

LewisRebecca_Assignment_6_2b

January 24, 2021

1 Assignment 6.2b

1.1 CIFAR10 Classification with dropout and augmentation

1.1.1 Rebecca Lewis

```
[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 (MaxPooling2D)	(None, 6, 6, 64)	0
conv2d_2 (Conv2D)	(None, 4, 4, 64)	36928
max_pooling2d_2 (MaxPooling2D)	(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
-----
```

```
[10]: from keras import optimizers
```

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

```
[11]: 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)
```

WARNING:tensorflow:From <ipython-input-11-f55e5816947f>:1: Model.fit_generator (from tensorflow.python.keras.engine.training) is deprecated and will be removed in a future version.

Instructions for updating:

Please use Model.fit, which supports generators.

Epoch 1/30

```
1250/1250 [=====] - 44s 35ms/step - loss: 2.1454 -
accuracy: 0.1874 - val_loss: 1.9897 - val_accuracy: 0.2526
```

Epoch 2/30

```
1250/1250 [=====] - 43s 35ms/step - loss: 1.9676 -
accuracy: 0.2548 - val_loss: 1.8919 - val_accuracy: 0.3018
```

Epoch 3/30

```
1250/1250 [=====] - 43s 35ms/step - loss: 1.9030 -
accuracy: 0.2889 - val_loss: 1.8305 - val_accuracy: 0.3288
```

Epoch 4/30

```
1250/1250 [=====] - 44s 35ms/step - loss: 1.8506 -
accuracy: 0.3101 - val_loss: 1.7836 - val_accuracy: 0.3450
```

Epoch 5/30

```
1250/1250 [=====] - 43s 35ms/step - loss: 1.8164 -
accuracy: 0.3235 - val_loss: 1.7600 - val_accuracy: 0.3562
```

Epoch 6/30

```
1250/1250 [=====] - 43s 35ms/step - loss: 1.7849 -
accuracy: 0.3406 - val_loss: 1.7348 - val_accuracy: 0.3595
```

Epoch 7/30

```
1250/1250 [=====] - 43s 35ms/step - loss: 1.7588 -
accuracy: 0.3508 - val_loss: 1.7096 - val_accuracy: 0.3796
```

Epoch 8/30

1250/1250 [=====] - 43s 35ms/step - loss: 1.7351 - accuracy: 0.3595 - val_loss: 1.6801 - val_accuracy: 0.3849
Epoch 9/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.7184 - accuracy: 0.3655 - val_loss: 1.6595 - val_accuracy: 0.3969
Epoch 10/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.7000 - accuracy: 0.3766 - val_loss: 1.6349 - val_accuracy: 0.4051
Epoch 11/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.6824 - accuracy: 0.3857 - val_loss: 1.6192 - val_accuracy: 0.4073
Epoch 12/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.6632 - accuracy: 0.3929 - val_loss: 1.6042 - val_accuracy: 0.4169
Epoch 13/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.6462 - accuracy: 0.4001 - val_loss: 1.5920 - val_accuracy: 0.4225
Epoch 14/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.6333 - accuracy: 0.4053 - val_loss: 1.5903 - val_accuracy: 0.4230
Epoch 15/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.6145 - accuracy: 0.4138 - val_loss: 1.5587 - val_accuracy: 0.4384
Epoch 16/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.6035 - accuracy: 0.4203 - val_loss: 1.5533 - val_accuracy: 0.4334
Epoch 17/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.5893 - accuracy: 0.4245 - val_loss: 1.5325 - val_accuracy: 0.4459
Epoch 18/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.5800 - accuracy: 0.4304 - val_loss: 1.5411 - val_accuracy: 0.4425
Epoch 19/30
1250/1250 [=====] - 44s 35ms/step - loss: 1.5646 - accuracy: 0.4368 - val_loss: 1.5550 - val_accuracy: 0.4398
Epoch 20/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.5530 - accuracy: 0.4351 - val_loss: 1.4924 - val_accuracy: 0.4705
Epoch 21/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.5460 - accuracy: 0.4439 - val_loss: 1.4894 - val_accuracy: 0.4673
Epoch 22/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.5381 - accuracy: 0.4447 - val_loss: 1.4784 - val_accuracy: 0.4700
Epoch 23/30
1250/1250 [=====] - 44s 35ms/step - loss: 1.5335 - accuracy: 0.4490 - val_loss: 1.4724 - val_accuracy: 0.4791
Epoch 24/30

```

1250/1250 [=====] - 43s 34ms/step - loss: 1.5220 -
accuracy: 0.4534 - val_loss: 1.4437 - val_accuracy: 0.4859
Epoch 25/30
1250/1250 [=====] - 44s 35ms/step - loss: 1.5128 -
accuracy: 0.4588 - val_loss: 1.4715 - val_accuracy: 0.4727
Epoch 26/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.5046 -
accuracy: 0.4589 - val_loss: 1.4352 - val_accuracy: 0.4897
Epoch 27/30
1250/1250 [=====] - 44s 35ms/step - loss: 1.4933 -
accuracy: 0.4631 - val_loss: 1.4468 - val_accuracy: 0.4751
Epoch 28/30
1250/1250 [=====] - 43s 34ms/step - loss: 1.4878 -
accuracy: 0.4632 - val_loss: 1.4368 - val_accuracy: 0.4890
Epoch 29/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.4787 -
accuracy: 0.4701 - val_loss: 1.3975 - val_accuracy: 0.4982
Epoch 30/30
1250/1250 [=====] - 43s 35ms/step - loss: 1.4802 -
accuracy: 0.4697 - val_loss: 1.4048 - val_accuracy: 0.4979

```

```

[12]: import matplotlib.pyplot as plt

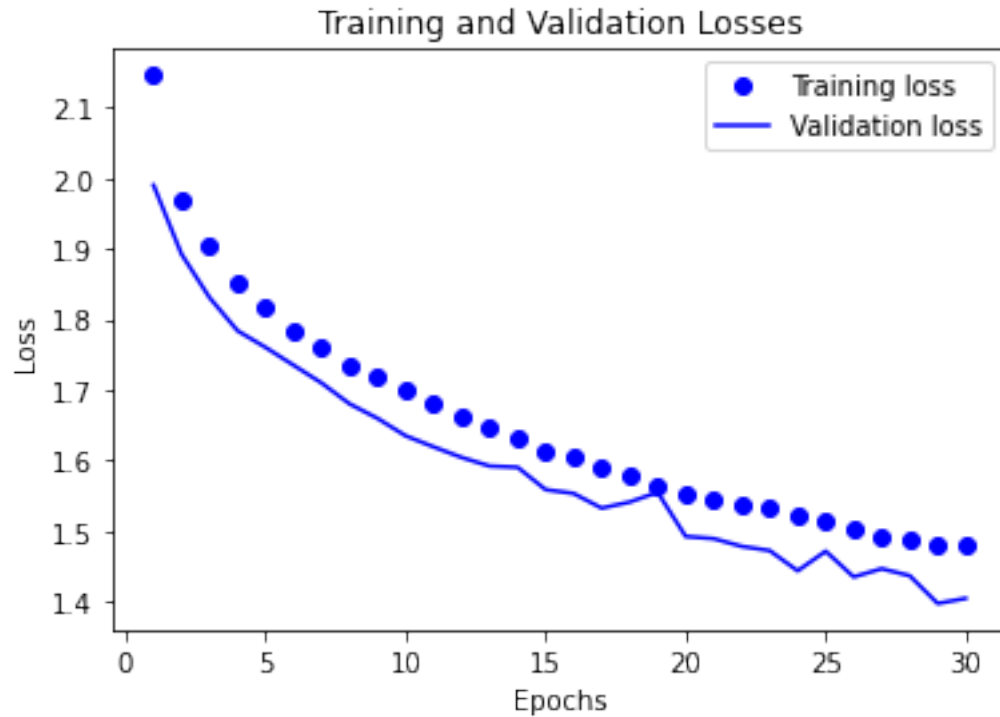
train_loss = history.history['loss']
val_loss = history.history['val_loss']

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/lewisrebecca_6_2b_lossplot.png')

```



<Figure size 432x288 with 0 Axes>

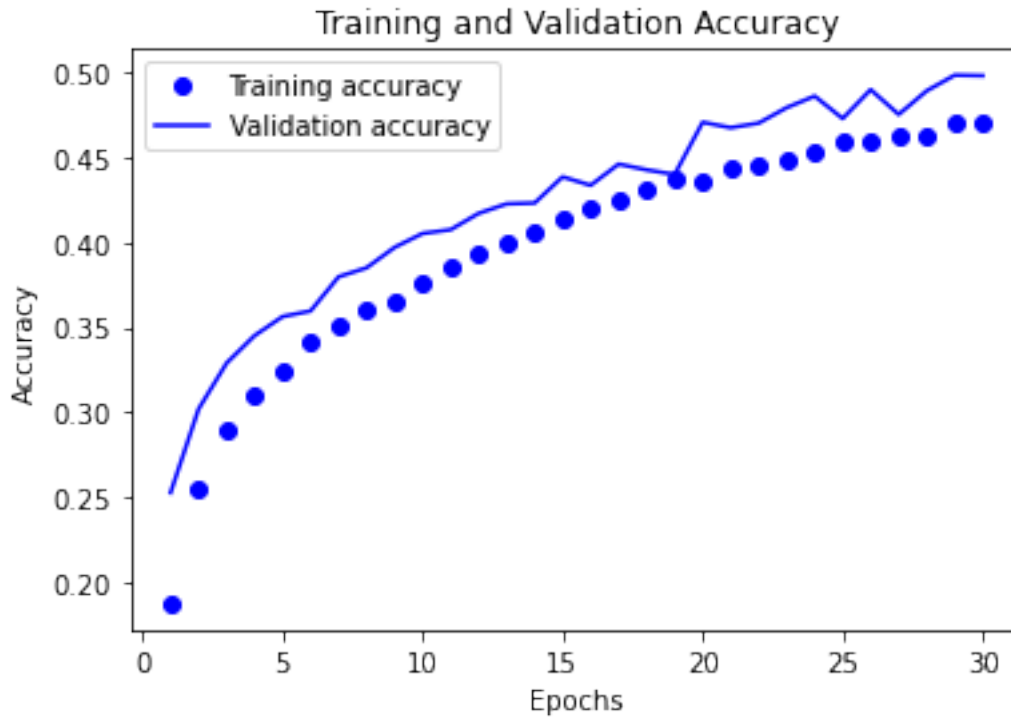
```
[13]: import matplotlib.pyplot as plt

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/lewisrebecca_6_2b_accplot.png')
```



<Figure size 432x288 with 0 Axes>

```
[15]: #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)
```

```
Epoch 1/16
1563/1562 [=====] - 45s 28ms/step - loss: 1.8227 -
accuracy: 0.3193
Epoch 2/16
1563/1562 [=====] - 45s 29ms/step - loss: 1.5316 -
accuracy: 0.4512
Epoch 3/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4854 -
```

```

accuracy: 0.4687
Epoch 4/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4678 -
accuracy: 0.4754
Epoch 5/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4525 -
accuracy: 0.4810
Epoch 6/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4461 -
accuracy: 0.4840
Epoch 7/16
1563/1562 [=====] - 45s 29ms/step - loss: 1.4384 -
accuracy: 0.4863
Epoch 8/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4299 -
accuracy: 0.4907
Epoch 9/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4221 -
accuracy: 0.4913
Epoch 10/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4192 -
accuracy: 0.4959
Epoch 11/16
1563/1562 [=====] - 45s 28ms/step - loss: 1.4092 -
accuracy: 0.4982
Epoch 12/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.4009 -
accuracy: 0.5009
Epoch 13/16
1563/1562 [=====] - 45s 28ms/step - loss: 1.3988 -
accuracy: 0.5039
Epoch 14/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.3878 -
accuracy: 0.5077
Epoch 15/16
1563/1562 [=====] - 44s 28ms/step - loss: 1.3875 -
accuracy: 0.5064
Epoch 16/16
1563/1562 [=====] - 45s 29ms/step - loss: 1.3832 -
accuracy: 0.5082
313/313 [=====] - 1s 4ms/step - loss: 251.8617 -
accuracy: 0.3811

```

```
[16]: model.save('results/lewisrebecca_6_2b_model.h5')
```

```
[17]: prediction_results = model.predict(x_test)
```



```
[18]: #write metrics to file
with open('results/lewisrebecca_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]))
```

```
[19]: import pandas as pd

predictions = pd.DataFrame(prediction_results,
    ↪columns=['0', '1', '2', '3', '4', '5', '6', '7', '8', '9'])
predictions.to_csv('results/lewisrebecca_6_2b_predictions.csv', index=False)
```

```
[ ]:
```