HW

Neural network design using Keras and Pytorch

You can use the code in our drive as a template. Upload your codes and your results as a report. Answer all the questions for both Keras and Pytorch implementations. Also analyze if there are any performance differences between both the implementations.

1. Read the datasets

* Regression: Use the Boston Housing dataset.
* Classification: Use the Iris dataset.

NET1: Baseline Network

* For regression: Design a 2-layer neural network with 2 hidden layers, each having 64 neurons.
* For classification: Design a 2-layer neural network with 2 hidden layers, each having 10 neurons.

Do :

* 1. Complete functions of the keras\_dnn.py file.
  2. Complete functions of the torch\_dnn.py file.
  3. Submit both to gradescope.

Train the network using the train data set. Test the network using the test data set.

Questions:

1. Compare the performance (test loss, accuracy, and training time) of the network using different activation functions (ReLU, Tanh, Sigmoid). Explain any observed differences.
2. Compare the performance (test loss, accuracy, and training time) of the network using different optimizers (SGD, Adam, RMSprop). Explain any observed differences. Be sure to include analysis of the SGD optimizer for the torch framework.
3. **NET2 - Increased Neurons**: Increase the number of neurons in each hidden layer to 128 and train the network using the same experimental setup as NET1. How does increasing the number of neurons affect the performance of the network? Compare the results of NET1 and NET2.
4. NET3 – Additional Hidden Layer: Add an additional hidden layer with the same number of neurons as the first hidden layer in NET1. Train the network using the same experimental setup as NET1. How does adding an additional hidden layer affect the performance of the network? Compare the results of NET1 and NET3.