

# LSTM neuronske mreže

primena nad problemom sekvencijalnog učenja

Nevena Soldat i Milena Kurtić

Matematički fakultet, Univerzitet u Beogradu

22. april 2020.

# Pregled

- Uvod
- Rekurentne neuronske mreže
- Problem generisanja teksta
- Rezultati
- Literatura

# Uvod

Neuronske mreže:

- jedna od najprimenjenijih metoda mašinskog učenja
- različite vrste (potpuno povezane, konvolutivne, rekurentne...)
- simulacija velike količine povezanih nervnih ćelija

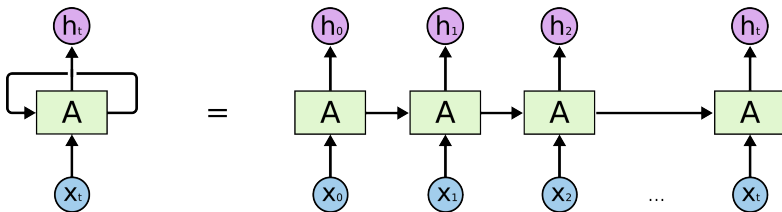
# Opis rada

- Implementacija LSTM neuronske mreže i primena na problem generisanja teksta
- Baza - opisi filmova
- Cilj: za ulaznu sekvencu reči pretpostaviti koje će biti naredne

# Rekurentne neuronske mreže

- Rekurentne neuronske mreže (eng. Recurrent Neural Network - RNN) - arhitektura neuronskih mreža specijalizovana za obradu sekvencijalnih podataka
- Sekvence nameću važnost redosleda zapažanja podataka
- Elementi ulazne sekvence se obrađuju u koracima
- Mreža ima skriveno stanje koje akumulira informaciju o elementima sekvence obrađenim u prethodnim koracima

# Arhitektura RNN



# Problemi sa RNN

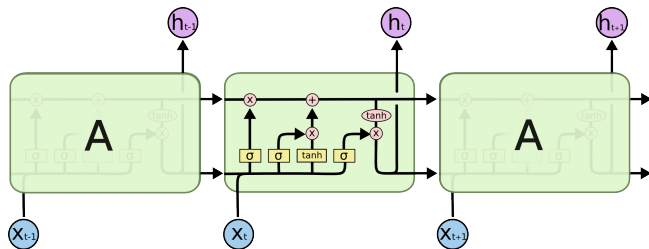
- 1 **Koordinate gradijenta eksplodiraju ili nestanu** prilikom izračunavanja gradijenta propagacijom u prošlost zbog velikog broja množenja
- 2 **Dugoročno čuvanje relevantnih informacija nije moguće** - doprinos starijih ulaza se brzo gubi pod uticajem novih.

# LSTM neuronske mreže

- duga kratkoročna memorija (eng. Long Short-Term Memory - LSTM)
- podvrsta rekurentnih neuronskih mreža
- rešavaju oba prethodno navedena problema
- pamćenje dugih sekvenci je praktično njihovo podrazumevano ponašanje
- osnovna ideja - postojanje ćelije koja čuva skriveno stanje, uz kontrolu pisanja, čitanja i zaboravljanja



# Arhitektura LSTM



# Problem generisanja teksta

- Generisanje teksta ima brojne primene, kao što su automatsko prevođenje jezika, sumarizacija teksta, generisanje opisa...
- Naš cilj je da dobijemo kratak opis filma, uz zadavanje ulaznih reči.
- Faze rešavanja:
  - 1 Preprocesiranje teksta
  - 2 Kreiranje i treniranje modela
  - 3 Rezultati

# Preprocesiranje teksta

- Neuronske mreže ne mogu da rade sa sirovim podacima, odnosno tekstem
- Kodiramo reči kao brojeve uz pomoć Tokenizer API-a
- Proces tokenizacije:
  - konvertovanje svih slova u mala
  - uklanjanje belina
  - uklanjanje znakova interpunkcije
  - ...
- **Token** - jedna reč kodirana kao broj

# Preprocesiranje teksta

- maksimalan broj reči - 10 000
- rečnik sa odgovarajućim vrednostima - reč : broj
- inverzni rečnik za dekodiranje - broj : reč

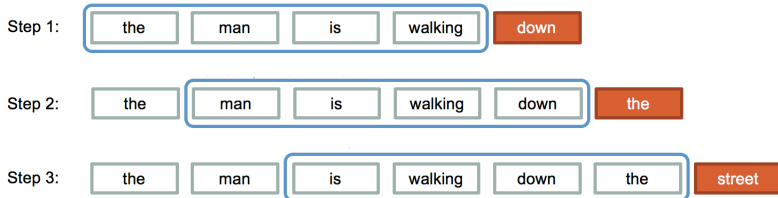
```
from keras.preprocessing.text import Tokenizer  
  
max_words = 10000  
tokenizer = Tokenizer(num_words = max_words)  
tokenizer.fit_on_texts(sample)  
  
reverse_word_map = dict(map(reversed, tokenizer.word_index.items()))
```

## Kreiranje ulaza i izlaza

- Biramo nasumično 100 filmova
- Uzimamo po 200 reči iz svakog opisa
- Sve opise spajamo u jedan korpus
- Ulaz - sekvenca od 20 reči
- Izlaz - naredna reč

# Primer

- Kreiranja ulaza i izlaza od sledeće rečenice: The man is walking down the street



# Arhitektura mreže

Mreža za rešavanje problema sastoji se od narednih slojeva:

- 1 Ulazni sloj
- 2 Prvi LSTM sloj
- 3 Sloj izbacivanja
- 4 Drugi LSTM sloj
- 5 Izlazni sloj

# Arhitektura mreže

```
model = Sequential()  
model.add(Embedding(vocab_size+1, 32, input_length = trainX.shape[1]))  
model.add(LSTM(200, return_sequences=True))  
model.add(Dropout(0.2))  
model.add(LSTM(200))  
model.add(Dense(trainy.shape[1], activation = 'softmax'))
```

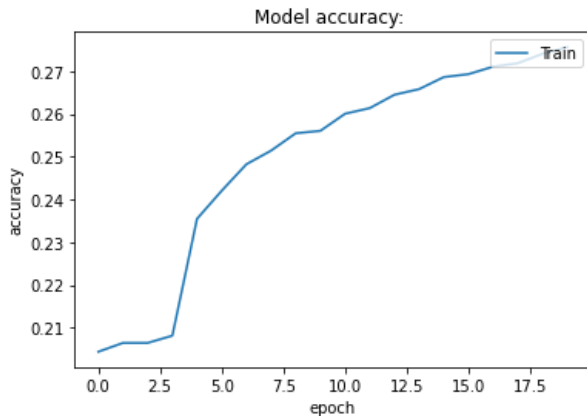


# Arhitektura mreže

```
model.compile(optimizer = 'adam', loss = 'categorical_crossentropy', metrics = ['acc'])  
filepath = "./weights.hdf5"  
checkpoint = ModelCheckpoint(filepath, monitor = 'loss',  
                             verbose = 1, save_best_only = True, mode = 'min')  
callbacks = [checkpoint]  
  
hist = model.fit(trainX, trainy, epochs = 20, batch_size = 128,  
                 verbose = 1, callbacks = callbacks, validation_split=0.2)
```

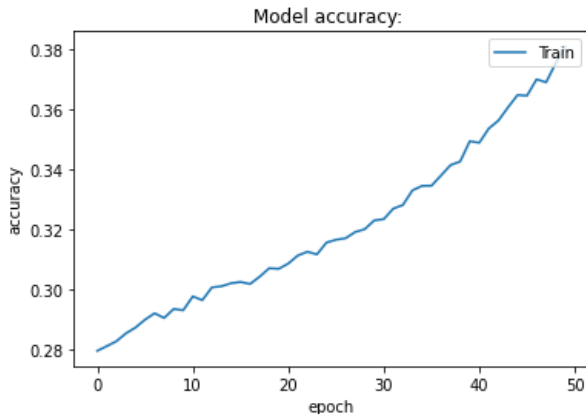
# Rezultati: Brzina konvergencije

- rezultat kroz 20 epoha



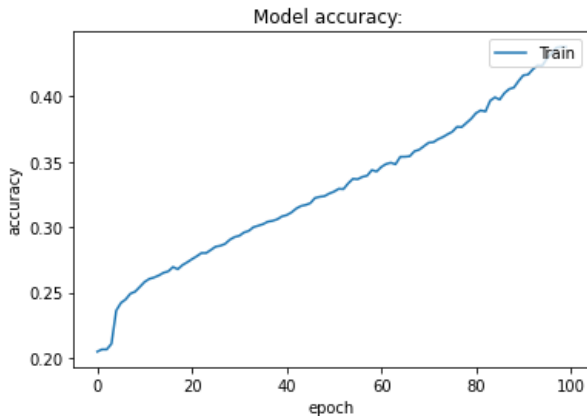
# Rezultati: Brzina konvergencije

- rezultat kroz 50 epoha



## Rezultati: Brzina konvergencije

- rezultat kroz 100 epoha



## Rezultati: Preciznost i greška

Broj epoha	Broj neurona	Preciznost	Gubitak
20	100	27.38	5.11
50	100	38.18	4.27
100	100	43.88	3.88
20	200	27.80	4.83
50	200	38.78	3.96

Rezultati: Izlaz

- rezultat kroz 20 epoha

```
print(generate_words("The movie", 50, model))
```

[illegible]

# Rezultati: Izlaz

## ■ rezultat kroz 50 epoha

```
print(generate_words("The movie", 50, model))
```

The Movie And Harford Phil Wheeler In Paradise Overhearing Reptiles The Fight Of A Local Woman And Has Been Forced To Be A Wife Technician And His Friend Who Is An Mother Who Is An Illegitimate Family Jr A J And Thief A Wife Technician And Says And Says And Tells By

# Rezultati: Izlaz

## ■ rezultat kroz 100 epoha

```
print(generate_words("The movie", 50, model))
```

The Movie And Everyone But Is Kidnapped To Be A Young Woman Who Is An Matador And Uses His Contented Hack Benet Br  
eaks A Merry Strife That He Has Been Forced To Be Changed By Ghostface In The Family But Been Slaughtered By His S  
traight Priest Who Is An Dominates Who Is



# Dalji razvoj

- povećanje broja epoha na više stotina
- eksperimentisanje sa brojem neurona i brojem slojeva mreže
- korišćenje GPU
- smanjenje broja ciljnih reči
- ...

**HVALA NA PAŽNJI!**