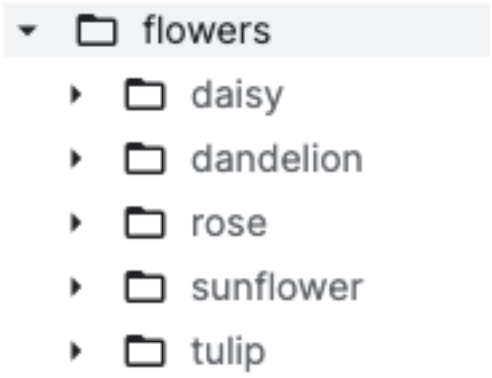
Assignment 4 Section 1: Flower Classification using LeNet5

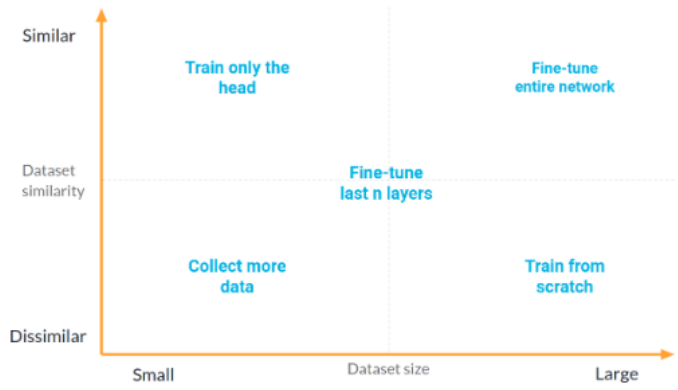
Step 0: Download Data
In this assignment, you'll be modeling a small dataset of fewer than 5000 natural images. Download the flower data from << https://s3.amazonaws.com/video.udacity-data.com/topher/2018/September/5baa60a0_flower-photos/flower-photos.zip >>, save it in the home directory of your program notebook and extract the zip file to get the directory



Step 1: Load the dataset in DataLoaders
Step 2: Visualize the Dataset
Step 3: Build a LeNet5 CNN Mode to flower classification
Step 4: Define Loss Function
Step 5: Define Optimizer (SDG)
Step 6: Training Loop (including Training & Validation & save checkpoints)
Step 7: Select Best Model (save the model)
Question: Show the Training and Testing Accuracy Graph
Step 8: Testing (Prediction)
Question: What is the testing accuracy?

Assignment 4 Section 2: Flower Classification using Transfer Learning

In many real-world situations you will not want to train a whole convolutional network from scratch, unless you happen to have a very large dataset and a lot of compute resources. Instead, in most cases you will be using Transfer Learning.



In this assignment, you'll be modeling a small dataset of fewer than 5000 natural images. As the diagram above suggests, in a case like this you can use the pre-trained network as a feature extractor and substitute the head (MLP section). Since 5000 is small but not very small, you can also try to squeeze some more performance by fine-tuning the rest of the network. You will need to use a learning rate small enough, so you do not destroy the filters that the network has learned on pre-trained ImageNet.

In this assignment you will load a pre-trained model from Torch, remove the head, and substitute it with your own head. Then you will freeze the feature extractor part and train the head. Be careful in using a small learning rate! Experimentation usually is needed to find the right learning rate. Here we will use 1/100th of the initial learning rate.

Create the pre-trained model and substitute the head

Let's now load a pretrained model and substitute its head. Let's use a model `xyz` from `torchvision` and substitute its head with a fully-connected layer (`Linear`) with the right input and output dimension.

Freeze the backbone and thaw the head

Now we need to freeze all layers except the one we just added. Let's keep track of the parameters we are freezing so we will be able to free them later.

Define Loss Function

Define Optimizer (SDG)

Training Loop

Question: Reflecting on the Performance Difference

In the code you created, we have seen that the performance we obtain with transfer learning is significantly better than the performance we obtain by training from scratch. Why?