```python
# First Fully Connected layer
model.add(Dense(33, activation='sigmoid', kernel_regularizer=l2(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 8s - 10ms/step - accuracy: 0.7811 - loss: 7.1333 - val_accuracy: 0.8586 - val_loss: 1.4247
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.8186 - loss: 1.1276 - val_accuracy: 0.8697 - val_loss: 0.9551
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8191 - loss: 1.0578 - val_accuracy: 0.8480 - val_loss: 0.9634
Epoch 4/10
750/750 - 4s - 5ms/step - accuracy: 0.8207 - loss: 1.0292 - val_accuracy: 0.8455 - val_loss: 0.9523
Epoch 5/10
750/750 - 5s - 6ms/step - accuracy: 0.8194 - loss: 1.0042 - val_accuracy: 0.8702 - val_loss: 0.8523
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.8205 - loss: 0.9919 - val_accuracy: 0.8341 - val_loss: 0.9433
Epoch 7/10
750/750 - 6s - 8ms/step - accuracy: 0.8201 - loss: 0.9766 - val_accuracy: 0.8609 - val_loss: 0.9045
Epoch 8/10
750/750 - 3s - 4ms/step - accuracy: 0.8191 - loss: 0.9717 - val_accuracy: 0.8361 - val_loss: 0.9201
Epoch 9/10
750/750 - 5s - 6ms/step - accuracy: 0.8210 - loss: 0.9625 - val_accuracy: 0.8540 - val_loss: 0.8487
Epoch 10/10
750/750 - 7s - 9ms/step - accuracy: 0.8217 - loss: 0.9543 - val_accuracy: 0.8871 - val_loss: 0.7656
Test Accuracy: 88.54%
```

1

88.54

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.1)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 7s - 10ms/step - accuracy: 0.8108 - loss: 7.1752 - val_accuracy: 0.8890 - val_loss: 1.1166
Epoch 2/10
750/750 - 5s - 7ms/step - accuracy: 0.8769 - loss: 0.9496 - val_accuracy: 0.8871 - val_loss: 0.8878
Epoch 3/10
750/750 - 4s - 5ms/step - accuracy: 0.8790 - loss: 0.8869 - val_accuracy: 0.8927 - val_loss: 0.8535
Epoch 4/10
750/750 - 4s - 5ms/step - accuracy: 0.8776 - loss: 0.8785 - val_accuracy: 0.9062 - val_loss: 0.7948
Epoch 5/10
750/750 - 3s - 4ms/step - accuracy: 0.8800 - loss: 0.8387 - val_accuracy: 0.9138 - val_loss: 0.7392
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.8787 - loss: 0.8472 - val_accuracy: 0.9164 - val_loss: 0.7441
Epoch 7/10
750/750 - 4s - 5ms/step - accuracy: 0.8772 - loss: 0.8424 - val_accuracy: 0.9190 - val_loss: 0.6921
Epoch 8/10
750/750 - 5s - 6ms/step - accuracy: 0.8811 - loss: 0.8246 - val_accuracy: 0.8904 - val_loss: 0.7516
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.8778 - loss: 0.8293 - val_accuracy: 0.9212 - val_loss: 0.7183
Epoch 10/10
750/750 - 7s - 9ms/step - accuracy: 0.8783 - loss: 0.8228 - val_accuracy: 0.9153 - val_loss: 0.6984
Test Accuracy: 91.15%
```

2

91.15

```python
# First Fully Connected layer
model.add(Dense(33, activation='sigmoid', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Third Fully Connected layer
model.add(Dense(51, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 6s - 8ms/step - accuracy: 0.7584 - loss: 2.4823 - val_accuracy: 0.8861 - val_loss: 1.0551
Epoch 2/10
750/750 - 6s - 7ms/step - accuracy: 0.8258 - loss: 0.9071 - val_accuracy: 0.8831 - val_loss: 0.6768
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8360 - loss: 0.8153 - val_accuracy: 0.8972 - val_loss: 0.6015
Epoch 4/10
750/750 - 5s - 7ms/step - accuracy: 0.8389 - loss: 0.7877 - val_accuracy: 0.8896 - val_loss: 0.6134
Epoch 5/10
750/750 - 4s - 6ms/step - accuracy: 0.8453 - loss: 0.7610 - val_accuracy: 0.8994 - val_loss: 0.5779
Epoch 6/10
750/750 - 4s - 5ms/step - accuracy: 0.8479 - loss: 0.7434 - val_accuracy: 0.8997 - val_loss: 0.5594
Epoch 7/10
750/750 - 3s - 4ms/step - accuracy: 0.8509 - loss: 0.7328 - val_accuracy: 0.9010 - val_loss: 0.5647
Epoch 8/10
750/750 - 3s - 4ms/step - accuracy: 0.8543 - loss: 0.7232 - val_accuracy: 0.9170 - val_loss: 0.5342
Epoch 9/10
750/750 - 4s - 6ms/step - accuracy: 0.8564 - loss: 0.7135 - val_accuracy: 0.9032 - val_loss: 0.5668
Epoch 10/10
750/750 - 4s - 5ms/step - accuracy: 0.8576 - loss: 0.7092 - val_accuracy: 0.9154 - val_loss: 0.5239
Test Accuracy: 91.54%
```

3

91.54

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 7s - 9ms/step - accuracy: 0.7099 - loss: 2.9934 - val_accuracy: 0.9092 - val_loss: 1.0498
Epoch 2/10
750/750 - 5s - 6ms/step - accuracy: 0.8683 - loss: 0.8667 - val_accuracy: 0.9290 - val_loss: 0.5676
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8846 - loss: 0.7046 - val_accuracy: 0.9394 - val_loss: 0.4801
Epoch 4/10
750/750 - 3s - 4ms/step - accuracy: 0.8903 - loss: 0.6488 - val_accuracy: 0.9398 - val_loss: 0.4498
Epoch 5/10
750/750 - 6s - 9ms/step - accuracy: 0.8938 - loss: 0.6266 - val_accuracy: 0.9373 - val_loss: 0.4521
Epoch 6/10
750/750 - 4s - 5ms/step - accuracy: 0.8994 - loss: 0.6055 - val_accuracy: 0.9415 - val_loss: 0.4341
Epoch 7/10
750/750 - 5s - 7ms/step - accuracy: 0.8984 - loss: 0.6109 - val_accuracy: 0.9373 - val_loss: 0.4393
Epoch 8/10
750/750 - 4s - 6ms/step - accuracy: 0.8995 - loss: 0.5914 - val_accuracy: 0.9434 - val_loss: 0.4249
Epoch 9/10
750/750 - 4s - 5ms/step - accuracy: 0.9008 - loss: 0.5918 - val_accuracy: 0.9487 - val_loss: 0.4008
Epoch 10/10
750/750 - 3s - 4ms/step - accuracy: 0.9034 - loss: 0.5747 - val_accuracy: 0.9453 - val_loss: 0.4118
Test Accuracy: 94.17%
```

4

94.17

```
 First Fully Connected layer
odel.add(Dense(33, activation='sigmoid', kernel_regularizer=l2(0.001)))
odel.add(BatchNormalization())
odel.add(Dropout(0.2))

 Second Fully Connected layer
odel.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.001)))
odel.add(BatchNormalization())
odel.add(Dropout(0.2))

 Third Fully Connected layer
odel.add(Dense(51, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
odel.add(BatchNormalization())
odel.add(Dropout(0.2))

 Fourth Fully Connected layer
odel.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
odel.add(BatchNormalization())
odel.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 8s - 10ms/step - accuracy: 0.7760 - loss: 1.1440 - val_accuracy: 0.9047 - val_loss: 0.6563
Epoch 2/10
750/750 - 5s - 6ms/step - accuracy: 0.8571 - loss: 0.7627 - val_accuracy: 0.9183 - val_loss: 0.5162
Epoch 3/10
750/750 - 6s - 8ms/step - accuracy: 0.8730 - loss: 0.6393 - val_accuracy: 0.9269 - val_loss: 0.4282
Epoch 4/10
750/750 - 4s - 5ms/step - accuracy: 0.8808 - loss: 0.5685 - val_accuracy: 0.9287 - val_loss: 0.3929
Epoch 5/10
750/750 - 3s - 4ms/step - accuracy: 0.8881 - loss: 0.5263 - val_accuracy: 0.9335 - val_loss: 0.3712
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.8928 - loss: 0.5012 - val_accuracy: 0.9420 - val_loss: 0.3289
Epoch 7/10
750/750 - 4s - 5ms/step - accuracy: 0.8969 - loss: 0.4770 - val_accuracy: 0.9467 - val_loss: 0.3222
Epoch 8/10
750/750 - 4s - 5ms/step - accuracy: 0.9004 - loss: 0.4621 - val_accuracy: 0.9460 - val_loss: 0.3122
Epoch 9/10
750/750 - 5s - 7ms/step - accuracy: 0.9032 - loss: 0.4545 - val_accuracy: 0.9441 - val_loss: 0.3132
Epoch 10/10
750/750 - 5s - 6ms/step - accuracy: 0.9062 - loss: 0.4469 - val_accuracy: 0.9523 - val_loss: 0.2927
Test Accuracy: 94.82%
```

5

94.82

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 8s - 10ms/step - accuracy: 0.7195 - loss: 1.3366 - val_accuracy: 0.9173 - val_loss: 0.6445
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.8748 - loss: 0.7541 - val_accuracy: 0.9284 - val_loss: 0.4903
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8933 - loss: 0.5985 - val_accuracy: 0.9403 - val_loss: 0.3890
Epoch 4/10
750/750 - 3s - 4ms/step - accuracy: 0.9040 - loss: 0.5068 - val_accuracy: 0.9453 - val_loss: 0.3353
Epoch 5/10
750/750 - 4s - 5ms/step - accuracy: 0.9103 - loss: 0.4585 - val_accuracy: 0.9503 - val_loss: 0.2986
Epoch 6/10
750/750 - 5s - 6ms/step - accuracy: 0.9156 - loss: 0.4255 - val_accuracy: 0.9526 - val_loss: 0.2775
Epoch 7/10
750/750 - 5s - 7ms/step - accuracy: 0.9180 - loss: 0.4074 - val_accuracy: 0.9519 - val_loss: 0.2721
Epoch 8/10
750/750 - 4s - 6ms/step - accuracy: 0.9205 - loss: 0.3932 - val_accuracy: 0.9532 - val_loss: 0.2705
Epoch 9/10
750/750 - 6s - 7ms/step - accuracy: 0.9240 - loss: 0.3816 - val_accuracy: 0.9537 - val_loss: 0.2630
Epoch 10/10
750/750 - 5s - 6ms/step - accuracy: 0.9236 - loss: 0.3797 - val_accuracy: 0.9544 - val_loss: 0.2664
Test Accuracy: 95.38%
```

6

95.38

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 8s - 10ms/step - accuracy: 0.7442 - loss: 1.0922 - val_accuracy: 0.9240 - val_loss: 0.4835
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.8808 - loss: 0.6150 - val_accuracy: 0.9358 - val_loss: 0.3839
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8941 - loss: 0.5166 - val_accuracy: 0.9438 - val_loss: 0.3191
Epoch 4/10
750/750 - 6s - 8ms/step - accuracy: 0.9056 - loss: 0.4498 - val_accuracy: 0.9427 - val_loss: 0.3021
Epoch 5/10
750/750 - 3s - 4ms/step - accuracy: 0.9097 - loss: 0.4230 - val_accuracy: 0.9481 - val_loss: 0.2770
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.9134 - loss: 0.3993 - val_accuracy: 0.9538 - val_loss: 0.2566
Epoch 7/10
750/750 - 6s - 8ms/step - accuracy: 0.9151 - loss: 0.3901 - val_accuracy: 0.9517 - val_loss: 0.2595
Epoch 8/10
750/750 - 5s - 6ms/step - accuracy: 0.9181 - loss: 0.3795 - val_accuracy: 0.9550 - val_loss: 0.2542
Epoch 9/10
750/750 - 5s - 7ms/step - accuracy: 0.9217 - loss: 0.3641 - val_accuracy: 0.9542 - val_loss: 0.2498
Epoch 10/10
750/750 - 4s - 5ms/step - accuracy: 0.9226 - loss: 0.3598 - val_accuracy: 0.9527 - val_loss: 0.2531
Test Accuracy: 95.37%
```

7

95.37

```
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.001)))
model.add(BatchNormalization())
model.add(Dropout(0.2))
```

```
Epoch 1/10
750/750 - 8s - 11ms/step - accuracy: 0.7122 - loss: 1.2877 - val_accuracy: 0.9206 - val_loss: 0.5781
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.8741 - loss: 0.6944 - val_accuracy: 0.9320 - val_loss: 0.4178
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.8933 - loss: 0.5394 - val_accuracy: 0.9419 - val_loss: 0.3280
Epoch 4/10
750/750 - 4s - 5ms/step - accuracy: 0.9033 - loss: 0.4565 - val_accuracy: 0.9457 - val_loss: 0.2780
Epoch 5/10
750/750 - 3s - 4ms/step - accuracy: 0.9082 - loss: 0.4136 - val_accuracy: 0.9495 - val_loss: 0.2536
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.9143 - loss: 0.3860 - val_accuracy: 0.9536 - val_loss: 0.2367
Epoch 7/10
750/750 - 3s - 4ms/step - accuracy: 0.9182 - loss: 0.3671 - val_accuracy: 0.9542 - val_loss: 0.2309
Epoch 8/10
750/750 - 4s - 5ms/step - accuracy: 0.9193 - loss: 0.3580 - val_accuracy: 0.9553 - val_loss: 0.2279
Epoch 9/10
750/750 - 3s - 5ms/step - accuracy: 0.9216 - loss: 0.3452 - val_accuracy: 0.9548 - val_loss: 0.2291
Epoch 10/10
750/750 - 5s - 6ms/step - accuracy: 0.9250 - loss: 0.3390 - val_accuracy: 0.9592 - val_loss: 0.2082
Test Accuracy: 95.59%
```

8

95.59

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.01)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 7s - 10ms/step - accuracy: 0.7895 - loss: 2.0829 - val_accuracy: 0.9259 - val_loss: 0.5802
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.8982 - loss: 0.5288 - val_accuracy: 0.9396 - val_loss: 0.3456
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.9149 - loss: 0.4275 - val_accuracy: 0.9409 - val_loss: 0.3145
Epoch 4/10
750/750 - 6s - 8ms/step - accuracy: 0.9257 - loss: 0.3761 - val_accuracy: 0.9502 - val_loss: 0.2830
Epoch 5/10
750/750 - 4s - 6ms/step - accuracy: 0.9286 - loss: 0.3559 - val_accuracy: 0.9524 - val_loss: 0.2741
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.9333 - loss: 0.3394 - val_accuracy: 0.9580 - val_loss: 0.2543
Epoch 7/10
750/750 - 4s - 6ms/step - accuracy: 0.9370 - loss: 0.3198 - val_accuracy: 0.9566 - val_loss: 0.2466
Epoch 8/10
750/750 - 4s - 5ms/step - accuracy: 0.9404 - loss: 0.3155 - val_accuracy: 0.9586 - val_loss: 0.2392
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.9419 - loss: 0.3060 - val_accuracy: 0.9571 - val_loss: 0.2494
Epoch 10/10
750/750 - 6s - 8ms/step - accuracy: 0.9444 - loss: 0.2957 - val_accuracy: 0.9574 - val_loss: 0.2492
Test Accuracy: 95.72%
```

9

95.72

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 7s - 9ms/step - accuracy: 0.8281 - loss: 0.6236 - val_accuracy: 0.9334 - val_loss: 0.2746
Epoch 2/10
750/750 - 6s - 8ms/step - accuracy: 0.9118 - loss: 0.3516 - val_accuracy: 0.9482 - val_loss: 0.2214
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.9267 - loss: 0.2905 - val_accuracy: 0.9552 - val_loss: 0.1953
Epoch 4/10
750/750 - 3s - 4ms/step - accuracy: 0.9344 - loss: 0.2701 - val_accuracy: 0.9555 - val_loss: 0.1904
Epoch 5/10
750/750 - 6s - 8ms/step - accuracy: 0.9388 - loss: 0.2528 - val_accuracy: 0.9633 - val_loss: 0.1709
Epoch 6/10
750/750 - 4s - 5ms/step - accuracy: 0.9423 - loss: 0.2402 - val_accuracy: 0.9631 - val_loss: 0.1725
Epoch 7/10
750/750 - 5s - 6ms/step - accuracy: 0.9464 - loss: 0.2265 - val_accuracy: 0.9628 - val_loss: 0.1687
Epoch 8/10
750/750 - 4s - 6ms/step - accuracy: 0.9448 - loss: 0.2236 - val_accuracy: 0.9621 - val_loss: 0.1679
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.9493 - loss: 0.2140 - val_accuracy: 0.9598 - val_loss: 0.1797
Epoch 10/10
750/750 - 3s - 4ms/step - accuracy: 0.9499 - loss: 0.2088 - val_accuracy: 0.9647 - val_loss: 0.1618
Test Accuracy: 96.17%
```

10

96.17

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.0001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 8s - 11ms/step - accuracy: 0.8163 - loss: 0.6551 - val_accuracy: 0.9312 - val_loss: 0.2779
Epoch 2/10
750/750 - 4s - 5ms/step - accuracy: 0.9112 - loss: 0.3506 - val_accuracy: 0.9501 - val_loss: 0.2120
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.9243 - loss: 0.3050 - val_accuracy: 0.9488 - val_loss: 0.2110
Epoch 4/10
750/750 - 5s - 6ms/step - accuracy: 0.9333 - loss: 0.2700 - val_accuracy: 0.9556 - val_loss: 0.1878
Epoch 5/10
750/750 - 4s - 5ms/step - accuracy: 0.9378 - loss: 0.2522 - val_accuracy: 0.9564 - val_loss: 0.1857
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.9421 - loss: 0.2394 - val_accuracy: 0.9602 - val_loss: 0.1789
Epoch 7/10
750/750 - 3s - 4ms/step - accuracy: 0.9446 - loss: 0.2264 - val_accuracy: 0.9588 - val_loss: 0.1761
Epoch 8/10
750/750 - 5s - 6ms/step - accuracy: 0.9465 - loss: 0.2215 - val_accuracy: 0.9617 - val_loss: 0.1669
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.9486 - loss: 0.2132 - val_accuracy: 0.9632 - val_loss: 0.1634
Epoch 10/10
750/750 - 3s - 4ms/step - accuracy: 0.9498 - loss: 0.2091 - val_accuracy: 0.9644 - val_loss: 0.1620
Test Accuracy: 96.20%
```

11

96.20

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 8s - 10ms/step - accuracy: 0.8248 - loss: 0.6006 - val_accuracy: 0.9327 - val_loss: 0.2289
Epoch 2/10
750/750 - 3s - 4ms/step - accuracy: 0.9119 - loss: 0.3038 - val_accuracy: 0.9461 - val_loss: 0.1834
Epoch 3/10
750/750 - 5s - 6ms/step - accuracy: 0.9266 - loss: 0.2545 - val_accuracy: 0.9492 - val_loss: 0.1706
Epoch 4/10
750/750 - 4s - 6ms/step - accuracy: 0.9337 - loss: 0.2268 - val_accuracy: 0.9551 - val_loss: 0.1511
Epoch 5/10
750/750 - 4s - 5ms/step - accuracy: 0.9389 - loss: 0.2067 - val_accuracy: 0.9571 - val_loss: 0.1486
Epoch 6/10
750/750 - 5s - 7ms/step - accuracy: 0.9429 - loss: 0.1966 - val_accuracy: 0.9588 - val_loss: 0.1425
Epoch 7/10
750/750 - 6s - 8ms/step - accuracy: 0.9458 - loss: 0.1848 - val_accuracy: 0.9623 - val_loss: 0.1298
Epoch 8/10
750/750 - 5s - 6ms/step - accuracy: 0.9482 - loss: 0.1744 - val_accuracy: 0.9607 - val_loss: 0.1344
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.9488 - loss: 0.1724 - val_accuracy: 0.9632 - val_loss: 0.1274
Epoch 10/10
750/750 - 5s - 6ms/step - accuracy: 0.9511 - loss: 0.1629 - val_accuracy: 0.9643 - val_loss: 0.1242
Test Accuracy: 96.21%
```

12

96.21

```python
# First Fully Connected layer
model.add(Dense(33, activation='sigmoid', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='sigmoid', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 7s - 10ms/step - accuracy: 0.8374 - loss: 0.5556 - val_accuracy: 0.9185 - val_loss: 0.2737
Epoch 2/10
750/750 - 4s - 6ms/step - accuracy: 0.8953 - loss: 0.3544 - val_accuracy: 0.9371 - val_loss: 0.2127
Epoch 3/10
750/750 - 3s - 5ms/step - accuracy: 0.9108 - loss: 0.3021 - val_accuracy: 0.9408 - val_loss: 0.1946
Epoch 4/10
750/750 - 6s - 8ms/step - accuracy: 0.9212 - loss: 0.2660 - val_accuracy: 0.9517 - val_loss: 0.1630
Epoch 5/10
750/750 - 4s - 5ms/step - accuracy: 0.9265 - loss: 0.2415 - val_accuracy: 0.9555 - val_loss: 0.1520
Epoch 6/10
750/750 - 3s - 5ms/step - accuracy: 0.9337 - loss: 0.2213 - val_accuracy: 0.9572 - val_loss: 0.1459
Epoch 7/10
750/750 - 3s - 4ms/step - accuracy: 0.9391 - loss: 0.2041 - val_accuracy: 0.9602 - val_loss: 0.1393
Epoch 8/10
750/750 - 5s - 7ms/step - accuracy: 0.9416 - loss: 0.1908 - val_accuracy: 0.9630 - val_loss: 0.1324
Epoch 9/10
750/750 - 5s - 7ms/step - accuracy: 0.9451 - loss: 0.1848 - val_accuracy: 0.9655 - val_loss: 0.1215
Epoch 10/10
750/750 - 4s - 5ms/step - accuracy: 0.9489 - loss: 0.1725 - val_accuracy: 0.9668 - val_loss: 0.1165
Test Accuracy: 96.41%
```

13

96.41

```python
# First Fully Connected layer
model.add(Dense(33, activation='sigmoid', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='sigmoid', kernel_regularizer=tf.keras.regularizers.l1(0.000001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 7s - 9ms/step - accuracy: 0.8204 - loss: 0.5991 - val_accuracy: 0.9352 - val_loss: 0.2146
Epoch 2/10
750/750 - 4s - 6ms/step - accuracy: 0.9018 - loss: 0.3297 - val_accuracy: 0.9450 - val_loss: 0.1837
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.9153 - loss: 0.2821 - val_accuracy: 0.9527 - val_loss: 0.1528
Epoch 4/10
750/750 - 5s - 6ms/step - accuracy: 0.9241 - loss: 0.2540 - val_accuracy: 0.9558 - val_loss: 0.1481
Epoch 5/10
750/750 - 6s - 8ms/step - accuracy: 0.9290 - loss: 0.2296 - val_accuracy: 0.9592 - val_loss: 0.1346
Epoch 6/10
750/750 - 3s - 4ms/step - accuracy: 0.9369 - loss: 0.2110 - val_accuracy: 0.9607 - val_loss: 0.1339
Epoch 7/10
750/750 - 3s - 4ms/step - accuracy: 0.9395 - loss: 0.2018 - val_accuracy: 0.9633 - val_loss: 0.1206
Epoch 8/10
750/750 - 3s - 4ms/step - accuracy: 0.9402 - loss: 0.1937 - val_accuracy: 0.9639 - val_loss: 0.1196
Epoch 9/10
750/750 - 5s - 7ms/step - accuracy: 0.9458 - loss: 0.1810 - val_accuracy: 0.9656 - val_loss: 0.1190
Epoch 10/10
750/750 - 3s - 4ms/step - accuracy: 0.9479 - loss: 0.1756 - val_accuracy: 0.9647 - val_loss: 0.1201
Test Accuracy: 96.52%
```

14

96.52

```python
# First Fully Connected layer
model.add(Dense(33, activation='relu', kernel_regularizer=l2(0.00001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Second Fully Connected layer
model.add(Dense(34, activation='relu', kernel_regularizer=l2(0.00001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Third Fully Connected layer
model.add(Dense(51, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))

# Fourth Fully Connected layer
model.add(Dense(22, activation='relu', kernel_regularizer=tf.keras.regularizers.l1(0.00001)))
model.add(BatchNormalization())
model.add(Dropout(0.1))
```

```
Epoch 1/10
750/750 - 8s - 11ms/step - accuracy: 0.7931 - loss: 0.7004 - val_accuracy: 0.9309 - val_loss: 0.2340
Epoch 2/10
750/750 - 4s - 5ms/step - accuracy: 0.9092 - loss: 0.3215 - val_accuracy: 0.9427 - val_loss: 0.1947
Epoch 3/10
750/750 - 3s - 4ms/step - accuracy: 0.9229 - loss: 0.2712 - val_accuracy: 0.9557 - val_loss: 0.1590
Epoch 4/10
750/750 - 5s - 6ms/step - accuracy: 0.9317 - loss: 0.2388 - val_accuracy: 0.9581 - val_loss: 0.1486
Epoch 5/10
750/750 - 3s - 4ms/step - accuracy: 0.9364 - loss: 0.2239 - val_accuracy: 0.9585 - val_loss: 0.1474
Epoch 6/10
750/750 - 5s - 7ms/step - accuracy: 0.9418 - loss: 0.2051 - val_accuracy: 0.9611 - val_loss: 0.1399
Epoch 7/10
750/750 - 6s - 8ms/step - accuracy: 0.9439 - loss: 0.1935 - val_accuracy: 0.9603 - val_loss: 0.1429
Epoch 8/10
750/750 - 3s - 4ms/step - accuracy: 0.9477 - loss: 0.1823 - val_accuracy: 0.9643 - val_loss: 0.1254
Epoch 9/10
750/750 - 3s - 4ms/step - accuracy: 0.9509 - loss: 0.1739 - val_accuracy: 0.9668 - val_loss: 0.1265
Epoch 10/10
750/750 - 4s - 5ms/step - accuracy: 0.9512 - loss: 0.1707 - val_accuracy: 0.9669 - val_loss: 0.1227
Test Accuracy: 96.68%
```

15

96.68