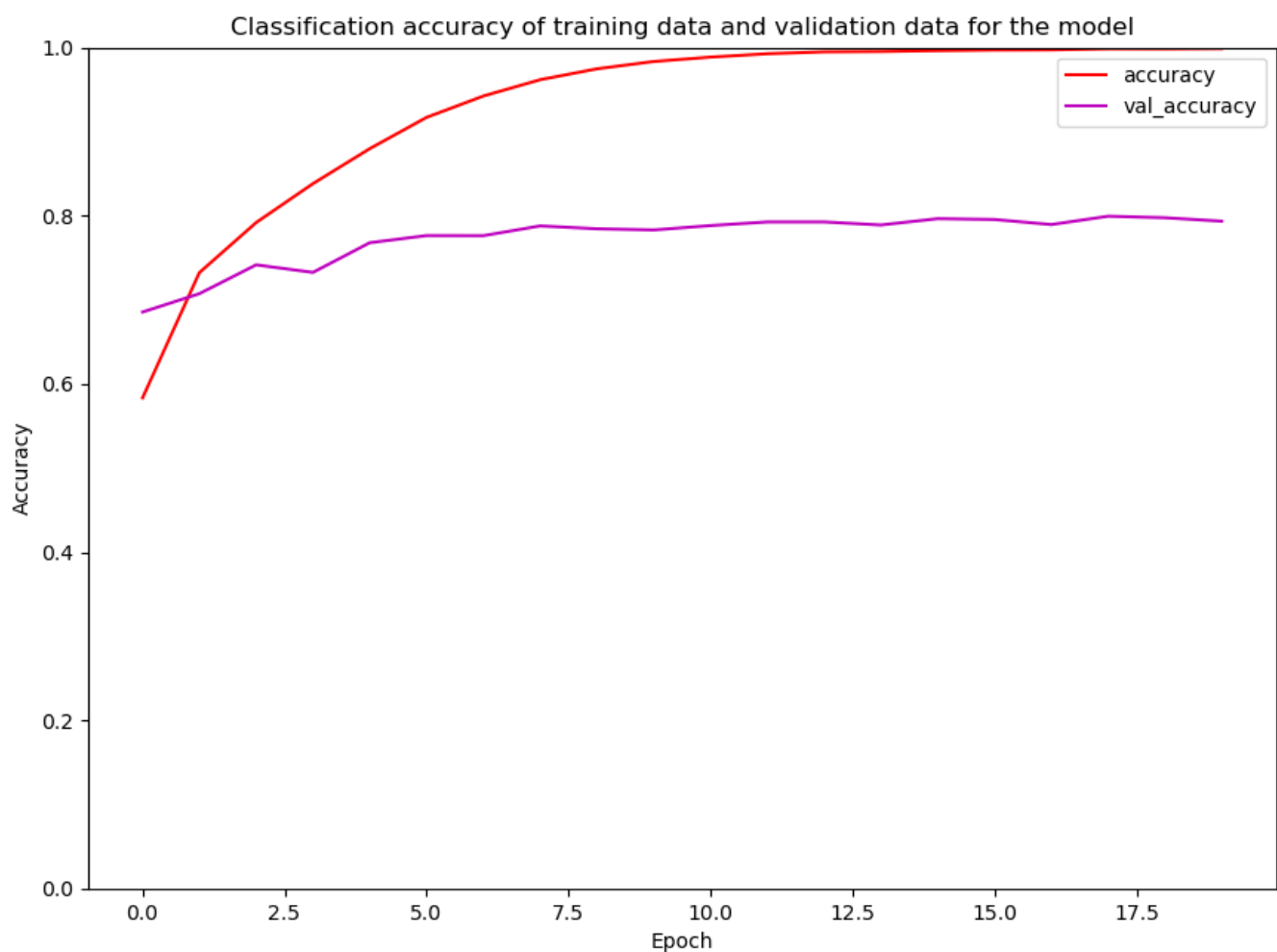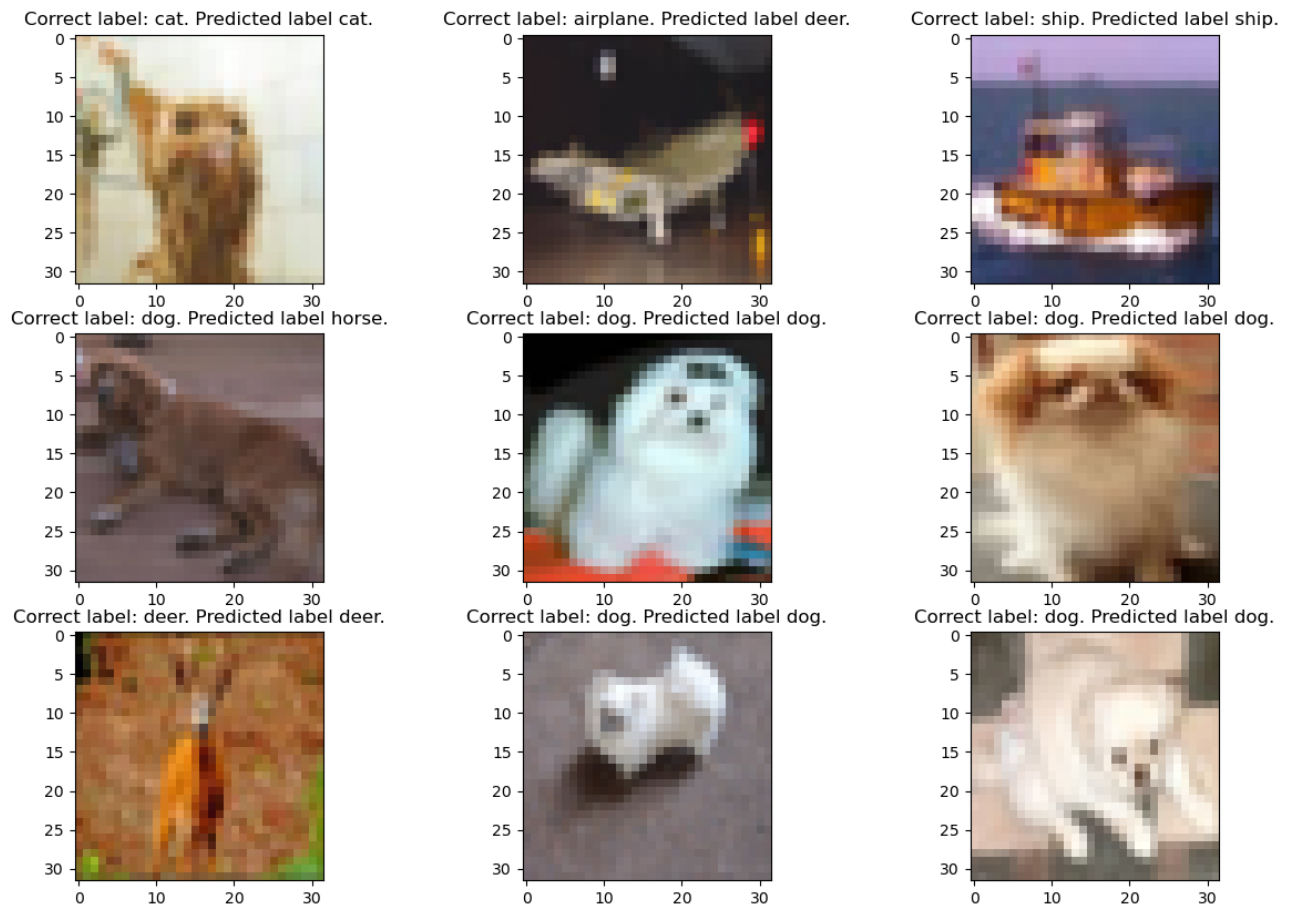```
(tf_gpu) C:\Users\Topsu\PycharmProjects\PRML\week5> python cifar10NeuralNetwork.py
Num GPUs Available: 1
2022-10-04 12:15:33.023650: I tensorflow/core/platform/cpu_feature_guard.cc:142] This TensorFlow binary is optimized with oneAPI Deep Neural Network Library (oneDNN) to use the following CPU instructions in performance-critical operations: AVX AVX2
To enable them in other operations, rebuild TensorFlow with the appropriate compiler flags.
2022-10-04 12:15:33.374773: I tensorflow/core/common_runtime/gpu/gpu_device.cc:1510] Created device /job:localhost/replica:0/task:0/device:GPU:0 with 6638 MB memory: -> device: 0, name: NVIDIA GeForce GTX 1070, pci bus id: 0000:01:00.0, compute capability: 6.1
2022-10-04 12:15:34.091988: I tensorflow/compiler/mlir/mlir_graph_optimization_pass.cc:185] None of the MLIR Optimization Passes are enabled (registered 2)
Epoch 1/20
2022-10-04 12:15:34.853541: I tensorflow/stream_executor/cuda/cuda_dnn.cc:369] Loaded cuDNN version 8201
2022-10-04 12:15:35.082349: E tensorflow/core/platform/windows/subprocess.cc:287] Call to CreateProcess failed. Error code: 2
2022-10-04 12:15:35.082924: E tensorflow/core/platform/windows/subprocess.cc:287] Call to CreateProcess failed. Error code: 2
2022-10-04 12:15:35.082989: W tensorflow/stream_executor/gpu/asm_compiler.cc:77] Couldn't get ptxas version string: Internal: Couldn't invoke ptxas.exe --version
2022-10-04 12:15:35.084408: E tensorflow/core/platform/windows/subprocess.cc:287] Call to CreateProcess failed. Error code: 2
2022-10-04 12:15:35.084619: W tensorflow/stream_executor/gpu/redzone_allocator.cc:314] Internal: Failed to launch ptxas
Relying on driver to perform ptx compilation.
Modify $PATH to customize ptxas location.
This message will be only logged once.
1250/1250 [==============================] - 10s 7ms/step - loss: 1.1692 - accuracy: 0.5845 - val_loss: 0.9221 - val_accuracy: 0.6731
Epoch 2/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.7653 - accuracy: 0.7303 - val_loss: 0.7740 - val_accuracy: 0.7277
Epoch 3/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.5973 - accuracy: 0.7931 - val_loss: 0.7270 - val_accuracy: 0.7533
Epoch 4/20
1250/1250 [==============================] - 8s 7ms/step - loss: 0.4637 - accuracy: 0.8396 - val_loss: 0.7261 - val_accuracy: 0.7459
Epoch 5/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.3476 - accuracy: 0.8795 - val_loss: 0.7183 - val_accuracy: 0.7627
Epoch 6/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.2513 - accuracy: 0.9161 - val_loss: 0.8273 - val_accuracy: 0.7412
Epoch 7/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.1740 - accuracy: 0.9445 - val_loss: 0.7538 - val_accuracy: 0.7691
Epoch 8/20
1250/1250 [==============================] - 8s 7ms/step - loss: 0.1154 - accuracy: 0.9665 - val_loss: 0.7327 - val_accuracy: 0.7873
Epoch 9/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0877 - accuracy: 0.9745 - val_loss: 0.7622 - val_accuracy: 0.7833
Epoch 10/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0650 - accuracy: 0.9823 - val_loss: 0.7842 - val_accuracy: 0.7861
Epoch 11/20
1250/1250 [==============================] - 8s 7ms/step - loss: 0.0495 - accuracy: 0.9877 - val_loss: 0.7858 - val_accuracy: 0.7906
Epoch 12/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0371 - accuracy: 0.9917 - val_loss: 0.8315 - val_accuracy: 0.7831
Epoch 13/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0288 - accuracy: 0.9938 - val_loss: 0.8141 - val_accuracy: 0.7909
Epoch 14/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0248 - accuracy: 0.9951 - val_loss: 0.8621 - val_accuracy: 0.7888
Epoch 15/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0230 - accuracy: 0.9953 - val_loss: 0.8082 - val_accuracy: 0.8000
Epoch 16/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0178 - accuracy: 0.9965 - val_loss: 0.8339 - val_accuracy: 0.7951
Epoch 17/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0158 - accuracy: 0.9971 - val_loss: 0.8755 - val_accuracy: 0.7868
Epoch 18/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0124 - accuracy: 0.9981 - val_loss: 0.8564 - val_accuracy: 0.7985
Epoch 19/20
1250/1250 [==============================] - 8s 7ms/step - loss: 0.0107 - accuracy: 0.9984 - val_loss: 0.8466 - val_accuracy: 0.7974
Epoch 20/20
1250/1250 [==============================] - 8s 6ms/step - loss: 0.0100 - accuracy: 0.9983 - val_loss: 0.8496 - val_accuracy: 0.7996
Time elapsed: 162.3658593 seconds.

313/313 [==============================] - 1s 4ms/step - loss: 0.8877 - accuracy: 0.7995
Classification accuracy for test data: 0.7994999885559082

(tf_gpu) C:\Users\Topsu\PycharmProjects\PRML\week5>
```



Classification accuracy of training data and validation data for the model

There is some overfitting happening. I managed to make it a bit better but could be honed more.

Classification of 9 random images from the test data

Correct label: cat. Predicted label cat.    Correct label: airplane. Predicted label deer.    Correct label: ship. Predicted label ship.

Correct label: dog. Predicted label horse.    Correct label: dog. Predicted label dog.    Correct label: dog. Predicted label dog.

Correct label: deer. Predicted label deer.    Correct label: dog. Predicted label dog.    Correct label: dog. Predicted label dog.

The classification accuracies and runtimes of different methods:

1 Nearest-Neighbour: 25,4% & 2824 seconds

Naïve Bayesian (1x1 images): 19,5% & 23 seconds

Bayesian (1x1 images): 24,7% & 13.3 seconds

Neural Network: 79,9% & 162 seconds (Tensorflow-GPU NVIDIA 1070)

From these we can clearly see that Neural Network is the superior method compared to the earlier ones.