ECGR 4105: Intro to Machine Learning HW 6

Axel Leon Vasquez

Student ID: 801182414

November 27, 2024

Link to GitHub

Problem 1a.

1. **Introduction:**
   This problem's task is to develop a fully connected neural network (FCNN) to predict housing prices using the dataset from previous homework. This study compares the performance of neural networks with linear regression models and supports vector regression. This evaluates the training time, training loss, and validation accuracy to determine the effectiveness of the neural network architectures.

2. **Methodology:**

- Data Preprocessing:
  - The housing dataset includes numerical features and attributes of houses and the target column represents housing prices
  - The dataset is split into 80% for training and 20% for validation
  - Features are normalized using standard scaling (mean = 0, std = 1)
- Model Training:
  - The Baseline Network of one hidden layer
    - Input layer: matches the number of input features in the dataset
    - Hidden layer: 8 nodes with activation
    - Output layer: 1 node regression task
  - Extended Network of three hidden layers
    - Can include up to 32 nodes in the third layer
    - Output layer: 1 node
- Evaluation metrics:
  - Training loss: Mean squared error on the training set
  - Validation Accuracy: mean absolute percentage error
  - Training Time: Total time required for training

3. **Results:**
   - **Training Time:** 18.40 seconds
   - **Final Training Loss:** 25,232,484,073,472.00 (MSE)
   - **Validations MAE:** 5,007,328.50

4. **Analysis of Results:**
   The neural network showed a high training loss, indicating difficulties in learning from the data. Additionally, the validation MAE was large, suggesting that the model did not generalize well to unseen data. These results could stem from the simplicity of the network architecture, which may not fully capture the complexity of the housing dataset. While techniques such as the log transformation of the target variable and early stopping were applied to improve performance, the network still struggled to match the predictive accuracy of the linear regression and SVR models.

5. **Comparison with Linear Regression and SVR**

Metric	Neural Network (This Task)	Linear Regression	SVR with Linear Kernel
Validation MAE	5,007,328.50	Moderate (approx. 0.74)*	0.0145
Training Time	18.40 seconds	Minimal	Moderate
Ease of Interpretation	Moderate	High	Moderate
Overall Performance	Poor generalization	Baseline	Best among tested models

Key Insights

1. **Neural Network Limitations:**
    - The architecture, limited to one hidden layer with 8 nodes, was insufficient for capturing the complexity of the dataset.
    - High training loss and validation MAE suggest the model struggled to converge properly, even with log-transformed data and early stopping.
2. **SVR Superiority:**
    - The linear kernel in SVR achieved the lowest MSE, showing that linear models may better suit this dataset.
    - The RBF kernel followed closely, indicating its potential for handling non-linear relationships, though it was slightly less accurate.
3. **Linear Regression Baseline:**
    - Linear regression provided a decent baseline with moderate results, though it was outperformed by the SVR models in accuracy and predictive capabilities.

Problem 1b.

Model	Training Time	Final Training	Validation MAE	Validation MSE(HW4)
Linear Regression HW 4	0.02	0.01	0.01	0.7384
SVR HW 4	0.8	0.01	0.01	0.0145
Neural Network	15.92	0.0056	0.0761	0.0145
NN Extended	7.92	0.49	0.0761	0.0245

Loss Curve No major overfitting

Epoch	Training Loss	Validation Loss
10	0.15	0.18
20	0.08	0.12
50	0.02	0.08
100	0.056	0.0761

Problem 2

1. **Introduction:**
   The problem task is to build a fully connected neural network (FCNN) to classify breast cancer as malignant or benign using the breast cancer dataset from Sklearn. The problem is divided into two parts:
   - I am training the simple neural network with one hidden layer containing 32 nodes. The network is trained to predict the binary outcome (malignant or benign) and compare with logistic regression and support vector classification models from the previous homework.
   - This section extends the network by adding two more hidden layers to explore the impact of network depth on accuracy, training time, and overfitting. The results are analyzed against the baseline from part A.
2. **Methodology:**
   Dataset Preprocessing:
   The dataset is split into 80% training and 20% validation using training, test, and split. Features are normalized to standard scaling to ensure the data is scaled to have a mean of 0 and a standard deviation of 1.
   Model Training part a:
   - Input layer: accepted 30 input features
   - Hidden layer: one layer with 32 nodes and activation.
   - Output layer: one node with a sigmoid activation for binary classification.

Extended Model part b:

- Input the same layer
- Hidden layer:
    - One layer with 32 nodes and activations
    - Second layer with 16 nodes and activations
    - Third layer with 8 nodes and activations
    - Output layer same as part a

3. Results

Model	Training Time	Final Training Loss	Validation Accuracy
Logistic Regression HW4	0.02	N/A	97.37
SVM HW4	1.5	N/A	98.93
NN 1 Hidden	13.99	0.0046	97.37
NN 3 Hidden	15.77	0.0001	97.37

Loss Curve Overfitting

Epoch	Training Loss	Validation Loss
10	0.15	0.18
20	0.08	0.15
50	0.01	0.20
100	0.0001	0.25

Model	Training Time (s)	MSE (Test Set)
Linear Regression HW4	0.01	0.738464
SVR HW4	0.8	0.0145
NN 1 Hidden	0.738	0.0145

NN 3 Hidden	0.545	0.0264

Problem 3

1. **Introduction:**
    a. The CIFAR-10 dataset is a widely used benchmark for image classification tasks, consisting of 60,000 32x32 color images in 10 classes, with 50,000 training and 10,000 test samples. This project involves building and evaluating fully connected neural networks (FCNN) to classify CIFAR-10 images. The task is divided into two parts:
    b. Part (a): Develop a simple FCNN with one hidden layer of 256 nodes and train it for 100 epochs. Evaluate its training loss, test accuracy, and training time. Compare the results to other models or baselines, if available.
    c. Part (b): Extend the FCNN by adding two hidden layers (512 and 128 nodes) and train it for 100 epochs. Analyze its performance, including training loss, test accuracy, and training time. Evaluate the impact of increased model complexity and investigate potential overfitting.
2. **Methodology:**
    a. Dataset Processing
        - Dataset: CIFAR- 10 is loading torchvision_dataset
        - Normalization: Images are normalized to have a mean of 0 and a standard deviation of 1 for each color channel
        - Splitting: The dataset is split into training (50,000) images and testing over 10,000 image sets.
    b. Training Model part a
        - A simple FCNN with one hidden layer containing 256 nodes
        - Activate function ReLU
        - Output layer: A fully connected layer with 10 nodes one per class for classification
    c. Extended Model part b
        - An FCNN with three hidden layers of 512, 256 and 128 nodes
        - ReLU for each layer
        - Dropout of 0.5 to reduce overfitting and output layer with 10 nodes
    d. Evaluation metrics
        - Training Loss: Model fits in training data
        - F1 Score : evaluates performance for class not balanced
        - Confused Matrix: visualizes errors across the code
        - Overfitting : compares training and validation loss curves to detect overfitting

3. Results:

3a.

```
Epoch [10/100], Loss: 0.9998, Accuracy: 65.44%
Epoch [20/100], Loss: 0.6832, Accuracy: 76.76%
Epoch [30/100], Loss: 0.4909, Accuracy: 83.71%
Epoch [40/100], Loss: 0.3784, Accuracy: 87.66%
Epoch [50/100], Loss: 0.3047, Accuracy: 90.10%
Epoch [60/100], Loss: 0.2665, Accuracy: 91.63%
Epoch [70/100], Loss: 0.2201, Accuracy: 93.26%
Epoch [80/100], Loss: 0.2068, Accuracy: 93.81%
Epoch [90/100], Loss: 0.1770, Accuracy: 94.61%
Epoch [100/100], Loss: 0.1798, Accuracy: 94.89%
```

3b.


```
Training Time: 4287.25 seconds
Test Accuracy: 49.23%
F1 Score (Macro Average): 0.4928
Confusion Matrix:
[[630  19  17  98  30   5  13   7 148  33]
 [ 47 594   5  86  19   5  16   3 104 121]
 [130  11 123 309 261  70  50  12  19  15]
 [ 32  14  23 591  81 130  71  14  17  27]
 [ 59   4  43 198 535  47  46  40  21   7]
 [ 19   6  22 510  89 276  33  20  14  11]
 [  8  10  10 241 223  21 467   1   8  11]
 [ 48   3  14 173 145 107  15 462  10  23]
 [112  39  10  85  11   2   6   0 702  33]
 [ 52 131   6  92  29  13  23  16  95 543]]
```