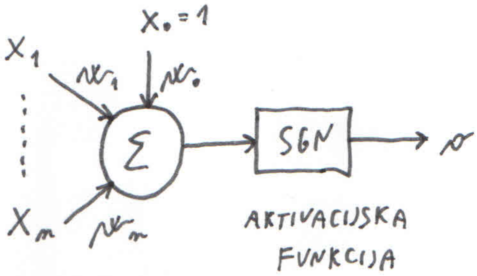
UMJETNE NEURONSKE MREŽE

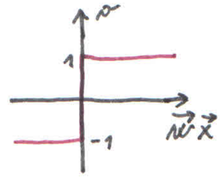
LITERATURA: „MACHINE LEARNING", T. MITCHELL (1997) [ML]

„DEEP LEARNING", I. GOODFELLOW, Y. BENGIO, A. COURVILLE (2016) [DL]

PERCEPTRON [ML P.86]



$$o(x_1, \dots, x_m) = \begin{cases} 1 & \text{AKO } \sum_{i=0}^{m} w_i x_i > 0 \\ -1 & \text{AKO } \sum_{i=0}^{m} w_i x_i \le 0 \end{cases}$$



$$o(\vec{w}\vec{x}) = SGN(\vec{w}\vec{x}) = \begin{cases} 1 & \text{AKO } \vec{w}\vec{x} > 0 \\ -1 & \text{AKO } \vec{w}\vec{x} \le 0 \end{cases}$$

PROSTOR HIPOTEZA: $H = \{\vec{w} \mid \vec{w} \in \mathbb{R}^{(m+1)}\}$

HIPERRAVNINE!

$t \in \{-1, 1\}$ CILJNA VARIJABLA
$o \in \{-1, 1\}$ IZLAZ

BINARNI KLASIFIKACIJSKI PROBLEM

UČENJE PERCEPTRONA:

$$\vec{w}^{(i+1)} \leftarrow \vec{w}^{(i)} + \Delta\vec{w}^{(i)}$$

$$\Delta\vec{w}^{(i)} = \eta(t - o)\vec{x}^{(i)}$$

STOPA UČENJA

KONVERGENCIJA SAMO U SLUČAJU LINEARNE ODVOJIVOSTI!

UZORCI $(\vec{x}^{(i)}, t^{(i)})$ SE PREDSTAVLJAJU JEDAN PO JEDAN I NAKON SVAKOG KRIVO KLASIFICIRANOG SE RADI UPDATE TEŽINA $\Delta\vec{w}^{(i)}$

$\boxed{\text{VC DIMENZIJA PERCEPTRONA JE } m+1}$

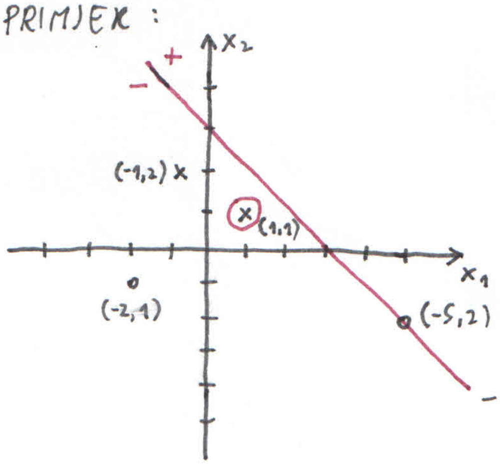
(NAJVEĆI BROJ PRIMJERA KOJE MOŽEMO RAZDVOJITI S PERCEPTRONOM BEZ OBZIRA NA LABELE)

INTUITIVNI DOKAZ:

AKO $o = -1$, $t = 1$ ONDA $t - o > 0$ I $\vec{w}\vec{x}$ SE TREBA POVEĆATI

AKO $o = 1$, $t = -1$ ONDA $t - o < 0$ I $\vec{w}\vec{x}$ SE TREBA SMANJITI

PRIMJER:



$-3 + x_1 + x_2 = 0$

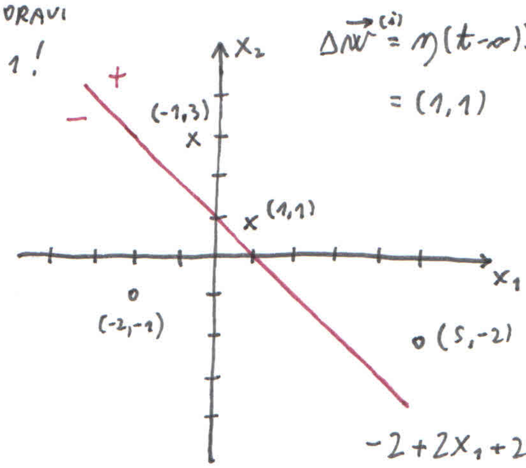
$\vec{x}^{(i)} = (1, 1)$

NE ZABORAVI NA $x_0 = 1$!

$t^{(i)} = +1$

$o^{(i)} = -1$

$\vec{w}^{(i)} = (-3, 1, 1)$



$\Delta\vec{w}^{(i)} = \eta(t - o)\vec{x}^{(i)} = (1, 1)$

$-2 + 2x_1 + 2x_2 = 0$

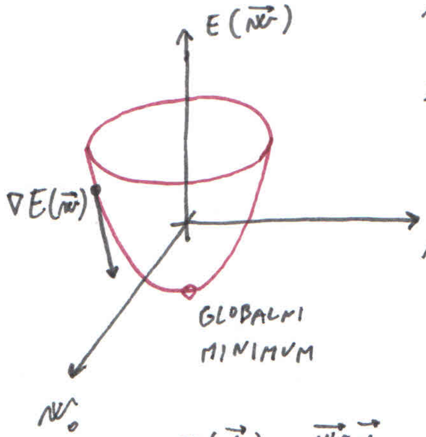
$-1 + x_1 + x_2 = 0$

GRADIJENTNI SPUST [ML P.89]

$o(\vec{x}) = \vec{w}\vec{x}$ IZLAZ BEZ AKTIVACIJSKE FUNKCIJE!

$$E(\vec{w}) = \frac{1}{2}\sum_{i \in D}(t^{(i)} - o^{(i)})^2$$

$$\nabla E(\vec{w}) = \left[\frac{\partial E}{\partial w_0}, \dots, \frac{\partial E}{\partial w_m}\right]$$



1. KONVEKSNOST!
2. BAYES-OPTIMALNA UZ PRETPOSTAVKU GAUSSOVOG ŠUMA

$o(\vec{x}) = \vec{w}\vec{x}$ LINEARNA FUNKCIJA } KONVEKSNOST!

$o(\vec{x}) = f(\vec{w}\vec{x})$ POTENCIJALNO NELINEARNA FUNKCIJA } NEKONVEKSNOST!

UČENJE:

$$\vec{w}^{(i+1)} \leftarrow \vec{w}^{(i)} + \Delta\vec{w}^{(i)}$$

$$\Delta\vec{w}^{(i)} = -\eta \nabla E(\vec{w})$$

$$w_k^{(i+1)} \leftarrow w_k^{(i)} + \Delta w_k^{(i)}$$

$$\Delta w_k^{(i)} = -\eta \frac{\partial E}{\partial w_k^{(i)}} = \underbrace{\eta \sum_{i \in D} \underbrace{(t^{(i)} - o^{(i)})\, x_k^{(i)}}_{\text{STOHASTIČKI GRAD. SPUST}}}_{\text{GRADIJENTNI SPUST}}$$

DOKAZ:

$$\frac{\partial E}{\partial w_k} = \frac{\partial}{\partial w_k}\frac{1}{2}\sum_{i \in D}(t^{(i)} - o^{(i)})^2 =$$

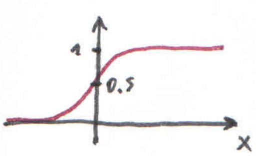
$$= \frac{1}{2}\sum_{i \in D} 2(t^{(i)} - o^{(i)}) \frac{\partial}{\partial w_k}\left(t^{(i)} - \vec{w}^{(i)}\vec{x}^{(i)}\right) = \sum_{i \in D}(t^{(i)} - o^{(i)})(-x_k^{(i)})$$

KOMENTARI:

- GRAD. SPUST MOŽEMO KORISTITI KAD IMAMO $\vec{w} \in \mathbb{R}^m$ I AKO $\nabla E(\vec{w})$ POSTOJI
- KONVERGENCIJA I KAD UZORCI NISU LINEARNO ODVOJIVI!
- MINIMIZACIJA KVADRATNE GREŠKE ($E(\vec{w})$), NE I UKUPNOG BROJA KRIVO KLASIFICIRANIH
- POTENCIJALNO SPORA KONVERGENCIJA (DODAVANJE MOMENTA I HESSIJANA)
- PROBLEMI AKO JE $E(\vec{w})$ NEKONVEKSNA (MINIMUM NIJE JEDINSTVEN)

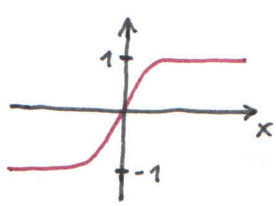
NELINEARNE AKTIVACIJSKE FUNKCIJE:

SIGMOIDA: $\sigma(\vec{w}\vec{x}) = \dfrac{1}{1 + e^{-\vec{w}\vec{x}}}$



$$\frac{d\,\sigma(y)}{dy} = \sigma(y)(1 - \sigma(y))$$

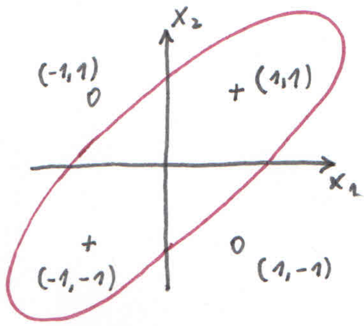
TANGENS HIPERBOLNI: $\tanh(\vec{w}\vec{x}) = \dfrac{-e^{-\vec{w}\vec{x}} + e^{\vec{w}\vec{x}}}{e^{-\vec{w}\vec{x}} + e^{\vec{w}\vec{x}}}$



$$\frac{d\,\tanh(y)}{dy} = 1 - \tanh^2(y)$$

VIŠESLOJNE NEURONSKE MREŽE [ML P.95, DL P.168]

MOTIVACIJSKI PROBLEM (XOR PROBLEM): [DL P.171]

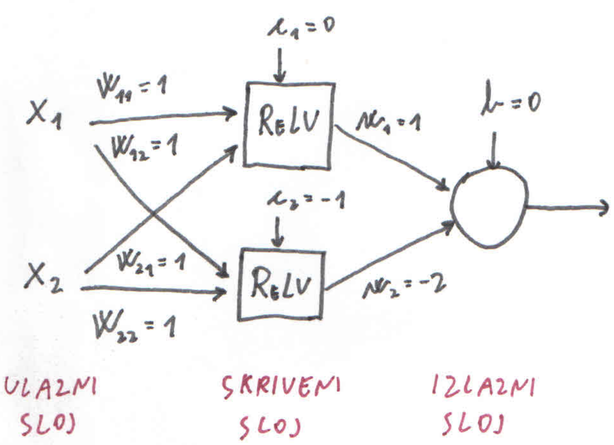


LINEARNO NESEPARABILAN, ALI MOGUĆE ODVOJITI S (RECIMO) ELIPSOM:

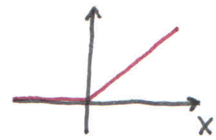
I DALJE LINEARAN PROBLEM (U PARAMETRIMA $\vec{w}$) ALI S EKSPLICITNIM NELINEARNIM MAPIRANJEM!

$$w_5 x_1^2 + w_4 x_1 x_2 + w_3 x_2^2 + w_2 x_1 + w_1 x_2 + w_0 = 0$$

S VIŠESLOJNIM NELINEARNIM NEUR. MR. MOŽEMO NAUČITI MAPIRANJE:



$$\mathrm{ReLU}(x) = \max\{0, x\}$$



RECTIFIED LINEAR UNIT

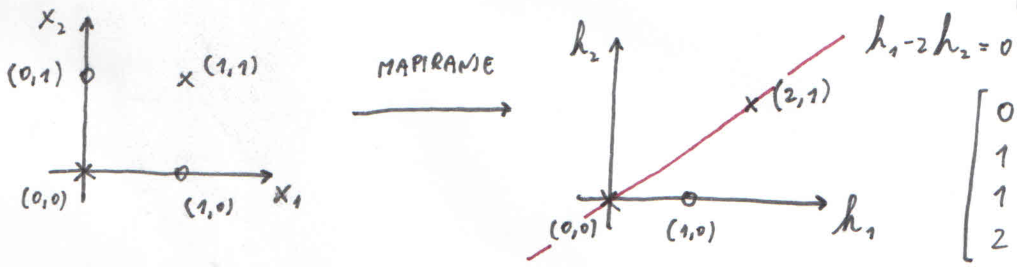
$$f(\vec{x}; W, \vec{c}, \vec{w}, b) = \overbrace{\vec{w}^T \underbrace{\max\{0, W^T\vec{x} + \vec{c}\}}_{\text{SKRIVENI SLOJ}} + b}^{\text{IZLAZNI SLOJ}}$$

$$W = \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \quad \vec{c} = \begin{bmatrix} 0 \\ -1 \end{bmatrix} \quad \vec{w} = \begin{bmatrix} 1 \\ -2 \end{bmatrix} \quad X = \begin{array}{c} \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \end{array} \begin{array}{c} + \\ 0 \\ 0 \\ + \end{array}$$

MAPIRANJE U SKRIVENOM SLOJU:

$$\max\{0, XW + \vec{c}\} = \max\left\{0, \begin{bmatrix} 0 & 0 \\ 0 & 1 \\ 1 & 0 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} + \begin{bmatrix} 0 \\ -1 \end{bmatrix}\right\} = \max\left\{0, \begin{bmatrix} 0 & -1 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}\right\} = \begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix}$$

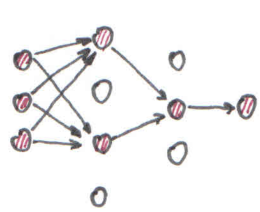
LINEARNO RAZDVOJIVI!



$$\begin{bmatrix} 0 & 0 \\ 1 & 0 \\ 1 & 0 \\ 2 & 1 \end{bmatrix} \begin{bmatrix} 1 \\ -2 \end{bmatrix} = \begin{bmatrix} 0 \\ 1 \\ 1 \\ 0 \end{bmatrix} \begin{array}{c} + \\ 0 \\ 0 \\ + \end{array}$$

KOMENTARI:

- KOMPOZICIJA LIN. FUN. JE I DALJE LIN. FUN., ZATO JE VIŠESLOJNA N.M. S LIN. AKT. FUN. EKVIVALENTNA JEDNOM PERCEPTRONU!
- ReLU NIJE DERIVABILNA, ZA RAZLIKU OD SIGMOIDE I TANH
- KORIŠTENJE ReLU JE EKVIVALENTNO KAO ODABIR PODSKUPA NEURONA I KORIŠTENJA LIN. AKT. FUN (ZA SVAKI UZORAK JE TAJ PODSKUP DRUGAČIJI!)



NEMA NESTAJUĆEG GRADIJENTA I RAČUNANJE JE EFIKASNIJE!

BACKPROPAGATION [ML P.97, DL P.204]

$$E_d(\vec{w}) = \frac{1}{2} \sum_{k \in OUT} (t_k - o_k)^2$$

FUNKCIJA GREŠKE ZA UZORAK d JE DEFINIRANA SAMO ZA IZLAZNI SLOJ!

$$\Delta w_{ji} = -\eta \frac{\partial E_d}{\partial w_{ji}}$$

UPDATE KOJI RADIMO ZA TEŽINU w_{ji} NAKON UZORKA d

j – NEURON, i – ULAZ U NEURON

$$\frac{\partial E_d}{\partial w_{ji}} = \frac{\partial E_d}{\partial\left(\sum_i w_{ji} x_{ji}\right)} \cdot \frac{\partial\left(\sum_i w_{ji} x_{ji}\right)}{\partial w_{ji}} = \underbrace{\frac{\partial E_d}{\partial\left(\sum_i w_{ji} x_{ji}\right)}}_{\delta_j} x_{ji}$$

CHAIN RULE

(DERIVACIJA KOMPOZICIJE FUNKCIJA)

IZLAZNI NEURONI: $\delta_j \leftarrow o_j(1-o_j)(t_j - o_j)$

SKRIVENI NEURONI: $\delta_j \leftarrow \underbrace{o_j(1-o_j)}_{\text{DERIVACIJA SIGMOIDE}} \underbrace{\sum_{k \in OUT} w_{kj}\delta_k}_{\text{KOLIKO JE NEURON } j \text{ ODGOVORAN ZA GREŠKU U IZLAZNOM NEURONU } k}$

ZA SIGMOIDALNU AKTIVACIJSKU FUN.!

$$\sigma(\vec{w}\vec{x}) = \frac{1}{1+e^{-\vec{w}\vec{x}}}$$

KOMENTARI:

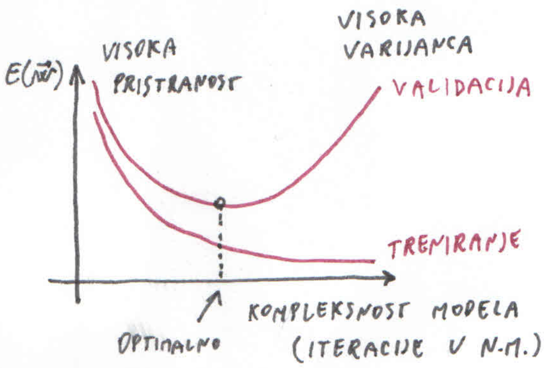
- GRADIJENT SE DISIPIRA ŠTO JE MREŽA DUBLJA, POČETNE TEŽINE SE MALO MIJENJAJU (VANISHING GRADIENT) – ReLU NEMA TAJ PROBLEM!

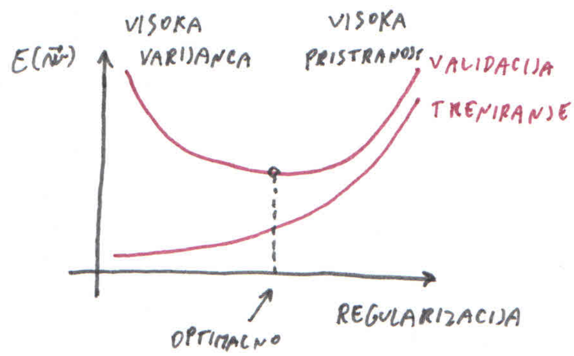
REGULARIZACIJA U NEURONSKIM MREŽAMA [ML P.108, DL P.228]

PRISTRANOST I VARIJANCA : [SCOT.FORTMANN-ROE.COM/DOCS/BIASVARIANCE.HTML]

$$Y = f(x) + \varepsilon \qquad \mathrm{Err}(x) = E\left[\left(Y - \hat{f}(x)\right)^2\right] =$$

$$= \underbrace{\left(E[\hat{f}(x)] - f(x)\right)^2}_{\text{PRISTRANOST}^2} + \underbrace{E\left[\left(\hat{f}(x) - E[\hat{f}(x)]\right)^2\right]}_{\text{VARIJANCA}} + \underbrace{\sigma_\varepsilon^2}_{\text{IREDUCIBILNA GREŠKA}}$$





REGULARIZACIJA U N.M. :

- PENALIZIRANJE TEŽINA $\vec{w}$

$$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} \sum_{k \in OUT} \left(t_k^{(d)} - o_k^{(d)}\right)^2 + \boxed{\gamma \sum_{i,j} w_{ij}^2}$$

KVADRATNA, L_2, TIKHONOVA REGULARIZACIJA

NAMEĆE GLATKOST U $\vec{w}$!

$$\boxed{\gamma \sum_{i,j} |w_{ij}|}$$

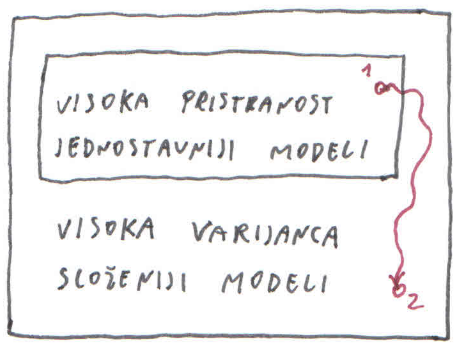
APSOLUTNA, L_1 REGULARIZACIJA

NAMEĆE RIJETKOST U $\vec{w}$!

- RANO PREKIDANJE UČENJA
- NAMETANJE INVARIJANCI (GENERIRANJE MODIFICIRANIH PRIMJERA, TRANS. ZNAČAJKI)
- LOKALNO POVEZIVANJE NEURONA I DIJELJENJE TEŽINA
- DROPOUT (NASUMIČNO „GAŠENJE" TEŽINA ILI NEURONA - IMPLICITNO STVARA ANSAMBL)

INICIJALIZACIJA TEŽINA:

PREPORUČA SE POČETI S MALIM TEŽINAMA JER TO ODGOVARA JEDNOSTAVNIJIM MODELIMA



REPREZENTACIJSKA MOĆ NEURONSKIH MREŽA [ML P.105, DL P.198]

UNIVERZALNI APROKSIMACIJSKI TEOREM (HORNIK, CYBENKO 1989.)

NEURONSKA MREŽA S BAREM JEDNIM SKRIVENIM SLOJEM SA „SQUASHING" AKT. FUN. (SIGMOIDA, TANH) I LINEARNIM VANJSKIM SLOJEM MOŽE PROIZVOLJNO TOČNO APROKSIMIRATI BILO KOJU BOREL-MJERLJIVU FUNKCIJU (SVAKA KONTINUIRANA FUN. NA ZATVORENOM PODSKUPU OD $\mathbb{R}^n$ JE BOREL-MJERLJIVA).

(LESHNO 1993.) EKSTENZIJA NA ReLU AKTIVACIJSKE FUNKCIJE

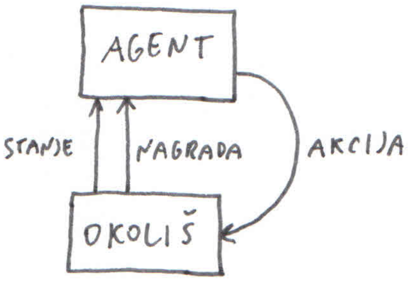
PROBLEMI:

- NE POSTOJI GARANCIJA DA JE MOGUĆE NAUČITI TEŽINE TAKVE MREŽE
- NE KAŽE SE KOLIKO JE NEURONA POTREBNO, U NAJGOREM SLUČAJU JE POTREBNO EKSPONENCIJALNI BROJ SKRIVENIH NEURONA
  (BROJ MOGUĆIH BINARNIH FUNKCIJA NA VEKTORIMA $v \in \{0,1\}^n$ JE 2^{2^n}, I POTREBNO JE 2^n BITOVA ZA SELEKCIJU JEDNE OD NJIH)

IPAK, INTUICIJA I PRAKSA POKAZUJU DA SU DUBOKE MREŽE PUNO EFIKASNIJE U UČENJU I REPREZENTACIJI:

NEURAL NETWORK PLAYGROUND:
PLAYGROUND.TENSORFLOW.ORG

UČENJE S PODRŠKOM



INTERAKCIJA S OKOLIŠEM : [ML P.368]

$$s_0 \xrightarrow{a_0} s_1 \xrightarrow{a_1} \dots\dots$$
$$\downarrow \qquad\qquad \downarrow$$
$$r_0 \qquad\qquad r_1$$

KUMULATIVNA NAGRADA :

$$r_0 + \gamma r_1 + \dots + \gamma^n r_n \qquad \cancel{0 \le \gamma \le 1} \quad 0 \le \gamma < 1$$

CILJ : NAUČI KONTROLNU STRATEGIJU $\pi : S \to A$

U NADZIRANOM UČENJU OČEKUJEMO PODATKE $(s, \pi(s))$

ZADATAK UČENJA : [ML P.370]

$$V^{\pi}(s_t) = r_t + \gamma r_{t+1} + \dots = \sum_{i=0}^{\infty} \gamma^i r_{t+i}$$ DISCOUNTED CUMMULATIVE REWARD

$$\pi^* \equiv \arg\max_{\pi} V^{\pi}(s), \ \forall s$$ OPTIMALNA STRATEGIJA

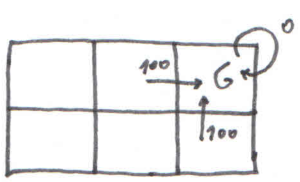
$$\pi^*(s) = \arg\max_{a} \left[r(s,a) + \gamma V^*(\delta(s,a)) \right]$$

FUNKCIJA NAGRADE — MAKSIMALNA KUMULATIVNA NAGRADA — FUNKCIJA PRIJELAZA

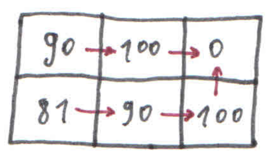
$\delta : S \times A \to S$ DETERMINISTIČKA

$\delta : S \times A \to \mathrm{PROB}(S)$ NEDETERMINISTIČKA

PRIMJER :



$\gamma = 0.9$



G - GRANIČNO STANJE

$r(s,a)$ - FUNKCIJA NAGRADE

$V^*(s)$ MAKSIMALNA KUMULATIVNA NAGRADA

$\pi^*(s)$ OPTIMALNA STRATEGIJA

Q LEARNING [ML P.373]

$$\pi^*(s) = \underset{a}{\mathrm{ARG\,MAX}} \left[r(s,a) + \gamma V^*(\delta(s,a)) \right]$$

KADA BI $r(s,a)$ I $\delta(s,a)$ BILE U POTPUNOSTI POZNATE GORNJA BI SE JEDNADŽBA MOGLA ISKORISTITI ZA UČENJE $\pi^*(s)$ (ONDA SE TO ZOVE DINAMIČKO PROGRAMIRANJE!). UMJESTO TOGA, UČIMO Q-FUNKCIJU:

$$Q(s,a) = r(s,a) + \gamma \max_{a'} Q(\delta(s,a), a')$$

AKCIJA DOSTUPNA IZ SLJEDEĆEG STANJA $\delta(s,a)$

$$V^*(s) = \max_{a'} Q(s,a')$$

FUNKCIJA STANJA

FUNKCIJA STANJA I AKCIJE (PRAKTIČNIJE)

UČENJE Q-FUNKCIJE: $\boxed{\hat{Q}(s,a) \leftarrow r(s,a) + \gamma \max_{a'} \hat{Q}(s',a')}$ $0 \le \gamma < 1$

KAKO ODABIRATI AKCIJE?

$$P(a_i|s) = \frac{k^{\hat{Q}(s,a_i)}}{\sum_j k^{\hat{Q}(s,a_j)}}$$

$k>0$, REGULIRA KOLIKO SE PREFERIRAJU AKCIJE S VELIKIM $Q(s,a)$ VRIJEDNOSTIMA (EXPLORATION VS EXPLOATATION)

UVJETI ZA KONVERGENCIJU Q UČENJA:

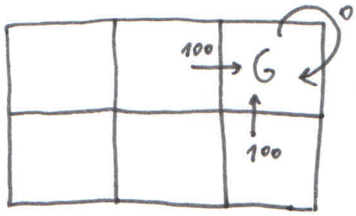
1. DETERMINISTIČKI MARKOVLJEV PROCES ODLUČIVANJA
2. NAGRADE SU OGRANIČENE
3. BESKONAČNO POSJEĆIVANJE SVAKOG PARA STANJE/AKCIJA

UVJET 3. JE NEREALISTIČAN, ZATO SE U PRAKSI Q-FUNKCIJA APROKSIMIRA NEURALNOM MREŽOM!

PRIMJERI IZ PRAKSE:

- „HUMAN-LEVEL CONTROL THROUGH DEEP R.L." (2015)
- „MASTERING THE GAME OF GO WITH DEEP N.N. AND TREE SEARCH" (2016)

PRIMJER ZA Q-UČENJE [ML p.376]

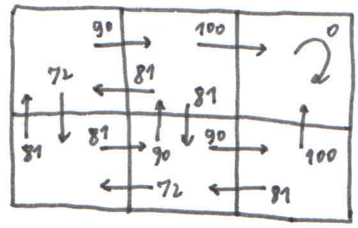


$\gamma = 0.9$

$r(s,a)$

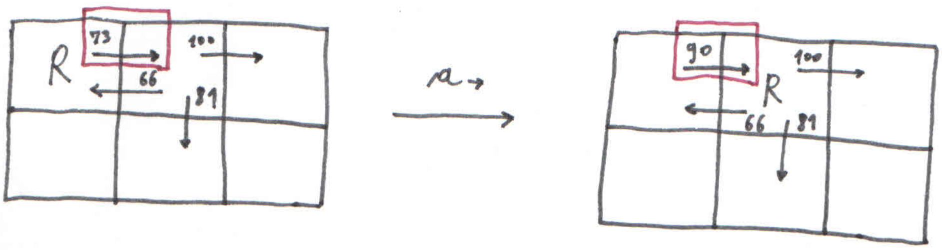
90	100	0
81	90	100

$V^*(s)$



$Q(s,a)$

JEDNA ITERACIJA Q-UČENJA:



$$\hat{Q}(s_1, a_\rightarrow) \leftarrow \underbrace{r(s_1, a_\rightarrow)}_{0} + \underbrace{\gamma}_{0.9} \underbrace{\max_{a'} \hat{Q}(s_2, a')}_{\max\{66, 81, 100\}} = 90$$