

Sep 24, 18 18:15

hw3_ec_gh.py

Page 1/2

```
'''
ECE471 Selected Topics in Machine Learning – Assignment 2
Submit by Sept. 26, 10pm
tldr: Classify mnist digits with a convolutional neural network. Get at least
95.5% accuracy on the test test.
'''

import numpy as np
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras.datasets.mnist import load_data

# Fix seed
np.random.seed(1618)
tf.set_random_seed(1618)

# Data parameters
NUM_CLASSES = 10
HEIGHT = 28
WIDTH = 28
NUM_CHANNELS = 1

# Load data and split into train, val, test sets
(x_train, y_train), (x_test, y_test) = load_data()
(x_train, y_train), (x_val, y_val) = \
    (x_train[:50000], y_train[:50000]), (x_train[50000:], y_train[50000:])

# Normalize and reshape data and labels
x_train, x_val, x_test = \
    map(lambda x: (x / 255.0).reshape([-1, HEIGHT, WIDTH, NUM_CHANNELS]),
        [x_train, x_val, x_test])
y_train, y_val, y_test = \
    map(lambda y: keras.utils.to_categorical(y, NUM_CLASSES),
        [y_train, y_val, y_test])

# Downsample images
x_train, x_val, x_test = \
    map(lambda x: x[:, ::2, ::2, :], [x_train, x_val, x_test])

# Hyperparameters
BATCH_SIZE = 32
NUM_EPOCHS = 10

# Create CNN using Keras API
activation = keras.activations.relu

model = keras.Sequential()
model.add(keras.layers.Conv2D(3, 3, activation=activation))
model.add(keras.layers.MaxPool2D(5))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(NUM_CLASSES, activation='softmax',
                             use_bias=False))

model.compile(loss=keras.losses.categorical_crossentropy,
              optimizer=keras.optimizers.Adam(),
              metrics=['accuracy'])

model.fit(x_train, y_train,
          batch_size=BATCH_SIZE,
          epochs=NUM_EPOCHS,
          verbose=1,
          validation_data=(x_val, y_val))

loss, acc = model.evaluate(x_test, y_test, verbose=1)
num_params = np.sum([np.prod(v.get_shape().as_list())
                     for v in tf.trainable_variables()])

print('Test loss:', loss)
print('Test accuracy:', acc)
```

Monday September 24, 2018

hw3_ec_gh.py

Sep 24, 18 18:15

hw3_ec_gh.py

Page 2/2

```
print('Number of parameters:', num_params)

''' Output
/Users/george/miniconda3/lib/python3.6/importlib/_bootstrap.py:219: RuntimeWarning: compiletime version 3.5 of m
odule 'tensorflow.python.framework.fast_tensor_util' does not match runtime version 3.6
  return f(*args, **kwargs)
Train on 50000 samples, validate on 10000 samples
Epoch 1/15
2018-09-24 01:21:28.264911: I tensorflow/core/platform/cpu_feature_guard.cc:141] Your CPU supports instructions t
hat this TensorFlow binary was not compiled to use: AVX2 FMA
50000/50000 [=====] - 4s 77us/step - loss: 1.7132 - acc: 0.4231 - val_loss: 1.155
4 - val_acc: 0.6233
Epoch 2/15
50000/50000 [=====] - 4s 74us/step - loss: 1.0180 - acc: 0.6720 - val_loss: 0.870
1 - val_acc: 0.7251
Epoch 3/15
50000/50000 [=====] - 4s 74us/step - loss: 0.8522 - acc: 0.7259 - val_loss: 0.771
4 - val_acc: 0.7526
Epoch 4/15
50000/50000 [=====] - 4s 74us/step - loss: 0.7821 - acc: 0.7462 - val_loss: 0.722
3 - val_acc: 0.7669
Epoch 5/15
50000/50000 [=====] - 4s 72us/step - loss: 0.7440 - acc: 0.7565 - val_loss: 0.692
6 - val_acc: 0.7734
Epoch 6/15
50000/50000 [=====] - 4s 71us/step - loss: 0.7191 - acc: 0.7653 - val_loss: 0.673
8 - val_acc: 0.7817
Epoch 7/15
50000/50000 [=====] - 4s 75us/step - loss: 0.7024 - acc: 0.7702 - val_loss: 0.665
5 - val_acc: 0.7840
Epoch 8/15
50000/50000 [=====] - 4s 72us/step - loss: 0.6905 - acc: 0.7760 - val_loss: 0.654
6 - val_acc: 0.7887
Epoch 9/15
50000/50000 [=====] - 4s 73us/step - loss: 0.6814 - acc: 0.7780 - val_loss: 0.642
2 - val_acc: 0.7941
Epoch 10/15
50000/50000 [=====] - 4s 71us/step - loss: 0.6739 - acc: 0.7794 - val_loss: 0.640
0 - val_acc: 0.7934
Epoch 11/15
50000/50000 [=====] - 4s 71us/step - loss: 0.6682 - acc: 0.7824 - val_loss: 0.632
5 - val_acc: 0.7958
Epoch 12/15
50000/50000 [=====] - 4s 71us/step - loss: 0.6633 - acc: 0.7826 - val_loss: 0.633
3 - val_acc: 0.7934
Epoch 13/15
50000/50000 [=====] - 4s 72us/step - loss: 0.6598 - acc: 0.7841 - val_loss: 0.630
8 - val_acc: 0.7967
Epoch 14/15
50000/50000 [=====] - 4s 71us/step - loss: 0.6565 - acc: 0.7865 - val_loss: 0.622
8 - val_acc: 0.7975
Epoch 15/15
50000/50000 [=====] - 4s 71us/step - loss: 0.6540 - acc: 0.7872 - val_loss: 0.620
0 - val_acc: 0.7983
10000/10000 [=====] - 0s 27us/step
Test loss: 0.6101281370639801
Test accuracy: 0.8052
Number of parameters: 458.0
'''
```

1/1