Algorithm

A Matlab program was implemented to train a perceptron to be able to recognise if a random given boolean function in given n dimensions is linearly separable or not. The program also counts how many boolean functions are linearly separable for given dimensions. The network learns iteratively over 20 epochs for each random boolean function. Each boolean function is only used only once for training. The program runs for 10^4 trials. The algorithm used to train the network was an iterative gradient descent (see chapter 5.2 in the book *Machine Learning for Neural Networks* by Bernard Mehlig.

Results

Table 1: Matlab results of linearly separable boolean functions in relation to n dimensions

n dimensions	Number of linearly separable boolean functions from Matlab results	Actual number of linear separable boolean functions	Number of boolean functions in n dimensions
2	14	14	16
3	104	104	256
4	262	1882	65536
5	0	94572	4294967296

Discussion and conclussion

The perceptron computes the right number of linearly separable functions in 2 and 3 dimensions. Though, not for 4 and 5. The number of boolean functions in 4 and 5 dimensions are 65536 respectively 4294967296. It's impossible for the perceptron to be trained for 65536 functions and beyond when the network is only capable of being trained for 10000 boolean functions (10^4 trials). The number of linearly separable boolean functions in 5 dimensions are 94572, the probability of finding *one* linearly separable function in 5 dimensions in one trial is therefore 0.00002 and 0.2 for 10^4 trials. That means that in one batch run, the perceptron will probably not find *any* linearly separable boolean functions in 5 dimensions. Similarly goes for 4 dimensions, the number of linearly separable boolean functions in 4 dimensions are 1882. The probability of finding *one* linearly separable function in 4 dimensions in one trial is therefore 0.0287 and 287 in 10^4 trials; the program should find about 287 linearly separable boolean functions for 4 dimensions which is roughly what the results show (see table 1). For 2 and 3 dimensions, the number of boolean functions are 16 respectively 256 which is well below 10^4 trials. This means that the perceptron will be trained for all boolean functions and likely find all linearly separable boolean functions for these dimensions, which the results shows as well. To conclude, in order to achieve accurate results from a trained network, the selected parameters involved must be evaluated and chosen carefully so that one doesn't blindly believe results from a network just because it's "trained".