

Neuronske mreže

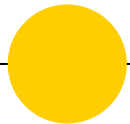
Rekurentne neuronske mreže



Agenda

- Motivacija
- Standardne *RNN*
- *LSTM*
- *GRU*
- Primeri
- Vizualizacija obučениh ćelija
- Dodatno ćitanje

Motivacija





Motivacija

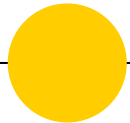
- Ljudi ne počinju da razmišljaju od nule svake sekunde
- Dok čitate nešto, razumete svaku reč na osnovu vašeg razumevanja prethodnih reči
 - Ne bacate sve i ponovo počinjete razmišljanje od nule
 - Vaše misli imaju postojanost
- Tradicionalne neuronske mreže ne mogu to da urade



Motivacija

- Npr.: ako želimo da klasifikujemo kakav se događaj dešava u svakom trenutku u filmu:
 - Nejasno je kako bi tradicionalna neuronska mreža mogla da koristi svoja razmišljanja o prethodnim događajima u filmu da informiše kasnije
 - Rekurentne neuronske mreže rešavaju ovaj problem.

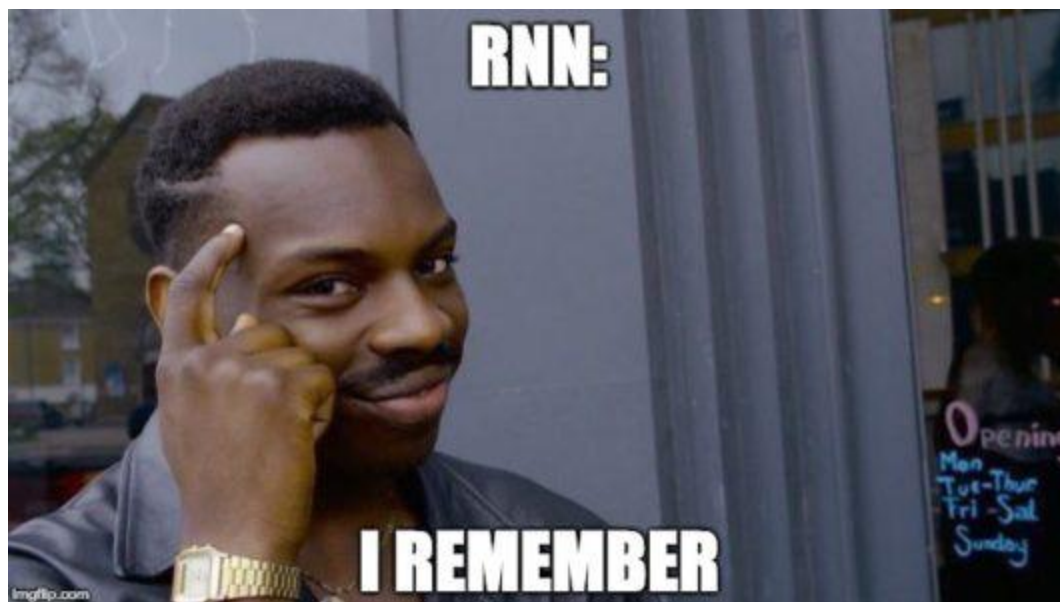
Standardne RNN





Standardne RNN

- Rekurentne neuronske mreže (*RNN*) su vrsta neuronskih mreža gde se izlaz neurona vraća na njegov ulaz
 - Pravi neku vrstu internog stanja mreže - memoriju:
 - Nosi informaciju o svim prethodnim izračunavanjima
 - U teoriji: mogu da čuvaju informaciju proizvoljno dugo
 - U praksi: čuvanje svega par koraka unazad
 - Idealne za procesiranje sekvenci



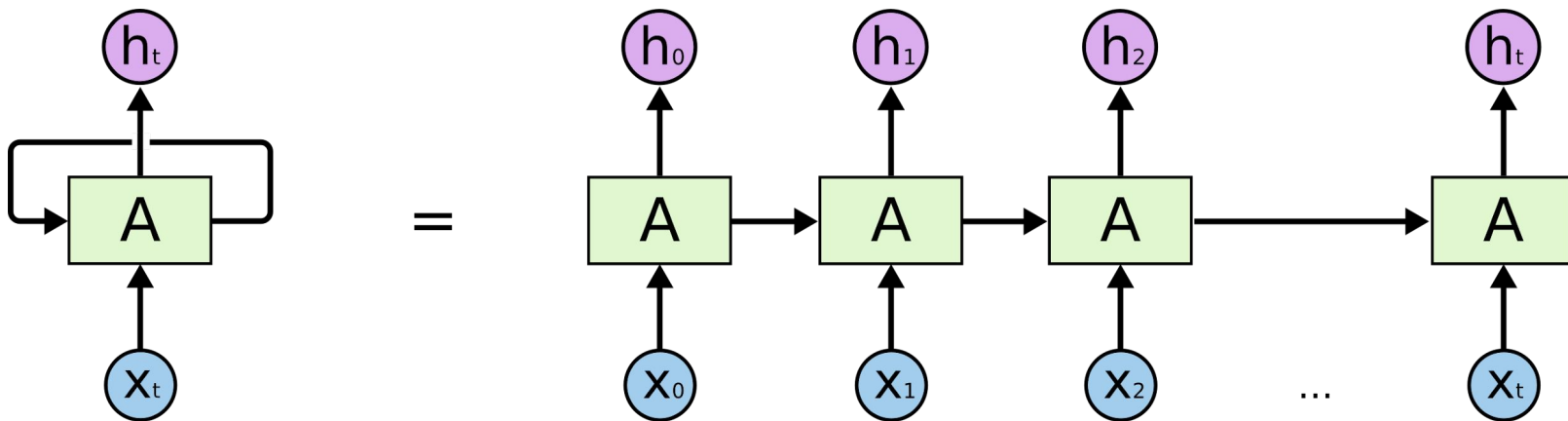


Standardne RNN

- Standardne *feed-forward* mreže:
 - Fiksni vektori kao ulazi (npr. slika), fiksni vektori kao izlazi (npr. verovatnoća klase)
 - Fiksna broj koraka u izračunavanju (broj slojeva u mreži)
 - Ulazni podaci “nezavisni”
- RNN:
 - Mogućnost procesiranja **sekvenci** vektora (tekst, govor, zvuk...)

Standardne RNN

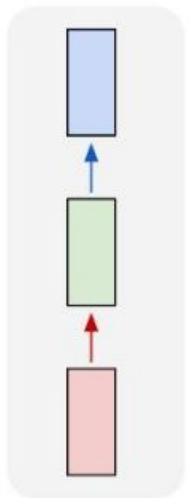
- RNN* se može smatrati višestrukim kopijama iste mreže, od kojih svaka prenosi poruku nasledniku





Standardne RNN

one to one



Vanilla NN

one to many

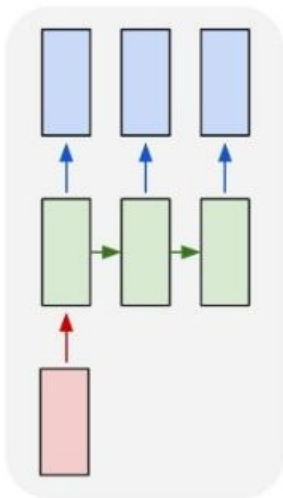
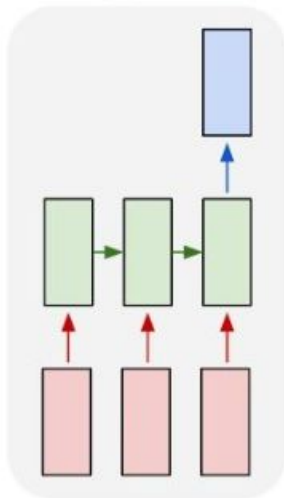


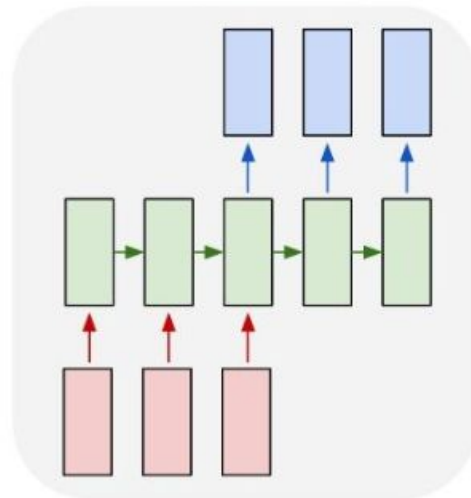
Image Captioning
slika => sekvenca reči

many to one



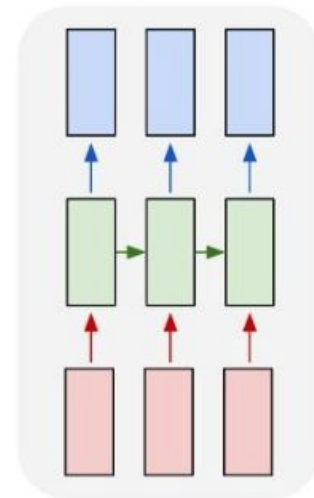
Klasifikacija sentimenta
sekvenca reči => sentiment

many to many



Neural Machine Translation
sekvenca reči => sekvenca reči

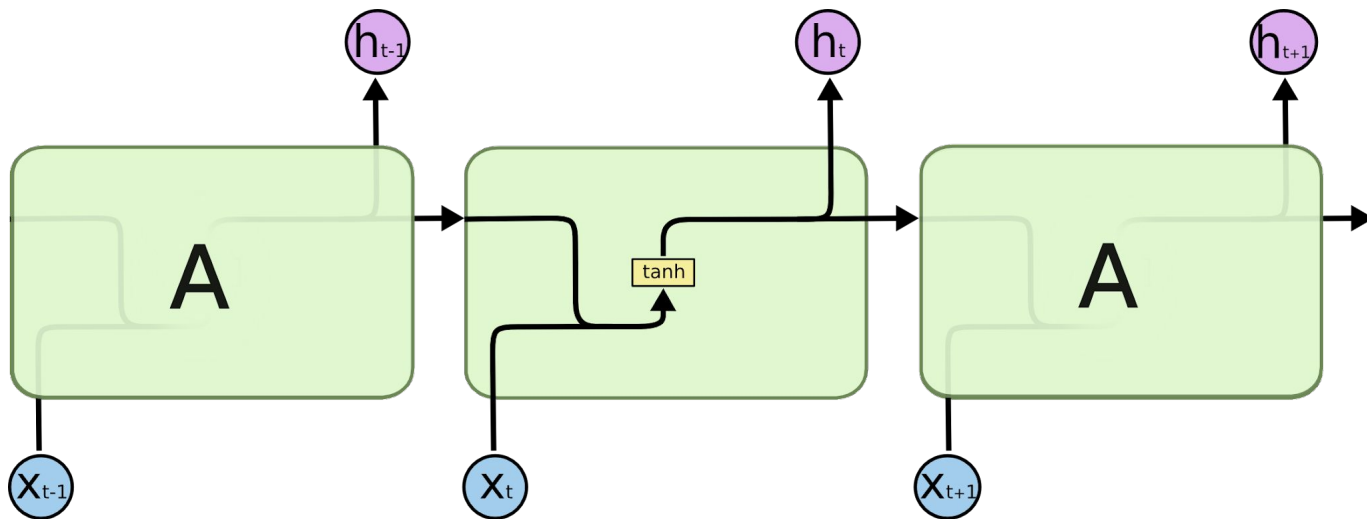
many to many



Klasifikacija video
sadržaja na nivou frejma

Standardne RNN

- Ponavljajući modul u standardnom *RNN*-u sadrži jedan sloj:





Standardne RNN

- Procesiranje sekvence vektora \mathbf{x} se može izvršiti primenom rekurentne formule u svakom trenutku vremena \mathbf{t}
- Kao funkcija se obično koristi hiperbolični tangens (**\tanh**)



Standardne RNN

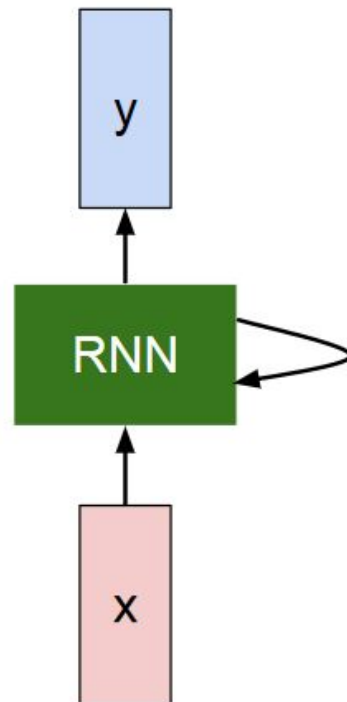
$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state

some function
with parameters W

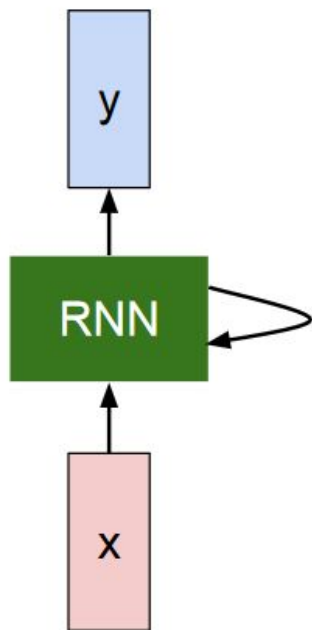
old state

input vector at
some time step





Standardne RNN



$$h_t = f_W(h_{t-1}, x_t)$$



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

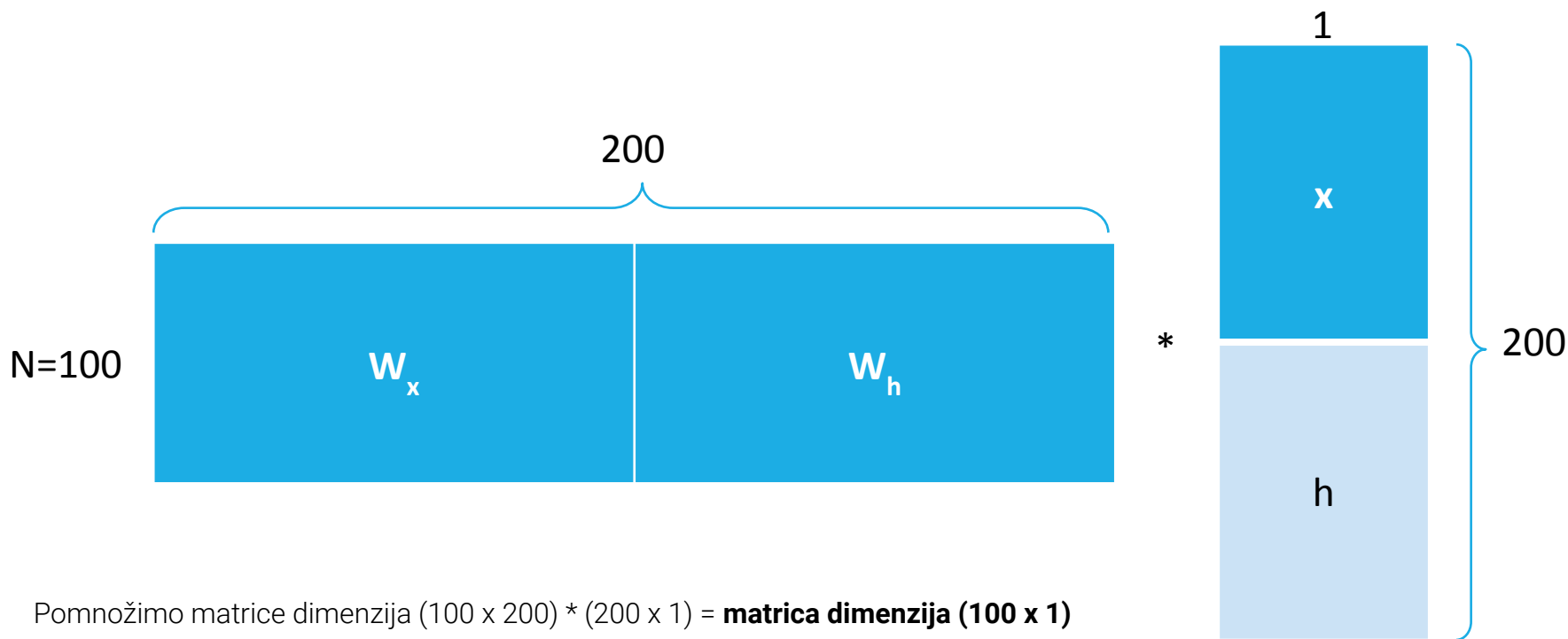


Standardne RNN

- Kako je moguće da je dimenzija izlaza iz ćelije ista kao dimenzija ulaza, ako se ulaz u ćeliju konkatenuira sa vektorom informacija iz prethodnog koraka (duplo veća dimenzija)?
- Pretpostavimo da je ulaz \mathbf{x} u rekurentnu ćeliju 100D:



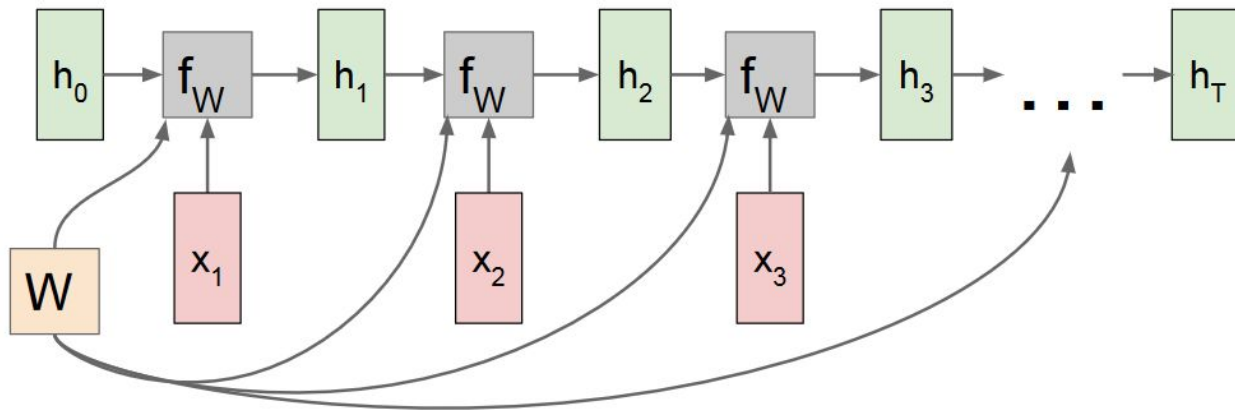
Standardne RNN



Standardne RNN

- Graf izračunavanja:

Re-use the same weight matrix at every time-step





Standardne RNN

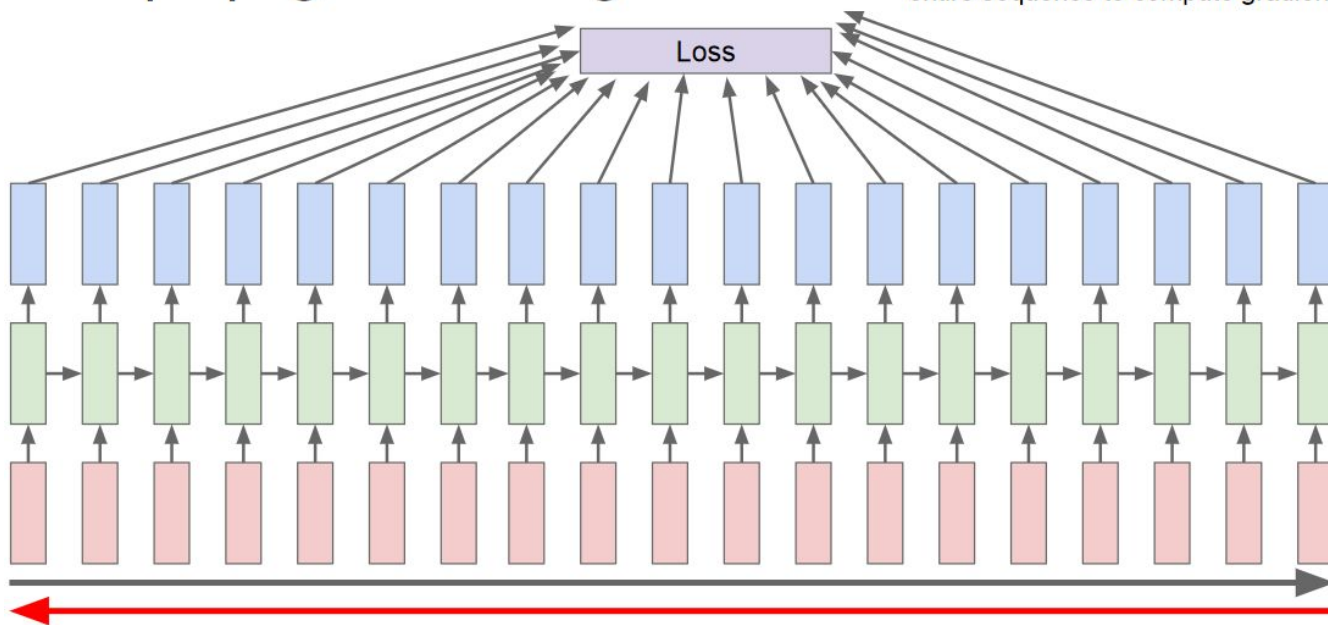
- Obučavanje:
 - *BPTT - Backpropagation through time*
 - Modifikacija standardnog *BP*:
 - Računanje gradijenta u prošlim koracima uzima u obzir gradijent iz budućih koraka
 - Zapravo samo *fancy* naziv za najobičniji *backpropagation unroll*-ovane *RNN*.



Standard RNN

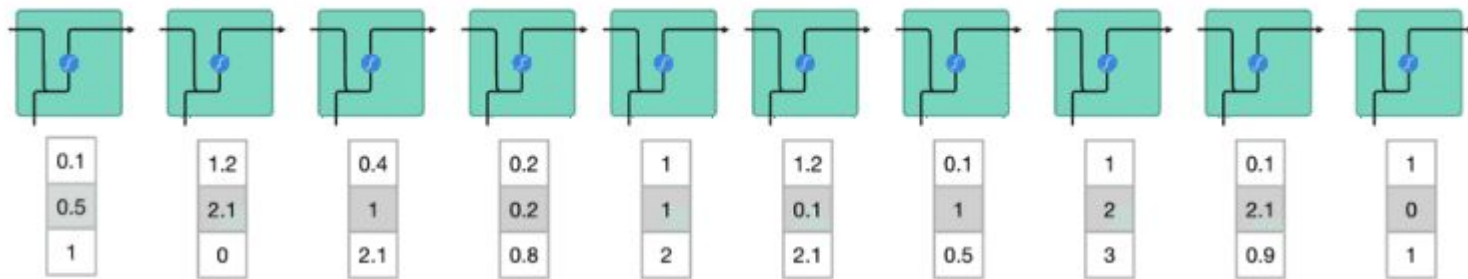
Backpropagation through time

Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient



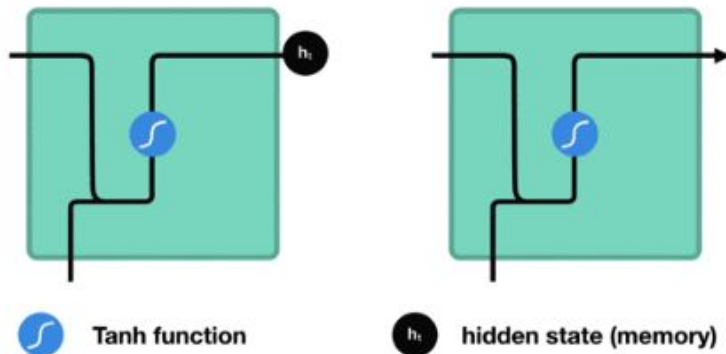
Standardne RNN

- Procesiranje sekvence dodavanjem novih primeraka u jedinici vremena (dimenzija **t**):



Standardne RNN

- U svakom trenutku t , ćelija dobija svoj ulaz x_t , ali takođe dobija i stanje ćelije iz prethodnog koraka h_{t-1}
- Ovo skriveno stanje iz prethodnog koraka se ponaša kao memorija

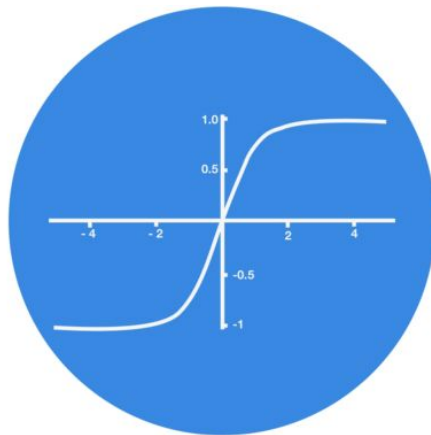




Standardne RNN

- Vektor ulaza i vektor skrivenog stanja iz prethodnog koraka se konkatenuiraju i provlače kroz **tanh**:
 - Na ovaj način se vrši regulisanje vrednosti izlaza unutar same neuronske mreže

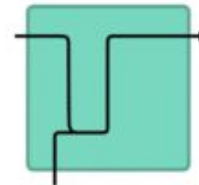
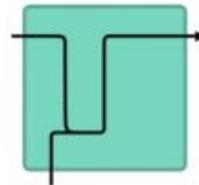
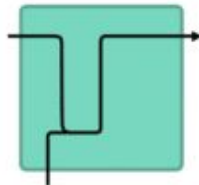
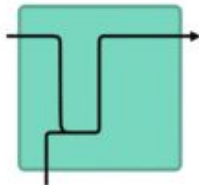
5
0.1
-0.5





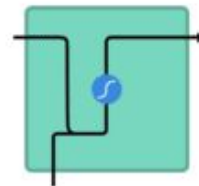
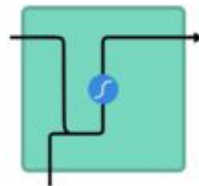
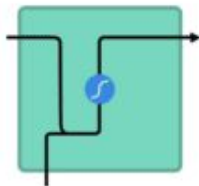
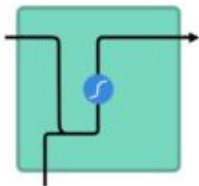
Standardne RNN

5
0.01
-0.5



Procesiranje bez **tanh**

5
0.01
-0.5



Procesiranje uz korišćenje **tanh**



Standardne RNN

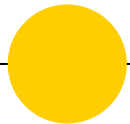
- Primer:
 - RNN from scratch



Standardne RNN

- Mane i problemi:
 - *Unrolled* mreža može biti ogromna (u zavisnosti koliko dugo treba da čuva informaciju):
 - Veoma duboka mreža
 - Teška za obučavanje (*vanishing/exploding gradient*)
 - Mreža nema dugoročnu memoriju

LSTM





LSTM

2

LSTM

Laughing-Silently-To-Myself. The REAL alternative to LOL, because really how often are you laughing out loud when you type LOL. Exactly. Not too often. And when you are laughing out loud, use LOL, otherwise: LSTM.

Sue: hey brad, guess what?

Brad: what?

Sue: I have Herpes :(

Brad: LSTM

Sue: That wasn't a joke!!

Brad: oh . . . sorry . . LOL!

#laughing #laugh #silent #real #lol #alternative

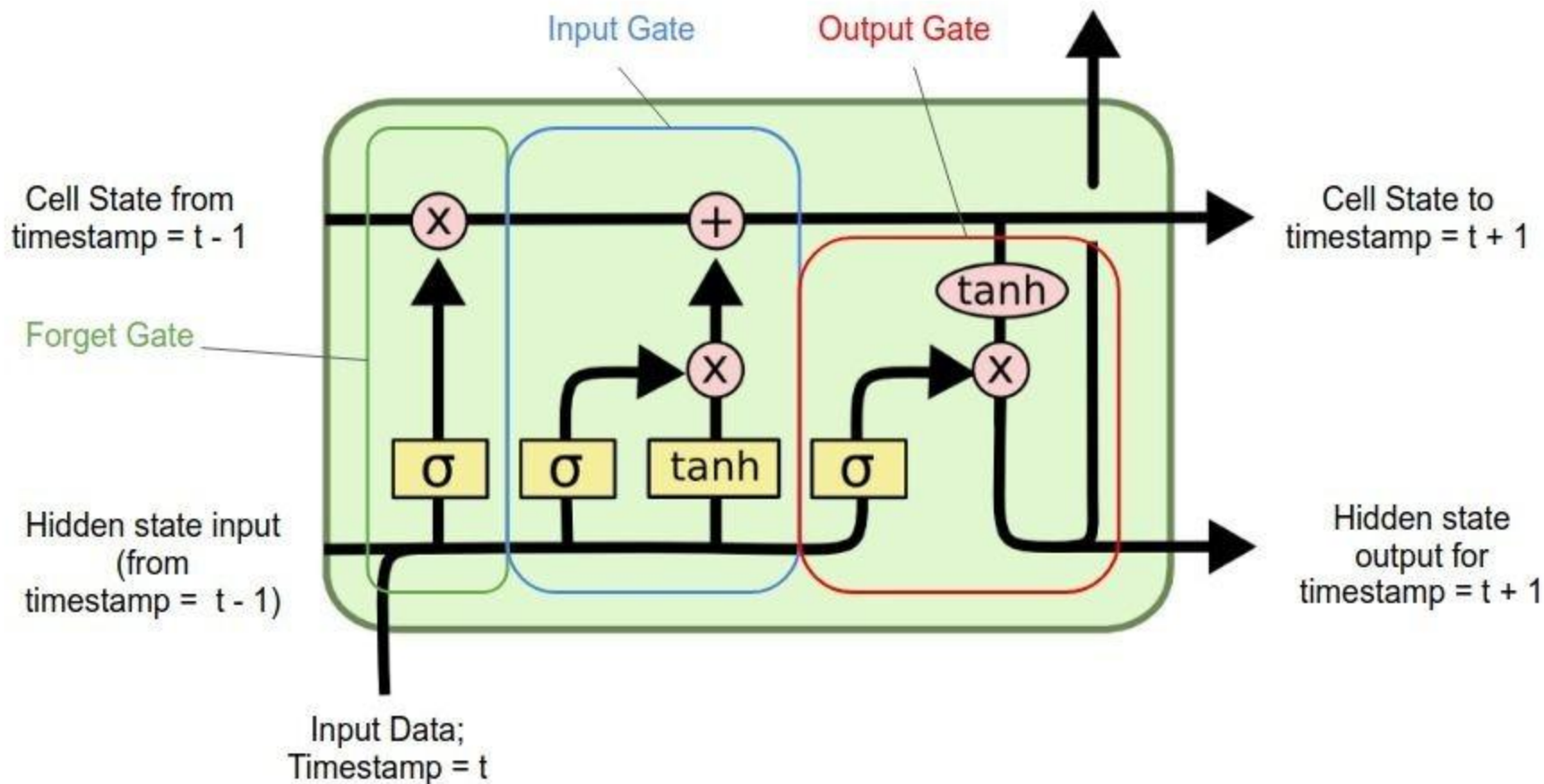
by **!jkm**? July 02, 2009





LSTM

- 3 nove komponente (*gates*):
 - **Input gate**
 - Da li prihvatiti ulaz?
 - **Cell state**
 - Koliko prihvatiti ulaz?
 - **Forget gate**
 - Da li izbrisati trenutno stanje/memoriju?
 - **Output gate**
 - Da li proslediti izlaz?



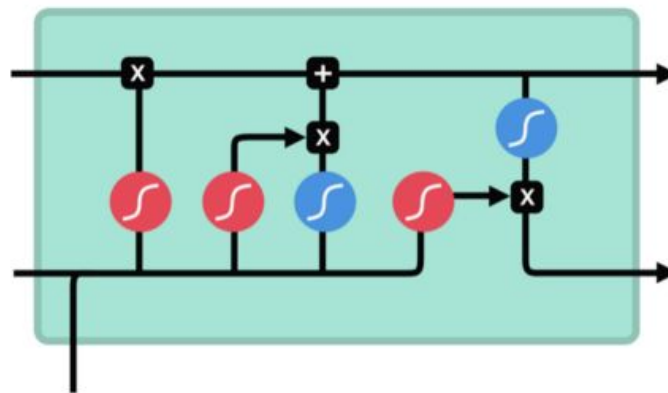


LSTM

- Sva 3 gejeta su zapravo obične logističke sigmoidalne funkcije
- Interno stanje (memorija) nije vidljivo “spolja”, već samo kroz aktivacionu funkciju.

● LSTM

- Koncepti:



sigmoid



tanh



pointwise
multiplication



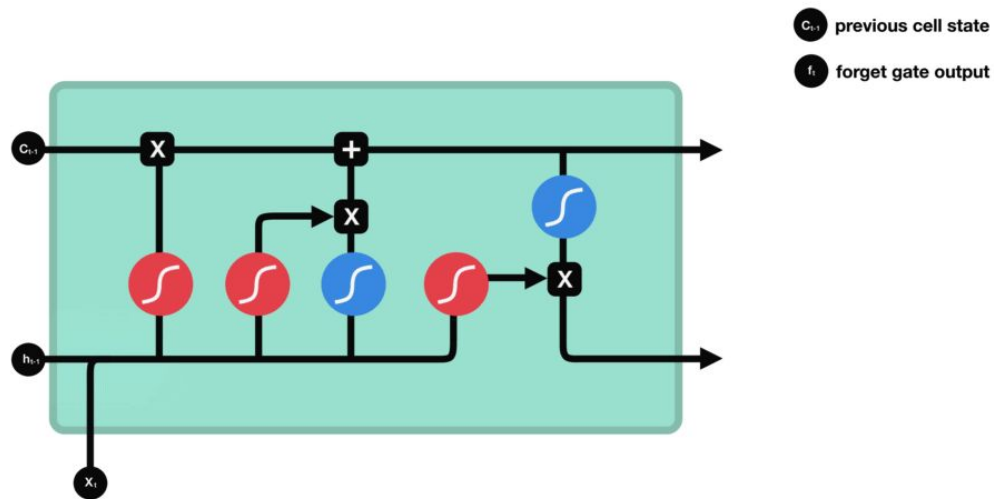
pointwise
addition



vector
concatenation

LSTM

- *Forget gate:*
 - Odlučuje koje informacije iz prethodnih koraka trebaju biti zaboravljene, a koje sačuvane

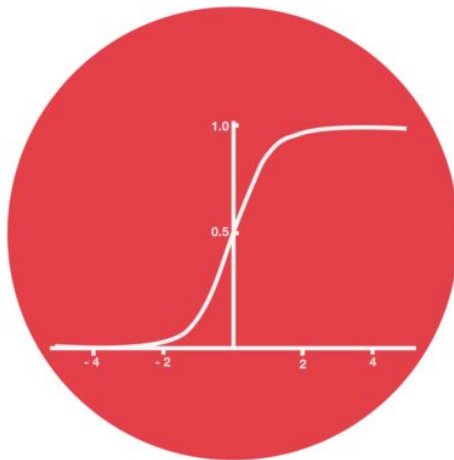




LSTM

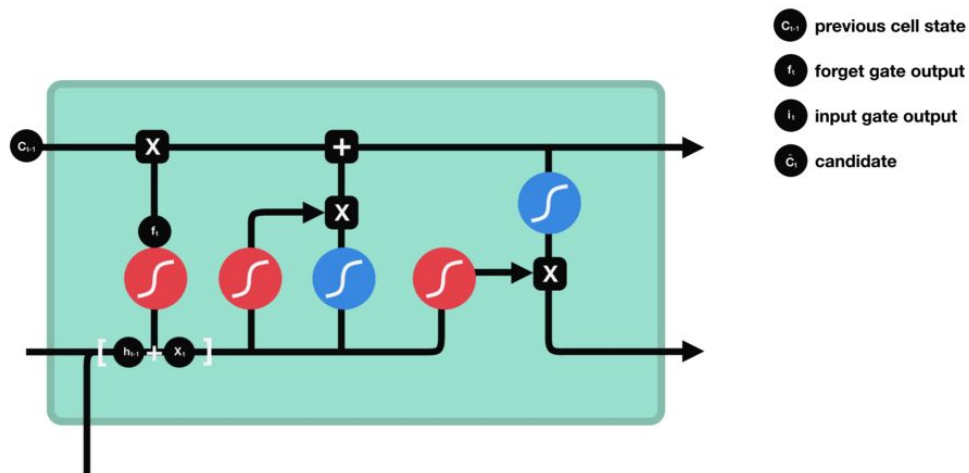
- *Forget gate:*
 - Koristi se sigmoidna funkcija:
 - Što je aktivacija bliža 0, ta informacija treba biti zaboravljena.

5
0.1
-0.5



LSTM

- *Input gate:*
 - Koristi se za osvežavanje stanja trenutne ćelije



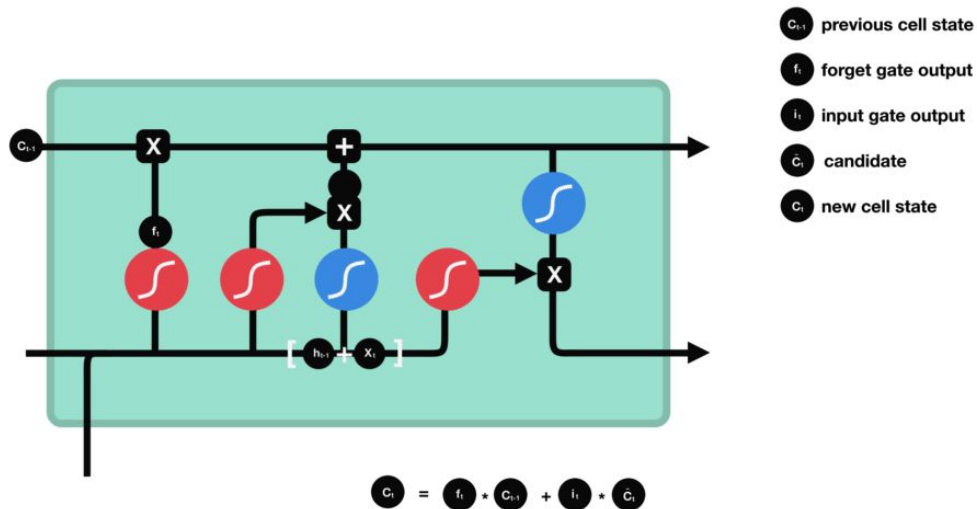


LSTM

- *Input gate:*
 - Sastoji se iz dva koraka:
 - Sigmoidalnim korakom se odlučuje koje informacije treba da budu ažurirane (0 - nebitne, 1 - bitne)
 - Hiperbolični tangens kreira kandidate za novo stanje trenutne ćelije
 - Rezultati ova dva koraka se množe:
 - Na taj način se dobija kandidat za novo stanje ćelije, gledajući i koje informacije su bitne, a koje ne.

LSTM

- *Cell state:*
 - Stanje ćelije se računa na osnovu rezultata iz *forget gate* i *input gate*



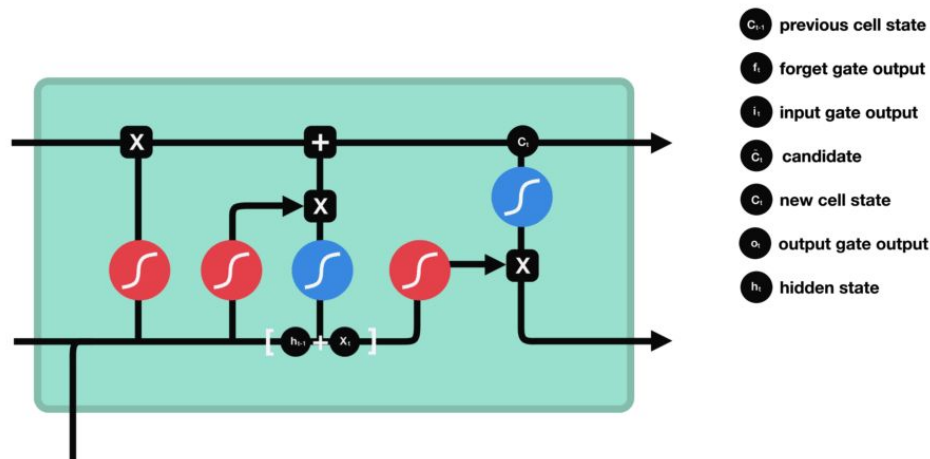


LSTM

- *Cell state*:
 - Stanje *forget gate*-a se množi sa stanjem iz prethodne ćelije (prethodnog vremenskog trenutka), da bi se izračunalo koje informacije iz prethodnih stanja treba sačuvati, a koje ne (*pointwise* množenje)
 - Vektor koji predstavlja rezultat prethodnog koraka (zaboravljanje) se sabira sa izlazom *input gate*-a:
 - Sabiranje je takođe *pointwise*
 - Rezultat ovog sabiranja je novi vektor stanja tekuće ćelije.

LSTM

- *Output gate:*
 - Određuje koje informacije izlaze iz *LSTM* ćelije i određuje skriveno stanje koje se prosleđuje u sledeću ćeliju sekvenci
 - Ovo stanje se koristi u predikciji.





LSTM

- *Output gate:*
 - Slično kao kod *input gate*-a, prethodno skriveno stanje i ulaz se konkatenuiraju i šalju na sigmoid funkciju, koja određuje važnost svih elemenata u tom vektoru
 - Novi vektor stanja ćelije (sa zaboravljenim informacijama - rezultat operacije sa *forget gate*-om) se provlači kroz tanh, da bi se vrednosti prebacile u interval $[-1, 1]$
 - Nakon toga se rezultat tanh funkcije množi sa rezultatom sigmoidalne funkcije, da bi se odredilo koje informacije treba proslediti u sledeću ćeliju.



LSTM

- Rezime:
 - *Forget gate* određuje šta je relevantno iz prethodnih koraka
 - *Input gate* određuje koje informacije su relevantne da se dodaju u trenutnom koraku
 - *Output gate* određuje kako će izgledati sledeće skriveno stanje
 - Ključ *LSTM*-a je u samom stanju ćelije (parametar C_t)

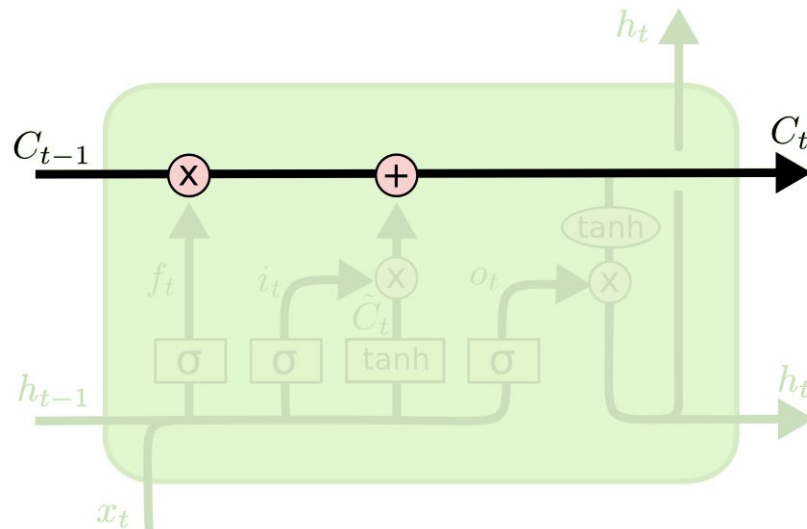
```
def LSTMCELL(prev_ct, prev_ht, input):
    combine = prev_ht + input
    ft = forget_layer(combine)
    candidate = candidate_layer(combine)
    it = input_layer(combine)
    Ct = prev_ct * ft + candidate * it
    ot = output_layer(combine)
    ht = ot * tanh(Ct)
    return ht, Ct
```

```
ct = [0, 0, 0]
ht = [0, 0, 0]
```

```
for input in inputs:
    ct, ht = LSTMCELL(ct, ht, input)
```

1. Prethodno skriveno stanje i trenutni ulaz se kombinuju.
2. Kombinacija se šalje u sloj za zaborav (kako bi uklonio nerelevantne podatke).
3. Sloj kandidata je kreiran korišćenjem kombinacije. Kandidat ima moguće vrednosti za dodavanje stanju ćelije.
4. Kombinacija se takođe šalje i u ulazni sloj. Ovaj sloj odlučuje koje podatke kandidata treba dodati u novo stanje ćelije.
5. Nakon izračunavanja sloja za zaborav, sloja kandidata i ulaznog sloja, stanje ćelije se izračunava korišćenjem tih vektora i prethodnog stanja ćelije.
6. Zatim se računa izlaz.
7. *Pointwise* množenje izlaza i novog stanja ćelije nam daje novo skriveno stanje.

LSTM

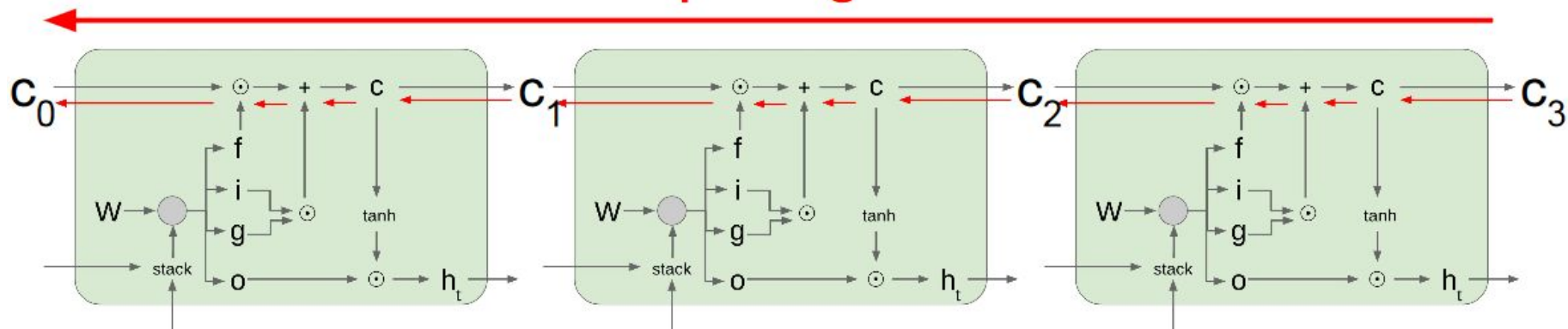


Novo stanje se izračunava na osnovu prethodnog stanja i na osnovu minornih linearnih interakcija koje grana ima sa gejtovima (gejtovi opciono određuju koje informacije da propuste dalje).

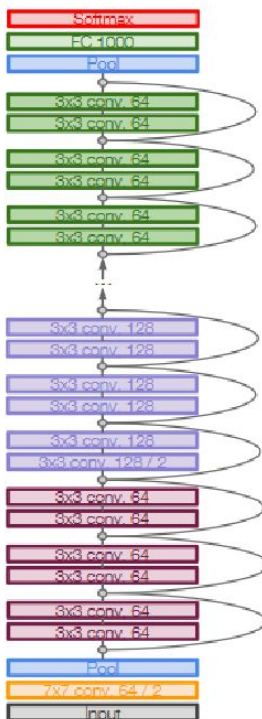
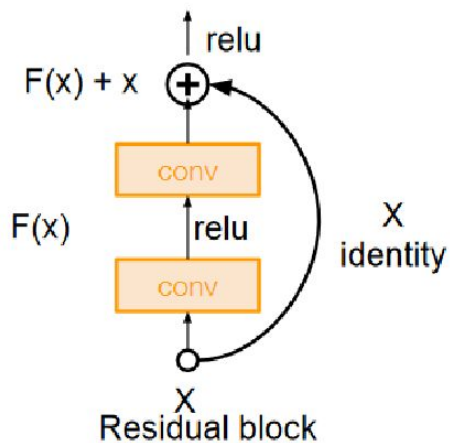


LSTM

Uninterrupted gradient flow!



LSTM



Revolution of Depth

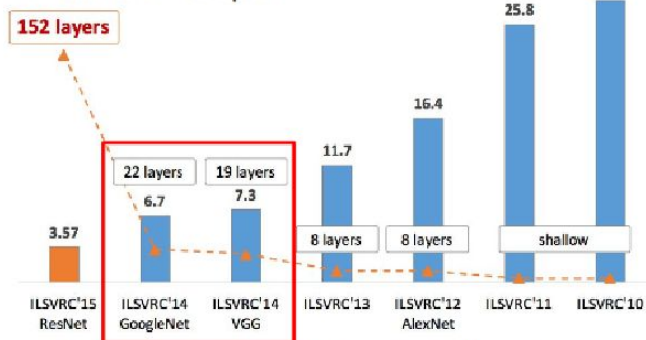


Figure copyright Kaiming He, 2016. Reproduced with permission.

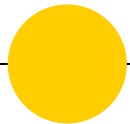


LSTM

When you don't use LSTM
for a long sequence RNN

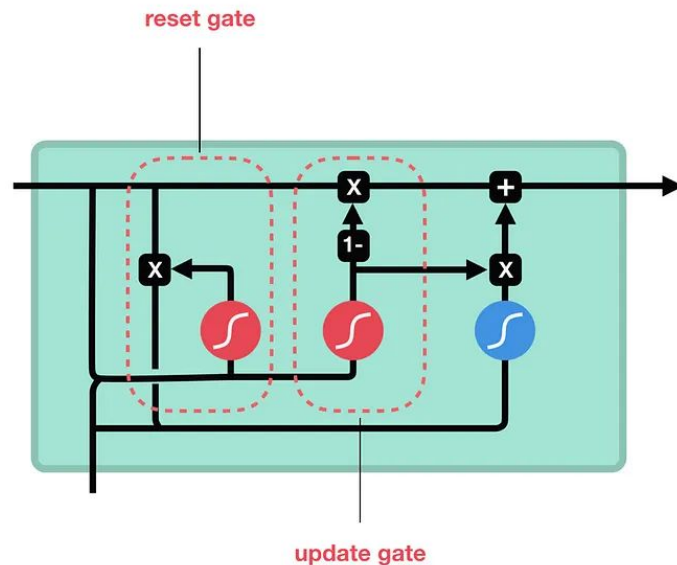


GRU



GRU

- Modifikacija *LSTM*
- Kombinovanje *input gate* i *forget gate* => **update gate** i **reset gate**:
 - **Reset gate** određuje koliko novi ulaz utiče na trenutno stanje
 - **Update gate** određuje koliko prethodno stanje utiče na trenutno stanje





GRU

- Nema *output gate*
- Nema aktivacionu funkciju nad memorijom:
 - Interno stanje vidljivo “spolja”



GRU

- *LSTM vs GRU:*
 - *LSTM* ima 3 gejtta, *GRU* samo 2:
 - *GRU* ima manje parametara
 - *LSTM* predložen 1997., *GRU* 2014. godine

GRU

- *LSTM vs GRU:*
 - Nad manjim skupovima podataka:

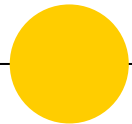




GRU

- *LSTM vs GRU:*
 - Ako je skup podataka veći, *LSTM* će verovatno dati bolje rezultate.

Primeri





Primeri

- Primene:
 - Puno uspeha u *NLP (Natural Language Processing)*:
 - Generisanje teksta, govora, muzike...
 - *Neural Machine Translation*
 - Prepoznavanje govora
 - *Text-to-speech*
 - ...

Primeri

- Primer klasifikacije sekvence:

What time is it?

Tokenizacija rečenice na reči

Primeri

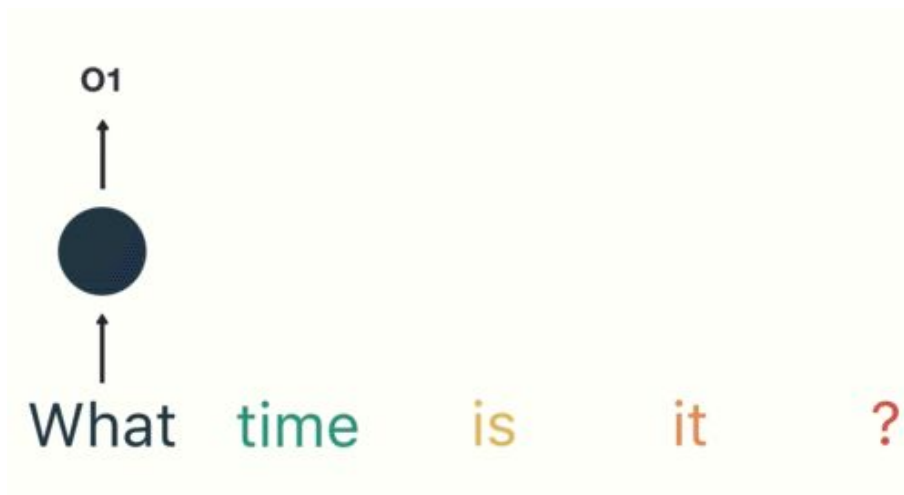
- Primer klasifikacije sekvence:

What time is it ?

Dovođenje prve reči na *RNN*. *RNN* kodira ulaz i vraća rezultat.

● Primeri

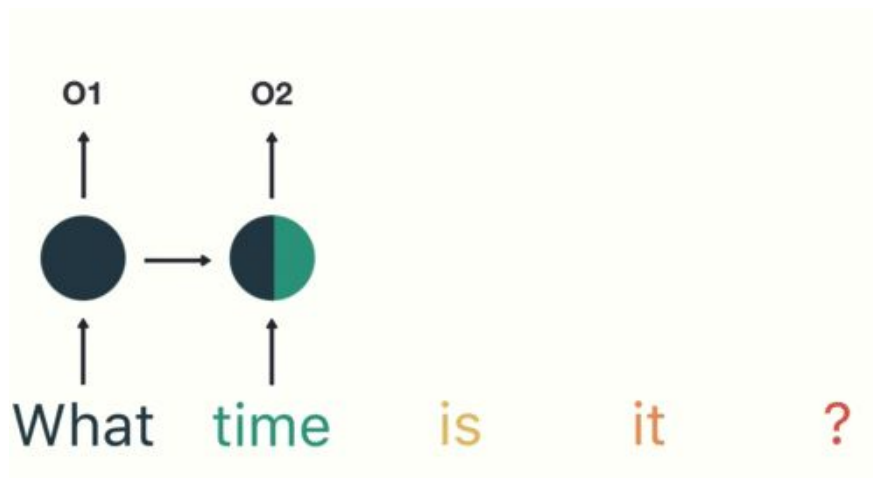
- Primer klasifikacije sekvence:



Dovođenje druge reči na *RNN*. Pored druge reči, dovodi se i skriveno stanje iz prethodnog koraka.
RNN ima transformaciju i "What" i "time".

● Primeri

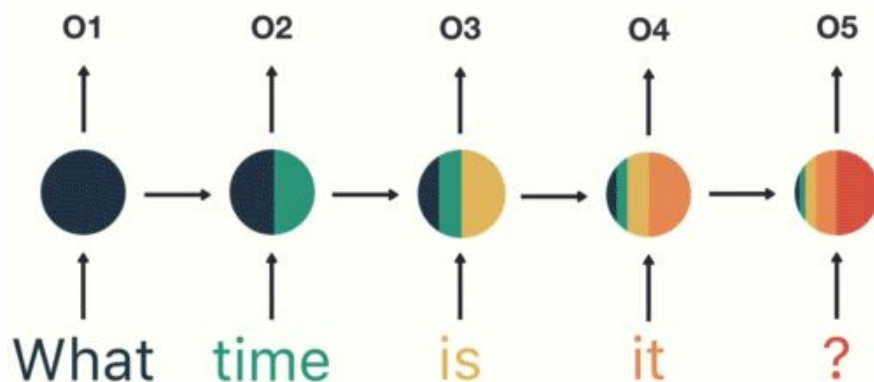
- Primer klasifikacije sekvence:



Proces se ponavlja do kraja sekvence. Rezultat je kodiranje cele ulazne sekvence reči.

● Primeri

- Primer klasifikacije sekvence:



Pošto se ovde koristi obična *RNN*, a ne *LSTM* ćelija, može se primetiti kako jako davna stanja gube na značaju kako sekvenca postaje duža (obratiti pažnju na boje).

LSTM bi putem svojih gejtova zaboravljao informacije koje nisu bitne za klasifikaciju i fokusirao se na one koje to jesu, i radio bi bolje na dužim sekvencama.

Pošto vektor **O5** sadrži kodirane informacije o celoj sekvenci, njega možemo poslati na neki klasifikator (recimo običan MLP) da bi izvršili klasifikaciju sekvence.



Primeri

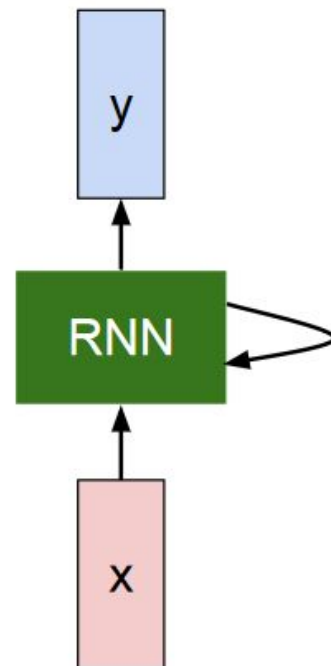
- Primer generisanja teksta:
 - Generisanje teksta na osnovu drugog teksta
 - Generisanje izvornog koda

THE SONNETS

by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the ripper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own buduriest thy content,
And tender churl mak'st waste in niggarding:
Pity the world, or else this glutton be,
To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
This were to be new made when thou art old,
And see thy blood warm when thou feel'st it cold.



at first:

tyntd-iafhatawiaoihrdemot lytdws e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrqd t o idoe ns,smtt h ne etie h,hregtrs nigtike,aoaenns lng



train more

"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuw y fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."



train more

Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.



train more

"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.

```

static void do_command(struct seq_file *m, void *v)
{
    int column = 32 << (cmd[2] & 0x80);
    if (state)
        cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
    else
        seq = 1;
    for (i = 0; i < 16; i++) {
        if (k & (1 << i))
            pipe = (in_use & UMXTHREAD_UNCCA) +
                ((count & 0x00000000ffffffff8) & 0x000000f) << 8;
        if (count == 0)
            sub(pid, ppc_md.kexec_handle, 0x20000000);
        pipe_set_bytes(i, 0);
    }
    /* Free our user pages pointer to place camera if all dash */
    subsystem_info = &of_changes[PAGE_SIZE];
    rek_controls(offset, idx, &soffset);
    /* Now we want to deliberately put it to device */
    control_check_polarity(&context, val, 0);
    for (i = 0; i < COUNTER; i++)
        seq_puts(s, "policy ");
}

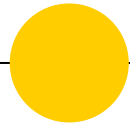
```



Primeri

- Primer:
 - Data acquisition
 - RNN Generator

Vizualizacija obučениh ćelija





Vizualizacija obučениh ćelija

- Ideja jeste prikazivanje delova teksta za koje se određene ćelije specijalizuju
- To rezultuje visokim aktivacijama ćelije kada naiđe na takav deo u tekstu
- Ovo je težak zadatak, pošto je interna reprezentacija obično visokodimenzionalna i distribuirana
- Obično je samo nad oko 5% ćelija moguće izvršiti nešto slično.



Vizualizacija obučениh ćelija

Cell sensitive to position in line:

The sole importance of the crossing of the Berezina lies in the fact that it plainly and indubitably proved the fallacy of all the plans for cutting off the enemy's retreat and the soundness of the only possible line of action--the one Kutuzov and the general mass of the army demanded--namely, simply to follow the enemy up. The French crowd fled at a continually increasing speed and all its energy was directed to reaching its goal. It fled like a wounded animal and it was impossible to block its path. This was shown not so much by the arrangements it made for crossing as by what took place at the bridges. When the bridges broke down, unarmed soldiers, people from Moscow and women with children who were with the French transport, all--carried on by vis inertiae--pressed forward into boats and into the ice-covered water and did not, surrender.



Vizualizacija obučениh ćelija

Cell that turns on inside quotes:

"You mean to imply that I have nothing to eat out of.... On the contrary, I can supply you with everything even if you want to give dinner parties," warmly replied Chichagov, who tried by every word he spoke to prove his own rectitude and therefore imagined Kutuzov to be animated by the same desire.

Kutuzov, shrugging his shoulders, replied with his subtle penetrating smile: "I meant merely to say what I said."



Vizualizacija obučениh ćelija

Cell that robustly activates inside if statements:

```
static int __dequeue_signal(struct sigpending *pending, sigset_t *mask,
                           siginfo_t *info)
{
    int sig = next_signal(pending, mask);
    if (sig) {
        if (current->notifier) {
            if (sigismember(current->notifier_mask, sig)) {
                if (!(current->notifier)(current->notifier_data)) {
                    clear_thread_flag(TIF_SIGPENDING);
                    return 0;
                }
            }
        }
        collect_signal(sig, pending, info);
    }
    return sig;
}
```



Vizualizacija obučениh ćelija

Cell that turns on inside comments and quotes:

```
/* duplicate LSM field information. The lsm_rule is opaque, so
 * re-initialized. */
static inline int audit_dupe_lsm_field(struct audit_field *df,
                                       struct audit_field *sf)
{
    int ret = 0;
    char *lsm_str;
    /* our own copy of lsm_str */
    lsm_str = kstrdup(sf->lsm_str, GFP_KERNEL);
    if (unlikely(!lsm_str))
        return -ENOMEM;
    df->lsm_str = lsm_str;
    /* our own (refreshed) copy of lsm_rule */
    ret = security_audit_rule_init(df->type, df->op, df->lsm_str,
                                   (void *)&df->lsm_rule);
    /* Keep currently invalid fields around in case they
     * become valid after a policy reload. */
    if (ret == -EINVAL) {
        pr_warn("audit rule for LSM '%s' is invalid\n",
                df->lsm_str);
        ret = 0;
    }
    return ret;
}
```



Vizualizacija obučениh ćelija

Cell that is sensitive to the depth of an expression:

```
#ifdef CONFIG_AUDITSYSCALL
static inline int audit_match_class_bits(int class, u32 *mask)
{
    int i;
    if (classes[class]) {
        for (i = 0; i < AUDIT_BITMASK_SIZE; i++)
            if (mask[i] & classes[class][i])
                return 0;
    }
    return 1;
}
```



Vizualizacija obučениh ćelija

- Međutim, većina ćelija se teško može protumačiti ljudskim rezonovanjem
- To ne mora značiti da ove ćelije ne doprinose rešavanju problema zbog koga se mreža obučavala
 - Kombinacijom ovakvih ćelija, neuronska mreža nauči da rešava problem u svom apstraktnom, distribuiranom i visokodimenzionalnom prostoru.

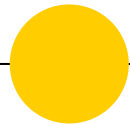


Vizualizacija obučениh ćelija

A large portion of cells are not easily interpretable. Here is a typical example:

```
/* Unpack a filter field's string representation from user-space
 * buffer. */
char *audit_unpack_string(void **bufp, size_t *remain, size_t len)
{
    char *str;
    if (!*bufp || (len == 0) || (len > *remain))
        return ERR_PTR(-EINVAL);
    /* Of the currently implemented string fields, PATH_MAX
     * defines the longest valid length.
     */
}
```

Dodatno čitanje





Dodatno čitanje

- *Introduction to RNN*
- *The Unreasonable Effectiveness of Recurrent Neural Networks*
- *Understanding LSTM Networks*
- *Illustrated Guide to LSTM's and GRU's: A step by step explanation*

Hvala na pažnji!

Pitanja?