NagarajVinay_Assignment_6_2b

April 24, 2021

0.1 Assignment 6.2b

0.1.1 CIFAR10 Classification with dropout and augmentation

[1]:
```
from keras.datasets import cifar10
from keras.utils import to_categorical
from keras.preprocessing.image import ImageDataGenerator

(x_train, y_train), (x_test, y_test) = cifar10.load_data()
```

[2]:
```
x_train.shape, y_train.shape
```

```
[2]: ((50000, 32, 32, 3), (50000, 1))
```

[3]:
```
x_test.shape, y_test.shape
```

```
[3]: ((10000, 32, 32, 3), (10000, 1))
```

[4]:
```
# Preprocess the data (these are NumPy arrays)
x_train = x_train.astype("float32")
x_test = x_test.astype("float32")

y_train = to_categorical(y_train)
y_test = to_categorical(y_test)
```

[5]:
```
# Reserve 10,000 samples for validation
x_val = x_train[-10000:]
y_val = y_train[-10000:]
x_train_2 = x_train[:-10000]
y_train_2 = y_train[:-10000]
```

[6]:
```
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)
```

```

test_datagen = ImageDataGenerator(rescale=1./255)

train_generator = train_datagen.flow(x_train_2, y_train_2,  batch_size=32)

validation_generator = train_datagen.flow(x_val, y_val, batch_size=32)
```

[7]:

```
#instantiate the model
from keras import models
from keras import layers

model = models.Sequential()
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(64, (3,3), activation='relu'))
model.add(layers.MaxPooling2D(2,2))
model.add(layers.Flatten())
model.add(layers.Dropout(0.5))
model.add(layers.Dense(64, activation='relu'))
model.add(layers.Dense(10, activation='softmax'))

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, 64)          36928
_________________________________________________________________
max_pooling2d_2 (MaxPooling2 (None, 2, 2, 64)          0
_________________________________________________________________
flatten (Flatten)            (None, 256)               0
_________________________________________________________________
dropout (Dropout)            (None, 256)               0
_________________________________________________________________
dense (Dense)                (None, 64)                16448
_________________________________________________________________
```

```
dense_1 (Dense)              (None, 10)                650
=================================================================
Total params: 73,418
Trainable params: 73,418
Non-trainable params: 0
_________________________________________________________________
```

[8]:
```python
from keras import optimizers

model.compile(optimizer=optimizers.RMSprop(lr=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])
```

[9]:
```python
history = model.fit_generator(train_generator,
                              steps_per_epoch=len(x_train_2) / 32,
                              epochs=30,
                              validation_data=validation_generator,
                              validation_steps=len(x_val) / 32)
```

```
/opt/conda/lib/python3.8/site-
packages/tensorflow/python/keras/engine/training.py:1844: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

```
Epoch 1/30
1250/1250 [==============================] - 44s 35ms/step - loss: 2.2161 -
accuracy: 0.1609 - val_loss: 2.0055 - val_accuracy: 0.2568
Epoch 2/30
1250/1250 [==============================] - 43s 35ms/step - loss: 1.9920 -
accuracy: 0.2504 - val_loss: 1.8961 - val_accuracy: 0.2979
Epoch 3/30
1250/1250 [==============================] - 42s 34ms/step - loss: 1.9093 -
accuracy: 0.2817 - val_loss: 1.8206 - val_accuracy: 0.3249
Epoch 4/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.8497 -
accuracy: 0.3087 - val_loss: 1.7776 - val_accuracy: 0.3418
Epoch 5/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.8120 -
accuracy: 0.3301 - val_loss: 1.7570 - val_accuracy: 0.3641
Epoch 6/30
1250/1250 [==============================] - 49s 39ms/step - loss: 1.7741 -
accuracy: 0.3423 - val_loss: 1.7185 - val_accuracy: 0.3729
Epoch 7/30
1250/1250 [==============================] - 51s 41ms/step - loss: 1.7549 -
accuracy: 0.3523 - val_loss: 1.6878 - val_accuracy: 0.3850
Epoch 8/30
1250/1250 [==============================] - 43s 35ms/step - loss: 1.7325 -
```

```
accuracy: 0.3604 - val_loss: 1.6704 - val_accuracy: 0.3969
Epoch 9/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.7020 -
accuracy: 0.3740 - val_loss: 1.7073 - val_accuracy: 0.3863
Epoch 10/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6854 -
accuracy: 0.3836 - val_loss: 1.6548 - val_accuracy: 0.4063
Epoch 11/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6722 -
accuracy: 0.3881 - val_loss: 1.6066 - val_accuracy: 0.4160
Epoch 12/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6497 -
accuracy: 0.3975 - val_loss: 1.6146 - val_accuracy: 0.4087
Epoch 13/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6259 -
accuracy: 0.4130 - val_loss: 1.5774 - val_accuracy: 0.4256
Epoch 14/30
1250/1250 [==============================] - 43s 35ms/step - loss: 1.6167 -
accuracy: 0.4133 - val_loss: 1.5786 - val_accuracy: 0.4354
Epoch 15/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6175 -
accuracy: 0.4164 - val_loss: 1.5328 - val_accuracy: 0.4449
Epoch 16/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.6017 -
accuracy: 0.4133 - val_loss: 1.5546 - val_accuracy: 0.4447
Epoch 17/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.5749 -
accuracy: 0.4322 - val_loss: 1.5044 - val_accuracy: 0.4636
Epoch 18/30
1250/1250 [==============================] - 42s 34ms/step - loss: 1.5679 -
accuracy: 0.4311 - val_loss: 1.5043 - val_accuracy: 0.4653
Epoch 19/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.5536 -
accuracy: 0.4377 - val_loss: 1.4941 - val_accuracy: 0.4649
Epoch 20/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.5439 -
accuracy: 0.4423 - val_loss: 1.5046 - val_accuracy: 0.4630
Epoch 21/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.5418 -
accuracy: 0.4486 - val_loss: 1.4780 - val_accuracy: 0.4738
Epoch 22/30
1250/1250 [==============================] - 44s 35ms/step - loss: 1.5143 -
accuracy: 0.4561 - val_loss: 1.4639 - val_accuracy: 0.4764
Epoch 23/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.5201 -
accuracy: 0.4523 - val_loss: 1.4628 - val_accuracy: 0.4729
Epoch 24/30
1250/1250 [==============================] - 43s 35ms/step - loss: 1.4954 -
```

```
accuracy: 0.4634 - val_loss: 1.4927 - val_accuracy: 0.4640
Epoch 25/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4944 -
accuracy: 0.4624 - val_loss: 1.4540 - val_accuracy: 0.4787
Epoch 26/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4870 -
accuracy: 0.4685 - val_loss: 1.4296 - val_accuracy: 0.4884
Epoch 27/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4768 -
accuracy: 0.4696 - val_loss: 1.4212 - val_accuracy: 0.4955
Epoch 28/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4743 -
accuracy: 0.4696 - val_loss: 1.4201 - val_accuracy: 0.4955
Epoch 29/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4749 -
accuracy: 0.4737 - val_loss: 1.4032 - val_accuracy: 0.5011
Epoch 30/30
1250/1250 [==============================] - 43s 34ms/step - loss: 1.4503 -
accuracy: 0.4797 - val_loss: 1.4026 - val_accuracy: 0.4942
```

[10]:

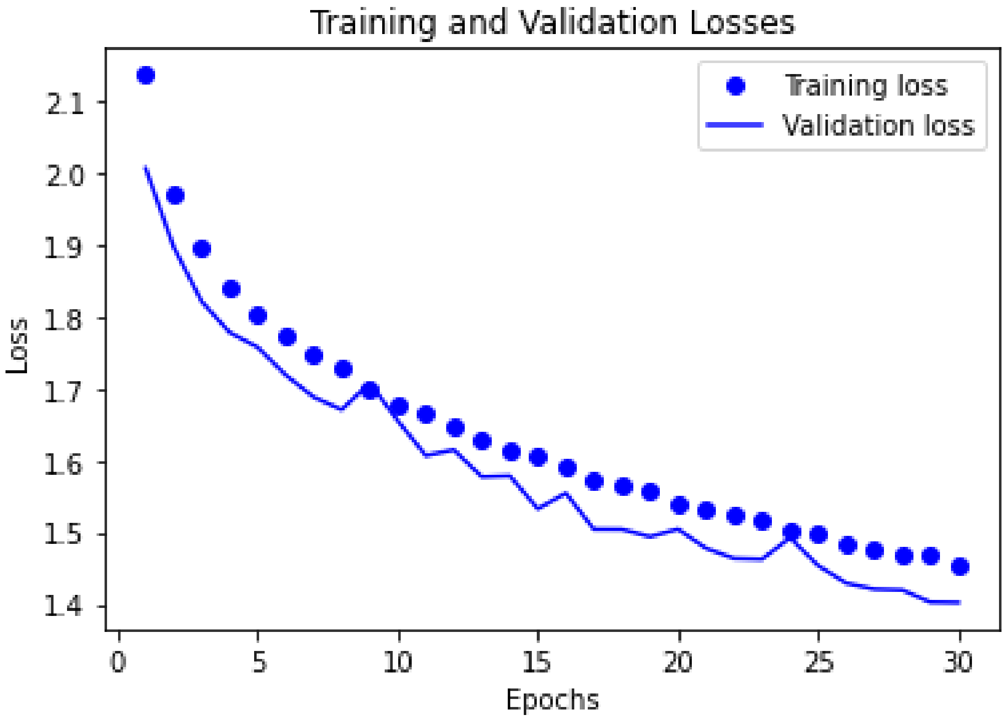
```python
import matplotlib.pyplot as plt

train_loss = history.history['loss']
val_loss = history.history['val_loss']

epochs = range(1, len(history.history['loss']) + 1)

plt.plot(epochs, train_loss, 'bo', label='Training loss')
plt.plot(epochs, val_loss, 'b', label='Validation loss')
plt.title('Training and Validation Losses')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()

plt.show()
plt.savefig('results/6_2b_lossplot.png')
```



```
<Figure size 432x288 with 0 Axes>
```

[11]:

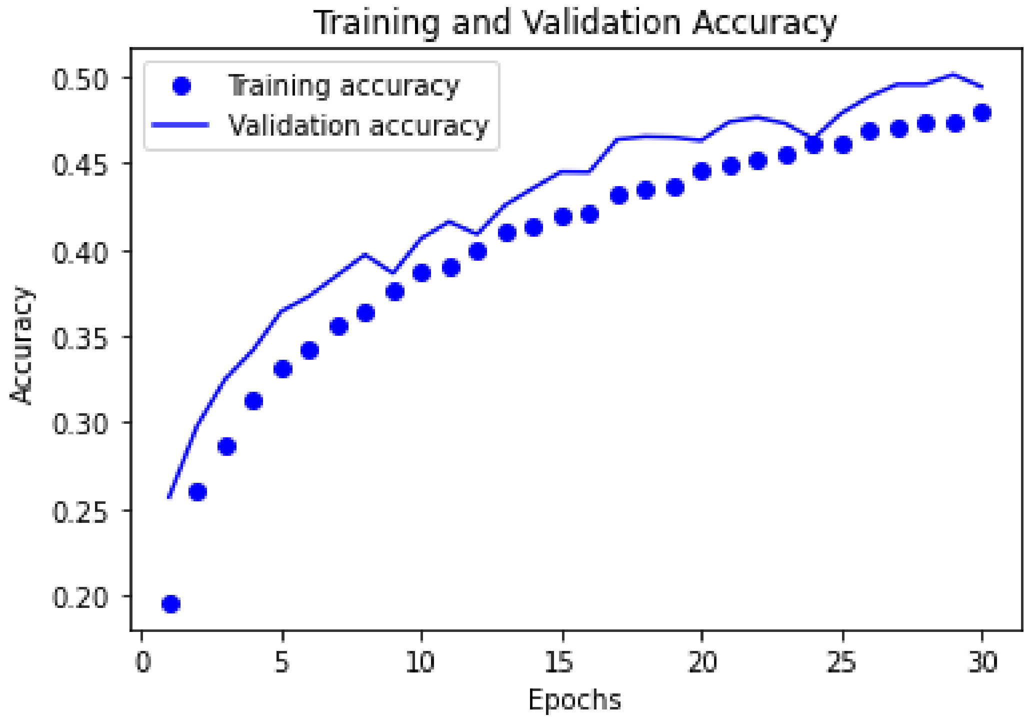
```
import matplotlib.pyplot as plt

train_loss = history.history['accuracy']
val_loss = history.history['val_accuracy']

epochs = range(1, len(history.history['accuracy']) + 1)

plt.plot(epochs, train_loss, 'bo', label='Training accuracy')
plt.plot(epochs, val_loss, 'b', label='Validation accuracy')
plt.title('Training and Validation Accuracy')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()

plt.show()
plt.savefig('results/6_2b_accplot.png')
```



```
<Figure size 432x288 with 0 Axes>
```

```
[12]: #retrain the model and evaluate on test
train_generator = train_datagen.flow(x_train, y_train,  batch_size=32)

model.compile(optimizer=optimizers.RMSprop(lr=1e-4),
              loss='categorical_crossentropy',
              metrics=['accuracy'])

#16 epochs chosen based on graphs above
history = model.fit_generator(train_generator,
                              steps_per_epoch=len(x_train) / 32,
                              epochs=16)
results = model.evaluate(x_test, y_test)
```

```
/opt/conda/lib/python3.8/site-
packages/tensorflow/python/keras/engine/training.py:1844: UserWarning:
`Model.fit_generator` is deprecated and will be removed in a future version.
Please use `Model.fit`, which supports generators.
  warnings.warn('`Model.fit_generator` is deprecated and '
```

```
Epoch 1/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.4564 -
accuracy: 0.4799
```

```
Epoch 2/16
1562/1562 [==============================] - 44s 28ms/step - loss: 1.4457 -
accuracy: 0.4815
Epoch 3/16
1562/1562 [==============================] - 45s 28ms/step - loss: 1.4408 -
accuracy: 0.4844
Epoch 4/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.4345 -
accuracy: 0.4864
Epoch 5/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.4245 -
accuracy: 0.4913
Epoch 6/16
1562/1562 [==============================] - 44s 28ms/step - loss: 1.4287 -
accuracy: 0.4875
Epoch 7/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.4113 -
accuracy: 0.4946
Epoch 8/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.4156 -
accuracy: 0.4961
Epoch 9/16
1562/1562 [==============================] - 45s 29ms/step - loss: 1.3995 -
accuracy: 0.5059
Epoch 10/16
1562/1562 [==============================] - 48s 31ms/step - loss: 1.4023 -
accuracy: 0.5019
Epoch 11/16
1562/1562 [==============================] - 56s 36ms/step - loss: 1.4012 -
accuracy: 0.5013
Epoch 12/16
1562/1562 [==============================] - 57s 36ms/step - loss: 1.3925 -
accuracy: 0.5038
Epoch 13/16
1562/1562 [==============================] - 57s 37ms/step - loss: 1.3662 -
accuracy: 0.5132
Epoch 14/16
1562/1562 [==============================] - 57s 37ms/step - loss: 1.3549 -
accuracy: 0.5171
Epoch 15/16
1562/1562 [==============================] - 57s 36ms/step - loss: 1.3738 -
accuracy: 0.5083
Epoch 16/16
1562/1562 [==============================] - 57s 36ms/step - loss: 1.3681 -
accuracy: 0.5101
313/313 [==============================] - 2s 4ms/step - loss: 284.4213 -
accuracy: 0.3377
```

[13]:

```
model.save('results/6_2b_model.h5')
```

[14]:

```
prediction_results = model.predict(x_test)
```

[15]:

```
#write metrics to file
with open('results/6_2b_metrics.txt', 'w') as f:
    f.write('Training Loss: {}'.format(str(history.history['loss'])))
    f.write('\nTraining Accuracy: {}'.format(str(history.history['accuracy'])))
    f.write('\nTest Loss: {}'.format(results[0]))
    f.write('\nTest Accuracy: {}'.format(results[1]))
```

[16]:

```
#write metrics to file
with open('results/6_2b_metrics.txt', 'w') as f:
    f.write('Training Loss: {}'.format(str(history.history['loss'])))
    f.write('\nTraining Accuracy: {}'.format(str(history.history['accuracy'])))
    f.write('\nTest Loss: {}'.format(results[0]))
    f.write('\nTest Accuracy: {}'.format(results[1]))
```

[ ]: