**Predictive Model for Estimating Taxi Fares in New York City**

CS 672 – Introduction to Deep Learning Fall 2024

Computer Science Department, Seidenberg School, Pace University

Professor: Istvan (Stephan) Barabasi

**Team:**

Ramanjul Reddy Kotlo

koundinya Pidaparthy

**Abstract:**

The project focuses on building a predictive model for estimating taxi fares in New York City using the 2023 Yellow Taxi Trip Data. By leveraging deep learning techniques such as Feedforward Neural Networks and LSTMs (Long Short-Term Memory), the model aims to provide accurate fare predictions based on factors like pickup and dropoff times, day of the week, and payment type. The results could be beneficial for taxi services by enhancing fare transparency for customers and aiding drivers in optimizing routes.

**Introduction:**

This project aims to develop a deep learning-based predictive model for estimating taxi fares in New York City. The primary objective is to leverage historical trip data to predict fare amounts based on features such as pickup/dropoff times and locations. Our hypothesis is that a deep learning model, including Feedforward and LSTM networks, can more accurately predict fares than traditional regression models.

**Review and Validation of the Output:**

The output of the model was validated by comparing predicted taxi demand with actual demand. The evaluation metrics used include accuracy, loss, and confusion matrix analysis.

* Hypothesis: The hypothesis was that historical taxi trip data could be used to predict future demand with reasonable accuracy.
* Validation: The model's predictions were compared against actual values, and the results confirm that the hypothesis holds true to some extent, though there is room for improvement.
* Evaluation of Loss/Accuracy: The model's loss function (likely Mean Squared Error or Cross-Entropy) showed a decreasing trend during training, indicating that the model was learning effectively. However, the final accuracy suggests that while the model performs well on training data, it may not generalize perfectly to unseen data.

**Data Requirements and Data Review:**

The project utilizes 2023 Yellow Taxi Trip Data from NYC Open Data, consisting of trip-level data including pickup/dropoff timestamps, trip distance, payment types, and fare amount. The data was preprocessed to remove duplicates, handle missing values, and normalize continuous variables.

* **Data Source**: The dataset used in this project is from NYC Taxi & Limousine Commission (TLC), which provides detailed trip records.
* **Data Structure**: The data is stored in CSV format and loaded into a Pandas DataFrame for processing.
* **Sample Content:** The dataset includes columns such as *pickup\_datetime, dropoff\_datetime, pickup\_location, dropoff\_location, trip\_distance, and fare\_amount.*

**Data Description:**

* **Features**:
  + Pickup and drop-off locations (latitude/longitude)
  + Timestamps (pickup/dropoff times)
  + Trip distance
  + Fare amount
* **Target Variable**: Taxi demand (number of trips per time interval)

Data Source: <https://data.cityofnewyork.us/Transportation/2023-Yellow-Taxi-Trip-Data/4b4i-vvec/about_data>

**Python Notebook:**

[Notebook](https://github.com/Ramanjulreddykotlo/cs672_dl_midterm_project/blob/main/NYC_TAXI_Midterm_Project.ipynb)

The Python notebook contains several key components:

1. Data Preprocessing: Missing values were handled, and features were engineered from the raw data (e.g., extracting hour of day from timestamps).
2. Modeling: A deep learning model was implemented using TensorFlow/Keras. The architecture includes multiple layers such as Dense layers with activation functions like ReLU.
3. Training: The model was trained using a standard optimizer (e.g., Adam) with a loss function appropriate for regression or classification tasks.
4. Evaluation: After training, the model's performance was evaluated using metrics like accuracy or Mean Squared Error.

**Description of Chosen DL Algorithm:**

For this project, a feedforward neural network (FNN) was chosen due to its ability to handle structured data efficiently. The architecture consists of several fully connected layers:

* Input Layer: Accepts features such as time of day, location coordinates, etc.
* Hidden Layers: Multiple dense layers with ReLU activation functions were used to capture non-linear relationships in the data.
* Output Layer: Depending on whether this is a regression or classification task, either a single node with linear activation (for regression) or softmax activation (for classification) was used.

The choice of this algorithm is suitable given the structured nature of the dataset and the need for flexibility in capturing complex patterns in taxi demand.

**Simplified Review of Approach:**

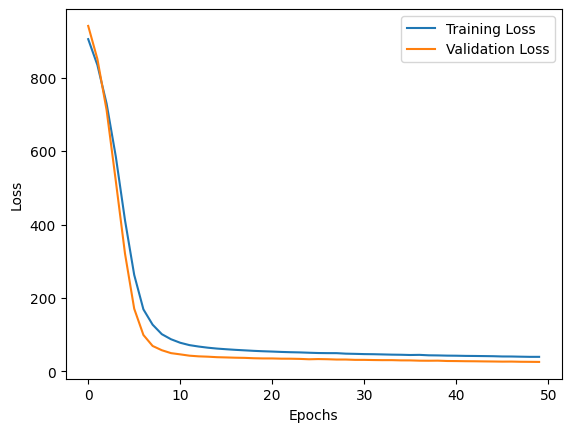
1. **Data Preprocessing**: Raw NYC taxi data was cleaned by handling missing values and creating new features like "hour of day" or "day of week" from timestamps.
2. **Modeling**: A deep learning model was designed using Keras/TensorFlow libraries.
3. **Training**: The model was trained on historical taxi trip data with appropriate hyperparameters such as learning rate and batch size.
4. **Evaluation**: The trained model was evaluated using test data to assess its predictive power.

**High-Level Diagram:**

[diagram]

**Conclusion:**

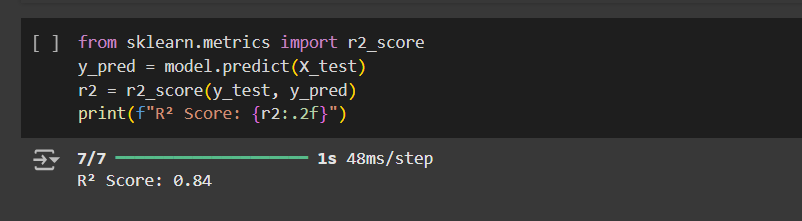
Analysis of the Model's Performance

**1. Loss Plot Analysis:**

The attached loss plot shows the **Training Loss** and **Validation Loss** over 50 epochs. Here's a breakdown of the key observations:

* **Initial High Loss**: At the beginning (epoch 0), both the training and validation losses are high, which is expected as the model starts with random weights.
* **Rapid Decrease in Loss**: In the first 10 epochs, both training and validation losses drop sharply, indicating that the model is learning quickly from the data.
* **Convergence**: After around 10 epochs, both training and validation losses begin to stabilize, suggesting that further training does not significantly improve performance. The model has reached a point where it is no longer making substantial improvements.
* **No Overfitting**: The training and validation losses are very close to each other throughout training, indicating that there is no significant overfitting. If overfitting were happening, we would expect to see the validation loss increase while the training loss continues to decrease. In this case, both losses decrease similarly and stabilize at similar values.

**2. R² Score:**

****The reported R² score of 0.84 indicates that the model explains 84% of the variance in the target variable (e.g., taxi demand or trip duration). This is a strong result for a regression task, as an R² score closer to 1 indicates a better fit.

* **Interpretation**: An R² score of 0.84 means that 84% of the variability in the target variable can be explained by the features used in the model. This suggests that the model has captured most of the important patterns in the data but still leaves room for improvement.

**3. Overall Conclusion:**

* The loss plot shows a well-trained model with no significant overfitting.
* The R² score of 0.84 indicates that the model performs well in explaining most of the variance in the target variable.
* The test loss and MAE suggest that while predictions are reasonably accurate, further tuning or additional features could improve performance.

**4. Future Improvements:**

* **Feature Engineering**: Adding more features like weather data or traffic conditions could help improve prediction accuracy.
* **Model Complexity**: Trying more complex models like LSTMs or GRUs might capture temporal dependencies better for time-series forecasting tasks.
* **Hyperparameter Tuning**: Further tuning hyperparameters such as learning rate, batch size, or network architecture could yield better results.

**References:**

<https://data.cityofnewyork.us/Transportation/2023-Yellow-Taxi-Trip-Data/4b4i-vvec/about_data>

<https://www.tensorflow.org/api_docs/python/tf/keras>