Machine Learning Task with DaPy

We will finish a famous machine learning task called 'Wine' with DaPy as a quick start tutorial.
You could download the data set from http://archive.ics.uci.edu/ml/datasets/Wine or https://github.com/JacksonWuxs/DaPy.

Loading Data

```
%matplotlib inline
import DaPy as dp   # Loading DaPy
```

```
data = dp.DataSet('testDB/Wine.csv')   # Initialize a DaPy.DataSet object
```

```
data.readcol()   # load your data from file and transform it into DaPy.SeriesSet structure.
```

Preparing Data

```
data.info  # Basic information of data (number of miss value, number of records & variable names).
```

```
1.  Structure: DaPy.SeriesSet
2.   Set Name: MySeries
3. Dimensions: Ln=178 | Col=14
4. Miss Value: 0 elements
5.    Columns:        Title        |  Miss Value  |  Column Type  |  Value Type
               --------------------+--------------+---------------+-------------
                       class       |      0       |     <list>    |    <int>
                      Alcohol      |      0       |     <list>    |   <float>
                     Malic acid    |      0       |     <list>    |   <float>
                        Ash        |      0       |     <list>    |   <float>
                 Alcalinity of ash |      0       |     <list>    |   <float>
                     Magnesium     |      0       |     <list>    |    <int>
                   Total phenols   |      0       |     <list>    |   <float>
                     Flavanoids    |      0       |     <list>    |   <float>
               Nonflavanoid phenols|      0       |     <list>    |   <float>
                  Proanthocyanins  |      0       |     <list>    |   <float>
                  Color intensity  |      0       |     <list>    |   <float>
                        Hue        |      0       |     <list>    |   <float>
                       OD280       |      0       |     <list>    |   <float>
                      Proline      |      0       |     <list>    |    <int>
               --------------------+--------------+---------------+-------------
```

```
data  # 'Watch' your data ('class' is our target)
```

```
                class: <1, 1, 1, 1, 1, ... ,3, 3, 3, 3, 3>
              Alcohol: <14.23, 13.2, 13.16, 14.37, 13.24, ... ,13.71, 13.4, 13.27, 13.17, 14.13>
           Malic acid: <1.71, 1.78, 2.36, 1.95, 2.59, ... ,5.65, 3.91, 4.28, 2.59, 4.1>
                  Ash: <2.43, 2.14, 2.67, 2.5, 2.87, ... ,2.45, 2.48, 2.26, 2.37, 2.74>
    Alcalinity of ash: <15.6, 11.2, 18.6, 16.8, 21.0, ... ,20.5, 23.0, 20.0, 20.0, 24.5>
            Magnesium: <127, 100, 101, 113, 118, ... ,95, 102, 120, 120, 96>
        Total phenols: <2.8, 2.65, 2.8, 3.85, 2.8, ... ,1.68, 1.8, 1.59, 1.65, 2.05>
           Flavanoids: <3.06, 2.76, 3.24, 3.49, 2.69, ... ,0.61, 0.75, 0.69, 0.68, 0.76>
 Nonflavanoid phenols: <0.28, 0.26, 0.3, 0.24, 0.39, ... ,0.52, 0.43, 0.43, 0.53, 0.56>
      Proanthocyanins: <2.29, 1.28, 2.81, 2.18, 1.82, ... ,1.06, 1.41, 1.35, 1.46, 1.35>
      Color intensity: <5.64, 4.38, 5.68, 7.8, 4.32, ... ,7.7, 7.3, 10.2, 9.3, 9.2>
                  Hue: <1.04, 1.05, 1.03, 0.86, 1.04, ... ,0.64, 0.7, 0.59, 0.6, 0.61>
                OD280: <3.92, 3.4, 3.17, 3.45, 2.93, ... ,1.74, 1.56, 1.56, 1.62, 1.6>
              Proline: <1065, 1050, 1185, 1480, 735, ... ,740, 750, 835, 840, 560>
-------------------------------------
SeriesSet{178 Records & 14 Variables}
```

```
data.shuffles()  # Randomly scrambled records
```

```
data.count_element(0)   # Statistic the distribution of column 0 ('class' variable).
```

```
{'class': Counter({1: 59, 2: 71, 3: 48})}
```

Since the output of the multi-layer perceptron must fall between 0 and 1, we need to convert the three categories of the target variable into three-dimensional space.

```
# Initialize a DaPy.Matrix object
data_target = dp.Matrix(columns=['class_1', 'class_2', 'class_3'])
# Create an empty matrix filled with 0.
data_target.make(178, 3, 0)
```

```
CLASS = data.pop_col('class')[1]   # Pick out the 'Target' column.
```

```
# Transform the match dimension from 0 to 1.
for i, line in enumerate(data_target):
    line[CLASS[i]-1] = 1
```

```
data.normalized(process='STANDARD')   # Standarlized each variable in your dataset.
```

```python
# We select the first 142 records of the data as the training set (79.7%).
train_variable, train_target = dp.Matrix(data[:142]), dp.Matrix(data_target[:142])
# The remaining data as a test set (20.2%).
test_variable = dp.Matrix(data[142:])
```

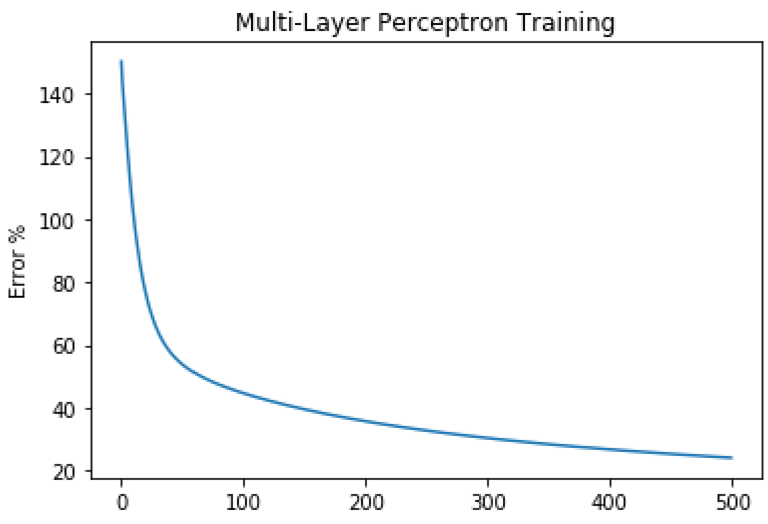
Multilayer Perceptron

```python
mlp = dp.MLP(train_variable, train_target)   # Initialized the DaPy.MLP object.
mlp.create()   # Create a new Neural Network.
```

```
"Create with 15 hidden Layer's cells"
```

```python
mlp.train(train_time=500)   # Training our MLP module for 500 periods.
```

```
MyMLP Start Training...
Initial Error: 150.23 %
Completed: 9.98     Remaining Time: 0.17 s
Completed: 19.96    Remaining Time: 0.12 s
Completed: 29.94    Remaining Time: 0.10 s
Completed: 39.92    Remaining Time: 0.08 s
Completed: 49.90    Remaining Time: 0.06 s
Completed: 59.88    Remaining Time: 0.05 s
Completed: 69.86    Remaining Time: 0.04 s
Completed: 79.84    Remaining Time: 0.03 s
Completed: 89.82    Remaining Time: 0.01 s
Completed: 99.80    Remaining Time: 0.00 s
Total Spent: 0.1 s  Errors: 24.013481 %
```



```
mlp.test(test_variable, data_target[142:])  # Test our MLP module.
```

```
'Classification Correct: 97.2222%'
```

```
mlp.topkl('MyMLP.pkl')   # 将训练好的神经网络保存至文件方便下次使用
```

Conclusion

Overall, the dataset of 'Wine' has 178 records and 14 variables. Because the data arrangement by 'class' variable as a ascending order, we shuffled our data with function 'shuffles()'. Due to the significant different magnitude between each variable, we standarlized our data with function 'normalized()'. In addition, We create and train a Multilayer Perceptron module. The test result of MLP module is amazing with the correct percentage more than 97.2%. Finally, we save our module as a file in order to use this module next time easily.