GE 461—Introduction to Data Science
Project – Supervised Learning
Assigned: April 21, 2021
Due: 23:55, May 1, 2021

PART 1 [25 pts]:

Please provide a clear and solid answer for each question below. Explanations are required for your answers.

(a) **[5 pts]** How does the number of convolutional layers influence the learning power of deep neural networks?

(b) **[10 pts]** Naïve implementation of very deep convolutional neural networks may not learn from the data. What would be the main reason(s) of the aforementioned issue? How can you handle such problems?

(c) **[10 pts]** Do we observe issues similar to the ones indicated in part 1(b) for basic (vanilla) version of recurrent neural networks? Please explain/discuss, provide an example, and indicate possible solutions.

PART 2 [75 pts]:

Implement an artificial neural network (ANN) regressor and learn its weights implementing the backpropagation algorithm. Your implementation should support an ANN for linear regression (no hidden layer) and an ANN with a single hidden layer. In your implementation,

- Use the sum of the squared errors as your loss function,
- Use sigmoid as your activation function to define the hidden units, and
- Use the stochastic learning algorithm.

After implementing your ANN regressors, test them on the dataset provided on the course web page. The instances in this dataset have 1-D inputs and 1-D outputs (so that you will easily plot and see what your ANNs learn). The dataset is provided as two text files (train1, test1). An individual line of each file corresponds to an instance where the first number corresponds to the input and the next one corresponds to the output.

While testing your implementation, follow the steps below:

(a) First find the configuration that learns a network on its training instances. When selecting your configuration, you may want to answer the following questions. Note that each question might affect whether or not your network learns.

- Is it sufficient to use a linear regressor or is it necessary to use an ANN with a single hidden layer? If it is latter, what will be the minimum number of hidden units?
- What is a good value for the learning rate?
- How to initialize the weights?
- How many epochs should you use? How to decide when to stop?
- Does **normalization** affect the learning process for this application?

(b) After finding such a configuration, plot the actual outputs for the given input points. Then draw a curve for the outputs estimated by your selected model (on the same plot). While drawing this estimated output curve, do not just use the given input points, but draw a curve for data points uniformly selected in the range of your input points (so that you can see a smooth curve). Provide one plot for the training and another one for the test set.

Then, write the answers to the aforementioned questions in the following format.

ANN used (specify the number of hidden units):
Learning rate:
Range of initial weights:
Number of epochs:
When to stop:
Is normalization used:
Training loss (averaged over training instances):
Test loss (averaged over test instances):

(c) Now, analyze the effects of the network complexity to the model performance. To this end, for each dataset, use the linear regressor and ANNs with a single hidden layer. Use 2, 4, 8, 16, and 32 hidden units. Note that for each ANN, you may need to use a different configuration. That is, try different configurations (different learning rates, different ranges of initial weights, etc.) when learning these ANNs.

Afterwards,

- For each ANN, provide the plot as explained in (b), but this time, just for the training set. Additionally, for the data points uniformly selected in the range of your input points, draw a curve for each of the hidden units (put all these curves on the same plot). Thus, you will have a single plot for each ANN.
- Prepare a table that reports the training and test set losses. In this table, for each network, report the training loss averaged over the training set instances and its standard deviation. Likewise, for each network, report the test loss averaged over the test set instances and its standard deviation.
- Observe how the complexity (number of hidden units) of an ANN determines its ability to learn. Summarize your observations in 6-10 sentences

The second part of this project/homework asks you to implement a neural network regressor by **writing your own codes**. Thus, you are not allowed using any machine learning package. In your implementation, you may use any programming language you would like.

You are expected to write your report neatly and properly. The format, structure, and writing style of your report as well as the quality of the tables and figures will be a part of your grade. Use reasonable font sizes, spacing, margin sizes, etc. You may submit either a one-column or a double-column document. In your report, do not give any screen shots. Do not forget to address the questions specifically asked to you. Your report should be a maximum of 4 pages.

Please upload your report (as a pdf file) and your source codes in a single ZIP archive using the Moodle submission form.

Please notice that this is not a group work, each student has to submit an individual report/project.