

# Neuronske mreže: Proces učenja

Prof. dr. sc. Sven Lončarić

Fakultet elektrotehnike i računarstva

sven.loncaric@fer.hr

[https://www.fer.unizg.hr/predmet/neumre\\_c](https://www.fer.unizg.hr/predmet/neumre_c)

# Pregled predavanja

- Uvod
- Učenje korekcijom pogreške
- Hebb-ovo učenje
- Kompetitivno učenje
- Boltzmann-ovo učenje
- Problem nagrađivanja
- Učenje pod nadzorom
- Učenje podrškom
- Učenje bez nadzora

# Uvod

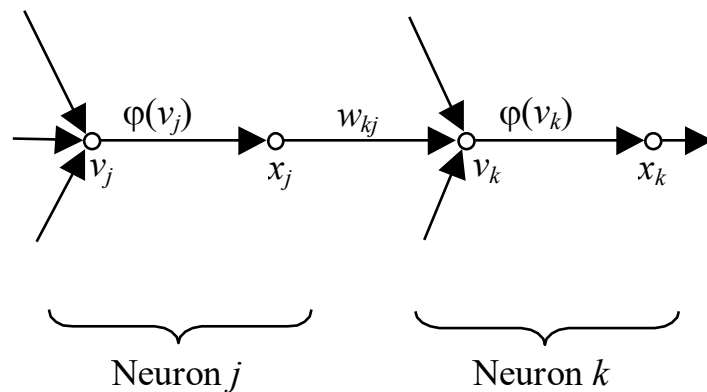
- Jedna od najvažnijih karakteristika neuronskih mreža je sposobnost učenja od okoline
- Neuronska mreža uči o okolini kroz iterativni proces podešavanja sinaptičkih težina i pragova
- Nakon svake iteracije učenja mreža bi trebala imati više znanja o okolini

# Definicija učenja

- Definicija učenja u kontekstu neuronskih mreža:
  - Učenje je proces kojim se slobodni parametri neuronske mreže adaptiraju kroz kontinuirani proces stimulacije od okoline u kojoj se mreža nalazi
  - Vrsta učenja je određena načinom kako se odvija promjena parametara
- Gornja definicija implicira ovaj slijed događaja:
  - Okolina stimulira neuronsku mrežu
  - Neuronska mreža se mijenja kao rezultat stimulacije
  - Zbog nastalih promjena neuronska mreža drugačije reagira na okolinu

# Notacija

- $v_j$  i  $v_k$  su interne aktivnosti neurona  $j$  i  $k$
- $x_j$  i  $x_k$  su izlazi neurona  $j$  i  $k$
- Neka je  $w_{kj}(n)$  sinaptička težina u trenutku  $n$



# Notacija

- Ako u koraku  $n$  promijenimo iznos sinaptičke težine  $w_{kj}(n)$  za  $\Delta w_{kj}(n)$  dobivamo novu vrijednost težine:

$$w_{kj}(n+1) = w_{kj}(n) + \Delta w_{kj}(n)$$

gdje su  $w_{kj}(n)$  i  $w_{kj}(n+1)$  stara i nova vrijednost težine između neurona  $k$  i  $j$

- Skup pravila za rješenje problema učenja zove se algoritam učenja
- Ne postoji *jedinstveni* algoritam učenja nego mnogo različitih algoritama od kojih svaki ima prednosti i mane

# Algoritmi i paradigme učenja

- Algoritmi učenja određuju način izračunavanja  $\Delta w_{kj}(n)$
- Paradigme učenja određuju odnos neuronske mreže prema okolini
- Tri osnovne paradigme učenja su:
  - Učenje pod nadzorom (nadzor vanjskog učitelja)
  - Učenje podrškom (reinforcement learning) (pokušaji i pogreške uz ocjenu suca)
  - Učenje bez nadzora (samoorganizirano - nije potreban učitelj)

# Podjela metoda učenja

- Prema algoritmu učenja:
  - Učenje korekcijom pogreške
  - Hebbovo učenje
  - Kompetitivno učenje
  - Boltzmannovo učenje
  - Thorndikeovo učenje
- Prema paradigmi učenja:
  - Učenje pod nadzorom
  - Učenje podrškom
  - Učenje bez nadzora



# Učenje korekcijom pogreške

- Pripada paradigmi učenja pod nadzorom
- Neka je  $d_k(n)$  željeni odziv neurona  $k$  u trenutku  $n$
- Neka je  $y_k(n)$  dobiveni odziv neurona  $k$  u trenutku  $n$
- Odziv  $y_k(n)$  dobiven je pobudom ulaznim vektorom  $\mathbf{x}(n)$
- Ulazni vektor  $\mathbf{x}(n)$  i željeni odziv  $d_k(n)$  predstavljaju jedan *primjer* koji je predstavljen mreži u trenutku  $n$
- Pogreška je razlika između željenog i dobivenog odziva neurona  $k$  u trenutku  $n$  i jednaka je:

$$e_k(n) = d_k(n) - y_k(n)$$

# Učenje korekcijom pogreške

- Cilj učenja korekcijom pogreške je da se minimizira funkcija pogreške temeljena na pogreškama  $e_k(n)$  tako da se dobiveni odziv svih neurona približava željenom odzivu u nekom statističkom smislu
- Najčešće se koristi srednja kvadratna pogreška kao funkcija pogreške:

$$J = E \left[ \frac{1}{2} \sum_k e_k^2(n) \right]$$

gdje je  $E[.]$  statistički operator očekivanja, a sumacija se radi za sve neurone u izlaznom sloju mreže

# Funkcija pogreške

- Problem s minimizacijom funkcije  $J$  je da je potrebno znati statistička svojstva slučajnog procesa  $e_k(n)$
- Zbog toga se kao *procjena* pogreške koristi trenutna vrijednost pogreške u nekom koraku  $n$  kao funkcija koja se minimizira:

$$\mathcal{E}(n) = \frac{1}{2} \sum_k e_k^2(n)$$

- Ovakav pristup daje približno rješenje

# Delta pravilo učenja

- Minimizacija pogreške  $J$  s obzirom na težine  $w_{kj}(n)$  daje zakon učenja korekcijom pogreške (delta pravilo):

$$\Delta w_{kj}(n) = \eta e_k(n) x_j(n)$$

gdje je  $\eta$  pozitivna konstanta koja određuje brzinu učenja

- Promjena težine proporcionalna je pogrešci na izlazu i iznosu pripadnog ulaza
- Konstanta  $\eta$  mora se pažljivo odabrati
  - mali  $\eta$  daje stabilnost ali je učenje sporo
  - veliki  $\eta$  ubrzava učenje ali donosi rizik nestabilnosti

# Ploha pogreške

- Ako nacrtamo ovisnost pogreške  $J$  o sinaptičkim težinama u mreži dobivamo višedimenzionalnu plohu pogreške (engl. error surface)
- Problem učenja se sastoji u nalaženju točke na plohi pogreške koja daje najmanju pogrešku

# Ploha pogreške

- S obzirom na vrstu neurona postoje dvije mogućnosti:
  - Mreža se sastoji od linearnih neurona i u tom slučaju je ploha pogreške kvadratična funkcija koja ima jedan minimum
  - Mreža ima nelinearne neurone i u tom slučaju ploha pogreške ima jedan (ili više) globalnih minimuma i više lokalnih minimuma
- Učenje počinje iz proizvoljne točke na plohi i minimizacijom korak po korak se:
  - u prvom slučaju približava se globalnom minimumu
  - u drugom slučaju može se završiti i u lokalnom minimumu

# Hebbovo učenje

- Hebbov princip učenja kaže (Hebb, The Organization of Behavior, 1942):
  - Kad je akson neurona A dovoljno blizu da aktivira neuron B i to ponavlja veći broj puta dolazi do metaboličkih promjena tako da se povećava efikasnost neurona A u aktiviranju neurona B
- Ekstenzija ovog principa (Stent, 1973) kaže:
  - Ako jedan neuron ne utječe na drugog (tj. neuroni se aktiviraju asinkrono) onda sinapsa među njima postaje slabija ili se potpuno eliminira

# Pravilo produkta aktivnosti

- Prema Hebbovu principu učenja težine se mijenjaju na slijedeći način:

$$\Delta w_{kj}(n) = F(y_k(n), x_j(n))$$

gdje su  $y_k(n)$  i  $x_j(n)$  izlaz i  $j$ -ti ulaz od  $k$ -tog neurona

- Jedan specijalni slučaj ovog principa je:

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n)$$

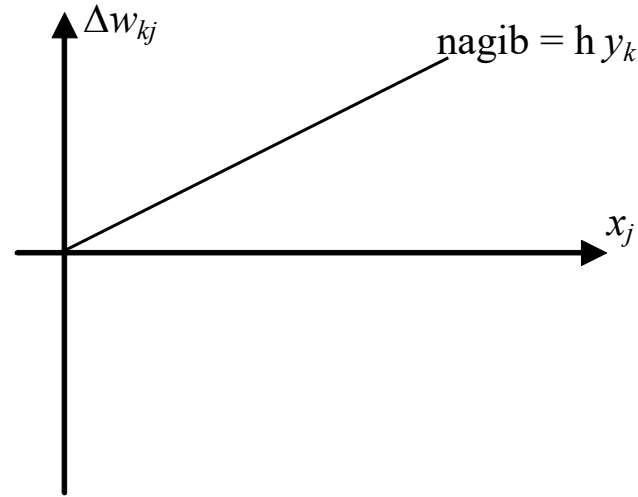
gdje konstanta  $\eta$  određuje brzinu učenja

- U ovom slučaju promjena težine proporcionalna je produktu ulaza i izlaza iz neurona
- Ovo pravilo zove se pravilo produkta aktivnosti (engl. activity product rule)



# Pravilo produkta aktivnosti

- Promjena težine  
proporcionalna je ulaznom  
signalu:  $\Delta w_{kj}(n) = \eta y_k(n) x_j(n)$
- Mana je da uzastopno  
ponavljanje ulazne pobude i  
postojanja izlaza dovodi do  
rasta težine  $w_{kj}$  i konačno  
zasićenja



# Generalizirano pravilo produkta

- Zbog problema zasićenja težine  $w_{kj}$  predložene su modifikacije koje ograničavaju rast težine  $w_{kj}$
- Ovaj cilj može se postići npr. uvođenjem nelinearnog faktora zaboravljanja (Kohonen, 1988):

$$\Delta w_{kj}(n) = \eta y_k(n) x_j(n) - \alpha y_k(n) w_{kj}(n)$$

gdje je  $\alpha$  pozitivna konstanta

- Gornji izraz može se pisati i kao:

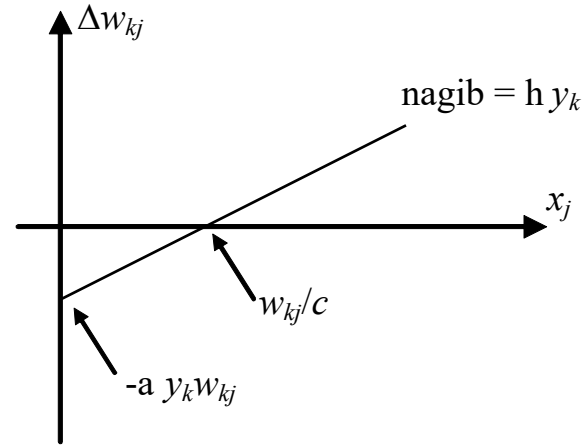
$$\Delta w_{kj}(n) = \alpha y_k(n) [c x_j(n) - w_{kj}(n)]$$

gdje je  $c = \eta / \alpha$

- Ovo pravilo zove se generalizirano pravilo produkta

# Generalizirano pravilo produkta

- Kod generaliziranog Hebbovog pravila svi ulazi koji imaju iznos  $x_j(n) < w_{kj}(n)/c$  uzrokuju smanjenje težine  $w_{kj}$
- Ulazi za koje vrijedi  $x_j(n) > w_{kj}(n)/c$  povećavaju težinu  $w_{kj}$

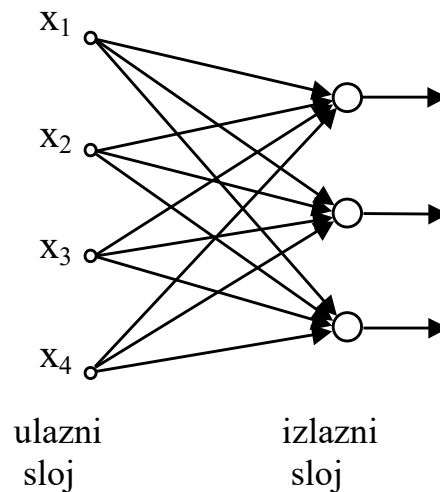


# Kompetitivno učenje

- Učenje bez nadzora
- Neuroni se natječu za priliku da se aktiviraju
- Samo jedan neuron može biti aktivan u bilo kojem trenutku
- Pogodno za probleme klasifikacije
- Tri osnovna elementa kompetitivnog učenja su:
  - Skup jednakih neurona koji imaju slučajno raspoređene težine i zbog toga reagiraju različito na dani skup ulaza
  - Ograničenje “jačine” svakog neurona
  - Mehanizam koji omogućuje natjecanje neurona za pravo na odziv na danu pobudu, tako da je samo jedan neuron (ili jedan po grupi) aktivan u jednom momentu (engl. winner-takes-all neuron)

# Kompetitivno učenje

- U primjeru na slici mreža ima jedan sloj neurona koji su potpuno vezani na sve ulaze



# Kompetitivno učenje

- Da bi neuron pobijedio njegova unutarnja aktivnost  $v_j$  za neki ulazni vektor  $x$  mora biti najveća od svih neurona
- Izlaz  $y_j$  pobjedničkog neurona  $j$  jednak je 1, a svi ostali 0
- Zakon učenja dan je izrazom:

$$\Delta w_{ji} = \begin{cases} \eta(x_i - w_{ji}) & \text{ako je neuron } j \text{ pobijedio} \\ 0 & \text{ako je neuron } j \text{ izgubio} \end{cases}$$

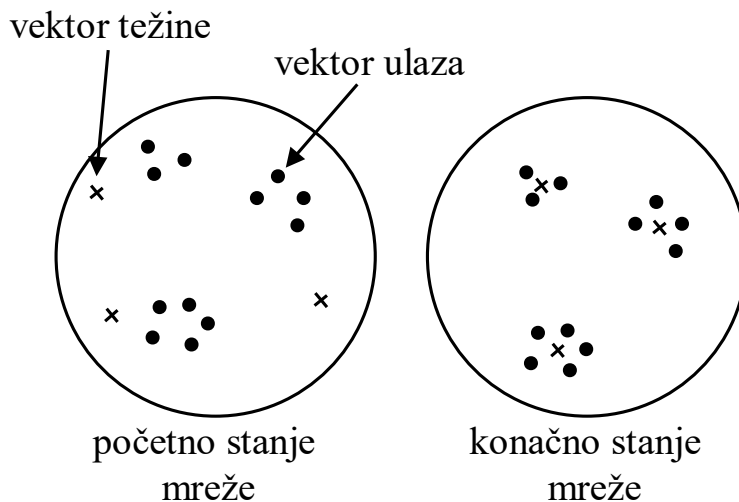
- Pravilo učenja ima efekt pomicanja vektora težine  $w_j$  pobjedničkog neurona prema ulaznom vektoru  $x$

# Primjer kompetitivnog učenja

- Pretpostavimo da svaki ulazni vektor ima normu (duljinu) jednaku jedan tako da se može prikazati kao točka na  $N$ -dimenzionalnoj jediničnoj sferi
- Isto tako pretpostavimo da vektori težina imaju normu jednaku jedan tako da se mogu prikazati kao točke na istoj  $N$ -dimenzionalnoj sferi
- U procesu treniranja ulazni vektori dovode se na ulaz mreže i težina pobjedničkog neurona mijenja se u skladu s zakonom učenja

# Primjer kompetitivnog učenja

- Proces treniranja može se prikazati kao kretanje križića po jediničnoj sferi





# Boltzmannovo učenje

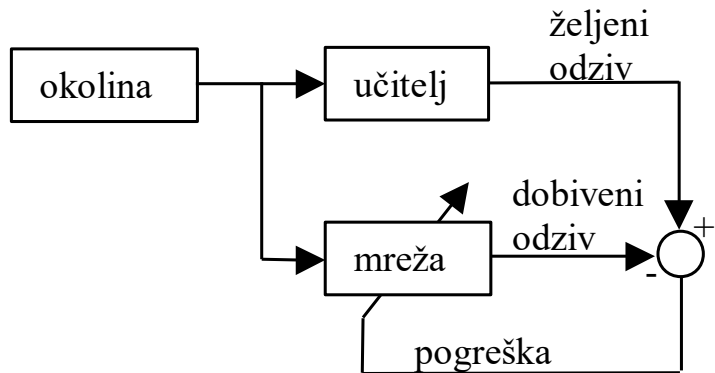
- Koristi se kod Boltzmannovog stroja
- Boltzmannov stroj sastoji se od stohastičkih neurona koji mogu poprimiti jedno od dva stanja: -1 ili +1
- Neuroni su međusobno povezani i postoje povratne veze
- Boltzmannovo pravilo učenja je stohastički algoritam temeljen na teoriji informacija i termodinamici

# Problem nagrađivanja

- Kod proučavanja algoritma učenja za distribuirane sustave važan je problem nagrađivanja (Minski, 1961)
- engl. credit-assignment problem
- Problem nagrađivanja se sastoji u davanju zasluga odnosno krivice za ukupni ishod učenja koji ovisi o većem broju internih odluka sustava za učenje

# Učenje pod nadzorom

- engl. supervised learning
- Glavna karakteristika učenja s nadzorom je prisutnost vanjskog učitelja



# Učenje pod nadzorom

- Učitelj ima znanje o okolini u obliku parova ulaz-izlaz
- Pogreška je razlika između željenog i dobivenog odziva za neki ulazni vektor
- Parametri mreže se mijenjaju pod utjecajem ulaznih vektora i signala pogreške
- Proces se iterativno ponavlja sve dok mreža ne nauči imitirati učitelja
- Nakon što je učenje završilo učitelj više nije potreban i mreža može raditi bez nadzora

# Učenje pod nadzorom

- Funkcija pogreške može biti srednja kvadratna pogreška u ovisnosti od slobodnih parametara mreže
- Funkcija pogreške se može promatrati kao multidimenzionalna ploha pogreške (engl. error surface)
- Bilo koja konfiguracija mreže definirana je parametrima i njoj je pridružena točka na plohi pogreške
- Iterativni proces učenja sastoji se od pomicanja točke prema dolje niz plohu pogreške, tj. prema globalnom minimumu funkcije pogreške

# Učenje pod nadzorom

- Točka na plohi pogreške kreće se prema minimumu na temelju znanja o gradijentu (zapravo ocjene gradijenta)
- Gradijent u nekoj točki plohe je vektor koji pokazuje smjer najbržeg porasta

# Učenje pod nadzorom

- Primjeri algoritama za učenje pod nadzorom su:
  - LMS (engl. least-mean-square) ili algoritam najmanjih kvadrata
  - BP (engl. back-propagation) ili algoritam s povratnom propagacijom pogreške
- Mana učenja pod nadzorom je da bez učitelja mreža ne može naučiti nove strategije koje nisu pokrivene primjerima koji su korišteni za učenje

# Učenje pod nadzorom

- Učenje pod nadzorom može se izvoditi u off-line ili on-line načinu rada
- U off-line načinu učenja:
  - sustav prvo uči
  - kad je učenje gotovo konfiguracija sustava više se ne mijenja i mreža zatim radi na statički način (bez promjena)
- U on-line načinu učenja:
  - sustav za vrijeme rada uči i mijenja svoju konfiguraciju u skladu s novim znanjem
  - učenje se odvija u stvarnom vremenu odnosno mreža je dinamička



# Učenje podrškom

- engl. reinforcement learning
- Učenje podrškom je on-line karaktera
- Uči se ulazno-izlazno preslikavanje kroz proces pokušaja i pogreške gdje se maksimizira indeks kojim se mjeri kvaliteta učenja
- Učenje podrškom rješava problem učenja pod nadzorom:
  - da bez učitelja mreža ne može naučiti nove strategije koje nisu pokrivene primjerima koji su korišteni za učenje

# Učenje podrškom

- Kod učenja podrškom ne postoji učitelj koji određuje kolika je pogreška za određeni par ulaz-izlaz, nego učitelj samo kaže koliko je određeni korak u učenju dobar (daje ocjenu ili podršku)
- Ocjena je skalarna vrijednost
- Kod učenja podrškom funkcija pogreške nije poznata
- Sustav za učenje mora sam ustanoviti “smjer” gibanja kroz prostor učenja metodom pokušaja i pogrešaka (nije poznata informacija o gradijentu kao kod učenja pod nadzorom)

# Thorndike-ov zakon

- engl. Thorndike's law of effect
- Princip učenja podrškom:
  - ako akcije sustava za učenje izazivaju pozitivni efekt tada je veća vjerojatnost da će sustav i ubuduće poduzimati iste akcije
  - inače se vjerojatnost da sustav poduzme te akcije smanjuje

# Učenje bez nadzora

- Kod učenja bez nadzora ili samoorganiziranog učenja nema učitelja koji upravlja procesom učenja
- Koristi se mjera kvalitete znanja koje mreža mora naučiti i slobodni parametri mreže se optimiziraju s obzirom na tu mjeru
- Za učenje bez nadzora može se koristiti kompetitivno pravilo učenja

# Učenje bez nadzora

- Kompetitivni sloj sastoji se od neurona koji se natječu za priliku da odgovore na značajke koji sadrži ulazni vektor
- U najjednostavnijem pristupu koristi se winner-takes-all strategija

# Usporedba učenja sa i bez nadzora

- Najpopularniji algoritam za učenje s nadzorom je povratna propagacija pogreške
- Nedostatak ovog algoritma je problem skaliranja koji se sastoji u tome da s povećanjem broja slojeva mreže i broja neurona vrijeme učenja eksponencijalno raste
- Jedno moguće rješenje ovog problema je da se koristi mreža s više slojeva gdje svaki sloj neovisno uči bez nadzora

# Zadaci

- Problem 2.1.
  - Delta pravilo i Hebbovo pravilo su dva različita načina učenja. Opisati razlike između ovih pravila
- Problem 2.5.
  - Signal iznosa 1 doveden je na ulaz sinaptičke veze čija je početna vrijednost jednaka 1. Treba izračunati promjenu težine s vremenom pomoću:
    - osnovnog Hebbovog pravila s parametrom brzine učenja  $\eta=0.1$
    - modificiranog Hebbovog pravila s  $\eta=0.1$  i  $c=0.1$