Machine Learning
Jacobs University Bremen
Zihan Qi & Danni Long

Course: CO22-320372
May 26th, 2018

Homework 11

Procedure

Structure:
MLP: input: 3 units, hidden layer: 3 units, output: 1 units.

We started with 2 Class, one for hidden layer, one for output layer. Both of them have initial weight matrix size defined based on number of input number, and corresponding initial weight value, forward and back propagation functions. Hidden layer has sigmoid function for wrapping outside potential value from linear combination of activation values of input sample. Output layer has mean square error function, and a threshold function for computing the error between true values and the output of hidden layer after threshold, for further back propagation.

Concrete Implementation:
We have 4 input, each of which has 3 units, first place is a bias term -1. Then based on initialization, now we have a 3×2 matrix, so that linear combination of the every units contributes to 2 units in hidden layer. A sigmoid function was wrap outside the result. Then a bias term was added on top of the 2 units. Based on our initialization of output layer, we have a 3×1 weight matrix, to make linear combination of 3 units from hidden layer for the one output unit in output layer.
Now we finished forward pass, and got a result. It is a value in [0,1], then we applied the threshold function to determined it to be $\{0,1\}$.
Afterwards, output of a list of 0/1 will compare with target output, and compute the error. Since there was no sigmoid function when compute output units, we could directly have its partial derivative for gradient descent. Then update second weight matrix from its random initialization. Then for hidden layer, we compute the error from we had in output layer, and update its weight.
This whole process is one epoch, and we repeat doing it for 3000 times, in step size 0.1.

We plot the MSE in Case 1: use values after threshold to compute with target values. It will have spikes, since both of values only have 0/1. Around 2600 epochs, MSE convergent to 0 error. Case 2: direct use output values, compare with target values. (but still use the threshold do back propagation), the error will be more smooth. Since each time output is a continuous value in [0,1]. Around 2600 epochs, the MLP convergent to a small fixed error.

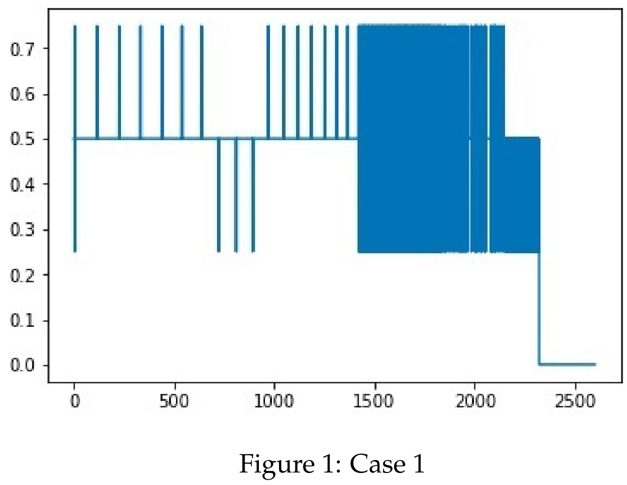

Figure 1: Case 1

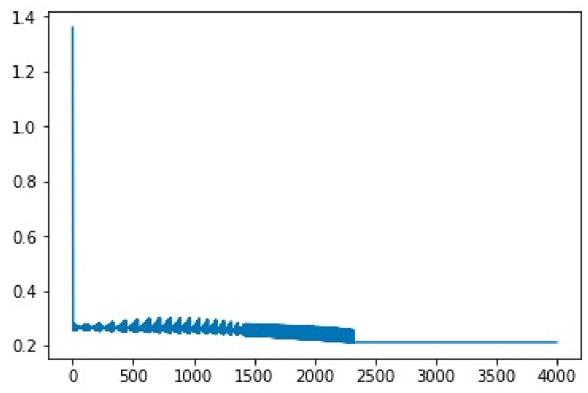

Figure 2: Case 2

Bonus:
We picked continuous version of XOR. We change activation function from sigmoid to tahn, since tahn maps value from [-1,1], which coincides with the sign function we are trying to approximate (But sigmoid will still do the work). The network wasn't convergent in the first place, we tune down the learning rate from 0.1 to 0.03, and added an addition unit in hidden layer, after 6000 epoch, the MSE convergence to a bit smaller than 1. We were suspicious about the error, but the output result are correct if you compare to the result from sign function. The reason why errors stay around 1 is what we believe cased by the threshold function we set. When doing back propagation, we let continuous value output which less than 0 to be -1, 0 to be 0 and larger than 0 to be 1. so as soon as the value over 0 a bit, it is recognized as 1, if the true label is 1 then the tuning process will leave this value alone. Therefore, If you look at output printed out, it is a vector of very small value around 0, yet they are correct. Our 2D contour plot has error, so we didn't include it.

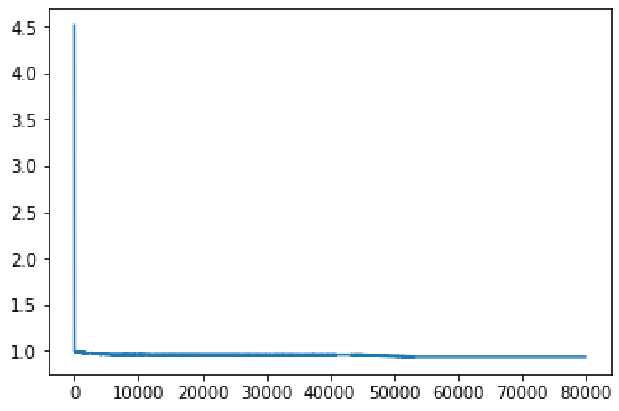

Figure 3: Bonus

Below is the case we choose learning rate to be 0.1, and using 2 units in hidden layer. You may observe that the network was not stable. so we tune down the learning rate, and magnitude is also decreasing.
Then we added more units in hidden layer, you could observe the MSE much more faster. In the case of hidden layer has 10 units, MSE convergence with epoch = 600.

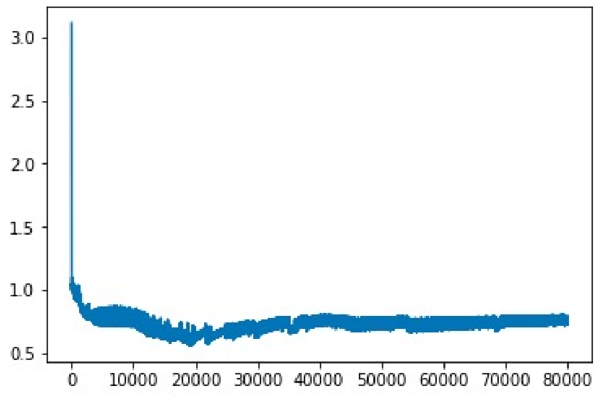

Figure 4: Bonus

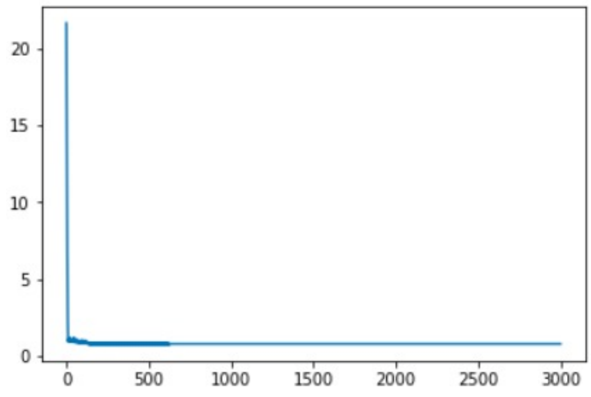

Figure 5: Bonus