CS550 - Machine Learning
Homework #2 - Artificial Neural Network Regressors

Ayhan Okuyan
ayhan.okuyan@bilkent.edu.tr
Bilkent University, Ankara, Turkey

I. Introduction

This homework required us to run regressor networks that also support linear regression to experiment on two datasets with one-dimensional input and outputs. Then we are required to optimize the networks either with or without hidden layers and comment on each, providing the necessary plots. For this implementation, we have decided to implement a general neural network framework that is able to run only ANNs using systematic matrix-based backpropagation. This way, we were able to configure the network in any way we wanted and was able to observe easily what each component represents and achieves. The implementation can be found under 'backend' section of the source code. The implementation coverage is as follows:

1) Initializers
   - Random Uniform
   - Truncated Normal
   - Zeros
   - Ones
   - Constant
   - Glorot Uniform
   - Glorot Normal
2) Activations
   - ReLU
   - Sigmoid
   - Tanh
   - Linear
   - Softmax (for completeness)
   - Leaky ReLU
3) Optimizers
   - Stochastic Gradient Descent (SGD)
   - Nesterov Momentum
   - Adaptive Momentum (ADAM)

For layers, only Dense layer is implemented and for loss options, only Mean Squared Error is implemented for regression purposes. Also, a batch generator is designed to feed the network with mini-batches during training. This homework is coded using Python 3, with using the **numpy, pandas, matplotlib and truncnorm from scipy.stats**.

II. Dataset

Before we moved on towards the implementation process, we have investigated the datasets. both datasets form a polynomial form as we were able to conceive, one 3^{rd} and the other 7^{th} degree. Also, we are able to observe when looking at both the training and test data, we see that the training data mostly covers the variance on the test data. There are 60 training and 41 test samples in dataset 1, while there is 229 training and 92 test instances in dataset 2. After observing we move on towards the implementation of the network. Throughout this homework labels are normalized as well as the features to improve the speed of convergence, and the losses based on the normalized versions are reported.

III. Part A

For the first data, below is the training and test curves.

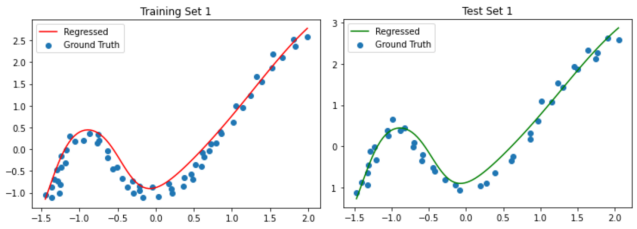

Fig. 1. Training and Test Data for Dataset 1 with the Regression Curves

ANN used (specify the number of hidden units): ANN with 30 Hidden Units
Selected activation function: Sigmoid
Selected loss function: MSE
Learning rate: 0.01
Range of initial weights: Each layer is specified with their own range. Ultimately, the Glorot Normal is used, which samples weights from a truncated normal distribution (2 standard deviations of truncation) with zero mean and a standard deviation of $\sqrt{\frac{6}{fan_{in}+fan_{out}}}$, where fan_{in} and fan_{out} are the number of inputs to a neuron and the number of output neurons respectively. Ultimately, all weights are sampled from a normal distribution of $\mathcal{N}(0, \sqrt{\frac{6}{31}})$, biases are initialized to zeros
Number of epochs: 5000
When to stop: Number of epochs are presented as the stop
Is momentum used (if so, value of the momentum factor): Yes, Adaptive Momentum is utilized for training with moments, presented on the original paper [1], $\beta_1 = 0.9 and \beta_2 = 0.999$
Is normalization used: Yes, it is used for the features, as well as the labels. Both parts of the test set are normalized with the mean and the standard deviations of their corresponding training counterparts.
Stochastic or batch learning: Mini-batch learning with a batch size of 15
Training loss (averaged over training instances):
Test loss (averaged over test instances):
The plots for the second data is as follows.

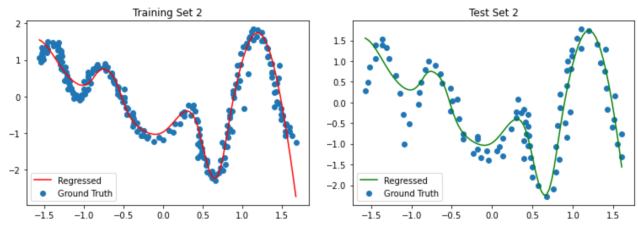

Fig. 2. Training Data for Dataset 1 with the Regression Curves for the Given Hidden Units

ANN used (specify the number of hidden units): ANN with 60 Hidden Units
Selected activation function: Sigmoid
Selected loss function: MSE
Learning rate: 0.001
Range of initial weights: Glorot Normal is used, meaning, weights are sampled from $\mathcal{N}(0, \sqrt{\frac{6}{61}})$, biases are initialized to zeros.
Number of epochs: 50000
When to stop: Number of epochs are presented as the stoppig criteria
Is momentum used (if so, value of the momentum factor): Yes, Adaptive Momentum is utilized for training with moments, presented on the original paper [1], $\beta_1 = 0.9 and \beta_2 = 0.999$
Is normalization used: Yes, it is used for the features, as well as the labels. Both parts of the test set are normalized with the mean and the standard deviations of their corresponding training counterparts.
Stochastic or batch learning: Mini-batch learning with a batch size of 15
Training loss (averaged over training instances):
Test loss (averaged over test instances):

IV. Part C

This part required us to do some series of experiment with both datasets. We were required to set the number of hidden units for both sets to $\{2, 4, 8, 15\}$ and the linear regression ANN and optimize them all, observe the results. The results on the 1^{st} dataset is as follows.

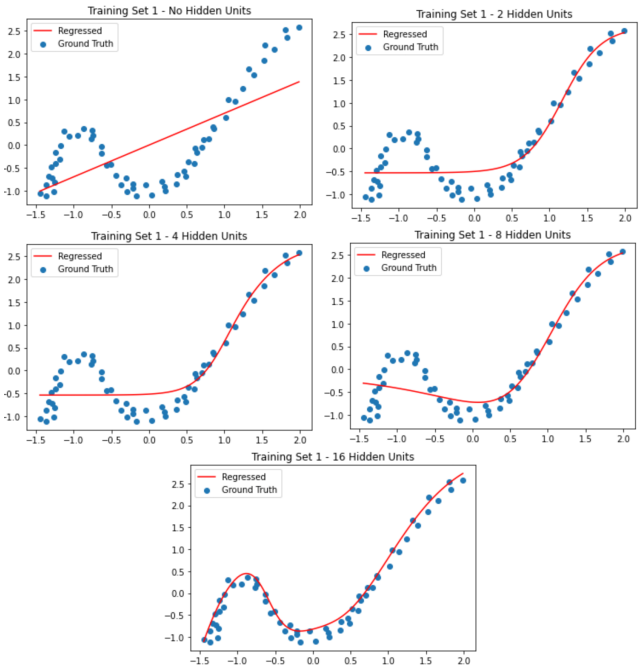

Fig. 3. Training and Test Data for Dataset 2 with the Regression Curves

The average loss and standard deviations are given below.XXX The results on the second set is provided below.

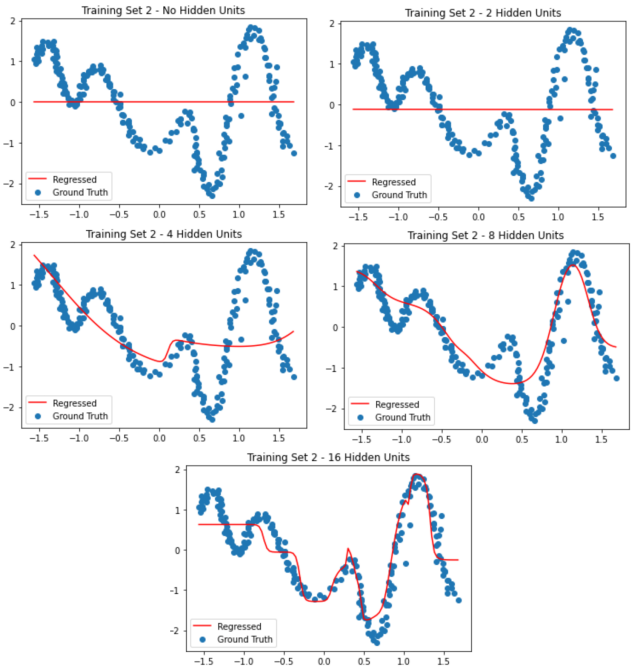

Fig. 4. Training and Test Data for Dataset 2 with the Regression Curves

The average loss and standard deviations are given below.XXX Here, we can observe the the number of neurons in the hidden layer determine the capacity of the network. Put more mathematically, as the number of hidden units increase the complexity of the decision increases. This can be concretely observed in both results, as the number of hidden units increase, the network learns more curvature and the loss gets better.

V. Part D

References

[1] D. P. Kingma and J. Ba, *Adam: A method for stochastic optimization*, 2017. arXiv: 1412.6980 [cs.LG].