IMDB assignment

For the first solution:

Hidden Layers num:2

```
 # First intermediate layer (16 units, ReLU activation)
layers.Dense(16, activation="relu"),
# Second intermediate layer (16 units, ReLU activation)
 layers.Dense(16, activation="relu"),
# Output layer (1 unit, Sigmoid activation for binary classification)
 layers.Dense(1, activation="sigmoid")
```

Result:

```
--- Final Results ---
Test Loss: 0.2852
Test Accuracy: 0.8874
```

Alternative: to 3 hidden layers

Hideen Layers num:3

```
# First intermediate layer (16 units, ReLU activation)
layers.Dense(16, activation="relu"),
# Second intermediate layer (16 units, ReLU activation)
layers.Dense(16, activation="relu"),
# Third intermediate layer(same), for alternative test
layers.Dense(16, activation="relu"),
# Output layer (1 unit, Sigmoid activation for binary classification)
```

layers.Dense(1, activation="sigmoid")

Result:

Test Loss: 0.3059

Test Accuracy: 0.8812

Alternative: turn the loss function from binary_crossentrophy to mse

Still use two hidden layers,

```
  # First intermediate layer (16 units, ReLU activation)
        layers.Dense(16, activation="relu"),
        # Second intermediate layer (16 units, ReLU activation)
        layers.Dense(16, activation="relu"),
        # Output layer (1 unit, Sigmoid activation for binary classification)
        layers.Dense(1, activation="sigmoid")
```

But replace the loss function

```
 model.compile(optimizer="rmsprop",
                    # Standard loss for binary classification with probability output
                    #loss="binary_crossentropy",
                    loss="mse",
                    metrics=["accuracy"])
    return model
```

Result:

Test Loss: 0.0855

Test Accuracy: 0.8853

Summary:

Configuration	Test Loss (L)	Test Accuracy (Acc)	Interpretation of Loss
Model 1 (With 2 hidden layers)	0.2852 (Binary Crossentropy)	**0.8874**	Best measure of probabilistic difference.
Model 2 (Alternative C: MSE Loss)	**0.0855** (Mean Squared Error)	0.8853	Lower numerical distance between predictions and targets.
Model 3 (With 3 hidden layers)	0.3059 (Binary Crossentropy)	0.8812	Highest loss, lowest accuracy among the three.

From the sheet we can tell:

For binary classification tasks, binary_crossentropy is the preferred choice of loss function.

For tasks that are relatively simple or do not have exceptionally large datasets, increasing model capacity (by adding more hidden layers) often leads to accelerated overfitting.