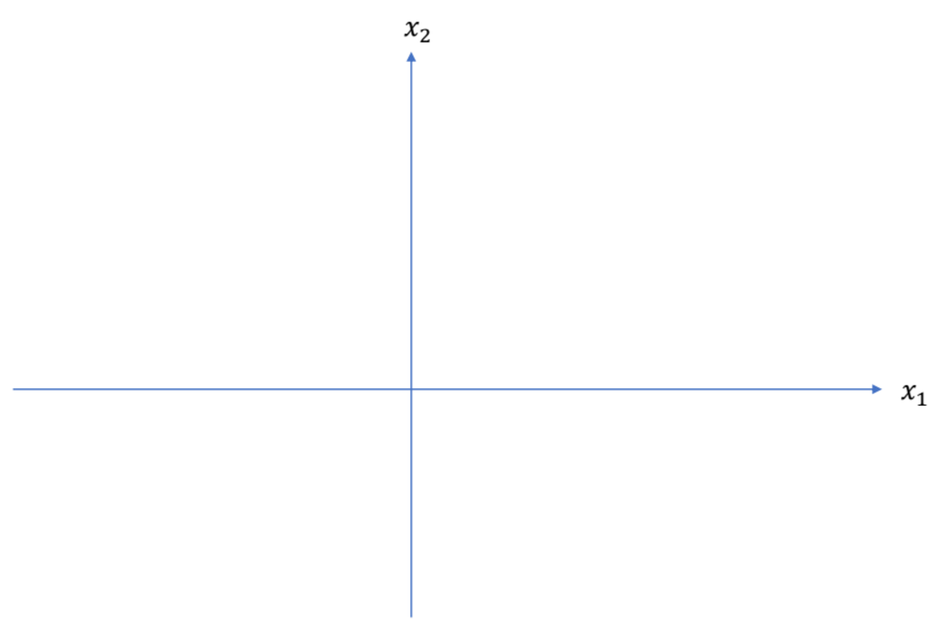
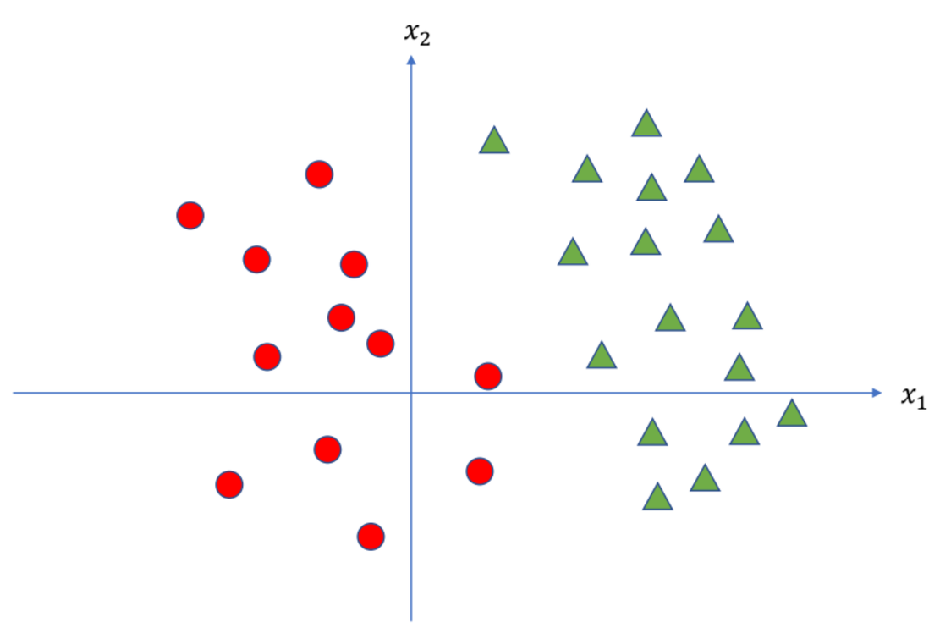
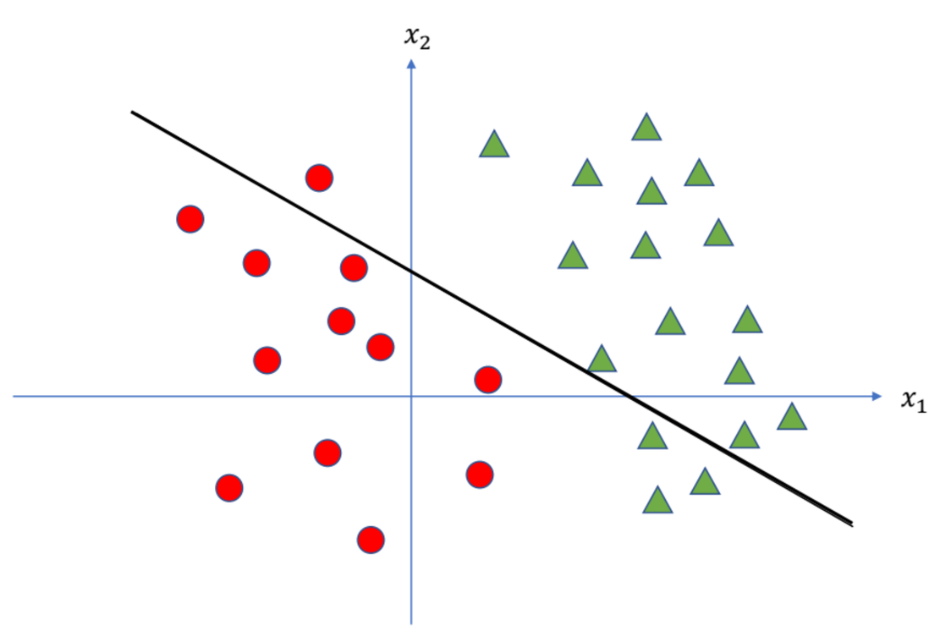
Perceptron 1.pdf

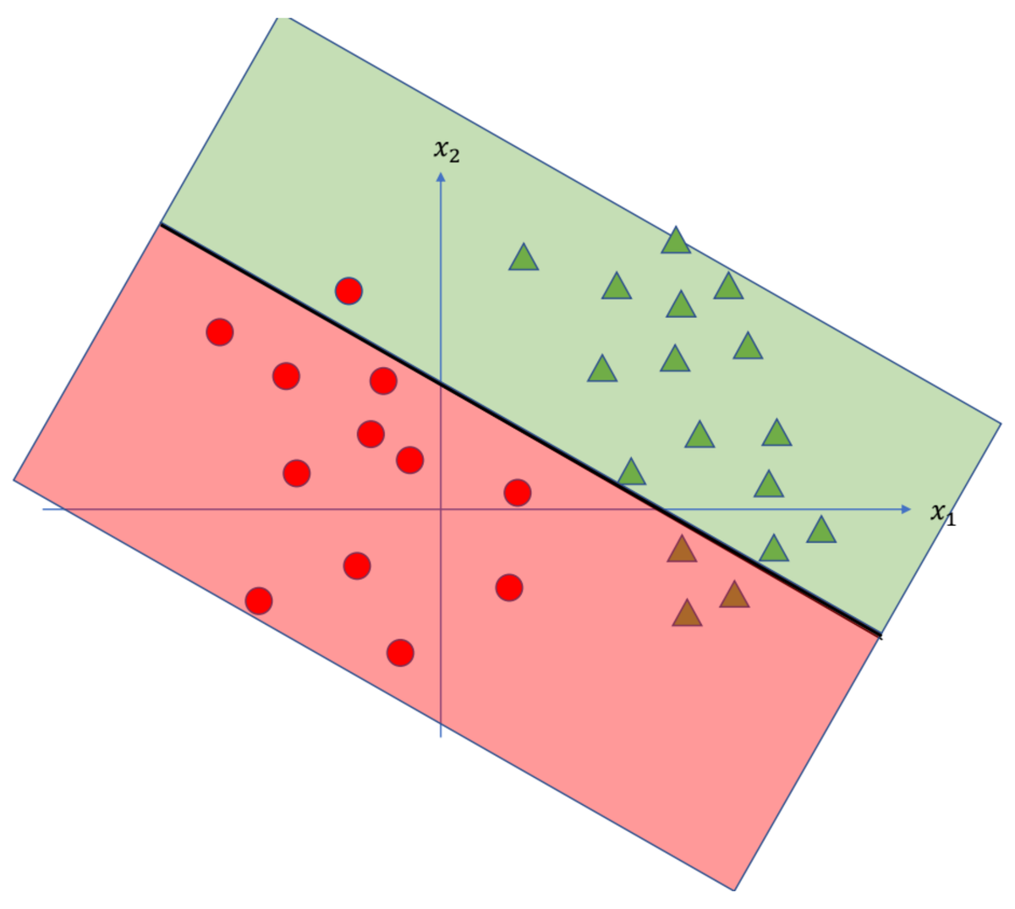
terça-feira, 4 de julho de 2023 15:05

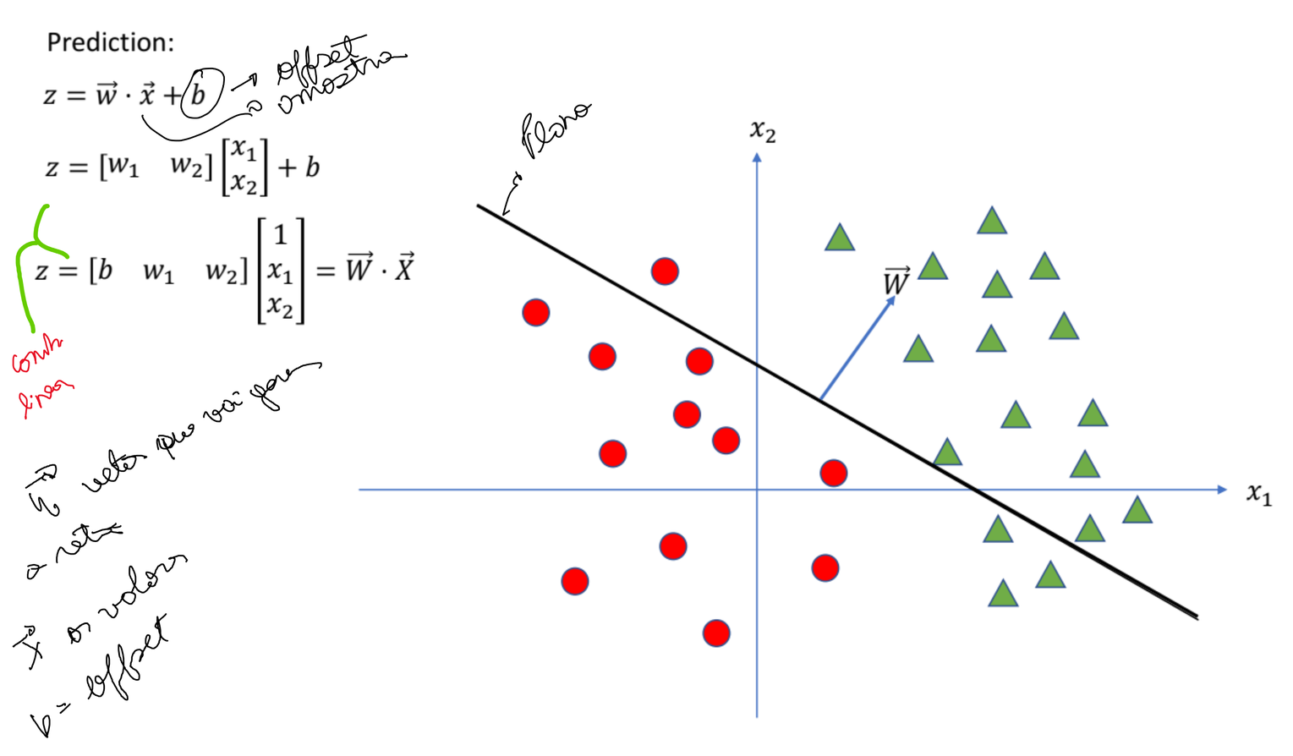
Perceptron 1.pdf











offset amostra

combinação linear

Plano

$\vec{W}$ vetor que vai girar a reta

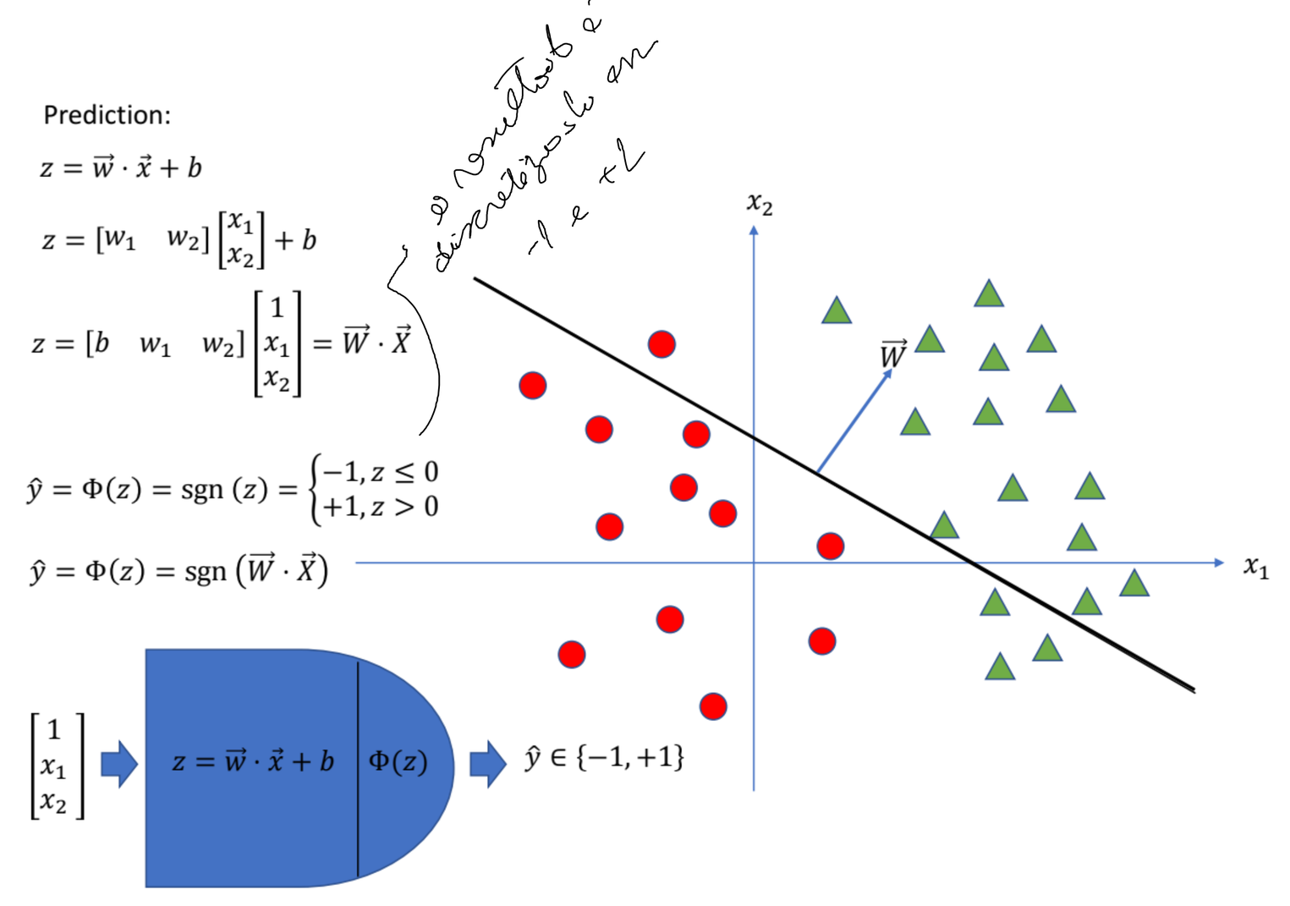
$\vec{X}$ os valores

b = offset

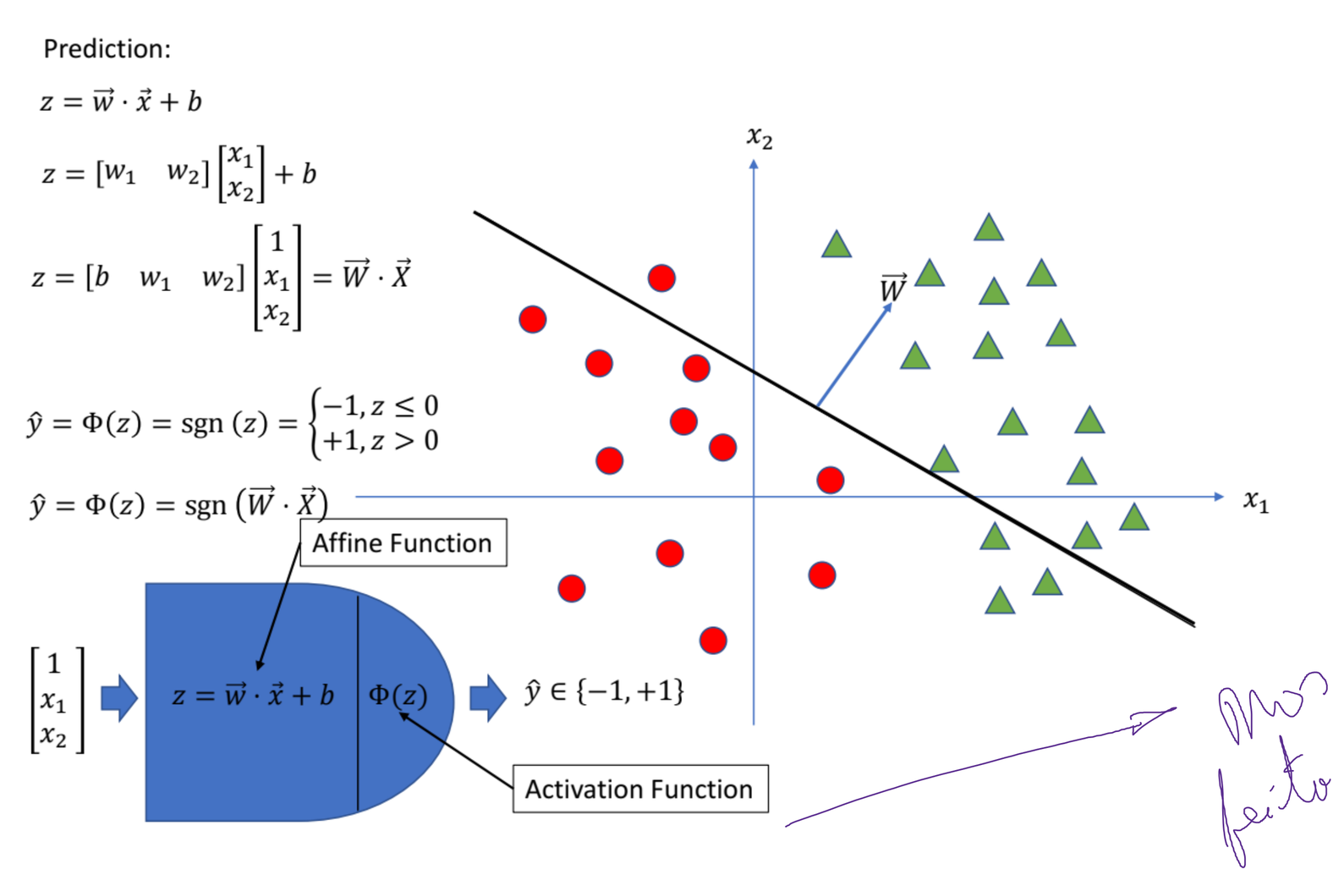
"linha", no que nosso contexto é um plano, mas a ideia é algo com

$y = mx + b$

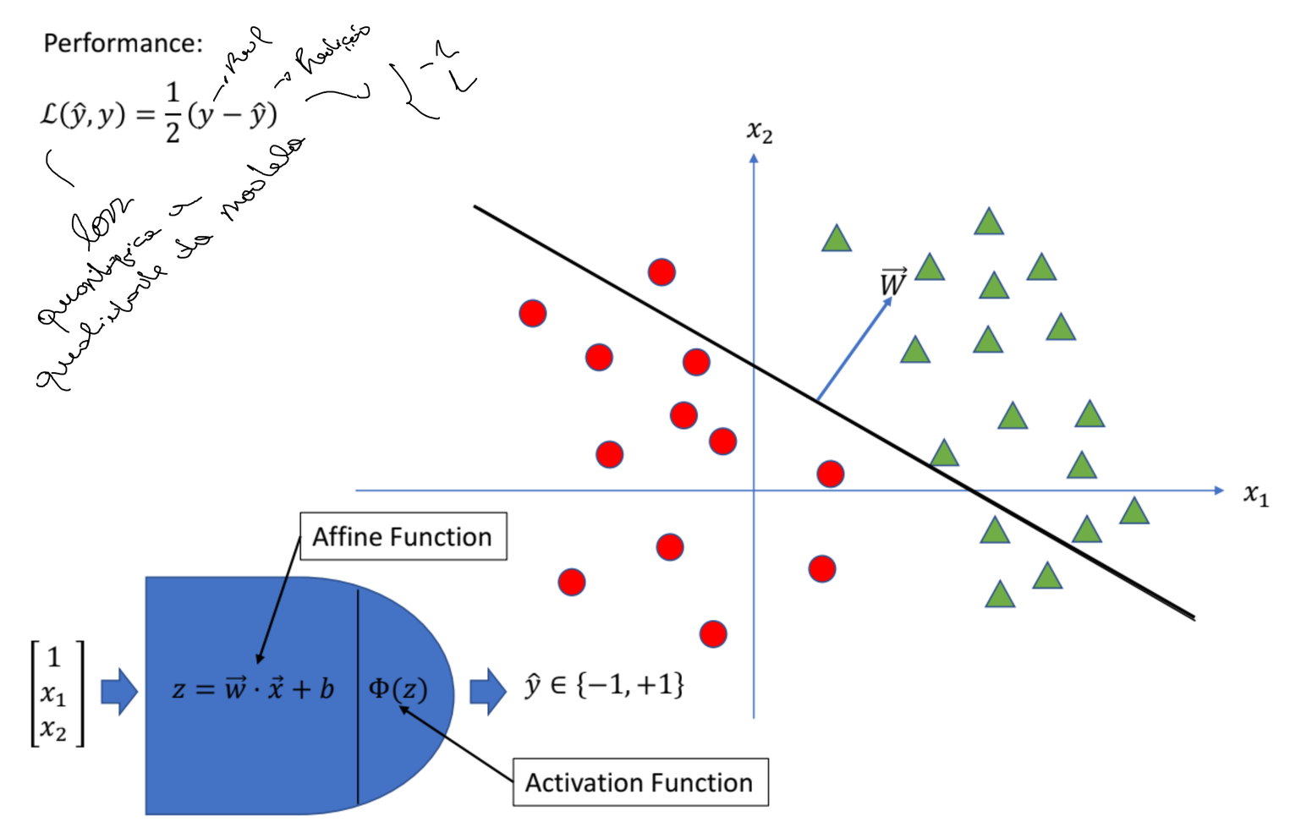
b → inclinação



o resultado é discretizado em -1 e +1

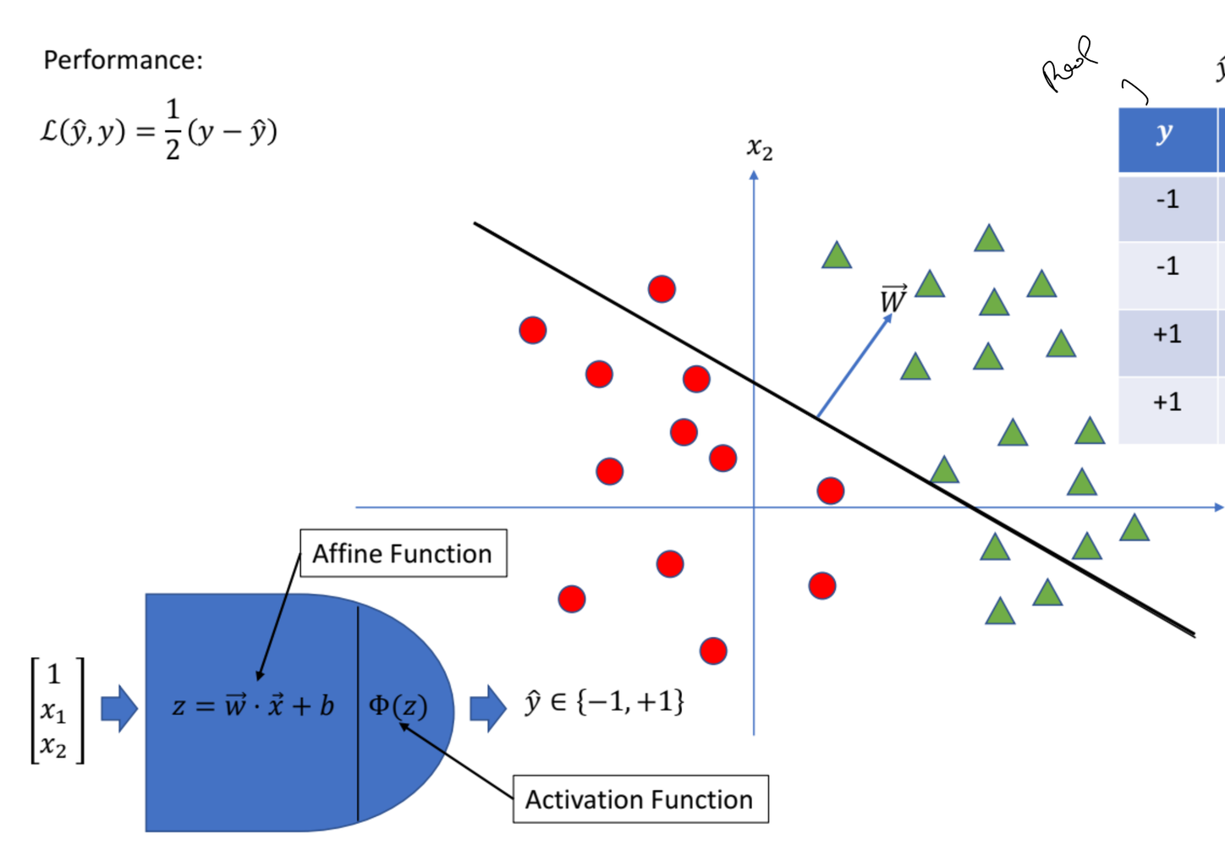


Mas como isso é feito no cálculo?



Real → Predição ~ {-1, 1}

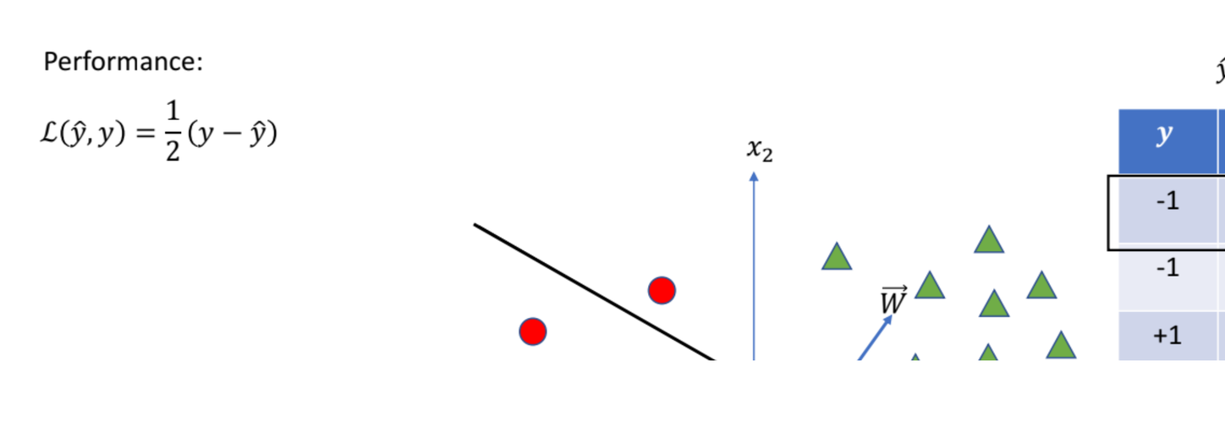
loss quantifica a qualidade do modelo



Real, Predição, erro

$\hat{y} = \text{sgn}(\vec{W} \cdot \vec{X})$

y	ŷ	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0



$\hat{y} = \text{sgn}(\vec{W} \cdot \vec{X})$

y	ŷ	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1

Perceptrons

Entrada: Vetor de números

Processo: calcular a combinação linear

Resultado: { 1 se o resultado for maior que um threshold; -1 caso contrário

→ cada entrada vai ter um peso associado

→ A partir das entradas, achar os pesos que se adequem a produzir o resultado ±1

↳ São algoritmos que irão fazer isso: Perceptron Rule, Delta Rule

→ A ideia de geração aleatória é com ela pegar algo, dar maior ou menor que o threshold? e a partir daí ir regulando seguindo um algoritmo

→ Uma opção é gerar pesos aleatórios que vai mudando de acordo com a regra de treinamento

↓

$w_i = w_i + \Delta w_i$

onde $\Delta w_i = \eta (t - o) x_i$ → input

taxa de aprendizado; output esperado; output gerado

→ A ideia é calcular novos pesos w baseado nessa equação, onde a taxa de aprendizado vai dizer quão "rápido" serão as mudanças

Ex.: Perceptron = -1 → logo, tem que reavaliar os pesos

Target = 1 → $\Delta w_i = 0.1 (1-(-1)) . 0.8 = 0.16$

$\eta = 0.1$

$x_i = 0.8$

→ Os casos acima são para linearmente separáveis

→ Delta Rule

↳ Tentar resolver os não lineares

↳ Gradient Descent

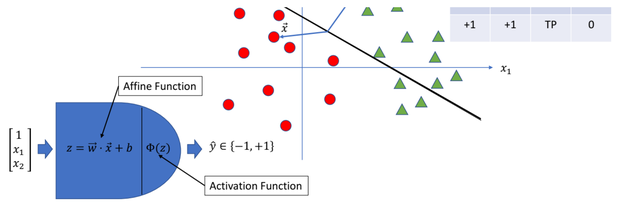
Partimos de uma unidade de perceptron que:

$o(\vec{x}) = \vec{w} \cdot \vec{x}$ → sem o threshold

Temos que o algoritmo vai ser baseado na taxa de erro:

$E(\vec{w}) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$

todos os training examples

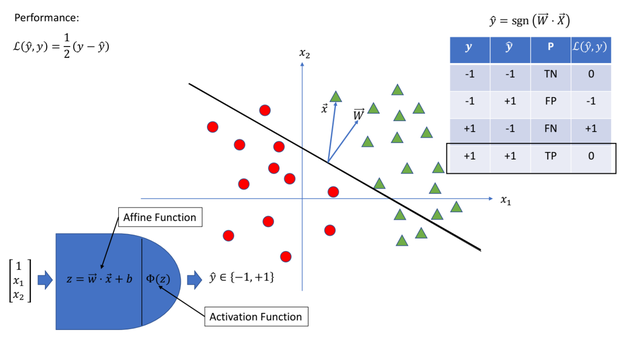


+1	+1	TP	0

Performance:

$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$

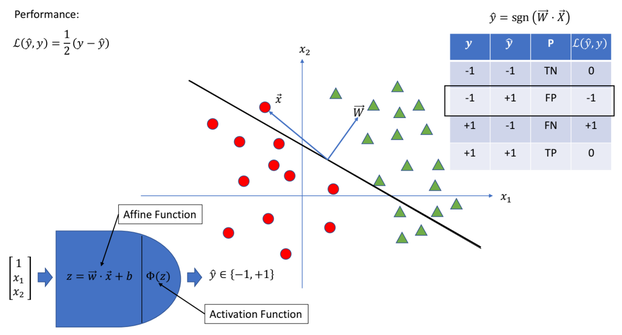


y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$

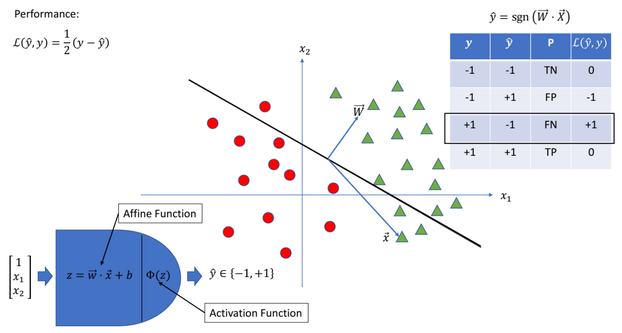


y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$



y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

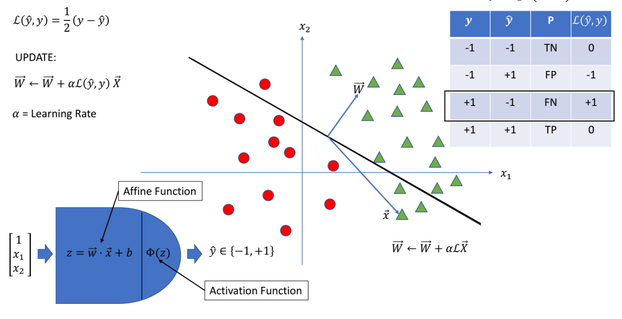
$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

UPDATE:

$\vec{W} \leftarrow \vec{W} + \alpha \mathcal{L}(\hat{y}, y)\, \vec{X}$

α = Learning Rate

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$



y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

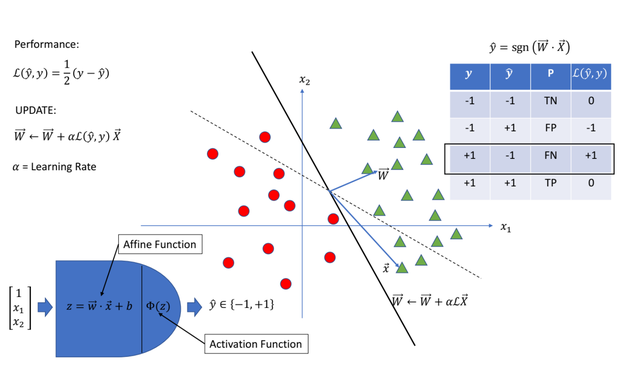
$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

UPDATE:

$\vec{W} \leftarrow \vec{W} + \alpha \mathcal{L}(\hat{y}, y)\, \vec{X}$

α = Learning Rate

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$



y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

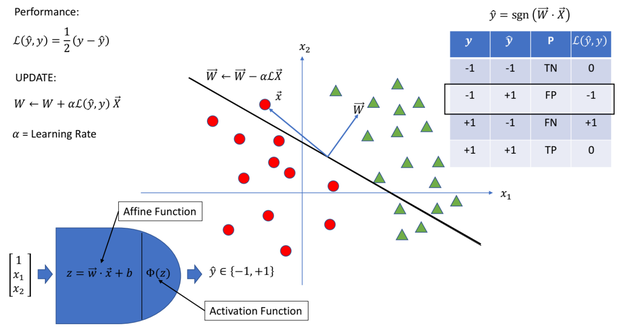
$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

UPDATE:

$W \leftarrow W + \alpha \mathcal{L}(\hat{y}, y)\, \vec{X}$

α = Learning Rate

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$



y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0

Performance:

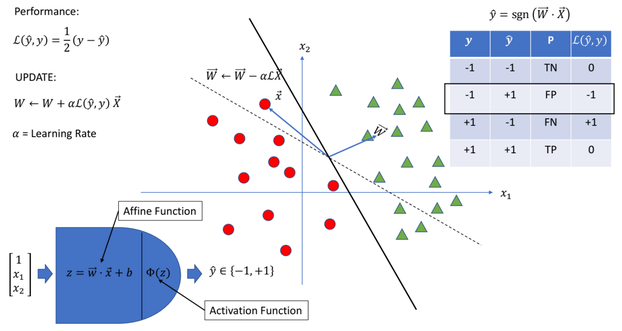
$\mathcal{L}(\hat{y}, y) = \frac{1}{2}(y - \hat{y})$

UPDATE:

$W \leftarrow W + \alpha \mathcal{L}(\hat{y}, y)\, \vec{X}$

α = Learning Rate

$\hat{y} = \text{sgn}\left(\vec{W} \cdot \vec{X}\right)$



y	$\hat{y}$	P	$\mathcal{L}(\hat{y}, y)$
-1	-1	TN	0
-1	+1	FP	-1
+1	-1	FN	+1
+1	+1	TP	0