

Artificial Neural Networks

For my perceptron implementation I have two special parameters (rate, epochs). The learning rate is the parameter that determines how much an updating step influences the current value of the weights while the epoch is simply the number of iterations that your neural network will train for. If you set the learning rate to a high value like more than 1, then the corrections to your neural network will be very dramatic and if you have a very small amount of epochs then you risk never finding the best fit. On the other side if you have a very small number like 0.0001 then your neural network will correct errors very slowly but you will need to have a lot more epochs to correctly fit your network.

ANN compared to other methods is a non-parametric model that allows to solve problems like classification or forecasting. ANN can basically solve most of the problems but this requires having enough data to feed in and also know how to tune them and the neural network structure that you should use that better fits your problem. Sadly this model is considered a black box model since it is very difficult to interpret what the model is doing or where the predictions are coming from. From what I've read this model is perfect when you are dealing with images with a structure called CNN but there are a lot of applications thanks to their flexibility. Another advantage is that you can speed up the process with GPU since you take advantage of the fast matrix operations.

It is worth implementing a small network in order to understand what is happening inside the algorithm and how the errors propagate across the layers but if you are looking to solve a problem fast then this is not the way to go. There are many libraries that allow you to create your own structures with a set of different layers, training parameters, hardware optimization, etc.

You should not use a neural network when you have very little data to train, if you don't have the time to tune it and understand what is going on then you should look for another algorithm, if you want to understand why your model is correctly or incorrectly predicting or classifying.