

Nevronske mreže in učenje s PSO

Umetni inteligentni sistemi za podporo odločanju

Jernej Sabadin



Mentorja: Prof. Dr. Igor Škrjanc, Univ. Dipl. Inž. El., doc. dr. Dejan Dovžan

Predmet: Inteligentni sistemi za podporo odločanju

Datum: 10. Marec 2023

Contents

1	Naloge	1
2	Grob opis nevronskega omrežja	1
3	Model nevrona	1
4	Opis triplastnega nevronskega omrežja	2
4.1	Učenje triplastnega perceptrona	2
5	Učenje s PSO	3
6	Implementacija algoritma v okolju Matlab	3
7	Poskus z linearno aktivacijsko funkcijo	4
8	Reference	5

Ključne besede: Nevronske mreže, PSO, rak dojke

1 Naloge

- Opis triplastnega nevronskega omrežja
- Programiranje perceptrona v okolju Matlab
- preiskus na zbirki diagnostike raka dojk

2 Grob opis nevronskega omrežja

”Umetno nevronske mreže je umetna oblika človekovih možganov. Ti so se sposobni naučiti novih stvari in se prilagajati na spreminjajoče se okolje. Na primer dojenček prepozna mater po vonju in glasu. Brati zmoremo pisave drugih ljudi. V slabi vidljivosti prepoznamo predmete po njihovi obliki. Zaradi svoje sposobnosti nadziranja telesa, razmišljanja, vizualizacije, sanjanja, domišljanja in učenja so možgani bolj zmogljivi kot najnaprednejši računalnik.

Možgani so sestavljeni iz nevronov, njihove povezave pa tvorijo nevronske mreže. Biološki nevron je sestavljen iz celičnega telesa, aksona in dendritov. Nevron sprejema signale od drugih nevronov prek dendritov. Ko moč signala preseže določen prag, ta nevron sproži lasten signal, ki se prenese na naslednje nevrone.”[4]

3 Model nevrona

”Gradnik nevronskega omrežja je nevron. Nevron prejme več vhodov in ima en sam izhod. Priključeni vhodi so pomnoženi z utežmi. Uteženi vhodi se seštejejo s

pragom nevrona. Prag nevrona predstavlja vrednost, pri kateri se aktivira izhod. Utežena vsota vstopa v aktivacijsko funkcijo, ki definira, kako se ta pretvori v izhod.

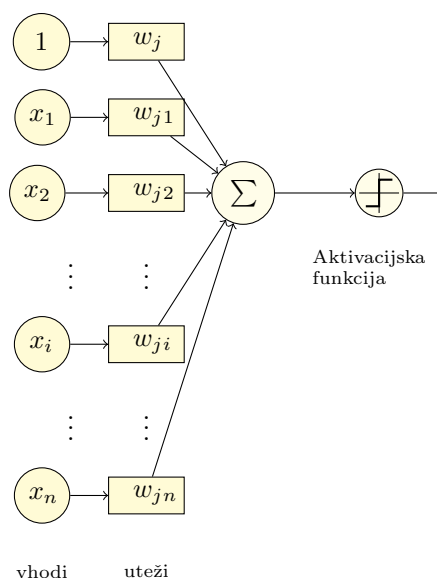


Fig. 1: Slika prikazuje model nevrona

Utež med i -tim in j -tim nevronom predstavimo z w_{ji} . Vsoto vhodov v j -ti nevron in njenega praga w_j zapišemo z izrazom:

$$z_j = \left(\sum_{i=1}^n w_{ji} x_i \right) - w_j \quad (1)$$

Utežena vsota vstopa v nelinearen element oz. aktivacijsko funkcijo. Izhod iz te definiramo kot:

$$y_j = f(z_j) \quad (2)$$

Aktivacijska funkcija je podobna aktivnostim v naših možganih, kjer se različni nevroni sprožijo z različnimi dražljaji. Najpogostejša v praksi je sigmoidalna funkcija.” [4]

4 Opis triplastnega nevronskega omrežja

”V nadaljevanju opišemo triplastno nevronskega omrežje, ki ga opišemo s pomočjo knjige Razpoznavanje vzorcev Nikole Pavšiča [1].

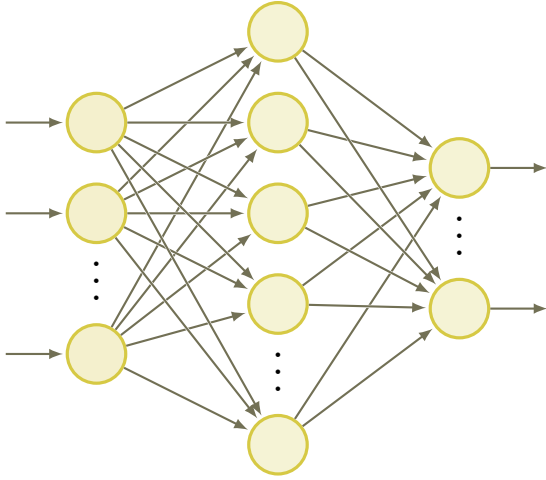


Fig. 2: Shema nevronskega omrežja

Izrazimo j -ti nevron v l -ti plasti z

$$x_j^{(l)} = f \left(\sum_{i=1}^{n_{l-1}+1} w_{ji}^{(l)} x_i^{(l-1)} \right) \quad (3)$$

kjer upoštevamo, da so vhodi $x_{n_{l-1}+1}^{(l-1)} = 1$. Števila nevronov v posamezni plasti predstavimo z n_l . Število $x_i^{(l-1)}$ pa predstavlja i -ti nevron v plasti $l-1$.

Za aktivacijsko funkcijo $f(z)$ uporabimo sigmoidalno funkcijo definirano z:

$$f(x) = \frac{1}{1 + e^{-x}} \quad (4)$$

Značilke $x_i^{(0)}$ vzorca, ki ga razvrščamo, razvrstimo v razred vzorcev C_i , če za izhode nevronov zadnje plasti velja

$$x_i^{(2)} > x_j^{(2)} \quad \text{za} \quad j = 1, 2, \dots, M; \quad j \neq i, \quad (5)$$

pri čemer je M število vseh razredov.

4.1 Učenje triplastnega perceptrona

V postopku vzratnega učenja (angl. back-propagation training algorithm) minimiziramo razliko med želenim poznanim $x^{(4)}$ in dejanskim t izhodom. Uteži se nastavijo na take vrednosti, da minimizirajo napako.

Postopek učenja temelji na gradientni metodi. Uteži se skozi postopek učenja popravljajo, dokler se ne ustalijo na določeni vrednosti. Nove vrednosti uteži izračunamo s pomočjo delnih odvodov funkcije napake. To storimo za vsako plast in vsak učni vzorec. V $k+1$ koraku popravimo uteži po naslednjem izrazu:

$$w_{ji}(k+1)^{(l)} = w_{ji}^{(l)}(k) - \beta \frac{\partial e(w(k))}{\partial w_{ji}^{(l)}(k)} = w_{ji}^{(l)}(k) + \Delta w_{ji}^{(l)}(k), \quad (6)$$

kjer je $e(w(k))$ napaka nevronskega omrežja v k -ti ponovitvi postopka:

$$e(w(k)) = \frac{1}{2} \sum_{p=1}^{n_2} \left(t_p(k) - x_p^{(2)}(k) \right)^2 \quad (7)$$

Pri tem je t želen odziv omrežja. Označimo ga z vektorjem ničel in eno enko, ki ponazarja dani razred.

Faktor β , ki je manjši od ena, določa korak premika k minimumu funkcije napake. Če ga nastavimo na vrednost 1, se bomo proti minimumu premikali prehitro in ga bomo preskočili. Če pa izberemo premajhnega, se računska zahtevnost učenja poveča.

Učenje nevronskega omrežja poteka tako s pomočjo delnih odvodov napake.” [4]

Za izhodno plast pišemo:

$$\begin{aligned} \frac{\partial e(w(k))}{\partial w_{ji}^{(2)}(k)} &= \frac{\partial e(w(k))}{\partial x_j^{(2)}(k)} \cdot \frac{\partial x_j^{(2)}(k)}{\partial z_j^{(2)}(k)} \cdot \frac{\partial z_j^{(2)}(k)}{\partial w_{ji}^{(2)}(k)} \\ &= \left(t_j(k) - x_j^{(2)}(k) \right) \left(1 - x_j^{(2)}(k) \right) x_j^{(2)}(k) x_i^{(1)}(k) \\ &= -d_j^{(2)}(k) x_i^{(1)}(k) \end{aligned} \quad (8)$$

kjer je

$$d_j^{(2)}(k) = \left(t_j(k) - x_j^{(2)}(k) \right) \left(1 - x_j^{(2)}(k) \right) x_j^{(2)}(k) \quad (9)$$

Popravki uteži $\Delta w_{ji}^{(2)}$ so tako definirani kot:

$$\Delta w_{ji}^{(2)} = \beta d_j^{(2)}(k) x_i^{(1)}(k) \quad (10)$$

popraski uteži v plasti $l=1$ pa so definirani kot:

$$\Delta w_{ji}^{(1)} = \beta d_j^{(1)}(k) x_i^{(0)}(k) \quad (11)$$

za $i = 1, 2, \dots, (n_{l-1} + 1)$, $j = 1, 2, \dots, n_l$ in $l = 1$, kjer je:

$$d_j^{(1)}(k) = \left(1 - x_j^{(1)}(k) \right) x_j^{(1)}(k) \sum_{p=1}^{n_2} d_p^{(2)}(k) w_{pj}^{(2)}(k) \quad (12)$$

Postopek učenja poteka tako, da za vhodne označene podatke izračunamo izhode. Glede na pričakovane izhode in dejanske izračunamo napako ter na podlagi nje popravimo vse uteži. To ponovimo tolikokrat, dokler se uteži ne umirijo.

5 Učenje s PSO

V naši nalogi bomo uteži optimizirali s pomočjo algoritma PSO.

PSO je globalni optimizacijski algoritem, ki se zgleduje po vedenju ptic ali rib, ki se zbirajo v jatah. Pri PSO se skupina kandidatov za rešitev (delcev) premika po iskalnem prostoru s ciljem najti optimalno rešitev. Položaj in hitrost vsakega delca se posodobita glede na lastne izkušnje (osebni rekord) in izkušnje celotne skupine (globalni najboljši). Glavni koraki algoritma so naslednji:

- Inicializacija: Algoritem se začne z naključno inicializacijo skupine delcev v iskalnem prostoru. Položaj in hitrost vsakega delca sta naključno dodeljena.
- Vrednotenje: primernost vsakega delca se oceni glede na njegovo vrednost kriterijske funkcije. Kriterijska funkcija je definirana glede na problem, ki se ga rešuje.
- Posodobitev osebnega rekorda: vsak delec posodobi svoj osebni najboljši položaj na podlagi lastne vrednosti kriterijske funkcije. Če ima trenutni položaj delca boljšo vrednost kriterijske funkcije od njegove osebne najboljše vrednosti kriterijske funkcije, delec posodobi svoj osebni najboljši položaj na trenutni položaj ter svojo osebno najboljšo vrednost kriterijske funkcije
- Globalna najboljša posodobitev: Globalna najboljša pozicija se posodobi tako, da se med vsemi delci izbere delec z najboljšo vrednostjo kriterijske funkcije.
- Posodobi položaj in hitrost: Hitrost vsakega delca se posodobi glede na njegovo trenutno hitrost, najboljši osebni položaj in najboljši globalni položaj. Nato se položaj vsakega delca posodobi glede na njegov trenutni položaj in hitrost.
- Mutacija: Z določeno verjetnostjo se za položaje delcev uporabi operacija mutacije, da se v proces iskanja vnesejo naključne spremembe.
- Ponovi: Koraki od 2 do 6 se ponavljajo, dokler ni izpolnjeno merilo za zaustavitev, kot je doseganje največjega števila ponovitev ali praga za vrednost kriterijske funkcije.

Inicializacija:

$$x_i^0 = \text{rand.}, \quad v_i^0 = \text{rand.}, \quad p_i^0 = x_i^0, \quad g^0 = \text{rand.} \quad (13)$$

kjer je *rand* vektor naključnih števil.

Posodobi položaj in hitrost:

$$v_i^{t+1} = w_t v_i^t + c_1 r_{1,i}^t \circ (p_i^t - x_i^t) + c_2 r_{2,i}^t \circ (g^t - x_i^t) \quad (14)$$

kjer vektor v_i^t označuje hitrost delca i v času t , c_1 in c_2 sta konstanti pospeška, w_t je vztrajnostni faktor in $r_{1,i}^t$ in $r_{2,i}^t$ sta vektorja naključnih števil med 0 in 1. Posodobi osebni in globalni rekord:

$$p_i^{t+1} = \begin{cases} x_i^{t+1}, & f(x_i^{t+1}) < f(p_i^t) \\ p_i^t, & \text{sicer} \end{cases} \quad (15)$$

$$g^{t+1} = \begin{cases} x_i^{t+1}, & f(x_i^{t+1}) < f(g^t) \\ g^t, & \text{sicer} \end{cases} \quad (16)$$

kjer je $f(x)$ kriterijska funkcija, ki jo je treba optimizirati.

Posodobi vztrajnostno težo:

$$w_t = w_{\max} - \frac{(w_{\max} - w_{\min})t}{T} \quad (17)$$

kjer sta w_{\max} in w_{\min} največja in najmanjša vztrajnostna utež, t je trenutna iteracija in T skupno število ponovitev.

Posodobi položaj delcev:

$$x_i^{t+1} = x_i^t + v_i^{t+1} \quad (18)$$

kjer x_i^t označuje položaj delca i v času t .

V PSO se mutacija lahko doda položaju delca z uvedbo verjetnosti mutacije p_m in faktorja mutacije δ_j . Pri vsaki ponovitvi se položaj vsakega delca spremeni z verjetnostjo p_m , kot sledi:

$$x_{i,j}^{t+1} = \begin{cases} x_{i,j}^t + \delta_j r_2 & \text{if } r_1 \leq p_m \\ x_{i,j}^t & \text{sicer} \end{cases} \quad (19)$$

kjer je $x_{i,j}^t$ j -ta komponenta položaja delca i v času t , $\delta_j < 1$ je konstanta mutacije, r_2 pa naključno število v mejah vnaprej predpisanih maksimalnih in minimalnih pozicij delcev (v našem primeru bo to število vzeto iz normalne porazdelitve med -1 in 1.). r_1 je naključno število vzeto iz enakomerno porazdeljenega intervala $[0, 1]$.

6 Implementacija algoritma v okolju Matlab

V naši izvedbi PSO delci predstavljajo uteži nevronske mreže. Položaj vsakega delca ustreza nizu uteži za nevronske mreže, kriterijska funkcija pa s temi utežmi oceni delovanje mreže. V našem primeru vzamemo za kriterijsko funkcijo kar MSE (ang. Mean Square Error). Algoritem PSO posodablja položaje delcev v iskalnem prostoru, da optimizira delovanje nevronske mreže.

V našem primeru omejimo hitrosti delcev ter prav tako njihove pozicije (interval $[-1, 1]$). Ker imamo za naš primer, ko določamo diagnozo raka dojk le 2 razreda, ki ju razvrščamo, ima tako naše nevronske omrežje le en nevron v izhodni plasti. Vhodno plast sestavlja 30 nevronov saj imamo 30 značilnk. V vmesni plasti bomo

uporabili le 5 nevronov, saj je naš primer določanja raka dojke ločljiv z linearno ločilno mejo (hiperravnino). Za aktivacijsko funkcijo vzamemo tangent sigmoidalno funkcijo definirano kot

$$f(x) = \frac{2}{1 + \exp(-2x)} - 1 \quad (20)$$

Spodaj prikažemo model našega nevronskega omrežja.

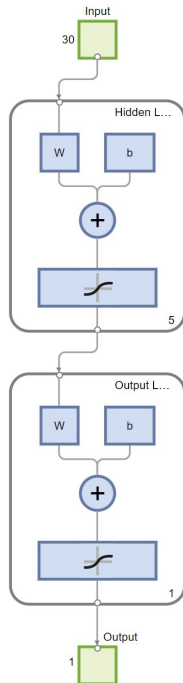


Fig. 3: Model nevronskega omrežja

Predno se lotimo optimizacije obdelamo podatke tako, da jih razdelimo v učno in testno množico, kjer učna množica vsebuje 70% vseh podatkov. Podatke razdelimo tako, da je porazdelitev razredov enaka v testnih in učnih podatkih. Zaženemo optimizacijo nevronskega omrežja in izrisujemo kako se spreminja stopnja razpoznavanja skozi iteracije, kjer učimo le učno zbirko.

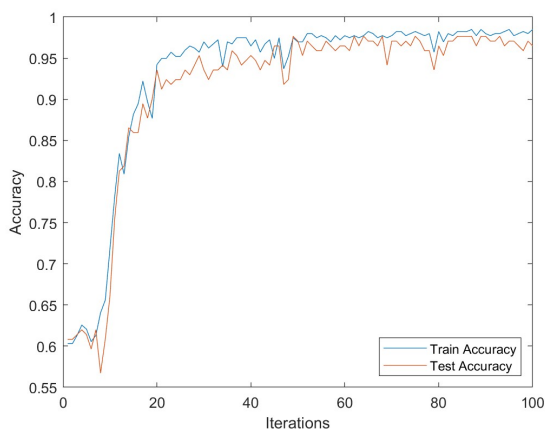


Fig. 4: Delež pravilnega razvrščanja skozi korake učenja

Učenje se zaključi po 100 korakih. Delež pravilno razpoznanih vzorcev znaša za zbirko na kateri smo učili 99%, za testno pa 97%.

Na splošno ni nenavadno, da je natančnost testne zbirke nižja od natančnosti učne zbirke, zlasti če je model kompleksen in ima veliko parametrov. Sicer je za naš primer odstopanje zelo majhno, prav tako je število parametrov majhno. Velik razkorak med natančnostjo učenja in preizkusa bi lahko kazal na overfitting ali težavo z modelom ali procesom učenja. Če želimo izboljšati posplošitev modela, moramo uporabiti nekatere tehnike, kot so zbiranje večje količine podatkov za učenje, uporaba zgodnje ustavitve, različne arhitekture modela ali nastavitve hiperparametrov.

Prikažemo tudi konfuzijsko matriko

Confusion Matrix		
Output Class	0	1
	<div>60 35.1%</div> <div>2 1.2%</div> <div>96.8% 3.2%</div>	<div>4 2.3%</div> <div>105 61.4%</div> <div>96.3% 3.7%</div>
	<div>93.8% 6.2%</div> <div>98.1% 1.9%</div> <div>96.5% 3.5%</div>	
Target Class		

Fig. 5: Konfuzijska matrika, kjer je na x osi informacija o dejanski diagnozi na y osi pa kako je razpoznavalec diagnosticiral raka dojke

Na sliki je prikazan tudi delež TP, TN, FP in FN, ki služijo za nadaljno analizo rezultatov. Maligne tumorje jemljemo za resnejše, ker se lahko širijo na druge dele telesa in postanejo življenjsko nevarni, medtem ko so benigni tumorji običajno manj zaskrbljujoči in se ne širijo na druge dele telesa. 0 na grafu predstavlja malign tumor, 1 pa benign. Vidimo, da napačne diagnoze v večini predstavljajo za bolnika hujše posledice, saj bo v primeru da ima malign tumor z pacient prejel diagnozo za benignega, kar je lahko za pacienta usodno.

7 Poskus z linearno aktivacijsko funkcijo

Ker so naši podatki ločljivi z linearno mejo v prostoru značilk, lahko za aktivacijsko funkcijo uporabimo kar linearno funkcijo.

Ko uporabimo takšno aktivacijsko funkcijo namesto tangent sigmoidalne, je izhod nevronov neposredno sorazmeren z vhom. Posledica tega je lahko bolj občutljivo omrežje, ki se močnejše odziva na majhne spremembe

v vходу, kar lahko povzroči, da je izhod omrežja bolj skokovit.

Določena stopnja skokovitega vedenja v rešitvi lahko pomeni, da algoritem PSO raziskuje različne rešitve in poskuša najti globalni optimum.

V nadaljevanju spremenimo parametre PSO in tako dobimo manj skokovito rešitev. To smo storili z povečanjem števila delcev, zmanjšanjem maximalne hitrosti ter zmanjšanjem faktorja δ_j .

Rezultate prikažemo na spodnji sliki.

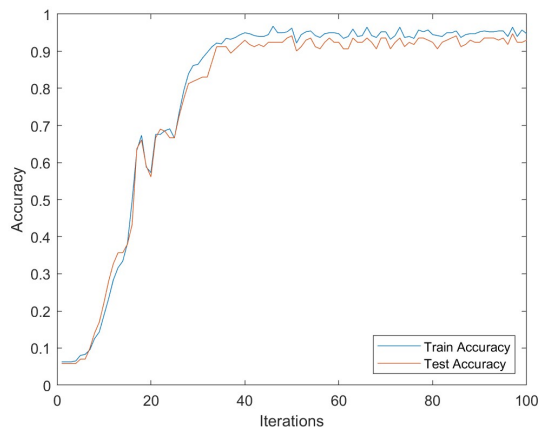


Fig. 6: Delež pravilnega razvrščanja skozi korake učenja

Confusion Matrix		
Output Class	0	<div>53 31.0%</div> <div>1 0.6%</div> <div>98.1% 1.9%</div>
	1	<div>11 6.4%</div> <div>106 62.0%</div> <div>90.6% 9.4%</div>
		<div>82.8% 17.2%</div> <div>99.1% 0.9%</div> <div>93.0% 7.0%</div>
Target Class		

Fig. 7: Konfuzijska matrika, kjer je na x osi informacija o dejanski diagnozi na y osi pa kako je razpoznavalnik diagnosticiral raka dojke

kjer 0 predstavlja benigni rak dojke, 1 pa maligni. Vidimo, da večina napačnih razvrščanj predstavlja za pacienta vsodno diagnozo, saj razpoznamo da ima benigno obliko raka v resnici pa ima maligno, ki je bolj nevarna.

8 Reference

- 1 N. Pavešić, 'Razpoznavanje vzorcev: Uvod v analizo in razumevanje vidnih in slušnih signalov', 3. izd., Ljubljana: Založba FE in FRI, 2012.
- 2 I. Škrjanc, Knjiga - Inteligentni sistemi za podporo odločanju, FE 2023
- 3 I. Škrjanc, Inteligentni sistemi za podporo odločanju, gradivo za predavanja, FE 2023