Mrežna inačica: ISSN 2459-6302
Tiskana inačica: ISSN 1849-1995
DOI: 10.19279/TVZ.PD.2021-9-2-06

ZNANJE I UČENJE JEDNOG NEURONA

KNOWLEDGE AND LEARNING OF A SINGLE NEURON

Predrag Valožić

Tehničko veleučilište u Zagrebu, profesor u mirovini

SAŽETAK

Analizirani su ishodi učenja linearnog umjetnog neurona s različitim prethodnim znanjem. Temeljem zajedničkog matematičkog modela neurona i jednakog predloška za učenje primjerima su ilustrirana svojstva različitih modela učenja, više – manje matematiziranih. Pokazano je kako su „napredniji" modeli brži, ali osjetljiviji su i rigidni u ponuđenim rješenjima. „Djetinjastiji", bazni modeli učenja su sporiji, ali univerzalniji su i „maštovitiji". Zajednički kriterij ocjene „ishoda učenja" je ispravnost rješenja problema – dizajn rekurzivnog generatora sinusne sekvence određenih ciljanih značajki: amplitude i frekvencije.

Ključne riječi: *linearni neuron, učenje, linearna kombinacija, inverz, pseudoinverz, povratna veza*

ABSTRACT

The learning outcomes of a linear artificial neuron with different prior knowledge were analyzed. Based on a common mathematical model of neurons and the same template for learning, the properties of different learning models more or less mathematized are illustrated with examples. It has been shown that "more advanced" models are faster but more sensitive and rigid in the solutions offered. "Childish", basic learning models are slower but more universal and "imaginative". A common criterion for assessing the "learning outcome" is the correctness of the solution to the problem – the design of a recursive sine wave generator of certain features: amplitude and frequency.

Keywords: *Linear neuron, learning, linear combination, inverse, pseudoinverse, feedback*

1. UVOD

1. INTRODUCTION

Kako učimo?

Kako i kada počinjemo učiti?

Ovisi li sposobnost učenja o naobrazbi (prethodnom učenju i stečenom znanju)?

Čemu služe učenje i znanje?

Vrijede li ova pitanja i odgovori na njih i za jedan, umjetni neuron?

Cilj ovoga rada nije odgovoriti na prethodna pitanja općenito nego, pod pretpostavkom da je odgovor na peto pitanja potvrdan, odgovoriti na prva četiri na primjeru jednog linearnog umjetnog neurona.

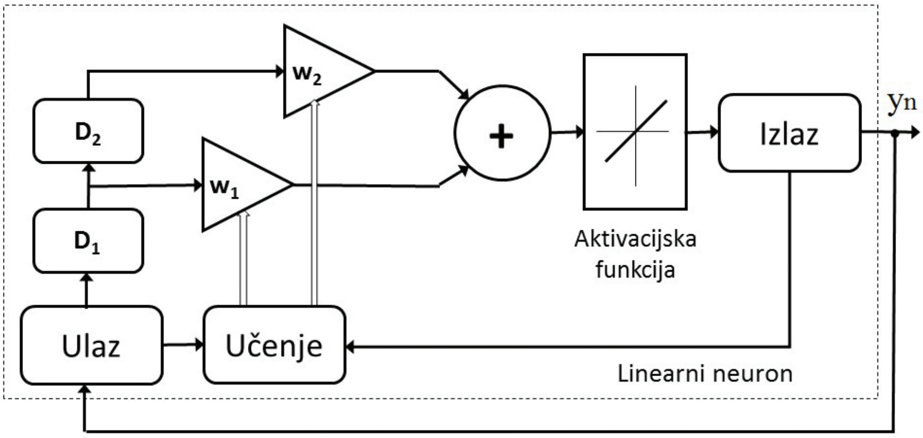
Definicija [1]: *Kaže se da računalni program uči iz iskustva **I** s obzirom na neku klasu zadataka **Z** i mjeru izvedbe **M**, ako se njegova izvedba na zadacima u **Z**, mjerena primjenom **M**, poboljšava s iskustvom **I**.*

Poput svih ostalih inteligentnih subjekata (ljudi, životinje i neki strojevi) i linearni neuron uči tako što mijenja ponašanje pod vanjskim utjecajem. Vanjski utjecaji su ulazi neurona. Ponašanje i izlaz ***y*** linearnog neurona matematički modeliramo linearnom kombinacijom *(1)*.

$$y_n = w_1 \cdot y_{n-1} + w_2 \cdot y_{n-2} + \cdots + w_k \cdot y_{n-k} + u_0 \cdot x_n + u_1 \cdot x_{n-1} + \cdots + u_l \cdot x_{n-l} \qquad (1)$$

Ako je linearni neuron ugrađen u rekurzivni generator digitalne sekvence (Slika 1), linearna kombinacija *(1)* zadrži samo rekurzivni dio:

$$y_n = w_1 \cdot y_{n-1} + w_2 \cdot y_{n-2} + \cdots + w_k \cdot y_{n-k} \qquad (2)$$



***Slika 1** Adaptivni rekurzivni generator sinusnog signala s ugrađenim linearnim neuronom [3]*

***Figure 1** Adaptive, recursive sine-wave generator with linear neuron embedded [3]*

Promjene ponašanja neurona i izlaza *y* odvijaju se promjenom vrijednosti težinskih koeficijenata w_i. Promjena težinskih koeficijenata događa se u periodu učenja. U odnosu na ukupno vrijeme nakon aktiviranja linearnog neurona, učenje može biti jednokratno, na početku rada neurona, povremeno ili kontinuirano. Analizirano je jednokratno učenje.

Pitanja s početka teksta moguće je reducirati na jedno, inicijalno: Kako će linearni neuron u fazi učenja iz ulaznih podataka (uzorka sekvence) odrediti težinske koeficijente w_i ?

U svakom od analiziranih slučajeva, učenje linearnog neurona temelji se na uzorku sekvence sinusnog signala. Razlika je u pretpostavljenom modelu učenja koji podrazumijeva stanovito predznanje – poznavanje matematike. Matematika [2] je egzaktna (točna, nedvojbena) znanost koja izučava aksiomatski definirane apstraktne strukture koristeći matematičku logiku. Kako svi analizirani modeli učenja nisu podjednako egzaktni, moguće je zaključiti kako primjena matematike nije podjednako dosljedna. Uz matematiku, u procesu učenja primjenjuju se i intuicija, profesionalno iskustvo te primjeri iz života.

Prvi, **referentni** model [3] je **trigonometrijski**. Uspoređuju se vrijednosti iz tablice sinusne funkcije izračunate funkcijom *sin()* pa se primjeni formula trigonometrijskih identiteta za *sin()* i *cos()* višestrukih kutova. Ekvivalentni model dobije se iz formule za **z-transformaciju** sinusne sekvence.

Drugi, **matrični** model temelji se na poznavanju **linearne algebre**. Pokušajima i pogreškama dođe se do reda linearnog modela *(2)*. Koeficijenti modela određuju se izračunom **inverza** matrice.

Treći model je proširenje prethodnog, matričnog modela. Koeficijenti linearnog modela različitih redova određuju se primjenom **pseudoinverza** matrice.

Najsloženiji i najmanje transparentan je četvrti, Matlab ***newlind*** model linearnog neurona.

Najjednostavniji, ali i najelastičniji, je peti, **kibernetski** model učenja gdje se metodom pokušaja i pogrešaka iterativno izračunavaju koeficijenti u linearnoj kombinaciji *(2)*. Konvergencija rješenja postiže se procesom nalik na Widrow-Hoff metodu učenja ili gradijentni spust.

2. REFERENTNI MODEL
2. REFERENCE MODEL

Diskretnu sinusnu sekvencu sn izračuna se primjenom formule:

$$s_n = S \cdot sin(n \cdot \Omega) \tag{3}$$

$$\Omega = 2 \cdot \pi \cdot f \,/\, fs \text{ [rad]} \tag{4}$$

Ω je diskretan korak promjene argumenta (kuta) sinusne funkcije *(3)*. Frekvencija sempliranja (uzorkovanja) je *fs* a frekvencija ciljanog signala označena je s *f*.

Primjenom formula za elementarne trigonometrijske jednakosti sinusa zbroja i razlike kutova, dobije se:

$$sin(n \cdot \Omega) = \mathbf{2} \cdot \boldsymbol{cos(\Omega)} \cdot sin((n-1) \cdot \Omega) \\ -\mathbf{1} \cdot sin((n-2) \cdot \Omega) \qquad (5)$$

Nova izlazna vrijednost $y_n = sin(n \cdot \Omega)$ sinusne sekvence izračuna se množenjem prethodne (rekurzivni algoritam) vrijednosti

$sin((n-1) \cdot \Omega)$ s konstantom

$R = 2 \cdot cos(\Omega)$

i oduzimanjem pret-prethodne vrijednosti

$sin((n-2) \cdot \Omega)$.

Prvi i drugi uzorak su zadane, početne vrijednosti. Od trećeg uzorka dalje *(5)* je moguće napisati kao:

$$y_n = 2 \cdot cos(\Omega) \cdot y_{n-1} - y_{n-2} \qquad (6)$$

$$y_n = R \cdot y_{n-1} - y_{n-2} \qquad (7)$$

U odnosu na formulu linearne kombinacije *(1)*, vidljivo je:

$$w_1 = R \qquad (7.1.)$$

$$w_2 = -1 \qquad (7.2.)$$

Ostalih težinskih koeficijenata iz *(1)* i *(2)* nema, jednaki su nuli. Linearna kombinacija je duljine tj. reda Nd=2.

Ako je $\Omega = \pi / 8$ [rad], prvi težinski koeficijent w_1 je $R = 2 \cdot cos(\Omega) = 1,847759065$. Rezultat je sinusna sekvenca s periodom 16 uzoraka.

3. MATRIČNI MODEL UČENJA

3. MATRIX LEARNING MODEL

Linearnu kombinaciju *(2)* moguće je prikazati matricama:

$$\begin{bmatrix} y_n \\ y_{n+1} \\ \cdots \\ y_{n+k} \end{bmatrix} = \begin{bmatrix} y_{n-1} & y_{n-2} & \cdots & y_{n-k} \\ y_n & y_{n-1} & \cdots & y_{n-k+1} \\ \cdots & \cdots & \cdots & \cdots \\ y_{n+k-1} & y_{n+k-2} & \cdots & y_n \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \\ \cdots \\ w_k \end{bmatrix} \qquad (8)$$

Kompaktniji je zapis:

$$\boldsymbol{Y_n = M_n * W} \qquad (9)$$

Iz *(9)* vidljivo je kako izračunati težinske koeficijente:

$$\boldsymbol{W = M_n^{-1} * Y_n} \qquad (10)$$

Za linearnu kombinaciju drugoga reda koeficijenti su w_1 i w_2:

$$\begin{bmatrix} y_3 \\ y_4 \end{bmatrix} = \begin{bmatrix} y_2 & y_1 \\ y_3 & y_2 \end{bmatrix} * \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} \qquad (11)$$

$$y_n = \sin((n-1) \cdot \Omega) \qquad (12)$$

Primjer algoritma i izračuna težinskih koeficijenata ***w*** u *OctaveOnline*:

```
N=16;
omega = 2*pi/N;
x=[0:3];
kut=x*omega;
R=2*cos(omega)
y=sin(kut)
M=[y(2:-1:1);y(3:-1:2)]
Mi=inv(M)
w=Mi*transpose(y(3:4))
```

Rezultati:

```
octave:52> source("m0_inv.m")
R =     1.8478
y =     0.00000    0.38268    0.70711
0.92388
M =     0.38268    0.00000
        0.70711    0.38268
Mi =    2.61313    0.00000
       -4.82843    2.61313
w =     1.84776
       -1.00000
```

Dakle, učenje linearnog neurona u matričnom modelu je primjenom jednadžbe *(10)*.

Pokušamo li s linearnom kombinacijom trećeg reda u *OctaveOnline*:

```
octave:45> m3
m3 =
0.70711    0.38268    0.00000
0.92388    0.70711    0.38268
1.00000    0.92388    0.70711
```

```
octave:46> inv(m3)
warning: matrix singular to machine
precision, rcond = 1.64503e-17
ans =
  -3.2498e+15   6.0048e+15
-3.2498e+15
   6.0048e+15  -1.1095e+16
6.0048e+15
  -3.2498e+15   6.0048e+15
-3.2498e+15

octave:49> det(m3)
ans =   -4.5064e-17
```

Determinanta matrice m3 praktično jednaka je nuli pa matrica m3 nije regularna.

Rješenje, naizgled, postoji, ali nije dobro:

```
w =
   1.50000
  -2.00000
   0.00000
```

Sinusna sekvenca *(12)* je:

```
octave:6> y(1:7)
ans =
0.00000   0.38268   0.70711
0.92388   1.00000   0.92388
0.70711
```

Primjenom *(2)*, dobije se:

```
octave:7> ya(1:7)
ans =
0.00000   0.38268   0.70711
0.29529  -0.97127  -2.04750
-1.12870
```

Generirana sekvenca *ya* razlikuje se od originalne sinusne sekvence *y* i nestabilna je.

Učenje je neuspješno!

3.1. PRILAGODBA MATRICE
3.1. MATRIX ADJUSTMENT

Rang matrice govori nam koliko je neovisnih redaka ili stupaca matrice. Za matricu m3 je:

```
octave:58> rank(m3)
ans =  2
```

Dakle, dva su neovisna retka ili stupca. Od matrice m3 dimenzije 3 x 3, submatrice dimenzije 2 x 2 možemo načiniti na više načina, npr.: odabirom 1. i 2., 2. i 3. te 1. i 3. retka i stupca:

```
octave:27> m12
m12 =
   0.70711   0.38268
   0.92388   0.70711
octave:4> w12
w12 =
   1.8478
  -1.0000
octave:28> m23
m23 =
   0.70711   0.38268
   0.92388   0.70711

octave:6> w23
w23 =
   2.4142
  -1.8478

octave:29> m13
m13 =
   0.70711   0.00000
   1.00000   0.70711
octave:5> w13
w13 =
   1.30656
  -0.54120
```

Matrice m12 i m23 jednake su. Isto vrijedi i za njihove inverze, ali y45 i y56 i su različiti pa izračunati težinski koeficijenti w12 i w23 nisu jednaki. Svi izračunati parovi koeficijenata podjednako su dobri rezultati učenja: svi daju korektnu sinusni sekvencu.

4. PSEUDOINVERZ

4. PSEUDOINVERSE

4.1. INVERZ I PSEUDOINVERZ

4.1. INVERSE AND PSEUDOINVERSE

Ako je matrica **M** kvadratna, regularna (determinanta ≠ 0) matrica, tada postoji inverzna (Octave funkcija ***inv()***) matrica $\mathbf{M}^{-1}$ tako da vrijedi:

$$\mathbf{M}^{-1} * \mathbf{M} = \mathbf{I} \qquad (13)$$

Ako je matrica M kvadratna i singularna (determinanta je 0 ili jako bliska nuli) tada ne postoji inverzna matrica, ali je moguće izračunati pseudoinverz $\mathbf{M}^{+}$ Octave funkcijom ***pinv()***.

Za pseudoinverz $\mathbf{M}^{+}$ vrijedi:

$$\mathbf{M}^{+} * \mathbf{M} \neq \mathbf{I} \qquad (14)$$

ali

$$\mathbf{M} * \mathbf{M}^{+} * \mathbf{M} = \mathbf{M} \qquad (15)$$

4.2. UČENJE PRIMJENOM PSEUDOINVERZA

4.2. LEARNING USING PSEUDOINVERSES

Za matricu **m3pi** = pinv(**m3**) :

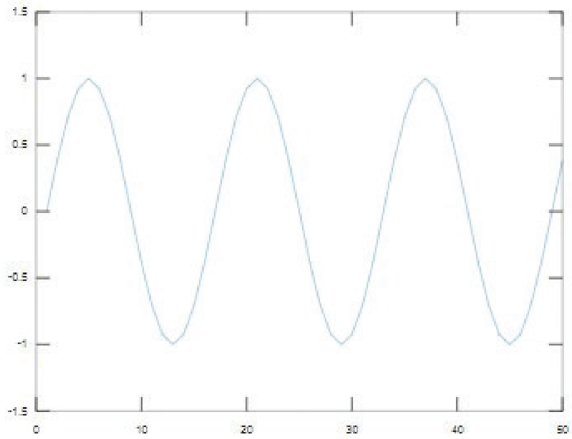
```
m3 =
   0.70711   0.38268   0.00000
   0.92388   0.70711   0.38268
   1.00000   0.92388   0.70711

m3pi =
   1.289465   0.430423  -0.494147
  -0.252136   0.096488   0.430423
  -1.755351  -0.252136   1.289465
```

Izračunati težinski koeficijenti w_1, w_2 i w_3 su:

```
wpi =
   1.16520
   0.26120
  -0.68256
```

Provjerom rada generatora s koeficijentima *w* dobijemo sekvencu *ya* koja se numerički i grafički podudara s originalnom sekvencom *y*.

```
octave:10> ya(1:7)
ans =
   0.00000   0.38268   0.70711
0.92388   1.00000   0.92388
0.70711
```



Slika 2 *Graf generirane sekvence ya, učenje s* ***pinv()***

Figure 2 *Graph of the generated sequence ya, learning with* ***pinv ()***

Primjenom pseudoinverza matrice **m3** dobiveni su koeficijenti linearne kombinacije $\boldsymbol{w_3}$=[1.16520 0.26120 -0.68256] koji generiraju ispravnu sinusnu sekvencu Slika 2.

Učenje je uspješno!

5. MATLAB LINEARNI NEURON

5. MATLAB LINEAR NEURON

Kao i u [3] za dizajn rekurzivnog generatora sinusne sekvence iskorištena je Matlab funkcija *newlind*. Ako je na raspolaganju ukupno N=7 uzoraka i ulazna linija za kašnjenje duljine je Nd=2, dobije se model neurona Slika 3.

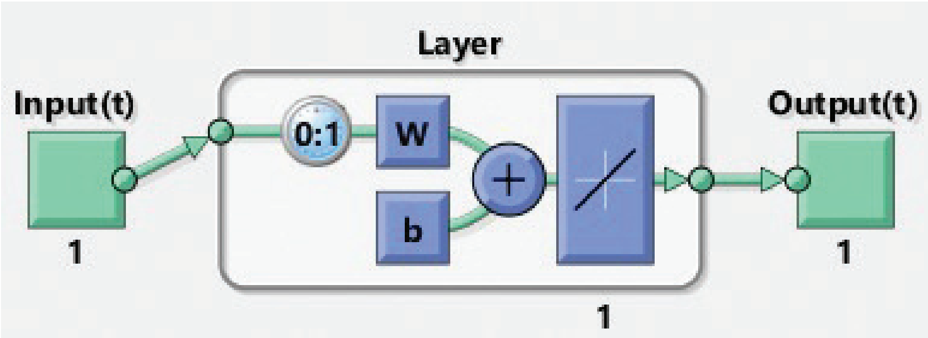
N = 7

Nd = 2

weights = 1.84775906502257 -1

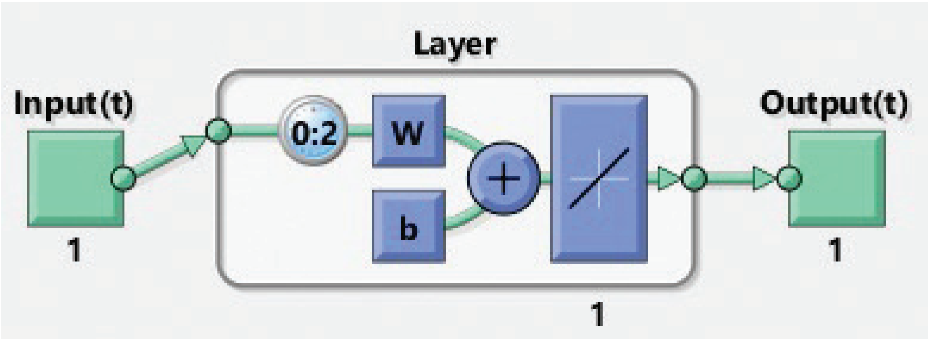
bias = 4.88826716915933e-16

R = 1.84775906502257



Slika 3 *Shema Matlab newlind neurona drugog reda: Nd=2*

Figure 3 *Diagram of the second order Matlab newlind neuron: Nd = 2*



Slika 4 *Shema Matlab newlind neurona trećeg reda: Nd=3*

Figure 4 *Diagram of the third order Matlab newlind neuron: Nd = 3*

Dobiveni težinski koeficijenti *w* jednaki su referentnim trigonometrijskim (7.1. i 7.2.).

Matlab *newlind* dopušta rješenja i s većim brojem koeficijenata. Za Nd = 3, dobije se:

N = 7 Nd = 3

weights = 1.30656296487638 0 -0.541196100146197

bias = 4.20955479178683e-16

Koeficijenti w_1 i w_3 jednaki su koeficijentima w_{13} iz prethodnog Octave **inverz** primjera.

Dakle, *newlind* je složeni algoritam učenja koji provjerava rang matrice pa određuje koeficijente za neku od submatrica odgovarajućeg ranga.

Preostaje pitanje odabira koeficijenata manje matrice. Od tri mogućnosti: 1. i 2. redak i stupac, 2. i 3. te 1. i 3. *newlind* se odlučio za treće. Kako bi dobiveni sustav bio 3. reda! Možda?

Kako je već opisano [3] linearni neuron Matlab *newlind* daje različita, primjenljiva rješenja pa je na korisniku – dizajneru odabir optimalnih koeficijenata rekurzivnog generatora sinusne sekvence. I provjera ispravnosti rada dizajniranog generatora, svakako.

6. KIBERNETSKI MODEL
6. CYBERNETIC MODEL

Kibernetski model učenja linearnog neurona temelji se na sustavu regulacije s povratnom vezom Slika 5 [5] i Slika 6.

Neuron započinje učenja s početnim vrijednostima koeficijenata $\mathbf{w_0} = [w_{01}\ w_{02}]$. Linearna kombinacija početnih vrijednosti sekvence $[yr_1\ yr_2]$ i vektora težinskih koeficijenata $\mathbf{w_0}$ daje

$y_3 = [yr_1\ yr_2] * \mathbf{w_0}^T$.

Izračuna se i

$y_4 = [yr_2\ yr_3] * \mathbf{w_0}^T$

Razlika izračunatih i referentnih vrijednosti sinusnih sekvence je

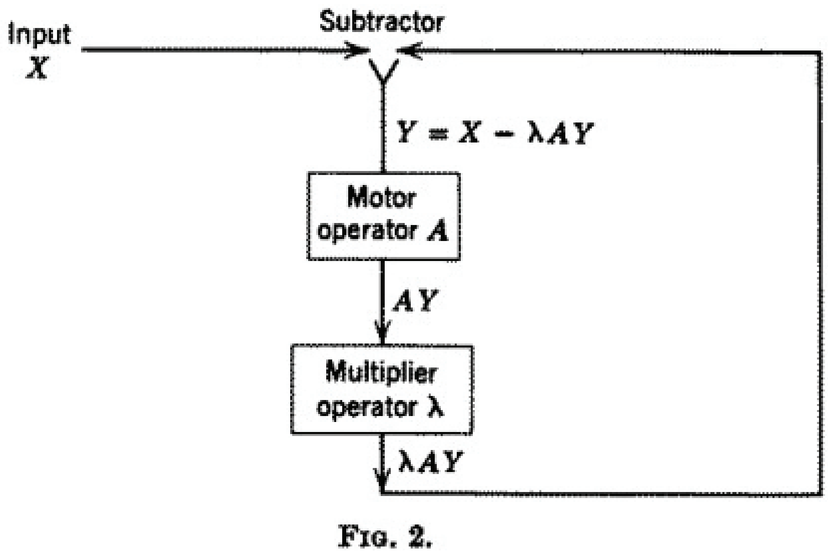
delta(1)=yr(3)-y(3);

delta(2)=yr(4)-y(4);

Umnožak intenziteta učenja *eta* i vektora greške ***delta*** korigira vektor koeficijenata

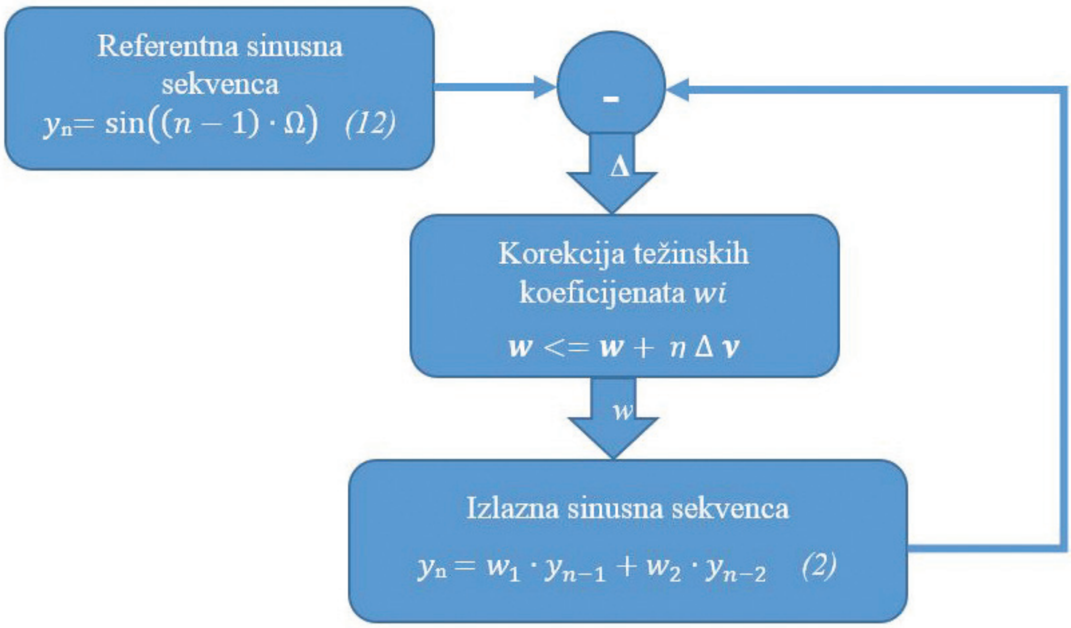
$\mathbf{w_{(n+1)}} = \mathbf{w_n} + eta*\mathbf{delta}$;

Postupak se ponavlja zadani broj puta Ni ili dok greška *delta* po apsolutnom iznosu ne padne ispod neke zadane veličine.



Slika 5 *Sustav regulacije s povratnom vezom N. Wiener 1948 [5]*

Figure 5 *Feedback control system N. Wiener 1948 [5]*



Slika 6 *Kibernetski model učenja linearnog neurona*

Figure 6 *Cybernetic model of linear neuron learning*

Za učenje trebamo najmanje 2·Nd uzoraka referentne sekvence.

```
octave:38> source("m1_2_Cyber.m")
Ni =  100
eta =  0.50000
w =    1.84776  -1.00000
```

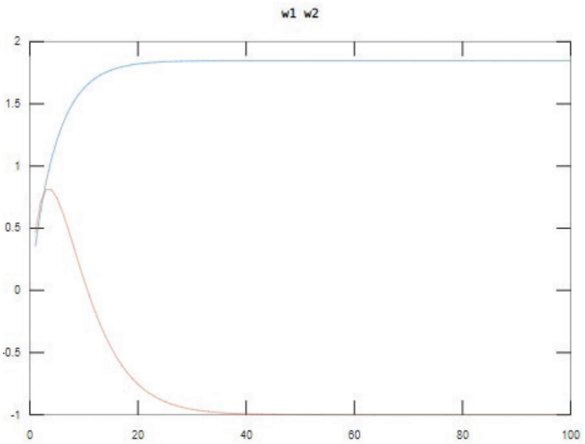
Na Slika 7 prikazane su promjene vrijednosti koeficijenata w_1 i w_2 od početnih [0 0] do konačnih 1.84776 i -1.

Nakon Ni = 100 iteracija, greške su:

```
delta =   0.00000000052267
-0.00000002194005
```

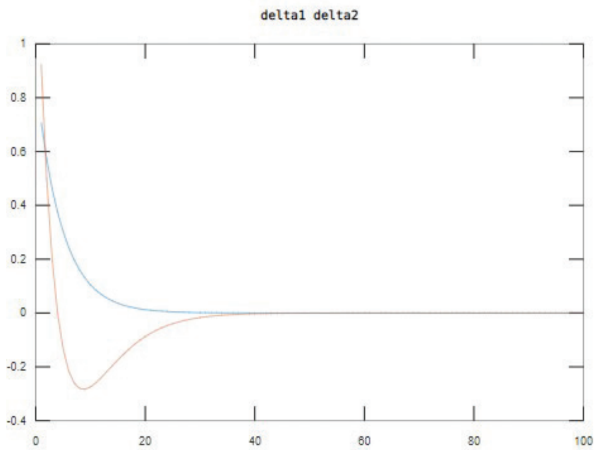
Promjene greške *delta* tijekom iterativnog postupka prikazane su na Slika 8.

Početne vrijednosti težinskih koeficijenata u prethodnom primjeru bile su w=[0 0]. Moguće je, npr. na stotinu primjera, provjeriti što se događa ako su početne vrijednosti slučajne s uniformnom razdiobom između -2,5 i 2,5: w=[5*(rand()-0.5) 5*(rand()-0.5)]. Rezultati su prikazani na Slika 9.



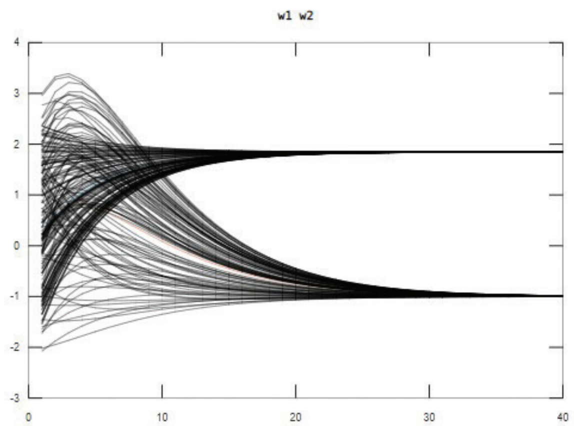
Slika 7 *Konvergencija težinskih koeficijenata u procesu učenja*

Figure 7 *Convergence of weight coefficients in the learning process*



Slika 8 *Konvergencija pogreške Δ u procesu učenja*

Figure 8 *Error convergence Δ in the learning process*



Slika 9 *Učenje s različitim početnim vrijednostima **w***

Figure 9 *Learning with different initial values **w***

Koje god bile početne vrijednosti težinskih koeficijenata ***w***, konačne vrijednosti su ispravne.

Učenje je uspješno!

Pokušamo li s linearnom kombinacijom duljine Nd=3, za početne vrijednosti težinskih koeficijenata $\boldsymbol{w_0}$:

```
w0 =     0.44449    0.75155    0.37331
```

rješenje je:

```
w =    1.05976    0.45603   -0.78800
```

Ako su, pak, početne vrijednosti:

```
w0 =    0.19996    0.22337    0.64388
```

rješenje za težinske koeficijente je drugačije:

```
w =     1.24482    0.11408   -0.60293
```

Težinski koeficijenti su različiti, ali generatori su podjednako dobri.

Neuron se snalazi u različitim situacijama!

7. ZAKLJUČAK

Zadaća generatora s linearnim neuronom je trajno generirati sinusnu sekvencu po raspoloživom predlošku. Predložak je odsječak sinusne sekvence od samo nekoliko sukcesivnih numeričkih vrijednosti kojih je ukupno manje od polovice perioda. Prethodno znanje koje posjeduje neuron je matematički (algebarski, računski) model – Linearna kombinacija, a što se podudara sa strukturom linearnog neurona Slika 1. Moglo bi se reći kako je taj dio znanja u strukturi neurona koja je programirana „genetskim kodom".

Drugi dio znanja je mehanizam učenja. Matrični (Inverz i Pseudoinverz) algoritmi učenja podrazumijevaju poznavanje više matematike – Linearne algebre. Matlab model linearnog neurona je najmanje transparentan te su tek postavljene hipoteze o ugrađenim algoritmima. Najjednostavniji, ujedno i najopćenitiji je kibernetski model: ustrajne korekcije parametara matematičkog modela dok se dovoljno precizno ne dobiju vrijednosti zadanog predloška.

Uz vrijednosti težinskih koeficijenata modela, varijabla modela je i njihov broj. Matrični inverz postupak daje rješenje samo ako matematički model odgovara fizikalnom modelu.

Pseudoinverz i Matlab model daju jedinstveno rješenje za sve pretpostavljene veličine modela. Prikazani kibernetski model daje rješenje jednako jedinstvenim rješenjima ostalih modela, neovisno o vrijednostima na početku učenja.

Na pitanja s početka teksta odgovori bi mogli biti:

Ovisi li sposobnost učenja o naobrazbi (prethodnom učenju i stečenom znanju)?

DA, „obrazovaniji“ neuron izračuna težinske koeficijente u jednom ciklusu.

Kako i kada počinjemo učiti?

Od samog „rođenja“; oponašanjem primjera iz okruženja i korekcijom vlastitog ponašanja. Viši stupnjevi obrazovanja omogućavaju kvalitetnije učenje, ali početni, prirođeni način učenja je univerzalniji i „maštovitiji“.

Kako učimo?

Ulaznim komunikacijskim kanalima pribavljamo informacije iz vanjskog svijeta i sustavno ih slažemo u ukupni korpus znanja.

Čemu služi učenje i znanje?

Ako subjekt nije kvalitetno naučio zadaću, neće opstati: neuron će biti resetiran a neispravan generator isključen i rastavljen.

Vrijede li ova pitanja i odgovori na njih i za jedan, umjetni neuron?

Primjerima je ilustrirano kako prethodni odgovori vrijede za linearni, umjetni neuron. Naša osobna iskustva potvrđuju kako vrijede i za nas. Dakle, učenje je zajedničko svim inteligentnim jedinkama i sustavima kibernetskog univerzuma. Neovisno o hardveru!

7. CONCLUSION

The task of a linear neuron generator is to permanently generate a sine sequence according to the available template. A template is a sine sequence of only a few successive numerical values that total less than half of the period. Previous knowledge possessed by a neuron is a mathematical (algebraic, computational) model: Linear combination, which coincides with the structure of a linear neuron Figure 1. It could be said that this part of knowledge is in the structure of neurons and which is programmed by the "genetic code".

The second part of knowledge is the learning mechanism. Matrix (Inverse and Pseudoinverse) learning algorithms imply knowledge of higher mathematics - Linear Algebra. The Matlab model of a linear neuron is the least transparent and hypotheses about embedded algorithms have just been set.

The simplest but also the most general is the cybernetic model: persistent corrections of the parameters of a mathematical model until the values of the given template are obtained with sufficient precision.

In addition to the values of the model weight coefficients, the model variable is also their number. The matrix inverse procedure gives a solution only if the mathematical model corresponds to the physical model. The pseudoinverse and Matlab model provide a unique solution for all assumed model sizes. The presented cybernetic model provides a solution equally unique to the solutions of other models, independent of the values at the beginning of learning.

The answers to the questions at the beginning of the text could be:

Does the ability to learn depend on education (previous learning and acquired knowledge)?

YES, a "more educated" neuron calculates weight coefficients in one cycle.

How and when do we start learning?

From the very "birth"; by imitating examples from the environment and correcting one's own behaviour. Higher levels of education enable better learning, but the initial, innate way of learning is more universal and "imaginative".

How do we learn?

Through incoming communication channels, we obtain information from the outside world and systematically combine it into a total body of knowledge.

What is the purpose of learning and knowledge?

If the subject has not learned the task well, it will not survive: the neuron will be reset and the faulty generator will be turned off and disassembled.

Are these questions and the answers to them valid for one, artificial neuron?

The examples illustrate how the previous answers apply to a linear, artificial neuron. Our personal experiences confirm that they are true for us as well. Thus, learning is common to all intelligent units and systems of the cybernetic universe. Regardless of the hardware!

8. REFERENCE
8. REFERENCES

[1.] Tom M. Mitchell: Machine Learning, McGraw-Hill, March 1, 1997.

[2.] https://hr.wikipedia.org/wiki/Matematika pristup 11.5.2020.

[3.] Predrag Valožić: Inteligencija i kreativnost jednog neurona, Polytechnic & Design, Vol. 6. No. 3. 2018.

[4.] Bernard Widrow: An Adaptive "ADALINE" Neuron Using Chemical "Memistors", Technical Report No. 1553-2, October 17, 1960. USA Office of Naval Research

[5.] Norbert Wiener: Cybernetics: Or Control and Communication in the Animal and the Machine. Paris, (Hermann & Cie) & Camb. Mass. (MIT Press) ISBN 978-0-262-73009-9; 1948, 2nd revised ed. 1961.

AUTOR · *AUTHOR*

● **Predrag Valožić -** nepromjenjena biografija nalazi se u časopisu Polytechnic & Design Vol. 8, No. 3, 2020.

Korespondencija · ***Correspondence***
pvalozic@tvz.hr
pvalozic@gmail.com