Lab Course Machine Learning

Exercise Sheet 8

Prof. Dr. Dr. Lars Schmidt-Thieme
Kiran Madhusudhanan
HiWi: Basharat Mubashir Ahmed
Sophia Damilola Lawal
Submission deadline : **January 15, 2023**

General Instructions

1. Perform a data analysis, deal with missing values and any outliers.

1 Deep Learning on Tabular Data (13 points)

For this exercise, we will try to implement a deep learning solution for tabular dataset. Consider this as a competition task, similar to a Kaggle competition. Download the *bank-additional-full.csv* file from the learn-web which contains the marketing campaigns of a Portuguese banking institution. The task is to predict if a client would subscribe to a term deposit (variable y).

Use Pytorch framework to create the model and train. I would also suggest exploring the PyTorch Lightning Framework for the task. Pytorch Lightning is an easy to use flexible framework built on top of Pytorch.

The completed task should contain:

1. **Preprocessing** : The dataset contains both categorical and numerical features. Use standard scaler (from sklearn) for standardizing and encode the categorical columns as required.
2. **Model** : The model should consist of an embedding layer for the categorical columns followed by linear layers (Fully connected layers) and ReLU activations. The exact model structure to be used should be considered as a hyperparameter. Print out the final model structure selected. You could also try out various optimizers and other hyper parameters to tune your model.
3. **Evaluation** : The task needs to be evaluated for K=3 fold cross validation. Visualize and report the best results. Compare the performance with a boosting model like Catboost for reference.
4. **Visualization** : The model should also store relevant scalar information like the epoch loss for train, validation and test using tensorboard. Additionally, you can store graphics for the model structure also using tensorboard. Upload screenshots from the generated reports for evaluation.
5. (Ungraded) Hyper parameter Optimizer : One could also try to use RayTune or Optuna for various hyper parameter tuning techniques.

2 NLP - Word2Vec Model (7 points)

In the previous exercise we learned the embedding vectors for categorical variables. Extending this idea further one could create vector representation of a word by also considering the context of the word. *Continuous bag-of-words (CBOW)* model takes 'n' words before and after a target word and tries to predict the target word.

Given the following raw text file (learn-web), create a simple CBOW model that takes 2 words to the left and 2 words to the right as context to generate word embeddings of size 100. The model structure should contain

- Embedding layer of size 100
- Linear layer of size 128
- ReLU Activation
- Linear layer for output

Train the model for 50 epochs and visualize the embeddings to understand similarity between the learned word embeddings.