practical_ANN

April 16, 2019

Gahan Saraiya (18MCEC10)

AIM: ANN on Iris dataset

```
In [29]: #Importing data and understanding your data
         import pandas as pd
         w = pd.read_csv('iris.data', header=None)
         w.head()
         %matplotlib notebook
```

```
In [17]: w.describe().transpose()
```

```
Out[17]:    count      mean       std  min  25%   50%  75%  max
         0  150.0  5.843333  0.828066  4.3  5.1  5.80  6.4  7.9
         1  150.0  3.054000  0.433594  2.0  2.8  3.00  3.3  4.4
         2  150.0  3.758667  1.764420  1.0  1.6  4.35  5.1  6.9
         3  150.0  1.198667  0.763161  0.1  0.3  1.30  1.8  2.5
```

```
In [25]: ##Saperating attributes and labels
         X = w.iloc[:, :4]
         y = w[4]

         ## dividing into train and test
         from sklearn.model_selection import train_test_split
         X_train, X_test, y_train, y_test = train_test_split(X, y)
```

```
Out[25]: 112
```

```
In [22]: ## Preprocessing the data
         from sklearn.preprocessing import StandardScaler
         scaler = StandardScaler()
         # Fit only to the training data
         scaler.fit(X_train)
         StandardScaler(copy=True, with_mean=True, with_std=True)
         # Now apply the transformations to the data:
         X_train = scaler.transform(X_train)
         X_test = scaler.transform(X_test)
```

```
In [48]: from sklearn.neural_network import MLPClassifier
         mlp = MLPClassifier(hidden_layer_sizes=(13, 13, 13), max_iter=5000)
         mlp.fit(X_train,y_train)
         predictions = mlp.predict(X_test)
         from sklearn.metrics import classification_report,confusion_matrix
         print(confusion_matrix(y_test,predictions))
         print(classification_report(y_test,predictions))
```

```
[[15  0  0]
 [ 0 11  2]
 [ 0  0 10]]
                 precision    recall  f1-score   support

    Iris-setosa       1.00      1.00      1.00        15
Iris-versicolor       1.00      0.85      0.92        13
 Iris-virginica       0.83      1.00      0.91        10

      micro avg       0.95      0.95      0.95        38
      macro avg       0.94      0.95      0.94        38
   weighted avg       0.96      0.95      0.95        38
```