Name: Gaurav Kedia

Roll No: 39

Batch: E2

Write a program to implement neural network on sonar dataset. Compare the result of

1. Backpropogation with adam optimizer
2. L1 and L2 regularization
3. Early stopping with p=5
4. dropout=0.2 (For input layers and 2nd hidden layer)

(No. of hidden layers=2 for all and no. of iterations=200)

```
import pandas as pd
import numpy as np
from keras.models import Sequential
from keras.layers import Dense
from keras.optimizers import Adam
from sklearn.model_selection import train_test_split
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
```

```
df=pd.read_csv("/content/sonar.csv",header=None)
df
```

	0	1	2	3	4	5	6	7	8	9	...
0	0.0200	0.0371	0.0428	0.0207	0.0954	0.0986	0.1539	0.1601	0.3109	0.2111	...
1	0.0453	0.0523	0.0843	0.0689	0.1183	0.2583	0.2156	0.3481	0.3337	0.2872	...
2	0.0262	0.0582	0.1099	0.1083	0.0974	0.2280	0.2431	0.3771	0.5598	0.6194	...
3	0.0100	0.0171	0.0623	0.0205	0.0205	0.0368	0.1098	0.1276	0.0598	0.1264	...
4	0.0762	0.0666	0.0481	0.0394	0.0590	0.0649	0.1209	0.2467	0.3564	0.4459	...
...	...	...	...	...	...	...	...	...	...	...	...
203	0.0187	0.0346	0.0168	0.0177	0.0393	0.1630	0.2028	0.1694	0.2328	0.2684	...
204	0.0323	0.0101	0.0298	0.0564	0.0760	0.0958	0.0990	0.1018	0.1030	0.2154	...
205	0.0522	0.0437	0.0180	0.0292	0.0351	0.1171	0.1257	0.1178	0.1258	0.2529	...
206	0.0303	0.0353	0.0490	0.0608	0.0167	0.1354	0.1465	0.1123	0.1945	0.2354	...
207	0.0260	0.0363	0.0136	0.0272	0.0214	0.0338	0.0655	0.1400	0.1843	0.2354	...

208 rows × 61 columns

```
from sklearn.model_selection import train_test_split

X=df.loc[:,0:59]
y=df.loc[:,60]


X_train, X_test, y_train, y_test = train_test_split(X, y, test_size=20, random_state=4)
```

```
encoder = LabelEncoder()
y = encoder.fit_transform(y)

model = Sequential()
model.add(Dense(16, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(8, activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer=Adam(lr=0.001), metrics=['accuracy'])

model.fit(X_train, y_train, epochs=100, batch_size=8, verbose=1)

y_pred = model.predict(X_test)
y_pred = np.round(y_pred).astype(int)
accuracy = accuracy_score(y_test, y_pred)
print(f"Accuracy: {accuracy}")
```

```
Epoch 1/100
/usr/local/lib/python3.9/dist-packages/keras/optimizers/legacy/adam.py:117: UserWarning: The `lr` argument is deprecated, use `l
  super().__init__(name, **kwargs)
21/21 [==============================] - 0s 2ms/step - loss: 0.6957 - accuracy: 0.5120
Epoch 2/100
21/21 [==============================] - 0s 2ms/step - loss: 0.6888 - accuracy: 0.5120
Epoch 3/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6858 - accuracy: 0.5181
Epoch 4/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6811 - accuracy: 0.5663
Epoch 5/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6766 - accuracy: 0.5482
Epoch 6/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6711 - accuracy: 0.5663
Epoch 7/100
21/21 [==============================] - 0s 2ms/step - loss: 0.6636 - accuracy: 0.5843
Epoch 8/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6551 - accuracy: 0.6386
Epoch 9/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6435 - accuracy: 0.6084
Epoch 10/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6322 - accuracy: 0.6988
Epoch 11/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6206 - accuracy: 0.7349
Epoch 12/100
21/21 [==============================] - 0s 2ms/step - loss: 0.6105 - accuracy: 0.6867
Epoch 13/100
21/21 [==============================] - 0s 1ms/step - loss: 0.6023 - accuracy: 0.7169
Epoch 14/100
21/21 [==============================] - 0s 1ms/step - loss: 0.5981 - accuracy: 0.6566
Epoch 15/100
21/21 [==============================] - 0s 1ms/step - loss: 0.5755 - accuracy: 0.7108
Epoch 16/100
21/21 [==============================] - 0s 1ms/step - loss: 0.5635 - accuracy: 0.7349
Epoch 17/100
21/21 [==============================] - 0s 1ms/step - loss: 0.5545 - accuracy: 0.7590
Epoch 18/100
21/21 [==============================] - 0s 2ms/step - loss: 0.5422 - accuracy: 0.7108
Epoch 19/100
21/21 [==============================] - 0s 2ms/step - loss: 0.5398 - accuracy: 0.7410
Epoch 20/100
21/21 [==============================] - 0s 2ms/step - loss: 0.5218 - accuracy: 0.7410
Epoch 21/100
21/21 [==============================] - 0s 2ms/step - loss: 0.5125 - accuracy: 0.7530
Epoch 22/100
21/21 [==============================] - 0s 2ms/step - loss: 0.5073 - accuracy: 0.7771
Epoch 23/100
21/21 [==============================] - 0s 1ms/step - loss: 0.4969 - accuracy: 0.7651
Epoch 24/100
21/21 [==============================] - 0s 2ms/step - loss: 0.4915 - accuracy: 0.8012
Epoch 25/100
21/21 [==============================] - 0s 2ms/step - loss: 0.4820 - accuracy: 0.7952
Epoch 26/100
21/21 [==============================] - 0s 2ms/step - loss: 0.4782 - accuracy: 0.7771
Epoch 27/100
21/21 [==============================] - 0s 2ms/step - loss: 0.4706 - accuracy: 0.7831
Epoch 28/100
```

```
history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)

loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
```

```
Epoch 1/50
6/6 [==============================] - 0s 36ms/step - loss: 0.0941 - accuracy: 0.9880 - val_loss: 0.3580 - val_accuracy: 0.8333
Epoch 2/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0963 - accuracy: 0.9880 - val_loss: 0.3618 - val_accuracy: 0.8333
Epoch 3/50
```

```
6/6 [==============================] - 0s 6ms/step - loss: 0.0935 - accuracy: 0.9880 - val_loss: 0.3581 - val_accuracy: 0.8095
Epoch 4/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0936 - accuracy: 0.9759 - val_loss: 0.3567 - val_accuracy: 0.8095
Epoch 5/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0939 - accuracy: 0.9759 - val_loss: 0.3545 - val_accuracy: 0.8095
Epoch 6/50
6/6 [==============================] - 0s 10ms/step - loss: 0.0911 - accuracy: 0.9819 - val_loss: 0.3550 - val_accuracy: 0.8095
Epoch 7/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0904 - accuracy: 0.9940 - val_loss: 0.3546 - val_accuracy: 0.8095
Epoch 8/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0896 - accuracy: 0.9880 - val_loss: 0.3535 - val_accuracy: 0.8095
Epoch 9/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0903 - accuracy: 0.9880 - val_loss: 0.3561 - val_accuracy: 0.8095
Epoch 10/50
6/6 [==============================] - 0s 8ms/step - loss: 0.0895 - accuracy: 0.9880 - val_loss: 0.3609 - val_accuracy: 0.8095
Epoch 11/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0880 - accuracy: 0.9880 - val_loss: 0.3587 - val_accuracy: 0.8095
Epoch 12/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0889 - accuracy: 0.9819 - val_loss: 0.3594 - val_accuracy: 0.8095
Epoch 13/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0871 - accuracy: 0.9940 - val_loss: 0.3610 - val_accuracy: 0.8333
Epoch 14/50
6/6 [==============================] - 0s 10ms/step - loss: 0.0873 - accuracy: 0.9880 - val_loss: 0.3601 - val_accuracy: 0.8333
Epoch 15/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0853 - accuracy: 0.9940 - val_loss: 0.3582 - val_accuracy: 0.8095
Epoch 16/50
6/6 [==============================] - 0s 8ms/step - loss: 0.0852 - accuracy: 0.9880 - val_loss: 0.3587 - val_accuracy: 0.8095
Epoch 17/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0851 - accuracy: 0.9880 - val_loss: 0.3596 - val_accuracy: 0.8095
Epoch 18/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0836 - accuracy: 0.9880 - val_loss: 0.3600 - val_accuracy: 0.8095
Epoch 19/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0843 - accuracy: 0.9880 - val_loss: 0.3621 - val_accuracy: 0.8095
Epoch 20/50
6/6 [==============================] - 0s 8ms/step - loss: 0.0835 - accuracy: 0.9880 - val_loss: 0.3594 - val_accuracy: 0.8095
Epoch 21/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0849 - accuracy: 0.9819 - val_loss: 0.3571 - val_accuracy: 0.8095
Epoch 22/50
6/6 [==============================] - 0s 11ms/step - loss: 0.0838 - accuracy: 0.9819 - val_loss: 0.3584 - val_accuracy: 0.8095
Epoch 23/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0820 - accuracy: 0.9880 - val_loss: 0.3572 - val_accuracy: 0.8095
Epoch 24/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0802 - accuracy: 0.9880 - val_loss: 0.3579 - val_accuracy: 0.8095
Epoch 25/50
6/6 [==============================] - 0s 7ms/step - loss: 0.0800 - accuracy: 0.9940 - val_loss: 0.3588 - val_accuracy: 0.8095
Epoch 26/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0796 - accuracy: 0.9940 - val_loss: 0.3593 - val_accuracy: 0.8095
Epoch 27/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0792 - accuracy: 0.9880 - val_loss: 0.3587 - val_accuracy: 0.8095
Epoch 28/50
6/6 [==============================] - 0s 6ms/step - loss: 0.0787 - accuracy: 0.9880 - val_loss: 0.3608 - val_accuracy: 0.8095
Epoch 29/50
```

```
from keras.regularizers import l1
model = Sequential()
model.add(Dense(32, input_dim=X_train.shape[1], activation='relu', kernel_regularizer=l1(0.01)))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)

loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
```

```
Epoch 1/50
6/6 [==============================] - 1s 38ms/step - loss: 3.1370 - accuracy: 0.4940 - val_loss: 3.0765 - val_accuracy: 0.3571
Epoch 2/50
6/6 [==============================] - 0s 7ms/step - loss: 3.0212 - accuracy: 0.4337 - val_loss: 2.9569 - val_accuracy: 0.4048
Epoch 3/50
6/6 [==============================] - 0s 11ms/step - loss: 2.9170 - accuracy: 0.4759 - val_loss: 2.8498 - val_accuracy: 0.5238
Epoch 4/50
6/6 [==============================] - 0s 10ms/step - loss: 2.8214 - accuracy: 0.4940 - val_loss: 2.7528 - val_accuracy: 0.5000
Epoch 5/50
6/6 [==============================] - 0s 6ms/step - loss: 2.7265 - accuracy: 0.5301 - val_loss: 2.6654 - val_accuracy: 0.5952
Epoch 6/50
6/6 [==============================] - 0s 6ms/step - loss: 2.6359 - accuracy: 0.5482 - val_loss: 2.5813 - val_accuracy: 0.5238
Epoch 7/50
6/6 [==============================] - 0s 6ms/step - loss: 2.5468 - accuracy: 0.5663 - val_loss: 2.4946 - val_accuracy: 0.5000
Epoch 8/50
6/6 [==============================] - 0s 10ms/step - loss: 2.4616 - accuracy: 0.5301 - val_loss: 2.4105 - val_accuracy: 0.5714
Epoch 9/50
6/6 [==============================] - 0s 7ms/step - loss: 2.3744 - accuracy: 0.5964 - val_loss: 2.3162 - val_accuracy: 0.5952
Epoch 10/50
6/6 [==============================] - 0s 6ms/step - loss: 2.2923 - accuracy: 0.6024 - val_loss: 2.2286 - val_accuracy: 0.5714
Epoch 11/50
6/6 [==============================] - 0s 7ms/step - loss: 2.2141 - accuracy: 0.6084 - val_loss: 2.1466 - val_accuracy: 0.6190
Epoch 12/50
```

```
6/6 [==============================] - 0s 6ms/step - loss: 2.1361 - accuracy: 0.6325 - val_loss: 2.0755 - val_accuracy: 0.5952
Epoch 13/50
6/6 [==============================] - 0s 7ms/step - loss: 2.0601 - accuracy: 0.6084 - val_loss: 2.0028 - val_accuracy: 0.6667
Epoch 14/50
6/6 [==============================] - 0s 10ms/step - loss: 1.9867 - accuracy: 0.6506 - val_loss: 1.9328 - val_accuracy: 0.6905
Epoch 15/50
6/6 [==============================] - 0s 7ms/step - loss: 1.9154 - accuracy: 0.6446 - val_loss: 1.8607 - val_accuracy: 0.6905
Epoch 16/50
6/6 [==============================] - 0s 7ms/step - loss: 1.8460 - accuracy: 0.6687 - val_loss: 1.7957 - val_accuracy: 0.7619
Epoch 17/50
6/6 [==============================] - 0s 7ms/step - loss: 1.7801 - accuracy: 0.6867 - val_loss: 1.7363 - val_accuracy: 0.7619
Epoch 18/50
6/6 [==============================] - 0s 6ms/step - loss: 1.7165 - accuracy: 0.6747 - val_loss: 1.6746 - val_accuracy: 0.7619
Epoch 19/50
6/6 [==============================] - 0s 7ms/step - loss: 1.6540 - accuracy: 0.6747 - val_loss: 1.6089 - val_accuracy: 0.7619
Epoch 20/50
6/6 [==============================] - 0s 7ms/step - loss: 1.5934 - accuracy: 0.6988 - val_loss: 1.5458 - val_accuracy: 0.7857
Epoch 21/50
6/6 [==============================] - 0s 6ms/step - loss: 1.5345 - accuracy: 0.6928 - val_loss: 1.4872 - val_accuracy: 0.7619
Epoch 22/50
6/6 [==============================] - 0s 7ms/step - loss: 1.4782 - accuracy: 0.6747 - val_loss: 1.4276 - val_accuracy: 0.6667
Epoch 23/50
6/6 [==============================] - 0s 7ms/step - loss: 1.4241 - accuracy: 0.6867 - val_loss: 1.3731 - val_accuracy: 0.6905
Epoch 24/50
6/6 [==============================] - 0s 7ms/step - loss: 1.3721 - accuracy: 0.6807 - val_loss: 1.3212 - val_accuracy: 0.6905
Epoch 25/50
6/6 [==============================] - 0s 6ms/step - loss: 1.3223 - accuracy: 0.6747 - val_loss: 1.2769 - val_accuracy: 0.7619
Epoch 26/50
6/6 [==============================] - 0s 7ms/step - loss: 1.2752 - accuracy: 0.6867 - val_loss: 1.2326 - val_accuracy: 0.7619
Epoch 27/50
6/6 [==============================] - 0s 10ms/step - loss: 1.2299 - accuracy: 0.6867 - val_loss: 1.1963 - val_accuracy: 0.7381
Epoch 28/50
6/6 [==============================] - 0s 6ms/step - loss: 1.1895 - accuracy: 0.6747 - val_loss: 1.1557 - val_accuracy: 0.7857
Epoch 29/50
6/6 [==============================] - 0s 7ms/step - loss: 1.1491 - accuracy: 0.6627 - val_loss: 1.1145 - val_accuracy: 0.7857
```

```
from keras.regularizers import l2
model = Sequential()
model.add(Dense(32, input_dim=X_train.shape[1], activation='relu', kernel_regularizer=l2(0.01)))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)

loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
```

```
Epoch 1/50
6/6 [==============================] - 1s 38ms/step - loss: 1.1373 - accuracy: 0.4880 - val_loss: 1.1352 - val_accuracy: 0.3810
Epoch 2/50
6/6 [==============================] - 0s 6ms/step - loss: 1.0938 - accuracy: 0.4880 - val_loss: 1.0816 - val_accuracy: 0.4048
Epoch 3/50
6/6 [==============================] - 0s 6ms/step - loss: 1.0597 - accuracy: 0.4880 - val_loss: 1.0478 - val_accuracy: 0.4286
Epoch 4/50
6/6 [==============================] - 0s 6ms/step - loss: 1.0352 - accuracy: 0.4639 - val_loss: 1.0169 - val_accuracy: 0.6429
Epoch 5/50
6/6 [==============================] - 0s 7ms/step - loss: 1.0133 - accuracy: 0.5361 - val_loss: 0.9919 - val_accuracy: 0.6190
Epoch 6/50
6/6 [==============================] - 0s 6ms/step - loss: 0.9930 - accuracy: 0.5723 - val_loss: 0.9748 - val_accuracy: 0.6429
Epoch 7/50
6/6 [==============================] - 0s 7ms/step - loss: 0.9730 - accuracy: 0.5843 - val_loss: 0.9560 - val_accuracy: 0.6429
Epoch 8/50
6/6 [==============================] - 0s 7ms/step - loss: 0.9549 - accuracy: 0.6024 - val_loss: 0.9382 - val_accuracy: 0.6429
Epoch 9/50
6/6 [==============================] - 0s 6ms/step - loss: 0.9374 - accuracy: 0.6084 - val_loss: 0.9205 - val_accuracy: 0.6905
Epoch 10/50
6/6 [==============================] - 0s 10ms/step - loss: 0.9206 - accuracy: 0.6446 - val_loss: 0.9030 - val_accuracy: 0.7381
Epoch 11/50
6/6 [==============================] - 0s 13ms/step - loss: 0.9047 - accuracy: 0.6506 - val_loss: 0.8869 - val_accuracy: 0.7857
Epoch 12/50
6/6 [==============================] - 0s 13ms/step - loss: 0.8892 - accuracy: 0.7169 - val_loss: 0.8691 - val_accuracy: 0.8095
Epoch 13/50
6/6 [==============================] - 0s 9ms/step - loss: 0.8753 - accuracy: 0.6747 - val_loss: 0.8503 - val_accuracy: 0.8095
Epoch 14/50
6/6 [==============================] - 0s 8ms/step - loss: 0.8621 - accuracy: 0.6988 - val_loss: 0.8337 - val_accuracy: 0.7619
Epoch 15/50
6/6 [==============================] - 0s 12ms/step - loss: 0.8491 - accuracy: 0.7048 - val_loss: 0.8188 - val_accuracy: 0.7857
Epoch 16/50
6/6 [==============================] - 0s 11ms/step - loss: 0.8353 - accuracy: 0.6988 - val_loss: 0.8085 - val_accuracy: 0.8333
Epoch 17/50
6/6 [==============================] - 0s 10ms/step - loss: 0.8217 - accuracy: 0.7048 - val_loss: 0.7996 - val_accuracy: 0.8571
Epoch 18/50
6/6 [==============================] - 0s 8ms/step - loss: 0.8113 - accuracy: 0.7229 - val_loss: 0.7959 - val_accuracy: 0.8333
Epoch 19/50
6/6 [==============================] - 0s 12ms/step - loss: 0.8010 - accuracy: 0.6928 - val_loss: 0.7842 - val_accuracy: 0.8333
Epoch 20/50
6/6 [==============================] - 0s 11ms/step - loss: 0.7895 - accuracy: 0.7229 - val_loss: 0.7692 - val_accuracy: 0.8333
Epoch 21/50
```

```
6/6 [==============================] - 0s 8ms/step - loss: 0.7786 - accuracy: 0.7530 - val_loss: 0.7577 - val_accuracy: 0.8333
Epoch 22/50
6/6 [==============================] - 0s 8ms/step - loss: 0.7685 - accuracy: 0.7651 - val_loss: 0.7440 - val_accuracy: 0.8571
Epoch 23/50
6/6 [==============================] - 0s 8ms/step - loss: 0.7578 - accuracy: 0.7470 - val_loss: 0.7231 - val_accuracy: 0.8095
Epoch 24/50
6/6 [==============================] - 0s 11ms/step - loss: 0.7522 - accuracy: 0.6928 - val_loss: 0.7063 - val_accuracy: 0.8333
Epoch 25/50
6/6 [==============================] - 0s 12ms/step - loss: 0.7472 - accuracy: 0.6747 - val_loss: 0.6953 - val_accuracy: 0.8095
Epoch 26/50
6/6 [==============================] - 0s 12ms/step - loss: 0.7341 - accuracy: 0.6988 - val_loss: 0.6947 - val_accuracy: 0.8333
Epoch 27/50
6/6 [==============================] - 0s 7ms/step - loss: 0.7247 - accuracy: 0.7530 - val_loss: 0.6909 - val_accuracy: 0.8571
Epoch 28/50
6/6 [==============================] - 0s 7ms/step - loss: 0.7167 - accuracy: 0.7771 - val_loss: 0.6881 - val_accuracy: 0.8571
Epoch 29/50
6/6 [==============================] - 0s 11ms/step - loss: 0.7108 - accuracy: 0.7771 - val_loss: 0.6793 - val_accuracy: 0.8571
```

```
from keras.callbacks import EarlyStopping
model = Sequential()
model.add(Dense(32, input_dim=X_train.shape[1], activation='relu'))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

early_stop = EarlyStopping(monitor='val_loss', patience=5)

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32, callbacks=[early_stop])

loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
```

```
Epoch 1/50
6/6 [==============================] - 1s 33ms/step - loss: 0.7474 - accuracy: 0.4880 - val_loss: 0.7616 - val_accuracy: 0.3810
Epoch 2/50
6/6 [==============================] - 0s 7ms/step - loss: 0.7111 - accuracy: 0.4819 - val_loss: 0.7050 - val_accuracy: 0.3571
Epoch 3/50
6/6 [==============================] - 0s 8ms/step - loss: 0.6918 - accuracy: 0.5060 - val_loss: 0.6797 - val_accuracy: 0.5476
Epoch 4/50
6/6 [==============================] - 0s 11ms/step - loss: 0.6842 - accuracy: 0.5422 - val_loss: 0.6634 - val_accuracy: 0.6429
Epoch 5/50
6/6 [==============================] - 0s 9ms/step - loss: 0.6772 - accuracy: 0.5904 - val_loss: 0.6553 - val_accuracy: 0.6429
Epoch 6/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6708 - accuracy: 0.5843 - val_loss: 0.6475 - val_accuracy: 0.6905
Epoch 7/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6656 - accuracy: 0.5663 - val_loss: 0.6432 - val_accuracy: 0.6905
Epoch 8/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6614 - accuracy: 0.5964 - val_loss: 0.6421 - val_accuracy: 0.6429
Epoch 9/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6581 - accuracy: 0.6024 - val_loss: 0.6448 - val_accuracy: 0.6429
Epoch 10/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6531 - accuracy: 0.6265 - val_loss: 0.6314 - val_accuracy: 0.6190
Epoch 11/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6464 - accuracy: 0.6566 - val_loss: 0.6158 - val_accuracy: 0.7143
Epoch 12/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6397 - accuracy: 0.6627 - val_loss: 0.6148 - val_accuracy: 0.6905
Epoch 13/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6346 - accuracy: 0.6807 - val_loss: 0.6084 - val_accuracy: 0.6905
Epoch 14/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6298 - accuracy: 0.6747 - val_loss: 0.5942 - val_accuracy: 0.7381
Epoch 15/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6242 - accuracy: 0.6747 - val_loss: 0.5927 - val_accuracy: 0.7143
Epoch 16/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6200 - accuracy: 0.6867 - val_loss: 0.5850 - val_accuracy: 0.7619
Epoch 17/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6148 - accuracy: 0.6807 - val_loss: 0.5734 - val_accuracy: 0.7857
Epoch 18/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6098 - accuracy: 0.6687 - val_loss: 0.5647 - val_accuracy: 0.8095
Epoch 19/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6052 - accuracy: 0.6747 - val_loss: 0.5613 - val_accuracy: 0.8095
Epoch 20/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5997 - accuracy: 0.7108 - val_loss: 0.5667 - val_accuracy: 0.8571
Epoch 21/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5954 - accuracy: 0.7470 - val_loss: 0.5685 - val_accuracy: 0.7857
Epoch 22/50
6/6 [==============================] - 0s 8ms/step - loss: 0.5905 - accuracy: 0.7349 - val_loss: 0.5575 - val_accuracy: 0.8333
Epoch 23/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5845 - accuracy: 0.7349 - val_loss: 0.5434 - val_accuracy: 0.8810
Epoch 24/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5801 - accuracy: 0.7590 - val_loss: 0.5358 - val_accuracy: 0.8810
Epoch 25/50
6/6 [==============================] - 0s 10ms/step - loss: 0.5757 - accuracy: 0.7590 - val_loss: 0.5327 - val_accuracy: 0.8810
Epoch 26/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5712 - accuracy: 0.7470 - val_loss: 0.5302 - val_accuracy: 0.8810
Epoch 27/50
6/6 [==============================] - 0s 6ms/step - loss: 0.5674 - accuracy: 0.7590 - val_loss: 0.5225 - val_accuracy: 0.8810
Epoch 28/50
6/6 [==============================] - 0s 7ms/step - loss: 0.5628 - accuracy: 0.7530 - val_loss: 0.5163 - val_accuracy: 0.8810
Epoch 29/50
```

```
6/6 [==============================] - 0s 7ms/step - loss: 0.5587 - accuracy: 0.7530 - val_loss: 0.5147 - val_accuracy: 0.8571
```

```
from keras.layers import Dense, Dropout
model = Sequential()
model.add(Dropout(0.2, input_shape=(X_train.shape[1],)))
model.add(Dense(32, activation='relu'))
model.add(Dropout(0.2))
model.add(Dense(1, activation='sigmoid'))

model.compile(loss='binary_crossentropy', optimizer='adam', metrics=['accuracy'])

history = model.fit(X_train, y_train, validation_data=(X_test, y_test), epochs=50, batch_size=32)

loss, accuracy = model.evaluate(X_test, y_test)
print('Test accuracy:', accuracy)
```

```
Epoch 1/50
6/6 [==============================] - 1s 34ms/step - loss: 0.7387 - accuracy: 0.5181 - val_loss: 0.6761 - val_accuracy: 0.6429
Epoch 2/50
6/6 [==============================] - 0s 10ms/step - loss: 0.7489 - accuracy: 0.4639 - val_loss: 0.6875 - val_accuracy: 0.5714
Epoch 3/50
6/6 [==============================] - 0s 6ms/step - loss: 0.7154 - accuracy: 0.4880 - val_loss: 0.6868 - val_accuracy: 0.5476
Epoch 4/50
6/6 [==============================] - 0s 6ms/step - loss: 0.7220 - accuracy: 0.4639 - val_loss: 0.6798 - val_accuracy: 0.5714
Epoch 5/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6887 - accuracy: 0.5602 - val_loss: 0.6796 - val_accuracy: 0.5714
Epoch 6/50
6/6 [==============================] - 0s 9ms/step - loss: 0.6907 - accuracy: 0.5181 - val_loss: 0.6742 - val_accuracy: 0.6190
Epoch 7/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6763 - accuracy: 0.5783 - val_loss: 0.6593 - val_accuracy: 0.6429
Epoch 8/50
6/6 [==============================] - 0s 7ms/step - loss: 0.7033 - accuracy: 0.4940 - val_loss: 0.6518 - val_accuracy: 0.6667
Epoch 9/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6562 - accuracy: 0.5542 - val_loss: 0.6423 - val_accuracy: 0.6905
Epoch 10/50
6/6 [==============================] - 0s 11ms/step - loss: 0.6840 - accuracy: 0.5301 - val_loss: 0.6344 - val_accuracy: 0.7143
Epoch 11/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6815 - accuracy: 0.5301 - val_loss: 0.6262 - val_accuracy: 0.7143
Epoch 12/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6715 - accuracy: 0.5964 - val_loss: 0.6178 - val_accuracy: 0.7143
Epoch 13/50
6/6 [==============================] - 0s 11ms/step - loss: 0.6688 - accuracy: 0.6024 - val_loss: 0.6159 - val_accuracy: 0.7381
Epoch 14/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6904 - accuracy: 0.5663 - val_loss: 0.6189 - val_accuracy: 0.7143
Epoch 15/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6553 - accuracy: 0.6325 - val_loss: 0.6109 - val_accuracy: 0.7381
Epoch 16/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6485 - accuracy: 0.6205 - val_loss: 0.6043 - val_accuracy: 0.7381
Epoch 17/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6616 - accuracy: 0.5843 - val_loss: 0.6012 - val_accuracy: 0.7143
Epoch 18/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6615 - accuracy: 0.5361 - val_loss: 0.5932 - val_accuracy: 0.7619
Epoch 19/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6582 - accuracy: 0.6024 - val_loss: 0.5864 - val_accuracy: 0.7857
Epoch 20/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6599 - accuracy: 0.5843 - val_loss: 0.5828 - val_accuracy: 0.7857
Epoch 21/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6390 - accuracy: 0.6325 - val_loss: 0.5744 - val_accuracy: 0.8095
Epoch 22/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6375 - accuracy: 0.6446 - val_loss: 0.5726 - val_accuracy: 0.8333
Epoch 23/50
6/6 [==============================] - 0s 10ms/step - loss: 0.6424 - accuracy: 0.6386 - val_loss: 0.5678 - val_accuracy: 0.8333
Epoch 24/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6382 - accuracy: 0.6687 - val_loss: 0.5715 - val_accuracy: 0.8095
Epoch 25/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6257 - accuracy: 0.6084 - val_loss: 0.5741 - val_accuracy: 0.8333
Epoch 26/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6090 - accuracy: 0.6687 - val_loss: 0.5691 - val_accuracy: 0.8095
Epoch 27/50
6/6 [==============================] - 0s 6ms/step - loss: 0.6293 - accuracy: 0.6687 - val_loss: 0.5608 - val_accuracy: 0.8333
Epoch 28/50
6/6 [==============================] - 0s 11ms/step - loss: 0.6171 - accuracy: 0.6084 - val_loss: 0.5600 - val_accuracy: 0.8095
Epoch 29/50
6/6 [==============================] - 0s 7ms/step - loss: 0.6171 - accuracy: 0.6687 - val_loss: 0.5533 - val_accuracy: 0.8095
```