CSC321 Assignment 1

Zikun Chen

September 7, 2018

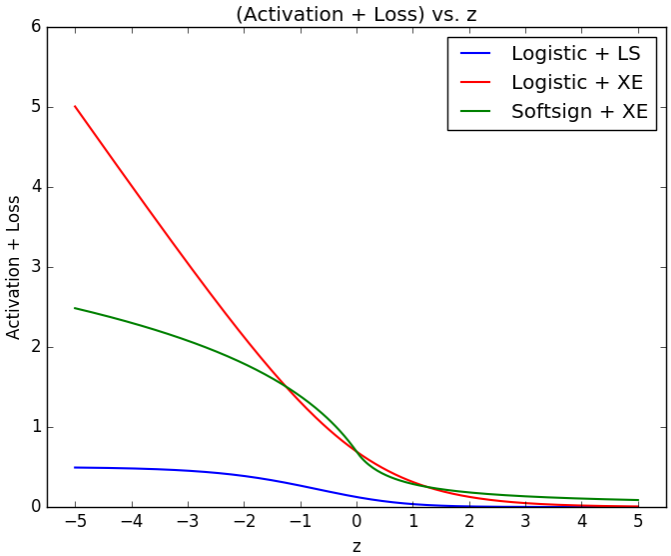
Part 1

Output:

```
dLdz = [ 0.72727273  0.1305405  -0.04729644 -0.51229508]
dEdW1[11, 487] = 0.010410179318
dEdW1[74, 434] = -0.010187383315
dEdb1[91] = -0.000362835119781
dEdb1[32] = -0.000438725272011
dEdw2[55] = 0.00952764403348
dEdw2[44] = -0.00958303219325
dEdb2 = 0.0113069930528
```

Part 2

1) **Graph of activation-loss functions in terms of z:**



Rate of descend: Based on the above plot, I think that the (logistic + cross-entropy) function would cause the training loss to decrease the fastest if we start from a very bad initialization (if z is initially very negative for a positive example). The slope of the function is the steepest when z is far out to the left, giving a very strong gradient signal to adjust the weights. On the other hand, the (logistic + squared error) activation-loss function would cause it to decrease the slowest for its lowest slope.

Result: The results supported my prediction. It shows that (logistic + cross-entropy) takes less epochs to reduce an initially large training error (double-digit). The first 20 epochs reduced the training loss to single-digit, which is really fast. On the other hand, it takes very long for (logistic + squared error) combination to make progress. The first 10 epochs only reduces around 0.001 training loss.

2) **Training Error:**

Based on the graph, I think that (logistic + cross-entropy) will get the lowest training error. Because with initial weights and biases that have small variability, (logistic + cross-entropy) will penalized any incorrect data points from the training set the most by giving the strongest gradient signal among the three (notice that it has a steeper slope than (softsign + cross-entropy) for most z), which will cause the model to fit the training set very well and very fast by the end of 1000 epochs.

On the other hand, the magnitude of the slope of (logistic + squared error) in terms of the linear prediction z is the lowest among the three, which means that it gives the lowest signal for gradient descent. So it would not fit the training set as well at the end of 1000 epochs because it is slower.

Validation Error:

When there is noise in the training data set, the first time the model encounter a mislabeled data, if the model has been fitting the correctly-labeled training data really well, it will penalize this data point harshly and adjust the weights to fit the training set better. It will continue to adjust away from the reality for the rest of the 10% noisy data. Consequently, the model will fit the training data nicely at the expense of losing the ability to perform well under the real validation set.
think that (logistic + squared error) will get the lowest validation error. Compared to the other two relatively faster converging activation-loss functions, (logistic + squared error) doesn't give as much of a gradient signal for mislabeled data. Therefore, it should perform the best when it comes to the validation set.

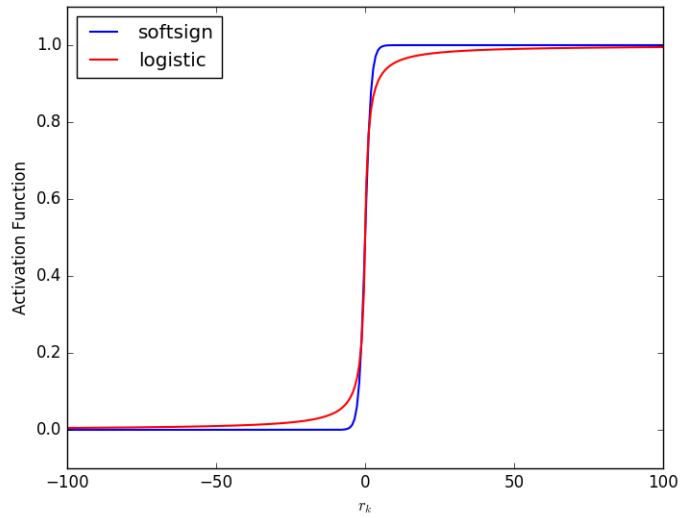
Result:

The result was that (logistic + cross-entropy) got the lowest training error, and (softsign + cross-entropy) got the highest.

And (logistic + squared error) got the lowest validation error.

Logistic+LS				
	1st run	2nd run	3rd run	average
training error	0.0825	0.08	0.0815	0.08133333
validation error	0.075	0.0725	0.075	0.07416667
Logistic+XE				
	1st run	2nd run	3rd run	average
training error	0.001	0.0005	0.001	0.00083333
validation error	0.0925	0.0975	0.0875	0.0925
Softsign+XE				
	1st run	2nd run	3rd run	average
training error	0.0105	0.011	0.013	0.0115
validation error	0.1025	0.0975	0.105	0.10166667

3) **Graph of softsign/logistic activation function in terms of r_k:**



I think that the logistic function is a better choice.
From the plot above, we can see that when the activation is close to 0 or 1, where r_k is large in magnitude, the slope of the logistic function is steeper than the softsign funciton, giving a stronger gradient signal, which means that the logistic function is more likely to get the saturated unit unstuck.