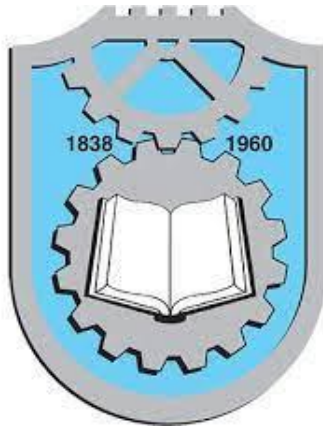


**Fakultet inženjerskih nauka  
Univerzitet u Kragujevcu**



# **Neuronske mreže treci domaći**

**- Mreže sa dugom kratkoročnom memorijom (LSTM) –**

**Student:**

Ilija Todorović 313/2023

**Predmetni nastavnik:**

Prof. Vesna Ranković

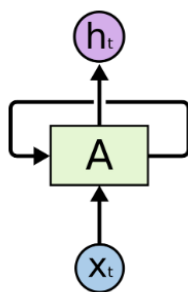
Kragujevac, 2024

# Sadržaj

|                         |    |
|-------------------------|----|
| Uvod.....               | 2  |
| Arhitektura LSTM-a..... | 3  |
| Zaključak.....          | 10 |
| Literatura.....         | 11 |

# Uvod

Mi ljudi ne počinjemo naše razmišljanje od nule svaki put. Kada nešto čitamo razumemo svaku reč na osnovu razumevanja onih prethodno napisanih. Naše misli imaju kontinuitet. Tradicionalne neuronske mreže to ne mogu da rade, što deluje kao velika mana. One imaju problem sa kratkoročnim pamćenjem, što znači da ako je sekvenca dovoljno duga teško im je da prenose informacije od ranijih do kasnijih koraka. Zbog toga se javlja potreba za Long Short-Term Memory (LSTM) mrežama koje su posebna vrsta rekurentke neuronske mreže (RNN). Ove mreže imaju sposobnost da pamte informacije tokom dužih vremenskih perioda. [1]

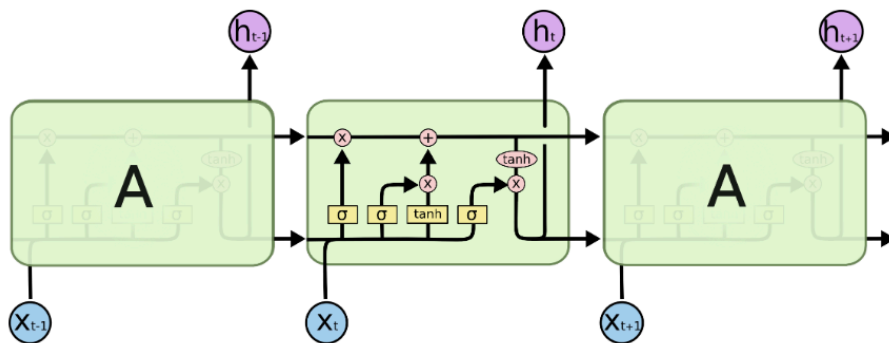


*Slika 1 - Loop koji koriste rekurentke neuronske mreže [1]*

RNN su neuronske mreže koje u sebi sadrže petlje (eng. *loop*) koje prenose informacije između različitih koraka unutar mreže. Ovaj koncept može biti viđen kao više kopija iste mreže koje međusobno razmenjuju informacije. Ova arhitektura pogodna je za rad sa listama i sekvencama [1]. U tradicionalnim neuronskim mrežama, ulazi i izlazi se tretiraju kao nezavisni, što nije pogodno za zadatke gde je važan kontekst (npr. predviđanje reči u rečenici). RNN koriste sekvencijalne podatke, jer se isti proces primenjuje na svaki element sekvence. Izlaz u RNN-u zavisi od prethodnih izračunavanja, što omogućava mreži da ima neku vrstu "memorije" koja pamti ranije informacije unutar sekvence [2]. Uz pomoć RNN-a postignut je veliki uspeh u oblastima kao što su prepoznavanje govora, prevođenje i opisivanje slika. Ključni uspeh ovih modela zasniva se na korišćenju LSTM-ova koji su poboljšana verzija običnih RNN. Nešto više o njima u nastavku.

## Arhitektura LSTM-a

LSTM mreže, što je skraćenica za "Long Short Term Memory" mreže, predstavljaju specijalnu vrstu rekurentnih neuronskih mreža (RNN), sposobnih da uče dugoročne zavisnosti. Prvi su ih predstavili Hochreiter i Schmidhuber 1997. godine, a kasnije su ih unapredili i popularizovali mnogi istraživači. Danas se LSTM mreže koriste u širokom spektru problema zbog svoje izvanredne efikasnosti. LSTM mreže su posebno osmišljene da reše problem dugoročnog pamćenja. Pamćenje informacija tokom dužih vremenskih perioda praktično je njihovo prirodno ponašanje, a ne nešto sa čim imaju poteškoće da nauče. Sve rekurentne neuronske mreže sastoje se od niza ponavljajućih modula neuronske mreže. U standardnim RNN mrežama, ovaj ponavljajući modul često ima jednostavnu strukturu, poput jednog sloja sa  $\tanh$  aktivacionom funkcijom. [1]

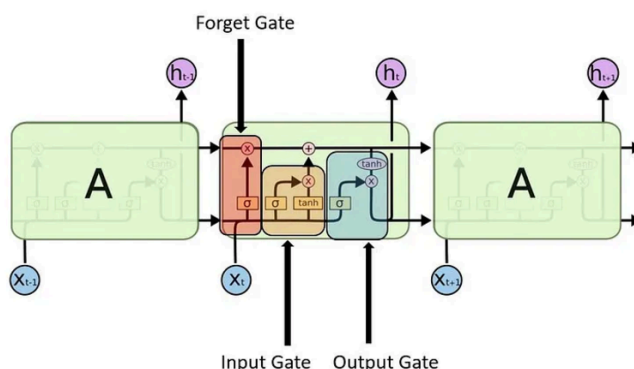


Slika 2 - Grafčki prikaz LSTM [1]

U gore prikazanom dijagramu, svaka linija prenosi čitav vektor, od izlaza jednog čvora do ulaza u druge. Ružičasti krugovi predstavljaju operacije koje se primenjuju na svaku komponentu vektora, poput sabiranja vektora, dok žute kutije označavaju slojeve neuronske mreže koji se uče. Spajanje linija označava konkatenciju (spajanje vektora), dok račvanje linije znači da se njen sadržaj kopira i šalje na različita mesta. [1]

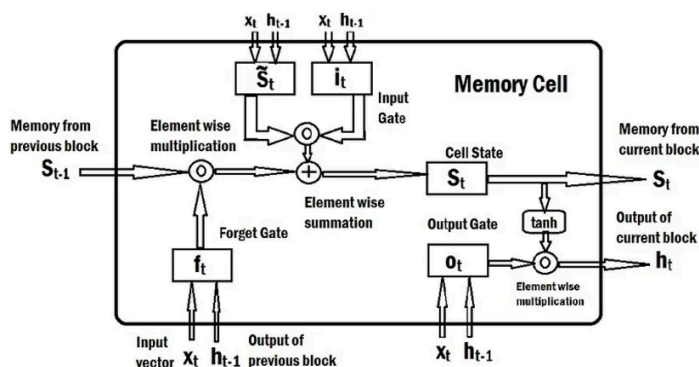
Ključ LSTM mreža je **stanje ćelije** (cell state), horizontalna linija koja prolazi kroz vrh dijagrama.

Stanje ćelije je poput pokretne trake. Ono se prostire duž celog lanca, sa samo manjim linearnim interakcijama. Informacije lako mogu da teku kroz stanje ćelije bez ikakvih promena. LSTM mreža ima sposobnost da uklanja ili dodaje informacije u stanje ćelije, pažljivo kontrolisano putem struktura koje se zovu **kapije** (gates). Kapije predstavljaju mehanizam za selektivno propuštanje informacija. LSTM koristi proces u tri koraka, što je prikazano na dijagramu ispod, gde LSTM modul ima tri kapije: **kapiju za zaboravljanje** (Forget gate), **ulaznu kapiju** (Input gate) i **izlaznu kapiju** (Output gate). [2]



*Slika 3 - Vrste kapija [2]*

Osnovna komponenta LSTM mreža je **memorijska ćelija**. Ona može održavati svoje stanje tokom vremena, a sastoji se od eksplicitne memorije (koja se naziva i **vektor stanja ćelije**) i jedinica za kontrolu protoka informacija, poznatih kao **kapije**. Kapije regulišu protok informacija u memoriju i iz nje, omogućavajući LSTM mrežama da precizno upravljaju time koje informacije treba sačuvati, zaboraviti ili koristiti u datom trenutku[2].



*Slika 4 - Memorijska ćelija [2]*

Objašnjenje pojmova sa dijagrama[2]:

### **Vektor stanja ćelije**

Vektor stanja ćelije predstavlja memoriju LSTM mreže i menja se zaboravljanjem starih informacija (putem kapije za zaboravljanje) i dodavanjem novih informacija (putem ulazne kapije).

### **Kapije**

Kapija se sastoji od sloja neuronske mreže sa **sigmoidnom** aktivacionom funkcijom, nakon čega sledi operacija koja se primenjuje na svaku komponentu vektora. Kapije kontrolišu protok informacija ka memoriji i iz nje. Kapije se kontrolišu spajanjem izlaza iz prethodnog vremenskog koraka i trenutnog ulaza, a opcionalno i vektora stanja ćelije.

### **Kapija za zaboravljanje**

Kontroliše koje informacije treba odbaciti iz memorije i odlučuje koliko prošlih informacija treba zadržati.

### **Kapija za ažuriranje/ulazna kapija**

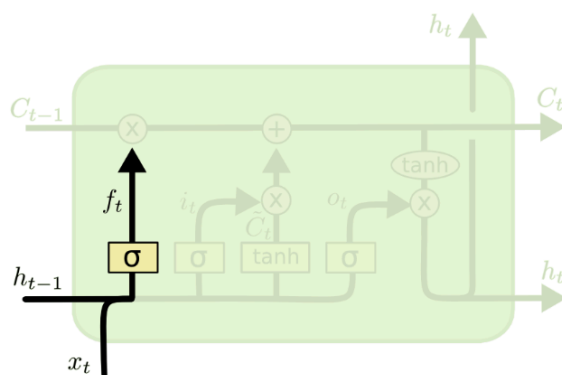
Kontroliše koje nove informacije se dodaju u stanje ćelije iz trenutnog ulaza i odlučuje koliko tih novih informacija treba dodati u trenutno stanje.

### **Izlazna kapija**

Uslovno odlučuje koje informacije treba izvući iz memorije i odlučuje koji deo trenutnog stanja ćelije postaje izlaz.

## Kako rade LSTM mreže korak po korak?

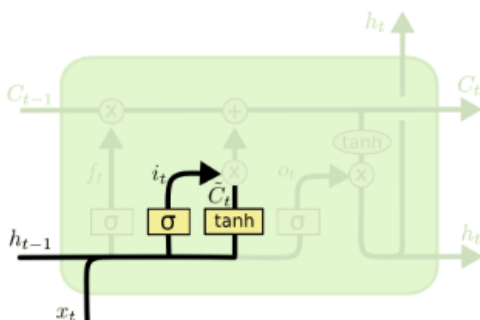
Prvi korak u LSTM mreži je odlučivanje koje informacije će se odbaciti iz stanja ćelije. Ovu odluku donosi sloj sa **sigmoidnom** aktivacionom funkcijom, nazvan **kapija za zaboravljanje** (Forget gate). On posmatra prethodni izlaz, označen sa  $h_{t-1}$ , i trenutni ulaz  $x_t$ , i daje rezultat u vidu broja između 0 i 1 za svaku vrednost u stanju ćelije  $C_{t-1}$ . Vrednost 1 znači "potpuno zadrži ovu informaciju", dok vrednost 0 znači "potpuno odbaci ovu informaciju" [1].



$$f_t = \sigma(W_f \cdot [h_{t-1}, x_t] + b_f)$$

Slika 5 - Forget gate [1]

Sledeći korak je odlučiti koje nove informacije ćemo sačuvati u stanju ćelije. Ovaj proces ima dva dela. Prvo, **sigmoidni** sloj, poznat kao "ulazna kapija" (*input gate*), odlučuje koje vrednosti ćemo ažurirati. Zatim, **tanh** sloj generiše vektor novih potencijalnih vrednosti, označenih kao  $\tilde{C}_t$ , koje bi mogle biti dodate trenutnom stanju. U narednom koraku kombinovaćemo ova dva rezultata kako bismo napravili ažuriranje stanja ćelije [1].

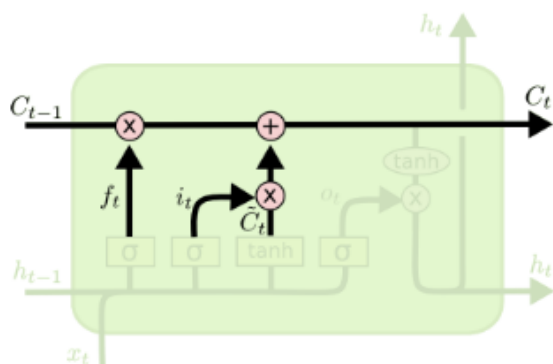


$$i_t = \sigma(W_i \cdot [h_{t-1}, x_t] + b_i)$$

$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] + b_C)$$

Slika 5 - Input gate [1]

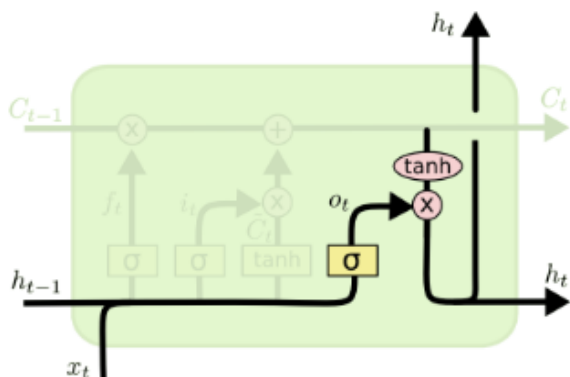
Sada je vreme da se ažurira staro stanje ćelije  $C_{t-1}$  u novo stanje  $C_t$ . Prethodni koraci su već odredili šta treba da se uradi, preostaje samo da se to odradi. Prvo se množi staro stanje sa  $f_t$ , čime se "zaboravljaju" one informacije koje smo prethodno odlučili da uklonimo. Zatim se dodaje  $i_t \cdot C_t$ . Ovo su nove kandidovane vrednosti, skalirane na osnovu toga koliko smo odlučili da ažuriramo svaku vrednost stanja [1].



$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Slika 6 - State line [1]

Na kraju, se treba odlučiti šta će se izbaciti kao izlaz. Ovaj izlaz će biti zasnovan na stanju ćelije, ali u filtriranoj verziji. Prvo se koristi **sigmoid** sloj, koji određuje koje delove stanja ćelije će se izbaciti kao izlaz. Zatim stanje ćelije prolazi kroz **tanh** funkciju (koja ograničava vrednosti između  $-1$  i  $1$ ) i množi se rezultat sa izlazom sigmoid kapije. Na taj način se izbacuju samo oni delovi stanja koje smo odlučili da sačuvamo kao izlaz [1].



$$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$$

$$h_t = o_t * \tanh(C_t)$$

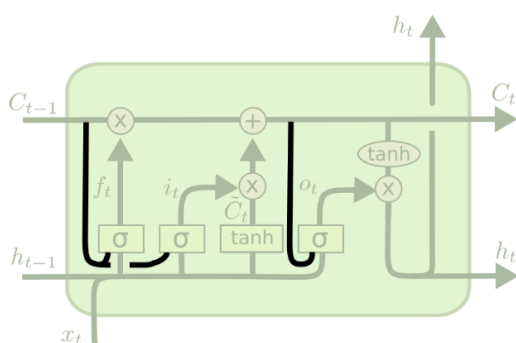
Slika 7 - Output gate [1]



## Vrste LSTM-ova

U prethodnom podnaslovu opsian je osnovni model LSTM, pored ove verzije postoje i druge koje se ne ralikuju previše, ali su ipak vredne pomena:

Jedna popularna varijanta LSTM-a, koju su uveli Gers i Schmidhuber (2000), uključuje dodavanje takozvanih **“peephole connections”** (rupica za virenje). To znači da se omogućava slojevima kapija da imaju direktan uvid u stanje ćelije, što im pruža dodatne informacije prilikom donošenja odluka [1].



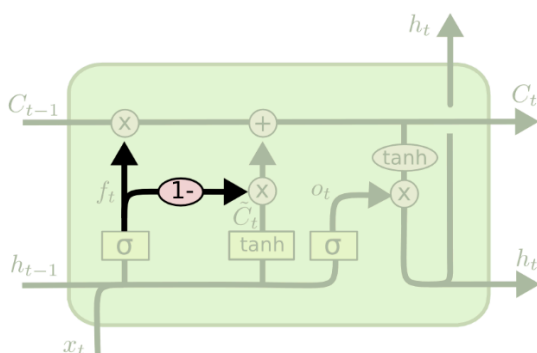
$$f_t = \sigma(W_f \cdot [C_{t-1}, h_{t-1}, x_t] + b_f)$$

$$i_t = \sigma(W_i \cdot [C_{t-1}, h_{t-1}, x_t] + b_i)$$

$$o_t = \sigma(W_o \cdot [C_t, h_{t-1}, x_t] + b_o)$$

Slika 8 - Model sa rupicama za virenje [1]

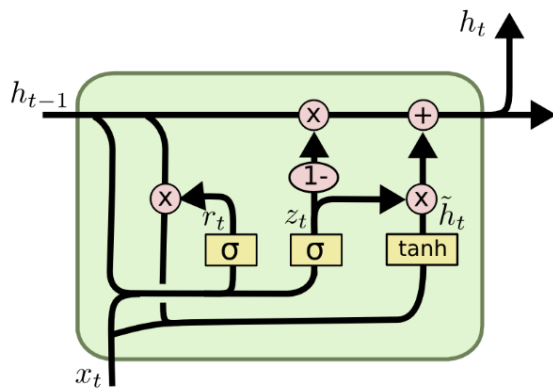
Druga vrsta koristi **povezane kapije za zaboravljanje i unos podataka** (*coupled forget and input gates*). Umesto da zasebno odlučuju šta će se zaboraviti i koje nove informacije će se dodati, ove odluke se donose zajedno [1].



$$C_t = f_t * C_{t-1} + (1 - f_t) * \tilde{C}_t$$

Slika 9 - Model sa povezanim kapijama [1]

Jedna značajnija varijacija LSTM-a je **Gated Recurrent Unit (GRU)**, koju su uveli Cho i saradnici (2014). GRU kombinuje kapije za zaboravljanje i unos u jednu jedinstvenu kapiju nazvanu "**kapija za ažuriranje**" (*update gate*). Pored toga, GRU spaja stanje ćelije i skriveno stanje u jednu komponentu i uvodi još nekoliko promena. Ovaj model je jedan od popularnijih [1].



$$z_t = \sigma(W_z \cdot [h_{t-1}, x_t])$$

$$r_t = \sigma(W_r \cdot [h_{t-1}, x_t])$$

$$\tilde{h}_t = \tanh(W \cdot [r_t * h_{t-1}, x_t])$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

*Slika 10 - GRU [1]*

## **Zaključak**

LSTM modeli su sjajan alat za obradu sekvencijalnih podataka, zahvaljujući sposobnosti da efikasno pamte i zaboravljaju informacije preko dužih vremenskih perioda. Standardni LSTM koristi mehanizam kapija za upravljanje protokom informacija, ali postoji više varijacija, poput peephole verzija, povezanih kapija i GRU modela, koje uvode pojednostavljenja ili dodatna prilagođavanja. Ove inovacije omogućavaju veću fleksibilnost i primenu LSTM modela u različitim problemima. Pored LSTM modela postoje i druge arhitekture poput Clockwork RNN-ova. Koji je najbolji izbor? Greff i saradnici (2015) su napravili odličnu uporednu analizu popularnih varijanti i zaključili da su sve prilično slične u performansama. S druge strane, Jozefowicz i saradnici (2015) testirali su više od deset hiljada arhitektura RNN-a i pronašli neke koje su bolje od LSTM-a za određene zadatke. [1]

# Literatura

[1] - Colah's blog - Understanding LSTM Networks - pristupljeno 25.11.2024.

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>

[2] - Medium - Long shot-term Memory(LSTM) networks in a nutshell - pristupljeno 25.11.2024.

<https://colah.github.io/posts/2015-08-Understanding-LSTMs/>