

REGRESSION FUNCTION FITTING

This repository implements a regression task. I set a neural network to fit a quadratic function. By doing so, I can check if my thoughts of full connected neural network are all right.

DATA SETS

How do I get the data sets for training and test?

It is much easier than what I think to get the datasets.

I will just use a binary function to generate some (x,y) pairs and add some noise (x,y) pairs as data sets.

Using `numpy.randn()` to add noise: $Y = a_1X^2 + a_2X + a_3 + \text{numpy.rand.randn}()$

NEURAL NETWORK STRUCTURE

Input -> Hidden layer -> Output.

Details

Optimise it with gradient descent

Forward: $X \rightarrow X(4,m) \rightarrow W(4,1) * X \rightarrow Y(1,m)$

Backward: $\text{cost(MSE)} \rightarrow \text{gradA}$

Update: $A = A - \text{learning_rate} * \text{gradA}$

Get Gradient

Below are my calculate form of how to get W1's gradient,

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

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

$$\frac{\partial cost}{\partial W1} = \frac{\partial cost}{\partial \hat{y}} \frac{\partial \hat{y}}{\partial W1} = \left(\frac{2}{m} \sum_{i=1}^m (\hat{y}_i - y_i) \right) \cdot A_0$$

so, after every iteration of learning, I will simultaneously update my parameter W1 like follows,

$$W1 := W1 - \alpha \cdot \frac{\partial cost}{\partial W1}$$

Above is my BFF project.

11th July, 2021

Today I just finished the basic framework of my neural network, just some linear things in it. Everything goes on as I wanted. But there still some bugs need to be finished.

- Learn how to choose a proper iteration figure and a proper learning_rate.
- Do error analyze on my framework.
- Standardize my output.