Ex No.10: *Applying Deep Learning Methods to Solve an application*

Problem Statement:

To develop a Deep Neural Network to predict cancer as malignant or benign.

Deep Learning

Deep learning is a subset of machine learning, which is essentially a neural network with three or more layers. These neural networks attempt to simulate the behaviour of the human brain—albeit far from matching its ability—allowing it to "learn" from large amounts of data. While a neural network with a single layer can still make approximate predictions, additional hidden layers can help to optimize and refine for accuracy.

Deep learning drives many artificial intelligence (AI) applications and services that improve automation, performing analytical and physical tasks without human intervention. Deep learning technology lies behind everyday products and services (such as digital assistants, voice-enabled TV remotes, and credit card fraud detection) as well as emerging technologies (such as self-driving cars).

Deep Neural Network to predict cancer as malignant or benign

Importing Dataset

In [ ]:

```
from sklearn.datasets import load_breast_cancer
dataset = load_breast_cancer()
```

In [ ]:

```
print(dataset.DESCR)
```

```
.. _breast_cancer_dataset:

Breast cancer wisconsin (diagnostic) dataset
--------------------------------------------

**Data Set Characteristics:**

    :Number of Instances: 569

    :Number of Attributes: 30 numeric, predictive attributes and the class

    :Attribute Information:
        - radius (mean of distances from center to points on the perimeter)
        - texture (standard deviation of gray-scale values)
        - perimeter
        - area
        - smoothness (local variation in radius lengths)
        - compactness (perimeter^2 / area - 1.0)
        - concavity (severity of concave portions of the contour)
        - concave points (number of concave portions of the contour)
        - symmetry
        - fractal dimension ("coastline approximation" - 1)

        The mean, standard error, and "worst" or largest (mean of the three
        largest values) of these features were computed for each image,
        resulting in 30 features.  For instance, field 3 is Mean Radius, field
        13 is Radius SE, field 23 is Worst Radius.

        - class:
                - WDBC-Malignant
                - WDBC-Benign

    :Summary Statistics:

    ===================================== ====== ======
                                           Min    Max
    ===================================== ====== ======
    radius (mean):                        6.981  28.11
    texture (mean):                       9.71   39.28
    perimeter (mean):                     43.79  188.5
    area (mean):                          143.5  2501.0
    smoothness (mean):                    0.053  0.163
    compactness (mean):                   0.019  0.345
    concavity (mean):                     0.0    0.427
    concave points (mean):                0.0    0.201
    symmetry (mean):                      0.106  0.304
    fractal dimension (mean):             0.05   0.097
    radius (standard error):              0.112  2.873
    texture (standard error):             0.36   4.885
    perimeter (standard error):           0.757  21.98
    area (standard error):                6.802  542.2
    smoothness (standard error):          0.002  0.031
    compactness (standard error):         0.002  0.135
    concavity (standard error):           0.0    0.396
    concave points (standard error):      0.0    0.053
    concave points (standard error):      0.0    0.053
    symmetry (standard error):            0.008  0.079
    fractal dimension (standard error):   0.001  0.03
    radius (worst):                       7.93   36.04
    texture (worst):                      12.02  49.54
    perimeter (worst):                    50.41  251.2
    area (worst):                         185.2  4254.0
    smoothness (worst):                   0.071  0.223
    compactness (worst):                  0.027  1.058
    concavity (worst):                    0.0    1.252
    concave points (worst):               0.0    0.291
    symmetry (worst):                     0.156  0.664
    fractal dimension (worst):            0.055  0.208
    ===================================== ====== ======

    :Missing Attribute Values: None

    :Class Distribution: 212 - Malignant, 357 - Benign

    :Creator:  Dr. William H. Wolberg, W. Nick Street, Olvi L. Mangasarian

    :Donor: Nick Street

    :Date: November, 1995

This is a copy of UCI ML Breast Cancer Wisconsin (Diagnostic) datasets.
https://goo.gl/U2Uwz2

Features are computed from a digitized image of a fine needle
aspirate (FNA) of a breast mass.  They describe
characteristics of the cell nuclei present in the image.

Separating plane described above was obtained using
Multisurface Method-Tree (MSM-T) [K. P. Bennett, "Decision Tree
Construction Via Linear Programming." Proceedings of the 4th
Midwest Artificial Intelligence and Cognitive Science Society,
pp. 97-101, 1992], a classification method which uses linear
programming to construct a decision tree.  Relevant features
were selected using an exhaustive search in the space of 1-4
features and 1-3 separating planes.

The actual linear program used to obtain the separating plane
in the 3-dimensional space is that described in:
[K. P. Bennett and O. L. Mangasarian: "Robust Linear
Programming Discrimination of Two Linearly Inseparable Sets",
Optimization Methods and Software 1, 1992, 23-34].

This database is also available through the UW CS ftp server:

ftp ftp.cs.wisc.edu
cd math-prog/cpo-dataset/machine-learn/WDBC/

.. topic:: References

   - W.N. Street, W.H. Wolberg and O.L. Mangasarian. Nuclear feature extraction
     for breast tumor diagnosis. IS&T/SPIE 1993 International Symposium on
     Electronic Imaging: Science and Technology, volume 1905, pages 861-870,
     San Jose, CA, 1993.
   - O.L. Mangasarian, W.N. Street and W.H. Wolberg. Breast cancer diagnosis and
     prognosis via linear programming. Operations Research, 43(4), pages 570-577,
     July-August 1995.
   - W.H. Wolberg, W.N. Street, and O.L. Mangasarian. Machine learning techniques
     to diagnose breast cancer from fine-needle aspirates. Cancer Letters 77 (1994)
     163-171.
```

In [ ]:

```
features = dataset.data
target = dataset.target
```

In [ ]:

```
print(features.shape)
print(features.shape)
```

```
(569, 30)
```

In [ ]:

```
print(target.shape)
```

```
(569,)
```

Splitting dataset into training set and test set

In [ ]:

```
from sklearn.model_selection import train_test_split
X_train, X_test, y_train, y_test = train_test_split(features, target, test_size = 0.2)
```

In [ ]:

```
print(X_train.shape)
```

```
(455, 30)
```

In [ ]:

```
print(y_train.shape)
```

```
(455,)
```

In [ ]:

```
print(X_test.shape)
```

```
(114, 30)
```

In [ ]:

```
print(y_test.shape)
```

```
(114,)
```

In [ ]:

```
from keras.models import Sequential
from keras.layers import Dense
```

In [ ]:

```
model = Sequential()
model.add(Dense(32, input_dim = 30, activation = 'relu')) ## hidden layer 1
model.add(Dense(64, activation = 'relu')) ## hidden layer 2
model.add(Dense(1, activation = 'sigmoid'))
```

In [ ]:

```
model.compile(loss = 'binary_crossentropy', optimizer = 'adam', metrics = ['accuracy'])
```

In [ ]:

```
model.summary()
```

```
Model: "sequential_8"
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
dense_24 (Dense)             (None, 32)                992
_________________________________________________________________
dense_25 (Dense)             (None, 64)                2112
_________________________________________________________________
dense_26 (Dense)             (None, 1)                 65
=================================================================
Total params: 3,169
Trainable params: 3,169
Non-trainable params: 0
_________________________________________________________________
```

In [ ]:

```
model.fit(X_train, y_train, epochs = 10)
```

```
Epoch 1/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0297 - accuracy: 0.9890
Epoch 2/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0180 - accuracy: 0.9934
Epoch 3/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0173 - accuracy: 0.9956
Epoch 4/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0263 - accuracy: 0.9890
Epoch 5/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0257 - accuracy: 0.9890
Epoch 6/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0161 - accuracy: 0.9978
Epoch 7/10
15/15 [==============================] - 0s 2ms/step - loss: 0.0172 - accuracy: 0.9934
Epoch 8/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0148 - accuracy: 0.9956
Epoch 9/10
15/15 [==============================] - 0s 1ms/step - loss: 0.0145 - accuracy: 0.9956
Epoch 10/10
15/15 [==============================] - 0s 2ms/step - loss: 0.0144 - accuracy: 0.9934
```

Out[ ]:

```
<keras.callbacks.History at 0x7fd5019442d0>
```

In [ ]:

```
scores = model.evaluate(X_test, y_test)
print(scores) ## returns loss and accuracy
```

```
4/4 [==============================] - 0s 2ms/step - loss: 0.4843 - accuracy: 0.9386
[0.48427480459213257, 0.9385964870452881]
```

In [ ]:

```
predictions = model.predict(X_test)
label = []
for pred in predictions:
  if pred>=0.5:
    print("Malignant")
  else:
    print("Benign")
```

```
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Benign
Malignant
Malignant
Benign
Benign
Malignant
Malignant
Malignant
Benign
Malignant
Malignant
Malignant
Malignant
Benign
Benign
```

```
Benign
Malignant
Benign
Benign
Malignant
Malignant
Benign
Benign
Malignant
Malignant
Benign
Malignant
Benign
Malignant
Benign
Malignant
Benign
Malignant
Benign
Benign
Malignant
Malignant
Malignant
Malignant
Malignant
Benign
Malignant
Benign
Benign
Benign
Benign
Malignant
Malignant
Benign
Malignant
Benign
Malignant
Benign
Malignant
Benign
Benign
Malignant
Malignant
Benign
Malignant
Malignant
Malignant
Malignant
Malignant
Benign
Benign
Malignant
Benign
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Malignant
Benign
Malignant
Malignant
Benign
Benign
Malignant
```

```
Malignant
Benign
Malignant
Malignant
Benign
Benign
Benign
Malignant
Benign
Malignant
Benign
Malignant
Malignant
Malignant
Benign
Malignant
Benign
Malignant
Benign
Benign
Benign
Malignant
```

RESULT

A Deep Neural Network was successfully developed to determine cancer as malignant or benign.

```
In [ ]:
```