Backpropagation Implementation Issues

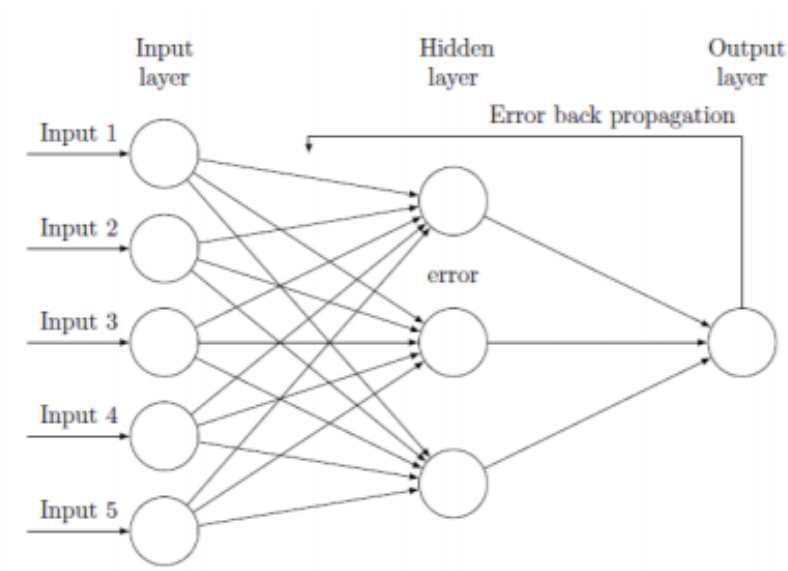
Backpropagation is a method used in Artificial Neural Networks to calculate a gradient that is needed to calculate the weights to be used in the network. With the backpropagation network, learning occurs during a training phase in which each input pattern in a training set is applied to the input units and then propagated forward. The pattern of activation arriving at the output layer is then compared with the associated output pattern to calculate an error signal. The error signal for such target output pattern is then backpropagated from the output to the input to appropriately adjust the weights in each layer of the network.

For instance, consider RoboCop. He wants to identify gang members and whether belong to the Jets or Sharks gang, by developing a reliable method to help predict gang membership. The police provided RoboCop access to their database of known gang members, which he used to as the dataset to train his network on. RoboCop arrests four individuals, whose information is not present in the training set, at a stakeout, and during interrogation learns their age, education, marital status, and occupation. RoboCop and his team need to figure out which gang these suspects belong to. They use relative activation of the Jets and Sharks units to make strong and weak predictions. When the Jet activation is large (near 1.0) and the Sharks activation is small (near 0.0), the network is making a strong prediction that the suspect is a Jet, and if the opposite is true then it predicts that the suspect is a Shark. In some cases, the network is stuck and unable to make a strong prediction (Jets and Sharks are near 0.5). Each new piece of information added to the database, RoboCop compares with the network's predictions to evaluate how well it generalizes. Each successful generalization shows that

RoboCop's predictor is reliable. He uses each mistake to improve the network's performance in future classifications.

The following are the implementation issues associated with the Backpropagation algorithm:

- Backpropagation with gradient descent is not guaranteed to find the global minimum of the error function, but only a local minimum. The algorithm always changes the weights in such a way that the error falls, this is after a brief rise in error. In this case, the algorithm gets stuck (it cannot go uphill) and the error will not decrease further.

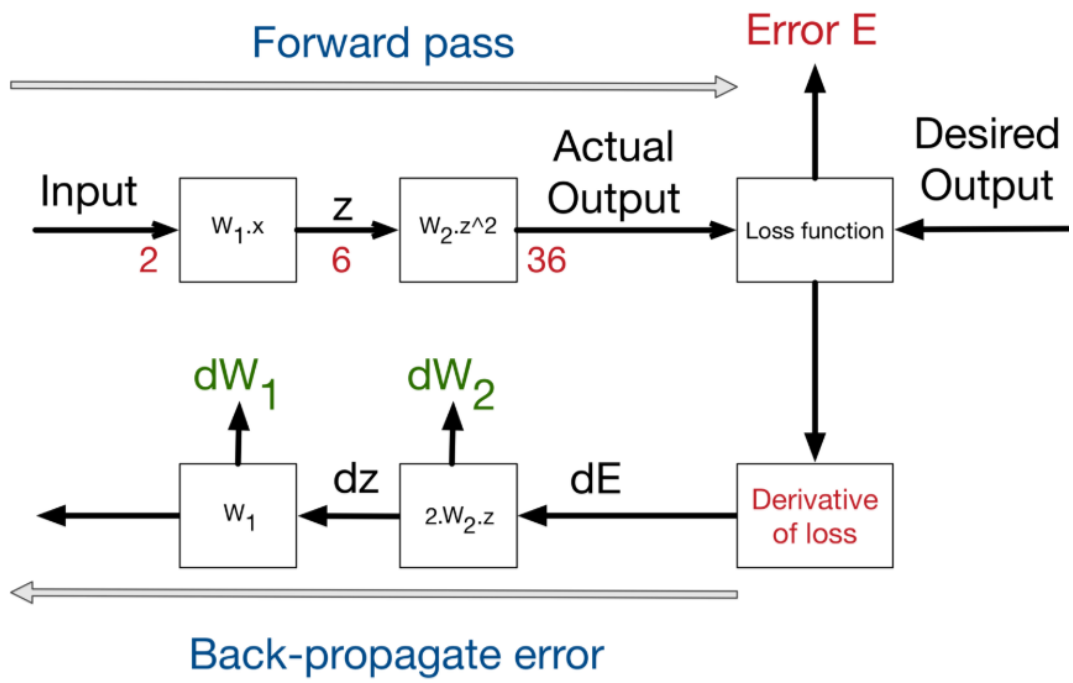


The figure above depicts the systematic diagram of Artificial Neural Network and its error backpropagation architecture

- Backpropagation suffers from network paralysis. Network paralysis occurs when the weights are adjusted to very large

values during training. Large weights can force most of the units to operate at extreme values, even in a region where the derivative of the activation function is very small.

- The learning rate and momentum have a large effect on the training time. The convergence of backpropagation tends to be very slow due to the constant step size of learning rate and momentum. Hence, it often yields sub-optimal solutions (Chaturvedi). The neural network may not converge at all if the initial weights are not selected properly.
- Network complexity i.e. number of hidden layers and number of hidden units and their interconnections with neuron of the other layers greatly affects the learning speed of ANN. This slowdown is due to the reduce in force/ effect of the error signal as it propagates backward through the layers of the network.
- Backpropagation also faces the moving target problem i.e. each unit in the interior of the network tries to evolve into a feature detector that will play some useful role in the network's overall computation, but this task is greatly complicated by the fact that all the other units are changing at the same time.
- Backpropagation faces a step-size problem since this method computed only the first partial derivative of the overall error function with respect to each weight in the network. In a practical learning system, you want to take fast steps and not infinitesimal steps. However, if we choose a step size that is too large, the network will not converge reliably to a good solution. And, to choose a reasonable step size, you will need to know the higher-order derivatives, the vicinity if current point in weight space, and the curvature of the error function and not just its slope, which is the only information available in the standard backpropagation algorithm.



The figure above depicts the process of backpropagation errors following this scheme: Input - Forward calls - Loss function - Derivative - Backpropagation of errors

- It has trouble crossing plateaus in the error function landscape. This issue is caused by the non-convexity of error functions in neural networks.
- The normalization range of training data and input output map affect the training time and accuracy (Chaturvedi).
- A multilayer backpropagation network with enough neurons can implement any function, but backpropagation will not always find the correct weights for the optimum solution.
- Backpropagation algorithms have a slow learning process, especially when large training sets or large networks are in use. For complex problems, it may require days or weeks to train the network or it may not even train at all since the learning phase

has intensive calculations. And, this results in non-optimum step size (Sivanandam, Deepa, 2006).

- Performance depends on initial weights. In most application of backpropagation, the range of initial weights is small. Increasing these weights significantly can improve learning rates, but when increased beyond a problem-dependent limit, performance degrades (Gowda, Mayya, 2014).

Gowda, C. Mayya, S. Comparison of Back Propagation Neural Network and Genetic Algorithm Neural Network for Stream Flow Prediction. 2014. Retrieved from: https://www.hindawi.com

Chaturvedi, D. Soft Computing: Techniques and its Applications in Electrical Engineering. Retrieved from: https://books.google.co.in

Sivanandam, S. Deepa, S. Introduction to Neural Networks Using Matlab 6.0. 2006. Retrieved from: https://books.google.co.in