OASIS ML Group TRAINING 03

◆ Logistic Regression simple practice – Multi-Classification

In this practice, you are asked to use the **whole** "**Iris Data Set**" and separate data into three varieties. Follow hints below to finish it.

▫ Practice：

■ Analysis datasets：

Isis datasets contains three varieties of Iris：「Setosa」、「Versocolour」and 「Virginica」. Each variety has 50 data, which means there're total 150 data in the datasets. Each data contains 4 features：「Sepal length」、「Sepal width」、「Petal length」and「Petal width」. You are asked to use **all of them** in this practice.

■ Concept：

Build multiple classifier which can output 0~1, and choose the class corresponded with classifier which has highest value as prediction.

■ One-hot encoding：

- Discuss what's one-hot encoding and why we need it?
- What's difference between one-hot encoding and label encoding?

Both **Scikit-learn** and **Pandas** can help us finish doing one-hot encoding, in this practice, **you shall use Scikit-learn in this practice**.

■ Softmax function：

Use Softmax func. Instead of Sigmoid func.

- What's Softmax function?.
- Discuss why using Softmax function?

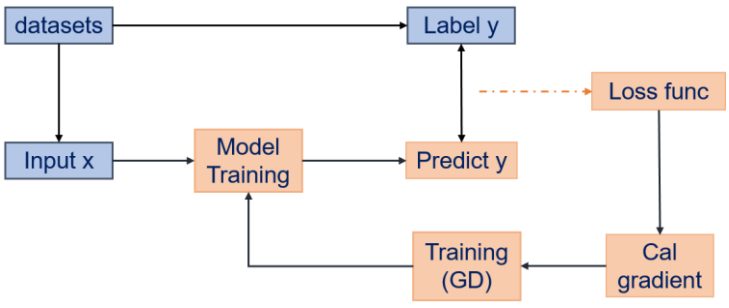
$$\sigma(\vec{z})_i = \frac{e^{z_i}}{\sum_{j=1}^{K} e^{z_j}}$$

■ Loss function：

- Using Cross entropy function as loss function.

$$\mathbf{H} = \frac{-1}{M}\sum_{m=0}^{M-1}(\mathbf{label\_y}_m \ \ \mathbf{log}\left(\mathbf{pred\_y}_m\right))$$

■ Training flow structure：



Copyright @ NCTU OASIS Lab, Shao-Wen, Cheng 2021

■ Recommend initial parameter：

Epoch：10000,　learning rate：0.01

- **Hint 01**：Import library you needed, but do not call API.
- **Hint 02**：Get the datasets from sklearn.

```
from sklearn.datasets import load_iris
iris = load_iris()
```

- **Hint 03**：In this time, you need to **choose 3 varieties and 4 features**.

```
Orginal data shape：(150, 4), Label data shape：(150,)
```

- **Hint 04**：**Convert label by using one-hot encoding.**

Reference：[sklearn.preprocessing.OneHotEncoder]

```
# do ONE-HOT ENCODING
from sklearn.preprocessing import OneHotEncoder

'''
    Code here
'''
print('Shape of orginal data：{}'.format(y_org.shape))
print('shape after One Hot-encoding：{}'.format(y_all_one.shape))
```

```
Shape of orginal data：(150,)
shape after One Hot-encoding：(150, 3)
```

- **Hint 05**：Separate datasets into training dataset and testing datasets by using train_test_split from sklearn. In this time, split data into 50% training data and 50% testing data
- **Hint 06**：Define function you need, for example：softmax, cross entropy……

```
def softmax(x):
    ### Code here ###
    return f_x

def predict_(x, w):
    return ### ?????? ###

def cross_entropy(yt, yp):
    return ### ?????? ###
```

- **Hint 07**：Calculating the accuracy score by using accuracy_score from sklearn.

```
from sklearn.metrics import accuracy_score
```

- **Hint 08**：After training, show the result.

```
First epoch result: Loss:1.091583, Accur：0.266667
Final epoch result: Loss:0.137235, Accur：0.960000
```

- **Hint 09**：Plot the figure of **"iteration vs. loss**" and "**iteration vs. accuracy**".

Copyright @ NCTU OASIS Lab, Shao-Wen, Cheng 2021

■ Bonus Problem：

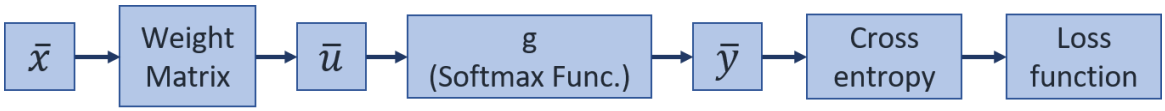


Figure shows the relationship of weight matrix, softmax and loss function.

➢ **Please dervie：**

When $err = pred\_y - label\_y$,

$$\frac{\partial L}{\partial W_{ij}} = x_j \cdot err_i$$

or

$$\frac{\partial L}{\partial W_{ij}} = \frac{1}{M}\sum_{m=0}^{M-1} x_{j(m)} \cdot err_{i(m)}$$

Copyright @ NCTU OASIS Lab, Shao-Wen, Cheng 2021