije hiperparametara perturbacija. Obje metode u komponente takvih perturbacija uključuju fotometrijske transforma-

cije (promjene kontrasta, svjetline, zasićenosti, invertiranje vrijendosti, kvantizacija boja, solarizacija), geometrijske transformacije (rotacija, translacija, smicanje), izjednačavanje histograma i promjenu oštrine. TrivialAugment [93] pokazuje da umjesto primjena više komponenata takve perturbacije zajedno, slučajni odabir samo jedne za svaki primjer može imati podjednak regularizacijski učinak.

Klasifikacijsku generalizaciju može poboljšati i konveksna interpolacija između parova slika i odgovarajućih parova oznaka, MixUp [94].

Za razliku od klasifikacije cijelih slika, gdje je točna oznaka često invarijantna na perturbacije (MixUp je iznimka), kod guste predikcije neke od tih perturbacija utječu na izlaz. U njih spadaju geometrijska izobličenja [21] i ljepljenje komada slike (CutMix) [28, 95]. Kod geometrijskih izobličenja se predikcija treba jednako deformirati kao slika npr. kao na slici 2.5. Nadalje, kod CutMix-a se u predikciju treba zalijepiti dio predikcije koji odgovara komadu druge slike koji je izrezan, kao na slici 3.1



Slika 3.1: Ilustracija primjene idealnog segmentacijskog primjerka modela h^* (odozgor prema dolje) i perturbacije CutMixa (s lijeva na desno), koja u sliku lijepi komad druge slike (koja nije prikazana). Uz ovakvu formalizaciju, parametri perturbacije $\boldsymbol{\tau}$ moraju sadržavati nalijepljeni komad slike i koordinate na koje se lijepi.

3.3 Postupci polunadziranog učenja

Postupci polunadziranog učenja često se oslanjaju na neke od sljedećih pretpostavki o razdiobi podataka [1]:

1. slični ulazi u područjima visoke gustoće odgovaraju sličnim izlazima (pretpostavka glatkoće),

2.ulazi čine nakupine (grupe) odvojene područjima niske gustoće i ulazi unutar grupa odgovaraju sličnim izlazima (pretpostavka nakupina),

3.podaci otprilike leže u niskodimenzionalnim mnogostrukostima (pretpostavka mnogostrukosti).

Algoritmi polunadziranog učenja na različite načine uključuju takve induktivne pristranosti ili i dodatno znanje koje je korisno za iskorištavanje informacija iz neoznačenih primjera ovisno o zadatku.

U nastavku ovog odjeljka slijede opisi nekih algoritama i opažanja iz literature, pri čemu se većina empirijskih opažanja temelji na istraživanju polunadziranog učenja na zadatku klasifikacije slika.

Minimizacija entropije [96] potiče visoku uvjerenost predikcija u neoznačenim ulazima. Uz pretpostavku nakupina, minimizacija entropije uz dodatnu induktivnu pristranost koja potiče glatkoću daje prednost granicama razreda koje su u područjima niske gustoće. **Učenje s pseudooznakama** (engl. *pseudo-label training, self-training, self-labeling*) [97, 98, 99] sastoji se od 2 ili više ponavljanja postupka učenja prema sljedećem algoritmu:

1.Uči se primjerak modela na označenim podacima.

2.Naučeni primjerak modela dobiva ulogu **učitelja** za dodjelu **pseudooznaka** neoznačenim primjerima.

3.Novi primjerak modela, učenik, uči se na kombinaciji označenih i pseudooznačenih podataka.

4.Postupak se može ponavljati od 2. koraka sa zadnjim učenikom u ulozi učitelja.

Kako bi bile korisne, pseudooznake moraju imati smanjenu entropiju u odnosu na predikcije. Predikcije učitelja se tipično izoštravamo tako da im smanjimo entropiju na 0. Drugim riječima, pseudooznaka odgovara najvjerojatnijem razredu. Osnovni oblici učenja s pseudooznakama često ne postižu kompetitivnu generalizacijsku performansu sami po sebi [100], ali mogu biti jako učinkoviti uz dodatna poboljšanja [101]. Pri tome tipično bude korisno izostaviti učiteljeve predikcije s niskom uvjerenošću i primjenjivati još neke načine obrade predikcija [97, 98, 102]. Neki autori [102] koriste riječ pseudooznaka (engl. *pseudo-label*) i za obrađenu učiteljevu predikciju kod jednosmjerne konzistencije, gdje se učitelj mijenja zajedno s učenikom, a samo se u učenikovoj grani računa gradijent po parametrima. U ovom radu izraz **pseudooznaka** se odnosi na obrađene ili neobrađene predikcije učitelja koji se ne mijenja tijekom učenja.

Pristupi **konzistencijskog učenja** se od učenja s pseudooznakama razlikuju po tome što učenje nema više faza i nema fiksnog učitelja, nego se zajedno s nadziranim gubitkom potiču konzistencija predikcija istog primjerka modela preko različitih inačica ulaza ili preko različitih primjeraka modela koji se zajedno optimiraju. Učinkovitost postupaka dobivanja različitih inačica ulaza (perturbacija ulaza) ovisi o tome koliko su informacije koje

perturbacije daje postupku učenja kroz konzistencijski gubitak korisne za generalizaciju na glavnom zadatku.

Kod zadataka kao što je klasifikacija cijelih slika točna predikcija je invarijantna na raznolike perturbacije ulaza, ali na zadacima guste predikcije mnoge perturbacije ulaza mijenjaju i točnu predikciju. Među njih spadaju perturbacije na koje je točna predikcija ekvivarijantna. Primjeri takvih perturbacija kod guste predikcije su geometrijske perturbacije, izrezivanje i lijepljenje, a i kompozicije više takvih perturbacija i kompozicije s perturbacijama na koje je točna predikcija invarijantna. U tom slučaju konzistencijski gubitak može imati oblik sličan izrazu (2.68), ali umjesto oznake treba biti predikcija u čistom ulazu:

$$L_{\text{cons-cp}}(\boldsymbol{\theta}, \boldsymbol{x}) = \mathop{\mathbf{E}}_{\boldsymbol{\tau}} D(h_{\boldsymbol{\theta}}(T_{\boldsymbol{\tau}}^{\mathbb{X}}(\boldsymbol{x})), T_{\boldsymbol{\tau}}^{\mathbb{P}_{\mathbb{Y}}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}))), \tag{3.3}$$

gdje je $T_{\boldsymbol{\tau}}^{\mathbb{X}}$ perturbacija u ulaznom prostoru, $T_{\boldsymbol{\tau}}^{\mathbb{P}_{\mathbb{Y}}}$ odgovarajuća perturbacija nad predikcijama ($\mathbb{P}_{\mathbb{Y}}$ označava skup razdioba nad mogućim oznakama iz $\mathbb{Y}$), a D je divergencija za usporedbu prediktiranih razdioba. Uključivanje znanja o ekvivarijatnosti je proučavano za razumijevanje i učenje korisnih slikovnih reprezentacija [25, 103] i poboljšavanje guste predikcije [104, 105, 106].

Vremensko ansambliranje [22] potiče konzistenciju između trenutne predikcije i pokretnog prosjeka prošlih predikcija. Isti rad pokazuje prednost vremenskog ansambliranja s obzirom na jednostavnu dvosmjernu konzistenciju s jednim čistim ulazom, koju nazivaju Π-modelom. Usrednjeni učitelj (engl. *Mean Teacher*) [5] umjesto predikcija usrednjuje parametre učitelja, tj. potiče konzistenciju s učiteljem čiji parametri su eksponencijalni pokretni prosjek učenikovih parametara.

Sličnost latentnih reprezentacija iste klase može se postići učenjem šetnji koje počinju u označenom primjeru, prelaze preko neoznačenog primjera i moraju se vratiti u označeni primjer koji ima istu oznaku u bipartitnom grafu između označenih i neoznačenih primjera [107]. Takav algoritam maksimizira vjerojatnost ispravnog puta. Vjerojatnost puta koji polazi iz primjera s indeksom i definira kao umnožak vjerojatnosti uzorkovanja označenog primjera i dviju uvjetnih vjerojatnosti prijelaza, svake za jedan korak puta. Uvjetnu vjerojatnost prijelaza iz primjera s indeksom i u primjer na drugoj strani grafa s indeksom j definira kao $\exp(h_{\boldsymbol{\theta}}(\boldsymbol{x})^{\mathsf{T}} h_{\boldsymbol{\theta}}(\boldsymbol{x}'))/\sum_{\boldsymbol{x}''} \exp(h_{\boldsymbol{\theta}}(\boldsymbol{x})^{\mathsf{T}} h_{\boldsymbol{\theta}}(\boldsymbol{x}''))$.

MixMatch [108] potiče konzistenciju između većeg broja predikcija u različitim slabim perturbacijama tako da kao konzistencijsku ciljnu oznaku koristi prosječnu predikciju više učitelja s umanjenom entropijom i još primjenjuje MixUp nad združenom minigrupom označenih i perturbiranih neoznačenih primjera s pridruženim konzistencijskim oznakama. *Deep co-training* [20] daje komplementarne primjerke modela poticanjem me-

đusobne konzistencije dok se svaki primjerak modela uči na neprijateljskim primjerima drugog primjerka modela.

Konzistencijski gubici mogu imati trivijalna (urušena) rješenja, kod kojih primjerak modela sve ili podskup ulaza preslikava u isti izlaz. Ovaj problem najčešće se javlja kod samonadziranog učenja jer tamo ne koristimo nadzirani gubitak [109]. Ipak, ovaj problem može se pojaviti i kod polunadziranog učenja, posebno ako se primjeri koji sudjeluju u konzistencijskom gubitku zbog jakih perturbacija dovoljno razlikuju od primjera koji sudjeluju u nadziranom gubitku.

Nedavno istraživanje samonadziranog učenja [109, 110, 111] pokazuje da učenje s konzistencijskim gubitkom bez eksplicitnih komponenata gubitka za sprječavanje trivijalnih rješenja i dalje može pronalaziti korisne (netrivijalne) reprezentacije pomoću algoritamskih komponenata kao što su normalizacija značajki i jednosmjerna konzistencija.

Učenje s virtualnim neprijateljskim primjerima (VAT, engl. *virtual adversarial training*) [6] potiče jednosmjernu konzistenciju između predikcija u neprijateljski perturbiranim ulazima i predikcija u originalnim ulazima. Predlažu perturbacije koje maksimiziraju kvadratnu aproksimaciju divergencije predikcije u maloj L^2-kugli oko ulaza. Dodatno poticanje visoke entropije može poboljšati performansu na klasifikaciji slika [96]. UDA (engl. *unsupervised data augmentation*) [7] isto koristi jednosmjernu konzistenciju, pri čemu se na ulaze modela dovode slučajne perturbacije motode RandAugment. FixMatch [112] pokazuje da "izoštravanje" i izostavljanje manje uvjerenijih učiteljevih predikcija mogu biti korisni u jednosmjernoj konzistenciji, pri čemu isto koristi RandAugment.

Ovaj rad prilagođava konzistencijsko učenje na zadatak semantičke segmentacije i istražuje različite formulacije konzistencijskog učenja s dvije grane. Radi jednostavnosti eksperimenti u ovom radu uključuju "izoštravanje" i izostavljanje manje uvjerenih predikcija učitelja samo kod reproduciranja algoritama iz literature.

3.4 Polunadzirana semantička segmentacija

Neki pristupi prilagodili su okvir generativnih suparničkih modela (GAN, engl. *generative adversarial networks*) za učenje značajki korisnih za polunadzirano učenje. Kod SGAN-a (engl. *semi-superised GAN*) klasifikator ima ulogu diskriminatora kojemu je zadatak razlikovati stvarne podatke (označene i neoznačene) od onih koje daje generator [15]. Takav pristup je prilagođen semantičkoj segmentaciji tako da diskriminator ujedno bude i segmentacijski model [8]. KE-GAN [113] dodatno potiče konzistenciju susjednih predikcija prema sličnosti oznaka izvedenoj prema tekstovnom korpusu MIT ConceptNet. AdvSemSeg [9] stavlja segmentacijski model u ulogu generatora, a diskriminator mora razlikovati stvarne oznake od predikcija. Diskriminator se još koristi i za odabir predikcija za učenje s

pseudooznakama. s4GAN + MLMT [10] još dodatno obrađuje guste predikcije na temelju respoznavanje razreda koje pronalazi klasifikator cijele slike učen usrednjenim učiteljem [5]. Autori primjećuju da takav klasifikator nije prikladan za skupove podataka kao što je Cityscapes, gdje se u skoro svakoj slici pojavljuje velik broj razreda.

Drugi nedavni radovi bave se pseudooznačavanjem u kontekstu guste predikcije [114, 115, 116]. Zhu et al. [114] opaža prednosti čvrstih pseudooznaka za segmentaciju, na što upućuje i druga literatura, i ne opažaju poboljšanja nakon više od 1 faze učenja s pseudooznakama. Pseudooznake se mogu značajno poboljšati agregiranjem učiteljevih predikcija po različitim veličinama ulaza (engl. *multi-scale inference*) i zrcaljenjima lijevo-desno.

Jedan nedavni pristup potiče konzistenciju između predikcija redundantih dekodera sa zašumljenim međurezultatima [117]. Za medicinsku segmentaciju nedavni rad [21] predlaže dvosmjernu konzistenciju s perturbiranjem obaju ulaza geometrijskim izobličavanjem. Naši eksperimenti pokazuju prednost jednosmjerne konzistencije s čistim učiteljem: perturbiranje samo učenikove grane ne zahtijeva inverznu transformaciju u predikcijama, učenje bolje generalizira i zahtijeva manje memorije. Nedavni pristup [118] uspješno primjenjuje prilagođeni kontrastni gubitak [119, 120] između dviju grana koje primaju preklapajuće isječke slike. Predlažu i zaustavljanje gradijenta na razini piksela - učitelj je onaj s većom sigurnošću. Kombiniranje usrednjenog učitelja s perturbacijama CutMixa postizalo je stanje tehnike na Cityscapesu na pola rezolucije [28] prije našeg rada.

Za razliku od većine prethodnih gustih pristupa [114, 115, 121], primjenjujemo jednosmjernu konzistenciju, koja ima sličnu memorijsku učinkovitost kao nadzirano učenje. U usporedbi s postupcima učenja s pseudooznakama [114, 115, 121], naš učitelj se ažurira u svakom koraku učenja, što pojednostavljuje postupak učenja na jednu fazu. U usporedbi s nedavnim gustim pristupom koji koristi usrednjenog učitelja [28], ovaj rad predlaže jači perturbacijski model koji daje bolju generalizaciju i pokazuje kompetitivnost jednostavne konzistencije s usrednjenim učiteljem.

Za razliku od prethodnih radova, ovaj rad ispituje polunadzirano učenje s učinkovitim modelima za gustu predikciju, proučava kompoziciju fotometrijskih i geometrijskih perturbacija, proučava više oblika konzistencije uz isti perturbacijski model i nudi objašnjenja prednosti jednosmjerne konzistencije s čistim učiteljem.

Poglavlje 4

Jednosmjerna konzistencija s čistim učiteljem

Ovo poglavlje opisuje algoritam za polunadzirano učenje semantičke segmentacije utemeljen na jednosmjernoj konzistenciji s čistim učiteljem i fotometrijsko-geometrijskim perturbacijama [26, 27]. Predloženi algoritam kao konzistencijski gubitak za gustu predikciju koristi srednju divergenciju između odgovarajućih piksela predikcija u čistoj i perturbiranoj slici. Slike se perturbiraju kompozicijom slučajnih fotometrijskih i geometrijskih transformacija. Konzistencijski gubitak je minimalan ako je predikcija invarijantna na fotometrijske transformacije i ekvivarijantna na geometrijske transformacije, što znači da se predikcija izobličava na isti način kao slika.

4.1 Oznake

Prosjek preko skupa označavamo slično kao očekivanje: $\mathbf{E}_{\boldsymbol{x}\in\mathbb{D}}\, f(\boldsymbol{x})$. Unakrsnu entropiju označavamo s $\mathrm{H}_q(p) \coloneqq \mathbf{E}_{y\sim p} \ln q(y)$, a entropiju s $\mathrm{H}(p)$, kao u odjeljku 2.2.1. Ulazne slike su $\boldsymbol{x} \in \mathbb{X} = [0,1]^{H\times W\times 3}$, a guste oznake $\boldsymbol{y} \in \mathbb{Y} = \{1..C\}^{H\times W}$. Označeni i neoznačeni skup podataka označavamo s $\mathbb{D}_\mathrm{l} \subset \mathbb{X}\times\mathbb{Y}$ i $\mathbb{D}_\mathrm{u} \subset \mathbb{X}$, gdje $\mathbb{D}_\mathrm{u}$ može sadržavati slike iz $\mathbb{D}_\mathrm{l}$. Primjerak modela $h_{\boldsymbol{\theta}}\colon \mathbb{X}\to\mathbb{P}$ s parametrima $\boldsymbol{\theta}$ preslikava sliku u polje kategoričkih razdioba: $h_{\boldsymbol{\theta}}(\boldsymbol{x})_{[i,j,c]} = \mathrm{P}(\underline{\boldsymbol{y}}_{[i,j]} = c|\boldsymbol{x},\boldsymbol{\theta})$. Radi jednostavnosti izlazne vektore vjerojatnosti razreda identificiramo s razdiobama: $h_{\boldsymbol{\theta}}(\boldsymbol{x})_{[i,j]} \equiv \mathrm{P}[\underline{\boldsymbol{y}}_{[i,j]}|\boldsymbol{x},\boldsymbol{\theta}]$.

4.2 Fotometrijsko-geometrijski perturbacijski model

Perturbacijski model za potrebe ovog rada možemo definirati kao par transformacija sa zajedničkim parametrima, od kojih se jedna primjenjuje na sliku, a druga na predikciju ili oznaku. Odabir perturbacijskih parametara $\boldsymbol{\tau}$ daje par transformacija $T_{\boldsymbol{\tau}}\colon \mathbb{X}\to\mathbb{X}$ i

$T'_{\boldsymbol{\tau}}\colon \mathbb{P} \to \mathbb{P}$, gdje je ulazni prostor $\mathbb{X}$ domena, a prostor predikcija $\mathbb{P}$ kodomena parametriziranog modela $h_{\boldsymbol{\theta}}$. $T_{\boldsymbol{\tau}}$ i $T'_{\boldsymbol{\tau}}$ moraju (idealno) biti takvi da u slučaju idealnog primjerka modela h^* vrijedi ekvivarijantnost:

$$h^*(T_{\boldsymbol{\tau}}(\boldsymbol{x})) = T'_{\boldsymbol{\tau}}(h^*(\boldsymbol{x})) \tag{4.1}$$

za svaki ispravan $\boldsymbol{\tau}$.

Predloženi perturbacijski model je kompozicija fotometrijske transformacije $T^{\mathrm{P}}_{\boldsymbol{\varphi}}$ i geometrijske transformacije $T^{\mathrm{G}}_{\boldsymbol{\gamma}}$ koju možemo izraziti parom:

$$T_{\boldsymbol{\tau}} = T^{\mathrm{G}}_{\boldsymbol{\gamma}} \circ T^{\mathrm{P}}_{\boldsymbol{\varphi}} \qquad \text{(u prostoru slike)}, \tag{4.2}$$

$$T'_{\boldsymbol{\tau}} = T^{\mathrm{G}}_{\boldsymbol{\gamma}} \qquad \text{(u izlaznom prostoru)}. \tag{4.3}$$

Pretpostavljamo da se geometrijske transformacija jednako primjenjuje na slike i predikcije, a da je fotometrijska perturbacija u izlaznom prostoru (u predikcijama i oznakama) identitet, tj. da primjerak modela treba biti invarijantan na fotometrijske perturbacije. Pri tome koristimo istu oznaku $T^{\mathrm{G}}_{\boldsymbol{\gamma}}$ za geometrijsku transformaciju u prostoru slike i odgovarajuću transformaciju u prostoru predikcije.

4.2.1 Fotometrijska komponenta perturbacijskog modela

Predložena fotometrijska perturbacija $T^{\mathrm{P}}_{\boldsymbol{\varphi}}$ je kompozicija 5 jednostavnijih transformacija s 5 parametara $\boldsymbol{\varphi} = (b, s, h, c, \boldsymbol{\pi})$. Te transformacije se primjenjuju jednako u svakom pikselu sljedećim redoslijedom:

1. svjetlini se dodaje b,
2. zasićenost se množi sa s,
3. nijansi se dodaje h,
4. mijenja se kontrast množenjem svih kanala u prostoru RGB sa c,
5. kanali u prostoru RGB se permutiraju prema $\boldsymbol{\pi}$.

Ovdje *nijansa*, *zasićenost* i *svjetlina* imaju značenje komponenata u prostoru boja HSV.

Parametri $\boldsymbol{\varphi}$ se uzorkuju ovako za svaku sliku: $b \sim \mathrm{U}(-0.25, 0.25)$, $s \sim \mathrm{U}(0.25, 2)$, $h \sim \mathrm{U}(-36°, 36°)$, $c \sim \mathrm{U}(0.25, 2)$ i $\boldsymbol{\pi} \sim \mathrm{U}(\mathbb{S}_3)$, gdje je $\mathbb{S}_3$ skup svih 6 permutacija troelementnog skupa.

4.2.2 Geometrijska komponenta perturbacijskog modela

Geometrijska komponenta predloženog perturbacijskog modela temelji se na vrsti poliharmoničke interpolacije (TPS, engl. *thin plate spline*) [122, 123], koja omogućuje glatko parametarsko izobličavanje.

Razmatramo 2D izobličenja kao funkciju $f\colon \mathbb{R}^2 \to \mathbb{R}^2$ koja parove slikovnih koordinata $\boldsymbol{q}$ preslikava u pomake. TPS-izobličenje interpolira skup parova koordinata i pomaka $\{(\boldsymbol{c}_i, \boldsymbol{d}_i)\colon i = 1..n\} \subset \mathbb{R}^2 \times \mathbb{R}^2$ zadovoljavanjem uvjeta $f(\boldsymbol{c}_i) = \boldsymbol{d}_i$ uz minimizaciju aproksimacije energije savijanja (zakrivljenosti), koju za k-tu ($k \in \{1,2\}$) komponentu izobličenja možemo izraziti kao integral zbroja kvadrata elemenata Hesseove matrice:

$$\int_{\mathrm{dom}(f)} \left\| \frac{\partial}{\partial \boldsymbol{q}} \left(\frac{\partial f(\boldsymbol{q})_{[k]}}{\partial \boldsymbol{q}} \right)^{\mathsf{T}} \right\|_{\mathrm{F}}^2 \mathrm{d}\boldsymbol{q}. \tag{4.4}$$

U slučaju 2D koordinata, rješenje tog problema ima oblik

$$f(\boldsymbol{q}) = \boldsymbol{A} \begin{bmatrix} 1 \\ \boldsymbol{q} \end{bmatrix} + \boldsymbol{W} \Big[\phi(\|\boldsymbol{q} - \boldsymbol{c}_i\|) \Big]_{i=1..n}^{\mathsf{T}}, \tag{4.5}$$

gdje $\phi(r) = r^2 \ln(r)$, $\boldsymbol{q}$ je vektor koordinata, $\boldsymbol{A}$ afina transformacijska matrica dimenzija 2×3, a $\boldsymbol{W}$ matrica koeficijenata n kontrolnih točaka dimenzija $2 \times n$ [122, 124]. Takva transformacija je ekvivarijantna na rotaciju i translaciju [122, 123], tj. $f(g(\boldsymbol{q})) = g(f(\boldsymbol{q}))$ za svaku kompoziciju rotacije i translacije g.

Parametri $\boldsymbol{A}$ i $\boldsymbol{W}$ se mogu dobiti kao rezultat linearnog sustava koji čini $2n$ ograničenja interpolacije pomaka $f(\boldsymbol{c}_i) = \boldsymbol{d}_i$ i dodatnih $2 \cdot 3$ ograničenja koja slijede iz zahtijeva kvadratne integrabilnosti drugih derivacija izobličenja f: $\boldsymbol{W}\boldsymbol{1}_n = \boldsymbol{0}_2$ i $\forall (j,k) \in \{1,2\}^2\colon \sum_{i=1}^{n} \boldsymbol{W}_{[j,i]} \mathbf{c}_{i[k]} = 0$ [125].

Uz poznate $\boldsymbol{A}$ i $\boldsymbol{W}$, lako se mogu deformirati cijele slika. Prvo ćemo razmatrati slike kao funkcije s kontinuiranom domenom, a kasnije se vratiti na slike kao nizove iz $[0,1]^{H \times W \times 3}$. Neka je $I\colon \mathrm{dom}(I) \to [0,1]^3$ originalna slika veličine (W,H), gdje $\mathrm{dom}(I) = [0,W] \times [0,H]$. Onda se transformirana slika I' može izraziti ovako:

$$I'(\boldsymbol{q} + f(\boldsymbol{q})) = \begin{cases} I(\boldsymbol{q}), & \boldsymbol{q} \in \mathrm{dom}(I), \\ \boldsymbol{0}, & \text{inače.} \end{cases} \tag{4.6}$$

Ovako izražena transformacija slike, koja izvorne koordinate $\boldsymbol{q} \in \mathrm{dom}(I)$ preslikava u odredišne $\boldsymbol{q}'$ naziva se *unaprijedno izobličenje* (engl. *forward warping*) [126] i složenija je za implementirati.

Ako uz pomake $\boldsymbol{d}_i$ umjesto polazišnih koordinata $\boldsymbol{c}_i$ zadamo odredišne koordinate $\boldsymbol{c}'_i = \boldsymbol{c}_i + \boldsymbol{d}_i$, možemo dobiti unatražnu transformaciju:

$$\tilde{I}(\boldsymbol{q}') = \begin{cases} I(\boldsymbol{q}' - \tilde{f}(\boldsymbol{q}')), & \boldsymbol{q}' - \tilde{f}(\boldsymbol{q}') \in \mathrm{dom}(I), \\ \boldsymbol{0}, & \text{inače.} \end{cases} \tag{4.7}$$

Ovako izražena transformacija odredišne koordinate $\boldsymbol{q}' \in \mathrm{dom}(I)$ preslikava u izvorne $\boldsymbol{q}$. Ona se naziva *unatražno izobličenje* (engl. *backward warping*) [126] i za diskretne slike može se jednostavno ostvariti pomoću bilinearne interpolacije: za svaki odredišni piksel može se izravno dobiti vrijednost iz odgovarajućih polazišnih koordinata. Suvremeni programski okviri već uključuju implementaciju koordinatnih transformacija slika na grafičkom sklopovlju. Ovisno o broju kontrolnih točaka n, najzahtjevniji dio računanja je određivanje parametara izobličenja iz izraza (4.5), za što je potrebno riješiti dva linearna sustava s $(n+3)^2$ varijabli [123].

U eksperimentima koristimo $n = 4$ kontrolne točke, čija polazišta odgovaraju središtima četiriju kvadranata slike:

$$(\boldsymbol{c}'_1, .., \boldsymbol{c}'_4) = \left(\left[\frac{1}{4}H, \frac{1}{4}W\right]^\mathsf{T}, .., \left[\frac{3}{4}H, \frac{3}{4}W\right]^\mathsf{T} \right). \tag{4.8}$$

Promjenjivi parametri naše geometrijske transformacije su 4 pomaka slikovnih koordinata $\boldsymbol{\gamma} = (\boldsymbol{d}_1, .., \boldsymbol{d}_4)$. Neka je $f_{\boldsymbol{\gamma}}$ rezultirajuće izobličenje s pomacima $\boldsymbol{\gamma}$. Takvu transformaciju slike možemo izraziti ovako:

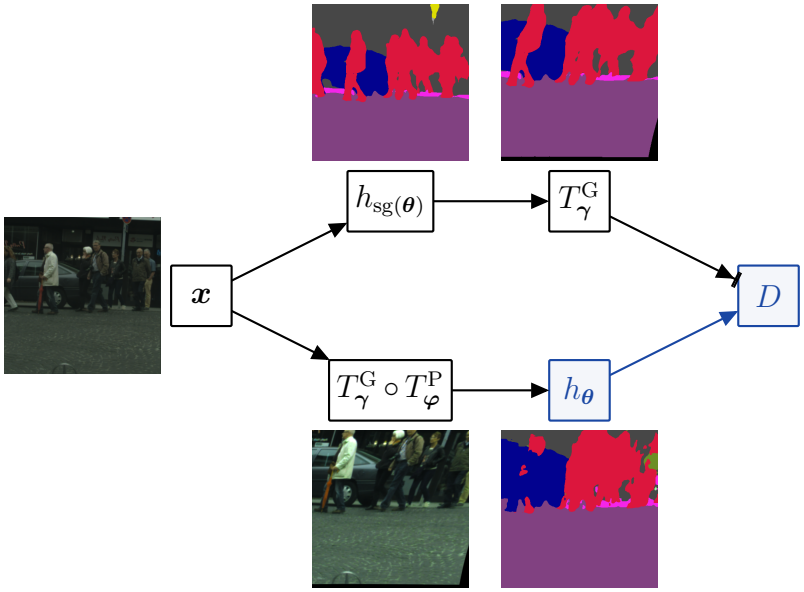
$$T^{\mathrm{G}}_{\boldsymbol{\gamma}}(\boldsymbol{x}) = \mathrm{backward\_warp}(\boldsymbol{x}, f_{\boldsymbol{\gamma}}). \tag{4.9}$$

Predloženi postupak učenja nasumično uzorkuje $\boldsymbol{\gamma}$ za svaku označenu sliku. Svaki pomak se uzorkuje iz 2D normalne razdiobe $\mathrm{N}(\mathbf{0}_2, 0.05H \cdot \mathbf{I}_2)$, gdje je H visina isječka slike za učenje.

4.3 Gusta jednosmjerna konzistencija

Prilagođavamo jednosmjernu konzistenciju [6, 7] na zadatak guste predikcije uz predloženi perturbacijski model $T_{\boldsymbol{\tau}} = T^{\mathrm{G}}_{\boldsymbol{\gamma}} \circ T^{\mathrm{P}}_{\boldsymbol{\varphi}}$, gdje je $T^{\mathrm{G}}_{\boldsymbol{\gamma}}$ glatko geometrijsko izobličenje, $T^{\mathrm{P}}_{\boldsymbol{\varphi}}$ fotometrijska perturbacija koja se jednako primjenjuje na sve piksele, a $\boldsymbol{\tau} = (\boldsymbol{\gamma}, \boldsymbol{\varphi})$ parametri perturbacije. $T^{\mathrm{G}}_{\boldsymbol{\gamma}}$ pomiče piksele s obzirom na gusto polje izobličenja. Takva geometrijska perturbacija se kod jednosmjerne konzistencije mora primijeniti na učenikov ulaz i učiteljev izlaz. Računanje gubitka guste jednosmjerne konzistencije s predloženim perturbacijskim modelom ilustriran je na slici 4.1 u obliku računskog grafa. Kod jednostavne jednosmjerne konzistencije učiteljevi parametri su smrznuta kopija učenikovih parametara: $\mathrm{sg}(\boldsymbol{\theta})$. Kod usrednjenog učitelja (engl. *Mean Teacher*) parametri su eksponencijalni pokretni prosjek učenikovih parametara. Kod jednostavne dvosmjerne konzistencije obje grane koriste iste parametre $\boldsymbol{\theta}$, za koje se u obje grane računa gradijent.

Općenita funkcija pogreške polunadziranog konzistencijskog učenja $E(\boldsymbol{\theta}; \mathbb{D}_{\mathrm{l}}, \mathbb{D}_{\mathrm{u}})$ može



Slika 4.1: Gusta jednosmjerna konzistencija s čistim učiteljem. Gornja grana: čisti ulaz se daje učitelju $h_{\mathrm{sg}(\boldsymbol{\theta})}$ i njegove predikcije se perturbiraju geometrijskom perturbacijom $T_{\boldsymbol{\gamma}}^{\mathrm{G}}$. Donja grana: ulaz se perturbira istom geometrijskom i fotometrijskom perturbacijom i daje učeniku $h_{\boldsymbol{\theta}}$. Funkcija gubitka D je prosječna KL divergencija između predikcija dviju grana. Gradijent se računa samo u plavom dijelu grafa.

se izraziti otežanim zbrojem srednjeg nadziranog gubitka L_{s} i srednjeg konzistencijskog gubitka L_{c}:

$$E(\boldsymbol{\theta}; \mathbb{D}_{\mathrm{l}}, \mathbb{D}_{\mathrm{u}}) = \mathop{\mathbf{E}}_{(\boldsymbol{x},\boldsymbol{y})\in\mathbb{D}_{\mathrm{l}}} L_{\mathrm{s}}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{y}) + \alpha \mathop{\mathbf{E}}_{\boldsymbol{x}\in\mathbb{D}_{\mathrm{u}}} \mathop{\mathbf{E}}_{\boldsymbol{\tau}} L_{\mathrm{c}}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{\tau}). \tag{4.10}$$

U našim eksperimentima, L_{s} je uobičajeni srednji gubitak unakrsne entropije po pikselima s L^2 regularizacijom. Očekivanje po perturbacijskim parametrima $\boldsymbol{\tau}$ procjenjujemo s jednim uzorkom po ulaznom primjeru po koraku učenja.

Za nenadzirani gubitak L_{c} u pikselu (i, j) koristimo divergenciju D između predikcije u perturbiranoj slici i perturbirane učiteljeve predikcije u čistoj slici:

$$L_{\mathrm{c}}^{i,j}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{\tau}) = D(T_{\boldsymbol{\gamma}}^{\mathrm{G}}(h_{\mathrm{sg}(\boldsymbol{\theta})}(\boldsymbol{x}))_{[i,j]}, h_{\boldsymbol{\theta}}((T_{\boldsymbol{\gamma}}^{\mathrm{G}} \circ T_{\boldsymbol{\varphi}}^{\mathrm{P}})(\boldsymbol{x}))_{[i,j]}). \tag{4.11}$$

Takav gubitak potiče ekvivarijantnost primjerka modela s obzirom na perturbacije. Na konačni konzistencijski gubitak utječe još i maska valjanosti $\boldsymbol{v}^{\boldsymbol{\gamma}} \in \{0,1\}^{H\times W}$:

$$\boldsymbol{v}_{[i,j]}^{\boldsymbol{\gamma}} = \left[\!\left[T_{\boldsymbol{\gamma}}^{\mathrm{G}}(\mathbf{1}_{H\times W})_{[i,j]} = 1 \right]\!\right]. \tag{4.12}$$

Maska valjanosti osigurava da se gubitak ne primijenjuje na dopunu uzorkovanu izvan koordinata slike $[1, H] \times [1, W]$. Vektor $T_{\boldsymbol{\gamma}}^{\mathrm{G}}(h_{\boldsymbol{\theta}}(\boldsymbol{x}))_{[i,j]}$ predstavlja razdiobu samo ako $\boldsymbol{v}_{[i,j]}^{\boldsymbol{\gamma}} = 1$. Gubitak konzistencije se onda može izraziti kao srednji gubitak unutar maske

valjanosti:

$$L_{\mathrm{c}}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{\tau}) = \frac{1}{\sum(\boldsymbol{v}^{\gamma})} \sum_{i,j} \boldsymbol{v}^{\gamma}_{[i,j]} L_{\mathrm{c}}^{i,j}(\boldsymbol{\theta}; \boldsymbol{x}, \boldsymbol{\tau}). \tag{4.13}$$

Primijetimo da se u izrazu (4.11) ne računa gradijent s obzirom na učiteljeve parametre $\mathrm{sg}(\boldsymbol{\theta})$. Takav oblik učenja nazivamo jednosmjernom konzistencijom s čistim učiteljem. Ona potiče da predikcije u perturbiranim ulazima budu sličnije predikcijama u čistim ulazima, koje su češće pouzdanije. Jednosmjerna konzistencija ima i računsku prednost jer se aktivacije učiteljeve grane ne moraju čuvati u memoriji za računanje gradijenta. Može se primijetiti da se dvosmjerna konzistencija [21, 22] dobije zamjenom $\mathrm{sg}(\boldsymbol{\theta})$ s $\boldsymbol{\theta}$.

Za divergenciju D koristimo relativnu entropiju (KL divergenciju), koja se može izraziti kao razlika unakrsne entropije učenikove razdiobe q s obzirom na učiteljevu razdiobu p i entropije učiteljeve razdiobe:

$$D(p, q) \coloneqq \mathrm{D}(p \parallel q) = - \mathop{\mathbf{E}}_{y \sim p} \ln \frac{q(y)}{p(y)} = \mathrm{H}(p \parallel q) - \mathrm{H}(p). \tag{4.14}$$

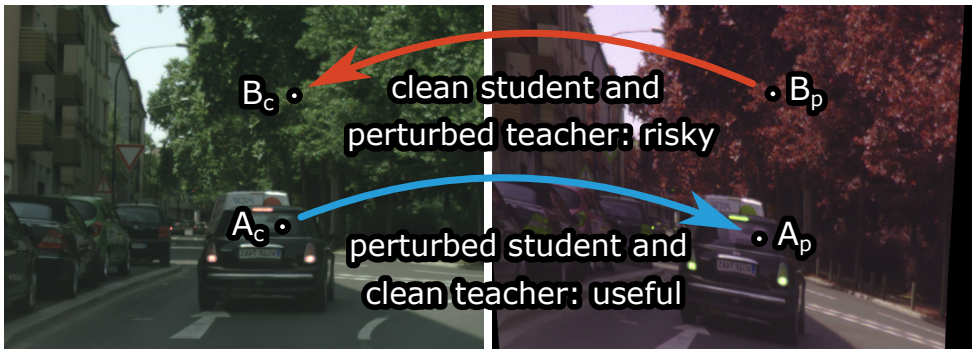
Budući da gradijent ne propagiramo kroz učiteljeve parametre $\mathrm{sg}(\boldsymbol{\theta})$, član $-\mathrm{H}(p)$ ne potiče povećavanje entropije i ne utječe na optimizaciju. Gubitak KL divergencije ekvivalentan je gubitku unakrsne entropije.

Neki istraživači su primijetili poboljšanje u generalizaciji nakon dodavanja člana koji potiče smanjivanje entropije [96] konzistencijskom gubitku [6, 7]. U našim početnim eksperimentima to nije imalo velik utjecaj.

Intuitivno ima smisla da jednosmjerna konzistencija bolje generalizira s čistim učiteljem nego s čistim učenikom jer jače perturbacije češće daju lošije predikcije (slika 1.1). Ako su perturbacije jače, algoritam učenja može biti skloniji tome da nauči razlikovati perturbirane od čistih slika i, ako se perturbiraju obje grane, naučiti sve perturbirane slike preslikavati u slične proizvoljne predikcije (npr. uvijek ista razdioba klasa) [109]. Slika 4.2 ilustrira da konzistencijsko učenje ima najbolje izglede za uspjeh ako učitelj dobiva čistu sliku, a učenik uči na perturbiranoj slici.

4.4 Memorijski učinkovit postupak učenja

Algoritam 4.1 pojednostavljeno prikazuje proceduru za računanje gradijenta predloženog polunadziranog gubitka (4.10) na paru mini-grupa označenih i neoznačenih primjera. Radi jednostavnosti neke oznake imaju malo drugačije značenje ovdje: $\boldsymbol{x}_{\mathrm{l}}$ i $\boldsymbol{y}_{\mathrm{l}}$ su mini-grupe veličina B_{l}, $\boldsymbol{x}_{\mathrm{u}}$, $\boldsymbol{\gamma}$ i $\boldsymbol{\varphi}$ mini-grupe veličine B_{u} i sve funkcije se primjenjuju na mini-grupe. Algoritam računa gradijent nadziranog gubitka, izbacuje spremljene međurezultate iz me-



Slika 4.2: Dvije varijante učenja jednosmjerne konzistencije na čistoj slici (lijevo) i njenoj perturbiranoj inačici (desno). Strelice označavaju smjer protoka informacija od učitelja prema učeniku. Jednosmjerna konzistencija s čistim učiteljem uči piksele predikcije u perturbiranoj slici (A_p) da budu konzistentne s odgovarajućim pikselima predikcije u čistoj slici (A_c). Obrnuti smjer učenja konzistencije (učenje u B_c prema predikciji u B_p) pogoršava performansu. Jedan od razloga pogoršavanja performanse je da jače perturbacije daju lošije predikcije.

morije, računa učiteljeve predikcije, primjenjuje gubitak konzistencije (4.11) i na kraju akumulira doprinose gradijenta.

Algoritam 1. Izračunavanje gradijenta predloženog polunadziranog gubitka uz mini-grupe uzorkovanih perturbacijskih parametara ($\boldsymbol{\gamma}$, $\boldsymbol{\varphi}$), označenih primjera ($\boldsymbol{x}_\mathrm{l}$, $\boldsymbol{y}_\mathrm{l}$) i neoznačenih primjera ($\boldsymbol{x}_\mathrm{u}$). CE označava srednju unakrsnu entropiju, a KL_masked srednju KL divergenciju po valjanim pikselima. Blok koji počinje s ključnom rječju "kontekst" odgovara bloku s ključnom ključnom rječju "with" u Pythonu, kakav se uz PyTorchev upravitelj konteksta `torch.no_grad()` koristi za isključivanje pamćenja računskog grafa i računanja gradijenta za operacije unutar bloka.

1# $\boldsymbol{x}_\mathrm{u}$, $\boldsymbol{\gamma}$, $\boldsymbol{\varphi}$, $\boldsymbol{p}_\mathrm{t}$, $\boldsymbol{p}_\mathrm{s}$,i $\boldsymbol{v}$ sumini −grupeod B_u elemenata.
2# $\boldsymbol{x}_\mathrm{l}$, $\boldsymbol{y}$,i $\boldsymbol{p}_\mathrm{l}$ sumini −grupeod B_l elemenata.
3 **procedura** izračunaj_gradijent_gubitka($h, \boldsymbol{\theta}, \boldsymbol{x}_\mathrm{l}, \boldsymbol{y}_\mathrm{l}, \boldsymbol{x}_\mathrm{u}, \boldsymbol{\gamma}, \boldsymbol{\varphi}$):
4 $\boldsymbol{\theta}' \leftarrow \mathrm{sg}(\boldsymbol{\theta})$#smrznutakopijaparametara
5
6#nadziranigubitak
7 $\boldsymbol{p}_\mathrm{l} \leftarrow h_{\boldsymbol{\theta}}(\boldsymbol{x}_\mathrm{l})$
8 $L_\mathrm{s} \leftarrow \mathrm{CE}(\boldsymbol{y}_\mathrm{l}, \boldsymbol{p}_\mathrm{l})$
9 $\boldsymbol{g} \leftarrow \nabla_{\boldsymbol{\theta}} L_\mathrm{s}$ #čistisačuvaneme đurezultate
10
11#nenadziranigubitak
12 **kontekst** bez_pamćenja_računskog_grafa():#me đurezultatineostajusačuvaniovdje
13 $\boldsymbol{p}_\mathrm{t} \leftarrow T_{\boldsymbol{\gamma}}^\mathrm{G}(h_{\boldsymbol{\theta}'}(\boldsymbol{x}_\mathrm{u}))$#čistiučitelj
14 $\boldsymbol{p}_\mathrm{s} \leftarrow h_{\boldsymbol{\theta}}((T_{\boldsymbol{\gamma}}^\mathrm{G} \circ T_{\boldsymbol{\varphi}}^\mathrm{P})(\boldsymbol{x}_\mathrm{u}))$#perturbiraniučenik
15 $\boldsymbol{v} \leftarrow \lfloor T_{\boldsymbol{\gamma}}^\mathrm{G}(\mathbf{1}_{B_\mathrm{u} \times H \times W}) \rfloor$ #maskavaljanosti
16 $L_\mathrm{c} \leftarrow \alpha \cdot \mathrm{KL\_masked}(\boldsymbol{p}_\mathrm{t}, \boldsymbol{p}_\mathrm{s}, \boldsymbol{v})$

17 $\boldsymbol{g} \leftarrow \boldsymbol{g} + \nabla_{\boldsymbol{\theta}} L_{\mathrm{c}}$

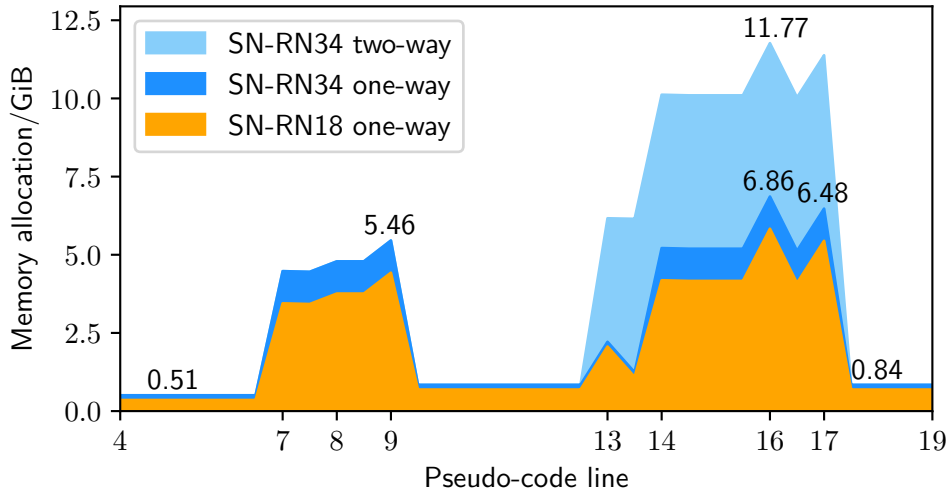
18

19 **vrati** $\boldsymbol{g}$

Slika 4.3 ilustrira zauzetost grafičke memorije tijekom iteracije polunadziranog učenja modela SwiftNet-RN34 s jednosmjernom i dvosmjernom konzistencijom uz isječke slika veličine 768×768 i veličine mini-grupa $(B_{\mathrm{l}}, B_{\mathrm{u}}) = (8, 8)$. Mjerenja su dobivena korištenjem procedura iz paketa `torch.cuda`: `max_memory_allocated`, `memory_allocated`, `reset_peak_memory_stats` i `empty_cache`. Za učenje je korištena grafička kartica RTX A4500.

Brojevi na x-osi odgovaraju linijama u pseudokodu u algoritmu 4.1. Linija 9 računa gradijent nadziranog gubitka propagacijom unatrag i zadržava gradijent u memoriji. Zauzetost memorije se podigne i onda spusti zbog automatskog brisanja čuvanih međurezultata koji više nisu potrebni. Linija 13 računa učiteljev izlaz bez zadržavanja međurezultata (koristi se `torch.no_grad`). Linija 16 računa nenadzirani gubitak, što zahtijeva čuvanje međurezultata, uključujući predikcije koje su na visokoj rezoluciji. To podiže zauzetost memorije na maksimalnih 6.86 GiB. Poslije toga se zauzetost spusti zbog brisanja perturbiranih ulaza i učiteljeve predikcije u stvarnoj implementaciji. Linija 17 računa gradijent konzistencijskog gubitka i akumulira ga. Zauzetost memorije naraste do 6.48 GiB zbog zadržavanja međurezultata gubitka i nakon toga se spusti zbog automatskog brisanja međurezultata pri računanju gradijenta. Na kraju je zauzetost memorije je malo veća nego kod linije 4 jer se u memoriji još zadržavaju nadzirane predikcije radi evaluacije performanse raspoznavanja na skupu za učenje.

Omjer između maksimalne zauzetosti memorije kod nadziranog gubitka (linija 16) i nenadziranog gubitka (linija 9) pokazuje da polunadzirani algoritam ne zahtijeva puno više memorije nego nadzirano učenje. Apsolutna razlika u zauzetosti memorije ne ovisi o modelu i ovisi o ukupnoj veličini perturbiranih ulaza, predikcija i međurezultata računanja KL divergencije. Kako se povećava model, tako se omjer vršnih zauzetosti memorije približava 1. U slučaju modela SwiftNet-RN34 omjer je približno 1.26.



Slika 4.3: Zauzetost grafičke memorije tijekom i poslije izvođenja određenih linija algoritma 4.1 tijekom druge iteracije učenja. Naša implementacija koja koristi PyTorch i uključuje modele SwiftNet-RN18 i SwiftNet-RN34 s jednosmjernom i dvosmjernom konzistencijom, isječke veličine 768×768 i veličine mini-grupa $(B_\mathrm{l}, B_\mathrm{u}) = (8, 8)$. Linija 9 računa nadzirani gradijent, linija 13 učiteljev izlaz (bez čuvanja međurezultata za računanje gradijenta). Linije 16 i 17 računaju konzistencijski gubitak i njegov gradijent.

Poglavlje 5

Eksperimenti

Eksperimenti istražuju generalizacijsku performansu jednosmjerne konzistencije s čistim učiteljem s obzirom na kompoziciju fotometrijskih i geometrijskih perturbacija ($T_\gamma^G \circ T_\phi^P$). Ispitujemo i uspoređujemo naš pristup s drugim oblicima konzistencije i stanjem tehnike u polunadziranoj semantičkoj segmentaciji.

Ako nije drugačije navedeno, "simple" označava jednostavnu jednosmjernu konzistenciju, a "MT" usrednjenog učitelja [5]. "PhTPS" označava naše perturbacije. U eksperimentima koji uspoređuju oblike konzistencije, "1w" (*one-way*) označava jednosmjernu, a "2w" (*two-way*) dvosmjernu konzistenciju, "ct" (*clean teacher*) označava da je učiteljev ulaz čist (slabo perturbiran), "cs" (*clean student*) da je učenik čist, a "2p" da su oba ulaza perturbirana. Polunadzirane eksperimente provodimo u nekoliko postavki s različitim udjelima oznaka i količinama podataka za semantičku segmentaciju i klasifikaciju slika.

Programska implementacija temelji se najviše na okviru PyTorch [53]. Izvorni kod za eksperimente je dostupan na `https://github.com/Ivan1248/semisup-seg-efficient`.

5.1 Skupovi podataka

Semantičku segmentaciju provodimo na podatkovnim skupovima Cityscapes [11] i PASCAL VOC 2012 [127] s proširenim skupom za učenje [128].

Fino označeni podskup Cityscapesa sastoji se od 2975 slika za učenje, 500 validacijskih i 1525 slika za testiranje rezolucije 1024×2048. Oznake su na razini piksela i predstavljaju 19 klasa i uz to jednu klasu koja označava piksele koji se ne uključuju u evaluaciju. Slike su pribavljene iz vozila u pokretu tijekom dana i po lijepom vremenu. Provodimo eksperimente sa slikama i oznakama na pola rezolucije i na punoj rezoluciji. U eksperimentima na pola rezolucije slike interpoliramo bilinearno. Radi uspoređivanja s prethodnim radom, oznake isto umanjujemo i interpoliramo najbližim susjedom (kod učenja i evaluacije). Neki eksperimenti također koriste grubo označeni podskup Cityscapesa

("train-extra"), koji sadrži 19998 slika s grubim oznakama.

Prošireni PASCAL VOC 2012 (PASCAL VOC) sastoji se od 10582 slika za učenje, 1449 validacijskih i 1456 slika za testiranje različitih rezolucija koje su blizu 500×500. Oznake su na razini piksela i predstavljaju 21 klasu i uz to jednu klasu koja označava piksele koji se ne uključuju u evaluaciju. Segmentacije obično uključuju pozadinu i jedan ili više predmeta, koji mogu biti neko vozilo, kućanski predmet, biljka u posudi ili životinja.

Klasifikaciju slika provodimo na skupu CIFAR-10. CIFAR-10 sastoji se od 50000 slika za učenje i 10000 slika za testiranje rezolucije 32×32. CIFAR-10 ima oznake za 10 klasa.

5.2 Postavke eksperimenata

Kod svih polunadziranih eksperimenata skup neoznačenih slika za nenadzirani gubitak $\mathbb{D}_\mathrm{u}$ uz neoznačeni skup za učenje uključuje i slike iz označenog skupa $\mathbb{D}_\mathrm{l}$. Koristimo B_l označenih i B_u neoznačenih slika u svakom koraku učenja. Svaka epoha ima $\lfloor |\mathbb{D}_\mathrm{l}|/B_\mathrm{l} \rfloor$ koraka učenja i odgovara jednom prolazu kroz označeni skup, osim $|\mathbb{D}_\mathrm{l}| - B_\mathrm{l} \lfloor |\mathbb{D}_\mathrm{l}|/B_\mathrm{l} \rfloor$ slučajnih primjera. Drugim riječima, svaka epoha "preskoči" nekoliko slika ako veličina označenog skupa nije djeljiva s veličinom označene grupe B_l.

Većina eksperimenata koristi isti perturbacijski model za sve zadatke i skupove podataka, pri čemu su pomaci kontrolnih točaka geometrijskog izobličenja proporcionalni veličini slike. Jedina iznimka su eksperimenti koji provode iscrpnu validaciju perturbacijskih hiperparametara. Hiperparametri perturbacijskog modela navedeni su na krajevima odjeljaka 4.2.1 i 4.2.2. U većini eksperimenata statistike normalizacije po grupi se ne ažuriraju samo u primjerku modela koji dobiva perturbirane ulaze. Iznimka su eksperimenti Cityscapesa na punoj rezoluciji, gdje je uključivanje ažuriranja statistika perturbiranog učenika postiglo bolje rezultate u validacijskim eksperimentima (v. odjeljak 5.8). Npr. u slučaju jednosmjerne konzistencije s čistim učiteljem, statistike se ažuriraju kod računanja nadziranog gubitka i u čistom učitelju, ali ne u perturbiranom učeniku kod računanja konzistencijskog gubitka. Učitelj uvijek koristi procjenu populacijskih statistika i ne ažurira ih. Usrednjeni učitelj (engl. *Mean Teacher*) koristi eksponencijalni pokretni prosjek učenikovih populacijskih statistika.

5.2.1 Postavke za semantičku segmentaciju

Eksperimenti vezani uz semantičku segmentaciju uključuju sljedeće modele: SwiftNet s okosnicom ResNet-18 (SwiftNet-RN18), SwiftNet s okosnicom ResNet-34 (SwiftNet-RN34) i DeepLab v2 s okosnicom ResNet-101 i Deeplab v3$^+$ s okosnicom ResNet-50. Inicijaliziramo okosnice predtreniranim parametrima za ImageNet.

Sve primjere, uključujući one za koje kažemo da su čisti (slabo perturbirani), perturbiramo slučajnim skaliranjem, izrezivanjem i horizontalnim zrcaljenjem.

Neki hiperparametri se razlikuju ovisno o skupu podataka. Dodatak A.1 predstavlja pregled i usporedbu hiperparametara s postupcima konzistencijskog učenja iz literature.

Semantička segmentacija na Cityscapesu

Veličina isječaka i parametri slučajnog skaliranja razlikuju se ovisno o skupu za učenje. U eksperimentima na Cityscapesu na pola rezolucije veličina isječka je 448×448 i logaritam faktora skaliranja se uzorkuje iz $\mathrm{U}(\ln(1.5^{-1}), \ln(1.5))$. U eksperimentima na punoj rezoluciji veličina isječka je 768×768 i logaritam faktora skaliranja se uzorkuje iz $\mathrm{U}(\ln(2^{-1}), \ln(2))$.

Radi jednostavnosti implementacije, za geometrijska izobličenja predikcija bilinearno interpoliramo logite. Numerička provjera upućuje na to da je razlika u vjerojatnostima nakon interpoliranja logita umjesto vjerojatnosti jako mala.

Raspored koraka optimizacije ima oblik 1/4 perioda kosinusa: $\eta(e) = \eta_0 \cos(e\pi/2)$, gdje je $e \in [0..1]$ udio završenih epoha, a η_0 osnovni korak optimizacije. Validacijski eksperimenti su pokazali da takav oblik ublažava pad generalizacije koji se javlja na kraju učenja sa standardnim rasporedom raspored oblika 1/2 perioda kosinusa [129]. Koristimo osnovni korak učenja $\eta_0 = 4 \cdot 10^{-4}$ za slučajno inicijalizirane parametre i $\eta_0 = 10^{-4}$ za predtrenirane parametre.

Koristimo optimizacijski algoritam Adam (opisan u odjeljku 2.5.2) s $(\beta_1, \beta_2) = (0.9, 0.99)$ [61], koji je otprilike* invarijantan na skaliranje gubitka. Težina L^2-regularizacije u nadziranim eksperimentima je 10^{-4} za slučajno inicijalizirane i $2.5 \cdot 10^{-5}$ za predtrenirane parametre [31]. Za eksperimente na punoj rezoluciji takva L^2-regularizacija je bila prejaka pa za njih koristimo $4\times$ manju težinu. Većina eksperimenata koristi težinu konzistencijskog gubitka $\alpha = 0.5$, do koje smo došli ranim validacijskim eksperimentima na Cityscapesu. Iznimka su eksperimenti koji ju validiraju.

Veličine mini-grupa za Cityscapes su $(B_\mathrm{l}, B_\mathrm{u}) = (8, 8)$ za SwiftNet-RN18 [31] i $(B_\mathrm{l}, B_\mathrm{u}) = (4, 4)$ za DeepLab v2 (s okosnicom ResNet-101) [30]. Veličina mini-grupe u odgovarajućim nadziranim eksperimentima je B_l.

U eksperimentima na Cityscapesu na pola rezolucije SwiftNet se uči $200 \cdot \frac{2975}{|\mathbb{D}_\mathrm{l}|}$ epoha (200 epoha ili 74200 iteracija kad se koriste sve oznake), a DeepLab v2 $100 \cdot \frac{2975}{|\mathbb{D}_\mathrm{l}|}$ epoha (100 epoha ili 74300 iteracija kad se sve oznake koriste). Na punoj rezoluciji SwiftNet se uči $250 \cdot \frac{2975}{|\mathbb{D}_\mathrm{l}|}$ epoha (250 epoha kad se koriste sve oznake). DeepLab v2 s okosnicom ResNet-101 je tijekom učenja na pola rezolucije oko 8 puta sporiji po slici od modela

*Adam bi bio invarijantan na skaliranje gubitka kad bi konstanta koju koristi za sprečavanje dijeljenja s 0 bila 0.

SwiftNeta-RN18, ali zahtijeva manje epoha, čemu vjerojatno doprinosi manji broj slučajno inicijaliziranih parametara. Posljedično, DeepLab v2 s okosnicom ResNet-101 uči se oko 4× sporije od SwiftNeta-RN18 na RTX 2080Ti. Dodatak A.2 predstavlja detaljniju usporedbu memorijskih i vremenskih zahtjeva različitih polunadziranih algoritama.

Semantička segmentacija na skupu PASCAL VOC

Eksperimenti na skupu PASAL VOC koriste većinu hiperparametara osnovice kao u [118] i hiperparametre naših polunadziranih algoritama kao za Cityscapes. U svim eksperimentima koristimo model DeepLab v3$^+$. Veličina isječaka slika za učenje je 320×320. Logaritam faktora skaliranja uzorkujemo iz $\mathrm{U}(\ln(1.5^{-1}), \ln(1.5))$, što se razlikuje se od [118], gdje se faktor skaliranja uzorkuje iz $\mathrm{U}(0.5, 2)$. Raspored koraka optimizacije ima oblik $\eta(e) = \eta_0(1-e)^{0.9}$. Osnovni korak učenja je $\eta_0 = 10^{-2}$ za slučajno inicijalizirane parametre i $\eta_0 = 10^{-3}$ za predtrenirane parametre. Koristimo stohastički gradijentni spust s inercijum (opisan u odjeljku 2.5.2) uz faktor prigušivanja $\gamma = 0.9$. Težina L^2-regularizacije u svim eksperimentima je 10^{-4}. Veličine mini-grupa su $(B_\mathrm{l}, B_\mathrm{u}) = (8, 8)$. Učenje traje $80 \cdot \frac{10582}{|\mathbb{D}_\mathrm{l}|}$ epoha (80 epoha ili 105760 iteracija kad se koriste sve oznake),

5.2.2 Postavke za klasifikaciju

Klasifikacijski eksperimenti koriste skup CIFAR-10 i model WRN-28-2 (Wide ResNet) sa standardnim hiperparametrima [130]. Za sve slike za učenje koristimo standardno slučajno zrcaljenje, dopunjavanje i izrezivanje.

Veličine mini grupe su $(B_\mathrm{l}, B_\mathrm{u}) = (128, 640)$, pa je broj iteracija po epohi $\left\lfloor \frac{|\mathbb{D}_\mathrm{l}|}{128} \right\rfloor$. Učenje na svim podacima traje 200 epoha, a polunadzirano učenje $1000 \cdot \frac{4000}{|\mathbb{D}_\mathrm{l}|}$ epoha. Ako broj oznaka $|\mathbb{D}_\mathrm{l}|$ nije premal, ukupan broj koraka učenja kad se ne koriste svi podaci je otprilike 40% broja koraka nadziranog učenja. Npr. učenje traje $1000 \cdot \frac{4000}{4000} = 1000$ epoha s po $\left\lfloor \frac{50000}{128} \right\rfloor = 390$ koraka kad se koristi 4000 oznaka, a 16000 epoha s po jednim korakom kad se koristi $|\mathbb{D}_\mathrm{l}| = 250$ oznaka. Broj koraka učenja je manji nego kad se ne koriste sve oznake radi bržeg izvršavanja eksperimenata i zbog ranije konvergencije kad je manje primjera označeno.

Koristimo izvorne hiperparametre VAT-a: norma inicijalne perturbacije je $\xi = 10^{-6}$, norma konačne perturbacije je $\epsilon = 10$, težina konzistencijskog gubitka je $\alpha = 1$ [6]. Hiperparametri perturbacija su isti kao za semantičku segmentaciju i pomaci kontrolnih točaka izobličenja uzorkuju se iz $\mathrm{N}(\mathbf{0}, 3.2 \cdot \mathbf{I}_2)$.

5.2.3 Evaluacija i prikaz rezultata

Naši eksperimenti prikazuju generalizacijsku performansu na kraju učenja na odgovarajućim validacijskim skupovima. Za semantičku segmentaciju mjerimo mIoU (srednji omjer presjeka i unije po klasama), dok za klasifikaciju mjerimo klasifikacijsku točnost.

Većina rezultata evaluacije izražena je sredinama i standardnim odstupanjima (s Besselovom korekcijom) uzorka odgovarajuće evaluacijske mjere za 5 izvršavanja učenja s različitim podskupovima oznaka. Kod učenja s podskupovima oznaka, svako od tih izvršavanja odgovara korištenju drugačijeg podskupa oznaka.

Gdje je prikladno, u tablicama su podebljani ukupno najbolji rezultati i podvučeni po odjeljku najbolji rezultati. Jednako su označeni odgovarajući rezultati koji se od najboljeg razlikuju za manje od standardne pogreške razlike sredina, $\sqrt{\frac{\sigma_i^2}{n_i} + \frac{\sigma_j^2}{n_j}}$, gdje su σ_i standardne devijacije, a n_i brojevi izvršavanja. U slučajevima nedostupnih statistika, pretpostavljamo $\sigma_i = 1$ i $n_i = 3$.

5.3 Semantička segmentacija na Cityscapesu na pola rezolucije

Ovdje učimo s različitim udjelima oznaka za učenje i evaluiramo mIoU na validacijskom skupu Cityscapesa na pola rezolucije.

5.3.1 Usporedba algoritama polunadziranog učenja

Tablica 5.1 uspoređuje predloženi pristup s prethodnim stanjem tehnike. Gornji odjeljak predstavlja prethodne radove [9, 10, 28, 116]. Srednji odjeljak predstavlja eksperimente koji koriste DeepLab v2 [30]. Zbog nekih razlika u učenju, naša osnovica daje bolje rezultate od prethodnog rada (razlike su opisane u odjeljku 5.2). Pravilnu usporedbu s [28] provodimo korištenjem našeg učenja u kombinaciji s njihovom polunadziranom metodom. MT-PhTPS bolje generalizira od MT-CutMixa s L^2-gubitkom i pragom uvjerenosti kada je dostupna 1/4 (743) ili više oznaka, dok je MT-CutMix bolji u slučaju 1/8 oznaka.

U donjem odjeljku su eksperimenti s učinkovitim modelom SwiftNet-RN18. Naš perturbacijski model je nadmašio CutMix (izrezivanje i lijepljenje pravokutnika) uz jednostavnu konzistenciju i uz usrednjenog učitelja. Usrednjeni učitelj uglavnom je dao bolju generalizacijsku performansu od jednostavne konzistencije. DeepLab v2 i SwiftNet-RN18 imaju sličnu korist od konzistencijskog gubitka. SwiftNet-RN18 je učinkovitiji za primjene zbog oko 12× bržeg zaključivanja od DeepLaba v2 s okosnicom ResNet-101 na RTX 2080Ti (v. dodatak A.2 za više detalja). Eksperimenti u srednjem i donjem odjeljku

koriste iste podskupove oznaka.

Tablica 5.1: Generalizacijska performansa semantičke segmentacije (mIoU/%) na validacijskom skupu Cityscapesa na pola rezolucije nakon učenja s različitim udjelima oznaka. Gornji odjeljak prikazuje rezultate iz prethodnog rada. Srednji odjeljak prikazuje naše eksperimente s DeepLabom v2. Donji odjeljak prikazuje naše eksperimente sa SwiftNetom-RN18. Eksperimente pokrećemo s 5 različitih podskupova oznaka za svaki broj oznaka i izvještavamo srednje mIoU-ove sa standardnim odstupanjima. Indeks "∼[28]" označava učenje s L^2-gubitkom s pragom uvjerenosti 0.97 i težinom $\alpha = 1$ prema [28]. Rezultati su izraženi prema opisu u odjeljku 5.2.3.

Postupak	**Broj oznaka**			
	371	743	1487	2975 (sve)
DLv2-RN101 nadzirano [10, 28]	56.2	60.2	64.6[1]	66.0
DLv2-RN101 s4GAN+MLMT [10]	59.3	61.9	–	65.8
DLv2-RN101 nadzirano [9]	55.5	59.9	64.1	66.4
DLv2-RN101 AdvSemSeg [9]	58.8	62.3	65.7	67.7
DLv2-RN101 nadzirano [116]	56.0	60.5	–	66.0
DLv2-RN101 ECS [116]	60.3	63.8	–	67.7
DLv2-RN101 MT-CutMix [28]	$60.3_{1.2}$	$63.9_{0.7}$	–	$67.7_{0.4}$
DLv2-RN101 nadzirano	$56.4_{0.4}$	$61.9_{1.1}$	$66.6_{0.6}$	$69.8_{0.4}$
DLv2-RN101 MT-CutMix$_{\sim[28]}$	$\mathbf{63.2}_{1.4}$	$65.6_{0.8}$	$67.6_{0.4}$	$70.0_{0.3}$
DLv2-RN101 MT-PhTPS	$61.5_{1.0}$	$\mathbf{66.4}_{1.1}$	$\mathbf{69.0}_{0.6}$	$\mathbf{71.0}_{0.7}$
SN-RN18 nadzirano	$55.5_{0.9}$	$61.5_{0.5}$	$66.9_{0.7}$	$70.5_{0.6}$
SN-RN18 simple-CutMix	$59.8_{0.5}$	$63.8_{1.2}$	$67.0_{1.4}$	$69.3_{1.1}$
SN-RN18 simple-PhTPS	$60.8_{1.6}$	$64.8_{1.5}$	$\mathbf{68.8}_{0.7}$	$\mathbf{71.4}_{0.6}$
SN-RN18 MT-CutMix$_{\sim[28]}$	$61.6_{0.9}$	$64.6_{0.5}$	$67.6_{0.7}$	$69.9_{0.6}$
SN-RN18 MT-CutMix	$59.3_{1.3}$	$63.3_{1.0}$	$66.8_{0.6}$	$69.7_{0.5}$
SN-RN18 MT-PhTPS	$62.0_{1.3}$	$\mathbf{66.0}_{1.0}$	$\mathbf{69.1}_{0.5}$	$\mathbf{71.2}_{0.7}$

5.3.2 Analiza utjecaja komponenata perturbacijskog modela

Ovaj odjeljak predstavlja ablacije i validaciju hiperparametara za simple-PhTPS uz SwiftNet-RN18.

Tablica 5.2 prikazuje ablacije perturbacijskog modela i uključuje nadzirano i polunadzirano učenje s perturbacijama PhTPS-a. Fotometrijsko rastresanje (Ph) imalo je velik doprinos generalizaciji kod svih udjela oznaka, dok je geometrijsko (TPS) naštetilo generalizaciji nadziranog učenja. Perturbiranje cijele ili pola mini-grupe PhTPS-om kod nadziranog učenja doprinijelo je generalizaciji u eksperimentima s manjim udjelima oznaka, dok geometrijska komponenta (TPS) nigdje nije pomogla. Eksperimenti s perturbacijama PhTPS-a u nadziranom učenju ("PhTPS ∘ st." i "st. | PhTPS ∘ st.") upućuju na to da je jako perturbiranje svih primjera za učenje korisnije ako je dostupno manje podataka, ali nije korisnije od osnovnog rastresanja kad se koristi više od 1/2 podataka. Pri tome jako perturbiranje pola mini-grupe ("st. | PhTPS ∘ st.") bolje generalizira kad se koristi više

od 1/8 (371) podataka. Polunadzirani eksperimenti upućuju na to da najviše doprinose fotometrijske perturbacije (Ph) i da geometrijske perturbacije (TPS) štete generalizaciji kad se koristi 1/2 (1487) ili više oznaka. Najbolje rezultate u tablici postiže jednosmjerna konzistencija s perturbacijama PhTPS-a, nakon čega dolaze jednosmjerna konzistencija s fotometrijskim perturbacijama (Ph) i nadzirano učenje s dodatnim fotometrijskim rastresanjem, koji su malo lošiji kad se koristi 1/8 oznaka.

Zadnji odjeljak tablice prikazuje konzistencijsko učenje s PhTPS-om bez korištenja neoznačenih primjera. On omogućuje provjeru doprinosa neoznačenih podataka i utjecaja korištenja učiteljevih predikcija umjesto oznaka kao u slučaju nadziranog učenja. Ako se ovdje učiteljeve predikcije zamijene točnim oznakama, dobije se nadzirano učenje s perturbiranjem pola mini-grupe PhTPS-om s malo drugačijim uzorkovanjem podataka. Bez neoznačenih podataka nije bilo značajne koristi od konzistencijskog gubitka (-0.1 pb do 0.5 pb) osim u slučaju kad su dostupni svi podaci (0.9 pb) i korištenje samo točnih oznaka je bilo korisnije od konzistencijskog gubitka s učiteljevim predikcijama kad se koristi 1/4 ili manje podataka.

Tablica 5.2: Ablacije perturbacijskog modela na validacijskom skupu Cityscapesa na pola rezolucije (mIoU/%) uz SwiftNet-RN18. Stupac "Osnovne perturbacije" kaže kakve perturbacije se primjenjuju na sve primjere. Drugi odjeljak uspoređuje rezultate nadziranog učenja s perturbacijskim modelima PhTPS, njegovom fotometrijskom komponentom (Ph) i njegovom geometrijskom komponentom (TPS). Oznake oblika "⟨perturbacijski model⟩ ∘ st." označavaju dodatno perturbiranje svake mini grupe dodatnim perturbacijskim modelom nakon standardnog (st.) slabog perturbiranja, a oznaka "st. | PhTPS ∘ st." dodatno perturbiranje polovine svake mini grupe PhTPS-om. Predzadnji odjeljak uspoređuje perturbacijske modele uz jednosmjernu konzistenciju, gdje se neoznačeni podaci (koji uključuju i slike iz označenog skupa) za konzistencijski gubitak još perturbiraju prema stupcu "Konz. perturb." Zadnji odjeljak prikazuje rezultate metode simple-PhTPS ako se koriste samo označeni primjeri. Prikazani su srednji rezultati uz 5 različitih podskupova oznaka za učenje. Indeksi prikazuju razlike od nadzirane osnovne metode iz prvog odjeljka.

Postupak	**Osnovne perturbacije**	**Konzist. perturb.**	**Korištenje neoznač. podataka**	**Broj oznaka** 371	743	1487	2975 (sve)
nadzirano	st.			55.5	61.5	66.9	70.5
nadzirano	Ph ∘ st.			58.1$_{+2.6}$	**64.5**$_{+3.0}$	**68.7**$_{+1.8}$	**71.2**$_{+0.7}$
nadzirano	TPS ∘ st.			54.5$_{-1.0}$	61.5$_{+0.0}$	65.0$_{-1.9}$	68.9$_{-1.6}$
nadzirano	PhTPS ∘ st.			57.4$_{+1.9}$	63.0$_{+1.5}$	66.4$_{-0.5}$	69.0$_{-1.5}$
nadzirano	st. | PhTPS ∘ st.			56.2$_{+0.7}$	62.2$_{+0.7}$	67.4$_{+0.5}$	70.4$_{-0.1}$
simple	st.	Ph	✓	59.2$_{+3.7}$	**64.9**$_{+3.4}$	**68.3**$_{+1.4}$	**71.8**$_{+1.3}$
simple	st.	TPS	✓	58.2$_{+3.1}$	63.4$_{+1.9}$	66.7$_{+0.2}$	70.1$_{+0.4}$
simple	st.	PhTPS	✓	**60.8**$_{+5.3}$	**64.8**$_{+3.3}$	**68.8**$_{+1.9}$	**71.4**$_{+0.9}$
simple	st.	PhTPS		55.4$_{-0.1}$	61.9$_{+0.4}$	67.4$_{+0.5}$	**71.4**$_{+0.9}$

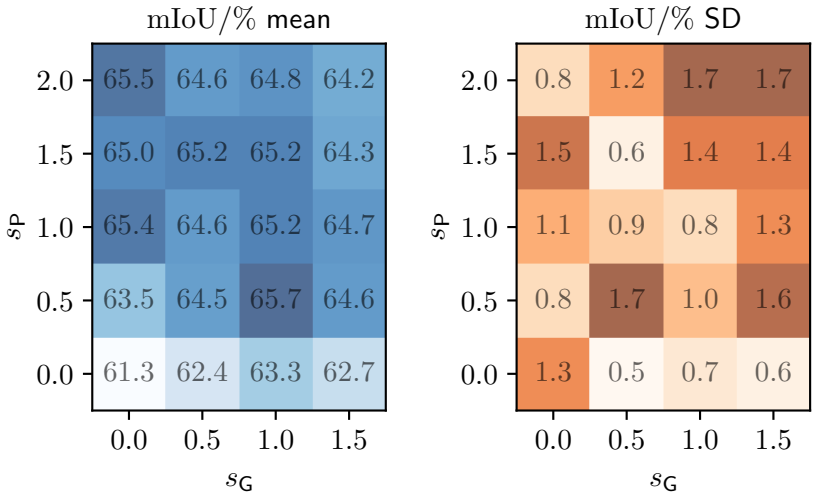
Slika 5.1 prikazuje validaciju jačine perturbacije uz korištenje 743 oznake. Stupci od-

govaraju faktoru s_G, koji množi standardnu devijaciju pomaka kontrolnih točaka određenu na kraju odjeljka 4.2.2. Reci odgovaraju jačini fotometrijske perturbacije s_P, koja slučajne fotometrijske parametre $(\underline{b}, \underline{s}, \underline{h}, \underline{c})$ mijenja ovako:

$$(\underline{b}, \underline{s}, \underline{h}, \underline{c}) \mapsto (s_P \cdot \underline{b}, \exp(s_P \cdot \ln(\underline{s})), s_P \cdot \underline{h}, \exp(s_P \cdot \ln(\underline{c}))). \tag{5.1}$$

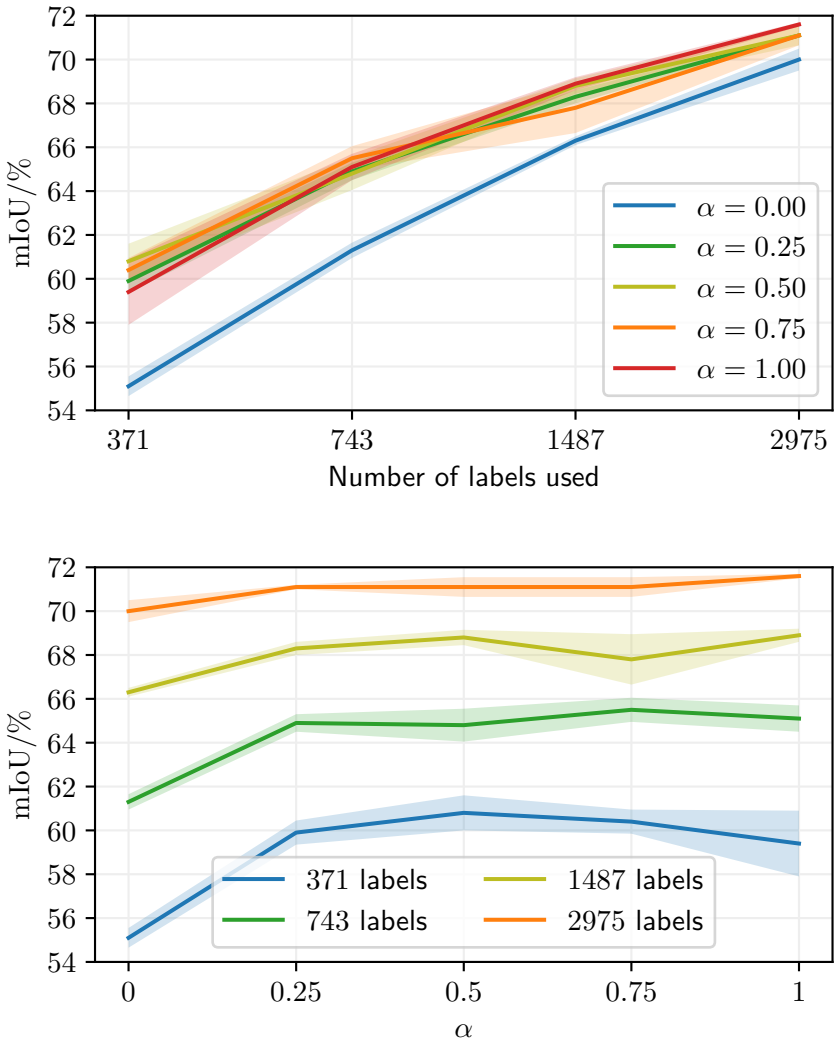
Nadalje, vjerojatnost slučajnog permutiranja kanala mijenja u $\min\{s_P, 1\}$. Prema tome, $s_P = 0$ mijenja fotometrijsku perturbaciju u funkciju identiteta. Stupac 743 oznake u tablici 5.2 koristi iste polunadzirane konfiguracije s jačinama $s_G, s_P \in \{0, 1\}$. U slučaju kad $(s_G, s_P) = (0, 0)$, ukupna perturbacija je funkcija identiteta, pa učenik i učitelj dobivaju isti ulaz i daju isti izlaz, što daje konzistencijski gubitak 0. Taj se slučaj razlikuje od nadzirane osnovice po tome što se statistike normalizacije po grupi ažuriraju u učeniku.

Slika 5.1 prikazuje da je fotometrijska komponenta perturbacije puno važnija i da jača fotometrijska komponenta može malo kompenzirati slabiju geometrijsku komponentu. Naš odabir jačine perturbacije $(s_G, s_P) = (1, 1)$ je blizu optimumu, za koji eksperimenti upućuju da je oko $(1, 0.5)$, ali i mnoge druge dovoljno velike vrijednosti su podjednako prikladne. Razlike u odnosu na tablicu 5.2 dolaze od varijance — procijenjene standardne pogreške sredine 5 pokretanja su između $0.7/\sqrt{5} \approx 0.31$ i $1.1/\sqrt{5} \approx 0.49$.



Slika 5.1: Validacija jačine perturbacija na validacijskom skupu Cityscapesa na pola rezolucije (mIoU/%). Eksperimenti su pokretani s 5 različitih podjela skupa za učenje uz 743 oznake. Hiperparametri s_P (fotometrijski) i s_G (geometrijski) su definirani u glavnom tekstu. SD označava standardno odstupanje.

Slika 5.2 prikazuje validaciju težine konzistencijskog gubitka α za simple-PhTPS s modelom SwiftNet-RN18. Najbolju generalizaciju postiže $\alpha \in [0.25..0.75]$. Ne skaliramo korak učenja s $(1+\alpha)^{-1}$ jer koristimo optimizacijski algoritam koji je invarijantan na skaliranje.



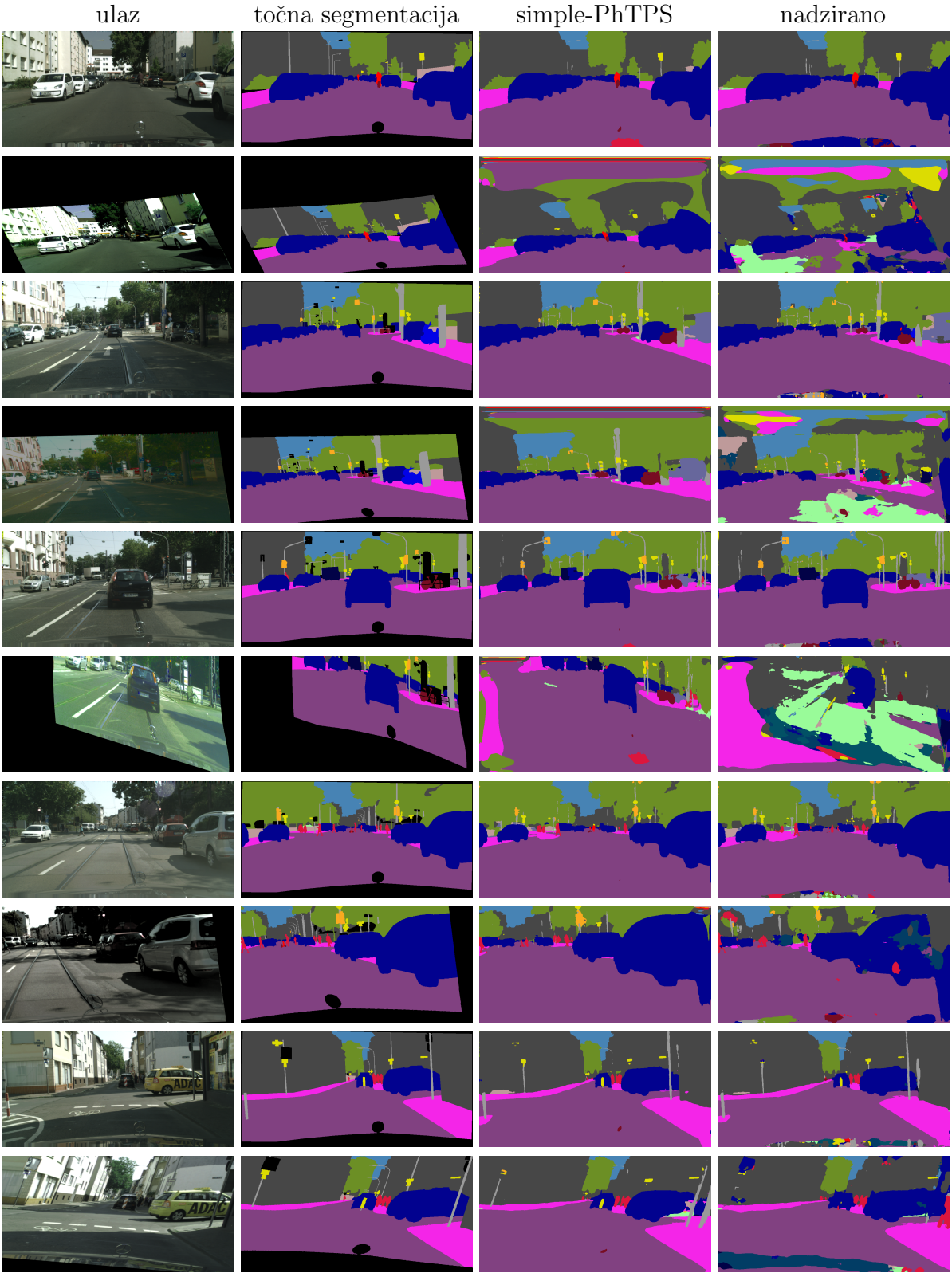
Slika 5.2: Validacija težine konzistencijskog gubitka α na validacijskom skupu Cityscapesa na pola rezolucije (mIoU/%). Isti rezultati su prikazani u dva grafa s različitim x-osima: broj oznaka (gore) i težina konzistencijskog gubitka α (dolje).

Dodatak 5.8 prikazuje učinak ažuriranja statistika normalizacije po grupi kad su ulazi perturbirani.

Slika 5.3 prikazuje kvalitativne rezultate simple-PhTPS-a na prvih nekoliko validacijskih slika sa SwiftNetom-RN18 učenim sa 743 oznake. Vidi se veća robusnost na jake perturbacije kao one tijekom učenja.

5.4 Semantička segmentacija na Cityscapesu na punoj rezoluciji

Tablica 5.3 prikazuje eksperimente na punoj rezoluciji s postavkama kao u tablici 5.1 i usporedbu s prethodnim radovima. U usporedbi s KE-GAN-om [113] (koji u učenju



Slika 5.3: Kvalitativni rezultati na prvih nekoliko validacijskih slika Cityscapesa na pola rezolucije od učenja SwiftNeta-RN18 sa 743 oznake. U neparnim recima su originalne, a u parnim recima su slike perturbirane PhTPS-om. Stupci su (s lijeva na desno): slika, točna segmentacija, predikcija simple-PhTPS-a i predikcija nadziranog učenja.

koristi tekstovni korpus) i ECS-om [116], SwiftNet-RN18 simple-PhTPS radi lošije na 1/8 (371) oznaka, a bolje na 1/2 ili više oznaka. DLv3$^+$-RN50 ECS zahtijeva 22 GiB grafičke memorije s veličinom mini-grupe 6 [116], dok SN-RN18 simple-PhTPS zahtijeva manje od 8 GiB s veličinom mini-grupe 8. Dodatak A.2 prikazuje detaljnije usporedbe memorije i vremena izvođenja.

DLv3$^+$-RN50 CAC [118] na svim udjelima oznaka ima bolje rezultate od SwiftNeta-RN18. Zato donji odjeljak još prikazuje eksperimente s okosnicom ResNet-34 umjesto ResNeta-18. Takav model i dalje ima manji kapacitet, ali uglavnom nadmašuje DLv3$^+$-RN50 CAC-a (simple-PhTPS je bolji na 1/1, a MT-PhTPS na 1/8 oznaka).

Učenje DLv3$^+$-RN50 CAC-a zahtijeva 3 grafičke kartice RTX 2080Ti [118], dok SN-RN34 simple-PhTPS zahtijeva manje od 9 GiB grafičke memorije i stane na jednu takvu grafičku karticu. SwiftNet-RN34 ima i oko 4× brže zaključivanje inference od DeepLaba v3$^+$-RN50 na RTX 2080Ti.

Tablica 5.3: Generalizacijska performansa semantičke segmentacije (mIoU/%) na validacijskom skupu Cityscapesa na punoj rezoluciji s različitim udjelima označenih podataka. Uspoređujemo simple-PhTPS i MT-PhTPS s nadziranim učenjem i prethodnim radom. DLv3$^+$-RN50 označava DeepLab v3$^+$ s okosnicom ResNet-50, a SN označava SwiftNet. Eksperimente pokrećemo s 5 različitih podjela na označeni i neoznačeni skup i izvještavamo srednje mIoU-ove sa standardnim odstupanjima. Rezultati su izraženi prema opisu u odjeljku 5.2.3.

Postupak	**Broj oznaka**			
	371	743	1487	2975 (sve)
KE-GAN [113]	66.9	70.6	72.2	75.3
DLv3$^+$-RN50 nadzirano [116]	63.2	68.4	<u>72.9</u>	74.8
DLv3$^+$-RN50 ECS [116]	67.4	70.7	<u>72.9</u>	74.8
DLv3$^+$-RN50 nadzirano [118]	63.9	68.3	71.2	76.3
DLv3$^+$-RN50 CAC [118]	<u>69.7</u>	<u>72.7</u>	–	<u>77.5</u>
SN-RN18 nadzirano	$61.1_{0.4}$	$67.3_{1.1}$	$71.9_{0.1}$	$75.4_{0.4}$
SN-RN18 simple-PhTPS	$66.3_{1.0}$	$71.0_{0.5}$	<u>74.3</u>$_{0.7}$	<u>75.8</u>$_{0.4}$
SN-RN18 MT-PhTPS	<u>68.6</u>$_{0.6}$	<u>72.0</u>$_{0.3}$	$73.8_{0.4}$	$75.0_{0.4}$
SN-RN34 nadzirano	$64.9_{0.8}$	$69.8_{1.0}$	$73.8_{1.4}$	$76.1_{0.8}$
SN-RN34 simple-PhTPS	$69.2_{0.8}$	$73.1_{0.7}$	$\mathbf{76.3}_{0.7}$	$\mathbf{77.9}_{0.2}$
SN-RN34 MT-PhTPS	$\mathbf{70.8}_{1.5}$	$\mathbf{74.3}_{0.5}$	$\mathbf{76.0}_{0.5}$	$77.3_{0.4}$

Sljedeći eksperimenti ispituju režim s puno podataka u kojem se koristi cijeli fino označeni podskup Cityscapesa (koji je dosad korišten) u $\mathbb{D}_\mathrm{l}$ i dodatni veliki grubo označeni skup ("train-extra"). Indeksi "l" (labeled) i "u" (unlabeled) označavaju koristi li se oznake. Prema tome $\mathrm{C_u}$ (od engl. *coarse*) u tablici označava da se grubo označeni podskup koristi bez oznaka. Tablica 5.4 ispituje učinak grubog podskupa na generalizacijsku performansu SwiftNeta na validacijskom skupu Cityscapesa na punoj rezoluciji.

U eksperimentima s cijelim fino označenim skupom, polunadzirano učenje (stupac

$(\mathrm{F_l}, \mathrm{F_u})$) je pomoglo u odnosu na nadzirano učenje (stupac $\mathrm{F_l}$) samo u slučaju jednostavnog učitelja. Poboljšanje je veće u slučaju SwiftNeta-RN34 (1.8 pb) nego u slučaju SwiftNeta-RN18 (0.4 pb).

Zanimljiva je i usporedba s nadziranim učenjem koje koristi i dodatni grubo označeni skup (stupac $(\mathrm{F_l}, \mathrm{C_l})$). U slučaju SwiftNeta-RN18, simple-PhTPS je lošiji (-0.4 pb), MT-PhTPS je puno lošiji (-1.4 pb), dok su u slučaju SwiftNeta-RN34, simple-PhTPS i MT-PhTPS značajno bolji (0.8 pb i 1 pb) od nadziranog učenja sa svim dostupnim oznakama. Ovi eksperimenti pokazuju da polunadzirano učenje može biti dobra alternativa velikom broju grubih oznaka.

Tablica 5.4: Učinci dodatnog velikog skupa u nadziranom i polunadziranom učenju na validacijskom skupu Cityscapesa na punoj rezoluciji (mIoU/%). F i C označavaju fino (engl. *fine*) i grubo (engl. *coarse*) označene skupove. Indeksi skupova označavaju koriste li se oznake (l) ili polunadzirano učenje (u).

Postupak	$\mathrm{F_l}$	$(\mathrm{F_l}, \mathrm{F_u})$	$(\mathrm{F_l}, \mathrm{F_u} \cup \mathrm{C_u})$	$(\mathrm{F_l}, \mathrm{C_l})$
SN-RN18 simple-PhTPS	$75.4_{0.4}$	$75.8_{0.4}$	$76.5_{0.3}$	$\mathbf{76.9}_{0.3}$
SN-RN18 MT-PhTPS		$75.0_{0.4}$	$75.5_{0.3}$	
SN-RN34 simple-PhTPS	$76.1_{0.8}$	$77.9_{0.2}$	$\mathbf{78.5}_{0.4}$	$77.7_{0.4}$
SN-RN34 MT-PhTPS		$77.3_{0.4}$	$\mathbf{78.7}_{0.3}$	

5.5 Semantička segmentacija na skupu PASCAL VOC

Tablica 5.5 uspoređuje predloženi pristup s prethodnim stanjem tehnike i paralelnim pristupom. Gornji odjeljak predstavlja postupke iz literature [10, 118]. Naša osnovica zbog nejasnih razloga daje bolje rezultate od prethodnog rada (razlike su opisane u odjeljku 5.2). Za svaki broj oznaka koristimo po 3 podskupa kao u [118]. Za razliku od naših eksperimenata na drugim skupovima, ovdje kao [118] ne uključujemo označene primjere iz $\mathbb{D}_\mathrm{l}$ u skup primjera za konzistencijski gubitak $\mathbb{D}_\mathrm{u}$. Donji odjeljak pokazuje rezultate naše metode s istim hiperparametrima kao u eksperimentima s Cityscapesom. jednostavna konzistencija (simple-PhTPS) je dala poboljšanja između 1.9 pb i 3.7 pb u odnosu na osnovicu u slučajevima kad se ne koriste sve oznake, dok nema velike razlike kad se koriste sve oznake. Međutim, usrednjeni učitelj (MT-PhTPS) je dao lošije rezultate od osnovice, što upućuje na to da nisu prikladni isti hiperparametri.

5.6 Klasifikacija slika na skupu CIFAR-10

Tablica 5.6 evaluira klasifikaciju slika na skupu CIFAR-10 uz 2 nadzirane osnovice i 4 polunadzirana algoritma. Prvi nadzirani algoritam koristi samo označene podatke sa

Tablica 5.5: Generalizacijska performansa semantičke segmentacije (mIoU/%) na validacijskom skupu Pascal VOC-a s različitim udjelima označenih podataka uz učenje na proširenom skupu sa učenje. Uspoređujemo simple-PhTPS i MT-PhTPS s nadziranim učenjem i prethodnim radom. DLv3$^+$-RN50 označava DeepLab v3$^+$ s okosnicom ResNet-50. Eksperimente pokrećemo s 3 različite podjele na označeni i neoznačeni skup prema [118] i izvještavamo srednje mIoU-ove sa standardnim odstupanjima. Rezultati su izraženi prema opisu u odjeljku 5.2.3.

Postupak	**Broj oznaka**			
	662	1323	2645	10582 (sve)
DLv3$^+$-RN50 nadzirano [118]	63.9	68.3	71.2	**76.3**
DLv3$^+$-RN50 ECS [116]	–	70.2	72.6	**76.3**
DLv3$^+$-RN50 CAC [118]	**70.1**	<u>72.4</u>	<u>74.0</u>	**76.5**
DLv3$^+$-RN50 nadzirano	$66.8_{0.5}$	$70.7_{0.8}$	$73.2_{0.5}$	$\mathbf{76.2}_{0.2}$
DLv3$^+$-RN50 simple-PhTPS	$\mathbf{70.5}_{1.3}$	$\mathbf{73.5}_{0.4}$	$\mathbf{75.1}_{0.5}$	$\mathbf{76.4}_{0.2}$
DLv3$^+$-RN50 MT-PhTPS	$64.4_{1.0}$	$69.7_{0.3}$	$71.3_{0.3}$	$75.8_{0.4}$

standardnim rastresanjem. Drugi algoritam računa srednji nadzirani gubitak sa standardnim rastresanjem i standardnim rastresanjem uz dodatno perturbiranje PhTPS-om. Treći algoritam je VAT s minimizacijom entropije [6]. Četvrti nadzirani algoritam je naš postupak simple-PHTPS. Naš postupak bolje generalizira od nadziranih pristupa i VAT-a. Jednosmjerna konzistenicja s čistim učenikom najlošije generalizira i u svim slučajevima smanjuje točnost u odnosu na nadziranu osnovicu, a dvosmjerna konzistencija s jednim čistim ulazom pomaže samo kad se koriste sve oznake. Bolja generalizacija na klasifikaciji slika može se postići kombiniranjem konzistencije s drugim postupcima koji se mogu pronaći u literaturi [7, 92, 108] i izvan su opsega ovog rada.

Tablica 5.6: Klasifikacijska točnost [%] na ispitnom podskupu skupa CIFAR-10 s modelom WRN-28-2. U gornjem odjeljku su dva nadzirana pristupa: prvo je obično nadzirano učenje, a drugo računa srednji nadzirani gubitak sa standardnim rastresanjem i standardnim rastresanjem uz dodatno perturbiranje PhTPS-om. U drugom odjeljku prvi redak je VAT s minimizacijom entropije [6], a zadnja 3 retka su dvosmjerna i jednosmjerna konzistencija s perturbacijama PhTPS-a. Rezultati su sredine i standardna odstupanja u 5 pokretanja s različitim označenim podskupovima.

Postupak	**Broj oznaka**			
	250	1000	4000	50000
nadzirano	$31.8_{0.6}$	$59.3_{1.4}$	$81.0_{0.2}$	$94.6_{0.2}$
nadzirano PhTPS-aug	<u>$48.7_{0.9}$</u>	<u>$67.2_{0.5}$</u>	<u>$81.7_{0.2}$</u>	<u>$94.8_{0.1}$</u>
VAT + entropy minimization	$41.0_{2.5}$	$73.2_{1.5}$	$84.2_{0.4}$	$90.5_{0.2}$
1w-cs simple-PhTPS	$27.7_{5.9}$	$51.7_{3.5}$	$69.3_{4.2}$	$91.6_{1.5}$
2w-c1 simple-PhTPS	$30.3_{1.8}$	$54.8_{1.5}$	$72.9_{2.6}$	$95.9_{0.2}$
1w-ct simple-PhTPS	$\mathbf{68.8}_{5.4}$	$\mathbf{84.2}_{0.4}$	$\mathbf{90.6}_{0.4}$	$\mathbf{96.2}_{0.2}$

5.7 Validacija oblika konzistencije

Tablica 5.7 prikazuje eksperimente s nadziranim učenjem i 4 oblika konzistencijskog polunadziranog učenja. Svi polunadzirani eksperimenti koriste iste perturbacije PhTPS-a na skupu CIFAR-10 (4000 oznaka i 50000 slika) i na Cityscapesu na pola rezolucije sa 743 oznake uz postavke kao u tablici 5.1).

Koristimo oznake 1w (*one-way*) i 2w (*two-way*) za smjer konzistencije i ct (*clean teacher*), cs (*clean student*) i 2p (oba ulaza perturbirana) za označavanje koji ulazi su perturbirani. Ispitujemo sljedeće oblike konzistencije: jednosmjerna s čistim učiteljem (1w-ct, usp. sa slikom 1.1c), jednosmjerna s čistim učenikom (1w-cs, usp. sa slikom 1.1b), dvosmjerna s jednim čistim ulazom (2w-c1, usp. sa slikom 1.1d) i jednosmjerna s perturbiranjem obaju ulaza (1w-p2).

Valja primijetiti da dvosmjerna konzistencija nije moguća s usrednjenim učiteljem i da, kad se oba ulaza perturbiraju (1w-p2), treba primijeniti inverznu geometrijsku transformaciju na gustim predikcijama kao kod [21]. Za inverznu transformaciju koristimo postupak unaprijednog izobličenja opisan u [131] s istim poljem pomaka. Dvosmjerna konzistencija perturbiranjem obaju ulaza (2w-p2) je isto moguća. Za nju se može očekivati slično ponašanje kao kod 1w-2p jer se može gledati kao zbroj dviju međusobno obrnutih jednsmjernih konzistencija i na to upućuju naši rani eksperimenti.

U tablici jednosmjerna konzistencija s čistim učiteljem (1w-ct) daje bolje rezultate od drugih oblika konzistencije. Dvosmjerna konzistencija s jednim perturbiranim ulazom (2w-c1) je između dvaju oblika jednosmjerne konzistencije (1w-ct i 1w-cs). To ide u korist hipotezi da su predikcije u čistim primjerima bolji konzistencijski ciljevi. Jednosmjerna konzistencija s oba ulaza perturbirana (1w-p2) često radi bolje nego s čistim učenikom (1w-cs), ali je uvijek lošija od čisotg učitelja (1w-ct). Pažljiviji pogled upućuje na to da obostrano perturbiranje (1w-p2) ponekad nauči varati konzistencijski gubitak davanjem sličnih izlaza za perturbirane slike. To se češće događa kad normalizacija po grupi koristi statistike mini-grupe procijenjene tijekom učenja. To se nije pojavilo u eksperimentima s perturbacijama CutMix-a, koje su blaže od PhTPS-a. Slično varanje se dogodilo u slučaju čistog učitelja (1w-cs) na Cityscapesu, ali još i uz jaku prenaučenost na skupu za učenje.

Polunadzirani eksperimenti na skupu CamVid [132] s dodatnim neoznačenim slikama iz istih videa dali su slične odnose između konzistencijskih varijanti, ali učenje s istim hiperparametrima kao za Cityscapes nije postiglo bolje rezultate od nadziranog učenja i ukupan učinak konzistencijskog gubitka je bio slabiji.

Tablica 5.7: Usporedba nadziranog učenja (nadz.) i 4 oblika konzistencije pod perturbacijama PhTPS-a: jednosmjerna s čistim učiteljem (1w-ct), jednosmjerna s čistim učenikom (1w-cs), dvosmjerna s jednim čistim ulazom (2w-c1), and jednosmjerna s perturbiranjem obaju ulaza (1w-p2). Algoritmi se evaluiraju na ispitnom podskupu skupa CIFAR-10 (točnost/%) uz 4000/50000 oznaka (CIFAR-10, 4k) i validacijskom skupu Cityscapesa na pola rezolucije (mIoU/%) uz učenje sa 743 oznake iz skupa za učenje (CS-half, 743).

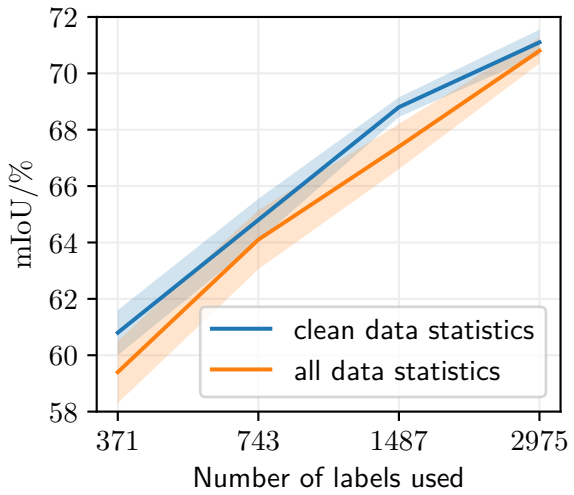
Skup	**Postupak**	nadz.	1w-ct	1w-cs	2w-c1	1w-p2
CIFAR-10, 4k	WRN-28-2 simple-PhTPS	$80.8_{0.4}$	$\mathbf{90.8}_{0.3}$	$69.3_{4.2}$	$72.9_{2.6}$	$73.3_{7.0}$
CIFAR-10, 4k	WRN-28-2 MT-PhTPS	$80.8_{0.4}$	$\mathbf{90.8}_{0.4}$	$80.5_{0.5}$	-	$73.4_{1.4}$
CS-half, 743	SN-RN18 simple-PhTPS	$61.5_{0.5}$	$\mathbf{65.3}_{1.9}$	$1.6_{1.0}$	$16.7_{3.0}$	$61.6_{0.5}$
CS-half, 743	SN-RN18 MT-PhTPS	$61.5_{0.5}$	$\mathbf{66.0}_{1.0}$	$61.5_{1.4}$	-	$62.0_{1.1}$

5.8 Validiranje normalizacije po grupi u perturbiranom učeniku

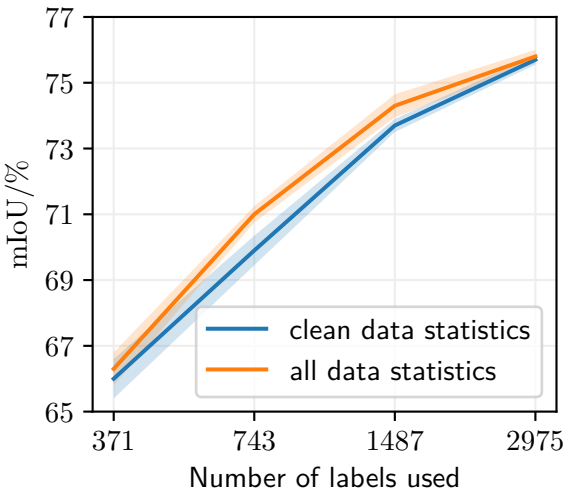
Komponente normalizacije po grupi tijekom učenja procjenjuju populacijske statistike značajki eksponencijalnim pokretnim prosjecima. Učenje na perturbiranim slikama može nepovoljno utjecati na prikladnost tih procjena. Zato ovdje ispitujemo ažuriranje statistika samo na čistim ulazima (kad se računa nadzirani gubitak) umjesto i na perturbiranim ulazima. Valja primijetiti da razlike u ažuriranju statistika ne utječu na optimizaciju parametara zato što se tijekom učenja uvijek koriste statistike mini-grupe.

Slika 5.4 prikazuje učinak isključivanja ažuriranja statistika normalizacije po grupi kad učenik prima perturbirane ulaze u našem polunadziranom učenju s jednosmjernom konzistencijom s čistim učiteljem. Eksperimenti koriste konfiguracije jednostavne jednosmjerne konzistencije (simple-PhTPS) s modelom SwiftNet-RN18 kao u tablicama 5.1 i 5.3 i simple-PhTPS s modelom WRN-28-2 kao u tablici 5.6. U slučaju skupova Cityscapes na pola rezolucije i PASCAL VOC, isključivanje ažuriranja u perturbiranom učeniku je povećalo validacijski mIoU za između 0.3 pb i 1.4 pb, ovisno o udjelu korištenih oznaka. Međutim, u slučaju Cityscapesa na punoj rezoluciji, događa se suprotno – mIoU je opao za između 0.1 pb i 1.1 pb. U eksperimentima na skupu CIFAR-10, učinak je slab.

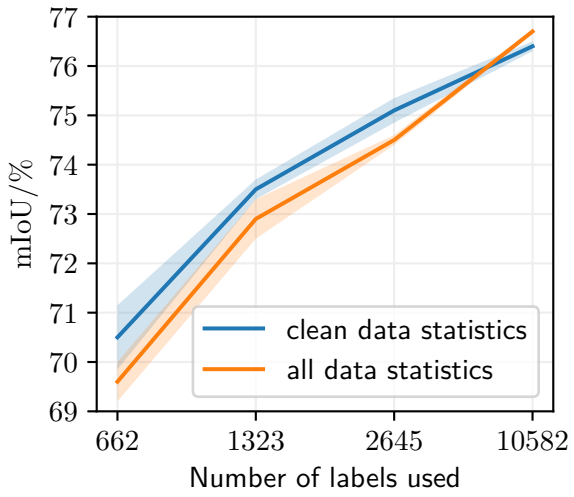
Slika 5.5 prikazuje učinak isključivanja ažuriranja statistika normalizacije po grupi na dodatnim eksperimentima na skupu PASCAL VOC. Vidi se da hiperparametri koji su prikladni za jednostavnu konzistenciu i usrednjenog učitelja na Cityscapesu i jednostavnu konzistenciju na skupu PASCAL VOC ne rade dobro s usrednjenim učiteljem na skupu PASCAL VOC. Po performansi na skupu za učenje, koja nije prikazana, vidi se da se događa prenaučenost koja počinje blizu početka učenja. Za sprječavanje takvih pojava literatura predlaže da se u prvih nekoliko epoha učenja primjenjuje samo nadzirani gubitak [28, 118]. U ovom radu se radi jednostavnosti konzistencijski gubitak primjenjuje od početka učenja.



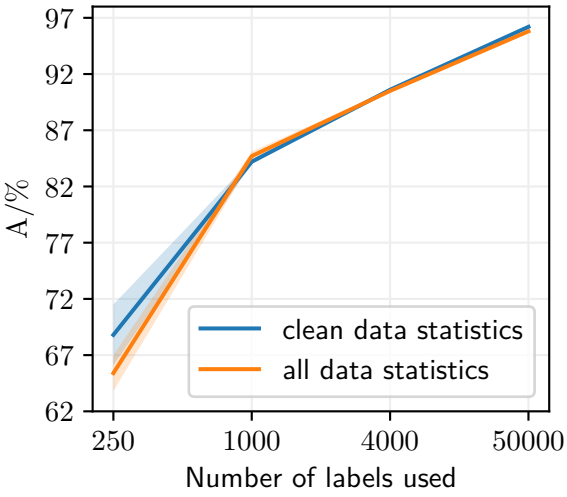
(a) Validacijski podskup Cityscapesa na pola rezolucije.



(b) Validacijski podskup Cityscapesa na punoj rezoluciji.

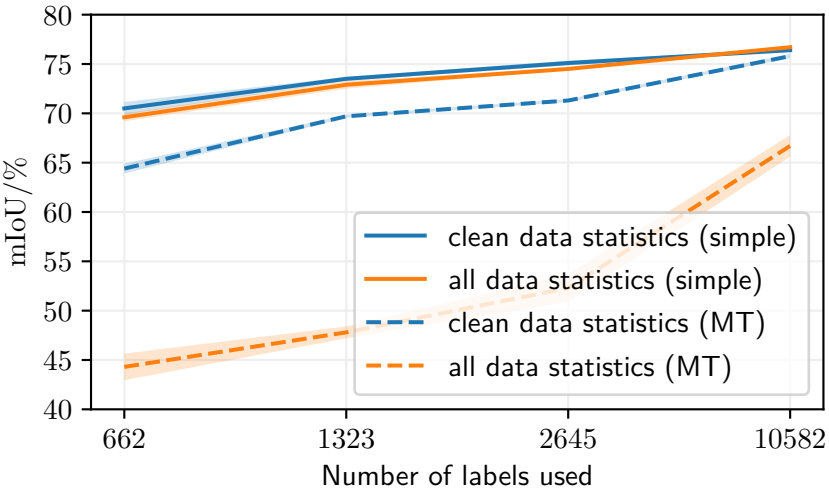


(c) Validacijski podskup skupa PASCAL VOC.



(d) Validacijski podskup skupa CIFAR-10.

Slika 5.4: Učinak ažuriranja statistika normalizacije po grupi u perturbiranom učeniku.



Slika 5.5: Učinak ažuriranja statistika normalizacije po grupi u perturbiranom učeniku na mIoU na skupu PASCAL VOC. Puna crta odgovara algoritmu simple-PhTPS, a crtkana odgovara algoritmu MT-PhTPS.

Poglavlje 6

Zaključak

Nadzirano učenje jedan je od najvažnijih mehanizama za ostvarivanje vizualnih kompetencija u autonomnoj vožnji, medicinskoj dijagnostici i brojnim drugim industrijskim primjenama. Međutim, skaliranje takvih pristupa otežavaju poteškoće vezane uz označavanje podataka. Te poteškoće posebno su ozbiljne kod semantičke segmentacije i ostalih vrsta guste predikcije zbog velike zahtjevnosti izrade oznaka na razini piksela.

Opisane probleme možemo ublažiti različitim oblicima učenja s nepotpunim nadzorom. Ova disertacija razmatra konzistencijsko polunadzirano učenje. Predložena metoda zasniva se na konzistencijskom učenju gdje zahtijevamo da različite perturbacije neoznačene slike dovode do konzistentnih segmentacijskih predikcija. Glavni doprinosi istraživanja odnose se na iscrpnu karakterizaciju i validaciju oblika konzistencijskog učenja, oblikovanje učinkovitog postupka učenja i izradu perturbacijskog modela za konzistencijsko polunadzirano učenje semantičku segmentaciju.

Rad pokazuje da jednosmjerna konzistencija s čistim učiteljem bolje generalizira od drugih oblika konzistencije s obzirom na odabire grana na kojima se primjenjuju perturbacije i računaju gradijenti nenadziranog gubitka. Jednosmjerna konzistencija ima i prednost u smislu da pažljiva implementacija može imati memorijske zahtjeve slične nadziranom učenju. Na bolju generalizacijsku performansu s čistim učiteljem ima utjecaj to što su predikcije u čistim slikama češće manje točne od predikcija u perturbiranim slikama. Nedostatak perturbiranja obiju grana je i veća sklonost učenju trivijalnih rješenja koja sve perturbirane ulaze preslikavaju u isti izlaz.

Rad predlaže učinkovit algoritam za polunadzirano učenje semantičke segmentacije koji se temelji na jednosmjernoj konzistenciji s čistim učiteljem i predloženom perturbacijskom modelu, koji je kompozicija fotometrijskih i geometrijskih transformacija. Konzistencijski gubitak potiče da primjerak modela bude invarijantan na fotometrijske perturbacije i ekvivarijantan na geometrijske perturbacije. Predloženi algoritam je nadmašio sve prethodne i suvremene postupke: u eksperimentima na podatkovnom skupu Cityscapes

zabilježili smo poboljšanu generalizacijsku performansu kod različitih količina označenih i neoznačenih podataka za učenje, kako za standardni model DeepLabv2-RN101 tako i za učinkovite modele SwiftNet-RN18 i SwiftNet-RN34. Učinkoviti model SwiftNet-RN18 generalizira slično kao i DeepLabv2-RN101, dok mu je zaključivanje oko 9× brže na pola rezolucije i oko 15× brže na punoj rezoluciji na RTX 2080Ti. Polunadzirano učenje s učinkovitim modelima može biti korisno za razvoj algoritama i praktične primjene u stvarnom vremenu kada veliki označeni skupovi nisu dostupni. Jednostavnost, kompetitivna performansa i računska učinkovitost čine ovakav pristup zanimljivom osnovicom za vrednovanje novih polunadziranih postupaka za gustu predikciju.

Rad uspoređuje i dva oblika učitelja za jednosmjernu konzistenciju. Kod jednostavne konzistencije učitelj ima iste parametre kao učenik, dok su kod usrednjenog učitelja učiteljevi parametri pokretni prosjek učenikovih. Eksperimenti na Cityscapesu upućuju na to da usrednjeni učitelj bolje generalizira od jednostavne konzistencije kad su podaci na manjoj rezoluciji ili kad je broj oznaka manji. Međutim, kad se koristi puno podataka i oznaka, jednostavna konzistencija i dalje nadmašuje nadzirano učenje, dok usrednjeni učitelj u nekim slučajevima lošije generalizira. Eksperimenti s dodatnim grubo označenim podskupom Cityscapesa pokazuju da polunadzirano učenje može biti bolja alternativa dodavanju velikog broja grubih oznaka u nadzirano učenje.

Postoji više smjerova budućeg rada koji se mogu nastaviti na ovo istraživanje. Bilo bi zanimljivo bolje razumjeti dinamiku optimizacije s jednosmjernim gubitkom. Osim toga nije jasno ni što uzrokuje razlike u ponašanju jednostavnog i usrednjenog učitelja u različitim uvjetima. Konačno, moglo bi biti korisno istražiti i robusnije oblike konzistencijskog gubitka koje bi povećale otpornost postupka na netočne predikcije učitelja u ranim iteracijama učenja.

Literatura

[1]Chapelle, O., Schlkopf, B., Zien, A., Semi-Supervised Learning, 1st ed. The MIT Press, 2010.

[2]Huang, Z., Wang, X., Wang, J., Liu, W., Wang, J., "Weakly-supervised semantic segmentation network with deep seeded region growing", 2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2018, str. 7014-7023, dostupno na: https://api.semanticscholar.org/CorpusID:51690586

[3]Kolesnikov, A., Zhai, X., Beyer, L., "Revisiting self-supervised visual representation learning", in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, str. 1920–1929.

[4]Wilson, G., Cook, D. J., "A survey of unsupervised deep domain adaptation", ACM Trans. Intell. Syst. Technol., Vol. 11, No. 5, jul 2020, dostupno na: https://doi.org/10.1145/3400066

[5]Tarvainen, A., Valpola, H., "Mean teachers are better role models: Weight-averaged consistency targets improve semi-supervised deep learning results", in Advances in neural information processing systems, 2017, str. 1195–1204.

[6]Miyato, T., Maeda, S., Koyama, M., Ishii, S., "Virtual adversarial training: A regularization method for supervised and semi-supervised learning", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 41, No. 8, 2019, str. 1979–1993.

[7]Xie, Q., Dai, Z., Hovy, E. H., Luong, T., Le, Q., "Unsupervised data augmentation for consistency training", in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., (ur.), 2020.

[8]Souly, N., Spampinato, C., Shah, M., "Semi supervised semantic segmentation using generative adversarial network", in IEEE International Conference on Computer

Vision, ICCV 2017, Venice, Italy, October 22-29, 2017. IEEE Computer Society, 2017, str. 5689–5697.

[9]Hung, W., Tsai, Y., Liou, Y., Lin, Y., Yang, M., "Adversarial learning for semi-supervised semantic segmentation", in BMVC, 2018, str. 65.

[10]Mittal, S., Tatarchenko, M., Brox, T., "Semi-supervised semantic segmentation with high- and low-level consistency", IEEE Transactions on Pattern Analysis and Machine Intelligence, 2019, str. 1-1.

[11]Cordts, M., Omran, M., Ramos, S., Scharw ächter, T., Enzweiler, M., Benenson, R., Franke, U., Roth, S., Schiele, B., "The cityscapes dataset", in CVPRW, 2015.

[12]Neuhold, G., Ollmann, T., Rota Bulò, S., Kontschieder, P., "Mapillary vistas dataset for semantic understanding of street scenes", in ICCV, 2017, str. 5000–5009.

[13]Maggiori, E., Tarabalka, Y., Charpiat, G., Alliez, P., "Can semantic labeling methods generalize to any city? the inria aerial image labeling benchmark", in 2017 IEEE International Geoscience and Remote Sensing Symposium, IGARSS 2017, Fort Worth, TX, USA, July 23-28, 2017. IEEE, 2017, str. 3226–3229.

[14]Rota Bulò, S., Porzi, L., Kontschieder, P., "In-place activated batchnorm for memory-optimized training of DNNs", in CVPR, June 2018.

[15]Salimans, T., Goodfellow, I. J., Zaremba, W., Cheung, V., Radford, A., Chen, X., "Improved techniques for training gans", in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain, 2016, str. 2226–2234.

[16]Tsai, Y., Hung, W., Schulter, S., Sohn, K., Yang, M., Chandraker, M., "Learning to adapt structured output space for semantic segmentation", in 2018 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2018, Salt Lake City, UT, USA, June 18-22, 2018. IEEE Computer Society, 2018, str. 7472–7481.

[17]Gao, H., Yao, D., Wang, M., Li, C., Liu, H., Hua, Z., Wang, J., "A hyperspectral image classification method based on multi-discriminator generative adversarial networks", Sensors, Vol. 19, No. 15, 2019, dostupno na: https://www.mdpi.com/1424-8220/19/15/3269

[18]Rasmus, A., Berglund, M., Honkala, M., Valpola, H., Raiko, T., "Semi-supervised learning with ladder networks", in Advances in Neural Information Processing Systems 28: Annual Conference on Neural Information Processing Systems 2015, December 7-12, 2015, Montreal, Quebec, Canada, 2015, str. 3546–3554.

[19]Sajjadi, M., Javanmardi, M., Tasdizen, T., "Mutual exclusivity loss for semi-supervised deep learning", in Advances in Neural Information Processing Systems 29: Annual Conference on Neural Information Processing Systems 2016, December 5-10, 2016, Barcelona, Spain. IEEE, 2016, str. 1163–1171.

[20]Qiao, S., Shen, W., Zhang, Z., Wang, B., Yuille, A., "Deep co-training for semi-supervised image recognition", in Proceedings of the european conference on computer vision (eccv), 2018, str. 135–152.

[21]Bortsova, G., Dubost, F., Hogeweg, L., Katramados, I., de Bruijne, M., "Semi-supervised medical image segmentation via learning consistency under transformations", in MICCAI, 2019.

[22]Laine, S., Aila, T., "Temporal ensembling for semi-supervised learning", in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017, dostupno na: https://openreview.net/forum?id=BJ6oOfqge

[23]Zheng, S., Song, Y., Leung, T., Goodfellow, I. J., "Improving the robustness of deep neural networks via stability training", in 2016 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2016, Las Vegas, NV, USA, June 27-30, 2016, 2016, str. 4480–4488.

[24]Tsipras, D., Santurkar, S., Engstrom, L., Turner, A., Madry, A., "Robustness may be at odds with accuracy", in 7th International Conference on Learning Representations, ICLR 2019, New Orleans, LA, USA, May 6-9, 2019. OpenReview.net, 2019, dostupno na: https://openreview.net/forum?id=SyxAb30cY7

[25]Lenc, K., Vedaldi, A., "Understanding image representations by measuring their equivariance and equivalence", Int. J. Comput. Vision, Vol. 127, No. 5, may 2019, str. 456–476, dostupno na: https://doi.org/10.1007/s11263-018-1098-y

[26]Grubišić, I., Oršić, M., Šegvić, S., "A baseline for semi-supervised learning of efficient semantic segmentation models", in 17th International Conference on Machine Vision and Applications, MVA 2021, Aichi, Japan, July 25-27, 2021. IEEE, 2021, str. 1–5, dostupno na: https://arxiv.org/abs/2106.07075

[27]Grubišić, I., Oršić, M., Šegvić, S., "Revisiting consistency for semi-supervised semantic segmentation", Sensors, Vol. 23, No. 2, 2023, dostupno na: https://www.mdpi.com/1424-8220/23/2/940

[28]French, G., Laine, S., Aila, T., Mackiewicz, M., Finlayson, G., "Semi-supervised semantic segmentation needs strong, varied perturbations", in BMVC, 2020.

[29]Krešo, I., Krapac, J., Šegvić, S., "Efficient ladder-style densenets for semantic segmentation of large images", IEEE Transactions on Intelligent Transportation Systems, 2020.

[30]Chen, L., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 40, No. 4, 2018, str. 834–848.

[31]Oršić, M., Krešo, I., Bevandić, P., Šegvić, S., "In defense of pre-trained imagenet architectures for real-time semantic segmentation of road-driving images", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, str. 12 607–12 616.

[32]ProofWiki, "Category:Definitions/Probability Theory — ProofWiki", https://proofwiki.org/wiki/Category:Definitions/Probability_Theory, [Online; accessed 2023-2-23]. 2023.

[33]Simon, B., Real Analysis, ser. A comprehensive course in analysis. American Mathematical Society, 2015, dostupno na: https://books.google.hr/books?id=7brzrQEACAAJ

[34]ProofWiki, "Category:Probability Theory — ProofWiki", https://proofwiki.org/wiki/Category:Probability_Theory, [Online; accessed 2023-2-23]. 2023.

[35]Olah, C., "Visual information theory", dostupno na: http://colah.github.io/posts/2015-09-Visual-Information/ 2015.

[36]Kirsch, A., Gal, Y., "A practical & unified notation for information-theoretic quantities in ML", CoRR, Vol. abs/2106.12062, 2021, dostupno na: https://arxiv.org/abs/2106.12062

[37]DeWeese, M. R., Meister, M., "How to measure the information gained from one symbol.", Network, Vol. 10 4, 1999, str. 325-40.

[38]Shannon, C. E., "A mathematical theory of communication", The Bell System Technical Journal, Vol. 27, 1948, str. 379–423, 623–656.

[39]Wikipedia, "Limiting density of discrete points — Wikipedia, the free encyclopedia", https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence, [Online; accessed 2023-3-27]. 2023.

[40]Wikipedia, "Kullback–Leibler divergence — Wikipedia, the free encyclopedia", https://en.wikipedia.org/wiki/Kullback%E2%80%93Leibler_divergence, [Online; accessed 2023-2-23]. 2023.

[41]Grubišić, I., "Nadzirani pristupi za procjenu nesigurnosti predikcija dubokih modela", https://github.com/Ivan1248/deep-learning-uncertainty/blob/master/thesis/diplomski.pdf, 2018.

[42]Wikipedia, "Mutual information — Wikipedia, the free encyclopedia", https://en.wikipedia.org/wiki/Mutual_information, [Online; accessed 2023-5-28]. 2023.

[43]Murphy, K. P., Probabilistic Machine Learning: Advanced Topics. MIT Press, 2023, dostupno na: http://probml.github.io/book2

[44]Garrabrant, S., Benson-Tilsen, T., Critch, A., Soares, N., Taylor, J., "Logical induction", CoRR, Vol. abs/1609.03543, 2016, dostupno na: http://arxiv.org/abs/1609.03543

[45]Lin, H., "Bayesian Epistemology (Stanford Encyclopedia of Philosophy)", https://plato.stanford.edu/entries/epistemology-bayesian/, [Online; accessed 2023-2-23]. 2022.

[46]Tokdar, S. T., Kass, R. E., "Importance sampling: a review", Wiley Interdisciplinary Reviews: Computational Statistics, Vol. 2, 2010.

[47]Brooks, S., Gelman, A., Jones, G., Meng, X.-L., Handbook of Markov Chain Monte Carlo. CRC press, 2011.

[48]Wolpert, D. H., "The lack of a priori distinctions between learning algorithms.", Neural Comput., Vol. 8, No. 7, 1996, str. 1341-1390.

[49]Goodfellow, I., Bengio, Y., Courville, A., Deep Learning. MIT Press, 2016.

[50]Hornik, K., Stinchcombe, M., White, H., "Multilayer feedforward networks are universal approximators", Neural Networks, Vol. 2, No. 5, 1989, str. 359-366.

[51]Glorot, X., Bordes, A., Bengio, Y., "Deep sparse rectifier neural networks", in Proceedings of the Fourteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Gordon, G., Dunson,

D., Dudík, M., (ur.), Vol. 15. Fort Lauderdale, FL, USA: PMLR, 11–13 Apr 2011, str. 315–323, dostupno na: https://proceedings.mlr.press/v15/glorot11a.html

[52]Rumelhart, D. E., Hinton, G. E., Williams, R. J., "Learning representations by back-propagating errors", Nature, Vol. 323, No. 6088, 1986, str. 533–536.

[53]Paszke, A., Gross, S., Massa, F., Lerer, A., Bradbury, J., Chanan, G., Killeen, T., Lin, Z., Gimelshein, N., Antiga, L., Desmaison, A., Kopf, A., Yang, E., DeVito, Z., Raison, M., Tejani, A., Chilamkurthy, S., Steiner, B., Fang, L., Bai, J., Chintala, S., "Pytorch: An imperative style, high-performance deep learning library", in Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019, str. 8024–8035, dostupno na: http://papers.neurips.cc/paper/9015-pytorch-an-imperative-style-high-performance-deep-learning-library.pdf

[54]Abadi, M., Agarwal, A., Barham, P., Brevdo, E., Chen, Z., Citro, C., Corrado, G. S., Davis, A., Dean, J., Devin, M., Ghemawat, S., Goodfellow, I., Harp, A., Irving, G., Isard, M., Jia, Y., Jozefowicz, R., Kaiser, L., Kudlur, M., Levenberg, J., Mané, D., Monga, R., Moore, S., Murray, D., Olah, C., Schuster, M., Shlens, J., Steiner, B., Sutskever, I., Talwar, K., Tucker, P., Vanhoucke, V., Vasudevan, V., Viégas, F., Vinyals, O., Warden, P., Wattenberg, M., Wicke, M., Yu, Y., Zheng, X., "TensorFlow: Large-scale machine learning on heterogeneous systems", dostupno na: https://www.tensorflow.org/ Software available from tensorflow.org. 2015.

[55]Pleiss, G., Chen, D., Huang, G., Li, T., van der Maaten, L., Weinberger, K. Q., "Memory-efficient implementation of densenets", arXiv preprint arXiv:1707.06990, 2017.

[56]Wikipedia, "Automatic differentiation — Wikipedia, the free encyclopedia", https://en.wikipedia.org/wiki/Automatic_differentiation, [Online; accessed 2023-4-15]. 2023.

[57]Masters, D., Luschi, C., "Revisiting small batch training for deep neural networks", ArXiv, Vol. abs/1804.07612, 2018.

[58]G ürbüzbalaban, M., Ozdaglar, A., Parrilo, P. A., "Why random reshuffling beats stochastic gradient descent", Math. Program., Vol. 186, No. 1–2, mar 2021, str. 49–84, dostupno na: https://doi.org/10.1007/s10107-019-01440-w

[59]Sutskever, I., Martens, J., Dahl, G., Hinton, G., "On the importance of initialization and momentum in deep learning", in Proceedings of the 30th International

Conference on Machine Learning, ser. Proceedings of Machine Learning Research, Dasgupta, S., McAllester, D., (ur.), Vol. 28, No. 3. Atlanta, Georgia, USA: PMLR, 17–19 Jun 2013, str. 1139–1147.

[60]Hinton, G., "Neural networks for machine learning, lecture 6a: Overview of mini-batch gradient descent", 2012, dostupno na: http://www.cs.toronto.edu/~tijmen/csc321/slides/lecture_slides_lec6.pdf

[61]Kingma, D. P., Ba, J., "Adam: A method for stochastic optimization", in 3rd International Conference on Learning Representations, ICLR 2015, San Diego, CA, USA, May 7-9, 2015, Conference Track Proceedings, Bengio, Y., LeCun, Y., (ur.), 2015, dostupno na: http://arxiv.org/abs/1412.6980

[62]Glorot, X., Bengio, Y., "Understanding the difficulty of training deep feedforward neural networks", in Proceedings of the Thirteenth International Conference on Artificial Intelligence and Statistics, ser. Proceedings of Machine Learning Research, Teh, Y. W., Titterington, M., (ur.), Vol. 9. Chia Laguna Resort, Sardinia, Italy: PMLR, 13–15 May 2010, str. 249–256, dostupno na: https://proceedings.mlr.press/v9/glorot10a.html

[63]Ioffe, S., Szegedy, C., "Batch normalization: Accelerating deep network training by reducing internal covariate shift", in Proceedings of the 32nd International Conference on Machine Learning, ser. Proceedings of Machine Learning Research, Bach, F., Blei, D., (ur.), Vol. 37. Lille, France: PMLR, 07–09 Jul 2015, str. 448–456, dostupno na: https://proceedings.mlr.press/v37/ioffe15.html

[64]Santurkar, S., Tsipras, D., Ilyas, A., Madry, A., "How does batch normalization help optimization?", in Advances in Neural Information Processing Systems, Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., (ur.), Vol. 31. Curran Associates, Inc., 2018, dostupno na: https://proceedings.neurips.cc/paper_files/paper/2018/file/905056c1ac1dad141560467e0a99e1cf-Paper.pdf

[65]Wu, Y., He, K., "Group normalization", in Computer Vision - ECCV 2018 - 15th European Conference, Munich, Germany, September 8-14, 2018, Proceedings, Part XIII, ser. Lecture Notes in Computer Science, Ferrari, V., Hebert, M., Sminchisescu, C., Weiss, Y., (ur.), Vol. 11217. Springer, 2018, str. 3–19, dostupno na: https://doi.org/10.1007/978-3-030-01261-8_1

[66]Wu, D., "L2 Regularization and Batch Norm", https://blog.janestreet.com/l2-regularization-and-batch-norm/, [Online; accessed 2023-4-30]. 2019.

[67]van Laarhoven, T., "L2 regularization versus batch and weight normalization", CoRR, Vol. abs/1706.05350, 2017, dostupno na: http://arxiv.org/abs/1706.05350

[68]He, K., Zhang, X., Ren, S., Sun, J., "Deep residual learning for image recognition", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2016, str. 770–778.

[69]Huang, G., Liu, Z., Pleiss, G., van der Maaten, L., Weinberger, K. Q., "Convolutional networks with dense connectivity", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 44, No. 12, 2022, str. 8704–8716.

[70]Hochreiter, S., Schmidhuber, J., "Long short-term memory", Neural Comput., Vol. 9, No. 8, nov 1997, str. 1735–1780, dostupno na: https://doi.org/10.1162/neco.1997.9.8.1735

[71]He, K., Zhang, X., Ren, S., Sun, J., "Identity mappings in deep residual networks", in European Conference on Computer Vision, 2016.

[72]Vaswani, A., Shazeer, N., Parmar, N., Uszkoreit, J., Jones, L., Gomez, A. N., Kaiser, L., Polosukhin, I., "Attention is all you need", in Proceedings of the 31st International Conference on Neural Information Processing Systems, ser. NIPS'17. Red Hook, NY, USA: Curran Associates Inc., 2017, str. 6000–6010.

[73]Li, H., Xu, Z., Taylor, G., Studer, C., Goldstein, T., "Visualizing the loss landscape of neural nets", in Advances in Neural Information Processing Systems, Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., (ur.), Vol. 31. Curran Associates, Inc., 2018, dostupno na: https://proceedings.neurips.cc/paper_files/paper/2018/file/a41b3bb3e6b050b6c9067c67f663b915-Paper.pdf

[74]Veit, A., Wilber, M., Belongie, S., "Residual networks behave like ensembles of relatively shallow networks", in Proceedings of the 30th International Conference on Neural Information Processing Systems, ser. NIPS'16. Red Hook, NY, USA: Curran Associates Inc., 2016, str. 550–558.

[75]Dosovitskiy, A., Beyer, L., Kolesnikov, A., Weissenborn, D., Zhai, X., Unterthiner, T., Dehghani, M., Minderer, M., Heigold, G., Gelly, S., Uszkoreit, J., Houlsby, N., "An image is worth 16x16 words: Transformers for image recognition at scale", in 9th International Conference on Learning Representations, ICLR 2021, Virtual Event, Austria, May 3-7, 2021. OpenReview.net, 2021, dostupno na: https://openreview.net/forum?id=YicbFdNTTy

[76]Dumoulin, V., Visin, F., "A guide to convolution arithmetic for deep learning", ArXiv e-prints, mar 2016.

[77]Long, J., Shelhamer, E., Darrell, T., "Fully convolutional networks for semantic segmentation", in 2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2015, str. 3431-3440.

[78]Ronneberger, O., Fischer, P., Brox, T., "U-net: Convolutional networks for biomedical image segmentation", in Medical Image Computing and Computer-Assisted Intervention - MICCAI 2015 - 18th International Conference Munich, Germany, October 5 - 9, 2015, Proceedings, Part III, ser. Lecture Notes in Computer Science, Navab, N., Hornegger, J., III, W. M. W., Frangi, A. F., (ur.), Vol. 9351, 2015, str. 234–241.

[79]Zhao, H., Qi, X., Shen, X., Shi, J., Jia, J., "Icnet for real-time semantic segmentation on high-resolution images", in ECCV, Vol. 11207, 2018, str. 418–434.

[80]Zhao, H., Shi, J., Qi, X., Wang, X., Jia, J., "Pyramid scene parsing network", in CVPR, 2017.

[81]Oršić, M., Šegvić, S., "Efficient semantic segmentation with pyramidal fusion", Pattern Recognition, Vol. 110, 2021, str. 107611, dostupno na: https://www.sciencedirect.com/science/article/pii/S0031320320304143

[82]Chen, L.-C., Papandreou, G., Kokkinos, I., Murphy, K., Yuille, A. L., "Deeplab: Semantic image segmentation with deep convolutional nets, atrous convolution, and fully connected crfs", IEEE Transactions on Pattern Analysis and Machine Intelligence, Vol. 40, No. 4, 2018, str. 834-848.

[83] Čaušević, D., "Modeliranje supojavljivanja semantičkih oznaka uvjetnim slučajnim poljima", http://www.zemris.fer.hr/~ssegvic/project/pubs/causevic16ms.pdf, 2016.

[84]Sun, K., Xiao, B., Liu, D., Wang, J., "Deep high-resolution representation learning for human pose estimation", in CVPR, 2019.

[85]Zoph, B., Ghiasi, G., Lin, T.-Y., Cui, Y., Liu, H., Cubuk, E. D., Le, Q. V., "Rethinking pre-training and self-training", in Proceedings of the 34th International Conference on Neural Information Processing Systems, ser. NIPS'20. Red Hook, NY, USA: Curran Associates Inc., 2020.

[86]Cheng, B., Schwing, A. G., Kirillov, A., "Per-pixel classification is not all you need for semantic segmentation", 2021.

[87]Zhang, W., Pang, J., Chen, K., Loy, C. C., "K-net: Towards unified image segmentation", CoRR, Vol. abs/2106.14855, 2021, dostupno na: https://arxiv.org/abs/2106.14855

[88]Cheng, B., Misra, I., Schwing, A. G., Kirillov, A., Girdhar, R., "Masked-attention mask transformer for universal image segmentation", 2022.

[89]Kirillov, A., He, K., Girshick, R. B., Rother, C., Dollár, P., "Panoptic segmentation", in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2019, Long Beach, CA, USA, June 16-20, 2019. Computer Vision Foundation / IEEE, 2019, str. 9404–9413, dostupno na: http://openaccess.thecvf.com/content_CVPR_2019/html/Kirillov_Panoptic_Segmentation_CVPR_2019_paper.html

[90]Milletari, F., Navab, N., Ahmadi, S.-A., "V-net: Fully convolutional neural networks for volumetric medical image segmentation", International Conference on 3D Vision, Jun 2016.

[91]Cubuk, E. D., Zoph, B., Mane, D., Vasudevan, V., Le, Q. V., "Autoaugment: Learning augmentation strategies from data", in Proceedings of the IEEE conference on computer vision and pattern recognition, 2019, str. 113–123.

[92]Cubuk, E. D., Zoph, B., Shlens, J., Le, Q., "Randaugment: Practical automated data augmentation with a reduced search space", in Advances in Neural Information Processing Systems 33: Annual Conference on Neural Information Processing Systems 2020, NeurIPS 2020, December 6-12, 2020, virtual, Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M., Lin, H., (ur.), 2020.

[93]M üller, S. G., Hutter, F., "Trivialaugment: Tuning-free yet state-of-the-art data augmentation", in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2021, str. 774-782.

[94]Zhang, H., Cissé, M., Dauphin, Y. N., Lopez-Paz, D., "mixup: Beyond empirical risk minimization", in 6th International Conference on Learning Representations, ICLR 2018, Vancouver, BC, Canada, April 30 - May 3, 2018, Conference Track Proceedings, 2018, dostupno na: https://openreview.net/forum?id=r1Ddp1-Rb

[95]Yun, S., Han, D., Oh, S. J., Chun, S., Choe, J., Yoo, Y., "Cutmix: Regularization strategy to train strong classifiers with localizable features", in Proceedings of the IEEE/CVF International Conference on Computer Vision (ICCV), October 2019.

[96]Grandvalet, Y., Bengio, Y., "Semi-supervised learning by entropy minimization", in Advances in Neural Information Processing Systems, Saul, L. K., Weiss, Y.,

Bottou, L., (ur.). MIT Press, 2005, str. 529–536, dostupno na: http://papers.nips.cc/paper/2740-semi-supervised-learning-by-entropy-minimization.pdf

[97]Yarowsky, D., "Unsupervised word sense disambiguation rivaling supervised methods", in 33rd Annual Meeting of the Association for Computational Linguistics. Cambridge, Massachusetts, USA: Association for Computational Linguistics, 6 1995, str. 189–196, dostupno na: https://aclanthology.org/P95-1026

[98]McClosky, D., Charniak, E., Johnson, M., "Effective self-training for parsing", in Proceedings of the Human Language Technology Conference of the NAACL, Main Conference. New York City, USA: Association for Computational Linguistics, 6 2006, str. 152–159, dostupno na: https://aclanthology.org/N06-1020

[99]hyun Lee, D., "Pseudo-label: The simple and efficient semi-supervised learning method for deep neural networks", in ICMLW, 2013.

[100]Oliver, A., Odena, A., Raffel, C. A., Cubuk, E. D., Goodfellow, I., "Realistic evaluation of deep semi-supervised learning algorithms", in Advances in Neural Information Processing Systems, Bengio, S., Wallach, H., Larochelle, H., Grauman, K., Cesa-Bianchi, N., Garnett, R., (ur.), Vol. 31. Curran Associates, Inc., 2018, dostupno na: https://proceedings.neurips.cc/paper/2018/file/c1fea270c48e8079d8ddf7d06d26ab52-Paper.pdf

[101]Xie, Q., Luong, M., Hovy, E. H., Le, Q. V., "Self-training with noisy student improves imagenet classification", in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. Computer Vision Foundation / IEEE, 2020, str. 10 684–10 695.

[102]Wang, Y., Wang, H., Shen, Y., Fei, J., Li, W., Jin, G., Wu, L., Zhao, R., Le, X., "Semi-supervised semantic segmentation using unreliable pseudo-labels", in IEEE/-CVF Conference on Computer Vision and Pattern Recognition, CVPR 2022, New Orleans, LA, USA, June 18-24, 2022. IEEE, 2022, str. 4238–4247.

[103]Gerken, J. E., Aronsson, J., Carlsson, O., Linander, H., Ohlsson, F., Petersson, C., Persson, D., "Geometric deep learning and equivariant neural networks", CoRR, Vol. abs/2105.13926, 2021.

[104]Wang, Y., Zhang, J., Kan, M., Shan, S., Chen, X., "Self-supervised equivariant attention mechanism for weakly supervised semantic segmentation", in Proc. IEEE Conference on Computer Vision and Pattern Recognition (CVPR), 2020.

[105]Cho, J. H., Mall, U., Bala, K., Hariharan, B., "Picie: Unsupervised semantic segmentation using invariance and equivariance in clustering", in CVPR, 2021.

[106]Patel, G., Dolz, J., "Weakly supervised segmentation with cross-modality equivariant constraints", Medical Image Anal., Vol. 77, 2022, str. 102374.

[107]H äusser, P., Mordvintsev, A., Cremers, D., "Learning by association - A versatile semi-supervised training method for neural networks", in 2017 IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2017, Honolulu, HI, USA, July 21-26, 2017. IEEE Computer Society, 2017, str. 626–635.

[108]Berthelot, D., Carlini, N., Goodfellow, I., Papernot, N., Oliver, A., Raffel, C. A., "Mixmatch: A holistic approach to semi-supervised learning", in Advances in Neural Information Processing Systems 32. Curran Associates, Inc., 2019, str. 5049–5059, dostupno na: http://papers.nips.cc/paper/8749-mixmatch-a-holistic-approach-to-semi-supervised-learning.pdf

[109]Grill, J.-B., Strub, F., Altché, F., Tallec, C., Richemond, P., Buchatskaya, E., Doersch, C., Avila Pires, B., Guo, Z., Gheshlaghi Azar, M., Piot, B., kavukcuoglu, k., Munos, R., Valko, M., "Bootstrap your own latent - a new approach to self-supervised learning", in Advances in Neural Information Processing Systems, Larochelle, H., Ranzato, M., Hadsell, R., Balcan, M. F., Lin, H., (ur.), Vol. 33. Curran Associates, Inc., 2020, str. 21 271–21 284, dostupno na: https://proceedings.neurips.cc/paper/2020/file/f3ada80d5c4ee70142b17b8192b2958e-Paper.pdf

[110]Chen, X., He, K., "Exploring simple siamese representation learning", in IEEE Conference on Computer Vision and Pattern Recognition, CVPR 2021, virtual, June 19-25, 2021. Computer Vision Foundation / IEEE, 2021, str. 15 750–15 758.

[111]Tian, Y., Chen, X., Ganguli, S., "Understanding self-supervised learning dynamics without contrastive pairs", in Proceedings of the 38th International Conference on Machine Learning, ICML 2021, 18-24 July 2021, Virtual Event, ser. Proceedings of Machine Learning Research, Meila, M., Zhang, T., (ur.), Vol. 139. PMLR, 2021, str. 10 268–10 278.

[112]Kurakin, A., Li, C.-L., Raffel, C., Berthelot, D., Cubuk, E. D., Zhang, H., Sohn, K., Carlini, N., Zhang, Z., "Fixmatch: Simplifying semi-supervised learning with consistency and confidence", in NeurIPS, 2020.

[113]Qi, M., Wang, Y., Qin, J., Li, A., "Ke-gan: Knowledge embedded generative adversarial networks for semi-supervised scene parsing", in Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition, 2019, str. 5237–5246.

[114]Zhu, Y., Zhang, Z., Wu, C., Zhang, Z., He, T., Zhang, H., Manmatha, R., Li, M., Smola, A. J., "Improving semantic segmentation via self-training", CoRR, Vol. abs/2004.14960, 2020.

[115]Chen, L., Lopes, R. G., Cheng, B., Collins, M. D., Cubuk, E. D., Zoph, B., Adam, H., Shlens, J., "Naive-student: Leveraging semi-supervised learning in video sequences for urban scene segmentation", in Computer Vision - ECCV 2020 - 16th European Conference, Glasgow, UK, August 23-28, 2020, Proceedings, Part IX, ser. Lecture Notes in Computer Science, Vedaldi, A., Bischof, H., Brox, T., Frahm, J., (ur.), Vol. 12354. Springer, 2020, str. 695–714.

[116]Mendel, R., Souza Jr, L., Rauber, D., Papa, J., Palm, C., "Semi-supervised segmentation based on error-correcting supervision", in ECCV, 2020.

[117]Ouali, Y., Hudelot, C., Tami, M., "Semi-supervised semantic segmentation with cross-consistency training", in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition, CVPR 2020, Seattle, WA, USA, June 13-19, 2020. IEEE, 2020.

[118]Lai, X., Tian, Z., Jiang, L., Liu, S., Zhao, H., Wang, L., Jia, J., "Semi-supervised semantic segmentation with directional context-aware consistency", in CVPR, 2021.

[119]van den Oord, A., Li, Y., Vinyals, O., "Representation learning with contrastive predictive coding", CoRR, Vol. abs/1807.03748, 2018, dostupno na: http://arxiv.org/abs/1807.03748

[120]He, K., Fan, H., Wu, Y., Xie, S., Girshick, R., "Momentum contrast for unsupervised visual representation learning", in 2020 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), 2020, str. 9726-9735.

[121]Yang, L., Zhuo, W., Qi, L., Shi, Y., Gao, Y., "St++: Make self-training work better for semi-supervised semantic segmentation", in CVPR, 2022.

[122]Duchon, J., "Splines minimizing rotation-invariant semi-norms in sobolev spaces", in Constructive theory of functions of several variables. Springer, 1977, str. 85–100.

[123]Bookstein, F. L., "Principal warps: Thin-plate splines and the decomposition of deformations.", IEEE Trans. Pattern Anal. Mach. Intell., Vol. 11, No. 6, 1989, str. 567-585.

[124]Keller, W., Borkowski, A., "Thin plate spline interpolation", Journal of Geodesy, Vol. 93, 02 2019.

[125]Donato, G., Belongie, S. J., "Approximate thin plate spline mappings", in Computer Vision - ECCV 2002, 7th European Conference on Computer Vision, Copenhagen, Denmark, May 28-31, 2002, Proceedings, Part III, ser. Lecture Notes in Computer Science, Heyden, A., Sparr, G., Nielsen, M., Johansen, P., (ur.), Vol. 2352. Springer, 2002, str. 21–31.

[126]Szeliski, R., Computer vision: algorithms and applications. Springer Science & Business Media, 2010.

[127]Everingham, M., Van Gool, L., Williams, C. K. I., Winn, J., Zisserman, A., "The PASCAL Visual Object Classes Challenge 2012 (VOC2012) Results", http://www.pascal-network.org/challenges/VOC/voc2012/workshop/index.html.

[128]Hariharan, B., Arbeláez, P., Bourdev, L., Maji, S., Malik, J., "Semantic contours from inverse detectors", in 2011 International Conference on Computer Vision, 2011, str. 991-998.

[129]Loshchilov, I., Hutter, F., "SGDR: stochastic gradient descent with warm restarts", in 5th International Conference on Learning Representations, ICLR 2017, Toulon, France, April 24-26, 2017, Conference Track Proceedings. OpenReview.net, 2017.

[130]Zagoruyko, S., Komodakis, N., "Wide residual networks", in Proceedings of the British Machine Vision Conference (BMVC), Richard C. Wilson, E. R. H., Smith, W. A. P., (ur.). BMVA Press, September 2016, str. 87.1-87.12, dostupno na: https://dx.doi.org/10.5244/C.30.87

[131]Niklaus, S., Liu, F., "Softmax splatting for video frame interpolation", in IEEE Conference on Computer Vision and Pattern Recognition, 2020.

[132]Brostow, G. J., Fauqueur, J., Cipolla, R., "Semantic object classes in video: A high-definition ground truth database", Pattern Recognition Letters, Vol. 30, No. 2, 2009, str. 88–97.

Dodatak A

Dodatne usporedbe algoritama

A.1 Hiperparametri

Tablice A.1 i A.2 uspoređuju hiperparametre konzistencijskih polunadziranih algoritama iz tablica 5.1 i 5.3.

Tablica A.1: Pregled hiperparametara konzistencijskih algoritama polunadziranog učenja za semantičku segmentaciju. Postavke naših eksperimenata označene su s "naše", H je visina i širina slučajnog isječka, $\boldsymbol{y}$ je učiteljeva predikcija, $\tilde{\boldsymbol{y}}$ je učenikova predikcija, hard je funkcija koja vektor koji predstavlja razdiobu preslikava u najbliži jednojedinični vektor ($\text{hard}_{[c]} = [\![c = \arg\max_k \boldsymbol{y}_{[k]}]\!]$), e je udio završenih epoha, α je težina konzistencijskog gubitka, a η_0 je osnovni korak učenja. Uvjet iza "|" u stupcu "Konzistencijski gubitak" označava korištenje praga uvjerenosti, koji određuje hoće li se gubitak primijeniti na pikselu.

Model	**Metoda**	H	**Skaliranje**	**Broj iter.**	**Broj epoha**	B_l	B_u	**Konzistencijski gubitak**	α	**Korak optim.**
Cityscapes na pola rezolucije										
DLv2	MT-CutMix [28]	321	[0.5..1.5]	40000	135 *	10	10	$\|\text{hard}(\boldsymbol{y}) - \tilde{\boldsymbol{y}}\|_2^2 \,\|_{\max(\boldsymbol{y})>0.97}$	0.5	$\eta_0(1-e)^{0.9}$
DLv2	MT-CutMix$_{\sim[28]}$	321	[0.5..1.5]	37100	100	4	4	$\|\text{hard}(\boldsymbol{y}) - \tilde{\boldsymbol{y}}\|_2^2 \,\|_{\max(\boldsymbol{y})>0.97}$	1	$\eta_0(1-e)^{0.9}$
DLv2	naše	448	[0.5..2]	74200	200	4	4	$D(\boldsymbol{y}, \tilde{\boldsymbol{y}})$	0.5	$\eta_0(1-e)^{0.9}$
SN	naše	448	[0.5..2]	74200	200	8	8	$D(\boldsymbol{y}, \tilde{\boldsymbol{y}})$	0.5	$\eta_0 \cos(e\pi/2)$
Cityscapes na punoj rezoluciji										
SN	naše	768	[0.5..2]	92750	250	8	8	$D(\boldsymbol{y}, \tilde{\boldsymbol{y}})$	0.5	$\eta_0 \cos(e\pi/2)$
DLv3$^+$	CAC [118]	720	[0.5..1.5]	92560	249 *	8	8	složeniji, vidjeti u [118]	$[\![e > \frac{5}{80}]\!] \cdot 0.1$	$\eta_0(1-e)^{0.9}$
DLv3$^+$	naše	720	[0.5..1.5]	92560	249 *	8	8	$D(\boldsymbol{y}, \tilde{\boldsymbol{y}})$	0.5	$\eta_0(1-e)^{0.9}$
PASCAL VOC 2012										
DLv3$^+$	CAC [118]	320	[0.5..2]	105760	80	8	8	složeniji, vidjeti u [118]	$[\![e > \frac{5}{80}]\!] \cdot 0.1$	$\eta_0(1-e)^{0.9}$
DLv3$^+$	naše	320	[0.5..1.5]	105760	80	8	8	$D(\boldsymbol{y}, \tilde{\boldsymbol{y}})$	0.5	$\eta_0(1-e)^{0.9}$

* Neki radovi koriste riječ "epoha" sa značenjem fiksnog broja iteracija koji ne ovisi o skupovima za učenje – 1000 iteracija [28] i 1157 iteracija [118].

Tablica A.2: Hiperparametri optimizacijskog algoritma za semantički segmentaciju izraženi u sintaksi sličnoj PyTorchovoj. Postavke naših eksperimenata označene su s "naše".

Model	**Metoda**	**Postavke optimizatora**	
		Osnovno	**Okosnica (razlika)**
Cityscapes			
DLv2	MT-CutMix [28]	`SGD, lr=`$3 \cdot 10^{-5}$`, momentum=`0.9`, weight_decay=`$5 \cdot 10^{-4}$	
DLv2	MT-CutMix$_{\sim[28]}$	`SGD, lr=`$3 \cdot 10^{-5}$`, momentum=`0.9`, weight_decay=`$5 \cdot 10^{-4}$	
DLv2	naše	`Adam, betas=`$(0.9, 0.99)$`, lr=`$4 \cdot 10^{-4}$`, weight_decay=`$1 \cdot 10^{-4}$	`lr=`$1 \cdot 10^{-4}$`, weight_decay=`$2.5 \cdot 10^{-5}$
SN	naše	`Adam, betas=`$(0.9, 0.99)$`, lr=`$4 \cdot 10^{-4}$`, weight_decay=`$2.5 \cdot 10^{-5}$	`lr=`$1 \cdot 10^{-4}$`, weight_decay=`$6.25 \cdot 10^{-6}$
DLv3$^+$	CAC [118]	`SGD, lr=`$1 \cdot 10^{-1}$`, momentum=`0.9	`lr=`$1 \cdot 10^{-2}$`, weight_decay=`$1 \cdot 10^{-4}$
DLv3$^+$	naše	`SGD, lr=`$1 \cdot 10^{-1}$`, momentum=`0.9	`lr=`$1 \cdot 10^{-2}$`, weight_decay=`$1 \cdot 10^{-4}$
PASCAL VOC 2012			
DLv3$^+$	CAC [118]	`SGD, lr=`$1 \cdot 10^{-2}$`, momentum=`0.9	`lr=`$1 \cdot 10^{-3}$`, weight_decay=`$1 \cdot 10^{-4}$
DLv3$^+$	naše	`SGD, lr=`$1 \cdot 10^{-2}$`, momentum=`0.9	`lr=`$1 \cdot 10^{-3}$`, weight_decay=`$1 \cdot 10^{-4}$

A.2 Vremenske i memorijske karakteristike

Tablica A.3 prikazuje memorijske zahtjeve i trajanje učenja algoritama iz tablica 5.1 i 5.3. Vremena uključuju učitavanje i obradu podataka, a ne uključuju vrednovanje na validacijskom skupu. Za mjerenje zauzetosti su korištene procedure `max_memory_allocated`* i `reset_peak_memory_stats` iz paketa `torch.cuda`. Neki algoritmi nisu stali u memoriju grafičke kartice GTX 2080Ti. Mjerenja memorije su veća ovdje nego u slici 4.3 jer se ovdje u memoriji još (nepotrebno) drže nadzirane predikcije i perturbirani ulazi i izlazi.

Za DeepLabv3$^+$-RN50 koristimo broj iteracija, veličinu mini-grupe i veličinu isječka iz [118]. Metoda iz [118] ima memorijske zahtjeve dvosmjerne konzistencije zbog smjera konzistencije koji ovisi o pojedinom pikselu.

Tablica A.3: Maksimalna zauzetost memorije i trajanje učenja na Cityscapesu na pola rezolucije (gornji odjeljak) i Cityscapesu (donji odjeljak) na grafičkom karticama NVidia A4500 i NVidia 2080Ti.

Model	**Metoda**	**Velič. isječ.**	**Broj iter.**	B_l	B_u	**Mem.** /MiB	**Trajanje**/min **A4500**	**2080Ti**
DLv2-RN101	MT-CutMix [28]	321	40000	10	10	16289	1067	–
	MT-CutMix$_{\sim[28]}$	321	37100	4	4	7037	794	1314
DLv2-RN101	supervised	448	74300	4	–	6611	338	602
	MT-PhTPS	448	74300	4	4	7021	816	1397
SN-RN18	supervised	448	74200	8	–	1646	119	161
	simple-PhTPS	448	74200	8	8	2398	228	279
	MT-PhTPS	448	74200	8	8	2456	234	297
SN-RN18	supervised	768	92750	8	–	4444	321	432
	simple-PhTPS	768	92750	8	8	6683	732	963
	MT-PhTPS	768	92750	8	8	6727	768	965
SN-RN34	supervised	768	92750	8	–	5500	422	570
	simple-PhTPS	768	92750	8	8	7737	994	1268
	MT-PhTPS	768	92750	8	8	7818	1013	1276
DLv3+-RN50	supervised	720	92560	8	–	11645	1229	–
	simple-PhTPS	720	92560	8	8	13384	1884	–
	CAC [118]	720	92560	8	8	25165†	>3000*	–

† Izvorna implementacija zahtijeva 36005 MiB. Otprilike 10.6 GiB može se uštediti čuvanjem akumuliranjem gradijenta kao u algoritmu 4.1.

* Procijenjeno pokretanjem na grafičkoj kartici NVidia A100.

Tablica A.4 prikazuje brojeve parametara modela, a tablica A.5 prikazuje brzine zaključivanja modela iz tablica 5.1 i 5.3.

*Dio troška memorije koji koristi PyTorch nije obuhvaćen mjerenjem. Više informacija daje dokumentacija o upravljanju memorijom PyTorcha: `https://pytorch.org/docs/master/notes/cuda.html`.

Tablica A.4: Brojevi parametara modela.

Model	**Broj parametara**
DeepLabv2-RN101	$43.80 \cdot 10^6$
DeepLabv3$^+$-RN50	$40.35 \cdot 10^6$
SwiftNet-RN34	$21.91 \cdot 10^6$
SwiftNet-RN18	$11.80 \cdot 10^6$

Tablica A.5: Brzine zaključivanja modela (broj slika po sekundi) na 3 različite grafičke kartice i 2 rezolucije ulaza. Ulazi se obrađuju jedan-po-jedan, bez preklapanja u računanju. Mjerenja uključuju računanje gubitka unakrsne entropije s obzirom na oznake i ne uključuju učitavanje i obradu podataka.

	1024×2048			512×1024		
	A4500	**2080Ti**	**1080Ti**	**A4500**	**2080Ti**	**1080Ti**
SwiftNet-RN18	56.5	45.3	34.6	139.5	115.8	98.4
SwiftNet-RN34	39.2	30.5	23.6	93.4	86.1	73.5
DeepLabv3+-RN50	16.1	9.6	5.2	54.2	30.7	23.5
DeepLabv2-RN101	5.1	3.0	1.5	19.6	12.2	6.3

Oznake

Objekti

Varijable su označene kosim slovima, a konstante uspravnim slovima. Vektori i drugi nizovi su označeni podebljanim slovima, a skupovi slovima s udvostručenim linijama ili velikim grčkim slovima. Slučajne varijable su podvučene.

x, X, θ	varijabla (općenito, često skalar)
$\boldsymbol{x}, \boldsymbol{\theta}$	vektor, niz ili višedimenzionalni niz (često vektor stupac)
$\boldsymbol{X}, \boldsymbol{\Theta}$	matrica ili višedimenzionalni niz
$\mathbb{X}, \Omega$	skup
$\underline{x}, \underline{\boldsymbol{x}}, \underline{\boldsymbol{X}}, \underline{\mathbb{X}}$	slučajna varijabla

Konstante

$\{\}$	prazni skup
ekonstanta za koju vrijedi	$\frac{\mathrm{d}}{\mathrm{d}x}\mathrm{e}^x = \mathrm{e}^x$
π	omjer duljine i promjera kružnice
$\mathbf{0}, \mathbf{1}$	vektor kojem su sve komponente 0 odnosno 1
$\mathbf{e}_i$	i-ti vektor kanonske baze, jednojedinični vektor
$\mathbf{I}, \mathbf{I}_n$	matrica identiteta (s n redaka i stupaca)
$\mathbb{N}, \mathbb{Z}, \mathbb{R}, \mathbb{C}$	poznati skup

Definiranje skupova i nizova

$\{a, .., b\}, \{a..b\}$	diskretni skup s elementima od a do b
$\{f(a) \colon P(a)\}, \{f(a)\}_{P(a)}$	skup definiran preko funkcije f i predikata P
$\{a_1, .., a_n\}, \{a_i\}_{i=1..n}$	skup s n elemenata
$(x_1, .., x_n)$	n-torka
$[x_1, .., x_n]$	retčani vektor dimenzije n
$[a..b), (a..b]$	poluzatvoreni realni interval

Donji i gornji indeks

U indeksima mogu biti oznake indeksiranja, izrezivanja ili druge oznake. Uglata zagrada u donjem indeksu označava indeksiranje ili izrezivanje po uzoru na Python.

$a^{\text{gore}}_{\text{dolje}}$	varijabla s oznakama u donjem i gornjem indeksu
$\boldsymbol{a}_{[i]}$	i-ti element (komponenta) vektora $\boldsymbol{a}$
$\boldsymbol{a}_{[i_1:i_2]}$	vektor koji čine elementi $\boldsymbol{a}_{[i_1]}, \boldsymbol{a}_{[i_1+1]}, .., \boldsymbol{a}_{[i_2]}$
$\boldsymbol{a}_{[(i_1\,..\,i_n)]}$	vektor koji čine elementi $\boldsymbol{a}_{[i_1]}, \boldsymbol{a}_{[i_2]}, .., \boldsymbol{a}_{[i_n]}$
$\boldsymbol{A}_{[i,j]}$	element i, j matrice $\boldsymbol{A}$
$\boldsymbol{A}_{[i,:]}$	i-ti redak matrice $\boldsymbol{A}$
$\boldsymbol{A}_{[:,i_1:i_2,j]}$	2D odsječak 3D niza $\boldsymbol{A}$

Operacije linearne algebre i operacije s nizovima

$\langle \boldsymbol{a} \vert \boldsymbol{b} \rangle, \boldsymbol{a}^\mathsf{T}\boldsymbol{b}$	skalarni produkt
$\boldsymbol{a} \odot \boldsymbol{b}$	umnožak po elementima; Hadamardov produkt
$\boldsymbol{a} \oslash \boldsymbol{b}$	dijeljenje po elementima
$\boldsymbol{a}^{\odot b}$	potenciranje po elementima
$\boldsymbol{AB}$, $\boldsymbol{Ab}$	matrično množenje
$\boldsymbol{A}^{-1}$	inverz matrice
$\boldsymbol{A}^\mathsf{T}$	transponiranje
$\operatorname{diag}(\boldsymbol{a})$	dijagonalna matrica kojoj dijagonalu čini vektor $\boldsymbol{a}$
$\det(\boldsymbol{A})$	determinanta matrice $\boldsymbol{A}$
$\left\|\boldsymbol{a}\right\|_p$	L^p-norma vektora $\boldsymbol{a}$
$\left\|\boldsymbol{A}\right\|_\mathrm{F}$	Frobeniusova norma matrice $\boldsymbol{A}$
$\operatorname{vec}(\boldsymbol{A})$	funkcija koja preslikava višedimenzionalni niz iz $\mathbb{R}^{d_1 \times \cdots \times d_n}$ u vektor iz $\mathbb{R}^{d_1 \cdots d_n}$

Diferencijalni račun

$\frac{\mathrm{d}y}{\mathrm{d}x}$, $\frac{\mathrm{d}}{\mathrm{d}x} f(x)$	derivacija funkcije $f = x \mapsto y$ po x (jakobijan u slučaju višedimenzionalne domene ili kodomene)
$\frac{\partial \boldsymbol{y}}{\partial \boldsymbol{x}}$, $\frac{\partial}{\partial \boldsymbol{x}} f(\boldsymbol{x}, \boldsymbol{\xi})$	parcijalna derivacija ili jakobijan funkcije f po ulazu $\boldsymbol{x}$
$\nabla_{\boldsymbol{x}} y, \nabla_{\boldsymbol{x}} f(\boldsymbol{x})$	gradijent funkcije $y = f(\boldsymbol{x})$ po $\boldsymbol{x}$
$\int_{x \in A} f(x)\,\mathrm{d}x$	određeni integral funkcije $f(x)$ po $x \in A$

Teorija vjerojatnosti

$\underline{a} \perp \underline{b}$	*slučajne varijable $\underline{a}$ i $\underline{b}$ su nezavisne*
$\mathrm{P}[\underline{a}], P_{\underline{a}}$	funkcija vjerojatnosti slučajne varijable $\underline{a}$
$\mathrm{p}[\underline{a}], p_{\underline{a}}$	funkcija gustoće vjerojatnosti slučajne varijable $\underline{a}$

$\mathrm{P}(\underline{a} = a), P_{\underline{a}}(a), \mathrm{P}(a)$vjerojatnost doga đaja $\underline{a} = a$
$\mathrm{p}(\underline{a} = a), p_{\underline{a}}(a), \mathrm{p}(a)$gustoća vjerojatnosti doga đaja $\underline{a} = a$
$p_{\underline{a},\underline{b}}(a,b), \mathrm{p}(a,b)$gustoća vjerojatnosti doga đaja $\underline{a} = a, \underline{b} = b$
$p_{\underline{a}|b}(a), \mathrm{p}(a \mid b)$uvjetna gustoća vjerojatnosti doga đaja $\underline{a} = a \mid \underline{b} = b$
$\mathbf{E}_{a\sim\mathrm{p}[\underline{a}]}\, f(a), \mathbf{E}_{\underline{a}}\, f(a)$očekivanje funkcije slučajne varijable $\underline{a}$
$\mathbf{E}_{a\in\mathcal{A}}\, f(a)$aritmetička sredina funkcije po skupu $\mathcal{A}$
$\mathrm{N}(\mu, \sigma^2)$normalna razdioba s očekivanjem μ i varijancom σ^2
$\mathrm{U}(\mathcal{A})$uniformna razdioba nad skupom $\mathcal{A}$

Teorija informacije

$\mathrm{I}(\mathcal{A})$informacijski sadržaj doga đaja $\mathcal{A}$
$\mathrm{H}[\underline{a}], \mathrm{H}(p)$entropija
$\mathrm{H}(p \parallel q)$unakrsna entropija q s obzirom na p
$\mathrm{D}(p \parallel q)$relativna entropija (Kullback-Leiblerova divergencija) q s obzirom na p
$\mathrm{I}[\underline{a}; \underline{b}]$uzajamna informacija
$\mathrm{H}[\underline{a} \mid \underline{b}]$uvjetna entropija

Ostale matematičke oznake

$f\colon \mathcal{A} \to \mathcal{B}$ funkcija s domenom $\mathcal{A}$ i kodomenom $\mathcal{B}$
$x \mapsto g(x)$funkcija koja preslikava x u $g(x)$
$|\mathcal{A}|$ kardinalitet skupa
$\mathcal{A} \perp \mathcal{B}$ *skupovi $\mathcal{A}$ i $\mathcal{B}$ su disjunktni*
$[\![P]\!]$ Iversonova uglata zagrada za tvrdnju P: $[\![P]\!] \coloneqq \begin{cases} 1, & P \equiv \top \\ 0, & P \equiv \bot \end{cases}$
$\mathrm{pa}_G(a)$skup čvorova roditelja čvora a u grafu G
$\mathrm{ch}_G(a)$skup čvorova djece čvora a u grafu G
$\mathrm{sg}(\boldsymbol{\theta})$smrznuta kopija varijable $\boldsymbol{\theta}$ (rezultat zaustavljanja gradijenta)

Izrazi

dimenzija vektorabroj komponenata (elemenata) vektora (kardinalitet baze vektorskog prostora)
n-dimenzionalni nizelement skupa $\mathbb{R}^{d_1\times\cdots\times d_n}$, gdje su $d_1, .., d_n$ dimenzije
i-ta dimenzija nizabroj mogućih vrijednosti i-tog indeksa niza
oblik niza n-torka koja redom sadrži sve dimenzije niza; $(d_1, .., d_n)$ za niz iz $\mathbb{R}^{d_1\times\cdots\times d_n}$
red nizabroj dimenzija niza; n za niz iz $\mathbb{R}^{d_1\times\cdots\times d_n}$

afini slojsloj modela	koji se može svesti na parametriziranu afinu transformaciju matričnog množenja s pribrajanjem vektora
nD	n-dimenzionalan (za broj n)
pbpostotni bod	
atraktorskup stanja	$\mathbb{A}$ za koji postoji drugi skup početnih stanja $\mathbb{B}$ takav da $\mathbb{B} \setminus \mathbb{A} \neq \{\}$ i broj iteracija nakon kojeg stanje dinamički sustav uvijek bude iz $\mathbb{A}$

Kazalo

Popis slika

Popis tablica

Životopis

Ivan Grubišić rođen je 1994. godine u Virovitici. Diplomirao je 2018. na Fakultetu elektrotehnike i računarstva Sveučilišta u Zagrebu. Po završetku diplomskog studija 2018. godine zaposlio se kao mlađi istraživač na projektu DATACROSS na Fakultetu elektrotehnike i računarstva u skupini profesora Siniše Šegvića. Od 2021. radi kao asistent na Fakultetu elektrotehnike i računarstva.

Sudjelovao je na istraživačkim projektima DATACROSS i ADEPT. Bio je u skupini koja je pobijedila na natjecanju Robust Vision Challenge 2020. godine. Autor je radova predstavljenih na međunarodnim konferencijama i časopisima i recenzent je na međunarodnim konferencijama.

Njegov primarni interes uključuje strojno učenje i računalni vidi, uz značajnije iskustvo u gustoj predikciji, polunadziranom učenju, robusnom učenju, normalizirajućim tokovima i razvoju programske potpore.

Popis objavljenih djela

Radovi u časopisima

1. **Grubišić, I., Oršić, M., Šegvić, S., "Revisiting consistency for semi-supervised semantic segmentation", Sensors, Vol. 23, No. 2, 2023**

Radovi na konferencijama

1.Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S., "Multi-domain semantic segmentation with overlapping labels", in IEEE/CVF Winter Conference on Applications of Computer Vision, 2022, pp. 2615-2624

2. **Grubišić, I., Oršić, M., Šegvić, S., "A baseline for semi-supervised learning of efficient semantic segmentation models", in 17th International Conference on Machine Vision and Applications, MVA 2021, Aichi, Japan, July 25-27, 2021. IEEE, 2021, str. 1–5**

3.Grcić M., Grubišić, I., Šegvić, S., "Densely connected normalizing flows", in Advances in Neural Information Processing Systems, Ranzato, M., Beygelzimer, A., Dauphin, Y., Liang, P., Vaughan, J. W., (ur.), Vol. 34. Curran Associates, Inc., 2021, str. 23 968–23 982

Ostali radovi

1.Bevandić, P., Oršić, M., Grubišić, I., Šarić, J., Šegvić, S., "Weakly supervised training of universal visual concepts for multi-domain semantic segmentation", arXiv preprint arXiv:2212.10340, 2022

Biography

Ivan Grubišić was born in 1994 in Virovitica, Croatia. He graduated in 2018 from the Faculty of Electrical Engineering and Computing, University of Zagreb. After completing his graduate studies in 2018, he was employed as a junior researcher on the DATACROSS project at the Faculty of Electrical Engineering and Computing in the group of professor Siniša Šegvić. Since 2021, he has been working as a teaching assistant at the Faculty.

He participated in the DATACROSS and ADEPT research projects. He was part of the group that won the Robust Vision Challenge 2020 competition. He is an author of papers presented at international conferences and journals and is a reviewer at international conferences.

His primary research interests include machine learning and computer vision, focusing on dense prediction, semi-supervised learning, robust learning, normalizing flows, and software development.