Fisayo Jassey- Jabarr

Professor Patricia McManus

ITAI 2376: Deep Learning in Artificial Intelligence

February 09, 2024

**Lab 03 Conclusion**

Data Analysis:

* The data exploration revealed insights into the numerical, categorical, and text features.
* Outliers in numerical features were removed for potentially improved model performance.
* Text features were cleaned using stop word removal and stemming.
* The dataset was split into training, validation, and test sets for model evaluation.

Data Processing:

* A data processing pipeline was created using ColumnTransformer to handle different feature types consistently.
* Numerical features were imputed using mean and scaled using MinMaxScaler.
* Categorical features were imputed with a placeholder and one-hot encoded.
* Text features were processed using CountVectorizer, considering memory usage for feature creation.

Neural Network Training:

* The code defines hyperparameters and data loaders was provided.

Overall:

* The provided code demonstrates a structured approach to data exploration, cleaning, transformation, and preparation for building a neural network model. However, the challenge was a bit difficult

Explanation of the steps taken for the challenge:

1. import the necessary layers from tensorflow.keras.layers.
2. create a Sequential model instance.
3. add two hidden layers with 64 units each, both using ReLU activation.
4. A Dropout layer with a rate of 0.2 (20%) is added after each hidden layer for regularization.
5. The final output layer is added with the desired number of units depending on your problem (e.g., number of classes for classification).
6. compiled the model with your chosen optimizer, loss function, and metrics

Conclusions:

* The code builds a Multilayer Perceptron (MLP) using the Sequential module.
* It uses 2 hidden layers of 64 units each with ReLU activation and 20% dropout for regularization.
* The output layer size needs to be specified based on the problem (e.g., number of classes for classification).
* Optimizations like hyperparameter tuning and different architectures may be needed depending on the task and data.