Task 1

Implement the OR Boolean logic gate using perceptron Neural Network. Inputs = x1, x2 and bias, weights should be fed into the perceptron with single Output = y. Display final weights and bias of each perceptron.

In [42]:

```
import numpy as np
```

In [43]:

```
def step(z):
    return 1 if z >= 1 else 0
```

In [44]:

```
w1 = 1
w2 = 1
b = 0

X =np.array([[0,0],
             [0,1],
             [1,0],
             [1,1]])
Y = np.array([0,1,1,1])
```

In [45]:

```
z = np.dot(X, np.array([w1, w2])) + b
predictions = np.array([step(i) for i in z])
```

In [46]:

```
predictions
```

```
Out[46]: array([0, 1, 1, 1])
```

Task 2

- Use the heart disease Dataset
- Create an Auto Encoder and fit it with our data using 3 neurons in the dense layer
- Display new reduced dimension values
- Plot loss for different Auto encoders

In [47]:

```
import pandas as pd
```

In [48]:

```
X = pd.read_csv('Data/heart.csv')
```

In [49]:

```
from sklearn.preprocessing import StandardScaler
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)
```

In [50]:

```
from tensorflow.keras.models import Sequential
from tensorflow.keras.layers import Dense, Input
```

In [51]:

```
autoencoder = Sequential([
    Input(shape = (X.shape[1], )),
    Dense(3, activation='relu'),
    Dense(X.shape[1], activation='sigmoid')
])
```

In [52]:

```
autoencoder.compile(optimizer='adam', loss='mse')
```

In [53]:

```
autoencoder.fit(X_scaled, X_scaled, epochs=50, batch_size=16, shuffle=True,
validation_split=0.2)
```

```
Epoch 1/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.2515
- val_loss: 1.1724
Epoch 2/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2214
- val_loss: 1.1460
Epoch 3/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1920
- val_loss: 1.1183
Epoch 4/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1609
- val_loss: 1.0897
Epoch 5/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1281
- val_loss: 1.0596
Epoch 6/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0948
- val_loss: 1.0293
Epoch 7/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0616
- val_loss: 0.9989
Epoch 8/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0309
- val_loss: 0.9719
Epoch 9/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0033
- val_loss: 0.9475
Epoch 10/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9790
- val_loss: 0.9265
Epoch 11/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9586
- val_loss: 0.9087
Epoch 12/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9414
- val_loss: 0.8940
Epoch 13/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 0.9272
- val_loss: 0.8813
Epoch 14/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9152
- val_loss: 0.8706
Epoch 15/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9053
- val_loss: 0.8615
Epoch 16/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8969
- val_loss: 0.8537
Epoch 17/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8897
- val_loss: 0.8470
Epoch 18/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8835
- val_loss: 0.8409
Epoch 19/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8780
- val_loss: 0.8354
Epoch 20/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8732
- val_loss: 0.8305
Epoch 21/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8689
- val_loss: 0.8262
Epoch 22/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8651
- val_loss: 0.8223
```

```
Epoch 23/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8616
- val_loss: 0.8187
Epoch 24/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8584
- val_loss: 0.8152
Epoch 25/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8556
- val_loss: 0.8124
Epoch 26/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8530
- val_loss: 0.8096
Epoch 27/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8506
- val_loss: 0.8071
Epoch 28/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8484
- val_loss: 0.8047
Epoch 29/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8463
- val_loss: 0.8027
Epoch 30/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8445
- val_loss: 0.8006
Epoch 31/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8427
- val_loss: 0.7985
Epoch 32/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8409
- val_loss: 0.7967
Epoch 33/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8393
- val_loss: 0.7944
Epoch 34/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8376
- val_loss: 0.7929
Epoch 35/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8362
- val_loss: 0.7914
Epoch 36/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8348
- val_loss: 0.7898
Epoch 37/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8335
- val_loss: 0.7883
Epoch 38/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8323
- val_loss: 0.7871
Epoch 39/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8312
- val_loss: 0.7858
Epoch 40/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8301
- val_loss: 0.7847
Epoch 41/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8290
- val_loss: 0.7837
Epoch 42/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8281
- val_loss: 0.7828
Epoch 43/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8272
- val_loss: 0.7816
Epoch 44/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8262
- val_loss: 0.7809
Epoch 45/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8253
- val_loss: 0.7800
Epoch 46/50
```

```
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8244
- val_loss: 0.7792
Epoch 47/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8236
- val_loss: 0.7784
Epoch 48/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8227
- val_loss: 0.7777
Epoch 49/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8220
- val_loss: 0.7771
Epoch 50/50
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8213
- val_loss: 0.7765
```

Out[53]:

```
<keras.src.callbacks.history.History at 0x1e91b9de390>
```

In [54]:

```
encoder  =Sequential([autoencoder.layers[0]])
encoded_data = encoder.predict(X_scaled)
```

```
[1m33/33[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 963us/step
```

The reduced dimention values are as follows:

In [55]:

```
encoded_data
```

Out[55]:

```
array([[2.6006954 , 0.        , 2.8786101 ],
       [4.6247406 , 4.141082  , 0.        ],
       [5.9165525 , 1.2400119 , 0.        ],
       ...,
       [6.4260826 , 0.27540517, 2.3368797 ],
       [0.2857455 , 2.299008  , 8.678795  ],
       [5.1289797 , 0.        , 1.1936338 ]],
      shape=(1025, 3), dtype=float32)
```

In [56]:

```
preds = []
loss = []
for i in range(1,5):
    autoencoder = Sequential([
        Input(shape = (X.shape[1], )),
        Dense(i, activation='relu'),
        Dense(X.shape[1], activation='sigmoid')
    ])
    autoencoder.compile(optimizer='adam', loss='mse')
    history = autoencoder.fit(X_scaled, X_scaled, epochs=20, batch_size=16,
shuffle=True, validation_split=0.2)
    preds.append(autoencoder.predict(X_scaled))
    loss.append(history.history['loss'])
    print(f'Encoding Dimension: {i}, Loss: {loss[-1]}')
```

```
Epoch 1/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.2385
- val_loss: 1.1537
Epoch 2/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2215
- val_loss: 1.1352
Epoch 3/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2050
- val_loss: 1.1178
Epoch 4/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1894
- val_loss: 1.1010
Epoch 5/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1743
- val_loss: 1.0853
Epoch 6/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1600
- val_loss: 1.0701
Epoch 7/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1465
- val_loss: 1.0565
Epoch 8/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1338
- val_loss: 1.0435
Epoch 9/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1218
- val_loss: 1.0313
Epoch 10/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1108
- val_loss: 1.0201
Epoch 11/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.1005
- val_loss: 1.0100
Epoch 12/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0909
- val_loss: 1.0009
Epoch 13/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0822
- val_loss: 0.9922
Epoch 14/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0742
- val_loss: 0.9845
Epoch 15/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0668
- val_loss: 0.9774
Epoch 16/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0602
- val_loss: 0.9715
Epoch 17/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0544
- val_loss: 0.9658
Epoch 18/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0488
- val_loss: 0.9607
```

```
Epoch 19/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0435
- val_loss: 0.9556
Epoch 20/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0385
- val_loss: 0.9513
[1m33/33[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 969us/step
Encoding Dimension: 1, Loss: [1.2384586334228516, 1.2215328216552734,
1.205014705657959, 1.1893672943115234, 1.1742677688598633, 1.1600158214569092,
1.1465078592300415, 1.133806586265564, 1.1218146085739136, 1.110803246498108,
1.1005158424377441, 1.0908856391906738, 1.0821701288223267, 1.0741585493087769,
1.0668381452560425, 1.0602295398712158, 1.0543524026870728, 1.048801302909851,
1.0435290336608887, 1.0384565591812134]
Epoch 1/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.2879
- val_loss: 1.2122
Epoch 2/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2598
- val_loss: 1.1866
Epoch 3/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2351
- val_loss: 1.1646
Epoch 4/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2125
- val_loss: 1.1445
Epoch 5/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1902
- val_loss: 1.1249
Epoch 6/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1679
- val_loss: 1.1048
Epoch 7/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1450
- val_loss: 1.0844
Epoch 8/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1219
- val_loss: 1.0637
Epoch 9/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0983
- val_loss: 1.0431
Epoch 10/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0747
- val_loss: 1.0222
Epoch 11/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0514
- val_loss: 1.0015
Epoch 12/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0292
- val_loss: 0.9820
Epoch 13/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0091
- val_loss: 0.9646
Epoch 14/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9918
- val_loss: 0.9499
Epoch 15/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9772
- val_loss: 0.9369
Epoch 16/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9650
- val_loss: 0.9264
Epoch 17/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9550
- val_loss: 0.9177
Epoch 18/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9469
- val_loss: 0.9100
Epoch 19/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9402
- val_loss: 0.9039
```

```
Epoch 20/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9348
- val_loss: 0.8985
[1m33/33[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 988us/step
Encoding Dimension: 2, Loss: [1.2878590822219849, 1.2598187923431396,
1.2351021766662598, 1.2124847173690796, 1.1901849508285522, 1.1678563356399536,
1.145045280456543, 1.121875524520874, 1.0983086824417114, 1.0746984481811523,
1.0514470338821411, 1.0292381048202515, 1.0090503692626953, 0.9917647838592529,
0.9772189259529114, 0.9649945497512817, 0.9550153017044067, 0.9468544721603394,
0.9402382969856262, 0.9347590208053589]
Epoch 1/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.3135
- val_loss: 1.2240
Epoch 2/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2749
- val_loss: 1.1902
Epoch 3/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2437
- val_loss: 1.1626
Epoch 4/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2171
- val_loss: 1.1377
Epoch 5/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1925
- val_loss: 1.1143
Epoch 6/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1682
- val_loss: 1.0899
Epoch 7/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1434
- val_loss: 1.0649
Epoch 8/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1180
- val_loss: 1.0392
Epoch 9/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0921
- val_loss: 1.0141
Epoch 10/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0669
- val_loss: 0.9899
Epoch 11/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0426
- val_loss: 0.9669
Epoch 12/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0191
- val_loss: 0.9453
Epoch 13/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9969
- val_loss: 0.9253
Epoch 14/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9764
- val_loss: 0.9074
Epoch 15/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9584
- val_loss: 0.8920
Epoch 16/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9429
- val_loss: 0.8790
Epoch 17/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9300
- val_loss: 0.8684
Epoch 18/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9193
- val_loss: 0.8597
Epoch 19/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9105
- val_loss: 0.8524
Epoch 20/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9033
- val_loss: 0.8465
```

```
[1m33/33[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 953us/step
Encoding Dimension: 3, Loss: [1.3135099411010742, 1.2749431133270264,
1.2437139749526978, 1.2170566320419312, 1.1925243139266968, 1.1682037115097046,
1.143405795097351, 1.1179910898208618, 1.092088222503662, 1.066867709159851,
1.0425903797149658, 1.01913583278656, 0.9968659281730652, 0.9763920307159424,
0.9583531022071838, 0.9428660869598389, 0.9299794435501099, 0.919308602809906,
0.9105072617530823, 0.9033305048942566]
Epoch 1/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 2ms/step - loss: 1.2920
- val_loss: 1.1891
Epoch 2/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2444
- val_loss: 1.1501
Epoch 3/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.2020
- val_loss: 1.1124
Epoch 4/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1620
- val_loss: 1.0776
Epoch 5/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.1245
- val_loss: 1.0448
Epoch 6/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0904
- val_loss: 1.0140
Epoch 7/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0595
- val_loss: 0.9858
Epoch 8/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0313
- val_loss: 0.9595
Epoch 9/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 1.0050
- val_loss: 0.9340
Epoch 10/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9805
- val_loss: 0.9111
Epoch 11/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9587
- val_loss: 0.8906
Epoch 12/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9398
- val_loss: 0.8732
Epoch 13/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9238
- val_loss: 0.8591
Epoch 14/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.9104
- val_loss: 0.8467
Epoch 15/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8991
- val_loss: 0.8363
Epoch 16/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8895
- val_loss: 0.8279
Epoch 17/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8814
- val_loss: 0.8209
Epoch 18/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8747
- val_loss: 0.8149
Epoch 19/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8689
- val_loss: 0.8099
Epoch 20/20
[1m52/52[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 1ms/step - loss: 0.8638
- val_loss: 0.8055
[1m33/33[0m [32m━━━━━━━━━━━━━━━━━━━━[0m[37m[0m [1m0s[0m 953us/step
Encoding Dimension: 4, Loss: [1.2919960021972656, 1.2444182634353638,
1.20195472240448, 1.1620312929153442, 1.1245213747024536, 1.0904380083084106,
1.0595048666000366, 1.0313063859939575, 1.004961609840393, 0.9805213212966919,
0.9586631059646606, 0.9397965669631958, 0.9238115549087524, 0.9103679656982422,
0.8990597128868103, 0.8894584774971008, 0.8814430236816406, 0.8746541738510132,
0.86887526512146, 0.8638182878494263]
```

In [57]:

```
loss
```

Out[57]:

```
[[1.2384586334228516,
  1.2215328216552734,
  1.205014705657959,
  1.1893672943115234,
  1.1742677688598633,
  1.1600158214569092,
  1.1465078592300415,
  1.133806586265564,
  1.1218146085739136,
  1.110803246498108,
  1.1005158424377441,
  1.0908856391906738,
  1.0821701288223267,
  1.0741585493087769,
  1.0668381452560425,
  1.0602295398712158,
  1.0543524026870728,
  1.048801302909851,
  1.0435290336608887,
  1.0384565591812134],
 [1.2878590822219849,
  1.2598187923431396,
  1.2351021766662598,
  1.2124847173690796,
  1.1901849508285522,
  1.1678563356399536,
  1.145045280456543,
  1.121875524520874,
  1.0983086824417114,
  1.0746984481811523,
  1.0514470338821411,
  1.0292381048202515,
  1.0090503692626953,
  0.9917647838592529,
  0.9772189259529114,
  0.9649945497512817,
  0.9550153017044067,
  0.9468544721603394,
  0.9402382969856262,
  0.9347590208053589],
 [1.3135099411010742,
  1.2749431133270264,
  1.2437139749526978,
  1.2170566320419312,
  1.1925243139266968,
  1.1682037115097046,
  1.143405795097351,
  1.1179910898208618,
  1.092088222503662,
  1.066867709159851,
  1.0425903797149658,
  1.01913583278656,
  0.9968659281730652,
  0.9763920307159424,
  0.9583531022071838,
  0.9428660869598389,
  0.9299794435501099,
  0.919308602809906,
  0.9105072617530823,
  0.9033305048942566],
 [1.2919960021972656,
  1.2444182634353638,
  1.20195472240448,
  1.1620312929153442,
  1.1245213747024536,
  1.0904380083084106,
  1.0595048666000366,
  1.0313063859939575,
  1.004961609840393,
  0.9805213212966919,
  0.9586631059646606,
  0.9397965669631958,
  0.9238115549087524,
  0.9103679656982422,
  0.8990597128868103,
  0.8894584774971008,
  0.8814430236816406,
  0.8746541738510132,
  0.86887526512146,
  0.8638182878494263]]
```

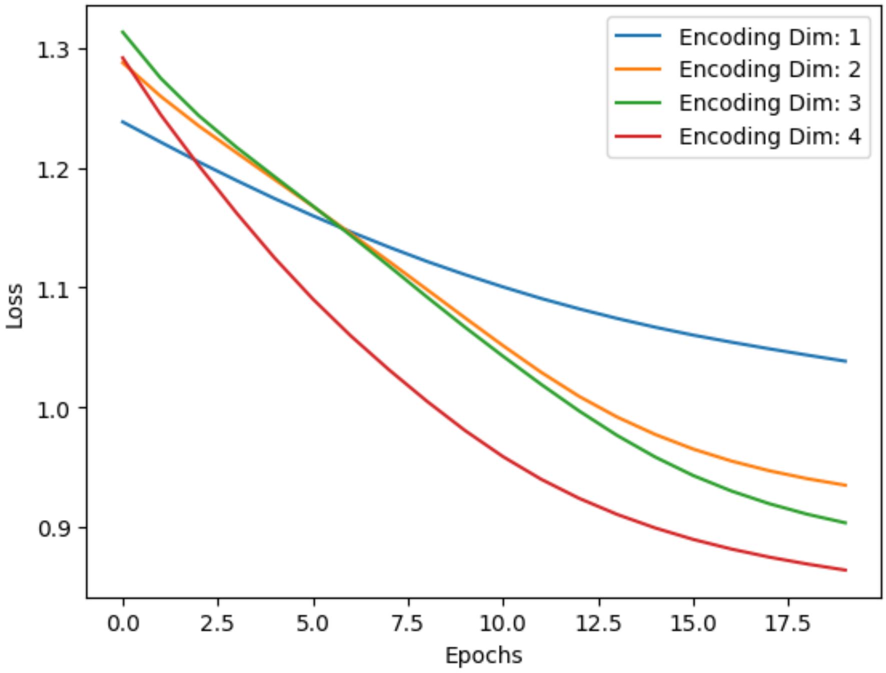
In [59]:

```
import matplotlib.pyplot as plt

for i in range(4):
    plt.plot(loss[i], label=f'Encoding Dim: {i+1}')
plt.xlabel('Epochs')
plt.ylabel('Loss')
plt.legend()
```

Out[59]:

```
<matplotlib.legend.Legend at 0x1e9253b64e0>
```


Encoding Dim: 1
Encoding Dim: 2
Encoding Dim: 3
Encoding Dim: 4
Loss
Epochs