COS30082 - Applied machine learning

Assignment 1

Name: Tran Hoang Hai Anh

Student ID: 104177513

1. Methodology

1. Using a base pretrained model

The BirdClassifier is based on the ResNet-50 backbone, which is a popular and successful deep neural network. ResNet-50 was chosen primarily for its residual learning capabilities, which allow for the creation and training of very deep network topologies without encountering the vanishing gradient problem. This is accomplished using skip connections, which allow information to travel unhindered through the network levels. The BirdClassifier uses this pre-trained model to profit from the massive hierarchy of picture characteristics learnt on the large and diverse ImageNet dataset, which contains over a million annotated photos across thousands of categories. This feature-rich model is especially well-suited to the demanding job of discriminating between various bird species in the CUB-200 dataset, which is notorious for its difficult, fine-grained picture classification problems.

Transfer learning is used to improve the performance of the model even further. This strategy allows the BirdClassifier to use previously learned patterns and features from ResNet-50, drastically reducing the need for vast amounts of training data. Transfer learning also speeds up the training process and improves accuracy because the model does not have to start from zero. Given the CUB-200 dataset's inherent quality and quantity concerns, transfer learning offers a reliable solution by fine-tuning the pre-trained model for bird species categorization.

2. Fine tuning process

The BirdClassifier uses a selective fine-tuning technique to strike a compromise between preserving important general traits and tailoring the model to the particular of bird species detection. Tuning_start_layer is a critical parameter that determines which layers of the network may be trained during the fine-tuning phase. In the present arrangement, the layers before the fifth are frozen, which means they keep their pre-trained weights and are not changed during training. This guarantees that the universal picture properties learnt from the ImageNet dataset, such as edge and texture recognition, remain intact. These broad properties are very portable and apply to the majority of picture classification applications, including bird recognition.

The model is fine-tuned for bird species-specific patterns in layers 5 and up. This selective strategy guarantees that the model responds to the particular constraints given by the CUB-200 dataset, which necessitates thorough separation between similar-looking species. When more thorough fine-tuning is required, setting tuning_start_layer to a negative number makes all layers trainable, boosting the model's ability to adapt to the dataset's unique properties.

3. Overfitting issues

The BirdClassifier employs a variety of regularization strategies to prevent overfitting and increase generalization. The 0.5-rate dropout layer prevents the model from being unduly reliant on any particular neuron or collection of features, allowing the network to learn more generalist and redundant feature representations. In addition, by freezing the initial layers, the model decreases the danger of overfitting on abstract characteristics, which are already well-suited for general picture categorization. This regularization strategy allows the model to perform effectively over a wide range of bird species, despite the sparse and possibly noisy data in the CUB-200 dataset.

4. Data processing

The BirdClassifier's success relies heavily on effective data preparation. Training pictures go through a number of modifications, including resizing, center cropping, tensor conversion, and normalization. These transformations guarantee that the input data meets the model's criteria and that the model is trained using standardized inputs. This allows the model to generalize more effectively by seeing a wider variety of data changes, making it more resilient to diverse picture circumstances and views.

5. Output layer blueprint

The basic output layer of ResNet-50 is intended to categorize the 1,000 classes in the ImageNet dataset. To make the model acceptable for bird species classification, the BirdClassifier contains a custom output layer that is tuned to the 200 bird species in the CUB-200 dataset. The redesigned output layer has the following components:
A linear layer decreases the feature dimensionality to 512, resulting in a more compact and useful feature representation.
A ReLU activation function is used to induce nonlinearity, which allows the model to capture more complicated patterns in the input.
A dropout layer with a dropout rate of 0.5 is used to randomly deactivate certain neurons during training, boosting resilience and generalization.
A final linear layer transforms the 512-dimensional characteristics into 200 output classes, each representing a different bird species.

6. Training, Validating and Evaluation

The BirdClassifier is trained using many important strategies that assure efficiency and robustness. The model employs Cross Entropy Loss, which is ideal for multi-class classification issues such as this one. Adam is the preferred optimizer, as it provides adaptable learning rates and momentum, allowing for speedier convergence. The initial learning rate is set to 0.001, which achieves a nice balance of stability and convergence speed.

To prevent overfitting during training, stop training is used. If the validation loss does not improve after a set number of epochs (patience), training is stopped to keep the model from becoming overly specialized to the training data. Additionally, a learning rate scheduler named ReduceLROnPlateau is used. If the validation loss reaches a plateau, this scheduler automatically slows the learning rate by 0.1, allowing the model to fine-tune its weights more efficiently in the final phases of training.

To evaluate the model's performance, the BirdClassifier includes an evaluation function that calculates both Top-1 accuracy and average accuracy per class. This guarantees that the model's performance is assessed both overall and per-class, resulting in a thorough assessment. The application of these parameters ensures that the classifier works well not just on the most frequent bird species, but also on more difficult, underrepresented ones.

2. Results and Discussion

1. Results

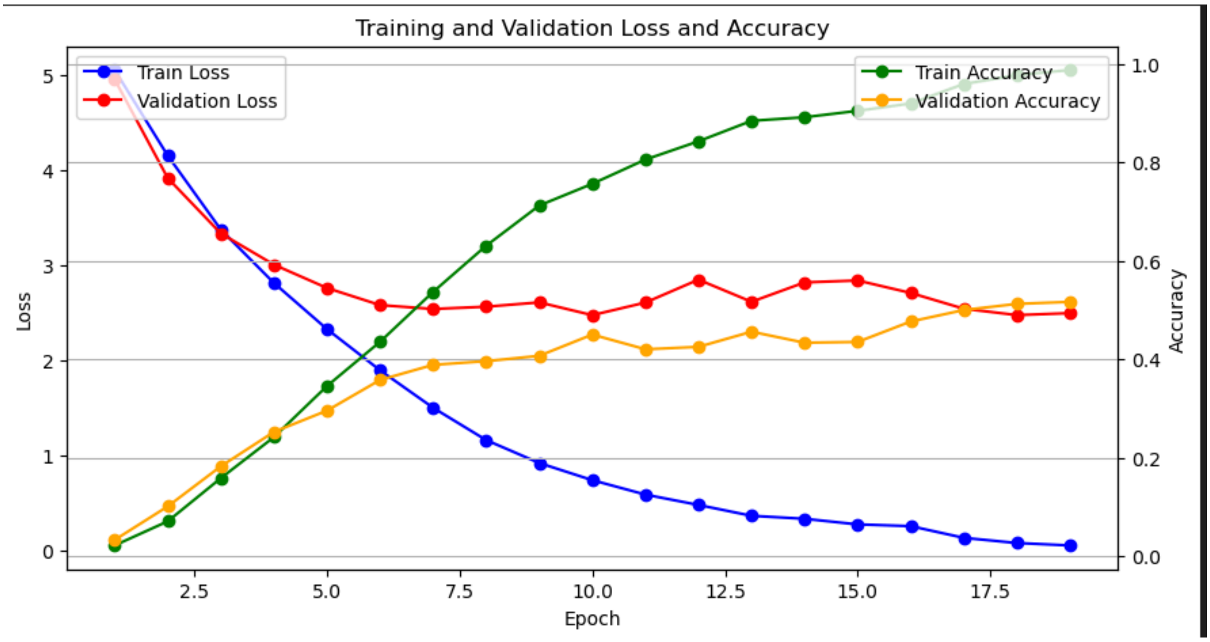

Fig1: Visualizing the Training and Validation Loss and Accuracy

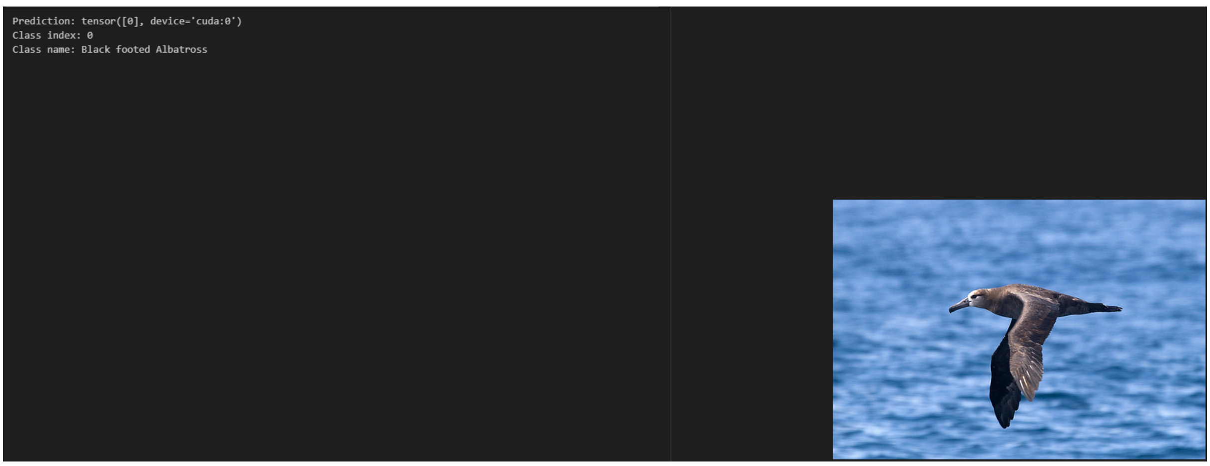

Fig2: Predicting the first image in the testing dataset

2. Analytics

1. Training performance

At the start of training (epoch 1), the model exhibits a high train loss of 5.06 and a very poor train accuracy of 2.2%. This suggests that the model initially fails to accurately categorize the bird species, as is typical when training on a complicated dataset like CUB-200. However, as the epochs continue, there is a definite pattern of improving training loss and accuracy. By epoch 10, the train loss has dropped to 0.74, while the train accuracy has increased to 75.6%, indicating that the model is learning successfully and modifying its weights to reduce loss.

In the last epoch (19), the model had a train loss of 0.059 and a train accuracy of 98.7%. This near-perfect accuracy shows that the model has learnt to properly categorize the majority of training data, which might indicate overfitting if validation performance does not match this increase.

2. Validation performance

The model's performance on the validation set reveals how well it can generalize to previously unknown data. At epoch 1, the validation loss is 4.96 and the validation accuracy is 3.3%, which is consistent with the poor start performance on the training set. However, as the model trains, both validation loss and accuracy improve, showing that it is learning relevant characteristics for bird species categorization.

By epoch 10, the validation accuracy has increased to 45.0%, a major improvement from the original 3.3%. The validation loss is reduced to 2.48, indicating that the model is generating better predictions on unseen data. This consistent improvement in validation performance indicates that the model is generalizing successfully up to this point.

In the last epoch (19), the model has a validation accuracy of 51.7% and a validation loss of 2.50. This accuracy, while not as high as the training accuracy, suggests that the model is pretty good at generalizing to new bird photos, albeit there is still potential for improvement.

3. Overfitting

One striking discovery is the growing disparity between training and validation accuracy. By epoch 19, the train accuracy is about 99%, while the validation accuracy is only about 52%. This mismatch shows that the model is overfitting to the training data, especially in the later phases of training. Overfitting happens when the model becomes overly specialized in learning patterns particular to the training set, reducing its effectiveness on new data.

The validation loss varies somewhat after epoch 10, showing that the model is struggling to develop further on the validation set. This might be an early indication that the model has reached its generalization limit for this dataset.The validation loss varies somewhat after epoch 10, showing that the model is struggling to develop further on the validation set. This might be an early indication that the model has reached its generalization limit for this dataset.

4. Learning performance

The model's performance progressively increases during the 19 epochs, with both training and validation accuracy exhibiting continuous increasing trends. Early stop training and a learning rate scheduler might assist to reduce overfitting and allow the model to focus on improving validation performance. If the learning rate gets too low after a given number of epochs, modifications to the learning rate or other hyperparameters may be considered in order to sustain further development.

3. In conclusion

In conclusion, the BirdClassifier has a robust learning trajectory, with significant gains in both training and validation accuracy across 19 epochs. However, the growing disparity between the high training accuracy (98.7%) and the relatively moderate validation accuracy (51.7%) indicates that the model is likely overfitting to the training data. To solve this, adding further regularization approaches, such as raising the dropout rate or expanding data augmentation, may help the model generalize more effectively. Fine-tuning the learning rate schedule or implementing a more severe early stop training criterion may also help to prevent overfitting in later epochs. While the model performs well, obtaining respectable accuracy on previously encountered data, more tweaks are required to improve its generalization capabilities.