In [2]:

```
# Import necessary modules
from keras.layers import Dense
from keras.models import Sequential
from keras.utils import to_categorical
import pandas as pd

df = pd.read_csv('project_image_001.csv')

print(df.shape)

X = df.iloc[:,1:785]

n_cols = X.shape[1]

# Convert the target to categorical: target
y = to_categorical(df.iloc[:,0])

input_shape = (n_cols,)

# Import EarlyStopping
from keras.callbacks import EarlyStopping

# Define early_stopping_monitor
early_stopping_monitor = EarlyStopping(patience=2)

# Create the model: model
model = Sequential()

# Add the first hidden layer
model.add(Dense(50, activation='relu', input_shape=(784,)))

# Add the second hidden layer
model.add(Dense(50, activation='relu'))

# Add the output layer
model.add(Dense(10, activation='softmax'))

# Compile the model
model.compile(optimizer='adam', loss='categorical_crossentropy', metrics=['accuracy'])

# Fit the model
model.fit(X, y, validation_split=0.3, epochs=60, callbacks=[early_stopping_monitor])
```

```
(2000, 785)
Train on 1400 samples, validate on 600 samples
Epoch 1/60
1400/1400 [==============================] - 1s 395us/step - loss: 11.7605 - acc: 0.2393 - va
l_loss: 10.0577 - val_acc: 0.3517
Epoch 2/60
1400/1400 [==============================] - 0s 79us/step - loss: 9.8712 - acc: 0.3750 - val_
loss: 9.7600 - val_acc: 0.3900
Epoch 3/60
1400/1400 [==============================] - 0s 76us/step - loss: 9.3760 - acc: 0.4121 - val_
loss: 9.6646 - val_acc: 0.3900
Epoch 4/60
1400/1400 [==============================] - 0s 83us/step - loss: 9.2429 - acc: 0.4164 - val_
loss: 10.0772 - val_acc: 0.3617
Epoch 5/60
1400/1400 [==============================] - 0s 78us/step - loss: 9.0574 - acc: 0.4307 - val_
loss: 9.5562 - val_acc: 0.3967
Epoch 6/60
1400/1400 [==============================] - 0s 78us/step - loss: 8.6157 - acc: 0.4571 - val_
loss: 10.0853 - val_acc: 0.3583
Epoch 7/60
1400/1400 [==============================] - 0s 76us/step - loss: 8.7417 - acc: 0.4521 - val_
loss: 8.9425 - val_acc: 0.4400
Epoch 8/60
1400/1400 [==============================] - 0s 78us/step - loss: 8.1917 - acc: 0.4836 - val_
loss: 8.2968 - val_acc: 0.4767
Epoch 9/60
1400/1400 [==============================] - 0s 79us/step - loss: 8.2195 - acc: 0.4821 - val_
loss: 8.5131 - val_acc: 0.4617
Epoch 10/60
1400/1400 [==============================] - 0s 82us/step - loss: 8.0568 - acc: 0.4900 - val_
loss: 8.2884 - val_acc: 0.4750
Epoch 11/60
1400/1400 [==============================] - 0s 80us/step - loss: 7.8389 - acc: 0.5043 - val_
loss: 8.0716 - val_acc: 0.4917
Epoch 12/60
1400/1400 [==============================] - 0s 80us/step - loss: 7.7713 - acc: 0.5100 - val_
loss: 8.2492 - val_acc: 0.4800
Epoch 13/60
1400/1400 [==============================] - 0s 80us/step - loss: 7.7212 - acc: 0.5171 - val_
loss: 8.2102 - val_acc: 0.4850
```

Out[2]: <keras.callbacks.History at 0x28e46904be0>

In [ ]: