

# Neuronske mreže za klasifikaciju glazbenog žanra

---

**Prolić, Ante**

**Undergraduate thesis / Završni 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://um.nsk.hr/um:nbn:hr:168:919376>

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

*Download date / Datum preuzimanja:* **2025-03-21**



*Repository / Repozitorij:*

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



SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1247

**NEURONSKE MREŽE ZA KLASIFIKACIJU GLAZBENOG  
ŽANRA**

Ante Prolić

Zagreb, lipanj 2024.

SVEUČILIŠTE U ZAGREBU  
FAKULTET ELEKTROTEHNIKE I RAČUNARSTVA

ZAVRŠNI RAD br. 1247

**NEURONSKE MREŽE ZA KLASIFIKACIJU GLAZBENOG  
ŽANRA**

Ante Prolić

Zagreb, lipanj 2024.

## ZAVRŠNI ZADATAK br. 1247

Pristupnik: **Ante Prolić (0036544148)**  
Studij: Elektrotehnika i informacijska tehnologija i Računarstvo  
Modul: Računarstvo  
Mentor: prof. dr. sc. Ivan Đurek

Zadatak: **Neuronske mreže za klasifikaciju glazbenog žanra**

### Opis zadatka:

Teoretski obradite neuronske mreže za klasifikaciju audio signala. Obradite najvažnije značajke audio signala koje se koriste za kreiranje neuronskih mreža. Na dostupnom setu audio zapisa, kreirajte neuronsku mrežu za klasifikaciju glazbenih žanrova. Analizirajte parametre neuronske mreže kako bi dobili najtočniju klasifikaciju.

Rok za predaju rada: 14. lipnja 2024.



1. UVOD.....	4
2. NEURONSKE MREŽE .....	5
3. ZVUK I PROCESUIRANJE AUDIO PODATAKA .....	9
4. KONVOLUCIJSKE NEURONSKE MREŽE .....	12
5. IMPLEMENTACIJA.....	14
5.1. BAZA PODATAKA ZA TESTIRANJE .....	14
5.2. ARHITEKTURA NEURONSKE MREŽE.....	16
5.3. TRENIRANJE I TESTIRANJE SLOJEVITE NEURONSKE MREŽE .....	20
5.4. ARHITEKTURA KONVOLUCIJSKE NEURONSKE MREŽE .....	21
5.5. TRENIRANJE I TESTIRANJE KONVOLUCIJSKE NEURONSKE MREŽE .....	25
6. ZAKLJUČAK.....	26
7. LITERATURA.....	27
8. SAŽETAK.....	28

# 1. UVOD

Tema ovog završnog rada su neuronske mreže za klasifikaciju glazbenog žanra. Cilj rada je istražiti što su to neuronske mreže, kako pripremiti i analizirati podatke potrebne za treniranje i testiranje neuronske mreže te na posljetku odrediti arhitekturu neuronske mreže koja će nam što preciznije odrediti glazbeni žanr pjesme.

Klasifikacija glazbenih žanrova odnosi se na automatsko prepoznavanje i kategorizaciju glazbe prema njenom žanru koristeći računalne algoritme. Duboko učenje, posebice konvolucijske neuronske mreže (CNN), pokazale su se izuzetno učinkovitima za zadatke obrade zvuka i slike. Korištenjem CNN-ova, možemo izvući značajke iz spektrograma, vizualnih prikaza zvučnih zapisa, kako bismo prepoznali karakteristike specifične za određene glazbene žanrove.

Jedan od ključnih izazova u ovoj oblasti je raznolikost i složenost glazbenih žanrova, koji često imaju slične karakteristike. Na primjer, podžanrovi mogu dijeliti mnoge sličnosti s glavnim žanrom, što otežava preciznu klasifikaciju. Način na koji se obrađuju podaci iz audio datoteke je također jako bitan i u organizaciji ulaza neuronskih mreža, igra dosta veliku ulogu u brzini i efikasnosti treniranja i testiranja istih.

Osim ulaznih podataka, točnost algoritma ovisi o aktivacijskim funkcijama neurona, slojevima težina, broju skrivenih slojeva u umjetnoj neuronskoj mreži i još mnogim drugim parametrima. Te parametre treba optimizirati i rezultate uspoređivati s drugim modelima kako bi se ponajviše unaprijedila preciznost određivanja žanra kojemu pjesma pripada.

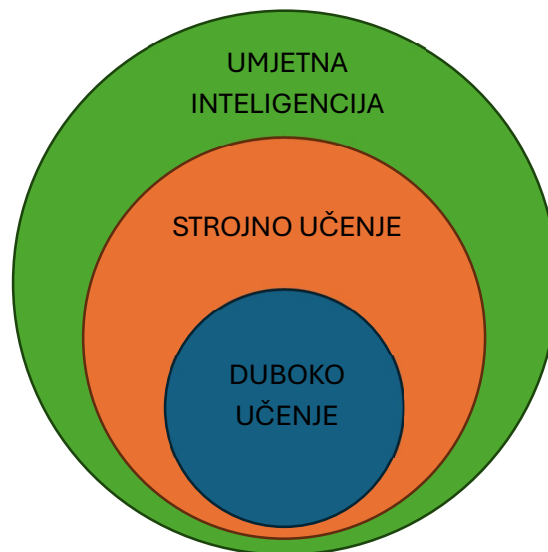
U ovom radu pokazat ću kako (i s kojim alatima) sam napisao i istrenirao konvolucijsku neuronsku mrežu čija je svrha raspoznavanje određenih glazbenih žanrova.

## 2. NEURONSKE MREŽE

“Umjetna inteligencija grana je računarske znanosti koja se bavi proučavanjem i oblikovanjem računarskih sustava koji pokazuju neki oblik inteligencije. Takvi sustavi mogu učiti, mogu donositi zaključke o svijetu koji ih okružuje, oni razumiju prirodni jezik te mogu spoznati i tumačiti složene vizualne scene te obavljati druge vrste vještina za koje se zahtijeva čovjekov tip inteligencije.”

- D. W. Patterson

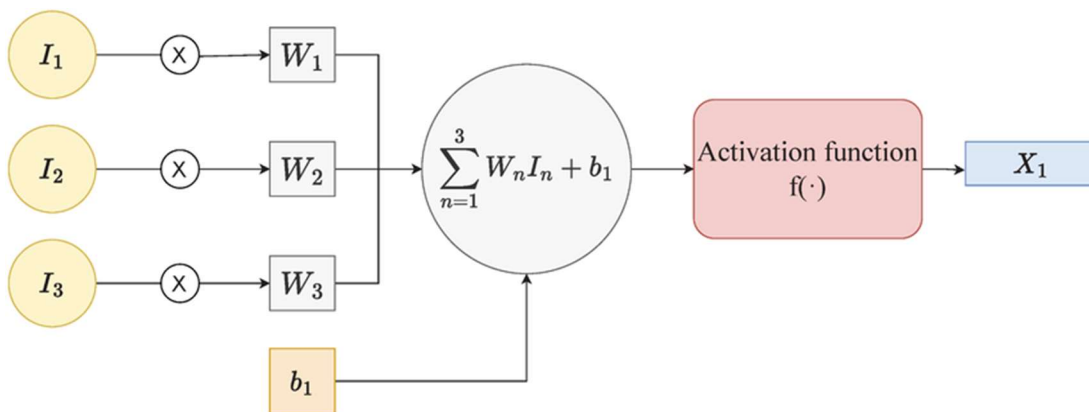
Strojno učenje je jedno od područja proučavanja umjetne inteligencije koje se bavi razvojem i proučavanjem algoritama čija je svrha učenje i generalizacija te korištenje istih za izvršavanje zadataka bez zadavanja jasnih uputa. U to područje svrstane su i neuronske mreže.



Umjetna neuronska mreža je skup međusobno povezanih neurona uređenih po uzoru na ljudski mozak. Omogućava obradu masivne količine podataka te koristi se za probleme klasifikacije i regresije.

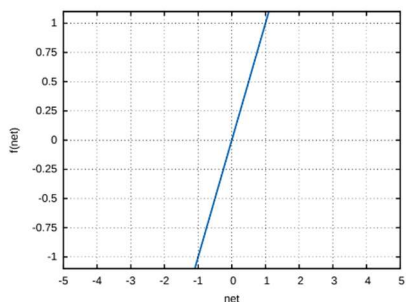


Glavna jedinka na kojoj se temelje neuronske mreže je umjetni neuron. Na temelju analize bioloških neurona u mozgu, McCulloch-Pitts je 1943. definirao prvi umjetni neuron zvan TLU-perceptron (Treshold Logic Unit). Sastoji se od 2 skupa: ulaza i osjetljivosti ulaza, a kad se ta dva skupa pomnože i sumiraju, ta se vrijednost propušta kroz prijenosnu funkciju. Time nastaje izlazna vrijednost neurona.

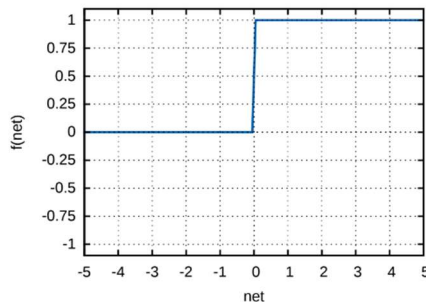


Slika 2.1. – Neuron TLU-perceptron

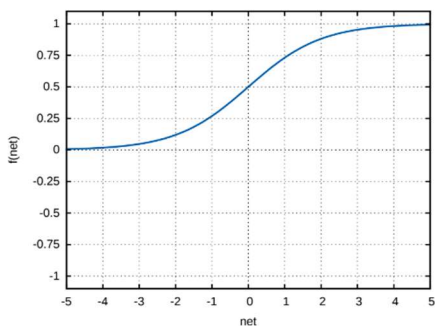
Svi umjetni neuroni ne koriste iste prijenosne funkcije, pa se ovisno o problemu i tipu ulaznih podataka bira ona funkcija koja osigurava da neuronska mreža što brže i točnije odradi zadatak za koji je istrenirana. Primjeri takvih funkcija:



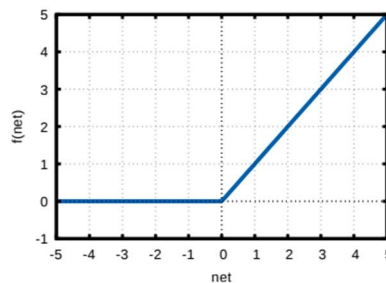
Slika 2.2. – Funkcija identiteta



Slika 2.3. – Funkcija skoka



Slika 2.4. – Sigmoidalna funkcija

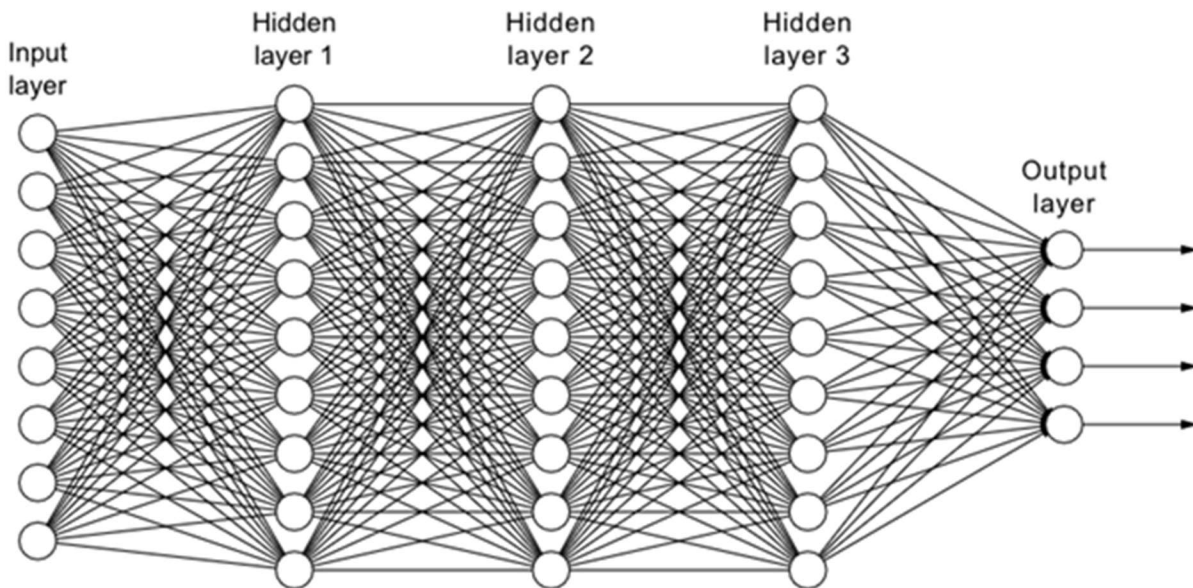


Slika 2.5. - Zglobnica

Klasifikacije dijelimo na binarne i one u više razreda. Binarna klasifikacija služi za razlučivanje 2 stanja (klase), dok klasifikacija u više stanja omogućava razlučivanje više od 2 stanja. Za slučaj klasifikacije glazbenog žanra najviše ima smisla koristiti onu u više stanja.

Kako bi dobili korisnu neuronsku mrežu koja obavlja onaj zadatak koji smo joj odredili, prvo se mora osmisliti njena arhitektura (tj. odrediti kako će neuroni biti spojeni, koliko ih ima, u koliko će se slojeva podijeliti itd.)

Slojevite neuronske mreže imaju 3 grupe slojeva: ulazni sloj (input layer), skrivene slojeve (hidden layers) i izlazni sloj (output layer). Svaki sloj može imati proizvoljan broj neurona, a ulazi neurona iz jednog sloja (osim ulaznog sloja) spojeni su na izlaze umjetnih neurona iz prethodnih slojeva.



Slika 2.6. – Arhitektura neuronske mreže

Kada se ostvari željena arhitektura umjetne neuronske mreže i podese težine (postavljaju se na slučajne vrijednosti), sljedeći korak do ostvarenja željenog cilja je treniranje (učenje) neuronske mreže. Treniranje se izvodi podešavanjem osjetljivosti ulaza neurona (težina) u ovisnosti o takozvanoj funkciji pogreške.

Funkcija pogreške služi kao indikator koliko se izlaz dosad istreniranog modela razlikuje od željenog ishoda. Najčešća funkcija koja se koristi u području je polovična suma srednjih kvadratnih odstupanja (1):

$$E = \frac{1}{2} \sum_{s=1}^N E(s) = \frac{1}{2} \sum_{s=1}^N \frac{1}{N} \sum_{i=1}^{N_o} (t_{s,i} - o_{s,i})^2 \quad (1)$$

Slika 2.7 – Polovična suma srednjih kvadratnih odstupanja

Jednom kad se odredi funkcija pogreške, koristi se postupak propagacije pogreške unatrag na temelju izračuna parcijalnih derivacija funkcije pogreške s obzirom na svaku težinu i prag (3):

$$\delta_j^{k+1} = o_{s,j} \cdot (1 - o_{s,j}) \cdot (t_{s,j} - o_{s,j}) \quad (2)$$

$$\delta_j^{(k)} = y_j^{(k)} \cdot (1 - y_j^{(k)}) \cdot (w_{j,1}^{(k)} \cdot \delta_1^{(k+1)} + \dots + w_{j,m}^{(k)} \cdot \delta_m^{(k+1)}) \quad (3)$$

Taj izračun nam pokazuje koliko će se rezultat neuronske mreže promijeniti ako modificiramo vrijednosti pojedinih težina. Postupak podešavanja težina ponavlja se dok neuronska mreža ne ostvari određenu točnost ili pogrešku (može se ograničiti i brojem epoha).

### 3. ZVUK I PROCESUIRANJE AUDIO PODATAKA

U fizici zvuk je definiran kao mehanički val u rasponu frekvencija od 20 do 20000 Hz. Širi se kroz prijenosne medije poput plina, tekućina ili krutina.

Jednostavan zvučni val može se prikazati jednostavnom sinusnom funkcijom (4).

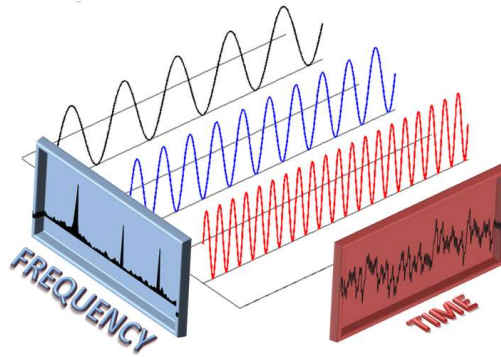
$$y(t) = A \sin(2\pi f t + \varphi) \quad (4)$$

gdje je:

- $t$  - vrijeme
- $A$  – amplituda
- $f$  – frekvencija
- $\varphi$  – faza

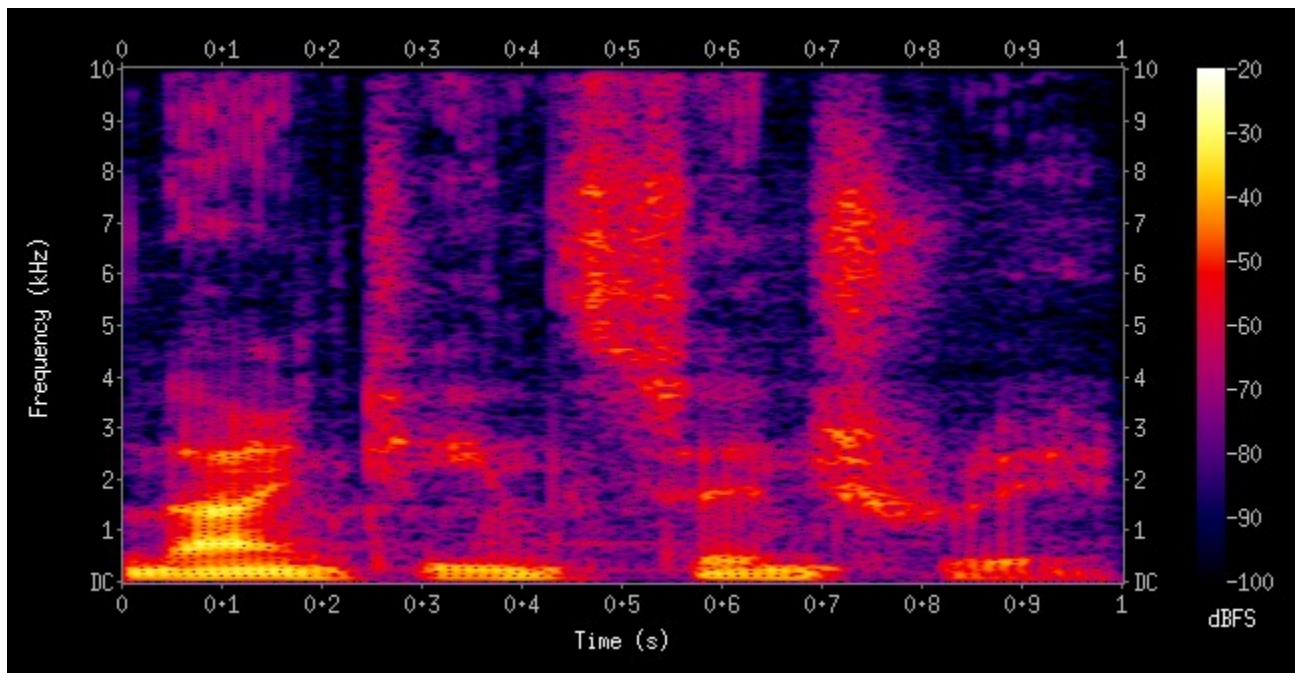
Kako bi analizirali zvuk za potrebe neuronskih mreža moramo napraviti analogno digitalnu pretvorbu. Zvučni signal se uzorkuje u uniformnim vremenskim intervalima (diskretizacija po vremenu) i amplituda se kvantizira s određenim brojem bitova. Frekvencija uzorkovanja najčešće se postavlja na 44.1 kHz dok je kvantizacija 16 bitna.

Osim A/D pretvorbe potrebna nam je i brza Fourierova transformacija (FFT) za pretvaranje valnog oblika u frekvencijski spektar. Koristi nam za rastavljanje složenog periodičkog zvuka u zbroj sinusnih valova koji osciliraju na različitim frekvencijama.



Slika 3.1. – Brza Fourierova transformacija

Ako koristimo kratkotrajnu Fourierovu transformaciju za određivanje sinusne frekvencije i faznog sadržaja lokalnih dijelova signala kako se on mijenja tijekom vremena, stvara se spektrogram:

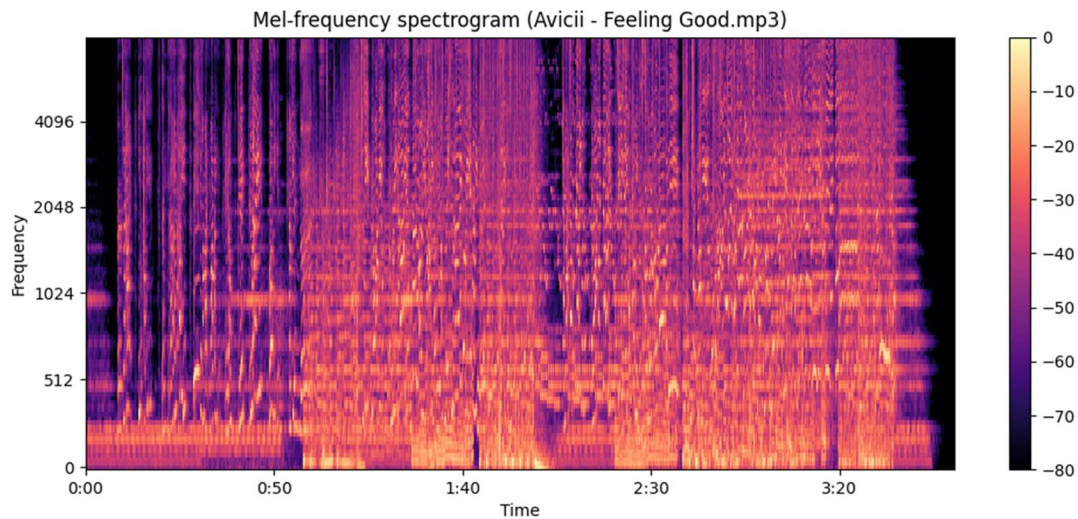


Slika 3.2. – Spektrogram nastao kratkotrajnom Fourierovom transformacijom

Na x osi je prikazano vrijeme, na y osi su prikazane frekvencije, a bojom je označena jačina zvuka u decibelima.

Nekada zvuk audio datoteke ima lošu kvalitetu ili se pojavljuju nekakvi šumovi u reprodukciji zvučnih signala. Kako bi se riješili takvih smetnji, koriste se mel-frekvencijski

kepralni koeficijent (MFCC). MFCC se temelji na ljudskom slušnom sustavu, pokušavajući oponašati način na koji ljudsko uho percipira zvuk. Najčešće se u industriji koristi za prepoznavanje govora, klasifikaciju žanrova, instrumenata itd.



Slika 3.3. – Primjer MEL-spektrograma

```
S = librosa.feature.melspectrogram(y=signal, sr=sample_rate, n_mels=128, fmax=8000)
S_dB = librosa.power_to_db(S, ref=np.max)

plt.figure().set_figwidth(12)
librosa.display.specshow(S_dB, x_axis="time", y_axis="mel", sr=sample_rate, fmax=8000)
plt.colorbar()

plt.title('Mel-frequency spectrogram ({}).format(sys.argv[1]))
plt.xlabel('Time')
plt.ylabel('Frequency')
plt.show()
```

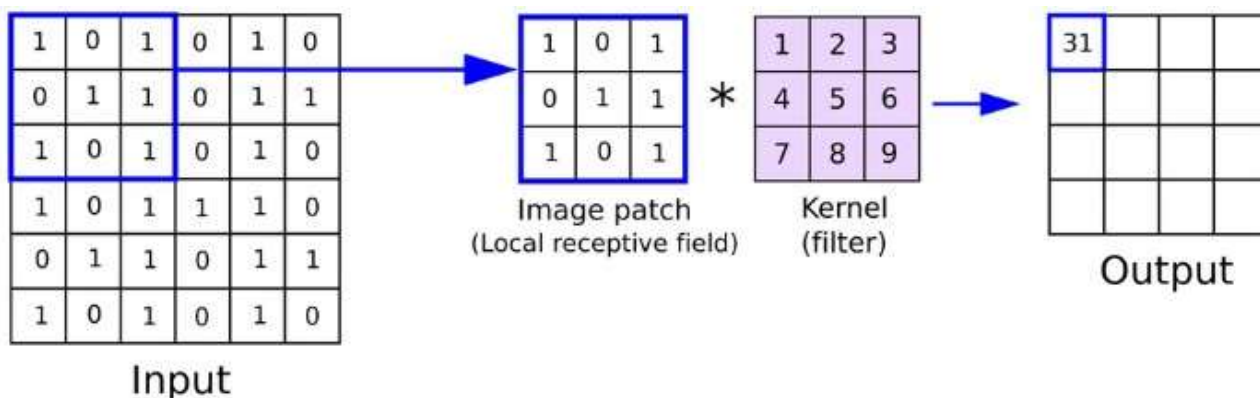
Slika 3.4. – Kod za stvaranje MEL-spektrograma

## 4. KONVOLUCIJSKE NEURONSKE MREŽE

Konvolucijske neuronske mreže (CNN, od engleskog "Convolutional Neural Networks") su tip umjetne neuronske mreže posebno dizajnirane za obradu strukturiranih rešetki podataka poput slika. CNN-ovi su vrlo učinkoviti za zadatke poput prepoznavanja slika, klasifikacije, segmentacije i prepoznavanja objekata. Varijante su višeslojnih perceptrona, dizajniranih da oponašaju ponašanje vidnog korteksa.

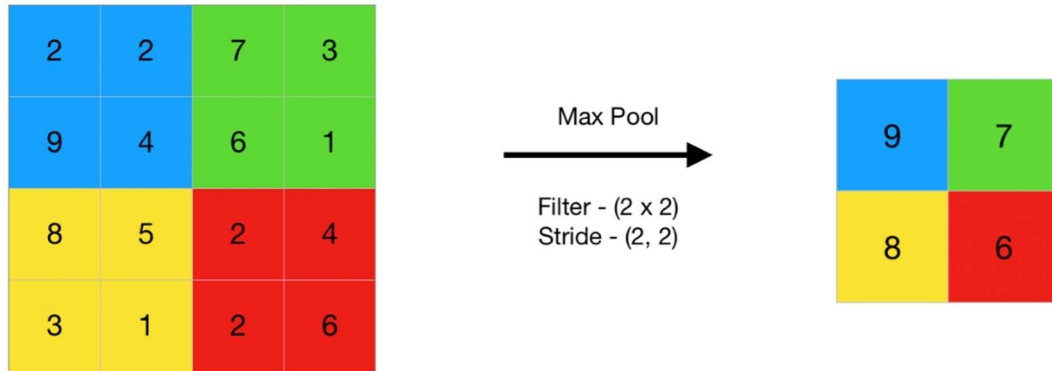
Osnovna 2 sloja u CNN-ovima su konvolucijski i *pooling* sloj:

- Konvolucijski sloj sadrži filtere koji se primjenjuju na ulazne podatke. Filter je matrica čiji su podaci težine i on ovisno o određenom pomaku konvoluirao preko ulazne slike i računa skalarni produkt između dijela slike na kojem se nalazi filter i same matrice filtra. Tijekom konvolucije koristi se *padding* za dodavanje nula oko granica slike kako bi se mogao izračunati skalarni produkt na rubovima. Filteri mogu imati i dubine, tj. više različitih setova vrijednosti za istu matricu (npr. za slike u boji, filter ima dubinu 3 za RGB).

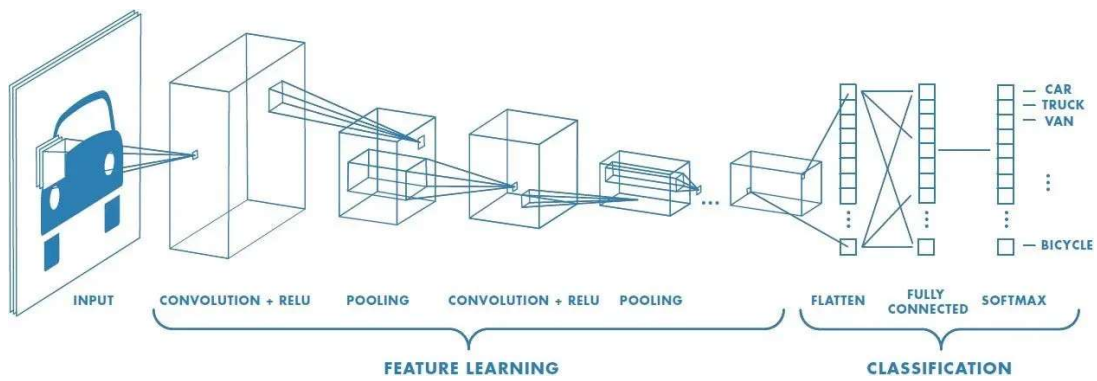


Slika 4.3. – Funkcionalnost jednog filtera

- *Pool* sloj smanjuje dimenzionalnost mape značajki uzimanjem maksimalne vrijednosti unutar unaprijed definiranog prozora (eng. *max pooling*). Ovim postupkom se smanjuje broj parametara i kontrolira prenaučenost.



Slika 4.2. – Funkcionalnost pool solja

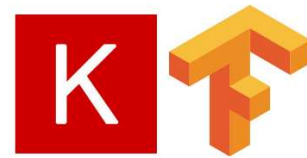


Slika 4.3. – Primjer arhitekture konvolucijske neuronske mreže



## 5. IMPLEMENTACIJA

Projekt je u potpunosti napisan u programskom jeziku Python. Za implementaciju i testiranje funkcionalnih umjetnih neuronskih mreža korištena je biblioteka TensorFlow iz koje je uvezen Keras.



### 5.1. BAZA PODATAKA ZA TESTIRANJE

Za treniranje i testiranje korišten je [online GTZAN Dataset](#) koji sadrži 1000 audio datoteka dugih 30 sekundi, podijeljene u 10 različitih žanrova.

Sljedeći korak je procesuiranje podataka kako bi se mogli koristiti na ulazu neuronske mreže.

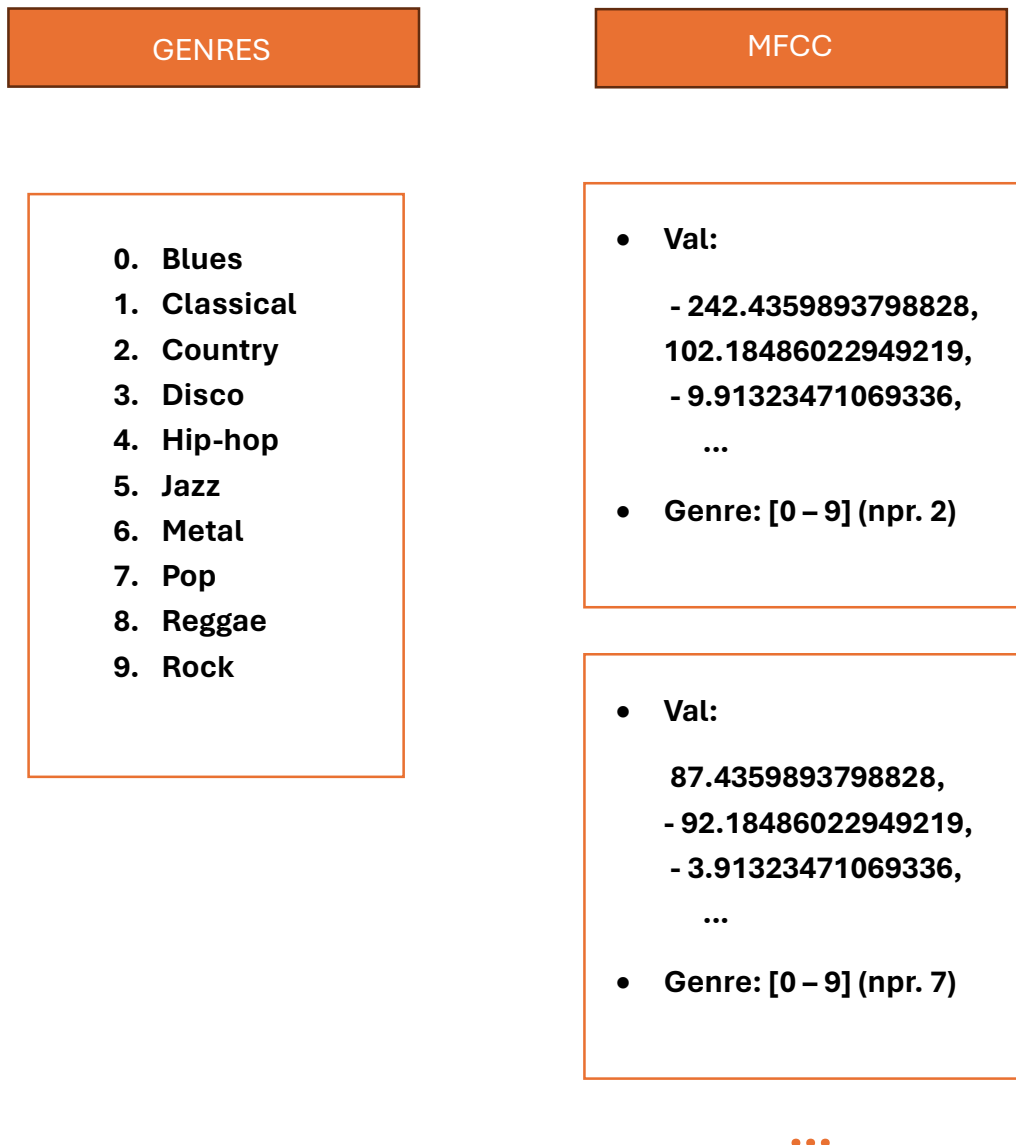
Datoteke su spremljene na način da se svaka nalazi u mapi nazvanoj prema žanru kojoj ta pjesma pripada. Kod u svakoj od tih mapa analizira datoteke i audio podatke dijeli na 10 segmenata, frekvencijom uzorkovanja od 22050 Hz. Svaki od tih segmenata koristimo za stvaranje MFCC-eva i njih spremamo u obliku Python rječnika s ključevima „val“ (value, spremamo niz MFCC-eva) i „genre“ (spremamo pripadajući žanr, ime direktorija u kojoj se nalazi datoteka).

MFCC-evi nastaju pozivom funkcije „mfcc“ iz biblioteke Librosa:

```
mfcc = librosa.feature.mfcc(y=partition, sr=sample_rate, n_mfcc=13, n_fft=2048, hop_length=512).T
```

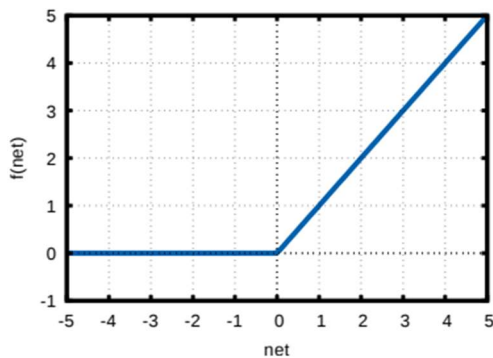
Funkcija prima segment koji se analizira, frekvenciju uzorkovanja (22050 Hz), broj MFCC-eva koji se vraćaju, duljinu prozora Fourierove brze transformacije i broj uzoraka između uzastopnih okvira.

Python rječnici su spremljeni u rječnik pod ključem „mfcc“, skupa s nizom svih žanrova pod ključem „genres“, koji je kasnije spremljen u JSON formatu pod nazivom „dataset.json“. Taj JSON služi kao baza podataka za treniranje i testiranje neuronske mreže.



## 5.2. ARHITEKTURA NEURONSKE MREŽE

Nakon obrade audio podataka, vrijeme je za sastavljanje arhitekture neuronske mreže. Mreža ima 1 ulazni, 1 izlazni i 3 skrivena sloja. Skriveni slojevi redom imaju 512, 256 i 64 neurona. Za skrivene slojeve koristio sam prijenosnu funkciju zglobnica (Rectified Linear Unit, ReLU) (5) koja je u praksi najefikasnija.



Slika 5.2.1. – Graf funkcije ReLU

$$f(\text{net}) = \max(0, \text{net}) \quad (5)$$

```
keras.layers.Dense(512, activation="relu", kernel_regularizer=keras.regularizers.l2(0.001)),
```

Slika 5.2.2. – Isječak koda gdje je funkcija ReLU prosljeđena kao parametar

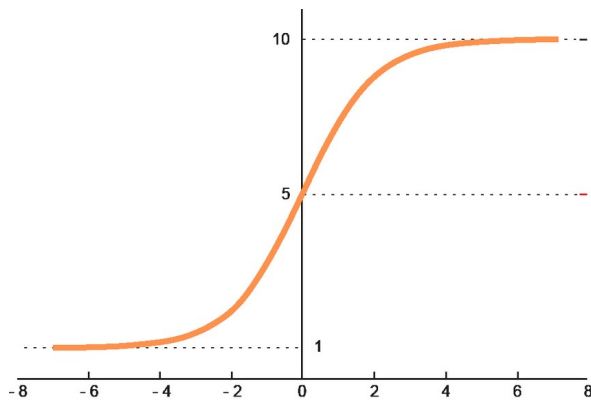
Stopa učenja mreže postavljena je na vrijednost 0.0001, a za pogrešku se koristila funkcija „sparse categorical cross entropy“ (6).

$$\text{Loss} = - \sum_{i=1}^{\text{output size}} y_i \cdot \log \hat{y}_i \quad (6)$$

```
model.compile(optimizer=optimizer, loss="sparse_categorical_crossentropy", metrics=["accuracy"])
```

Slika 5.2.3. – Isječak koda gdje je funkcija „sparse categorical cross entropy“ prosljeđena kao parametar

Izlazni sloj neurona sadrži koristi prijenosnu funkciju „softmax“(7).



Slika 5.2.4. – Graf funkcije „softmax“

$$\sigma(\mathbf{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}} . \quad (7)$$

```
keras.layers.Dense(len(self.genres), activation="softmax")
```

Slika 5.2.5. – Isječak koda gdje je funkcija „softmax“ prosljeđena kao parametar

Kako bi se smanjila prenaučenosť, koristio sam Keras metodu Dropout. Dropout nasumično postavlja ulazne jedinice na 0 s određenom učestalosti na svakom koraku tijekom treniranja modela. U ovom slučaju učestalost je stavljena na 0.3:

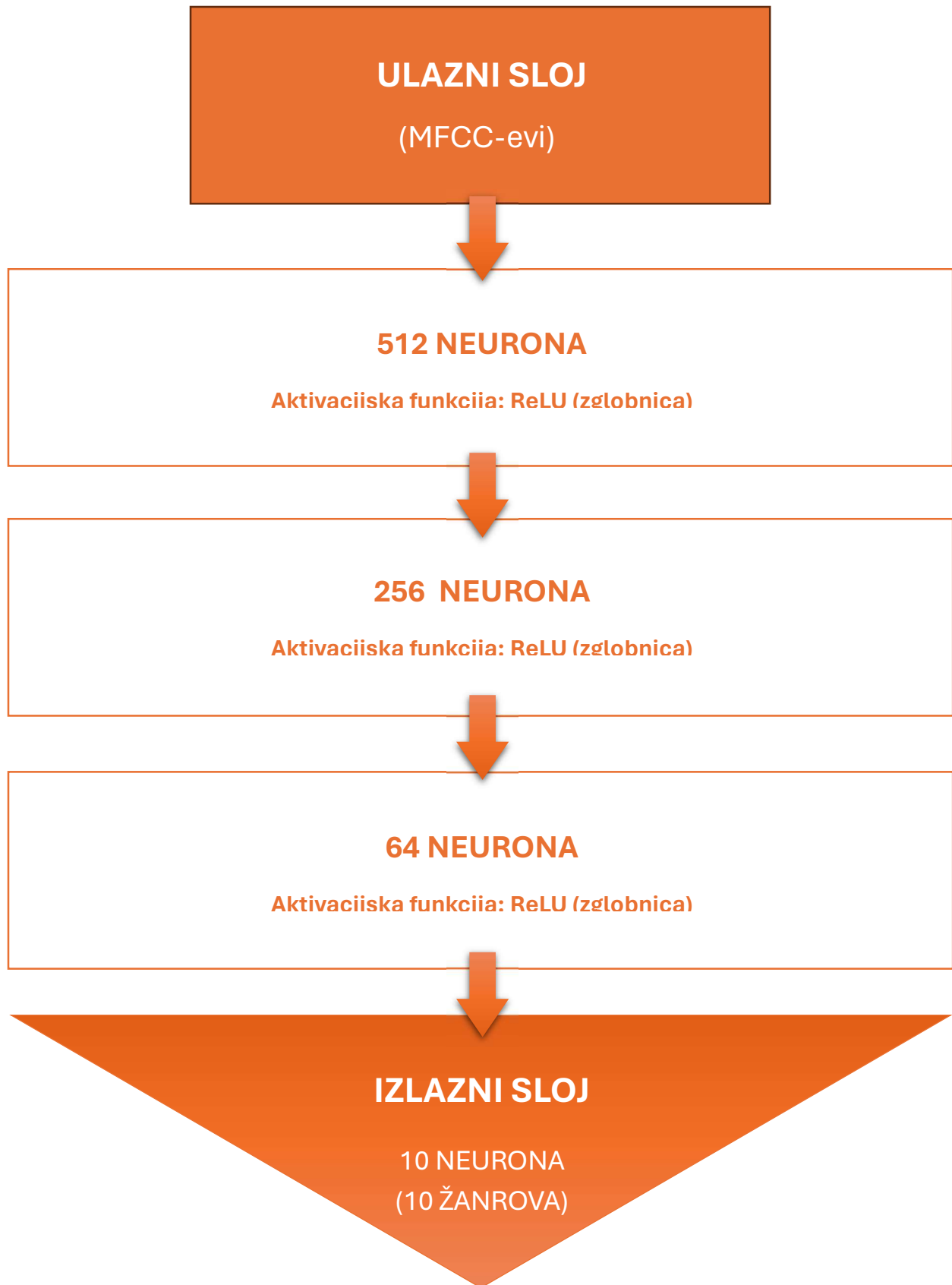
```
keras.layers.Dense(32, activation='relu'),  
keras.layers.Dropout(0.3),
```

Konačno model izgleda ovako:

```
layers = [  
    keras.layers.Flatten(input_shape=(data_inputs.shape[1], data_inputs.shape[2])),  
    keras.layers.Dense(512, activation="relu", kernel_regularizer=keras.regularizers.l2(0.001)),  
    keras.layers.Dropout(0.3),  
    keras.layers.Dense(256, activation="relu", kernel_regularizer=keras.regularizers.l2(0.001)),  
    keras.layers.Dropout(0.3),  
    keras.layers.Dense(64, activation="relu", kernel_regularizer=keras.regularizers.l2(0.001)),  
    keras.layers.Dropout(0.3),  
    keras.layers.Dense(len(self.genres), activation="softmax")  
]  
  
model = keras.Sequential(layers)  
  
optimizer = keras.optimizers.Adam(learning_rate=0.0001)  
model.compile(optimizer=optimizer, loss="sparse_categorical_crossentropy", metrics=["accuracy"])  
  
model.summary()  
  
history = model.fit(train_inputs, train_outputs, validation_data=(test_inputs, test_outputs), batch_size=32, epochs=70)
```

Slika 5.2.6. – Kod slojeva u slojevitoj neuronskoj mreži

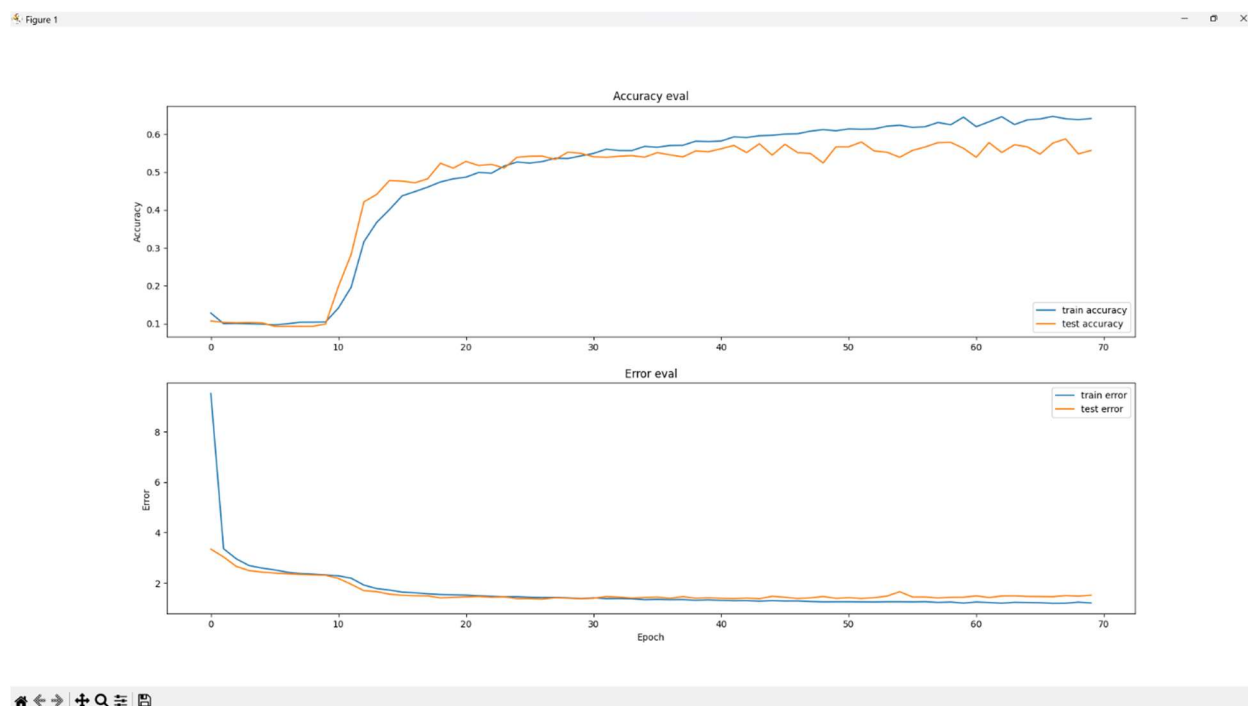
Model se trenira u 70 epoha, a „batch\_size“ što je veći, to više smanjuje količinu ulaznih podataka kako bi se ubrzalo treniranje i smanjila prenaučenosť.



### 5.3. TRENIRANJE I TESTIRANJE SLOJEVITE NEURONSKE MREŽE

Kada je „batch\_size“ postavljen na 1, svaka epoha u slučaju GTZAN baze podataka obrađuje po 6990 uzoraka. Mreža je trenirana u 70 epoha, a 1 epoha prosječno je trajala po 80 sekundi. Treniranje je završilo nakon otprilike sat i 33 minute, a preciznost modela je dosegla maksimalno 60 %.

```
Epoch 63/70  
6990/6990 ————— 81s 12ms/step - accuracy: 0.6501 - loss: 1.1861 - val_accuracy: 0.5517 - val_loss: 1.4898  
Epoch 64/70  
6990/6990 ————— 81s 12ms/step - accuracy: 0.6268 - loss: 1.2308 - val_accuracy: 0.5721 - val_loss: 1.4937  
Epoch 65/70  
6990/6990 ————— 82s 12ms/step - accuracy: 0.6418 - loss: 1.2168 - val_accuracy: 0.5668 - val_loss: 1.4723  
Epoch 66/70  
6990/6990 ————— 83s 12ms/step - accuracy: 0.6313 - loss: 1.2202 - val_accuracy: 0.5474 - val_loss: 1.4671  
Epoch 67/70  
6990/6990 ————— 80s 11ms/step - accuracy: 0.6489 - loss: 1.1800 - val_accuracy: 0.5764 - val_loss: 1.4607  
Epoch 68/70  
6990/6990 ————— 81s 12ms/step - accuracy: 0.6440 - loss: 1.1910 - val_accuracy: 0.5878 - val_loss: 1.5046  
Epoch 69/70  
6990/6990 ————— 77s 11ms/step - accuracy: 0.6339 - loss: 1.2568 - val_accuracy: 0.5481 - val_loss: 1.4828  
Epoch 70/70  
6990/6990 ————— 79s 11ms/step - accuracy: 0.6451 - loss: 1.1972 - val_accuracy: 0.5571 - val_loss: 1.5204
```



Slika 5.3.1. Graf točnosti i greške u ovisnosti o epohama tijekom treniranja neuronske mreže

Točnost je već oko 30. epohe na testnoj grupi podataka krenula stagnirati.

Vremenska neučinkovitost i nedovoljna točnost potakla me da implementiram drukčiji pristup rješenju: konvolucijske neuronske mreže.

## 5.4. ARHITEKTURA KONVOLUCIJSKE NEURONSKE MREŽE

Implementirana konvolucijska mreže sastoji se od 3 konvolucijska sloja, sa zglobnicom kao aktivacijskom funkcijom, čiji filtri imaju dimenzije redom: 5x5, 3x3 i 2x2, 3 „max pooling“ čiji su ulazi spojeni s izlazima konvolucijskih slojeva (svaki konvolucijski sloj ima 1 vlastiti „pooling“ sloj koji slijedi iza njega).

Slojevi za normalizaciju (eng. batch normalization) služe kako bi se stabilizirao i ubrzao proces treniranja modela. Normaliziraju ulaze za svaki sloj.

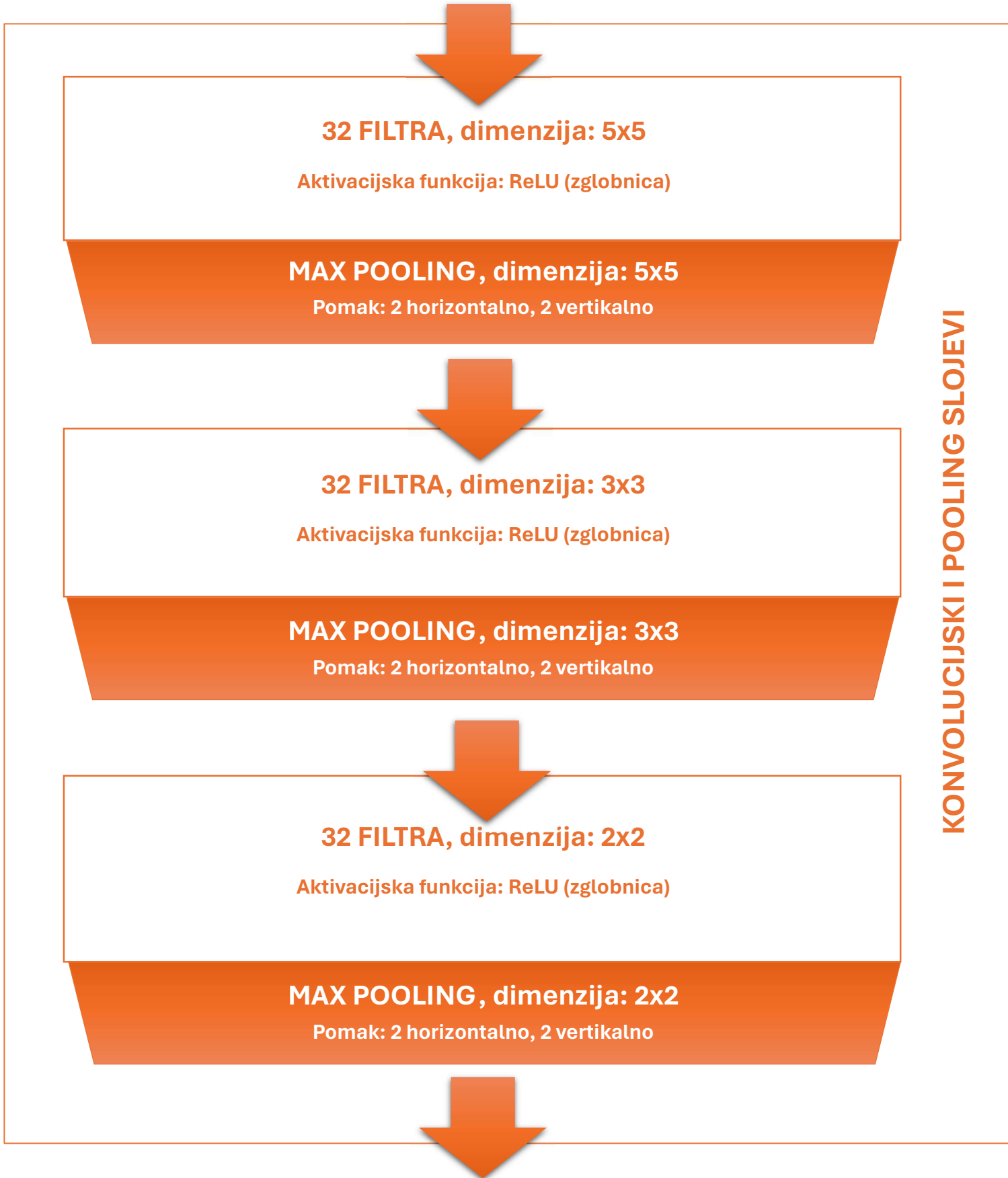
```
layers = [  
    keras.layers.Conv2D(32, (5, 5), activation='relu', input_shape=(train_inputs.shape[1], train_inputs.shape[2], 1)),  
    keras.layers.MaxPooling2D((5, 5), strides=(2, 2), padding='same'),  
    keras.layers.BatchNormalization(),  
  
    keras.layers.Conv2D(32, (3, 3), activation='relu'),  
    keras.layers.MaxPooling2D((3, 3), strides=(2, 2), padding='same'),  
    keras.layers.BatchNormalization(),  
  
    keras.layers.Conv2D(32, (2, 2), activation='relu'),  
    keras.layers.MaxPooling2D((2, 2), strides=(2, 2), padding='same'),  
    keras.layers.BatchNormalization(),  
  
    keras.layers.Flatten(),  
    keras.layers.Dense(128, activation='relu'),  
    keras.layers.Dropout(0.3),  
  
    keras.layers.Dense(64, activation='relu'),  
    keras.layers.Dropout(0.3),  
  
    keras.layers.Dense(32, activation='relu'),  
    keras.layers.Dropout(0.3),  
  
    keras.layers.Dense(len(self.genres), activation='softmax')  
]  
  
model = keras.Sequential(layers)  
model.compile(optimizer=keras.optimizers.Adam(learning_rate=0.0001), loss='sparse_categorical_crossentropy', metrics=['accuracy'])  
model.summary()
```

Slika 5.4.1. – Kod slojeva u konvolucijskoj neuronskoj mreži



Nakon toga su još implementirana 3 skrivena sloja čiji su brojevi neurona redom: 128, 64 i 32 te svaki sloj također koristi aktivacijsku funkciju ReLU (zglobnica). Izlazni sloj, slično prijašnje opisanoj slojevitoj neuronskoj mreži, ima aktivacijsku funkciju „softmax“. Stopa učenja je također 0.0001, sa „sparse cross entropy“ funkcijom postavljenom za računanje gubitka.





## 5.5. TRENIRANJE I TESTIRANJE KONVOLUCIJSKE NEURONSKE MREŽE

Za preciznije oblikovanje podatke sam podijelio u 3 grupe: grupa za treniranje, provjeru i testiranje. Grupa za provjeru služi kako bi se ustanovilo koji istrenirani model ima najveću točnost. S veličinom „batch\_size“ postavljenom na 8 i brojem epoha smanjenim na 50, model konvolucijske mreže imao je točnost vrijednosti otprilike 70 %, a treniranje je završilo u otprilike 4 minute i 10 sekundi:

```
Epoch 47/50  
525/525 ————— 5s 10ms/step - accuracy: 0.7073 - loss: 0.8687 - val_accuracy: 0.6946 - val_loss: 0.9223  
Epoch 48/50  
525/525 ————— 5s 10ms/step - accuracy: 0.7111 - loss: 0.8697 - val_accuracy: 0.6910 - val_loss: 0.9085  
Epoch 49/50  
525/525 ————— 5s 10ms/step - accuracy: 0.7146 - loss: 0.8592 - val_accuracy: 0.6906 - val_loss: 0.9270  
Epoch 50/50  
525/525 ————— 5s 10ms/step - accuracy: 0.7112 - loss: 0.8690 - val_accuracy: 0.6903 - val_loss: 0.9161  
94/94 - 1s - 6ms/step - accuracy: 0.6903 - loss: 0.9418  
Model is 69.03% accurate.  
Saving trained model...  
Model saved successfully.
```



Slika 5.5.1. – Graf točnosti i greške u ovisnosti o epohama tijekom treniranja CNN

## 6. ZAKLJUČAK

Cilj ovog rada bilo je modeliranje i treniranje neuronskih mreža za klasifikaciju glazbenih žanrova. Ovisno o sadržaju audio datoteke, model, s određenom stopom točnosti, mora znati pretpostaviti i prepoznati glazbeni žanr pjesme.

Prvi korak je učenje i razumijevanje što su, te kako funkcioniraju neuronske mreže. Istraživanje je izvršeno analizom online članaka na temu i predmetnih materijala dostupnih na stranicama [FER Intraneta](#). Također je bitna teorija i rješenja za pretvorbu zvuka iz analognog u digitalni format. Zvuk se mora parametrizirati kako bi na temelju tih parametara model mogao generalizirati i klasificirati kojem žanru glazba točno pripada. Jednom kad je to svladano, programiranje može započeti.

Drugi korak je pronalazak i instalacija alata za stvaranje arhitekture umjetne neuronske mreže. Također su potrebne biblioteke za A/D pretvorbu i analizu audio datoteka. Za arhitekturu neuronske mreže, broj neurona, broj slojeva, aktivacijske funkcije itd. igraju ključnu ulogu u treniranju modela. Ako je broj ulaznih primjeraka prevelik, mreži će trebati previše vremena i resursa za treniranje. Mali broj neurona ili korištenje krivih aktivacijskih funkcija može dovesti do manje točnosti od one koju priželjkujemo. Takve parametre moramo mijenjati i analizirati njihov utjecaj na rezultat.

Mreža je više puta trenirana, a arhitektura se mijenjala. S pomoću testnih primjera i grafova bilježen je napredak prema cilju. Modeli su spremni i brisani kako su se pojavljivali obećavajući rezultati. Na poslijetku je nađen model koji zadovoljava potrebne kriterije i dovoljno dobro obavlja svoj zadatak.

## 7. LITERATURA

Članci i online izvori:

1. <https://www.fer.unizg.hr/predmet/uuui>
2. <https://www.ibm.com/topics/convolutional-neural-networks>
3. <https://www.tensorflow.org/tutorials/images/cnn>
4. <https://www.linkedin.com/advice/0/what-some-tips-optimizing-convolutional-neural-p38uc>
5. [https://en.wikipedia.org/wiki/Neural\\_network\\_\(machine\\_learning\)#Models](https://en.wikipedia.org/wiki/Neural_network_(machine_learning)#Models)
6. <https://en.wikipedia.org/wiki/Sound>
7. <https://www.britannica.com/science/sound-physics>
8. <https://www.youtube.com/playlist?list=PL-wATfeyAMNqlee7cH3q1bh4QJFAaeNv0>
9. <https://www.youtube.com/playlist?list=PL-wATfeyAMNrtbkCNsLcpoAyBBRJZVInf>

Izvori slika:

- [https://www.researchgate.net/figure/Basic-neuron-architecture-called-the-threshold-logic-unit-TLU-showing-inputs-I\\_fig5\\_368702279](https://www.researchgate.net/figure/Basic-neuron-architecture-called-the-threshold-logic-unit-TLU-showing-inputs-I_fig5_368702279)
- <https://images.spiceworks.com/wp-content/uploads/2022/05/18113202/The-Architecture-of-a-Neural-Network.png>
- <https://www.researchgate.net/profile/Nuwan-Kuruwitaarachchi/publication/323281289/figure/fig6/AS:701217198059529@1544194616847/Fast-Fourier-Transformation-2-Fast-Fourier-Transformation-To-increase-the-performance.ppm>
- <https://upload.wikimedia.org/wikipedia/commons/c/c5/Spectrogram-19thC.png>
- [https://cdn.prod.website-files.com/614c82ed388d53640613982e/646371e3bdc5ca90dee5331b\\_convolutional-neural-network%20\(1\).webp](https://cdn.prod.website-files.com/614c82ed388d53640613982e/646371e3bdc5ca90dee5331b_convolutional-neural-network%20(1).webp)
- <https://media.geeksforgeeks.org/wp-content/uploads/20190721025744/Screenshot-2019-07-21-at-2.57.13-AM.png>
- <https://saturncloud.io/images/blog/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way.webp>
- [https://miro.medium.com/v2/resize:fit:828/format:webp/1\\*m0H6-tUbW6grMlezlb52yw.png](https://miro.medium.com/v2/resize:fit:828/format:webp/1*m0H6-tUbW6grMlezlb52yw.png)
- [https://miro.medium.com/v2/resize:fit:828/format:webp/0\\*BrC7o-KTt54z948C.jpg](https://miro.medium.com/v2/resize:fit:828/format:webp/0*BrC7o-KTt54z948C.jpg)
- [https://miro.medium.com/v2/resize:fit:640/format:webp/1\\*MoiYQrW3Qaft6lfPQYbUbw.png](https://miro.medium.com/v2/resize:fit:640/format:webp/1*MoiYQrW3Qaft6lfPQYbUbw.png)

## 8. SAŽETAK

Rad se bavi implementacijom neuronskih mreža i konvolucijskih neuronskih mreža za prepoznavanje glazbenih žanrova. Glavni cilj istraživanja je poboljšati točnost klasifikacije žanrova analizom zvučnih zapisa. Modeli neuronskih mreža i CNN modeli su trenirani na online bazi podataka koja uključuje 10 različitih glazbenih žanrova. Klasične neuronske mreže koriste slojeve umjetnih neurona (TLU-perceptrona) za obradu značajki, dok CNN modeli koriste konvolucijske i pooling slojeve za izdvajanje kompleksnih uzoraka iz audio signala. Rezultati treniranja modela pokazuju da CNN nadmašuje standardne neuronske mreže u točnosti klasifikacije zbog svoje sposobnosti automatskog izdvajanja relevantnih značajki. Rad naglašava važnost dubokog učenja u obradi audio podataka i demonstrira potencijal CNN-a u poboljšanju prepoznavanja glazbenih žanrova, pružajući time vrijedne uvide za daljnja istraživanja i praktične primjene u području glazbene analitike.