

ECGR_4105_HW7_Problem_1

December 9, 2024

```
[ ]: import matplotlib.pyplot as plt
import time
import tensorflow as tf

from tensorflow.keras import models, layers
from tensorflow.keras.datasets import cifar10
```

```
[ ]: # Load CIFAR-10 dataset
(x_train, y_train), (x_test, y_test) = cifar10.load_data()

x_train = x_train.astype('float32') / 255.0
x_test = x_test.astype('float32') / 255.0
```

Downloading data from <https://www.cs.toronto.edu/~kriz/cifar-10-python.tar.gz>
170498071/170498071 5s
0us/step

```
[ ]: # Part a: Basic CNN
def build_cnn():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Flatten(),
        layers.Dense(64, activation='relu'),
        layers.Dense(10, activation='softmax')
    ])
    return model

basic_cnn = build_cnn()
basic_cnn.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    ↪ metrics=['accuracy'])

# Train the basic CNN
start_time = time.time()
# history_basic = basic_cnn.fit(x_train, y_train, epochs=200,
    ↪ validation_data=(x_test, y_test))
```

```

# each epoch takes really long time to train, I limit it down to 20 to save time
history_basic = basic_cnn.fit(x_train, y_train, epochs=20,
    ↪validation_data=(x_test, y_test))
training_time_basic = time.time() - start_time

# Results
print("Basic CNN Training Time: {:.2f} seconds".format(training_time_basic))
print("Basic CNN Final Accuracy: {:.2f}%".format(history_basic.
    ↪history['val_accuracy'][-1] * 100))

```

```

Epoch 1/20
1563/1563      8s 4ms/step -
accuracy: 0.3882 - loss: 1.6860 - val_accuracy: 0.5686 - val_loss: 1.2078
Epoch 2/20
1563/1563      5s 3ms/step -
accuracy: 0.5960 - loss: 1.1481 - val_accuracy: 0.6369 - val_loss: 1.0426
Epoch 3/20
1563/1563      5s 3ms/step -
accuracy: 0.6543 - loss: 0.9845 - val_accuracy: 0.6609 - val_loss: 0.9817
Epoch 4/20
1563/1563      5s 3ms/step -
accuracy: 0.6904 - loss: 0.8934 - val_accuracy: 0.6721 - val_loss: 0.9569
Epoch 5/20
1563/1563      6s 3ms/step -
accuracy: 0.7176 - loss: 0.8210 - val_accuracy: 0.6642 - val_loss: 0.9863
Epoch 6/20
1563/1563      4s 3ms/step -
accuracy: 0.7354 - loss: 0.7598 - val_accuracy: 0.6906 - val_loss: 0.9021
Epoch 7/20
1563/1563      6s 3ms/step -
accuracy: 0.7535 - loss: 0.7062 - val_accuracy: 0.6976 - val_loss: 0.9012
Epoch 8/20
1563/1563     10s 3ms/step -
accuracy: 0.7686 - loss: 0.6598 - val_accuracy: 0.7008 - val_loss: 0.8843
Epoch 9/20
1563/1563      5s 3ms/step -
accuracy: 0.7873 - loss: 0.6135 - val_accuracy: 0.6960 - val_loss: 0.9330
Epoch 10/20
1563/1563      4s 3ms/step -
accuracy: 0.7988 - loss: 0.5825 - val_accuracy: 0.7045 - val_loss: 0.9258
Epoch 11/20
1563/1563      6s 3ms/step -
accuracy: 0.8076 - loss: 0.5479 - val_accuracy: 0.6927 - val_loss: 0.9660
Epoch 12/20
1563/1563      5s 3ms/step -
accuracy: 0.8241 - loss: 0.5041 - val_accuracy: 0.6854 - val_loss: 0.9788
Epoch 13/20

```

```

1563/1563          4s 3ms/step -
accuracy: 0.8296 - loss: 0.4792 - val_accuracy: 0.6994 - val_loss: 1.0159
Epoch 14/20
1563/1563          5s 3ms/step -
accuracy: 0.8434 - loss: 0.4503 - val_accuracy: 0.6969 - val_loss: 1.0231
Epoch 15/20
1563/1563          5s 3ms/step -
accuracy: 0.8508 - loss: 0.4205 - val_accuracy: 0.6890 - val_loss: 1.0832
Epoch 16/20
1563/1563          4s 3ms/step -
accuracy: 0.8601 - loss: 0.4040 - val_accuracy: 0.6946 - val_loss: 1.0683
Epoch 17/20
1563/1563          5s 3ms/step -
accuracy: 0.8713 - loss: 0.3686 - val_accuracy: 0.6892 - val_loss: 1.1390
Epoch 18/20
1563/1563          5s 3ms/step -
accuracy: 0.8774 - loss: 0.3475 - val_accuracy: 0.6747 - val_loss: 1.2649
Epoch 19/20
1563/1563          4s 3ms/step -
accuracy: 0.8820 - loss: 0.3360 - val_accuracy: 0.6868 - val_loss: 1.2070
Epoch 20/20
1563/1563          6s 3ms/step -
accuracy: 0.8918 - loss: 0.3043 - val_accuracy: 0.6842 - val_loss: 1.2418
Basic CNN Training Time: 107.89 seconds
Basic CNN Final Accuracy: 68.42%

```

```

[ ]: # Part b: Extended CNN
def build_extended_cnn():
    model = models.Sequential([
        layers.Conv2D(32, (3, 3), activation='relu', input_shape=(32, 32, 3)),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(64, (3, 3), activation='relu'),
        layers.MaxPooling2D((2, 2)),
        layers.Conv2D(128, (3, 3), activation='relu'), # Additional layer
        layers.MaxPooling2D((2, 2)),
        layers.Flatten(),
        layers.Dense(128, activation='relu'),
        layers.Dense(10, activation='softmax')
    ])
    return model

extended_cnn = build_extended_cnn()
extended_cnn.compile(optimizer='adam', loss='sparse_categorical_crossentropy',
    metrics=['accuracy'])

# Train the extended CNN
start_time = time.time()

```

```
# history_extended = extended_cnn.fit(x_train, y_train, epochs=200,
    ↪validation_data=(x_test, y_test))
history_extended = extended_cnn.fit(x_train, y_train, epochs=20,
    ↪validation_data=(x_test, y_test))
training_time_extended = time.time() - start_time

# Report results
print("Extended CNN Training Time: {:.2f} seconds".
    ↪format(training_time_extended))
print("Extended CNN Final Accuracy: {:.2f}%".format(history_extended.
    ↪history['val_accuracy'][-1] * 100))
```

```
Epoch 1/20
1563/1563          9s 5ms/step -
accuracy: 0.3526 - loss: 1.7426 - val_accuracy: 0.5552 - val_loss: 1.2347
Epoch 2/20
1563/1563          5s 3ms/step -
accuracy: 0.5890 - loss: 1.1541 - val_accuracy: 0.6187 - val_loss: 1.0910
Epoch 3/20
1563/1563          5s 3ms/step -
accuracy: 0.6682 - loss: 0.9496 - val_accuracy: 0.6786 - val_loss: 0.9361
Epoch 4/20
1563/1563          5s 3ms/step -
accuracy: 0.7086 - loss: 0.8293 - val_accuracy: 0.6633 - val_loss: 0.9707
Epoch 5/20
1563/1563          4s 3ms/step -
accuracy: 0.7417 - loss: 0.7362 - val_accuracy: 0.6986 - val_loss: 0.8723
Epoch 6/20
1563/1563          6s 4ms/step -
accuracy: 0.7642 - loss: 0.6741 - val_accuracy: 0.6974 - val_loss: 0.8713
Epoch 7/20
1563/1563          5s 3ms/step -
accuracy: 0.7848 - loss: 0.6131 - val_accuracy: 0.7024 - val_loss: 0.8860
Epoch 8/20
1563/1563          5s 3ms/step -
accuracy: 0.8042 - loss: 0.5561 - val_accuracy: 0.7105 - val_loss: 0.8950
Epoch 9/20
1563/1563          5s 3ms/step -
accuracy: 0.8174 - loss: 0.5142 - val_accuracy: 0.7207 - val_loss: 0.8664
Epoch 10/20
1563/1563         11s 3ms/step -
accuracy: 0.8382 - loss: 0.4636 - val_accuracy: 0.7313 - val_loss: 0.8656
Epoch 11/20
1563/1563          5s 3ms/step -
accuracy: 0.8524 - loss: 0.4214 - val_accuracy: 0.7218 - val_loss: 0.9385
Epoch 12/20
1563/1563          5s 3ms/step -
```

```

accuracy: 0.8652 - loss: 0.3801 - val_accuracy: 0.7032 - val_loss: 1.0703
Epoch 13/20
1563/1563          6s 3ms/step -
accuracy: 0.8748 - loss: 0.3496 - val_accuracy: 0.7260 - val_loss: 0.9743
Epoch 14/20
1563/1563          4s 3ms/step -
accuracy: 0.8860 - loss: 0.3154 - val_accuracy: 0.7194 - val_loss: 1.0310
Epoch 15/20
1563/1563          6s 3ms/step -
accuracy: 0.9002 - loss: 0.2820 - val_accuracy: 0.7086 - val_loss: 1.0806
Epoch 16/20
1563/1563          9s 3ms/step -
accuracy: 0.9055 - loss: 0.2619 - val_accuracy: 0.7059 - val_loss: 1.1580
Epoch 17/20
1563/1563          5s 3ms/step -
accuracy: 0.9147 - loss: 0.2396 - val_accuracy: 0.7070 - val_loss: 1.1595
Epoch 18/20
1563/1563         10s 3ms/step -
accuracy: 0.9206 - loss: 0.2193 - val_accuracy: 0.7010 - val_loss: 1.3501
Epoch 19/20
1563/1563         10s 3ms/step -
accuracy: 0.9262 - loss: 0.2046 - val_accuracy: 0.7085 - val_loss: 1.3151
Epoch 20/20
1563/1563          6s 3ms/step -
accuracy: 0.9309 - loss: 0.1906 - val_accuracy: 0.7053 - val_loss: 1.3330
Extended CNN Training Time: 127.40 seconds
Extended CNN Final Accuracy: 70.53%

```

```

[ ]: # Compare results
print("Basic CNN Model Size: {} parameters".format(basic_cnn.count_params()))
print("Extended CNN Model Size: {} parameters".format(extended_cnn.
    ↪count_params()))

# Plot accuracy curves
plt.plot(history_basic.history['val_accuracy'], label='Basic CNN')
plt.plot(history_extended.history['val_accuracy'], label='Extended CNN')
plt.title('Validation Accuracy Comparison')
plt.xlabel('Epochs')
plt.ylabel('Accuracy')
plt.legend()
plt.show()

```

```

Basic CNN Model Size: 167562 parameters
Extended CNN Model Size: 160202 parameters

```

