Artificial Neural Network

Kaitaku Takeda, Kamesh Kantharaj

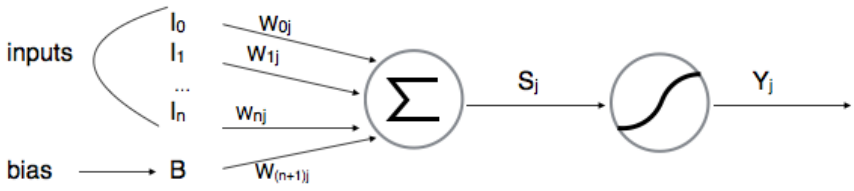
June 13, 2017

Abstract

We successfully built a simple *Artificial Neural Network* model trained to solve simple binary operations. Our graphical interface is built using p5.js and the underlying calculation is done using synaptic.js. The model illustrates chaotic behavior of output, given various values of input.

1 Introduction to Artificial Neural Network

Artificial neural network (ANN) is a computational model motivated by biological neural network of human brain. Neurons are smallest unit of the network, and are represented by nodes in ANN. Each node has inputs, an activation function, and output values. They are analogous to dendrites, nucleus, and axons of a human brain respectively. Connection between nodes have weights associated to them, which are randomly calculated in the initial setup. Each input value gets scaled by associated weights, and their sum gets fed to the activation function to produce an output value. Here is a simple diagram.



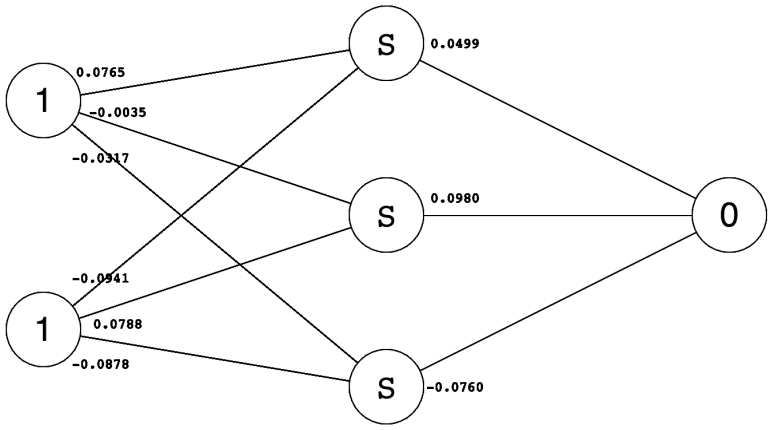
S_j in the diagram denotes the linear combination of all inputs with respective weights, and Y_j is the output of activation function on S_j. In our model we used the *Logistic Sigmoid Function* for this activation function.

2 Modeling

Our model initially illustrates the simplest network architecture known as *Feed-forward network*. Connections between nodes are unidirectional, therefore they

never form a cycle. The network can be trained using *back-propagation* algorithm, which is encapsulated in the Javascript library we used. In a single propagation, the network computes an output, compares it to the given *training set*, and makes justification to the weights to reduce the level of error. Our model allows modification of marginal error, maximal number of iterations (the number of times to be trained), as well as the learning rate.

The network we created learns how to execute XOR binary operation with given inputs. We were able to observe a variance in the output as we modified the values of iterations, error, and learning rate. Here is a model of binary operation on $(1, 1)$.



As you can see, each edge has an random weight associated to it. The initial value computed in this test case was 0.5036, which would actually display an incorrect value when rounded. However after 400 iterations of training, the output became 0.1655, which is fairly close to the target value of 0.

We then created the option to allow *feedback network*, where the computed output is routed back as an input, thus creating a cycle. In this network, we observed that computed output values slowly converge without any training performed. In contrast, the values oscillated when we trained the network.

3 Conclusion

Not only did the model provide a great visualization of an artificial neural network, but it also presented chaotic behavior depending on initial inputs. Neural network is a rather complex topic in mathematics and computer science. However, construction of such model helped us understand the basic flow of *Feed-forward network*, as well as a sample behavior of a *Feed-back network*.