**Lab 7**

**OBJECTIVE: Classifications Using Neural Networks**

**Lab Task**

1. **Increase or decrease number of neurons in hidden layers**

**CODE**

import pandas as pd

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.datasets import load\_breast\_cancer

breast\_cancer = load\_breast\_cancer()

df = pd.DataFrame(breast\_cancer.data, columns=breast\_cancer.feature\_names)

df['target'] = breast\_cancer.target

print(df.head())

X = breast\_cancer.data

y = breast\_cancer.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)))

model.add(Dense(32, activation='relu'))

model.add(Dense(16, activation='relu'))

model.add(Dense(1, activation='sigmoid')) # Binary classification

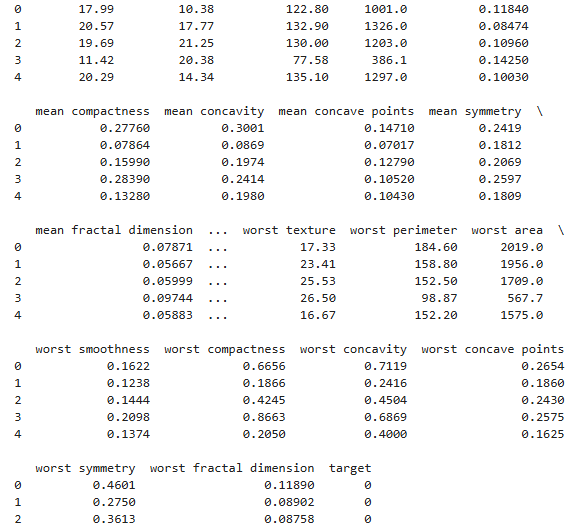
model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train\_scaled, y\_train, epochs=10, batch\_size=32, verbose=1)

loss, accuracy = model.evaluate(X\_test\_scaled, y\_test, verbose=0)

print("Accuracy on Test Data:", round(accuracy \* 100, 2), "%")

print("Loss Value:", round(loss \* 100, 2))





1. **Add more hidden layers**

**CODE**

import pandas as pd

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.datasets import load\_breast\_cancer

breast\_cancer = load\_breast\_cancer()

df = pd.DataFrame(breast\_cancer.data, columns=breast\_cancer.feature\_names)

df['target'] = breast\_cancer.target

print(df.head())

X = breast\_cancer.data

y = breast\_cancer.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)))

model.add(Dense(32, activation='relu'))



model.add(Dense(16, activation='relu'))

model.add(Dense(12, activation='relu'))

model.add(Dense(15, activation='relu'))

model.add(Dense(1, activation='sigmoid')) # Binary classification

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train\_scaled, y\_train, epochs=10, batch\_size=32, verbose=1)

loss, accuracy = model.evaluate(X\_test\_scaled, y\_test, verbose=0)

print("Accuracy on Test Data:", round(accuracy \* 100, 2), "%")

print("Loss Value:", round(loss \* 100, 2))

1. **Increase number of epochs**

import pandas as pd

from tensorflow.keras.models import Sequential

from tensorflow.keras.layers import Dense

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

from sklearn.datasets import load\_breast\_cancer

breast\_cancer = load\_breast\_cancer()

df = pd.DataFrame(breast\_cancer.data, columns=breast\_cancer.feature\_names)

df['target'] = breast\_cancer.target

print(df.head())

X = breast\_cancer.data

y = breast\_cancer.target

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

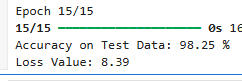
X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)))

model.add(Dense(32, activation='relu'))



model.add(Dense(16, activation='relu'))

model.add(Dense(12, activation='relu'))

model.add(Dense(15, activation='relu'))

model.add(Dense(1, activation='sigmoid')) # Binary classification

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train\_scaled, y\_train, epochs=15, batch\_size=32, verbose=1)

loss, accuracy = model.evaluate(X\_test\_scaled, y\_test, verbose=0)

print("Accuracy on Test Data:", round(accuracy \* 100, 2), "%")

print("Loss Value:", round(loss \* 100, 2))

1. **change the dataset to any easily available classification dataset on Kaggle do mention the link**

<https://www.kaggle.com/datasets/nancyalaswad90/breast-cancer-dataset>

CODE

import pandas as pd

from tensorflow.keras.models import Sequential



from tensorflow.keras.layers import Dense

from sklearn.model\_selection import train\_test\_split

from sklearn.preprocessing import StandardScaler

df = pd.read\_csv('data.csv')

print(df.head())

X = df.drop('diagnosis', axis=1).values

y = df['diagnosis'].values

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size=0.2, random\_state=42)

scaler = StandardScaler()

X\_train\_scaled = scaler.fit\_transform(X\_train)

X\_test\_scaled = scaler.transform(X\_test)

model = Sequential()

model.add(Dense(64, activation='relu', input\_shape=(X\_train\_scaled.shape[1],)))

model.add(Dense(32, activation='relu'))

model.add(Dense(16, activation='relu'))

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

model.compile(optimizer='adam', loss='binary\_crossentropy', metrics=['accuracy'])

model.fit(X\_train\_scaled, y\_train, epochs=5, batch\_size=32, verbose=1)

loss, accuracy = model.evaluate(X\_test\_scaled, y\_test, verbose=0)

print("Accuracy on Test Data:", round(accuracy \* 100, 2), "%")

print("Loss Value:", round(loss \* 100, 2))

