Machine Learning

Parte 3

Evandro J.R. Silva

ejrs.profissional@gmail.com

Bacharelado em Ciência da Computação
Faculdade Estácio Teresina

23 de julho de 2022

Sumário

Principais Algoritmos

Principais Algoritmos

- Aprendizado Supervisionado
  - Classificação
    - Naive Bayes
    - k-NN
    - Árvore de Decisão
    - Redes Neurais Artificiais
  - Regressão
    - Regressão Logística
- Aprendizado Não Supervisionado
  - k-Means

Aprendizado Supervisionado

Classificação

Redes Neurais Artificiais

Redes Neurais Artificiais

O que são Redes Neurais Artificiais?

"Redes Neurais Artificiais (RNAs) são sistemas paralelos distribuídos compostos por unidades de processamento simples (nodos) que calculam determinadas funções matemáticas (normalmente não-lineares). Tais unidades são dispostas em uma ou mais camadas interligadas por um grande número de conexões, geralmente unidirecionais. Na maioria dos modelos estas conexões estão associadas a pesoas, os quais armazenam o conhecimento representado no modelo e servem para ponderar a entrada recebida por cada neurônio da rede. O funcionamento destas redes é inspirado em uma estrutura física concebida pela natureza: o cérebro humano."[1]

Redes Neurais Artificiais

- Existem vários tipos de RNAs, e vamos ver duas básicas e comuns: Perceptron e MLP (*Multi Layer Perceptron*).

Redes Neurais Artificiais

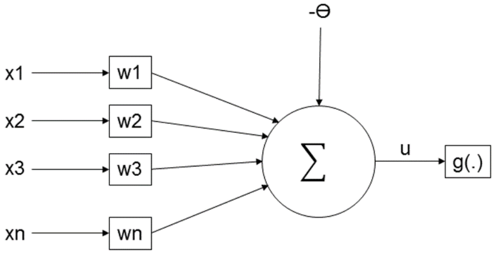
- Existem vários tipos de RNAs, e vamos ver duas básicas e comuns: Perceptron e MLP (*Multi Layer Perceptron*).
- Não veremos as outras por questão de tempo e também porque estamos em um minicurso introdutório.

Redes Neurais Artificiais

- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.

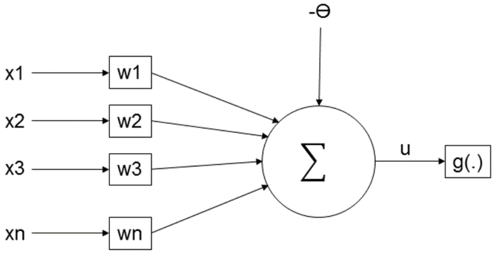
Redes Neurais Artificiais

- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



Redes Neurais Artificiais

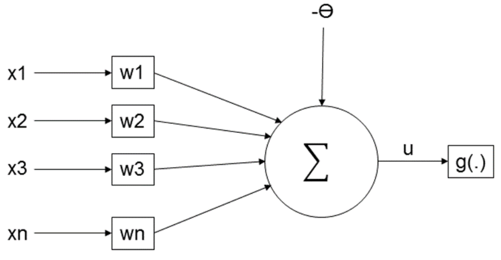
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



    - Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;

Redes Neurais Artificiais

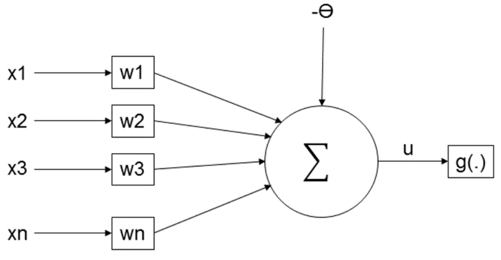
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



    - Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
    - Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;

Redes Neurais Artificiais

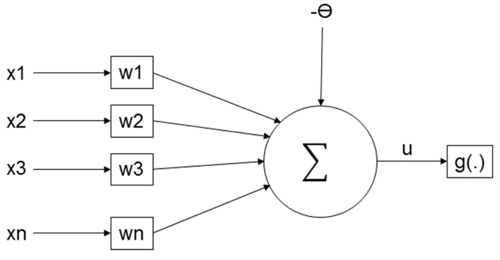
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



    - Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
    - Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;
    - Combinador linear: Σ;

Redes Neurais Artificiais

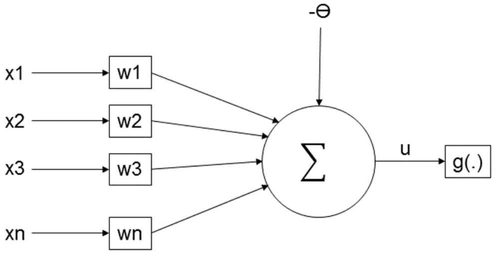
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



    - Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
    - Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;
    - Combinador linear: Σ;
    - Limiar de ativação: θ;

Redes Neurais Artificiais

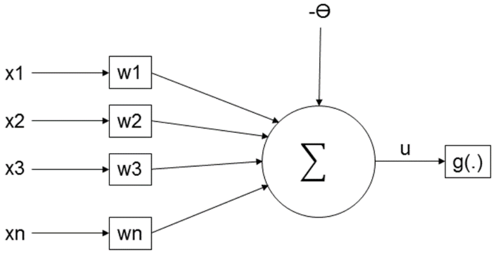
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



- Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
- Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;
- Combinador linear: Σ;
- Limiar de ativação: θ;
- Potencial de ativação: $u = \sum_{i=1}^{n} w_i \cdot x_i - \theta$;

Redes Neurais Artificiais

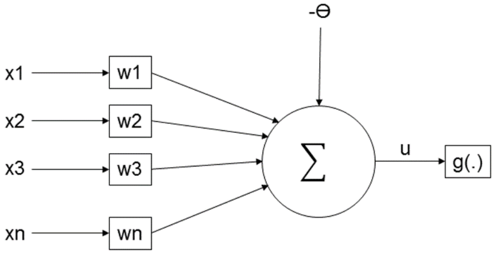
- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



- Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
- Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;
- Combinador linear: Σ;
- Limiar de ativação: θ;
- Potencial de ativação: $u = \sum_{i=1}^{n} w_i \cdot x_i - \theta$;
- Função de ativação: $g(.)$;

Redes Neurais Artificiais

- Perceptron
  - O Perceptron consiste em apenas um único neurônio artificial, o qual recebe várias entradas, é capaz de processá-las e retorna uma saída.



- Sinais de entrada: $\{x_1, x_2, x_3, ..., x_n\}$;
- Pesos sinápticos: $\{w_1, w_2, w_3, ..., w_n\}$;
- Combinador linear: Σ;
- Limiar de ativação: θ;
- Potencial de ativação: $u = \sum_{i=1}^{n} w_i \cdot x_i - \theta$;
- Função de ativação: $g(.)$;
- Sinal de saída: $y = g(u)$;

Redes Neurais Artificiais

- Perceptron
  - Funcionamento básico:

Redes Neurais Artificiais

- Perceptron
  - Funcionamento básico:
    1. Cada entrada é um valor. Multiplique cada entrada pelo peso correspondente;

Redes Neurais Artificiais

- Perceptron
  - Funcionamento básico:
    1. Cada entrada é um valor. Multiplique cada entrada pelo peso correspondente;
    2. Faça a combinação de todas as entradas (já multiplicadas com os pesos) com o limiar de ativação. A combinação é a soma de todos os valores.

Redes Neurais Artificiais

- Perceptron
  - Funcionamento básico:
    1. Cada entrada é um valor. Multiplique cada entrada pelo peso correspondente;
    2. Faça a combinação de todas as entradas (já multiplicadas com os pesos) com o limiar de ativação. A combinação é a soma de todos os valores.
    3. Execute a função de ativação. Essa função tem como entrada a combinação das entradas e limiar de ativação. A partir do valor recebido a função vai retornar outro valor. Por exemplo: se o valor for ≥ 0 a função retorna 1, senão, retorna 0. O retorno dessa função é justamente a saída y.

Redes Neurais Artificiais

- Perceptron
  - Funcionamento básico:
    1. Cada entrada é um valor. Multiplique cada entrada pelo peso correspondente;
    2. Faça a combinação de todas as entradas (já multiplicadas com os pesos) com o limiar de ativação. A combinação é a soma de todos os valores.
    3. Execute a função de ativação. Essa função tem como entrada a combinação das entradas e limiar de ativação. A partir do valor recebido a função vai retornar outro valor. Por exemplo: se o valor for ≥ 0 a função retorna 1, senão, retorna 0. O retorno dessa função é justamente a saída y.
    4. Se o aprendizado for supervisionado e a saída estiver errada, ajuste os pesos de acordo com alguma função escolhida.

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Dada uma rede do tipo Perceptron formada por um neurônio com três terminais de entrada, utilizado os pesos iniciais $w_0 = 0,4$, $w_1 = -0,6$ e $w_2 = 0,6$, limiar $\theta = 0,5$ e uma taxa de aprendizado $\eta = 0,4$. Responda os itens abaixo:
  1. Ensinar a rede a gerar a saída -1 para o padrão 001 e a saída +1 para o padrão 110;
  2. A que classe pertencem os padrões 111, 000, 100, 011?

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Perceba que são três entradas: **0**, **0** e **1**. Ou **1**, **1** e **0**.

Redes Neurais Artificiais

- Perceptron

Exemplo

- Perceba que são três entradas: **0**, **0** e **1**. Ou **1**, **1** e **0**.
- Lembre também que o limiar de ativação é multiplicado por -1. Ou seja, podemos considerar que θ também é um peso associado.

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Perceba que são três entradas: **0**, **0** e **1**. Ou **1**, **1** e **0**.
- Lembre também que o limiar de ativação é multiplicado por -1. Ou seja, podemos considerar que θ também é um peso associado.
- A função de ativação $g(u)$ que utilizaremos vai retornar +1 se a combinação for $\geq$ 0, e -1 se a combinação $<$ 0.

Redes Neurais Artificiais

- Perceptron

Exemplo

- Perceba que são três entradas: **0**, **0** e **1**. Ou **1**, **1** e **0**.
- Lembre também que o limiar de ativação é multiplicado por -1. Ou seja, podemos considerar que θ também é um peso associado.
- A função de ativação $g(u)$ que utilizaremos vai retornar +1 se a combinação for $\geq$ 0, e -1 se a combinação $<$ 0.
- Por fim, temos uma taxa de aprendizado, a qual será utilizada na função de ajuste de pesos.

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$
  1. u = 0(0,4) + 0(-0,6) + 1(0,6) - 1(0,5) = 0,1

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$
  1. u = 0(0,4) + 0(-0,6) + 1(0,6) - 1(0,5) = 0,1

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$
  1. u = 0(0,4) + 0(-0,6) + 1(0,6) - 1(0,5) = 0,1
     $y = g(u) = +1$ (uma vez que 0,1 $\geq$ 0)

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$

1. u = 0(0,4) + 0(-0,6) + 1(0,6) - 1(0,5) = 0,1
   $y = g(u) = +1$ (uma vez que $0{,}1 \geq 0$)
2. Atualização dos pesos: $w_n = w_n + \Delta w_n$
   onde w_n é o peso n e $\Delta w_n =$ taxa de aprendizado $\cdot$ *entrada* $\cdot$ *erro*. E o *erro* $=$ saída desejada $-$ saída real.
   Ou seja: $\Delta w_n = \eta \cdot x_n \cdot (d - y)$.

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **001**. Saída desejada: $d = -1$
  1. u = 0(0,4) + 0(-0,6) + 1(0,6) - 1(0,5) = 0,1
     $y = g(u) = +1$ (uma vez que 0,1 $\geq$ 0)
  2. Atualização dos pesos: $w_n = w_n + \Delta w_n$
     onde w_n é o peso n e $\Delta w_n =$ taxa de aprendizado $\cdot$ *entrada* $\cdot$ *erro*. E o *erro* $=$ saída desejada $-$ saída real.
     Ou seja: $\Delta w_n = \eta \cdot x_n \cdot (d - y)$.
  3. Atualizando os pesos:
     $w_0 = 0,4 + 0,4 \cdot 0 \cdot (-1 - (+1)) = 0,4$
     $w_1 = -0,6 + 0,4 \cdot 0 \cdot (-1 - (+1)) = -0,6$
     $w_2 = 0,6 + 0,4 \cdot 1 \cdot (-1 - (+1)) = -0.2$
     $w_\theta = 0,5 + 0,4 \cdot -1 \cdot (-1 - (+1)) = 1,3$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **110**. Saída desejada: $d = +1$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **110**. Saída desejada: $d = +1$
  1. u = 1(0,4) + 1(-0,6) + 0(-0,2) - 1(1,3) = -1,5

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **110**. Saída desejada: $d = +1$
  1. u = 1(0,4) + 1(-0,6) + 0(-0,2) - 1(1,3) = -1,5

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **110**. Saída desejada: $d = +1$
  1. u = 1(0,4) + 1(-0,6) + 0(-0,2) - 1(1,3) = -1,5
     $y = g(u) = -1$ (uma vez que $-1,5 < 0$)

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **110**. Saída desejada: $d = +1$
  1. u = 1(0,4) + 1(-0,6) + 0(-0,2) - 1(1,3) = -1,5
     $y = g(u) = -1$ (uma vez que $-1,5 < 0$)
  2. Atualizando os pesos:
     $w_0 = 0,4 + 0,4 \cdot 1 \cdot (1 - (-1)) = 1,2$
     $w_1 = -0,6 + 0,4 \cdot 1 \cdot (1 - (-1)) = 0,2$
     $w_2 = -0,2 + 0,4 \cdot 0 \cdot (1 - (-1)) = -0.2$
     $w_\theta = 1,3 + 0,4 \cdot -1 \cdot (1 - (-1)) = 0,5$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.
- Novamente o padrão **110**. Saída desejada: $d = +1$

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.
- Novamente o padrão **110**. Saída desejada: $d = +1$
  1. u = 1(1,2) + 1(0,2) + 0(-0,2) - 1(0,5) = 0,9

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.
- Novamente o padrão **110**. Saída desejada: $d = +1$
  1. u = 1(1,2) + 1(0,2) + 0(-0,2) - 1(0,5) = 0,9

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.
- Novamente o padrão **110**. Saída desejada: $d = +1$
  1. u = 1(1,2) + 1(0,2) + 0(-0,2) - 1(0,5) = 0,9
     $y = g(u) = +1$ (uma vez que $0,9 \geq 0$)

Redes Neurais Artificiais

- Perceptron

Exemplo

- Novamente o padrão **001**. Saída desejada: $d = -1$
  1. u = 0(1,2) + 0(0,2) + 1(-0,2) - 1(0,5) = -0,7
     $y = g(u) = -1$ (uma vez que $-0,7 < 0$)
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.
- Novamente o padrão **110**. Saída desejada: $d = +1$
  1. u = 1(1,2) + 1(0,2) + 0(-0,2) - 1(0,5) = 0,9
  2. Atualizando os pesos:
     Como $d = y$ não há necessidade de atualizar os pesos.

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Dada uma rede do tipo Perceptron formada por um neurônio com três terminais de entrada, utilizado os pesos iniciais $w_0 = 0,4$, $w_1 = -0,6$ e $w_2 = 0,6$, limiar $\theta = 0,5$ e uma taxa de aprendizado $\eta = 0,4$. Responda os itens abaixo:
  1. Ensinar a rede a gerar a saída -1 para o padrão 001 e a saída +1 para o padrão 110;
  2. A que classe pertencem os padrões 111, 000, 100, 011?

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **111**
  u = 1(1,2) + 1(0,2) + 1(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **111**
  u = 1(1,2) + 1(0,2) + 1(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$
- Padrão **000**
  u = 0(1,2) + 0(0,2) + 0(-0,2) - 1(0,5) = -0,5 $\therefore y = -1$

Estácio

Redes Neurais Artificiais

- Perceptron

Exemplo

- Padrão **111**
  u = 1(1,2) + 1(0,2) + 1(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$
- Padrão **000**
  u = 0(1,2) + 0(0,2) + 0(-0,2) - 1(0,5) = -0,5 $\therefore y = -1$
- Padrão **100**
  u = 1(1,2) + 0(0,2) + 0(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$

Redes Neurais Artificiais

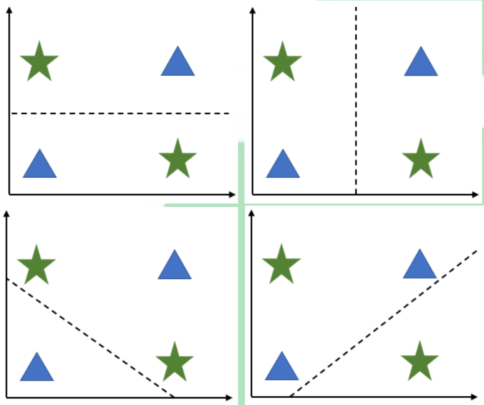
- Perceptron

Exemplo

- Padrão **111**
  u = 1(1,2) + 1(0,2) + 1(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$
- Padrão **000**
  u = 0(1,2) + 0(0,2) + 0(-0,2) - 1(0,5) = -0,5 $\therefore y = -1$
- Padrão **100**
  u = 1(1,2) + 0(0,2) + 0(-0,2) - 1(0,5) = 0,7 $\therefore y = +1$
- Padrão **011**
  u = 0(1,2) + 1(0,2) + 1(-0,2) - 1(0,5) = -0,5 $\therefore y = -1$

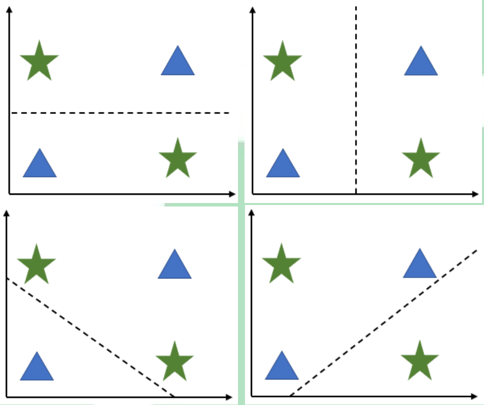
Redes Neurais Artificiais

- MLP
  - Um Perceptron só é capaz de treinar sobre porblemas cuja solução seja linearmente separável.



Redes Neurais Artificiais

- MLP
  - Um Perceptron só é capaz de treinar sobre porblemas cuja solução seja linearmente separável.
  - Isso faz com que determinados problemas (até simples) sejam impossíveis de serem resolvidos:



Redes Neurais Artificiais

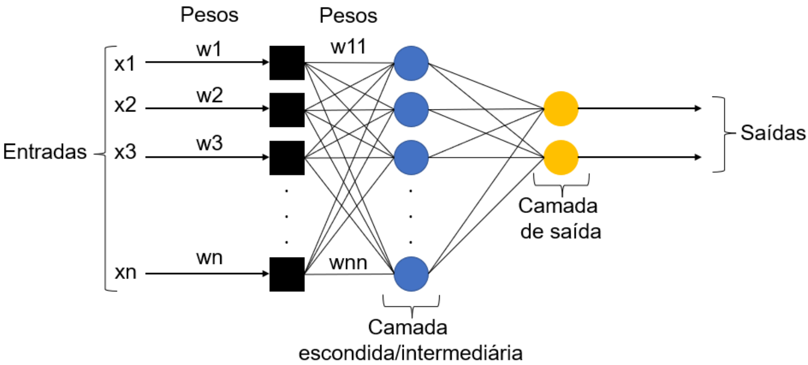
- MLP
  - Solução: redes com uma ou mais camadas escondidas/intermediárias.

Redes Neurais Artificiais

- MLP
  - Solução: redes com uma ou mais camadas escondidas/intermediárias.
  - Duas camadas intermediárias são suficientes para a aproximação de qualquer função [1].

Redes Neurais Artificiais

- MLP
  - Solução: redes com uma ou mais camadas escondidas/intermediárias.
  - Duas camadas intermediárias são suficientes para a aproximação de qualquer função [1].



Redes Neurais Artificiais

- MLP
  - A MLP funciona de forma semelhante ao Perceptron (pesos, função de ativação, etc.).
  - Entretanto, fica em aberto quantas camadas e quantos neurônios em cada camada escondida são necessários/suficientes.
  - Além disso fica também em aberto como os neurônios vão se ligar (abordagem mais comum: completamente ligados).
  - Por fim, o aprendizado, ou seja, a atualização dos pesos se torna mais complexa (Backpropagation foi o algoritmo que "salvou"as RNAs de serem descartadas para sempre).

Estácio

Redes Neurais Artificiais

- MLP
  - Vídeo de MLP sendo usado para aprender o jogo do dinossauro do Chrome [2].

Aprendizado Supervisionado

Regressão

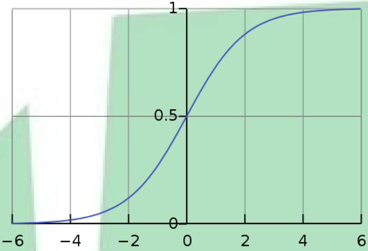
Regressão Logística

Regressão Logística

- Imagine um conjunto de dados contendo N pontos. Cada ponto i consiste de uma série de m variáveis de entrada $x_{1,i}, \ldots, x_{m,i}$ (variáveis independentes, ou atributos) e uma saída binária Y_i (variável dependente, ou classe), ou seja, a saída só pode assumir dois valores 0 ou 1. O objetivo da Regressão Logística é utilizar o conjunto de dados para criar um modelo preditivo para a variável de saída.

Regressão Logística

- A função logística é uma função sigmoide:



Regressão Logística

- Função logística:
  $\delta : \mathbb{R} \to (0, 1)$
  $$\delta(t) = \frac{\mathrm{e}^t}{\mathrm{e}^t + 1} = \frac{1}{1 + \mathrm{e}^{-t}}$$

Regressão Logística

- Função logística:
  $\delta : \mathbb{R} \to (0, 1)$
  $$\delta(t) = \frac{\mathrm{e}^t}{\mathrm{e}^t + 1} = \frac{1}{1 + \mathrm{e}^{-t}}$$
- Assumindo que t é uma função linear, então $t = \beta_0 + \beta_1 x$
  Substituindo: $p(x) = \delta(t) = \dfrac{1}{1 + \mathrm{e}^{-(\beta_0 + \beta_1 x)}}$

Regressão Logística

- Pulando toda a matemática que ainda tem depois da função apresentada, vamos ver um pequeno exemplo.

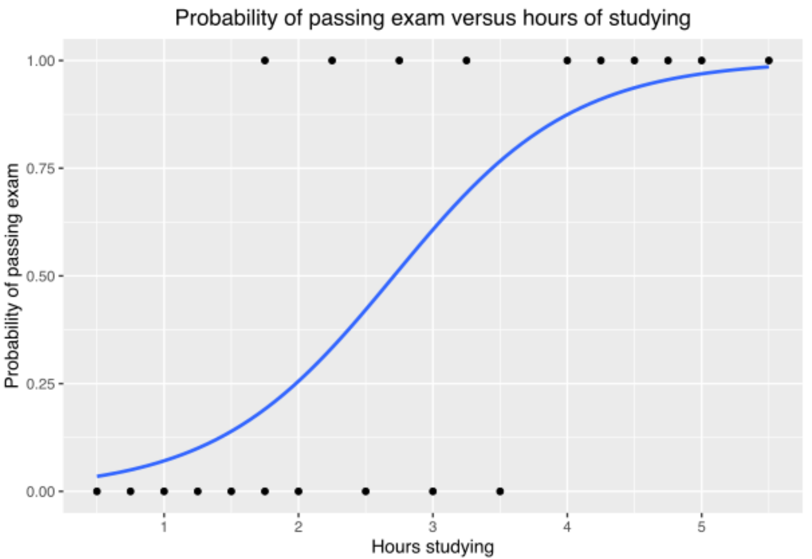
Exemplo

- Um grupo de 20 estudantes passou de 0 a 6 horas estudando para uma prova. Os dados mostram, para cada aluno, o quanto ele estudou e se passou (1) ou não (0). Com a regressão logística, vamos ver a probabilidade de passar na prova levando em consideração o tempo de estudo.

Horas(x_k)	0,5	0,75	1,00	1,25	1,50	1,75	1,75	2,00	2,25	2,50	2,75	3,00	3,25	3,50	4,00	4,25	4,50	4,75	5,00	5,50
Resultado(y_k)	0	0	0	0	0	0	1	0	1	0	1	0	1	0	1	1	1	1	1	1

Tabela: Base de Dados

Regressão Logística



Aprendizado Não Supervisionado

k-Means

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.
- Pseudo-Algoritmo:

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.
- Pseudo-Algoritmo:
    1. Posicione k centroides aleatoriamente no espaço amostral;

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.
- Pseudo-Algoritmo:
  1. Posicione k centroides aleatoriamente no espaço amostral;
  2. $\forall$ a $\in$ A $\{a \in k_i | dist(a, k_i) = min(dist(a, k))\}$;

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.
- Pseudo-Algoritmo:
  1. Posicione k centroides aleatoriamente no espaço amostral;
  2. $\forall$ a $\in$ A $\{a \in k_i | dist(a, k_i) = min(dist(a, k))\}$;
  3. Atualize os centroides;

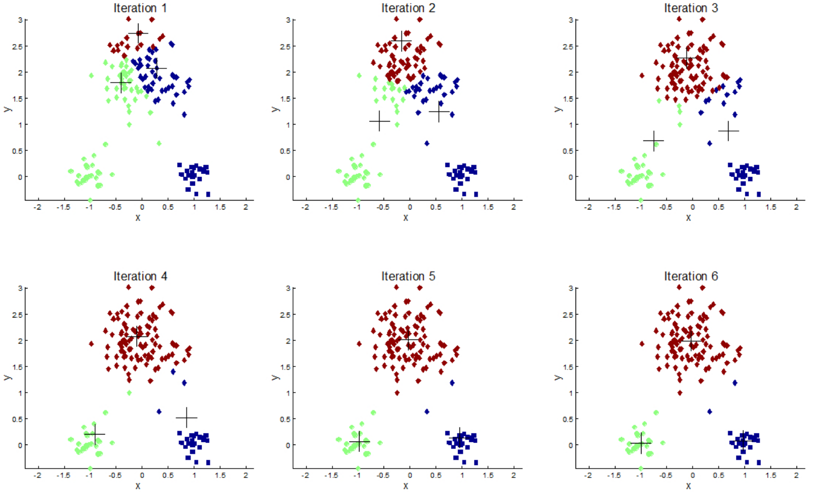
Estácio

k-Means

- k-Means é um algoritmo de *clusterização* (ou agrupamento).
- O termo significa k médias, onde k é um número inteiro qualquer.
- Pseudo-Algoritmo:
  1. Posicione k centroides aleatoriamente no espaço amostral;
  2. $\forall$ a $\in$ A $\{a \in k_i | dist(a, k_i) = min(dist(a, k))\}$;
  3. Atualize os centroides;
  4. Enquanto a posição dos centroides modifica, repita os passos 2 e 3;

Estácio

k-Means



Estácio

FIM

Finalmente terminamos!
Espero que tenham gostado!

Obrigado pela atenção e paciência!

Referências I

[1] Antônio de Pádua Braga, Teresa Bernarda Ludermir e André Ponce de Leon F. de Carvalho. **Redes Neurais Artificiais: Teoria e Aplicações**. LTC Editora, 2000.

[2] Universo Programado. **Inteligência Artificial destruindo no dinossauro da Google! (Rede Neural)**. `https://www.youtube.com/watch?v=NZlIYr1slAk`. Último acesso em: 23/07/22.

Estácio