Assignments

Assignment for Saturday March 06th

1. Implement 2 hidden layer neural network in numpy to approximate sin(X)

https://github.com/bismillahkani/neural-network-numpy

2. Read more about backprop from deeplearningbook.org

- [x] ~~Reading complete~~

3. Read and summarize in a doc - https://www.iro.umontreal.ca/~vincentp/ift3395/lectures/backprop_old.pdf the original paper from Nature magazine by Hinton

Summary:

The paper describe a procedure called back propagation which iteratively adjust the weights of neural network with the aim to minimize the error between the actual output and desired output.

Forward propagation of neural network is given by two equations. Equation 1 is a matrix multiplication of weights and inputs.

$$x_j = \sum_i y_i w_{ji}$$

Equation 2 is a non-linear activation function on outputs of equation 1.

$$y_j = \frac{1}{1+e^{-x_j}}$$

The repeated operations of linear and non-linear operations defines the forward propagation.

The aim is to determine the set of weights of the neural network which minimizes the error E between the actual output y and the desired output d defined as follows,

$$E = \frac{1}{2}\sum_c \sum_j (y_{j,c} - d_{j,c})^2$$

The optimal weights are determined using gradient descent method and for that we need to compute the partial derivative of E with respect to each weight in the network.

Backward Propagation is based mainly on two things i.e. partial derivative and chain rule of calculus.

The backprop starts with the last layer and repeat the procedure for successive earlier layers. The derivatives computed are then used to change the weights. The weights can be changed for every input-output pair or as an alternative the derivatives can be accumulated over all samples before changing weights.

$$\partial E/\partial y_i = \sum_j \partial E/\partial x_j \cdot w_{ji}$$

The drawback of the gradient descent is that it is not guaranteed to find a global minimum and sometimes it might converge at local minimum. The current form of learning is different from the way how brain learns and it might be interesting to find more biological ways of doing gradient descent in neural networks.

4. Read and summarize in a doc - https://www.jmlr.org/papers/volume2/bousquet02a/bousquet02a.pdf

This is a high level summary.

Summary:

The paper is about the stability and generalization of learning algorithm. From the empirical and leave one out error we can use the notions of stability defined in this paper to derive the bounds of generalization error of the learning algorithm. The method can be used for classification and regression problem. The approach used in this paper is based on the sensitive analysis which determines the influence of variation in input on the output. The two types of randomness an algorithm have to handle are the sampling randomness and the noise in the data. The stability of an algorithm is defined in many ways such as hypothesis stability, pointwise hypothesis stability, error stability, uniform stability. The main result of the paper is that the an algorithm that has good stability properties has exponential bound on generalization error. The stability of regularized algorithm is control by the regularization parameter lambda.