

Neizrazito, evolucijsko i neuroračunarstvo Samoorganizirajuće neuronske mreže.

prof.dr.sc. Bojana Dalbelo Bašić prof.dr.sc. Marin Golub
dr.sc. Marko Čupić

Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu
Akademska godina 2013./2014.

12. prosinca 2013.

Vrste učenja neuronskih mreža

- Učenje s učiteljem (engl. *supervised learning*)
 - Tipičan primjer je FF-ANN i algoritam učenja *Backpropagation*
 - Karakteristike: poznati su ulazi i odgovarajući izlazi; mreža uči preslikavanje ulaz-izlaz
 - Mreža obavlja **funkcijsku regresiju** ili **klasifikaciju** podataka
- Učenje bez učitelja (engl. *unsupervised learning*)
 - Dostupni su samo ulazni podatci; nema specificiranog preslikavanja koje mreža mora naučiti
 - Mreža može analizirati sličnost ulaznih podataka
 - Pojedini neuroni mogu "željeti" biti različiti od drugih neurona u mreži
 - Mreža obavlja **grupiranje** (engl. *klastering*)

Natjecanje

Opažanje (S. Grossberg): **mehanizam natjecanja** dovodi do optimizacije funkcije elemenata distribuiranog sustava (društvo, biologija, računarski sustavi, ...)

- natjecanje vodi do optimizacije na lokalnoj razini bez potrebe za globalnom kontrolom: vrlo česta pojava u biologiji
- svi procesni elementi (PE) mreže dobivaju identičan input, ali se natječu:
 - lateralnim vezama
 - formulacijom principa učenja
- procesni elementi specijaliziraju se za određena područja ulaznog prostora
 - neuroni mreže počinju oslikavati strukturu ulaznih podataka

Natjecanje

Kod učenja neuronskih mreža procesni se elementi specijaliziraju na određeni način:

- algoritam Backpropagation postiže specijalizaciju neurona indirektno i uz velike troškove (spora konvergencija)
- **kompetitivne mreže specijaliziraju se postupkom natjecanja**

Prvi primjer: mreža Winner-takes-all

Što želimo?

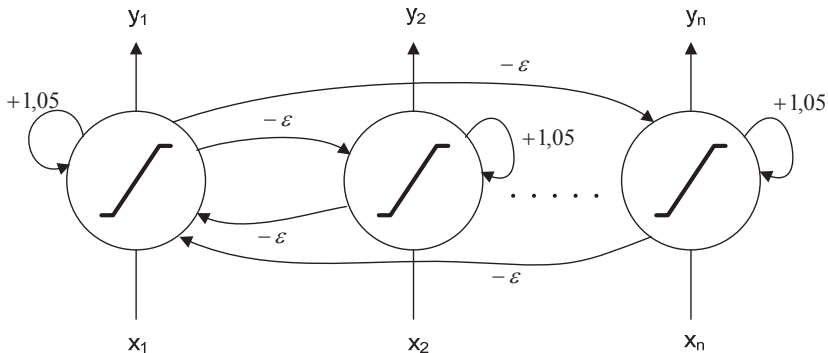
- želimo ostvariti neuronsku mrežu koja ima n ulaza i n izlaza
- svaki ulaz ograničen je na interval $[0, 1]$
- želimo na izlazu generirati vrijednost 1 uz onaj ulaz na koji je dovedena najveća vrijednost:
npr. $(0.2, 0.9, 0.3) \Rightarrow (0, 1, 0)$
- treba riješiti decentralizirano

Ideja rješenja:

- dinamička mreža izgrađena od upravo n neurona
- svaki neuron vidi svoj ulaz
- svaki neuron sam sebe potiče da generira veći izlaz
- svaki neuron guši izlaze drugih neurona

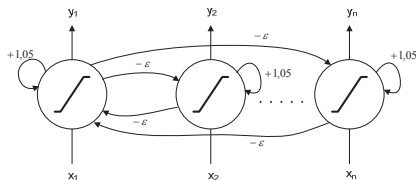
Prvi primjer: mreža Winner-takes-all

y_1, \dots, y_n : izlazi



x_1, \dots, x_n : ulazi

Prvi primjer: mreža Winner-takes-all



- N procesnih elemenata
- samo onaj element čiji je ulaz $x_i > x_j \quad \forall j \neq i$ na izlazu daje 1, svi ostali 0
- postoji **jedinična pozitivna povratna veza** elementa
- postoje **negativne lateralne veze** veličine ϵ ($0 < \epsilon < \frac{1}{N}$) prema svim ostalim procesnim elementima
- nakon prelaznog razdoblja izlaz se stabilizira

Prvi primjer: mreža Winner-takes-all

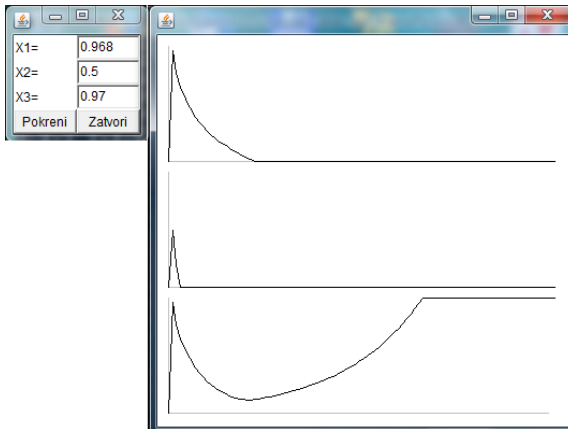
- Mreži se u prvom trenutku predoči ulazni uzorak, i zatim se makne

... simulacija ...

- Lateralne negativne veze sve izlaze tjeraju eksponencijalno prema nuli
- PE s najvećim ulaznim uzorkom sve manje osjeća $-\epsilon$ veze i počinje ponovno rasti (zbog $+1,05$ veze) pa još više guši preostale procesne elemente

Prvi primjer: mreža Winner-takes-all

Nakon kraćeg prelaznog razdoblja samo jedan od neurona za izlazu će generirati vrijednost 1 dok svi ostali neuroni imaju izlaz koji je pritegnut na 0:



Četiri zahtjeva na samoorganizirajuće NN

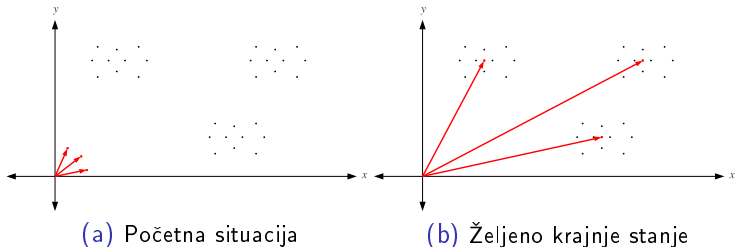
- 1 Težine u neuronu moraju biti predstavnici razreda uzoraka, tako da svaki neuron predstavlja drugi razred.
- 2 Ulazni uzorak predočava se svim neuronima i svaki je neuron izlazni. Vrijednost izlaza je mjera sličnosti između ulaznog uzorka i uzorka pohranjenog u neuronu.
- 3 Koristi se natjecateljska strategija učenja (engl. *competitive learning strategy*) koja odabire neuron s najvećim izlazom.
- 4 Postoje metode poticanja najvećeg izlaza.

Podjele s obzirom na tip natjecanja

- ① **Čvrsto natjecanje** (engl. *hard competition*) - postoji samo jedan pobjednik i samo on uči.
 - ② **Meko natjecanje** (engl. *soft competition*) – postoji samo jedan pobjednik ali u određenoj mjeri uči i njegova okolina.
-
- Mreže *INSTAR* i *OUTSTAR* predstavnici su mreža koje koriste meko natjecanje.
 - Kohonenov SOM predstavnik je mreže koja koristi čvrsto natjecanje.

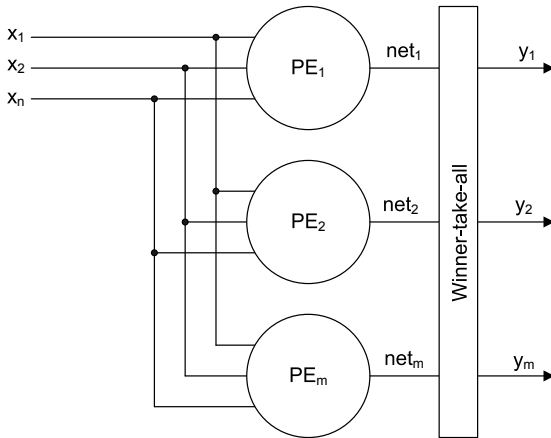
Željena funkcionalnost

Grupiranje podataka: ulazi su vektori n -dimenzijskog prostora, svaki od m neurona kroz svoje težine predstavlja centar jednog razreda uzoraka.



Izlazi bi trebali biti binarni: kada mreža dobije ulaz, neuron u čiji razred pripada taj uzorak trebao bi generirati na izlazu vrijednost 1; svi ostali trebaju imati izlaz postavljen na 0.

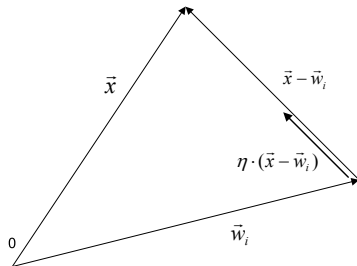
Arhitektura mreže



Rad mreže

- u prvom sloju neurona svaki je neuron ulazni i dobiva cjelokupni uzorak
- izlaz tog sloja je mjera sličnosti pohranjenog i predloženog uzorka: što je sličnost veća, izlaz je veći
- konceptualno, na prvi sloj dodan je WTA-sloj kako bi se utvrdio neuron pobjednik
- taj neuron (pobjednik) uči modificirajući svoje težine tako da postanu sličnije predloženom uzorku

Pravilo učenja



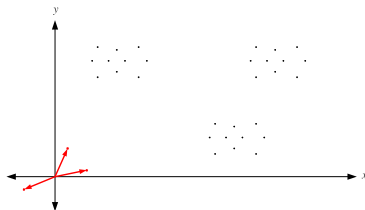
$$\vec{W}_j \leftarrow \vec{W}_j + \eta \cdot (\vec{X} - \vec{W}_j) \cdot y_j$$

pri čemu je $y_j = 1$ ako je j -ti neuron pobjednik; inače je 0.
Raspisano po komponentama:

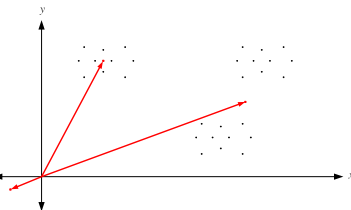
$$w_{ij}(k+1) \leftarrow w_{ij}(k) + \eta \cdot (x_i - w_{ij}) \cdot y_j.$$

Pravilo učenja

- kao mjera sličnosti može se koristiti kosinus kuta ili pak mjera izvedena iz udaljenosti vektora
 - kosinus uzorke preslikava na obod jedinične kružnice: ponekad nije najpovoljniji odabir
- kako ne znamo kako inicijalno odabrati težine, postupak može rezultirati mrtvim neuronima
 - rješenje je dodavanje *savjesti*: pogledati u knjizi detaljnije



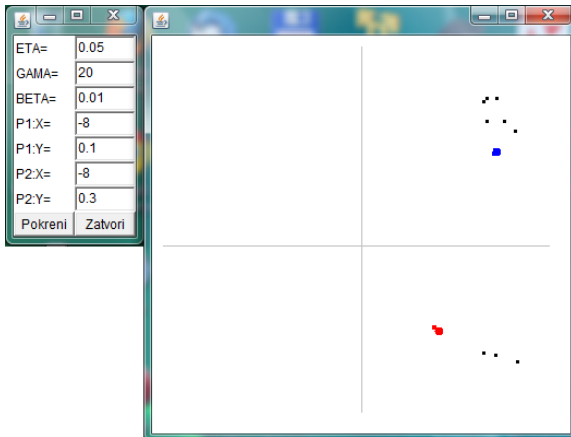
(c) Početna situacija



(d) Krajnje stanje (mrtav neuron)

Prikaz rada

... simulacija ...

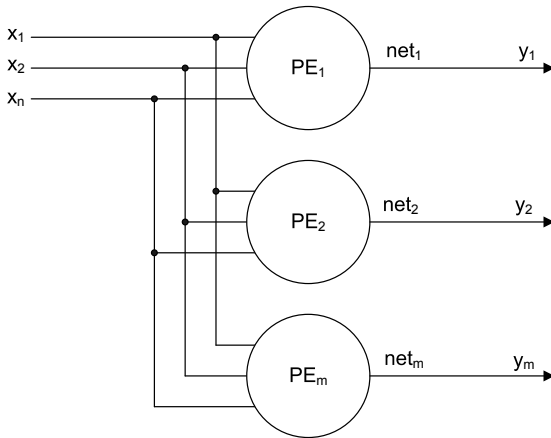


Željena funkcionalnost

Zadaća mreže OUTSTAR je pamćenje i rekonstrukcija pohranjenih uzoraka.

- Mreža ima n ulaza pri čemu ulazi moraju biti binarni (0/1) i samo jedan od ulaza smije biti postavljen na 1.
- Idejno, mrežu učimo sljedećemu: *kada je ovaj ulaz u 1, na izlazu rekonstruiraj ovaj podatak; kada je ovaj ulaz u 1, na izlazu rekonstruiraj ovaj podatak...*
- Kako svaki neuron računa *net* kao težinsku sumu ulaza i kako je ulaz ograničen na binarni uz samo jedan ulaz postavljen na 1, pojedine težine svakog neurona pamte pojedine komponente uzoraka koje želimo pohraniti.

Arhitektura mreže



Učenje mreže

Označimo s w_{ij} i -tu težinu j -tog neurona. Neka je na t -tom ulazu dovedena vrijednost 1.

- Izlaz mreže tada će biti $(w_{t1}, w_{t2}, \dots, w_{tm})$.
- Željeni izlaz mreže dan je kao uzorak za učenje (y_1, y_2, \dots, y_m) .
- Postupkom učenja želimo težinu w_{tj} približiti željenom izlazu za j -ti neuron: y_j .
- Stoga je formula prema kojoj se radi učenje:

$$w_{ij} \leftarrow w_{ij} + \eta (y_j - w_{ij}) \cdot x_i$$

- Ako za isti ulaz imamo više željenih izlaza, mreža će naučiti rekonstruirati centar predložene grupe izlaza.

Željena funkcionalnost

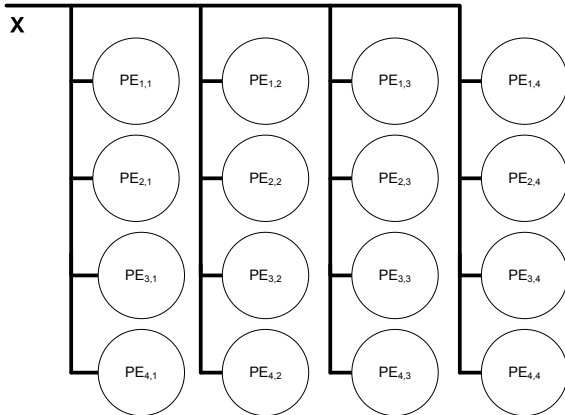
Želimo topološki poduprto grupiranje podataka.

- Želimo funkcionalnost koja je slična mreži INSTAR.
- Razlika je što na strukturu neurona želimo utisnuti topologiju (definirati susjednost) kako bismo postigli da topološki susjedni (ili bliski) neuroni postanu predstavnici razreda koji su međusobno bliski u prostoru uzoraka.
- Kako bi to postigao, Kohonenov SOM koristi *meko* natjecanje: uči pobjednik i blisko susjedstvo ali što je susjed dalje od pobjednika, to u manjoj mjeri korigira svoje težine.

Primjeri topologija

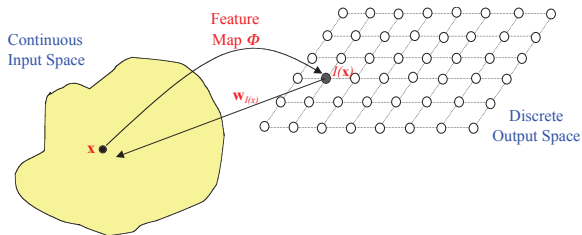
- jednodimenzijska, pri čemu su neuroni međusobno povezani u lanac pa svaki neuron osim krajnjih ima dva direktna susjeda (jednog lijevo i jednog desno);
- dvodimenzijska rešetka, pri čemu svaki neuron osim rubnih ima četiri direktna susjeda (jednog iznad, jednog ispod, jednog lijevo i jednog desno)

Arhitektura mreže



Što radi SOM

SOM obavlja preslikavanje uzoraka iz kontinuiranog ulaznog prostora u diskretni izlazni prostor. To će preslikavanje imati nekoliko vrlo zanimljivih svojstava.



Pravilo učenja

Neka je $d(i, j)$ udaljenost između dva neurona u nametnutoj topologiji (**ne** udaljenost pohranjenih uzoraka!).

- Pravilo ažuriranja težina koje se koristi kod samoorganizirajuće mape glasi:

$$\vec{w}_i \leftarrow \vec{w}_i + \eta(n) \cdot \Lambda_{i,i^*}(n) \cdot (\vec{X} - \vec{w}_i)$$

- Pri tome je u iteraciji n s $\Lambda_{i,i^*}(n)$ definirana funkcija koja određuje mjeru u kojoj uči neuron i ako je pobjednik neuron i^* : ta mjera opada s porastom udaljenosti između neurona i, i^* ; npr.

$$\Lambda_{i,j}(n) = e^{\frac{-d_{i,j}^2}{2\sigma^2(n)}}$$

Pravilo učenja

Postupak učenja odvija se u dvije faze.

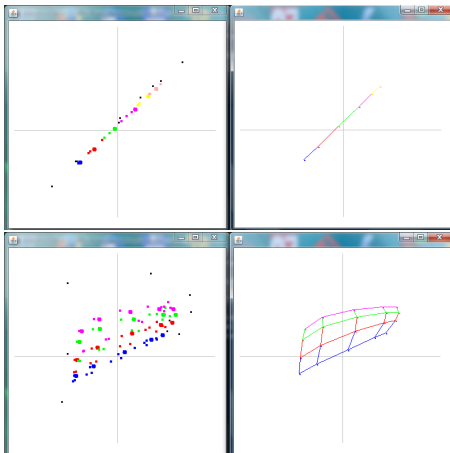
- *Inicijalna faza* osigurava grubo podešavanje samoorganizirajuće mape: stope učenja su velike, susjedstva su široka. Stopa učenja i susjedstvo mijenjaju se prema:

$$\eta(n) = \eta_0 \left(1 - \frac{n}{N_0 + K} \right) \quad \sigma(n) = \sigma_0 \left(1 - \frac{n}{N_0} \right)$$

- *Faza finog podešavanja* traje bitno dulje od inicijalne faze; susjedstvo je malo; stopa je mala.

Prikaz rada

... simulacija ...



Svojstva SOM-a

- 1 Preslikavanje koje je naučio SOM odnosno skup naučenih težina $\{\vec{w}_i\}$ predstavlja dobru aproksimaciju uzoraka iz ulaznog prostora. Taj je skup manje dimenzionalnosti.
- 2 Preslikavanje naučeno od SOM-a je topološki uređeno u smislu da prostorni položaj neurona u mreži odgovara prostornom odnosu uzoraka u ulaznom prostoru.
- 3 Oslikavanje raspodjele uzoraka: razredi koji imaju veću gustoću uzoraka u ulaznom prostoru dobit će više neurona koji će ih predstavljati u SOM-u.

Ilustracija primjene SOM-a

Kohonen: *Phonetic typewriter for the Finnish language*

- Ulaz: govor
- Izlaz: pisani tekst
- Arhitektura: 2D polje od 96 PE-a
- Svaki PE dobiva na 15 ulaza dijelove spektra govora
- Nakon procesa samoorganizacije uočava se grupiranje sličnih verzija istog fonema
- Prepoznavanje: praćenje trajektorije