L04-Backpropagation-Lecture Notes

{Submitted By: 15MI427 & 15MI412}

Introduction:

Backpropagation, is an algorithm for supervised learning of artificial neural networks using gradient descent. Given an artificial neural network and an error function, the method calculates the gradient of the error function with respect to the neural network's weights. It is a generalization of the delta rule for perceptron to multilayer feedforward neural networks.

The "backwards" part of the name stems from the fact that calculation of the gradient proceeds backwards through the network, with the gradient of the final layer of weights being calculated first and the gradient of the first layer of weights being calculated last. Partial computations of the gradient from one layer are reused in the computation of the gradient for the previous layer. This backwards flow of the error information allows for efficient computation of the gradient at each layer versus the naive approach of calculating the gradient of each layer separately.

Backpropagation's popularity has experienced a recent resurgence given the widespread adoption of deep neural networks for image recognition and speech recognition. It is considered an efficient algorithm, and modern implementations take advantage of specialized GPUs to further improve performance.

Formal Definition:

Backpropagation is analogous to calculating the delta rule for a multilayer feedforward network. Thus, like the delta rule, backpropagation requires three things:

I. Dataset consisting of input-output pairs $(x_{i,}y_i)$, where x_i is the input and y_i is the desired output of network on input x_i. The set of input-output pairs of size NN is denoted $X = \{(x_{1,}y_1),\ldots, (x_{N,}y_N)\}$.

II. A feedforward neural network, as formally defined in the article concerning feedforward neural networks, whose parameters are collectively denoted θ. In backpropagation, the parameters of primary interest are w^k_{ij}, the weight between node j in layer l_k and node i in l_{k-1}, and b^k_i, the bias for node i in the layer l_k. There are no connections between nodes in the same layer and layers are fully connected.

III. An error function, E(X,θ), which defines the error between the desired output y_i and the calculated output $y_i^{\wedge}$ of the neural network on input x_i for a set of input-output pairs $(x_{i,}y_i) \in X$ and a particular value of the parameters θ.

Training a neural network with gradient descent requires the calculation of the gradient of the error function E(X,θ) with respect to the weights wkij and biases bki. Then, according to the learning rate α, each iteration of gradient descent updates the weights and biases (collectively denoted θ) according to

$$\theta^{t+1} = \theta^t - \alpha \frac{\partial E(X, \theta^t)}{\partial \theta},$$

where θ^t denotes the parameters of the neural network at iteration t in gradient descent.

Advantages:

- It is fast, simple and easy to program.
- It has no parameters to tune (except for the number of input).
- This is a shift in mind set for the learning-system designer instead of trying to design a learning algorithm that is accurate over the entire space.
- It requires no prior knowledge about the weak learner and so can be flexible.

Disadvantages:

- The actual performance of Backpropagation on a particular problem is clearly dependent on the input data.
- Backpropagation can be sensitive to noisy data and outliers.
- Fully matrix-based approach to backpropagation over a mini-batch.

Applications:

- Mapping character strings into phonemes so they can be pronounced by a computer.
- Neural network trained how to pronounce each letter in a word in a sentence, given the three letters before and three letters after it in a window
- In the field of Speech Recognition.
- In the field of Character Recognition.
- In the field of Face Recognition.