UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Redes Perceptron de Uma Camada

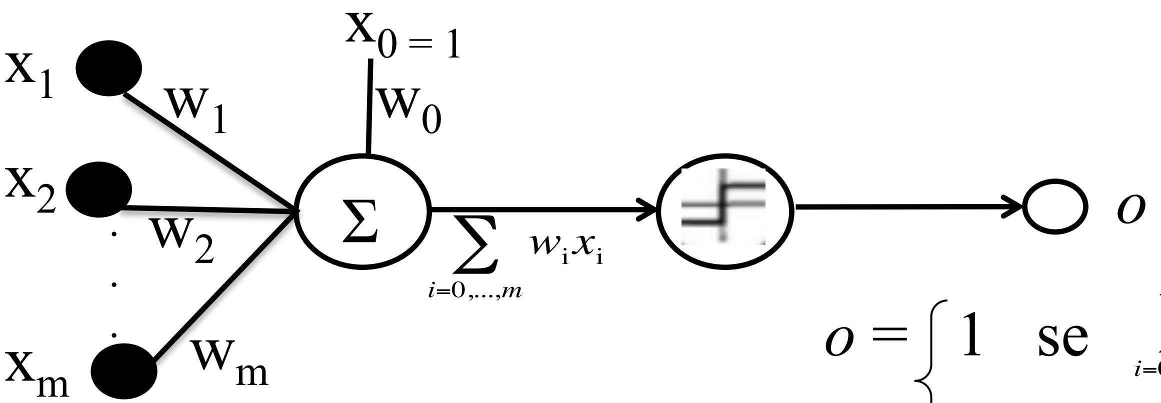
Gisele L. Pappa

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Perceptron de uma Camada

- Primeiro modelo para aprendizado supervisionado
- Padrões linearmente separáveis



$$o = \begin{cases} 1 & \text{se} \quad \sum_{i=0,\dots,m} w_i x_i > 0 \\ -1 & \text{caso contrário} \end{cases}$$

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Aprendizado Supervisionado

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Treinamento

Entrada / Saída

A_1	A_2	A_3	A_4	C
0	0	1	0	$\boxed{1}$
1	1	0	1	$\boxed{1}$
0	1	0	0	$\boxed{0}$
0	0	0	1	$\boxed{0}$

Rede Neural Artificial

Teste

1,0,1,0,?
0,0,0,1,?
1,1,0,0,?
0,1,1,1,?

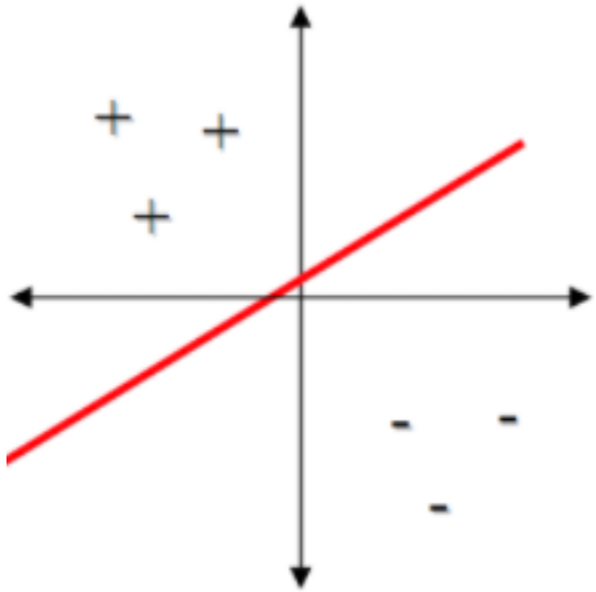
Rede Neural Artificial

1,0,1,0,1
0,0,0,1,0
1,1,0,0,1
0,1,1,1,1

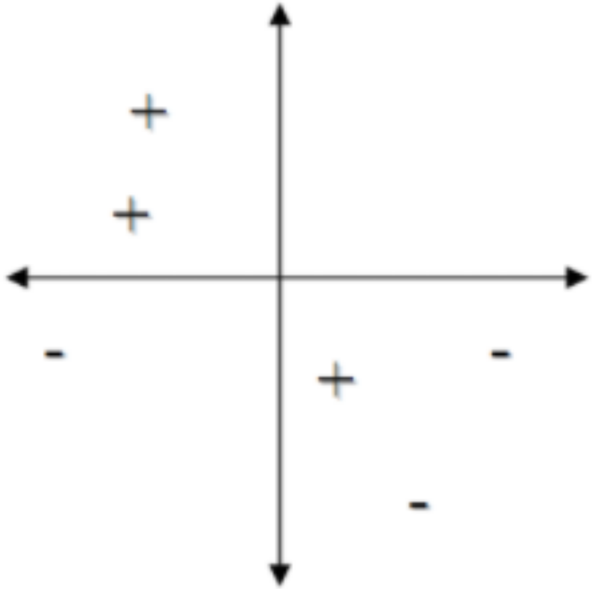
• Capacidade de Generalização da rede

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Padrões de Dados



Linearmente separável



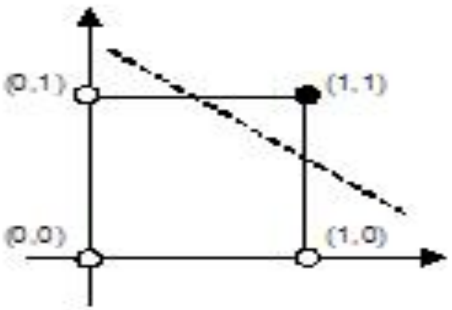
Não-linearmente separável

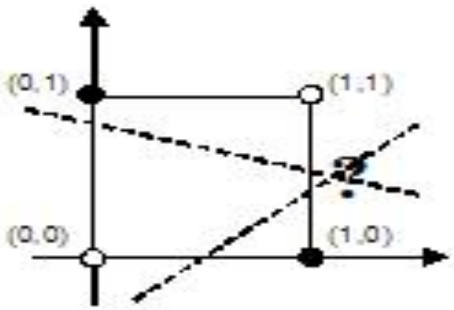
Perceptron de uma Camada

- Primeiro modelo para aprendizado supervisionado
- Padrões linearmente separáveis

Inputs		Output
x_1	x_2	x_1 AND x_2
0	0	0
0	1	0
1	0	0
1	1	1

Inputs		Output
x_1	x_2	x_1 XOR x_2
0	0	0
0	1	1
1	0	1
1	1	0





Aprendizado

- Seja w_i um peso sináptico de um dado neurônio.
- O ajuste Δw_i é aplicado ao peso sináptico w_i gerando o valor corrigido w_i, na forma:

$$w_i = w_i + \Delta w_i$$

- Várias maneira de obter $\Delta w(t)$:
  - regra de Hebb, regra do perceptron, regra Delta, algoritmo de *backpropagation*, estratégias de competição, máquina de Boltzmann

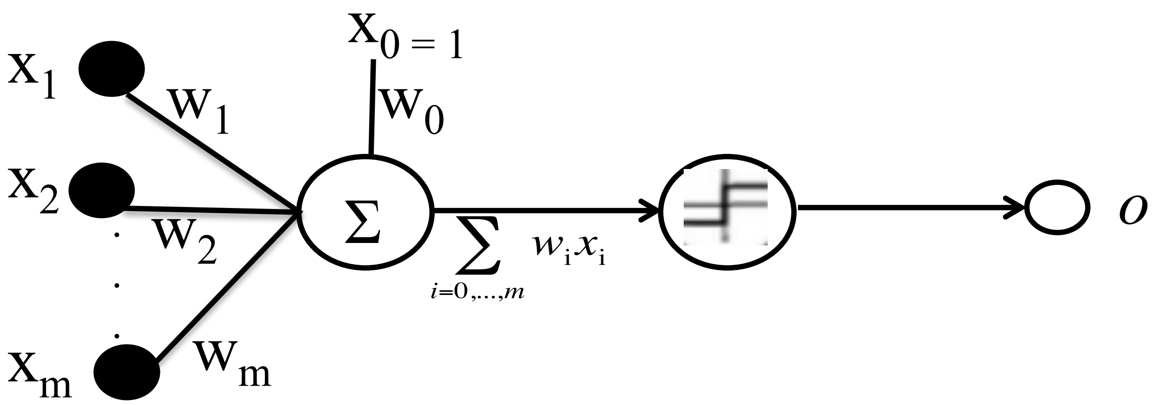
UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Regra do Perceptron

$$\overbrace{x_1,\ x_2,\ \ldots,\ x_m}^{\text{Entrada}},\quad \overset{\text{Saída}}{y}$$

0,	0,	…,	0,	1
1,	1,	…,	1,	1

Conjunto de treinamento



$$w_i = w_i + \Delta w_i$$

$$\Delta w_i = \eta\ (y\text{-}o)\ x_i$$

η é a taxa de aprendizado

– No perceptron de uma camada, a taxa de aprendizado têm pouco impacto e pode ser igual a 1

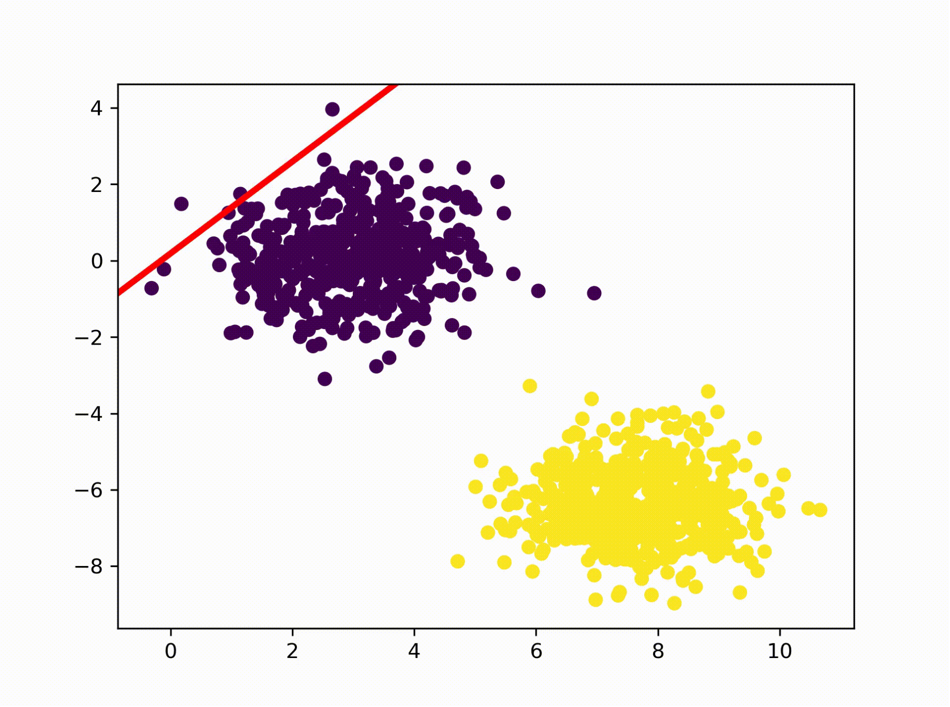
DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Intuição da Regra do Perceptron

- Dada uma instância (x,y)
  - Se o erro é positivo (saída real é maior que a retornada)
    - Quero aumentar $w_k x_k$
  - Se o erro é negativo (saída real é menor que a retornada)
    - Quero diminuir $w_k x_k$
- Se não exite erro, não muda pesos

UFMG UNIVERSIDADE FEDERAL DE MINAS GERAIS
DCC DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Aprendizado do Perceptron



Fonte: https://towardsdatascience.com/from-biology-to-ai-the-perceptron-81abfdc788bf

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS
DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Propriedades da regra do Perceptron

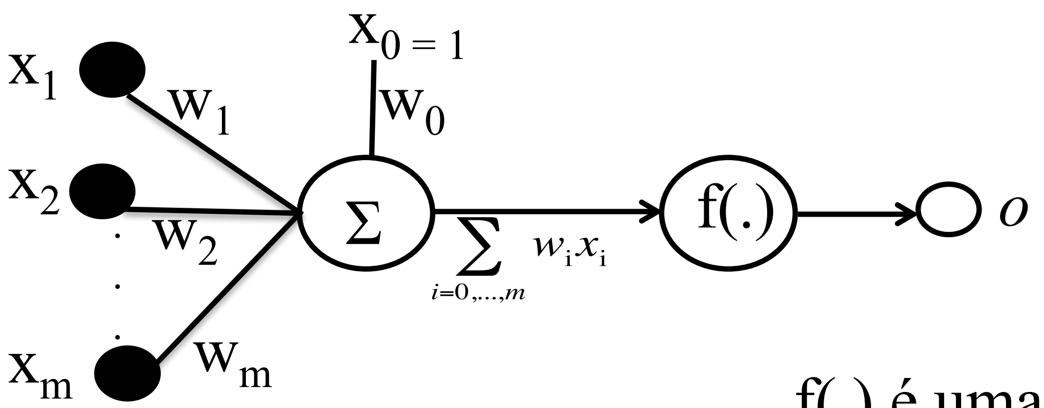
- Garante convergência quando os pontos são linearmente separáveis
- Não garante convergência para mínimo local quando pontos **não** são linearmente separáveis
- Pontos classificados corretamente não influenciam no treino

Regra Delta

- Como a regra do perceptron não converge em casos de exemplos não-linearmente separáveis, criou-se a regra delta
- A regra delta converge para a melhor aproximação da saída quando os exemplos não são linearmente separáveis
- Ela usa a descida do gradiente para buscar os pesos.

Perceptron

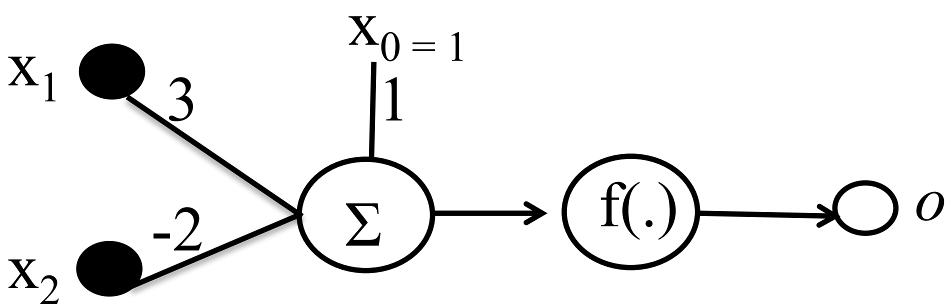
- Para entender a regra, ao invés de usar a saída usando o perceptron com o limiar, vamos considerar um perceptron usando uma função linear



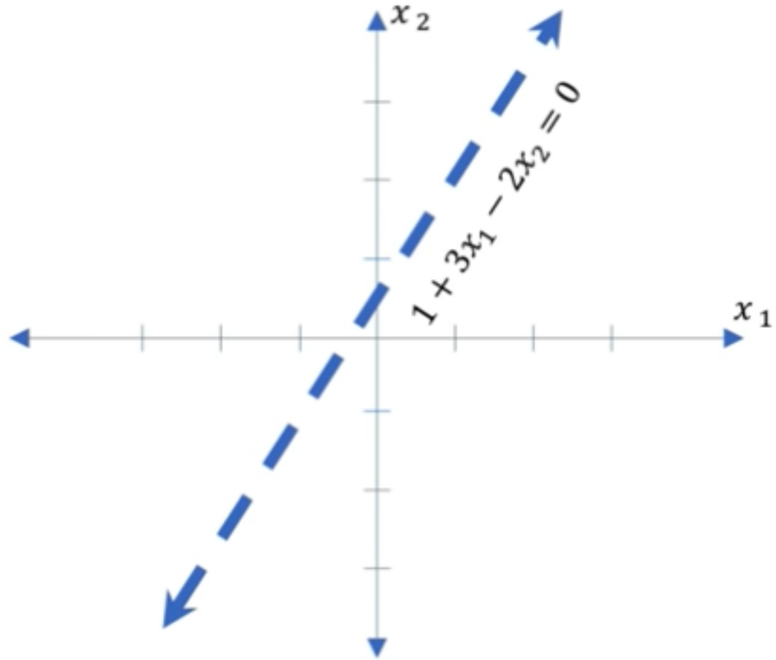
f(.) é uma função linear

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Perceptron



$$o = f(3x_1 - 2x_2 + 1)$$

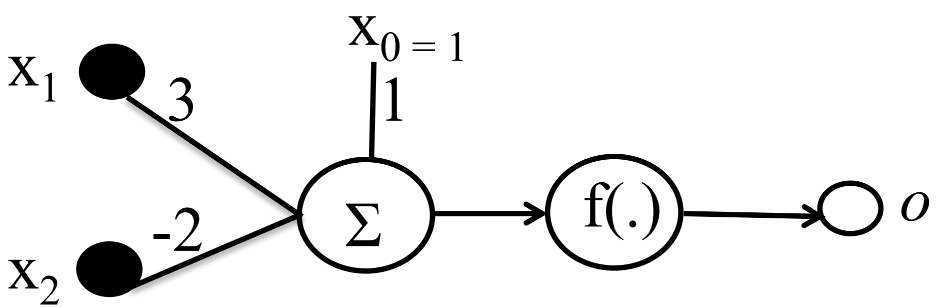


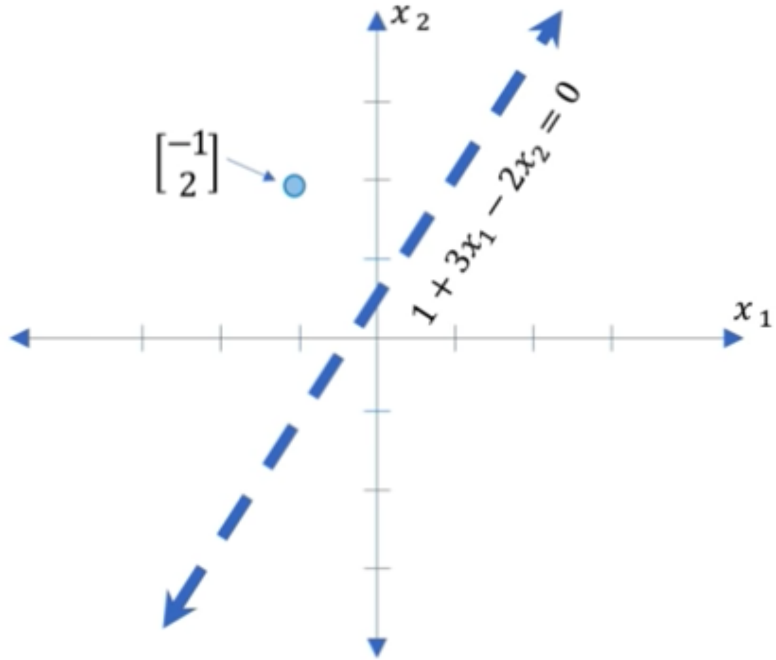
DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

introtodeeplearning.com

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Perceptron





Entrada: -1, 2

$$o = f(3x_1 - 2x_2 + 1)$$
$$= f((3* -1) - (2*2) + 1$$
$$= f(-6) = -6$$

$\boxed{\text{Considerando } f(x) = x}$

introtodeeplearning.com

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Aprendizado do Perceptron

- Consiste em mudar os pesos da rede de forma a reduzir o erro entre a predição da rede e a saída real.

- Erro pode ser calculado utilizando diversas funções de erro/ou de *loss*

Aprendizado do Perceptron

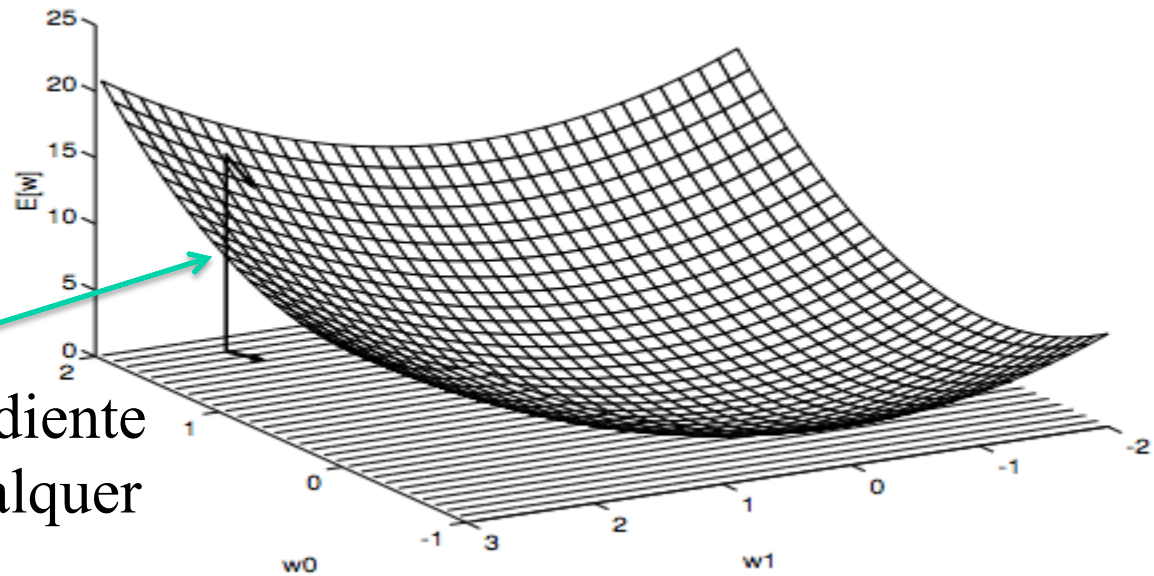
- Vamos utilizar o erro quadrático médio

$$\boxed{E(\overrightarrow{w}) = \frac{1}{2}\sum_{d \in D}(y_d - o_d)^2}$$

onde D é o número de exemplos de treinamento

Aprendizado do Perceptron

U F m G
UNIVERSIDADE FEDERAL
DE MINAS GERAIS



Negativa do gradiente
de um ponto qualquer

Superfície de erro para diferentes conjuntos de pesos w_0 e w_1

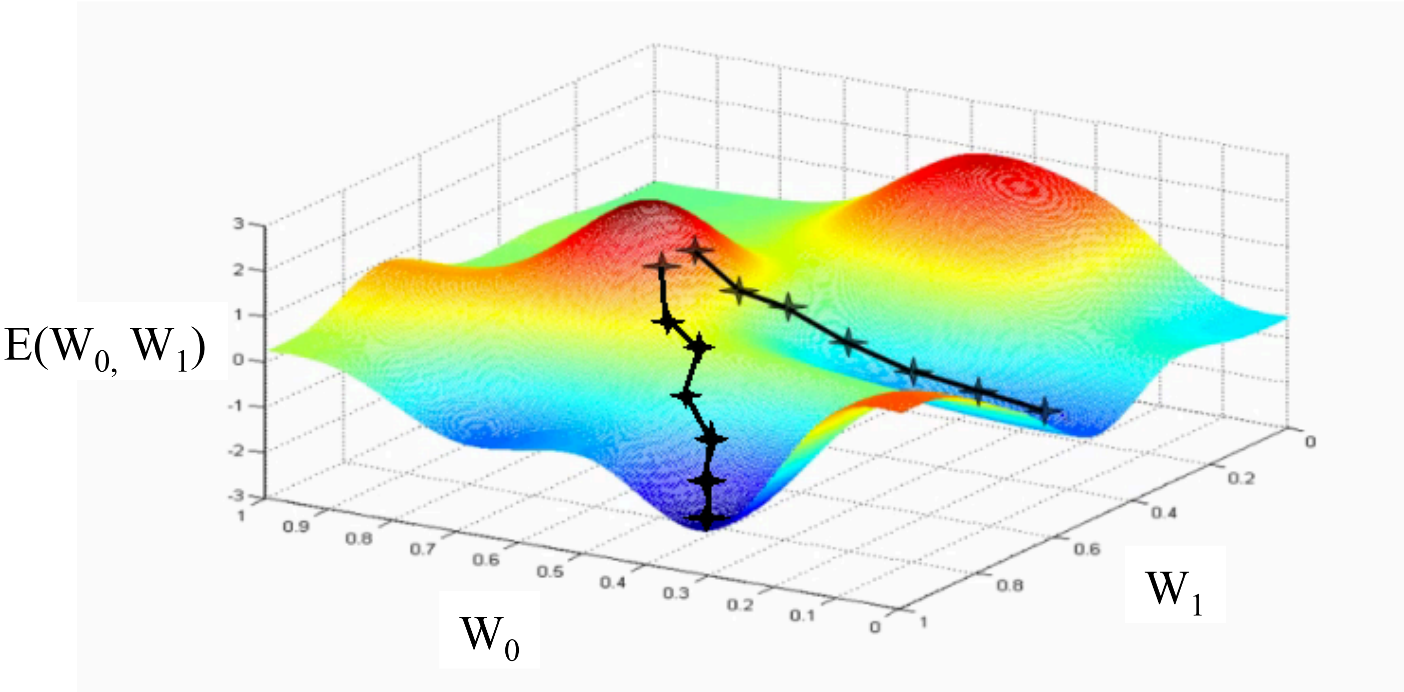
DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Aprendizado do Perceptron

- Usa o algoritmo de descida do gradiente
  - Algoritmo de otimização utilizado para encontrar os parâmetros capazes de minimizar uma função.
- **Gradiente** de uma função f(x,y) no ponto (x_0,y_0) ($\nabla f(x_0,y_0)$):
  - Para um dado ponto (x_0,y_0), o gradiente fornece a direção de maior crescimento de f(x)
    - É um vetor cujos componentes são as derivadas parciais de f(x,y)
- Descida do gradiente vai na "direção contrária" do gradiente

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Aprendizado do Perceptron



https://www.analyticsvidhya.com/blog/2017/03/introduction-to-gradient-descent-algorithm-along-its-variants/

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Descida do Gradiente

- Inicialize os pesos da rede aleatoriamente
- Enquanto (não converge)
  - Compute o gradiente da função de erro considerando os pesos
  - Atualize os pesos usando o negativo do gradiente
- Retorne os pesos

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Regra Delta

$$\nabla E[\vec{w}] \equiv \left[\frac{\partial E}{\partial w_0}, \frac{\partial E}{\partial w_1}, \cdots \frac{\partial E}{\partial w_n}\right]$$

$$\Delta \vec{w} = -\eta \nabla E[\vec{w}]$$

$$\Delta w_i = -\eta \frac{\partial E}{\partial w_i}$$

$$\boxed{\frac{\partial E}{\partial w_i} = \sum_d (t_d - o_d)(-x_{i,d})}$$

Regra Delta

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

- Como os gradientes são obtidos?

$X_{i,d}$ é a entrada *i* para o exemplo *d*

$$
\begin{aligned}
\frac{\partial E}{\partial w_i} &= \frac{\partial}{\partial w_i}\frac{1}{2}\sum_d (t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d \frac{\partial}{\partial w_i}(t_d - o_d)^2 \\
&= \frac{1}{2}\sum_d 2(t_d - o_d)\frac{\partial}{\partial w_i}(t_d - o_d) \\
&= \sum_d (t_d - o_d)\frac{\partial}{\partial w_i}(t_d - \vec{w}\cdot\vec{x_d}) \\
\frac{\partial E}{\partial w_i} &= \sum_d (t_d - o_d)(-x_{i,d})
\end{aligned}
$$

Lembre-se que:

$y = u^2$: $\frac{dy}{dx} = \frac{dy}{du}\frac{du}{dx}$ with $u = t_d - o_d$

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Regra Delta

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

- O algoritmo de descida do gradiente (GD) mostrado otimiza o erro considerando todos os exemplos (*batch)*
- Existe uma versão estocástica do GD que atualiza os pesos da rede após cada exemplo ser apresentado (*stochastic* GD)
- *Mini-batch* GD – usa uma amostra pequena dos dados de treino por época

UFMG
UNIVERSIDADE FEDERAL DE MINAS GERAIS

Esquemas de treinamento

- Batch gradient descent: usa todos os exemplos de treinamento a cada iteração.
- Stochastic gradient descent: usa um exemplo de treinamento a cada iteração
- Mini-batch gradient descent: usa *b* exemplos de treinamento a cada iteração, onde *b* normalmente varia entre 2 e 100 e é potência de 2
  - Vantagem: é fácil de paralelizar, tornando o aprendizado mais rápido que o stochastic

DCC
DEPARTAMENTO DE CIÊNCIA DA COMPUTAÇÃO

Treinamento do Perceptron

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

- Diferentes conjuntos iniciais de pesos para o perceptron podem levar a diferentes superfícies de decisão.
  - O problema de ajuste supervisionado de pesos pode ser visto como um processo de busca por um conjunto de pesos que otimizam uma determinada superfície de erro.
  - Uma escolha inadequada da condição inicial da rede pode levar o algoritmo a uma convergência para ótimos locais desta superfície de erro.

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

Treinamento do Perceptron

- **Parâmetros de treinamento**
  - Taxa de aprendizado
- **Treinamento versus aplicação da rede**
  - Diferenciar entre o processo de treinamento e aplicação da rede.
  - O treinamento da rede corresponde ao processo de ajuste de pesos.
  - Após treinada, verificar a qualidade do aprendizado para verificar sua *capacidade de generalização*.

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Redes Perceptron de Uma Camada

Gisele L. Pappa

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO

UFMG
UNIVERSIDADE FEDERAL
DE MINAS GERAIS

Leitura Recomendada

- A Brief Introduction to Neural Networks, http://www.dkriesel.com/en/science/neural_networks, Parte 1

DCC
DEPARTAMENTO DE
CIÊNCIA DA COMPUTAÇÃO