Phase 3 Report: Model Engineering

Artur Zagitov, Ildar Zalialiev, Artem Nazarov

Contents

1 Chapter 4: Model Engineering

1.1 Literature Research on Similar Problems

Based on our review of existing literature, we identified several studies that address problems similar to ours, namely demand prediction for online advertisements and click-through prediction.

In the first relevant study [1], the authors utilize both text features, processed through TfIdf vectorization and FastText embeddings, and different image features, such as features extracted using NIMA for aesthetic evaluation, to predict e-commerce advertisement demand. They experimented with various models, including AdaBoost, XGBoost, LightGBM, Multi-layer Perceptrons (MLP), Long Short-Term Memory (LSTM) networks, and Gated Recurrent Unit (GRU) networks. The most effective models were found to be LightGBM, combined GRU and DNN, and MLP.

Another study [2] focuses on click-through prediction for display advertising, which is a task of predicting, whether an ad will be clicked or not, quite similar to our problem. They compare various deep learning approaches, including a proprietary model based on residual networks, against traditional methods. The study uses datasets from iPinYou, Criteo, and Avazu, and demonstrates superior performance of deep neural networks, particularly the residual network-based approach.

These insights are relevant to our project as they provide proven methodologies and baselines for performance, guiding our approach in model selection and implementation.

1.2 Quality Measures

The quality of the machine learning models in our project is assessed using the following criteria:

- **Root Mean Squared Error (RMSE):** The primary metric for evaluating the accuracy of our predictions. It measures the average difference between predicted and actual deal probabilities, with lower values indicating better performance. RMSE gives a relatively high weight to large errors and is more sensitive to outliers. It is chosen as a primary metric, since it is better suited for cases where large errors are particularly undesirable.
- **Model Scalability:** Ability to handle increasing amounts of data without a significant degradation in performance or speed.
- **Model Stability:** Consistency of the model's performance over time and across various data segments.

These measures are crucial because they ensure that the model not only performs well on average but also behaves predictably across different scenarios, which is crucial for deployment in dynamic environments like online marketplaces.

1.3 Model Selection

For our project, we selected two models: a simple Multi-layer Perceptron (MLP) and a more complex Residual Network (ResNet). These choices are driven by the need to balance between model complexity and computational efficiency, in line with our business objectives and data characteristics.

1.3.1 MLP Architecture

The MLP model consists of three layers with the hyperparameter configurations presented in Table 1. This architecture serves as a baseline model and is known for its ability to learn complex patterns in data. The model's architecture is illustrated in Figure 1.

Parameter	Options	Best Value
Hidden Layer 1 Size	[32, 64, 128]	32
Hidden Layer 2 Size	[16, 32, 64]	16
Hidden Layer 3 Size	[8, 16, 32]	32

Table 1: MLP Model HyperParameters

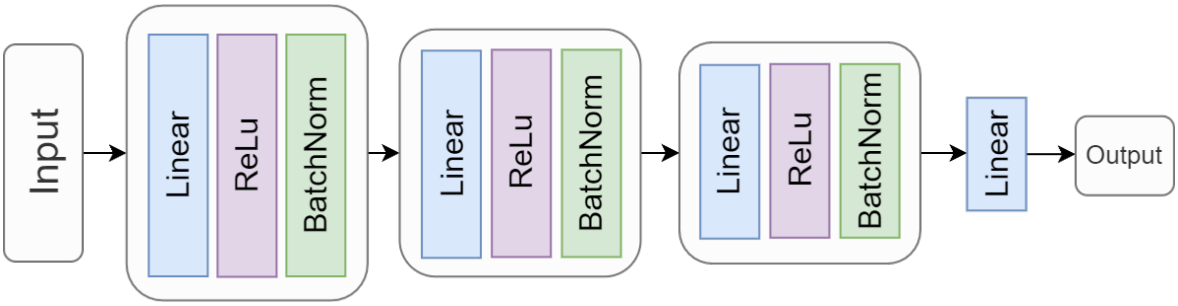

Figure 1: MLP Model Architecture

1.3.2 ResNet Architecture

The ResNet model uses residual blocks, each consisting of two linear layers accompanied by batch normalization and dropout at the end, to facilitate learning deeper representations without the vanishing gradient problem, while dropout layer helps in preventing overfitting. The embedding size and the number of residual blocks were optimized as shown in Table 2. The model's architecture is illustrated in Figure 2.

Parameter	Options	Best Value
Embedding Size	[16, 32, 64, 128]	128
Number of Residual Blocks	[1, 3, 5, 10]	3
Dropout Rate	[0.35, 0.5, 0.75]	0.75

Table 2: ResNet Model HyperParameters

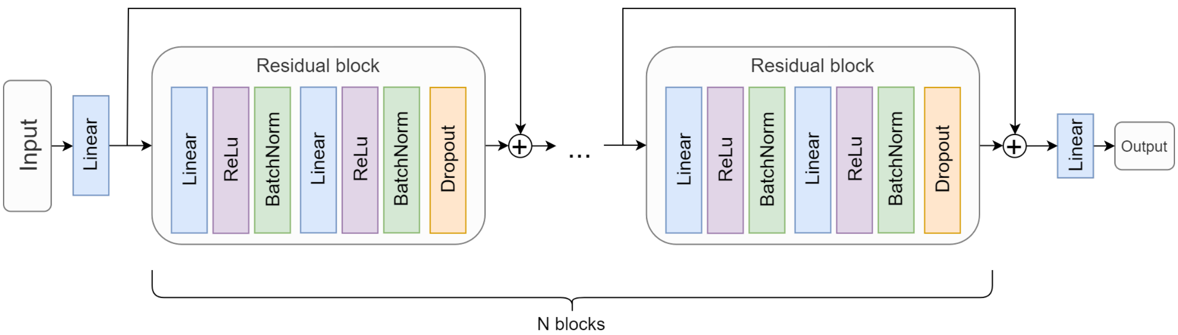

Figure 2: Residual Network Model Architecture

1.3.3 Model Signature

The input and output dimensions of both models are as follows:

- **Input Dimension:** 125
- **Output Dimension:** 1

1.4 Domain Knowledge Incorporation

Incorporating domain knowledge involves ensuring that the selected models and metrics align with the business needs of predicting advertisement demand effectively. The use of RMSE aligns with the business's need for accurate predictions, while the choice of MLP and ResNet architectures is influenced by their proven effectiveness in similar tasks as identified through our literature research.

1.5 Model Training

For training, we employed a 3-fold cross-validation strategy using GridSearch for hyperparameter optimization. The test datasets comprise the next 10,000 samples, ensuring no overlap with training data. We used Root Mean Squared Error (RMSE) as our loss function, as in our experiments we found, that RMSE loss results in better models, then MSE loss. For the optimization of our models, we selected the AdamW optimizer. This choice was made after testing showed that AdamW has superior performance for our datasets and model architectures. Both models were trained using a fixed learning rate of 5×10^{-4} and a weight decay of 10.0. We limited the training to 25 epochs based on validation performance, since larger number of epochs usually was resulting in overfitting. The performance of both models after training is summarized in Table 3.

The performance of both models after training is summarized below:

Model	**RMSE**	**MSE**	**MAE**
MLP	0.242	0.0585	0.162
ResNet	0.246	0.0606	0.166

Table 3: Model Performance Summary

These results indicate that both models meet the project's success criteria, with the MLP slightly outperforming the ResNet.

We also used parallel plots to visually analyze the impact of hyperparameters on model performance. For the MLP, the parallel plot (Figure 3) indicated that variations in hyperparameters did not significantly impact the performance. On the other hand, the ResNet model showed a more significant variation in RMSE across different hyperparameter settings, as shown in the parallel plot in Figure 4.

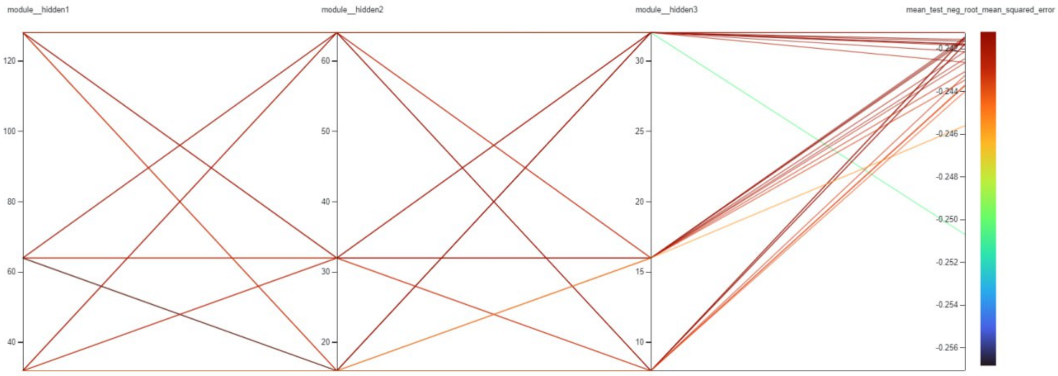

Figure 3: Parallel Plot for MLP Hyperparameters

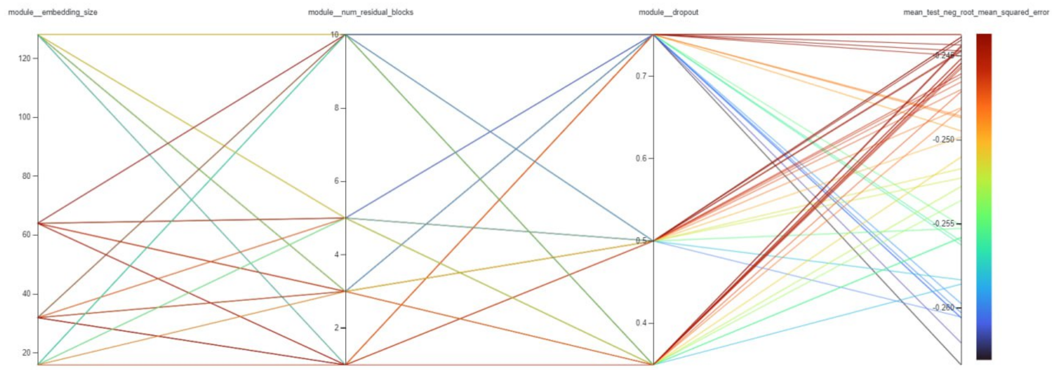

Figure 4: Parallel Plot for ResNet Hyperparameters

1.6 Assure Reproducibility

1.6.1 Method Reproducibility

To ensure the reproducibility of our models, we have documented the following:

- **Model Architectures:** Detailed descriptions and diagrams of both the MLP (Figure 1) and ResNet (Figure 2) architectures are provided.
- **Hyperparameters:** The specific hyperparameters used for each model are outlined in Tables 1 and 2.
- **Software and Libraries:** We utilized skorch and PyTorch to implement the models. Skorch allows the use of PyTorch models with grid search methods from scikit-learn.
- **Environment:** The models were trained and tested in a WSL2 environment on a Windows PC.
- **Random Seed:** We primarily used the seed '42' for hyperparameter optimization and evaluation.

1.6.2 Result Reproducibility

To verify result reproducibility, we evaluated our models using five different random seeds. The evaluation metrics for both models across these seeds are presented in Tables 4 and 5.

Metric	Values	Average	Variance
MAE	[0.1606, 0.1615, 0.1631, 0.1632, 0.1625]	0.1622	9.67×10^{-7}
MSE	[0.0586, 0.0586, 0.0585, 0.0586, 0.0586]	0.0586	1.28×10^{-9}
RMSE	[0.2421, 0.2421, 0.2419, 0.2420, 0.2420]	0.2420	5.47×10^{-9}

Table 4: MLP Test Metrics for Different Seeds

Metric	Values	Average	Variance
MAE	[0.1683, 0.1589, 0.1578, 0.1659, 0.1804]	0.1663	6.59×10^{-5}
MSE	[0.0597, 0.0607, 0.0628, 0.0592, 0.0607]	0.0606	1.51×10^{-6}
RMSE	[0.2444, 0.2464, 0.2506, 0.2433, 0.2464]	0.2462	6.16×10^{-6}

Table 5: ResNet Test Metrics for Different Seeds

The MLP demonstrates very low variance in performance across different seeds, with an average RMSE lower than that of ResNet. While ResNet has slightly higher variance, it still remains relatively low.

1.6.3 Experimental Documentation

Our experiment tracking involves detailed logs of all model training and validation steps, ensuring that each model iteration is fully documented and reproducible.

References

[1] S. Rai, A. Gupta, A. Anand, A. Trivedi, and S. Bhadauria, "Demand prediction for e-commerce advertisements: A comparative study using state-of-the-art machine learning methods," in *2019 10th International Conference on Computing, Communication and Networking Technologies (ICCNT)*, pp. 1–6, 2019.

[2] M. Liu, S. Cai, Z. Lai, L. Qiu, Z. Hu, and Y. Ding, "A joint learning model for click-through prediction in display advertising," *Neurocomputing*, vol. 445, pp. 206–219, 2021.