**AI Neural Network**

**Short note about Neural Network**:

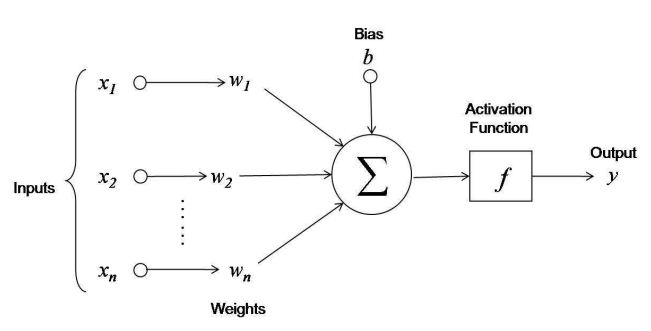
* **Machine learning** is the branch of computer science that has to do with building algorithms that are guided by data. Rather than relying on human programmers to provide explicit instructions, machine learning algorithms use training sets of real-world data to infer models that are more accurate and sophisticated than humans could devise on their own.

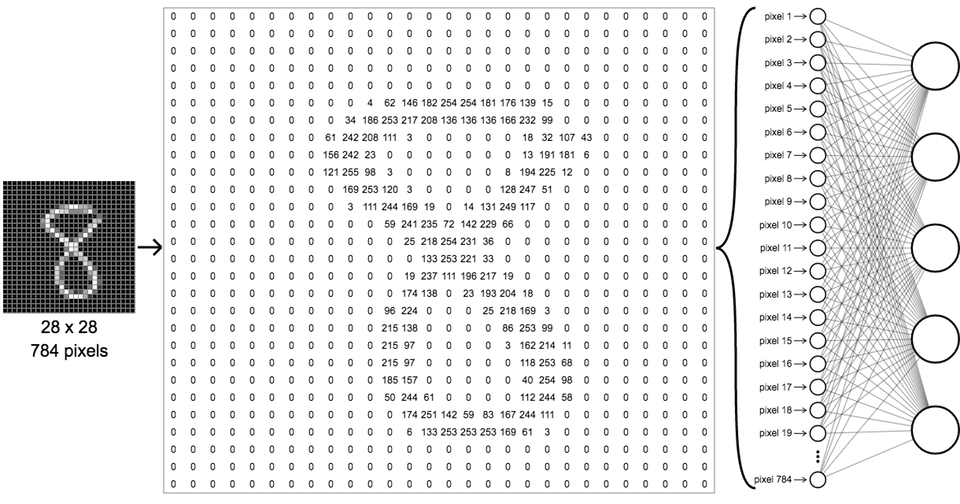
**Neural Networks-**

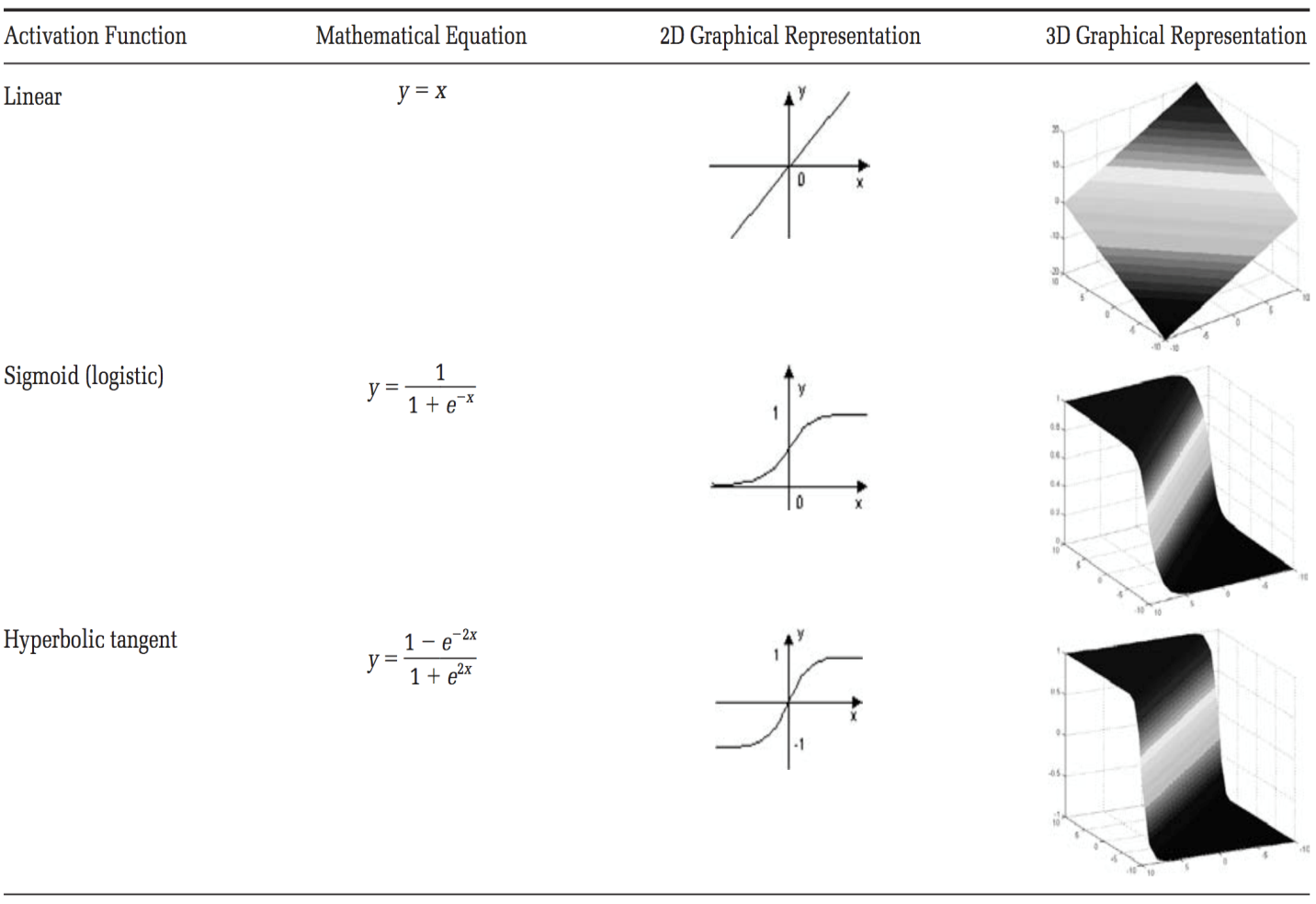
* Within the field of machine learning, **neural networks** are a subset of algorithms built around a model of artificial neurons spread across three or more layers (we’ll get into the details shortly). There are plenty of other machine learning techniques that don’t rely on neural networks.

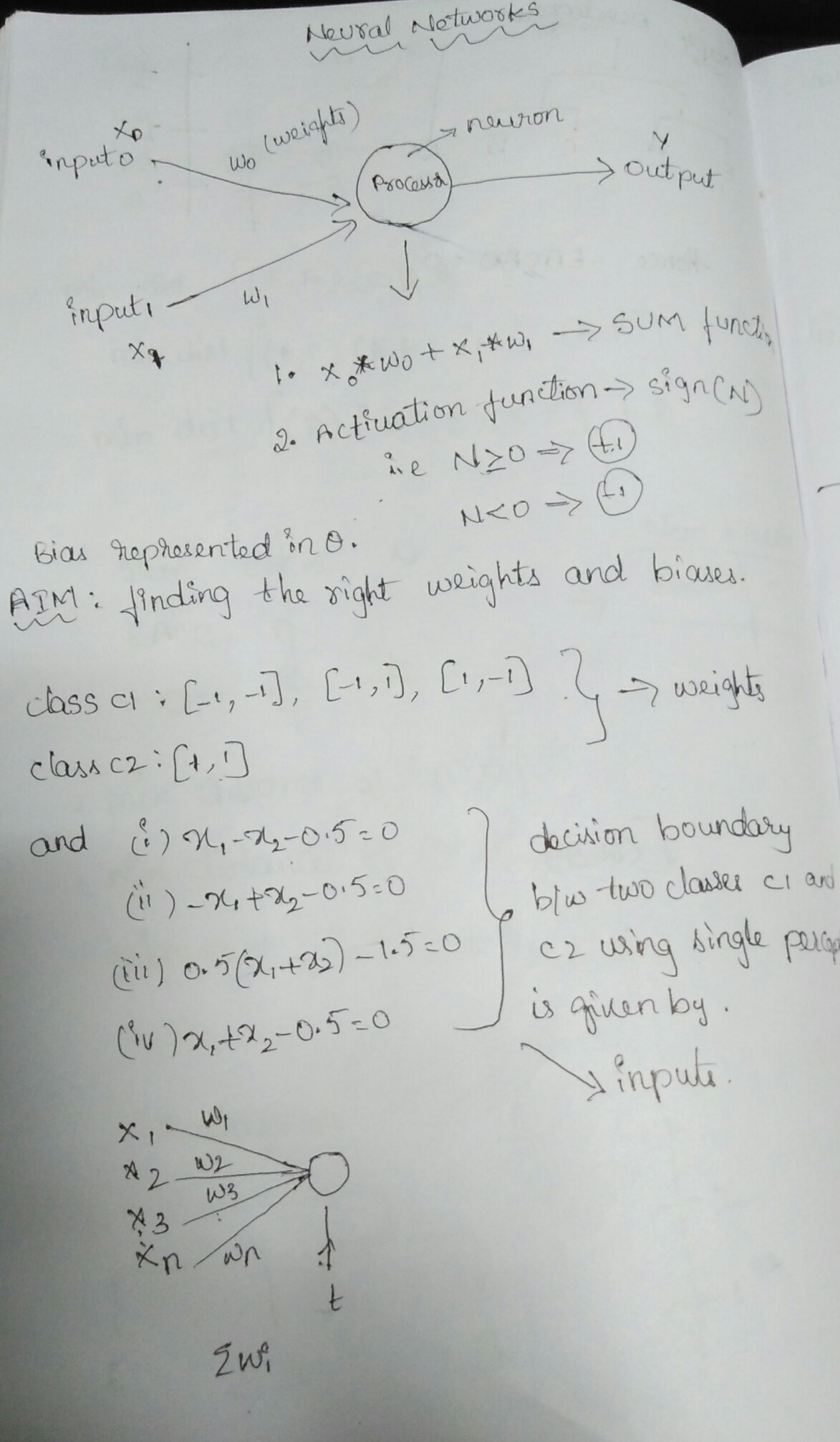
**DeepLearning-**

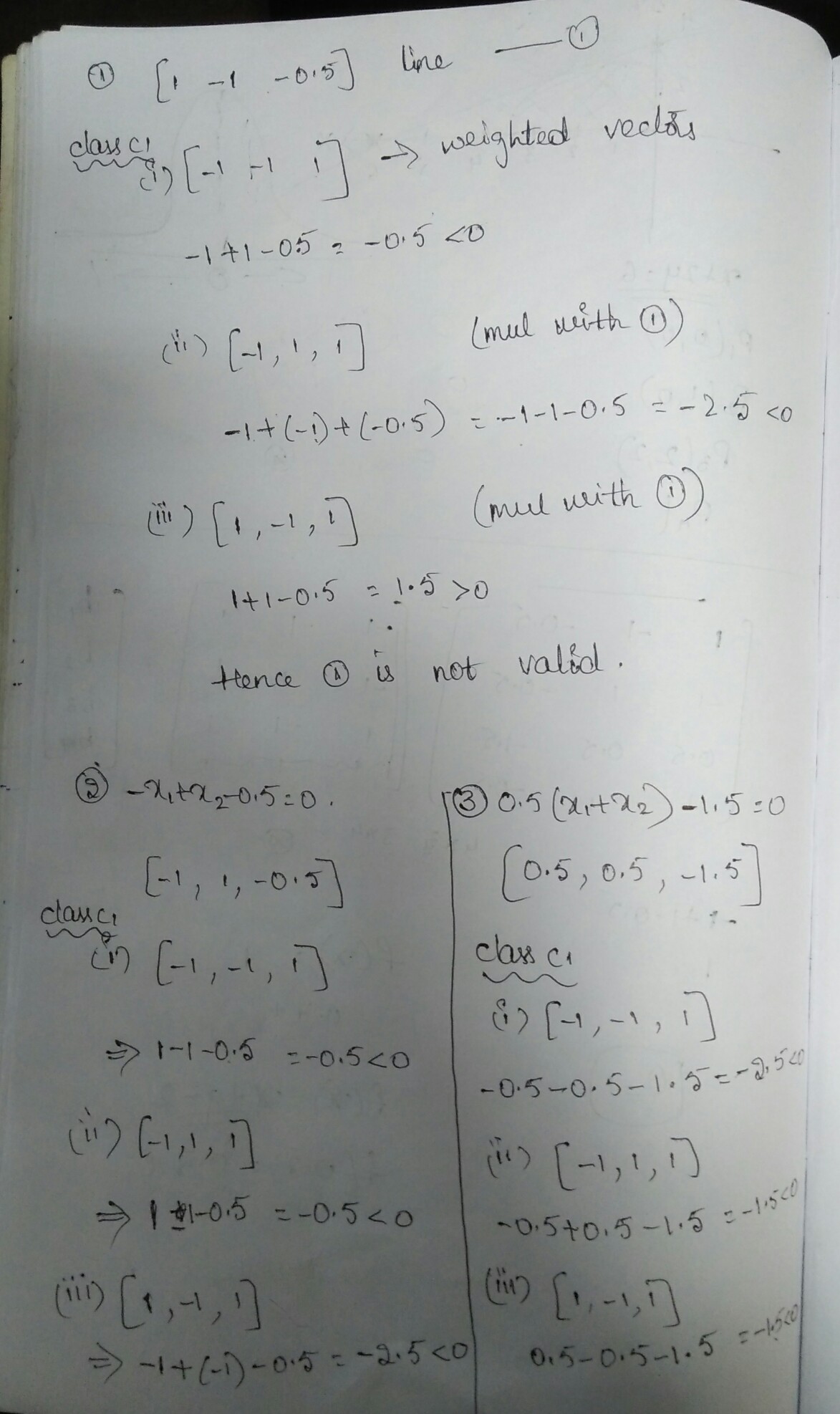
* Within neural networks, **deep learning** is generally used to describe particularly complex networks with many more layers than normal. The advantage of these added layers is that the networks are able to develop much greater levels of abstraction, which is necessary for certain complex tasks, like image recognition and automatic translation.

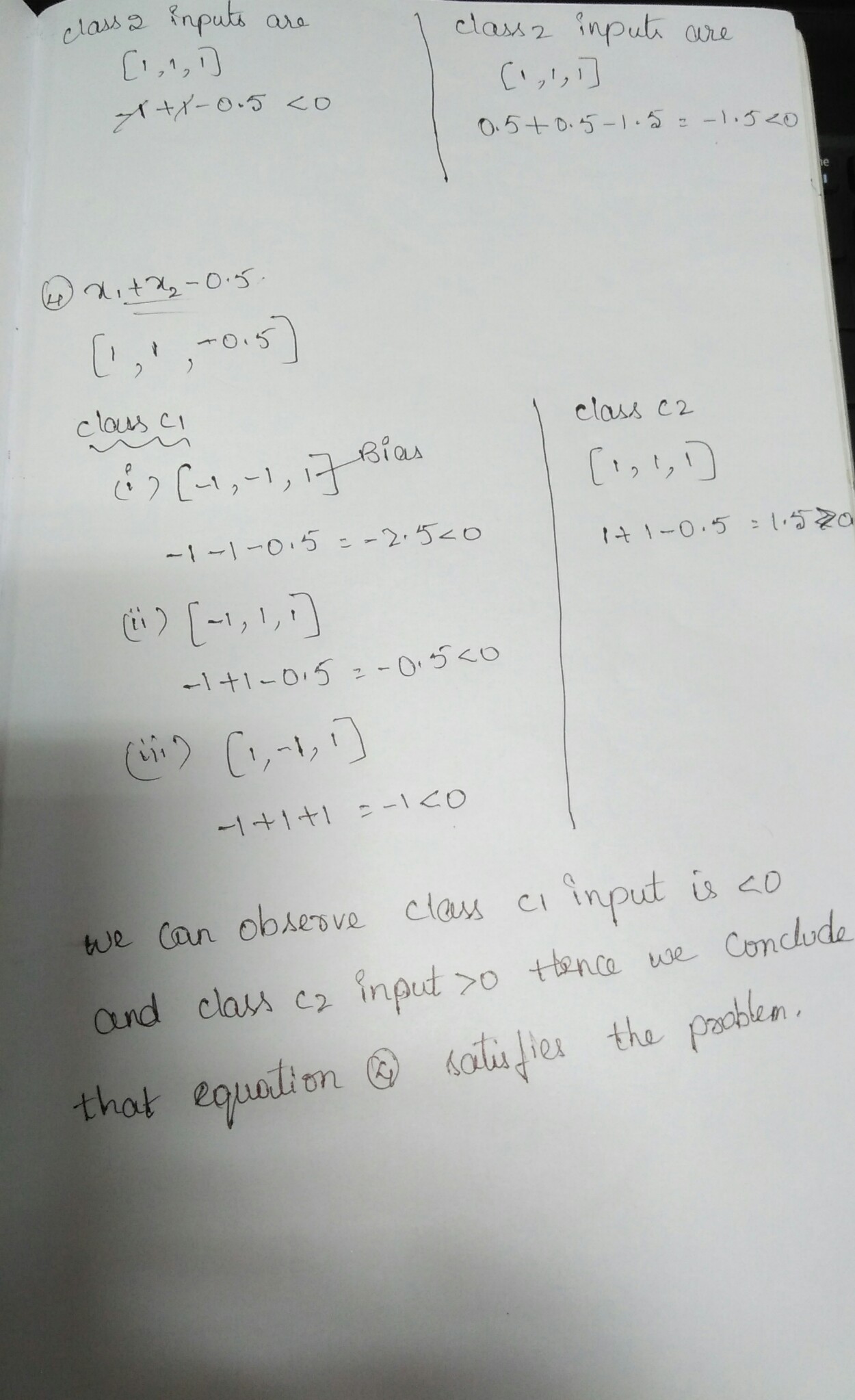












**Use Cases-**

* Market Price and cost modeling.
* Customer Segmentation.
* Insurance claim and fraud Detection.
* Elgibility to apply for credit card
* Chances of getting selected in Interview
* Chances of apply use cases without error.
* To mention the Population Growth.
* To check the Credit Risk in banks.
* Finding the transactions happened in ATM per day.
* Finding the changes in Pollution
* To check the existence of the customer in the bank.
* To check the market strategy.
* To find the price fluctuation of Amazon products.

Python-

import numpy as np

import matplotlib.pyplot as plt

import pandas as pd

# Importing the dataset

dataset = pd.read\_csv('C:\\Users\\Rama\\Desktop\\Datbase1.csv')

X = dataset.iloc[:, 3:13].values

y = dataset.iloc[:, 13].values

# Encoding categorical data

from sklearn.preprocessing import LabelEncoder, OneHotEncoder

labelencoder\_X\_1 = LabelEncoder()

X[:, 1] = labelencoder\_X\_1.fit\_transform(X[:, 1])

labelencoder\_X\_2 = LabelEncoder()

X[:, 2] = labelencoder\_X\_2.fit\_transform(X[:, 2])

onehotencoder = OneHotEncoder(categorical\_features = [1])

X = onehotencoder.fit\_transform(X).toarray()

X = X[:, 1:]

# Splitting the dataset into the Training set and Test set

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

X\_train, X\_test, y\_train, y\_test = train\_test\_split(X, y, test\_size = 0.2, random\_state = 0)

# Feature Scaling

from sklearn.preprocessing import StandardScaler

sc = StandardScaler()

X\_train = sc.fit\_transform(X\_train)

X\_test = sc.transform(X\_test)

# Part 2 - Now let's make the ANN!

# Importing the Keras libraries and packages

import keras

from keras.models import Sequential

from keras.layers import Dense

# Initialising the ANN

classifier = Sequential()

# Adding the input layer and the first hidden layer

classifier.add(Dense(units = 6, kernel\_initializer = 'uniform', activation = 'relu', input\_dim = 11))

# Adding the second hidden layer

classifier.add(Dense(units = 6, kernel\_initializer = 'uniform', activation = 'relu'))

# Adding the output layer

classifier.add(Dense(units = 1, kernel\_initializer = 'uniform', activation = 'sigmoid'))

# Compiling the ANN

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

# Fitting the ANN to the Training set

classifier.fit(X\_train, y\_train, batch\_size = 10, epochs = 100)

# Part 3 - Making predictions and evaluating the model

# Predicting the Test set results

y\_pred = classifier.predict(X\_test)

y\_pred = (y\_pred > 0.5)

# Making the Confusion Matrix

from sklearn.metrics import confusion\_matrix

cm = confusion\_matrix(y\_test, y\_pred)