National Tsing Hua University

11220IEEM 513600

Deep Learning and Industrial Applications

Homework 2

Name: Li-Chiao Wang (王麗喬)   Student ID: 112034804

Completed on March 21, 2024

1. (20 pts) Select 2 hyper-parameters of the artificial neural network used in Lab 2, and set 3 different values for each. Perform experiments to compare the effects of varying these hyper-parameters on the loss and accuracy metrics across the training, validation, and test datasets. Present your findings with appropriate tables.

➢ Hyper-parameter 1: **batch_size**

Experimental results:

batch_size	*Training loss*	*Validation loss*	*Training acc.*	*Validation acc.*	*Test acc.*
16	0.3528	0.3308	85.1852%	83.9506%	80.6452%
32	**0.3923**	**0.4311**	**80.4233%**	**79.0123%**	**67.7419%**
64	0.4043	0.6351	80.9524%	70.3704%	70.9677%
128	0.4924	0.6187	77.7778%	66.6667%	61.2903%

Default: batch_size = 32

Set: batch_size = 16, 64, 128

➢ Hyper-parameter 2: **lr** (learning rate)

Experimental results:

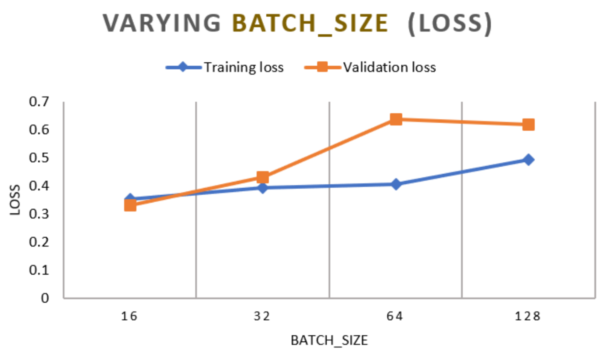
lr	*Training loss*	*Validation loss*	*Training acc.*	*Validation acc.*	*Test acc.*
1e-3	**0.3923**	**0.4311**	**80.4233%**	**79.0123%**	**67.7419%**
5e-3	0.3528	0.5240	84.1270%	74.0741%	70.9677%
1e-2	0.2855	0.4024	86.2434%	80.2469%	80.6452%
5e-2	0.6873	0.6971	55.5556%	51.8519%	48.3871%

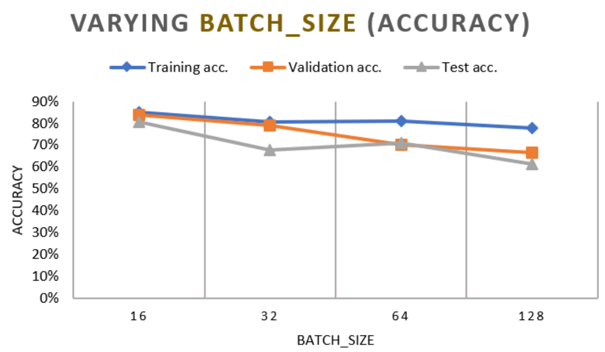
Default: lr = 1e-3

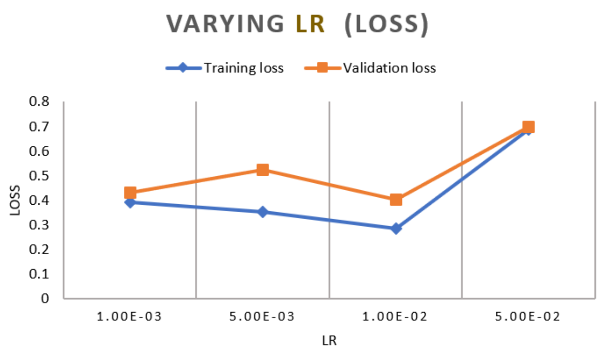
Set: lr = 5e-3, 1e-2, 5e-2

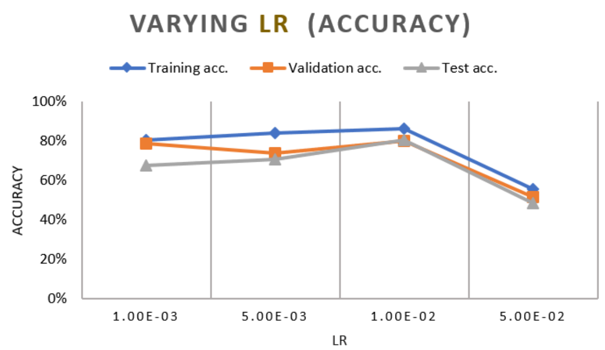
2. (20 pts) Based on your experiments in Question 1, analyze the outcomes. What differences do you observe with the changes in hyper-parameters? Discuss whether these adjustments contributed to improvements in model performance, you can use plots to support your points. (Approximately 100 words.)

Plots for varying the hyper-parameters:









Adjusting hyperparameters such as "batch size" and "learning rate" directly influences model performance. Increasing the batch size raises the loss and slightly lowers accuracy across training, validation, and test datasets. Similarly, raising the learning rate initially reduces the loss, but then it begins to increase again, while accuracy first improves and then declines. The optimal test accuracy is attained with a learning rate set to 1e-2. Therefore, it is evident that fine-tuning these hyperparameters significantly influences the model's performance. Based on our experiments, it is recommended to employ a smaller batch size (e.g., 16) and a learning rate of 1e-2 for training in this scenario.

3. (20 pts) In Lab 2, you may have noticed a discrepancy in accuracy between the training and test datasets. What do you think causes this occurrence? Discuss potential reasons for the gap in accuracy. (Approximately 100 words.)

The difference in accuracy between the training and test datasets could be due to model overfitting. This occurs when the model becomes too complex, capturing noise rather than the actual patterns in the data. To address this, we can employ techniques like regularization, hyperparameter tuning, or simplifying the model.

Another factor could be data imbalance in the training data, where there is a bias towards certain classes. Additionally, differences in the distributions of training and test data can contribute to this issue. In this case, there are more instances of target 1 in the training data, while the distribution of target values is more balanced in the test data.

Furthermore, insufficient training data can lead to poor performance on the test data. Increasing the volume of training data would help reduce this discrepancy.

4. (20 pts) Discuss methodologies for selecting relevant features in a tabular dataset for machine learning models. Highlight the importance of feature selection and how it can impact model performance. You are encouraged to consult external resources to support your arguments. Please cite any sources you refer to. (Approximately 100 words, excluding reference.)

Selecting the right features is key for building robust and efficient models. Not all features are equally important, and some may not contribute at all. By choosing features wisely, we can enhance our model's performance. Methods like Recursive Feature Elimination (RFE), Feature Importance from Trees, Principal Component Analysis (PCA), L1 Regularization (Lasso), and Correlation-based Feature Selection aid in this process. RFE removes less important features one by one, while Feature Importance from Trees ranks them using decision trees. PCA identifies vital features by explaining the most variance. L1 Regularization simplifies feature selection by penalizing coefficient size, and Correlation-based Feature Selection identifies highly correlated features.

Sources:

(1) Feature Selection Techniques in Machine Learning (Updated 2024) by Aman Gupta: https://www.analyticsvidhya.com/blog/2020/10/feature-selection-techniques-in-machine-learning/

(2) Feature Selection Techniques in Machine Learning with Python by Rahil Shaikh: https://towardsdatascience.com/feature-selection-techniques-in-machine-learning-with-python-f24e7da3f36e

5. (20 pts) While artificial neural networks (ANNs) are versatile, they may not always be the most efficient choice for handling tabular data. Identify and describe an alternative deep learning model that is better suited for tabular datasets. Explain the rationale behind its design specifically for tabular data, including its key features and advantages. Ensure to reference any external sources you consult. (Approximately 150 words, excluding reference.)

FT-Transformer is a specialized deep learning model designed for tabular datasets. It employs the Transformer architecture, enhanced with self-attention, to effectively capture feature relationships. By incorporating positional and feature embeddings, it is adept at understanding the sequential and relational nature of tabular data. Additionally, with features such as feature-wise normalization, it efficiently handles varying scales within the dataset. FT-Transformer focuses on scalability and performance in tasks like regression and classification on tabular data, making it a promising alternative to traditional ANNs.

Sources:

FT-Transformer (Introduced by Gorishniy et al. in Revisiting Deep Learning Models for Tabular Data): https://paperswithcode.com/method/ft-transformer