HW2

1.

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Batch size | epochs | Train(loss/acc) | Val(loss/acc) | Test(acc) |
| 32 | 80 | 0.3658/85.1852 | 0.5589/72.8395 | 70.9677 |
| 32 | 100 | 0.3063/89.4180 | 0.5803/79.0123 | 77.4193 |
| 32 | 120 | 0.3078/88.8889 | 0.6398/72.8395 | 77.41 |
| 64 | 80 | 0.4483/78.8360 | 0.6304/70.3704 | 64.5161 |
| 64 | 100 | 0.4054/82.0106 | 0.6152/72.8395 | 70.9677 |
| 64 | 120 | 0.4241/81.4815 | 0.6282/67.9012 | 64.5161 |
| 48 | 80 | 0.4723/78.3069 | 0.5199/71.6049 | 54.8387 |
| 48 | 100 | 0.4069/83.5979 | 0.4824/76.5432 | 61.2903 |
| 48 | 120 | 0.4131/79.8942 | 0.5218/70.3704 | 64.5161 |

2.

When adjusting the hyperparameters batch size and epochs, I found that increasing the epoch at the same batch size can improve accuracy, but higher epochs don't always result in better performance; too many epochs can actually decrease accuracy. Similarly, when increasing the batch size, I noticed that the accuracy decreased when the batch size was 48, but increased when the batch size was further increased to 64. Additionally, having higher training accuracy does not necessarily mean higher test accuracy.

3.

In lab2, I found that although training accuracy is generally higher, test accuracy tends to be higher as well, but this is not always the case. Sometimes, even when both training and validation accuracy are high, test accuracy can be very low. This phenomenon may be due to a lack of generalization, suggesting that the model may have overfit to the training set and performs poorly on data outside of it.

4.

Feature selection involves selecting the most important and relevant features from a dataset to improve the efficiency of machine learning models. After feature engineering, there may still be irrelevant data that can lead to poor predictions or redundant variables that make the learning process difficult and can cause overfitting. Selecting features that can exclude these factors can reduce the number of features, improve model accuracy, and reduce runtime. Additionally, selecting truly relevant features can simplify the model and help understand the data generation process.

Ref：<https://medium.com/ai%E5%8F%8D%E6%96%97%E5%9F%8E/%E7%89%B9%E5%BE%B5%E5%B7%A5%E7%A8%8B%E4%B9%8B%E7%89%B9%E5%BE%B5%E9%81%B8%E6%93%87%E6%A6%82%E5%BF%B5-ca11745db63c>

<https://ithelp.ithome.com.tw/articles/10245037>

5.

An alternative deep learning model that is specifically designed for handling tabular datasets is TabNet. TabNet is a novel deep learning architecture introduced by Google Cloud AI researchers in 2019. It is based on a sequential attention mechanism that can learn to select and process important features from tabular data, making it well-suited for tasks such as classification and regression on structured data.

TabNet is designed to learn decision tree-like mappings to gain the advantages of decision tree interpretability, and because it uses deep learning methods, it can gain the benefits of high performance. TabNet uses a machine learning technology called Sequential Attention. Users can understand each step in the model and the reasons for selecting model features. This mechanism can explain the process of the model obtaining the final prediction result, and can also be used to improve the model. Increase accuracy.

Ref：

<https://www.ithome.com.tw/news/139718>