Deep Learning Report3

Shivangi Rai 2321003

March 27, 2024

1 Introduction

This report presents the development of a dense neural network model tailored for the classification task on the Street View House Numbers (SVHN) dataset. The SVHN dataset comprises colored digit images, which are inherently challenging due to their real-world variability and complexity. This report covers the preprocessing of the dataset, the architecture of the neural network, and the hyperparameter tuning strategies employed to optimize the model's performance.

2 Dataset and Preprocessing

The SVHN dataset, accessible via TensorFlow Datasets, contains 32x32 pixel colored images of house numbers captured from Google Street View images. The preprocessing steps implemented for this dataset involve:

Normalization: Each image's pixel values are scaled to a range between 0 and 1 by dividing by 255. This step is crucial for stabilizing the neural network's training process by ensuring consistent data scale and distribution. Data Format: The images retain their original 3D shape of 32x32x3 (width, height, color channels) without any flattening, allowing the model to process spatial information even within a dense network architecture.

3 Neural Network Architecture

The neural network is structured as follows:

Input Layer: Accommodates the input images with the shape (32, 32, 3), matching the dimensions of the SVHN dataset images.

Flattening Layer: Converts the 2D images into 1D vectors to facilitate processing by the subsequent dense layers.

Hidden Layers:

Two hidden layers are included in the model, with the number of neurons and activation function in each layer subject to optimization through hyperparameter tuning. The Rectified Linear Unit (ReLU) activation function is employed for both hidden layers due to its effectiveness in adding non-linearity and mitigating the vanishing gradient problem.

Output Layer: Comprises 10 neurons corresponding to the ten digit classes (0-9) in the SVHN dataset, utilizing the softmax activation function to yield a probability distribution over these classes.

4 Hyperparameter Tuning

Hyperparameter tuning is conducted using the Hyperband tuner, an extension of random search with early stopping, to efficiently identify the optimal hyperparameters. The key hyperparameters subjected to tuning include:

Number of Neurons in Each Hidden Layer: The range for the number of neurons is set between 32 to 512, allowing the model to explore various levels of complexity and capacity. Learning Rate: The learning rate choices are 0.01, 0.001, and 0.0001, enabling the identification of an optimal learning pace for the Adam optimizer.

5 Training and Evaluation

The model undergoes training on the preprocessed training set, with a portion of the data reserved for validation to monitor and mitigate overfitting. The performance of the best model, as determined through hyperparameter tuning, is then evaluated based on its accuracy and loss on the validation set.

Comparing trial-25 and trial-26

Trial 25:

- **Hyperparameters:**
  - $units\_1 : 352$
  - $units\_2 : 96$
  - $learning\_rate : 0.001$
  - $tuner/epochs : 10$
- **Training Progress:**
  - The model starts with a moderate loss of 2.1425 and accuracy of 0.2347 on the training set.
  - Over epochs, both loss and accuracy improve steadily.
  - By the end of training, the loss reduces to 0.9220, and accuracy increases to 0.7204.
- **Validation Performance:**

  - The validation loss decreases from 1.6855 to 1.0129, and validation accuracy increases from 0.4157 to 0.7032.

- **Overall Assessment:**
  - The model seems to be learning effectively, as both training and validation metrics improve over epochs.
  - However, the final validation accuracy of 0.7032 suggests that there might still be room for improvement.

Trial 26:

- **Hyperparameters:**
  - $units\_1 : 320$
  - $units\_2 : 128$
  - $learning\_rate : 0.01$
  - $tuner/epochs : 10$
- **Training Progress:**
  - The model starts with a relatively high loss of 2.3889 and accuracy of 0.1870 on the training set.
  - However, the loss and accuracy do not improve significantly over epochs, indicating potential issues with the chosen hyperparameters.
- **Validation Performance:**
  - The validation metrics remain stagnant throughout training, with a final accuracy of 0.1959.
- **Overall Assessment:**
  - This trial seems to have suffered from poor hyperparameter choices, as the model fails to learn effectively.
  - The extremely low validation accuracy of 0.1959 suggests that the model is essentially making random predictions.

Comparison:

- Trial 25 outperforms Trial 26 significantly in terms of validation accuracy.
- Trial 25 has a more appropriate learning rate (0.001) compared to Trial 26 (0.01), which likely contributed to its better performance.
- The choice of units in the hidden layers also seems more effective in Trial 25, contributing to its better performance.

In conclusion, Trial 25 is more successful due to better hyperparameter choices, resulting in significantly better validation performance compared to Trial 26.

6 Conclusion and Future Work

Upon completion of the hyperparameter tuning process, the optimal model configuration is determined, including the best number of neurons in each hidden layer and the ideal learning rate. The performance metrics of this model provide insights into its classification efficacy on the SVHN dataset.

This report detailed the methodology for developing a dense neural network for digit classification on the SVHN dataset, emphasizing the preprocessing approach, neural network architecture, and hyperparameter tuning strategy. Future endeavors might explore the incorporation of convolutional layers to better capture spatial hierarchies in image data, the application of advanced regularization techniques to enhance model generalization, and the utilization of alternative hyperparameter optimization methods to further refine the model's performance.