Daily Temperature Prediction for Istanbul Using GRU Neural Networks

Mustafa Efe Kurt
040200253
Electronics and Communication Engineering
Istanbul Technical University

***Abstract*—This paper describes the creation of a weather forecasting model utilizing Gated Recurrent Unit (GRU) neural networks. Using historical weather data, the model estimates daily temperatures in Istanbul for November 2024. The procedure preprocessing, scaling, and creating time series datasets. The model showed impressive accuracy results, with a Mean Square Error (MSE) of 2.42 and a Mean Absolute Error (MAE) of 1.18. The results suggest that the model is effective in short-term weather forecasting jobs.**

I. Methodology

The methodology involves data preprocessing, model training, and evaluation. The steps are detailed below:

A. Dataset

The dataset has daily weather data. It includes attributes such as *tempmax*, *tempmin*, *feelslikemax*, *feelslikemin*, *humidity*, and *temp* (target variable). The dataset covers the period from May 1, 2022, to October 31, 2024. The last 33 days of data were reserved as the test set for November 2024 predictions.

B. Preprocessing

- If there are NaN values in the dataset they are removed.
- The *datetime* column was converted to a datetime format.
- Data is normalized using MinMaxScaler for features and target. Features were scaled between 0 and 1, while the target (temp) was scaled based on the original temperature range.

C. Feature Selection

The selected features include:

- Maximum Temperature (*tempmax*)
- Minimum Temperature (*tempmin*)
- Feels Like Maximum Temperature (*feelslikemax*)
- Feels Like Minimum Temperature (*feelslikemin*)
- Humidity (*humidity*)
- Windspeed (*windspeed*)
- Precipitation (*precip*)

D. Sequence Generation

To predict the temperature on the fourth day, time-series were generated with a three-day sequence. This enabled the model to use past data for forecasts. Model tries to use past 3 days data and moves the frame.

E. Model Architecture

The GRU model comprises:

- Two GRU layers with 64 and 32 units, respectively.
- Batch Normalization layers after each GRU layer to improve stability.
- Leaky ReLU activation with `alpha=0.01` to prevent overfitting (negative section allows a small gradient instead of being completely zero).
- Dropout layers with a rate of 30% to avoid overfitting.
- A fully connected Dense layer with 16 units and ReLU activation.
- An output layer with one unit for predicting the target variable (*temp*).

F. Model Improvement Techniques

To enhance the model's performance and prevent overfitting, the following techniques were applied:

- **Epoch Tuning:** The number of epochs was initially set to 50, with training halted early using `EarlyStopping` based on validation loss.
- **Dropout Layers:** Dropout layers were added after each GRU layer to prevent overfitting by randomly setting a fraction of the input units to zero during training.
- **Leaky ReLU Activation:** Leaky ReLU was implemented to allow small negative gradients when the input is less than zero, improving the model's ability to generalize.
- **Early Stopping:** Early stopping was used to monitor validation loss and stop the training cycle when there is no effective change.
- **Learning Rate Adjustment:** `ReduceLROnPlateau` was used to reduce the learning rate when the validation loss plateaued, improving convergence.

II. Model Architecture

Model used for daily temperature prediction in Istanbul is based on a Gated Recurrent Unit (GRU) architecture. Main goal is to predict the temperature for the next day with the data of previous three days. The architecture includes the following components:

- **First GRU Layer:**
  - 64 units with *tanh* activation function.
  - `return_sequences=True` to pass the output to the next GRU layer.

- **LeakyReLU Activation:**
  - `alpha=0.01` to allow small negative gradients.
- **Dropout Layer:**
  - Dropout rate of 30% to prevent overfitting by randomly dropping neurons during training.
- **Second GRU Layer:**
  - 32 units with *tanh* activation function.
- **LeakyReLU Activation:**
  - `alpha=0.01` to continue allowing small negative gradients.
- **Dropout Layer:**
  - Dropout rate of 30% again to prevent overfitting.
- **Dense Layer:**
  - 16 units with *ReLU* activation function to introduce non-linearity.
- **Output Layer:**
  - 1 unit for predicting the temperature (*temp*).

The model is compiled using the **Adam** optimizer, and the learning rate was 0.0016. Loss function is **Mean Squared Error (MSE)**. The evaluation metric used is **Mean Absolute Error (MAE)**. These metrics are widely used in the projects and they give reasonable outputs.

The model was trained maximum of 50 epochs, but training was stopped early if there was no significant improvement in the validation loss, using the **EarlyStopping** callback.

III. Results

A. Model Performance

The model was evaluated using Mean Squared Error (MSE) and Mean Absolute Error (MAE). The results are as follows:

- MSE: 2.42
- MAE: 1.18

B. Visualization

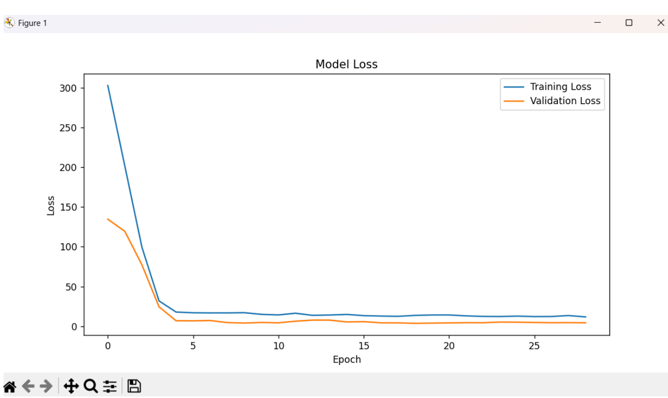

Fig. 1. Training and Validation Loss During Model Training

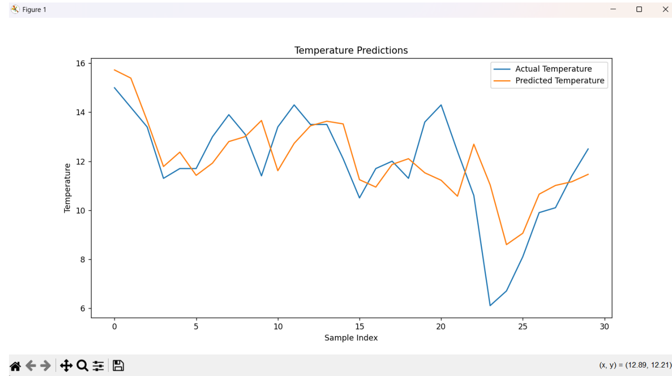

Fig. 2. Test Set Predictions for November 2024

C. Error Analysis

A heatmap (Fig. 3) was generated to visualize the daily prediction errors for November 2024. Errors were generally okay. There was a drastically decrease in the temperature in November 20th. It can be considered as an anomaly.

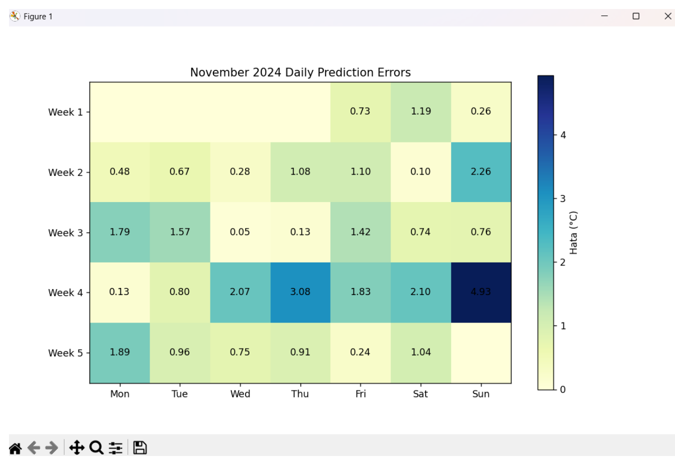

Fig. 3. Daily Prediction Errors for November 2024

D. Accuracy and Errors

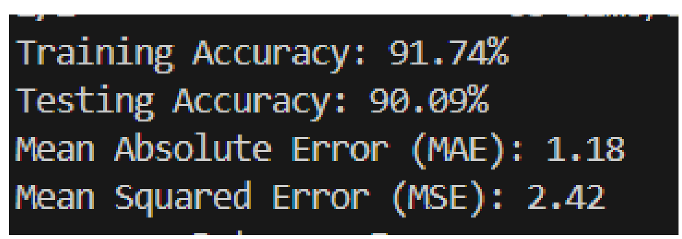

Fig. 4. Accuracy and Errors

IV. Conclusion

I tried to show GRU's predictive skills for daily temperature predictions. Epoch tuning, the addition of Dropout layers, Leaky ReLU activation, and Early Stopping, helped me to greatly reduce overfitting. Future work could be using more parameters, such as atmospheric pressure and solar radiation, to enhance accuracy. I also tried adding more GRU layers to model, but it did not give better accuracy.