

Machine Learning Package

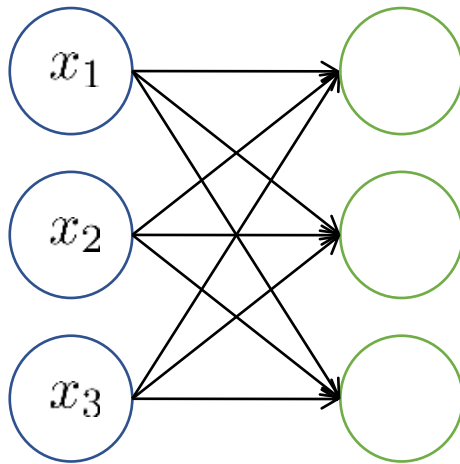
Portfolio de algoritmos de Machine Learning

Sumário

- Os modelos de redes neuronais são uma analogia do funcionamento do cérebro humano. Uma rede neuronal é um processador paralelo capaz de adquirir conhecimento em problemas não lineares e complexos.
- Iremos implementar um modelo de rede neuronal genérico que permite definir arquiteturas (topologias) complexas - *NN*
- Iremos implementar a primeira fase do algoritmo *Backpropagation* que permite treinar um modelo de redes neuronais – *forward propagation*
- Iremos implementar uma rede neuronal totalmente conectada – *Dense layer*
- Iremos implementar uma rede neuronal de ativação tendo como base a função sigmoid – *Sigmoid Activation layer*

Objeto *Dense*

- Onde podemos implementar a rede neuronal do tipo *Dense*?
- Quais os parâmetros de uma rede neuronal do tipo *Dense*?
- Quais os atributos da rede neuronal do tipo *Dense*?
- Como podemos calcular o output para a próxima *layer*?
 - Temos de considerar os pesos armazenados na rede neuronal



$$a_1^{(2)} = g(\Theta_{10}^{(1)} x_0 + \Theta_{11}^{(1)} x_1 + \Theta_{12}^{(1)} x_2 + \Theta_{13}^{(1)} x_3)$$

$$a_2^{(2)} = g(\Theta_{20}^{(1)} x_0 + \Theta_{21}^{(1)} x_1 + \Theta_{22}^{(1)} x_2 + \Theta_{23}^{(1)} x_3)$$

$$a_3^{(2)} = g(\Theta_{30}^{(1)} x_0 + \Theta_{31}^{(1)} x_1 + \Theta_{32}^{(1)} x_2 + \Theta_{33}^{(1)} x_3)$$

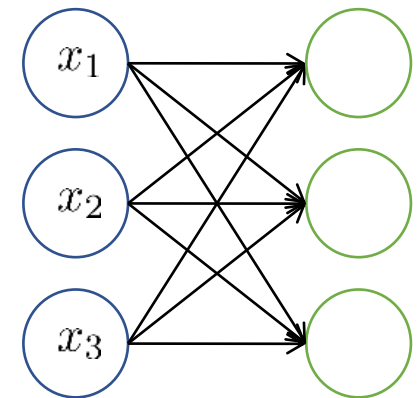
Objeto *SigmoidActivation*

- Onde podemos implementar a rede neuronal do tipo *SigmoidActivation*?
- Quais os parâmetros de uma rede neuronal do tipo *SigmoidActivation*?
- Quais os atributos da rede neuronal do tipo *SigmoidActivation*?
- Como podemos calcular o output para a próxima *layer*?
 - Temos de aplicar a função sigmoid aos dados de input

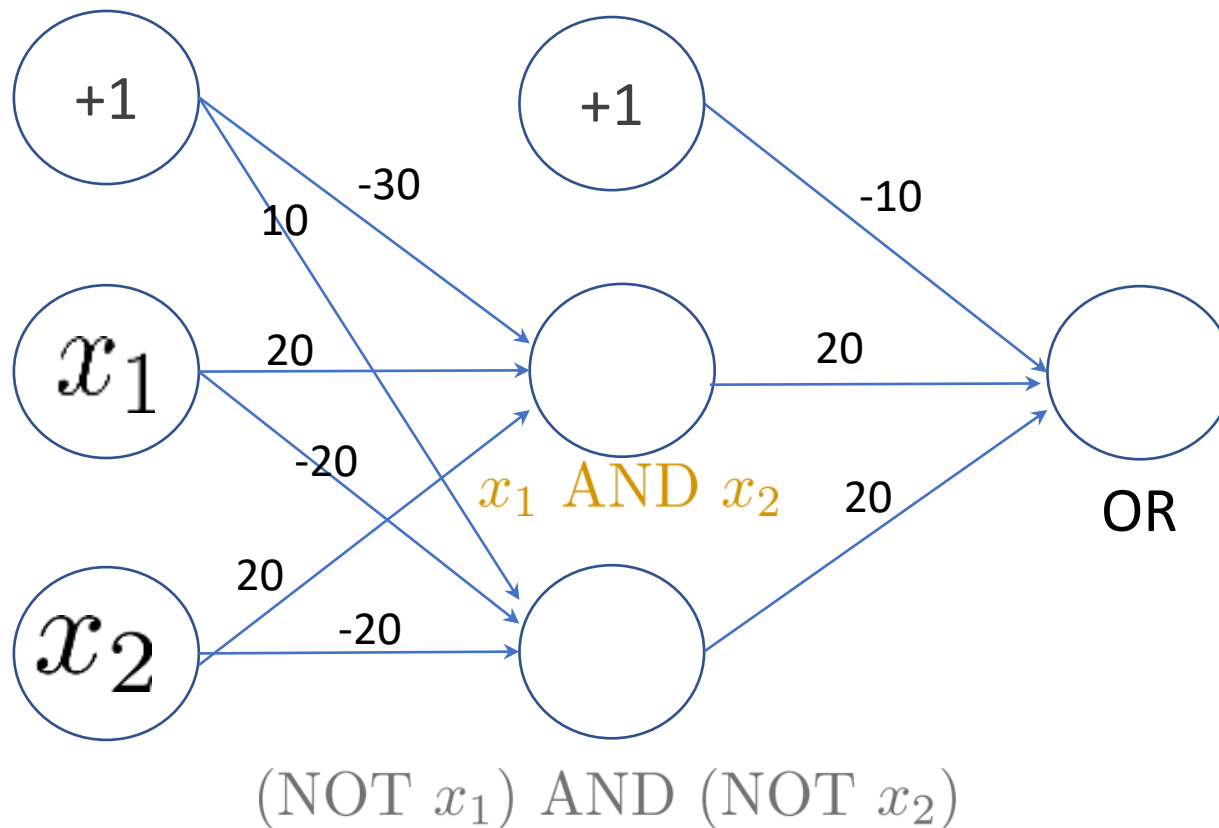
$$\frac{1}{1 + e^{-z}}$$

Objeto *NN*

- Onde podemos implementar o modelo *NN*?
- Quais os parâmetros de um modelo genérico de redes neuronais?
- Quais os atributos de um modelo genérico de redes neuronais?
- Como podemos treinar um modelo de redes neuronais?
 - Considera apenas a primeira fase do algoritmo de treino *Backpropagation* – *forward propagation*
- Como podemos obter previsões a partir de um modelo de redes neuronais treinado?



Exemplo X1 XNOR X2



Os pesos:

- 30, 20, 20

10, -20, -20

-10, 20, 20

- Quantas *Dense layers* constituem a arquitetura deste modelo?
- Quais as suas dimensões e pesos?
- Quantas *SigmoidActivation layers* constituem a arquitetura deste modelo?
- Obtém previsões para o caso de estudo X1 XNOR X2

Avaliação

■ Exercício 10: Redes neuronais, *layers* de ativação, regressão e *multiclass*

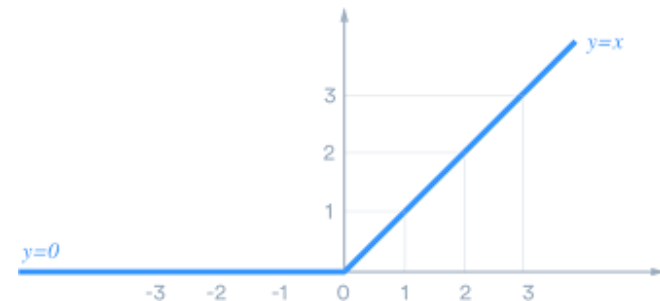
- 10.1) Adiciona uma nova *layer* de ativação chamada *SoftMaxActivation*
 - Esta *layer* deve calcular a probabilidade de ocorrência de cada classe usando a seguinte formula:

$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^K e^{z_j}}$$

- Sendo que
 - z_i - $X - \max(X)$
 - e^{z_i} - exponencial do vetor z_i
 - $\sum_{j=1}^K e^{z_j}$ - soma da exponencial do vetor z_i - considera a seguinte função do numpy: `np.sum(..., axis=1, keepdims=True)`
- Esta *layer* é particularmente útil para problemas *multiclass*

Avaliação

- Exercício 10: Redes neurais, *layers* de ativação, regressão e *multiclass*
- 10.2) Adiciona uma nova *layer* de ativação chamada *ReLUActivation*
 - Esta *layer* deve calcular a relação linear retificada. Ou seja, debes considerar a parte positiva do seu argumento.
 - Considera a seguinte função do numpy para implementar a função ReLU
 - `np.maximum(mínimo, X)`



Avaliação

- Exercício 10: Redes neuronais, *layers* de ativação, regressão e *multiclass*
- 10.3) Constrói um modelo de redes neuronais considerando a seguinte topologia:
 - O *dataset* de treino contém **32 *features***
 - O problema é do tipo **binário**
 - O modelo deve conter 3 *Dense layers*:
 - *Dense layer de input*
 - *Dense layer com redução do número de neurónios (units) para metade*
 - *Dense layer final (output)*
 - Usa a *SigmoidActivation* como *layer* de ativação

Avaliação

- Exercício 10: Redes neuronais, *layers* de ativação, regressão e *multiclass*

- 10.4) Constrói um modelo de redes neuronais considerando a seguinte topologia:
 - O *dataset* de treino contém **32 features**
 - O problema é do tipo ***multiclass* com 3 classes**
 - O modelo deve conter 3 *Dense layers*:
 - *Dense layer de input*
 - *Dense layer com redução do número de neurónios (units) para metade*
 - *Dense layer final (output)*
 - Usa a *SigmoidActivation* como *layer* de ativação
 - Usa a *SoftMaxActivation* como última *layer* de ativação

Avaliação

- Exercício 10: Redes neuronais, *layers* de ativação, regressão e *multiclass*

- 10.5) Constrói um modelo de redes neuronais considerando a seguinte topologia:
 - O *dataset* de treino contém **32 features**
 - O problema é do tipo **regressão**
 - O modelo deve conter 3 *Dense layers*:
 - *Dense layer de input*
 - *Dense layer com redução do número de neurónios (units) para metade*
 - *Dense layer final (output)*
 - Usa a *ReLU* como *layer* de ativação
 - Considera que o modelo deve acabar com uma ativação linear