Definitions:
Activation Function: An activation function determines the output of a node given an input. It determines if the output of the neuron will be important to future neurons. The output of an activation function is determined by multiplying a neuron's input by a weight. In most cases an output of 0 means the neuron has not fired and its output can be "ignored". This is because the weight of the neuron is 0 meaning that output has no weight in the final solution.

Filter: Filters are produced by the kernel to determine important features of the image. Produced using the method listed in the kernel definition. It is a mapping of subgroups of pixels. They can be used to determine more general important features when categorizing images.

Kernel: In convolutional neural networks, the kernel helps to determine the important features of an image. The kernel analyzes image pixels in subgroups by sliding across and down the image, most standardly by one column and row at a time. The sections of the kernel's grid contains numbers that are respective scalars for each pixel that the kernel is analyzing. The kernel is responsible for making filters by multiplying its grid section weights by each respective pixel and adding them together. Each time this operation is performed, one section of a filter is completed. The weights in the kernel determine which parts of the sub image may be important.

Fashion_mnist_keras: input cell 6-original model

```
# Must define the input shape in the first layer of the neural network

model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=2, padding='same', activation='relu',
input_shape=(28,28,1)))
```

This first line initializes the input to the model of size 28 x 28, and sends it to a convolution layer. Convolution layers are used in image processing to determine the "important" parts of an image, or parts of the image that the model should use more to determine the final output, relative to other parts. Convolution layers use filters which is basically a number of important features that you want the neural network to find in the image. I say important lightly, because depending on the kernel size, and the weights for every section of the kernel size, some parts of the image might be more important than others.

The kernel size determines how many pixels of the image are observed at one time when trying to build future maps. For example, the current input images 28 x 28. That means the kernel will look at four pixels at a time. The first set of pixels would be the top four in the left corner, the next four would be in the top two rows of the image, but the kernel would slide over one to the right.

The padding being the same means that the output future maps will be the same size as the original image, despite the fact that the kernel size is 2 x 2 and would normally make the feature map smaller than the current image size. This is achieved by adding pixels with a value of zero to the outside of the original image.

The rectified linear unit activation function is used throughout this model. This activation function is usually more useful than other activation functions because it's easy to compute the derivative for gradient descent while also avoiding the vanishing gradient problem. In this activation function the output of any negative inputs is always zero, and that the input is positive the output is the same as the input. The rectified linear unit avoids the vanishing gradient problem because if you have any asymptotic tendencies, such as with the sigmoid function, that would cause it's only output large positive or large negative numbers. Avoiding the vanishing gradient problem is very important in deep learning because if the gradient is too small you experience no error correction, and learn nothing. If the gradient is too large, the error is overcorrected which means the ideal solution will always be missed.

In this specific convolution layer the output is 64 features that were created by looking at 2 x 2 sets of pixels from the input image, with the zero padding. The output of the feature maps were

determined by the rectified linear unit function which means it's output feature map was the same as it's input after the kernel mapping.

In this instance, there are more features with the smaller kernel size which means the model probably started by looking at the more detailed differences in the image.

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

Max pooling is used to find the generalities in the future map. It replaces a set of, in this case 2 x 2 pixels, with the largest value pixel lin that section of the future map. This keeps the model from over fitting because it allows the model to look at generalities as opposed to smaller details of the specific pixel shading. It is important to keep you kernel size small, so the model doesn't overgeneralize.

model.add(tf.keras.layers.Dropout(0.3))

Dropout is used to remove connections between two layers in a neural network with a random probability. In this case a connection will dropout with a 30% probability between the max pooling layer and the next convolution layer. Dropout can be used to avoid over fitting because it causes the model to lose some information about the training set. It is usually better to use dropout with smaller percentages, so you don't lose too much information in one the risk of under fitting.

model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=2, padding='same', activation='relu'))

This is again a convolution layer. Everything about this layer is the same as the previous layer, except for the number of filters. The reduction in the number of filters allows the model to narrow in on the more important features in the model. The reduction in filter size could also be due to the use of max pooling. Max pooling caused the future map images to be less detailed meaning there are less features that can be observed from them to begin with.

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

This is another max pooling layer that is exactly the same as the first. Max pooling layers are usually used after convolution layers to have a pattern of detail to generality. You get detailed features from the convolution layers and then you can generalize them to avoid over fitting using

the max pooling layers. So at this point the model is looking at some pretty general features of different clothing items, while trying to avoid under fitting.

model.add(tf.keras.layers.Dropout(0.3))

This is another dropout layer. In this instance it is between the max pooling layer in the flattening layer. This is again used in small amounts to lose some information from the feature maps and max pooling to allow the model to lgeneralize more, and reduce the amount of data that goes into the flattening layer.

model.add(tf.keras.layers.Flatten())

The flattening layer is used to take the feature maps of the square images, and turn them into a vector to be put through the final output layerll to determine the different images.

model.add(tf.keras.layers.Dense(256, activation='relu'))

The dense layer has connections from every input node to every output node. Again it uses the rectified linear unit function for easier calculations, and avoiding the vanishing gradient. In this case the layer has 256 output nodes. The layer is particularly important because it takes the feature map vector and determines which parts of the vector (aka pixels) are most related to one and other. That is why the particular output for this layer is large because it is looking at all possible combinations of pixels.

model.add(tf.keras.layers.Dropout(0.5))

The dropout between the dense layer in the final output layer is slightly larger than the other dropout layers because the dense layer contains so much information it might be difficult for the output layer to make its final determination with so much information. With this large amount of information, the model could also have the pppossibility of over fitting. In this case, connections will drop with the roughly 50% probability.

model.add(tf.keras.layers.Dense(10, activation='softm'))

The final layer uses the softmax function because the softmax function outputs a binary value to say whether or not an image is or isn't a certain value. In this case the layer has 10 outputs because they attend possible clothing articles.

Final Model Notebook: Input cell 13- Model 1

```
# Must define the input shape in the first layer of the neural network

model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, padding='same', activation='relu',
input_shape=(28,28,1)))

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=2, padding='same', activation='relu'))

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

model.add(tf.keras.layers.Flatten())
```

```
model.add(tf.keras.layers.Dense(256, activation='relu'))

model.add(tf.keras.layers.Dropout(0.3))

model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

This model is similar to the original, except the number of filters in the convolution layers increase over time instead of decrease, and the dropout only occurs at the end. I increase the filter size over time to simulate more general observations, that then become more detailed. I thought this would be a better way to run the model because the model for the very beginning can meet the general distinction between a top and a shoe for example, due to how much shading is in the image relative to the backgrounll.

My first convolutional layer is also more general because I use a kernel size of three instead of two. That means larger areas were considered when creating the feature maps which makes them more general. Since the first convolutional layer has less features and is probably more adept at determining the outer bounds of objects. The second convolutional layer, since it has more features, and a smaller kernel size should be more adept at noticing gaps in pixilation, such as distinctions between the sleeve of a shirt, and the body of a shirt.

I don't use dropout until the very end of the model. This allows the model to retain as much information as possible before the image data is flattened. Storing as much image data as possible helps get more comparisons for higher accuracy. The other reason why I waited to use dropout on to the very end is because it allows the model to train faster. In this instance I picked a dropout percentage of 30%. I just tested it and assumed it was probably enough to keep the model from being overwhelmed with too much information and over fitting, but also small enough that the model doesn'ppt run the risk of possibly losing some important information in large quantities.

Final Model Notebook: Input cell 18- Model 2

```
# Must define the input shape in the first layer of the neural network

model.add(tf.keras.layers.Conv2D(filters=32, kernel_size=3, padding='same', activation='relu', input_shape=(28,28,1)))

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

model.add(tf.keras.layers.Conv2D(filters=64, kernel_size=2, padding='same', activation='relu'))

model.add(tf.keras.layers.MaxPooling2D(pool_size=2))

model.add(tf.keras.layers.Dropout(0.3))

model.add(tf.keras.layers.Flatten())

model.add(tf.keras.layers.Dense(256, activation='relu'))

model.add(tf.keras.layers.Dropout(0.3))

model.add(tf.keras.layers.Dense(10, activation='softmax'))
```

This model is very similar to model one except there is an extra 30% dropout between the second max pooling layer, and the flatten layer. I wanted to try this because I thought it would be good to see if removing some of the connections from the image feature data before it is flattened would help the network perform better. This is because flattening ultimately changes the shape of the data and could possibly affect the final output.

Results: input cells 17 and 20

My initial run of both models resulted in about 92% test set accuracy. Model one had a training set accuracy of 95%, while model two had a training set accuracy of 92%. I wanted to determine if having higher training set accuracy, like model one, allowed for higher test set accuracy in general. To test this I ran each model 10 times and averaged their test set accuracy. The results are shown below.

```
Model 1


Run: 1
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4639 - accur
acy: 0.8332
Epoch 00001: val_loss improved from inf to 0.32213, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 23s 411us/sample - loss: 0.463
7 - accuracy: 0.8333 - val_loss: 0.3221 - val_accuracy: 0.8880
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3055 - accur
acy: 0.8896
Epoch 00002: val_loss improved from 0.32213 to 0.26652, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 386us/sample - loss: 0.305
5 - accuracy: 0.8896 - val_loss: 0.2665 - val_accuracy: 0.9068
Epoch 3/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.2643 - accur
acy: 0.9027
Epoch 00003: val_loss improved from 0.26652 to 0.24552, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 382us/sample - loss: 0.264
3 - accuracy: 0.9027 - val_loss: 0.2455 - val_accuracy: 0.9132
Epoch 4/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.2341 - accur
acy: 0.9150
Epoch 00004: val_loss improved from 0.24552 to 0.22967, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 381us/sample - loss: 0.234
4 - accuracy: 0.9149 - val_loss: 0.2297 - val_accuracy: 0.9172
```

```
Epoch 5/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.2115 - accur
acy: 0.9214
Epoch 00005: val_loss improved from 0.22967 to 0.21778, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 386us/sample - loss: 0.211
6 - accuracy: 0.9214 - val_loss: 0.2178 - val_accuracy: 0.9196
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1894 - accur
acy: 0.9291
Epoch 00006: val_loss improved from 0.21778 to 0.21773, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 419us/sample - loss: 0.189
4 - accuracy: 0.9291 - val_loss: 0.2177 - val_accuracy: 0.9178
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1693 - accur
acy: 0.9375
Epoch 00007: val_loss improved from 0.21773 to 0.20141, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 449us/sample - loss: 0.169
3 - accuracy: 0.9375 - val_loss: 0.2014 - val_accuracy: 0.9286
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1511 - accur
acy: 0.9429
Epoch 00008: val_loss did not improve from 0.20141
55000/55000 [==============================] - 23s 418us/sample - loss: 0.151
0 - accuracy: 0.9429 - val_loss: 0.2015 - val_accuracy: 0.9294
Epoch 9/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.1381 - accur
acy: 0.9481
Epoch 00009: val_loss did not improve from 0.20141
55000/55000 [==============================] - 23s 412us/sample - loss: 0.138
2 - accuracy: 0.9480 - val_loss: 0.2041 - val_accuracy: 0.9282
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1239 - accur
acy: 0.9538
Epoch 00010: val_loss did not improve from 0.20141
55000/55000 [==============================] - 21s 389us/sample - loss: 0.123
9 - accuracy: 0.9538 - val_loss: 0.2062 - val_accuracy: 0.9278

 Test accuracy: 0.9225


Run: 2
```

```
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4706 - accur
acy: 0.8308
Epoch 00001: val_loss improved from inf to 0.32614, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 22s 395us/sample - loss: 0.470
5 - accuracy: 0.8308 - val_loss: 0.3261 - val_accuracy: 0.8838
Epoch 2/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.3093 - accur
acy: 0.8891
Epoch 00002: val_loss improved from 0.32614 to 0.26476, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 424us/sample - loss: 0.309
2 - accuracy: 0.8891 - val_loss: 0.2648 - val_accuracy: 0.9052
Epoch 3/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2633 - accur
acy: 0.9030
Epoch 00003: val_loss improved from 0.26476 to 0.23560, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 419us/sample - loss: 0.263
4 - accuracy: 0.9030 - val_loss: 0.2356 - val_accuracy: 0.9136
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2339 - accur
acy: 0.9141
Epoch 00004: val_loss improved from 0.23560 to 0.22715, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 417us/sample - loss: 0.233
8 - accuracy: 0.9141 - val_loss: 0.2271 - val_accuracy: 0.9158
Epoch 5/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2129 - accur
acy: 0.9208
Epoch 00005: val_loss improved from 0.22715 to 0.22024, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 418us/sample - loss: 0.212
9 - accuracy: 0.9208 - val_loss: 0.2202 - val_accuracy: 0.9234
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1935 - accur
acy: 0.9290
Epoch 00006: val_loss improved from 0.22024 to 0.21775, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 422us/sample - loss: 0.193
5 - accuracy: 0.9290 - val_loss: 0.2177 - val_accuracy: 0.9212
Epoch 7/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.1758 - accur
acy: 0.9356
Epoch 00007: val_loss improved from 0.21775 to 0.20451, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 422us/sample - loss: 0.175
9 - accuracy: 0.9356 - val_loss: 0.2045 - val_accuracy: 0.9242
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1595 - accur
acy: 0.9399
Epoch 00008: val_loss did not improve from 0.20451
55000/55000 [==============================] - 23s 425us/sample - loss: 0.159
4 - accuracy: 0.9399 - val_loss: 0.2208 - val_accuracy: 0.9218
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1428 - accur
acy: 0.9464
Epoch 00009: val_loss did not improve from 0.20451
55000/55000 [==============================] - 23s 424us/sample - loss: 0.142
8 - accuracy: 0.9464 - val_loss: 0.2146 - val_accuracy: 0.9210
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1259 - accur
acy: 0.9524
Epoch 00010: val_loss did not improve from 0.20451
55000/55000 [==============================] - 22s 391us/sample - loss: 0.125
9 - accuracy: 0.9524 - val_loss: 0.2111 - val_accuracy: 0.9270

 Test accuracy: 0.9181


Run: 3
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.4636 - accur
acy: 0.8332
Epoch 00001: val_loss improved from inf to 0.33747, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 23s 412us/sample - loss: 0.463
3 - accuracy: 0.8333 - val_loss: 0.3375 - val_accuracy: 0.8788
Epoch 2/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.3038 - accur
acy: 0.8894
Epoch 00002: val_loss improved from 0.33747 to 0.28953, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 399us/sample - loss: 0.303
6 - accuracy: 0.8895 - val_loss: 0.2895 - val_accuracy: 0.8952
```

```
Epoch 3/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2611 - accur
acy: 0.9047
Epoch 00003: val_loss improved from 0.28953 to 0.24724, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 404us/sample - loss: 0.260
9 - accuracy: 0.9048 - val_loss: 0.2472 - val_accuracy: 0.9070
Epoch 4/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2341 - accur
acy: 0.9139 ETA: 0s - loss: 0.2339 -
Epoch 00004: val_loss improved from 0.24724 to 0.22401, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 407us/sample - loss: 0.234
2 - accuracy: 0.9138 - val_loss: 0.2240 - val_accuracy: 0.9188
Epoch 5/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2084 - accur
acy: 0.9227
Epoch 00005: val_loss improved from 0.22401 to 0.21640, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 408us/sample - loss: 0.208
2 - accuracy: 0.9227 - val_loss: 0.2164 - val_accuracy: 0.9180
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1917 - accur
acy: 0.9272
Epoch 00006: val_loss improved from 0.21640 to 0.20735, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 399us/sample - loss: 0.191
7 - accuracy: 0.9272 - val_loss: 0.2074 - val_accuracy: 0.9244
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1720 - accur
acy: 0.9357
Epoch 00007: val_loss improved from 0.20735 to 0.20573, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 398us/sample - loss: 0.172
0 - accuracy: 0.9357 - val_loss: 0.2057 - val_accuracy: 0.9270
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1549 - accur
acy: 0.9411
Epoch 00008: val_loss improved from 0.20573 to 0.20469, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 402us/sample - loss: 0.155
0 - accuracy: 0.9411 - val_loss: 0.2047 - val_accuracy: 0.9280
Epoch 9/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.1398 - accur
acy: 0.9477
Epoch 00009: val_loss improved from 0.20469 to 0.20306, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 22s 405us/sample - loss: 0.139
7 - accuracy: 0.9477 - val_loss: 0.2031 - val_accuracy: 0.9268
Epoch 10/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.1251 - accur
acy: 0.9525
Epoch 00010: val_loss did not improve from 0.20306
55000/55000 [==============================] - 22s 406us/sample - loss: 0.125
0 - accuracy: 0.9525 - val_loss: 0.2067 - val_accuracy: 0.9298

 Test accuracy: 0.9217


Run: 4
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4690 - accur
acy: 0.8299
Epoch 00001: val_loss improved from inf to 0.32748, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 23s 419us/sample - loss: 0.468
6 - accuracy: 0.8301 - val_loss: 0.3275 - val_accuracy: 0.8792
Epoch 2/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.3088 - accur
acy: 0.8871
Epoch 00002: val_loss improved from 0.32748 to 0.27647, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 413us/sample - loss: 0.308
6 - accuracy: 0.8871 - val_loss: 0.2765 - val_accuracy: 0.8966
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2664 - accur
acy: 0.9013
Epoch 00003: val_loss improved from 0.27647 to 0.24608, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 416us/sample - loss: 0.266
4 - accuracy: 0.9014 - val_loss: 0.2461 - val_accuracy: 0.9090
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2378 - accur
acy: 0.9128
Epoch 00004: val_loss improved from 0.24608 to 0.23806, saving model to model
.weights.best.hdf5
```

```
55000/55000 [==============================] - 23s 417us/sample - loss: 0.238
0 - accuracy: 0.9127 - val_loss: 0.2381 - val_accuracy: 0.9092
Epoch 5/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2128 - accur
acy: 0.9211
Epoch 00005: val_loss improved from 0.23806 to 0.22386, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 420us/sample - loss: 0.212
9 - accuracy: 0.9210 - val_loss: 0.2239 - val_accuracy: 0.9146
Epoch 6/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1944 - accur
acy: 0.9282
Epoch 00006: val_loss did not improve from 0.22386
55000/55000 [==============================] - 23s 422us/sample - loss: 0.194
4 - accuracy: 0.9282 - val_loss: 0.2346 - val_accuracy: 0.9132
Epoch 7/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1735 - accur
acy: 0.9342
Epoch 00007: val_loss improved from 0.22386 to 0.20998, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 424us/sample - loss: 0.173
6 - accuracy: 0.9341 - val_loss: 0.2100 - val_accuracy: 0.9224
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1581 - accur
acy: 0.9408
Epoch 00008: val_loss improved from 0.20998 to 0.20971, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 431us/sample - loss: 0.158
1 - accuracy: 0.9408 - val_loss: 0.2097 - val_accuracy: 0.9258
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1426 - accur
acy: 0.9464
Epoch 00009: val_loss did not improve from 0.20971
55000/55000 [==============================] - 24s 431us/sample - loss: 0.142
6 - accuracy: 0.9465 - val_loss: 0.2159 - val_accuracy: 0.9264
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1297 - accur
acy: 0.9511
Epoch 00010: val_loss did not improve from 0.20971
55000/55000 [==============================] - 24s 433us/sample - loss: 0.129
7 - accuracy: 0.9511 - val_loss: 0.2230 - val_accuracy: 0.9220

 Test accuracy: 0.922
```

```
Run: 5
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4628 - accur
acy: 0.8350
Epoch 00001: val_loss improved from inf to 0.30591, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 24s 442us/sample - loss: 0.462
7 - accuracy: 0.8350 - val_loss: 0.3059 - val_accuracy: 0.8934
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3026 - accur
acy: 0.8911
Epoch 00002: val_loss improved from 0.30591 to 0.25453, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 438us/sample - loss: 0.302
6 - accuracy: 0.8911 - val_loss: 0.2545 - val_accuracy: 0.9042
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2591 - accur
acy: 0.9044
Epoch 00003: val_loss improved from 0.25453 to 0.24896, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.259
1 - accuracy: 0.9044 - val_loss: 0.2490 - val_accuracy: 0.9092
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2331 - accur
acy: 0.9127
Epoch 00004: val_loss improved from 0.24896 to 0.22811, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 444us/sample - loss: 0.233
1 - accuracy: 0.9127 - val_loss: 0.2281 - val_accuracy: 0.9168
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2105 - accur
acy: 0.9211
Epoch 00005: val_loss did not improve from 0.22811
55000/55000 [==============================] - 25s 449us/sample - loss: 0.210
4 - accuracy: 0.9211 - val_loss: 0.2303 - val_accuracy: 0.9170
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1877 - accur
acy: 0.9298
Epoch 00006: val_loss improved from 0.22811 to 0.20628, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 450us/sample - loss: 0.187
8 - accuracy: 0.9297 - val_loss: 0.2063 - val_accuracy: 0.9220
```

```
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1710 - accur
acy: 0.9368
Epoch 00007: val_loss did not improve from 0.20628
55000/55000 [==============================] - 25s 457us/sample - loss: 0.171
0 - accuracy: 0.9368 - val_loss: 0.2210 - val_accuracy: 0.9174
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1541 - accur
acy: 0.9434
Epoch 00008: val_loss improved from 0.20628 to 0.20527, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 460us/sample - loss: 0.154
2 - accuracy: 0.9434 - val_loss: 0.2053 - val_accuracy: 0.9230
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1389 - accur
acy: 0.9476
Epoch 00009: val_loss did not improve from 0.20527
55000/55000 [==============================] - 26s 470us/sample - loss: 0.138
9 - accuracy: 0.9476 - val_loss: 0.2279 - val_accuracy: 0.9208
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1213 - accur
acy: 0.9548
Epoch 00010: val_loss did not improve from 0.20527
55000/55000 [==============================] - 26s 476us/sample - loss: 0.121
3 - accuracy: 0.9548 - val_loss: 0.2188 - val_accuracy: 0.9258

 Test accuracy: 0.9201


Run: 6
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4685 - accur
acy: 0.8312
Epoch 00001: val_loss improved from inf to 0.31240, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 27s 485us/sample - loss: 0.468
2 - accuracy: 0.8312 - val_loss: 0.3124 - val_accuracy: 0.8886
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3041 - accur
acy: 0.8893 ETA: 0s - loss: 0.3042 - accuracy: 0.
Epoch 00002: val_loss improved from 0.31240 to 0.26051, saving model to model
.weights.best.hdf5
```

```
55000/55000 [==============================] - 27s 487us/sample - loss: 0.304
2 - accuracy: 0.8892 - val_loss: 0.2605 - val_accuracy: 0.9058
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2598 - accur
acy: 0.9055
Epoch 00003: val_loss improved from 0.26051 to 0.24327, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 27s 498us/sample - loss: 0.259
8 - accuracy: 0.9055 - val_loss: 0.2433 - val_accuracy: 0.9132
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2299 - accur
acy: 0.9141
Epoch 00004: val_loss improved from 0.24327 to 0.22487, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 504us/sample - loss: 0.229
9 - accuracy: 0.9141 - val_loss: 0.2249 - val_accuracy: 0.9166
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2046 - accur
acy: 0.9251
Epoch 00005: val_loss improved from 0.22487 to 0.21087, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 515us/sample - loss: 0.204
6 - accuracy: 0.9251 - val_loss: 0.2109 - val_accuracy: 0.9214
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1852 - accur
acy: 0.9314
Epoch 00006: val_loss did not improve from 0.21087
55000/55000 [==============================] - 29s 524us/sample - loss: 0.185
1 - accuracy: 0.9313 - val_loss: 0.2273 - val_accuracy: 0.9178
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1661 - accur
acy: 0.9389
Epoch 00007: val_loss did not improve from 0.21087
55000/55000 [==============================] - 29s 535us/sample - loss: 0.166
1 - accuracy: 0.9389 - val_loss: 0.2112 - val_accuracy: 0.9218
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1491 - accur
acy: 0.9460
Epoch 00008: val_loss did not improve from 0.21087
55000/55000 [==============================] - 27s 496us/sample - loss: 0.149
1 - accuracy: 0.9460 - val_loss: 0.2234 - val_accuracy: 0.9206
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1350 - accur
acy: 0.9496
```

```
Epoch 00009: val_loss did not improve from 0.21087
55000/55000 [==============================] - 24s 428us/sample - loss: 0.135
0 - accuracy: 0.9496 - val_loss: 0.2222 - val_accuracy: 0.9210
Epoch 10/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1182 - accur
acy: 0.9552
Epoch 00010: val_loss did not improve from 0.21087
55000/55000 [==============================] - 24s 439us/sample - loss: 0.118
2 - accuracy: 0.9553 - val_loss: 0.2301 - val_accuracy: 0.9252

 Test accuracy: 0.9159


Run: 7
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4724 - accur
acy: 0.8296
Epoch 00001: val_loss improved from inf to 0.30568, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 25s 455us/sample - loss: 0.472
3 - accuracy: 0.8297 - val_loss: 0.3057 - val_accuracy: 0.8938
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3134 - accur
acy: 0.8854
Epoch 00002: val_loss improved from 0.30568 to 0.26497, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 460us/sample - loss: 0.313
4 - accuracy: 0.8854 - val_loss: 0.2650 - val_accuracy: 0.9052
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2708 - accur
acy: 0.9004
Epoch 00003: val_loss improved from 0.26497 to 0.23986, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 458us/sample - loss: 0.270
8 - accuracy: 0.9003 - val_loss: 0.2399 - val_accuracy: 0.9140
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2389 - accur
acy: 0.9119
Epoch 00004: val_loss improved from 0.23986 to 0.23834, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 464us/sample - loss: 0.239
0 - accuracy: 0.9119 - val_loss: 0.2383 - val_accuracy: 0.9110
Epoch 5/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.2187 - accur
acy: 0.9186
Epoch 00005: val_loss improved from 0.23834 to 0.22413, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 467us/sample - loss: 0.218
8 - accuracy: 0.9186 - val_loss: 0.2241 - val_accuracy: 0.9182
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1987 - accur
acy: 0.9265
Epoch 00006: val_loss improved from 0.22413 to 0.21685, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 470us/sample - loss: 0.198
7 - accuracy: 0.9266 - val_loss: 0.2168 - val_accuracy: 0.9226
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1796 - accur
acy: 0.9327
Epoch 00007: val_loss improved from 0.21685 to 0.21106, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 478us/sample - loss: 0.179
6 - accuracy: 0.9327 - val_loss: 0.2111 - val_accuracy: 0.9198
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1625 - accur
acy: 0.9389
Epoch 00008: val_loss did not improve from 0.21106
55000/55000 [==============================] - 27s 484us/sample - loss: 0.162
5 - accuracy: 0.9389 - val_loss: 0.2132 - val_accuracy: 0.9250
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1475 - accur
acy: 0.9448
Epoch 00009: val_loss did not improve from 0.21106
55000/55000 [==============================] - 27s 485us/sample - loss: 0.147
5 - accuracy: 0.9448 - val_loss: 0.2179 - val_accuracy: 0.9272
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1319 - accur
acy: 0.9510
Epoch 00010: val_loss did not improve from 0.21106
55000/55000 [==============================] - 27s 490us/sample - loss: 0.131
9 - accuracy: 0.9510 - val_loss: 0.2165 - val_accuracy: 0.9234

 Test accuracy: 0.9149


Run: 8
Train on 55000 samples, validate on 5000 samples
```

```
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4640 - accur
acy: 0.8317
Epoch 00001: val_loss improved from inf to 0.31145, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 28s 508us/sample - loss: 0.463
9 - accuracy: 0.8317 - val_loss: 0.3114 - val_accuracy: 0.8922
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3016 - accur
acy: 0.8907
Epoch 00002: val_loss improved from 0.31145 to 0.26983, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 508us/sample - loss: 0.301
5 - accuracy: 0.8907 - val_loss: 0.2698 - val_accuracy: 0.9006
Epoch 3/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.2567 - accur
acy: 0.9052
Epoch 00003: val_loss improved from 0.26983 to 0.24726, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 500us/sample - loss: 0.256
5 - accuracy: 0.9054 - val_loss: 0.2473 - val_accuracy: 0.9110
Epoch 4/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2284 - accur
acy: 0.9167
Epoch 00004: val_loss improved from 0.24726 to 0.22296, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 422us/sample - loss: 0.228
4 - accuracy: 0.9167 - val_loss: 0.2230 - val_accuracy: 0.9220
Epoch 5/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2039 - accur
acy: 0.9241
Epoch 00005: val_loss did not improve from 0.22296
55000/55000 [==============================] - 23s 420us/sample - loss: 0.203
9 - accuracy: 0.9241 - val_loss: 0.2283 - val_accuracy: 0.9164
Epoch 6/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.1829 - accur
acy: 0.9320
Epoch 00006: val_loss improved from 0.22296 to 0.20210, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 384us/sample - loss: 0.182
8 - accuracy: 0.9321 - val_loss: 0.2021 - val_accuracy: 0.9280
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1641 - accur
acy: 0.9383
```

```
Epoch 00007: val_loss did not improve from 0.20210
55000/55000 [==============================] - 21s 386us/sample - loss: 0.164
1 - accuracy: 0.9383 - val_loss: 0.2243 - val_accuracy: 0.9168
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1484 - accur
acy: 0.9449
Epoch 00008: val_loss did not improve from 0.20210
55000/55000 [==============================] - 21s 384us/sample - loss: 0.148
3 - accuracy: 0.9449 - val_loss: 0.2142 - val_accuracy: 0.9226
Epoch 9/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1335 - accur
acy: 0.9503
Epoch 00009: val_loss did not improve from 0.20210
55000/55000 [==============================] - 21s 385us/sample - loss: 0.133
7 - accuracy: 0.9502 - val_loss: 0.2098 - val_accuracy: 0.9248
Epoch 10/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1169 - accur
acy: 0.9571
Epoch 00010: val_loss did not improve from 0.20210
55000/55000 [==============================] - 21s 384us/sample - loss: 0.116
8 - accuracy: 0.9571 - val_loss: 0.2169 - val_accuracy: 0.9238

 Test accuracy: 0.9197


Run: 9
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54848/55000 [============================>.] - ETA: 0s - loss: 0.4587 - accur
acy: 0.8344
Epoch 00001: val_loss improved from inf to 0.31033, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 22s 393us/sample - loss: 0.458
3 - accuracy: 0.8346 - val_loss: 0.3103 - val_accuracy: 0.8908
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3011 - accur
acy: 0.8907
Epoch 00002: val_loss improved from 0.31033 to 0.27401, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 21s 386us/sample - loss: 0.301
0 - accuracy: 0.8907 - val_loss: 0.2740 - val_accuracy: 0.8996
Epoch 3/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2581 - accur
acy: 0.9051
```

```
Epoch 00003: val_loss improved from 0.27401 to 0.23791, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 410us/sample - loss: 0.258
1 - accuracy: 0.9051 - val_loss: 0.2379 - val_accuracy: 0.9150
Epoch 4/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2294 - accur
acy: 0.9157
Epoch 00004: val_loss improved from 0.23791 to 0.22551, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 410us/sample - loss: 0.229
6 - accuracy: 0.9156 - val_loss: 0.2255 - val_accuracy: 0.9182
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2053 - accur
acy: 0.9242
Epoch 00005: val_loss improved from 0.22551 to 0.21356, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 417us/sample - loss: 0.205
3 - accuracy: 0.9242 - val_loss: 0.2136 - val_accuracy: 0.9234
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1881 - accur
acy: 0.9316
Epoch 00006: val_loss did not improve from 0.21356
55000/55000 [==============================] - 23s 419us/sample - loss: 0.188
1 - accuracy: 0.9316 - val_loss: 0.2184 - val_accuracy: 0.9170
Epoch 7/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1681 - accur
acy: 0.9376
Epoch 00007: val_loss improved from 0.21356 to 0.20934, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 426us/sample - loss: 0.168
1 - accuracy: 0.9377 - val_loss: 0.2093 - val_accuracy: 0.9234
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1491 - accur
acy: 0.9451
Epoch 00008: val_loss did not improve from 0.20934
55000/55000 [==============================] - 23s 427us/sample - loss: 0.149
1 - accuracy: 0.9451 - val_loss: 0.2140 - val_accuracy: 0.9210
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1366 - accur
acy: 0.9500
Epoch 00009: val_loss did not improve from 0.20934
55000/55000 [==============================] - 24s 428us/sample - loss: 0.136
6 - accuracy: 0.9500 - val_loss: 0.2162 - val_accuracy: 0.9260
Epoch 10/10
```

```
54912/55000 [============================>.] - ETA: 0s - loss: 0.1232 - accur
acy: 0.9534
Epoch 00010: val_loss improved from 0.20934 to 0.20489, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 23s 424us/sample - loss: 0.123
2 - accuracy: 0.9534 - val_loss: 0.2049 - val_accuracy: 0.9310

 Test accuracy: 0.9231


Run: 10
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4715 - accur
acy: 0.8276
Epoch 00001: val_loss improved from inf to 0.30679, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 24s 437us/sample - loss: 0.471
5 - accuracy: 0.8275 - val_loss: 0.3068 - val_accuracy: 0.8906
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3052 - accur
acy: 0.8891
Epoch 00002: val_loss improved from 0.30679 to 0.27642, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 436us/sample - loss: 0.305
2 - accuracy: 0.8891 - val_loss: 0.2764 - val_accuracy: 0.9026
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2651 - accur
acy: 0.9034
Epoch 00003: val_loss improved from 0.27642 to 0.25428, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.265
1 - accuracy: 0.9034 - val_loss: 0.2543 - val_accuracy: 0.9124
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2390 - accur
acy: 0.9110
Epoch 00004: val_loss improved from 0.25428 to 0.22462, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 443us/sample - loss: 0.238
9 - accuracy: 0.9110 - val_loss: 0.2246 - val_accuracy: 0.9162
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2142 - accur
acy: 0.9211
Epoch 00005: val_loss did not improve from 0.22462
```

```
55000/55000 [==============================] - 25s 453us/sample - loss: 0.214
2 - accuracy: 0.9211 - val_loss: 0.2322 - val_accuracy: 0.9186
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1950 - accur
acy: 0.9276
Epoch 00006: val_loss improved from 0.22462 to 0.21826, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.195
0 - accuracy: 0.9275 - val_loss: 0.2183 - val_accuracy: 0.9200
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1751 - accur
acy: 0.9350
Epoch 00007: val_loss improved from 0.21826 to 0.20353, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 456us/sample - loss: 0.175
2 - accuracy: 0.9349 - val_loss: 0.2035 - val_accuracy: 0.9276
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1574 - accur
acy: 0.9410
Epoch 00008: val_loss did not improve from 0.20353
55000/55000 [==============================] - 25s 453us/sample - loss: 0.157
5 - accuracy: 0.9410 - val_loss: 0.2213 - val_accuracy: 0.9236
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1433 - accur
acy: 0.9461
Epoch 00009: val_loss did not improve from 0.20353
55000/55000 [==============================] - 25s 463us/sample - loss: 0.143
3 - accuracy: 0.9461 - val_loss: 0.2339 - val_accuracy: 0.9140
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1256 - accur
acy: 0.9528
Epoch 00010: val_loss did not improve from 0.20353
55000/55000 [==============================] - 26s 465us/sample - loss: 0.125
6 - accuracy: 0.9528 - val_loss: 0.2107 - val_accuracy: 0.9272

 Test accuracy: 0.9177


Average Model 1 Accuracy: 0.9195699989795685



Model 2
```

```
Run: 1
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4781 - accur
acy: 0.8278
Epoch 00001: val_loss improved from inf to 0.32479, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 24s 431us/sample - loss: 0.478
0 - accuracy: 0.8278 - val_loss: 0.3248 - val_accuracy: 0.8830
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3260 - accur
acy: 0.8804
Epoch 00002: val_loss improved from 0.32479 to 0.27195, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 433us/sample - loss: 0.326
0 - accuracy: 0.8805 - val_loss: 0.2720 - val_accuracy: 0.9006
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2850 - accur
acy: 0.8949
Epoch 00003: val_loss improved from 0.27195 to 0.25259, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 437us/sample - loss: 0.285
1 - accuracy: 0.8949 - val_loss: 0.2526 - val_accuracy: 0.9090
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2560 - accur
acy: 0.9055
Epoch 00004: val_loss improved from 0.25259 to 0.23696, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 436us/sample - loss: 0.256
0 - accuracy: 0.9055 - val_loss: 0.2370 - val_accuracy: 0.9178
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2380 - accur
acy: 0.9109
Epoch 00005: val_loss improved from 0.23696 to 0.22261, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 438us/sample - loss: 0.238
0 - accuracy: 0.9109 - val_loss: 0.2226 - val_accuracy: 0.9200
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2180 - accur
acy: 0.9192
Epoch 00006: val_loss improved from 0.22261 to 0.22236, saving model to model
.weights.best.hdf5
```

```
55000/55000 [==============================] - 24s 438us/sample - loss: 0.218
0 - accuracy: 0.9191 - val_loss: 0.2224 - val_accuracy: 0.9180
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2038 - accur
acy: 0.9243
Epoch 00007: val_loss improved from 0.22236 to 0.20472, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.203
8 - accuracy: 0.9243 - val_loss: 0.2047 - val_accuracy: 0.9228
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1901 - accur
acy: 0.9290
Epoch 00008: val_loss improved from 0.20472 to 0.20058, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 437us/sample - loss: 0.190
1 - accuracy: 0.9290 - val_loss: 0.2006 - val_accuracy: 0.9250
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1777 - accur
acy: 0.9317
Epoch 00009: val_loss improved from 0.20058 to 0.19673, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 440us/sample - loss: 0.177
6 - accuracy: 0.9317 - val_loss: 0.1967 - val_accuracy: 0.9304
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1671 - accur
acy: 0.9362
Epoch 00010: val_loss improved from 0.19673 to 0.19246, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 438us/sample - loss: 0.167
0 - accuracy: 0.9362 - val_loss: 0.1925 - val_accuracy: 0.9320

 Test accuracy: 0.9198


Run: 2
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4807 - accur
acy: 0.8260
Epoch 00001: val_loss improved from inf to 0.30993, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 25s 459us/sample - loss: 0.480
7 - accuracy: 0.8260 - val_loss: 0.3099 - val_accuracy: 0.8928
Epoch 2/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.3260 - accur
acy: 0.8809
Epoch 00002: val_loss improved from 0.30993 to 0.26864, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 451us/sample - loss: 0.326
1 - accuracy: 0.8809 - val_loss: 0.2686 - val_accuracy: 0.9050
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2841 - accur
acy: 0.8955
Epoch 00003: val_loss improved from 0.26864 to 0.25528, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 447us/sample - loss: 0.284
2 - accuracy: 0.8955 - val_loss: 0.2553 - val_accuracy: 0.9098
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2582 - accur
acy: 0.9039
Epoch 00004: val_loss improved from 0.25528 to 0.24295, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 449us/sample - loss: 0.258
2 - accuracy: 0.9039 - val_loss: 0.2430 - val_accuracy: 0.9104
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2388 - accur
acy: 0.9122
Epoch 00005: val_loss improved from 0.24295 to 0.22725, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 448us/sample - loss: 0.238
8 - accuracy: 0.9121 - val_loss: 0.2272 - val_accuracy: 0.9136
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2219 - accur
acy: 0.9173
Epoch 00006: val_loss improved from 0.22725 to 0.21427, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 466us/sample - loss: 0.221
9 - accuracy: 0.9173 - val_loss: 0.2143 - val_accuracy: 0.9190
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2031 - accur
acy: 0.9243
Epoch 00007: val_loss did not improve from 0.21427
55000/55000 [==============================] - 25s 453us/sample - loss: 0.203
1 - accuracy: 0.9243 - val_loss: 0.2143 - val_accuracy: 0.9204
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1910 - accur
acy: 0.9286
Epoch 00008: val_loss did not improve from 0.21427
```

```
55000/55000 [==============================] - 25s 453us/sample - loss: 0.191
0 - accuracy: 0.9286 - val_loss: 0.2208 - val_accuracy: 0.9184
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1799 - accur
acy: 0.9326
Epoch 00009: val_loss improved from 0.21427 to 0.20817, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 27s 488us/sample - loss: 0.179
9 - accuracy: 0.9325 - val_loss: 0.2082 - val_accuracy: 0.9230
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1683 - accur
acy: 0.9360
Epoch 00010: val_loss improved from 0.20817 to 0.19794, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 469us/sample - loss: 0.168
4 - accuracy: 0.9359 - val_loss: 0.1979 - val_accuracy: 0.9270

 Test accuracy: 0.921


Run: 3
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4787 - accur
acy: 0.8254
Epoch 00001: val_loss improved from inf to 0.32570, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 27s 486us/sample - loss: 0.478
7 - accuracy: 0.8254 - val_loss: 0.3257 - val_accuracy: 0.8824
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3267 - accur
acy: 0.8813
Epoch 00002: val_loss improved from 0.32570 to 0.26946, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 459us/sample - loss: 0.326
6 - accuracy: 0.8813 - val_loss: 0.2695 - val_accuracy: 0.9012
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2828 - accur
acy: 0.8949 E
Epoch 00003: val_loss did not improve from 0.26946
55000/55000 [==============================] - 25s 451us/sample - loss: 0.282
8 - accuracy: 0.8949 - val_loss: 0.2723 - val_accuracy: 0.8998
Epoch 4/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.2553 - accur
acy: 0.9044
Epoch 00004: val_loss improved from 0.26946 to 0.23522, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 456us/sample - loss: 0.255
3 - accuracy: 0.9044 - val_loss: 0.2352 - val_accuracy: 0.9134
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2352 - accur
acy: 0.9122
Epoch 00005: val_loss improved from 0.23522 to 0.22489, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 437us/sample - loss: 0.235
2 - accuracy: 0.9122 - val_loss: 0.2249 - val_accuracy: 0.9164
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2186 - accur
acy: 0.9186
Epoch 00006: val_loss improved from 0.22489 to 0.21426, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 449us/sample - loss: 0.218
6 - accuracy: 0.9186 - val_loss: 0.2143 - val_accuracy: 0.9170
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2023 - accur
acy: 0.9237
Epoch 00007: val_loss improved from 0.21426 to 0.20054, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 449us/sample - loss: 0.202
3 - accuracy: 0.9237 - val_loss: 0.2005 - val_accuracy: 0.9266
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1875 - accur
acy: 0.9292
Epoch 00008: val_loss did not improve from 0.20054
55000/55000 [==============================] - 25s 454us/sample - loss: 0.187
5 - accuracy: 0.9292 - val_loss: 0.2025 - val_accuracy: 0.9260
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1752 - accur
acy: 0.9347
Epoch 00009: val_loss improved from 0.20054 to 0.19670, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 456us/sample - loss: 0.175
2 - accuracy: 0.9347 - val_loss: 0.1967 - val_accuracy: 0.9294
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1649 - accur
acy: 0.9381
```

```
Epoch 00010: val_loss improved from 0.19670 to 0.19228, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 463us/sample - loss: 0.165
0 - accuracy: 0.9381 - val_loss: 0.1923 - val_accuracy: 0.9294

 Test accuracy: 0.9249


Run: 4
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4659 - accur
acy: 0.8316
Epoch 00001: val_loss improved from inf to 0.31928, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 26s 472us/sample - loss: 0.465
9 - accuracy: 0.8316 - val_loss: 0.3193 - val_accuracy: 0.8854
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3175 - accur
acy: 0.8841
Epoch 00002: val_loss improved from 0.31928 to 0.26321, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 467us/sample - loss: 0.317
4 - accuracy: 0.8841 - val_loss: 0.2632 - val_accuracy: 0.9060
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2776 - accur
acy: 0.8974
Epoch 00003: val_loss improved from 0.26321 to 0.24253, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 473us/sample - loss: 0.277
6 - accuracy: 0.8974 - val_loss: 0.2425 - val_accuracy: 0.9090
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2483 - accur
acy: 0.9083
Epoch 00004: val_loss improved from 0.24253 to 0.22904, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 470us/sample - loss: 0.248
3 - accuracy: 0.9083 - val_loss: 0.2290 - val_accuracy: 0.9168
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2322 - accur
acy: 0.9151
Epoch 00005: val_loss improved from 0.22904 to 0.21427, saving model to model
.weights.best.hdf5
```

```
55000/55000 [==============================] - 26s 481us/sample - loss: 0.232
1 - accuracy: 0.9151 - val_loss: 0.2143 - val_accuracy: 0.9224
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2150 - accur
acy: 0.9200
Epoch 00006: val_loss improved from 0.21427 to 0.20904, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 475us/sample - loss: 0.214
9 - accuracy: 0.9200 - val_loss: 0.2090 - val_accuracy: 0.9214
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1982 - accur
acy: 0.9265
Epoch 00007: val_loss did not improve from 0.20904
55000/55000 [==============================] - 26s 480us/sample - loss: 0.198
2 - accuracy: 0.9265 - val_loss: 0.2109 - val_accuracy: 0.9220
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1855 - accur
acy: 0.9307 ETA: 0s - loss: 0.1856 - accuracy:
Epoch 00008: val_loss improved from 0.20904 to 0.19484, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 27s 483us/sample - loss: 0.185
5 - accuracy: 0.9307 - val_loss: 0.1948 - val_accuracy: 0.9302
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1742 - accur
acy: 0.9343
Epoch 00009: val_loss did not improve from 0.19484
55000/55000 [==============================] - 27s 490us/sample - loss: 0.174
2 - accuracy: 0.9342 - val_loss: 0.1964 - val_accuracy: 0.9270
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1643 - accur
acy: 0.9383
Epoch 00010: val_loss did not improve from 0.19484
55000/55000 [==============================] - 27s 495us/sample - loss: 0.164
2 - accuracy: 0.9383 - val_loss: 0.2046 - val_accuracy: 0.9250

 Test accuracy: 0.9209


Run: 5
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4763 - accur
acy: 0.8257
```

```
Epoch 00001: val_loss improved from inf to 0.31145, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 28s 510us/sample - loss: 0.476
3 - accuracy: 0.8257 - val_loss: 0.3115 - val_accuracy: 0.8908
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3218 - accur
acy: 0.8818
Epoch 00002: val_loss improved from 0.31145 to 0.26618, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 508us/sample - loss: 0.321
8 - accuracy: 0.8818 - val_loss: 0.2662 - val_accuracy: 0.9066
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2792 - accur
acy: 0.8979
Epoch 00003: val_loss improved from 0.26618 to 0.24412, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 514us/sample - loss: 0.279
3 - accuracy: 0.8979 - val_loss: 0.2441 - val_accuracy: 0.9126
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2542 - accur
acy: 0.9067
Epoch 00004: val_loss did not improve from 0.24412
55000/55000 [==============================] - 29s 520us/sample - loss: 0.254
3 - accuracy: 0.9067 - val_loss: 0.2447 - val_accuracy: 0.9092
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2345 - accur
acy: 0.9123
Epoch 00005: val_loss improved from 0.24412 to 0.21708, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 29s 527us/sample - loss: 0.234
4 - accuracy: 0.9123 - val_loss: 0.2171 - val_accuracy: 0.9214
Epoch 6/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2158 - accur
acy: 0.9202
Epoch 00006: val_loss improved from 0.21708 to 0.21568, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 29s 532us/sample - loss: 0.215
9 - accuracy: 0.9202 - val_loss: 0.2157 - val_accuracy: 0.9182
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2014 - accur
acy: 0.9248
Epoch 00007: val_loss did not improve from 0.21568
55000/55000 [==============================] - 30s 540us/sample - loss: 0.201
4 - accuracy: 0.9248 - val_loss: 0.2214 - val_accuracy: 0.9164
```

```
Epoch 8/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1888 - accur
acy: 0.9297
Epoch 00008: val_loss improved from 0.21568 to 0.19971, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 31s 556us/sample - loss: 0.188
9 - accuracy: 0.9297 - val_loss: 0.1997 - val_accuracy: 0.9270
Epoch 9/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1766 - accur
acy: 0.9339
Epoch 00009: val_loss did not improve from 0.19971
55000/55000 [==============================] - 31s 562us/sample - loss: 0.176
6 - accuracy: 0.9339 - val_loss: 0.2125 - val_accuracy: 0.9234
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1676 - accur
acy: 0.9370
Epoch 00010: val_loss improved from 0.19971 to 0.19265, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 509us/sample - loss: 0.167
6 - accuracy: 0.9369 - val_loss: 0.1926 - val_accuracy: 0.9278

 Test accuracy: 0.9192


Run: 6
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4699 - accur
acy: 0.8300
Epoch 00001: val_loss improved from inf to 0.31789, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 25s 446us/sample - loss: 0.469
7 - accuracy: 0.8301 - val_loss: 0.3179 - val_accuracy: 0.8858
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3224 - accur
acy: 0.8829
Epoch 00002: val_loss improved from 0.31789 to 0.26064, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 428us/sample - loss: 0.322
4 - accuracy: 0.8829 - val_loss: 0.2606 - val_accuracy: 0.9094
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2803 - accur
acy: 0.8963
```

```
Epoch 00003: val_loss improved from 0.26064 to 0.25092, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 432us/sample - loss: 0.280
4 - accuracy: 0.8963 - val_loss: 0.2509 - val_accuracy: 0.9130
Epoch 4/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2547 - accur
acy: 0.9050
Epoch 00004: val_loss improved from 0.25092 to 0.22857, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 436us/sample - loss: 0.255
0 - accuracy: 0.9050 - val_loss: 0.2286 - val_accuracy: 0.9136
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2349 - accur
acy: 0.9138
Epoch 00005: val_loss improved from 0.22857 to 0.22056, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 436us/sample - loss: 0.234
9 - accuracy: 0.9138 - val_loss: 0.2206 - val_accuracy: 0.9164
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2177 - accur
acy: 0.9196
Epoch 00006: val_loss improved from 0.22056 to 0.20865, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 439us/sample - loss: 0.217
7 - accuracy: 0.9196 - val_loss: 0.2086 - val_accuracy: 0.9212
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2038 - accur
acy: 0.9235
Epoch 00007: val_loss did not improve from 0.20865
55000/55000 [==============================] - 24s 444us/sample - loss: 0.203
8 - accuracy: 0.9235 - val_loss: 0.2217 - val_accuracy: 0.9218
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1862 - accur
acy: 0.9299
Epoch 00008: val_loss improved from 0.20865 to 0.19579, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.186
1 - accuracy: 0.9299 - val_loss: 0.1958 - val_accuracy: 0.9266
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1777 - accur
acy: 0.9331
Epoch 00009: val_loss did not improve from 0.19579
55000/55000 [==============================] - 24s 439us/sample - loss: 0.177
6 - accuracy: 0.9331 - val_loss: 0.2126 - val_accuracy: 0.9220
```

```
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1649 - accur
acy: 0.9379
Epoch 00010: val_loss improved from 0.19579 to 0.19230, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 24s 441us/sample - loss: 0.165
0 - accuracy: 0.9379 - val_loss: 0.1923 - val_accuracy: 0.9286

 Test accuracy: 0.9244


Run: 7
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4831 - accur
acy: 0.8242
Epoch 00001: val_loss improved from inf to 0.31229, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 25s 455us/sample - loss: 0.483
3 - accuracy: 0.8241 - val_loss: 0.3123 - val_accuracy: 0.8886
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3305 - accur
acy: 0.8785
Epoch 00002: val_loss improved from 0.31229 to 0.27554, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 458us/sample - loss: 0.330
5 - accuracy: 0.8785 - val_loss: 0.2755 - val_accuracy: 0.8996
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2895 - accur
acy: 0.8934
Epoch 00003: val_loss improved from 0.27554 to 0.25142, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 448us/sample - loss: 0.289
4 - accuracy: 0.8934 - val_loss: 0.2514 - val_accuracy: 0.9096
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2618 - accur
acy: 0.9028
Epoch 00004: val_loss did not improve from 0.25142
55000/55000 [==============================] - 24s 445us/sample - loss: 0.261
7 - accuracy: 0.9028 - val_loss: 0.2532 - val_accuracy: 0.9036
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2412 - accur
acy: 0.9105
```

```
Epoch 00005: val_loss improved from 0.25142 to 0.22893, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 449us/sample - loss: 0.241
2 - accuracy: 0.9105 - val_loss: 0.2289 - val_accuracy: 0.9132
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2232 - accur
acy: 0.9156
Epoch 00006: val_loss improved from 0.22893 to 0.21231, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 448us/sample - loss: 0.223
3 - accuracy: 0.9155 - val_loss: 0.2123 - val_accuracy: 0.9190
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2077 - accur
acy: 0.9214
Epoch 00007: val_loss improved from 0.21231 to 0.20779, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 451us/sample - loss: 0.207
7 - accuracy: 0.9214 - val_loss: 0.2078 - val_accuracy: 0.9218
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1960 - accur
acy: 0.9262
Epoch 00008: val_loss improved from 0.20779 to 0.20575, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 451us/sample - loss: 0.196
0 - accuracy: 0.9262 - val_loss: 0.2058 - val_accuracy: 0.9230
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1811 - accur
acy: 0.9314
Epoch 00009: val_loss did not improve from 0.20575
55000/55000 [==============================] - 25s 450us/sample - loss: 0.181
1 - accuracy: 0.9314 - val_loss: 0.2163 - val_accuracy: 0.9200
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1703 - accur
acy: 0.9355
Epoch 00010: val_loss improved from 0.20575 to 0.19464, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 456us/sample - loss: 0.170
3 - accuracy: 0.9355 - val_loss: 0.1946 - val_accuracy: 0.9294

 Test accuracy: 0.921


Run: 8
Train on 55000 samples, validate on 5000 samples
```

```
Epoch 1/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.4839 - accur
acy: 0.8268
Epoch 00001: val_loss improved from inf to 0.31099, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 25s 463us/sample - loss: 0.483
6 - accuracy: 0.8269 - val_loss: 0.3110 - val_accuracy: 0.8902
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3314 - accur
acy: 0.8793
Epoch 00002: val_loss improved from 0.31099 to 0.29696, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 464us/sample - loss: 0.331
3 - accuracy: 0.8793 - val_loss: 0.2970 - val_accuracy: 0.8922
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2893 - accur
acy: 0.8938
Epoch 00003: val_loss improved from 0.29696 to 0.25608, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 460us/sample - loss: 0.289
3 - accuracy: 0.8938 - val_loss: 0.2561 - val_accuracy: 0.9046
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2587 - accur
acy: 0.9046
Epoch 00004: val_loss improved from 0.25608 to 0.23984, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 460us/sample - loss: 0.258
7 - accuracy: 0.9045 - val_loss: 0.2398 - val_accuracy: 0.9114
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2386 - accur
acy: 0.9112
Epoch 00005: val_loss improved from 0.23984 to 0.21762, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 460us/sample - loss: 0.238
6 - accuracy: 0.9112 - val_loss: 0.2176 - val_accuracy: 0.9188
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2188 - accur
acy: 0.9192
Epoch 00006: val_loss improved from 0.21762 to 0.21722, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 25s 458us/sample - loss: 0.218
8 - accuracy: 0.9192 - val_loss: 0.2172 - val_accuracy: 0.9198
Epoch 7/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.2037 - accur
acy: 0.9238
Epoch 00007: val_loss improved from 0.21722 to 0.20890, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 465us/sample - loss: 0.203
7 - accuracy: 0.9238 - val_loss: 0.2089 - val_accuracy: 0.9176
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1928 - accur
acy: 0.9284
Epoch 00008: val_loss improved from 0.20890 to 0.20100, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 465us/sample - loss: 0.192
8 - accuracy: 0.9284 - val_loss: 0.2010 - val_accuracy: 0.9242
Epoch 9/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1751 - accur
acy: 0.9336
Epoch 00009: val_loss did not improve from 0.20100
55000/55000 [==============================] - 26s 476us/sample - loss: 0.175
1 - accuracy: 0.9337 - val_loss: 0.2012 - val_accuracy: 0.9252
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1665 - accur
acy: 0.9367
Epoch 00010: val_loss improved from 0.20100 to 0.19973, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 26s 474us/sample - loss: 0.166
5 - accuracy: 0.9367 - val_loss: 0.1997 - val_accuracy: 0.9248

 Test accuracy: 0.9223


Run: 9
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4822 - accur
acy: 0.8253
Epoch 00001: val_loss improved from inf to 0.34114, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 27s 497us/sample - loss: 0.482
2 - accuracy: 0.8252 - val_loss: 0.3411 - val_accuracy: 0.8786
Epoch 2/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.3275 - accur
acy: 0.8811
Epoch 00002: val_loss improved from 0.34114 to 0.28331, saving model to model
.weights.best.hdf5
```

```
55000/55000 [==============================] - 27s 495us/sample - loss: 0.327
6 - accuracy: 0.8811 - val_loss: 0.2833 - val_accuracy: 0.8946
Epoch 3/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2842 - accur
acy: 0.8950
Epoch 00003: val_loss improved from 0.28331 to 0.24849, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 27s 500us/sample - loss: 0.284
3 - accuracy: 0.8949 - val_loss: 0.2485 - val_accuracy: 0.9134
Epoch 4/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2566 - accur
acy: 0.9051
Epoch 00004: val_loss improved from 0.24849 to 0.23257, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 504us/sample - loss: 0.256
5 - accuracy: 0.9052 - val_loss: 0.2326 - val_accuracy: 0.9120
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2376 - accur
acy: 0.9119
Epoch 00005: val_loss improved from 0.23257 to 0.22980, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 28s 503us/sample - loss: 0.237
7 - accuracy: 0.9119 - val_loss: 0.2298 - val_accuracy: 0.9148
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2182 - accur
acy: 0.9194
Epoch 00006: val_loss did not improve from 0.22980
55000/55000 [==============================] - 28s 513us/sample - loss: 0.218
2 - accuracy: 0.9193 - val_loss: 0.2505 - val_accuracy: 0.9086
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2050 - accur
acy: 0.9238
Epoch 00007: val_loss improved from 0.22980 to 0.21190, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 29s 519us/sample - loss: 0.205
0 - accuracy: 0.9238 - val_loss: 0.2119 - val_accuracy: 0.9224
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1923 - accur
acy: 0.9291
Epoch 00008: val_loss did not improve from 0.21190
55000/55000 [==============================] - 29s 523us/sample - loss: 0.192
3 - accuracy: 0.9291 - val_loss: 0.2180 - val_accuracy: 0.9190
Epoch 9/10
```

```
54976/55000 [============================>.] - ETA: 0s - loss: 0.1774 - accur
acy: 0.9335
Epoch 00009: val_loss improved from 0.21190 to 0.20956, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 29s 528us/sample - loss: 0.177
4 - accuracy: 0.9335 - val_loss: 0.2096 - val_accuracy: 0.9274
Epoch 10/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1687 - accur
acy: 0.9370
Epoch 00010: val_loss improved from 0.20956 to 0.19283, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 30s 540us/sample - loss: 0.168
7 - accuracy: 0.9370 - val_loss: 0.1928 - val_accuracy: 0.9296

 Test accuracy: 0.9206


Run: 10
Train on 55000 samples, validate on 5000 samples
Epoch 1/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.4896 - accur
acy: 0.8226
Epoch 00001: val_loss improved from inf to 0.34298, saving model to model.wei
ghts.best.hdf5
55000/55000 [==============================] - 31s 563us/sample - loss: 0.489
5 - accuracy: 0.8227 - val_loss: 0.3430 - val_accuracy: 0.8762
Epoch 2/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.3318 - accur
acy: 0.8791
Epoch 00002: val_loss improved from 0.34298 to 0.28180, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 31s 557us/sample - loss: 0.331
8 - accuracy: 0.8792 - val_loss: 0.2818 - val_accuracy: 0.8986
Epoch 3/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2912 - accur
acy: 0.8928
Epoch 00003: val_loss improved from 0.28180 to 0.25772, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 31s 565us/sample - loss: 0.291
2 - accuracy: 0.8928 - val_loss: 0.2577 - val_accuracy: 0.9062
Epoch 4/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.2626 - accur
acy: 0.9024
```

```
Epoch 00004: val_loss improved from 0.25772 to 0.24391, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 32s 580us/sample - loss: 0.262
6 - accuracy: 0.9024 - val_loss: 0.2439 - val_accuracy: 0.9094
Epoch 5/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2417 - accur
acy: 0.9112
Epoch 00005: val_loss improved from 0.24391 to 0.22267, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 32s 589us/sample - loss: 0.241
7 - accuracy: 0.9112 - val_loss: 0.2227 - val_accuracy: 0.9164
Epoch 6/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2217 - accur
acy: 0.9181
Epoch 00006: val_loss improved from 0.22267 to 0.21748, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 34s 613us/sample - loss: 0.221
7 - accuracy: 0.9181 - val_loss: 0.2175 - val_accuracy: 0.9196
Epoch 7/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.2093 - accur
acy: 0.9216
Epoch 00007: val_loss improved from 0.21748 to 0.21060, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 35s 633us/sample - loss: 0.209
2 - accuracy: 0.9216 - val_loss: 0.2106 - val_accuracy: 0.9186
Epoch 8/10
54976/55000 [============================>.] - ETA: 0s - loss: 0.1943 - accur
acy: 0.9262
Epoch 00008: val_loss improved from 0.21060 to 0.20436, saving model to model
.weights.best.hdf5
55000/55000 [==============================] - 36s 653us/sample - loss: 0.194
3 - accuracy: 0.9262 - val_loss: 0.2044 - val_accuracy: 0.9226
Epoch 9/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1800 - accur
acy: 0.9321
Epoch 00009: val_loss did not improve from 0.20436
55000/55000 [==============================] - 35s 645us/sample - loss: 0.180
2 - accuracy: 0.9320 - val_loss: 0.2144 - val_accuracy: 0.9234
Epoch 10/10
54912/55000 [============================>.] - ETA: 0s - loss: 0.1690 - accur
acy: 0.9370
Epoch 00010: val_loss did not improve from 0.20436
55000/55000 [==============================] - 34s 613us/sample - loss: 0.169
0 - accuracy: 0.9371 - val_loss: 0.2107 - val_accuracy: 0.9238
```

```
Test accuracy: 0.9206
```

```
Average Model 2 Accuracy: 0.9214699923992157
```

These results show that model one and model two have approximately the same test accuracy, even though their training accuracy is a very different relative to their test set accuracy. Model one has a training accuracy of 95%, while model two has a training accuracy of 93%. One can also see that their validation set accuracies are roughly the same at 92% on average.

This information leads me to conclude that model two may actually be better than model one, despite the differences in training accuracy. The validation set and test set accuracies are the same on average. That means model one performs no better than model two in practice. Since model one has higher training than test accuracy that likely means it's over fitting and performing worse on the test and validation sets because it's learning the training set too well. Model two is most likely performing better because it has the extra dropout later before the flattening step. This means that model two has less training information available which would allow it to generalize more.