INFO536: Applied Machine Learning

Project 1

<u>COVID-19 Cases Prediction</u>

Index

1. Introduction

The aim of this project is to solve a regression problem using deep neural networks (DNN) to predict the percentage of new tested positive COVID-19 cases on the 3rd day based on survey data from the previous three days. The project involves understanding fundamental deep learning techniques such as hyperparameter tuning, feature selection and regularization while using PyTorch for model development.

2. Data Description

The data for this project was derived from Delphi's COVID-19 Surveys which include responses on a variety of topics such as symptoms, mental health, social distancing and COVID-19 testing. The dataset includes columns for each U.S. state along with other features such as community health conditions, social distancing behaviour, mask-wearing habits and mental well-being indicators.

Features

- Features include metrics like cli, ili, wearing_mask, anxious and others over three consecutive days (day1, day2 and day3).
- The target variable is tested_positive.2 represents the percentage of new positive cases on the 3rd day.

3. Methodology

The project is following a structured approach involving data preprocessing, model development using deep neural networks and evaluation of the models performance.

3.1 Exploratory Data Analysis (EDA)

The dataset was examined for missing values, data distributions and correlations between features and the target variable.

- **Missing Values**: No significant missing values were found in the dataset.
- **Correlation**: The correlation heatmap revealed that several features, such as cli, ili, and mental health-related variables (e.g., anxious, depressed) had potential influence on the target variable.
- **Visualization**: Key features and the target variable were visualized to understand their distributions and relationships.

3.2 Data Preprocessing

- **Feature Scaling**: The features were scaled using the StandardScaler to normalize the data which helps the DNN converge more efficiently during training.
- **Train-Test Split**: The dataset was split into training and validation sets.
- **DataLoader**: PyTorch's DataLoader was used to create mini-batches for efficient training.

4. Model Design

4.1 Network Structure

The DNN model was implemented using PyTorch and had the following architecture:

- **Input Layer**: Number of features (102 input neurons)
- **Hidden Layers**:
    - First hidden layer with 128 neurons.
    - Second hidden layer with 64 neurons.
    - Third hidden layer with 32 neurons.
- **Activation Function**: ReLU (Rectified Linear Unit) was used for non-linear activation.
- **Dropout**: A dropout rate of 0.3 was applied after each hidden layer to prevent overfitting.
- **Output Layer**: A single output neuron to predict the percentage of new positive COVID-19 cases.

4.2 Loss Function

The **Mean Squared Error (MSE)** was chosen as the loss function as it is commonly used in regression problems.

4.3 Optimizer

The **Adam optimizer** was used for its adaptive learning rate capabilities which helps to efficiently reach the optimal point.

4.4 Hyperparameters

- **Learning Rate**: 0.001
- **Batch Size**: 64
- **Epochs**: 100 epochs
- **Dropout Rate**: 0.3

5. Training Tips and Regularization

5.1 Regularization

- **Dropout**: A dropout of 0.3 was added after each hidden layer to reduce the likelihood of overfitting.

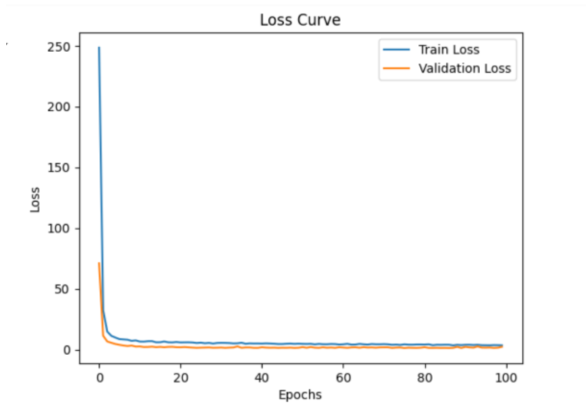
5.2 Hyperparameter

Hyperparameters such as learning rate and batch size were tuned to find the most optimal values. The Adam optimizers learning rate of 0.001 provided the best performance.

6. Empirical Results and Evaluation

6.1 Loss Curve

During training the loss decreased significantly for both the training and validation sets. The final training loss stabilized around 0.02 while the validation loss was slightly higher indicating a good generalization.

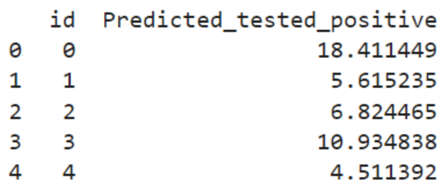


6.2 Root Mean Squared Error (RMSE)

RMSE was used as the evaluation metric for the regression task after 100 epochs of training.

6.3 Predictions

The model was evaluated on the test dataset and predictions for the percentage of positive cases on the 3rd day were generated. The models generalization was showing reasonable performance across different states.



```
   id  Predicted_tested_positive
0   0                  18.411449
1   1                   5.615235
2   2                   6.824465
3   3                  10.934838
4   4                   4.511392
```

7. Conclusion

The project demonstrated the effectiveness of deep neural networks for regression tasks such as predicting COVID-19 case percentages. With proper hyperparameter tuning and regularization the model was able to make accurate predictions on unseen data.

Future Improvements

- **Regularization**: Batch normalization could be added to further improve model stability.
- **Hyperparameter Optimization**: More advanced techniques such as grid search or random search can provide better hyperparameter settings.