Building a Neural Network with Single Hidden Layer

CS 440/640 Programming assignment 1

My name: Chris-Emio Raymond

Teammates' names: Zhan Hao Xu, Ivan Mata

Date: October 9, 2019

Problem Definition

Give a concise description of current problem. What needs to be solved? Why is the result useful? Do you make any assumptions? What are the anticipated difficulties?

Overall Problem and Current Problem:

We were unable to train the neural network because we were unable to do back propagation. We overall did most of the build for the neural network, part of it is complete but not thoroughly tested because of the unfinished backpropagation method

Problems that needs to be solved:

The problems that were originally given to us were to: read a csv file to get the data from it, split it into 5 fold, and then trained and test out neural network using them. To train them we needed to run forward and backward propagation and change the weight to so the network could make correct decisions when tested

Why is the result useful?

Neural networks could prove that computers are able to process data of complex patterns through predictions and training as the basis of algorithms

Did we make any assumptions?

We assumed that all the data given would be simple. Simple as in just 0s and 1s. Each csv sheet would have the right amount of corresponding labels.

Anticipated Difficulties:

We anticipated that it would be hard to actually come up with an equation to implement forward and back propagation.

Method and Implementation

Give a concise description of the implemented method. For example, you might describe the motivation of your idea, the algorithmic steps of your methods, or the mathematical formulation of your method.

Methods:

Methods:

__Init__: Modified the function to make it easier to work on. Identified the number of input, out layers, and number of nodes. Furthermore, weights are randomly generated and defined regulation lambda.

Fit: We were suppose to run forward propagation, backpropagation to find the gradient. However, we could not finish it

Predict: Runs forward propagation only on a specific input to predict what the output would be.

Forward: First calculate layer 2 activation by getting the dot product of the X input and the layer 1 weights. Then calculate layer 2 activity by getting the sigmoid of layer 2 activation. After that calculate layer 3 activation by getting the

dot product of the layer 2 activity and the layer 2 weights. Lastly calculate the sigmoid of layer 3 activation and you'll have the y predictions.

sigmoid : Defined the sigmoid function as 1/(1+e^-x). The sigmoid function is useful because it reduces the number range to between (0,1), thus we use it to predict the probability of an output

Backward: Did not do

getCost: The purpose of this function is to compute the lost/cost in terms of cross entropy. However, we could not figure out how to correctly implement cross entropy. Instead, we implemented the Mean Squared error cost function.

Equation: $\frac{1}{n} \sum_{n=1}^{n} (Predicted - Desired)$

getData: This simply gets the names of the data sets. Looks for it in the current directory and reads the data on it using built in read csv method

Splitdata: Uses vsplit built in fuction to split the input matrix into even sets of K fold, assigns each of them an index. Makes sure each of the indices is chosen as a test set and separate the rest for the training sets.

Train: Initializes the neural network, calls fit on it to train it, and returns the trained model for testing. Also plots it using plotDecisionSummary which was given to us in the skeleton code.

Test: Test simply uses the train model to call predict on the test data set

getConfusionMatrix: It's pretty simple and based on the first homework. We tallied the total number of predictions, which is 100. Then we calculated which ones were true positives/negatives, and which ones were false positives/negatives. Then we put them in an array.

getPerformanceScore: The performance score uses the confusion matrix to calculate the accuracy, precision, recall, and F1 scores. We used the eq uations we learned from homework 1.

Briefly outline the functions you created in your code to carry out the algorithmic steps you described earlier.

Experiments

Describe your experiments, including the number of tests that you performed, and the relevant parameter values.

We ran our forward propagation method twice, each time on one set of 100 X's and Y's (Each time was a different set of 100 X's and Y's).

Define your evaluation metrics, e.g., detection rates, accuracy, running time.

Results

List your experimental results. Provide examples of input images and output images. If relevant, you may provide images showing any intermediate steps. If your work involves videos, do not submit the videos but only links to them.

Results

Trial	Source Image	Result Image
trial 1		
trial 2		
trial 3		

Discussion

Discuss your method and results:

- What are the strengths and weaknesses of your method?
- Our biggest weakness is the neural network is hardcoded for the amount of hidden layers that we have. It made the calculations simple and easy but it only works for a specific neural network

- Do your results show that your method is generally successful or are there limitations? Describe what you expected to find in your experiments, and how that differed or was confirmed by your results.
- Among the methods that we completed, we tested them individually to make sure that they do what they're meant to do. However, due to the fact that we didn't finish backpropagation, we never ran a full thorough test on the entire neural network. Our neural networks isn't being trained properly and therefore lacks accurate and significant results.

- Potential future work. How could your method be improved? What would you try (if you had more time) to overcome the failures/limitations of your work?
    - First we would need to finish implement backpropagation. Second, we could make it more universal when initializing the network so it isn't hard coded for 2 inputs, and 1 hidden layer. Third, run a full test to improve bad implementation of the currently finished methods

---

Conclusions

Based on your discussion, what are your conclusions? What is your main message?

-In conclusion, we needed to complete our testing to figure out if our neural network could be used for as a trustworthy system.

Credits and Bibliography

Cite any papers or other references you consulted while developing your solution. Citations to papers should include the authors, the year of publication, the title of the work, and the publication information (e.g., book name and publisher; conference proceedings and location; journal name, volume and pages; technical report and institution).

Material on the web should include the url and date of access.

Credit any joint work or discussions with your classmates.

- Deep learning, chapter 1-4 - 3Blue1Brown

- https://www.bogotobogo.com/python/scikit-learn/Artificial-Neural-Network-ANN-2-Forward-Propagation.php?fbclid=IwAR0DYowxQvFw-eXZLLgM6HzLT3P4ZbiDDVI_yzBCTOXk93ks7bsTlqXwjV8
- Class, lab, and lecture materials