Assignment_6.2b

January 15, 2021

0.1 File information

File: Assignment_6.2b.ipynb

Name: Amie Davis

Date: 1/14/2021

Course: DSC650 - Big Data

Assignment Number: 6.2b

Purpose: Create a ConvNet model that classifies images from the Keras CIFAR10 small images classification dataset, include dropout and data augmentation

1 Train the convnet on CIFAR10 small images

1.1 This file contains code from Deep Learning with Python

www.manning.com/books/deep-learning-with-python

Copyright 2018 Francois Chollet

1.2 Data Source: The CIFAR10 dataset - comes packaged with Keras.

This is a dataset of 50,000 32x32 color training images and 10,000 test images, labeled over 10 categories.

1.2.1 References: https://keras.io/api/preprocessing/image/#imagedatagenerator-class

[1]:
```python
# Import required packages
import keras

from keras import layers
from keras import models
from keras.datasets import cifar10
from keras.utils import to_categorical

import matplotlib.pyplot as plt

import os
```

```
from pathlib import Path
```

[2]:
```
# Set results directory for writing
import os

current_dir = Path(os.getcwd()).absolute()
results_dir = current_dir.joinpath('results')
results_dir.mkdir(parents=True, exist_ok=True)

model_path = results_dir.joinpath('6.2b_model.h5')
output_path = results_dir.joinpath('6.2b_output.txt')
image1_path = results_dir.joinpath('6.2b_image1.png')
image2_path = results_dir.joinpath('6.2b_image2.png')
```

1.3 Build Convolutional Neural Network Model

[3]:
```
# Instantiate ConvNet
model = models.Sequential()

# Build ConvNet
# Stack of Conv2D and MaxPooling2D layers
# Input shape is height x width x channel
model.add(layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(64, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
model.add(layers.Conv2D(128, (3, 3), activation='relu'))
model.add(layers.MaxPooling2D((2, 2)))
```

[4]:
```
# Show layer details
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 128)         0
=================================================================
Total params: 93,248
Trainable params: 93,248
Non-trainable params: 0
_________________________________________________________________
```

[5]:
```python
# Last output is of shape (2, 2, 128)
# Flatten 3D output to 1D
model.add(layers.Flatten())

# Add dropout layer
# Randomly sets input units to 0 to help prevent overfitting
# Rate = 0.2 drops 20% of the units
#model.add(layers.Dropout(0.2))

# Add densely-connected classifier network
# Final output has 10 classifications
# Use softmax activation fxn since multi-classification problem
model.add(layers.Dense(128, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))
```

[6]:
```python
# Show layer details
model.summary()
```

```
Model: "sequential"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d (Conv2D)              (None, 30, 30, 32)        896
_________________________________________________________________
max_pooling2d (MaxPooling2D) (None, 15, 15, 32)        0
_________________________________________________________________
conv2d_1 (Conv2D)            (None, 13, 13, 64)        18496
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 6, 6, 64)          0
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 4, 4, 128)         73856
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 128)         0
_________________________________________________________________
flatten (Flatten)            (None, 512)               0
_________________________________________________________________
dense (Dense)                (None, 128)               65664
_________________________________________________________________
dense_1 (Dense)              (None, 10)                1290
=================================================================
Total params: 160,202
Trainable params: 160,202
```

```
Non-trainable params: 0
_________________________________________________________________
```

[7]:
```
# Output model summary to file
with open(output_path, 'w') as f:
    f.write('Model Summary:')
    f.write('\n')

    # Pass the file handle in as a lambda function to make it callable
    model.summary(print_fn=lambda x: f.write(x + '\n'))
```

1.4 Load the data

[8]:
```
# Load data
# cifar10 data is in the shape (num_samples, height, width, channel) so no need␣
 ↪to reshape
(train_images, train_labels), (test_images, test_labels) = cifar10.load_data()

# Prepare training data
# Normalize to values btwn 0 and 1 since pizel range from 0-255
train_images = train_images.astype('float32') / 255

# Prepare test data
# Normalize to values btwn 0 and 1 since pizel range from 0-255
test_images = test_images.astype('float32') / 255

# Convert labels from integer to categorical
# cifar10 labels are in the shape (num_samples, 1)
# Transform label indices to one-hot encoded vectors
train_labels = to_categorical(train_labels, num_classes=10)
test_labels = to_categorical(test_labels, num_classes=10)
```

1.5 Data Augmentation

[9]:
```
# Setup Data Augmentation generator
# Views images different ways to increase dataset
from keras.preprocessing.image import ImageDataGenerator

train_datagen = ImageDataGenerator(
      rescale=1./255,
      rotation_range=40,
      width_shift_range=0.2,
      height_shift_range=0.2,
      shear_range=0.2,
      zoom_range=0.2,
      horizontal_flip=True,
      fill_mode='nearest')
```

```python
# Validation data should not be augmented - only training
test_datagen = ImageDataGenerator(rescale=1./255)

# Generates batches of augmentated data
# Uses flow method since data & labels in arrays
train_generator = train_datagen.flow(
        train_images,
        train_labels,
        batch_size=64,
        shuffle=True)

validation_generator = test_datagen.flow(
        test_images,
        test_labels,
        batch_size=20,
        shuffle=True)
```

[15]:
```python
# Train ConvNet on the CIFAR10 images
# Use categorical_crossentropy since multi-classification
model.compile(optimizer='rmsprop',
              loss='categorical_crossentropy',
              metrics=['accuracy'])

# Train Model using generator
# Collect measurement logs
history = model.fit(train_generator,
                    steps_per_epoch=200,
                    epochs=200,
                    validation_data=validation_generator,
                    validation_steps=50)
```

```
Epoch 1/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6822 -
accuracy: 0.3889 - val_loss: 1.4695 - val_accuracy: 0.4800
Epoch 2/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6614 -
accuracy: 0.4039 - val_loss: 1.4682 - val_accuracy: 0.4800
Epoch 3/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6574 -
accuracy: 0.4120 - val_loss: 1.4597 - val_accuracy: 0.4820
Epoch 4/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6470 -
accuracy: 0.4063 - val_loss: 1.4448 - val_accuracy: 0.4790
Epoch 5/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6487 -
accuracy: 0.4070 - val_loss: 1.4605 - val_accuracy: 0.4680
```

```
Epoch 6/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6486 -
accuracy: 0.4087 - val_loss: 1.4616 - val_accuracy: 0.4690
Epoch 7/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6568 -
accuracy: 0.4075 - val_loss: 1.4327 - val_accuracy: 0.5150
Epoch 8/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6385 -
accuracy: 0.4169 - val_loss: 1.4477 - val_accuracy: 0.4850
Epoch 9/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6540 -
accuracy: 0.4059 - val_loss: 1.4561 - val_accuracy: 0.4790
Epoch 10/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6556 -
accuracy: 0.4009 - val_loss: 1.4294 - val_accuracy: 0.5130
Epoch 11/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6546 -
accuracy: 0.4031 - val_loss: 1.5187 - val_accuracy: 0.4690
Epoch 12/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6617 -
accuracy: 0.3979 - val_loss: 1.4865 - val_accuracy: 0.4710
Epoch 13/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6695 -
accuracy: 0.3959 - val_loss: 1.4518 - val_accuracy: 0.4950
Epoch 14/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6464 -
accuracy: 0.4101 - val_loss: 1.4504 - val_accuracy: 0.4980
Epoch 15/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6455 -
accuracy: 0.4075 - val_loss: 1.4648 - val_accuracy: 0.4650
Epoch 16/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6509 -
accuracy: 0.4102 - val_loss: 1.4727 - val_accuracy: 0.4840
Epoch 17/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6405 -
accuracy: 0.4100 - val_loss: 1.4313 - val_accuracy: 0.4980
Epoch 18/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6456 -
accuracy: 0.4083 - val_loss: 1.5327 - val_accuracy: 0.4640
Epoch 19/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6564 -
accuracy: 0.4054 - val_loss: 1.4578 - val_accuracy: 0.4880
Epoch 20/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6348 -
accuracy: 0.4141 - val_loss: 1.5785 - val_accuracy: 0.4560
Epoch 21/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6298 -
accuracy: 0.4152 - val_loss: 1.5618 - val_accuracy: 0.4370
```

```
Epoch 22/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6452 -
accuracy: 0.4090 - val_loss: 1.5149 - val_accuracy: 0.4630
Epoch 23/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6299 -
accuracy: 0.4138 - val_loss: 1.5496 - val_accuracy: 0.4440
Epoch 24/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6432 -
accuracy: 0.4091 - val_loss: 1.6283 - val_accuracy: 0.4360
Epoch 25/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6396 -
accuracy: 0.4090 - val_loss: 1.5035 - val_accuracy: 0.4630
Epoch 26/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6444 -
accuracy: 0.4086 - val_loss: 1.5168 - val_accuracy: 0.4550
Epoch 27/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6537 -
accuracy: 0.4101 - val_loss: 1.4694 - val_accuracy: 0.5030
Epoch 28/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6368 -
accuracy: 0.4059 - val_loss: 1.3980 - val_accuracy: 0.4840
Epoch 29/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6423 -
accuracy: 0.4141 - val_loss: 1.4694 - val_accuracy: 0.4850
Epoch 30/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6311 -
accuracy: 0.4100 - val_loss: 1.4395 - val_accuracy: 0.5000
Epoch 31/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6379 -
accuracy: 0.4111 - val_loss: 1.4359 - val_accuracy: 0.4870
Epoch 32/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6317 -
accuracy: 0.4112 - val_loss: 1.4542 - val_accuracy: 0.4900
Epoch 33/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6302 -
accuracy: 0.4156 - val_loss: 1.4385 - val_accuracy: 0.4990
Epoch 34/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6431 -
accuracy: 0.4115 - val_loss: 1.4708 - val_accuracy: 0.4660
Epoch 35/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6373 -
accuracy: 0.4090 - val_loss: 1.4637 - val_accuracy: 0.4910
Epoch 36/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6430 -
accuracy: 0.4128 - val_loss: 1.5301 - val_accuracy: 0.4470
Epoch 37/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6351 -
accuracy: 0.4111 - val_loss: 1.4662 - val_accuracy: 0.4890
```

```
Epoch 38/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6250 -
accuracy: 0.4227 - val_loss: 1.5946 - val_accuracy: 0.4380
Epoch 39/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6328 -
accuracy: 0.4131 - val_loss: 1.4174 - val_accuracy: 0.4890
Epoch 40/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6366 -
accuracy: 0.4093 - val_loss: 1.4180 - val_accuracy: 0.5040
Epoch 41/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6217 -
accuracy: 0.4140 - val_loss: 1.4701 - val_accuracy: 0.4860
Epoch 42/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6274 -
accuracy: 0.4117 - val_loss: 1.4107 - val_accuracy: 0.5120
Epoch 43/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6295 -
accuracy: 0.4169 - val_loss: 1.4443 - val_accuracy: 0.4770
Epoch 44/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6285 -
accuracy: 0.4192 - val_loss: 1.5312 - val_accuracy: 0.4570
Epoch 45/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6220 -
accuracy: 0.4190 - val_loss: 1.4329 - val_accuracy: 0.4760
Epoch 46/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6156 -
accuracy: 0.4165 - val_loss: 1.4359 - val_accuracy: 0.4870
Epoch 47/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6181 -
accuracy: 0.4134 - val_loss: 1.4479 - val_accuracy: 0.4860
Epoch 48/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6326 -
accuracy: 0.4186 - val_loss: 1.4566 - val_accuracy: 0.4820
Epoch 49/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6224 -
accuracy: 0.4177 - val_loss: 1.4421 - val_accuracy: 0.4870
Epoch 50/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6348 -
accuracy: 0.4145 - val_loss: 1.4981 - val_accuracy: 0.4690
Epoch 51/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6158 -
accuracy: 0.4247 - val_loss: 1.4882 - val_accuracy: 0.4780
Epoch 52/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6126 -
accuracy: 0.4207 - val_loss: 1.4424 - val_accuracy: 0.4820
Epoch 53/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6133 -
accuracy: 0.4206 - val_loss: 1.4218 - val_accuracy: 0.4880
```

```
Epoch 54/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6188 -
accuracy: 0.4178 - val_loss: 1.5083 - val_accuracy: 0.4550
Epoch 55/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6177 -
accuracy: 0.4178 - val_loss: 1.3915 - val_accuracy: 0.4900
Epoch 56/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6365 -
accuracy: 0.4132 - val_loss: 1.3621 - val_accuracy: 0.5300
Epoch 57/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6097 -
accuracy: 0.4235 - val_loss: 1.4712 - val_accuracy: 0.4850
Epoch 58/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6226 -
accuracy: 0.4146 - val_loss: 1.3858 - val_accuracy: 0.5170
Epoch 59/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6206 -
accuracy: 0.4150 - val_loss: 1.4727 - val_accuracy: 0.4910
Epoch 60/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6165 -
accuracy: 0.4206 - val_loss: 1.4731 - val_accuracy: 0.4900
Epoch 61/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6143 -
accuracy: 0.4205 - val_loss: 1.3876 - val_accuracy: 0.5170
Epoch 62/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6205 -
accuracy: 0.4187 - val_loss: 1.4379 - val_accuracy: 0.4760
Epoch 63/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6049 -
accuracy: 0.4208 - val_loss: 1.4437 - val_accuracy: 0.4730
Epoch 64/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6147 -
accuracy: 0.4240 - val_loss: 1.4194 - val_accuracy: 0.5030
Epoch 65/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6089 -
accuracy: 0.4226 - val_loss: 1.4011 - val_accuracy: 0.5140
Epoch 66/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6048 -
accuracy: 0.4308 - val_loss: 1.5324 - val_accuracy: 0.4590
Epoch 67/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6217 -
accuracy: 0.4212 - val_loss: 1.4083 - val_accuracy: 0.4970
Epoch 68/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6191 -
accuracy: 0.4237 - val_loss: 1.3531 - val_accuracy: 0.5170
Epoch 69/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5953 -
accuracy: 0.4293 - val_loss: 1.5316 - val_accuracy: 0.4610
```

```
Epoch 70/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6009 -
accuracy: 0.4292 - val_loss: 1.4202 - val_accuracy: 0.4770
Epoch 71/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6089 -
accuracy: 0.4209 - val_loss: 1.3747 - val_accuracy: 0.5190
Epoch 72/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5997 -
accuracy: 0.4236 - val_loss: 1.4033 - val_accuracy: 0.5030
Epoch 73/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6075 -
accuracy: 0.4241 - val_loss: 1.4399 - val_accuracy: 0.5050
Epoch 74/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6027 -
accuracy: 0.4246 - val_loss: 1.3657 - val_accuracy: 0.5050
Epoch 75/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6026 -
accuracy: 0.4298 - val_loss: 1.4277 - val_accuracy: 0.4950
Epoch 76/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6109 -
accuracy: 0.4254 - val_loss: 1.4283 - val_accuracy: 0.4990
Epoch 77/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5959 -
accuracy: 0.4279 - val_loss: 1.4661 - val_accuracy: 0.4690
Epoch 78/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6100 -
accuracy: 0.4198 - val_loss: 1.4000 - val_accuracy: 0.5110
Epoch 79/200
200/200 [==============================] - 9s 45ms/step - loss: 1.6132 -
accuracy: 0.4195 - val_loss: 1.3716 - val_accuracy: 0.5110
Epoch 80/200
200/200 [==============================] - 9s 47ms/step - loss: 1.6010 -
accuracy: 0.4268 - val_loss: 1.4813 - val_accuracy: 0.4640
Epoch 81/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6090 -
accuracy: 0.4238 - val_loss: 1.4475 - val_accuracy: 0.4870
Epoch 82/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5974 -
accuracy: 0.4239 - val_loss: 1.4093 - val_accuracy: 0.4990
Epoch 83/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5961 -
accuracy: 0.4330 - val_loss: 1.5120 - val_accuracy: 0.4740
Epoch 84/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5984 -
accuracy: 0.4270 - val_loss: 1.4194 - val_accuracy: 0.4900
Epoch 85/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5963 -
accuracy: 0.4295 - val_loss: 1.4268 - val_accuracy: 0.4850
```

```
Epoch 86/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5905 -
accuracy: 0.4295 - val_loss: 1.3984 - val_accuracy: 0.4870
Epoch 87/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6070 -
accuracy: 0.4205 - val_loss: 1.4212 - val_accuracy: 0.4880
Epoch 88/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5887 -
accuracy: 0.4277 - val_loss: 1.4361 - val_accuracy: 0.4820
Epoch 89/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5960 -
accuracy: 0.4277 - val_loss: 1.4089 - val_accuracy: 0.4860
Epoch 90/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5795 -
accuracy: 0.4339 - val_loss: 1.4105 - val_accuracy: 0.5010
Epoch 91/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5864 -
accuracy: 0.4308 - val_loss: 1.4176 - val_accuracy: 0.5030
Epoch 92/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5883 -
accuracy: 0.4294 - val_loss: 1.4085 - val_accuracy: 0.5150
Epoch 93/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5931 -
accuracy: 0.4277 - val_loss: 1.4259 - val_accuracy: 0.4880
Epoch 94/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5918 -
accuracy: 0.4305 - val_loss: 1.4270 - val_accuracy: 0.4920
Epoch 95/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5944 -
accuracy: 0.4285 - val_loss: 1.4627 - val_accuracy: 0.4690
Epoch 96/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6085 -
accuracy: 0.4244 - val_loss: 1.4087 - val_accuracy: 0.5130
Epoch 97/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5955 -
accuracy: 0.4283 - val_loss: 1.4383 - val_accuracy: 0.4840
Epoch 98/200
200/200 [==============================] - 9s 46ms/step - loss: 1.6107 -
accuracy: 0.4210 - val_loss: 1.3737 - val_accuracy: 0.5260
Epoch 99/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5946 -
accuracy: 0.4277 - val_loss: 1.3768 - val_accuracy: 0.5270
Epoch 100/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5918 -
accuracy: 0.4259 - val_loss: 1.4127 - val_accuracy: 0.4940
Epoch 101/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5953 -
accuracy: 0.4228 - val_loss: 1.4373 - val_accuracy: 0.5010
```

```
Epoch 102/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5834 -
accuracy: 0.4344 - val_loss: 1.3396 - val_accuracy: 0.5330
Epoch 103/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5972 -
accuracy: 0.4274 - val_loss: 1.4431 - val_accuracy: 0.4970
Epoch 104/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5822 -
accuracy: 0.4355 - val_loss: 1.3747 - val_accuracy: 0.4980
Epoch 105/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5936 -
accuracy: 0.4313 - val_loss: 1.4588 - val_accuracy: 0.4810
Epoch 106/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5852 -
accuracy: 0.4322 - val_loss: 1.5845 - val_accuracy: 0.4490
Epoch 107/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5871 -
accuracy: 0.4330 - val_loss: 1.3880 - val_accuracy: 0.5090
Epoch 108/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5846 -
accuracy: 0.4286 - val_loss: 1.3160 - val_accuracy: 0.5460
Epoch 109/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5862 -
accuracy: 0.4317 - val_loss: 1.4424 - val_accuracy: 0.4710
Epoch 110/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5803 -
accuracy: 0.4305 - val_loss: 1.3705 - val_accuracy: 0.5080
Epoch 111/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5747 -
accuracy: 0.4352 - val_loss: 1.4063 - val_accuracy: 0.5250
Epoch 112/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5823 -
accuracy: 0.4250 - val_loss: 1.3758 - val_accuracy: 0.5390
Epoch 113/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5837 -
accuracy: 0.4289 - val_loss: 1.4716 - val_accuracy: 0.5030
Epoch 114/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5855 -
accuracy: 0.4288 - val_loss: 1.4158 - val_accuracy: 0.4920
Epoch 115/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5898 -
accuracy: 0.4323 - val_loss: 1.3459 - val_accuracy: 0.5290
Epoch 116/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5761 -
accuracy: 0.4402 - val_loss: 1.3882 - val_accuracy: 0.5050
Epoch 117/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5808 -
accuracy: 0.4392 - val_loss: 1.4156 - val_accuracy: 0.4870
```

```
Epoch 118/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5768 -
accuracy: 0.4323 - val_loss: 1.4704 - val_accuracy: 0.4800
Epoch 119/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5769 -
accuracy: 0.4348 - val_loss: 1.4076 - val_accuracy: 0.4910
Epoch 120/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5933 -
accuracy: 0.4288 - val_loss: 1.3834 - val_accuracy: 0.5040
Epoch 121/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5845 -
accuracy: 0.4295 - val_loss: 1.4101 - val_accuracy: 0.5080
Epoch 122/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5816 -
accuracy: 0.4343 - val_loss: 1.4409 - val_accuracy: 0.4770
Epoch 123/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5790 -
accuracy: 0.4342 - val_loss: 1.4466 - val_accuracy: 0.4920
Epoch 124/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5772 -
accuracy: 0.4382 - val_loss: 1.4558 - val_accuracy: 0.4910
Epoch 125/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5735 -
accuracy: 0.4388 - val_loss: 1.3597 - val_accuracy: 0.5260
Epoch 126/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5756 -
accuracy: 0.4385 - val_loss: 1.4358 - val_accuracy: 0.4890
Epoch 127/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5788 -
accuracy: 0.4359 - val_loss: 1.4056 - val_accuracy: 0.5180
Epoch 128/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5937 -
accuracy: 0.4280 - val_loss: 1.3877 - val_accuracy: 0.4930
Epoch 129/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5716 -
accuracy: 0.4384 - val_loss: 1.3862 - val_accuracy: 0.4940
Epoch 130/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5637 -
accuracy: 0.4421 - val_loss: 1.4481 - val_accuracy: 0.4990
Epoch 131/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5767 -
accuracy: 0.4391 - val_loss: 1.4822 - val_accuracy: 0.4700
Epoch 132/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5709 -
accuracy: 0.4319 - val_loss: 1.3662 - val_accuracy: 0.5340
Epoch 133/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5635 -
accuracy: 0.4389 - val_loss: 1.3811 - val_accuracy: 0.4970
```

```
Epoch 134/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5718 -
accuracy: 0.4398 - val_loss: 1.4674 - val_accuracy: 0.4820
Epoch 135/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5611 -
accuracy: 0.4416 - val_loss: 1.4159 - val_accuracy: 0.4780
Epoch 136/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5612 -
accuracy: 0.4424 - val_loss: 1.3939 - val_accuracy: 0.4990
Epoch 137/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5601 -
accuracy: 0.4399 - val_loss: 1.4435 - val_accuracy: 0.4940
Epoch 138/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5699 -
accuracy: 0.4320 - val_loss: 1.3817 - val_accuracy: 0.5100
Epoch 139/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5733 -
accuracy: 0.4373 - val_loss: 1.3441 - val_accuracy: 0.5360
Epoch 140/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5744 -
accuracy: 0.4433 - val_loss: 1.4103 - val_accuracy: 0.4810
Epoch 141/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5874 -
accuracy: 0.4341 - val_loss: 1.4339 - val_accuracy: 0.4940
Epoch 142/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5606 -
accuracy: 0.4427 - val_loss: 1.4095 - val_accuracy: 0.4990
Epoch 143/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5725 -
accuracy: 0.4368 - val_loss: 1.3732 - val_accuracy: 0.5010
Epoch 144/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5656 -
accuracy: 0.4430 - val_loss: 1.3784 - val_accuracy: 0.5070
Epoch 145/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5787 -
accuracy: 0.4375 - val_loss: 1.4101 - val_accuracy: 0.5110
Epoch 146/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5738 -
accuracy: 0.4377 - val_loss: 1.4004 - val_accuracy: 0.4990
Epoch 147/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5515 -
accuracy: 0.4455 - val_loss: 1.4960 - val_accuracy: 0.4890
Epoch 148/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5644 -
accuracy: 0.4386 - val_loss: 1.3596 - val_accuracy: 0.5300
Epoch 149/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5706 -
accuracy: 0.4388 - val_loss: 1.3966 - val_accuracy: 0.4920
```

```
Epoch 150/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5648 -
accuracy: 0.4390 - val_loss: 1.4538 - val_accuracy: 0.4710
Epoch 151/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5636 -
accuracy: 0.4409 - val_loss: 1.3369 - val_accuracy: 0.5210
Epoch 152/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5791 -
accuracy: 0.4368 - val_loss: 1.3661 - val_accuracy: 0.5120
Epoch 153/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5699 -
accuracy: 0.4419 - val_loss: 1.3803 - val_accuracy: 0.5110
Epoch 154/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5634 -
accuracy: 0.4489 - val_loss: 1.3997 - val_accuracy: 0.4920
Epoch 155/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5613 -
accuracy: 0.4435 - val_loss: 1.3150 - val_accuracy: 0.5420
Epoch 156/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5678 -
accuracy: 0.4384 - val_loss: 1.4030 - val_accuracy: 0.5180
Epoch 157/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5730 -
accuracy: 0.4348 - val_loss: 1.4035 - val_accuracy: 0.5130
Epoch 158/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5668 -
accuracy: 0.4384 - val_loss: 1.3655 - val_accuracy: 0.5220
Epoch 159/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5447 -
accuracy: 0.4498 - val_loss: 1.4292 - val_accuracy: 0.4920
Epoch 160/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5620 -
accuracy: 0.4446 - val_loss: 1.3696 - val_accuracy: 0.5090
Epoch 161/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5557 -
accuracy: 0.4441 - val_loss: 1.4075 - val_accuracy: 0.5070
Epoch 162/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5469 -
accuracy: 0.4474 - val_loss: 1.3906 - val_accuracy: 0.5140
Epoch 163/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5666 -
accuracy: 0.4450 - val_loss: 1.3707 - val_accuracy: 0.5320
Epoch 164/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5601 -
accuracy: 0.4412 - val_loss: 1.4008 - val_accuracy: 0.5100
Epoch 165/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5577 -
accuracy: 0.4434 - val_loss: 1.3446 - val_accuracy: 0.5300
```

```
Epoch 166/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5532 -
accuracy: 0.4471 - val_loss: 1.3169 - val_accuracy: 0.5460
Epoch 167/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5452 -
accuracy: 0.4435 - val_loss: 1.3544 - val_accuracy: 0.5260
Epoch 168/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5603 -
accuracy: 0.4410 - val_loss: 1.3996 - val_accuracy: 0.4950
Epoch 169/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5641 -
accuracy: 0.4428 - val_loss: 1.4127 - val_accuracy: 0.5050
Epoch 170/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5580 -
accuracy: 0.4435 - val_loss: 1.4196 - val_accuracy: 0.4810
Epoch 171/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5423 -
accuracy: 0.4514 - val_loss: 1.3477 - val_accuracy: 0.5300
Epoch 172/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5535 -
accuracy: 0.4465 - val_loss: 1.3132 - val_accuracy: 0.5190
Epoch 173/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5507 -
accuracy: 0.4486 - val_loss: 1.4145 - val_accuracy: 0.4880
Epoch 174/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5648 -
accuracy: 0.4426 - val_loss: 1.3202 - val_accuracy: 0.5410
Epoch 175/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5561 -
accuracy: 0.4461 - val_loss: 1.4079 - val_accuracy: 0.5010
Epoch 176/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5621 -
accuracy: 0.4431 - val_loss: 1.3285 - val_accuracy: 0.5120
Epoch 177/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5574 -
accuracy: 0.4422 - val_loss: 1.4250 - val_accuracy: 0.4940
Epoch 178/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5626 -
accuracy: 0.4385 - val_loss: 1.3768 - val_accuracy: 0.5150
Epoch 179/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5543 -
accuracy: 0.4427 - val_loss: 1.3962 - val_accuracy: 0.5070
Epoch 180/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5533 -
accuracy: 0.4473 - val_loss: 1.3620 - val_accuracy: 0.5120
Epoch 181/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5473 -
accuracy: 0.4466 - val_loss: 1.3382 - val_accuracy: 0.5220
```

```
Epoch 182/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5509 -
accuracy: 0.4473 - val_loss: 1.3605 - val_accuracy: 0.5250
Epoch 183/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5600 -
accuracy: 0.4419 - val_loss: 1.3676 - val_accuracy: 0.5320
Epoch 184/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5563 -
accuracy: 0.4443 - val_loss: 1.3860 - val_accuracy: 0.5090
Epoch 185/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5538 -
accuracy: 0.4470 - val_loss: 1.4734 - val_accuracy: 0.4800
Epoch 186/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5509 -
accuracy: 0.4422 - val_loss: 1.3647 - val_accuracy: 0.5180
Epoch 187/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5441 -
accuracy: 0.4514 - val_loss: 1.4115 - val_accuracy: 0.5030
Epoch 188/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5428 -
accuracy: 0.4492 - val_loss: 1.4096 - val_accuracy: 0.4980
Epoch 189/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5436 -
accuracy: 0.4473 - val_loss: 1.5431 - val_accuracy: 0.4640
Epoch 190/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5410 -
accuracy: 0.4495 - val_loss: 1.3432 - val_accuracy: 0.5270
Epoch 191/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5415 -
accuracy: 0.4468 - val_loss: 1.3966 - val_accuracy: 0.4990
Epoch 192/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5566 -
accuracy: 0.4381 - val_loss: 1.3737 - val_accuracy: 0.5130
Epoch 193/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5607 -
accuracy: 0.4450 - val_loss: 1.3821 - val_accuracy: 0.5230
Epoch 194/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5372 -
accuracy: 0.4498 - val_loss: 1.3398 - val_accuracy: 0.5440
Epoch 195/200
200/200 [==============================] - 9s 45ms/step - loss: 1.5535 -
accuracy: 0.4410 - val_loss: 1.3433 - val_accuracy: 0.5300
Epoch 196/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5530 -
accuracy: 0.4427 - val_loss: 1.2997 - val_accuracy: 0.5400
Epoch 197/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5815 -
accuracy: 0.4373 - val_loss: 1.2914 - val_accuracy: 0.5480
```

```
Epoch 198/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5550 -
accuracy: 0.4436 - val_loss: 1.4314 - val_accuracy: 0.4810
Epoch 199/200
200/200 [==============================] - 9s 46ms/step - loss: 1.5629 -
accuracy: 0.4402 - val_loss: 1.3659 - val_accuracy: 0.5270
Epoch 200/200
200/200 [==============================] - 9s 47ms/step - loss: 1.5461 -
accuracy: 0.4453 - val_loss: 1.3939 - val_accuracy: 0.5180
```

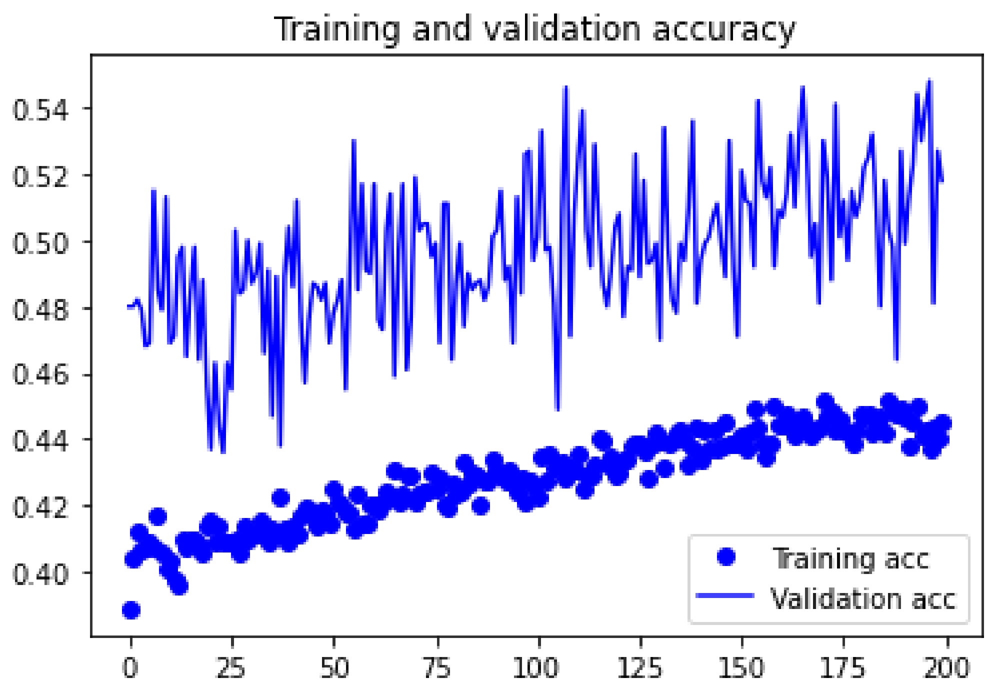
[16]:
```python
# Save Model
model.save(model_path)
```

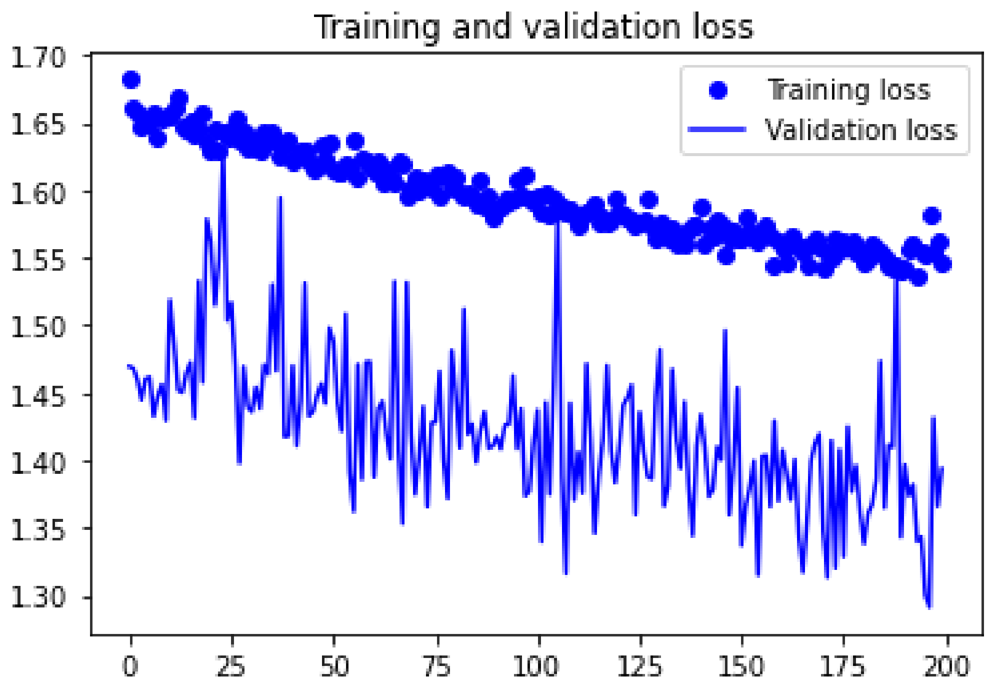
[17]:
```python
# Write history dictionary to file
import json

with open(output_path, 'a') as file:
    file.write('Final epoch metrics for original trained model:')
    json.dump(str(history.history), file)
```

1.6 Validate Model

[18]:
```python
# Plot the training and validation loss

acc = history.history['accuracy']
val_acc = history.history['val_accuracy']
loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(len(acc))

plt.plot(epochs, acc, 'bo', label='Training acc')
plt.plot(epochs, val_acc, 'b', label='Validation acc')
plt.title('Training and validation accuracy')
plt.legend()

# Save figure1 to drive
plt.savefig(image1_path)

plt.figure()

plt.plot(epochs, loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and validation loss')
plt.legend()

# Save figure2 to drive
```

```
plt.savefig(image2_path)

plt.show()
```





Although the accuracy is pretty low. The model is not overfit. It just levels out steadily.

```
[ ]:
```