Neuronske mreže

Poglavlje 18.7

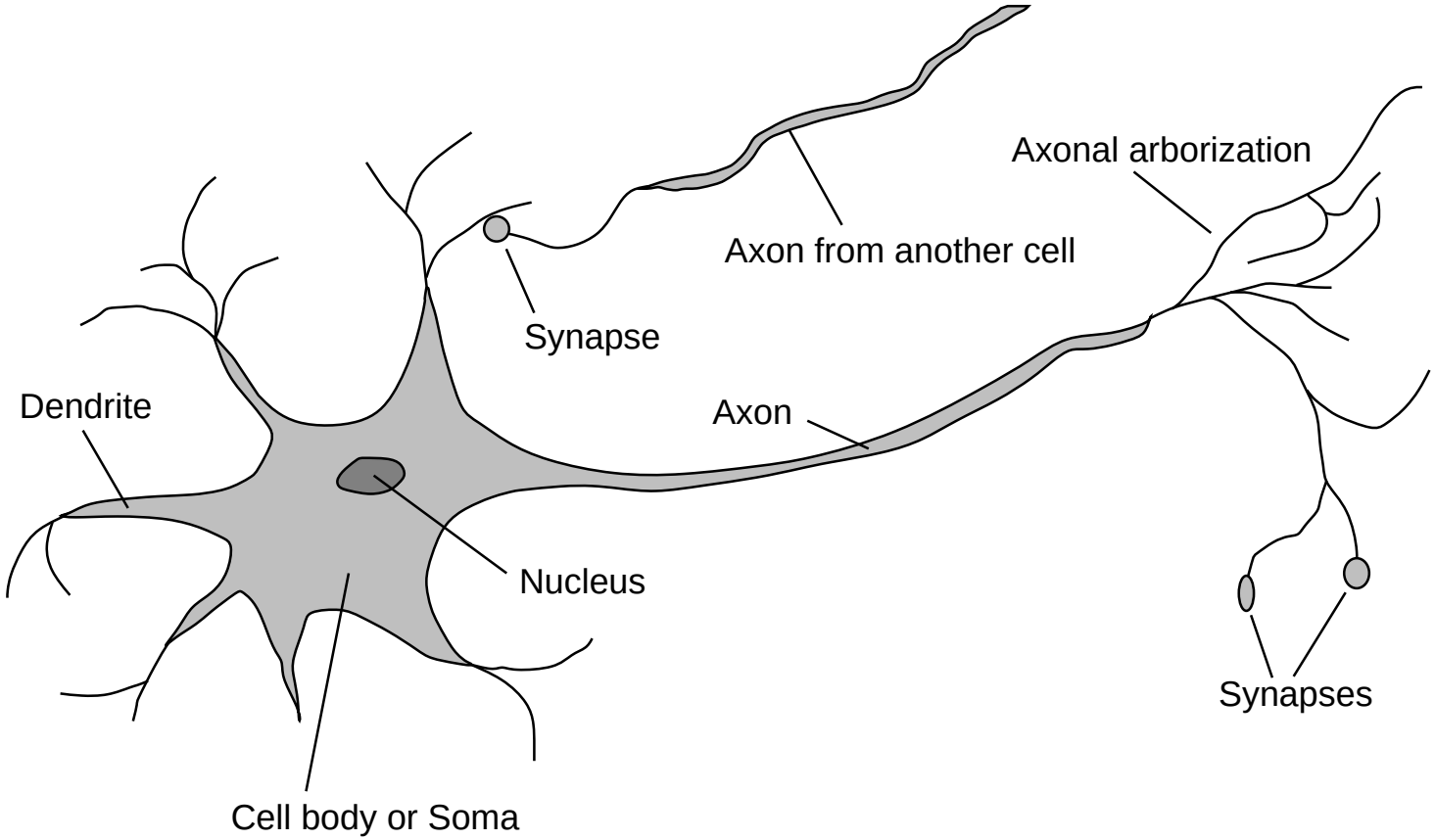
Prema slajdovima Stuarta Russella (hvala)!

Sadržaj

- ♢ Mozgovi
- ♢ Neuronske mreže
- ♢ Perceptroni
- ♢ Višeslojni perceptroni
- ♢ Primjena neuronskih mreža

Mozgovi

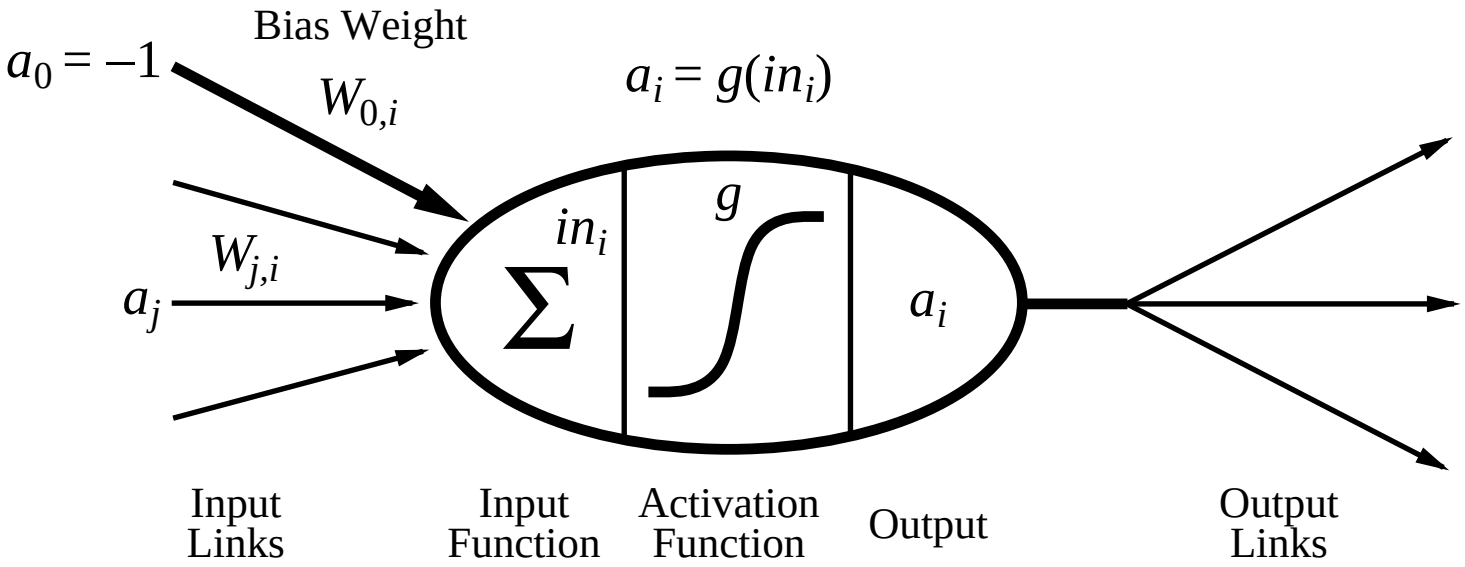
10^{11} neurona s > 20 tipova, 10^{14} sinapsi, 1 ms–10 ms vrijeme ciklusa
Signali su niz "šiljaka" električnog potencijala = "okidanje" (sa šumom)



McCulloch–Pitts "jedinice"

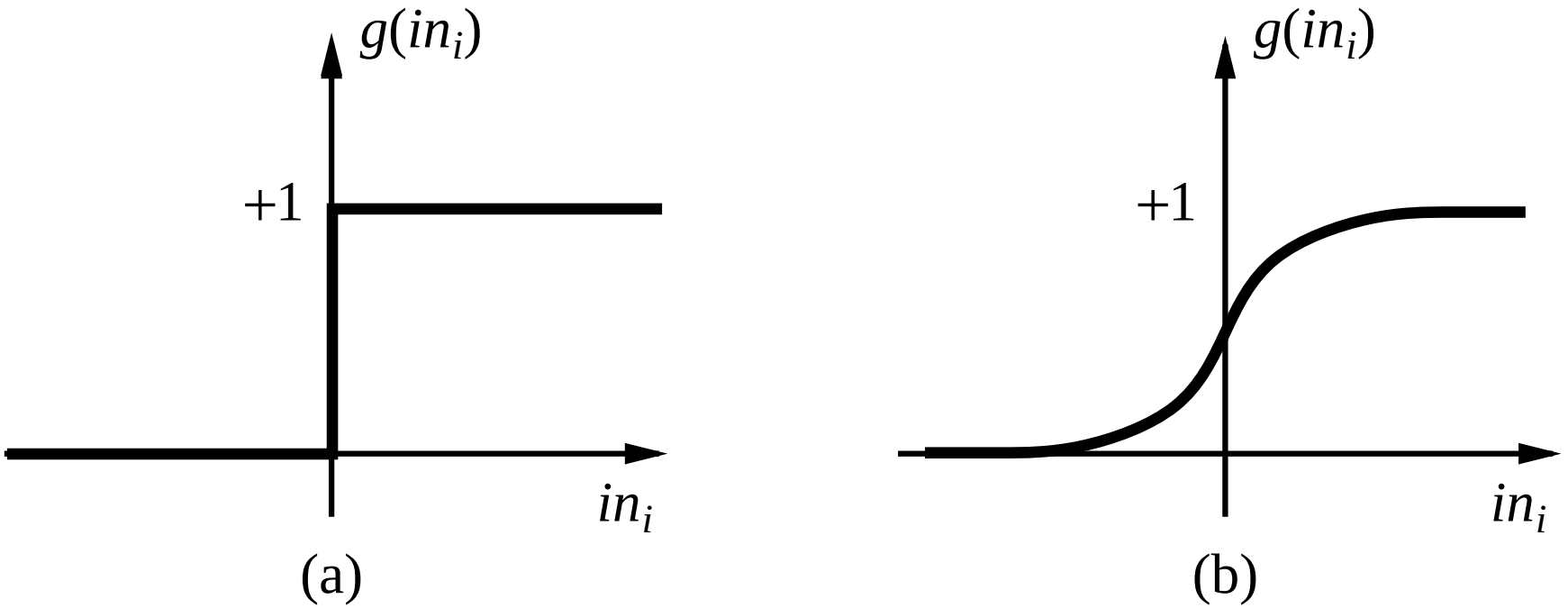
Izlaz je "zgnječena" linearna funkcija ulaza:

$$a_i \leftarrow g(in_i) = g\left(\sum_j W_{j,i} a_j\right)$$



Veliko pojednostavljivanje stvarnih neurona, ali njegova je svrha razviti razumijevanje što mreža jednostavnih jedinica može napraviti

Funkcije aktivacije

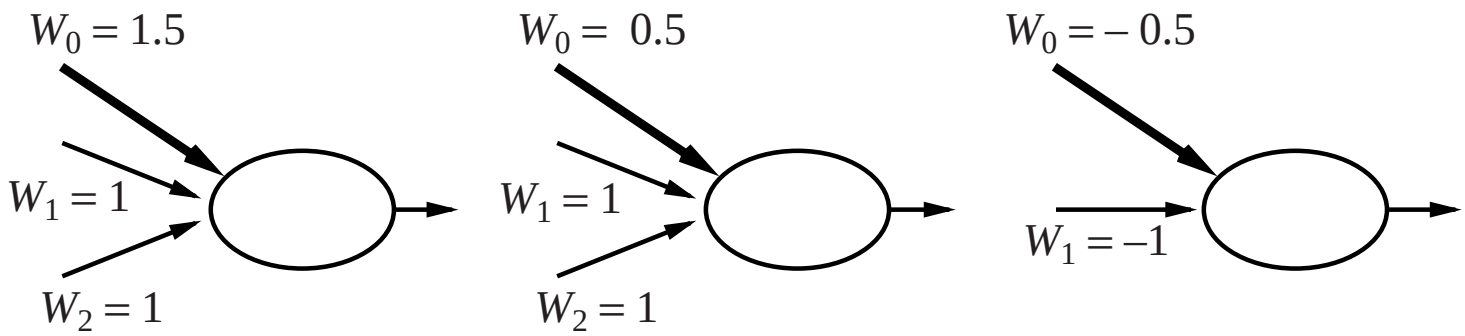


(a) je step funkcija ili granična funcija

(b) je sigmoidalna funkcija $1/(1+e^{-x})$

Promjena dodane težine $W_{0,i}$ pomiče granicu

Implementacija logičkih funkcija



McCulloch i Pitts: svaka Booleova funkcija može se implementirati

Struktura mreže

Mreže s vezama unaprijed (engl. feed–forward = "hranjenje unaprijed"):

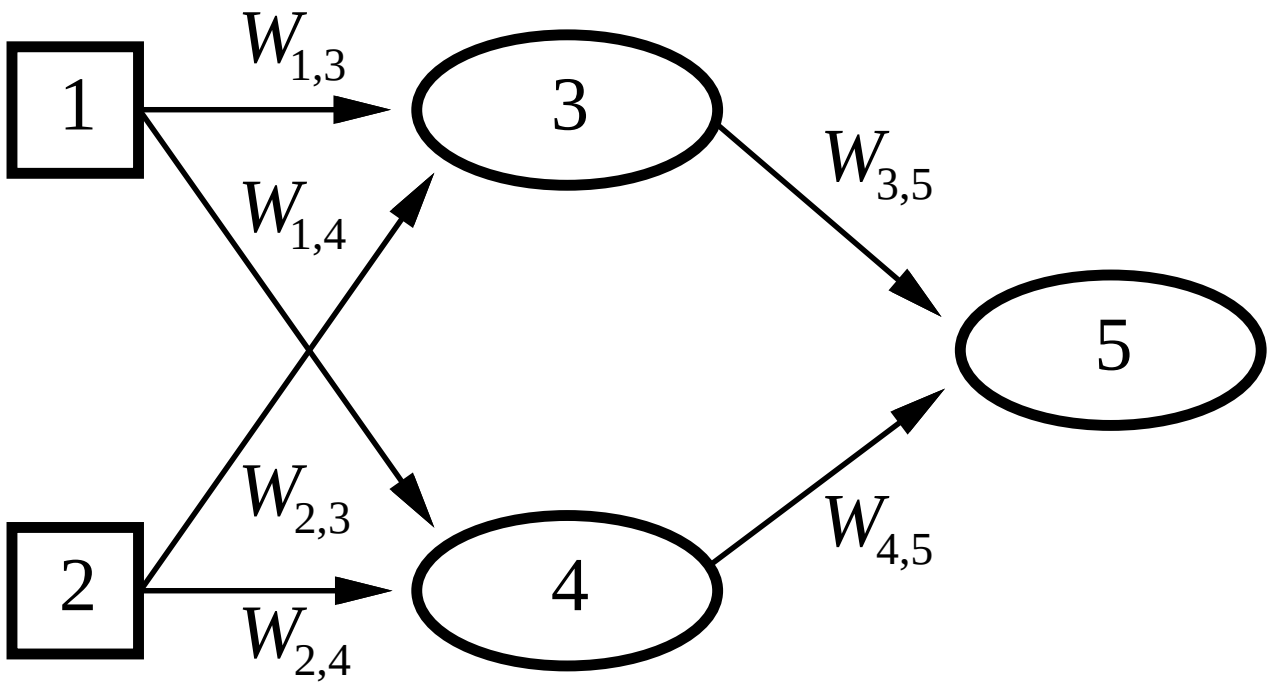
- jednoslojni perceptroni
- višeslojni perceptroni

Mreže s vezama unaprijed implementiraju funkcije, nemaju internih stanja

Rekurzivne mreže:

- Hopfieldove mreže imaju simetrične težine ($W_{i,j} = W_{j,i}$)
  $g(x) = \text{sign}(x)$, $a_i = \pm 1$;
  holografska asocijativna memorija
- Boltzmannovi strojevi koriste stohastičke aktivacijske funkcije,
  $\approx$ MCMC u Bayesovim mrežama
- rekurzivne neuronske mreže imaju vođene cikluse s kašnjenjem
  $\Rightarrow$ imaju interna stanja (kao flip-flap), mogu oscilirati itd.

Primjer mreža s vezama unaprijed

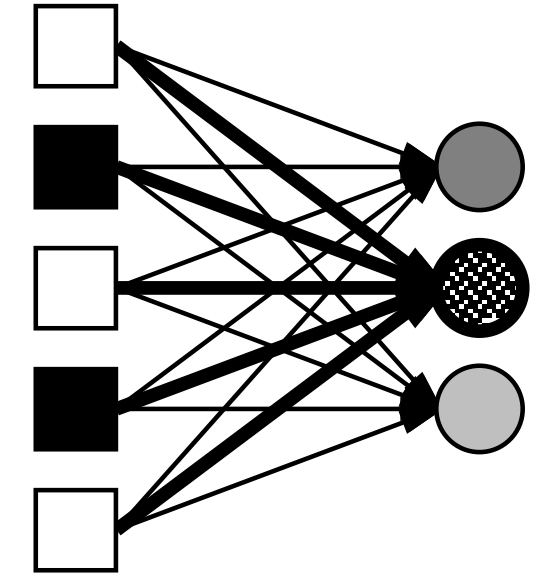


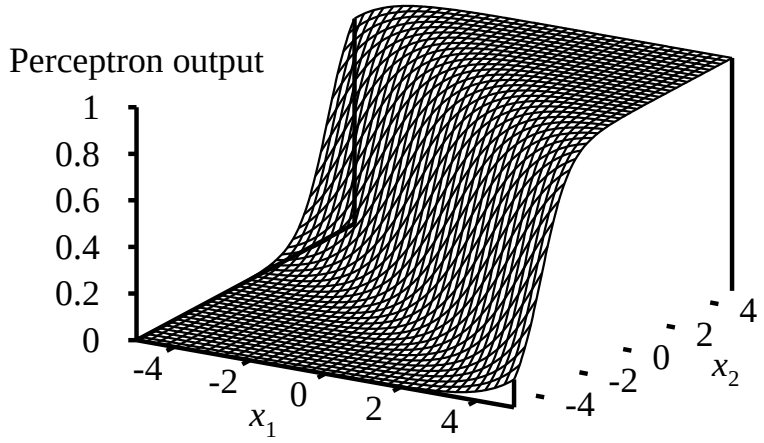
Mreža s vezama unaprijed = parametrizirana familija nelinearnih funkcija:

$$a_5 = g(W_{3,5} \cdot a_3 + W_{4,5} \cdot a_4)$$
$$= g(W_{3,5} \cdot g(W_{1,3} \cdot a_1 + W_{2,3} \cdot a_2) + W_{4,5} \cdot g(W_{1,4} \cdot a_1 + W_{2,4} \cdot a_2))$$

Promjena težina mijenja funkciju: učenje se može napraviti i tako!

Jednoslojni perceptroni





Svaka od izlaznih jedinica radi posebno—nema dijeljenih težina

Promjena težina mijenja lokaciju, orijentaciju, i strminu funkcije

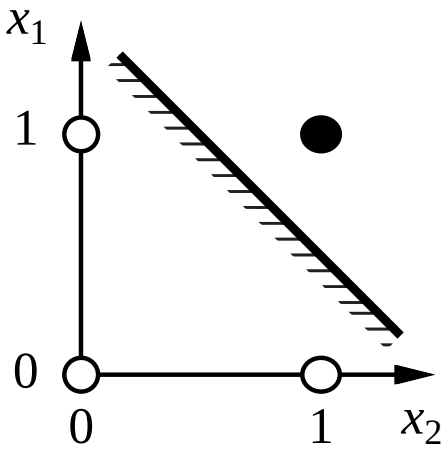
Izražajnost perceptrona

Promatrajno perceptron sa $g =$ step funkcija (Rosenblatt, 1957, 1960)

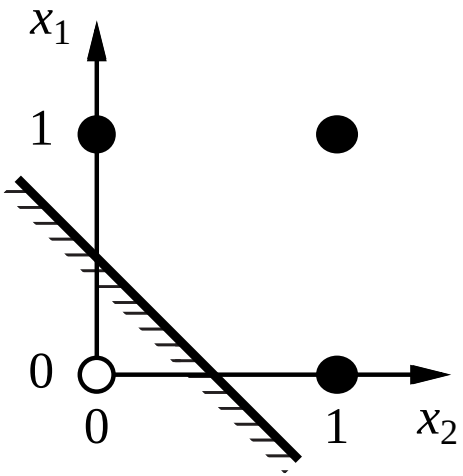
Može reprezentirali AND, OR, NOT, većinu, itd, ali ne može XOR

Reprezentira linearni separator u ulaznom prostoru:

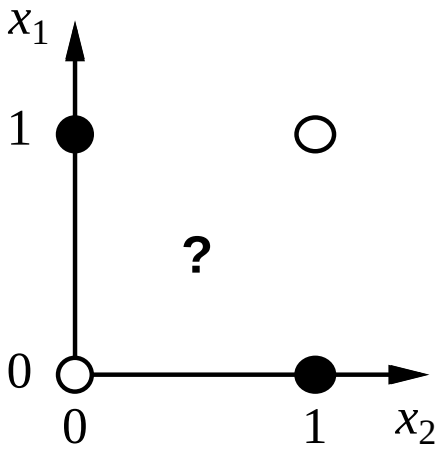
$$\sum_j W_j x_j > 0 \quad \text{or} \quad \mathbf{W} \cdot \mathbf{x} > 0$$



(a) x_1 **and** x_2



(b) x_1 **or** x_2



(c) x_1 **xor** x_2

Učenje perceptrona

Učenje promjenom težina da se smanji greška na skupu za treniranje.

Kvadrat greške za primjer s ulazom $\mathbf{x}$ i pravim izlazom y je

$$E = \frac{1}{2}Err^2 \equiv \frac{1}{2}(y - h_{\mathbf{W}}(\mathbf{x}))^2.$$

Rješenje = optimizacijsko traženje gradijentnim silaskom:

$$\begin{aligned}\frac{\partial E}{\partial W_j} &= Err \times \frac{\partial Err}{\partial W_j} = Err \times \frac{\partial}{\partial W_j}\left(y - g(\textstyle\sum_{j=0}^{n} W_j x_j)\right) \\ &= -Err \times g'(in) \times x_j\end{aligned}$$

Jednostavno pravilo ažuriranja težina: α = korak (parametar učenja)

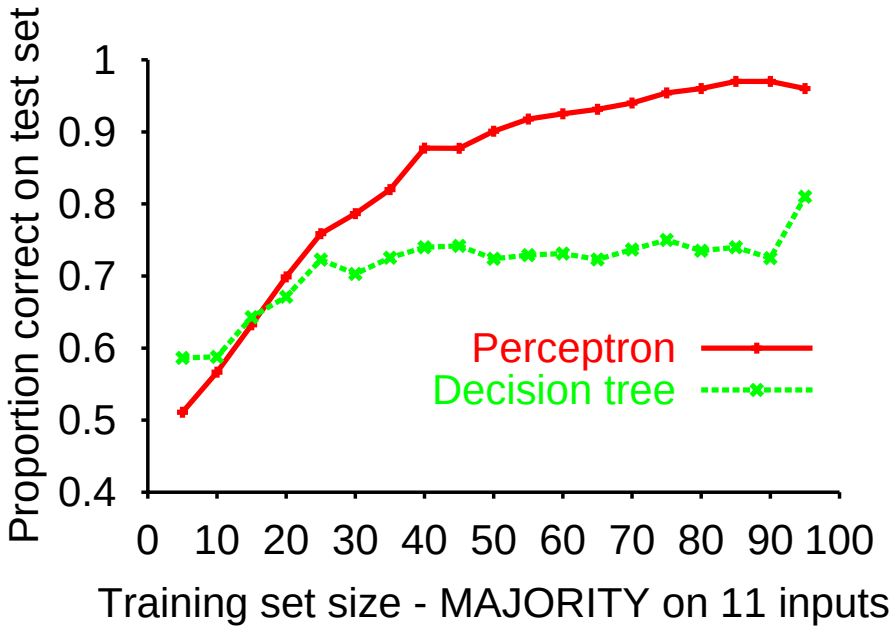
$$W_j \leftarrow W_j + \alpha \times Err \times g'(in) \times x_j$$

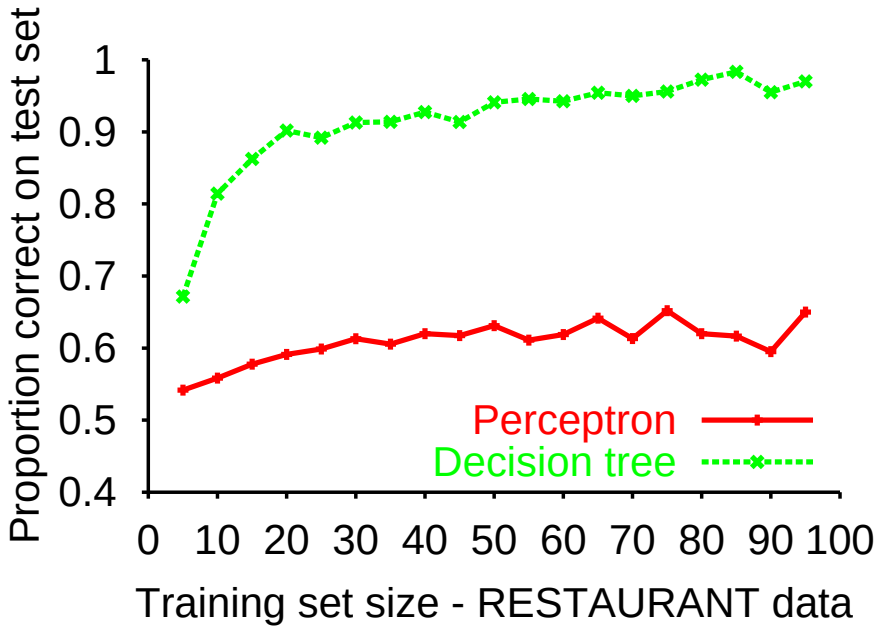
Na primjer, +greška $\Rightarrow$ treba povećati mrežni izlaz

$\Rightarrow$ povećati težine na +ulazima, smanjiti na -ulazima.

Učenje perceptrona nastavak

Pravilo za učenje perceptrona konvergira konzistentnoj funkciji
za bilo koji linearno separabilni skup podataka



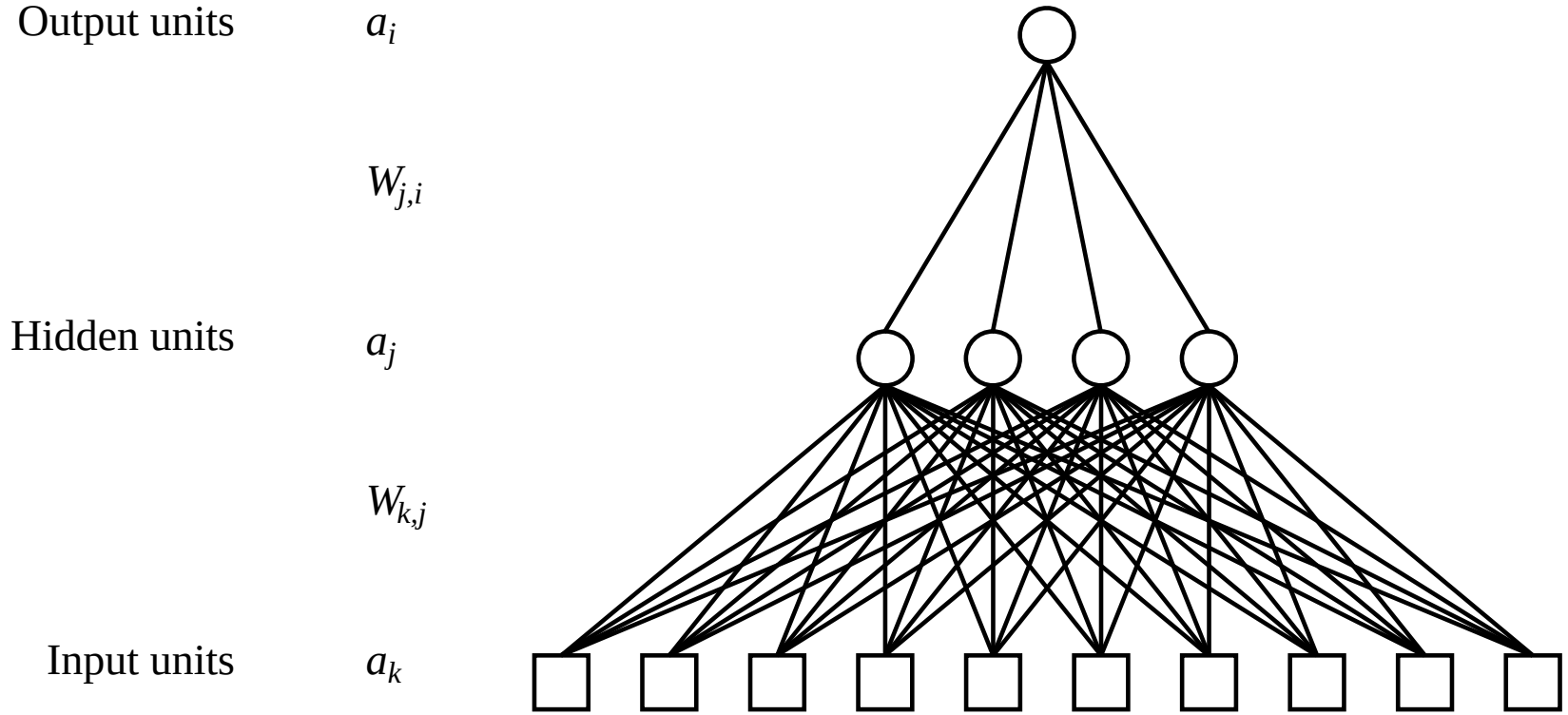


Perceptroni lako uče funkciju za većinu, DTL je beznadan

DTL lako uče funkciju restorana, perceptroni je ne mogu reprezentirati

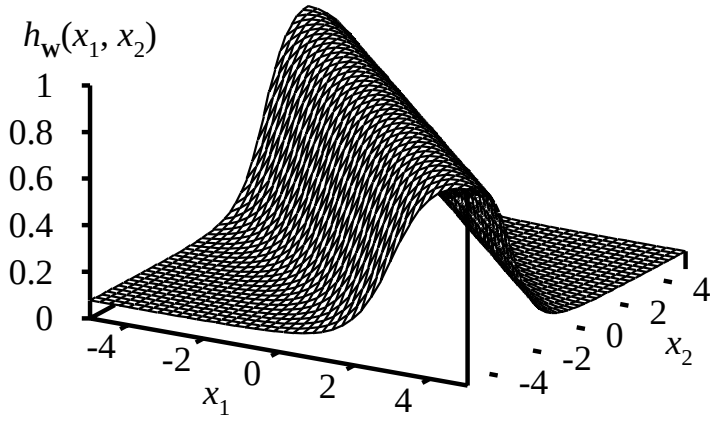
Višeslojni perceptroni — MLP

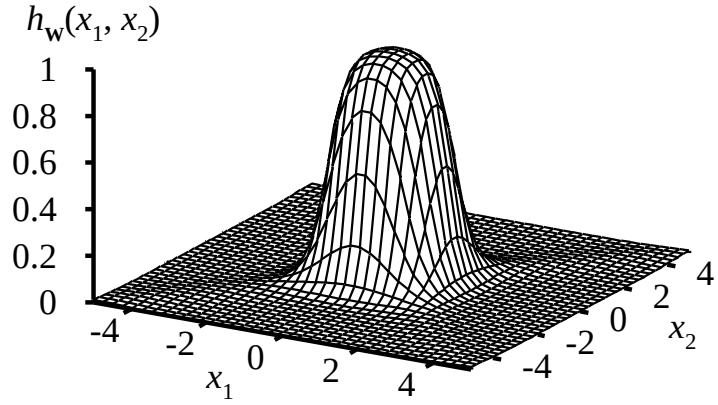
Slojevi su uobičajeno potpuno povezani;
broj skrivenih jedinica tipično se bira "ručno"



Izražajnost MLP-ova

Sve neprekidne funkcije idu s 2 sloja (jedan skriveni), ili sve funkcije s 3 sloja





Kombinirati dvije funkcije granice (koje gledaju u razne strane) da se dobije greben

Kombinirati dva okomita grebena da se dobije ispupčenje (brdo)

Dodati ispupčenja različitih veličina i lokacija tako da aproksimiraju bilo koju površinu

(Dokaz treba eksponencijalno mnogo skrivenih jedinica)

Učenje propagiranjem unatrag

Izlazni sloj: isti kao kod jednoslojnog perceptrona,

$$W_{j,i} \leftarrow W_{j,i} + \alpha \times a_j \times \Delta_i$$

gdje $\Delta_i = Err_i \times g'(in_i)$

Skriveni sloj: **propagira unatrag** grešku iz izlaznog sloja:

$$\Delta_j = g'(in_j) \sum_i W_{j,i} \Delta_i \ .$$

Pravilo ažuriranja težina u skrivenom sloju:

$$W_{k,j} \leftarrow W_{k,j} + \alpha \times a_k \times \Delta_j \ .$$

(Većina neuroznanstvenika opovrgava tvrdnju da je propagiranje unatrag događa u mozgu)

Izvod propagiranja unatrag

Kvadrirana greška na jednom primjeru definira se kao

$$E = \frac{1}{2}\sum_i (y_i - a_i)^2 \ ,$$

gdje je suma po svim čvorovima izlaznog sloja.

$$\begin{aligned}
\frac{\partial E}{\partial W_{j,i}} &= -(y_i - a_i)\frac{\partial a_i}{\partial W_{j,i}} = -(y_i - a_i)\frac{\partial g(in_i)}{\partial W_{j,i}} \\
&= -(y_i - a_i)g'(in_i)\frac{\partial in_i}{\partial W_{j,i}} = -(y_i - a_i)g'(in_i)\frac{\partial}{\partial W_{j,i}}\left(\sum_j W_{j,i}a_j\right) \\
&= -(y_i - a_i)g'(in_i)a_j = -a_j\Delta_i
\end{aligned}$$

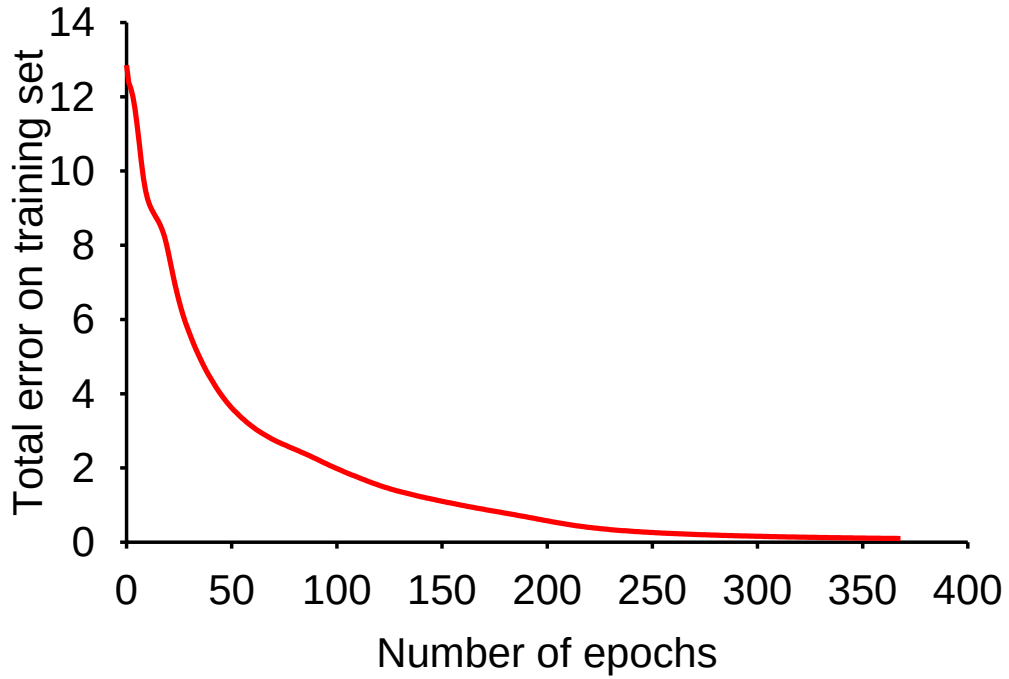
Izvod propagiranja unatrag nastavak

$$
\begin{aligned}
\frac{\partial E}{\partial W_{k,j}} &= -\sum_i (y_i - a_i)\frac{\partial a_i}{\partial W_{k,j}} = -\sum_i (y_i - a_i)\frac{\partial g(in_i)}{\partial W_{k,j}} \\
&= -\sum_i (y_i - a_i) g'(in_i)\frac{\partial in_i}{\partial W_{k,j}} = -\sum_i \Delta_i \frac{\partial}{\partial W_{k,j}}\left(\sum_j W_{j,i} a_j\right) \\
&= -\sum_i \Delta_i W_{j,i}\frac{\partial a_j}{\partial W_{k,j}} = -\sum_i \Delta_i W_{j,i}\frac{\partial g(in_j)}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial in_j}{\partial W_{k,j}} \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j)\frac{\partial}{\partial W_{k,j}}\left(\sum_k W_{k,j} a_k\right) \\
&= -\sum_i \Delta_i W_{j,i} g'(in_j) a_k = -a_k \Delta_j
\end{aligned}
$$

Učenje propagiranjem unatrag nastavak

U svakoj epohi, zbroji sve korekcije gradijenata za sve primjere i primijeni ih

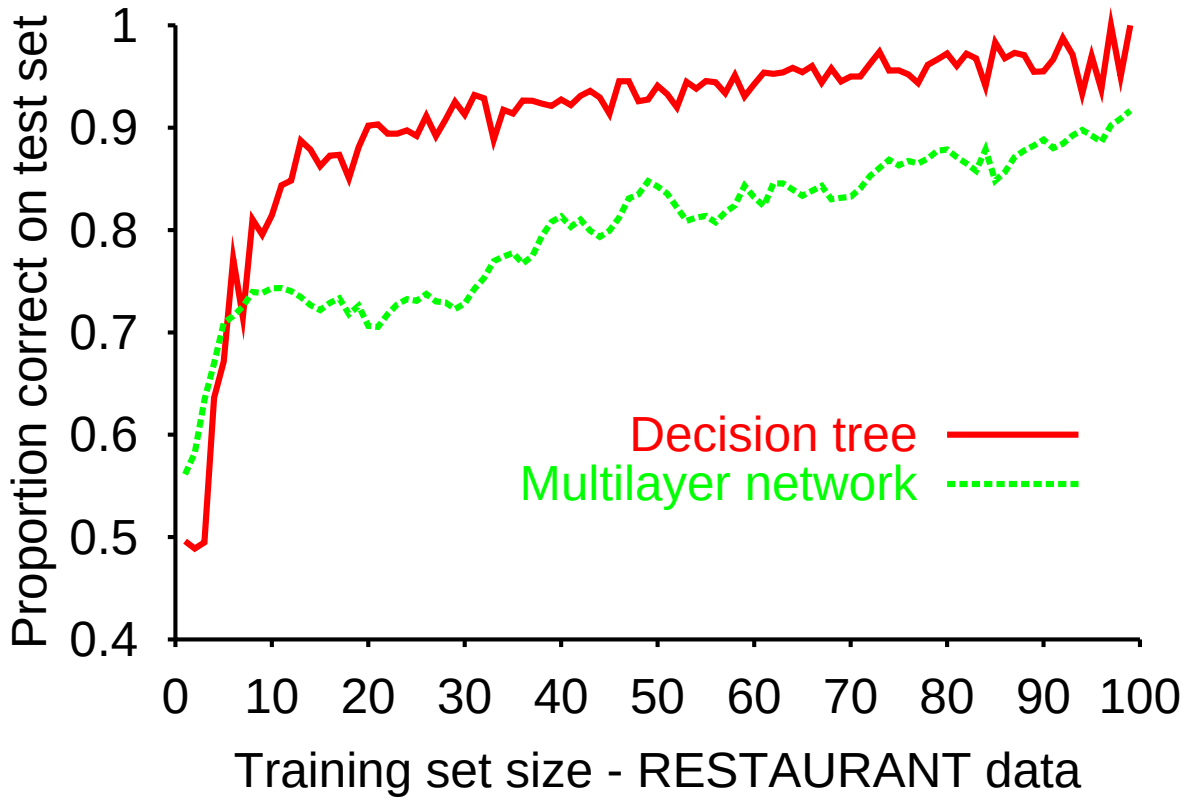
Krivulja treniranja za primjer 100 restorana: nalazi egzaktnu aproksimaciju



Tipični problemi: spora konvergencija, lokalni minimum

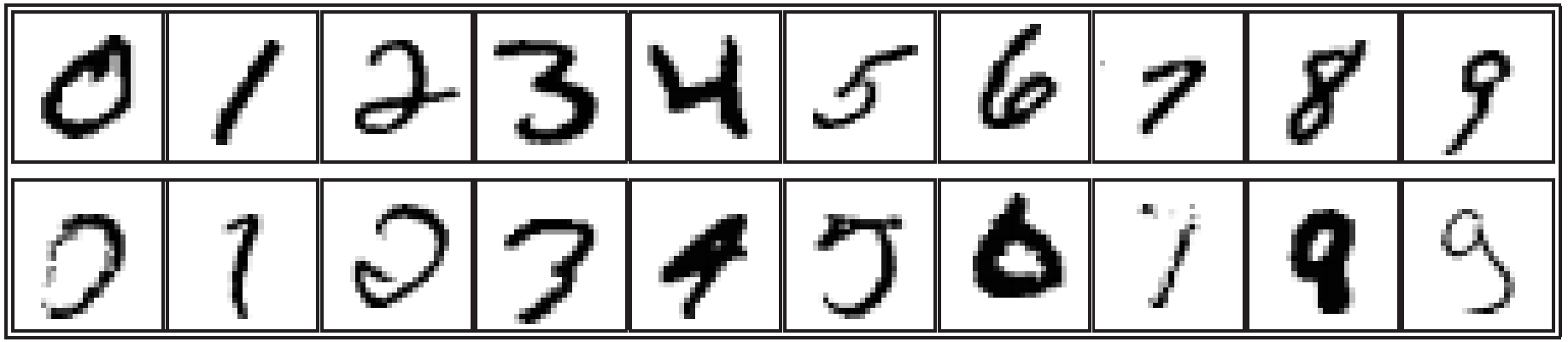
Učenje propagiranjem unatrag nastavak

Krivulja treniranja za MLP s 4 skrivene jedinice:



MLP-i su prilično dobri za zadatke složenog prepoznavanja oblika, ali rezultirajuće hipoteze ne mogu se jednostavno razumjeti

Prepoznavanje rukom pisanih znamenki



3-najbliža-susjeda = 2.4% greška
400–300–10 jedinica MLP = 1.6% greška
LeNet: 768–192–30–10 jedinica MLP = 0.9% greška

Trenutno najbolji (jezgreni strojevi, algoritmi vida) $\approx$ 0.6% greške

Sažetak

Većina mozgova ima mnogo neurona; svaki neuron $\approx$ linearno–granična jedinica (?)

Perceptroni (jednoslojne mreže) nedovoljno izražajni

Višeslojne mreže su dovoljno izražajne; mogu se trenirati metodom smanjenja gradijenata, tj., propagiranjem greške unatrag

Mnogo primjena: govor, vožnja, rukopis, detekcija prijevare, itd.

Inženjerstvo, kognitivno modeliranje, modeliranje neuronskih sustava primjene su uvelike divergirale