

CS3006 – Machine Learning Algorithms

CIA2

1. Implement a neural network from scratch. Take any dataset. If you take a regression problem, use the equations derived in the class. If you take a classification problem, use the below equations: (run minimum 200 iterations and get the result). Use the gradient descent optimization technique for weight optimization.

Forward propagation

$$Z^{[1]} = W^{[1]}X + b^{[1]}$$

$$A^{[1]} = g_{\text{ReLU}}(Z^{[1]})$$

$$Z^{[2]} = W^{[2]}A^{[1]} + b^{[2]}$$

$$A^{[2]} = g_{\text{softmax}}(Z^{[2]})$$

Backward propagation

$$dZ^{[2]} = A^{[2]} - Y$$

$$dW^{[2]} = \frac{1}{m} dZ^{[2]} A^{[1]T}$$

$$dB^{[2]} = \frac{1}{m} \sum dZ^{[2]}$$

$$dZ^{[1]} = W^{[2]T} dZ^{[2]} * g^{[1]'}(z^{[1]})$$

$$dW^{[1]} = \frac{1}{m} dZ^{[1]} A^{[0]T}$$

$$dB^{[1]} = \frac{1}{m} \sum dZ^{[1]}$$

Parameter updates

$$W^{[2]} := W^{[2]} - \alpha dW^{[2]}$$

$$b^{[2]} := b^{[2]} - \alpha db^{[2]}$$

$$W^{[1]} := W^{[1]} - \alpha dW^{[1]}$$

$$b^{[1]} := b^{[1]} - \alpha db^{[1]}$$

2. For the same dataset, build a neural network using keras library. Run the same number of epochs and compare the results obtained with your model vs the built-in keras model.

Hint: you shall use make_classification function to create a new dataset for classification problem/ use MNIST dataset or use the load_boston dataset for regression problem.

Write your inferences in the python notebook itself and submit the same as a zip file in the LMS.