

```

import numpy as np
import matplotlib.pyplot as plt
import pandas as pd
import tensorflow as tf
from tensorflow import keras
from tensorflow.keras import layers
from tensorflow.keras.datasets import imdb
import random

# Set seed for reproducibility
def initialize_seed(seed_value=42):
    """
    Initializes the random seed for NumPy, TensorFlow, and Python random.
    """
    np.random.seed(seed_value)
    tf.random.set_seed(seed_value)
    random.seed(seed_value)

initialize_seed(42)

# Load IMDB dataset
print("Loading IMDB dataset: ")
(train_reviews, train_labels), (test_reviews, test_labels) = imdb.load_data(num_words=10000)

# Function to convert sequences into binary multi-hot encoding
def encode_sequences(sequences, vocab_size=10000):
    """
    Converts sequences of word indices into a binary multi-hot encoded matrix.
    """
    output_matrix = np.zeros((len(sequences), vocab_size))
    for idx, seq in enumerate(sequences):
        output_matrix[idx, seq] = 1.0
    return output_matrix

# Transform training and testing data into vectorized format
x_train = encode_sequences(train_reviews)
x_test = encode_sequences(test_reviews)

y_train = np.asarray(train_labels).astype("float32")
y_test = np.asarray(test_labels).astype("float32")

# Split the data into training and validation sets
x_validation = x_train[:10000]
x_partial_train = x_train[10000:]
y_validation = y_train[:10000]
y_partial_train = y_train[10000:]

# Function to construct the neural network model
def construct_model(hidden_layers=2, neurons=16, activation_fn='relu', dropout=0.0, loss_fn=
    """

```

```

Builds and compiles a sequential neural network model.
"""
model = keras.Sequential()
model.add(layers.Dense(neurons, activation=activation_fn, input_shape=(10000,)))

for _ in range(hidden_layers - 1):
    model.add(layers.Dense(neurons, activation=activation_fn))
    if dropout > 0.0:
        model.add(layers.Dropout(dropout))

model.add(layers.Dense(1, activation='sigmoid'))

model.compile(optimizer='rmsprop',
              loss=loss_fn,
              metrics=['accuracy', 'precision', 'recall', 'auc'])
return model

# Dictionary to store experiment results
experiment_results = {
    'Hidden Layers': {},
    'Neuron Count': {},
    'Loss Function': {},
    'Activation Function': {},
    'Dropout Rate': {}
}

# Function to visualize training and validation metrics
def visualize_metrics(history_dict, title, y_label):
    """
    Plots accuracy or loss over epochs for different experimental setups.
    """
    plt.figure(figsize=(10, 6))
    for label, history in history_dict.items():
        epochs = range(1, len(history.history['accuracy']) + 1)
        plt.plot(epochs, history.history['accuracy'], label=f'Training {y_label}')
        plt.plot(epochs, history.history['val_accuracy'], label=f'Validation {y_label}')
    plt.title(title)
    plt.xlabel('Epochs')
    plt.ylabel(y_label)
    plt.legend()
    plt.show()

➡ Loading IMDB dataset:

# Experiment 1: Testing different numbers of hidden layers
print("\n--- Experiment 1: Number of Hidden Layers ---")
hidden_layer_configs = [1, 2, 3, 4, 5]
histories_hidden_layers = {}

for num_layers in hidden_layer_configs:
    print(f"Training model with {num_layers} hidden layer(s)...")

```

```
model = construct_model(hidden_layers=num_layers, neurons=16)
history = model.fit(x_partial_train, y_partial_train,
                    epochs=20,
                    batch_size=512,
                    validation_data=(x_validation, y_validation),
                    verbose=1)
histories_hidden_layers[num_layers] = history
val_acc = history.history['val_accuracy'][-1]
test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
experiment_results['Hidden Layers'][num_layers] = {'Validation Accuracy': val_acc, 'Test

visualize_metrics(histories_hidden_layers, 'Hidden Layers - Accuracy', 'Accuracy')
visualize_metrics(histories_hidden_layers, 'Hidden Layers - Loss', 'Loss')
```



--- Experiment 1: Number of Hidden Layers ---

Training model with 1 hidden layer(s)...

Epoch 1/20

**30/30**  4s 81ms/step - accuracy: 0.7302 - auc: 0.8075 - loss: 0.5782

Epoch 2/20

**30/30**  1s 38ms/step - accuracy: 0.8906 - auc: 0.9549 - loss: 0.3561

Epoch 3/20

**30/30**  2s 51ms/step - accuracy: 0.9144 - auc: 0.9700 - loss: 0.2788

Epoch 4/20

**30/30**  2s 44ms/step - accuracy: 0.9279 - auc: 0.9778 - loss: 0.2327

Epoch 5/20

**30/30**  2s 37ms/step - accuracy: 0.9376 - auc: 0.9827 - loss: 0.2016

Epoch 6/20

**30/30**  1s 37ms/step - accuracy: 0.9453 - auc: 0.9861 - loss: 0.1786

Epoch 7/20

**30/30**  1s 36ms/step - accuracy: 0.9516 - auc: 0.9888 - loss: 0.1603

Epoch 8/20

**30/30**  1s 36ms/step - accuracy: 0.9572 - auc: 0.9910 - loss: 0.1449

Epoch 9/20

**30/30**  1s 34ms/step - accuracy: 0.9621 - auc: 0.9927 - loss: 0.1316

Epoch 10/20

**30/30**  1s 37ms/step - accuracy: 0.9663 - auc: 0.9942 - loss: 0.1199

Epoch 11/20

**30/30**  1s 44ms/step - accuracy: 0.9707 - auc: 0.9953 - loss: 0.1093

Epoch 12/20

**30/30**  2s 39ms/step - accuracy: 0.9738 - auc: 0.9963 - loss: 0.1000

Epoch 13/20

**30/30**  1s 35ms/step - accuracy: 0.9767 - auc: 0.9970 - loss: 0.0916

Epoch 14/20

**30/30**  1s 38ms/step - accuracy: 0.9796 - auc: 0.9976 - loss: 0.0839

Epoch 15/20

**30/30**  1s 38ms/step - accuracy: 0.9824 - auc: 0.9981 - loss: 0.0771

Epoch 16/20

**30/30**  1s 38ms/step - accuracy: 0.9840 - auc: 0.9985 - loss: 0.0708

Epoch 17/20

**30/30**  1s 37ms/step - accuracy: 0.9868 - auc: 0.9988 - loss: 0.0659

Epoch 18/20

**30/30**  1s 37ms/step - accuracy: 0.9885 - auc: 0.9991 - loss: 0.0607

Epoch 19/20

**30/30**  1s 38ms/step - accuracy: 0.9904 - auc: 0.9993 - loss: 0.0553

Epoch 20/20

**30/30**  1s 48ms/step - accuracy: 0.9918 - auc: 0.9994 - loss: 0.0511

Training model with 2 hidden layer(s)...

Epoch 1/20

**30/30**  4s 91ms/step - accuracy: 0.7028 - auc: 0.7760 - loss: 0.5959

Epoch 2/20

**30/30**  4s 67ms/step - accuracy: 0.8849 - auc: 0.9508 - loss: 0.3399

Epoch 3/20

**30/30**  1s 39ms/step - accuracy: 0.9181 - auc: 0.9726 - loss: 0.2485

Epoch 4/20

**30/30**  1s 35ms/step - accuracy: 0.9374 - auc: 0.9823 - loss: 0.1960

Epoch 5/20

**30/30**  1s 36ms/step - accuracy: 0.9449 - auc: 0.9865 - loss: 0.1666

Epoch 6/20

**30/30**  1s 35ms/step - accuracy: 0.9524 - auc: 0.9902 - loss: 0.1418

```

Epoch 7/20
30/30 ————— 1s 35ms/step - accuracy: 0.9665 - auc: 0.9940 - loss: 0.1154
Epoch 8/20
30/30 ————— 1s 37ms/step - accuracy: 0.9738 - auc: 0.9960 - loss: 0.0972
Epoch 9/20
30/30 ————— 1s 38ms/step - accuracy: 0.9772 - auc: 0.9970 - loss: 0.0844
Epoch 10/20
30/30 ————— 1s 37ms/step - accuracy: 0.9816 - auc: 0.9981 - loss: 0.0714
Epoch 11/20
30/30 ————— 2s 62ms/step - accuracy: 0.9833 - auc: 0.9985 - loss: 0.0643
Epoch 12/20
30/30 ————— 2s 35ms/step - accuracy: 0.9873 - auc: 0.9990 - loss: 0.0547
Epoch 13/20
30/30 ————— 1s 37ms/step - accuracy: 0.9895 - auc: 0.9993 - loss: 0.0467
Epoch 14/20
30/30 ————— 1s 38ms/step - accuracy: 0.9933 - auc: 0.9997 - loss: 0.0362
Epoch 15/20
30/30 ————— 1s 36ms/step - accuracy: 0.9952 - auc: 0.9998 - loss: 0.0300
Epoch 16/20
30/30 ————— 1s 38ms/step - accuracy: 0.9969 - auc: 0.9999 - loss: 0.0260
Epoch 17/20
30/30 ————— 1s 35ms/step - accuracy: 0.9968 - auc: 0.9999 - loss: 0.0233
Epoch 18/20
30/30 ————— 1s 36ms/step - accuracy: 0.9978 - auc: 1.0000 - loss: 0.0204
Epoch 19/20
30/30 ————— 1s 35ms/step - accuracy: 0.9952 - auc: 0.9999 - loss: 0.0210
Epoch 20/20
30/30 ————— 1s 45ms/step - accuracy: 0.9986 - auc: 1.0000 - loss: 0.0143
Training model with 3 hidden layer(s)...
Epoch 1/20
30/30 ————— 5s 90ms/step - accuracy: 0.6970 - auc: 0.7623 - loss: 0.5997
Epoch 2/20
30/30 ————— 4s 39ms/step - accuracy: 0.8907 - auc: 0.9518 - loss: 0.3301
Epoch 3/20
30/30 ————— 1s 38ms/step - accuracy: 0.9247 - auc: 0.9746 - loss: 0.2300
Epoch 4/20
30/30 ————— 1s 39ms/step - accuracy: 0.9422 - auc: 0.9846 - loss: 0.1769
Epoch 5/20
30/30 ————— 1s 39ms/step - accuracy: 0.9478 - auc: 0.9869 - loss: 0.1556
Epoch 6/20
30/30 ————— 1s 47ms/step - accuracy: 0.9583 - auc: 0.9903 - loss: 0.1296
Epoch 7/20
30/30 ————— 1s 39ms/step - accuracy: 0.9722 - auc: 0.9950 - loss: 0.0976
Epoch 8/20
30/30 ————— 1s 39ms/step - accuracy: 0.9779 - auc: 0.9964 - loss: 0.0822
Epoch 9/20
30/30 ————— 2s 61ms/step - accuracy: 0.9841 - auc: 0.9976 - loss: 0.0669
Epoch 10/20
30/30 ————— 2s 60ms/step - accuracy: 0.9822 - auc: 0.9978 - loss: 0.0629
Epoch 11/20
30/30 ————— 1s 37ms/step - accuracy: 0.9864 - auc: 0.9987 - loss: 0.0509
Epoch 12/20
30/30 ————— 1s 39ms/step - accuracy: 0.9928 - auc: 0.9993 - loss: 0.0358
Epoch 13/20
30/30 ————— 1s 35ms/step - accuracy: 0.9964 - auc: 0.9998 - loss: 0.0259
Epoch 14/20
30/30 ————— 1s 37ms/step - accuracy: 0.9992 - auc: 0.9999 - loss: 0.0268

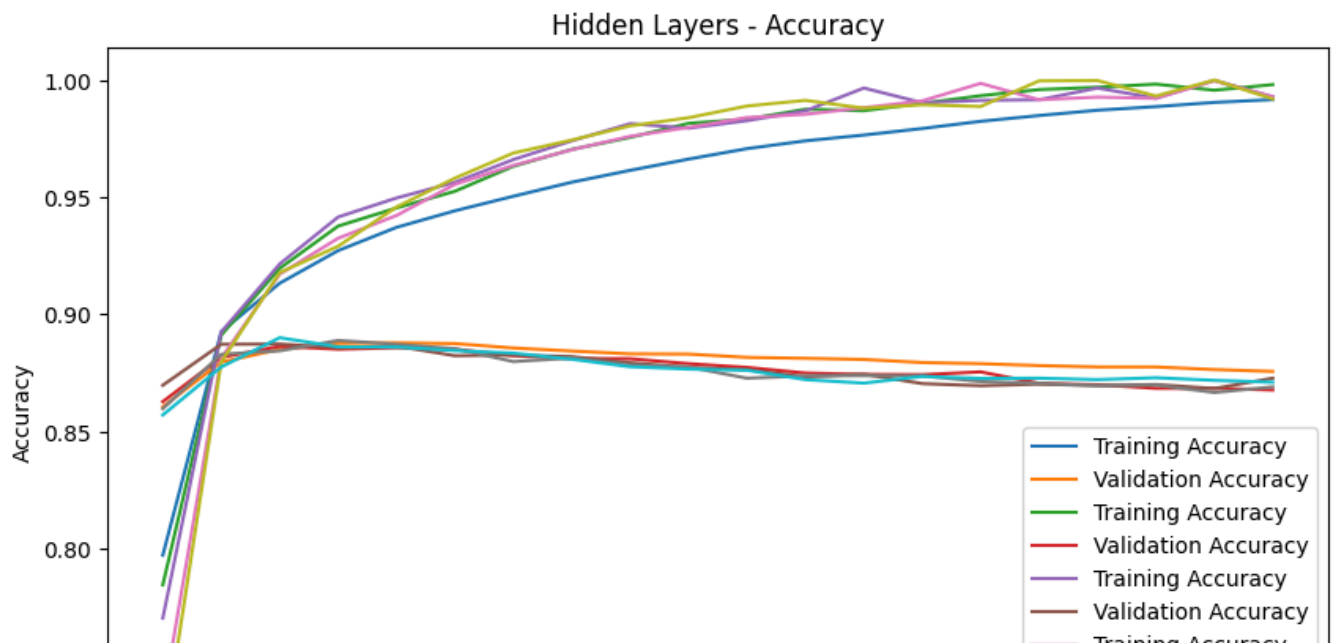
```

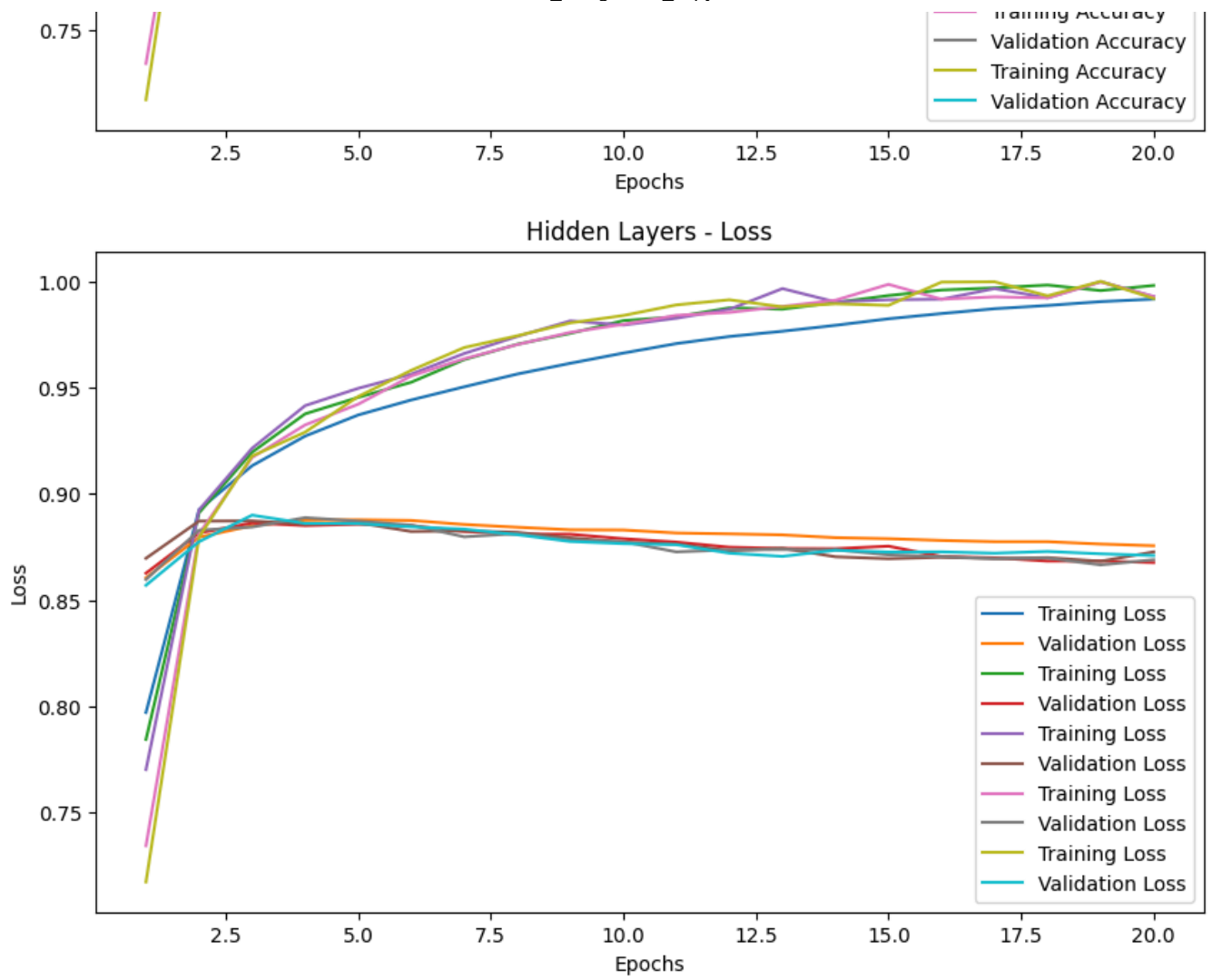
```

Epoch 15/20
30/30 ————— 1s 39ms/step - accuracy: 0.9921 - auc: 0.9992 - loss: 0.0299
Epoch 16/20
30/30 ————— 1s 36ms/step - accuracy: 0.9942 - auc: 0.9996 - loss: 0.0222
Epoch 17/20
30/30 ————— 1s 38ms/step - accuracy: 0.9981 - auc: 0.9999 - loss: 0.0132
Epoch 18/20
30/30 ————— 2s 50ms/step - accuracy: 0.9973 - auc: 0.9997 - loss: 0.0150
Epoch 19/20
30/30 ————— 2s 45ms/step - accuracy: 0.9999 - auc: 1.0000 - loss: 0.0064
Epoch 20/20
30/30 ————— 1s 38ms/step - accuracy: 0.9983 - auc: 0.9998 - loss: 0.0097
Training model with 4 hidden layer(s)...
Epoch 1/20
30/30 ————— 5s 112ms/step - accuracy: 0.6467 - auc: 0.7149 - loss: 0.6541
Epoch 2/20
30/30 ————— 1s 39ms/step - accuracy: 0.8740 - auc: 0.9389 - loss: 0.4015
Epoch 3/20
30/30 ————— 1s 37ms/step - accuracy: 0.9143 - auc: 0.9681 - loss: 0.2634
Epoch 4/20
30/30 ————— 1s 36ms/step - accuracy: 0.9320 - auc: 0.9784 - loss: 0.2030
Epoch 5/20
30/30 ————— 1s 37ms/step - accuracy: 0.9415 - auc: 0.9850 - loss: 0.1652
Epoch 6/20
30/30 ————— 1s 37ms/step - accuracy: 0.9597 - auc: 0.9904 - loss: 0.1311
Epoch 7/20
30/30 ————— 1s 36ms/step - accuracy: 0.9655 - auc: 0.9930 - loss: 0.1107
Epoch 8/20
30/30 ————— 1s 36ms/step - accuracy: 0.9732 - auc: 0.9949 - loss: 0.0917
Epoch 9/20
30/30 ————— 1s 36ms/step - accuracy: 0.9795 - auc: 0.9964 - loss: 0.0746
Epoch 10/20
30/30 ————— 2s 59ms/step - accuracy: 0.9825 - auc: 0.9972 - loss: 0.0637
Epoch 11/20
30/30 ————— 2s 53ms/step - accuracy: 0.9866 - auc: 0.9982 - loss: 0.0511
Epoch 12/20
30/30 ————— 2s 37ms/step - accuracy: 0.9902 - auc: 0.9985 - loss: 0.0448
Epoch 13/20
30/30 ————— 1s 36ms/step - accuracy: 0.9944 - auc: 0.9990 - loss: 0.0321
Epoch 14/20
30/30 ————— 1s 36ms/step - accuracy: 0.9966 - auc: 0.9992 - loss: 0.0244
Epoch 15/20
30/30 ————— 1s 35ms/step - accuracy: 0.9984 - auc: 0.9993 - loss: 0.0185
Epoch 16/20
30/30 ————— 1s 36ms/step - accuracy: 0.9841 - auc: 0.9967 - loss: 0.0531
Epoch 17/20
30/30 ————— 1s 37ms/step - accuracy: 0.9879 - auc: 0.9980 - loss: 0.0404
Epoch 18/20
30/30 ————— 1s 37ms/step - accuracy: 0.9960 - auc: 0.9995 - loss: 0.0165
Epoch 19/20
30/30 ————— 1s 46ms/step - accuracy: 0.9997 - auc: 1.0000 - loss: 0.0067
Epoch 20/20
30/30 ————— 2s 64ms/step - accuracy: 0.9967 - auc: 0.9996 - loss: 0.0133
Training model with 5 hidden layer(s)...
Epoch 1/20
30/30 ————— 5s 82ms/step - accuracy: 0.6414 - auc: 0.6932 - loss: 0.6552

```

Epoch 2/20  
**30/30** ————— **2s** 65ms/step - accuracy: 0.8729 - auc: 0.9373 - loss: 0.3936  
 Epoch 3/20  
**30/30** ————— **2s** 39ms/step - accuracy: 0.9156 - auc: 0.9692 - loss: 0.2542  
 Epoch 4/20  
**30/30** ————— **1s** 39ms/step - accuracy: 0.9276 - auc: 0.9778 - loss: 0.1988  
 Epoch 5/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9515 - auc: 0.9872 - loss: 0.1503  
 Epoch 6/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9606 - auc: 0.9919 - loss: 0.1185  
 Epoch 7/20  
**30/30** ————— **1s** 37ms/step - accuracy: 0.9716 - auc: 0.9945 - loss: 0.0951  
 Epoch 8/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9762 - auc: 0.9955 - loss: 0.0820  
 Epoch 9/20  
**30/30** ————— **1s** 39ms/step - accuracy: 0.9780 - auc: 0.9964 - loss: 0.0719  
 Epoch 10/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9821 - auc: 0.9970 - loss: 0.0622  
 Epoch 11/20  
**30/30** ————— **2s** 61ms/step - accuracy: 0.9853 - auc: 0.9981 - loss: 0.0498  
 Epoch 12/20  
**30/30** ————— **2s** 51ms/step - accuracy: 0.9911 - auc: 0.9989 - loss: 0.0349  
 Epoch 13/20  
**30/30** ————— **2s** 35ms/step - accuracy: 0.9896 - auc: 0.9977 - loss: 0.0406  
 Epoch 14/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9961 - auc: 0.9994 - loss: 0.0178  
 Epoch 15/20  
**30/30** ————— **1s** 37ms/step - accuracy: 0.9968 - auc: 0.9996 - loss: 0.0147  
 Epoch 16/20  
**30/30** ————— **1s** 38ms/step - accuracy: 0.9997 - auc: 1.0000 - loss: 0.0069  
 Epoch 17/20  
**30/30** ————— **1s** 35ms/step - accuracy: 0.9997 - auc: 1.0000 - loss: 0.0049  
 Epoch 18/20  
**30/30** ————— **1s** 36ms/step - accuracy: 0.9897 - auc: 0.9979 - loss: 0.0368  
 Epoch 19/20  
**30/30** ————— **1s** 36ms/step - accuracy: 0.9999 - auc: 1.0000 - loss: 0.0020  
 Epoch 20/20  
**30/30** ————— **2s** 58ms/step - accuracy: 0.9967 - auc: 0.9988 - loss: 0.0160







```
# Experiment 2: Changing number of neurons
print("\n--- Experiment 2: Varying Neuron Count ---")
neuron_options = [32, 64, 128, 256]
histories_neurons = {}

for neurons in neuron_options:
    print(f"Training model with {neurons} neurons per layer...")
    model = construct_model(hidden_layers=2, neurons=neurons)
    history = model.fit(x_partial_train, y_partial_train,
                        epochs=20,
                        batch_size=512,
                        validation_data=(x_validation, y_validation),
                        verbose=1)
    histories_neurons[neurons] = history
    val_acc = history.history['val_accuracy'][-1]
    test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
    experiment_results['Neuron Count'][neurons] = {'Validation Accuracy': val_acc, 'Test Acc

visualize_metrics(histories_neurons, 'Neuron Count - Accuracy', 'Accuracy')
visualize_metrics(histories_neurons, 'Neuron Count - Loss', 'Loss')
```



--- Experiment 2: Varying Neuron Count ---

Training model with 32 neurons per layer...

Epoch 1/20

**30/30**  4s 85ms/step - accuracy: 0.7030 - auc: 0.7676 - loss: 0.5940

Epoch 2/20

**30/30**  2s 59ms/step - accuracy: 0.8883 - auc: 0.9511 - loss: 0.3281

Epoch 3/20

**30/30**  3s 66ms/step - accuracy: 0.9176 - auc: 0.9727 - loss: 0.2376

Epoch 4/20

**30/30**  2s 53ms/step - accuracy: 0.9308 - auc: 0.9808 - loss: 0.1921

Epoch 5/20

**30/30**  3s 55ms/step - accuracy: 0.9500 - auc: 0.9877 - loss: 0.1554

Epoch 6/20

**30/30**  2s 53ms/step - accuracy: 0.9578 - auc: 0.9914 - loss: 0.1301

Epoch 7/20

**30/30**  2s 53ms/step - accuracy: 0.9595 - auc: 0.9928 - loss: 0.1174

Epoch 8/20

**30/30**  1s 42ms/step - accuracy: 0.9686 - auc: 0.9957 - loss: 0.0959

Epoch 9/20

**30/30**  3s 60ms/step - accuracy: 0.9744 - auc: 0.9967 - loss: 0.0837

Epoch 10/20

**30/30**  2s 54ms/step - accuracy: 0.9815 - auc: 0.9982 - loss: 0.0656

Epoch 11/20

**30/30**  2s 42ms/step - accuracy: 0.9854 - auc: 0.9987 - loss: 0.0559

Epoch 12/20

**30/30**  3s 43ms/step - accuracy: 0.9897 - auc: 0.9992 - loss: 0.0463

Epoch 13/20

**30/30**  1s 43ms/step - accuracy: 0.9921 - auc: 0.9996 - loss: 0.0379

Epoch 14/20

**30/30**  1s 43ms/step - accuracy: 0.9851 - auc: 0.9981 - loss: 0.0496

Epoch 15/20

**30/30**  3s 60ms/step - accuracy: 0.9839 - auc: 0.9975 - loss: 0.0517

Epoch 16/20

**30/30**  2s 53ms/step - accuracy: 0.9953 - auc: 0.9996 - loss: 0.0241

Epoch 17/20

**30/30**  1s 44ms/step - accuracy: 0.9974 - auc: 0.9999 - loss: 0.0162

Epoch 18/20

**30/30**  1s 43ms/step - accuracy: 0.9973 - auc: 0.9996 - loss: 0.0170

Epoch 19/20

**30/30**  1s 42ms/step - accuracy: 0.9998 - auc: 1.0000 - loss: 0.0085

Epoch 20/20

**30/30**  3s 42ms/step - accuracy: 0.9983 - auc: 0.9997 - loss: 0.0114

Training model with 64 neurons per layer...

Epoch 1/20

**30/30**  5s 102ms/step - accuracy: 0.7005 - auc: 0.7644 - loss: 0.5875

Epoch 2/20

**30/30**  5s 85ms/step - accuracy: 0.8883 - auc: 0.9495 - loss: 0.3122

Epoch 3/20

**30/30**  5s 69ms/step - accuracy: 0.9135 - auc: 0.9700 - loss: 0.2320

Epoch 4/20

**30/30**  2s 68ms/step - accuracy: 0.9389 - auc: 0.9833 - loss: 0.1757
















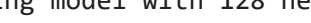












Epoch 5/20

**30/30**  2s 61ms/step - accuracy: 0.9375 - auc: 0.9850 - loss: 0.1608

Epoch 6/20

**30/30**  4s 120ms/step - accuracy: 0.9589 - auc: 0.9919 - loss: 0.1215

```

Epoch 7/20
30/30  3s 62ms/step - accuracy: 0.9665 - auc: 0.9948 - loss: 0.0984
Epoch 8/20
30/30  2s 61ms/step - accuracy: 0.9757 - auc: 0.9961 - loss: 0.0817
Epoch 9/20
30/30  2s 68ms/step - accuracy: 0.9773 - auc: 0.9971 - loss: 0.0718
Epoch 10/20
30/30  2s 67ms/step - accuracy: 0.9859 - auc: 0.9986 - loss: 0.0521
Epoch 11/20
30/30  3s 94ms/step - accuracy: 0.9884 - auc: 0.9994 - loss: 0.0431
Epoch 12/20
30/30  4s 68ms/step - accuracy: 0.9947 - auc: 0.9998 - loss: 0.0271
Epoch 13/20
30/30  2s 62ms/step - accuracy: 0.9828 - auc: 0.9981 - loss: 0.0515
Epoch 14/20
30/30  3s 68ms/step - accuracy: 0.9948 - auc: 0.9996 - loss: 0.0231
Epoch 15/20
30/30  4s 102ms/step - accuracy: 0.9998 - auc: 1.0000 - loss: 0.0076
Epoch 16/20
30/30  2s 68ms/step - accuracy: 0.9924 - auc: 0.9988 - loss: 0.0266
Epoch 17/20
30/30  2s 63ms/step - accuracy: 0.9999 - auc: 1.0000 - loss: 0.0044
Epoch 18/20
30/30  2s 68ms/step - accuracy: 0.9978 - auc: 0.9995 - loss: 0.0106
Epoch 19/20
30/30  3s 68ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0036
Epoch 20/20
30/30  3s 116ms/step - accuracy: 0.9994 - auc: 0.9999 - loss: 0.0042
Training model with 128 neurons per layer...
Epoch 1/20
30/30  7s 169ms/step - accuracy: 0.6496 - auc: 0.7200 - loss: 0.6054
Epoch 2/20
30/30  9s 144ms/step - accuracy: 0.8783 - auc: 0.9412 - loss: 0.3207
Epoch 3/20
30/30  4s 96ms/step - accuracy: 0.9084 - auc: 0.9648 - loss: 0.2432
Epoch 4/20
30/30  3s 94ms/step - accuracy: 0.9372 - auc: 0.9819 - loss: 0.1769
Epoch 5/20
30/30  6s 140ms/step - accuracy: 0.9494 - auc: 0.9879 - loss: 0.1442
Epoch 6/20
30/30  4s 101ms/step - accuracy: 0.9598 - auc: 0.9923 - loss: 0.1174
Epoch 7/20
30/30  6s 117ms/step - accuracy: 0.9674 - auc: 0.9946 - loss: 0.0967
Epoch 8/20
30/30  5s 155ms/step - accuracy: 0.9806 - auc: 0.9977 - loss: 0.0659
Epoch 9/20
30/30  3s 96ms/step - accuracy: 0.9855 - auc: 0.9970 - loss: 0.0562
Epoch 10/20
30/30  4s 117ms/step - accuracy: 0.9932 - auc: 0.9988 - loss: 0.0338
Epoch 11/20
30/30  3s 96ms/step - accuracy: 0.9970 - auc: 0.9999 - loss: 0.0190
Epoch 12/20
30/30  6s 117ms/step - accuracy: 0.9972 - auc: 0.9995 - loss: 0.0187
Epoch 13/20
30/30  5s 96ms/step - accuracy: 0.9997 - auc: 1.0000 - loss: 0.0125
Epoch 14/20
30/30  4s 136ms/step - accuracy: 0.9997 - auc: 1.0000 - loss: 0.0056

```

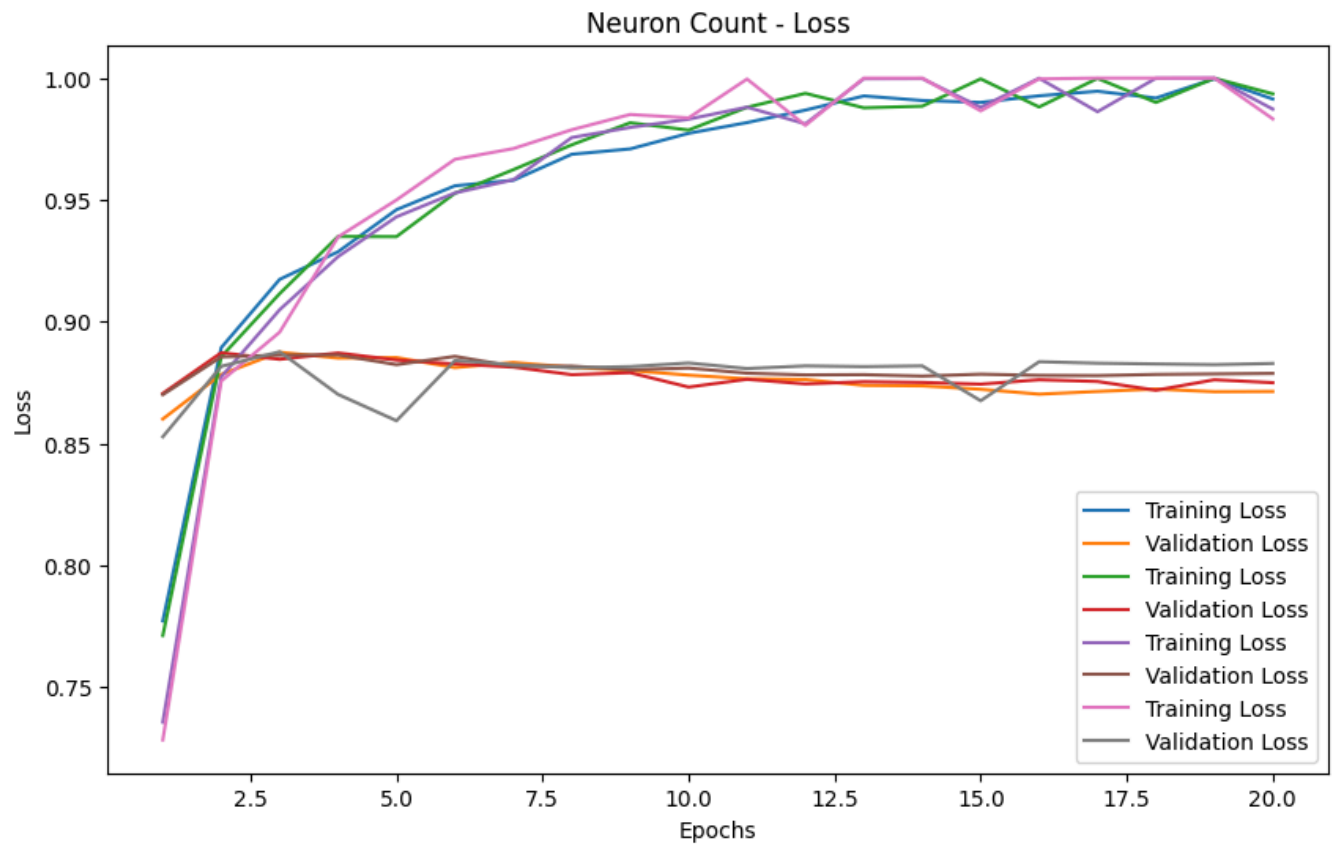
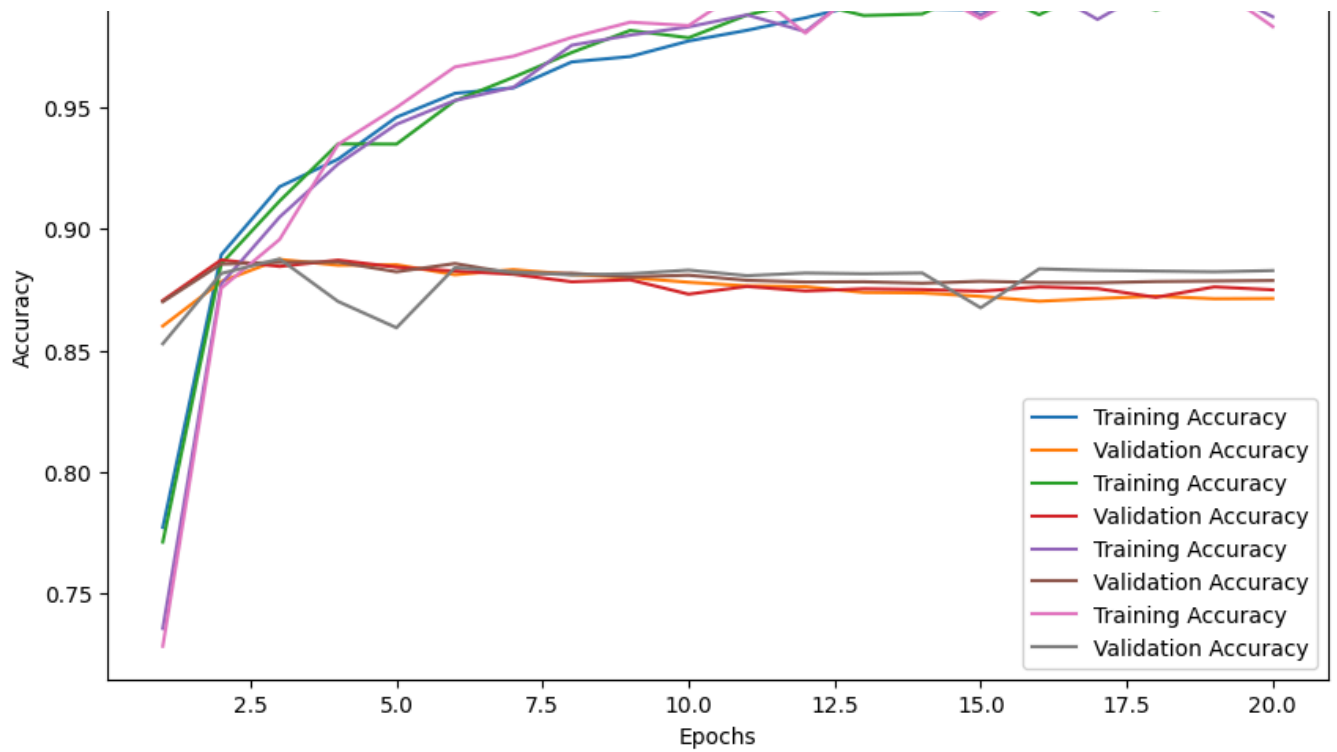
```

Epoch 15/20
30/30 4s 119ms/step - accuracy: 0.9846 - auc: 0.9951 - loss: 0.0598
Epoch 16/20
30/30 3s 116ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0032
Epoch 17/20
30/30 6s 132ms/step - accuracy: 0.9904 - auc: 0.9983 - loss: 0.0307
Epoch 18/20
30/30 5s 118ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0026
Epoch 19/20
30/30 3s 115ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0015
Epoch 20/20
30/30 6s 153ms/step - accuracy: 0.9926 - auc: 0.9975 - loss: 0.0323
Training model with 256 neurons per layer...
Epoch 1/20
30/30 9s 209ms/step - accuracy: 0.6450 - auc: 0.7088 - loss: 0.6159
Epoch 2/20
30/30 10s 211ms/step - accuracy: 0.8693 - auc: 0.9391 - loss: 0.3221
Epoch 3/20
30/30 10s 194ms/step - accuracy: 0.8955 - auc: 0.9593 - loss: 0.2599
Epoch 4/20
30/30 9s 159ms/step - accuracy: 0.9379 - auc: 0.9833 - loss: 0.1681
Epoch 5/20
30/30 6s 203ms/step - accuracy: 0.9490 - auc: 0.9893 - loss: 0.1376
Epoch 6/20
30/30 5s 166ms/step - accuracy: 0.9611 - auc: 0.9943 - loss: 0.1049
Epoch 7/20
30/30 6s 213ms/step - accuracy: 0.9725 - auc: 0.9951 - loss: 0.0838
Epoch 8/20
30/30 9s 164ms/step - accuracy: 0.9828 - auc: 0.9969 - loss: 0.0605
Epoch 9/20
30/30 6s 210ms/step - accuracy: 0.9908 - auc: 0.9989 - loss: 0.0338
Epoch 10/20
30/30 9s 171ms/step - accuracy: 0.9918 - auc: 0.9987 - loss: 0.0302
Epoch 11/20
30/30 10s 159ms/step - accuracy: 0.9995 - auc: 1.0000 - loss: 0.0081
Epoch 12/20
30/30 6s 204ms/step - accuracy: 0.9958 - auc: 0.9980 - loss: 0.0247
Epoch 13/20
30/30 10s 210ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0051
Epoch 14/20
30/30 9s 169ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0026
Epoch 15/20
30/30 6s 211ms/step - accuracy: 0.9986 - auc: 0.9994 - loss: 0.0083
Epoch 16/20
30/30 5s 158ms/step - accuracy: 0.9987 - auc: 1.0000 - loss: 0.0067
Epoch 17/20
30/30 7s 206ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 0.0014
Epoch 18/20
30/30 5s 159ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 8.5046
Epoch 19/20
30/30 5s 166ms/step - accuracy: 1.0000 - auc: 1.0000 - loss: 7.1691
Epoch 20/20
30/30 6s 197ms/step - accuracy: 0.9831 - auc: 0.9935 - loss: 0.0757

```

Neuron Count - Accuracy







```
# Experiment 3: Comparing loss functions
print("\n--- Experiment 3: Loss Function Comparison ---")
histories_loss_functions = {}

print("Training model with MSE loss function...")
mse_model = construct_model(loss_fn='mse')
history_mse = mse_model.fit(x_partial_train, y_partial_train,
                             epochs=20,
                             batch_size=512,
                             validation_data=(x_validation, y_validation),
                             verbose=1)
histories_loss_functions['MSE'] = history_mse
val_acc = history.history['val_accuracy'][-1]
test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
experiment_results['Neuron Count'][neurons] = {'Validation Accuracy': val_acc, 'Test Accuracy': test_acc}

visualize_metrics(histories_loss_functions, 'Loss Function Comparison - Accuracy', 'Accuracy')
visualize_metrics(histories_loss_functions, 'Loss Function Comparison - Loss', 'Loss')
```



--- Experiment 3: Loss Function Comparison ---

Training model with MSE loss function...

Epoch 1/20

**30/30**  **4s** 78ms/step - accuracy: 0.6700 - auc: 0.7590 - loss: 0.2204

Epoch 2/20

**30/30**  **1s** 37ms/step - accuracy: 0.8801 - auc: 0.9431 - loss: 0.1232

Epoch 3/20

**30/30**  **1s** 35ms/step - accuracy: 0.9110 - auc: 0.9647 - loss: 0.0872

Epoch 4/20

**30/30**  **1s** 36ms/step - accuracy: 0.9196 - auc: 0.9714 - loss: 0.0717

Epoch 5/20

**30/30**  **1s** 37ms/step - accuracy: 0.9296 - auc: 0.9772 - loss: 0.0613

Epoch 6/20

**30/30**  **1s** 36ms/step - accuracy: 0.9452 - auc: 0.9831 - loss: 0.0509

Epoch 7/20

**30/30**  **2s** 59ms/step - accuracy: 0.9553 - auc: 0.9862 - loss: 0.0438

Epoch 8/20

**30/30**  **1s** 46ms/step - accuracy: 0.9620 - auc: 0.9882 - loss: 0.0384

Epoch 9/20

**30/30**  **2s** 36ms/step - accuracy: 0.9660 - auc: 0.9892 - loss: 0.0349

Epoch 10/20

**30/30**  **1s** 36ms/step - accuracy: 0.9708 - auc: 0.9907 - loss: 0.0299

Epoch 11/20

**30/30**  **1s** 34ms/step - accuracy: 0.9757 - auc: 0.9912 - loss: 0.0268

Epoch 12/20

**30/30**  **1s** 34ms/step - accuracy: 0.9796 - auc: 0.9918 - loss: 0.0237

Epoch 13/20

**30/30**  **1s** 34ms/step - accuracy: 0.9830 - auc: 0.9923 - loss: 0.0210

Epoch 14/20

**30/30**  **1s** 33ms/step - accuracy: 0.9859 - auc: 0.9926 - loss: 0.0194

Epoch 15/20

**30/30**  **1s** 34ms/step - accuracy: 0.9823 - auc: 0.9925 - loss: 0.0198

Epoch 16/20

**30/30**  **2s** 54ms/step - accuracy: 0.9892 - auc: 0.9933 - loss: 0.0150

Epoch 17/20

**30/30**  **2s** 36ms/step - accuracy: 0.9889 - auc: 0.9935 - loss: 0.0152

Epoch 18/20

**30/30**  **1s** 34ms/step - accuracy: 0.9895 - auc: 0.9934 - loss: 0.0142

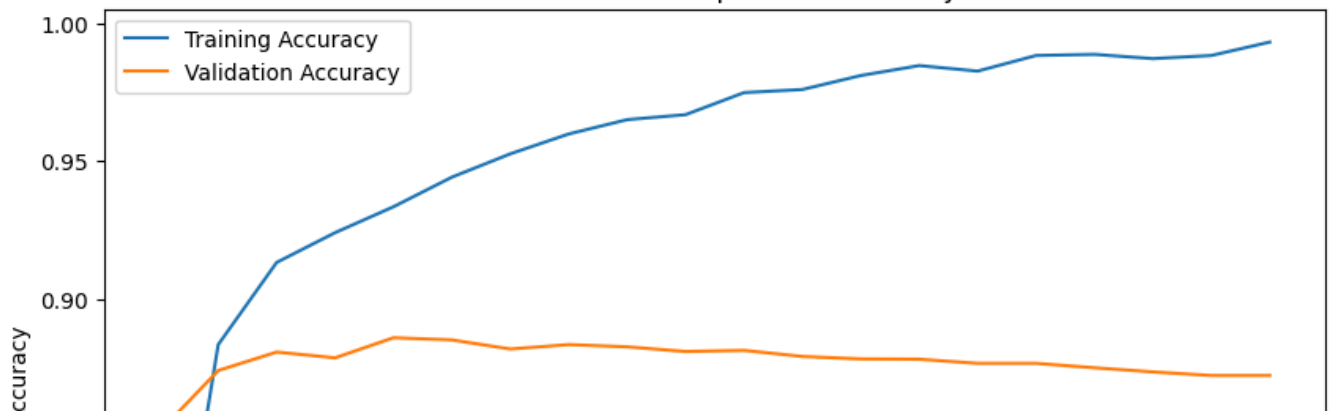
Epoch 19/20

**30/30**  **1s** 33ms/step - accuracy: 0.9919 - auc: 0.9935 - loss: 0.0110

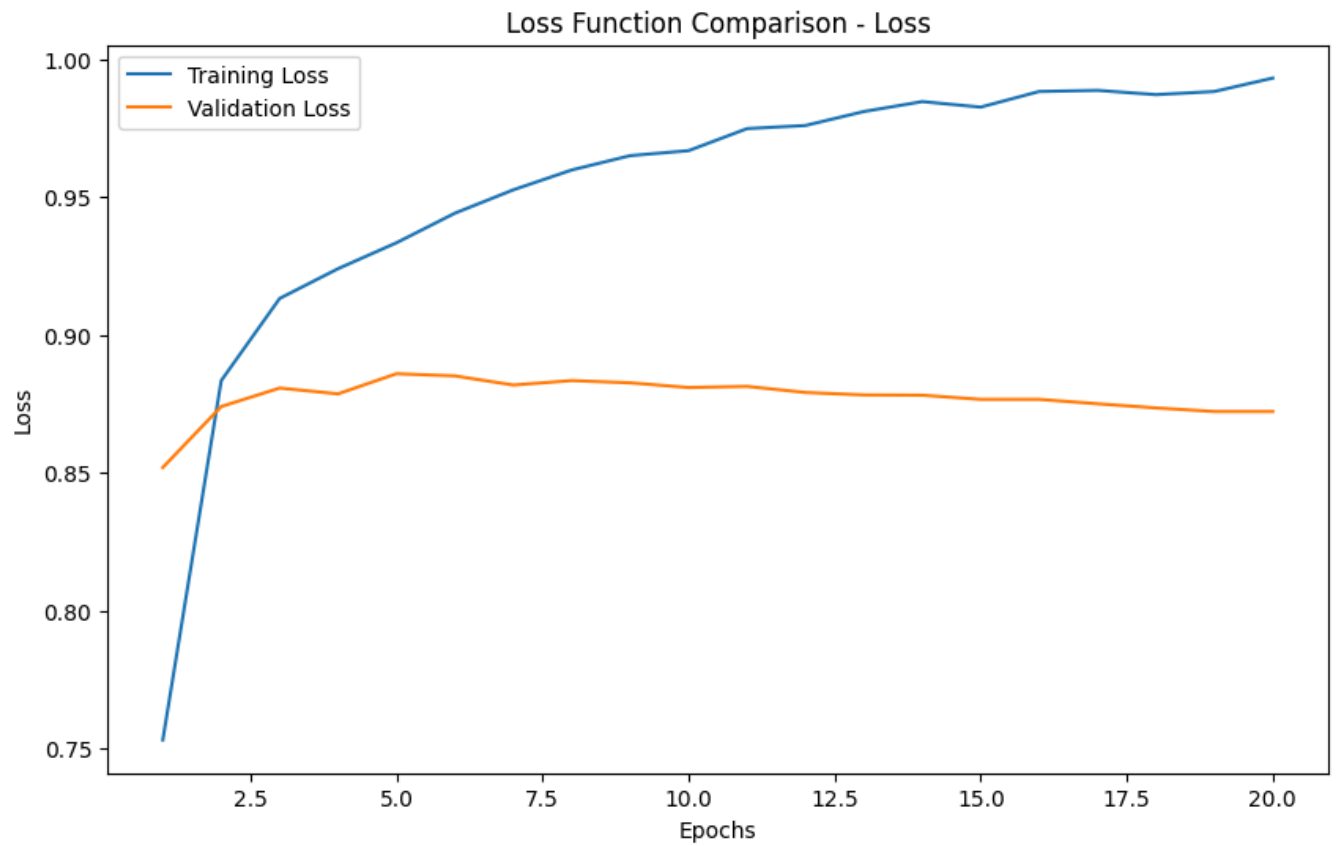
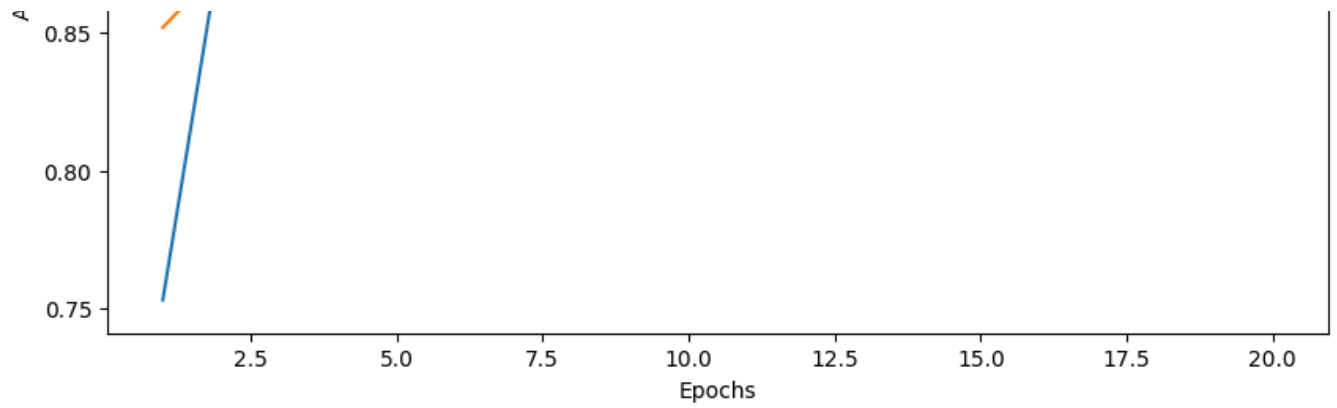
Epoch 20/20

**30/30**  **1s** 35ms/step - accuracy: 0.9933 - auc: 0.9938 - loss: 0.0095

Loss Function Comparison - Accuracy







```
# Experiment 4: Testing activation functions
print("\n--- Experiment 4: Activation Function Comparison ---")
histories_activation = {}

print("Training model with tanh activation function...")
tanh_model = construct_model(activation_fn='tanh')
history_tanh = tanh_model.fit(x_partial_train, y_partial_train,
                              epochs=20,
                              batch_size=512,
                              validation_data=(x_validation, y_validation),
                              verbose=1)
histories_activation['tanh'] = history_tanh
val_acc = history.history['val_accuracy'][-1]
test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
experiment_results['Neuron Count'][neurons] = {'Validation Accuracy': val_acc, 'Test Accuracy': test_acc}

visualize_metrics(histories_activation, 'Activation Function Comparison - Accuracy', 'Accuracy')
visualize_metrics(histories_activation, 'Activation Function Comparison - Loss', 'Loss')
```



--- Experiment 4: Activation Function Comparison ---

Training model with tanh activation function...

Epoch 1/20

**30/30**  **4s** 77ms/step - accuracy: 0.7133 - auc: 0.7933 - loss: 0.5752

Epoch 2/20

**30/30**  **1s** 49ms/step - accuracy: 0.8980 - auc: 0.9596 - loss: 0.3253

Epoch 3/20

**30/30**  **1s** 37ms/step - accuracy: 0.9239 - auc: 0.9750 - loss: 0.2355

Epoch 4/20

**30/30**  **1s** 36ms/step - accuracy: 0.9368 - auc: 0.9826 - loss: 0.1837


Epoch 5/20

**30/30**  **2s** 45ms/step - accuracy: 0.9546 - auc: 0.9885 - loss: 0.1418

Epoch 6/20

**30/30**  **2s** 37ms/step - accuracy: 0.9628 - auc: 0.9909 - loss: 0.1190


Epoch 7/20

**30/30**  **1s** 35ms/step - accuracy: 0.9726 - auc: 0.9938 - loss: 0.0939

Epoch 8/20

**30/30**  **1s** 36ms/step - accuracy: 0.9734 - auc: 0.9945 - loss: 0.0849

Epoch 9/20

**30/30**  **1s** 38ms/step - accuracy: 0.9848 - auc: 0.9964 - loss: 0.0627

Epoch 10/20

**30/30**  **1s** 37ms/step - accuracy: 0.9867 - auc: 0.9971 - loss: 0.0503

Epoch 11/20

**30/30**  **1s** 37ms/step - accuracy: 0.9919 - auc: 0.9978 - loss: 0.0409

Epoch 12/20

**30/30**  **1s** 36ms/step - accuracy: 0.9927 - auc: 0.9982 - loss: 0.0357

Epoch 13/20

**30/30**  **1s** 35ms/step - accuracy: 0.9955 - auc: 0.9986 - loss: 0.0264

Epoch 14/20

**30/30**  **1s** 44ms/step - accuracy: 0.9950 - auc: 0.9983 - loss: 0.0276

Epoch 15/20

**30/30**  **2s** 36ms/step - accuracy: 0.9988 - auc: 0.9991 - loss: 0.0144

Epoch 16/20

**30/30**  **1s** 35ms/step - accuracy: 0.9917 - auc: 0.9983 - loss: 0.0305

Epoch 17/20

**30/30**  **1s** 35ms/step - accuracy: 0.9956 - auc: 0.9993 - loss: 0.0182

Epoch 18/20

**30/30**  **1s** 35ms/step - accuracy: 0.9951 - auc: 0.9985 - loss: 0.0231

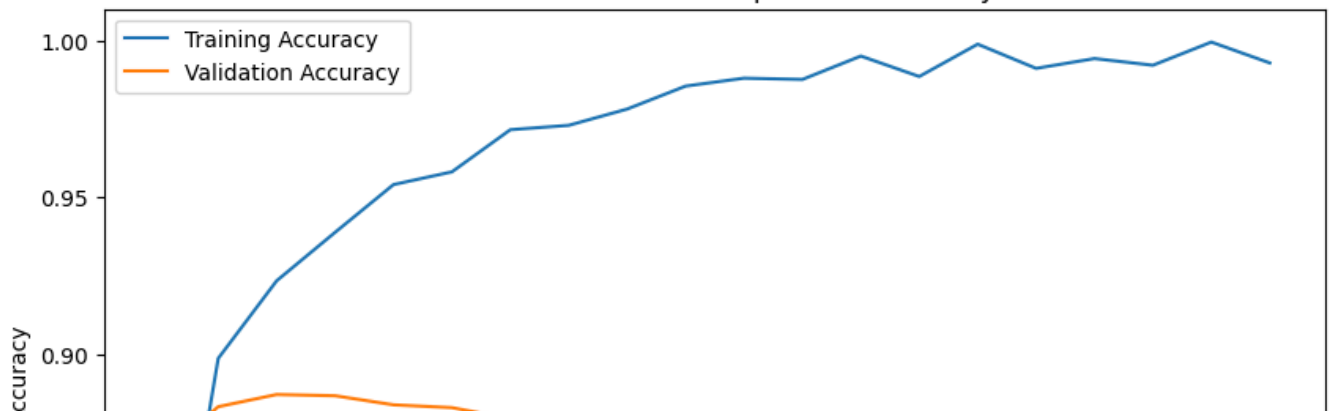
Epoch 19/20

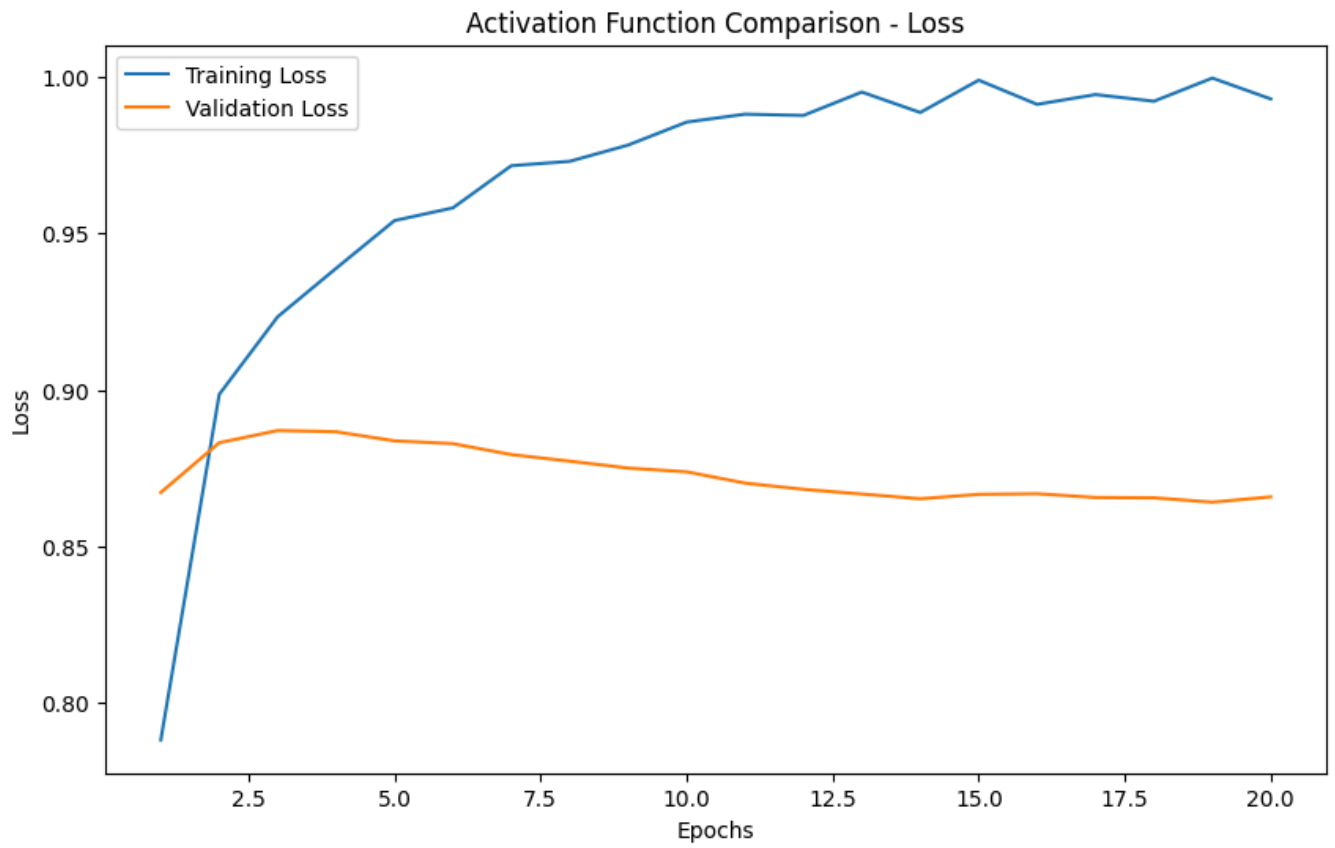
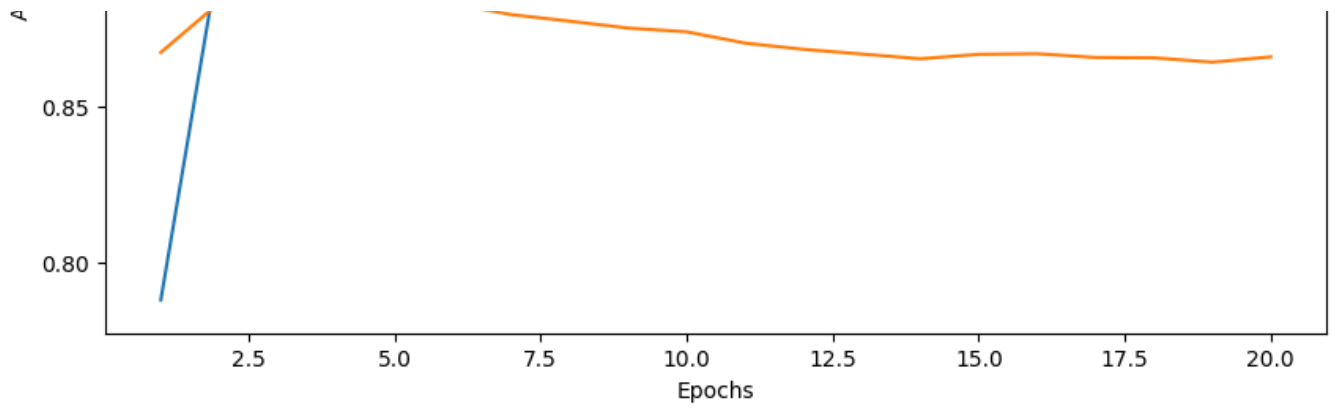
**30/30**  **1s** 35ms/step - accuracy: 0.9994 - auc: 0.9997 - loss: 0.0075

Epoch 20/20

**30/30**  **1s** 36ms/step - accuracy: 0.9964 - auc: 0.9992 - loss: 0.0150

Activation Function Comparison - Accuracy





```

# Experiment 5: Dropout regularization
print("\n--- Experiment 5: Dropout Regularization ---")
dropout_rates = [0.3, 0.5, 0.7]
histories_dropout = {}

for rate in dropout_rates:
    print(f"Training model with dropout rate {rate}...")
    model = construct_model(dropout=rate)
    history = model.fit(x_partial_train, y_partial_train,
                        epochs=20,
                        batch_size=512,
                        validation_data=(x_validation, y_validation),
                        verbose=1)
    histories_dropout[rate] = history
val_acc = history.history['val_accuracy'][-1]
test_acc = model.evaluate(x_test, y_test, verbose=0)[1]
experiment_results['Neuron Count'][neurons] = {'Validation Accuracy': val_acc, 'Test Accuracy': test_acc}

visualize_metrics(histories_dropout, 'Dropout Rate - Accuracy', 'Accuracy')
visualize_metrics(histories_dropout, 'Dropout Rate - Loss', 'Loss')

```



```

--- Experiment 5: Dropout Regularization ---
Training model with dropout rate 0.3...
Epoch 1/20
30/30 ————— 4s 77ms/step - accuracy: 0.6591 - auc: 0.7271 - loss: 0.6289
Epoch 2/20
30/30 ————— 1s 48ms/step - accuracy: 0.8556 - auc: 0.9256 - loss: 0.4164
Epoch 3/20

```