Question:

Write a python program to create a single neuron model to simulate different transfer functions over the iris dataset.

Solution:

In [ ]:

```python
# Imporing Libries
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
```

Transfer Functions:

3 Different transfer function i.e. hardlimit, sigmoid and hyperbolic tan (tansig).

In [ ]:

```python
def hardlim(n):

    if n<0:
        return -1
    elif n >=0:
        return 1
    else:
        return 0

def sigmoid(n):
    return 1/(1+np.exp(-n))

def tansig(n):
    num = np.exp(n) - np.exp(-n)
    den = np.exp(n) + np.exp(-n)
    return num/den
```

Loading and pre processing of Data

In [ ]:

```python
df = pd.read_csv('iris.data')
df.head()
```

Out[ ]:

	5.1	3.5	1.4	0.2	Iris-setosa
0	4.9	3.0	1.4	0.2	Iris-setosa
1	4.7	3.2	1.3	0.2	Iris-setosa
2	4.6	3.1	1.5	0.2	Iris-setosa
3	5.0	3.6	1.4	0.2	Iris-setosa
4	5.4	3.9	1.7	0.4	Iris-setosa

In [ ]:

```python
X = np.array(df)[:,0:4]
X[:5]
```

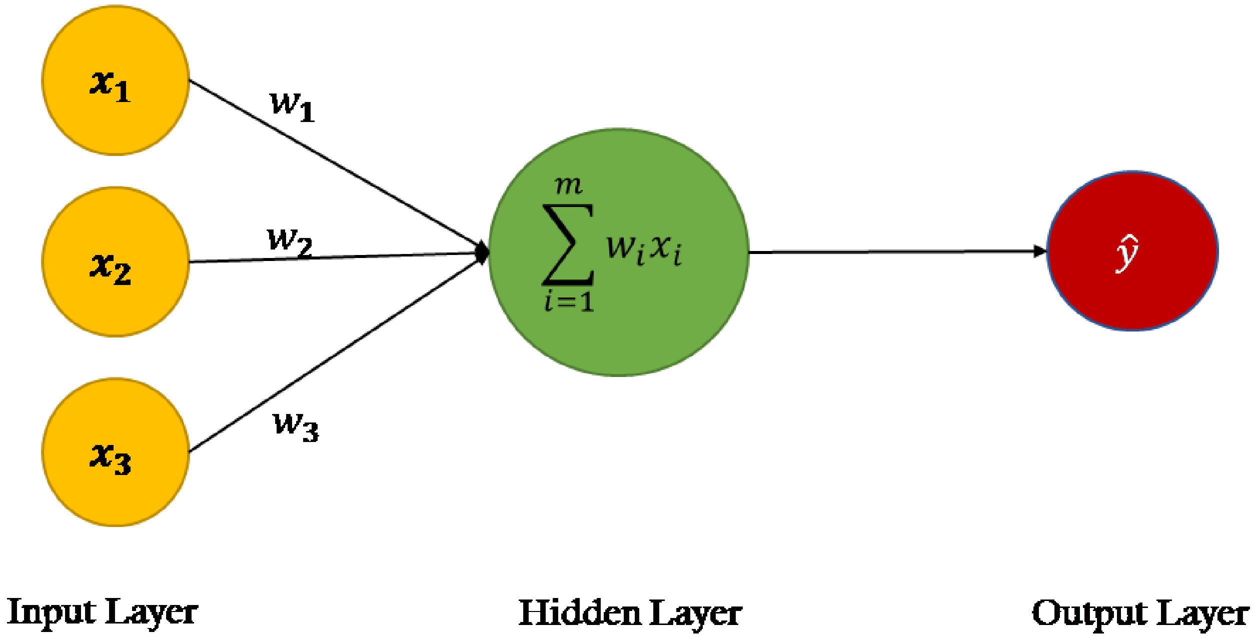
Out[ ]:

```
array([[4.9, 3.0, 1.4, 0.2],
       [4.7, 3.2, 1.3, 0.2],
       [4.6, 3.1, 1.5, 0.2],
       [5.0, 3.6, 1.4, 0.2],
       [5.4, 3.9, 1.7, 0.4]], dtype=object)
```

Single Nural Network:

Single Nural Network to draw histogram of the actual output of three different transfer function.

Single Nuron:

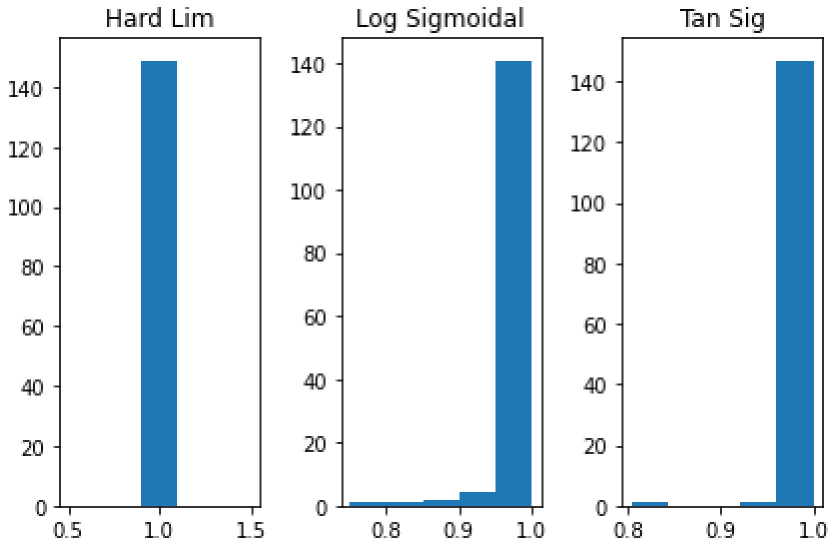


In [ ]:

```
def nuron_drawhist(x):
    ao1_list = []
    ao2_list = []
    ao3_list = []

    #Looping Over All the Patterns
    for i in x:

        # Randomly Generated weights
        w = np.random.rand(4)

        net = np.dot(i,w.T)

        #Actual Output for 3 different Transfer Functions
        ao1 = hardlim(net)
        ao2 = sigmoid(net)
        ao3 = tansig(net)

        ao1_list.append(ao1)
        ao2_list.append(ao2)
        ao3_list.append(ao3)

    # Plot of Three Histograms
    plt.subplot(1,3,1)
    plt.hist(ao1_list,bins= 5)
    plt.title('Hard Lim')

    plt.subplot(1,3,2)
    plt.hist(ao2_list, bins=5)
    plt.title('Log Sigmoidal')

    plt.subplot(1,3,3)
    plt.hist(ao3_list, bins= 5)
    plt.title('Tan Sig')

    plt.tight_layout()

    plt.show()
```

In [ ]:

```
nuron_drawhist(X)
```



Single Nural Network:

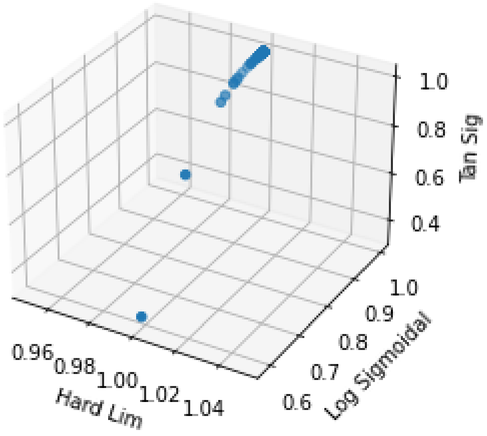
Single Nural Network to draw 3D scatter plot of the actual output of three different transfer function.

In [ ]:

```
def nuron_drawScatter(x):
    ao1_list = []
    ao2_list = []
    ao3_list = []

    #Looping Over All the Patterns
    for i in x:
        # Randomly Generated weights
        w = np.random.rand(4)

        net = np.dot(i,w.T)

        #Actual Output for 3 different Transfer Functions
        ao1 = hardlim(net)
        ao2 = sigmoid(net)
        ao3 = tansig(net)

        ao1_list.append(ao1)
        ao2_list.append(ao2)
        ao3_list.append(ao3)
    # 3D Scatter Plot of All the transfer functions
    ax = plt.axes(projection = '3d')
    ax.scatter3D(ao1_list,ao2_list, ao3_list)
    ax.set_xlabel('Hard Lim')
    ax.set_ylabel('Log Sigmoidal')
    ax.set_zlabel('Tan Sig')

    plt.show()
```

In [ ]:

```
nuron_drawScatter(X)
```


1.0
0.8
0.6
0.4
Tan Sig
1.0
0.9
0.8
0.7
0.6
Log Sigmoidal
0.96
0.98
1.00
1.02
1.04
Hard Lim