**EE3314-01**

**Introduction to Artificial Intelligence**

**Project Report**

**Due:** December 20, 2019


**Group:**

| | |
|---|---|
| Kevin Chung | 2019843301 |
| Justin Kim | 2019843708 |
| Kevin Liu | 2019843296 |
| Walter Nam | 2019843014 |

## 1. Introduction

Within this project, we learned about convolutional neural networks (CNN) and implemented our knowledge collaboratively. In the first question, we implemented a simple CNN and classified/visualized the distribution of MINIST features as a beginning procedure in order to familiarize ourselves with training CNNs. Following, we put that training into practice by adopting the AlexNet CNN model with the CIFAR-10 dataset. Lastly, we visualized the capability of transfer learning in the third question by using the trained AlexNet model with the IMDB_WIKI dataset and attempted to observe its performance. By working on this project, we were able to develop more understanding behind CNNs and gain first-hand experiences working with these networks.

## 2. Problems and Solutions

### 2.1 Question 1: MINIST Features Visualization with CNN

For the first part of question one, we needed to implement a simple CNN to visualize the distribution of MNIST features in a two-dimensional scatter plot. Thus, we trained our network with the given specifications and used the following hyperparameters.

| **Training Hyperparameters** | |
|---|---|
| Learning Rate Schedule | 0.1 |
| Momentum Coefficient | 0.9 |
| L2 Coefficient | 1e-6 |
| Dropout Rate | 0.5 |
| Batch Number | 64 |

*Table 1. Training Hyperparameters for CNN*

After training our network, we plotted two graphs: “Cross Entropy Loss vs. Epoch” and “Classification Accuracy vs. Epoch”. In the following plots, the blue and red lines represents the training and testing set respectively.

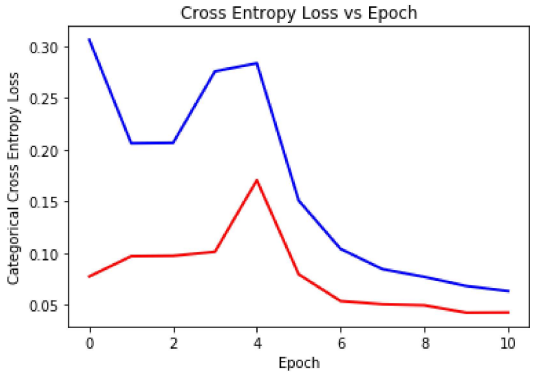



*Figure 1. Cross Entropy Loss vs. Epoch*

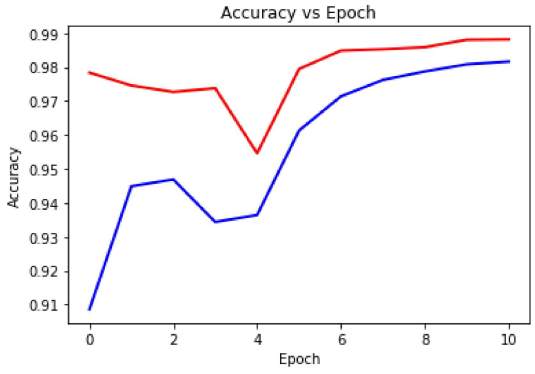



*Figure 2. Classification Accuracy vs. Epoch*

| **Training Accuracy (%)** | **Testing Accuracy (%)** |
|---|---|
| 98.17 | 98.82 |

*Table 2. Accuracy for First CNN*

For the second part, we modified the CNN to follow the given specifications on the various stages and trained our network again. This time around, we used the following values for the hyperparameters and calculated the accuracies in the following table. Then, we plotted the scatter plot with the activation values.

| **Training Hyperparameters** | |
|---|---|
| Learning Rate Schedule | ReduceLR on Plateau |
| Momentum Coefficient | 0.9 |
| L2 Coefficient | 1e-6 |
| Dropout Rate | 0.5 |
| Batch Number | 64 |

*Table 3. Training Hyperparameters for Modified CNN*

| Training Accuracy (%) | Testing Accuracy (%) |
|---|---|
| 98.75 | 99.08 |

*Table 4. Accuracy for Modified CNN*

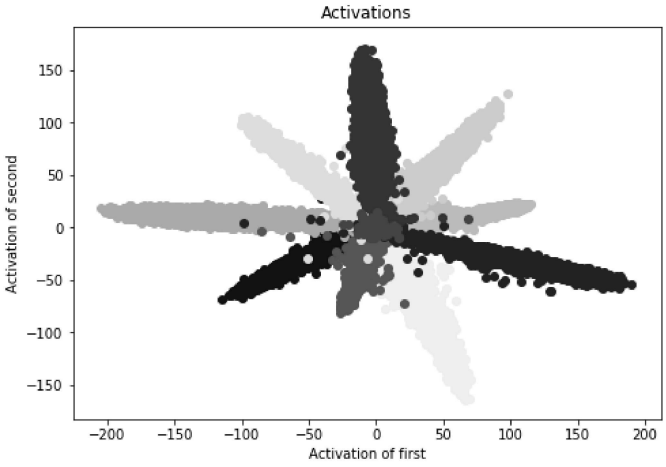



*Figure 3. Activation Distributed Scatter Plot*

### 2.2 Question 2: AlexNet with CIFAR-10

For this question, we needed to adopt an AlexNet CNN to train the CIFAR-10 dataset. In order to do so, we utilized the following hyperparameters for the training setup and had the model layer setup as stated in the next table.

| Training Hyperparameters | |
|---|---|
| Learning Rate Schedule | 0.0001 |
| Momentum Coefficient | 0.9 |
| L2 Coefficient | 0.01 |
| Dropout Rate (5 layers used) | 0.2/0.3/0.4/0.5/0.5 |
| Batch Number | 128 |

*Table 6. Training Hyperparameters for CNN*

| Layer | # of filters/filter size (Window)/Stride/Zero Padding | Feature Map Size | # of weights | # of neurons (Memory) | # of conv operations | Receptive Field |
|---|---|---|---|---|---|---|
| Input | - | 32x32x3 | 0 | 3072 | 0 | 1 |
| Conv1 | 32/3x3/1/same | 32x32x32 | 864 | 32768 | 884736 | 3 |
| Pool1 | 32/2x2/2/none | 16x16x32 | 0 | 8192 | 32768 | 4 |
| Conv2 | 64/3x3/1/same | 16x16x64 | 18432 | 16384 | 4718592 | 8 |
| Pool2 | 64/2x2/2/none | 8x8x64 | 0 | 4096 | 16384 | 10 |
| Conv3 | 128/3x3/1/same | 8x8x128 | 73728 | 8192 | 4718592 | 18 |
| Conv4 | 256/3x3/1/same | 8x8x256 | 294912 | 16384 | 18874368 | 26 |
| Conv5 | 512/3x3/1/same | 8x8x512 | 1179648 | 32768 | 75497472 | 34 |
| Pool3 | 512/2x2/2/none | 4x4x512 | 0 | 8192 | 32768 | 38 |
| FC1 | - | 512 | 4194304 | 512 | 262144 | - |
| FC2 | - | 256 | 131072 | 256 | 131072 | - |
| Output | - | 10 | 20480 | 10 | 20480 | - |
| **Total** | **-** | **-** | **5913440** | **130826** | **105189376** | **-** |

*Table 5. AlexNet Network Layers*

After training the network, we plotted the loss and accuracy versus the epochs; in the following plots, the blue and red lines represents the training and testing set respectively. The final accuracy is written in the following table.

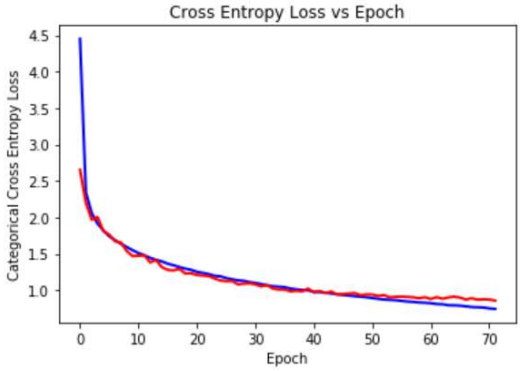



*Figure 4. Cross Entropy Loss vs. Epoch*

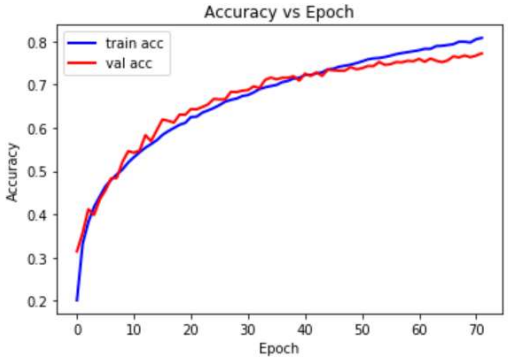



*Figure 5. Classification Accuracy vs. Epoch*

| Training Accuracy (%) | Testing Accuracy (%) |
|---|---|
| 80.8 | 77.2 |

*Table 6. Accuracy for AlexNet CNN*

### 2.3 Question 3: Transfer Learning

For this part of the question, we needed to use transfer training and implement the baseline model by retraining the last (softmax) layer with the IMDB_WIKI face training samples. Thus, we utilized the following hyperparameters and attempted to plot the loss and accuracy versus epochs, receiving the final accuracy values in the last table. Secondly, we needed to fine-tune a few of the convolution and fully-connected layers and train the networks again. After training the network once again, we would need to plot the graphs and record the final accuracy values once again.

| Training Hyperparameters | |
|---|---|
| Learning Rate Schedule | 0.01 |
| Number of Epochs | 500 |

*Table 7. Training Hyperparameters for Transfer Learning*

Note: Following 2 graphs are for the baseline model

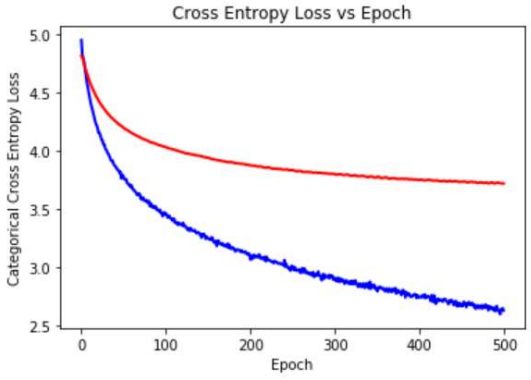



*Figure 6. Cross Entropy Loss vs. Epoch*

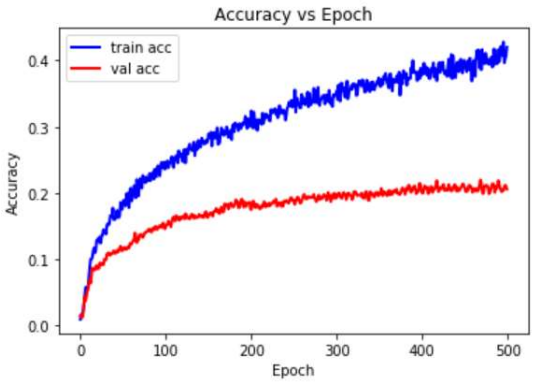



*Figure 7. Classification Accuracy vs. Epoch*

| Training Accuracy (%) | Testing Accuracy (%) |
|---|---|
| 42 | 21 |

*Table 8. Accuracy for Baseline Model*

For the fine tuned model, we attempted to change the model so that the two fully connected layers as well as the last convolution layers were trainable when running the IMDB face dataset. The same number of epochs (500) were run.

Note: Following 2 graphs are for the fine tuned model

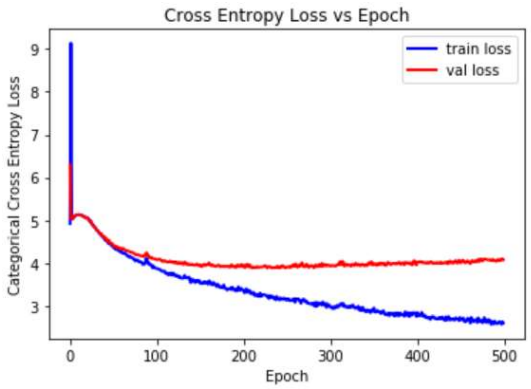



*Figure 6. Cross Entropy Loss vs. Epoch*

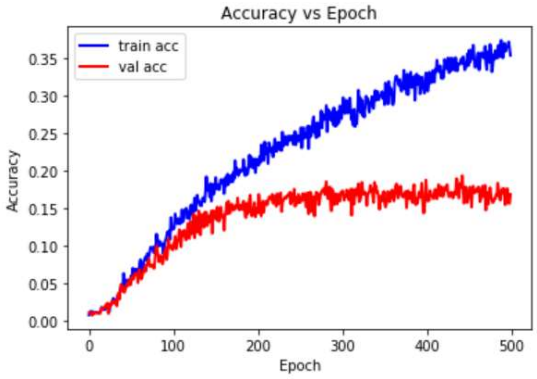



*Figure 7. Classification Accuracy vs. Epoch*

| Training Accuracy (%) | Testing Accuracy (%) |
|---|---|
| 35 | 16 |

*Table 8. Accuracy for Tuned Model*

## 3. Discussion

### 3.1 Question 1: MINIST Features Visualization with CNN

Reflecting on the plot, the classification performance is generally accurate, showcasing the features from different classes in clear colored streaks in the outer

wings of the scatter plot. However, there are some outliers that can be seen from various colored points outside of their set around the center of the plot; there is no clear distinction between the different classes towards the center unlike the training set scatter plot, meaning that it is very difficult to differentiate the outputs' classes when the activation values of both first and second neurons approaches to zero. Otherwise, the distinction between classes are clearly evident in the plot. Thus, the classification is generally accurate despite some outliers seen in the plot.

### 3.2 Question 2: AlexNet with CIFAR-10

For Question 2, we trained an AlexNet architecture CNN with the CIFAR-10 dataset. Our deep learning library of choice was keras tensorflow. Due to the input dimensions of the images, the feature map vanishes at the later layers, so we had to manually tweak the stride and kernel dimensions to prevent vanishing filters. In solving this question, we experienced various obstacles that we needed to overcome. An issue was finding the hyperparameters manually, using various test cases in order to find the proper variables to obtain an accuracy over eighty percent. An issue that came with this problem was having a slow machine due to an inferior CPU/GPU, making it very difficult to quickly run the network when changing the values. In addition, convergence was rather slow. Thus, we utilized Google Colab in order to help alleviate that problem which sped up the epoch calculations and produced an output quicker.

### 3.3 Question 3: Transfer Learning

For transfer learning, we encountered problems when attempting to generalize the results from the CIFAR-10 database to the IMDB database. When we attempted to train the network we got from CIFAR-10 with IMDB, the model trained extremely

slowly and did not converge to an acceptable accuracy value. We made several attempts to alleviate the problem such as by unfreezing certain layers, increasing the learning rate, changing the way we augmented the image data, and other methods, but nothing seemed to improve our training speed. While the model did increase in accuracy, it did so very slowly and very slightly. Because of this, we either assumed that our model did not generalize well to the context of the IMDB database, or that the problem between classifying random objects versus classifying specific people is not similar enough to be generalized for transfer learning.

When attempting to enhance our model through fine tuning, we found that even changing hyperparameters and allowing certain layers to be trainable did not help in increasing training accuracy. This could be attributed to an error in either our model or image data setup. Additionally, it could also mean that the model learned for CIFAR-10 cannot easily generalize to the IMDB database even with more trainable layers.

## 4. Conclusion

By completing all three questions in this project, we have become more versed in the area of convolutional neural networks, knowing how to implement CNNs models in artificial intelligence in order to utilize them in analyzing the datasets. We implemented a simple CNN and observed its performance, adopted the AlexNet model and trained it to the CIFAR-10 dataset, and utilized transfer learning with the AlexNet model to generalize the IMDB-WIKI dataset. By working these given problems, we were able to complete the project and get hands-on experience with implementing convolutional neural networks of artificial intelligence.

## 5. Contributions

| **Group Member** | **Section** |
|---|---|
| Kevin Chung | Question 2 & Question 3 |
| Justin Kim | Question 1 & Question 3 |
| Kevin Liu | Question 2 & Report |
| Walter Nam | Question 2 & Report |

```
# Introduction to AI: Final Project
# Question 1, Part 1


import keras
import matplotlib.pyplot as plt
from keras import backend as K
import numpy as np
```

```
Using TensorFlow backend.
```

The default version of TensorFlow in Colab will soon switch to TensorFlow 2.x.
We recommend you upgrade now or ensure your notebook will continue to use TensorFlow 1.x via the %tenso

```
from keras.datasets import mnist

# input image dimensions
img_rows, img_cols = 28, 28

num_classes = 10

# the data, split between train and test sets
(x_train, y_train_num), (x_test, y_test_num) = mnist.load_data()


x_train = x_train.reshape(x_train.shape[0], img_rows, img_cols, 1)
x_test = x_test.reshape(x_test.shape[0], img_rows, img_cols, 1)
input_shape = (img_rows, img_cols, 1)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255

y_train = keras.utils.to_categorical(y_train_num, num_classes)
y_test = keras.utils.to_categorical(y_test_num, num_classes)


#hyper_parameters
batch_size = 64
epochs = 11
dropout_rate = .5
alpha = .2


from keras.models import Sequential
from keras.layers import Dense, Conv2D, Flatten, MaxPooling2D, Dropout


from keras.layers import LeakyReLU
from keras.layers import ZeroPadding2D
```

```
model2 = Sequential()

# Stage 1
model2.add(ZeroPadding2D(2))
model2.add(Conv2D(32,
                  kernel_size=(5,5),
                  strides=1,
                  #padding='valid',
                  input_shape=(28,28,1)))
                  #activation=LeakyReLU(),
model2.add(LeakyReLU(alpha=alpha))

model2.add(ZeroPadding2D(padding=(2, 2)))
model2.add(Conv2D(32,
                  kernel_size=(5,5),
                  strides=1))
                  #padding='valid'))
                  #activation=LeakyReLU()))
model2.add(LeakyReLU(alpha=alpha))


model2.add(MaxPooling2D(pool_size=(2,2),
                        strides=(2,2),
                        padding='valid'))
model2.add(Dropout(dropout_rate))

# Stage 2
for i in range(2):
  model2.add(ZeroPadding2D(2))
  model2.add(Conv2D(64,
                    kernel_size=(5,5),
                    strides=1))
                    #padding='valid'))
                    #activation=LeakyReLU()))
  model2.add(LeakyReLU(alpha=alpha))


model2.add(MaxPooling2D(pool_size=(2,2),
                        strides=(2,2),
                        padding='valid'))
model2.add(Dropout(dropout_rate))

# Stage 3
for i in range(2):
  model2.add(ZeroPadding2D(2))
  model2.add(Conv2D(128,
                    kernel_size=(5,5),
                    strides=1))
                    #padding='valid'))
  model2.add(LeakyReLU(alpha=alpha))

                    #activation=LeakyReLU()))
```

```
                  #activation=LeakyReLU()))
model2.add(MaxPooling2D(pool_size=(2,2),
                        strides=(2,2),
                        padding='valid'))


# Stage 4
model2.add(Flatten())
model2.add(Dense(2,
                 name='stage4',
                activation='linear'))
#model2.add(Dropout(dropout_rate))

# Stage
model2.add(Dense(10,
                activation='softmax'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
```

```
model = Sequential()

# Stage 1
model.add(Conv2D(20,
                 kernel_size=(5,5),
                 strides=1,
                 padding='valid',
                 activation='relu',
                 input_shape=input_shape))

model.add(MaxPooling2D(pool_size=(2,2),
                       strides=(2,2),
                       padding='valid'))
#model.add(Dropout(0.25))

# Stage 2
model.add(Conv2D(50,
                 kernel_size=(5,5),
                 strides=1,
                 padding='valid',
                 activation='relu'))

model.add(MaxPooling2D(pool_size=(2,2),
                       strides=(2,2),
                       padding='valid'))
model.add(Dropout(0.45))

# Stage 3
model.add(Flatten())
model.add(Dense(500,
                activation='relu'))
```

```
model.add(Dropout(0.25))

# Stage 4
model.add(Dense(10,
                activation='softmax'))
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
Instructions for updating:
Please use `rate` instead of `keep_prob`. Rate should be set to `rate = 1 - keep_prob`.
```

```
from keras.optimizers import SGD


sgd = SGD(lr=0.1, decay=1e-6, momentum=.9)

model.compile(optimizer=sgd,
              loss=keras.losses.categorical_crossentropy,
              metrics=['accuracy'])
model2.compile(optimizer=sgd,
              loss=keras.losses.categorical_crossentropy,
              metrics=['accuracy'])
```

```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/optimizers.py:793:

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_
```

## Part 1

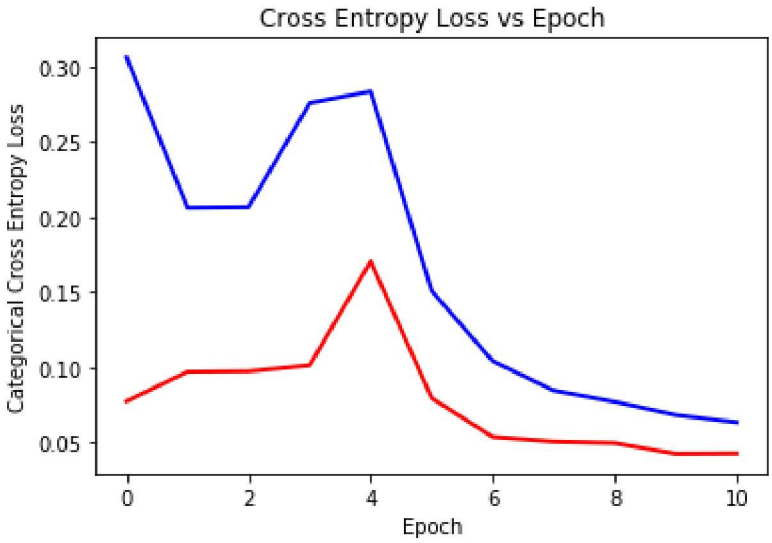
```
from keras.callbacks import ReduceLROnPlateau

reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.2, patience=3, min_lr=0.0000001)

history = model.fit(x_train,
          y_train,
          validation_data=(x_test, y_test),
          callbacks=[reduce_lr],
          batch_size=batch_size,
          epochs=epochs)
```

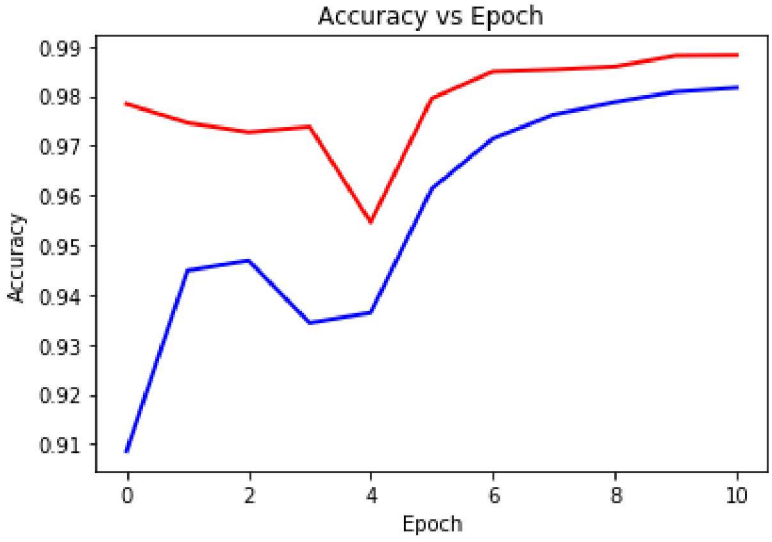
```
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/tensorflow_core/python/op
Instructions for updating:
Use tf.where in 2.0, which has the same broadcast rule as np.where
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

Train on 60000 samples, validate on 10000 samples
Epoch 1/11
WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

WARNING:tensorflow:From /usr/local/lib/python3.6/dist-packages/keras/backend/tensorflow_

60000/60000 [==============================] - 11s 181us/step - loss: 0.3062 - acc: 0.90
Epoch 2/11
60000/60000 [==============================] - 4s 66us/step - loss: 0.2063 - acc: 0.9449
Epoch 3/11
60000/60000 [==============================] - 4s 68us/step - loss: 0.2067 - acc: 0.9469
Epoch 4/11
60000/60000 [==============================] - 4s 67us/step - loss: 0.2758 - acc: 0.9344
Epoch 5/11
60000/60000 [==============================] - 4s 67us/step - loss: 0.2835 - acc: 0.9364
Epoch 6/11
60000/60000 [==============================] - 4s 66us/step - loss: 0.1510 - acc: 0.9613
Epoch 7/11
60000/60000 [==============================] - 4s 66us/step - loss: 0.1043 - acc: 0.9714
Epoch 8/11
60000/60000 [==============================] - 4s 64us/step - loss: 0.0848 - acc: 0.9762
Epoch 9/11
60000/60000 [==============================] - 4s 65us/step - loss: 0.0773 - acc: 0.9788
Epoch 10/11
60000/60000 [==============================] - 4s 65us/step - loss: 0.0685 - acc: 0.9808
Epoch 11/11
60000/60000 [==============================] - 4s 67us/step - loss: 0.0636 - acc: 0.9817
```

Model 1: Train acc = 98.17%, val acc = 98.82%

```
train_loss = history.history['loss']
val_loss   = history.history['val_loss']
train_acc  = history.history['acc']
val_acc    = history.history['val_acc']

# Plot loss
fig = plt.figure()
```

```
plt.plot(train_loss, color='blue', linewidth=2)
plt.plot(val_loss, color='red', linewidth=2)
plt.title("Cross Entropy Loss vs Epoch")
plt.ylabel("Categorical Cross Entropy Loss")
plt.xlabel("Epoch")
plt.show()

# Plot accuracy
fig = plt.figure()
plt.plot(train_acc, color='blue', linewidth=2)
plt.plot(val_acc, color='red', linewidth=2)
plt.title("Accuracy vs Epoch")
plt.ylabel("Accuracy")
plt.xlabel("Epoch")
plt.show()
```




```
from keras.callbacks import ReduceLROnPlateau

reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.2, patience=3, min_lr=0.0000001)
# hyper_parameters used: batch_size, epochs
history2 = model2.fit(x_train
```

```
          y_train,
          validation_data=(x_test, y_test),
          callbacks=[reduce_lr],
          batch_size=batch_size,
          epochs=epochs)
```

```
Train on 60000 samples, validate on 10000 samples
Epoch 1/11
60000/60000 [==============================] - 11s 177us/step - loss: 0.6612 - acc: 0.79
Epoch 2/11
60000/60000 [==============================] - 10s 167us/step - loss: 0.2321 - acc: 0.94
Epoch 3/11
60000/60000 [==============================] - 10s 166us/step - loss: 0.1932 - acc: 0.95
Epoch 4/11
60000/60000 [==============================] - 10s 167us/step - loss: 0.1698 - acc: 0.96
Epoch 5/11
60000/60000 [==============================] - 10s 166us/step - loss: 0.1560 - acc: 0.96
Epoch 6/11
60000/60000 [==============================] - 10s 165us/step - loss: 0.1499 - acc: 0.96
Epoch 7/11
60000/60000 [==============================] - 10s 166us/step - loss: 0.1558 - acc: 0.96
Epoch 8/11
60000/60000 [==============================] - 10s 164us/step - loss: 0.1752 - acc: 0.96
Epoch 9/11
60000/60000 [==============================] - 10s 165us/step - loss: 0.1762 - acc: 0.96
Epoch 10/11
60000/60000 [==============================] - 10s 164us/step - loss: 0.0754 - acc: 0.98
Epoch 11/11
60000/60000 [==============================] - 10s 164us/step - loss: 0.0585 - acc: 0.98
```

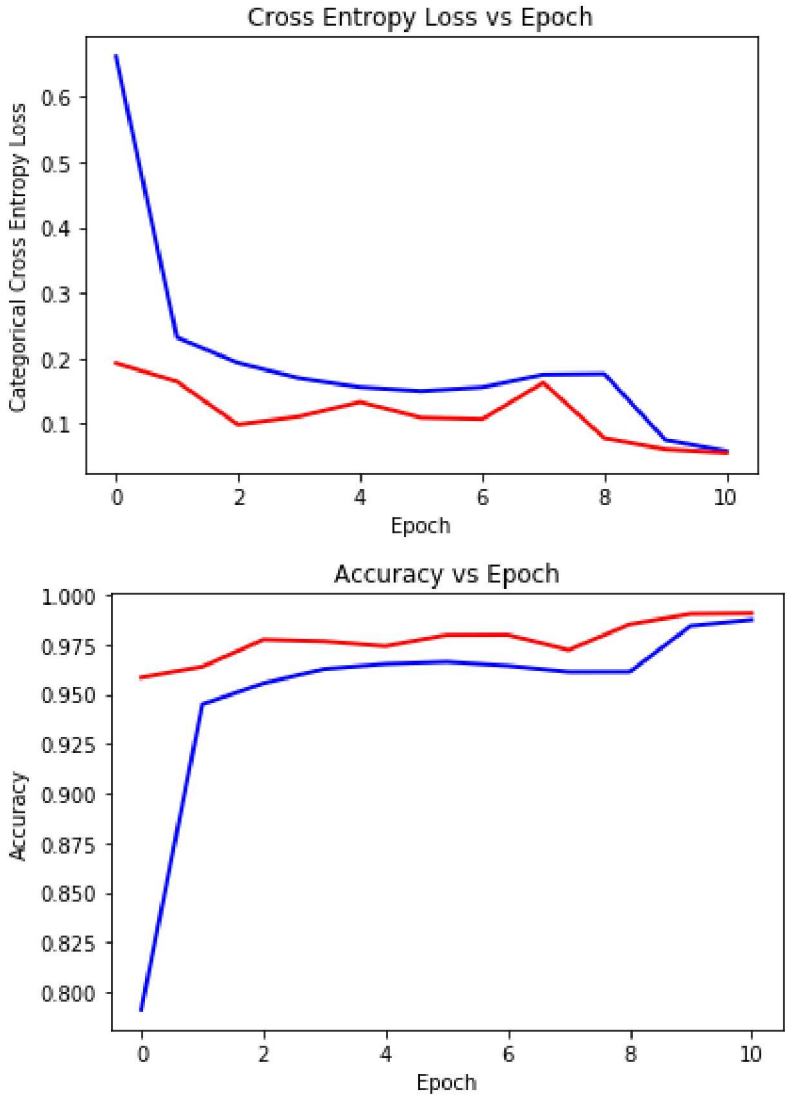
Model 2: Train acc = 98.75%, val acc = 99.08%

```
model2.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
zero_padding2d_7 (ZeroPaddin (None, 32, 32, 1)         0
_________________________________________________________________
conv2d_9 (Conv2D)            (None, 28, 28, 32)        832
_________________________________________________________________
leaky_re_lu_7 (LeakyReLU)    (None, 28, 28, 32)        0
_________________________________________________________________
zero_padding2d_8 (ZeroPaddin (None, 32, 32, 32)        0
_________________________________________________________________
conv2d_10 (Conv2D)           (None, 28, 28, 32)        25632
_________________________________________________________________
leaky_re_lu_8 (LeakyReLU)    (None, 28, 28, 32)        0
_________________________________________________________________
max_pooling2d_6 (MaxPooling2 (None, 14, 14, 32)        0
_________________________________________________________________
dropout_5 (Dropout)          (None, 14, 14, 32)        0
_________________________________________________________________
zero_padding2d_9 (ZeroPaddin (None, 18, 18, 32)        0
_________________________________________________________________
conv2d_11 (Conv2D)           (None, 14, 14, 64)        51264
_________________________________________________________________
leaky_re_lu_9 (LeakyReLU)    (None, 14, 14, 64)        0
_________________________________________________________________
zero_padding2d_10 (ZeroPaddi (None, 18, 18, 64)        0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 14, 14, 64)        102464
_________________________________________________________________
leaky_re_lu_10 (LeakyReLU)   (None, 14, 14, 64)        0
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 7, 7, 64)          0
_________________________________________________________________
dropout_6 (Dropout)          (None, 7, 7, 64)          0
_________________________________________________________________
zero_padding2d_11 (ZeroPaddi (None, 11, 11, 64)        0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 7, 7, 128)         204928
_________________________________________________________________
leaky_re_lu_11 (LeakyReLU)   (None, 7, 7, 128)         0
_________________________________________________________________
zero_padding2d_12 (ZeroPaddi (None, 11, 11, 128)       0
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 7, 7, 128)         409728
_________________________________________________________________
leaky_re_lu_12 (LeakyReLU)   (None, 7, 7, 128)         0
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 3, 3, 128)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 1152)              0
_________________________________________________________________
stage4 (Dense)               (None, 2)                 2306
_________________________________________________________________
dense_4 (Dense)              (None, 10)                30
=================================================================
Total params: 797,184
```

```
Trainable params: 797,184
Non-trainable params: 0
_________________________________________________________________
```

```
train_loss2 = history2.history['loss']
val_loss2   = history2.history['val_loss']
train_acc2  = history2.history['acc']
val_acc2    = history2.history['val_acc']

# Plot loss
fig = plt.figure()
plt.plot(train_loss2, color='blue', linewidth=2)
plt.plot(val_loss2, color='red', linewidth=2)
plt.title("Cross Entropy Loss vs Epoch")
plt.ylabel("Categorical Cross Entropy Loss")
plt.xlabel("Epoch")
plt.show()

# Plot accuracy
fig = plt.figure()
plt.plot(train_acc2, color='blue', linewidth=2)
plt.plot(val_acc2, color='red', linewidth=2)
plt.title("Accuracy vs Epoch")
plt.ylabel("Accuracy")
plt.xlabel("Epoch")
plt.show()
```

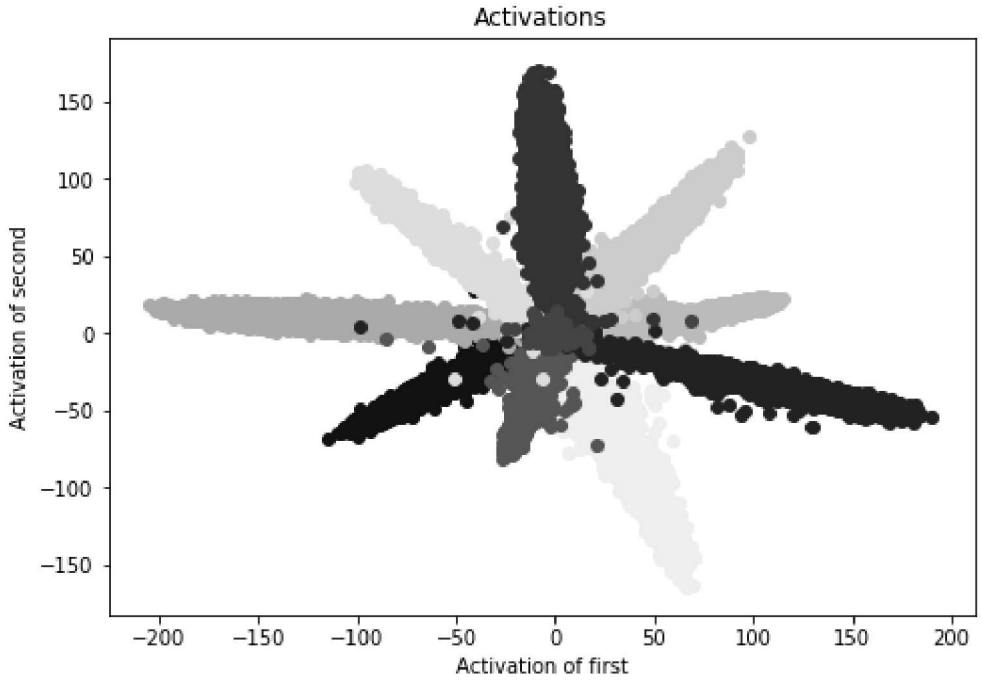



```
name_last_layer = str(model2.layers[-1])
model2_activation = Sequential()
for layer in model2.layers:
    if str(layer) != name_last_layer:
        model2_activation.add(layer)


pred_test = model2_activation.predict(x_test)
pred_train = model2_activation.predict(x_train)


# Plot accuracy
fig = plt.figure()
plt.plot(train_acc2, color='blue', linewidth=2)
plt.plot(val_acc2, color='red', linewidth=2)
plt.title("Accuracy vs Epoch")
plt.ylabel("Accuracy")
plt.xlabel("Epoch")
plt.show()
```

```
plt.show()
```

```
(28, 28, 1)
```

```
activations = {}

for i in range(len(y_train_num)):
  res = y_train_num[i]
  output = pred_train[i]

  if res not in activations:
    activations[res] = [[],[]]

  activations[res][0].append(output[0])
  activations[res][1].append(output[1])


for i in range(len(y_test_num)):
  res = y_test_num[i]
  output = pred_test[i]

  if res not in activations:
    activations[res] = []
  activations[res][0].append(output[0])
  activations[res][1].append(output[1])


#girls_grades = [89, 90, 70, 89, 100, 80, 90, 100, 80, 34]
#boys_grades = [30, 29, 49, 48, 100, 48, 38, 45, 20, 30]
#grades_range = [10, 20, 30, 40, 50, 60, 70, 80, 90, 100]
colors = ['#aaaaaa','#bbbbbb','#cccccc','#dddddd','#eeeeee','#111111','#222222','#333333','#4
fig=plt.figure()
ax=fig.add_axes([0,0,1,1])

for val in activations:
  output = activations[val]
  first = output[0]
  second = output[1]
  ax.scatter(first, second, color=colors[val])

ax.set_xlabel('Activation of first')
ax.set_ylabel('Activation of second')
ax.set_title('Activations')
plt.show()
```




```
print(activations[0][1])
```

```
[13.764669, 13.549252, 7.602485, 14.78693, 7.690344, 13.0722685, 7.018775, 11.264408, 6.
```

```
from __future__ import print_function
import numpy as np
from keras.callbacks import EarlyStopping
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D
from keras.optimizers import Adam
from keras.layers.pooling import MaxPooling2D
from keras.utils import to_categorical
from keras.callbacks import ReduceLROnPlateau
from keras import regularizers
import matplotlib.pyplot as plt


# Load the dataset
(X_train, Y_train), (X_test, Y_test) = cifar10.load_data()

X_train = X_train.astype('float32')
X_test  = X_test.astype('float32')
X_train = X_train / 255.0
X_test  = X_test / 255.0
print(Y_train)
Y_train = to_categorical(Y_train)
Y_test  = to_categorical(Y_test)
```

```
[[6]
 [9]
 [9]
 ...
 [9]
 [1]
 [1]]


# Create the model
model = Sequential()
model.add(Conv2D(32, kernel_size=(3, 3), activation='relu', padding='same', input_shape=(32,
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Conv2D(64, kernel_size=(3, 3), activation='relu', padding='same', kernel_regularize
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))


model.add(Conv2D(128, kernel_size=(3, 3), activation='relu', padding='same', kernel_regulariz
model.add(Conv2D(256, kernel_size=(3, 3), activation='relu', padding='same', kernel_regulariz
model.add(Conv2D(512, kernel_size=(3, 3), activation='relu', padding='same', kernel_regulariz

model.add(Dropout(0.5))
```

```
model.add(MaxPooling2D(pool_size=(2, 2)))
model.add(Dropout(0.5))

model.add(Flatten())
model.add(Dense(512, activation='relu'))
model.add(Dense(256, activation='relu'))

model.add(Dense(10, activation='softmax'))



# Hyper parameters
batch_size = 128
num_epochs = 100


# Compile the model
optimizer = Adam(lr=0.0001, decay=1e-6)
model.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])



reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.2, patience=3, min_lr=0.0000001)
#early_stopping = EarlyStopping(min_delta=0.001, patience=3)
# Train the model
history = model.fit(X_train, Y_train,
                    batch_size=batch_size,
                    shuffle=True,
                    epochs=num_epochs,
                    validation_data=(X_test, Y_test),
                    callbacks=[reduce_lr])
```

```
Train on 50000 samples, validate on 10000 samples
Epoch 1/100
50000/50000 [==============================] - 11s 221us/step - loss: 4.4807 - acc: 0.26
Epoch 2/100
50000/50000 [==============================] - 10s 203us/step - loss: 2.3630 - acc: 0.39
Epoch 3/100
50000/50000 [==============================] - 10s 202us/step - loss: 2.0287 - acc: 0.42
Epoch 4/100
50000/50000 [==============================] - 10s 202us/step - loss: 1.8847 - acc: 0.45
Epoch 5/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.7986 - acc: 0.46
Epoch 6/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.7199 - acc: 0.48
Epoch 7/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.6603 - acc: 0.50
Epoch 8/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.6073 - acc: 0.51
Epoch 9/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.5684 - acc: 0.52
Epoch 10/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.5326 - acc: 0.53
Epoch 11/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.5018 - acc: 0.54
Epoch 12/100
50000/50000 [==============================] - 10s 202us/step - loss: 1.4777 - acc: 0.55
Epoch 13/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.4526 - acc: 0.56
Epoch 14/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.4326 - acc: 0.57
Epoch 15/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.4130 - acc: 0.57
Epoch 16/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.3965 - acc: 0.58
Epoch 17/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.3778 - acc: 0.58
Epoch 18/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.3563 - acc: 0.59
Epoch 19/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.3461 - acc: 0.59
Epoch 20/100
50000/50000 [==============================] - 10s 201us/step - loss: 1.3316 - acc: 0.60
Epoch 21/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.3163 - acc: 0.60
Epoch 22/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.2993 - acc: 0.61
Epoch 23/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.2856 - acc: 0.62
Epoch 24/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.2748 - acc: 0.62
Epoch 25/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.2648 - acc: 0.62
Epoch 26/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.2523 - acc: 0.62
Epoch 27/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.2375 - acc: 0.63
Epoch 28/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.2268 - acc: 0.63
```

```
Epoch 29/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.2146 - acc: 0.64
Epoch 30/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.2070 - acc: 0.64
Epoch 31/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.1891 - acc: 0.65
Epoch 32/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.1834 - acc: 0.65
Epoch 33/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.1724 - acc: 0.65
Epoch 34/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.1592 - acc: 0.66
Epoch 35/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.1542 - acc: 0.66
Epoch 36/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.1429 - acc: 0.66
Epoch 37/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.1263 - acc: 0.67
Epoch 38/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.1188 - acc: 0.67
Epoch 39/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.1125 - acc: 0.68
Epoch 40/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.0987 - acc: 0.68
Epoch 41/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0946 - acc: 0.68
Epoch 42/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0830 - acc: 0.69
Epoch 43/100
50000/50000 [==============================] - 10s 199us/step - loss: 1.0791 - acc: 0.69
Epoch 44/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.0688 - acc: 0.69
Epoch 45/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.0632 - acc: 0.69
Epoch 46/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0558 - acc: 0.69
Epoch 47/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0450 - acc: 0.70
Epoch 48/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0359 - acc: 0.70
Epoch 49/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0328 - acc: 0.70
Epoch 50/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.0234 - acc: 0.71
Epoch 51/100
50000/50000 [==============================] - 10s 200us/step - loss: 1.0230 - acc: 0.71
Epoch 52/100
50000/50000 [==============================] - 10s 198us/step - loss: 1.0164 - acc: 0.71
Epoch 53/100
50000/50000 [==============================] - 10s 197us/step - loss: 1.0039 - acc: 0.71
Epoch 54/100
50000/50000 [==============================] - 10s 197us/step - loss: 0.9977 - acc: 0.71
Epoch 55/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9914 - acc: 0.72
Epoch 56/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9817 - acc: 0.72
Epoch 57/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9817 - acc: 0.72
```

```
Epoch 58/100
50000/50000 [==============================] - 10s 197us/step - loss: 0.9791 - acc: 0.72
Epoch 59/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9715 - acc: 0.73
Epoch 60/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9660 - acc: 0.72
Epoch 61/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9589 - acc: 0.73
Epoch 62/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9574 - acc: 0.73
Epoch 63/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9497 - acc: 0.73
Epoch 64/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9497 - acc: 0.73
Epoch 65/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9359 - acc: 0.74
Epoch 66/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9371 - acc: 0.73
Epoch 67/100
50000/50000 [==============================] - 10s 200us/step - loss: 0.9319 - acc: 0.74
Epoch 68/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9278 - acc: 0.74
Epoch 69/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9220 - acc: 0.74
Epoch 70/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9192 - acc: 0.74
Epoch 71/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.9087 - acc: 0.75
Epoch 72/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9070 - acc: 0.75
Epoch 73/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9009 - acc: 0.75
Epoch 74/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.9009 - acc: 0.75
Epoch 75/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8940 - acc: 0.75
Epoch 76/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8912 - acc: 0.75
Epoch 77/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8807 - acc: 0.75
Epoch 78/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8763 - acc: 0.76
Epoch 79/100
50000/50000 [==============================] - 10s 200us/step - loss: 0.8791 - acc: 0.75
Epoch 80/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8675 - acc: 0.76
Epoch 81/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8700 - acc: 0.76
Epoch 82/100
50000/50000 [==============================] - 10s 200us/step - loss: 0.8653 - acc: 0.76
Epoch 83/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8585 - acc: 0.76
Epoch 84/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8541 - acc: 0.76
Epoch 85/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8522 - acc: 0.76
Epoch 86/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8454 - acc: 0.77
```

```
50000/50000 [==============================] - 10s 198us/step - loss: 0.8454 - acc: 0.77
Epoch 87/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8404 - acc: 0.77
Epoch 88/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8418 - acc: 0.77
Epoch 89/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8343 - acc: 0.77
Epoch 90/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8345 - acc: 0.77
Epoch 91/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8301 - acc: 0.77
Epoch 92/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8267 - acc: 0.77
Epoch 93/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8249 - acc: 0.77
Epoch 94/100
50000/50000 [==============================] - 10s 200us/step - loss: 0.8170 - acc: 0.78
Epoch 95/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8097 - acc: 0.78
Epoch 96/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8105 - acc: 0.78
Epoch 97/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8069 - acc: 0.78
Epoch 98/100
50000/50000 [==============================] - 10s 199us/step - loss: 0.8058 - acc: 0.78
Epoch 99/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.8014 - acc: 0.78
Epoch 100/100
50000/50000 [==============================] - 10s 198us/step - loss: 0.7955 - acc: 0.78
```

```
# Evaluate the model
scores = model.evaluate(X_test, Y_test)
print('Loss: %.3f' % scores[0])
print('Accuracy: %.3f' % scores[1])
```

```
10000/10000 [==============================] - 1s 124us/step
Loss: 0.932
Accuracy: 0.754
```

```
train_loss = history.history['loss']
val_loss   = history.history['val_loss']
train_acc  = history.history['acc']
val_acc    = history.history['val_acc']

# Plot loss
fig = plt.figure()
plt.plot(train_loss, color='blue', linewidth=2, label='train loss')
plt.plot(val_loss, color='red', linewidth=2, label='val acc')
plt.title("Cross Entropy Loss vs Epoch")
plt.ylabel("Categorical Cross Entropy Loss")
plt.xlabel("Epoch")
```

```
plt.legend()
plt.show()

# Plot accuracy
fig = plt.figure()
plt.plot(train_acc, color='blue', linewidth=2, label='train acc')
plt.plot(val_acc, color='red', linewidth=2, label='val acc')
plt.title("Accuracy vs Epoch")
plt.ylabel("Accuracy")
plt.xlabel("Epoch")
plt.legend()
plt.show()
```

```
model.save("q2.h5")

model.summary()
```

```
from __future__ import print_function
import numpy as np
from keras.callbacks import EarlyStopping
from keras.datasets import cifar10
from keras.models import Sequential
from keras.layers.core import Dense, Dropout, Flatten
from keras.layers.convolutional import Conv2D
from keras.optimizers import Adam
from keras.layers.pooling import MaxPooling2D
from keras.utils import to_categorical
from keras.callbacks import ReduceLROnPlateau
from keras import regularizers
import matplotlib.pyplot as plt
from keras.models import load_model
import numpy as np
from keras.preprocessing.image import ImageDataGenerator



import zipfile
with zipfile.ZipFile("face.zip","r") as zip_ref:
    zip_ref.extractall("faces")


import os
from PIL import Image
import cv2

names = []

x_train = []
y_train = []

for name in os.listdir('./faces/downsized/test'):
  names.append(name)
names.sort()
i = 0
for name in os.listdir('./faces/downsized/test'):
  for im in os.listdir('./faces/downsized/test/' + name):
    im = Image.open('./faces/downsized/test/' + name + '/' + im)
    x = np.array(im)

    x_train.append(x)
    y_train.append(names.index(name))

x_test = []
y_test = []

for name in os.listdir('./faces/downsized/train'):
  for im in os.listdir('./faces/downsized/train/' + name):
    im = Image.open('./faces/downsized/train/' + name + '/' + im)
```

```
        x = np.array(im)
        x_test.append(x)
        y_test.append(names.index(name))

x_train = np.array(x_train)
x_test = np.array(x_test)

x_train = x_train.astype('float32')
x_test = x_test.astype('float32')
x_train /= 255
x_test /= 255


y_train = to_categorical(np.expand_dims(np.array(y_train),axis=1))
y_test = to_categorical(np.expand_dims(np.array(y_test),axis=1))




model = load_model('./q2.h5')
#model.save_weights('./q2_weights.h5')
```

Double-click (or enter) to edit

```
name_last_layer = str(model.layers[-1])
model2 = Sequential()
for layer in model.layers:
    if str(layer) != name_last_layer:
        model2.add(layer)

model2.add(Dense(100, activation='softmax', name='dense_3_transfer'))


model2.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_11 (Conv2D)           (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_7 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 8, 8, 128)         73856
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 8, 8, 256)         295168
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 8, 8, 512)         1180160
_________________________________________________________________
dropout_8 (Dropout)          (None, 8, 8, 512)         0
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 4, 4, 512)         0
_________________________________________________________________
dropout_9 (Dropout)          (None, 4, 4, 512)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_7 (Dense)              (None, 512)               4194816
_________________________________________________________________
dense_8 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_3_transfer (Dense)     (None, 100)               25700
=================================================================
Total params: 5,920,420
Trainable params: 5,920,420
Non-trainable params: 0
_________________________________________________________________
```

```
# retrain but only with softmax dense layer

for layer in model2.layers:
  if layer.name != 'dense_3_transfer':
    layer.trainable=False
  else:
    layer.trainable = True

#model2.load_weights('q2_weights.h5', by_name=True)


model2.summary()
```

```
Model: "sequential_3"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_11 (Conv2D)           (None, 32, 32, 32)        896
_________________________________________________________________
max_pooling2d_7 (MaxPooling2 (None, 16, 16, 32)        0
_________________________________________________________________
conv2d_12 (Conv2D)           (None, 16, 16, 64)        18496
_________________________________________________________________
max_pooling2d_8 (MaxPooling2 (None, 8, 8, 64)          0
_________________________________________________________________
dropout_7 (Dropout)          (None, 8, 8, 64)          0
_________________________________________________________________
conv2d_13 (Conv2D)           (None, 8, 8, 128)         73856
_________________________________________________________________
conv2d_14 (Conv2D)           (None, 8, 8, 256)         295168
_________________________________________________________________
conv2d_15 (Conv2D)           (None, 8, 8, 512)         1180160
_________________________________________________________________
dropout_8 (Dropout)          (None, 8, 8, 512)         0
_________________________________________________________________
max_pooling2d_9 (MaxPooling2 (None, 4, 4, 512)         0
_________________________________________________________________
dropout_9 (Dropout)          (None, 4, 4, 512)         0
_________________________________________________________________
flatten_3 (Flatten)          (None, 8192)              0
_________________________________________________________________
dense_7 (Dense)              (None, 512)               4194816
_________________________________________________________________
dense_8 (Dense)              (None, 256)               131328
_________________________________________________________________
dense_3_transfer (Dense)     (None, 100)               25700
=================================================================
Total params: 5,920,420
Trainable params: 25,700
Non-trainable params: 5,894,720
_________________________________________________________________
```

```
# Hyper parameters
batch_size = len(x_train)
num_epochs = 500


# Compile the model
optimizer = Adam(lr=0.01, decay=1e-6)

model2.compile(loss='categorical_crossentropy', optimizer=optimizer, metrics=['accuracy'])


from keras.preprocessing.image import ImageDataGenerator

reduce_lr = ReduceLROnPlateau(monitor='loss', factor=0.2, patience=100, min_lr=0.0000001)
#early_stopping = EarlyStopping(min_delta=0.001, patience=3)
```

```
# Train the model
history = model2.fit(x_train, y_train,
                     batch_size=batch_size,
                     shuffle=True,
                     epochs=num_epochs,
                     validation_data=(x_test, y_test),
                     callbacks=[reduce_lr])
```

```
Train on 2000 samples, validate on 1000 samples
Epoch 1/500
2000/2000 [==============================] - 1s 291us/step - loss: 4.9514 - acc: 0.0095
Epoch 2/500
2000/2000 [==============================] - 0s 42us/step - loss: 4.8299 - acc: 0.0170 -
Epoch 3/500
2000/2000 [==============================] - 0s 43us/step - loss: 4.7995 - acc: 0.0145 -
Epoch 4/500
2000/2000 [==============================] - 0s 41us/step - loss: 4.7731 - acc: 0.0200 -
Epoch 5/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.7159 - acc: 0.0310 -
Epoch 6/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.6582 - acc: 0.0420 -
Epoch 7/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.6106 - acc: 0.0575 -
Epoch 8/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.5651 - acc: 0.0565 -
Epoch 9/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.5311 - acc: 0.0580 -
Epoch 10/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.4847 - acc: 0.0585 -
Epoch 11/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.4483 - acc: 0.0735 -
Epoch 12/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.4092 - acc: 0.0880 -
Epoch 13/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.3830 - acc: 0.1005 -
Epoch 14/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.3539 - acc: 0.1000 -
Epoch 15/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.3222 - acc: 0.1030 -
Epoch 16/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.2858 - acc: 0.1125 -
Epoch 17/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.2562 - acc: 0.1175 -
Epoch 18/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.2324 - acc: 0.1100 -
Epoch 19/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.2187 - acc: 0.1130 -
Epoch 20/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.1988 - acc: 0.1280 -
Epoch 21/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.1586 - acc: 0.1300 -
Epoch 22/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.1588 - acc: 0.1275 -
Epoch 23/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.1395 - acc: 0.1335 -
Epoch 24/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.1145 - acc: 0.1245 -
Epoch 25/500
2000/2000 [==============================] - 0s 39us/step - loss: 4.0910 - acc: 0.1250 -
Epoch 26/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.0888 - acc: 0.1365 -
Epoch 27/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.0538 - acc: 0.1370 -
Epoch 28/500
2000/2000 [==============================] - 0s 43us/step - loss: 4.0551 - acc: 0.1450 -
```

```
Epoch 29/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.0254 - acc: 0.1405 -
Epoch 30/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.0080 - acc: 0.1415 -
Epoch 31/500
2000/2000 [==============================] - 0s 40us/step - loss: 4.0000 - acc: 0.1380 -
Epoch 32/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.9786 - acc: 0.1510 -
Epoch 33/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.9727 - acc: 0.1540 -
Epoch 34/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.9525 - acc: 0.1520 -
Epoch 35/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.9366 - acc: 0.1540 -
Epoch 36/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.9160 - acc: 0.1660 -
Epoch 37/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.9127 - acc: 0.1650 -
Epoch 38/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.9141 - acc: 0.1650 -
Epoch 39/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.8859 - acc: 0.1775 -
Epoch 40/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.8746 - acc: 0.1600 -
Epoch 41/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.8643 - acc: 0.1610 -
Epoch 42/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.8604 - acc: 0.1605 -
Epoch 43/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.8403 - acc: 0.1655 -
Epoch 44/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.8378 - acc: 0.1740 -
Epoch 45/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.8266 - acc: 0.1650 -
Epoch 46/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.8119 - acc: 0.1630 -
Epoch 47/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.8218 - acc: 0.1660 -
Epoch 48/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.7618 - acc: 0.1860 -
Epoch 49/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.7746 - acc: 0.1830 -
Epoch 50/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7911 - acc: 0.1755 -
Epoch 51/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7637 - acc: 0.1865 -
Epoch 52/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.7543 - acc: 0.1895 -
Epoch 53/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7514 - acc: 0.1865 -
Epoch 54/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7553 - acc: 0.1805 -
Epoch 55/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.7217 - acc: 0.1990 -
Epoch 56/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.7183 - acc: 0.1925 -
Epoch 57/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7071 - acc: 0.1995 -
```

```
Epoch 58/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.7083 - acc: 0.1780 -
Epoch 59/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6893 - acc: 0.2045 -
Epoch 60/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.7041 - acc: 0.2075 -
Epoch 61/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.6750 - acc: 0.1880 -
Epoch 62/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.6726 - acc: 0.1920 -
Epoch 63/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6571 - acc: 0.2015 -
Epoch 64/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.6541 - acc: 0.2115 -
Epoch 65/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6457 - acc: 0.2050 -
Epoch 66/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6334 - acc: 0.1930 -
Epoch 67/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.6179 - acc: 0.2205 -
Epoch 68/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6274 - acc: 0.2150 -
Epoch 69/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.6332 - acc: 0.1955 -
Epoch 70/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.6263 - acc: 0.2105 -
Epoch 71/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5887 - acc: 0.2220 -
Epoch 72/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5979 - acc: 0.2180 -
Epoch 73/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.5780 - acc: 0.2075 -
Epoch 74/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5972 - acc: 0.2265 -
Epoch 75/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.5804 - acc: 0.2170 -
Epoch 76/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.5677 - acc: 0.2210 -
Epoch 77/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5678 - acc: 0.2155 -
Epoch 78/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5741 - acc: 0.2245 -
Epoch 79/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5537 - acc: 0.2180 -
Epoch 80/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5558 - acc: 0.2220 -
Epoch 81/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5671 - acc: 0.2190 -
Epoch 82/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5220 - acc: 0.2295 -
Epoch 83/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5459 - acc: 0.2185 -
Epoch 84/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5295 - acc: 0.2285 -
Epoch 85/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5118 - acc: 0.2275 -
Epoch 86/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.5225 - acc: 0.2315
```

```
2000/2000 [==============================] - 0s 39us/step - loss: 3.5225 - acc: 0.2315 -
Epoch 87/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.5204 - acc: 0.2210 -
Epoch 88/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.5048 - acc: 0.2360 -
Epoch 89/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.5194 - acc: 0.2265 -
Epoch 90/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4991 - acc: 0.2245 -
Epoch 91/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4855 - acc: 0.2390 -
Epoch 92/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4913 - acc: 0.2415 -
Epoch 93/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4790 - acc: 0.2340 -
Epoch 94/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4912 - acc: 0.2260 -
Epoch 95/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4644 - acc: 0.2390 -
Epoch 96/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4621 - acc: 0.2465 -
Epoch 97/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4366 - acc: 0.2435 -
Epoch 98/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4662 - acc: 0.2375 -
Epoch 99/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4693 - acc: 0.2465 -
Epoch 100/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.4514 - acc: 0.2360 -
Epoch 101/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4590 - acc: 0.2465 -
Epoch 102/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4468 - acc: 0.2425 -
Epoch 103/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4244 - acc: 0.2490 -
Epoch 104/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4359 - acc: 0.2380 -
Epoch 105/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4406 - acc: 0.2440 -
Epoch 106/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4197 - acc: 0.2465 -
Epoch 107/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.4153 - acc: 0.2475 -
Epoch 108/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4090 - acc: 0.2550 -
Epoch 109/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3963 - acc: 0.2435 -
Epoch 110/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4168 - acc: 0.2385 -
Epoch 111/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.4068 - acc: 0.2485 -
Epoch 112/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3791 - acc: 0.2540 -
Epoch 113/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.3776 - acc: 0.2565 -
Epoch 114/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3819 - acc: 0.2555 -
Epoch 115/500
```

```
2000/2000 [==============================] - 0s 39us/step - loss: 3.3613 - acc: 0.2455 -
Epoch 116/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3821 - acc: 0.2640 -
Epoch 117/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.3664 - acc: 0.2590 -
Epoch 118/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3685 - acc: 0.2560 -
Epoch 119/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3609 - acc: 0.2590 -
Epoch 120/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3610 - acc: 0.2500 -
Epoch 121/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.3804 - acc: 0.2475 -
Epoch 122/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3501 - acc: 0.2650 -
Epoch 123/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3563 - acc: 0.2660 -
Epoch 124/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3487 - acc: 0.2455 -
Epoch 125/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3426 - acc: 0.2620 -
Epoch 126/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.3243 - acc: 0.2665 -
Epoch 127/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3312 - acc: 0.2590 -
Epoch 128/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3137 - acc: 0.2700 -
Epoch 129/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3121 - acc: 0.2740 -
Epoch 130/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3356 - acc: 0.2665 -
Epoch 131/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3242 - acc: 0.2675 -
Epoch 132/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3122 - acc: 0.2640 -
Epoch 133/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.3103 - acc: 0.2685 -
Epoch 134/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3085 - acc: 0.2770 -
Epoch 135/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3167 - acc: 0.2565 -
Epoch 136/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.3023 - acc: 0.2720 -
Epoch 137/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2891 - acc: 0.2650 -
Epoch 138/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2961 - acc: 0.2665 -
Epoch 139/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2996 - acc: 0.2795 -
Epoch 140/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2885 - acc: 0.2710 -
Epoch 141/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2868 - acc: 0.2690 -
Epoch 142/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2685 - acc: 0.2700 -
Epoch 143/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2519 - acc: 0.2830 -
Epoch 144/500
```

```
2000/2000 [==============================] - 0s 39us/step - loss: 3.2831 - acc: 0.2765 -
Epoch 145/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2688 - acc: 0.2790 -
Epoch 146/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2821 - acc: 0.2740 -
Epoch 147/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.2693 - acc: 0.2745 -
Epoch 148/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2414 - acc: 0.2795 -
Epoch 149/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2610 - acc: 0.2760 -
Epoch 150/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2495 - acc: 0.2790 -
Epoch 151/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2461 - acc: 0.2895 -
Epoch 152/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2404 - acc: 0.2835 -
Epoch 153/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2335 - acc: 0.2865 -
Epoch 154/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.2394 - acc: 0.2920 -
Epoch 155/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2559 - acc: 0.2710 -
Epoch 156/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2485 - acc: 0.2895 -
Epoch 157/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2186 - acc: 0.2835 -
Epoch 158/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2118 - acc: 0.2915 -
Epoch 159/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2072 - acc: 0.2830 -
Epoch 160/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2246 - acc: 0.2850 -
Epoch 161/500
2000/2000 [==============================] - 0s 43us/step - loss: 3.1793 - acc: 0.2985 -
Epoch 162/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2176 - acc: 0.2735 -
Epoch 163/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2263 - acc: 0.2745 -
Epoch 164/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2092 - acc: 0.2760 -
Epoch 165/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1966 - acc: 0.2905 -
Epoch 166/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2014 - acc: 0.2985 -
Epoch 167/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1921 - acc: 0.2885 -
Epoch 168/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.2042 - acc: 0.2950 -
Epoch 169/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1827 - acc: 0.2870 -
Epoch 170/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1805 - acc: 0.2895 -
Epoch 171/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1897 - acc: 0.2895 -
Epoch 172/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1820 - acc: 0.3000 -
Epoch 173/500
```

```
Epoch 173/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1734 - acc: 0.2975 -
Epoch 174/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.2015 - acc: 0.2840 -
Epoch 175/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1636 - acc: 0.2850 -
Epoch 176/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1894 - acc: 0.2840 -
Epoch 177/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.1587 - acc: 0.2945 -
Epoch 178/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1587 - acc: 0.2805 -
Epoch 179/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1703 - acc: 0.2995 -
Epoch 180/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1574 - acc: 0.3035 -
Epoch 181/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1614 - acc: 0.2925 -
Epoch 182/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1527 - acc: 0.3060 -
Epoch 183/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1332 - acc: 0.3045 -
Epoch 184/500
2000/2000 [==============================] - 0s 43us/step - loss: 3.1433 - acc: 0.3030 -
Epoch 185/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.1498 - acc: 0.2990 -
Epoch 186/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1492 - acc: 0.2940 -
Epoch 187/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1472 - acc: 0.3050 -
Epoch 188/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1436 - acc: 0.3035 -
Epoch 189/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1303 - acc: 0.2975 -
Epoch 190/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1438 - acc: 0.3010 -
Epoch 191/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1378 - acc: 0.3060 -
Epoch 192/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1259 - acc: 0.3030 -
Epoch 193/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1269 - acc: 0.3080 -
Epoch 194/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1248 - acc: 0.3160 -
Epoch 195/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1328 - acc: 0.3085 -
Epoch 196/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1170 - acc: 0.2975 -
Epoch 197/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1094 - acc: 0.3125 -
Epoch 198/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1066 - acc: 0.3000 -
Epoch 199/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1166 - acc: 0.3035 -
Epoch 200/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1023 - acc: 0.3035 -
Epoch 201/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0619 - acc: 0.3240 -
```

```
Epoch 202/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0802 - acc: 0.3150 -
Epoch 203/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0990 - acc: 0.2980 -
Epoch 204/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0737 - acc: 0.3220 -
Epoch 205/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1036 - acc: 0.3095 -
Epoch 206/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.1012 - acc: 0.2980 -
Epoch 207/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0850 - acc: 0.3100 -
Epoch 208/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0927 - acc: 0.2945 -
Epoch 209/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0872 - acc: 0.3110 -
Epoch 210/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0833 - acc: 0.3105 -
Epoch 211/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.1089 - acc: 0.3100 -
Epoch 212/500
2000/2000 [==============================] - 0s 42us/step - loss: 3.0953 - acc: 0.3060 -
Epoch 213/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0717 - acc: 0.3035 -
Epoch 214/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0727 - acc: 0.3205 -
Epoch 215/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0777 - acc: 0.3050 -
Epoch 216/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0792 - acc: 0.3090 -
Epoch 217/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0621 - acc: 0.3195 -
Epoch 218/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0879 - acc: 0.3135 -
Epoch 219/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0490 - acc: 0.3190 -
Epoch 220/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0872 - acc: 0.3140 -
Epoch 221/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0561 - acc: 0.3155 -
Epoch 222/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0630 - acc: 0.3100 -
Epoch 223/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0557 - acc: 0.3085 -
Epoch 224/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0568 - acc: 0.3260 -
Epoch 225/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0494 - acc: 0.3210 -
Epoch 226/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0457 - acc: 0.3255 -
Epoch 227/500
2000/2000 [==============================] - 0s 43us/step - loss: 3.0457 - acc: 0.3215 -
Epoch 228/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0330 - acc: 0.3260 -
Epoch 229/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0526 - acc: 0.3275 -
Epoch 230/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0423 - acc: 0.3185 -
```

```
Epoch 231/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0457 - acc: 0.3240 -
Epoch 232/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0412 - acc: 0.3135 -
Epoch 233/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0304 - acc: 0.3175 -
Epoch 234/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0293 - acc: 0.3320 -
Epoch 235/500
2000/2000 [==============================] - 0s 41us/step - loss: 3.0287 - acc: 0.3220 -
Epoch 236/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0327 - acc: 0.3245 -
Epoch 237/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0271 - acc: 0.3200 -
Epoch 238/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0216 - acc: 0.3255 -
Epoch 239/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0224 - acc: 0.3250 -
Epoch 240/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0134 - acc: 0.3215 -
Epoch 241/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0596 - acc: 0.3100 -
Epoch 242/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0014 - acc: 0.3175 -
Epoch 243/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0278 - acc: 0.3280 -
Epoch 244/500
2000/2000 [==============================] - 0s 40us/step - loss: 3.0036 - acc: 0.3300 -
Epoch 245/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9984 - acc: 0.3370 -
Epoch 246/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9869 - acc: 0.3340 -
Epoch 247/500
2000/2000 [==============================] - 0s 40us/step - loss: 2.9892 - acc: 0.3315 -
Epoch 248/500
2000/2000 [==============================] - 0s 41us/step - loss: 2.9685 - acc: 0.3400 -
Epoch 249/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0032 - acc: 0.3295 -
Epoch 250/500
2000/2000 [==============================] - 0s 39us/step - loss: 3.0104 - acc: 0.3325 -
Epoch 251/500
2000/2000 [==============================] - 0s 40us/step - loss: 2.9903 - acc: 0.3360 -
Epoch 252/500
2000/2000 [==============================] - 0s 40us/step - loss: 2.9722 - acc: 0.3555 -
Epoch 253/500
2000/2000 [==============================] - 0s 41us/step - loss: 2.9993 - acc: 0.3220 -
Epoch 254/500
2000/2000 [==============================] - 0s 40us/step - loss: 2.9876 - acc: 0.3380 -
Epoch 255/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9687 - acc: 0.3430 -
Epoch 256/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9757 - acc: 0.3360 -
Epoch 257/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9682 - acc: 0.3325 -
Epoch 258/500
2000/2000 [==============================] - 0s 39us/step - loss: 2.9797 - acc: 0.3285 -
Epoch 259/500
```