Neural Networks

Assignment 6

Ravikiran Bhat
Rubanraj Ravichandran
Ramesh Kumar

November 19, 2017

1 Summary

- MLPs consist of the input layer, one or more hidden computation layers and an output layer

- MLPS can solve some difficult problems when trained in a supervised manner using the error back-propagation algorithm that is based on the error correction learning rule.

- The error back-propagation algorithm consists of 2 passes through the network.

- In the forward pass, an activity pattern applied to the inputs propagates through different layers to produce a set of outputs.

- Synaptic weights are fixed during forward pass.

- During backward pass, the synaptic weights are adjusted according to the error correction rule.

- Model of each neuron has a non-linear activation function represented by the sigmoidal logistic function.

- The network exhibits high degree of connectivity determined by synapses of the network.

- The back-propagation algorithm provides a computationally efficient method for training of multilayer perceptrons.

  Prelimnaries

- A multilayer perceptron consists of two kinds of signals, namely function signals which is an input signal, and the error signal.

- At each neuron, the input signal is calculated as a function of the inputs and the weights applied to the neuron.

- The error signal originates at the output of the network and propageates backwards.

- Each hidden or output neuron computes the function signal and an estimate of the gradient vector.

Back-propagation

- Each neuron j in the network has an instantaneous error energy value $\xi(n)$ and an average squared energy value ξ_{av} which are a function of all free parameters of the network

- ξ_{av} represents the cost function as a measure of learning performance.

- The free parameters (namely the synaptic weights and bias levels) of the network must be adjusted to minimise ξ_{av}.

- For each neuron j in the network excited by an input from neuron i, the weight correction $\Delta w_{ji}(n)$ applied to $w_{ji}(n)$ after accounting for gradient descent in weight space is given by : $-\eta\delta_j(n)y_i(n)$ where $\delta_j(n)$ is the local gradient given by $e_j(n)\phi_j^{'}(v_j(n))$, with $\phi_j^{'}(v_j(n))$ being the derivation of the associated activation function.

- If j is the output neuron, $e_j(n)$ calculation is straightforward since it is supplied with a desired response.

- If j is a hidden neuron, error signal is calulated recursively in terms of error signals of all neurons to which it is directly connected as : $\phi_j^{'}(v_j(n)) \sum_k \delta_k(n)w_{kj}(n)$

- In the forward pass of the back-propagation algorithm, synaptic weights remain unaltered throughout the network, but undergo changes according to the delata rule during the backward pass.

- Computation of the δ of each neuron requires knowledge of the derivative of the activation function of the neuron.

- Two commonly used continuously differentiable non-linear activation functons in MLPs are : the Logistic function and the Hyperbolic tangent function.

- The learning-rate parameter should not be a constant but be a connection dependent parameter η_{ji}.

- In the back-propagation algorithm, learning may proceed for a given training set in 2 ways:

  - Sequential mode : Weights are updated after presenting each training example.

– Batch mode : Weights are updated after presenting all training examples that constitute an epoch.

- Sequential mode is stochastic in nature, requires less storage for each synaptic connection and less likely to be trapped in local minimum.

- Sequential mode is preferred as it provides simple to implement and provides easy solutions to large and difficult problems.

- The back-propagation algorithm is said to have converged when the absolute rate of change in the average squared error per epoch is sufficiently small.

Summary of Back-propagation

- Using a sequential mode of operation, the algorithm cycles through a training sample as follows:

  – Initialization : The synaptic weights and thresholds are picked from a uniform distribution with zero mean.

  – Presentation of training examples : Present the network with an epoch of training examples.

  – Forward Computation : Compute the error signal at the output for each training example in the epoch.

  – Backward computation : Compute the δs (local gradients) for each hidden neuron and output neuron in the network.

  – Iteration : Iterate the forward and backward computations until the stopping criterion is met.

XOR problem

- The XOR problem is a special case of classifying points in the unit hypercube, with the need to consider only the 4 corners of the unit square, with the input patterns : (0,0),(0,1),(1,0) and (1,1).

- We solve the XOR problem by using a single hidden layer with 2 neurons.

- Each neuron is represented by the McCulloh-Pitts model with a threshold function as the activation function.

- The bottom hidden neuron has an excitatory connection to output neuron while the top hidden neuron has a stronger inhibitory connection.

- The function of the output neuron is to construct a linear combination of the decision boundaries formed by the 2 hidden neurons.

  Heuristics for making back-propagation algorithm perform better

- Selecting sequential mode of back-propagation results in faster computations for large and redundant data.

- The training examples used should result in largest training error and must be radically different from all those previously used in order to be able to search more of the weight space.

- a MLP learns faster when the sigmoid activation function is antisymmetric than when it is nonsymmetric.

- The desired response of output neuron should be offset by some value from the limiting value of the sigmoid function.

- Each input variable should be preprocessed so that its mean value averaged over the training set is close to zero. Also the input variable should be uncorrelated and the decorrelated inputs should be scaled so that their covariances are approximately equal.

- Uniform distribution from which synaptic weights are selected should have zero mean and variance equal to reciprocal of number of synaptic connections of a neuron.

- Including prior information about the input-output mapping function in the learning process allows to generalize learning by examples to include learning by hints.

- All neurons in MLPs should ideally learn at the same rate. The η should be assigned a smaller value in the last layer however, compared to the front layers.

2   Exercise 2

This exercise is solved using ipython and ipython file/pdf has been attached along with the submission files.

3   Exercise 3

This exercise is solved using ipython and ipython file/pdf has been attached along with the submission files.