Neuronske mreže: Samoorganizirajuće mreže

Prof. dr. sc. Sven Lončarić

Fakultet elektrotehnike i računarstva

https://www.fer.unizg.hr/predmet/neumre_c

Uvod

- Samoorganizirajuće mreže su one koje uče bez nadzora
- Kod učenja bez nadzora ne postoji učitelj
- Nema informacije o tome koju bi vrijednosti izlazi trebali imati za pojedine ulaze
- Mreža mora sama otkriti uzorke, korelacije ili kategorije u ulaznim podacima
- Da bi učenje bez nadzora bilo moguće potrebna je redundancija u ulaznim podacima
- Bez redundancije ulazni podaci izgledali bi kao slučajni šum i učenje ne bi bilo moguće

Uvod

- Vrsta uzorka koju mreža otkriva u ulaznim podacima ovisi o arhitekturi mreže
- Postoji više različitih arhitektura samoorganizirajućih mreža koje se mogu koristiti za razne namjene
- Općenito, samoorganizirajuća mreža može detektirati razna svojstva ulaznih podataka navedena u nastavku

Što predstavljaju izlazi ?

1. Sličnost: Jedan kontinuirani izlaz može pokazivati koliko je novi ulaz sličan dosadašnjim (prosječnim) uzorcima
2. Analiza glavnih komponenti: Proširenje mjerenja sličnosti s jednim na mjerenje s više vektora. Npr. upotreba vlastitih vektora korelacijske matrice ulaznih uzoraka.
3. Grupiranje: Skup binarnih izlaza gdje je samo jedan aktivan u jednom trenutku može nam reći kojoj grupi pripada ulazni uzorak

Što predstavljaju izlazi ?

4. Prototyping: Slično kao grupiranje samo izlaz je predstavnik grupe kojoj pripada ulazni uzorak (asocijativna memorija)
5. Kodiranje: Izlaz je kodirana verzija ulaza, uz korištenje manje bitova i očuvanje što više relevantnih informacija (za kompresiju podataka)
6. Preslikavanje značajki (engl. feature mapping): Ako su izlazni neuroni u nekom geometrijskom rasporedu (npr. dvodimenzionalno polje) i ako je samo jedan aktivan u svakom momentu

Pregled predavanja

- Hebbovo učenje bez nadzora
  - Samoorganizirajuća mreža s jednim neuronom
  - Ojino pravilo učenja bez nadzora
  - Analiza glavnih komponenti
  - Jednoslojna samoorganizirajuća mreža s bez povratnih veza
  - Sangerovo pravilo učenja bez nadzora za jednoslojnu mrežu
- Kompetitivno učenje bez nadzora

Jedan linearni neuron

- Neka je ξ N-dimenzionalni slučajni vektor s komponentama ξ_i, $i = 1, \ldots, N$
- Samoorganizirajuća mreža s jednim linearnim neuronom radi tako da se u svakom koraku generira slučajni vektor ξ i postavi na ulaz mreže
- Nakon što je mreža "vidjela" dovoljan broj uzoraka može na izlazu dati odgovor kako pojedini ulazni uzorak odgovara svojoj distribuciji
- Najjednostavniji je slučaj s jednim *linearnim* neuronom

Jedan linearni neuron

- Izlaz linearnog neurona dan je izrazom:

$$v = \sum_{j=1}^{N} w_j \xi_j = \mathbf{w}^T \boldsymbol{\xi} = \boldsymbol{\xi}^T \mathbf{w}$$

  gdje je **w** vektor težina
- Kad imamo samo jedan neuron onda nam izlaz može biti mjera sličnosti uzorka s ostalima: Što je vjerojatniji određeni ulazni vektor $\boldsymbol{\xi}$ to mora biti veći izlaz *v*
- Da bi to postigli, možemo koristiti Hebbov zakon učenja:

$$\Delta w_i = \eta v \xi_i$$

Ojino pravilo učenja

- Problem Hebbovog učenja je da vektor težina stalno raste i učenje nikad nije gotovo
- Jedno rješenje ovog problema našao je Oja kroz modifikaciju originalnog Hebbovog pravila
- Ojino pravilo učenja bez nadzora:

$$\Delta w_i = \eta v(\xi_i - v w_i)$$

Oja pravilo: Svojstva

- Svojstva Ojinog algoritma učenja su takva da vektor težina konvergira ka vektoru **w** koji ima slijedeća svojstva:

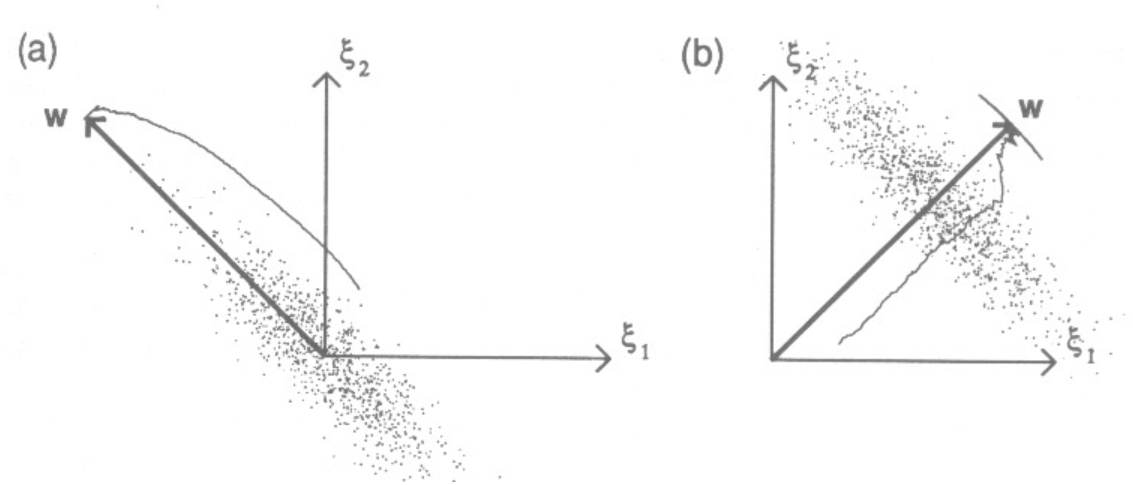
  1. $|\mathbf{w}|=1$

  2. **w** ima smjer maksimalnog vlastitog vektora korelacijske matrice **C** ulaznih vektora

  3. **w** ima smjer koji maksimizira $E[v^2]$

Primjer Oja učenja

- Slika prikazuje promjenu vektora težine za učenje prema Ojinom pravilu



Analiza glavnih komponenti

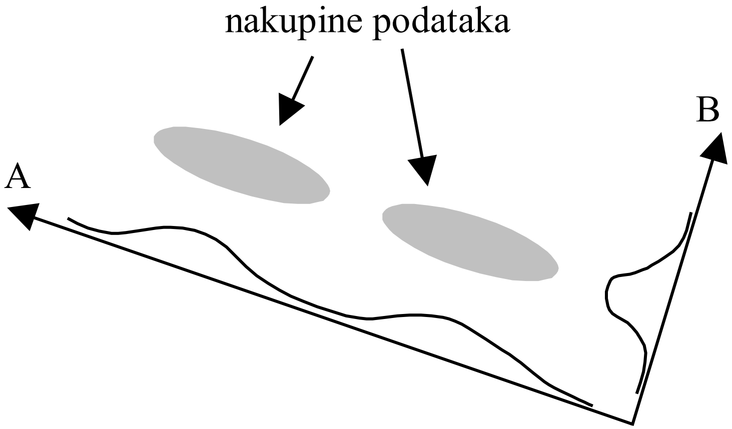
- engl. principal component analysis
- Analiza glavnih komponenti je poznata metoda u raspoznavanju uzoraka i statistici
- Analiza glavnih komponenti poznata je u analizi signala i slika pod nazivom Karhunen-Loeve transformacija

Analiza glavnih komponenti

- Ideja je da se za skup *N*-dimenzionalnih vektora nađe $M<N$ ortogonalnih vektora u smjeru maksimalne varijacije podataka (ulaznih vektora)
- Projekcija podataka iz originalnog *N*-dimenzionalnog prostora u *M*-dimenzionalni obavlja redukciju dimenzionalnosti vektora
- Obično je $M<<N$ tako da su reducirani podaci puno lakši za analizu kada npr. želimo naći grupe (engl. clusters) podataka

Analiza glavnih komponenti

- Grupe podataka je lakše prepoznati kad su podaci projecirani u smjeru velike varijance podataka (A) nego u smjeru male varijance podataka (B)



Analiza glavnih komponenti

- Neka je ξ N-dimenzionalni slučajni vektor
- Korelacijska matrica **C** ovog slučajnog vektora definirana je kao: **C** = $E[\xi\xi^T]$
- Elementi matrice **C** su korelacije parova slučajnih varijabli ξ_i *i* ξ_j : $c_{ij} = E[\xi_i\xi_j]$
- Dijagonalni elementi predstavljaju varijance pojedinih slučajnih varijabli: $c_{ii} = E[\xi_i^2]$
- **C** je simetrična matrica

Analiza glavnih komponenti

- Tvrdnja: Smjer maksimalne varijacije podataka je smjer prve glavne komponente tj. maksimalnog vlastitog vektora (vektora koji odgovara maksimalnoj vlastitoj vrijednosti matrice **C**)
- Dokaz:

$$E[v^2] = E[(\mathbf{w}^T \boldsymbol{\xi})^2] = E[\mathbf{w}^T \boldsymbol{\xi}\boldsymbol{\xi}^T \mathbf{w}] = \mathbf{w}^T \mathbf{C}\mathbf{w}$$

  Za fiksni |**w**| i danu simetričnu matricu **C** poznato je iz linearne algebre da kvadratna forma $\mathbf{w}^T\mathbf{C}\mathbf{w}$ ima maksimalnu vrijednost za vektor **w** koji je u smjeru maksimalnog vlastitog vektora korelacijske matrice **C**

Analiza glavnih komponenti

- Može se pokazati da za ostale glavne komponente vrijedi sličan rezultat kako slijedi
- Druga glavna komponenta je ortogonalna na prvu i predstavlja glavnu komponentu za preostali
  N-1-dimenzionalni vektorski podprostor (u tom podprostoru druga komponenta predstavlja smjer najveće varijacije podataka)
- Druga glavna komponenta je u smjeru vlastitog vektora koji odgovara drugoj najvećoj vlastitoj vrijednosti

Analiza glavnih komponenti: Sažetak

- Neka je A ortogonalna matrica čiji su stupci ortonormalizirani vlastiti vektori korelacijske matrice $C=E[\xi\xi^T]$ ulaznog slučajnog vektora
- Navedeni vlastiti vektori pokazuju glavne komponente, tj. smjerove maksimalne varijacije podataka
- Transformacijom originalnog vektora ξ u novi vektor $y=A^T\xi$ obavljamo projekciju iz originalnog prostora u novi prostor glavnih komponenti

Analiza glavnih komponenti: Sažetak

- Ako zadržimo samo prvih *M<N* komponenti vektora y u novom prostoru onda smo uz najmanju moguću kvadratnu pogrešku prikazali originalni vektor ξ
- Ne postoji niti jedna druga linearna transformacija koja će s M komponenti točnije (u smislu minimalne kvadrante pogreške) prikazati originalni vektor
- Promatranjem samo prvih *M<N* komponenti novog vektora y možemo lakše riješiti problem klasifikacije originalnih uzoraka ξ

Unaprijedna jednoslojna mreža

- Ojino pravilo se može iskoristiti za učenje u mreži s *M* izlaza koja daje prvih *M* glavnih komponenti
- Sanger i Oja su predložili takve mreže 1989
- Mreže su linearne i *i*-ti izlaz dan je izrazom:

$$v_i = \sum_{j=1}^{N} w_{ij}\xi_j = \mathbf{w}_i{}^T \boldsymbol{\xi} = \boldsymbol{\xi}^T \mathbf{w}_i$$

gdje je $\mathbf{w}_i$ *N*-dimenzionalni vektor težina za *i*-ti neuron

Sangerovo pravilo učenja

- Kako je ranije rečeno, Ojino pravilo učenja za mrežu s jednim linearnim neuronom je:

$$\Delta w_i = \eta v(\xi_i - v w_i)$$

- Sangerovo pravilo učenja za jednoslojnu mrežu s *M* linearnih neurona definirano je izrazom:

$$\Delta w_{ij} = \eta v_i(\xi_j - \sum_{k=1}^{i} v_k w_{kj})$$

- Za prvi neuron ovo je pravilo jednako Ojinom, dakle njegov vektor težina konvergira ka prvoj glavnoj komponenti

Sangerovo pravilo učenja

- Svaki od M linearnih neurona daje jedan izlaz koji odgovara jednog glavnoj komponenti
- Izlazi su prvih M glavnih komponenti za skup slučajnih ulaznih vektora

Kompetitivno učenje bez nadzora

- Kod Hebbovog učenja može istovremeno biti aktivno više neurona
- Kod kompetitivnog učenja samo je jedan neuron (ili jedan neuron u grupi) aktivan u jednom trenutku
- Zato se ovakve mreže zovu engl. winner-takes-all (WTA) mreže
- Namjena ovakvih mreža je grupiranje ili kategorizacija podataka (slični ulazni vektori klasificiraju se u istu grupu)
- Mogu se koristiti za klasifikaciju uzoraka (objekata) u računalnom vidu ili za vektorsku kvantizaciju

Winner-takes-all mreža

- U najjednostavnijem slučaju WTA mreža ima jedan sloj izlaznih neurona od kojih je svaki spojen na ulaze u mrežu
- Samo jedan neuron se može aktivirati (pobjednik)
- Pobjednik je neuron s najvećom aktivacijom:

$$h_i = \sum_j w_{ij} \xi_j = \mathbf{w}_i{}^T \boldsymbol{\xi}$$

- Za pobjednički neuron *k* vrijedi:

$$\mathbf{w}_k^T \boldsymbol{\xi} \geq \mathbf{w}_i^T \boldsymbol{\xi} \quad \forall i$$

Winner-takes-all mreža

- Ako su težine normirane tako da je npr. $|\mathbf{w}_i|=1$ onda je prethodni izraz ekvivalentan izrazu:

$$\left|\mathbf{w}_k - \boldsymbol{\xi}\right| \leq \left|\mathbf{w}_i - \boldsymbol{\xi}\right| \quad \forall i$$

- Drugim riječima pobjednik je neuron s težinom najbližom ulaznom vektoru ξ
- Problem je sada kako odrediti težine $\mathbf{w}_i$
- Jedan način učenja koji se može koristiti je standardno pravilo kompetitivnog učenja

Kompetitivno učenje

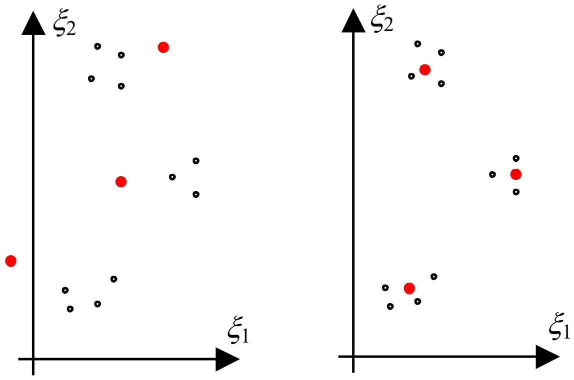
- Vektori težina se inicijaliziraju na neke slučajne početne iznose
- Na ulaz mreže se dovode ulazni vektori prema slučajnom redoslijedu
- Za svaki ulazni vektor ξ odredi se neuron pobjednik i njegove težine se modificiraju prema pravilu:

$$\Delta w_{kj} = \eta(\xi_j - w_{kj})$$

- Standardno pravilo kompetitivnog učenja pomiče vektor težina u smjeru ulaznog vektora

Kompetitivno učenje

- Primjer s tri neurona i dvodimenzionalnim vektorima
- Vektori težina su prikazani crveno, a ulazni vektori crno
- Lijeva slika prikazuje početno stanje
- Desna slika prikazuje stanje nakon završetka učenja



Mjerenje kvalitete grupiranja

- Često je potrebno na neki način izmjeriti kvalitetu grupiranja koja je postignuta učenjem bez nadzora
- Standardnom pravilu za kompetitivno učenje pridružena je funkcija cijene definirana izrazom:

$$E = \frac{1}{2} \sum_{l=1}^{L} \sum_{i=1}^{C} \sum_{j=1}^{N} m_{il} (\xi_j^{\,l} - w_{ij})^2$$

gdje je *L* broj vektora, *C* broj grupa, *N* dimenzija vektora, a **M**=$\{m_{il}\}$ matrica pripadnosti grupama koja pokazuje da li uzorak $\xi^{\,l}$ pripada grupi *i* :

$$m_{il} = \begin{cases} 1, & \xi^l \in C_i \\ 0, & \xi^l \notin C_i \end{cases}$$

Mjerenje kvalitete grupiranja

- Ekvivalentan zapis funkcije cijene koja pokazuje kvalitetu grupiranja je:

$$E = \frac{1}{2}\sum_{l=1}^{L}\left|\boldsymbol{\xi}^{l} - \mathbf{w}_{k(l)}\right|^{2}$$

  gdje je $k(l)$ indeks pobjedničkog neurona za l-ti ulazni uzorak ξ^l
- Gornji izraz pokazuje sumu kvadrata udaljenosti svih ulaznih vektora od centara grupa kojima pripadaju
- To je mjera kvalitete grupiranja jer što je taj broj manji vektori su bolje koncentrirani oko centara grupa

Primjene kompetitivnog učenja

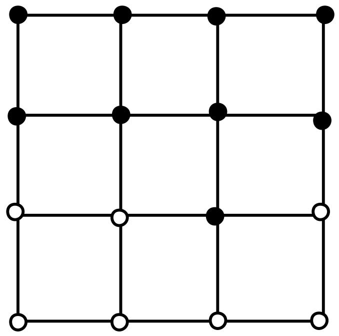
- Primjeri primjena kompetitivnog učenja su:
  - Biparticija grafa
  - Kvantizacija vektora za kompresiju
  - Klasifikacija vektora za prepoznavanje objekata

Primjer: Biparticija grafa

- Problem biparticije grafa sastoji se u podjeli grafa u dva podgrafa sa što manje grana koja ih povezuju
- Problem biparticije može se riješiti mrežom koja ima jedan binarni ulaz za svaki čvor grafa i dva izlazna neurona za indikaciju particije
- Ulazni skup vektora formira se tako da za svaku granu aktiviramo binarne ulaze koji odgovaraju incidentnim čvorovima
- Upotrebom kompetitivnog učenja mreža nauči podijeliti graf u dva dijela, tako da za svaki ulaz (granu) daje odgovor kojoj particiji ta grana pripada

Primjer: Biparticija grafa

- Primjer rezultata biparticije grafa neuronskom mrežom
- Crni čvorovi pripadaju jednoj particiji, a bijeli drugoj

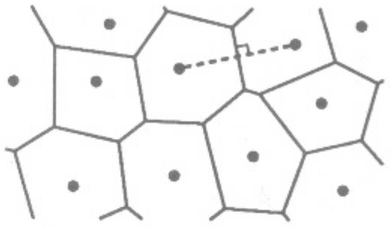


Primjer: Kvantizacija vektora

- Problem se sastoji u tome da se kategorizira dani skup vektora u M klasa i zatim svaki vektor predstavi klasom kojoj pripada
- Na taj način se postiže kompresija podataka
- Prenosi se ili pohranjuje samo indeks klase u kodnoj stranici a ne cijeli vektor
- Kodna stranica sadrži po jedan vektor predstavnik pojedine klase
- Za svaki ulazni vektor traži se najbliži predstavnik u smislu Euklidske udaljenosti (to vodi na Voronoi teselaciju prostora)

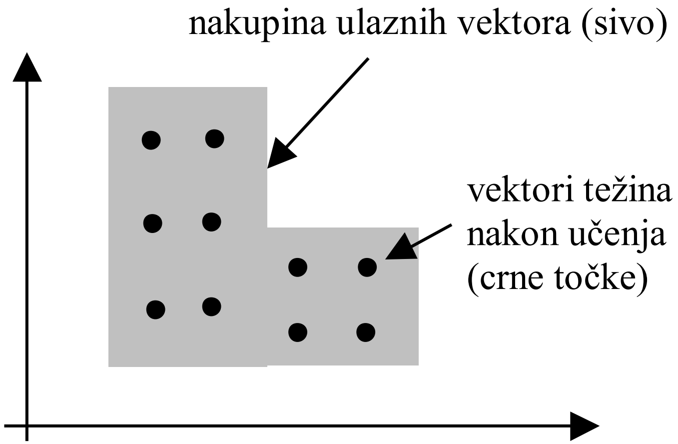
Voronoi teselacija prostora

- Kod Voronoi teselacije prostor je podijeljen u poligonalne regije koje su određene prototipnim vektorima
- Svaka regija predstavlja skup točaka koje su najbliže (u smislu Euklidskom udaljenosti) pripadnom prototipnom vektoru
- Granice su okomite na linije koje povezuju centre regija



Primjer: Kvantizacija vektora

- Ilustracija upotrebe kompetitivnog učenja za vektorsku kvantizaciju



Preslikavanje značajki

- engl. feature mapping
- Do sada nismo obraćali pažnju na geometrijski raspored kompetitivnih izlaznih neurona
- Često je korisno rasporediti izlazne neurone u 1-D raspored ili u 2-D raspored
- Tada možemo zamisliti mreže gdje bliski ulazi odgovaraju bliskim izlazima odnosno ako ulazi nisu blizu onda ni izlazi ne smiju biti blizu
- To je onda preslikavanje koje čuva topologiju
- Korisno pri vizualizaciji višedimenzionalnih prostora

Kohonenov algoritam

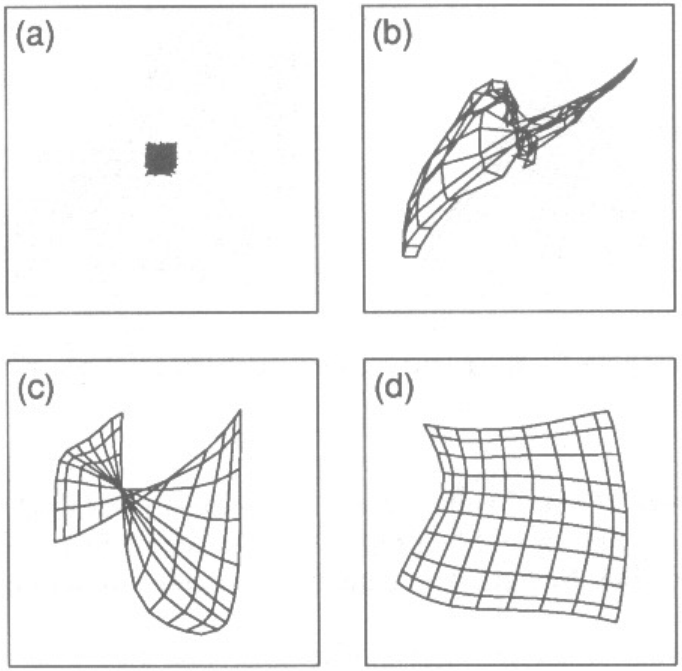
- Kohonen je koristio kompetitivnu mrežu i način učenja
- Kohonenov algoritam učenja podešava težine u nekoj okolini pobjedničkog neurona:

$$\Delta w_{ij} = \eta \Lambda(i,k)(\xi_j - w_{ij})$$

gdje je $\Lambda(i,k)$ funkcija koja se smanjuje s udaljenošću od pobjedničkog neurona k i koja određuje susjedstvo unutar kojeg se vrši učenje

Primjer 2-D preslikavanja značajki

- Kohonenovo 2-D preslikavanje značajki iz kvadratne regije ravnine na polje od 10×10 neurona
- Prikazano je četiri stanja u procesu učenja, gdje je ulaz 2-D slučajni vektor



Zaključak

- Samoorganizirajuće mreže koriste učenje bez nadzora
- Učenje bez nadzora korisno je u velikom broju aplikacija kad nije raspoloživa informacija o željenom izlazu
- Primjene uključuju: klasifikaciju objekata ili uzoraka, vektorsku kvantizaciju za upotrebu u kompresiji, grupiranje (clustering) za analizu podataka, preslikavanje značajki