CAP 6618 Spring 2019
Christian Garbin
Module 7 Assignment - Supervised Image Classification Using TensorFlow

Objective of this assignment

The objective of this assignment is to *"[d]esign and implement a complete solution to the problem of image classification using TensorFlow."*

Files submitted

One Jupyter notebook is submitted as part of this assignment, CAP6618_PP5_cgarbin_final.ipynb.

The notable parts of the notebook are:

- Section 2: experiments are defined with the help of the `ExperimentData` class. In section 2.2 a specific experiment is configured. The remainder of the notebook is driven by the experiment data.
- Section 5: loads the dataset accounting for possible class imbalances.
- Section 6: the training function supports early stopping, has a more granular progress indicator and saves the model at the end of each epoch, to go back to the best model when early stopping is invoked.

Structure of this report

Because this will be longer than the usual report, the structure of the report will follow the inverted pyramid structure. The conclusions will be presented first, followed by the details of the datasets used in the experiments. The report ends with a few samples of the datasets.

Conclusions

The major lesson learned with these experiments is that transfer learning is not a simple exercise of taking a trained network and swapping its classifier. This was already clear in the "When and how to fine-tune" section of CS231n's transfer learning chapter, but stumbling through it in real life was a memorable learning exercise. However, to do that I had to try several datasets and I cannot claim I had a bigger plan for that. I was lucky that the chosen datasets had such different characteristics.

The most important takeaway is to first understand well the domain on which the pretrained model was trained and the domain and size of the new dataset. If they significantly overlap, retraining only the classifier layer(s) will likely yield good results, as the "natural image" dataset showed. If the domains don't overlap, we may need a completely new classifier and perhaps even additional techniques (such as image segmentation to find the area of interest, useful in Harvard's HAM10000) or unfreeze some of the layers (when there is some overlap for specific areas, as it was the case with Stanford's Cars dataset).

The other lesson learned in this experiment was to not rely on accuracy when working with highly imbalanced datasets (Harvard's HAM10000) and to think of the base rate estimate from the start. Until the confusion matrix

was put in place, the dilettante author of this report could not understand why training accuracy was high, while the prediction of individual item was abysmal. After seeing the confusion matrix, he made the connection with the base rate estimation for an imbalanced dataset. This would be a good time to learn about better metrics (e.g. F1 score), and be less fixated on "accuracy".

Experiments performed

In this experiment the pretrained network was kept constant and the datasets varied. The network was the same used in the starter notebook, an Inception V3 network trained on the ImageNet dataset. The network is available in the TensorFlow repository. The code is based on Géron's *Hands-on Machine Learning with Scikit-Learn and Tensorflow,* chapter 13, available here.

Four datasets were used in the experiments.

- 5-class flower from Géron's notebook
- Harvard's HAM10000 ("skin cancer MNIST")
- Natural images
- Stanford Cars dataset

The appendix has samples from each dataset.

The following sections explain the datasets, the reason to choose them and the results. Each dataset has different characteristics, allowing the experiments to verify how a network pretrained in general categories performed in those cases.

5-class flower dataset

From the 5-class flowers dataset in https://www.tensorflow.org/tutorials/image_retraining.

This is the dataset from the original notebook. It is used to check the functionality of the code. Without fine tuning it should achieve over 60% accuracy.

Why this dataset was used: It provides a baseline to test the code. It should achieve over 60% accuracy in ten epochs or fewer.

Results from the experiment: It achieved 66% on the test set after about ten epochs of training, similar to what Géron's original notebook achieved. This fulfills its basic function of checking that the code in the notebook is still sound, after all the adaptations done for this assignment.

Classes and confusion matrix:

```
['daisy', 'dandelion', 'roses', 'sunflowers', 'tulips']
[[125   0   0   2   0]
 [ 46 126   0   8   0]
 [ 14   1  69  34  11]
 [ 73   3   2  61   1]
 [ 26   3   5  38  88]]
```

Harvard's HAM10000 ("skin cancer MNIST")

From Harvard's Dataverse HAM10000, a large collection of multi-source dermatoscopic images of common pigmented skin lessions, downloaded via Kaggle, where it is described as the "skin cancer MNIST", to "*get biology and medicine students more excited about machine learning and image processing.*"

> *Training of neural networks for automated diagnosis of pigmented skin lesions is hampered by the small size and lack of diversity of available dataset of dermatoscopic images. We tackle this problem by releasing the HAM10000 ("Human Against Machine with 10000 training images") dataset. We collected dermatoscopic images from different populations, acquired and stored by different modalities. The final dataset consists of 10015 dermatoscopic images which can serve as a training set for academic machine learning purposes.*

Why this dataset was used: It tests the pretrained network on a domain it was not trained before (medical images). This gives an idea of how powerful the concept transfer is (or is not). It is also a highly imbalanced dataset, another complicating factor machine learning.

Results from the experiment: Although this dataset achieved almost 70% accuracy during training, it performed poorly when verifying the actual predictions. Almost all samples in the test dataset were predicted as class 5, the class with the highest number of samples by far. This means we basically trained a classifier that returns "5" for all samples, and that the 70% "accuracy" is nothing more than our base rate for this imbalanced dataset. All in all, the experiment with this dataset can be filed under "great learning experience". A few lessons learned: 1) imbalanced datasets are hard to work with, 2) do not use a network trained on natural images to classify images from a completely different domain (or at least be prepared to unfreeze some of the CNN layers), 3) do not perform random image manipulations when there is clearly one region of interest in the image…

Classes and confusion matrix:

```
['akiec', 'bcc', 'bkl', 'df', 'mel', 'nv', 'vasc']
[[   0    0    0    0    0   66    0]
 [   0    0    0    0    0  103    0]
 [   0    0    0    0    0  220    0]
 [   0    0    0    0    0   23    0]
 [   0    0    0    0    1  222    0]
 [   0    0    0    0    1 1340    0]
 [   0    0    0    0    0   29    0]]
```

Natural Images

Downloaded from Kaggle, originally created for the paper Effects of Degradations on Deep Neural Network Architectures.

It contains eight classes, compiled from other datasets:

- Airplane images obtained from http://host.robots.ox.ac.uk/pascal/VOC
- Car images obtained from https://ai.stanford.edu/~jkrause/cars/car_dataset.html
- Cat images obtained from https://www.kaggle.com/c/dogs-vs-cats

- Dog images obtained from https://www.kaggle.com/c/dogs-vs-cats
- Flower images obtained from http://www.image-net.org
- Fruit images obtained from https://www.kaggle.com/moltean/fruits
- Motorbike images obtained from http://host.robots.ox.ac.uk/pascal/VOC
- Person images obtained from http://www.briancbecker.com/blog/research/pubfig83-lfw-dataset

Why this dataset was used: It combines images from important datasets, such as PASCAL VOC, Stanford Cars dataset and ImageNet. Most of the images are from the domain the pretrained model was tested on, so the expectation is that it will perform well.

Results from the experiment: This dataset achieved almost 100% with only a few epochs. This is likely a side effect of the large overlap between its classes and the classes in ImageNet (used to train the model we are using).

Classes and confusion matrix:

```
['airplane', 'car', 'cat', 'dog', 'flower', 'fruit', 'motorbike', 'person']
[[146   0   0   0   0   0   0   0]
 [  0 194   0   0   0   0   0   0]
 [  0   0 177   0   0   0   0   0]
 [  0   0   2 139   0   0   0   0]
 [  0   0   0   0 169   0   0   0]
 [  0   0   0   0   0 200   0   0]
 [  0   3   0   0   0   0 155   0]
 [  0   0   0   0   0   0   0 198]]
```

Stanford Cars

The Stanford Cars dataset, downloaded from Kaggle, where it is already split in folders.

The content of the dataset:

> *The Cars dataset contains 16,185 images of 196 classes of cars. The data is split into 8,144 training images and 8,041 testing images, where each class has been split roughly in a 50-50 split. Classes are typically at the level of Make, Model, Year, e.g. 2012 Tesla Model S or 2012 BMW M3 coupe.*

Why this dataset was used: This dataset tests what the pretrained model can do with a dataset that is not completely unknown, but contains only a fine-grained version of categories used to train the model. As described in Stanford's FGComp 2013 site (emphasis added), "*This challenge will target fine-grained classification, i.e. classification among categories which are both visually and semantically very similar. This is a very difficult regime which is even challenging for humans without careful training, and is critical for establishing a more detailed understanding of the visual world. Specifically, this challenge will span a range of domains, and within each domain will test classification among the many fine-grained categories.*"

Results from the experiment: Never surpassed 10% accuracy during training and achieved 9% accuracy on the test data. This is similar to the result of the Infor_FG team in the FGComp 2013 competition. The team achieved 4.45% accuracy in the "Cars" category using a network based on the original ImageNet paper network (although it is unclear if the network was pretrained on ImageNet itself - likely not). The CafeNet team achieved 80% by fine-tuning more layers: "*pre train[ing] the lower levels of [a network based on the one that*

won the 2012 ImageNet challenge] with a large collection of images from Imagenet to learn the most generic visual features at different levels. At the time of fine-tuning, we remove the 2 top trained layers, the classifier and the fully connected hidden layer, and replace those by a much smaller hidden layer and a classifier for the specific task.".

NOTE: Because this is a very large dataset, only the 'test' folder was used (which was then partitioned in a train and a test dataset). Some images also caused `IndexError` when loading, which resulted in creating a cleanup step in section 3, to toss out those images. Another lesson: don't trust the dataset blindly. Fortunately the number of images removed is small, not affecting the experiment.

The large number of classes and low accuracy make the confusion matrix uninteresting in this case:

```
[[1 0 0 ... 0 0 0]
 [0 0 0 ... 0 0 0]
 [0 0 1 ... 0 0 0]
 ...
 [0 0 0 ... 0 0 0]
 [0 0 0 ... 0 4 0]
 [0 0 0 ... 0 1 0]]
```

Appendix

Samples from the 5-class flower dataset

Samples from the skin cancer MNIST dataset

Class: akiec

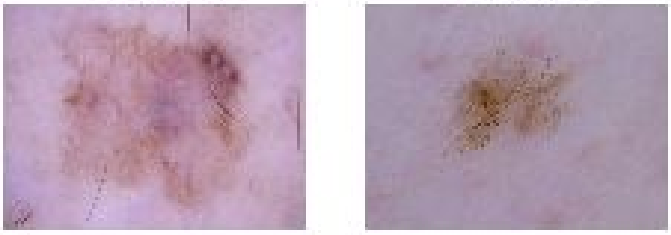
600x450 600x450

Class: bcc

600x450 600x450

Class: bkl

600x450 600x450



Samples from the natural image dataset

Class: airplane

300x104

286x113

Class: car

100x100

100x100

Class: cat

236x396

438x242

Samples from the Stanford Cars dataset

Class: AM General Hummer SUV 2000

96x64

250x144

Class: Acura Integra Type R 2001

900x600

800x458

Class: Acura RL Sedan 2012

640x450

425x255

Class: Acura TL Sedan 2012

1024x683

640x480