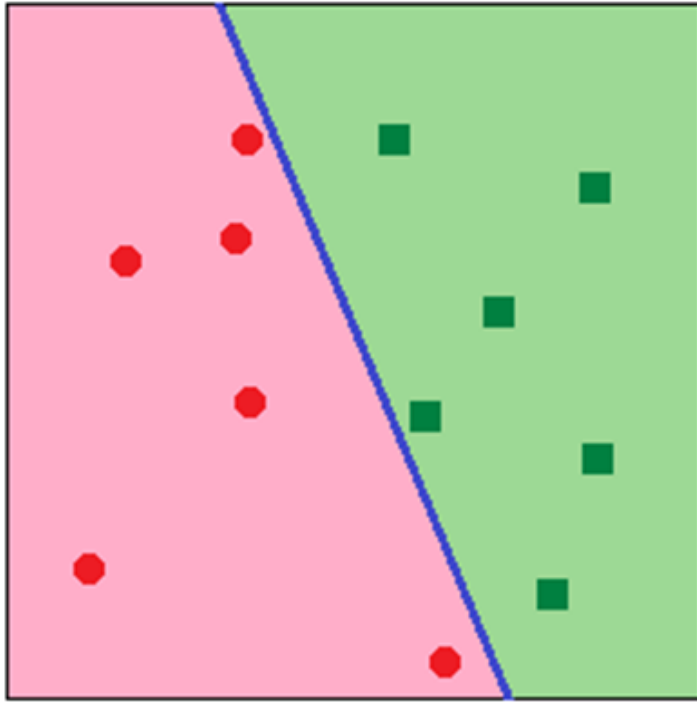
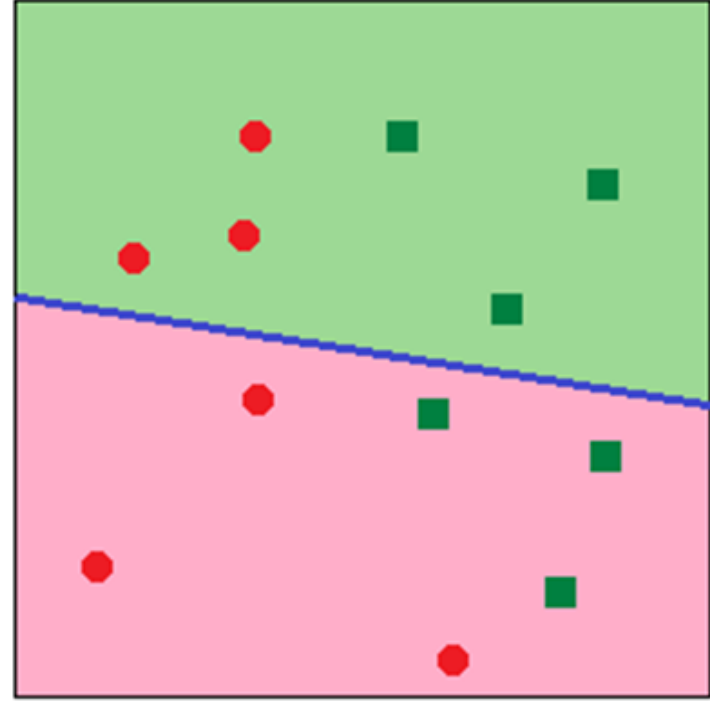


Cilj

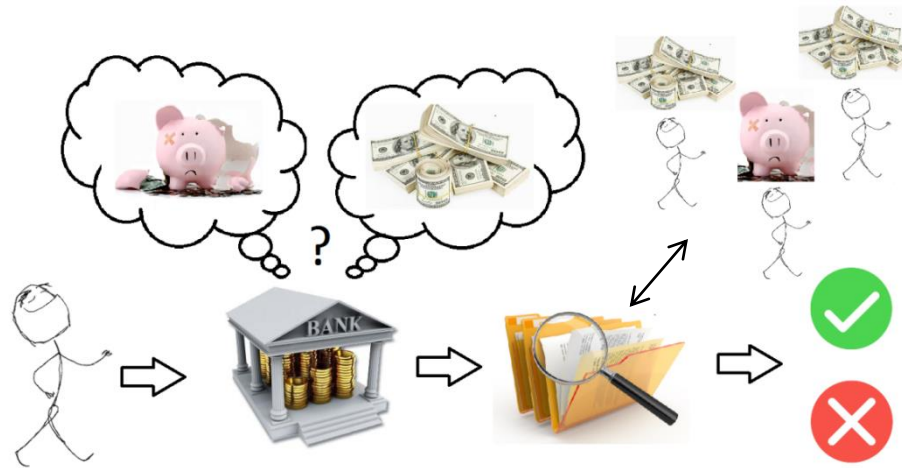


Inicijalizacija (slučajna)



Perceptron

Primer klasifikacije – odobravanje kredita



starost	23 godine
pol	Muški
Godišnja zarada	\$30 000
Trenutni dug	\$15 000
Posедује nekretninu	Da
...	...

- Mušterija je predstavljena nizom obeležja $x = (x_1, x_2, \dots, x_d)$
- Svakom obeležju će biti dodeljena težina θ_i zavisno od toga koliko je važan za problem
- Npr. informacija o da li mušterija ima dug je veoma važna – veliki dug znači da ne treba odobriti kredit. Težina može da bude negativna u ovom slučaju

Perceptron

- Pristup logističke regresije modifikovan na taj način da je „prisiljen“ da izbacuje vrednosti koje su tačno -1 ili +1:

$$h(x, \theta) = \begin{cases} +1, & \sum_{i=1}^d \theta_i x_i > \text{threshold} \\ -1, & \sum_{i=1}^d \theta_i x_i < \text{threshold} \end{cases}$$

$$h(x) = \text{sign} \left(\left(\sum_{i=1}^d \theta_i x_i \right) - \text{threshold} \right)$$

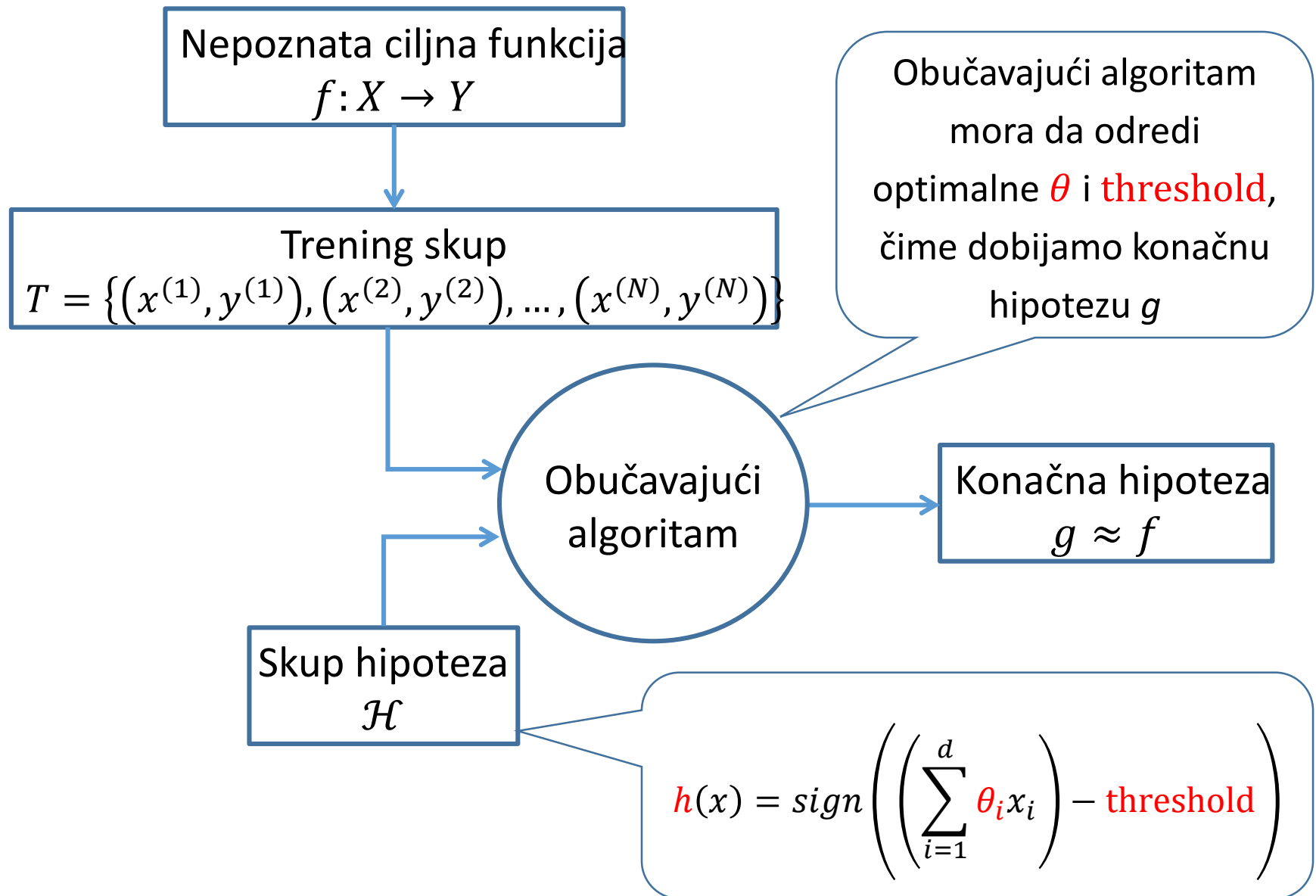
linear
formula
 $h \in \mathcal{H}$

Perceptron

$$h(x) = \text{sign} \left(\left(\sum_{i=1}^d \theta_i x_i \right) - \text{threshold} \right)$$

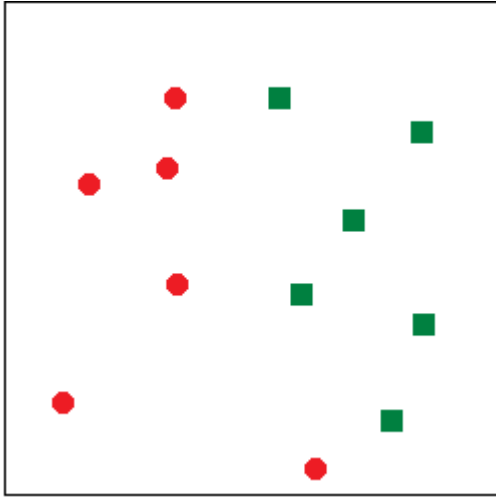
- Ono što razlikuje jednu hipotezu od druge je naš izbor vrednosti *parametara modela*: θ_i i **threshold**

Komponente obučavanja (podsetnik)

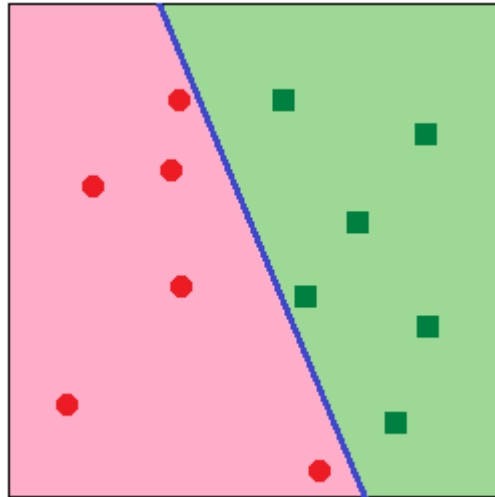


Linearno separabilni podaci

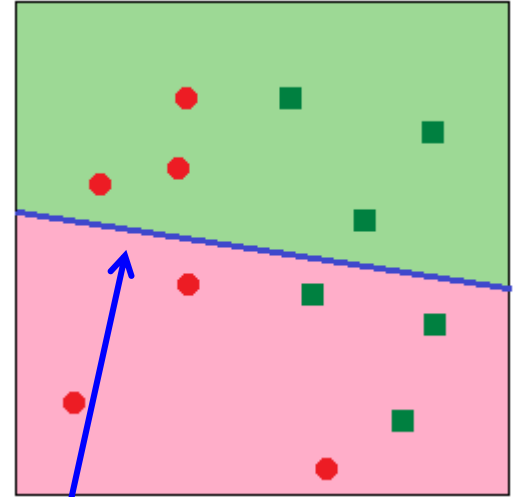
Podaci su linearno separabilni



Cilj



Inicijalizacija (slučajna)



$$h(x) = \text{sign} \left(\left(\sum_{i=1}^d \theta_i x_i \right) - \text{threshold} \right)$$

Skup hipoteza

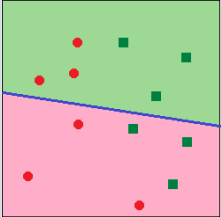
- Malo ćemo izmeniti notaciju:

$$h(x) = \text{sign} \left(\left(\sum_{i=1}^d \theta_i x_i \right) - \text{threshold} \right) \rightarrow h(x) = \text{sign} \left(\left(\sum_{i=1}^d \theta_i x_i \right) - \theta_0 \right)$$

- Uvešćemo $x_0 = 1$:

$$h(x) = \text{sign} \left(\sum_{i=0}^d \theta_i x_i \right) = \text{sign}(\theta^T x)$$

Perceptron Learning Algorithm (PLA)

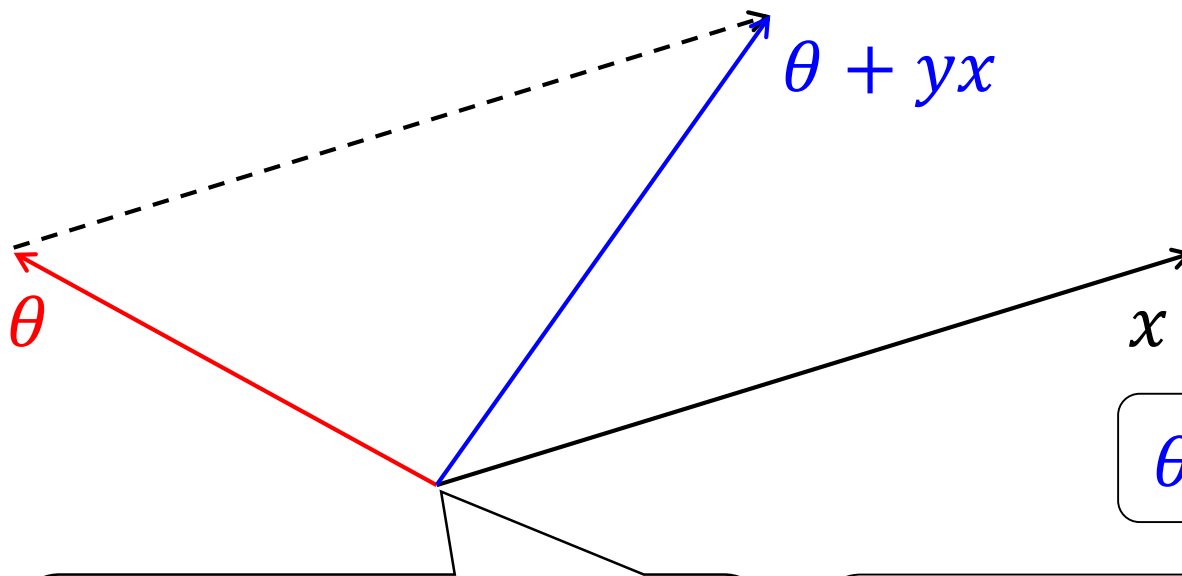
Ulaz	<ul style="list-style-type: none">• $T = \{(x^{(i)}, y^{(i)}), i = 1, \dots, N\}$• maxlters 
Postupak	<p>for $t = 1, 2, \dots, \text{maxlters}$:</p> <ul style="list-style-type: none">• Iz T na slučajan način odabrati pogrešno klasifikovanu tačku $(x^{(n)}, y^{(n)})$• Na osnovu $(x^{(n)}, y^{(n)})$ ažurirati parametre θ: $\theta^{(t)} \leftarrow \theta^{(t-1)} + y^{(n)} x^{(n)}$
Izlaz	$h_{\theta}(x) = \text{sign}(\theta^T x)$

PLA ažuriranje parametara

Pogrešno klasifikovana tačka:

$$\text{sign}(\theta^T x^{(n)}) \neq y^{(n)}$$

$$\theta \leftarrow \theta + y^{(n)} x^{(n)}$$



Ako je ugao između vektora θ i x tup, onda je $\text{sign}(\theta x) = -1$

Pošto je tačka pogrešno klasifikovana, to znači da je (tačno) $y = +1$

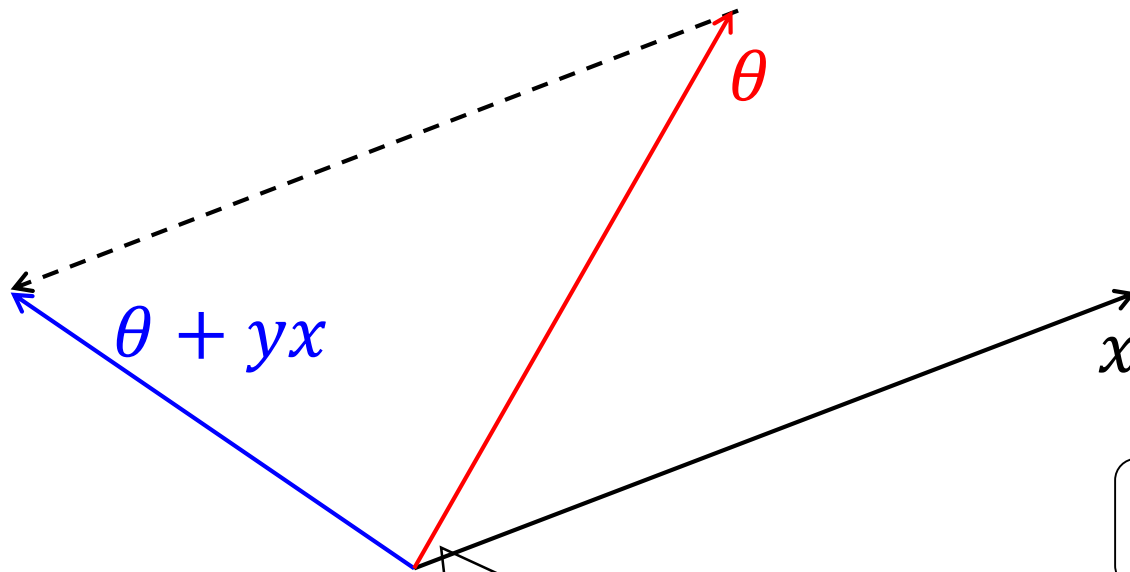
$$\theta \leftarrow \theta + x^{(n)}$$

PLA ažuriranje parametara

Pogrešno klasifikovana tačka:

$$\text{sign}(\theta^T x^{(n)}) \neq y^{(n)}$$

$$\theta \leftarrow \theta + y^{(n)} x^{(n)}$$

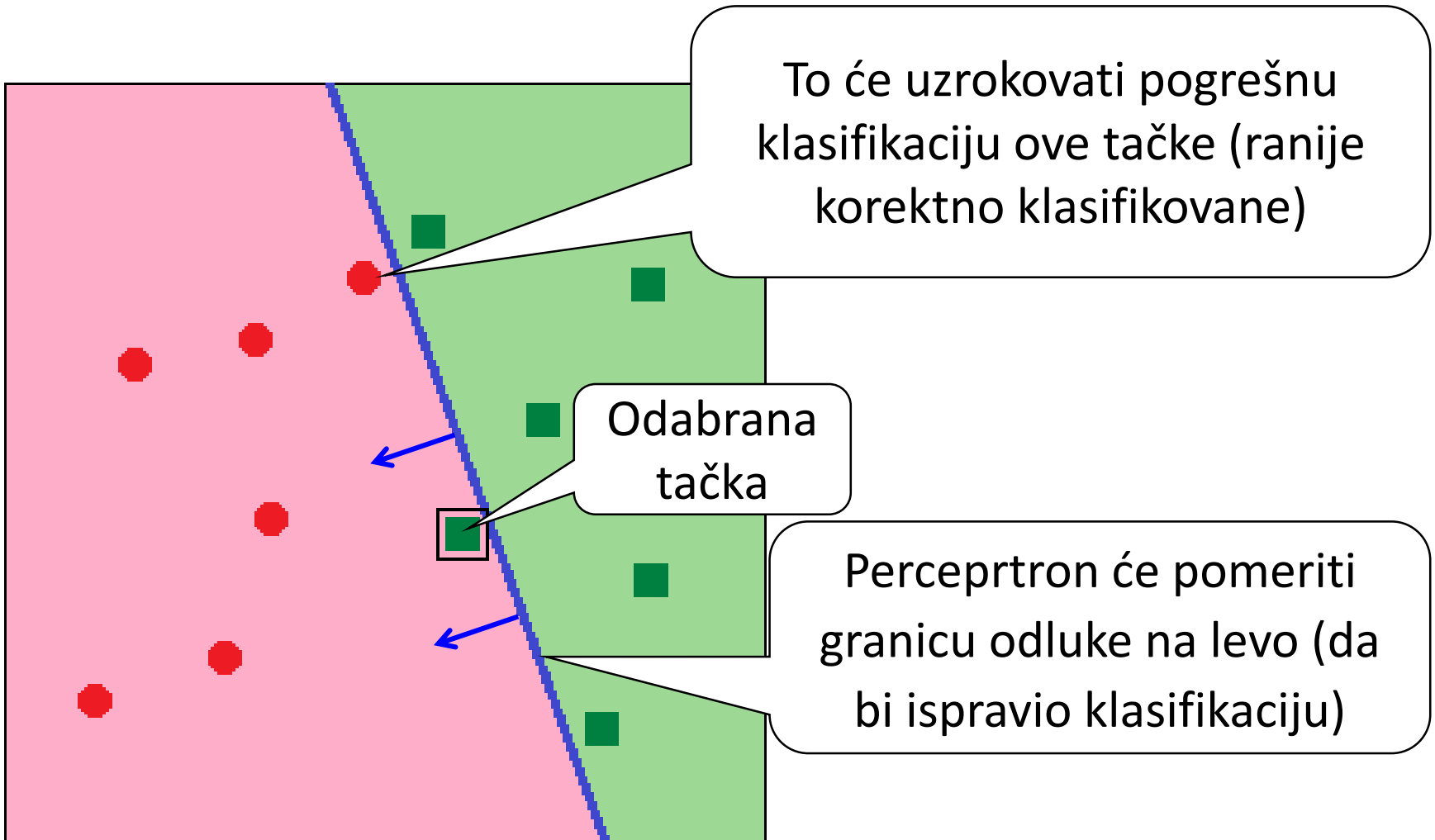


$$\theta \leftarrow \theta - x^{(n)}$$

Ako je ugao između vektora θ i x oštar, onda je $\text{sign}(\theta x) = +1$

Pošto je tačka pogrešno klasifikovana, to znači da je (tačno) $y = -1$

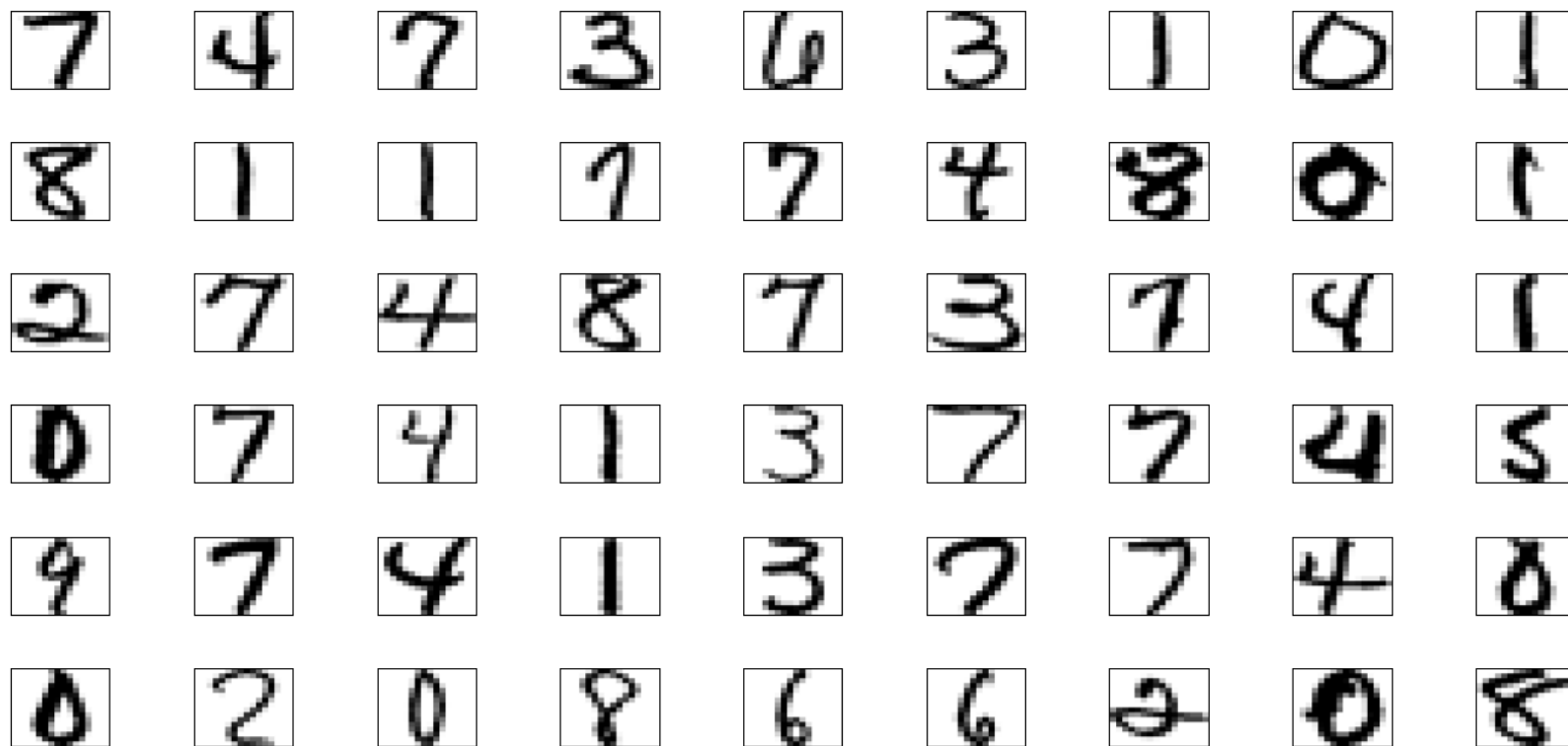
PLA ažuriranje parametara



PLA ažuriranje parametara

- Dakle, pokušavamo da sredimo jednu tačku, ali pritom možemo da pokvarimo sve ostale tačke jer ih ne uzimamo u obzir prilikom pomeranja granice odluke
- Ipak, ako su podaci **linearno separabilni**, tokom vremena ćemo dobiti korektno rešenje

Primer: OCR



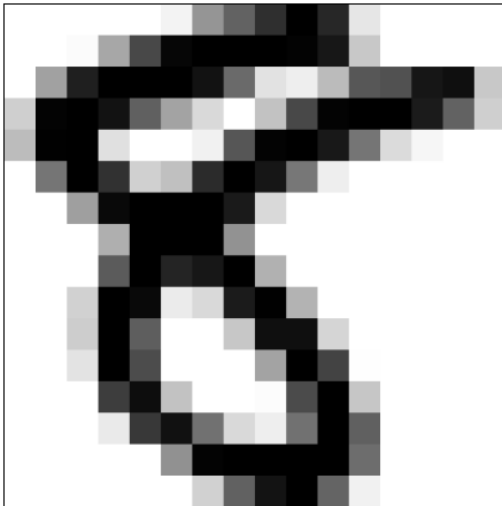
Slike 16 x 16 piksela (realne vrednosti)

„Sirov“ ulaz $x = (x_0, x_1, x_2, \dots, x_{256})$

Linearan model: $(\theta_0, \theta_1, \dots, \theta_{256})$

Primer: OCR

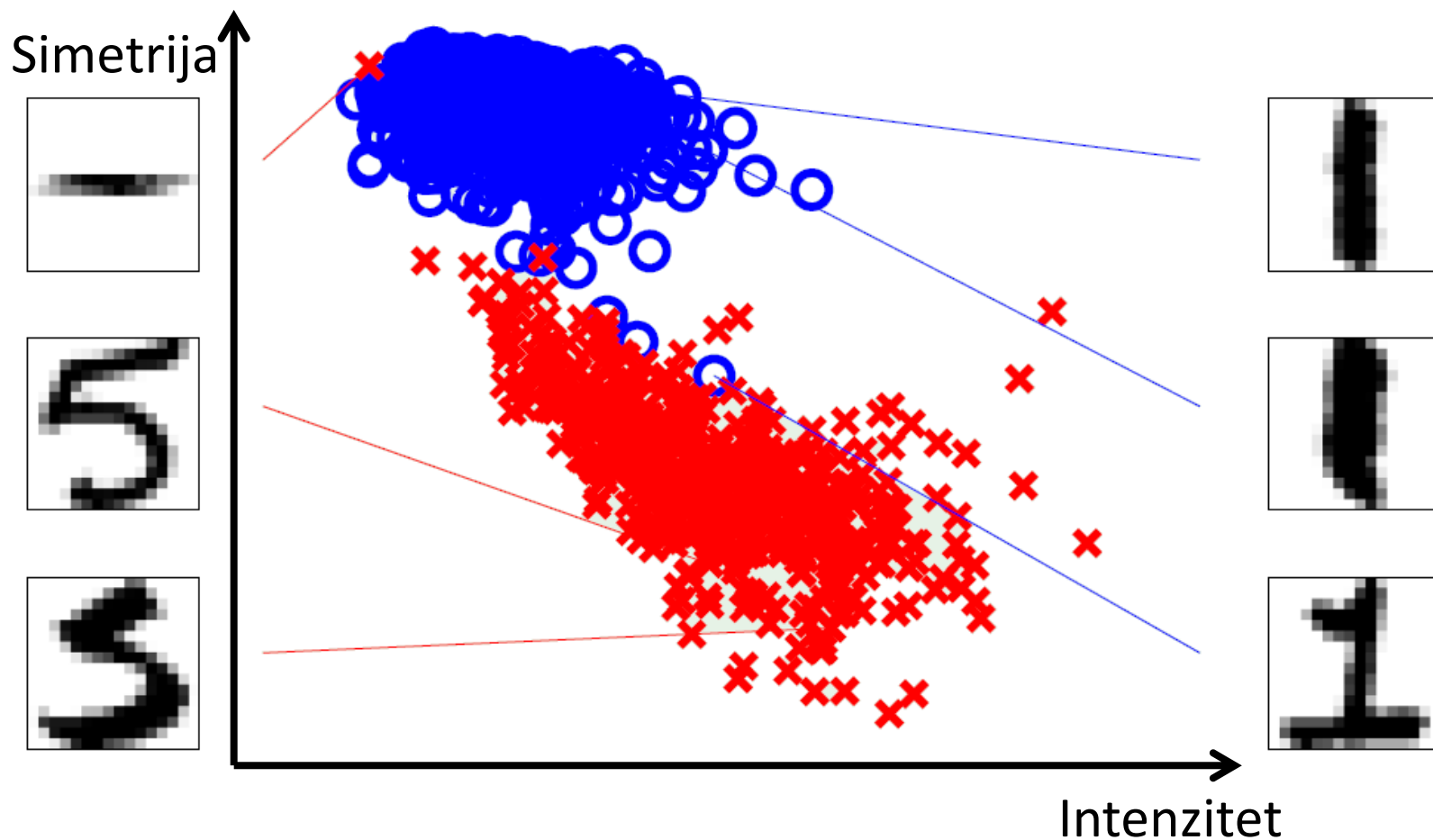
- Pretprocesiranje: izvući ćemo korisne informacije – obeležja koja će nam pomoći da lakše rešimo problem
- Npr. ispostavilo se da su za problem prepoznavanja rukom napisanih cifara važni
 - Simetrija
 - Prosečan intenzitet



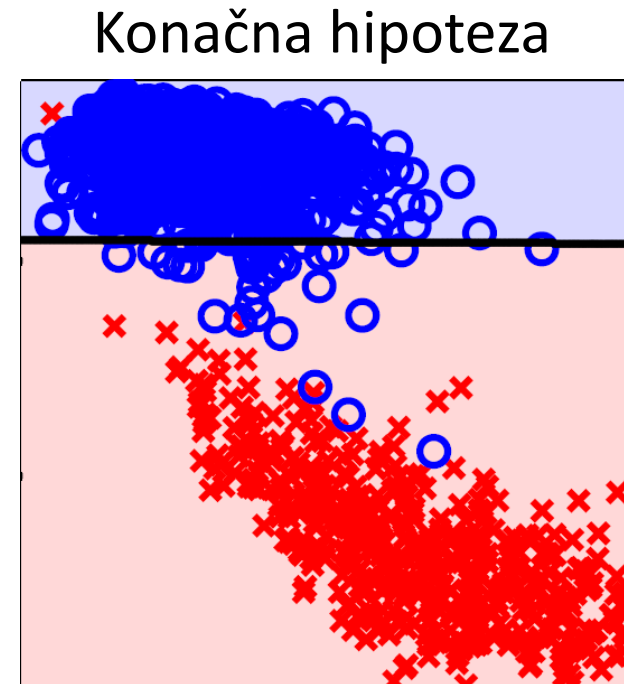
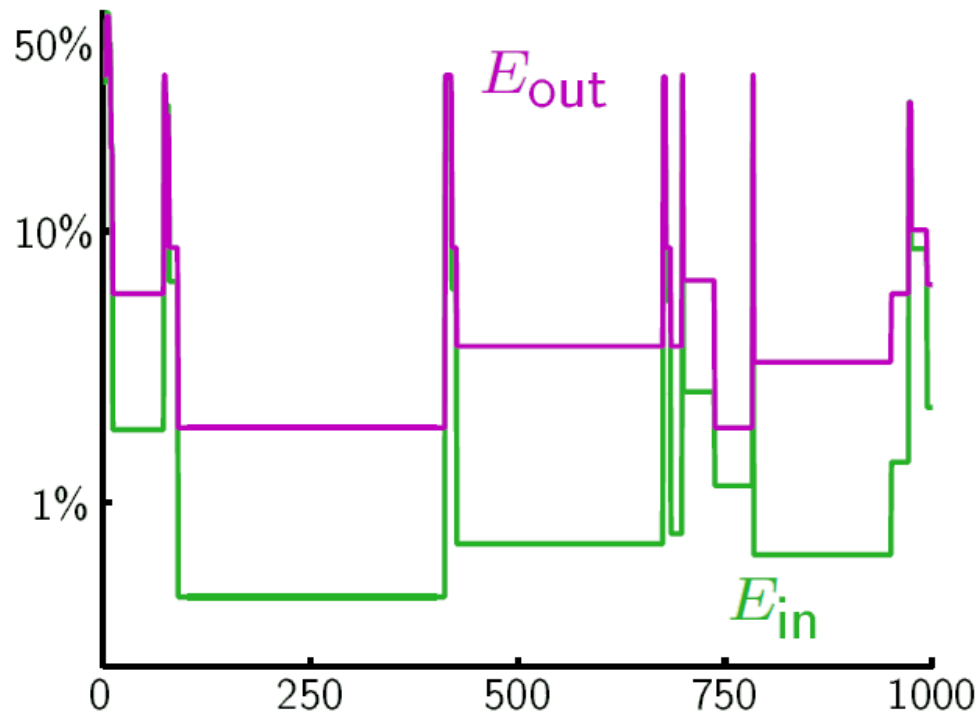
$$x = (x_0, x_1, x_2)$$

Linearan model: $(\theta_0, \theta_1, \theta_2)$

Primer: OCR

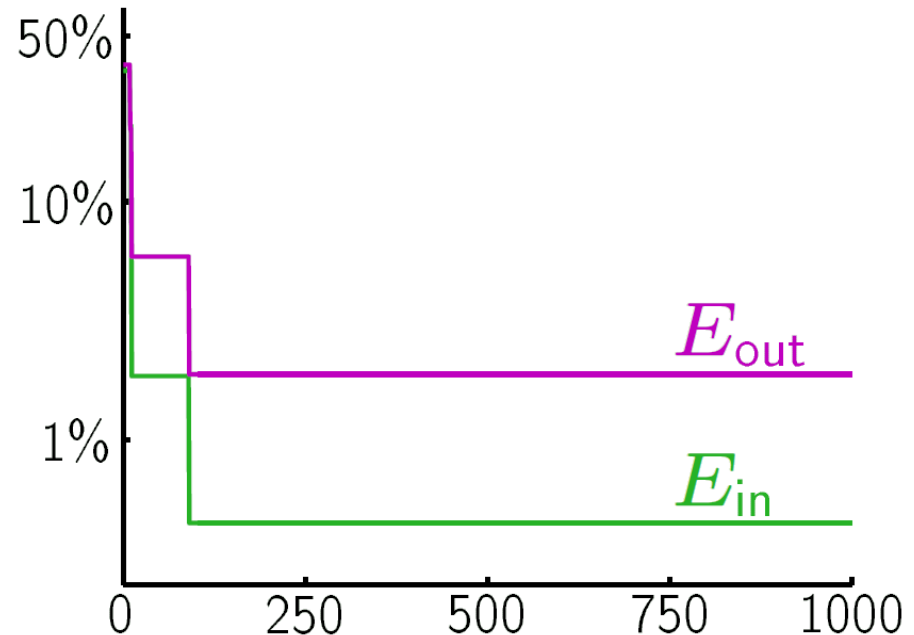
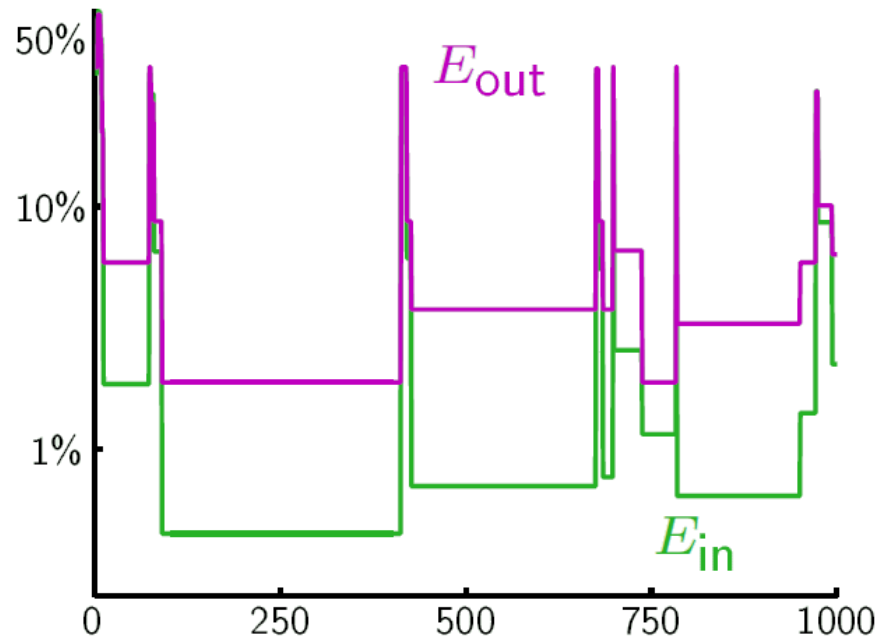


PLA na OCR primeru



- Podaci nisu linearno separabilni
 - perceptron se u ovom slučaju ponaša loše
 - Nikada neće konvergirati, prekidamo ga nakon 1000 iteracija

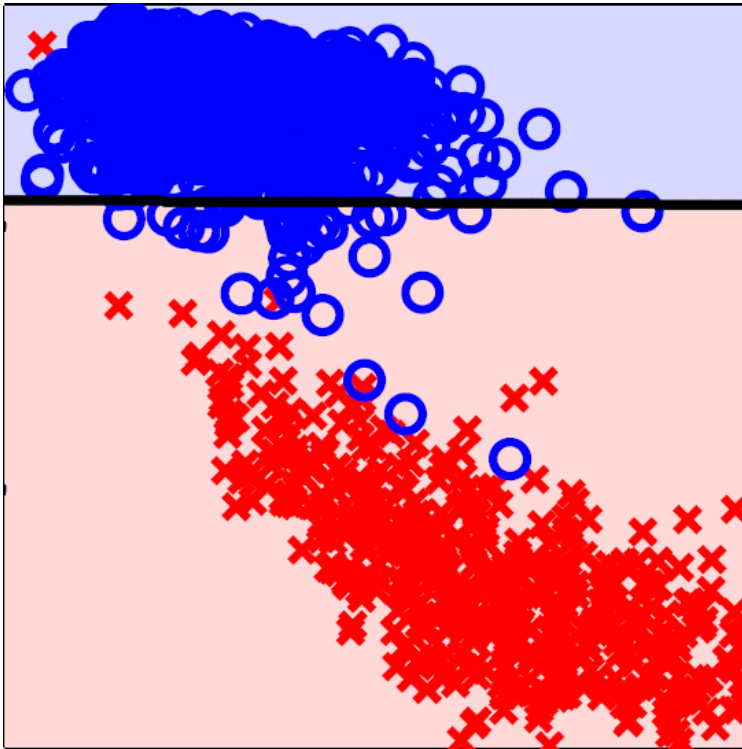
Modifikacija perceptrona – „Pocket“



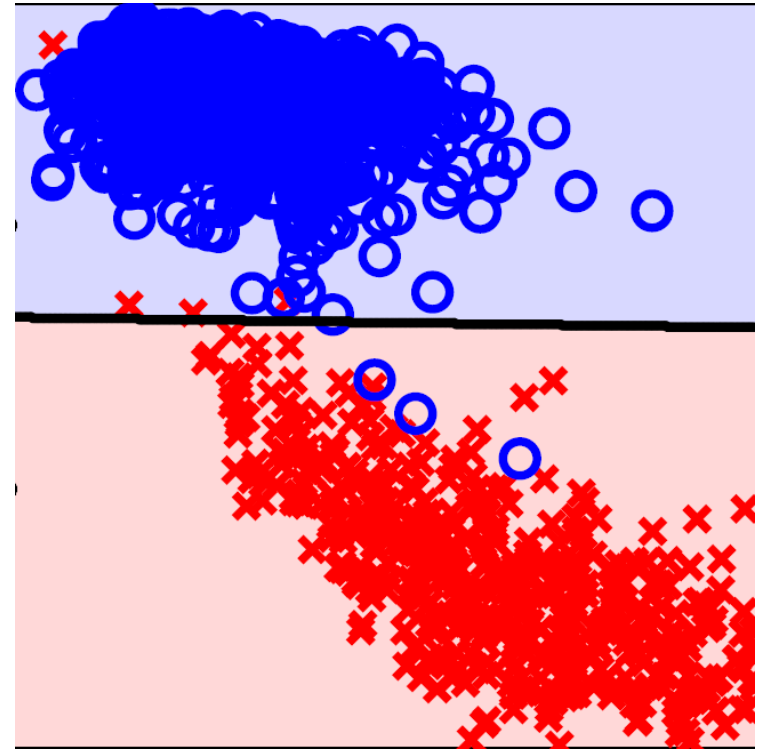
- „Pocket“: primenjujemo PLA, ali čuvamo najbolju hipotezu (prema E_{in}). Prekidamo nakon zadatog broja iteracija
- Bolji od perceptrona na podacima koji nisu linearno separabilni

Konačna hipoteza

PLA



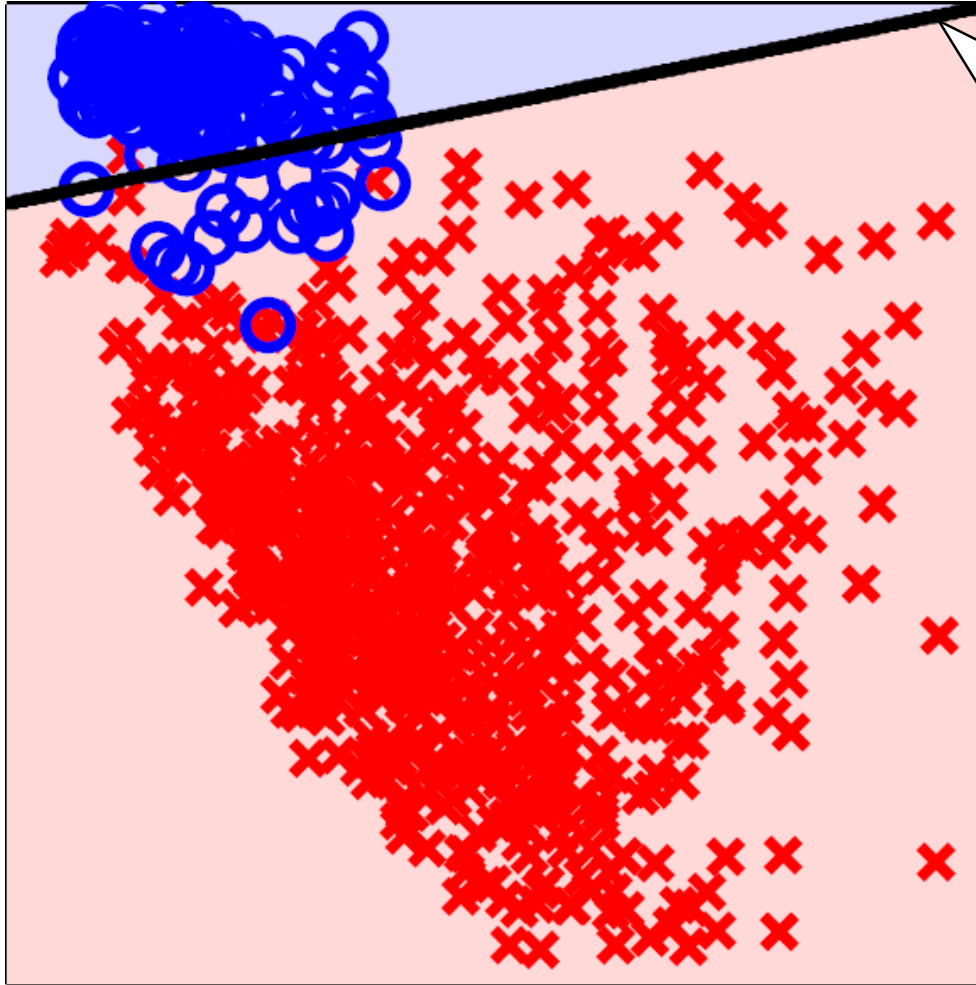
Pocket



Može li linearna regresija pomoći?

- Ukoliko su podaci linearno separabilni, PLA će vremenom konvergirati
- Ali, treba nam dosta iteracija...
- Linearna regresija nije dobar izbor za klasifikacione probleme
- Ali možda nam može pomoći pri inicijalizaciji
- Inicijalizacija: primenimo linearnu regresiju da odredimo θ

Može li linearna regresija pomoći?



Granica odluke
dobijena pomoću
linearne regresije

Perceptron – zaključak

- U 1960-im perceptron je uveden kao grub model rada individualnih neurona u mozgu
- Iako je perceptron „kozmetički“ sličan logističkoj regresiji i linearnoj regresiji, on je ustvari veoma različit
- Krucijalna razlika je teškoća da za predikcije perceptrona navede probabilistička interpretacija ili da se perceptron algoritam izvede putem metoda maksimalne verodostojnosti
- Međutim, i danas je u upotrebi