Practical 3 - Keras

November 27, 2018

In [1]:
```
###### ALTERNATIVE CODE FOR PERSONAL EXPERIMENTATION USING KERAS #########

import tensorflow as tf
import numpy as np
import matplotlib.pyplot as plt

from tensorflow import keras
from tensorflow.examples.tutorials.mnist import input_data

sess = tf.Session(config=tf.ConfigProto(log_device_placement=True))

mnist = input_data.read_data_sets('MNIST_data', one_hot=True)

print("\n\n\n\n\n")

Ntrain = mnist.train.images.shape[0]
Ntest  = mnist.test.images.shape[0]

reshaped_train_images = np.zeros(shape=(Ntrain, 28, 28, 1))
reshaped_test_images  = np.zeros(shape=(Ntest, 28, 28, 1))
for i in range(Ntrain):
        reshaped_train_images[i] = mnist.train.images[i].reshape(28, 28, 1)
for i in range(Ntest):
  reshaped_test_images[i] = mnist.test.images[i].reshape(28, 28, 1)

# Show train dataset
plt.figure(figsize=(10,10))
for i in range(16):
    plt.subplot(4,4,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(reshaped_train_images[i].reshape(28, 28), cmap='Greys_r')
plt.show()

# Build model
model = keras.Sequential([
    keras.layers.Conv2D(filters=25, kernel_size=(12, 12), strides=(2, 2), padding='vali
```

```
    keras.layers.Conv2D(filters=64, kernel_size=(5, 5), strides=(1, 1), padding='same'
    keras.layers.MaxPooling2D(pool_size=(2, 2)),
    keras.layers.Flatten(),
    keras.layers.Dense(1024, activation='relu'),
        keras.layers.Dense(10),
        keras.layers.Activation('softmax')
])

model.compile(optimizer=keras.optimizers.Adam(lr=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train model
model.fit(reshaped_train_images, mnist.train.labels, batch_size=50, epochs=2)

# Evaluate and print accuracy
loss, acc = model.evaluate(reshaped_test_images, mnist.test.labels)
print('Accuracy:', acc)


# Find and show set of first 25 trained weights
weights_mat = model.get_weights()[0]
weights = np.split(weights_mat, 25, 3)

plt.figure(figsize=(10,10))
for i in range(25):
    plt.subplot(5,5,i+1)
    plt.xticks([])
    plt.yticks([])
    plt.grid(False)
    plt.imshow(weights[i].reshape(12, 12), cmap='Greys_r')
plt.show()

# Find and show 12 best activations for first 5 filters
intermediate_layer_model = keras.Model(inputs=model.inputs, outputs=model.layers[0].ou
intermediate_outputs = intermediate_layer_model.predict(reshaped_test_images)
transp_outputs = np.transpose(intermediate_outputs, [0, 3, 1, 2]) #NCHW

best_act_values = [0.0 for _ in range(12)]
best_patches = [np.zeros(shape=(12, 12)) for _ in range(12)]

for i in range(len(transp_outputs)): # for all images in test dataset
  activations = transp_outputs[i]

  for j in range(5): # first 5 filters
    activation = activations[j]

    for (x,y), val in np.ndenumerate(activation):
```

```
            for k in range(len(best_act_values)):
              if val > best_act_values[k]:
                best_act_values[k] = val
                best_patches[k] = reshaped_test_images[i, (x*2):(x*2+12), (y*2):(y*2+12)]
                break

    plt.figure(figsize=(10,10))
    for i in range(12):
        plt.subplot(4,3,i+1)
        plt.xticks([])
        plt.yticks([])
        plt.grid(False)
        plt.imshow(best_patches[i].reshape(12, 12), cmap='Greys_r')
    plt.show()
```

```
/Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/h5py/__init__.py:36: FutureWarning:
  from ._conv import register_converters as _register_converters
```

```
WARNING:tensorflow:From <ipython-input-1-98bcf2c49dab>:10: read_data_sets (from tensorflow.cont
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
WARNING:tensorflow:From /Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/tensorflow/c
Instructions for updating:
Please write your own downloading logic.
WARNING:tensorflow:From /Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/tensorflow/c
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data/train-images-idx3-ubyte.gz
WARNING:tensorflow:From /Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/tensorflow/c
Instructions for updating:
Please use tf.data to implement this functionality.
Extracting MNIST_data/train-labels-idx1-ubyte.gz
WARNING:tensorflow:From /Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/tensorflow/c
Instructions for updating:
Please use tf.one_hot on tensors.
Extracting MNIST_data/t10k-images-idx3-ubyte.gz
Extracting MNIST_data/t10k-labels-idx1-ubyte.gz
WARNING:tensorflow:From /Users/lorcandelaney/anaconda3/lib/python3.6/site-packages/tensorflow/c
Instructions for updating:
Please use alternatives such as official/mnist/dataset.py from tensorflow/models.
```

```
<matplotlib.figure.Figure at 0x1220f9cf8>


Epoch 1/2
55000/55000 [==============================] - 60s 1ms/step - loss: 0.3717 - acc: 0.8976
Epoch 2/2
55000/55000 [==============================] - 51s 920us/step - loss: 0.1102 - acc: 0.9669
10000/10000 [==============================] - 3s 289us/step
Accuracy: 0.9773


<matplotlib.figure.Figure at 0x13911b4a8>


<matplotlib.figure.Figure at 0x12c536128>
```