National Tsing Hua University

1130IEEM 513600

Deep Learning and Industrial Applications

Homework 2

Name: 楊東偉 **Student ID: 113003815**

1. hidden_sizes = [8, 16, 32]

   learning_rates = [0.001, 0.01, 0.1]

train_acc	val_acc	test_acc	train_loss	val_loss	test_loss	hidden_size	learning_rate
0.842593	0.925926	0.709677	0.405541	0.241608	0.642166	16	0.001
0.847222	0.925926	0.709677	0.415144	0.253939	0.685191	32	0.001
0.916667	0.907407	0.677419	0.711477	0.596657	1.470861	8	0.010
0.819444	0.888889	0.677419	0.407594	0.259426	0.639823	8	0.001
0.898148	0.870370	0.709677	3.401132	1.804441	5.153871	8	0.100
0.893519	0.851852	0.709677	1.333676	1.036434	1.907168	16	0.010
0.888889	0.851852	0.741935	3.702525	3.323871	6.544828	16	0.100
0.893519	0.851852	0.741935	2.088002	1.504202	2.245832	32	0.010
0.888889	0.851852	0.741935	3.653885	2.440895	7.215479	32	0.100

2. From the experiments, increasing the hidden layer size from 8 to 16 or 32 improved both validation and test accuracy, suggesting that a larger model better captures underlying patterns. Regarding learning rate, 0.01 yielded the most balanced performance across all datasets. A high learning rate like 0.1 caused unstable loss, while a low rate (0.001) often resulted in slower convergence and lower accuracy. These results indicate that carefully tuning these hyper-parameters significantly influences model performance.

3. The discrepancy between training and test accuracy is mainly due to overfitting. Since the dataset is relatively small, the model might learn patterns specific to the training data, including noise, rather than generalizable features. As a result, it performs worse on unseen test data. Other possible reasons include lack of regularization, overly complex model architecture, or too many training epochs, all of which can cause the model to fit the training data too closely and reduce its generalization performance.

4. n tabular datasets, feature selection methods include filter methods (e.g., correlation scores, chi-squared test), wrapper methods (e.g., Recursive Feature Elimination), and embedded methods (e.g., L1 regularization). Feature selection reduces noise by removing irrelevant or redundant features, which helps prevent overfitting, improves model training speed, and enhances overall generalization performance. Selecting meaningful features not only leads to better accuracy but also improves model interpretability, especially important in medical or financial applications.

[1] J. Brownlee, "An Introduction to Feature Selection", Machine Learning Mastery, 2019.

[2] https://scikit-learn.org/stable/modules/feature_selection.html

5. Although artificial neural networks (ANNs) are effective in domains like images and speech, they often underperform on tabular data due to weak feature interactions and sparse structures. TabNet, proposed by Google, is a deep learning architecture specifically designed for tabular data. It uses a sequential attention mechanism to select relevant features at each decision step, improving interpretability and model efficiency. Unlike traditional ANNs, TabNet handles missing values, learns complex non-linear relationships between features, and maintains high performance without requiring heavy feature engineering. In many benchmarks, TabNet has outperformed both ANNs and traditional models like XGBoost, making it a strong candidate for tabular data tasks. Its ability to balance accuracy, transparency, and scalability makes it a suitable choice for real-world tabular applications.