OneLayerTensorFlow

October 12, 2018

TensorFlow Neural Network Lab

In this lab, you'll use all the tools you learned from *Introduction to TensorFlow* to label images of English letters! The data you are using, notMNIST, consists of images of a letter from A to J in different fonts.

The above images are a few examples of the data you'll be training on. After training the network, you will compare your prediction model against test data. Your goal, by the end of this lab, is to make predictions against that test set with at least an 80% accuracy. Let's jump in!

To start this lab, you first need to import all the necessary modules. Run the code below. If it runs successfully, it will print "`All modules imported`".

```
In [1]: import hashlib
        import os
        import pickle
        from urllib.request import urlretrieve

        import numpy as np
        from PIL import Image
        from sklearn.model_selection import train_test_split
        from sklearn.preprocessing import LabelBinarizer
        from sklearn.utils import resample
        from tqdm import tqdm
        from zipfile import ZipFile

        print('All modules imported.')
```

```
All modules imported.
```

The notMNIST dataset is too large for many computers to handle. It contains 500,000 images for just training. You'll be using a subset of this data, 15,000 images for each label (A-J).

```
In [2]: def download(url, file):
            """
            Download file from <url>
            :param url: URL to file
            :param file: Local file path
            """
            if not os.path.isfile(file):
```

```
                print('Downloading ' + file + '...')
                urlretrieve(url, file)
                print('Download Finished')

        # Download the training and test dataset.
        download('https://s3.amazonaws.com/udacity-sdc/notMNIST_train.zip', 'notMNIST_train.zip'
        download('https://s3.amazonaws.com/udacity-sdc/notMNIST_test.zip', 'notMNIST_test.zip')

        # Make sure the files aren't corrupted
        assert hashlib.md5(open('notMNIST_train.zip', 'rb').read()).hexdigest() == 'c8673b3f28f4
                'notMNIST_train.zip file is corrupted.  Remove the file and try again.'
        assert hashlib.md5(open('notMNIST_test.zip', 'rb').read()).hexdigest() == '5d3c7e653e634
                'notMNIST_test.zip file is corrupted.  Remove the file and try again.'

        # Wait until you see that all files have been downloaded.
        print('All files downloaded.')
```

```
All files downloaded.
```

```
In [3]: def uncompress_features_labels(file):
            """
            Uncompress features and labels from a zip file
            :param file: The zip file to extract the data from
            """
            features = []
            labels = []

            with ZipFile(file) as zipf:
                # Progress Bar
                filenames_pbar = tqdm(zipf.namelist(), unit='files')

                # Get features and labels from all files
                for filename in filenames_pbar:
                    # Check if the file is a directory
                    if not filename.endswith('/'):
                        with zipf.open(filename) as image_file:
                            image = Image.open(image_file)
                            image.load()
                            # Load image data as 1 dimensional array
                            # We're using float32 to save on memory space
                            feature = np.array(image, dtype=np.float32).flatten()

                        # Get the the letter from the filename.  This is the letter of the image
                        label = os.path.split(filename)[1][0]

                        features.append(feature)
                        labels.append(label)
```