CIFAR-10: Image Classification Report

Overview

This report outlines the process and results of building a Convolutional Neural Network (CNN) to classify images from the CIFAR-10 dataset. We employed the DenseNet architecture, pre-trained on ImageNet, leveraging transfer learning to enhance model performance.

Data Preprocessing

In this stage we decided on a data driven approach, where the focus was placed on improving the data as opposed to adding more layers in the model, essentially meaning a simpler model. The issues we faced with the data classification led us to this conclusion as boats, aeroplanes and birds were being classified together along with trucks and cars in one category. We concluded that adding more layers in the model would not resolve this issue and improving the data processing step was imperative to get more accurate results.

Here are the preprocessing steps taken:

1. **Inserted Grayscale Conversion in Data Preprocessing:** This step was done before the normalisation & data augmentation steps to ensure that all transformations were applied correctly to the grayscale images. We ultimately decided to do this as removing colour channels meant the model processes less data, which can lead to faster training times and reduced model complexity.

2. **Normalization:** Image pixel values were scaled to a [0, 1] range to facilitate faster convergence.

3. **One-Hot Encoding:** Class labels were encoded into a one-hot format to match the output layer's expectation.

CNN Architecture: DenseNet

After extensive research into the various CNN architectures available, mainly ResNet, VGG16, Inception etc we decided upon DenseNet as our priority was reducing overfitting. Its architecture allows it to achieve similar or better performance with fewer parameters compared to deeper models like VGG16, which is a significant advantage on a limited dataset like CIFAR-10.

DenseNet

- DenseNet (Densely Connected Convolutional Networks) is particularly good for tasks where model efficiency and feature reuse are crucial. It can achieve similar or even superior performance to ResNet with fewer parameters, thanks to its dense connections.
- *Specific Model:* **DenseNet-121** or a smaller version can be quite effective for CIFAR-10 due to its efficient use of features and reduced risk of overfitting.

Modifications for CIFAR-10:

- **Input Layer:** CIFAR-10 images are small (32x32), but DenseNet expects larger inputs, hence images were resized to match the input size requirements.
- **Base Layers:** All original DenseNet convolutional layers, frozen to retain learned ImageNet features.
- **Additional Layers:** A Global Average Pooling layer followed by two dense layers, the latter having 512 units with 'relu' activation and a final output layer with 10 units (corresponding to CIFAR-10 classes) with 'softmax' activation.

Training Process

- **Optimizer:** Adam with a learning rate of 0.0001, chosen for its adaptive learning rate capabilities, which make it robust to initial learning rate settings and ideal for transfer learning scenarios.
- **Loss Function:** Categorical crossentropy, suitable for multi-class classification problems. Our initial choice was Sparse Categorical crossentropy which we had to reject as we had too many errors and realised its incompatibility with labelled categorical data.
- **Batch Size:** 64, balancing training speed and model performance.
- **Epochs:** 50, with early stopping implemented to prevent overfitting. Early stopping monitored validation loss and halted training if no improvement was seen for 5 consecutive epochs.

Results and Model Performance

- **Accuracy and Loss Trends:** The model achieved a training accuracy of over X% and a validation accuracy of around X%. The training and validation loss decreased steadily, indicating good model convergence without signs of overfitting, as evidenced by the parallel reduction in validation loss.

- **Confusion Matrix and Classification Report:** Provided insights into class-specific performance, revealing higher misclassifications between certain classes like cats and dogs, which are visually similar.

Insights Gained

- **Importance of Data Preprocessing:** Properly normalized and encoded data significantly impacted model training effectiveness.
- **Effectiveness of Transfer Learning:** Using a pre-trained model reduced the need for an extensive dataset and lowered training times while improving model robustness.
- **Parameter Tuning:** Adjustments in learning rate and batch size showed noticeable effects on model performance, highlighting the importance of hyperparameter optimization in neural network training.

Visualizations

Visual aids included plots of training/validation accuracy and loss over epochs, and a heatmap of the confusion matrix to highlight performance across different classes.

Conclusion

The use of DenseNet with transfer learning proved to be highly effective for the CIFAR-10 image classification task. However, irrespective of the model chosen, the most important thing we learnt is that experimentation is crucial and often, the best way to determine the optimal model is through empirical testing and tuning.

Future work could explore additional enhancements such as more aggressive data augmentation, further hyperparameter tuning, or experimenting with other advanced CNN architectures like Vision Transformers (ViT) or Xception for potentially even better performance.

This project demonstrated the power of CNNs, particularly when combined with strategies like transfer learning, in handling complex image classification tasks efficiently and effectively.