

# TEORIJSKA SKRIPTA - OSNOVI RAČUNARSKE INTELIGENCIJE

## 1. NEURONSKE MREŽE

### Šta je neuronska mreža?

Neuronska mreža je model inspirisan biološkim neuronima. Sastoji se od:

- **Neurona (perceptrona)** - osnovna gradivna jedinica
- **Slojeva** - grupe neurona povezanih između sebe
- **Težina** - parametri koji se menjaju tokom učenja
- **Bias** - dodatni parametar za svaki neuron
- **Aktivacione funkcije** - određuju izlaz neurona

### Komponente neurona:

- n ulaza
- bias
- težine
- sabirač
- aktivaciona funkcija

### Treniranje neuronske mreže:

1. **Forward pass** - propagacija podataka kroz mrežu
2. **Loss funkcija** - meri grešku između predviđanja i stvarnih vrednosti
3. **Backpropagation** - računanje gradijenata
4. **Ažuriranje parametara** - menjanje težina na osnovu gradijenata

## 2. TEMPO UČENJA (LEARNING RATE)

### Definicija:

Tempo učenja je hiperparametar koji kontroliše koliko brzo model uči. Određuje veličinu koraka pri ažuriranju parametara tokom optimizacije.

### Uticaj tempa učenja:

- **Previše veliki:** model može "preskočiti" minimum, nestabilan trening
- **Previše mali:** sporo učenje, može zaglaviti u lokalnom minimumu
- **Optimalan:** efikasno učenje koje konvergira ka dobrom rešenju

### Strategije:

- Fiksni tempo učenja
- Adaptivni tempo (smanjuje se tokom treninga)
- Različiti algoritmi optimizacije (Adam, SGD, RMSprop)

## 3. KLASIFIKACIJA

### Tipovi klasifikacije:

- **Binarna** - 2 klase (pas/mačka)
- **Višeklasna** - više klasa (pas/mačka/slon)
- **Multi-label** - više oznaka istovremeno

### Metrije evaluacije:

- **Accuracy** - procenat tačnih predviđanja
- **Precision** - od predviđenih pozitivnih, koliko je stvarno pozitivno
- **Recall** - od stvarno pozitivnih, koliko je predviđeno
- **F1-score** - harmonijska sredina precision i recall

## 4. K-NEAREST NEIGHBORS (KNN)

### Princip rada:

- Traži K najbližih suseda u trening skupu
- Klasifikuje na osnovu većinske klase među susedima
- "Lazy learning" - ne uči eksplicitni model

### Prednosti:

- Jednostavan za implementaciju
- Ne pravi pretpostavke o podacima
- Radi dobro sa malim skupovima podataka

### Mane:

- Spor za velike skupove podataka
- Osetljiv na izbor K
- Loš sa visokim dimenzijama (curse of dimensionality)
- Osetljiv na skaliranje podataka

## 5. NAIVNI BAYES

### Osnova:

Koristi Bayesovu teoremu:  $P(A|B) = P(B|A) \times P(A) / P(B)$

### "Naivna" pretpostavka:

Sve features su nezavisne (što često nije tačno, ali algoritam i dalje radi dobro)

### Prednosti:

- Brz za trening i predviđanje
- Radi dobro sa malim skupovima podataka
- Dobra baseline metoda
- Otporan na irelevantne features

## Mane:

- Pretpostavka nezavisnosti često nije tačna
- Može biti nadguran od strane diskriminativnih metoda
- Potreban je Laplace smoothing za nove kategorije

## 6. K-MEANS KLASTEROVANJE

### Algoritam:

1. Odaberi K centroida nasumično
2. Dodeli svaku tačku najbližem centroidu
3. Pomeri centroide na sredinu svojih klastera
4. Ponavljaj 2-3 dok se centriodi ne stabilizuju

### PREDNOSTI K-MEANS:

- **Jednostavan** za razumevanje i implementaciju
- **Efikasan** za velike skupove podataka
- **Garantovana konvergencija**
- **Skalabilan** -  $O(n \times k \times i)$  složenost

### MANE K-MEANS:

- **Mora unapred znati K** (broj klastera)
- **Osetljiv na inicijalizaciju** (može zaglaviti u lokalnom minimumu)
- **Radi samo sa sfernim klasterima** - loš za kompleksne oblike
- **Osetljiv na outliere**
- **Pretpostavlja klasteri iste veličine**
- **Deterministički rezultat zavisi od početnih centroida**

### Izbor optimalnog K:

- **Elbow metoda** - traži "lakat" na grafiku SSE vs K
- **Silhouette analiza**
- **Gap statistic**

## 7. DBSCAN KLASTEROVANJE

### Parametri:

- **eps ( $\epsilon$ )** - maksimalna udaljenost između tačaka u klasteru
- **min\_samples** - minimalan broj tačaka potreban za formiranje klastera

### Tipovi tačaka:

- **Core points** - imaju dovoljno suseda u eps okolini
- **Border points** - u eps okolini core point-a
- **Noise points** - nisu core niti border

### Prednosti:

- Ne zahteva unapred definisan broj klastera
- Može identifikovati noise/outliere
- Radi sa klasterima bilo kog oblika
- Otporan na outliere

### Mane:

- Osetljiv na izbor parametara (eps, min\_samples)
- Problematican kada klasteri imaju različite gustine
- Može biti spor za velike skupove podataka

## 8. EVALUACIJA MODELA

### Train/Validation/Test podela:

- **Training set** - za učenje modela
- **Validation set** - za tuning hiperparametara
- **Test set** - za finalno testiranje

### Cross-validation:

- K-fold cross validation

- Stratified cross validation
- Leave-one-out cross validation

### Overfitting vs Underfitting:

- **Overfitting** - model memorisao trening podatke, loš na novim
- **Underfitting** - model previše jednostavan, loš i na trening podacima
- **Regularizacija** - tehnike za sprečavanje overfitting-a

## 9. PRAKTIČNI SAVETI

### Pre-processing podataka:

- Normalizacija/standardizacija
- Rukovanje nedostajućim vrednostima
- Feature engineering
- Encoding kategorijskih varijabli

### Evaluacija:

- Uvek koristi validation set
- Prati više metrika
- Visualizuj rezultate (confusion matrix, ROC kriva)

### Debugging modela:

- Počni sa jednostavnim modelom
- Proveri podatke pre treniranja
- Prati learning curves
- Testiraj na malom skupu podataka

# Teorija neuronskih mreža - CNN i RNN

## 1. KONVOLUTIVNE NEURONSKE MREŽE (CNN)

### Osnovne komponente CNN-a

CNN se sastoji od tri osnovna tipa slojeva:

1. **Konvolutivni sloj** (Convolution layer)
2. **Sloj sažimanja** (Pooling layer)
3. **Potpuno povezan sloj** (Fully connected layer)

### Konvolucija - matematička osnova

Konvolucija je matematički operator koji od dve funkcije  $f$  i  $g$  proizvodi treću:

$$(f * g)(x) = \int_{-\infty}^{\infty} f(t) \cdot g(x - t) dt$$

**2D konvolucija** predstavlja skalarni proizvod konvolucionog kernela sa delom slike nad kojim se vrši konvolucija.

### Konvolutivni sloj

**Šta radi:**

- Primenjuje filtere (kernele) na ulaznu sliku
- Svaki filter uči da prepozna određene karakteristike (ivice, uglove, teksture)
- Umesto ručnog kreiranja filtera, mreža sama uči optimalne filtere tokom treniranja

**Izlaz:** N mapa osobina (feature maps), gde je N broj filtera

**Faktori koji utiču na dimenzije izlaza:**

1. **Broj filtera** - određuje dubinu izlaza
2. **Korak (stride)** - za koliko pozicija pomeramo kernel
3. **Padding** - dodavanje piksela oko ivica da se spreči smanjenje dimenzija

## Sloj sažimanja (Pooling)

### Funkcija:

- Vršiti downsampling nad mapama osobina
- Smanjuje broj parametara
- Čini mrežu invarijantnom na manje transformacije

### Tipovi:

- **Max pooling** - uzima najveću vrednost
- **Average pooling** - računa prosek vrednosti

**Primer:** Slika  $32 \times 32 \times 1$  sa filterom  $2 \times 2$  i korakom 2  $\rightarrow$  rezultat  $16 \times 16 \times 1$  (4 puta manje parametara)

## Potpuno povezan sloj

- Standardni MLP sloj za finalnu klasifikaciju
- Prima "spljoštene" feature mape kao ulaz
- Generiše finalne predikcije

## 2. REKURENTNE NEURONSKE MREŽE (RNN)

### Osnovna karakteristika

RNN imaju **memoriju** - čuvaju stanje između različitih ulaza u sekvenci, za razliku od feed-forward mreža koje svaki ulaz obrađuju nezavisno.

### Tipovi sekvencijalnih problema

1. **Jedan ulaz  $\rightarrow$  sekvenca izlaza** (generisanje opisa slike)
2. **Sekvenca ulaza  $\rightarrow$  jedan izlaz** (analiza sentimenta)
3. **Sekvenca ulaza  $\rightarrow$  sekvenca izlaza** (mašinsko prevođenje)
4. **Sinhronizovane sekvence** (klasifikacija video kadra po kadra)



## Arhitektura RNN-a

### Ključni elementi:

- **x** - ulazna sekvenca
- **h** - skriveno stanje (memorija)
- **o** - izlaz
- **U** - matrica težina na ulazu
- **W** - matrica težina na skrivenom stanju
- **V** - matrica težina na izlazu

### Formule:

$$h_t = \tanh(x_t \cdot U^T + b_u + h_{t-1} \cdot W^T + b_w)$$
$$o_t = \text{activation}(h_t \cdot V^T + b_v)$$

## Vanilla RNN

- Osnovni tip RNN-a
- U praksi se češće koriste naprednije varijante (LSTM, GRU) zbog problema nestajanja gradijenata

## 3. PRAKTIČNI ASPEKTI

### CNN - Primena na MNIST

#### Tipična arhitektura:

```
# Konvolutivni deo (feature extraction)
Conv2d(1, 16, kernel_size=5, padding=2) → ReLU → MaxPool2d(2)
Conv2d(16, 32, kernel_size=5, padding=2) → ReLU → MaxPool2d(2)

# Klasifikacioni deo
Flatten() → Linear(7*7*32, 10)
```

## RNN - Primena na klasifikaciju imena

### Reprezentacija teksta:

- One-hot encoding na nivou karaktera
- Vokabular: ASCII slova + specijalni karakteri
- Sekvencijalno procesiranje karakter po karakter

### Ključne funkcije:

- Konverzija Unicode → ASCII
- Kreiranje tensora od stringova
- Sekvencijalno prosleđivanje kroz RNN

## 4. TEORETSKI KONCEPTI ZA RAZUMEVANJE

### Zašto CNN funkcioniše za slike?

1. **Lokalna povezanost** - susedni pikseli su povezani
2. **Deljenje parametara** - isti filter se koristi kroz celu sliku
3. **Hijerarhijsko učenje** - od jednostavnih ka složenim karakteristikama

### Zašto RNN funkcioniše za sekvence?

1. **Memorija** - čuva informacije o prethodnim elementima
2. **Fleksibilna dužina** - može da obradi sekvence različitih dužina
3. **Kontekstualno razumevanje** - odluke zavise od celog konteksta

### Praktični saveti

#### CNN:

- Koristiti padding da se održe dimenzije
- Eksperimentisati sa brojem i veličinom filtera
- Kombinovati više konvolutivnih slojeva za dublje učenje

#### RNN:

- Paziti na problem nestajanja gradijenata
- Koristiti LSTM/GRU za duže sekvence
- Voditi računa o balansiranosti podataka kod klasifikacije

## 5. ČESTI PROBLEMI I REŠENJA

### CNN problemi:

- **Prefikovanje** - koristiti dropout, data augmentation
- **Sporo treniranje** - batch normalization, learning rate scheduling

### RNN problemi:

- **Nestajanje gradijenata** - LSTM/GRU, gradient clipping
- **Eksplוזija gradijenata** - gradient clipping, pravilna inicijalizacija

### Opšti saveti:

- Početi sa jednostavnim arhitekturama
- Koristiti pretrenirane modele kad god je moguće
- Validacioni skup za praćenje performansi
- Eksperimentisati sa hiperparametrima sistematično