

## Python Machine Learning Notes

1. Machine learning mainly has 3 categories: supervised, unsupervised and reinforcement learning;
2. Supervised learning refers to a set of samples where the desired output signals (labels) are already known;
3. Reinforcement learning is to develop an agent that improves its performance based on interactions with the environment;
4. Unsupervised learning is to explore the structure of our data to extract meaningful information without guidance of a known outcome variable or reward function;
5. Dimensionality reduction is used in unsupervised learning to remove noise from the data, which can also degrade the predictive performance of certain algorithms;
6. Sample is each instance in our epochs, and it has multi-dimensional features, the target output value is called label;
7. Machine learning programming should go through pre-processing stage: where we input and divide labels and raw data into training dataset & test dataset; then learning phase where we apply algorithm to build the optimal model; the next one is evaluation, using reserved test dataset to test our final model to estimate the performance of our output models; Eventual stage is prediction, where we use our model to handle real world data and predict the output labels;
8. CSV stands for Comma Separate Values, we can use pandas' `pd.read_csv('path', header = None)` to import our pre-processing data;
9. Single layer neuron: multiple signals arrive at the dendrites, are then integrated into the cell body, and, if the accumulated signal exceeds a certain threshold, an output signal is generated and will be passed on by the axon;
10. For basic single layer neuron, unit step function is its decision function, in machine learning, the negative threshold is usually called the bias unit;
11. Basic perceptron learning rule: compute the output value `net_output`, calculate the error and update the weights; Adaptive Linear Neuron (Adaline) add the cost function, using the gradient descent to find the minimum cost that the cost function can converge. If the learning rate is wrong, it may overshoot the cost, we need to choose proper learning rate for Adaline learning algorithm;
12. Standardisation can optimise the sample's normal distribution, which help the gradient descent learning to converge more quickly. Simple just subtract the sample mean from every training sample and divide it by its standard deviation; (page 42)
13. When the dataset can go beyond millions, stochastic gradient descent can help to converge quicker to the minimum cost but it doesn't reach the global minimum, stochastic gradient descent can also used for online learning, when a model is trained on the fly as new training data arrives from time to time;
14. 5 steps to choose a classification algorithm: 1) selecting features and collecting training samples; 2) Choosing a performance metric; 3) choosing a classifier and optimization algorithm; 4) Evaluating the performance of the model 5) Tuning the algorithm
15. SciKit Learn has Iris dataset, after import datasets from sklearn, using `datasets.load_iris()` can load the data into a programme; from `sklearn.model_selection` import `train_test_split` can help us split datasets into training and test sets, they will also be automatically shuffled; from

sklearn.preprocessing import StandardScaler, we can standardize our datasets; from sklearn.linear\_model import logisticRegression, we can use the built-in logisticRegression classifier directly;

16. Print string and array at one-line, in Python 2.7.15 version, we just need commas to separate them, for instance:

```
demo = [1, 2, 3, 4]
```

```
print 'Result is: ', demo [: 2]
```

**OUTPUT:** [1,2,3];

17. Perceptron & Adaline has the drawback: they never converge unless the samples need to be linearly separable, otherwise the weight update won't stop unless we set up maximum epochs; Logistic Regression is a classifier, not regression. Its cost function is a S-shape sigmoid function, its logit function takes as input values in the range 0 to 1 and transforms them to values over the entire real-number range. If we predict correctly to 0 or 1, the cost on the y-axis approaches 0 or 1, However, if the prediction is wrong, the cost goes towards infinity (main point we penalize wrong predictions with an increasingly larger cost);
18. Overfitting means a model performs well on training data but does not generalize well to unseen data (test data), if a model suffers from overfitting, it has a high variance, it's basically "too sensitive about randomness"; Similarly, underfitting means a model is not complex enough to capture the pattern in the training data well and therefore also suffers from low performance on unseen data (high bias), we need to fine tune a model to get a "good compromise";
19. Regularisation is a very useful method to handle collinearity (high correlation among features), filter out noise from data and eventually prevent overfitting. The concept behind regularisation is to introduce additional information (bias) to penalize extreme parameter (weight) values. The term C parameter in sklearn's Logistic Regression classifier is directly related to the regularisation parameter, which is its inverse. By regularisation, the weight coefficients shrink if we decrease the parameter C, that is, if we increase the regularisation strength;

20.