**DSC 275/475: Time Series Analysis and Forecasting (Fall 2021)**

**Project 2 – Feedforward Neural Networks**

**Total points: 30 (undergraduate students); 40 (graduate students; and with extra credit for undergraduate students)**

***Instructions:***

* You are welcome to work on this project individually or in teams (up to 2 members in each team max).
* If you plan to use PyTorch, a good resource is to review and modify the attached example code (we plan to review this example code in class as well) for feedforward networks. For this project, you will need to modify the following sections of the example code for each question:
  + Network configuration (number of layers and neurons) in the class “Net”
  + Activation function (“relu” is recommended)
  + Decision boundary limits in the function “plot\_decision\_boundary” as per each question
  + Learning rate “learning\_rate”
  + Number of epochs and/or apply a stopping condition when the desired accuracy and/or loss is achieved
  + Optimizer: “torch.optim.SGD” or “torch.optim.Adam”. The latter may provide faster convergence

**Overview**

In this project, you will create and train feedforward neural networks on synthetic, separable data. Although traditional machine learning tasks involve measuring performance on sets of data not seen by the network during training (e.g., test data), in this project we will mostly be concerned with architecture, hyperparameter and learning parameter tuning to maximize performance on the training set. This is an often-overlooked aspect of machine learning: practitioners seldom verify their algorithms have the right capacity for the data by examining their behavior on the training set. In this project, all performance metrics of interest will be computed on the training data.

In order to construct networks that are capable of performing the tasks being posed, you will need to adjust parameters such as learning rate, type of network activation function, number of layers and number of neurons per layer. Lastly, you will need to monitor the loss and the accuracy on the training set as the learning takes place.

Please make sure to hand in a pdf/Word/text document with the final version of your code as well as answers to the questions below. In addition, please submit the code file also. Points highlighted in red denote problems *required* for graduate students and *extra credit* optional submissions for undergraduate students.

1. Classification of XOR data (15 points)

At first sight, it may seem as if separating the XOR data is a simple task. However, due to the fact that the data is not linearly separable and that fitting the data requires non-trivial learning, the XOR problem has been a case study of interest on many topics related to training of feedforward networks. In the 1960s, Minsky and Papert’s observations that perceptrons (neural network ancestors) were unable to fit the XOR data contributed to the rise of the first AI winter. XOR data makes for such an interesting case study that papers describing learning properties of networks trained on it are still being published to this day.

Create arrays containing the input data and the corresponding output labels for the XOR operator. Recall that XOR takes as input two binary variables, and outputs a 0/1 if they have the same/different value. Your XOR data should look like the following table:

|  |  |  |
| --- | --- | --- |
| ***x*1** | ***x*2** | ***y*** |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

In other words, the XOR training includes four, two-dimensional data samples with labels. Create and train a network with at least three hidden layers that separates the XOR data, that is, a network that gets 100% performance on the four training samples above. Monitor the accuracy on the training set as the training progresses.

1. Plot the decision boundaries of the earliest network in the training process that achieves 100% accuracy by plotting the network outputs in a densely sampled region around [-0.5,1.5] × [-0.5,1.5]. *(5 points)*
2. Plot the decision boundaries of a network after the loss falls below 1×10-4. *(5 points)*
3. Gradually decrease the capacity of the network above. Find the smallest network that can still separate the data, i.e., find the least number of hidden layers and neurons that produces an accuracy of 1 on the training set? *(5 points) [A portion of the total points is allocated to your rank amongst your peers in achieving the smallest network]*
4. Classification of Separable, Synthetic data (15 +10 points)

In this section, you will attempt to design a network that is able to classify synthetically created data that is still separable, that is, where no overlap exists between the classes in the native feature space. In this case, however, the tolerances (i.e., the smallest separation between samples in different classes) are much tighter than those seen in the XOR case.

* 1. File Feedforward\_Data\_ellipse.csv contains 13312 two-dimensional data points (feature values located in columns 1 and 2) and their respective binary label (labels located in column 3). Create and train a network that separates the data. Report your best loss and accuracy values. Plot the decision boundaries of your best network by plotting the network outputs in a densely sampled region around [-1.0,1.0] × [-1.0,1.0]. Report the number of hidden layers, type of activation function and number of neurons per layer used. *(15 points)*. *[A portion of the total points is allocated based on your rank amongst your peers in achieving the best loss and accuracy values]*
  2. File Feedforward\_Data\_hexa.csv contains 13312 two-dimensional data points (feature values located in columns 1 and 2) and their respective binary label (labels located in column 3). Create and train a network that separates the data. Report your best loss and accuracy values. Plot the decision boundaries of your best network by plotting the network outputs in a densely sampled region around [-1.0,1.0] × [-1.0,1.0]. Report the number of hidden layers, type of activation function and number of neurons per layer used. *(10 points)* *[A portion of the total points is allocated based on your rank amongst your peers in achieving the best loss and accuracy values]*