

Neuronske mreže: Committee Machines

Prof. dr. sc. Sven Lončarić
Doc. dr. sc. Marko Subašić

Fakultet elektrotehnike i računarstva
Sveučilište u Zagrebu

http://www.fer.hr/predmet/neumre_b

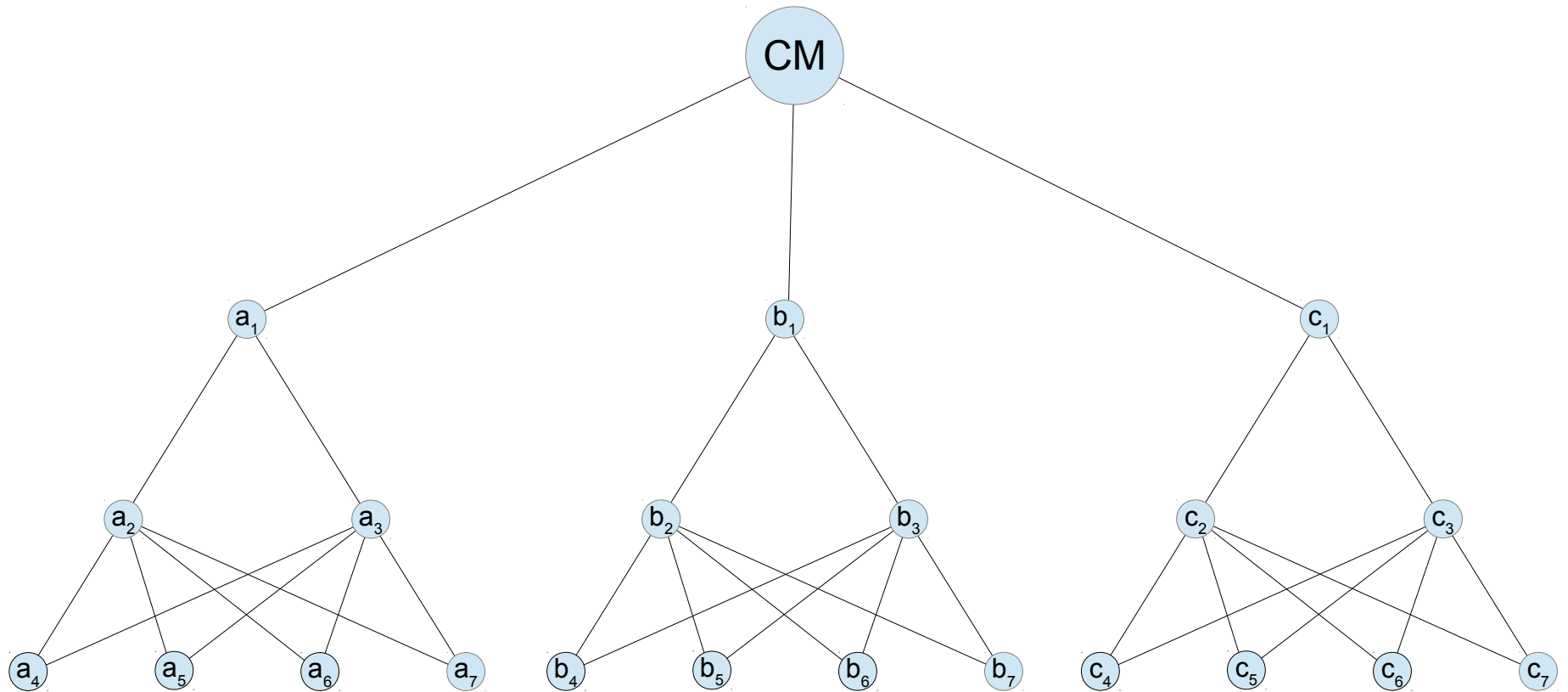
Pregled predavanja

- Kalsifikacija committee mašina
- Statičke strukture
 - Grupno usrednjavanje
 - Boosting
- Dinamičke strukture
 - Mješavina eksperata
 - Hijerarhiska mješavina eksperata

Uvod

- Kompleksni probleme teško može riješiti samo jedan ekspert
 - Takav ekspert bi treba biti vrlo "jak"
- Rješenje: kombiniranje "mišljenja" više "slabih" eksperata
- Cilj je dobiti konačno "mišljenje" koje je točnije od "mišljenja" svakog pojedinog eksperta
- Fuzija znanja
- Modularnost

Committee Machines



Committee Machines

- Konačni rezultat dosta nalikuje na jednu kompleksnu neuronsku mrežu
 - Pogotovo ako su pojedini eksperti isto neuronske mreže
- Pristup je otporniji na problem pretreniranja
- Omogućuje segmentaciju problema i specijalizaciju pojedinih eksperata
 - Pretreniranje je poželjno!?
- Dodatna sloboda u načinu kombiniranja pojedinih eksperata

Dvije glavne kategorije

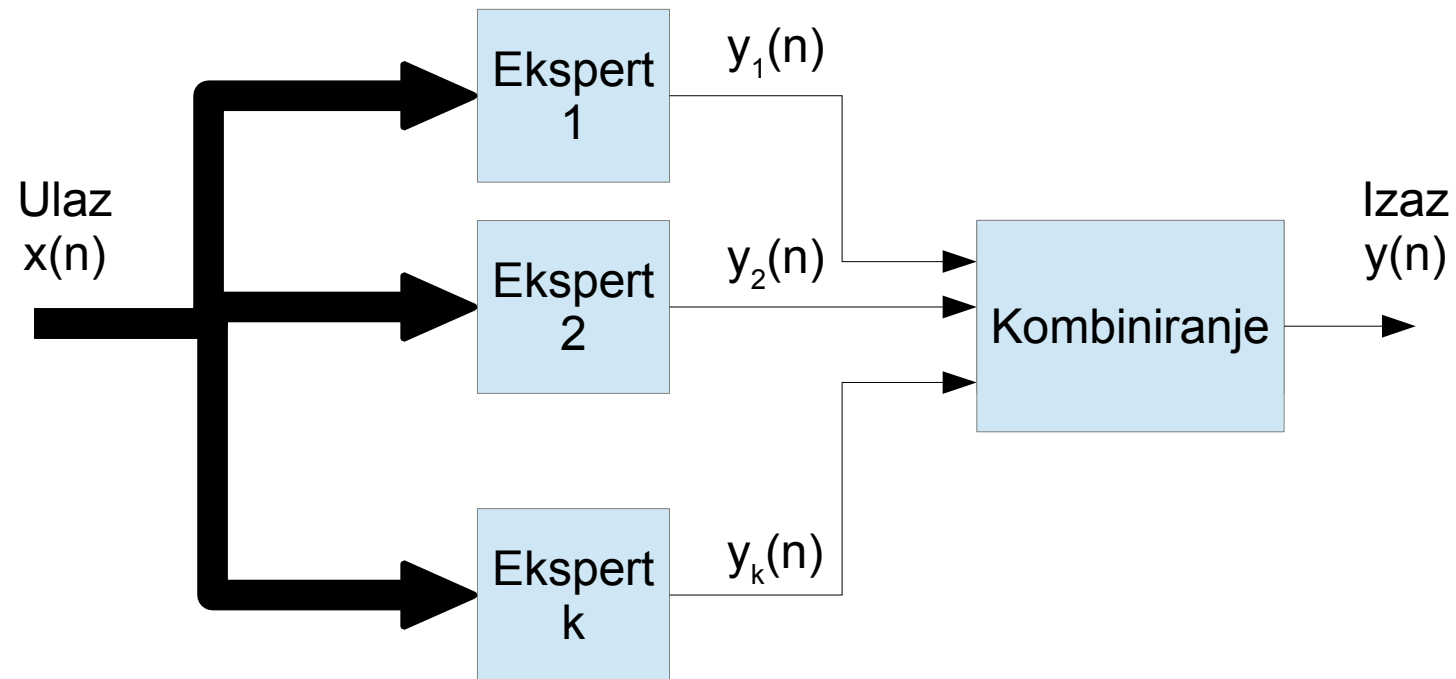
1. Statička struktura – kombiniranje eksperata se odvija bez utjecaja ulaznih informacija

- Grupno usrednjavanje – linearna kombinacija mišljenja eksperata
- Pojačavanje (boosting) – slabi eksperti se kombiniraju u jednog jakog eksperta

2. Dinamička struktura – ulazne informacije utječu na način kombiniranja

- Mješavina eksperata – nelinearno kombiniranje
- Hijerarhijska mješavina eksperata – višekratno nelinearno kombiniranje/odbacivanje

Statička struktura



Grupno usrednjavanje

- Motivacija:
 - Treniranje svih eksperata i usrednjavanja kao jedne mreže traje duže od paralelnog treniranja pojedinih eksperata
 - Rizik od pretreniranja je veći ako je mreža veća i kompleksnija

Grupno usrednjavaње

- Svi eksperti treniraju se na istim podacima
- Može se pokazati da je u slučaju kombiniranja više eksperata, pojedine eksperte dobro i pretrenirati
- Negativni efekti pretreniranja umanjiti će se kombiniranjem

Pojačavanje (Boosting)

- Generalna metoda za poboljšanje bilo kojeg na učenju temeljenog algoritma
- Pojedini eksperti treniraju se na drugačijim skupovima podataka
- Pojedini eksperti nazivaju se i slabi kalsifikatori ili slabi učenici
 - Poželjno je da su slabi – da nisu "predobri"

Slabi učenik

- Jaki učenik
 - Prihvatljiva vjerojatnost uspjeha je malo manje od 100%
- Slabi učenik
 - Prihvatljiva vjerojatnost uspjeha je malo više od 50%
 - Tek malo bolje od bacanja novčića
- Pojačanje iz više slabih učenika radi jednog jakog kroz njihovo kombiniranje

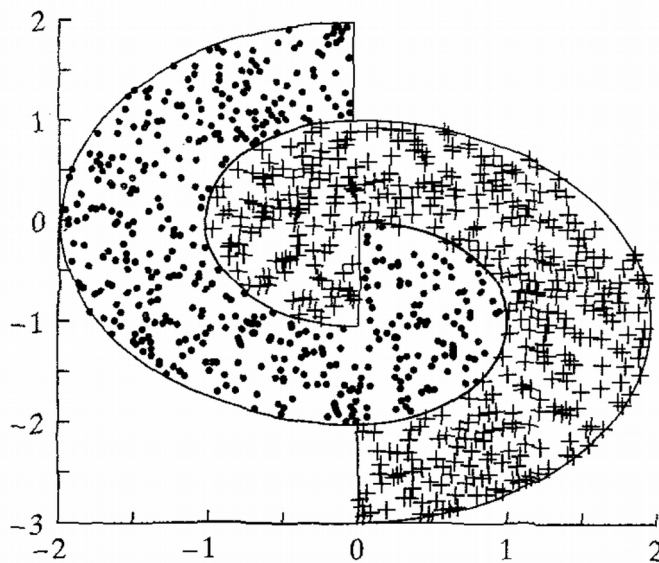
Varijante pojačavanja

1. Pojačavanje filtriranjem – odbacivanje ili prihvatanje pojedinih uzoraka za treniranje
 - Mali memorijski zahtjevi
2. Pojačavanje uzorkovanjem – višekratno uzorkovanje uzoraka za učenje ovisno o distribuciji vjerojatnosti tijekom učenja
3. Pojačavanje težinama – pridjeljivanje težina uzorcima za učenje

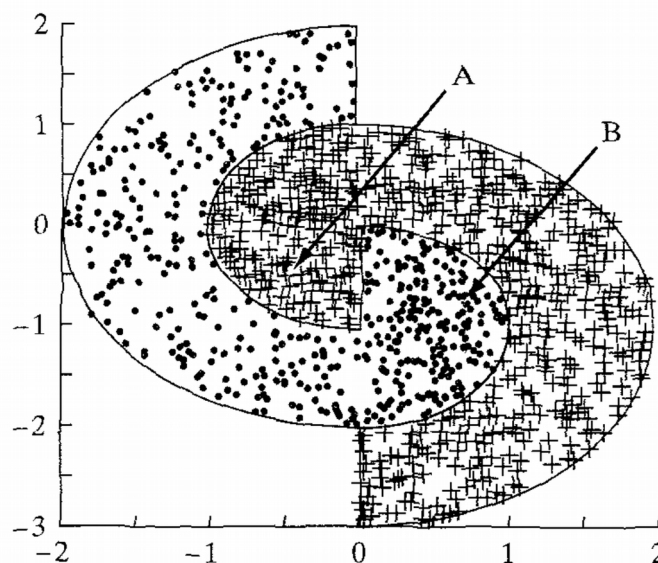
Pojačavanje filtriranjem

- Pretpostavlja veći skup za treniranje
- Tri slaba učenika
- 1. se trenira sa N_1 uzoraka
- 2. se trenira sa N_1 nekorisćenih uzoraka na kojima 1. griješi 50%
 - Slučajno se odabiru uzorci na kojima je 1. učenik pogodio i oni na kojima je pogriješio prebiranjem N_2 uzoraka – traži se da je 1. imao ukupnu točnost veću od 50%
 - Stvara se drugačija distribucija skupa učenja za 2. učenika
- 3. se trenira sa N_1 nekorisćenih uzoraka za koje 1. i 2. učenik daju različite ocjene
 - Prebire se N_3 uzoraka
- Ukupni broj potrebnih uzoraka je $N_1 + N_2 + N_3$

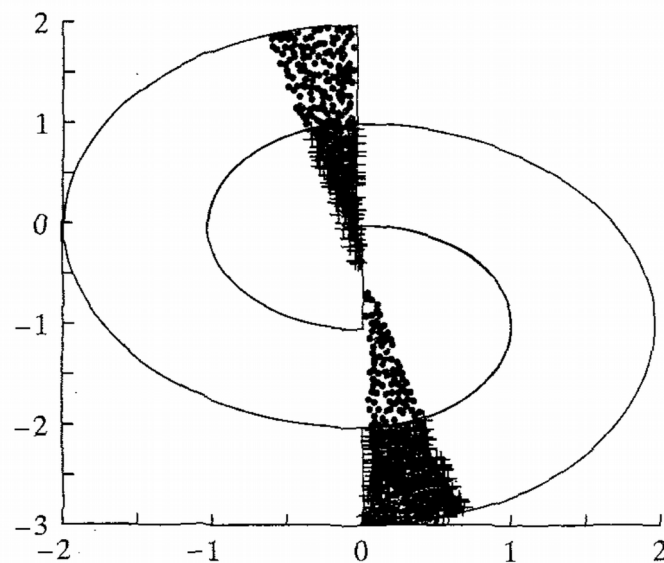
Pojačavanje filtriranjem



(a)

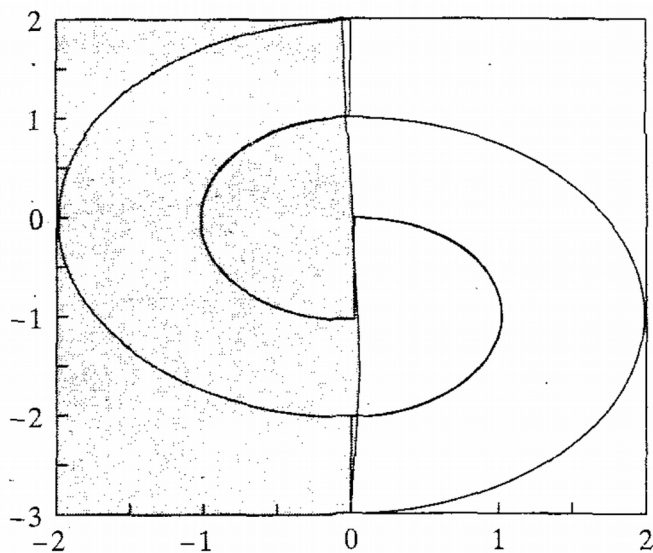


(b)

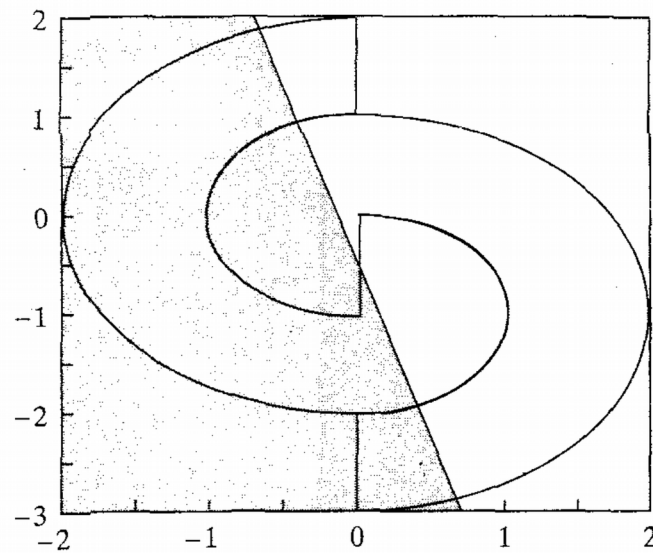


(c)

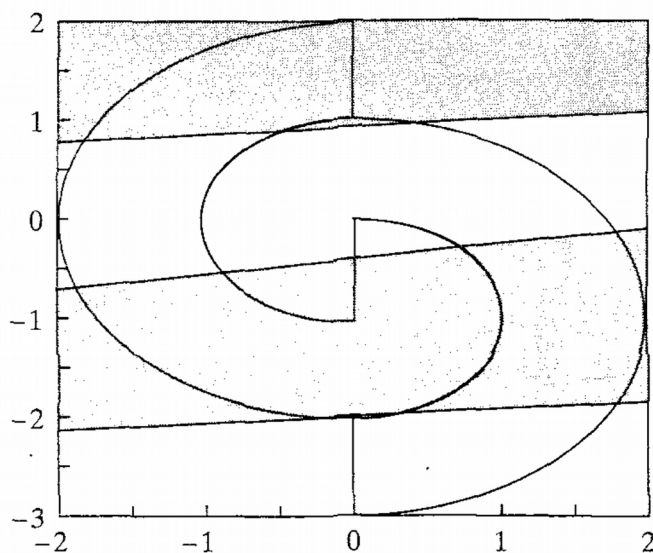
Pojačavanje filtriranjem



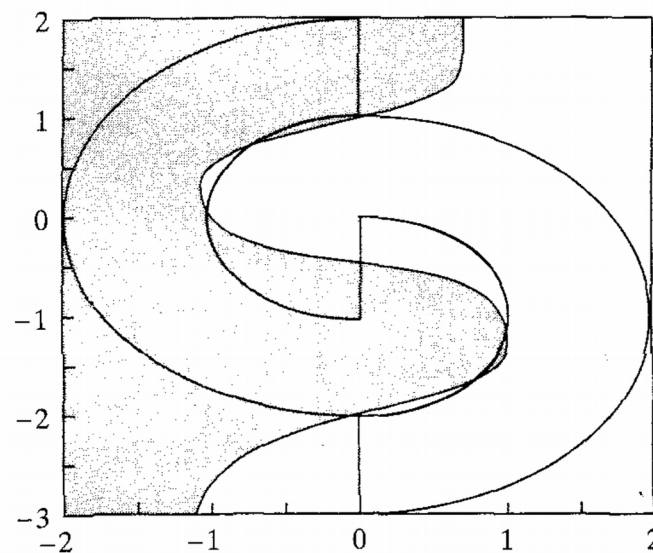
(a)



(b)



(c)



(d)

Pojačavanje filtriranjem

- Ostaje pitanje kombiniranja ocjena sva tri učenika
 - Glasanje
 - Srednja ocjena
- Potreba za velikim skupom uzoraka za treniranje

AdaBoost

- Pojačavanje uzorkovanjem
- "recikliranje" skupa za treniranje
- Algoritam se ADaptira na greške pojedinih slabih učenika
- Očekivana greška svakog slabog učenika je nešto manja od 50%

AdaBoost

- 1.Svakom uzorku se pridijeli jednaka težina
- 2.Trenaira se slabi učenik na određenoj distribuciji slučajnim odabirom uzoraka za treniranje
- 3.Odredi kvalitetu slabog učenika prema njegovoj konačnoj grešci**
 - Koristi se za kombiniranje slabih učenika
- 4.Povećaj težine uzoraka koje je slabi učenik pogrešno ocijenio i odredi novu distribuciju**
- 5.Odaberi novog slabog učenika i vrati se na 2. korak

AdaBoost pseudokod

1. Inicijaliziraj distribuciju $D_1(i) = 1/N$
2. Treniraj slabog učenika na distribuciji $D(n)$
3. Odredi izlaz slabog učenika $F_n: \mathbf{X} \rightarrow Y$

4. Izračunaj grešku
$$e_n = \sum_{i: F_n(\mathbf{x}_i) \neq d_i} D_n(i)$$

5.
$$\beta_n = \frac{e_n}{1 - e_n}$$

6. Osvježi distribuciju

$$D_{n+1}(i) = \frac{D_n(i)}{Z_n} * \begin{cases} \beta_n & \text{ako je } F_n(\mathbf{x}_i) = d_i \\ 1 & \text{inače} \end{cases}$$

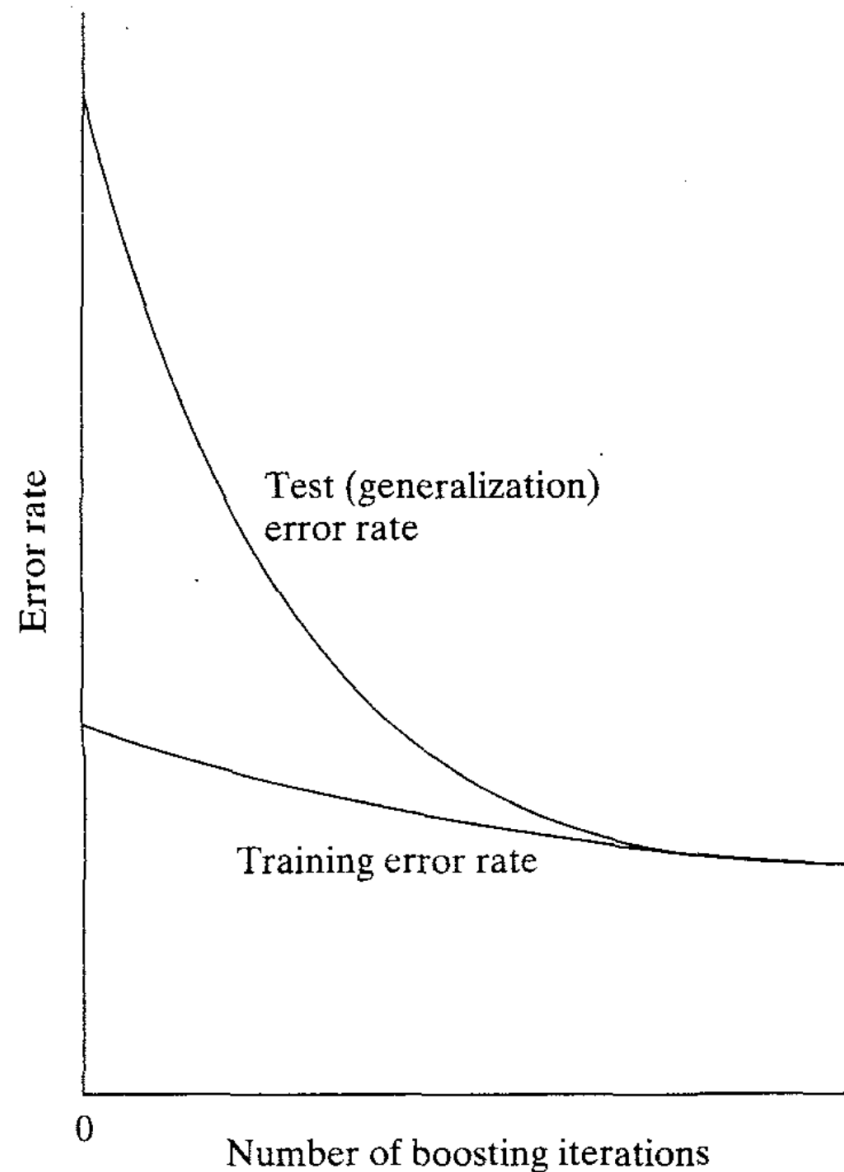
Z_n – normalizacijska konstanta

Konačna ocjena
$$F_{fin}(\mathbf{x}) = \arg \max_{d \in D} \sum_{i: F_n(\mathbf{x}) = d} \log \frac{1}{\beta_n}$$

AdaBoost

- Dodavanjem novih slabih učenika koji imaju grešku nešto manju od 50% ukupna pogreška treniranja eksponencijalno teži 0
- Pokazalo se da se i greška testiranja smanjuje (bolja generalizacija), čak i kada novi slabi učenici neznatno smanjuju grešku treniranja
 - Povećanje kompleksnosti poboljšava generalizaciju!
 - Tumačenje je da novi slabi učenici povećavaju marginu između klasa (poveznica sa SVM) te tako smanjuju grešku i kod skupa za testiranje
 - Nema pretreniranja!

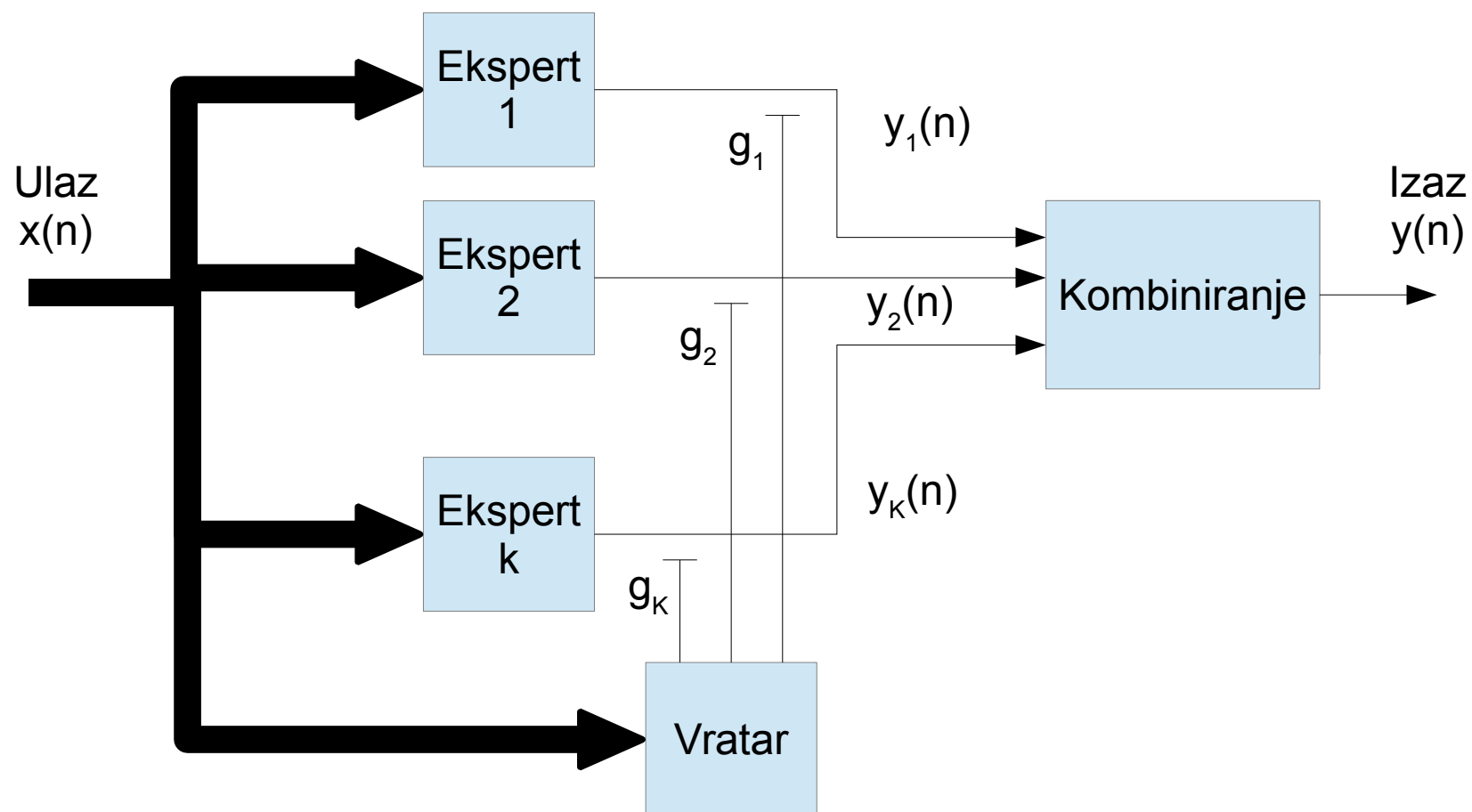
AdaBoost poboljšanje generalizacije



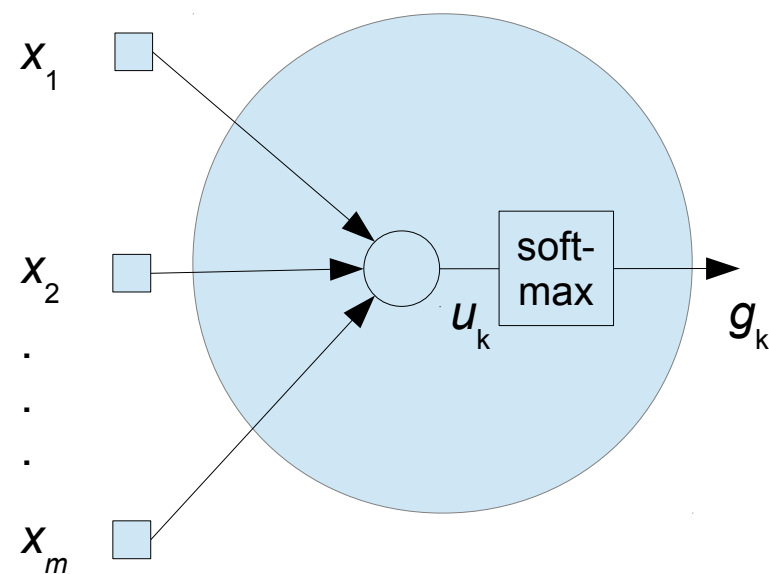
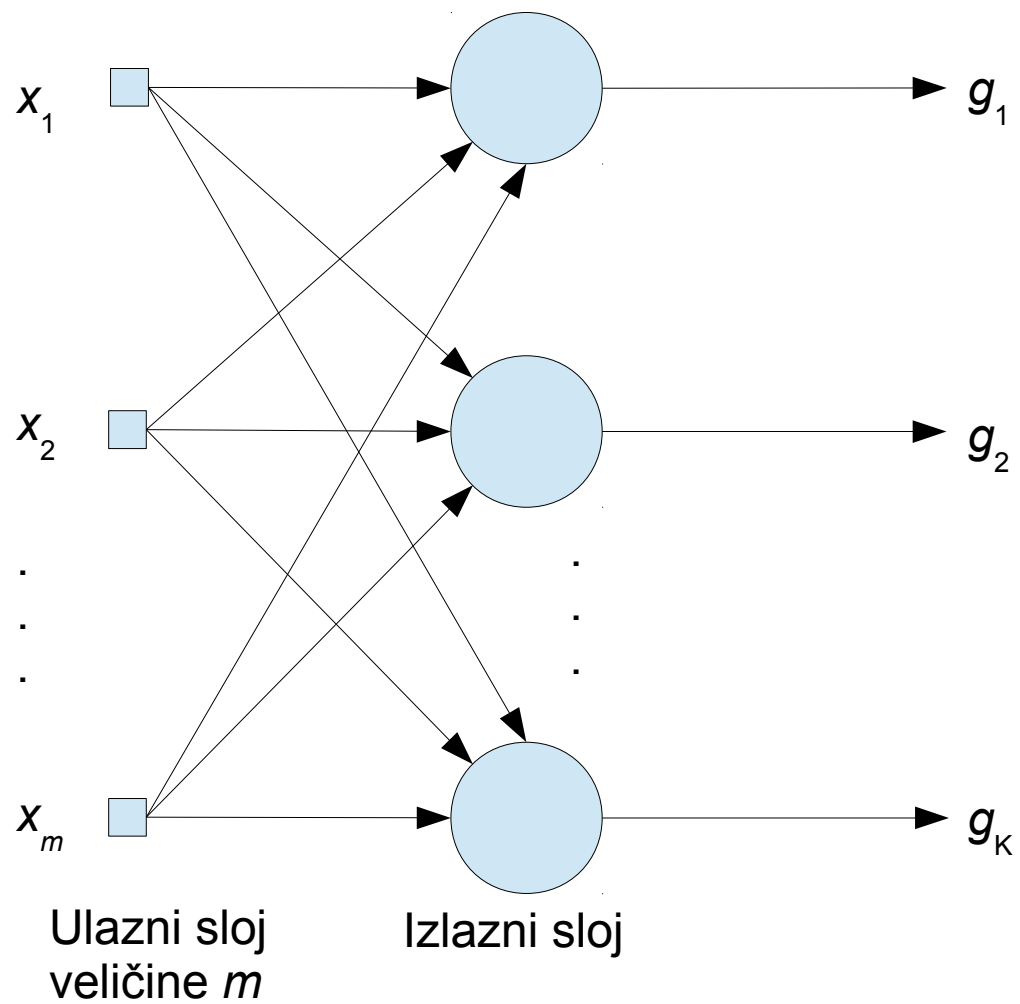
Dinamička struktura

- Dinamička struktura – kombiniranje eksperata ovisi o ulaznim podacima
- Eksperti se sami organiziraju i prilagođavaju ulazim podacima
- Pojedini eksperti specijalizirani su za podskupve ulaznih podataka, ali kao cjelina dobro funkcioniraju na cijelom skupu ulaznih podataka
- Dinamičko grupiranje obavlja vratar (gating network)

Mješavina eksperata



Vratar



Mješavina eksperata

- Neuroni vratara su nelinearni
 - Nelinearna funkcija od \mathbf{x}
- Aktivacijska funkcija:

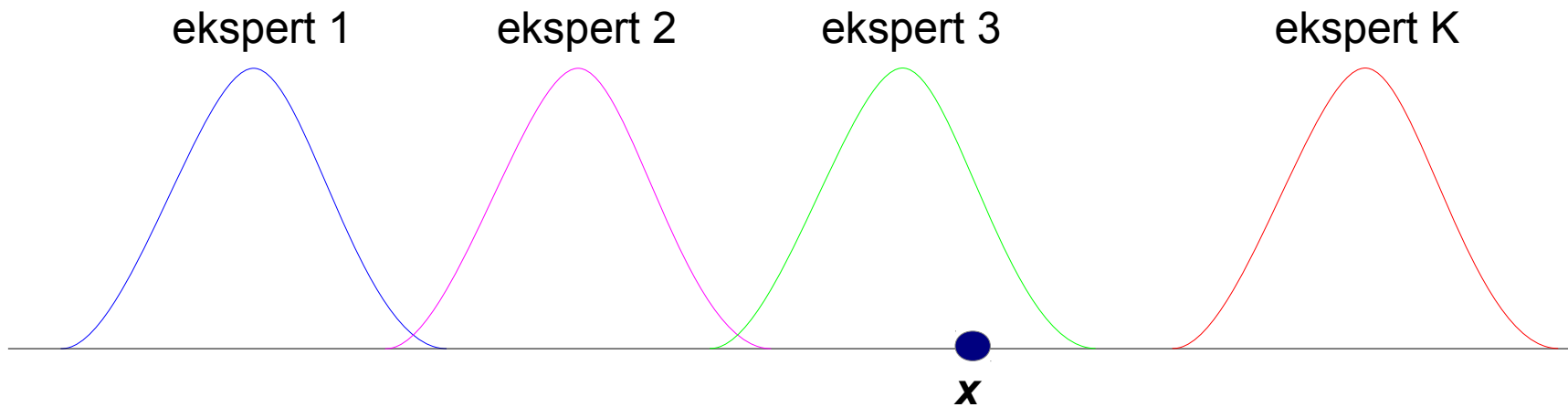
$$g_k = \frac{\exp(u_k)}{\sum_{j=1}^K \exp(u_j)}$$

$$u_k = \mathbf{a}_k^T \mathbf{x}$$

- Derivablina varijanta winner-takes-all – softmax

Vratar

- Možemo ga promatrati kao klasifikatora koji mapira vektor \mathbf{x} u multimodalnu distribuciju tako da odabire odgovarajućeg eksperta odgovornog za jedan "mod"



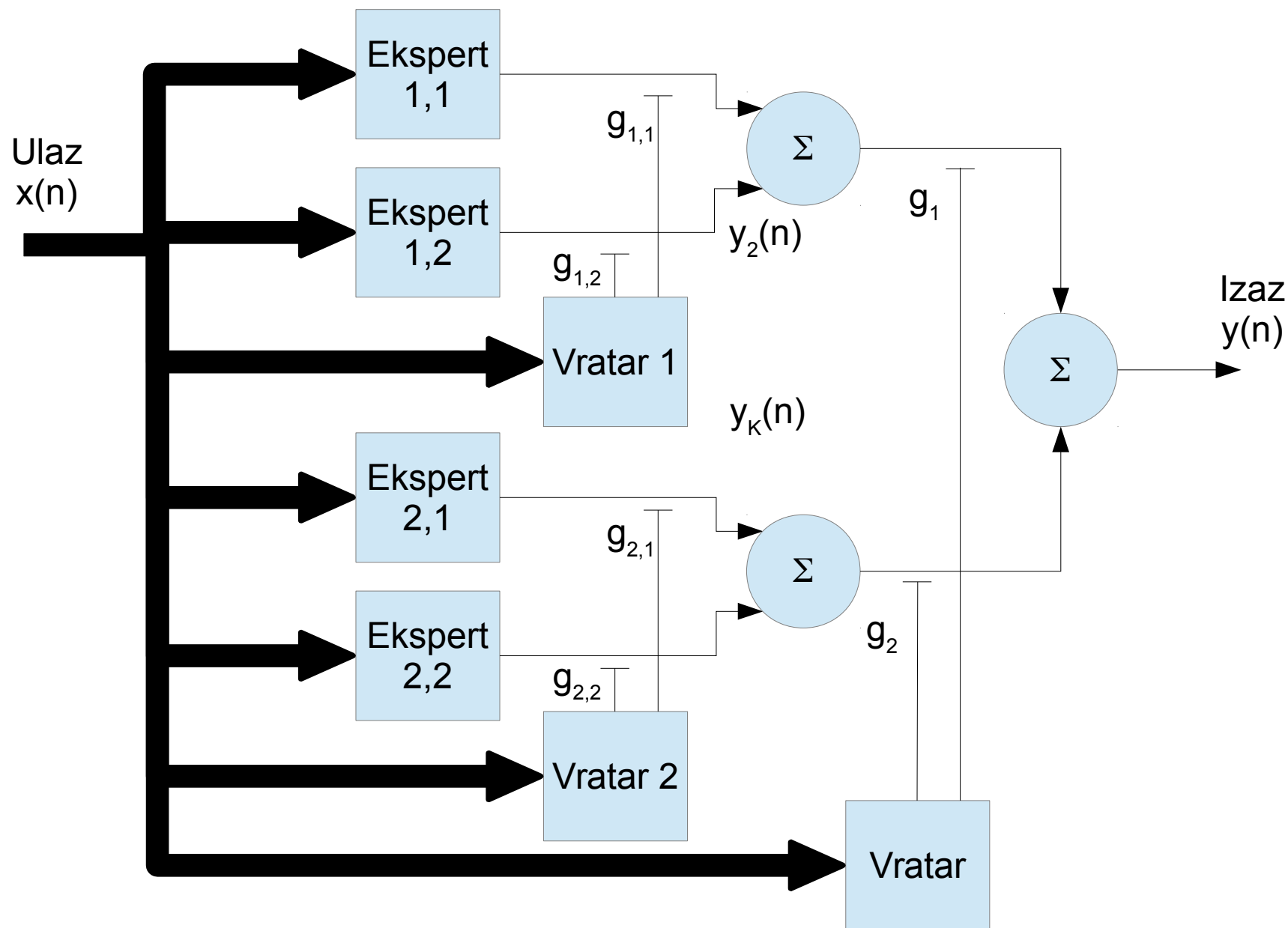
Konačan odziv

$$y = \sum_{k=1}^K g_k y_k$$

$$\sum_{k=1}^K g_k = 1 \qquad 0 \leq g_k \leq 1 \text{ za sve } k$$

- Preostaje još naučiti y_k i g_k

Hijerarhijska mješavina eksperata



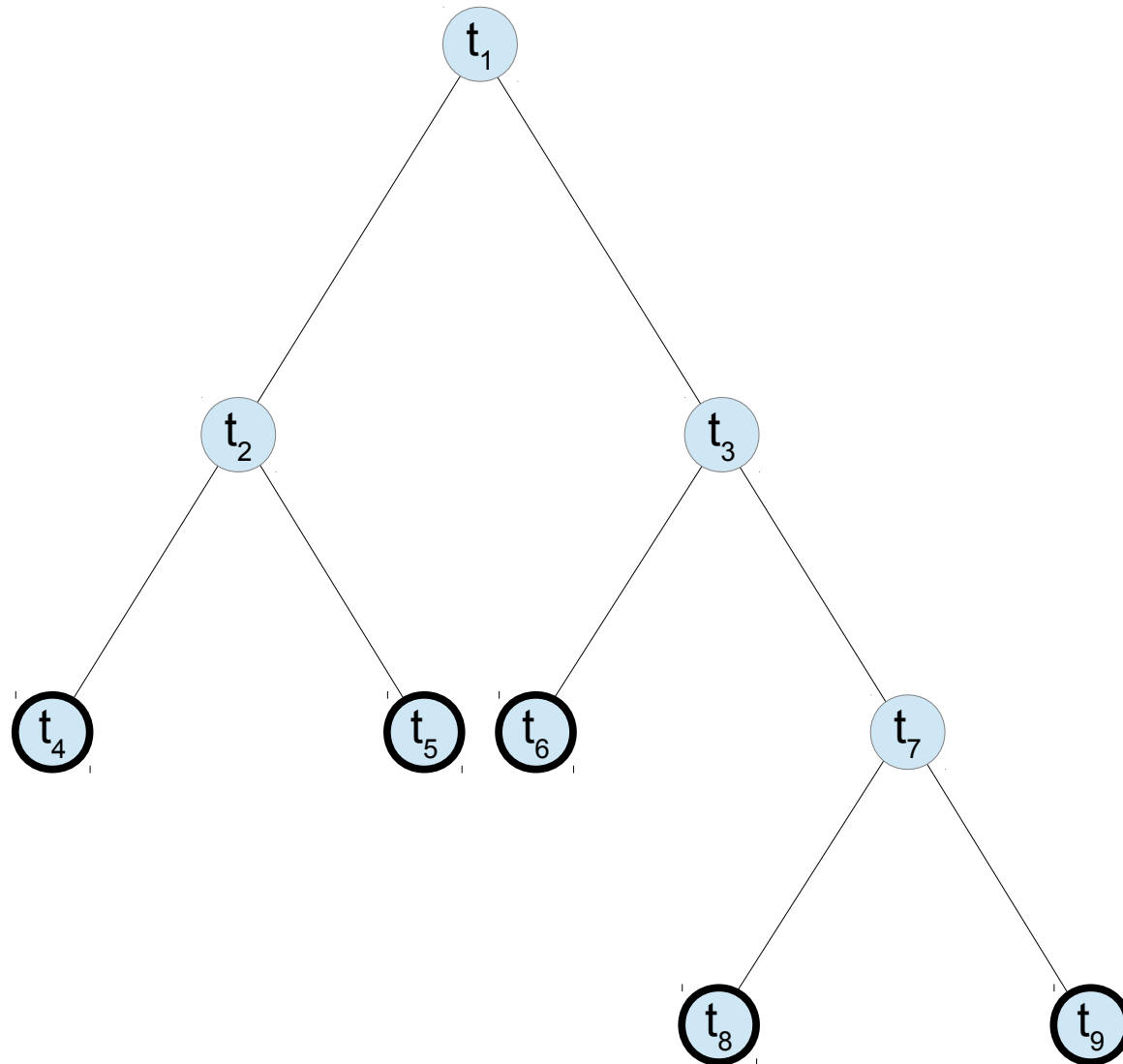
Hijerarhijska mješavina eksperata

- Ekstenzija mješavine eksperata
- Stablasta struktura
- Podjela ulaznog prostora na ugniježdene potprostore
- Strategija podjeli pa vladaj – dobro je dijeliti ulazni prostor na potprostore
- Slično stablu odluka koje u grananjima donosi fiksne odluke (da/ne)
 - Fiksne odluke dovode do gubitka informacija
 - Problem pohlepe – jednom donesena odluka se u kasnijim grananjima ne može izmijeniti

Hijerarhijska mješavina eksperata

- Prije procjene parametara mreže potrebno je odrediti model: broj i povezanost grananja
- Jedna mogućnost je izvođenje standardnog stabla odluke kao recimo CART (classification and regression tree) radi inicijalizacije
 - CART dijeli ulazni prostor uzastopnim binarnim grananjima
 - Grananja u CART-u postaju vratari u HME
 - Koristi se jednostavnost CART-a u odnosu na HME za brzo i pametno određivanje arhitekture HME
 - CART se poboljšava sa *soft* odlučivanjem

CART



CART algoritam

1. Odabir mjesta grananja:

- Nađi srednju ocjenu čvora t
$$\bar{d}(t) = \frac{1}{N(t)} \sum_{x_i \in t} d_i$$

- Odredi srednju kvadratnu pogrešku za čvor t

$$E(t) = \frac{1}{N} \sum_{x_i \in t} (d_i - \bar{d}(t))^2$$

- Odredi ukupnu kvadratnu pogrešku za krajnje čvorove stabla

$$E(T) = \sum_{t \in T} E(t)$$

- Podjela čvora t je ona koja minimizira $E(T)$

CART algoritam

- Od svih mogućih podjela S na čvoru t odaberi onu podjelu s^* koja dijeli čvor t na t_L i t_D tako da

$$\Delta E(s, t) = E(t) - E(t_L) - E(t_D)$$

$$\Delta E(s^*, t) = \max_{s \in S} \Delta E(s, t)$$

- Na taj način minimizira se $E(T)$

CART algoritam

2. Određivanje krajnjih čvorova prema unaprijed zadanom pragu β

$$\max_{s \in S} \Delta E(s, t) < \beta$$

3. Određivanje parametara krajnjih čvorova

$$\mathbf{w}(t) = \mathbf{X}^+ \mathbf{d}(t)$$

gdje je \mathbf{X}^+ pseudoinverz matrice \mathbf{X} koja se sastoji od svih ulaza $x_i \in t$, $\mathbf{d}(t)$ se sastoji od svih d_i u t

- Konačan rezultat je minimiziranje sume kvadrata pogrešaka

Inicijalizacija HME

- Svaka podjela u stablu definira višedimezionalnu površinu

$$\mathbf{a}^T \mathbf{x} + b = 0$$

1. Primjeni CART na podacima za treniranje
 2. Postavi parametre težina pojedinih eksperata na odgovarajuće težine \mathbf{w} krajnjih čvorova CART-a
 3. Za vratane (softmax):
 - a) postavi težine vratarata tako da pokazuju okomito na dotičnu ravninu razdvajanja
 - b) slučajno odaberi vektora težina male dužine
- Za estimaciju svih parametara koristi se EM algoritam

Diskusija

- Mješavina eksperata
 - Smanjenje greške pomoću pretreniranja
 - Rješavanje problema prevelike raznolikosti podataka kroz različite početne uvjete treniranja
- Boosting
 - Smanjivanje greške do proizvoljnog iznosa uzastopnim dodavanjem slabih učenika
- Hijerarhijska mješavina eksperata
 - Kompromis između jednostavnosti CART (jednostavnost -> lagani uvid u suštinu problema) i složenosti MLP (iako moćan, pristup crne kutije ne daje nikakav uvid u suštinu problema)

Zadaci

1. Razmislite kako bi se pojačanje filtriranjem ili AdaBoost ponašali da koriste jake učenike umjesto slabih
2. Usporedite računalnu složenost MLP-a s dva skrivena sloja (10-10-1) i mješavine eksperata sa jednim vratarom (10-2) i ekspertima (linearne mreže 10-1)