Razvoj rekurentne neuronske mreže i primena na analizi vremenskih serija

Seminarski rad u okviru kursa
Računarska inteligencija

Kristina Pantelić, 91/2016, mi16091@matf.bg.ac.rs
Nevena Mesar, 107/2015, mi15107@matf.bg.ac.rs

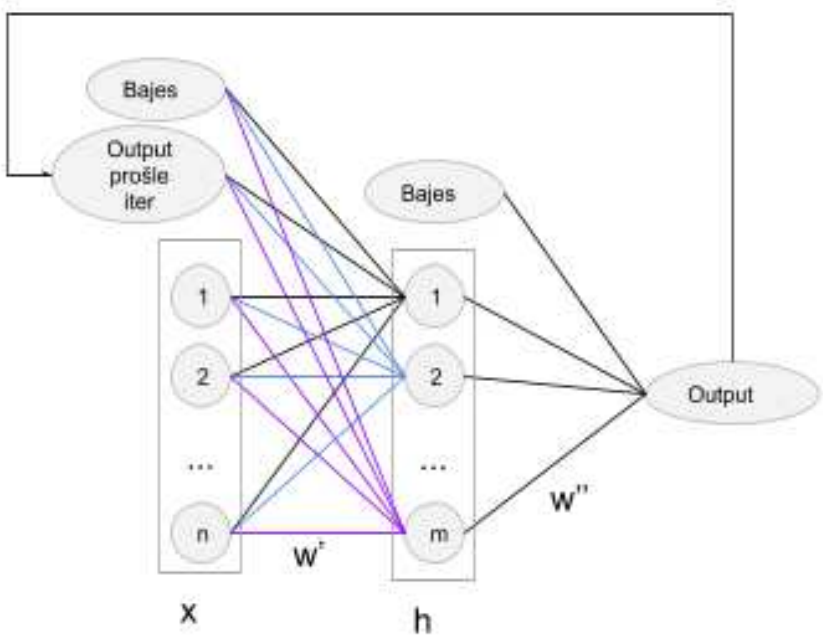
Matematički fakultet

20.6.2020.

Uvod

- Tradicionalna neuronska mreža
  - ulazi i izlazi nezavisni jedni od drugih
- Rekurentna neuronska mreža
  - svojstvo pamćenja naučenog znanja iz prethodnih trening instanci
  - omogućava predikciju u oblasti vremenskih serija
- Jordanova rekurentna neuronska mreža (eng. *Jordan SRNN*)
  - kopija izlaznog sloja se sprovodi na ulaz

Jordan SRNN



Legenda:

x - skup ulaznih čvorova
h - skup čvorova skrivenog sloja
Output - količina padavina za 7. dan
Output prošle iter - kopija izlaznog sloja
w' - težine veza ka skrivenom sloju
w'' - težine veza ka izlaznom sloju
Bajes - uvek jednak 1

Figure: Jordanova SRNN sa jednim skrivenim slojem

Model RNN-a

- **Aktivaciona funkcija** skrivenog i izlaznog sloja:

$$f(x) = (1 + e^{-x})^{-1}$$

- Greška izlaznog sloja neurona k:

$$E_k = \frac{1}{2}(y_k - o_k)^2$$

- Pri ažuriranju vrednosti w''_{jk}, važi $w''_{jk} = w''_{jk} + \Delta w''_j k$, gde je

$$\Delta w''_{jk} = -\eta \frac{\partial E_k}{\partial w''_{jk}} + \alpha \Delta w''_{jk}$$

η uticaj parcijalnog izvoda greške E_k po w''_{jk}
α uticaj prethodne vrednosti $\Delta w''_{jk}$

Algoritam

1. Ulazni podaci $(x_1^{(l)}, x_2^{(l)}, \ldots, x_{n+p}^{(l)})$ i $(y_1, y_2, \ldots, y_p)$, $x_0^{(l)} = 1, \forall l$ iz skupa podataka
2. Init η, α i kriterijum zaustavljanja
   Init w'_{ij} i w''_{jk} i $\Delta w'_{ij} = \Delta w''_{jk} = 0$
3. Novi par ulaznog i izlaznog vektora
4. Odrediti u'_j i h_j. Postaviti $h_0^{(l)} = 1, \forall l$ iz skupa podataka
5. Odrediti u''_k i o_k. Ukoliko je ispunjen kriterijum zaustavljanja, prekinuti izvršavanje.
6. Odrediti $\Delta w''_{jk}$ i ažurirati vrednosti w''_{jk}.
7. Odrediti $\Delta w'_{ij}$ i ažurirati vrednosti w'_{ij}.
8. Preći na korak 3.

Implementacija algoritma

Potrebne biblioteke

- `numpy` – zeros, array, append, concatenate, multiply, vstack, matrix, around, random
- `pandas` – DataFrame, Series, read_csv, errors, concat
- `sklearn`
  - metrics – mean_absolute_error, mean_squared_error
  - model_selection – train_test_split
  - preprocessing – MinMaxScaler
- `matplotlib` – pyplot

Izvor podataka

Basel, Švajcarska
31.12.1990. do 31.12.2019

Implementacija algoritma

Predprocesiranje

- `test : train` = 3 : 7
  test – 3398, train – 7928
- `sklearn.model_selection` – `test_train_split`
- `MinMaxScaler` – [0.1, 0.9]
- precipitation
  100→ [0.1, 0.9]

Inicijalizacija

- Broj dana za predikciju: `N`
- Ulazni čvorovi:
  `n = N * n_attrs + 1`
- Patern: 6 dana + output
- Broj paterna:
  `len(x_train) - N`
- Broj atributa za jedan dan:
  `n_attrs`
- Čvorovi skrivenog sloja:
  `m = (int)((N * n_attrs) * 4) + 1`
- `eta = 0.3; alpha = 0.2`

Implementacija algoritma – paterni

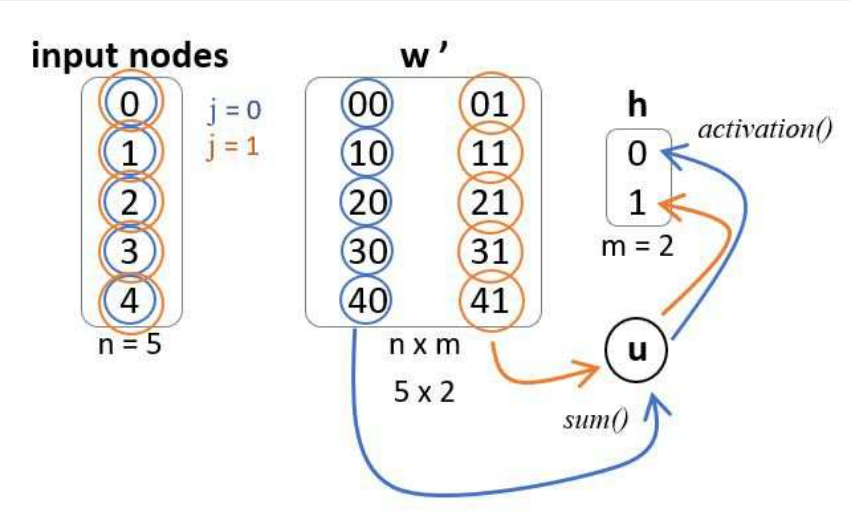
1	2	3	4	5	6	→	7
2	3	4	5	6	7	→	8
... ...							
7923	7924	7925	7926	7927	7928	→	7929

- Svaka ćelija je skup atributa za jedan dan
- Jedan patern, `input_nodes`, predstavlja 6 dana zaredom

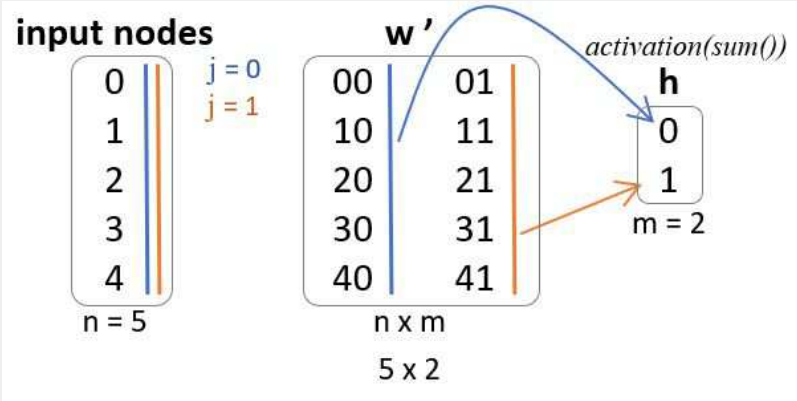
```
concatenate((bias,
             (x_train[day:(day+N)]).reshape(-1),
             output_arr))
```

Izračunavanje h1,..., hm čvorova skrivenog sloja

```
# Naivan pristup
h[0] = 1.0
for j in range(1, m+1):
    u = 0
    for i in range(0, n+1):
        u += input_nodes[i]
             * w_[i][j]
    h[j] = activation_f(u)
```




```
# Poboljsan pristup
u = []
for i in w_.T:
    u += [activation_f(
            sum(multiply(
                i,
                input_nodes))
          )]
h = array(u)
h[0] = 1
```



Izračunavanje o1,...,op i greške

Računanje izlaznog čvora

```
# Naivna implementacija
for k in range(1, p+1):
    u = 0
    for j in range(0, m+1):
        u += h[j] * w__[j][k]
    o[day] = activation_f(u)

# Poboljsana implementacija
o[day] = activation_f(sum(multiply(h, w__)))
output_arr = array([o[day]])
```

Računanje greške

```
error = (y_train[day+N] - o[day]) * o[day] *
(1.0 - o[day])
```

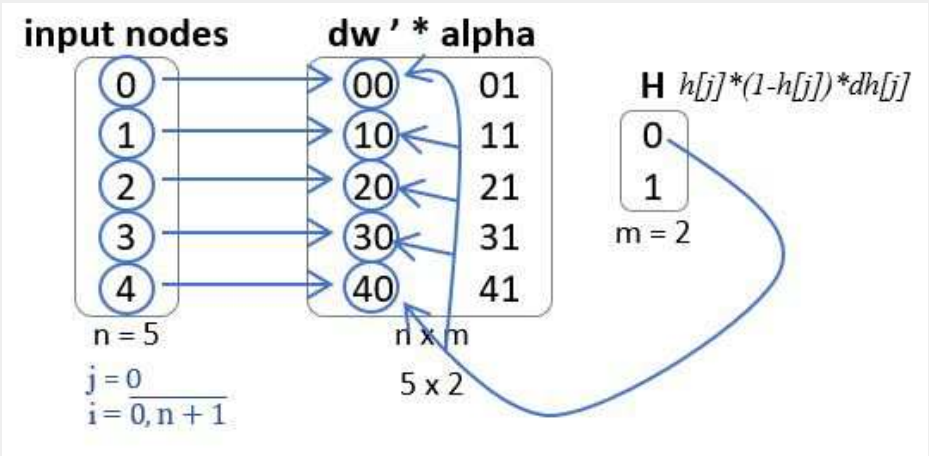
Izračunavanje deltaH(j)

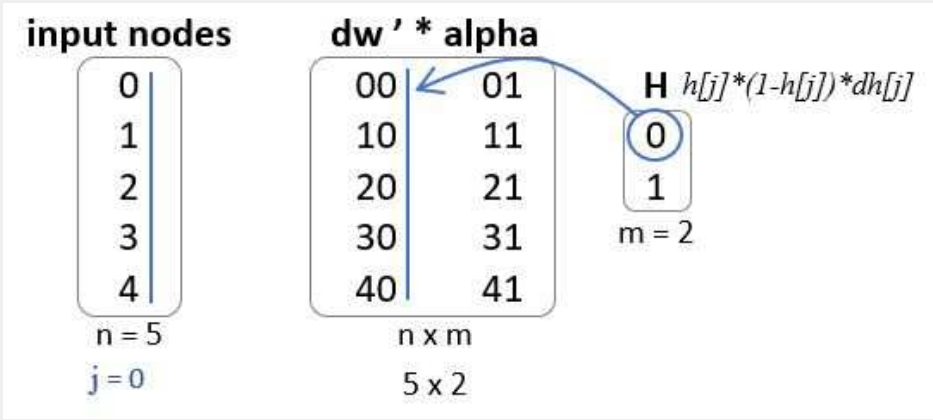
```
# Naivna implementacija
for j in range(1, m+1):
    dh[j] = 0.0
    for k in range(1, p+1):
        dh[j] += w__[j][k] * error

# Poboljsana implementacija
dh = w__ * error
```

Ažuriranje W'(ij) i deltaW'(ij)

```
# Naivna implementacija
for j in range(1, m+1):
    H = h[j] * (1 - h[j]) * dh[j]
    for i in range(0, n+1):
        dw_[i][j] = eta * input_nodes[i] * H
                    + alpha * dw_[i][j]
        w_[i][j] += dw_[i][j]
```

```
# Poboljsana implementacija
H = h * (1-h) * dh * eta
dw_ *= alpha
for j in range(0, m):
    dw_.T[j] += (input_nodes * H[j])
w_ += dw_
```





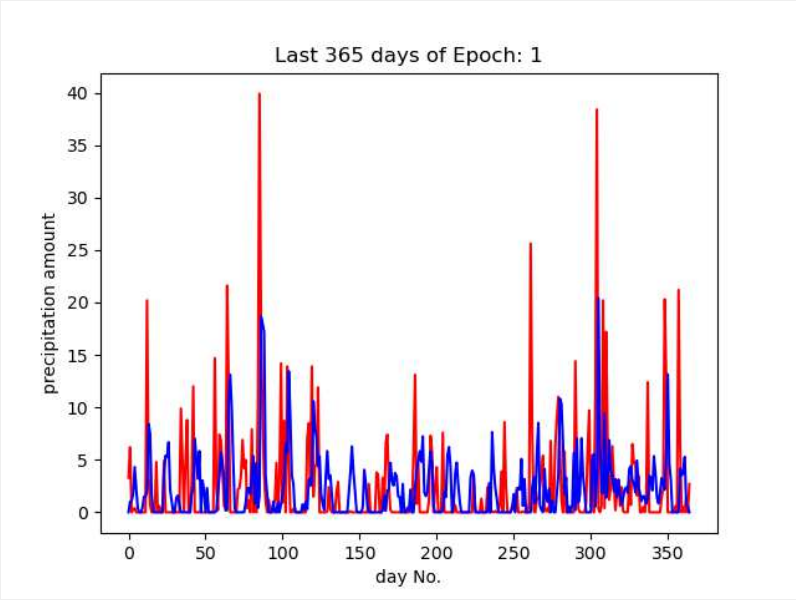
Ažuriranje W”(jk) i deltaW”(jk)

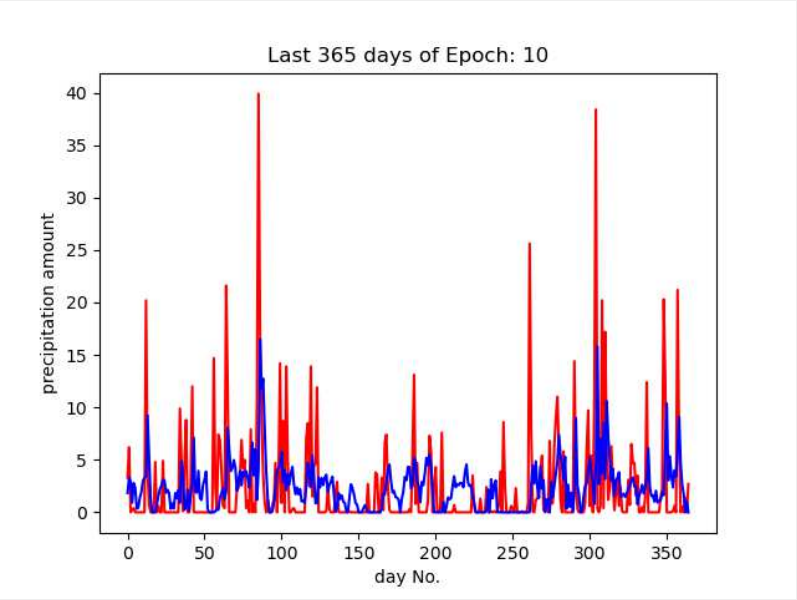
```
# Naivna implementacija
for k in range(1, p):
    for j in range(0, m):
        dw__[j][k] = eta * h[j] * error
                   + alpha * dw__[j][k]
        w__[j][k] += dw__[j][k]


# Poboljsana implementacija
dw__ *= alpha
dw__ += (eta * h * error)
w__ += dw__
```

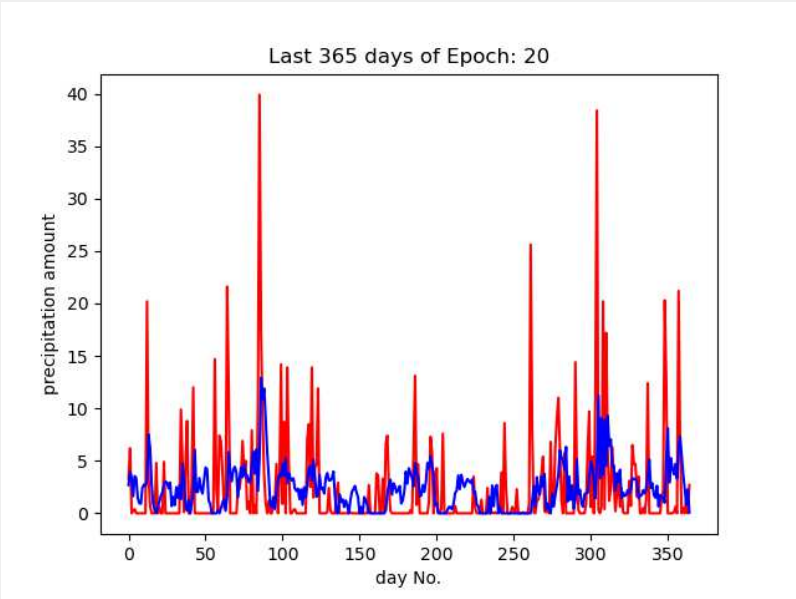
Model n = 54, m = 270, alpha = 0.5, eta = 0.3

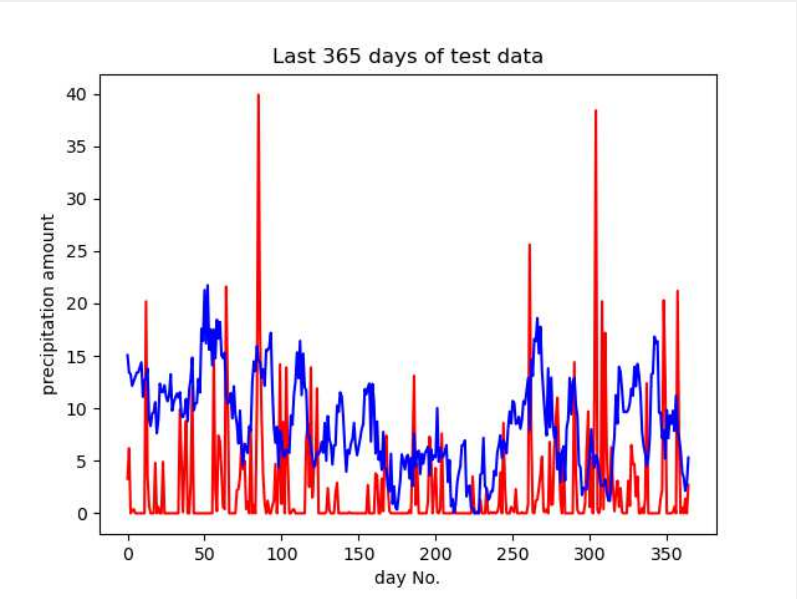
```
broj dana za predikciju= 6; broj ulaznih cvorova= 54; broj cvorova skriveni sloj= 270
alpha = 0.5 eta = 0.3
epoch:   0/20   mse: 0.00230558769024192      mae: 0.030745924289642186      rmse: 0.048016535591834615
epoch:   1/20   mse: 0.0020691819015563008    mae: 0.02903043113699024       rmse: 0.04548826114017001
epoch:   2/20   mse: 0.001938548994169053     mae: 0.027713894391377336      rmse: 0.04402895631478281
epoch:   3/20   mse: 0.0018782308177129618    mae: 0.02718125800641104       rmse: 0.04333856040194416
epoch:   4/20   mse: 0.0018541484309476885    mae: 0.02706152932361909       rmse: 0.04305982386108527
epoch:   5/20   mse: 0.0018472157254949184    mae: 0.02711965092474976       rmse: 0.04297924761434195
epoch:   6/20   mse: 0.001847096882086492     mae: 0.027240328981882785      rmse: 0.04297786502476004
epoch:   7/20   mse: 0.0018484149606793445    mae: 0.02736813761528103       rmse: 0.04299319667900195
epoch:   8/20   mse: 0.001848603137872986     mae: 0.027468230387172166      rmse: 0.04299538507645891
epoch:   9/20   mse: 0.0018466459437720528    mae: 0.027535487234559175      rmse: 0.04297261853520277
epoch:  10/20   mse: 0.0018423689617472       mae: 0.027561658799244805      rmse: 0.04292282564961445
epoch:  11/20   mse: 0.0018360429200479063    mae: 0.027550815717757516      rmse: 0.042849071402399214
epoch:  12/20   mse: 0.0018281416207006791    mae: 0.027509304146725484      rmse: 0.0427567728050268
epoch:  13/20   mse: 0.0018191847531010947    mae: 0.02745052855455493       rmse: 0.042651902104139446
epoch:  14/20   mse: 0.0018096434011468487    mae: 0.02737767645017048       rmse: 0.04253990363349274
epoch:  15/20   mse: 0.0017998957765097856    mae: 0.0272990199920374        rmse: 0.042425178567800814
epoch:  16/20   mse: 0.0017902188217654774    mae: 0.027220601868053973      rmse: 0.04231097755624984
epoch:  17/20   mse: 0.0017808003071023023    mae: 0.027141672344482493      rmse: 0.042199529702382964
epoch:  18/20   mse: 0.0017717585383886071    mae: 0.02706356552362999       rmse: 0.042092262215146
epoch:  19/20   mse: 0.0017631615172861157    mae: 0.02698921856530739       rmse: 0.041990016876468576
epoch:  20/20   mse: 0.001755042129148845     mae: 0.026918010213622875      rmse: 0.041893222950124584
Cuvanje modela: model10_34_23.txt
Uspesno cuvanje modela...
Da li zelite da ucitate neki od postojecih modela? Inace ce se raditi sa trenutnim. (Y/N)
n
MSE - test: 0.007809762722696304      MAE - test: 0.06803416142550417
```

Model n = 54, m = 270, alpha = 0.5, eta = 0.3



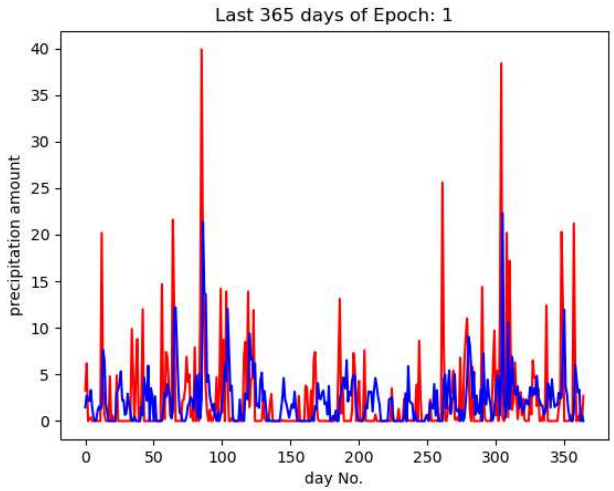




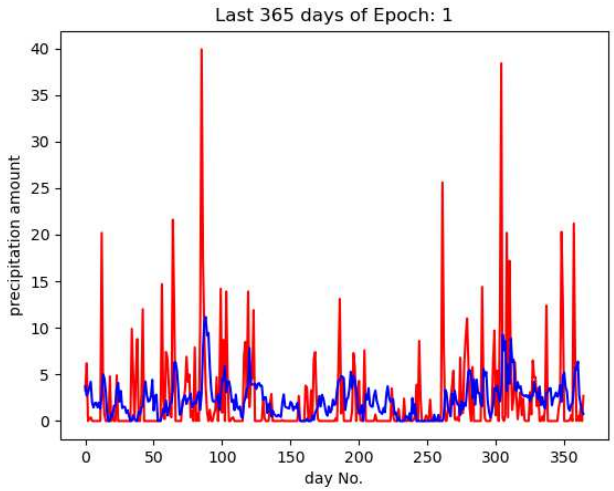


```
mse: 0.0020269406544871807
mse: 0.001890504081243114
mse: 0.0018196302392409823
mse: 0.0017822232490548485
mse: 0.0017627793303375927
mse: 0.0017530898416978208
mse: 0.0017484690989736473
mse: 0.0017462094607348239
mse: 0.0017447740486098808
mse: 0.0017433371103788757
mse: 0.001741505525559831 3
mse: 0.0017391416649945803
mse: 0.001736249819904523
mse: 0.0017329051905294111
mse: 0.0017292112152060485
mse: 0.0017252751760931007
mse: 0.0017211952721594226
mse: 0.001717054744560637
mse: 0.0017129202224039387
mse: 0.0017088424555715819
mse: 0.0017048582457663906
a = 0.2 e = 0.3 m = 270
```

```
mse: 0.0018333969136350868
mse: 0.0016754780687101315
mse: 0.0016590487256534369
mse: 0.0016544457597040723
mse: 0.0016543808344862826
mse: 0.001655983693567097
mse: 0.001657923819381214
mse: 0.00165959587624372455
mse: 0.0016607316316972973
mse: 0.00166129863196318 5
mse: 0.0016613203279609002
mse: 0.0016608669737958744
mse: 0.0016600215298744568
mse: 0.0016588664753508535
mse: 0.0016574772979276228
mse: 0.0016559196408595365
mse: 0.0016542485113777163
mse: 0.0016525086452444057
mse: 0.0016507354798691062
mse: 0.0016489563946454236
mse: 0.001647192008298528
a = 0.2 e = 0.3 m = 108
```

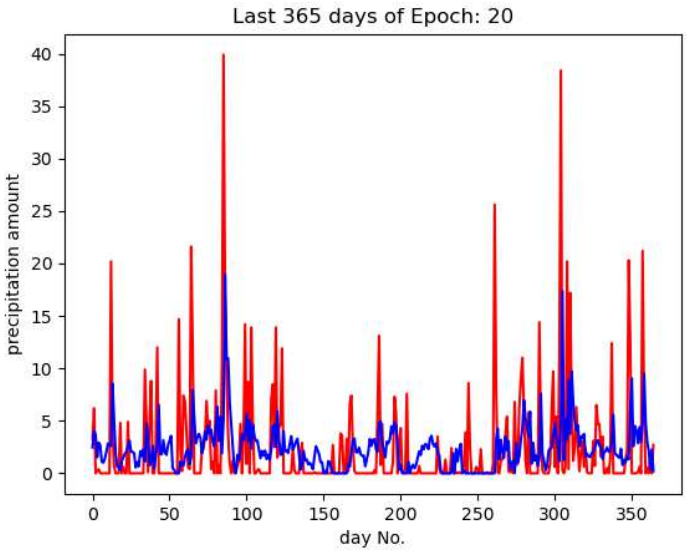


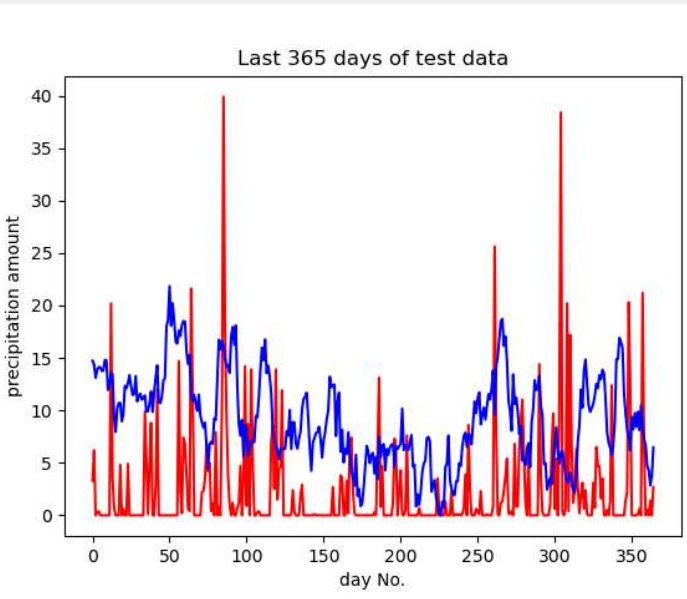
m = 270, α = 0.2, η = 0.3



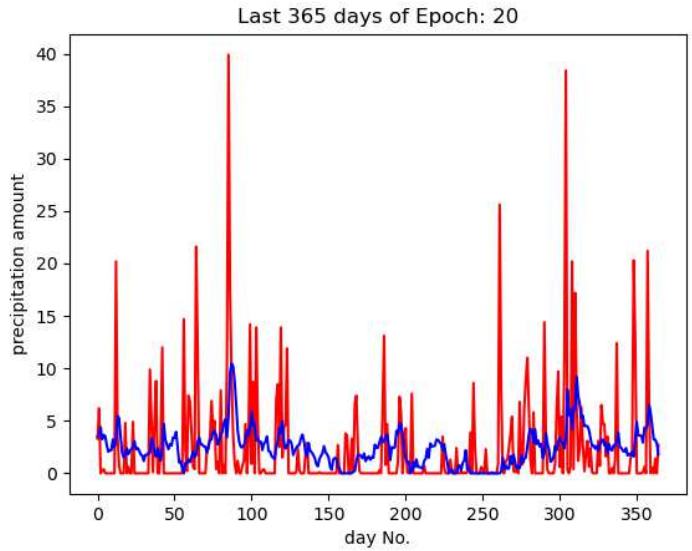
m = 108, α = 0.2, η = 0.3

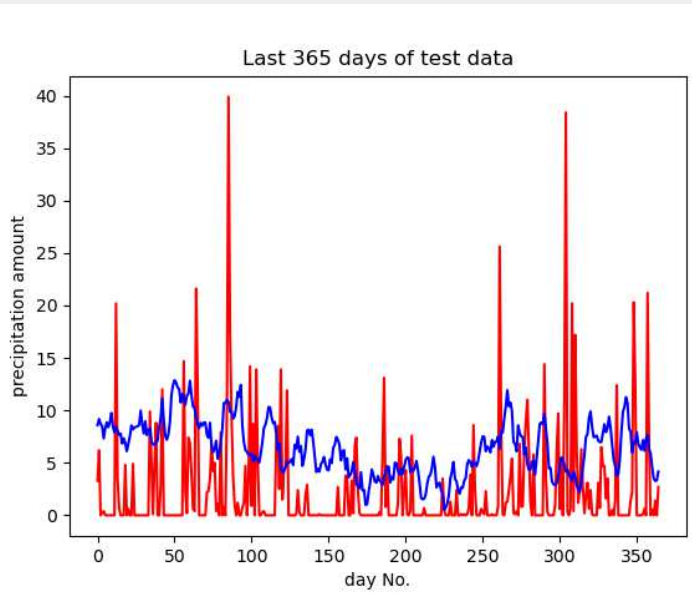
M = 270, α = 0.2, η = 0.3





M = 108, α = 0.2, η = 0.3





Zaključak

References

Model rekurentne neuronske mreže.
on-line at: http://poincare.matf.bg.ac.rs/~stefan/ri/index.htm.

Uvod u RNN.
on-line at: https://www.geeksforgeeks.org/introduction-to-recurrent-neural-network/.

Uvod u RNN i poređenje sa tradicionalnom NN.
on-line at: https://towardsdatascience.com/recurrent-neural-networks-d4642c9bc7ce.

Sara Gavran.
Veštačke neuronske mreže u istraživanju podataka: pregled i primena.
Master's thesis, Univerzitet u Beogradu, Matematički fakultet, 2016.