National Tsing Hua University
1130IEEM 513600
Deep Learning and Industrial Applications
Homework 2

Name: 黃蘇珊 Student ID:113034421
Due on 2025.03.27

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-parameters:
1.Learning Rate (lr): 0.01, 0.001, 0.0001
2.Number of Hidden Units: 64, 128, 256

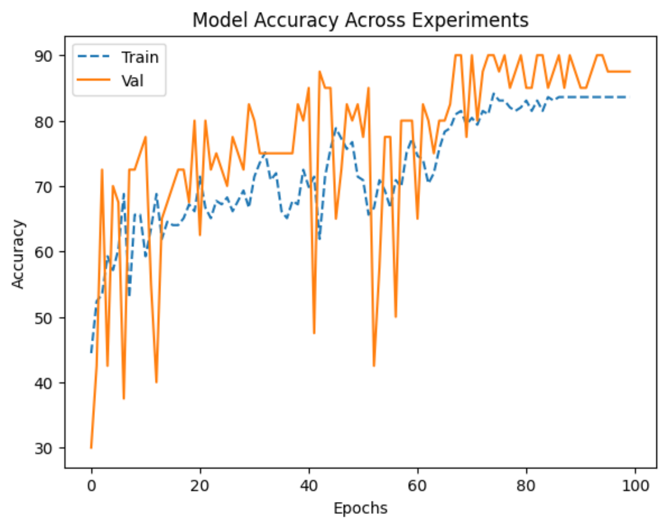
Learning Rate	Hidden Units	Training Loss	Training Acc	Validation Loss	Validation Acc	Test Loss	Test Acc
0.01	64	0.4609	80.9524%	0.6185	85.0000%	0.5479	67.74%
0.01	128	0.4271	80.4233%	0.4936	87.5000%	0.5494	70.97%
0.01	256	0.3833	81.4815%	0.4745	87.5000%	0.5420	70.97%
0.001	64	0.4541	77.7778%	0.4748	87.5000%	0.5818	70.97%
0.001	128	0.4885	75.6614%	0.4952	87.5000%	0.5566	64.52%
0.001	**256**	**0.3922**	**84.1270%**	**0.4305**	**92.5000%**	**0.4945**	**80.65%**
0.0001	64	0.4784	77.2487%	0.5044	87.5000%	0.5693	67.74%
0.0001	128	0.4379	81.4815%	0.4836	87.5000%	0.6009	74.19%
0.0001	256	0.4210	83.0688%	0.4637	90.0000%	0.5387	74.19%

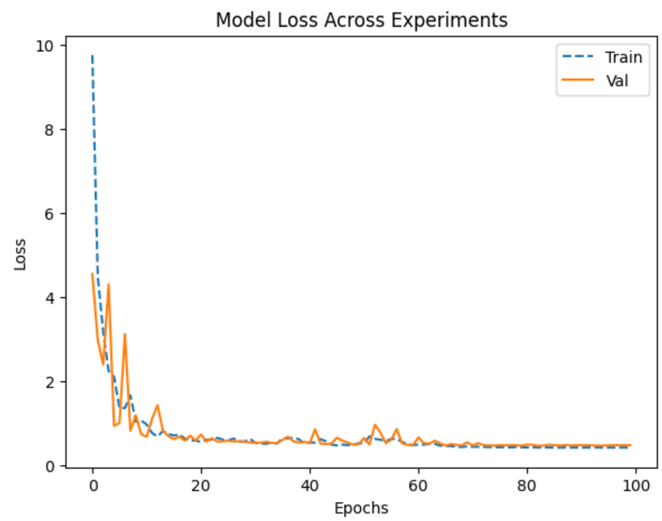
Findings:
Optimal Learning Rate (~0.001)
- Balances **fast convergence** while **avoiding instability**.
- Allows **efficient optimization** without skipping over optimal weights.

Conclusion:
- **a too high LR leads to instability, and a too-low LR leads to slow or stuck learning.**
- **Increasing hidden units improves model capacity only if trained with the right LR.**

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.)

From my experiments, the learning rate (η) and the number of hidden units **significantly influenced model performance.** The best configuration—learning rate = 0.001 and 256 hidden units—resulted in the lowest validation loss (0.4305) and highest validation accuracy (92.5%), i**ndicating strong generalization.**





- The accuracy plot shows that training accuracy steadily increases, but validation accuracy fluctuates before stabilizing. This suggests early instability, which is expected as the model optimizes. **A lower learning rate (0.0001) caused slower convergence, while a higher rate (0.01) led to overshooting.**
- The loss plot confirms smooth loss reduction over epochs**, validating that 0.001 was optimal for balance between speed and stability.**

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.)

From Excel data, we can conclude:

- **Training Dataset:**
  - Positive Cases: **149 (54.58%)**
  - Negative Cases: **123 (45.42%)**
  - Total Samples: **273**
- **Test Dataset:**
  - Positive Cases: **15 (48.39%)**
  - Negative Cases: **16 (51.61%)**
  - Total Samples: **31**

Potential Reasons for the Gap in Accuracy

a. **Sample size issue:** The training dataset has 273 samples, whereas the test dataset only has 31 samples. A small test set may lead to higher variance in accuracy due to the limited number of samples**, making it sensitive to small changes in predictions.**

b. **Data Imbalance:** The training dataset has 54.58% positive cases and 45.42% negative cases**.** In contrast, the test dataset is more balanced, with 48.39% positive cases and 51.61% negative cases. **This imbalance in the training data could lead to the model overfitting to the majority class.**

c. **Hyperparameter Selection:** From my previous experiments, I found that a learning rate of 0.001 with 256 hidden units performed the best, achieving the highest validation and test accuracy. **Suboptimal hyperparameter selection could contribute to the observed performance gap.**

d. **Regularization Techniques:** **Lack of techniques like dropout** may allow the model to memorize training data instead of learning meaningful representations, **leading to a lower test accuracy.**

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.)

Feature selection is an important step in machine learning that helps **improve model performance by selecting the most relevant features while reducing dimensionality and computational cost.** Effective feature selection enhances model interpretability, reduces overfitting, and improves generalization. Common methodologies include:

1. **Filter Methods** – Uses statistical tests (e.g., Pearson correlation, chi-square test) to rank features independently of the model.
2. **Wrapper Methods** – Employs iterative search techniques like Recursive Feature Elimination (RFE) to assess feature importance based on model performance.
3. **Embedded Methods** – Feature selection occurs during model training, such as LASSO (L1 regularization) in linear regression.
4. **Dimensionality Reduction** – Techniques like Principal Component Analysis (PCA) transform features to reduce redundancy.

References:

- Guyon, I., & Elisseeff, A. (2003). An introduction to variable and feature selection. *Journal of Machine Learning Research*, 3, 1157–1182.
- Chandrashekar, G., & Sahin, F. (2014). A survey on feature selection methods. *Computers & Electrical Engineering*, 40(1), 16-28.

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

you to reference any external sources you consult. (Approximately 150 words, excluding reference.)

Alternative Deep Learning Model for Tabular Data: TabNet

- **Why ANNs Struggle with Tabular Data?**

Traditional ANNs often underperform on tabular data compared to gradient boosting models (e.g., XGBoost, LightGBM). This happens because tabular data has complex feature interactions, which ANNs struggle to capture effectively. Also, ANNs treat all features equally, making them less efficient for structured datasets.

- **Why TabNet is a Better Alternative?**
    1. **Sequential attention mechanism:** Instead of processing all features at once, TabNet learns which features to focus on at each step, improving interpretability and efficiency.
    2. **More Interpretable**: It tells us why it decided by showing which features it looked at, helping us understand the model better.
    3. **Efficient with Data:** TabNet works well with less data and needs less preprocessing compared to other deep models.
    4. **Handles Feature Relationships**: It captures complex feature interactions, similar to tree-based models like XGBoost, but with the power of deep learning.
    5. **End-to-End Learning**: No need for lots of feature engineering — TabNet can learn directly from raw tabular data.

References:

https://arxiv.org/abs/1908.07442
https://cloud.google.com/blog/products/ai-machine-learning/tabnet-google-clouds-new-deep-learning-architecture-for-tabular-data
https://www.kaggle.com/learn/tabular-playground-series