

Machine Learning Package

Portfolio de algoritmos de Machine Learning

Sumário

- Iremos continuar a implementação do modelo genérico de redes neuronais - *NN*
- Iremos implementar parte da segunda fase do algoritmo *Backpropagation* que permite treinar um modelo de redes neuronais – *backward propagation*
- Iremos implementar a propagação do erro para as *layers* anteriores

Objeto *Dense*

- Como podemos propagar o erro *layer a layer*?
 - Considera a segunda fase do algoritmo *Backpropagation* – *backward propagation*
 - Considera que o erro a passar à *layer* anterior consiste na derivada do erro atual em ordem aos dados de input (X)

$$\frac{\partial E}{\partial X} = \frac{\partial E}{\partial Y} W^t$$

$$\frac{\partial E}{\partial W} = X^t \frac{\partial E}{\partial Y}$$

$$\frac{\partial E}{\partial B} = \frac{\partial E}{\partial Y}$$

Objeto *SigmoidActivation*

- Como podemos propagar o erro nas *layers* de ativação?
 - Considera que estas *layers* não têm parâmetros associados à aprendizagem
 - Considera que o erro a passar à *layer* anterior consiste na derivada do erro atual em ordem aos dados de input (X)

$$\frac{dE}{dY} * f'(X)$$

$$\frac{1}{1 + e^{-x}} * \left(1 - \frac{1}{1 + e^{-x}}\right)$$

Avaliação

- Exercício 12: Redes neuronais e *Backpropagation*
 - 12.1) Implementa o método de *backward propagation* da *ReLUActivation layer*.
 - Considera o seguinte para a propagação do erro numa *layer* de ativação *ReLU*:
 - Substituir valores de erro superiores a 0 por 1
 - Substituir valores de erro inferiores a 0 por 0
 - Multiplicação elemento a elemento entre o erro e os valores anteriores
 - 12.2) Constrói um modelo de redes neuronais adequado ao dataset `breast-bin.csv`.
 - 12.3) Constrói um modelo de redes neuronais adequado ao dataset `cpu.csv`.