Giorgio Mendoza
CS549-E24-E01
Dr. Alexandros Lioulemes

Week 9 Lab

Part A_1:

Objective:
The objective of this task is to perform transfer learning by adding a new classifier to a pretrained MobileNet V2 model, which has been trained on the ImageNet dataset. The new classifier will be trained from scratch using a dataset of cats and dogs. This approach leverages the feature extraction capabilities of the MobileNet V2 model while fine-tuning it for a specific binary classification task.

Code Description:
The code starts with data preprocessing, where the Cats and Dogs dataset is downloaded, extracted, and divided into training, validation, and test sets. The images are resized to 160x160 pixels, and the data is batched for efficient processing. To enhance the diversity of the training data, data augmentation techniques such as random horizontal flipping and rotation are applied, making the model more robust to variations in input data.
The MobileNet V2 model is loaded with pretrained weights, excluding its top layer, and a new classifier is added, consisting of a global average pooling layer followed by a dense output layer for binary classification. The base model's weights are frozen, preventing them from being updated during training. The new classifier is then trained for 10 epochs using the Adam optimizer and binary cross-entropy loss. After training, the model is evaluated on the test dataset, and its accuracy is reported.
Finally, the training process is visualized by plotting the accuracy and loss for both training and validation sets over the epochs. These plots provide insights into the model's performance, indicating how well it generalizes to new data and whether it is overfitting, underfitting, or well-fitted.

Results:
After 10 epochs, the MobileNet V2 model demonstrates effective transfer learning by achieving over 90% accuracy on both training and validation datasets, with close alignment indicating good generalization. The consistent decrease in loss further confirms the model's successful adaptation to the new task.

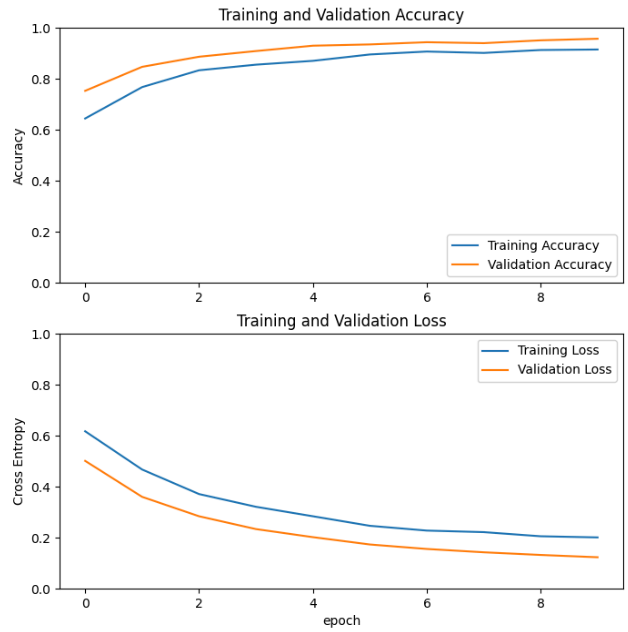

Figure 1. Training & Validation Accuracy and Training & Validation Loss

PART A_2:

Objective:
The objective of this task is to reimplement the given MATLAB tutorial in Python using TensorFlow and Keras to classify images from the Merch dataset. The task involves leveraging transfer learning by using two pretrained networks, VGG-19 and Inception-V3, to build and train a new classifier. The goal is to freeze the initial layers of these pretrained models, augment the training data, train the networks with similar hyperparameters, and compare the performance of the two models in terms of accuracy and loss.

Code Description:

For this task Google Collab was used because the A100 GPU is much faster than my PC to run the libraries used. The implementation begins with setting up the environment by loading the Merch dataset from Google Drive and applying data augmentation using the `ImageDataGenerator` from TensorFlow. The data is split into training and validation sets, and images are preprocessed to match the input size expected by the VGG-19 and Inception-V3 models.

Two pretrained models, VGG-19 and Inception-V3, are loaded with their weights trained on the ImageNet dataset. The initial layers of both models are frozen to retain the learned features, and a global average pooling layer is added to each model to reduce the spatial dimensions. Fully connected layers are then appended to both models to enable classification into the five categories of the Merch dataset. Additionally, a dropout layer is included in the VGG-19 model to mitigate overfitting.

The models are compiled using the Adam optimizer with a specified learning rate, and they are trained on the augmented dataset for 10 epochs. Early stopping and model checkpointing callbacks are utilized during training to prevent overfitting and to save the best-performing models. After training, the models are evaluated on validation data to compare their performance in terms of accuracy and loss.

Results:
The VGG-19 model showed a steady improvement in accuracy and a corresponding decrease in loss over the course of 10 epochs. Initially, the model struggled with an accuracy of 41.67% on the training set and 42.86% on the validation set. However, by the end of training, the accuracy on the training set had increased to 81.25%, with a validation accuracy of 78.57%. The loss values reflected this improvement, decreasing from 1.54 to 0.59 on the training set, and from 1.35 to 0.67 on the validation set. These results could indicate that the VGG-19 model was able to learn effectively and generalize well to unseen data.

Similarly, the Inception-V3 model demonstrated significant progress during the training process. Starting with an accuracy of 12.50% on the training set and 28.57% on the validation set, the model improved to achieve a perfect 100% accuracy on the training set and 78.57% on the validation set by the final epoch. The loss also decreased substantially from 1.93 to 0.50 on the training set and from 1.52 to 0.73 on the validation set. These results could indicate that the Inception-V3 model also learned effectively, although its performance was comparable to VGG-19 on the validation set.

```
Epoch 1/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step - accuracy: 0.4167 - loss: 1.5365 - val_accuracy: 0.4286 - val_loss: 1.3526
Epoch 2/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step - accuracy: 0.4688 - loss: 1.2707 - val_accuracy: 0.5714 - val_loss: 1.1598
Epoch 3/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.5714 - val_loss: 1.1437
Epoch 4/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step - accuracy: 0.6562 - loss: 0.9999 - val_accuracy: 0.5714 - val_loss: 1.0332
Epoch 5/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 1s/step - accuracy: 0.5417 - loss: 1.0606 - val_accuracy: 0.7143 - val_loss: 0.9984
Epoch 6/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 208ms/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.6429 - val_loss: 1.1344
Epoch 7/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 3s 3s/step - accuracy: 0.5833 - loss: 1.0182 - val_accuracy: 0.8571 - val_loss: 0.6133
Epoch 8/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 287ms/step - accuracy: 0.8125 - loss: 0.5736 - val_accuracy: 0.8571 - val_loss: 0.7023
Epoch 9/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 207ms/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.7143 - val_loss: 0.6508
Epoch 10/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 2s 2s/step - accuracy: 0.8125 - loss: 0.5925 - val_accuracy: 0.7857 - val_loss: 0.6724
```

Figure 2. VGG19 model results

```
Epoch 1/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 16s 16s/step - accuracy: 0.1250 - loss: 1.9310 - val_accuracy: 0.2857 - val_loss: 1.5152
Epoch 2/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 4s 4s/step - accuracy: 0.2500 - loss: 1.6069 - val_accuracy: 0.6429 - val_loss: 1.4101
Epoch 3/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 231ms/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.6429 - val_loss: 1.3412
Epoch 4/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 908ms/step - accuracy: 0.6250 - loss: 1.2249 - val_accuracy: 0.6429 - val_loss: 1.2039
Epoch 5/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 223ms/step - accuracy: 0.7500 - loss: 1.0496 - val_accuracy: 0.7857 - val_loss: 1.0376
Epoch 6/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 214ms/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 0.7857 - val_loss: 1.0172
Epoch 7/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 938ms/step - accuracy: 0.7917 - loss: 0.8085 - val_accuracy: 0.7857 - val_loss: 0.8934
Epoch 8/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 232ms/step - accuracy: 0.9375 - loss: 0.6635 - val_accuracy: 0.8571 - val_loss: 0.6558
Epoch 9/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 0s 216ms/step - accuracy: 0.0000e+00 - loss: 0.0000e+00 - val_accuracy: 1.0000 - val_loss: 0.6753
Epoch 10/10
1/1 ━━━━━━━━━━━━━━━━━━━━ 1s 928ms/step - accuracy: 1.0000 - loss: 0.5015 - val_accuracy: 0.7857 - val_loss: 0.7346
```

Figure 3. InceptionV3 model results

One observation is that for some epochs, the accuracy is logged as 0.0 for both models. This might suggest that the batch size is small or the classes are underrepresented in certain batches. Another possibility is an issue with the data augmentation or generator, where an empty batch could have been processed, resulting in zero accuracy and loss for that epoch. Despite these occasional anomalies, both models demonstrated strong learning capabilities, with VGG-19 showing consistent improvements and Inception-V3 obtaining higher accuracy on the training set by the end of the training period.