DETAILED ANALYSIS OF TUNING EXPERIMENT

By Blessing Ilesanmi

29th November, 2024

Introduction

In this report, I evaluate the performance of convolutional neural networks (CNNs) using a grid search methodology. The goal of this experiment was to explore various combinations of hyperparameters to determine which configuration yields the best model performance. The hyperparameters tested during the grid search were:

- conv_layers: The number of convolutional layers in the model (set to 2 or 3).
- dense_units: The number of units in the dense layer (set to 64 or 128).
- batch_size: The batch size for training (set to 32 or 64).
- epochs: The number of epochs for training (set to 5 or 10).

Each configuration was evaluated by training the model and measuring its accuracy on the test dataset. Below are the results from the grid search, which highlight the performance of each combination of hyperparameters. The best-performing configuration is highlighted at the end of the results section.

Results

Model Configuration 1: {'batch_size': 32, 'conv_layers': 2, 'dense_units': 64, 'epochs': 5}

- Test Accuracy: 29.96%
- Notes: The model demonstrated some improvement over the initial epochs, but accuracy remained below 30%, indicating the need for more epochs or a more complex architecture.

Model Configuration 2: {'batch_size': 32, 'conv_layers': 2, 'dense_units': 64, 'epochs': 10}

- Test Accuracy: 32.89%
- Notes: The additional epochs improved performance, but the gap between training and test accuracy suggests that the model is overfitting.

Model Configuration 3: {'batch_size': 32, 'conv_layers': 2, 'dense_units': 128, 'epochs': 5}

- Test Accuracy: 34.67%
- Notes: Increasing the units in the dense layer led to a slight increase in test accuracy, showing that the model could capture more complex patterns.

Model Configuration 4: {'batch_size': 32, 'conv_layers': 2, 'dense_units': 128, 'epochs': 10}

- Test Accuracy: 32.19%
- Notes: Although the model achieved a high training accuracy, it did not improve significantly in test accuracy. This suggests potential overfitting.

Model Configuration 5: {'batch_size': 32, 'conv_layers': 3, 'dense_units': 64, 'epochs': 5}

- Test Accuracy: 31.79%
- Notes: Adding a third convolutional layer did not lead to a significant improvement in test accuracy, and the performance was still lower than other configurations.

Model Configuration 6: {'batch_size': 32, 'conv_layers': 3, 'dense_units': 64, 'epochs': 10}

- Test Accuracy: 32.33%
- Notes: Although the model was trained for 10 epochs, the test accuracy did not show a major improvement compared to the previous configurations.

Model Configuration 7: {'batch_size': 32, 'conv_layers': 3, 'dense_units': 128, 'epochs': 5}

- Test Accuracy: 35.23%
- Notes: The addition of a third convolutional layer with 128 dense units improved performance slightly, but the gap between training and test accuracy remained.

Model Configuration 8: {'batch_size': 32, 'conv_layers': 3, 'dense_units': 128, 'epochs': 10}

- Test Accuracy: 31.89%
- Notes: Despite a high training accuracy, the model did not generalize well to the test set, as shown by the limited improvement in test accuracy.

Model Configuration 9: {'batch_size': 64, 'conv_layers': 2, 'dense_units': 64, 'epochs': 5}

- Test Accuracy: 32.90%
- Notes: Increasing the batch size to 64 improved training stability, but the test accuracy did not show a significant increase.

Model Configuration 10: {'batch_size': 64, 'conv_layers': 2, 'dense_units': 64, 'epochs': 10}

- Test Accuracy: 33.32%
- Notes: The model showed an improvement in test accuracy, suggesting that a larger batch size with more epochs may improve generalization.

Model Configuration 11: {'batch_size': 64, 'conv_layers': 2, 'dense_units': 128, 'epochs': 5}

- Test Accuracy: 33.81%
- Notes: Increasing both the dense units and the batch size resulted in a small but noticeable improvement in test accuracy.

Model Configuration 12: {'batch_size': 64, 'conv_layers': 2, 'dense_units': 128, 'epochs': 10}

- Test Accuracy: 34.87%

- Notes: This configuration showed the best test accuracy among configurations with 64 batch size, indicating that a larger batch size with more dense units and epochs can improve model performance.

Model Configuration 13: {'batch_size': 64, 'conv_layers': 3, 'dense_units': 64, 'epochs': 5}

- Test Accuracy: 28.94%
- Notes: The third convolutional layer did not provide significant improvement compared to earlier configurations with 64 batch size.

Model Configuration 14: {'batch_size': 64, 'conv_layers': 3, 'dense_units': 64, 'epochs': 10}

- Test Accuracy: 31.94%
- Notes: A third convolutional layer with a larger batch size did not result in substantial improvements in test accuracy.

Model Configuration 15: {'batch_size': 64, 'conv_layers': 3, 'dense_units': 128, 'epochs': 5}

- Test Accuracy: 34.56%
- Notes: This configuration achieved better test accuracy compared to models with 64 batch size and 5 epochs, indicating that a more complex architecture might help improve performance.

Model Configuration 16: {'batch_size': 64, 'conv_layers': 3, 'dense_units': 128, 'epochs': 10}

- Test Accuracy: 35.15%
- Notes: This configuration achieved the highest test accuracy of all the models tested, making it the best-performing model in the grid search

Conclusion

The grid search results indicate that increasing the complexity of the model, such as adding more convolutional layers, increasing the number of units in the dense layer, and using a larger batch size, generally led to better training accuracy. However, the most notable finding was that **Model Configuration 7**: {'batch_size': 32, 'conv_layers': 3, 'dense_units': 128, 'epochs': 5} achieved the highest test accuracy of 35.23%. This configuration outperformed all other combinations and showed the best generalization ability to the test dataset.

Despite the high training accuracy observed with several configurations, overfitting remained an issue for many of the models. The key to improving performance lies in balancing model complexity with generalization, and **Model Configuration 7** seems to be the best trade-off between these factors in the current experiment.