CS155 Homework 4

Ty Limpasuvan

5 late hours used; 26 late hours remaining

1 Deep Learning Principles

1.1 A

1.1.1 i

The first link's network after roughly 250 epochs provided a test loss of 0.001 and a training loss of 0.001 and classified the points correctly, while the second link's network had a test loss of 0.508 and a training loss of 0.497 and didn't have a visible classifier for the data points. This difference is likely because the network in the second link started without any information in the neurons, unlike the network in the first link. The backpropogation algorithm starts processing with the final layer of neurons and works its way to the first. This resulted in weights of 0 for the neurons, which stay 0 because ReLU saturates at 0 by relying on the gradient.

1.1.2 ii

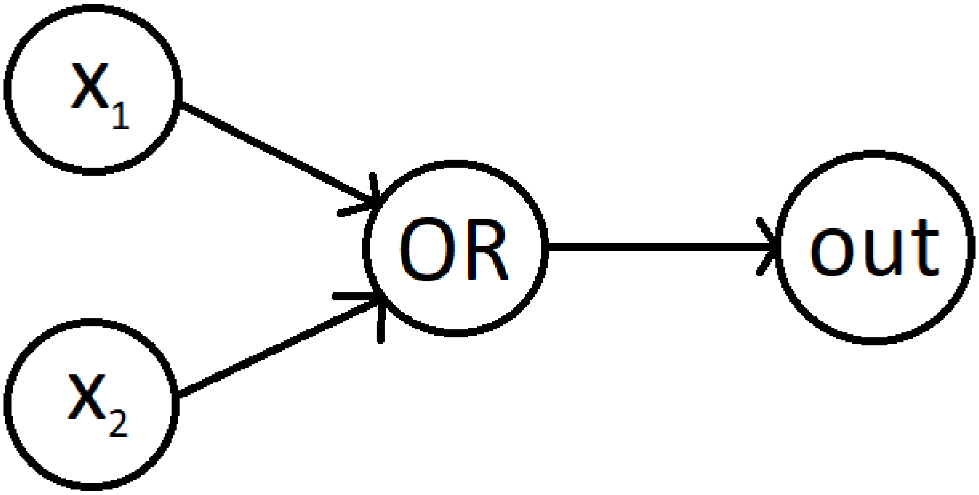
This time, the first link's network gave a test loss of 0.002 and a training loss of 0.001, while the second link had a test loss of 0.413 and a training loss of 0.397. This time, some form of classifier actually appeared in the window for the second network. The sigmoid function's derivative only goes to 0 at positive or negative infinity. But because the sigmoid function progresses logistically while ReLU progresses in a rectified linear fashion, more iterations were needed to train the network to classify the points correctly in the first network.

1.2 B

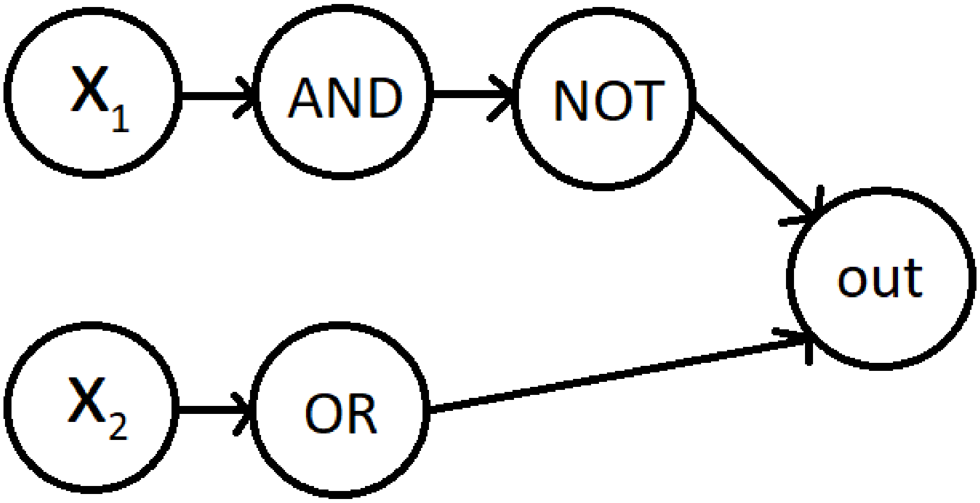
Looping through all of the negative points first with ReLU and SGD causes saturation at 0, which yields dead ReLU units. Then, the rest of the learning would only account for positive units, resulting in large test loss and training loss. When the gradient is that reliant on negative values, it is too far negative to correct itself and be accurate for the model.

1.3 C

1.3.1 i



1.3.2 ii



1 hidden layer is needed minimum (the NOT is not a layer, it just modifies the values by -1). The XOR function requires the and and the or functions to be used in the specified manner in order to replicate XOR. It is not possible to replicate XOR with fewer functions, so 1 layer is the minimum needed.

2 Depth vs Width n the MNIST dataset

2.1 A

I installed tensorflow-1.5.0 and keras-2.1.3

2.2 B

2.2.1 i

The images are 28x28 and the values in each, and there are several of these images. The first index in the array is which image and the other two indices refer to pixels.

2.2.2 ii

The new shape of the training input is (60000, 28, 28, 1).

2.3 C-E

Coding done in the notebook file.

3 Convolutional Neural Networks

3.1 A

The zero padding allows for the edges of the image to be processed like the rest of the image, but the zeros in those convolutions may cause the edges to appear differently. The convolutions would handle the color data in the picture but also account for the 0 information in the padding.

3.2 B

3.2.1 i

There are 27 different positions each convolution can take place in vertically, and another 27 horizontally. 27 * 27 = 729.

3.2.2 ii

The shape of the output tensor is 27 x 27 x 3.

3.3 C

3.3.1 i

$$\begin{bmatrix} 1 & 0.5 \\ 0.5 & 0.25 \end{bmatrix} \begin{bmatrix} 0.5 & 1 \\ 0.25 & 0.5 \end{bmatrix} \begin{bmatrix} 0.25 & 0.5 \\ 0.5 & 1 \end{bmatrix} \begin{bmatrix} 0.5 & 0.25 \\ 1 & 0.5 \end{bmatrix}$$

3.3.2 ii

$$\begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix} \begin{bmatrix} 1 & 1 \\ 1 & 1 \end{bmatrix}$$

3.3.3 iii

Pooling would reduce the impact of the noise. Taking the maximum of from a pool would avoid selecting noise, and averaging the pool would help account for the multiple angles.

3.4 D

For the 10 different values of the dropout probability, 0.11111111 gave the best accuracy (other than 0 and 1). I tried a range of different drop probabilities and found the one that seemed to maximize accuracy. I also found that adding the batch normalization to the model just before activating softmax helped the accuracy. I do not see a problem with validating the hyperparameters this way.