In [ ]:
```
input_size=4
hidden_layers=2
neurons_in_hidden_layer=5
output_size=3
learning_rate=0.1

model=ANN(input_size,hidden_layers,neurons_in_hidden_layer,output_size,learning_rate)


import pandas as pd
import numpy as np
import seaborn as sns

df=sns.load_dataset('iris')

x=df.drop(['spcies'],axis=1)
y=df['spcies']


df


import numpy as np
from sklearn.preprocessing import LabelEncoder
label_encoder = LabelEncoder()
encoded_labels = label_encoder.fit_transform(y)

from sklearn.model_selection import train_test_split
x_train,x_test,y_train,y_test=train_test_split(x,encoded_labels,test_size=0.2,random_state=62)

from tensorflow.keras.utils import to_categorical

labels = np.array(y_train)
y_train = to_categorical(labels)
y_train=np.array(y_train)


x_train = x_train.to_numpy().T
x_test = x_test.to_numpy()
```

```
print(x_train)
print(x_test)
print(y_train)


epochs=100000
model.train(x_train,y_train,epochs)


hh=model.forward([[6.7],[3.0],[5.2],[2.3]])
print(hh[-1])


test_sample=[]
for i in x_test:
    test_sample.append([[x] for x in i.tolist()])


y_pred = np.argmax(np.hstack(model.predict(test_sample)), axis=0)


print(y_pred)
print(y_test)


from sklearn.metrics import f1_score,precision_score,recall_score,accuracy_score
accuracy=accuracy_score(y_test,y_pred)
precision=precision_score(y_test,y_pred,average='weighted')
recall=recall_score(y_test,y_pred,average='weighted')
f1score=f1_score(y_test,y_pred,average='weighted')
print("Accuracy:", accuracy)
print("Precision:", precision)
print("Recall:", recall)
print("F1-score:", f1score)
```