Assignments 2 & 3, Applied ML

John MacLaren Walsh

Fall, AY 2020-2021

1 Overview

Two scores will be created from your submission to this assignment, one regarding unsupervised learning, one regarding transfer learning for image classification.

2 Transfer Learning Image Classification with Keras

Using Keras in TensorFlow, create and train a deep neural network image classifier for the classes of flowers found in the Oxford Flowers 102 dataset. You can access the dataset through the TensorFlow datasets API[1]. You will submit a saved TensorFlow2 model in .h5 format and all of the code used to train it. You may employ transfer learning by loading one of the models from keras.applications. **Aside from loading one of those imagenet pretrained models, however, your weights may only be trained on the training and/or validation portions of the Oxford Flowers 102 dataset.** We reserve the right to retrain your model by rerunning your code, and will do so if we have reason to believe the model you have submitted accessed the testing data and is not the one your code creates. Your code must include a complete definition of a function preprocess which can be passed to testingSet.map(preprocess), where testingSet is a testing tensorflow dataset formatted identically to the training and validation portions of the Oxford Flowers 102 dataset. Your top-k accuracy, with $k \in \{1, 5, 10\}$, will be combined to generate an overall score that will determine your grade on this portion of the assignment.

2.1 Your Submission

You will submit a file assignment2and3.zip or assignment2and3.tgz containing the following files

1. buildAndTrainFlowersModel.py: This must contain all of the code necessary to train your model then save it.
2. preprocessDefinition.py: This must contain all of the code necessary to define a method preprocess which will operate on each (image,label) pair in the evaluation dataset via evalSet.map(preprocess). It should leave the label unchanged, and process the image (of arbitrary size!) into the format expected by your model. Typically this will call either tf.image.resize or tf.image.resize_with_pad, followed by the keras.applications.XXX.preprocess_input method for the imagenet model you are transfer learning from.
3. flowersModel.h5: This is your final saved trained TensorFlow2 Keras model that we will evaluate on the testing data.

The code used to evaluate your model will be something like

```
import tensorflow.compat.v2 as tf
import tensorflow_datasets as tfds
from tensorflow import keras
from preprocessDefinition import preprocess
model=tf.keras.models.load_model('flowersModel.h5')
evalset,info = tfds.load(name='oxford_flowers102', split='test',as_supervised=True,with_info=True)
evalPipe=evalset.map(preprocess,num_parallel_calls=16).batch(128).prefetch(1)
for feats,lab in evalPipe.unbatch().batch(6000).take(1):
    probPreds=trainedModel1.predict(feats)
top1err=tf.reduce_mean(keras.metrics.sparse_top_k_categorical_accuracy(lab,probPreds,k=1))
top5err=tf.reduce_mean(keras.metrics.sparse_top_k_categorical_accuracy(lab,probPreds,k=5))
top10err=tf.reduce_mean(keras.metrics.sparse_top_k_categorical_accuracy(lab,probPreds,k=10))
```

[1] Download the tensorflow-datasets package with pip if you do not already have it.

3 Unsupervised Learning Analysis of Transfer Learning Based Features

One can view the activations from the lower layers of a network pre-trained on imagenet as feature vectors that you are giving to the upper layers of your combined model. Write python code to do the following

1. Load the Keras model from keras.applications, pretrained on imageNet, that you used for transfer learning in the previous section, without the top layer.
2. Stack on top of this pretrained network a global average pooling 2D layer
3. Input into this stack (pretrained model followed by global average pooling 2D) all of the images, appropriately preprocessed with your preprocess function from the previous section, from the testing portion of the oxford flowers 102 dataset.
4. Perform PCA on the resulting series of feature vectors.
5. Generate a plot of the explained variance ratio versus number of dimensions kept under PCA

3.1 Your Submission

In addition to the items described in §2.1, your submission assignment2and3.zip or assignment2and3.tgz must also include

1. explainedVariancePlot.png: A plot of the explained variance ratio as a function of the dimension kept under PCA as described above.
2. generatePCAplot.py: The python code used to generate this plot.