

Duboki generativni modeli u vrednovanju financijske imovine

Šutalo, Karlo

Master's thesis / Diplomski rad

2024

Degree Grantor / Ustanova koja je dodijelila akademski / stručni stupanj: **University of Zagreb, Faculty of Electrical Engineering and Computing / Sveučilište u Zagrebu, Fakultet elektrotehnike i računarstva**

Permanent link / Trajna poveznica: <https://urn.nsk.hr/urn:nbn:hr:168:422259>

Rights / Prava: [In copyright](#)/[Zaštićeno autorskim pravom.](#)

Download date / Datum preuzimanja: **2025-04-01**



Repository / Repozitorij:

[FER Repository - University of Zagreb Faculty of Electrical Engineering and Computing repository](#)



SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 532

**DUBOKI GENERATIVNI MODELI U VREDNOVANJU
FINANCIJSKE IMOVINE**

Karlo Šutalo

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

DIPLOMSKI RAD br. 532

**DUBOKI GENERATIVNI MODELI U VREDNOVANJU
FINANCIJSKE IMOVINE**

Karlo Šutalo

Zagreb, lipanj 2024.

DIPLOMSKI ZADATAK br. 532

Pristupnik: **Karlo Šutalo (0036522932)**
Studij: Računarstvo
Profil: Znanost o podacima
Mentor: prof. dr. sc. Zvonko Kostanjčar

Zadatak: **Duboki generativni modeli u vrednovanju financijske imovine**

Opis zadatka:

U okviru ovog rada proučava se primjena dubokog učenja za vrednovanje financijske imovine, s naglaskom na vjerojatnosne pristupe. Vrednovanje financijske imovine se često svodi na predviđanje očekivanih povrata i predviđanja kovarijacijske matrice odgovarajućih povrata. S obzirom na to da je financijsko tržište vremenski promjenjiv kompleksan sustav, vektor očekivanih povrata i kovarijacijska matrica povrata mijenjaju se u vremenu, što implicira da klasični pristupi zasnovani na stacionarnosti ili ergodičnosti ne funkcioniraju dovoljno dobro. U radu je potrebno razviti model koji predviđa vektor očekivanja i kovarijacijsku matricu distribucije budućih povrata temeljem povijesnih realizacija, uz mogućnost kontroliranja ranga kovarijacijske matrice regularizacijskim parametrom. Model je potrebno testirati nad povijesnim podacima vodećih tržišnih indeksa nekoliko imovinskih klasa te usporediti rezultate sa standardnim modelima u području.

Rok za predaju rada: 28. lipnja 2024.

Sadržaj

| | |
|---|-----------|
| 1. Uvod | 3 |
| 2. Teorijski okvir | 5 |
| 2.1. Financijski vremenski nizovi | 5 |
| 2.1.1. Povrat na investiciju | 5 |
| 2.1.2. Pretpostavke na povrate | 6 |
| 2.2. Analiza linearnih vremenskih nizova | 8 |
| 2.2.1. Stacionarnost | 9 |
| 2.2.2. Korelacija i autokorelacijska funkcija | 10 |
| 2.3. Autoregresivni modeli | 10 |
| 2.3.1. Uvjet slabe stacionarnosti | 12 |
| 2.4. Modeliranje volatilnosti | 13 |
| 2.4.1. Karakteristike volatilnosti | 13 |
| 2.4.2. ARCH model | 15 |
| 2.5. Neuronske mreže | 17 |
| 2.5.1. Učenje temeljeno na gradijentu | 18 |
| 2.5.2. Aktivacijske funkcije | 21 |
| 2.5.3. Arhitektura mreže | 24 |
| 2.5.4. Regularizacija | 27 |
| 3. Generiranje sintetičkih podataka | 28 |
| 3.1. Stilizirane činjenice o povratima | 28 |
| 3.2. Generiranje podataka | 29 |
| 3.3. Karakteristike generiranih podataka | 31 |
| 4. Modeliranje povrata | 35 |

| | |
|--|-----------|
| 4.1. Generativno modeliranje | 35 |
| 4.2. Priprema podataka za treniranje | 36 |
| 4.3. Modeliranje povrata korištenjem neuronske mreže | 37 |
| 4.4. Postupak treniranja | 38 |
| 4.5. Evaluacija performansi | 40 |
| 5. Zaključak | 44 |
| 5.1. Sumiranje rezultata | 44 |
| 5.2. Preporuke za buduća istraživanja | 44 |
| Literatura | 46 |
| Sažetak | 48 |
| Abstract | 49 |

1. Uvod

Analiza financijskih vremenskih nizova, kao što su cijene dionica, tečajevi valuta i kamatne stope, ključna je za razumijevanje i predviđanje kretanja financijskih tržišta. Pouzdane metode modeliranja ovih nizova igraju značajnu ulogu u različitim financijskim primjenama, uključujući predviđanje cijena, upravljanje rizicima i donošenje investicijskih odluka [1]. Iz tog razloga, posljednjih desetljeća, razvijeni su mnogi modeli kojima se pokušava analizirati i predvidjeti financijske vremenske nizova. Primjeri takvih modela, koji se koriste u ovom radu, su autoregresivni modeli (AR), modeli autoregresivne uvjetne heteroskedastičnosti (ARCH) i neuronske mreže [2] [3] [4] [5].

Autoregresivni modeli (AR) koriste povijesne podatke za predviđanje budućih vrijednosti pretpostavljajući linearnu zavisnost podataka u određenom razdoblju čime se mogu modelirati povrati imovina [3]. GARCH modeli, kao poopćenje AR modela, pogodni su za modeliranje volatilnosti što je ključna karakteristika financijskih vremenskih nizova [3]. Razvijanjem dubokog učenja, modeliranje povrata i volatilnosti može se raditi i neuronskim mrežama koje na temelju povijesnih podataka mogu naučiti kompleksne nelinearne zavisnosti [5].

Cilj ovog rada je analizirati kako količina podataka i metode generiranja sintetičkih financijskih vremenskih nizova utječu na performanse neuronskih mreža u modeliranju istih. Generiranje sintetičkih podataka daje nam mogućnost kreiranja eksperimenta u kontroliranim uvjetima što sa stvarnim financijskim podacima je teško moguće zbog velike količine šuma u podacima.

U drugom poglavlju rada razmotrit će se teoretski okvir financijskih vremenskih nizova i modela koji će se koristiti u ovom radu. Treće poglavlje bavit će se metodologijom generiranja sintetičkih podataka korištenjem navedenih modela, dok će četvrto poglav-

lje biti posvećeno analizi utjecaja količine podataka i metoda generiranja podataka na performanse neuronskih mreža. U petom poglavlju bit će sažeti glavni rezultati i dane preporuke za buduća istraživanja.

2. Teorijski okvir

2.1. Financijski vremenski nizovi

Prilikom analiziranja financijskih vremenskih nizova većinom se koriste povrati umjesto cijena imovina. Prema Campbell, Lo, and MacKinlay [1] postoje dva glavna razloga za to. Prvi razlog je što, za prosječnog investitora, povrat imovine daje kompletnu informaciju o investiciji bez obzira na skalu cijene. Drugi razlog je taj što povrati, kao vremenski niz, imaju bolja statistička svojstva za analizu.

2.1.1. Povrat na investiciju

Za početak, potrebno je uvesti pojmove aritmetičkog povrata R_t i logaritamskog povrata r_t u trenutku t . Neka je P_t vrijednost imovine u trenutku t .

Aritmetički povrat od trenutka $t - 1$ do trenutka t definira se kao

$$R_t = \frac{P_t - P_{t-1}}{P_{t-1}} = \frac{P_t}{P_{t-1}} - 1. \quad (2.1)$$

Logaritamski povrat od trenutka t do trenutka $t - 1$ definira se kao

$$r_t = \log \frac{P_t}{P_{t-1}}. \quad (2.2)$$

Mjera povrata nam daje informaciju o tome kako se mijenjala vrijednost financijske imovine bez obzira na njezinu cijenu. No kako povrati nisu statični, potrebno je definirati mjeru povrata na duljem periodu. U slučaju da posjedujemo imovinu na k perioda

između vremena $t - k$ i t definiramo k -periodni aritmetički povrat $R_t[k]$ kao:

$$\begin{aligned}
 1 + R_t[k] &= \frac{P_t}{P_{t-k}} = \frac{P_t}{P_{t-1}} \times \frac{P_{t-1}}{P_{t-2}} \times \dots \times \frac{P_{t-k+1}}{P_{t-k}} \\
 &= (1 + R_t)(1 + R_{t-1}) \dots (1 + R_{t-k+1}) \\
 &= \prod_{j=0}^{k-1} (1 + R_{t-j})
 \end{aligned} \tag{2.3}$$

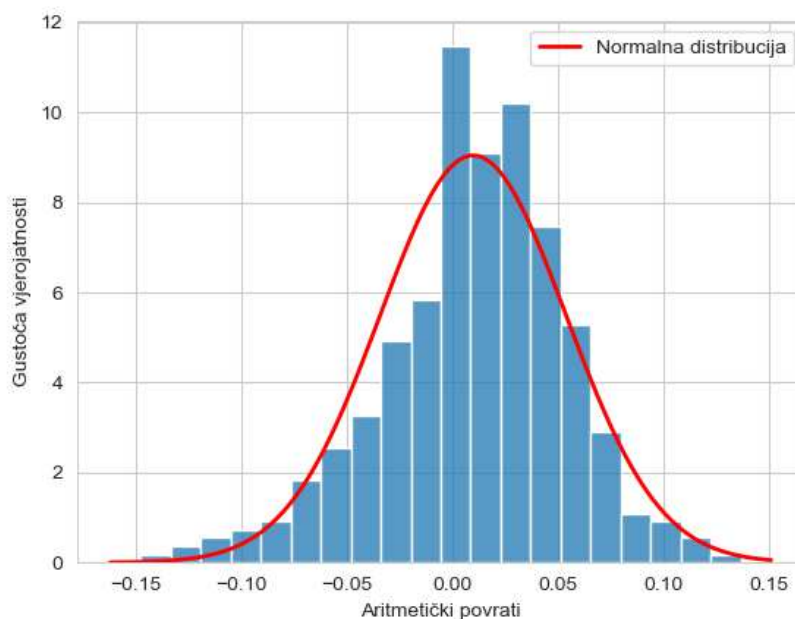
Također, na sličan način može se definirati i k -periodni logaritamski povrat $r_t[k]$ kao:

$$\begin{aligned}
 r_t[k] &= \log(1 + R_t[k]) = \log[(1 + R_t)(1 + R_{t-1}) \dots (1 + R_{t-k+1})] \\
 &= \log(1 + R_t) + \log(1 + R_{t-1}) + \dots + \log(1 + R_{t-k+1}) \\
 &= \sum_{j=0}^{k-1} r_{t-j}
 \end{aligned} \tag{2.4}$$

Iz jednadžbi 2.3 i 2.4 vidljivo je da se logaritamski povrati samo zbrajaju u slučaju dužeg perioda što nije slučaj za aritmetičke povrate. Uz to, logaritamski povrati imaju bolja statistička svojstva.

2.1.2. Pretpostavke na povrate

Često korištena pretpostavka prilikom proučavanja financijskih vremenskih nizova je da su aritmetički povrati $R_{it|t=1,\dots,T}$ nezavisni i jednako distribuirani po normalnoj distribuciji s fiksnom srednjom vrijednosti i varijancom [1]. Takva pretpostavka omogućuje da statistička svojstva povrata budu traktabilna. Međutim, ta pretpostavka nailazi na nekoliko poteškoća. Prvi problem je taj što je donja granica povrata -1 dok normalna distribucija ima domenu na cijelom skupu realnih brojeva te samim time nema donju granicu. Drugo, ako su R_{it} normalno distribuirani, tada višeperiodni aritmetički povrat $R_t[k]$ nije normalno distribuiran jer produkt normalnih distribucija nije normalna distribucija. Treće, pretpostavka normalnosti nije podržana empirijskim rezultatima koji pokazuju da stvarni povrati imaju teže repove [6].



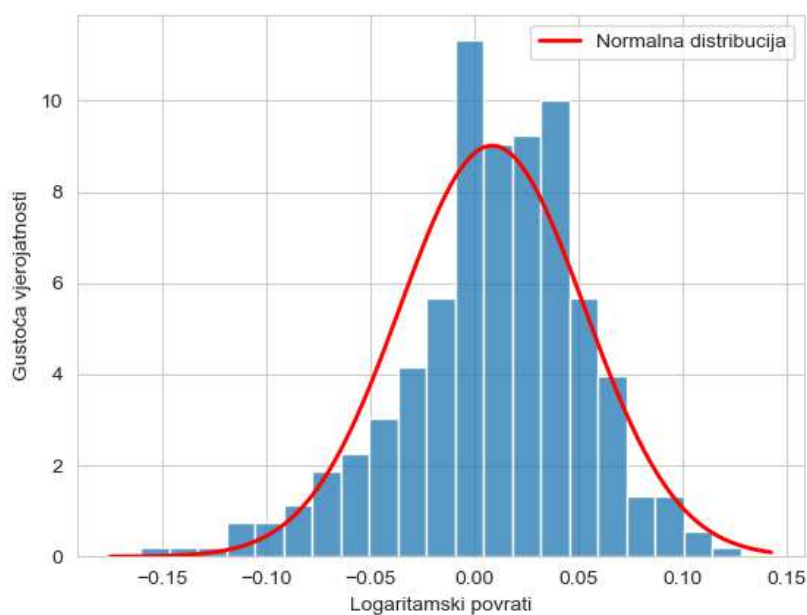
Slika 2.1. Histogram aritmetičkih povrata i funkcija gustoće normalne razdiobe
Podatci: S&P 500 index

Još jedna često korištena pretpostavka je da su logaritamski povrati nezavisni i jednako distribuirani po normalnoj razdiobi s fiksnom srednjom vrijednosti i varijancom [1]. Aritmetički povrati su onda nezavisni i jednako distribuirani po log-normalnoj razdiobi sa srednjom vrijednosti i varijancom danima kao:

$$E(R_t) = \exp\left(\mu + \frac{\sigma^2}{2}\right) - 1 \quad (2.5)$$

$$VAR(R_t) = \exp(2\mu + \sigma^2) [\exp(\sigma^2) - 1] \quad (2.6)$$

Kako je suma konačno mnogo normalnih razdioba open normalna razdioba, $r_t[k]$ je također normalno distribuirana pod pretpostavkom normalnosti logaritamskih povrata r_t . Dodatno, r_t nema donju granicu te je donja granica R_t zadovoljena jednadžbom $1 + R_t = \exp(r_t)$. Međutim, pretpostavka log-normalnosti aritmetičkih povrata, odnosno normalnosti logaritamskih povrata, nije konzistentna sa svim svojstvima povijesnih povrata. To se može primijetiti na slici 2.2. gdje je vidljivo da stvarni povrati imaju viši mod distribucije, teže repove te blagu pozitivnu asimetriju.



Slika 2.2. Histogram logaritamskih povrata i funkcija gustoće normalne razdiobe
Podatci: S&P 500 index

2.2. Analiza linearnih vremenskih nizova

Tretiranja povrata imovine (npr. logaritamskih povrata r_t) kao skup slučajnih varijabli u vremenu, dobije se vremenski niz $\{r_t\}$. Analiza linearnih vremenskih nizova daje prirodan okvir za proučavanje dinamike takvih nizova [3]. Neki od modela osmišljenih u tu svrhu su:

- Autoregresivni modeli (AR, eng. *autoregressive models*)
- Modeli pokretnog prosjeka (MA, eng. *moving-average models*)
- Autoregresivni modeli pokretnog prosjeka (ARMA eng. *Autoregressive moving-average models*)
- Sezonalni modeli

U slučaju povrata imovine r_t , svi ti modeli pokušavaju objasniti linearnu povezanost između r_t i informacija dostupnih prije trenutka t . Te informacije mogu biti sadržane u prijašnjoj cijeni imovine, ali isto tako i u nekom slučajnom vektoru koji opisuje ekonomsko okruženje u kojem se imovina promatra. Korelacija ima ključnu ulogu u razumi-

jevanju ovih modela. Posebno se naglašava analiza linearnih vremenskih nizova kroz korelacije između ciljne varijable i njezinih prethodnih vrijednosti. Ove korelacije, poznate kao serijske korelacije ili autokorelacije, predstavljaju osnovni alat za proučavanje stacionarnih vremenskih nizova.

2.2.1. Stacionarnost

Stacionarnost je osnova proučavanja vremenskih nizova. Za vremenski niz $\{r_t\}$ kažemo da je strogo stacionaran ako je zajednička distribucija $(r_{t_1}, \dots, r_{t_k})$ identična distribuciji $(r_{t_1+t}, \dots, r_{t_k+t})$ za svaki t , gdje je k proizvoljan cijeli broj i $(r_{t_1}, \dots, r_{t_k})$ je skup od k pozitivnih cijelih brojeva. Drugim riječima, stroga stacionarnost podrazumijeva da zajednička distribucija $(r_{t_1}, \dots, r_{t_k})$ ne zavisi od vremena. Međutim, ovaj uvjet je vrlo rigorozan i teško ga je empirijski provjeriti. Stoga se uvodi pojam slabe stacionarnosti. Vremenski niz r_t je slabo stacionaran ako su srednja vrijednost r_t i kovarijanca između r_t i r_{t-l} vremenski nepromjenjive, gdje je l proizvoljan cijeli broj. Točnije rečeno, $\{r_t\}$ je slabo stacionaran ako vrijedi:

$$\begin{aligned} E(r_t) &= \mu \\ \text{Cov}(r_t, r_{t-l}) &= \gamma_l, \end{aligned} \tag{2.7}$$

gdje je μ konstanta, a γ_l ovisi samo o vremenskom pomaku l . Slaba stacionarnost implicira da bi vremenski diagram sadržavao točke koje fluktuiraju s konstantnom varijancom oko konstantne vrijednosti.

Slaba stacionarnost pretpostavlja da su prva dva momenta od r_t konačna. Pod tom pretpostavkom, ako je r_t strogo stacionaran i ima konačne prve dvije momente, onda je i slabo stacionaran. Obratno ne vrijedi u općem slučaju. Ipak, ako je r_t normalno distribuiran, tada su slaba i stroga stacionarnost ekvivalentne.

Kovarijanca $\gamma_l = \text{Cov}(r_t, r_{t-l})$ naziva se autokovarijanca odmaka l vremenskog niza r_t . Ona posjeduje dva bitna svojstva, a to su: (a) $\gamma_0 = \text{Var}(r_t)$ i (b) $\gamma_{-l} = \gamma_l$. Drugo svojstvo vrijedi zbog

$$\begin{aligned} \text{Cov}(r_t, r_{t-(-l)}) &= \text{Cov}(r_{t-(-l)}, r_t) \\ &= \text{Cov}(r_{t+l}, r_t) \\ &= \text{Cov}(r_{t_1}, r_{t_1-l}), \quad t_1 = t + l \end{aligned} \tag{2.8}$$

U financijskoj literaturi, uobičajena pretpostavka je da su nizovi povrata imovina slabo stacionarne. Takva pretpostavka može se provjeriti uz dovoljnu količinu podataka.

2.2.2. Korelacija i autokorelacijska funkcija

Koeficijent korelacije između dvije slučajne varijable X i Y definira se kao:

$$\rho_{x,y} = \frac{Cov(X, Y)}{\sqrt{Var(X)Var(Y)}} = \frac{E[(X - \mu_x)(Y - \mu_y)]}{\sqrt{E(X - \mu_x)^2 E(Y - \mu_y)^2}}, \quad (2.9)$$

gdje su μ_x i μ_y srednje vrijednosti varijabli X i Y te se pretpostavlja da varijance postoje. Taj koeficijent ukazuje na mjeru linearne zavisnosti između X i Y te se može pokazati da vrijedi $-1 \leq \rho_{x,y} \leq 1$ i $\rho_{x,y} = \rho_{y,x}$. Varijable su nekorelirane ako je $\rho_{x,y} = 0$. Dodatno, ako su varijable normalno distribuirane, nekoreliranost je ekvivalentna s nezavisnošću.

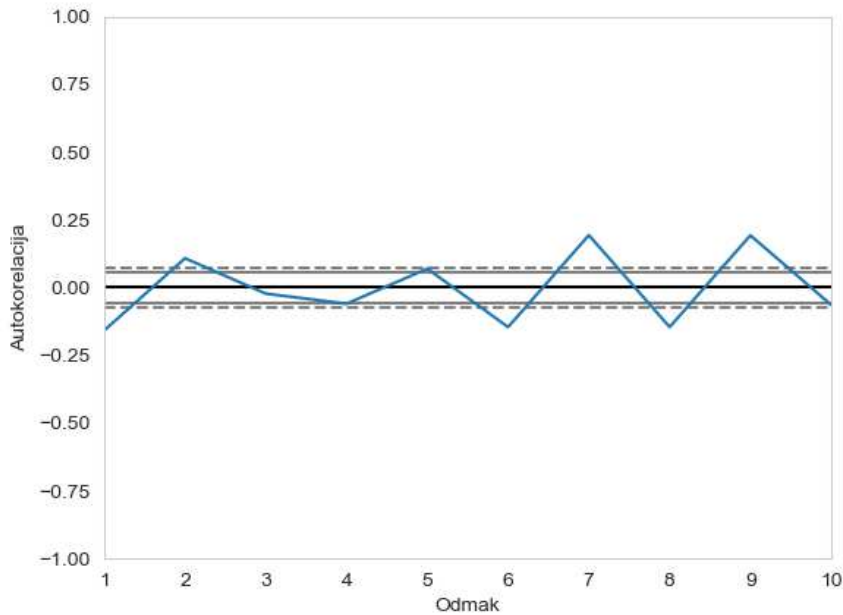
U slučaju da se promatra slabo stacionaran niz r_t , koncept korelacije može se općiti na autokorelacijsku funkciju (ACF, eng. *Autocorrelation Function*). Tada se ne analizira linearna povezanost dviju varijabli nego niza r_t i njezinih prošlih vrijednosti r_{t-l} od interesa. Koeficijent korelacije između r_t i r_{t-l} naziva se autokorelacija odmaka k i označava se s ρ_l . Pod pretpostavkom slabe stacionarnosti, ρ_l je funkcija ovisna samo o vremenskom pomaku l , a definira se kao

$$\rho_l = \frac{Cov(r_t, r_{t-l})}{\sqrt{Var(r_t)Var(r_{t-l})}} = \frac{Cov(r_t, r_{t-l})}{Var(r_t)} = \frac{\gamma_l}{\gamma_0}, \quad (2.10)$$

gdje se koristi svojstvo $Var(r_t) = Var(r_{t-l})$ koje proizlazi iz pretpostavke slabe stacionarnosti. Iz definicije 2.10 može se pokazati da vrijedi $\rho_0 = 1$, $\rho_l = \rho_{-l}$ i $-1 \leq \rho_l \leq 1$. Dodatno, slabo stacionarni niz r_t nije serijski koreliran ako i samo ako $\rho_l = 0$ za svaki $l > 0$.

2.3. Autoregresivni modeli

Činjenica da mjesečni povrati r_t Američkog indeksa S&P 500 imaju statistički značajnu autokorelaciju odmaka 1, vidljivo na slici 2.3., ukazuje na to da prošli povrat r_{t-1} može biti od koristi u predviđanju r_t .



Slika 2.3. Autokorelacija dnevnih logaritamskih povrata
Podatci: S&P 500 index

Jednostavan model koji se koristi prošlim povratom za predviđanje idućeg je:

$$r_t = \phi_0 + \phi_1 r_{t-1} + a_t, \quad (2.11)$$

gdje se pretpostavlja da je $\{a_t\}$ niz bijelog šuma sa srednjom vrijednosti nula i varijancom σ^2 . U literaturi vremenskih nizova, model 2.11 naziva se autoregresivnim (AR) modelom stupnja 1 ili jednostavnije AR(1) modelom. Iz definicije modela 2.11 može se pokazati kako vrijede sljedeća svojstva:

$$\begin{aligned} E(r_t | r_{t-1}) &= \phi_0 + \phi_1 r_{t-1} \\ \text{Var}(r_t | r_{t-1}) &= \text{Var}(a_t) = \sigma^2 \end{aligned} \quad (2.12)$$

To jest, ako znamo vrijednost r_{t-1} , r_t centriran je oko $\phi_0 + \phi_1 r_{t-1}$ sa standardnog devijacijom σ_a . Ovdje je prikazano Markovljevo svojstvo koje govori da je r_t nekoreliran od r_{t-i} za $i > 1$ ako znamo vrijednost r_{t-1} .

Međutim, postoje situacije u kojima sam r_{t-1} nema dovoljnu informaciju na temelju koje može predvidjeti r_t . Poopćenjem AR(1) modela dobije se AR(p) model definiran

kao;

$$\begin{aligned} r_t &= \phi_0 + \phi_1 r_{t-1} + \dots + \phi_p r_{t-p} + a_t \\ &= \phi_0 + \sum_{j=1}^p (\phi_j r_{t-j}) + a_t, \end{aligned} \quad (2.13)$$

gdje je p nenegativni cijeli broj i $\{a_t\}$ je definiran u modelu 2.11. Ovakav model govori da zadnjih p vrijednosti r_{t-i} ($i = 1, \dots, p$) zajedno određuje uvjetnu očekivanu vrijednost r_t .

2.3.1. Uvjet slabe stacionarnosti

Ako se pretpostavi slaba stacionarnost niza r_t onda vrijedi $E(r_t) = \mu$, $Var(r_t) = \gamma_0$ i $Cov(r_t, r_{t-j}) = \gamma_j$, gdje su μ i γ_0 konstante, a γ_j funkcija ovisna samo o j . Uzimajući očekivanje jednadžbe 2.11 i uzimajući u obzir da je po definiciji $E(a_t) = 0$ vrijedi:

$$E(r_t) = \phi_0 + \phi_1 E(r_{t-1}). \quad (2.14)$$

Pod uvjetom stacionarnosti, $E(r_t) = E(r_{t-1})$ i zbog toga vrijedi

$$\mu = \phi_0 + \phi_1 \mu \quad \text{ili} \quad E(r_t) = \mu = \frac{\phi_0}{1 - \phi_1} \quad (2.15)$$

Ovaj rezultat govori dvije stvari o r_t . Prvo, srednja vrijednost postoji samo ako $\rho_1 \neq 1$. Drugo, srednja vrijednost je nula samo ako $\rho_0 = 0$.

Sljedeće, ako se u jednadžbu 2.11 uvrsti $\phi_0 = (1 - \phi_1)\mu$ tada se ta jednadžba može zapisati kao

$$r_t - \mu = \phi_1(r_{t-1} - \mu) + a_t. \quad (2.16)$$

Ponavljajući supstituciju, jednadžba se na kraju može izraziti kao

$$\begin{aligned} r_t - \mu &= a_t + \phi_1 a_{t-1} + \phi_2 a_{t-2} + \dots \\ &= \sum_{i=0}^{\infty} (\phi_1^i a_{t-i}). \end{aligned} \quad (2.17)$$

Kvadriranjem jednadžbe 2.16, njezino očekivanje postaje

$$Var(r_t) = \phi_1^2 Var(r_{t-1}) + \sigma_a^2, \quad (2.18)$$

gdje je σ_a^2 varijanca slučajne varijable a_t te se koristi činjenica da je kovarijanca između r_{t-1} i a_t nula. To se može vidjeti iz toga što r_{t-1} ovisi samo o realizacijama niza do vremena $t - 1$ te realizacija a_t u trenutku t ne utječe na r_{t-1} . Pod pretpostavkom stacionarnosti vrijedi da je $Var(r_t) = Var(r_{t-1})$ tako da vrijedi

$$Var(r_t) = \frac{\sigma_a^2}{1 - \phi_1^2} \quad (2.19)$$

uz uvjet da je $\phi_1^2 < 1$. Taj uvjet proizlazi iz toga što je varijanca slučajne varijable uvijek nenegativna. Ovime je pokazano da je nužan i dovoljan uvjet za slabu stacionarnost niza r_t to da mora vrijediti $|\phi_1| < 1$.

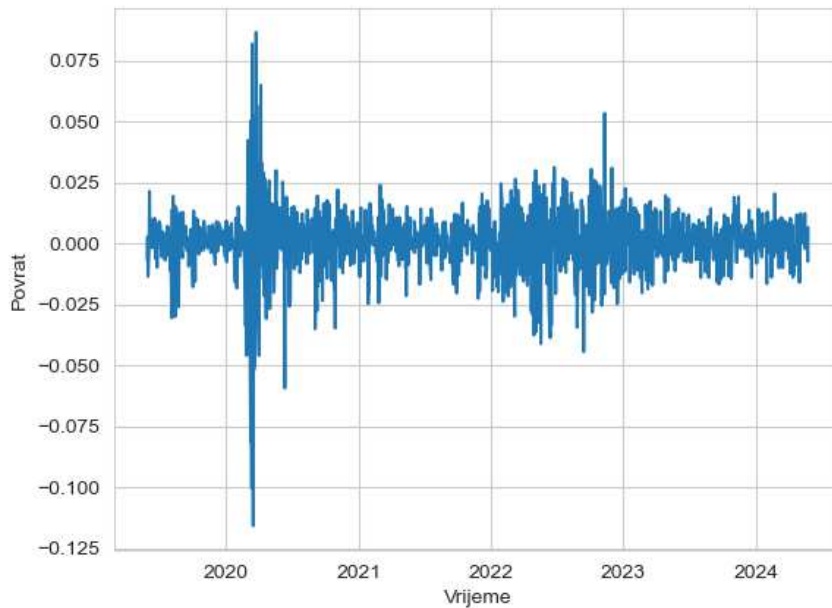
2.4. Modeliranje volatilnosti

Iako korisni za modeliranje povrata, obični AR modeli nemaju karakteristike potrebne za modeliranje volatilnosti. Modeliranje volatilnosti je ključno iz nekoliko razloga. Prvi razlog je taj što omogućava bolje razumijevanje rizika te shodno tome donosi mogućnost osiguravanja od istog. Visoka volatilnost može donijeti velike dobitke, ali isto tako i velike gubitke. Drugi razlog je taj što procjena volatilnosti omogućuje učinkovitije strategije upravljanja rizikom kao što je, na primjer, diverzifikacija portfelja. Uz to, procjena volatilnosti omogućuje procjenu vrijednosti financijskih instrumenata poput opcija ili derivata. Konačno, modeliranje volatilnosti doprinosi boljem razumijevanju dinamike tržišta te pomaže u donošenju informiranih odluka. Iz tog razloga, bitno je upoznati se s karakteristikama volatilnosti kako bi modeli što vjernije prikazali podatke.

2.4.1. Karakteristike volatilnosti

Posebna značajka volatilnosti financijskih imovina je da nije neposredno osmotriva [7]. Ako se promatraju dnevni podatci određene imovine, iz njih se ne može uočiti volatilnost jer je osmotriva samo jedna realizacija u danu. Ako postoje podatci unutar dana o cijeni te imovine, tada se volatilnost na dnevnoj razini može procijeniti iz tih podataka. Međutim, volatilnost između dva dana može se podijeliti na unutar-dnevnu i preko-noćnu volatilnost. Unutar-dnevne cijene imaju jako limitiranu informaciju o preko-noćnoj volatilnosti što stvara probleme prilikom modeliranja ukupne volatilnosti.

Iako volatilitnost nije direktno osmotriva, postoje neke karakteristike koje su često uočene kod povrata imovina. Prva od njih je da postoji grupiranje volatilitnosti, odnosno da volatilitnost može biti visoka za određena razdoblja i niska za druga [3]. Ta karakteristika vidljiva je na slici 2.4. koja prikazuje logaritamske povrate indeksa S&P 500 gdje se jasno vide područja veće i manje volatilitnosti. Druga karakteristika je da volatilitnost se mijenja kontinuirano kroz vrijeme, to jest, skokovi volatilitnosti su rijetki [3]. Treće, volatilitnost ne divergira u beskonačnost, točnije postoji neki fiksni raspon unutar kojeg volatilitnost varira [3]. Ova karakteristika ukazuje na to da je volatilitnosti često stacionarna. I četvrta karakteristika je da volatilitnost drugačije reagira na veliki skok i na veliki pad cijene [3]. Ove karakteristike igraju ključnu ulogu u razvijanju modela volatilitnosti.



Slika 2.4. Graf dnevni logaritamskih povrata
Podatci: S&P 500 index

Dodatno, potrebno je razmotriti uvjetno očekivanje i varijancu od r_t s obzirom na F_{t-1}

$$\begin{aligned}\mu_t &= E(r_t | F_{t-1}) \\ \sigma_t^2 &= Var(r_t | F_{t-1}) = E[(r_t - \mu_t)^2 | F_{t-1}],\end{aligned}\tag{2.20}$$

gdje F_{t-1} predstavlja skup dostupnih informacija u trenutku $t-1$. Uobičajeno, F_{t-1} se sastoji od svih linearnih funkcija prošlih povrata. Kao što je ranije rečeno, serijska ovisnost

povrata mala je ako uopće postoji te zbog toga μ_t se može predstaviti nekom jednostavnom funkcijom kao što je AR(p) model 2.21

$$r_t = \mu_t + a_t \quad (2.21)$$

$$\mu_t = \phi_0 + \sum_{i=1}^p (\phi_i r_{t-i})$$

Spajanjem jednažbi 2.20 i 2.21 dobije se

$$\sigma_t^2 = Var(r_t | F_{t-1}) = Var(a_t | F_{t-1}) \quad (2.22)$$

2.4.2. ARCH model

Osnovna ideja iza proučavanja volatilnosti je da je niz r_t ili serijski nekorelirana ili postoje male autokorelacije nižeg reda, ali je zavisna niz [2]. Modeli volatilnosti pokušavaju obuhvatiti takvu ovisnost u nizu povrata. Prvi model koji pruža sustavni okvir za modeliranje volatilnosti je ARCH (*eng. Autoregressive Conditional Heteroskedasticity*) model [2]. Ideja ARCH modela je da su reziduali a_t serijski nekorelirani, ali zavisni i da se zavisnost a_t može objasniti kvadratnom formulom prošlih vrijednosti a_{t-i} $t > 0$ Konkretno, ARCH(m) model pretpostavlja da vrijedi

$$a_t = \sigma_t \epsilon_t \quad (2.23)$$

$$\sigma_t^2 = \alpha_0 + \alpha_1 a_{t-1}^2 + \dots + \alpha_m a_{t-m}^2,$$

gdje je $\{\epsilon_t\}$ niz nezavisnih i jednako distribuiranih (iid) slučajnih varijabli sa srednjom vrijednosti 0 i varijancom 1, $\alpha_0 > 0$, $\alpha_i \geq 0$ $i > 0$. Uobičajeno se koristi pretpostavka da ϵ_t dolazi iz standardne normalne ili standardne studentove distribucije.

Iz strukture modela vidljivo je da veliki prijašnji kvadrirani reziduali $\{a_{t-i}^2\}_{i=1}^m$ impliciraju veliku uvjetnu varijancu σ_t^2 za a_t . To znači da nakon velikog reziduala vjerojatno, ali ne sigurno, slijedi još jedan veliki rezidual. Nije sigurno da će uslijediti veliki rezidual jer velika varijanca ne implicira veliku realizaciju. Međutim, povećanjem varijance, vjerojatnost događaja daljih od srednje vrijednosti se povećava. Ova karakteristika modela slična je grupiranju volatilnosti koje je opaženo kod stvarnih povrata imovina.

Karakteristike ARCH modela

Za razumijevanje ARCH modela isplati se prvo proučiti ARCH(1) model

$$\begin{aligned}a_t &= \sigma_t \epsilon_t \\ \sigma_t^2 &= \alpha_0 + \alpha_1 a_{t-1}^2,\end{aligned}\tag{2.24}$$

gdje su $\alpha_0 > 0$ i $\alpha_1 \geq 0$. Prvo, očekivana srednja vrijednost od a_t ostaje 0 zbog toga što vrijedi

$$E(a_t) = E[E(a_t|F_{t-1})] = E[\sigma_t E(\epsilon_t)] = 0.\tag{2.25}$$

Drugo, varijanca od a_t može se dobiti kao

$$\begin{aligned}Var(a_t) &= E(a_t^2) = E[E(a_t^2|F_{t-1})] \\ &= E(\alpha_0 + \alpha_1 a_{t-1}^2) = \alpha_0 + \alpha_1 E(a_{t-1}^2).\end{aligned}\tag{2.26}$$

Budući da je $\{a_t\}$ stacionaran proces s $E(a_t) = 0$, $Var(a_t) = Var(a_{t-1}) = E(a_{t-1}^2)$. Ako se to ubaci u jednadžbu 2.26 dobije se

$$\begin{aligned}Var(a_t) &= \alpha_0 + \alpha_1 Var(a_t) \\ &= \frac{\alpha_0}{1 - \alpha_1}\end{aligned}\tag{2.27}$$

Budući da varijanca od a_t mora biti pozitivna, dobije se uvjet koji mora biti zadovoljen a to je $0 \leq \alpha_1 < 1$.

Nedostatci ARCH modela

Prvi nedostatak ARCH modela je da pretpostavlja da pozitivni i negativni reziduali imaju isti utjecaj na volatilitnost zbog toga što ovisi o kvadratu prijašnjih reziduala. Kao što je ranije navedeno, empirijskom analizom povijesnih povrata pokazano je da to nije slučaj. Drugi nedostatak je da ARCH model ne pruža nikakav novi uvid u razumijevanje izvora varijance financijskog vremenskog niza. On samo pruža mehanički način za opisivanje ponašanja uvjetne varijance. Ne daje naznake o tome što uzrokuje takvo ponašanje. Treće je da će ARCH model vjerojatno precijeniti volatilitnost jer sporo reagira na velike izolirane rezidualne povratnog niza.

2.5. Neuronske mreže

Duboke unaprijedne neuronske mreže, ili višeslojni perceptroni (MLPs), fundamentalni su modeli dubokog učenja. Cilj jedne takve mreže je aproksimacija neke funkcije f^* . Na primjeru klasifikatora, $y = f^*(x)$ preslikava ulaz x u kategoriju y . Unaprijedna mreža definira preslikavanje $y = f^*(x|\theta)$ i uči parametre θ koji rezultiraju najboljom aproksimacijom funkcije [5].

Ovi modeli nazivaju se unaprijednim jer informacija teče kroz funkciju koja se procjenjuje iz ulaza x , kroz kalkulacije koje definiraju funkciju f i konačno do izlaza y . U takvim modelima ne postoji povratna veza u kojoj se izlaz modela vraća na ulaz. Međutim, ako se unaprijedne mreže prošire s povratnim vezama tada nastaju povratne neuronske mreže.

Unaprijedne neuronske mreže nazivaju se mrežama jer se obično sastoje od kombinacije više različitih funkcija. Model je predstavljen usmjerenim acikličkim grafom koji prikazuje kako su te funkcije povezane. Na primjer, tri funkcije $f^{(1)}$, $f^{(2)}$ i $f^{(3)}$ mogu biti povezane u lanac i tvoriti $f(x) = f^{(3)}(f^{(2)}(f^{(1)}(x)))$. U tom slučaju, $f^{(1)}$ se naziva prvim slojem, $f^{(2)}$ drugim slojem, i tako dalje. Duljina tog lanca označava dubinu modela. Zadnji sloj mreže naziva se izlazni sloj. Prilikom treniranja, funkcija $f(x)$ približava se ciljnoj funkciji $f^*(x)$. Podatci za treniranje daju zašumljene, približne primjere funkcije $f^*(x)$ procijenjene na različitim točkama. Svakom primjeru x pridružena je oznaka $y \approx f^*(x)$. Primjeri za treniranje točno određuju željeni izlaz za svaku točku x , to jest, model mora proizvesti vrijednost blizu y . Međutim, to ne određuje ponašanje ostalih slojeva. Kako se ponašaju skriveni slojevi i kako ih sastaviti za željeni izlaz je zadaća algoritma učenja.

Za kraj, ove mreže nazivaju se neuronskima jer su djelomično inspirirane neuroznanosti. Skriveni slojevi obično imaju vektorske vrijednosti. Dimenzionalnost ovog slojeva određuje širinu modela. Svaki element sloja može biti interpretiran slično neuronu. Umjesto razmišljanja da sloj izvršava jednu vektorsku funkciju, sloj se može interpretirati kao da izvršava mnogo funkcija više varijabli paralelno. Svaka jedinica nalikuje neuronu u smislu da prima ulazne podatke od mnogih drugih jedinica i izračunava vlastitu aktivacijsku vrijednost.

Kako bi se shvatile neuronske mreže, najbolje je početi od linearnih modela i vidjeti

kako poboljšati njihove nedostatke. Linearni modeli, kao što su linearne i logistička regresija, privlačni su jer se njihovi parametri mogu procijeniti učinkovito i pouzdano, bilo u zatvorenom obliku ili s konveksnom optimizacijom. Međutim, njihov nedostatak je da su limitirani samo na linearne funkcije stoga ne mogu modelirati interakcije bilo koje dvije varijable. Kako bi se proširili linearni modeli na nelinearne funkcije od x , linearni model može se primijeniti ne na x nego na transformirani ulaz $\phi(x)$, gdje ϕ predstavlja nelinearnu transformaciju. Ta transformacija može se interpretirati kao dodavanje značajki koje opisuju x ili kao potpuno nova reprezentacija za x .

Postavlja se pitanje kako odabrati transformaciju ϕ . Prva mogućnost je odabrati neku generičku transformaciju, kao na primjer beskonačno dimenzionalnu ϕ RBF jezgru. Ako je $\phi(x)$ dovoljno dimenzionalna, onda je uvijek moguće imati dovoljno kapaciteta da se nauče podatci za treniranje, ali svojstvo generaliziranja na neviđene podatke će ostati loše. Druga mogućnost je ručno kreiranje ϕ . Prije razvoja dubokog učenja ovo je bio dominantan pristup. Međutim, takav pristup zahtjeva mnogo ljudskog truda i domenskog znanja kako bi nove značajke bile korisne za zadatak koji se odrađuje. Treća mogućnost, koju koristi duboko učenje, je učenje reprezentacija ϕ . U tom slučaju, model se može zapisati kao

$$y = f(x; \theta, w) = \phi(x; \theta)^T w. \quad (2.28)$$

Kao što je vidljivo iz jednadžbe 2.28 sada postoje dva skupa parametara, θ koji se koriste za učenje preslikavanja ϕ i w koji služe za preslikavanje $\phi(x)$ u y . U ovom slučaju domensko znanje i trud su potrebni samo za definiranje familije funkcija preslikavanja $\phi(x)$, a ne više za doslovno kreiranje novih značajki. Ovakav pristup omogućava da mreža sama nauči koju funkciju, od svih dopuštenih, treba koristiti te samim time povećava kapacitet mreže. Kapacitet mreže, u slučaju dubokog učenja, predstavlja sposobnost mreže da nauči kompleksnija preslikavanja.

2.5.1. Učenje temeljeno na gradijentu

Dizajn i treniranje neuronskih mreža slični su treniranju drugih modela strojnog učenja pomoću gradijentnog spusta. Najveća razlika leži u tome što nelinearnost neuronskih mreža uzrokuje da većina funkcija gubitka postane nekonveksna. To znači da se neuronske mreže najčešće treniraju iterativnim postupcima baziranim na gradijentu

koji smanjuju funkciju gubitka na neku malu vrijednost, umjesto rješavanja linearnih jednadžbi koje se koriste prilikom treniranja linearne regresije ili algoritama konveksne optimizacije koji imaju garanciju konvergencije kod, na primjer, logističke regresije ili SVM-a [5]. Stohastički gradijentni spust primijenjen na nekonveksne funkcije nema garanciju konvergencije i osjetljiv je na vrijednosti početnih parametara. Za unaprijedne neuronske mreže važno je inicijalizirati sve težine W na male slučajne vrijednosti [8]. Pomaci b mogu se inicijalizirati ili na nulu ili na male pozitivne vrijednosti. Učenje parametara stohastičkim gradijentnim spustom odvija se u iteracijama definiranim:

$$\theta := \theta - \eta \nabla f(\theta) = \theta - \frac{\eta}{n} \sum_{i=1}^n \nabla f_i(\theta), \quad (2.29)$$

gdje $\nabla f(\theta)$ predstavlja gradijent funkcije gubitka koja ovisi o parametrima θ . Ako se radi s grupama od više podataka, stohastički gradijentni spust ažurira parametre s obzirom na prosjek gradijenata funkcije gubitka po svim primjerima.

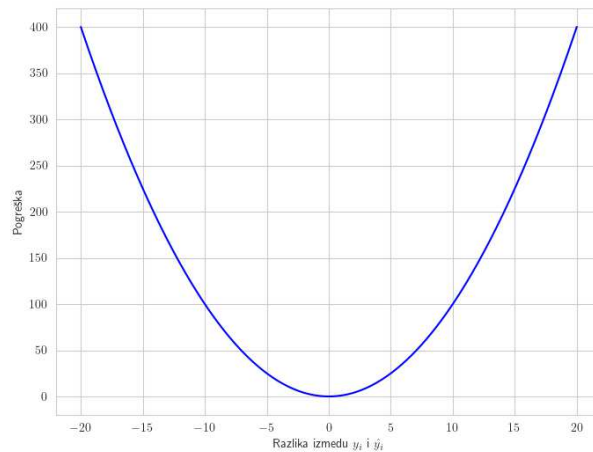
Funkcija gubitka

Kako bi se moglo provesti učenje temeljeno na gradijentu, potrebno je odrediti funkciju gubitka koja će dati informaciju o tome koliko dobro model radi. Ovisno o zadatku, funkcija gubitka unosi pretpostavke o tome kako bi model trebao raditi. Iako je moguće napraviti prilagođene funkcije gubitka, postoje neke često korištene ovisno o zadatku koji je potrebno riješiti.

Jedna od čestih funkcija koja se javlja prilikom regresijskih zadataka je funkcija gubitka najmanjih kvadrata (*eng. Mean Squared Error, MSE*), prikazana na slici 2.5., koja se definira kao

$$MSE = \frac{1}{n} \sum_{i=1}^n (y_i - \hat{y}_i)^2, \quad (2.30)$$

gdje y_i predstavlja stvarnu vrijednost oznake, a \hat{y}_i predstavlja izlaz modela. MSE se koristi iz više razloga, a neki od njih su jednostavnost, interpretabilnost i diferencijabilnost. Također, pošto se vrijednost odstupanja kvadrira, ovakva funkcija pogreške jako penalizira velika odstupanja. Međutim, ako u skupu podataka postoje stršće vrijednosti, na primjer zbog pogreške u mjerenju podataka, ovakva funkcija gubitka će im se jako prilagoditi.

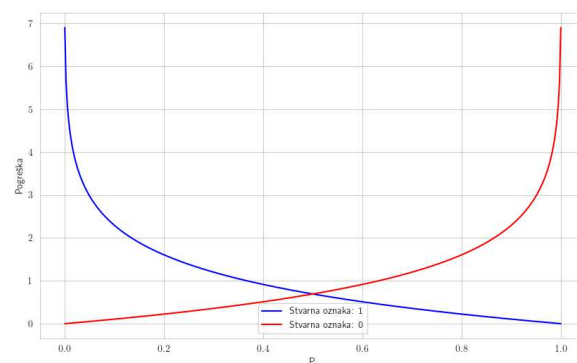


Slika 2.5. Funkcija gubitka najmanjih kvadrata

Ako je zadatak binarna klasifikacija, često korištena funkcija gubitka je gubitak unakrsne entropije (eng. *Cross-Entropy Loss, CE*), prikazana na slici 2.6., definirana kao

$$CE = -\frac{1}{n} \sum_{i=1}^n [y_i \log(\hat{y}_i) + (1 - y_i) \log(1 - \hat{y}_i)]. \quad (2.31)$$

Prednost ove funkcije je što penalizacija predikcija udaljenih od binarnih vrijednosti (0, 1). Također, korištenje logaritama u formuli 2.31 doprinosi penaliziranju niskih vjerojatnosti te numeričkoj stabilnosti. Uz to, ovakva formulacija pogreške se može lako generalizirati na više klasa korištenjem softmax funkcije. Kao kao i MSE, pogreška unakrsne entropije je diferencijabilna što omogućava učenje temeljeno na gradijentu.



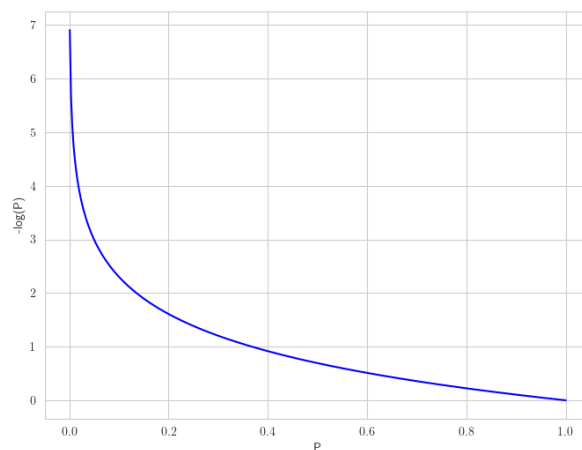
Slika 2.6. Funkcija gubitka unakrsne entropije

Još jedna funkcija gubitka koja se koristi u klasifikacijskim problemima je negativna

logaritamska izglednost (eng. *Negative Log-Likelihood, NLL*), prikazana na slici 2.7., definirana kao

$$NLL = - \sum_{i=1}^n (\log(P(y_i|x_i))), \quad (2.32)$$

gdje $P(y_i|x_i)$ predstavlja vjerojatnost pripadanja klase y_i za ulaz x_i . Prednost ovakvog modela je što je prikladan za probabilističke modele, odnosno ima vjerojatnosnu interpretaciju. Također, korištenje logaritama u 2.32 poboljšava performanse modela.



Slika 2.7. Funkcija gubitka negativne logaritamske izglednosti

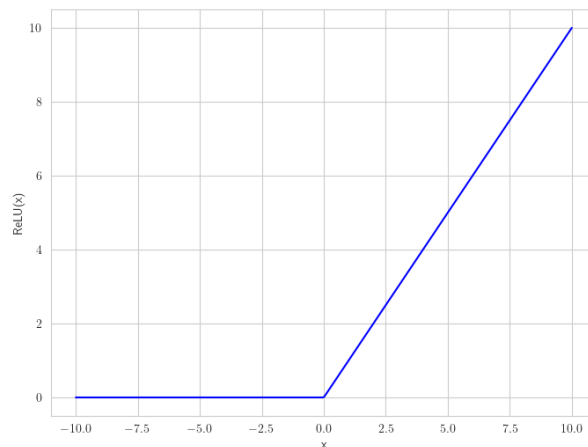
2.5.2. Aktivacijske funkcije

U jednadžbi 2.28 uvedena je nelinearnost kroz transformaciju ulaznih podataka ϕ . Međutim, što ako je mreža sastavljena od više slojeva? Ako svaki se svaki element sloja promatra kao jedinica koja obavlja matrično množenje $X^T w$, spajanjem više takvih slojeva zapravo se ne povećava kapacitet mreže. Razlog tome je što linearna kombinacija više linearnih kombinacija opet tvori linearnu kombinaciju. Odnosno, drugačije rečeno, ako se zbraja više matričnih množenja ulaza x_i s težinama w_i , identičan rezultat bi se dobio ako to sve zamijenimo jednim matričnim množenjem $z = W^{*T} x^* + b^*$. Zbog toga je potrebno uvesti i aktivacijske funkcije $g(z)$ koje će uvoditi nelinearnost na kraju svakog skrivenog sloja.

ReLU

Primjer jedne takve funkcije je ReLU (eng. *Rectified Linear Units*), prikazana na slici 2.8. koja je definirana kao

$$g(z) = \max\{0, z\}. \quad (2.33)$$



Slika 2.8. ReLU funkcija

Ovakva funkcija je jednostavna za optimizaciju jer je slična linearnoj funkciji. Jedina razlika između linearne funkcije i ReLU je u tome što ReLU vraća nulu za negativne vrijednosti ulaza. Zbog toga derivacije kroz ReLU ostaju velike kad god je jedinica aktivna. Gradijenti nisu samo veliki, već su i konzistentni. Druga derivacija operacije ReLU je gotovo posvuda 0, dok je prva derivacija 1 svugdje gdje je jedinica aktivna. To znači da je smjer gradijenta mnogo korisniji za učenje nego kod aktivacijskih funkcija koje uvode učinke drugog reda.

ReLU se obično koristi povrh affine transformacije:

$$h = g(W^T x + b). \quad (2.34)$$

Prilikom inicijalizacije parametara affine transformacije, preporučeno je postaviti sve elemente b na male pozitivne vrijednosti, na primjer 0.1. Time se dobiva da će, vrlo vjerojatno, izlaz affine transformacije biti aktivan za sve ulaze te će dozvoliti prolazak gradijenta.

Jedan nedostatak ReLU funkcije je taj što ne mogu učiti putem metoda temeljenih

na gradijentu na primjerima za koje je njihova aktivacija nula. Različite generalizacije ispravljenih linearnih jedinica jamče da dobivaju gradijent posvuda. Neke generalizacije ReLU funkcije temelje se na korištenju nenultog nagiba α_i za $z_i < 0$:

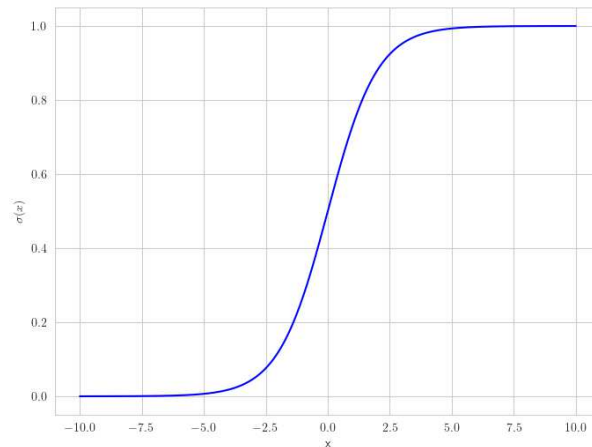
$$h_i = g(z, \alpha)_i = \max(0, z_i) + \alpha_i \min(0, z_i). \quad (2.35)$$

Primjeri takvih generalizacija su propusni ReLU (*eng. leaky ReLU*) koji postavi malu vrijednost za α_i , na primjer 0.01 i parametarski ReLU (*eng. parametric ReLU, PReLU*) koji tretira α_i kao parametar koji se uči. ReLU i njegove generalizacije temelje se na činjenici da je modele jednostavnije optimirati što su oni linearniji.

Sigmoidalna funkcija

Prije uvođenja ReLU-a, većina neuronskih mreža koristila je sigmoidalnu aktivacijsku funkciju

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}. \quad (2.36)$$



Slika 2.9. Sigmoidalna funkcija

Za razliku od ReLU funkcije, sigmoida je u zasićenju u većem dijelu svoje domene što je vidljivo na slici 2.9. U zasićenju je kada je z jako pozitivan ili jako negativan te nije u zasićenju samo kada je z oko nule. Raširena zasićenost sigmoidalne funkcije može otežati učenje temeljeno na gradijentu. Iz tog razloga, preporučuje se ne koristiti ju za aktivacijsku funkciju skrivenih slojeva.

Međutim, sigmoida se primjenjuje na drugim mjestima u dubokom učenju. Ako se radi o klasifikacijskom problemu te je potrebna vjerojatnosna interpretacija izlaza, sigmoida se može primijeniti na izlaz zadnjeg sloja te onda vrijednost izlaza mreže može se interpretirati kao vjerojatnost da podatak pripada određenoj klasi.

2.5.3. Arhitektura mreže

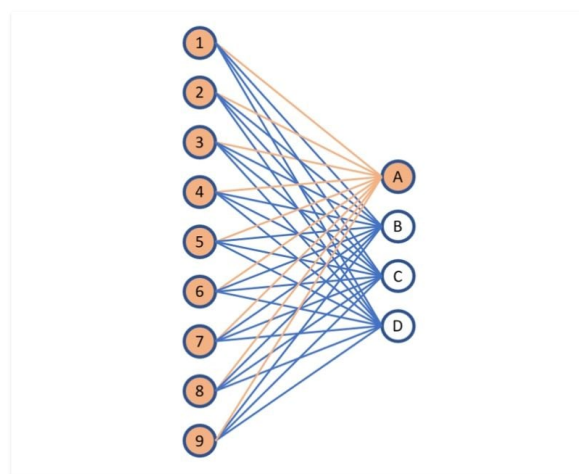
Kao što je ranije spomenuto, dodavanje novih slojeva i proširivanje trenutnih povećava kapacitet mreže te omogućuje učenje složenijih odnosa značajki. Međutim, prevelik kapacitet modela može vrlo lako dovesti do prenaučivosti, gdje se model previše prilagođava podacima za treniranje te gubi svojstvo generalizacije [5]. Stoga su razvijeni mnogi tipovi slojeva koji se mogu koristiti u neuronskim mrežama kako bi se na pametan način smanjio kapacitet mreže te unijela pristranost koja pomaže pri učenju takvih modela.

Potpuno povezani slojevi

Temeljni sloj neuronskih mreža je potpuno povezani (*eng. fully connected layer*) [5]. Potpuno povezani sloj, prikazan na slici 2.10., može se formulirati kao sloj u kojem su svi neuroni povezani s neuronima iz prethodnog sloja i definira se kao

$$h = g(W^T x + b), \quad (2.37)$$

gdje je h izlaz sloja, g je aktivacijska funkcija, W su težine, b je pomak, a x ulaz u sloj.



Slika 2.10. Potpuno povezani sloj. ¹

¹ <https://builtin.com/machine-learning/fully-connected-layer>

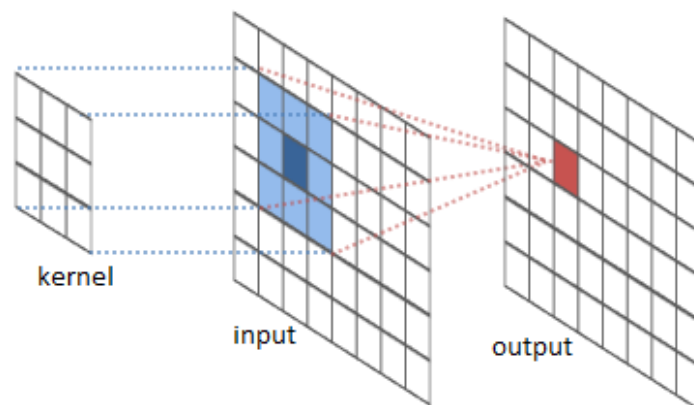
Iako jednostavni, ovakvi slojevi su vrlo fleksibilni i imaju velik kapacitet. Također, univerzalni su, odnosno mogu se primjenjivati na širokom rasponu zadataka. Međutim, kao što je ranije rečeno, veliki kapacitet ovakvih slojeva dovodi do prenaučnosti i nemogućnosti generalizacije. Velik kapacitet ovakvih slojeva proizlazi iz velikog broja parametara koji ga opisuju. Da bi se to izbjeglo, potrebno je upotrijebiti metode regulacije ili povećati skup podataka.

Konvolucijski slojevi

Još jedan način da se izbjegne prenaučnost je promjena modela s ciljem smanjenja broja parametara. Jedan od takvih slojeva je konvolucijski sloj, prikazan na slici 2.11. koji je osmišljen s ciljem otkrivanja lokalnih zavisnosti u podacima koji imaju prostornu strukturu [5]. Umjesto množenja težina W sa svim prijašnjim aktivacijama x , konvolucijski slojevi koriste jezgru (eng. *kernel*) K koja konvolucijom utječe na manji broj susjednih aktivacija.

$$h = X * K = \sum_{i=0}^m \sum_{j=0}^n X(m, n)K(i - m, j - n), \quad (2.38)$$

gdje m i n označavaju dimenzije ulaznog podatka. Ovakva formulacija lako se može proširiti na proizvoljan broj dimenzija.



Slika 2.11. Konvolucijski sloj. ²

Prednost ovakvog sloja je što dodavanje novih jezgara ne dodaje puno parametara jer se svaka jezgra primjenjuje na cijelu sliku, te se tako uči više značajki odjednom. Također, s obzirom na to da se vrši konvolucija, ovakav sloj je invarijantan na pomake te

² <https://towardsdatascience.com/beginners-guide-to-understanding-convolutional-neural-networks-ae9ed58bb17d>

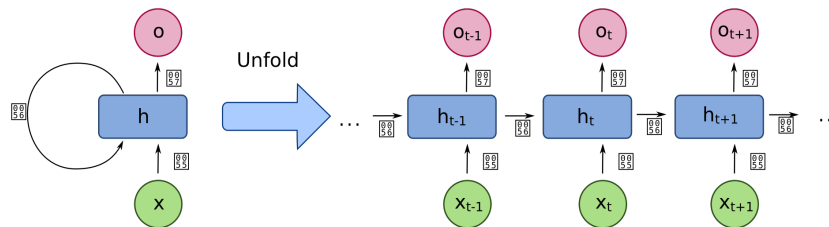
može naučiti značajke neovisno o njihovom položaju na slici. Međutim, mogući problem ovakvih slojeva je što zahtijevaju konstantnu veličinu ulaza, odnosno nije moguće primijeniti istu mrežu na više podataka različitih dimenzija.

Povratni slojevi

Za razliku od konvolucijskih slojeva, koji uče prostornu zavisnost podataka, povratni slojevi, prikazani na slici 2.12., služe za učenje vremenske zavisnosti podataka [5]. Oni koriste povratne veze kako bi omogućili informacijama da kruže unutar mreže te tako omogućuju zadržavanje kontekstualne informacije s prethodnih koraka. Informacija o prethodnim koracima spremljena je u skriveno stanje h_t na koje utječu trenutni ulaz x i prijašnje skriveno stanje h_{t-1} te rekurzivno onda i svi raniji ulaza i skrivena stanja. Najjednostavniji povratni sloj definiran je kao

$$\begin{aligned} h_t &= g_h(W_h x_t + U_h h_{t-1} + b_h) \\ y_t &= g_y(W_y h_t + b_y), \end{aligned} \tag{2.39}$$

gdje W i U predstavljaju matrice težina kojima se ulaz i prošlo skriveno stanje preslikavaju u prostor trenutnog skrivenog stanja.



Slika 2.12. Povratni sloj. ³

Iako korisni, ovakvi slojevi imaju više problema. Jedan od njih je problem eksplodirajućeg i nestajućeg gradijenta. Prilikom propagacije gradijenta unatrag, zbog množenja gradijenata svakog skrivenog sloja, može doći do nestajanja gradijenta. Ako su gradijenti blizu 0, kada se više njih pomnoži, tada ukupni gradijent teži u 0 zbog konačne preciznosti prikazivanja brojeva u računali. Također, ako su gradijenti veliki, njihovo množenje

³ <https://www.analyticsvidhya.com/blog/2022/03/a-brief-overview-of-recurrent-neural-networks-rnn/>

dovodi do toga da dođe do preljeva u decimalnom zapisu. Međutim, taj problem riješen je uvođenjem raznih novih slojeva baziranih na običnom povratnom sloju.

2.5.4. Regularizacija

Središnji problem strojnog učenja je izraditi algoritme koji će dobro raditi ne samo na podacima za treniranje, već i na novim, neviđenim podacima. Mnoge strategije korištene u strojnom učenju osmišljene su kako bi smanjile pogrešku na testnom skupu, čak i ako to znači povećanje pogreške na skupu za treniranje.

L_1 i L_2 regularizacija

Jedan od pristupa regularizaciji je kažnjavanje velike norme parametara modela kroz funkciju gubitka.

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha \|\theta\|_n, \quad (2.40)$$

gdje $\tilde{J}(\theta; X, y)$ predstavlja novu funkciju gubitka, a $\|\theta\|_n$ n-tu normu parametara. Pretpostavka ovakve regularizacije je da su velike težine znak prenaučivosti jer takvi modeli nisu robusni, odnosno, mala promjena u ulaznim podacima uzrokuje velike promjene na izlazu.

$$\|x\|_p = \left(\sum_{i \in \mathbb{N}} (|x|^p) \right)^{\frac{1}{p}} \quad (2.41)$$

Najčešće korištene norme su L_1 i L_2 norme. L_1 norma kažnjava s obzirom na apsolutnu vrijednost parametara te će takva regularizacija rezultirati rijetkim modelima, odnosno većina vrijednosti parametara θ bit će postavljene na 0. Međutim, L_1 regularizacija nije diferencijabilna stoga nije preferirana.

L_2 norma kažnjava s obzirom na kvadrat vrijednosti parametara te će jače kažnjavati veće parametre. Ali, kako se radi o kvadratu vrijednosti, za parametre manje od 1 snaga regularizacije se smanjuje te neće rezultirati rijetkim modelima.

3. Generiranje sintetičkih podataka

Cilj generiranja sintetičkih podataka, u ovom slučaju, je kreiranje eksperimenta u kontroliranim uvjetima. Takvim eksperimentom želi se provjeriti mogućnost neuronskih mreža u predviđanju idućeg povrata. Stvarni financijski podatci nisu pogodni za stvaranje takvog eksperimenta zbog niskog omjera informacije prema šumu što otežava testiranje korištenih metoda. Međutim, kako je cilj raditi predikciju na financijskim podacima, bilo bi poželjno da generirani podatci preslikavaju stvarna svojstva financijski nizova.

3.1. Stilizirane činjenice o povratima

Financijski podatci, kao što su cijene dionica i roba i tržišni indeksi, proučavaju se već više od pola desetljeća. Dostupnost velikih skupova podataka visokofrekventnih nizova cijena i pristup računalnim metodama za analizu podataka omogućili su uočavanje statističkih svojstava financijskih nizova bez pretpostavki modela [6]. Ta statistička svojstva nazvana su stilizirane činjenice, a neke od njih su navedene:

1. **Odsustvo autokorelacije:**

Autokorelacije povrata su često statistički neznačajne, osim u kratkim unutar dnevnim skalama (≈ 20 minuta)

2. **Teški repovi:**

Repovi (bezuovjetne) distribucije povrata slični su repovima Pareto distribucije s indeksom repa većim od 2 i manjim od 5 za većinu proučavanih skupova podataka.

3. **Asimetrija distribucije:**

Distribucija povrata ima negativan treći moment što govori da se veliki negativni povrati događaju češće nego veliki pozitivni povrati. Na grafu distribucije je vidljivo

preko toga što je lijevi rep distribucije duži, a masa distribucije je koncentrirana na desnoj strani.

4. **Asimetrija dobitka/gubitka:**

Uočeno su veliki padovi cijena dionica, ali ne i jednako veliki prirasti.

5. **Grupiranje volatilnosti:**

Različite metode mjerenja volatilnosti pokazuju pozitivnu autokorelaciju što kvantificira činjenicu da trenutci visoke volatilnosti budu grupirani.

6. **Sporo opadanje autokorelacije apsolutnih povrata:**

Autokorelacijska funkcija apsolutnih povrata polako opada kao funkcija vremenskog odmaka, otprilike kao potencijski zakon s eksponentom $\beta \in [0.2, 0.4]$. To se može interpretirati kao znat dugotrajne ovisnosti.

3.2. **Generiranje podataka**

Prilikom generiranja podataka uvedene su određene pretpostavke. Prva takva pretpostavka je da srednja vrijednost i standardna devijacija idućeg povrata ovise o prošlim n , u ovom slučaju 5, realizacija povrata. Druga uvedena pretpostavka je da uvjetna distribucija povrata s obzirom na prošlim n realizacija povrata prati normalnu razdiobu. Tako da je proces generiranja podataka podijeljen u tri djela: generiranje srednje vrijednosti, generiranje varijance i generiranje povrata na temelju dobivenih parametara.

Generiranje srednje vrijednosti

Srednje vrijednosti povrata modelirane su autoregresivnim modelom opisanim u poglavlju 2.3. Pretpostavka da takav model vrijedi proizlazi iz činjenice da povrati imaju statistički značajnu autokorelaciju malog odmaka te da se informacija o prošlim realizacijama povrata mogu iskoristiti za bolju predikciju idućih povrata. Konačni model kojim su generirane srednje vrijednosti glasi

$$\mu_t = \phi_0 + \sum_{i=1}^5 (\phi_i r_{t-i}). \quad (3.1)$$

Parametar ϕ_0 postavljen je na nulu jer tada očekivanje srednje vrijednosti μ_t je isto nula. Kako bi se modelirala jačina zavisnosti o prijašnjim povratima, parametri ϕ_i , $1 \geq i \geq 5$ postavljeni su da kvadratno opadaju s faktorom 0,25, odnosno $\phi_i = 0.25^i$. Tako najveći utjecaj dodijeljen je vremenski najbližem prošlom povratu i sve manji povratima prije toga. Također, pošto su svi parametri ϕ_i manji od 1, zadovoljen je uvjet stacionarnosti.

Generiranje varijance

Modeliranje varijance, isto kao i srednje vrijednosti, iskorištava informacije prošlih povrata za predikciju novih. Razlika je u tome što varijanca, kao mjera disperzije, mjeri odstupanje podataka od srednje vrijednosti te predznak odstupanja ne bi trebao biti bitan. Zbog toga se prošli povrti kvadriraju i onda je model varijance definiran kao

$$\sigma_t^2 = \theta_0 + \sum_{i=1}^5 (\theta_i r_{t-i}^2). \quad (3.2)$$

Parametar θ_0 postavljen je na malu vrijednost 0.01 dok su parametri θ_i , $1 \geq i \geq 5$ postavljeni na vrijednost 0.2. U slučaju volatilnosti, kvadrati prošlih povrata ne daju informaciju o trendu, kao u modeliranju srednje vrijednosti, nego o magnitudi odstupanja. Ako su prošli povrti bili veliki, veću vjerojatnost dajemo događajima ponovne veće volatilnosti. Pretpostavka takvog modela proizlazi iz toga što je opaženo da povrti imaju sporo opadajuću funkciju apsolutnih i kvadriranih povrata.

Generiranje povrata

Generiranje povrata provodi se uzorkovanjem iz normalne distribucije uvjetovane prošlim povratima.

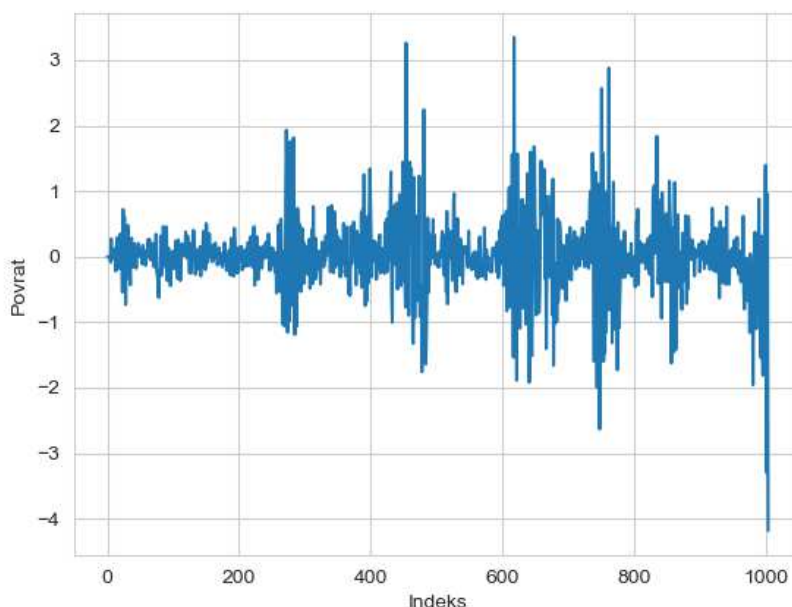
$$r_t \sim \mathcal{N}(\mu_t(r_{t-1}, \dots, r_{t-5}), \sigma_t(r_{t-1}, \dots, r_{t-5})) \quad (3.3)$$

Uvjetnost prošlim povratima dolazi zato što parametri normalne distribucije nisu konstantni nego su funkcije koje ovise o prošlim povratima. Takvim generiranjem povrata očuvana je pretpostavka normalnosti koja se u financijskim analizama često koristi, ali je fleksibilnija jer dopušta promjenu parametara s obzirom na prijašnje povrate.

3.3. Karakteristike generiranih podataka

Cilj generiranja sintetičkih podataka je da vjerno preslikavaju karakteristike stvarnih podataka. U ovom slučaju, radi se o generiranju sintetičkih financijskih vremenskih nizova pa je cilj da takav niz ima karakteristike koje su opisane u poglavlju 3.1.

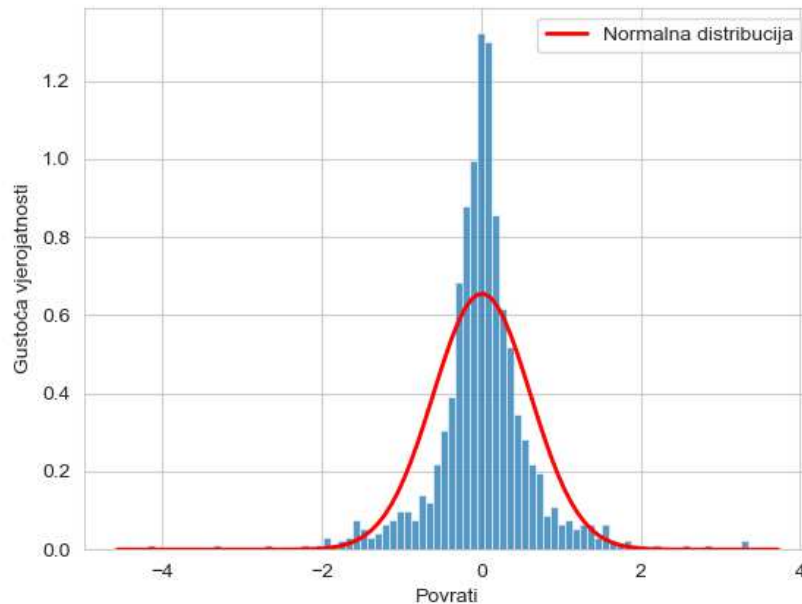
Jedan generirani niz prikazan je na slici 3.1. Iako skala povrata nije prigodna za stvarne financijske podatke, taj problem može se lako riješiti skaliranjem cijelog niza. Kako je očekivanje niza generiranog modelima 3.1 i 3.2 jednako nuli, skaliranje neće utjecati na tu karakteristiku. Također, mjera autokorelacija neovisna je o skali podataka tako da skaliranje neće utjecati niti na nju.



Slika 3.1. Graf generiranih povrata

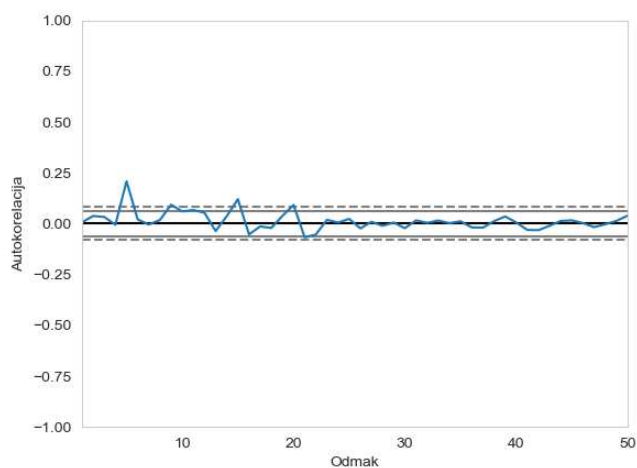
Za početak potrebno je proučiti distribuciju generiranih povrata. Na slici 3.2. prikazan je histogram generiranih povrata i funkcija gustoće normalne razdiobe sa parametrima srednje vrijednosti i varijance izračunatima na podacima. Prvo svojstvo koje je održano je da distribucija povrata ima teže repove od normalne distribucije što se očituje sa više velikih povrata po apsolutnoj vrijednosti te višim modom distribucije. To potvrđuje i četvrti moment distribucije koji za ovaj niz iznosi 1.313 što ukazuje na teže repove i viši mod distribucije. Iako se na slici 3.2. može izgledati kao da je očuvano i svojstvo

asimetrije distribucije, odnosno da je lijevi rep teži, treći moment distribucije koji to opisuje je -0.048 što govori da je distribucija skoro simetrična. Također, generiranjem 100 nizova, srednja vrijednost razlike apsolutnih vrijednosti najvećeg negativnog i najvećeg pozitivnog povrata iznosi -33.043 što ukazuje da ovako generirani nizovi održavaju svojstvo asimetrije gubitka i dobitka.

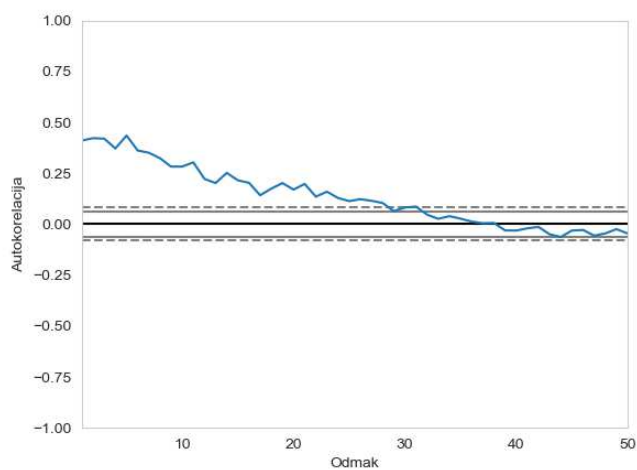


Slika 3.2. Histogram generiranih povrata i funkcija gustoće normalne razdiobe

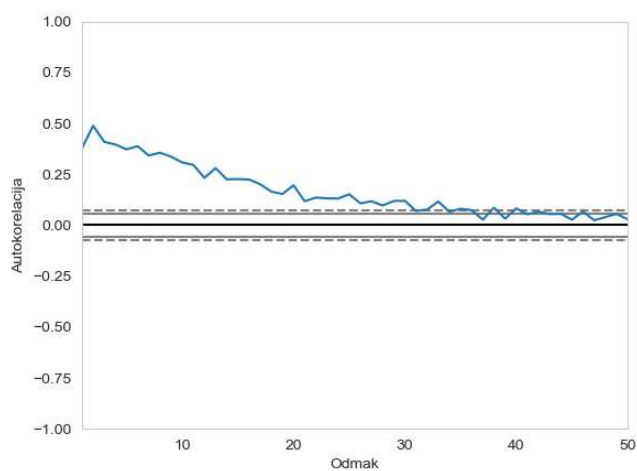
Bitna svojstva koja je još potrebno provjeriti su odsustvo autokorelacije i sporo opadajuća autokorelacija apsolutnih povrata. Na slici 3.3. prikazana je autokorelacija generiranih povrata. Može se primijetiti kako postoji jedna vrijednost autokorelacijske funkcije koja je statistički značajna, a to je vrijednost na odmaku 5. Ostale vrijednosti su blizu nule te se može reći kako je očuvano svojstvo odsustva autokorelacije. Uz to, na slici 3.4. prikazana je autokorelacijska funkcija apsolutnih generiranih povrata koja sporo opada. Usporedbom apsolutne autokorelacije generiranih povrata 3.4. i apsolutne autokorelacije stvarnih podataka 3.5. vidljivo je da oba niza imaju sličnu strukturu te slične odmake sa statistički značajnim autokorelacijama.



Slika 3.3. Autokorelacija generiranih povrata



Slika 3.4. Autokorelacija apsolutnih generiranih povrata



Slika 3.5. Autokorelacija apsolutnih dnevnih povrata indeksa S&P 500

I zadnja stilizirana činjenica koju generirani povrati poželjno imaju je grupiranje volatilnosti. To svojstvo najbolje je vidljivo iz grafa povrata u vremenu 3.1. gdje se vidi da su područja više volatilnosti podijeljena u više grupa koje su isprekidane područjima manje volatilnosti. Uz to, svojstvo grupiranja volatilnosti može se vidjeti i iz autokorelacijskog grafa apsolutnih povrata 3.4. jer grupiranje volatilnosti pridonosi vremenskoj zavisnosti odstupanja, što se u ovom slučaju mjeri apsolutnom vrijednosti povrata.

4. Modeliranje povrata

Modeliranje povrata financijske imovine ključno je za razumijevanje kretanja tržišta i donošenje informiranih investicijskih odluka. Povrati, kao ključni pokazatelji performansi financijskih instrumenata, pružaju uvid u promjene vrijednosti imovine kroz vrijeme. Tradicionalni modeli, poput autoregresivnih (AR) i generalizirane uvjetne heteroskedastičnosti (GARCH), dugo su bili standardni alati za analizu financijskih vremenskih nizova. Iako, u ovom radu korišteni za generiranje podataka, takvi modeli imaju ograničenja na složenost koju mogu uhvatiti.

S razvojem dubokog učenja, neuronske mreže su se pokazale kao moćan alat za modeliranje složenih i nelinearnih odnosa u podacima. Neuronske mreže, zbog svoje sposobnosti učenja iz velikih količina podataka i prepoznavanja složenih obrazaca, pružaju značajne prednosti u odnosu na tradicionalne modele.

Glavni razlog za korištenje neuronskih mreža u modeliranju povrata leži u njihovoj sposobnosti da se nose s visokom dimenzionalnošću podataka i prepoznaju nelinearne zavisnosti koje su često prisutne u financijskim vremenskim nizovima. Tradicionalni linearni modeli često ne mogu adekvatno uhvatiti ove složenosti, dok neuronske mreže, zahvaljujući svojim višeslojnim strukturama, mogu modelirati složene interakcije između varijabli. Osim toga, duboki generativni modeli omogućuju nam generiranje sintetičkih podataka koji mogu biti korisni za testiranje različitih hipoteza i simulaciju tržišnih uvjeta u kontroliranim okruženjima.

4.1. Generativno modeliranje

Generativno modeliranje je pristup u strojnom učenju koje se fokusira na učenje distribucije podataka $P(X)$, gdje je X prostor podataka. Za razliku od diskriminativnih modela,

koji se koriste za klasifikaciju ili regresiju, generativni modeli pokušavaju razumjeti unutarnju strukturu podataka. Uz razumijevanje, učenje distribucije podataka omogućuje i generiranje sintetičkih podataka koji prate distribuciju naučenu iz podataka. Glavni zadatak treniranja takvih modela je naučiti parametre distribucije koja opisuje podatke.

Postoji nekoliko pristupa generativnom modeliranju, a neki od najpopularnijih su:

- **Naivni Bayes:** koristi se za modeliranje uvjetne vjerojatnosti podataka pretpostavljajući neovisnost među značajkama. Ako pretpostavka o neovisnosti ne vrijedi za sve značajke, ovaj model može se proširiti na grupe međusobno neovisnih značajki.
- **Boltzmannove mašine:** koriste energetske funkcije za modeliranje distribucije podataka.
- **Varijacijski autoenkoderi (VAE):** koriste neuronske mreže za mapiranje podataka na latentni prostor i natrag. Latentni prostor, obično manje dimenzije od ulaznog prostora, omogućuje izvlačenje bitnih značajki iz podataka te rekonstrukciju istih iz tog manje zašumljenog prostora.
- **Generativne suparničke mreže (GAN):** koriste dvije suparničke mreže (generator i diskriminator) koje se natječu jedna protiv druge kako bi poboljšale kvalitetu generiranih podataka

4.2. Priprema podataka za treniranje

Kako bi treniranje mreže bilo brže i stabilnije, potrebno je pripremiti podatke. Za početak, kako modeli za generiranje povrata ovise o prošlim povratima, potrebno je inicijalizirati početne vrijednosti povrata na temelju kojih će se generirati niz. Prvih n , u ovom slučaju 5, vrijednosti povrata inicijalizirane su na vrijednost 0. Pomoću tih vrijednosti generiran je niz povrata duljine 1005.

Jedan od trikova kako ubrzati učenje je normalizacija podataka prije treniranja (Efficient BackProp ref). Stoga je od niza povrata oduzeta njegova srednja vrijednost i podijeljena je sa standardnom devijacijom. Tako transformirani niz ima srednju vrijednost nula i standardnu devijaciju 1 te samim time, vrijednosti niza nisu niti prevelike niti pre-

male te se težine mreže mogu lakše naučiti. Ako bi ulazni podatci bili preveliki, model bi to kompenzirao manjim težinama te bi bila potrebna veća promjena na ulazu kako bi se promijenio izlaz. Također, ako bi ulazni podatci bili premali, model bi naučio veće težine kako bi kompenzirao množenje te bi svaka manja promjena na ulazu uzrokovala velike promjene na izlazu.

Normalizirani niz povrata podijeljena je u primjere koji sadrže ulazne podatke, odnosno 5 uzastopnih povrata i ciljanu vrijednost povrata koja neposredno slijedi. Skup podataka sada sadrži 1000 instanci primjera. Međutim, kako bi se mogle testirati performanse modela, podatke je potrebno podijeliti u tri podskupa, skup za treniranje, skup za validaciju i skup za testiranje. Skup za treniranje služi za treniranje modela te on sadržava 70% svih podataka. Skup za validaciju koristi se kako bi se provjeravalo je li model u području prenaučnosti ili podnaučnosti te se na temelju performansi na tom skupu određuje najbolji model tijekom treniranja. Skup za testiranje, kao i skup za validaciju, sadrže po 15% ukupnog broja podatka. Nakon procesa treniranja, skup za testiranje koristi se kako bi se testirale performanse gotovog modela.

4.3. Modeliranje povrata korištenjem neuronske mreže

U ovom radu korištene su i uspoređene tri arhitekture za učenje distribucije povrata. Sve arhitekture sastoje se od dvije neuronske mreže, jedna za procjenu srednje vrijednosti i druga za procjenu varijance, koje se treniraju zajedničkom funkcijom gubitka. Razlog zbog kojeg se koristi jedna funkcija gubitka je taj što srednja vrijednost i varijanca idućeg povrata nisu osmotrivi, nego je idući povrat jedna realizacija varijable koja je opažena na tržištu. Zbog toga učenje mreža odvojeno nije moguće jer ne postoji informacija s kojom bi se izračunali posebni gubitci srednje vrijednosti i varijance nego se oba parametra koriste za procjenu logaritamske izglednosti idućeg podatka. Također, mreža koja procjenjuje varijabilnost na izlazu daje logaritam varijance zato što varijanca uvijek mora bit nenegativna te logaritmiranjem osiguramo da izlaz mreže koji ja u skupu realnih broje rezultira nenegativnom varijancom.

Prva arhitektura sastoji se od dvije potpuno povezane mreže, od kojih je jedna namijenjena procjeni srednje vrijednosti a druga procijeni varijance. Mreža za procjenu srednje vrijednosti sastoji se od četiri potpuno povezana sloja s po 512 neurona, pri čemu svi slo-

jevi, osim posljednjeg koriste ReLU aktivacijsku funkciju. Mreža za procjenu varijance sastoji se od pet potpuno povezanih slojeva s po 1024 neurona, također koristeći ReLU aktivacijsku funkciju u svim slojevima osim u završnom sloju.

Druga arhitektura integrira LSTM slojeve uz potpuno povezane slojeve. Mreža za procjenu srednje vrijednosti sastavljena je od dva LSTM sloja sa skrivenim stanjem veličine 512, te tri potpuno povezana sloja s po 512 neurona koji su, svi osim posljednjeg, aktivirani ReLU funkcijom. Svrha LSTM slojeva je učenje vremenske zavisnosti podataka, što je relevantno budući da se za procjenu parametara koriste vremenski uzastopni podatci s prethodno opisanim karakteristikama. Mreža za procjenu varijance sastoji se od dva LSTM sloja sa skrivenim stanjem veličine 512 te četiri potpuno povezana sloja s po 1024 neurona i ReLU aktivacijskom funkcijom.

Treća arhitektura predstavlja kombinaciju prvih dviju, pri čemu se za procjenu varijance koriste isključivo potpuno povezani slojevi, dok se za procjenu srednje vrijednosti koriste i LSTM slojevi. Mreža za procjenu srednje vrijednosti ima dva LSTM sloja sa skrivenim stanjem veličine 512 i tri potpuno povezana sloja s po 512 neurona, dok mreža za procjenu varijance ima pet potpuno povezanih slojeva s po 1024 neurona. LSTM slojevi nemaju aktivacijsku funkciju jer je na njihovom izlazu već postavljena sigmoida te nije potrebno dodatno unositi nelinearnost.

4.4. Postupak treniranja

Model na ulaz dobiva 5 prijašnjih realizacija povrata i na temelju njih procjenjuje srednju vrijednost i varijancu idućeg povrata. Iz tog razlog funkcija gubitka definirana je kao negativna logaritamska izglednost idućeg povrata po normalnoj distribuciji s parametrima procijenjenima s pomoću modela.

$$\begin{aligned}
 L_t(r_t; \hat{\mu}_t, \hat{\sigma}_t) &= - \left[\log \left(\frac{1}{\sqrt{2\pi\hat{\sigma}_t^2}} \right) + \log \left(\exp \left(-\frac{(r_t - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2} \right) \right) \right] \\
 &= \frac{1}{2} \log(2\pi\hat{\sigma}_t^2) + \frac{(r_t - \hat{\mu}_t)^2}{2\hat{\sigma}_t^2}
 \end{aligned} \tag{4.1}$$

Prilikom treniranja, podatci za treniranje podijeljeni su u mini grupe veličine 256.

Time se dobiva nepreciznija procjena gradijenta, jer se ne koriste svi podatci, te time mreža može izbjeći lokalne optimume. Također, podjelom u mini grupe, efektivno mreža prođe kroz više iteracija treniranja jer se u jednoj epohi više puta izračuna i primjeni gradijent.

Treniranje se odvijalo u 200 epoha. Tijekom svake epohe, mreža je dobivala grupe podataka za treniranje te na temelju njih izračunavala gradijent pogreške. Za propagaciju pogreške po slojevima koristio se ADAM (eng. *Adaptive Moment Estimation*) optimizator koji, uz gradijent, koristi prvi m i drugi moment v . Moment se koristi kako bi se treniranje na monotonim dijelovima ubrzalo. Procjena prvog i drugog momenta provodi se prilikom svake iteracije i ovisi o trenutnom gradijentu i prošlim momentima.

$$\begin{aligned}
 g_t &= \nabla_{\theta} f(\theta_{t-1}) \\
 m_t &= \frac{\beta_1 * m_{t-1} + (1 - \beta_1) * g_t}{1 - \beta_1^t} \\
 v_t &= \frac{\beta_2 * v_{t-1} + (1 - \beta_2) * g_t^2}{1 - \beta_2^t} \\
 \theta_t &= \theta_{t-1} - \eta * \frac{m_t}{\sqrt{v_t} + \epsilon}
 \end{aligned} \tag{4.2}$$

U ovom slučaju, ϵ je mala konstanta, uobičajeno 10^{-8} , koja prevenira dijeljenje s nulom. Početna stopa učenja η postavljena je na 10^{-3} . Međutim, stopa učenja nije konstantna prilikom cijelog procesa treniranja. Na završetku svake epohe, model je evaluiran na podacima za validaciju te ako se validacijska pogreška ne bi smanjivala određen broj epoha, stopa učenja bi se smanjila za određeni faktor. Konkretno, u ovom radu korišten je faktor 0.5 koji bi se primjenjivao ako 10 epoha nije došlo do poboljšanja. Također, postavljena je minimalna stopa učenja na 10^{-6} kako se učenje ne bi skroz zaustavilo.

Dodatno, na kraju svake epohe, kada je model evaluiran na podacima za validaciju, ako je validacijska pogreška trenutnog modela najmanja do sada, taj model se sprema u memoriju. Tako nije potrebno implementirati rano zaustavljanje u borbi protiv prenaučivosti jer na kraju postupka treniranja moguće je učitati model s najboljom validacijskom pogreškom te njega koristiti. To je napravljeno i u ovom radu te su svi rezultati koji će biti prikazani dobiveni na takvim modelima.

4.5. Evaluacija performansi

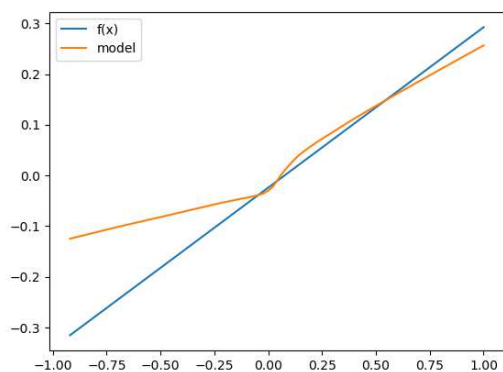
Evaluacija performansi modela provedenih u ovom radu uključuje grafičku i kvantitativnu analizu. Za početak, kako su podatci generirani s pomoću funkcija navedenih u poglavlju 3.2., moguće je provjeriti performanse modela usporedbom stvarne funkcije i funkcije dobivene modelom. Na taj način moguće je proučiti u kojim područjima model bolje ili lošije funkcionira te pokušati otkriti razlog toga.

Kako bi se mogla provesti grafička analiza potrebno je napraviti preslikavanje ulaznog prostora, koji je peterodimenzionalan, na jednu dimenziju. Takvo preslikavanje omogućuje vizualizaciju rezultata u dvije dimenzije. Za početak, potrebno je odrediti domenu na kojoj će se modeli evaluirati. U ovom slučaju domena je skup od 101 jednako udaljen realan broj n_i između -1 i 1. Međutim, kako model zahtjeva peterodimenzionalan ulaz, podatci su grupirani u skupove od 5 podataka tako da podatak x na mjestu i sadrži brojeve s indeksima $i, i + 1, i + 2, i + 3, i + 4$, odnosno

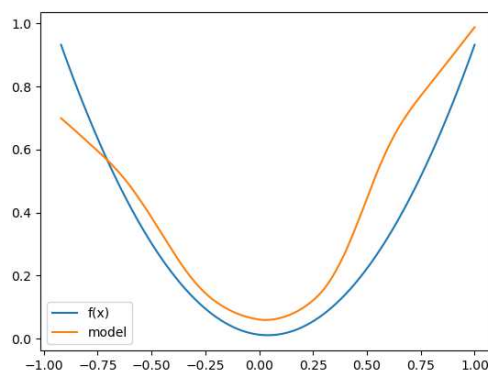
$$x_i = (n_i, n_{i+1}, n_{i+2}, n_{i+3}, n_{i+4}). \quad (4.3)$$

U svim idućim vizualnim prikazima rezultata y-os označava vrijednost tražene funkcije kojom su generirani podatci te vrijednost izlaza modela dok x-os označava najmanju vrijednost u skupu x_i odnosno vrijednost n_i nad kojim je provedena evaluacija.

Na slikama 4.1. vidljivi su rezultati grafičke usporedbe izlaza linearnih modela i funkcija preko kojih su podatci generirani. Može se primijetiti kako je model bolje naučio pozitivne srednje vrijednosti povrata od negativnih. Suprotno tome, model je bolje naučio varijance za primjere s negativnim prijašnjim povratima. To se može interpretirati tako da je model kompenzirao procjenu jedne varijable drugom jer obje variable utječu na funkciju gubitka. Razlog tome može biti manji broj podataka u tom području te je model bolje mogao naučiti, na primjer, varijabilnost prijašnjih povrata nego točnu srednju vrijednost idućeg povrata.



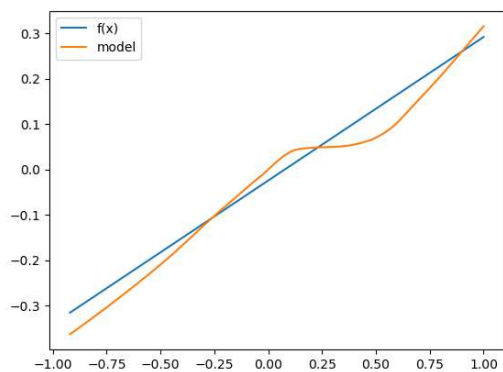
(a) Srednje vrijednosti



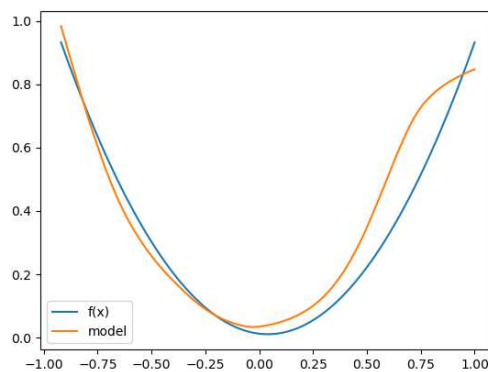
(b) Varijance

Slika 4.1. Grafovi funkcija za generiranje podataka i procjena funkcija pomoću linearnih modela

U slučaju modela s povratnim vezama, odnosno modela koji u sebi sadrži LSTM slojeve, rezultati su prikazani na slikama 4.2. Rezultati takvog modela su vidljivo bolji, te procjene takvog modela su bliže stvarnim funkcijama koje su generirale podatke. Takvi rezultati imaju smisla zbog toga što je u taj model ugrađena pretpostavka o vremenski zavisnim događajima te se informacija propagira u vremenu što dovodi do bolje procjene parametara.



(a) Srednje vrijednosti

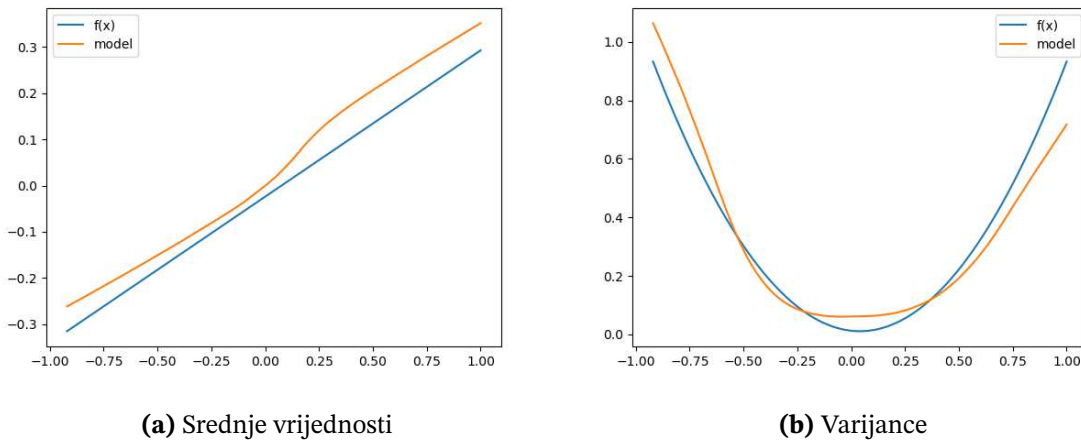


(b) Varijance

Slika 4.2. Grafovi funkcija za generiranje podataka i procjena funkcija pomoću LSTM modela

Treća arhitektura koja je korištena je spoj LSTM arhitekture za procjenu srednje vrijednosti i potpuno povezane arhitekture za procjenu varijance. Rezultati takvog modela prikazani su na slikama 4.3. Ono što se može primijetiti, kao kod modela s povratnim vezama, ovakav je model dobro naučio vremensku zavisnost povrata te funkcija procije-

njene srednje vrijednosti slični funkciji generiranja srednje vrijednosti. Također, potpuno povezani model dobro procjenjuje varijancu.



Slika 4.3. Grafovi funkcija za generiranje podataka i procjena funkcija pomoću linearnog i LSTM modela

Uz grafički analizu, provedena je i kvantitativna analiza performansi. Ranije navedeni modeli trenirani su na tri skupa stvarnih financijskih podataka. Prvi skup podataka čine povrati indeksa *Standards and Poor's (SPY)* 500 koji prati 500 najvećih američkih kompanija. Drugi skup podataka su povrati indeksa *iShares 20+ Year Treasury Bond ETF (TLT)* koji prati američke obveznice s dospijecem preko 20 godina. I treći skup korišten u analizi su povrati indeksa *Invesco Optimum Yield Diversified Commodity Strategy No K-1 ETF (PDBC)* koji prati robne terminske ugovore. Sva tri skupa sadržavaju povijesne podatke od datuma 28. 5. 2019. do datuma 24. 5. 2024.

Rezultati analize prikazani su u tablici 4.1. Stupci tablice predstavljaju indeks na čijim se podacima izvršavala analiza, a redci predstavljaju modele. Prvi redak prikazuje rezultate najjednostavnijeg modela koji pretpostavlja da svi povrati dolaze iz iste normalne distribucije i parametri te distribucije procjenjuju se na svim podacima za treniranje. Iduća tri retka prikazuju rezultate modela korištenih u ovom radu te "Model 1" predstavlja model koji za koristi potpuno povezane modele i za procjenu srednje vrijednosti i za procjenu varijance. "Model 2" je model koji koristi LSTM slojeve za procjenu srednje vrijednosti i varijance, a "Model 3" predstavlja model koji je kombinacija prva dva modela odnosno koristi LSTM slojeve za procjenu srednje vrijednosti te potpuno povezane slojeve za procjenu varijance. Kako se radi o logaritamskoj izglednosti,

veća vrijednost predstavlja bolji rezultat te je u svakom stupcu podebljan najbolji rezultat. Može se vidjeti kako su svi modeli korišteni u ovom radu pokazali bolje rezultate od referentnog modela te da u dva od tri slučaja najbolje rezultate daje "Model 3".

| Rezultati | | | |
|-------------------|----------------|----------------|---------------|
| | SPY | TLT | PDBC |
| Normalna razdioba | -1.4612 | -1.438 | -1.5378 |
| Model 1 | -1.3777 | -1.4333 | -1.3002 |
| Model 2 | -1.3225 | -1.3978 | -1.295 |
| Model 3 | -1.3164 | -1.3889 | -1.3139 |

Tablica 4.1. Logaritamske izglednosti modela na stvarnim podacima

5. Zaključak

5.1. Sumiranje rezultata

Ovaj rad se fokusirao na istraživanje i evaluaciju različitih modela za predviđanje povrata financijske imovine koristeći duboke generativne modele. U eksperimentalnom dijelu rada uspoređena su četiri modela. Referentni model korišten u ovom radu je normalna razdioba koja proizlazi iz često korištene pretpostavke da su povrati normalno distribuirani. Idući model koji je korišten su dvije potpuno povezane neuronske mreže koje služe za procjenu srednje vrijednosti i varijance. Treći model uveo je povratne veze u oba modela koristeći LSTM slojeve te tako pokušao uhvatiti vremensku zavisnost podataka. I zadnji model korišten je kombinacija drugog i trećeg modela te je koristio povratnu neuronsku mrežu za procjenu srednje vrijednosti te potpuno povezanu neuronsku mrežu za procjenu varijance.

Rezultati pokazuju da svi modeli korišteni u ovom radu daju bolje rezultate u odnosu na referentni model. Konkretno, model složen od povratne mreže i potpuno povezane mreže daje najbolje rezultate u dva od tri slučaja. Ovo sugerira da kombinacija LSTM slojeva za procjenu srednje vrijednosti i potpuno povezanih slojeva za procjenu varijance može bolje modelirati povrate financijske imovine. Ovi rezultati potvrđuju hipotezu da složenije arhitekture dubokog učenja mogu bolje modelirati kompleksne obrasce u financijskim podacima. Uz to, korištenjem LSTM slojeva model je uspio uhvatiti dinamičke promjene u vremenskim nizovima čime je poboljšana predikcija.

5.2. Preporuke za buduća istraživanja

Iako su ovi modeli dali bolje rezultate od referentnog modela, postoji nekoliko smjerova za buduća istraživanja. Buduća istraživanja trebala bi se usmjeriti na kombiniranje i

testiranje različitih arhitektura neuronskih mreža kao što su transformer mreže ili varijacijski autoenkoderi. Transformer mreže, poznate po svojoj sposobnosti da efikasno modeliraju sekvencijske podatke, mogle bi ponuditi prednosti u predikciji vremenskih nizova zbog svoje sposobnosti da hvataju dugoročne ovisnosti bolje od tradicionalnih LSTM mreža. Primjena varijacijskih autoenkodera mogla bi ponuditi dodatne uvide u strukturu podataka s obzirom na to da model daje uvid u naučenu latentnu reprezentaciju koja je sadržajnije informacijama te otpornija na šumove.

Analiza u ovom radu bila je ograničena na univarijatne vremenske nizove, što znači da je svaki modeliran povrat financijske imovine razmatran izolirani. Buduća istraživanja trebala bi istražiti proširenje analize na multivarijatne slučajeve, gdje se istovremeno modelira više imovina. Ovo bi omogućilo razumijevanje međusobnih odnosa između različitih financijskih instrumenata i moglo bi značajno unaprijediti preciznost predikcija u portfeljskim strategijama. Multivarijatni modeli mogu koristiti složenije strukture podataka koje uključuju zajedničke informacije iz više vremenskih nizova. Na primjer, korištenje zajedničkih informacijskih sadržaja među različitim imovinama može poboljšati modeliranje korelacija i razumijevanje zajedničkih tržišnih šokova. Ovo bi uključivalo primjenu naprednih tehnika poput Vector AutoRegression (VAR), multivarijatnih LSTM mreža, te korištenje korelacijskih i kovarijacijskih matrica za bolje hvatanje međuovisnosti među imovinama. Ova istraživanja trebala bi obuhvatiti i evaluaciju portfeljskih rizika i povrata, te analizu robusnosti modela u različitim tržišnim uvjetima. Implementacija takvih modela mogla bi značajno pridonijeti razvoju sofisticiranijih alata za upravljanje rizicima i donošenje investicijskih odluka.

Provedbom ovih preporuka, buduća istraživanja mogu dodatno unaprijediti sposobnost modela za točno predviđanje povrata financijske imovine te pridonijeti razvoju pouzdanih alata za donošenje investicijskih odluka.

Literatura

- [1] J. Y. Campbell, A. W. Lo, i A. MacKinlay, *The Econometrics of Financial Markets*. Princeton University Press, 1997. [Mrežno]. Adresa: <http://www.jstor.org/stable/j.ctt7skm5>
- [2] R. F. Engle, “Autoregressive conditional heteroscedasticity with estimates of the variance of united kingdom inflation”, *Econometrica*, sv. 50, br. 4, str. 987–1007, 1982. [Mrežno]. Adresa: <http://www.jstor.org/stable/1912773>
- [3] G. Box i G. Jenkins, *Time Series Analysis: Forecasting and Control*, ser. Holden-Day series in time series analysis and digital processing. Holden-Day, 1976. [Mrežno]. Adresa: <https://books.google.hr/books?id=1WVHAAAAMAAJ>
- [4] S. Borovkova i H. R. M. K. Raadsveld, *Time Series Analysis and Forecasting by Example*. Boca Raton, FL: CRC Press, 2019.
- [5] I. Goodfellow, Y. Bengio, i A. Courville, *Deep Learning*. MIT Press, 2016., <http://www.deeplearningbook.org>.
- [6] R. Cont, “Empirical properties of asset returns: Stylized facts and statistical issues”, *Quantitative Finance*, sv. 1, br. 2, str. 223–236, 2001. <https://doi.org/10.1080/713665670>
- [7] R. S. Tsay, *Analysis of Financial Time Series*, 2. izd., ser. Wiley Series in Probability and Statistics. Hoboken, NJ: Wiley-Interscience, 2005., includes bibliographical references and index.
- [8] Y. A. LeCun, L. Bottou, G. B. Orr, i K.-R. Müller, *Efficient BackProp*, G. Montavon, G. B. Orr, i K.-R. Müller, Ur. Berlin, Heidelberg: Springer Berlin Heidelberg, 2012.

https://doi.org/10.1007/978-3-642-35289-8_3

Sažetak

Duboki generativni modeli u vrednovanju financijske imovine

Karlo Šutalo

U radu se istražuju duboki generativni modeli i njihova primjena u vrednovanju financijske imovine. Korištenjem različitih metoda, kao što su autoregresivni modeli, modeli autoregresivne uvjetne heteroskedastičnosti (ARCH) i neuronske mreže, analiziraju se financijski vremenski nizovi kako bi se predvidjele buduće vrijednosti imovine. Poseban naglasak stavljen je na generiranje sintetičkih podataka i modeliranje povrata korištenjem neuronskih mreža. Evaluacija performansi modela pokazuje njihovu učinkovitost u predviđanju kretanja cijena i volatilnosti na financijskim tržištima.

Ključne riječi: financijski vremenski nizovi, duboki generativni modeli, neuronske mreže, autoregresivni modeli, volatilnost, sintetički podaci

Abstract

Deep generative models in asset pricing

Karlo Šutalo

This thesis explores deep generative models and their application in asset pricing. Financial time series are analysed using various methods, such as autoregressive model, autoregressive conditional heteroskedasticity (ARCH) models, and neural networks to predict future asset values. Special emphasis is placed on the generation of synthetic data and the modelling of returns using neural networks. The performance evaluation of the models demonstrates their effectiveness in forecasting price movements and volatility in financial markets.

Keywords: financial time series, deep generative models, neural networks, autoregressive models, volatility, synthetic data