Predmet: "Vještačka inteligencija"
Laboratorijska vježba 5: Neuronske mreže (3/4)
Odgovorna nastavnica: Vanr. prof. dr Amila Akagić

Sadržaj vježbe:

1 Cilj vježbe

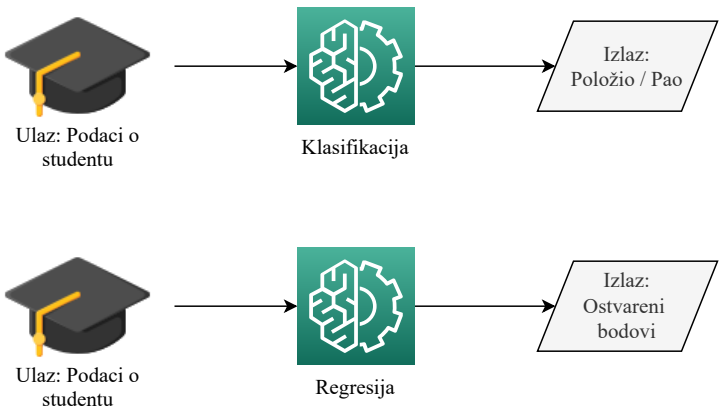
Cilj vježbe je upoznavanje sa drugim tipom nadziranog učenja - regresijom. Studenti se upoznaju sa razlikama pri dizajniranju arhitekture neuronske mreže za regresiju u odnosu na klasifikaciju te rješavaju jednostavan zadatak regresije koristeći Keras razvojni okvir.

2 Neuronske mreže u regresiji

2.1 Problem regresije

U prethodnim vježbama su rađeni zadaci u kojima je bilo potrebno podatku na ulazu dodijeliti diskretnu klasu i ovaj problem se naziva klasifikacija. Drugi problem nadziranog učenja je regresija. Ona podrazumijeva predviđanje kontinualne vrijednosti na izlazu na osnovu datih podataka na ulazu. Primjeri regresije uključuju predviđanje sutrašnje vremenske prognoze na osnovu postojećih meteoroloških podataka ili predviđanje koliko vremena će biti potrebno da se jedan projekat završi uzimajući u obzir poznate specifikacije.

Kao podsjetnik na razliku između problema klasifikacije i regresije može poslužiti slika 1 sa treće laboratorijske vježbe:



Slika 1: Razlika između klasifikacije i regresije. Klasifikacija ulazne podatke pridruži nekoj predefinisanoj klasi, dok regresija pokušava odrediti neku kontinualnu vrijednost na osnovu ulaza.

Kao što je pokazano primjerom na slici 1, za slučaj klasifikacije na osnovu ulaznih podataka o studentu, izlaz bi bila predikcija da li će student položiti ili pasti, dok za slučaj regresije izlaz bi bio broj ostvarenih bodova koji, u opštem

slučaju, može biti bilo koji realan broj. Konkretno za ovaj primjer taj realan broj je ograničen na raspon od 0 do 100.

2.2 Dizajniranje neuralne mreže za regresiju

Dizajniranje neuralne mreže za svrhu regresije je slično dizajniranju neuralne mreže za klasifikaciju. Jedna od očitih stvari koje treba promijeniti je izlazni sloj. Pri klasifikaciji se koristila aktivacijska funkcija na posljednjem sloju kako bi omogućila nelinearnost i kako bi se izlaz ograničio na neki opseg vrijednosti. Međutim, kao što je ranije navedeno, izlaz regresije je kontinualna vrijednost, te se iz tog razloga često izostavlja aktivacijska funkcija na posljednjem sloju i kao rezulat se dobija linearna funkcija, ili se eventualno koristi neki drugi oblik linearne aktivacijske funkcije koji je u tom slučaju potrebno eksplicitno postaviti. Iako je izostavljanje aktivacijske funkcije na posljednjem sloju najčešći slučaj, postoje određeni problemi u regresiji za koje je potrebno ograničiti izlaz na neki opseg vrijednosti. Primjer toga je predikcija vjerovatnoće da će sutra padati kiša. Za takve primjene najčešće se koristi `sigmoid` aktivacijska funkcija koja je korištena i u binarnoj klasifikaciji.

Druga razlika u dizajnu neuralne mreže za regresiju u odnosu na klasifikaciju je korištena mjera kvaliteta, odnosno funkcija gubitka. U regresiji se najčešće koriste **srednja kvadratna greška (eng. *Mean Squared Error - MSE*)** i **srednja apsolutna greška (eng. *Mean Absolute Error - MAE*)**.

Srednja kvadratna greška (MSE) je možda i najjednostavnija korištena funkcija gubitka. Računa se prema formuli:

$$MSE = \frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2$$

Za svaki ulazni podatak se računa razlika stvarne poznate vrijednosti od predviđene vrijednosti izlaza te se taj rezultat kvadrira. Iz ovoga se zaključuje da funkcija MSE uvijek ima nenegativnu vrijednost. Također, zbog operacije kvadrianja, ova funkcija gubitka naglašava velike greške, odnosno odstupanja od tačne vrijednosti. Ovo je upravo prednost ove funkcije gubitka jer se na ovaj način osigurava da model vrši relativno dobre predikcije i za podatke koji su ekstremni (eng. *outlier*). Međutim ovo može predstavljati problem, jer ako model napravi jednu veoma lošu predikciju tokom procesa treniranja, onda će to značajno utjecati na vrijednost funkcije gubitka te će se model nastojati modifikovati iako možda u većini drugih slučajeva već vrši izuzetno dobre predikcije.
Sa druge strane, srednja apsolutna greška (MAE) za prednost ima upravo ovo što je nedostatak MSE funkcije gubitka. MAE se računa prema formuli:

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

S obzirom da se u ovom slučaju ne vrši kvadriranje greške, to znači da se sve greške tretiraju sa istom težinom te prema tome ne daje se previše značaja ekstremnim vrijednostima (eng. *outlierima*). Ukoliko je bitno da model uzima u obzir i ekstremne vrijednosti, onda ova metrika nije pravi izbor.

Metrike koje se prate pri treniranju i testiranju modela za regresiju su također različite u odnosu na klasifikaciju. Tačnost nije upotrebljiva jer vjerovatnoća da će predviđena i stvarna vrijednost biti potpuno jednake je bliska nuli s obzirom da u općem slučaju izlaz modela za regresiju može biti bilo koji realan broj. Iz tog razloga je potrebno koristiti neke druge funkcije kao metrike, a korisnim se pokazuju već spomenute funkcije srednje kvadratne greške i srednje apsolutne greške.

3 Zadaci za rad u laboratoriji

Predviđeno je da se zadatak u nastavku radi u sklopu Jupyter Notebook okruženja. Svaki podzadatak treba biti zasebna Jupyter ćelija.

Zadatak 1 - Predviđanje cijena kuća u Bostonu

U ovom zadatku cilj je izvršiti predikciju cijene prilikom kupovine kuće u Bostonu na bazi podataka iz 1970. godine. Za potrebe vježbe koristiti će se historijski podaci, a oni uključuju informacije kao što je nivo kriminala u regiji, lokalni porez, itd. Ciljne (eng. *target*) vrijednosti su izražene u hiljadama dolara.

a) Učitati *The Boston Housing Price* skup podataka. Skup podataka je dio `Keras` biblioteke te se može učitati sljedećim kodom:

```
from keras.datasets import boston_housing
(train_data, train_targets), (test_data, test_targets) = boston_housing.
    load_data()
```

b) Koji je oblik podataka za treniranje, a koji za testiranje? Koliko ima ukupno atributa (značajki)?

c) Koji je opseg vrijednosti (*min* i *max*) za svaki atribut u skupu podataka? A koji je opseg vrijednosti za ciljne (eng. *target*) vrijednosti?

d) S obzirom da podaci zauzimaju široke spektre vrijednosti, izvršiti skaliranje korištenjem `MinMaxScaler`-a. Obavezno povesti računa da se parametri `scaler`-a za skaliranje izračunaju nad trening skupom podataka, a da se skaliranje izvrši i nad trening i nad test skupom;

e) Napisati pomoćnu funkciju `def model_mreze()` koja će vraćati model mreže (`return model`) opisan u nastavku. Model treba da ima 2 skrivena `Dense` sloja sa po 64 neurona sa `relu` aktivacijskim funkcijama. Dimenzije ulaznog sloja odrediti na osnovu broja atributa skupa podataka. Treći, koji je i posljednji sloj, treba imati samo jedan neuron i ne treba imati aktivacijsku funkciju;

f) Kompajlirati model da koristi `mse` funkciju gubitka, `adam` optimizator i `mae` metriku. S obzirom da su ciljne vijednosti izražene u hiljadama dolara, metrika MAE govori koliko model griješi u hiljadama dolara. Na primjer, ukoliko je vrijednost $MAE = 4.2$, to znači da predviđanja modela odstupaju u prosjeku za \$4200;

g) Izvršiti treniranje modela na 100 epoha, sa veličinom *batch*-a jednakoj 1. Koristiti 10% trening skupa za validaciju. Kolika je postignuta vrijednost funkcije gubitka na kraju treniranja, a kolika je vrijednost metrike MAE? Grafički prikazati; [1]

h) U prethodnom grafičkom prikazu problem predstavlja skala i visoka varijansa nad validacijskim skupom podataka. Iz tog razloga je korisno prikazati usrednjene vrijednosti i zanemariti prvih 10 uzoraka radi visoke skale u početnim epohama. Ovakav prikaz nad validacijskim skupom podataka se može postići sljedećim kodom:

```
def smooth_curve(points, factor=0.9):
  smoothed_points = []
  for point in points:
    if smoothed_points:
      previous = smoothed_points[-1]
      smoothed_points.append(previous * factor + point * (1 - factor))
    else:
      smoothed_points.append(point)
  return smoothed_points

mae_history = history['val_mae']

smooth_mae_history = smooth_curve(mae_history[10:])
plt.plot(range(1, len(smooth_mae_history) + 1), smooth_mae_history)
plt.xlabel('Epochs')
plt.ylabel('Validation MAE')
plt.show()
```

[1] Iskoristiti kod za grafički prikaz sa vježbe 3, no voditi računa da se više ne radi o metrici `accuracy`

i) Ponovo definisati model i istrenirati ga na 500 epoha. Ostale parametre ostaviti nepromijenjenim. Izvršiti grafički prikaz usrednjenih vrijednosti kao u prethodnom zadatku. Uporediti ova dva grafička prikaza. Šta se može zaključiti iz ovih grafika?

j) Izvršiti evaluaciju modela nad testnim skupom podataka. Koja je postignuta vrijednost funkcije gubitka, a koja vrijednost metrike MAE? Koliko prosječno u dolarima griješi model u svojim predikcijama?