

## 9. Konvolucijske neuronske mreže.

### 9.1 Cilj Vježbe

Upoznati se s konvolucijskim neuronskim mrežama i njihovom izgradnjom u Keras aplikacijskom okviru za strojno učenje. Primijeniti znanje stečeno o konvolucijskim neuronskim mrežama na problem klasifikacije podatkovnog skupa CIFAR-10.

### 9.2 Teorijska pozadina

Konvolucijske neuronske mreže (engl. *Convolutional Neural Networks - CNNs*) vrsta su dubokih neuronskih mreža koje se prvenstveno koriste za rješavanje raznih zadaća u području računalnog vida (kao npr. klasifikacije slika). U ovoj vježbi studenti se upoznaju s konvolucijskim neuronskim mrežama, njihovim slojevima i načinom učenja ovakvih modela u Keras aplikacijskom okviru za strojno učenje.

#### 9.2.1 Konvolucijska neuronska mreža

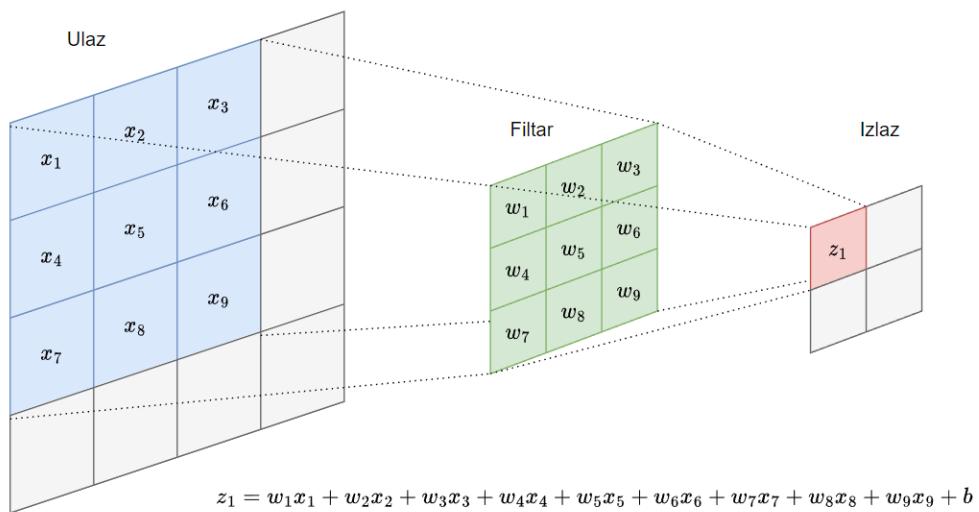
Konvolucijska neuronska mreža sastavljena je od više različitih slojeva:

1. konvolucijski slojevi
2. slojevi sažimanja
3. aktivacijski slojevi
4. potpuno povezani slojevi

Osnova CNN je operacija 2D konvolucije tj. **konvolucijski sloj**. Operacija 2D konvolucije izvodi se pomicanjem filtra po lokacijama u ulaznom volumenu, množenjem filtra i preklapajućeg dijela ulaznog volumena po elementima, te zbrajanjem rezultata da bi se proizvela mapa značajki kao što je prikazano na slici 9.1. Mapa značajki predstavlja odgovor filtra na ulaz.

Konvolucijski sloj najčešće ima veći broj filtara, a tipične prostorne dimenzije su 3x3, 5x5 i sl. dok dubina filtra ovisi o dubini ulaznog volumena. Korištenje višestrukih filtara omogućuje CNN-u izdvajanje više vrsta značajki iz ulaznih podataka poput rubova, kuteva, tekstura i sl. Ova operacija konvolucije jezgra je CNN-a i ponavlja se više puta kako bi se izdvojile značajke na više

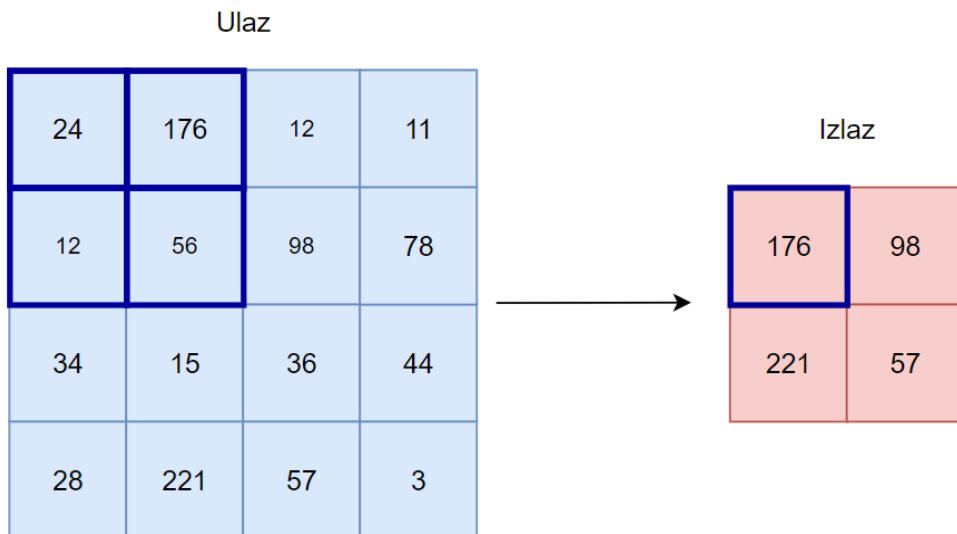
razina. Dobivene aktivacijske mape se slažu jedna pored druge pri čemu stvaraju novi volumen čija dubina ovisi o broju primjenjenih filtara. Širina i visina dobivenog volumena ovisi koristi li se nadopunjavanje (engl. *padding*) ulaznog volumena ili ne. U slučaju nadopunjavanja izlazni volumen ima jednaku visinu i širinu kao ulazni volumen. Filtar se može pomicati za jedan ili više elemenata po ulaznom filtru što također utječe na visinu i širinu rezultirajućeg volumena. Ovo definira stride konvolucijskog sloja. Težine filtara su parametri mreže koje je potrebno procijeniti na temelju dostupnog skupa podataka za učenje. Glavna prednost CNN-ova u odnosu na druge tradicionalne algoritme za klasifikaciju slika je upravo njihova sposobnost da automatski izluče najvažnije značajke iz ulaznih podataka, smanjujući potrebu za ručnim izdvajanjem značajki.



Slika 9.1: Operacija 2D konvolucije.

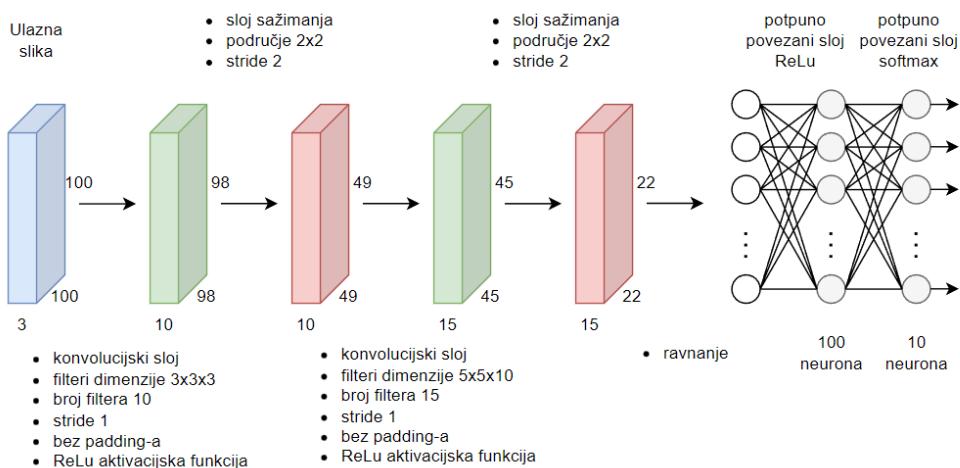
**Aktivacijska funkcija** (najčešće ReLu) primjenjuje se na svaki element mape značajki koju proizvodi konvolucijski sloj. Aktivacijska funkcija koristi se za uvođenje nelinearnosti u mrežu, kako bi mreža mogla aproksimirati složene uzorke i reprezentacije podataka. Sloj koji vrši **sažimanje po maksimalnoj vrijednosti** (engl. *max pooling*) je sloj za smanjenje prostorne veličine mape značajki koje proizvodi konvolucijski sloj. Ideja ovog sloja je odabir maksimalne vrijednosti iz skupa susjednih elemenata u mapi značajki (tipično područje je veličine 2x2) i koristiti je kao reprezentativnu vrijednost za tu regiju. To ima za posljedicu smanjenje veličine mape značajki, a također pomaže u očuvanju najvažnijih informacija u mapi značajki dok se manje važne informacije odbacuju. Sažimanje po maksimalnoj vrijednosti radi po svakoj aktivacijskoj mapi zasebno. Ovaj sloj ima nekoliko prednosti, uključujući smanjenje računalnih zahtjeva mreže smanjenjem broja parametara i poboljšanje robusnosti mreže čineći je invariantnom na male pomake ulaza. Sažimanje po maksimalnoj vrijednosti često se koristi u kombinaciji s konvolucijskim slojem. Važno je napomenuti da sloj sažimanja nema parametre koji se određuju postupkom učenja. Na slici 9.2 ilustriran je princip rada sloja koji provodi sažimanje po maksimalnoj vrijednosti.

CNN se najčešće sastoji od niza konvolucijskih slojeva i slojeva sažimanja koji se izmjenjuju. Ekstrahirane značajke zatim se transformiraju u 1D vektor pomoću operacije ravnjanja (engl. *flatten*) te se obrađuju dodatnim slojevima kao što su potpuno povezani slojevi kako bi se proizveo konačni rezultat. Na slici 9.3 prikazan je primjer konvolucijske neuronske mreže koja ima dva konvolucijska



Slika 9.2: Primjer sažimanja po maksimalnoj vrijednosti.

sloja, dva sloja sažimanja i dva potpuno povezana sloja od 100 i 10 neurona.



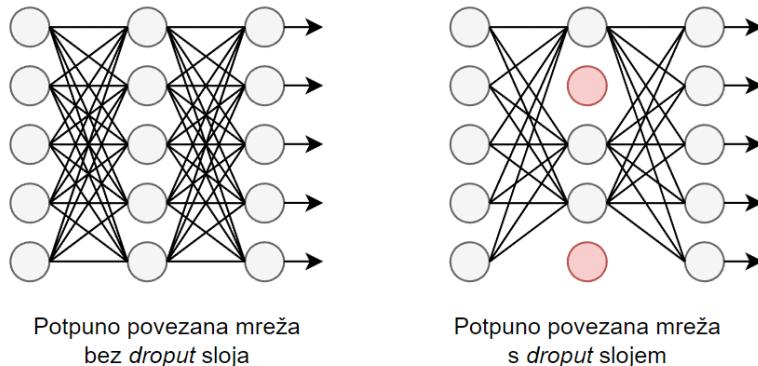
Slika 9.3: Primjer konvolucijske neuronske mreže.

### 9.2.2 Tehnike sprječavanja pretjeranog usklađivanja konvolucijske mreže

#### *Dropout* sloj

Jedan od jednostavnih načina sprječavanja pretjeranog usklađivanja na podatke za učenje dubokih neuronskih mreža je korištenje *dropout* sloja. Ovaj sloj nasumično isključuje određeni postotak neurona nekog sloja tijekom učenja. Ovo pomaže u smanjenju ovisnosti modela o bilo kojoj pojedinačnoj značajci i stoga smanjuje rizik od pretjeranog usklađivanja. Postotak neurona koji se nasumično isključuje je hiperparametar kojeg korisnik treba odabrati. Tipične vrijednosti su 0.2 - 0.5. *Dropout* sloj obično se postavlja između potpuno povezanih slojeva u CNN-u. Važno je napomenuti

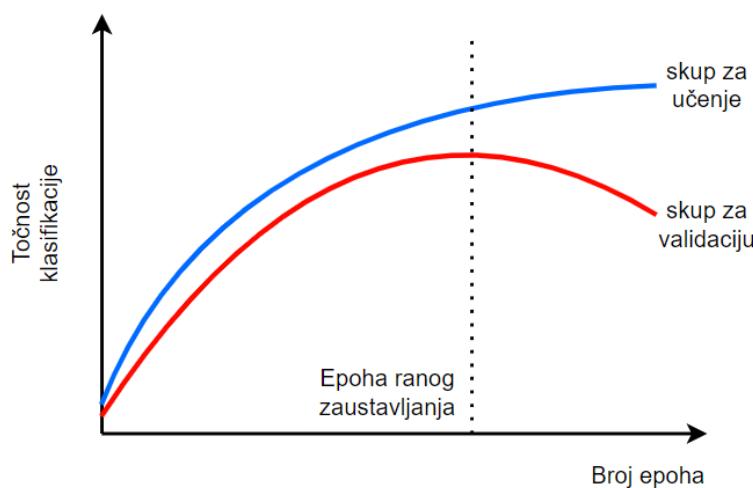
da se tijekom faze evaluacije odnosno testiranja mreže svi neuroni održavaju aktivnima.



Slika 9.4: Princip rada dropout sloja.

### Rano zaustavljanje

Tijekom učenja dubokih neuronskih mreža potrebni je pratiti performanse mreže na zasebnom, validacijskom skupu podataka. Rano zaustavljanje predstavlja zaustavljanje procesa učenja duboke neuronske mreže kada odgovarajuća metrika (npr. točnost klasifikacije) na skupu za validaciju prestane poboljšavati s iteracijama optimizacijskog algoritma. To znači da je započeo proces pretjeranog usklađivanja na podatke za učenje. Rano zaustavljanje najčešće se implementira na način da se parametri modela spremaju svaku epohu, a onda se prema vrijednostima postignutim na validacijskom skupu odabiru konačni parametri modela kako je prikazano na slici 9.5



Slika 9.5: Ilustracija ranog zaustavljanja.

### 9.2.3 Izgradnja konvolucijskih neuronskih mreža u Kerasu

#### Slojevi CNN

U okviru Keras Layers API-a dostupni su sljedeći slojevi koji omogućuju izgradnju konvolucijske neuronske mreže:

- conv2D klasa - kreira konvolucijski sloj; najvažniji argumenti
  - `filters` – broj konvolucijskih filtera, pozitivna cijelobrojna vrijednost
  - `kernel_size` – cijelobrojna vrijednost ili *tuple* cijelobrojnih vrijednosti koji definira visinu i širinu konvolucijskih filtera
  - `strides` – cijelobrojna vrijednost ili *tuple* cijelobrojnih vrijednosti koji definira pomak filtera po visini i širini
  - `padding` - "same" nadopunjavanje ulaznog volumena s nulama kako bi izlaz imao jednaku dimenziju, "valid" nema nadopunjavanja.
- MaxPooling2D klasa – kreira sloj sažimanja po maksimalnoj vrijednosti; najvažniji argumenti
  - `pool_size` - cijelobrojna vrijednost ili *tuple* cijelobrojnih vrijednosti koji definira veličinu prozora za operaciju sažimanja
  - `strides` - cijelobrojna vrijednost ili *tuple* cijelobrojnih vrijednosti koji definira pomak prozora po visini i širini; ako se ne definira onda je jednak `pool_size`
- Dropout klasa – kreira sloj za nasumično isključivanje ulaznih vrijednosti; najvažniji argument:
  - `rate` – decimalna vrijednost u rasponu 0 do 1 koja definira postotak isključivanja
- Flatten klasa – kreira sloj ravnjanja

Primjer 9.1 prikazuje programski kod koji kreira konvolucijsku neuronsku mrežu prikazanu na slici 9.3.

#### ■ Primjer 9.1

```
from tensorflow import keras
from tensorflow.keras import layers

model = keras.Sequential()
model.add(layers.Input(shape=(100, 100, 3)))
model.add(layers.Conv2D(10, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(15, (5, 5), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Flatten())
model.add(layers.Dense(100, activation="relu"))
model.add(layers.Dropout(0.3))
model.add(layers.Dense(10, activation="softmax"))
```

#### Keras funkcije povratnog poziva

Funkcije povratnog poziva (engl. *callbacks*) su dio *Keras Callbacks API* i prosljeđuju se drugim funkcijama koje ih onda u određenim trenucima izvršavaju. Keras funkcije povratnog poziva omogućuju razne korisne akcije tijekom učenja modela poput periodičko spremanje modela na trajnu memoriju računala, rano zaustavljanje, dinamičko podešavanje stope učenja i sl. Metoda `.fit` sadrži argument `callbacks` preko kojeg je procesu učenja moguće predati funkcije povratnog poziva.

**Tensorboard** je alat za vizualizaciju informacija tijekom učenja modela. Najčešće se tijekom učenja konvolucijske neuronske mreže za klasifikaciju prikazuju točnost klasifikacije i prosječna funkcija gubitka na skupu podataka za učenje i skupu podataka za validaciju. U Kerasu je na raspolaganju odgovarajuća klasa naziva Tensorboard koja zapisuje podatke u formatu kojeg može raščlaniti Tensorboard i prikazati ih u web pregledniku. Najvažniji argumenti Tensorboard klase su:

- `log_dir` – string, putanja do direktorija u koji se spremaju informacije o procesu učenja
- `update_freq` – 'epoch', 'batch' ili cjelobrojna vrijednost i definira frekvenciju zapisivanja informacija u direktorij (nakon svake epohe, nakon svake serije, ili nakon određenog broja serija)

Rano zaustavljanje je implementirano pomoću klase `EarlyStopping`. Najvažniji argumenti `EarlyStopping` klase su:

- `monitor` – string koji definira metriku koja se prati
- `patience` – broj epoha nakon koji se zaustavlja učenje ako ne postoji poboljšanje s obzirom na korištenu metriku

Primjer 9.2 prikazuje isječak koda koji definira funkcije povratnog poziva. Kada se pokrene učenje pomoću metode `.fit` aktivno je rano zaustavljanje i prikaz podataka u Tensorboard alatu.

### ■ Primjer 9.2

```
from tensorflow import keras

my_callbacks = [
    keras.callbacks.EarlyStopping(monitor="val_loss",
        patience = 12,
        verbose = 1),
    keras.callbacks.TensorBoard(log_dir = 'logs/cnn_3',
        update_freq = 100)
]

model.fit(X_train_n,
    y_train,
    epochs = 50,
    batch_size = 64,
    callbacks = my_callbacks,
    validation_split = 0.1)
```

## 9.3 Priprema za vježbu

1. Proučite poglavlje 9.2.
2. Po potrebi dodatno proučite dijelove Keras dokumentacije
3. Odredite ukupan broj parametara mreže koja je prikazana na slici 9.3 (točan odgovor je: 731155).

## 9.4 Rad na vježbi

Riješite dane zadatke.

**Zadatak 9.4.1** Skripta `Zadatak_1.py` učitava CIFAR-10 skup podataka. Ovaj skup sadrži 50000 slika u skupu za učenje i 10000 slika za testiranje. Slike su RGB i rezolucije su 32x32. Svakoj slici je pridružena jedna od 10 klasa ovisno koji je objekt prikazan na slici. Potrebno je:

1. Proučite dostupni kod. Od kojih se slojeva sastoji CNN mreža? Koliko ima parametara mreža?
2. Pokrenite učenje mreže. Pratite proces učenja pomoću alata Tensorboard na sljedeći način.

Pokrenite Tensorboard u terminalu pomoću naredbe:

```
tensorboard -logdir=logs
```

i zatim otvorite adresu <http://localhost:6006/> pomoću web preglednika.

3. Proučite krivulje koje prikazuju točnost klasifikacije i prosječnu vrijednost funkcije gubitka na skupu podataka za učenje i skupu podataka za validaciju. Što se dogodilo tijekom učenja mreže? Zapišite točnost koju ste postigli na skupu podataka za testiranje.

**Zadatak 9.4.2** Modificirajte skriptu iz prethodnog zadatka na način da na odgovarajuća mjesta u mrežu dodate *dropout* slojeve. Prije pokretanja učenja promijenite Tensorboard funkciju povratnog poziva na način da informacije zapisuje u novi direktorij (npr. `=/log/cnn_dropout`). Pratite tijek učenja. Kako komentirate utjecaj *dropout* slojeva na performanse mreže?

**Zadatak 9.4.3** Dodajte funkciju povratnog poziva za rano zaustavljanje koja će zaustaviti proces učenja nakon što se 5 uzastopnih epoha ne smanji prosječna vrijednost funkcije gubitka na validacijskom skupu.

**Zadatak 9.4.4** Što se događa s procesom učenja:

1. ako se koristi jako velika ili jako mala veličina serije?
2. ako koristite jako malu ili jako veliku vrijednost stope učenja?
3. ako izbacite određene slojeve iz mreže kako biste dobili manju mrežu?
4. ako za 50% smanjite veličinu skupa za učenje?

## 9.5 Izvještaj s vježbe

Kao izvještaj s vježbe prihvata se web link na repozitorij pod nazivom OSU\_LV.