

Deep Learning for Computer Vision, IEEE AI/ML Tutorial, August 2018

Santosh Chapaneri (santosh.chapaneri@ieee.org)

Note: This is NOT a stand-alone notebook; rather use it along with the course material shared during the tutorial.

While using this material, you might get some errors. Try to understand the error and troubleshoot it. Learn from mistakes and progress further.

There are several thought-provoking **questions** as well as **homework problems** below.

Make sure you solve them to gain detailed understanding.

Most of the answers were discussed during the Tutorial session on 10th Aug.

In []:

```
train = pd.read_csv("Data/fashion-mnist_train.csv")
Y_train = train["label"]
X_train = train.drop(labels = ["label"],axis = 1)
Y_train.value_counts()
img_rows, img_cols = 28, 28
X_train_mlp = X_train.values.reshape(X_train.shape[0],img_rows*img_cols)
X_train = X_train.values.reshape(X_train.shape[0],img_rows,img_cols,1)

# Question: Why did we have 1 in the last part?
# Question: Is X_train a scalar? vector? matrix? or ...s...? (fill in the dots)

X_train_mlp /= 255

# Question: Is this really needed? What is this operation called? N..m...z... (fill in the dots)
# Homework: similar for test data...practice!

num_classes = 10
Y_train_mlp = keras.utils.to_categorical(Y_train, num_classes)

# Homework: Learn about one-hot-encoding

X_train_mlp, X_val_mlp, Y_train_mlp, Y_val_mlp = train_test_split(X_train_mlp, Y_train_mlp, test_size=0.2)

# Question: Why are we splitting the train data?
```

Multi Layer Perceptron

In []:

```
batch_size = 256
num_epochs = 50
model = Sequential()
model.add(Dense(input_dim=784, activation='sigmoid',units=625, kernel_initializer='normal'))

# Homework: Learn about kernel_initializer

model.add(Dense(input_dim=625, activation='softmax',units=10, kernel_initializer='normal'))
model.compile(optimizer=SGD(lr=0.05), loss='categorical_crossentropy', metrics=['accuracy'])

# Homework: Learn about categorical_crossentropy
# Homework: What is the formula for accuracy? (We covered it during the tutorial, check your notes)

history = model.fit(X_train_mlp, Y_train_mlp,batch_size = batch_size,epochs = num_epochs,verbose = 1,validation_data = (X_val_mlp, Y_val_mlp))

# Homework: Recall the role of validation data

score = model.evaluate(X_test_mlp, Y_test_mlp, verbose = 1)

# Question: Why are we sending Y_test_mlp here during evaluation?
# Isn't it equivalent to telling the answers at exam time? What are your thoughts?
```

Deep Multi Layer Perceptron

In []:

```
model = Sequential()
model.add(Dense(input_dim=784, units=625, kernel_initializer='normal'))
model.add(Activation('relu'))
model.add(Dropout(0.2))

# Homework: Try with different dropout values and check the model performance

model.add(Dense(input_dim=625, units=625, kernel_initializer='normal'))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(input_dim=625, units=625, kernel_initializer='normal'))
model.add(Activation('relu'))
model.add(Dropout(0.2))

# Homework: Try adding one more dense layer...does it improve the accuracy further?

model.add(Dense(input_dim=625, units=10, kernel_initializer='normal'))
model.add(Activation('softmax'))
model.compile(optimizer=RMSprop(lr=0.001, rho=0.9), loss='categorical_crossentropy', metrics=['accuracy'])

# Homework: Learn about RMSprop optimizer

history = model.fit(X_train_mlp, Y_train_mlp, batch_size = batch_size, epochs = num_epochs, verbose = 1, validation_data = (X_val_mlp, Y_val_mlp))
score = model.evaluate(X_test_mlp, Y_test_mlp, verbose = 1)
```

Convolutional Neural Networks

In []:

```
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', kernel_initializer='normal',
, input_shape=input_shape))
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.25))

# Homework: Play with different dropout values

model.add(Conv2D(64, kernel_size=(3, 3), activation='relu'))

# Question: Should the number of filters be always power of 2?

model.add(MaxPooling2D(pool_size=(2, 2)))

# Question: What is the role of MaxPooling? Any other pooling options? Possible pool-si
zes?

model.add(Dropout(0.25))
model.add(Conv2D(128, kernel_size=(3, 3), activation='relu'))
model.add(Dropout(0.4))
model.add(Flatten())

# Question: How does Flatten actually work behind the scenes?
# Eg: If X = [1 4; 5 8], what is the outcome of flatten of X? [1 4 5 8] or [1 5 4 8]?

model.add(Dense(128, activation='relu'))
model.add(Dropout(0.3))
model.add(Dense(num_classes, activation='softmax'))

# Question: Can we any other activation function in last layer?

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

# Homework: Time to Learn about Adam optimizer

history = model.fit(X_train, Y_train,batch_size = batch_size,epochs = num_epochs,verbo
se = 1,validation_data = (X_val, Y_val))
score = model.evaluate(X_test, Y_test, verbose = 1)
plt.plot(epochs, accuracy, 'b', label='Training accuracy')

# Homework: Learn about matplotlib's plot functionality

plt.plot(epochs, val_accuracy, 'r', label='Validation accuracy')
plt.plot(epochs, loss, 'b', label='Training loss')
plt.plot(epochs, val_loss, 'r', label='Validation loss')
predicted_classes = model.predict_classes(X_test)

# Question: What is the shape of predicted_classes? What is your interpretation?

correct = np.nonzero(predicted_classes == Y_true)
incorrect = np.nonzero(predicted_classes != Y_true)

# Question: What is the shape of correct and incorrect? What do they represent?

print(classification_report(Y_true, predicted_classes, target_names = target_names))

# Homework: Learn about Precision, Recall and F-score
```

```
Y_pred = model.predict(X_test)

# Question: What is the shape of Y_pred? What is your interpretation?

confusion_mtx = confusion_matrix(Y_true, Y_pred_classes)

# Homework: Learn about the confusion matrix
```

Transfer Learning

In []:

```
train = pd.read_csv("Data/mnist_train.csv",header=None)

# Question: Think why we header=None here?

Y_train = train[0]
X_train = train.drop(labels = [0],axis = 1)
X_train = np.array(X_train)

# Question: What is np.array operation? How does it help here?

X_train_lt5 = X_train[Y_train < 5]

# Question: What's going on here? Interpret this one-line code.

Y_train_lt5 = Y_train[Y_train < 5]
X_train_gte5 = X_train[Y_train >= 5]
Y_train_gte5 = Y_train[Y_train >= 5] - 5

# similar for test data

batch_size = 128, num_classes = 5, epochs = 5
num_filters = 32, pool_size = 2, kernel_size = 3

def train_model(model, train, test, num_classes):
    # Homework: just repeat what we have above in CNN...time for practice!

    feature_layers = [Convolution2D(num_filters, (kernel_size, kernel_size),input_shape = i
nput_shape,padding = 'valid'),Activation('relu'),Convolution2D(num_filters, (kernel_siz
e, kernel_size)),Activation('relu'),MaxPooling2D((pool_size, pool_size)),Dropout(0.25),
Flatten()]
    classification_layers = [Dense(128),Activation('relu'),Dropout(0.5),Dense(num_classes),
Activation('softmax')]
    model = Sequential(feature_layers + classification_layers)

    # Question: What is this '+' doing here?

    train_model(model, (X_train_lt5, Y_train_lt5), (X_test_lt5, Y_test_lt5), num_classes)
    for layer in feature_layers:
        layer.trainable = False

    # Question: What is the magic happening here?

    train_model(model, (X_train_gte5, Y_train_gte5), (X_test_gte5, Y_test_gte5), num_classe
s)

    # Homework: Try increasing the number of epochs. Does the performance improve further?

    # Homework: Try changing the pool size, kernel size, number of filters. What is the imp
act on model performance?
```

Practice...practice...practice. Troubleshoot the errors.

If still it doesn't work, you can email me; however, I would like to see the efforts you took to solve the errors before I give the answer.

On the way to becoming a Data Scientist :)