Exercise 8

8.1 Problem Statement:

Design and implement a radial basis function neural network to solve function approximation or regression problem.

8.2 Description of Machine Learning Algorithm:

Radial basis function (RBF) networks are a commonly used type of artificial neural network for function approximation problems. Radial basis function networks are distinguished from other neural networks due to their universal approximation and faster learning speed. An RBF network is a type of feed forward neural network composed of three layers, namely the input layer, the hidden layer and the output layer. Each of these layers has different tasks.

The first layer corresponds to the inputs of the network, the second is a hidden layer consisting of a number of RBF non-linear activation units, and the last one corresponds to the final output of the network. Activation functions in RBFNs are conventionally implemented as Gaussian functions.

8.3 Description of Data Set:

Title of the data set: Bank marketing dataset

The data is related with direct marketing campaigns of a Portuguese banking institution. The marketing campaigns were based on phone calls. Often, more than one contact to the same client was required, in order to access if the product (bank term deposit) would be ('yes') or not ('no') subscribed.

8.4 Data Preprocessing and Exploratory Data Analysis:

Data preprocessing is the process of transforming raw data into an understandable format. It is also an important step in data mining as we cannot work with raw data. The quality of the data should be checked before applying machine learning or data mining algorithms.

Major Tasks in Data Preprocessing:

1. Data cleaning
2. Data integration
3. Data reduction
4. Data transformation

Exploratory data analysis (EDA) is a technique that data professionals can use to understand a dataset before they start to model it. Some people refer to EDA as data exploration. The goal of conducting EDA is to determine the characteristics of the dataset. Conducting EDA can help data analysts make predictions and assumptions about data. Often, EDA involves data visualization, including creating graphs like histograms, scatter plots and box plots.

Major Tasks in EDA:

1. Observe your dataset
2. Find any missing values
3. Categorize your values
4. Find the shape of your dataset
5. Identify relationships in your dataset
6. Locate any outliers in your dataset

8.5 Machine Learning Package Used for Model building:

For classification of model we use scikit-learn

Scikit-learn is an open source Machine Learning Python package that offers functionality supporting supervised and unsupervised learning. Additionally, it provides tools for model development, selection and evaluation as well as many other utilities including data pre-processing functionality.

More specifically, scikit-learn's main functionality includes classification, regression, clustering, dimensionality reduction, model selection and pre-processing. sThe library is very simple to use and most importantly efficient as it is built on **NumPy**, **SciPy** and **matplotlib.**

Neural networks are a machine learning method inspired by how the human brain works. They are particularly good at doing pattern recognition and classification tasks, often using images as inputs. They're a well established machine learning technique that has been around since the 1950s but have gone through several iterations since that have overcome fundamental limitations of the previous one. The current state of the art neural networks are often referred to as deep learning.

8.6 Implementation:

```
import math
import pandas as pd
from sklearn.preprocessing import LabelEncoder
from sklearn.metrics import accuracy_score
from sklearn.model_selection import train_test_split
from sklearn.cluster import KMeans
from sklearn.preprocessing import StandardScaler
import numpy as numpy
Data= pd.read_csv("bank-full.csv")
cols = ["age","balance","day","duration","campaign","pdays","previous"]
data_encode= Data.drop(cols, axis= 1)
data_encode= data_encode.apply(LabelEncoder().fit_transform)
data_rest= Data[cols]
Data= pd.concat([data_rest,data_encode], axis= 1)
data_train, data_test= train_test_split(Data, test_size= 0.33, random_state= 4)
X_train= data_train.drop("y", axis= 1)
Y_train= data_train["y"]
X_test= data_test.drop("y", axis=1)
Y_test= data_test["y"]
scaler= StandardScaler()
scaler.fit(X_train)
X_train= scaler.transform(X_train)
X_test= scaler.transform(X_test)
K_cent= 8
km= KMeans(n_clusters= K_cent, max_iter= 100)
km.fit(X_train)
cent= km.cluster_centers_
max=0
for i in range(K_cent):
    for j in range(K_cent):
        d = numpy.linalg.norm(cent[i]-cent[j])
        if(d> max):
            max= d
d= max

sigma= d/math.sqrt(2*K_cent)
shape= X_train.shape
row= shape[0]
column= K_cent
G= numpy.empty((row,column), dtype= float)
for i in range(row):
    for j in range(column):
        dist= numpy.linalg.norm(X_train[i]-cent[j])
        G[i][j]= math.exp(-math.pow(dist,2)/math.pow(2*sigma,2))
GTG= numpy.dot(G.T,G)
GTG_inv= numpy.linalg.inv(GTG)
fac= numpy.dot(GTG_inv,G.T)
W= numpy.dot(fac,Y_train)
row= X_test.shape[0]
column= K_cent
G_test= numpy.empty((row,column), dtype= float)
for i in range(row):
    for j in range(column):
        dist= numpy.linalg.norm(X_test[i]-cent[j])
        G_test[i][j]= math.exp(-math.pow(dist,2)/math.pow(2*sigma,2))
prediction= numpy.dot(G_test,W)
prediction= 0.5*(numpy.sign(prediction-0.5)+1)
score= accuracy_score(prediction,Y_test)
print ('The accuracy of the RBF Neural Network is: ' , "{0:0.2f}".format(score.
mean()*100), '%')
The accuracy of the RBF Neural Network is:  88.77 %
```

8.7 Results and Discussion:

Implementation of a radial basis function neural network to solve function approximation or regression problem has been successfully completed and the neural network is 88.77% accurate.