ml-lab-ex9

March 24, 2024

1 ML EX9

Implement a neural network from scratch. Take any dataset. If you take a regression problem, use the equations derived in the class. If you take a classification problem, use the below equations: (run minimum 200 iterations and get the result). Use the gradient descent optimization technique for weight optimization.

1.1 Scratch

Dataset Used: MNIST

[ ]:
```
import numpy as np
import pandas as pd
from matplotlib import pyplot as plt
```

[ ]:
```
df = pd.read_csv('mnist.csv')
```

[ ]:
```
df = np.array(df)
m, n = df.shape
np.random.shuffle(df)

test_data = df[0:1000].T
Y_test = test_data[0]
X_test = test_data[1:n]
X_test = X_test / 255.

train_data = df[1000:m].T
Y_train = train_data[0]
X_train = train_data[1:n]
X_train = X_train / 255.
_,m_train = X_train.shape
```

[ ]:
```
def init_params():
    W1 = np.random.rand(10, 784) - 0.5
    b1 = np.random.rand(10, 1) - 0.5
    W2 = np.random.rand(10, 10) - 0.5
    b2 = np.random.rand(10, 1) - 0.5
    return W1, b1, W2, b2
```

```python
def ReLU(Z):
    return np.maximum(Z, 0)

def softmax(Z):
    A = np.exp(Z) / sum(np.exp(Z))
    return A

def forward_prop(W1, b1, W2, b2, X):
    Z1 = W1.dot(X) + b1
    A1 = ReLU(Z1)
    Z2 = W2.dot(A1) + b2
    A2 = softmax(Z2)
    return Z1, A1, Z2, A2

def ReLU_deriv(Z):
    return Z > 0

def one_hot(Y):
    one_hot_Y = np.zeros((Y.size, Y.max() + 1))
    one_hot_Y[np.arange(Y.size), Y] = 1
    one_hot_Y = one_hot_Y.T
    return one_hot_Y

def backward_prop(Z1, A1, Z2, A2, W1, W2, X, Y):
    one_hot_Y = one_hot(Y)
    dZ2 = A2 - one_hot_Y
    dW2 = 1 / m * dZ2.dot(A1.T)
    db2 = 1 / m * np.sum(dZ2)
    dZ1 = W2.T.dot(dZ2) * ReLU_deriv(Z1)
    dW1 = 1 / m * dZ1.dot(X.T)
    db1 = 1 / m * np.sum(dZ1)
    return dW1, db1, dW2, db2

def update_params(W1, b1, W2, b2, dW1, db1, dW2, db2, alpha):
    W1 = W1 - alpha * dW1
    b1 = b1 - alpha * db1
    W2 = W2 - alpha * dW2
    b2 = b2 - alpha * db2
    return W1, b1, W2, b2
```

[ ]:
```python
def get_predictions(A2):
    return np.argmax(A2, 0)

def get_accuracy(predictions, Y):
    print(predictions, Y)
    return np.sum(predictions == Y) / Y.size
```

```python
def gradient_descent(X, Y, alpha, iterations):
    W1, b1, W2, b2 = init_params()
    for i in range(iterations):
        Z1, A1, Z2, A2 = forward_prop(W1, b1, W2, b2, X)
        dW1, db1, dW2, db2 = backward_prop(Z1, A1, Z2, A2, W1, W2, X, Y)
        W1, b1, W2, b2 = update_params(W1, b1, W2, b2, dW1, db1, dW2, db2,␣
 ↪alpha)
        if i % 10 == 0:
            print("Iteration: ", i)
            predictions = get_predictions(A2)
            print(get_accuracy(predictions, Y))
    return W1, b1, W2, b2
```

[ ]:
```python
W1, b1, W2, b2 = gradient_descent(X_train, Y_train, 0.10, 200)
```

```
Iteration:  0
[7 4 7 … 0 6 3] [1 6 5 … 3 6 9]
0.07221951219512195
Iteration:  10
[3 4 9 … 8 2 5] [1 6 5 … 3 6 9]
0.1721951219512195
Iteration:  20
[1 3 9 … 3 2 7] [1 6 5 … 3 6 9]
0.2585121951219512
Iteration:  30
[1 3 9 … 3 2 7] [1 6 5 … 3 6 9]
0.34868292682926827
Iteration:  40
[1 3 9 … 3 2 9] [1 6 5 … 3 6 9]
0.4182682926829268
Iteration:  50
[1 6 7 … 3 2 9] [1 6 5 … 3 6 9]
0.47548780487804876
Iteration:  60
[1 6 7 … 3 2 9] [1 6 5 … 3 6 9]
0.5172439024390244
Iteration:  70
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.5547317073170732
Iteration:  80
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.5845121951219512
Iteration:  90
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.6099756097560975
Iteration:  100
```

```
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.633219512195122
Iteration:  110
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.6541707317073171
Iteration:  120
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.6723658536585366
Iteration:  130
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.6876585365853658
Iteration:  140
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7018780487804878
Iteration:  150
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7139268292682927
Iteration:  160
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7236829268292683
Iteration:  170
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7321707317073171
Iteration:  180
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7409024390243902
Iteration:  190
[1 6 7 … 3 6 9] [1 6 5 … 3 6 9]
0.7480487804878049
```

[ ]:

```
def make_predictions(X, W1, b1, W2, b2):
    _, _, _, A2 = forward_prop(W1, b1, W2, b2, X)
    predictions = get_predictions(A2)
    return predictions

def test_prediction(index, W1, b1, W2, b2):
    current_image = X_train[:, index, None]
    prediction = make_predictions(X_train[:, index, None], W1, b1, W2, b2)
    label = Y_train[index]
    print("Prediction: ", prediction)
    print("Label: ", label)

    current_image = current_image.reshape((28, 28)) * 255
    plt.gray()
    plt.imshow(current_image, interpolation='nearest')
    plt.show()
```

```
[ ]: dev_predictions = make_predictions(X_test, W1, b1, W2, b2)
     print("Accuracy: ", get_accuracy(dev_predictions, Y_test))
```

```
[0 6 9 1 7 4 0 3 0 9 3 2 4 1 9 0 0 1 4 5 7 4 0 7 6 2 1 6 9 1 3 0 3 3 9 1 9
 0 0 1 0 9 9 1 4 4 4 0 9 4 0 9 6 0 2 6 2 1 9 9 3 9 1 6 5 6 1 9 6 2 1 8 8 9
 1 6 4 0 2 4 5 4 3 1 5 4 3 1 2 5 2 0 2 5 7 6 0 8 0 3 6 4 1 4 0 9 5 8 6 1 8
 7 8 9 2 4 4 4 6 3 4 7 8 1 5 6 2 1 1 4 8 6 1 8 7 8 8 6 9 9 2 4 0 1 8 9 0 6
 7 2 5 8 3 6 2 8 7 9 8 4 2 7 9 8 4 3 8 1 0 7 6 7 9 3 7 6 6 0 6 1 4 8 2 7 3
 8 7 3 8 3 4 3 5 8 9 4 0 4 3 4 6 4 2 9 7 2 7 0 8 3 1 6 2 1 9 2 8 2 7 7 1 6
 4 4 9 4 6 0 7 0 8 4 6 1 0 1 7 4 8 7 7 8 0 3 6 5 2 5 6 5 7 3 3 3 9 1 9 3 1
 8 3 7 2 2 9 4 1 1 6 2 9 1 6 1 5 8 0 1 9 4 3 2 8 7 7 9 7 1 7 3 7 2 8 4 4 6
 3 7 9 0 0 4 7 2 1 5 4 8 2 6 2 0 1 5 6 0 6 5 8 9 7 1 6 5 3 6 1 7 3 7 6 5 1
 8 4 6 1 8 6 1 5 6 5 0 2 8 7 1 8 7 2 7 9 9 9 5 4 8 2 3 8 2 5 5 8 7 9 4 9 4
 9 6 1 9 7 8 4 7 7 8 1 1 4 9 9 3 9 3 1 9 9 1 0 8 6 7 4 5 9 9 1 1 3 4 4 2 9
 8 2 2 4 3 8 1 2 1 5 2 2 6 7 5 5 3 9 1 1 2 1 0 9 9 6 3 7 3 0 4 5 7 4 7 3 5
 5 7 4 1 3 8 8 4 6 5 6 9 0 3 7 8 8 2 6 4 3 4 3 2 4 2 6 7 0 3 7 6 1 3 0 3 1
 0 7 1 1 2 1 1 3 7 1 3 2 9 6 3 1 7 9 8 9 8 4 8 6 1 0 5 3 0 1 6 2 6 6 4 8 7
 7 2 7 0 7 0 2 8 8 1 3 8 1 5 4 2 8 9 8 7 1 5 8 3 8 1 7 2 7 5 3 7 1 2 1 3 1
 8 4 2 1 8 0 2 2 3 5 7 6 2 0 6 4 6 2 1 7 5 0 0 2 1 6 7 0 9 8 1 4 3 5 6 1 7
 8 8 2 0 7 1 8 2 6 6 3 5 9 0 8 0 9 0 5 2 6 3 8 5 9 1 1 5 0 1 7 9 2 4 1 1 8
 8 6 8 1 0 1 8 7 3 2 9 4 0 3 6 1 9 6 6 9 5 7 5 0 0 8 7 0 8 4 6 9 6 1 3 8 7
 1 8 1 4 9 9 1 3 8 1 8 6 3 0 9 5 5 4 9 1 6 7 1 8 0 6 0 8 8 8 7 7 2 7 2 8 5
 1 4 1 4 7 2 3 8 6 3 0 5 5 6 7 7 4 8 7 4 6 2 9 2 2 4 2 1 2 6 3 1 5 1 5 7 1
 3 7 3 1 6 3 1 5 0 6 4 4 1 4 3 1 7 1 4 3 8 0 4 2 9 7 0 3 5 3 9 4 8 3 9 9 6
 5 5 7 1 5 7 6 9 8 9 3 2 6 2 5 1 8 9 3 2 8 3 3 9 2 3 1 3 6 7 4 8 1 3 1 6 3
 7 9 3 7 2 0 9 8 4 9 7 7 5 1 1 6 8 0 1 1 1 8 1 3 7 8 3 1 3 9 7 0 3 3 6 1 8
 3 5 8 8 8 0 9 0 5 0 9 9 2 0 8 9 8 3 6 3 1 1 7 4 3 1 1 5 0 2 6 0 5 7 0 3 0
 9 6 8 4 8 9 3 7 2 6 6 1 3 9 8 6 8 6 4 0 6 1 2 9 6 0 2 4 4 1 4 8 0 1 9 3 9
 9 3 5 4 0 5 4 9 5 6 7 6 2 8 6 7 2 1 8 4 8 9 0 4 9 9 8 7 7 4 6 1 9 3 6 7 8
 8 6 2 7 4 0 9 8 5 0 3 1 3 2 7 2 4 1 2 4 6 1 7 9 3 0 7 8 0 3 2 9 6 6 9 7 7
 5] [0 6 9 1 5 4 0 5 0 9 3 6 4 1 9 0 0 8 4 5 7 4 0 7 6 2 1 2 9 1 3 0 2 3 9 2 3
 0 6 1 0 9 9 1 2 4 4 0 9 4 0 9 6 0 2 6 2 1 9 8 3 0 1 6 5 6 1 9 6 2 1 5 8 7
 1 6 4 0 6 4 5 4 3 1 5 4 3 1 2 5 2 0 3 3 7 4 0 8 0 3 6 4 1 4 0 7 5 8 6 1 2
 9 3 5 2 4 2 4 6 8 4 7 8 1 6 6 2 1 1 4 6 6 1 2 7 3 7 6 4 9 2 4 0 1 8 9 0 6
 7 2 8 8 3 6 2 7 7 9 8 4 6 7 9 9 7 3 8 1 0 7 6 7 5 3 3 6 6 0 6 1 4 8 2 3 3
 8 7 8 8 3 4 3 3 3 8 9 0 4 5 4 5 4 2 9 7 2 7 0 2 3 1 6 2 2 9 2 8 2 7 7 1 2
 4 4 9 9 6 0 8 0 8 4 6 1 5 1 7 4 3 7 7 2 0 3 5 5 2 5 6 6 7 3 3 3 9 1 9 3 1
 8 3 9 3 3 9 5 1 3 6 2 9 1 6 1 5 4 0 1 9 4 6 2 2 7 2 9 7 1 7 5 5 2 8 4 4 6
 5 7 4 0 0 7 8 6 9 5 4 8 2 6 2 0 1 5 6 5 6 5 2 9 3 1 6 5 3 6 1 7 8 7 6 5 2
 8 4 6 1 8 6 1 5 6 5 7 2 5 7 1 8 7 2 7 9 9 9 8 4 8 8 3 7 2 3 5 5 7 9 4 9 4
 8 6 1 9 7 9 9 3 8 8 1 8 4 4 9 3 9 2 7 9 7 1 0 8 6 7 4 5 9 9 1 1 3 4 7 2 9
 8 6 2 4 5 8 1 3 9 5 6 3 6 7 5 0 3 2 4 1 2 1 0 8 9 6 3 7 3 0 9 0 7 4 7 3 5
 5 7 9 1 3 3 5 4 6 5 6 7 0 3 7 8 4 3 4 4 3 4 3 2 3 2 6 7 0 3 7 6 1 3 0 3 1
 0 7 1 1 5 1 1 6 7 1 3 2 9 7 8 1 7 9 3 5 8 4 8 6 1 0 5 5 0 1 6 2 6 0 4 8 7
 5 2 3 0 9 0 2 9 5 2 3 8 1 5 4 6 7 9 8 7 1 5 8 7 8 1 7 2 7 5 6 7 1 6 1 9 1
 1 2 2 1 8 0 6 2 5 5 3 1 8 0 6 4 8 2 8 5 5 0 0 2 1 6 9 0 5 8 1 9 5 5 2 1 7
 8 8 2 0 7 1 8 2 6 6 3 5 4 0 8 0 4 0 5 2 6 3 8 5 9 1 1 5 0 7 7 9 2 4 1 8 8
```

```
 2 6 8 1 4 1 8 7 3 2 9 4 0 3 6 1 9 6 6 9 5 7 5 0 0 8 9 0 8 4 6 9 2 1 3 5 7
 1 7 1 7 9 9 1 3 8 1 8 6 3 0 8 8 5 4 4 3 6 7 1 8 0 6 0 8 8 8 7 7 2 3 2 8 0
 8 4 1 4 3 2 8 9 6 5 0 5 0 6 7 7 3 6 7 4 5 8 5 1 2 9 7 7 6 6 8 1 8 1 0 7 1
 3 7 5 1 6 3 1 5 0 6 4 8 1 6 3 1 2 1 4 3 5 0 4 0 8 7 0 3 5 3 9 4 8 3 9 9 6
 0 3 7 1 5 7 6 4 3 4 1 2 6 6 0 1 2 7 3 2 8 3 3 9 6 3 1 3 6 7 4 8 1 8 1 6 3
 7 9 3 7 2 0 9 2 4 9 7 7 5 1 1 6 8 0 1 1 1 8 1 3 7 3 3 1 3 9 7 0 3 3 3 1 8
 3 5 8 8 5 0 9 0 5 0 9 9 2 0 8 4 1 5 6 8 1 1 7 4 5 1 1 5 0 2 6 0 5 7 5 3 0
 5 6 8 4 8 5 3 7 2 6 6 1 8 9 3 6 8 6 9 0 4 1 2 4 6 5 2 4 4 1 4 3 0 1 9 3 8
 8 3 5 4 0 5 5 9 2 6 7 6 8 1 3 7 2 1 8 7 5 9 0 4 9 9 5 3 7 4 6 1 4 3 2 5 2
 8 6 2 7 4 0 9 8 5 5 8 1 3 2 7 2 4 1 2 4 6 1 7 7 3 0 7 8 0 2 6 9 6 6 9 7 7
 5]
Accuracy:  0.755
```

2 Keras

[ ]:
```python
import numpy as np
import matplotlib.pyplot as plt

from keras.datasets import mnist
from keras.models import Sequential
from keras.layers import Dense, Dropout, Activation
from tensorflow.keras.utils import to_categorical
```

[ ]:
```python
X_train = X_train.reshape(60000, 784)
X_train = X_train.astype('float32')
X_train /= 255
print("Training matrix shape", X_train.shape)

X_test = X_test.reshape(10000, 784)
X_test = X_test.astype('float32')
X_test /= 255
print("Testing matrix shape", X_test.shape)
```

```
Training matrix shape (60000, 784)
Testing matrix shape (10000, 784)
```

[ ]:
```python
nb_classes = 10
```

[ ]:
```python
Y_train = to_categorical(y_train, nb_classes)
Y_test = to_categorical(y_test, nb_classes)
```

[ ]:
```python
model = Sequential()
model.add(Dense(512, input_shape=(784,)))
model.add(Activation('relu'))
model.add(Dropout(0.2))
model.add(Dense(512))
model.add(Activation('relu'))
```

```
model.add(Dropout(0.2))
model.add(Dense(10))
model.add(Activation('softmax'))
```

```
2024-03-24 23:20:04.733422: I metal_plugin/src/device/metal_device.cc:1154]
Metal device set to: Apple M1
2024-03-24 23:20:04.733463: I metal_plugin/src/device/metal_device.cc:296]
systemMemory: 8.00 GB
2024-03-24 23:20:04.733469: I metal_plugin/src/device/metal_device.cc:313]
maxCacheSize: 2.67 GB
2024-03-24 23:20:04.733746: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:303]
Could not identify NUMA node of platform GPU ID 0, defaulting to 0. Your kernel
may not have been built with NUMA support.
2024-03-24 23:20:04.733776: I
tensorflow/core/common_runtime/pluggable_device/pluggable_device_factory.cc:269]
Created TensorFlow device (/job:localhost/replica:0/task:0/device:GPU:0 with 0
MB memory) -> physical PluggableDevice (device: 0, name: METAL, pci bus id:
<undefined>)
```

[ ]:
```
model.compile(loss = 'categorical_crossentropy', optimizer = 'adam',␣
↪metrics=['accuracy'])
```

[ ]:
```
model.fit(X_train, Y_train, batch_size=128, epochs=200, verbose=1,␣
↪validation_data=(X_test, Y_test))
```

```
Epoch 1/200
```

```
2024-03-24 23:20:12.978669: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2024-03-24 23:20:12.999596: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954]
PluggableGraphOptimizer failed: INVALID_ARGUMENT: Unparseable
tensorflow.GraphDef proto
2024-03-24 23:20:13.009331: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954]
PluggableGraphOptimizer failed: INVALID_ARGUMENT: Unparseable
tensorflow.GraphDef proto
```

```
469/469 [==============================] - ETA: 0s - loss: 0.2509 - accuracy:
0.9248
```

```
2024-03-24 23:20:19.630822: I
tensorflow/core/grappler/optimizers/custom_graph_optimizer_registry.cc:114]
Plugin optimizer for device_type GPU is enabled.
2024-03-24 23:20:19.639395: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954]
PluggableGraphOptimizer failed: INVALID_ARGUMENT: Unparseable
```

```
tensorflow.GraphDef proto
2024-03-24 23:20:19.643379: E
tensorflow/core/grappler/optimizers/meta_optimizer.cc:954]
PluggableGraphOptimizer failed: INVALID_ARGUMENT: Unparseable
tensorflow.GraphDef proto
```

```
469/469 [==============================] - 7s 11ms/step - loss: 0.2509 -
accuracy: 0.9248 - val_loss: 0.1061 - val_accuracy: 0.9683
Epoch 2/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0991 -
accuracy: 0.9697 - val_loss: 0.0793 - val_accuracy: 0.9761
Epoch 3/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0720 -
accuracy: 0.9780 - val_loss: 0.0728 - val_accuracy: 0.9772
Epoch 4/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0552 -
accuracy: 0.9829 - val_loss: 0.0658 - val_accuracy: 0.9810
Epoch 5/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0425 -
accuracy: 0.9861 - val_loss: 0.0600 - val_accuracy: 0.9819
Epoch 6/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0378 -
accuracy: 0.9878 - val_loss: 0.0697 - val_accuracy: 0.9817
Epoch 7/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0339 -
accuracy: 0.9885 - val_loss: 0.0683 - val_accuracy: 0.9815
Epoch 8/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0283 -
accuracy: 0.9905 - val_loss: 0.0631 - val_accuracy: 0.9831
Epoch 9/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0256 -
accuracy: 0.9911 - val_loss: 0.0720 - val_accuracy: 0.9817
Epoch 10/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0245 -
accuracy: 0.9925 - val_loss: 0.0695 - val_accuracy: 0.9813
Epoch 11/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0224 -
accuracy: 0.9924 - val_loss: 0.0747 - val_accuracy: 0.9824
Epoch 12/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0209 -
accuracy: 0.9933 - val_loss: 0.0682 - val_accuracy: 0.9822
Epoch 13/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0184 -
accuracy: 0.9937 - val_loss: 0.0771 - val_accuracy: 0.9814
Epoch 14/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0212 -
accuracy: 0.9930 - val_loss: 0.0831 - val_accuracy: 0.9812
Epoch 15/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0176 -
accuracy: 0.9941 - val_loss: 0.0753 - val_accuracy: 0.9806
Epoch 16/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0176 -
accuracy: 0.9942 - val_loss: 0.0730 - val_accuracy: 0.9822
Epoch 17/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0158 -
accuracy: 0.9946 - val_loss: 0.0792 - val_accuracy: 0.9839
Epoch 18/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0162 -
accuracy: 0.9945 - val_loss: 0.0808 - val_accuracy: 0.9840
Epoch 19/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0127 -
accuracy: 0.9958 - val_loss: 0.0905 - val_accuracy: 0.9816
Epoch 20/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0169 -
accuracy: 0.9945 - val_loss: 0.0881 - val_accuracy: 0.9832
Epoch 21/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0133 -
accuracy: 0.9959 - val_loss: 0.0872 - val_accuracy: 0.9828
Epoch 22/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0121 -
accuracy: 0.9959 - val_loss: 0.0897 - val_accuracy: 0.9822
Epoch 23/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0123 -
accuracy: 0.9960 - val_loss: 0.0806 - val_accuracy: 0.9856
Epoch 24/200
469/469 [==============================] - 5s 12ms/step - loss: 0.0139 -
accuracy: 0.9959 - val_loss: 0.0973 - val_accuracy: 0.9818
Epoch 25/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0120 -
accuracy: 0.9962 - val_loss: 0.0829 - val_accuracy: 0.9848
Epoch 26/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0131 -
accuracy: 0.9961 - val_loss: 0.0893 - val_accuracy: 0.9833
Epoch 27/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0114 -
accuracy: 0.9965 - val_loss: 0.0960 - val_accuracy: 0.9821
Epoch 28/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0110 -
accuracy: 0.9964 - val_loss: 0.0930 - val_accuracy: 0.9851
Epoch 29/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0112 -
accuracy: 0.9964 - val_loss: 0.0858 - val_accuracy: 0.9832
Epoch 30/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0095 -
accuracy: 0.9970 - val_loss: 0.0906 - val_accuracy: 0.9846
Epoch 31/200
```

```
469/469 [==============================] - 4s 10ms/step - loss: 0.0110 -
accuracy: 0.9965 - val_loss: 0.1015 - val_accuracy: 0.9815
Epoch 32/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0138 -
accuracy: 0.9957 - val_loss: 0.1015 - val_accuracy: 0.9833
Epoch 33/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0091 -
accuracy: 0.9973 - val_loss: 0.0973 - val_accuracy: 0.9833
Epoch 34/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0086 -
accuracy: 0.9972 - val_loss: 0.0972 - val_accuracy: 0.9840
Epoch 35/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0089 -
accuracy: 0.9972 - val_loss: 0.1203 - val_accuracy: 0.9827
Epoch 36/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0115 -
accuracy: 0.9966 - val_loss: 0.1080 - val_accuracy: 0.9823
Epoch 37/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0096 -
accuracy: 0.9972 - val_loss: 0.1031 - val_accuracy: 0.9840
Epoch 38/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0089 -
accuracy: 0.9972 - val_loss: 0.1112 - val_accuracy: 0.9826
Epoch 39/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0101 -
accuracy: 0.9968 - val_loss: 0.1165 - val_accuracy: 0.9837
Epoch 40/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0081 -
accuracy: 0.9976 - val_loss: 0.1133 - val_accuracy: 0.9826
Epoch 41/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0100 -
accuracy: 0.9971 - val_loss: 0.1087 - val_accuracy: 0.9828
Epoch 42/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0084 -
accuracy: 0.9974 - val_loss: 0.1193 - val_accuracy: 0.9825
Epoch 43/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0099 -
accuracy: 0.9972 - val_loss: 0.1205 - val_accuracy: 0.9821
Epoch 44/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0084 -
accuracy: 0.9977 - val_loss: 0.1254 - val_accuracy: 0.9830
Epoch 45/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0104 -
accuracy: 0.9971 - val_loss: 0.1144 - val_accuracy: 0.9845
Epoch 46/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0090 -
accuracy: 0.9977 - val_loss: 0.1281 - val_accuracy: 0.9830
Epoch 47/200
```

```
469/469 [==============================] - 4s 10ms/step - loss: 0.0122 -
accuracy: 0.9968 - val_loss: 0.1108 - val_accuracy: 0.9836
Epoch 48/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0090 -
accuracy: 0.9977 - val_loss: 0.1129 - val_accuracy: 0.9839
Epoch 49/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0060 -
accuracy: 0.9983 - val_loss: 0.1215 - val_accuracy: 0.9833
Epoch 50/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0069 -
accuracy: 0.9980 - val_loss: 0.1203 - val_accuracy: 0.9831
Epoch 51/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0088 -
accuracy: 0.9974 - val_loss: 0.1324 - val_accuracy: 0.9831
Epoch 52/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0108 -
accuracy: 0.9972 - val_loss: 0.1160 - val_accuracy: 0.9847
Epoch 53/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0085 -
accuracy: 0.9976 - val_loss: 0.1365 - val_accuracy: 0.9827
Epoch 54/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0105 -
accuracy: 0.9973 - val_loss: 0.1117 - val_accuracy: 0.9842
Epoch 55/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0079 -
accuracy: 0.9977 - val_loss: 0.1189 - val_accuracy: 0.9865
Epoch 56/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0064 -
accuracy: 0.9982 - val_loss: 0.1278 - val_accuracy: 0.9844
Epoch 57/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0076 -
accuracy: 0.9977 - val_loss: 0.1258 - val_accuracy: 0.9851
Epoch 58/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0111 -
accuracy: 0.9969 - val_loss: 0.1257 - val_accuracy: 0.9835
Epoch 59/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0072 -
accuracy: 0.9979 - val_loss: 0.1196 - val_accuracy: 0.9845
Epoch 60/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0065 -
accuracy: 0.9981 - val_loss: 0.1261 - val_accuracy: 0.9849
Epoch 61/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0059 -
accuracy: 0.9983 - val_loss: 0.1380 - val_accuracy: 0.9837
Epoch 62/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0080 -
accuracy: 0.9979 - val_loss: 0.1313 - val_accuracy: 0.9855
Epoch 63/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0079 -
accuracy: 0.9979 - val_loss: 0.1384 - val_accuracy: 0.9847
Epoch 64/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0075 -
accuracy: 0.9983 - val_loss: 0.1436 - val_accuracy: 0.9837
Epoch 65/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0099 -
accuracy: 0.9974 - val_loss: 0.1437 - val_accuracy: 0.9829
Epoch 66/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0082 -
accuracy: 0.9981 - val_loss: 0.1306 - val_accuracy: 0.9842
Epoch 67/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0089 -
accuracy: 0.9977 - val_loss: 0.1419 - val_accuracy: 0.9839
Epoch 68/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0095 -
accuracy: 0.9976 - val_loss: 0.1223 - val_accuracy: 0.9853
Epoch 69/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0061 -
accuracy: 0.9981 - val_loss: 0.1254 - val_accuracy: 0.9851
Epoch 70/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0070 -
accuracy: 0.9983 - val_loss: 0.1164 - val_accuracy: 0.9858
Epoch 71/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0049 -
accuracy: 0.9987 - val_loss: 0.1439 - val_accuracy: 0.9825
Epoch 72/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0079 -
accuracy: 0.9981 - val_loss: 0.1600 - val_accuracy: 0.9832
Epoch 73/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0089 -
accuracy: 0.9978 - val_loss: 0.1364 - val_accuracy: 0.9853
Epoch 74/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0086 -
accuracy: 0.9978 - val_loss: 0.1507 - val_accuracy: 0.9824
Epoch 75/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0072 -
accuracy: 0.9980 - val_loss: 0.1384 - val_accuracy: 0.9849
Epoch 76/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0060 -
accuracy: 0.9983 - val_loss: 0.1242 - val_accuracy: 0.9858
Epoch 77/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0096 -
accuracy: 0.9977 - val_loss: 0.1312 - val_accuracy: 0.9863
Epoch 78/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0057 -
accuracy: 0.9983 - val_loss: 0.1310 - val_accuracy: 0.9846
Epoch 79/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0076 -
accuracy: 0.9981 - val_loss: 0.1391 - val_accuracy: 0.9857
Epoch 80/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0062 -
accuracy: 0.9981 - val_loss: 0.1282 - val_accuracy: 0.9868
Epoch 81/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0074 -
accuracy: 0.9982 - val_loss: 0.1502 - val_accuracy: 0.9840
Epoch 82/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0073 -
accuracy: 0.9985 - val_loss: 0.1566 - val_accuracy: 0.9846
Epoch 83/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0085 -
accuracy: 0.9978 - val_loss: 0.1536 - val_accuracy: 0.9839
Epoch 84/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0065 -
accuracy: 0.9983 - val_loss: 0.1469 - val_accuracy: 0.9852
Epoch 85/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0069 -
accuracy: 0.9982 - val_loss: 0.1671 - val_accuracy: 0.9839
Epoch 86/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0126 -
accuracy: 0.9976 - val_loss: 0.1629 - val_accuracy: 0.9842
Epoch 87/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0067 -
accuracy: 0.9985 - val_loss: 0.1362 - val_accuracy: 0.9862
Epoch 88/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0073 -
accuracy: 0.9983 - val_loss: 0.1449 - val_accuracy: 0.9853
Epoch 89/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0088 -
accuracy: 0.9982 - val_loss: 0.1468 - val_accuracy: 0.9843
Epoch 90/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0058 -
accuracy: 0.9983 - val_loss: 0.1568 - val_accuracy: 0.9857
Epoch 91/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0055 -
accuracy: 0.9986 - val_loss: 0.1458 - val_accuracy: 0.9866
Epoch 92/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0062 -
accuracy: 0.9985 - val_loss: 0.1577 - val_accuracy: 0.9836
Epoch 93/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0076 -
accuracy: 0.9983 - val_loss: 0.1501 - val_accuracy: 0.9850
Epoch 94/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0077 -
accuracy: 0.9983 - val_loss: 0.1503 - val_accuracy: 0.9840
Epoch 95/200
```

```
469/469 [==============================] - 5s 10ms/step - loss: 0.0061 -
accuracy: 0.9985 - val_loss: 0.1551 - val_accuracy: 0.9859
Epoch 96/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0084 -
accuracy: 0.9979 - val_loss: 0.1586 - val_accuracy: 0.9856
Epoch 97/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0065 -
accuracy: 0.9986 - val_loss: 0.1536 - val_accuracy: 0.9848
Epoch 98/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0086 -
accuracy: 0.9981 - val_loss: 0.1658 - val_accuracy: 0.9854
Epoch 99/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0073 -
accuracy: 0.9984 - val_loss: 0.1553 - val_accuracy: 0.9850
Epoch 100/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0061 -
accuracy: 0.9986 - val_loss: 0.1464 - val_accuracy: 0.9860
Epoch 101/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0069 -
accuracy: 0.9984 - val_loss: 0.1498 - val_accuracy: 0.9853
Epoch 102/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0069 -
accuracy: 0.9981 - val_loss: 0.1544 - val_accuracy: 0.9854
Epoch 103/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0069 -
accuracy: 0.9984 - val_loss: 0.1850 - val_accuracy: 0.9850
Epoch 104/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0094 -
accuracy: 0.9980 - val_loss: 0.1694 - val_accuracy: 0.9847
Epoch 105/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0073 -
accuracy: 0.9984 - val_loss: 0.1678 - val_accuracy: 0.9858
Epoch 106/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0078 -
accuracy: 0.9983 - val_loss: 0.1614 - val_accuracy: 0.9851
Epoch 107/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0077 -
accuracy: 0.9984 - val_loss: 0.1862 - val_accuracy: 0.9832
Epoch 108/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0054 -
accuracy: 0.9986 - val_loss: 0.1674 - val_accuracy: 0.9844
Epoch 109/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0044 -
accuracy: 0.9989 - val_loss: 0.1623 - val_accuracy: 0.9868
Epoch 110/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0066 -
accuracy: 0.9987 - val_loss: 0.1915 - val_accuracy: 0.9843
Epoch 111/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0063 -
accuracy: 0.9984 - val_loss: 0.2185 - val_accuracy: 0.9815
Epoch 112/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0078 -
accuracy: 0.9982 - val_loss: 0.1759 - val_accuracy: 0.9845
Epoch 113/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0062 -
accuracy: 0.9985 - val_loss: 0.1962 - val_accuracy: 0.9845
Epoch 114/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0047 -
accuracy: 0.9991 - val_loss: 0.1654 - val_accuracy: 0.9863
Epoch 115/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0052 -
accuracy: 0.9989 - val_loss: 0.1715 - val_accuracy: 0.9875
Epoch 116/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0081 -
accuracy: 0.9984 - val_loss: 0.1999 - val_accuracy: 0.9849
Epoch 117/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0096 -
accuracy: 0.9982 - val_loss: 0.1833 - val_accuracy: 0.9853
Epoch 118/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0058 -
accuracy: 0.9986 - val_loss: 0.1944 - val_accuracy: 0.9844
Epoch 119/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0097 -
accuracy: 0.9982 - val_loss: 0.1733 - val_accuracy: 0.9854
Epoch 120/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0062 -
accuracy: 0.9989 - val_loss: 0.1860 - val_accuracy: 0.9857
Epoch 121/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0077 -
accuracy: 0.9987 - val_loss: 0.1818 - val_accuracy: 0.9855
Epoch 122/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0067 -
accuracy: 0.9985 - val_loss: 0.1857 - val_accuracy: 0.9858
Epoch 123/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0083 -
accuracy: 0.9984 - val_loss: 0.1858 - val_accuracy: 0.9870
Epoch 124/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0073 -
accuracy: 0.9986 - val_loss: 0.2034 - val_accuracy: 0.9852
Epoch 125/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0077 -
accuracy: 0.9983 - val_loss: 0.1767 - val_accuracy: 0.9864
Epoch 126/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0069 -
accuracy: 0.9988 - val_loss: 0.1940 - val_accuracy: 0.9851
Epoch 127/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0082 -
accuracy: 0.9986 - val_loss: 0.1752 - val_accuracy: 0.9862
Epoch 128/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0044 -
accuracy: 0.9991 - val_loss: 0.1758 - val_accuracy: 0.9852
Epoch 129/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0076 -
accuracy: 0.9986 - val_loss: 0.1832 - val_accuracy: 0.9871
Epoch 130/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0088 -
accuracy: 0.9984 - val_loss: 0.1829 - val_accuracy: 0.9852
Epoch 131/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0071 -
accuracy: 0.9987 - val_loss: 0.1709 - val_accuracy: 0.9861
Epoch 132/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0072 -
accuracy: 0.9988 - val_loss: 0.1859 - val_accuracy: 0.9841
Epoch 133/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0068 -
accuracy: 0.9986 - val_loss: 0.1915 - val_accuracy: 0.9854
Epoch 134/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0071 -
accuracy: 0.9986 - val_loss: 0.2025 - val_accuracy: 0.9851
Epoch 135/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0056 -
accuracy: 0.9987 - val_loss: 0.2029 - val_accuracy: 0.9839
Epoch 136/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0051 -
accuracy: 0.9988 - val_loss: 0.1947 - val_accuracy: 0.9852
Epoch 137/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0071 -
accuracy: 0.9988 - val_loss: 0.2144 - val_accuracy: 0.9857
Epoch 138/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0066 -
accuracy: 0.9987 - val_loss: 0.2163 - val_accuracy: 0.9841
Epoch 139/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0095 -
accuracy: 0.9981 - val_loss: 0.2474 - val_accuracy: 0.9836
Epoch 140/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0064 -
accuracy: 0.9988 - val_loss: 0.2026 - val_accuracy: 0.9857
Epoch 141/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0086 -
accuracy: 0.9984 - val_loss: 0.2084 - val_accuracy: 0.9847
Epoch 142/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0057 -
accuracy: 0.9989 - val_loss: 0.1842 - val_accuracy: 0.9873
Epoch 143/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0055 -
accuracy: 0.9990 - val_loss: 0.2150 - val_accuracy: 0.9835
Epoch 144/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0069 -
accuracy: 0.9988 - val_loss: 0.1894 - val_accuracy: 0.9868
Epoch 145/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0092 -
accuracy: 0.9984 - val_loss: 0.1913 - val_accuracy: 0.9855
Epoch 146/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0062 -
accuracy: 0.9989 - val_loss: 0.2001 - val_accuracy: 0.9861
Epoch 147/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0043 -
accuracy: 0.9992 - val_loss: 0.2118 - val_accuracy: 0.9853
Epoch 148/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0058 -
accuracy: 0.9988 - val_loss: 0.2361 - val_accuracy: 0.9840
Epoch 149/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0078 -
accuracy: 0.9988 - val_loss: 0.1917 - val_accuracy: 0.9860
Epoch 150/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0074 -
accuracy: 0.9988 - val_loss: 0.2244 - val_accuracy: 0.9841
Epoch 151/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0098 -
accuracy: 0.9984 - val_loss: 0.2284 - val_accuracy: 0.9855
Epoch 152/200
469/469 [==============================] - 5s 11ms/step - loss: 0.0061 -
accuracy: 0.9990 - val_loss: 0.2202 - val_accuracy: 0.9847
Epoch 153/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0083 -
accuracy: 0.9986 - val_loss: 0.2072 - val_accuracy: 0.9850
Epoch 154/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0060 -
accuracy: 0.9988 - val_loss: 0.2106 - val_accuracy: 0.9845
Epoch 155/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0060 -
accuracy: 0.9989 - val_loss: 0.2175 - val_accuracy: 0.9852
Epoch 156/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0045 -
accuracy: 0.9992 - val_loss: 0.2200 - val_accuracy: 0.9851
Epoch 157/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0082 -
accuracy: 0.9985 - val_loss: 0.2377 - val_accuracy: 0.9859
Epoch 158/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0063 -
accuracy: 0.9991 - val_loss: 0.2145 - val_accuracy: 0.9869
Epoch 159/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0052 -
accuracy: 0.9990 - val_loss: 0.2212 - val_accuracy: 0.9861
Epoch 160/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0058 -
accuracy: 0.9989 - val_loss: 0.2478 - val_accuracy: 0.9850
Epoch 161/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0064 -
accuracy: 0.9989 - val_loss: 0.2486 - val_accuracy: 0.9834
Epoch 162/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0097 -
accuracy: 0.9986 - val_loss: 0.2469 - val_accuracy: 0.9853
Epoch 163/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0065 -
accuracy: 0.9987 - val_loss: 0.2381 - val_accuracy: 0.9857
Epoch 164/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0046 -
accuracy: 0.9990 - val_loss: 0.2669 - val_accuracy: 0.9843
Epoch 165/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0061 -
accuracy: 0.9988 - val_loss: 0.2538 - val_accuracy: 0.9840
Epoch 166/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0067 -
accuracy: 0.9989 - val_loss: 0.3067 - val_accuracy: 0.9832
Epoch 167/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0058 -
accuracy: 0.9989 - val_loss: 0.2565 - val_accuracy: 0.9848
Epoch 168/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0087 -
accuracy: 0.9985 - val_loss: 0.2951 - val_accuracy: 0.9831
Epoch 169/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0090 -
accuracy: 0.9985 - val_loss: 0.2623 - val_accuracy: 0.9848
Epoch 170/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0061 -
accuracy: 0.9990 - val_loss: 0.2390 - val_accuracy: 0.9858
Epoch 171/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0064 -
accuracy: 0.9991 - val_loss: 0.2386 - val_accuracy: 0.9854
Epoch 172/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0076 -
accuracy: 0.9988 - val_loss: 0.2355 - val_accuracy: 0.9851
Epoch 173/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0060 -
accuracy: 0.9989 - val_loss: 0.2466 - val_accuracy: 0.9839
Epoch 174/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0058 -
accuracy: 0.9988 - val_loss: 0.2201 - val_accuracy: 0.9850
Epoch 175/200
```

```
469/469 [==============================] - 4s 10ms/step - loss: 0.0070 -
accuracy: 0.9988 - val_loss: 0.2104 - val_accuracy: 0.9858
Epoch 176/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0056 -
accuracy: 0.9990 - val_loss: 0.2486 - val_accuracy: 0.9856
Epoch 177/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0061 -
accuracy: 0.9989 - val_loss: 0.2632 - val_accuracy: 0.9852
Epoch 178/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0084 -
accuracy: 0.9988 - val_loss: 0.2543 - val_accuracy: 0.9852
Epoch 179/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0095 -
accuracy: 0.9985 - val_loss: 0.2730 - val_accuracy: 0.9848
Epoch 180/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0069 -
accuracy: 0.9988 - val_loss: 0.2765 - val_accuracy: 0.9848
Epoch 181/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0068 -
accuracy: 0.9989 - val_loss: 0.2314 - val_accuracy: 0.9858
Epoch 182/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0064 -
accuracy: 0.9989 - val_loss: 0.2613 - val_accuracy: 0.9842
Epoch 183/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0054 -
accuracy: 0.9992 - val_loss: 0.2415 - val_accuracy: 0.9857
Epoch 184/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0059 -
accuracy: 0.9988 - val_loss: 0.3257 - val_accuracy: 0.9832
Epoch 185/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0063 -
accuracy: 0.9990 - val_loss: 0.3103 - val_accuracy: 0.9832
Epoch 186/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0049 -
accuracy: 0.9993 - val_loss: 0.3172 - val_accuracy: 0.9831
Epoch 187/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0084 -
accuracy: 0.9988 - val_loss: 0.2831 - val_accuracy: 0.9845
Epoch 188/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0047 -
accuracy: 0.9991 - val_loss: 0.2947 - val_accuracy: 0.9844
Epoch 189/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0055 -
accuracy: 0.9991 - val_loss: 0.3230 - val_accuracy: 0.9844
Epoch 190/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0066 -
accuracy: 0.9991 - val_loss: 0.3165 - val_accuracy: 0.9814
Epoch 191/200
```

```
469/469 [==============================] - 4s 9ms/step - loss: 0.0077 -
accuracy: 0.9990 - val_loss: 0.2710 - val_accuracy: 0.9851
Epoch 192/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0082 -
accuracy: 0.9989 - val_loss: 0.2647 - val_accuracy: 0.9863
Epoch 193/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0092 -
accuracy: 0.9987 - val_loss: 0.2570 - val_accuracy: 0.9858
Epoch 194/200
469/469 [==============================] - 4s 9ms/step - loss: 0.0066 -
accuracy: 0.9990 - val_loss: 0.2892 - val_accuracy: 0.9835
Epoch 195/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0071 -
accuracy: 0.9990 - val_loss: 0.2651 - val_accuracy: 0.9849
Epoch 196/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0037 -
accuracy: 0.9991 - val_loss: 0.2600 - val_accuracy: 0.9854
Epoch 197/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0045 -
accuracy: 0.9991 - val_loss: 0.2455 - val_accuracy: 0.9857
Epoch 198/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0077 -
accuracy: 0.9991 - val_loss: 0.2912 - val_accuracy: 0.9849
Epoch 199/200
469/469 [==============================] - 5s 10ms/step - loss: 0.0063 -
accuracy: 0.9990 - val_loss: 0.3127 - val_accuracy: 0.9854
Epoch 200/200
469/469 [==============================] - 4s 10ms/step - loss: 0.0025 -
accuracy: 0.9995 - val_loss: 0.2548 - val_accuracy: 0.9850
```

```
[ ]: <keras.src.callbacks.History at 0x28f032b30>
```

```
[ ]: score = model.evaluate(X_test, Y_test, verbose=0)
     print('Test score:', score[0])
     print('Test accuracy:', score[1])
```

```
Test score: 0.25479933619499207
Test accuracy: 0.9850000739097595
```

In conclusion, while the custom implementation achieves 75% accuracy and the Keras implementation excels with 98%, both models demonstrate notable performance. On testing the models I found that despite the accuracy gap, the custom implementation proves effective and comparable to Keras.