CAP 4611 – Fall 2022 – Project 3

Due: November 30, 2022

Instructions:

- Remember that the homework is **individual** work.
- Points add up to 40, with 30 + 10 bonus points

Homework journal:

For this homework, you need to document the process of the solution in a Homework Journal. This is a series of screenshots on the progress of the homework, labeled as nnn-text.jpg. Screenshots should document the tries, challenges, the breakthroughs etc. of the process.
The easiest way (on Windows) to do this is by hitting Windows Key + PrtScr when a step is achieved. You can crop and select the pictures later.
Please crop the pictures from any private information from your screen.

Step 1. Set up you deep learning environment (12 pts)

Set up a deep learning environment on your computer. At the end of this step, you should be able to run TensorFlow 2.x in a Jupyter notebook under Python 3. To ensure that you have a working environment, try to run the tensorflow quickstart tutorial:

https://www.tensorflow.org/tutorials/quickstart/beginner

The purpose of this exercise is for you to understand what is required to do this **on your own computer**. Using Google Colab or other preconfigured tools is **not** acceptable. You can use Anaconda.
If you have an Nvidia video card, configure Tensorflow to use it.
What to submit: the homework journal for this installation.

Step 2: Set up training for RaccoonSquirrelWombat (18 pts)

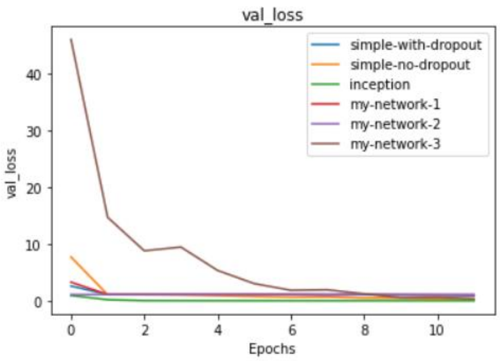
Download from the class webpage the P3-RaccoonSquirrelWombat file. Unzip it in a convenient location on your computer.
Orient yourself in the content of the unzipped directory:

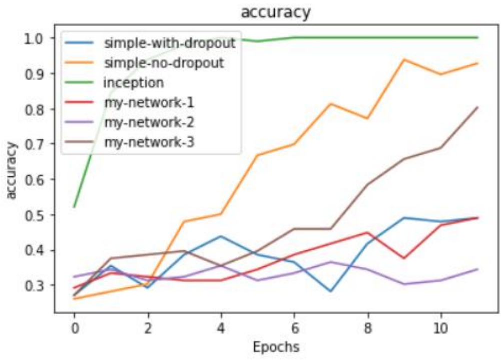
- CAP4611-P3-DeepLearning.ipynb contains the code for the training framework as a notebook. Basically, the content of this file trains a number of different networks as classifiers for the raccoon-squirrel-wombat dataset and compares them.
- ImageDataset.py contains several auxiliary functions for loading the training data. You should not change this.
- The data subdirectory contains the training and validation data (as directories). Each of these have subdirectories for raccoons, squirrels and wombats.

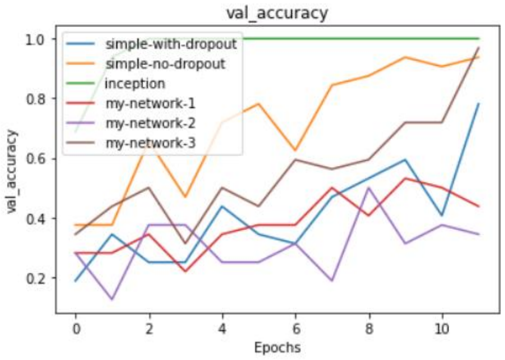
- Under the _Temporary directory, the training will create checkpoints of the trained networks as the training proceeds. Each of the networks is saved in its own directory. If the network already exists, it will not train the network again, just loads it.

Run all the cells of CAP4611-P3-DeepLearning.ipynb in Jupyter. Perform the necessary configurations fixes that might be necessary. As a note: depending on the performance of your computer, this training might take some time (from maybe 30 seconds to tens of minutes). Should you need to interrupt the training or the system crash, the next time you run it, it will start from where you left it. If all the networks are trained, it will skip the training and will just plot the comparison.

What to submit: the homework journal for this step. Make sure that you screenshot the results, which should look like this:







Step 3: Improve the classification (10 pts)

In the CAP4611-P3-DeepLearning.ipynb file you will find functions to create 3 networks: create_my_network_1, 2 and 3. As you see from the graphs above, these do not work very well.

Your task is to experiment in finding a network that classifies the best the dataset. Note that if you change the network x, you will also need to remove the corresponding subdirectory in _Temporary, for the retraining to happen.

Please look around for different models on the web but be aware that what works on other datasets and other computers might not work on this (very small) dataset and on your computer which might or might not be performant. Things that you might want to vary:

- Number of layers in the network
- Number of channels in the convolutional network
- Number of nodes in the fully connected layer
- Activation function
- Without or with dropout
- Optimizer
- Learning rate
- Use a different pre-trained network as a starting point (but please only use this idea for one of the choices).

Document what was the setup and how well it trained (training/validation accuracy and loss). Comment on your impressions on the performance of the model, and what do you learned out of it. The type of comment I want you to make is along the lines:

“Trial #765: Network with 800 convolutional layers with 2 channels each, with dropout 0.9 after every layer. Training went very slow, taking >6 hours. The training loss kept going up and down, while the validation accuracy, after reaching value = 0.4, did not improve at all. Conclusion: very deep, very narrow networks with excessive dropout do not train well.”

Retain in the create_my_network_1,2,3 functions the three different networks that appear to work well.

Submit: homework journal + ipynb file. Make sure that you screenshot the results.