

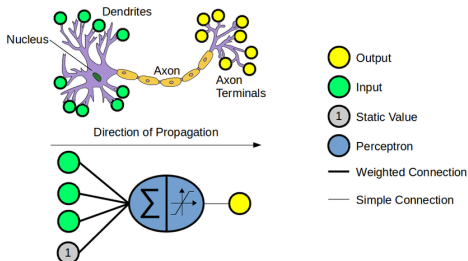
# Вештачке неуронске мреже

Ненад Митић

Математички факултет  
[nenad@matf.bg.ac.rs](mailto:nenad@matf.bg.ac.rs)

# Вештачке неуронске мреже (ВНМ)

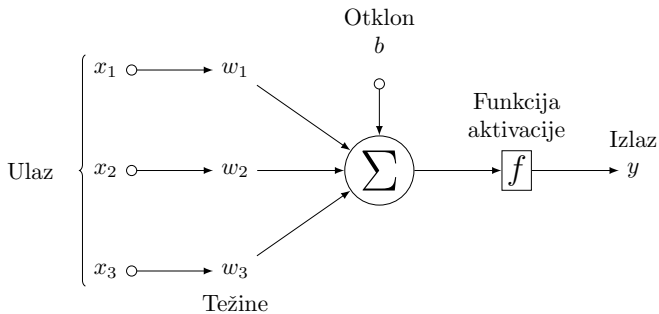
- Идеја: симулација рада биолошких нервних система
- Аналогно структури у мозгу, ВНМ чине чворови и везе између њих
- У ВНМ чворови представљају неуроне или јединце



# Перцептрон

- Најједноставнија верзија ВНМ
- Моделира појединачну ћелију
- Два типа (улазни, излазни) чворова који су повезани везом са *тежинама*
- Тежине симулирају јачину синаптичке везе у биолошким неуронима
- Тренирање (обучавање) перцептрона укључује промену вредности тежине
- Тренирање траје док се синхронизују улазно/излазне зависности података

# Перцептрон



# Перцептрон

- Рачуна излазну вредност  $\bar{y}$  као тежинску суму улазних вредности уз одузимање отклона (енг. *bias*) уз проверу знака резултата
- Претходни перцептрон за вредности тежина 0.3 и отклона 0.4 представља модел за израчунавање

$$\bar{y} = \begin{cases} 1, & \text{ако } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 > 0 \\ -1, & \text{ако } 0.3x_1 + 0.3x_2 + 0.3x_3 - 0.4 < 0 \end{cases}$$

Математички, излазни модел перцептрона је једнак

$$\begin{aligned} \bar{y} &= \text{sign}(w_d x_d + w_{d-1} x_{d-1} + \dots + w_2 x_2 + w_1 x_1 - b) \\ &= \text{sign}(w \cdot x) \end{aligned}$$

# Обучавање перцептрона

## Алгоритам за обучавање перцептронаа

```

Neka je  $D = \{(x_i, y_i) | i = 1, \dots, n\}$  skup primera za trening
Inicijalizovati tezine slucajnim vrednostima  $w^{(0)}$ 
repeat
  for svaki trening primer  $(x_i, y_i) \in D$  do
    Izracunaj predvidjeni izlaz  $\bar{y}_i^{(k)}$ 
    for svaka teжина  $w_j$  do
      azuriraj tezinu  $w_j^{(k+1)} = w_j^{(k)} + \lambda(y_i - \bar{y}_i^{(k)})x_{ij}$ 
    end for
  end for
until dostignut izlazni kriterijum

```

где су  $w^{(k)}$  тежина придружена  $i$ -тој улазној грани после  $k$ -те итерације,  $\lambda$  је брзина учења, а  $x_{ij}$  је вредност  $j$ -тог атрибута у примеру за обуку  $x_i$ ;

# Обучавање перцептрона

Нова вредност  $w^{(k+1)}$  је комбинација старе вредности  $w^{(k)}$  и вредности пропорционалне грешци предвиђања  $(y - \bar{y})$

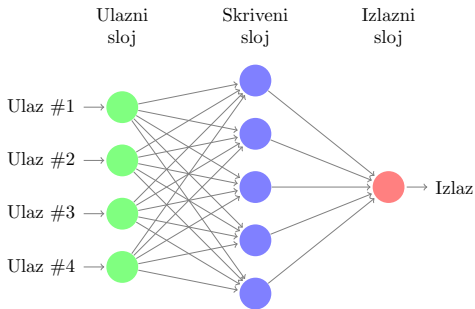
- Ако је предвиђање коректно, тежине се не мењају
- Ако је  $y = +1, \bar{y} = -1$ , тада је грешка предвиђања  $(y - \bar{y}) = 2$ , и да би се отклонила повећава се вредност предвиђеног излаза повећањем тежина у свим везама са позитивним, и смањењем у везама са негативним улазом
- Ако је  $y = -1, \bar{y} = +1$ , тада је грешка предвиђања  $(y - \bar{y}) = -2$ , и да би се отклонила смањује се вредност предвиђеног излаза смањењем тежина у свим везама са позитивним, а смањује у везама са негативним улазом

# Обучавање перцептрона

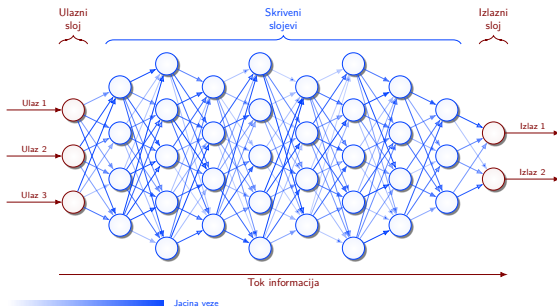
- Параметар  $\lambda \in [0, 1]$  се користи за контролу подешавања у свакој итерацији
  - Ако је  $\lambda$  близу нуле, нове вредности тежина највише зависе од старих вредности тежина
  - Ако је  $\lambda$  близу 1, нове вредности тежина су осетљиве на количину подешавања у текућој итерацији
- Претходни модел перцептрона је линеаран по параметрима  $w$  и атрибутима  $x$
- Вредност  $\bar{y} = 0$  представља линеарну хиперраван која раздваја податке у две класе ( $y = 1, y = -1$ )
- Алгоритам обучавања перцептрона гарантује конвергенцију ка оптималном решењу за линеарно раздвојиве класификационе проблеме



# ВНМ са више слојева



# ВНМ са више слојева



# Врсте ВНМ

- Са пропагацијом унапред
- Са пропагацијом уназад
- Рекурентне - везе унутар истог или претходног слоја
- ...
- 
- 
- 

Могу да се користе различите активационе функције (линеарна, сигмоид, тангенс хиперболички, сигн, ...)

# Обучавање ВНМ

Циљ обучавања ВНМ је одредити скуп тежина  $w$  који минимизује укупну квадратну грешку

$$E(w) = \frac{1}{2} \sum_{i=1}^n (y_i - \bar{y}_i)^2$$

Заменом  $\bar{y} = w \cdot x$  добија се да је функција грешке квадратна функција - може се одредити глобални минимум

# Обучавање ВНМ

- Ако активациона ф-ја није линеарна излаз из ВНМ је нелинеарна функција параметара
- начина да се добије глобални минимум
- Користи се метода градијентног спуста за решење оптимизационог проблема

$$w_j \leftarrow w_j - \lambda \frac{\partial E(w)}{\partial w_j}$$

- Како је функција грешке нелинеарна, могуће је да се помоћу градијентног спуста не добије глобални већ локални минимум