Report: Implementation of Neural Networks from scratch

Saurav Garg

Implementing the blocks of a neural network

I have added a small constant for numerical stability in the following two functions:

1. `nn.computeLoss` I have used the cross entropy loss as $-y\log(p+10^{-7})-(1-y)\log(1-p+10^{-7})$. The 10^{-7} helps to avoid the Python error when either p or $1-p$ becomes very very close to 0.
2. `nn.backpropagate` There is a similar change in the derivative of cross entropy used for last layer i.e. $-y/(p+10^{-7})+(1-y)/(1-p+10^{-7})$

Applications of Neural Network

The topology, hyperparameters and seed values for each of the datasets are as follows:

1. **XOR Dataset**

   (a) Topology: The input is of size 2 nodes corresponding to x and y coordinate. This is connected to 4 nodes in a hidden layer using fully connected layer with softmax as activation. These 4 nodes are connected to the 2 output nodes using a second fully connected layer again with softmax activation.

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,4)
Fully Connected	softmax	(n,4)	(n,2)

   Here n is the number of input data samples.

   (b) Hyperparameters: Learning rate is 0.03, batch size is 10, and number of epochs is 30.

   (c) Seed values: 10 seed values from 0 to 9 (both included)

   The model achieved an average test accuracy of **99.3%** using the above. This is the minimal topology as is clear by the following.

   Test accuracies for other topologies:
   (using the same hyperparameters and avergaed over the same seed values)

   - Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,3)
Fully Connected	softmax	(n,3)	(n,2)

     Average accuracy: 81.24%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,2)	(n,3)
Fully Connected	softmax	(n,3)	(n,2)

Average accuracy: 78.15%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,2)
Fully Connected	softmax	(n,2)	(n,2)

Average accuracy: 56.47%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,2)	(n,2)
Fully Connected	softmax	(n,2)	(n,2)

Average accuracy: 73.16%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,1)
Fully Connected	softmax	(n,1)	(n,2)

Average accuracy: 49.77%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,2)	(n,1)
Fully Connected	softmax	(n,1)	(n,2)

Average accuracy: 60.35%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,2)

Average accuracy: 52.33%

Hence, we have shown empirically that smaller topologies give less than 90% accuracy.

Below is the decision boundary generated using the neural network. Here, the yellow and purple part represents the training data. The light blue and brown part represent the prediction for the purple and yellow data points repectively.

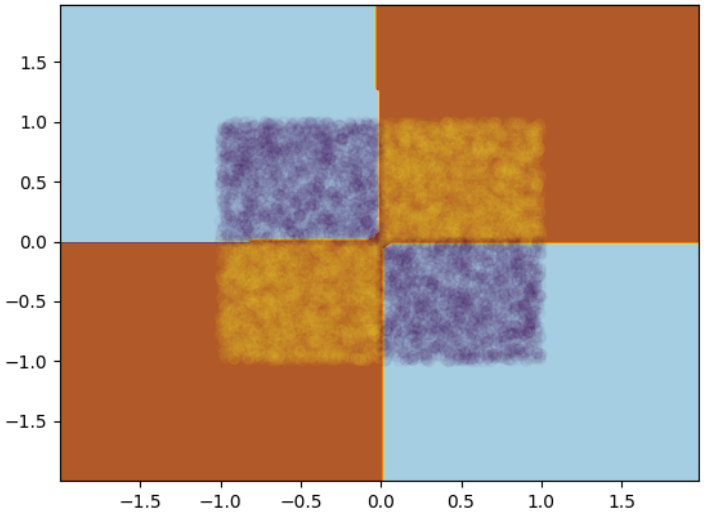

Figure 1: Decision boundary for XOR dataset

2. **Circle Dataset**

   (a) Topology: The input is of size 2 nodes corresponding to x and y coordinate. This is connected to 3 nodes in a hidden layer using fully connected layer with softmax as activation. These 3 nodes are connected to the 2 output nodes using a second fully connected layer again with softmax activation.

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,3)
Fully Connected	softmax	(n,3)	(n,2)

   Here n is the number of input data samples.

   (b) Hyperparameters: Learning rate is 0.03, batch size is 10, and number of epochs is 30.

   (c) Seed values: 10 seed values from 0 to 9 (both included)

   The model achieved an average test accuracy of **97.54%** using the above. This is the minimal topology as is clear by the following.

   Test accuracies for other topologies:
   (using the same hyperparameters and averaged over the same seed values)

   - Topology:

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,2)	(n,2)
Fully Connected	softmax	(n,2)	(n,2)

     Average accuracy: 90.58%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,2)
Fully Connected	softmax	(n,2)	(n,2)

Average accuracy: 81.55%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,2)	(n,1)
Fully Connected	softmax	(n,1)	(n,2)

Average accuracy: 73.42%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,1)
Fully Connected	softmax	(n,1)	(n,2)

Average accuracy: 49.77%

- Topology:

Layer	Activation	Input size	Output size
Fully Connected	softmax	(n,2)	(n,2)

Average accuracy: 77.69%

Here, although using 2 hidden nodes with relu as activation gives an average accuracy of 90.58% but if we use that topology, the accuracy is not consistent for different seeds. eg taking average test accuracy over 0-4 or 0-19 as seed values, both ends included, doesn't cross 90%. Also, if we take average over 0-99, the average test accuracy is less than 90%. Out of 0-99, for approx 60% of seeds, the model is able to achieve a high accuracy of more than 95% but for the rest 40%, it mostly varies 70-80%. So, I chose minimal topology as having 3 nodes which consistently gives a high accuracy above 95% for almost all the seed values. Hence, we have shown empirically that smaller topologies give less than 90% accuracy.

Below is the decision boundary generated using the neural network. Here, the yellow and purple part represents the training data. The light blue and brown part represent the prediction for the purple and yellow data points repectively. We can see that the network doesn't converge to a circular decision boundary. This happens because the data provided to the model is not a completer circle. And it achieves high accuracy even with this decision boundary.

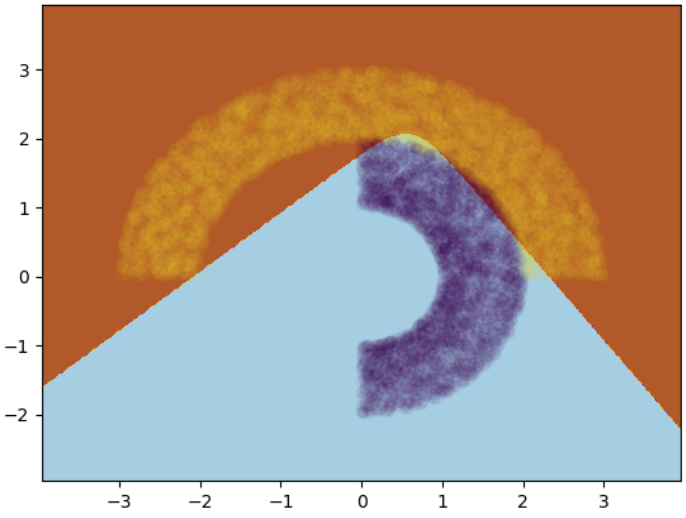

Figure 2: Decision boundary for circle dataset

3. **MNIST Dataset**

   (a) Topology: The input is of size 784 nodes corresponding to the flattened 28 x 28 size image pixels. This is connected to 28 nodes in a hidden layer using fully connected layer with relu as activation. These 28 nodes are connected to the 10 output nodes using a second fully connected layer with softmax activation.

Layer	Activation	Input size	Output size
Fully Connected	relu	(n,784)	(n,28)
Fully Connected	softmax	(n,28)	(n,10)

   Here n is the number of input data samples.

   (b) Hyperparameters: Learning rate is 0.03, batch size is 100, and number of epochs is 10.

   (c) Seed values: 10 seed values from 0 to 9 (both included)

   The model achieved an average test accuracy of **93.81%** using the above. Here, I decided the topology based on the reasoning that 1 hidden layer is sufficient for any function and tried a few values for hidden nodes like 16, 28, 32 etc and found that 28 was working good.

4. **CIFAR10 Dataset**

   (a) Topology: The input is of size 3 x 32 x 32 nodes corresponding to the 3 layers RGB each of size 32 x 32 pixels. This is passed through a convolution layer having 16 filters each of size 5 x 5 with relu as activation using stride equal to 1 resulting in an output size of 16 x 28 x 28. This is followed by a max pool of size 2 x 2 using stride equal to 2 resulting in an output size of 16 x 14 x 14. This is further passed through another convolution layer having 20 filters each of size 5 x 5 with relu as activation using stride equal to 1 resulting in an output size of 20 x 10 x 10. This is followed by a max pool of size 2 x 2 using stride equal to 2 resulting in an output size of 20 x 5 x 5. This is followed by a flattening layer resulting

in 500 nodes which are connected to 10 output nodes using a fully connected layer with softmax as activation.

Layer	Filter size	No of filters	Stride	Activation	Input size	Output size
Convolution	(5,5)	16	1	relu	(n,3,32,32)	(n,16,28,28)
Max Pooling	(2,2)	-	2	-	(n,16,28,28)	(n,16,14,14)
Convolution	(5,5)	20	1	relu	(n,16,14,14)	(n,20,10,10)
Max Pooling	(2,2)	-	2	-	(n,20,10,10)	(n,20,5,5)
Flatten	-	-	-	-	(n,20,5,5)	(n,500)
Fully Connected	-	-	-	softmax	(n,500)	(n,10)

Here n is the number of input data samples.

(b) Hyperparameters: Learning rate is 0.03, batch size is 100, and number of epochs is 40.

(c) Seed value: The provided model is trained using seed equal to 1. But I have verified the accuracy is above 35% for seeds 0 and 2 as well.

The model achieved the test accuracy of **47.5%** using the above (seed value 1). Here the network topology is decided based on the typical architecture given in the lectures. Two pairs of conv-pool followed by a fully connected layer. The convolution filter size and the max pooling size was chosen empirically. I have kept the number of epochs as 40 but the network gives test accuracy greater than 35% even on 20 epochs. I kept it high because the test accuracy increase by a small amount on more no of epochs.