A LSTM-based Methodology for Learning Curve Prediction: Leveraging Learning Patterns from Multiple Models to Predict the Performance of a New Model

Abstract

Training deep learning models requires significant time and computational resources, especially when comparing the performance of various models such as ResNet-34, ResNet-50, Plain-34, and Plain-50. It's challenging to determine a model's final performance from the insufficient data of the early training stages. To address this, this paper proposes a methodology to predict the learning curve of Plain-50 using data from ResNet-34, ResNet-50, and Plain-34. We utilize two predictive models: Polynomial Regression and a Long Short-Term Memory (LSTM) network. The LSTM model is enhanced by including not just accuracy but also the gradient (rate of change in accuracy per epoch) as a feature to improve prediction accuracy. Our experiments show that the LSTM-based model, having learned patterns from multiple models, is more effective than polynomial regression at capturing the nonlinear trends and convergence behavior of Plain-50's learning curve. This suggests that leveraging the learning trends of existing models can effectively predict a new model's performance early on, thus reducing unnecessary resource consumption.

1 Introduction

Training deep learning models can be a time-consuming process, spanning dozens or even hundreds of epochs depending on the model's complexity and dataset size. When comparing different architectures like ResNet-34, ResNet-50, Plain-34, and Plain-50, training each one to completion demands substantial time and computational resources. The fundamental problem is that it is difficult to predict a model's final performance from the limited data of the initial few epochs (e.g., 5 epochs).

This paper proposes learning curve prediction as a solution to this problem. By using early training data to predict a model's future learning curve, we can estimate its final performance without full training. Specifically, we aim to train a predictive model on the data from ResNet-34, ResNet-50, and Plain-34 and then use it to forecast the learning curve of a new model, Plain-50. This approach validates the generalizability of our predictive model. Existing methods often use parameter-based regression models such as logistic or power-law functions to model learning curves. However, these models assume a specific curve shape, limiting their flexibility in handling diverse training scenarios. To overcome this, we propose an LSTM-based methodology, which is well-suited for analyzing sequential data and can learn complex nonlinear patterns. For comparison, we also utilize a traditional Polynomial Regression model to provide a comprehensive analysis of both methods' strengths and weaknesses.

2 Methodology

2.1 Polynomial Regression-based Prediction

Polynomial regression is a traditional regression analysis method that finds the best-fit polynomial function for the initial learning curve data. Using epoch-accuracy pairs from the training data, the polynomial coefficients are estimated using the method of least squares. In this study, we use the combined data from ResNet-34, ResNet-50, and Plain-34 to train the model, which then predicts the learning curve of Plain-50.

2.2 LSTM-based Prediction

LSTM excels at learning temporal dependencies in sequential data. To maximize its predictive power, our LSTM model uses the epoch, accuracy, and gradient ($\Delta y_t = y_t - y_{t-1}$) as input features. The combined learning curve data from ResNet-34, ResNet-50, and Plain-34 is used to train the LSTM model. The model then predicts the future accuracy of Plain-50 based on its initial data. This is a recursive prediction process where the model's output is fed back into the input sequence to predict the next time step's accuracy.

2.3 Performance Evaluation Metrics

Each prediction method's performance is evaluated by comparing its predicted curve to the actual learning curve. The main evaluation metrics are:

- **Root Mean Squared Error (RMSE)**: Measures the average magnitude of the error between predicted and actual values.

$$RMSE = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i - \hat{y}_i)^2}$$

where y_i is the actual accuracy, $\hat{y}_i$ is the predicted accuracy, and N is the number of data points.

- **Mean Absolute Error (MAE)**: Measures the average of the absolute errors. Unlike RMSE, it's less sensitive to outliers.

$$MAE = \frac{1}{N}\sum_{i=1}^{N}|y_i - \hat{y}_i|$$

3 Results and Discussion

3.1 Analysis of LSTM Model Performance

After training on the learning patterns of ResNet-34, ResNet-50, and Plain-34, the LSTM model's prediction accuracy for Plain-50 significantly improved as the initial training data increased beyond 20 epochs. At 40 epochs, the predicted curve almost perfectly converged with the actual learning curve. This demonstrates that the LSTM effectively learned to capture the nonlinear convergence patterns and long-term trends of the learning process.

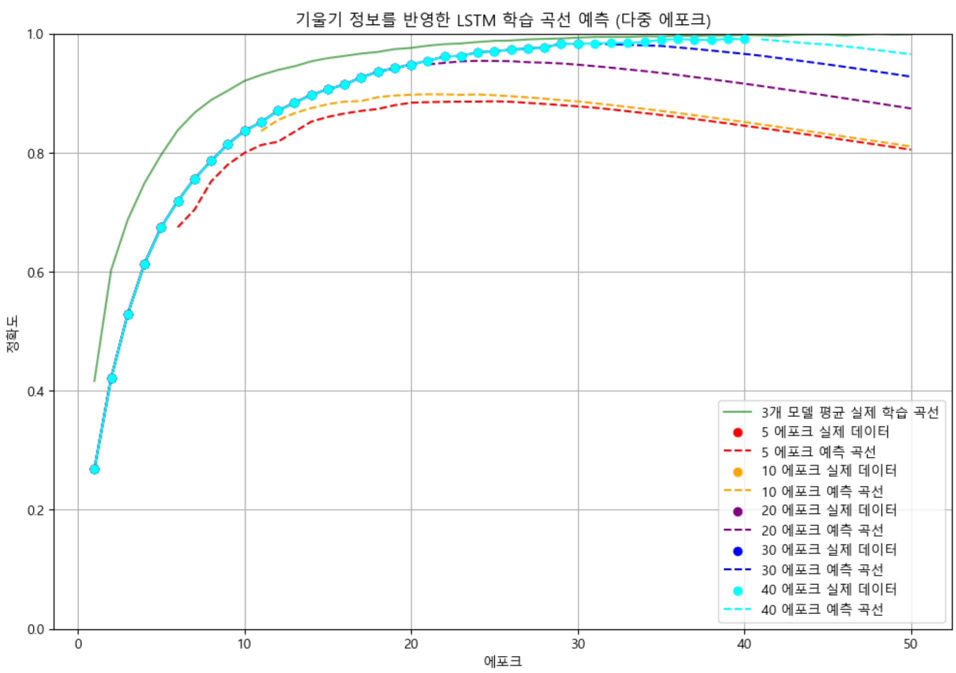

Figure 1: LSTM-based learning curve prediction results

3.2 Analysis of Polynomial Regression Model Performance

The polynomial regression model showed a tendency to be highly sensitive to the local patterns in the initial training data. While it performed reasonably well up to 10 epochs, it began to diverge significantly from the actual learning curve beyond 20 epochs. This shows the limitations of polynomial functions in accurately approximating the later, more subtle convergence stages of the learning curve.

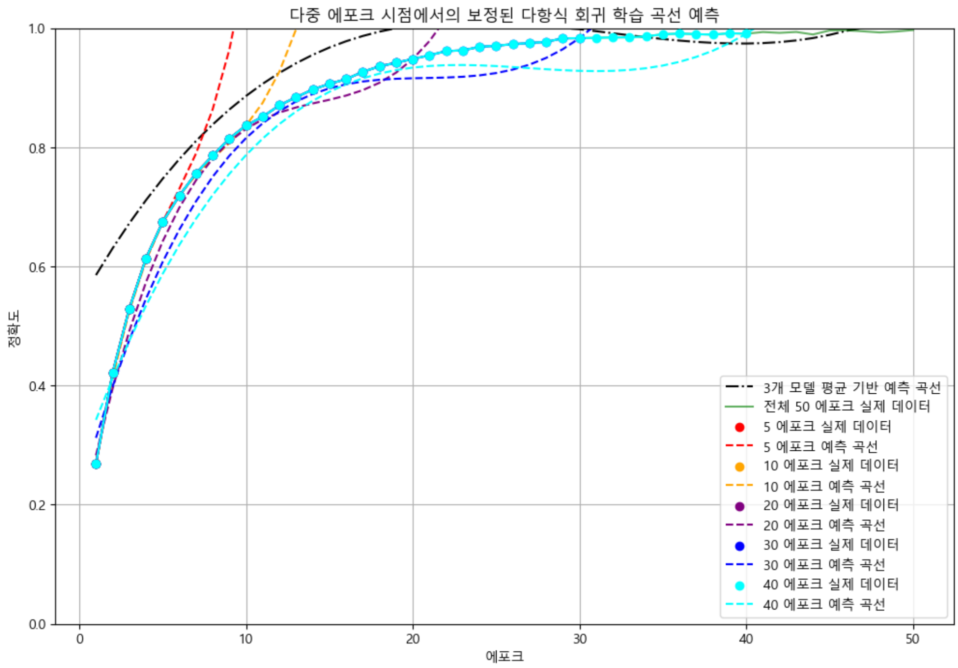

Figure 2: Polynomial regression-based learning curve prediction results

3.3 Model Comparison and Conclusion

Our results suggest that a data-driven recurrent neural network model is more flexible and provides stronger predictive performance for learning curves than a parameter-based regression model. The inclusion of gradient as a time-series feature was particularly crucial, allowing the model to grasp the dynamic changes in the learning curve.

4 Future Work

This study's limitation is that it only used Train Accuracy to predict the learning curve. To improve prediction accuracy, it would be beneficial to incorporate additional metrics like Train Loss, Test Loss, and Test Accuracy into the learning process. These metrics provide crucial information about the model's generalization performance and overfitting tendencies, allowing the predictive model to learn from a richer context.

Furthermore, while this study focused on LSTM, future research could explore other state-of-the-art time-series models such as Transformers or Causal Convolutional Networks (CNNs). Transformers, with their self-attention mechanism, are especially good at capturing global relationships in data, which could enable them to model the long-term patterns of learning curves more effectively. These additions could further enhance the performance and generalizability of learning curve prediction models.

5 References

1 Vianna, L.S. et al. "Analysis of learning curves in predictive modeling using exponential curve fitting with an asymptotic approach." *PLOS One*, 2024.

2 Goedhart, J.M. et al. "Estimation of Predictive Performance in High-Dimensional Data Settings using Learning Curves." *arXiv preprint*, 2022.

3 Sabry, Y. "Prediction of cryptocurrency prices using LSTM, SVM and polynomial regression." *ResearchGate*, 2024.

4 Yang, Y. et al. "Regression analysis and prediction using LSTM model and machine learning methods." *Journal of Physics: Conference Series*, 2022.

6 6. Reference Image

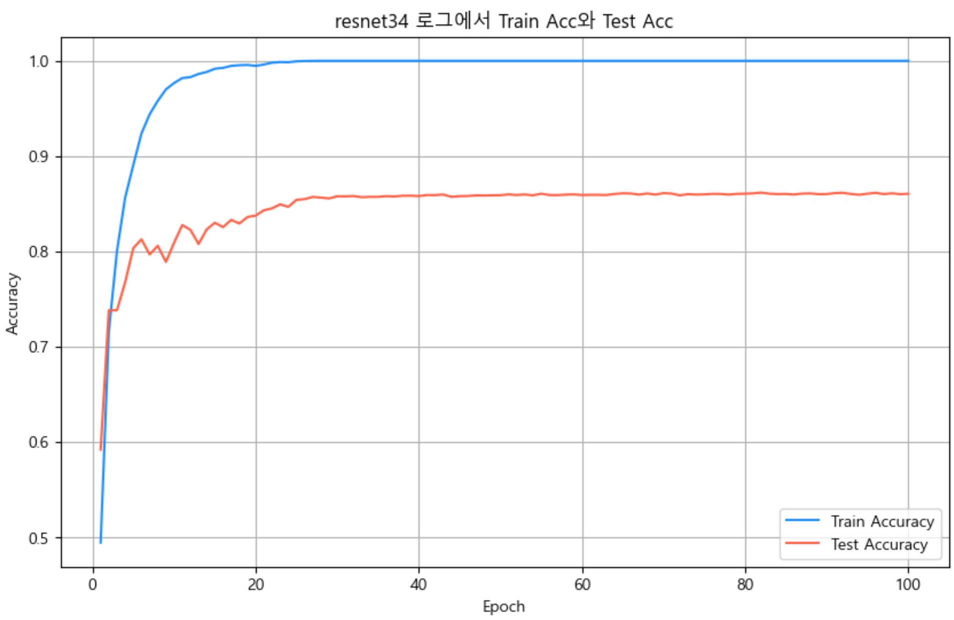

Figure 3: Comparison of Train Accuracy and Test Accuracy for ResNet-34

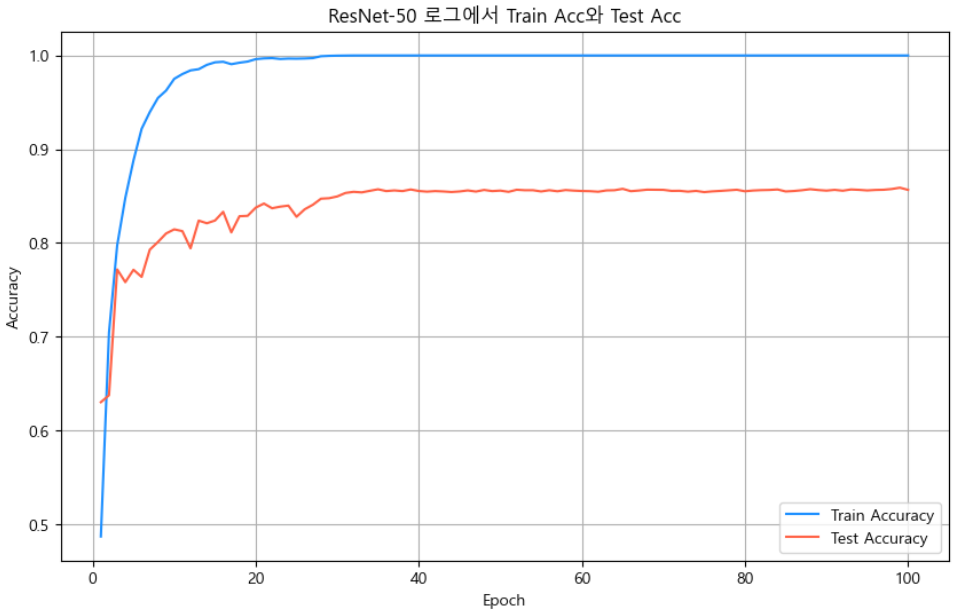

Figure 4: Comparison of Train Accuracy and Test Accuracy for ResNet-50

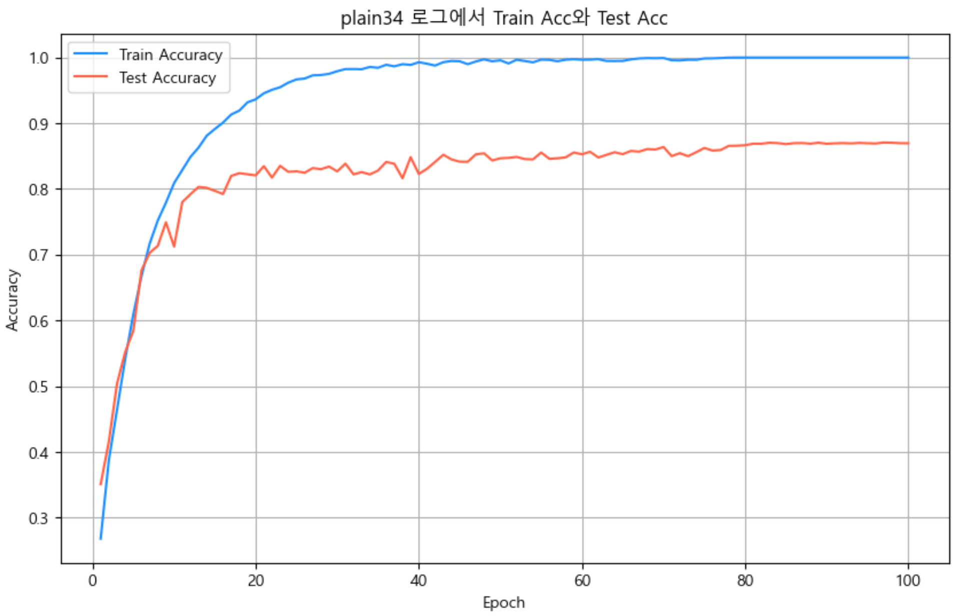

Figure 5: Comparison of Train Accuracy and Test Accuracy for Plain-34

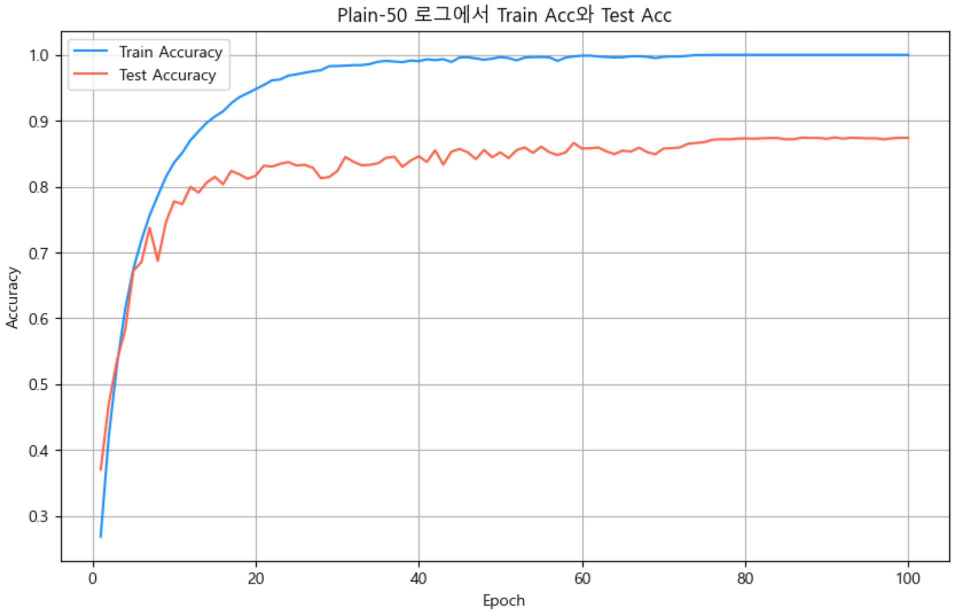

Figure 6: Comparison of Train Accuracy and Test Accuracy for Plain-50