

Transfer Learning Project

AHCS, Fall 2023

The goal of this project is for you to create your own Transfer Learning program. You will need to do all of the following:

- Use a dataset different from the defungi dataset we used in class. You can use the UC-Irvine Machine Learning Repository, Kaggle or Huggingface to find new datasets. See the Blackbaud Topics page "Transfer Learning" for links.
- You will need to work alone on this project, I want everyone to implement an ML model from start to finish by themselves.
- Feel encouraged to cut down whatever dataset you downloaded to make it more amenable to computation on your computer. A couple of suggestions of how to do this:
 - Use only certain categories/classes, which you can probably accomplish by just grabbing folders from the downloaded dataset.
 - Use only some of the images for each category/class, which may be best done by cranking out a quick script to move/copy some images into new folders.
- Customize your project in some additional ways (pick at least 3 things, links are there to give you a start on how to do each of these):
 - Transfer from a [different model](#) besides Resnet.
 - [Save](#) your model. This should either be paired with:
 - A separate program that loads the model and uses it to classify images, preferably from outside the downloaded dataset. ([Example here](#) with images from inside the dataset.)
 - Checkpoints that save the model automatically every so often during training. ([Example here](#))
 - Use matplotlib to graph the loss and accuracy (both training and validation). [This is an example](#) where I do that, but with output from pytorch instead of tensorflow, so it would need to be heavily modified. I also only had training loss there, you would need four line graphs, I only had three. You should be able to run the existing parseEpochs.py on any of the .txt files in that repository.
 - Use a [class-based model](#) instead of the sequence of functions/"Functional API" version that I used when I set up the Transfer Learning. ([Example here](#))
 - Do a comparison using at least 3 different (each collection counts separately):
 - [Optimizers](#)
 - [Loss functions](#) (some loss functions need different output format than what we had set up).
 - Other things that you find interesting.