firstcnn_plaidml

December 8, 2020

[1]:
```
# Importing PlaidML
import plaidml.keras
plaidml.keras.install_backend()
import os
os.environ["KERAS_BACKEND"] = "plaidml.keras.backend"
```

[2]:
```
import keras
from keras.models import Sequential
from keras.layers import Dense, Dropout, Flatten
from keras.layers import Conv2D, MaxPooling2D
from keras import backend as K

# Download fashion dataset from Keras
fashion_mnist = keras.datasets.fashion_mnist
(x_train, y_train), (x_test, y_test) = keras.datasets.fashion_mnist.load_data()

# Reshape and normalize the data
x_train = x_train.astype('float32').reshape(60000,28,28,1) / 255
x_test = x_test.astype('float32').reshape(10000,28,28,1) / 255
```

```
Downloading data from http://fashion-mnist.s3-website.eu-
central-1.amazonaws.com/train-labels-idx1-ubyte.gz
32768/29515 [==================================] - 0s 4us/step
Downloading data from http://fashion-mnist.s3-website.eu-
central-1.amazonaws.com/train-images-idx3-ubyte.gz
26427392/26421880 [==============================] - 3s 0us/step
Downloading data from http://fashion-mnist.s3-website.eu-
central-1.amazonaws.com/t10k-labels-idx1-ubyte.gz
8192/5148 [================================================] - 0s 0us/step
Downloading data from http://fashion-mnist.s3-website.eu-
central-1.amazonaws.com/t10k-images-idx3-ubyte.gz
4423680/4422102 [==============================] - 1s 0us/step
```

[3]:
```
# Build a CNN model. You should see "INFO:plaidml:Opening device xxx" after you␣
↪run this chunk
model = keras.Sequential()
model.add(keras.layers.Conv2D(filters=64, kernel_size=2, padding='same',␣
↪activation='relu', input_shape=(28,28,1)))
```

```
model.add(keras.layers.MaxPooling2D(pool_size=2))
model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.Conv2D(filters=32, kernel_size=2, padding='same',␣
 ↪activation='relu'))
model.add(keras.layers.MaxPooling2D(pool_size=2))
model.add(keras.layers.Dropout(0.3))
model.add(keras.layers.Flatten())
model.add(keras.layers.Dense(256, activation='relu'))
model.add(keras.layers.Dropout(0.5))
model.add(keras.layers.Dense(10, activation='softmax'))
```

```
INFO:plaidml:Opening device "metal_amd_radeon_rx_5700_xt.0"
```

[4]:
```
# Compile the model
model.compile(optimizer='adam',
              loss=keras.losses.sparse_categorical_crossentropy,
              metrics=['accuracy'])
```

[5]:
```
# Fit the model on training set
model.fit(x_train, y_train,
          batch_size=64,
          epochs=10)

# Evaluate the model on test set
score = model.evaluate(x_test, y_test, verbose=0)
# Print test accuracy
print('\n', 'Test accuracy:', score[1])
```

```
Epoch 1/10
60000/60000 [==============================] - 10s 164us/step - loss: 0.5958 -
acc: 0.7798
Epoch 2/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.4168 -
acc: 0.8502
Epoch 3/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.3672 -
acc: 0.8677
Epoch 4/10
60000/60000 [==============================] - 7s 119us/step - loss: 0.3350 -
acc: 0.8783
Epoch 5/10
60000/60000 [==============================] - 7s 117us/step - loss: 0.3167 -
acc: 0.8853
Epoch 6/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.3002 -
acc: 0.8904
Epoch 7/10
```

```
60000/60000 [==============================] - 7s 116us/step - loss: 0.2890 -
acc: 0.8938
Epoch 8/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.2760 -
acc: 0.8984
Epoch 9/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.2669 -
acc: 0.9009
Epoch 10/10
60000/60000 [==============================] - 7s 116us/step - loss: 0.2600 -
acc: 0.9048

 Test accuracy: 0.915
```

```
[ ]:
```