

Machine Learning Package

Portfolio de algoritmos de Machine Learning

Sumário

- Iremos continuar a implementação do modelo de rede neuronal genérico - *NN*
- Iremos implementar parte da segunda fase do algoritmo *Backpropagation* que permite treinar um modelo de redes neuronais – *backward propagation*
- Iremos implementar uma medida de cálculo do erro e a sua derivada – *mse* e *mse_derivative*

Objeto *NN*

- Considera agora todo algoritmo de treino *Backpropagation*!
- Que parâmetros devemos adicionar ao objeto *NN*?
 - Algoritmo iterativo
 - Cálculo do erro entre valores previstos e valores reais
 - Podemos modificar a taxa de aprendizagem do algoritmo?
- Que atributos devemos adicionar ao objeto *NN*?
 - Talvez seja interessante acompanhar a diminuição do erro?
- Como podemos treinar um modelo de redes neurais?
 - Considera a primeira fase do algoritmo *Backpropagation* – *forward propagation*
 - Considera a segunda fase do algoritmo *Backpropagation* – *backward propagation*
 - Algoritmo iterativo que ajusta os pesos de cada *layer* tendo em conta o erro entre os valores previstos e os valores reais
- Como podemos obter o erro entre os valores reais e os valores previstos pelo modelo?

mse metric

- Consulta a medida de erro *mse* no *metrics* sub-package
- *def mse*
 - assinatura/argumentos:
 - *y_true* – valores reais de Y
 - *Y_pred* – valores estimados de Y
 - output esperado:
 - O valor do erro entre *y_true* e *y_pred*
 - Confirma que a função implementada respeita a seguinte formula da *mse*
 - Sendo que:
 - y_i - valores reais
 - \hat{y}_i - valores previstos
 - n – número de exemplos

$$\frac{\sum_{i=1}^n (y_i - \hat{y}_i)^2}{2n}$$

mse metric

- Adiciona a função *mse_derivative* que permite calcular o erro usando a derivada da *mse*
- *def mse_derivative*
 - assinatura/argumentos:
 - *y_true* – valores reais de Y
 - *Y_pred* – valores estimados de Y
 - output esperado:
 - O valor do erro entre *y_true* e *y_pred*
 - algoritmo:
 - Calcula o erro usando a derivada da *mse* em ordem a \hat{y}_i

Avaliação

■ Exercício 11: Medidas de erro

- 11.1) Adiciona uma nova medida de erro chamada *cross entropy*
 - Deves adicionar a função *cross_entropy* ao módulo *cross_entropy.py* no sub-package *metrics*
 - Considera a seguinte formula para implementares a medida de erro *cross entropy*:
$$\frac{-\sum_{i=1}^n y_i * \ln \hat{y}_i}{n}$$
 - Sendo que:
 - y_i - valores reais
 - \hat{y}_i - valores previstos
 - n – número de exemplos

Avaliação

■ Exercício 11: Medidas de erro

- 11.2) Adiciona agora a derivada da medida de erro *cross entropy*
 - Deves adicionar uma nova função chamada *cross_entropy_derivative* ao módulo *cross_entropy.py*
 - Considera a formula da *cross entropy* apresentada no slide anterior.
 - Deriva a *cross entropy* em ordem a \hat{y}_i