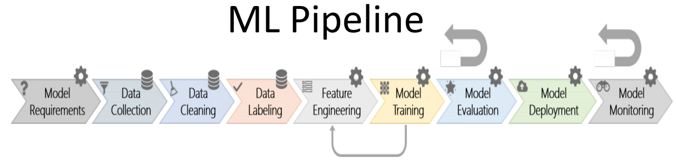
Week 3: ML Pipeline

ML Pipeline



- **Data Preparation and Feature Engineering**
  - Create new feature column with attribute combinations
  - Feature imputation
  - Process categorical feature. E.g .1-hot encoding
  - Feature transformation. E.g. Standardization
  - Create feature process pipeline
  - Prepare training set: Stratified Sampling v.s. Random Sampling
- **Model Selection/Evaluation**
  - Cross validation
  - Hyper-parameter Tuning

    Fine tune the model. E.g. GridSearchCV

What we have learnt so far

- **Data Preparation**

  feature scaling, stratified sampling

  Prepare training set and test set
- **Model Selection:**

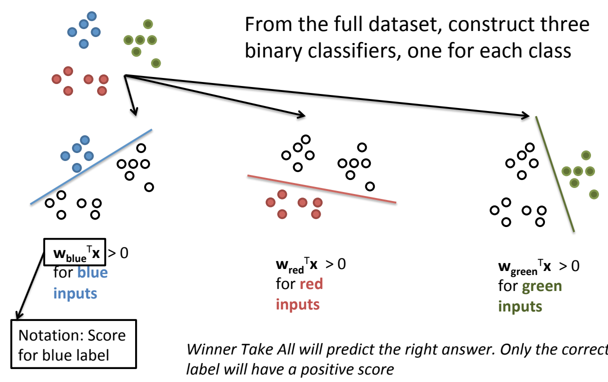
  Binary classification models: Perceptron and Adaline

  Both are linear models
- **Model Evaluation:**

  Accuracy score: how many the classifier got it right

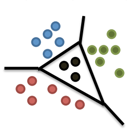
One-Vs-Rest: Multi-class classification

From the full dataset, construct three binary classifiers, one for each class



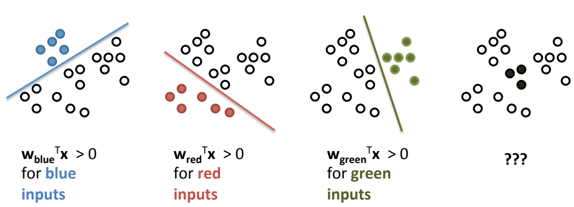
Winner Take All will predict the right answer. Only the correct label will have a positive score

One-Vs-Rest may not always work



Black points are not separable with a single binary classifier

The decomposition will not work for these cases!

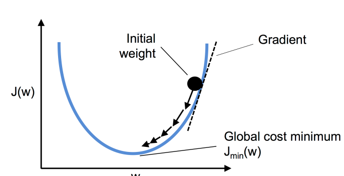


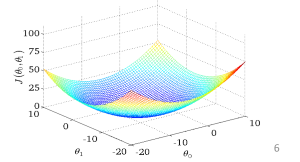
Gradient Descent Revisit

An optimization algorithm used for finding the weights (or coefficients) of machine learning algorithms.

Cost Function for Adaline: Sum of Squared Errors

$$J(\boldsymbol{w}) = \frac{1}{2}\sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right)\right)^2$$





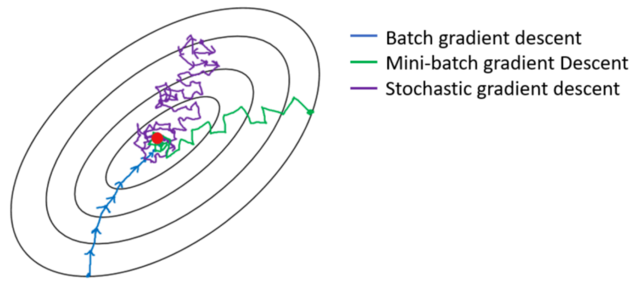
Stochastic Gradient Descent

$$\Delta \boldsymbol{w} = \eta \sum_i \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \boldsymbol{x}^{(i)}$$

$$\approx \eta \left(y^{(i)} - \phi\left(z^{(i)}\right) \right) \boldsymbol{x}^{(i)}$$

- Enables inexpensive evaluations of the sum-function and the sum gradient (an approximation)
- Adjust weights with each sample
- Samples are selected randomly & re-shuffle the training set for every epoch.

Mini-Batch Gradient Descent



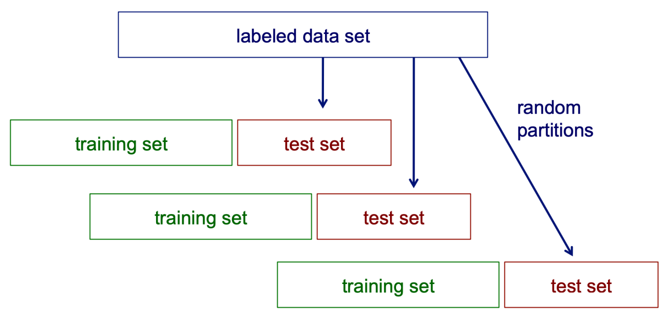
The **`batch size`** is a hyper-parameter of that controls the number of training samples in each iteration of weight update

Model Evaluation

Limitation of using a single training/test partition:

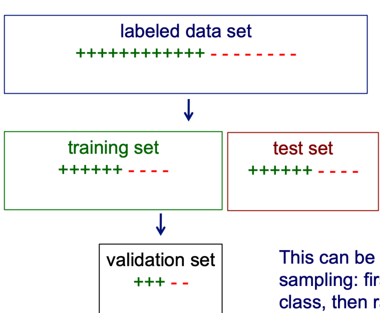
- May not have enough data
- A single training set doesn't tell us how sensitive accuracy is to a particular training sample.

Random Partitions



Stratified Sampling

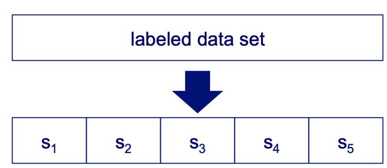
When randomly selecting training or validation sets, we may want to ensure that class proportions are maintained in each selected set



This can be done via stratified sampling: first stratify instances by class, then randomly select instances from each class proportionally.

Cross Validation

partition data into *n* subsamples



iteratively leave one subsample out for the test set, train on the rest

iteration	train on	test on
1	s_2 s_3 s_4 s_5	s_1
2	s_1 s_3 s_4 s_5	s_2
3	s_1 s_2 s_4 s_5	s_3
4	s_1 s_2 s_3 s_5	s_4
5	s_1 s_2 s_3 s_4	s_5

Performance Evaluation

- **Accuracy:** the measure as the percentage of predicted results that match the expected results.
- But…

Cross Validation - Example

Suppose we have 100 instances, and we want to estimate accuracy with cross validation

iteration	train on	test on	correct
1	s_2 s_3 s_4 s_5	s_1	11 / 20
2	s_1 s_3 s_4 s_5	s_2	17 / 20
3	s_1 s_2 s_4 s_5	s_3	16 / 20
4	s_1 s_2 s_3 s_5	s_4	13 / 20
5	s_1 s_2 s_3 s_4	s_5	16 / 20

accuracy = 73/100 = 73%

Confusion Matrix

		actual class: positive	actual class: negative
predicted class	positive	true positives (TP)	false positives (FP)
	negative	false negatives (FN)	true negatives (TN)

$$\text{accuracy} = \frac{TP + TN}{TP + FP + FN + TN}$$

Performance Evaluation

- **Precision**: the measure of the proportion of positive predictions was actually correct.

  precision = TP / (TP + FP)

- **Recall** (Sensitivity): the measure of the proportion of actual positives was identified correctly.

  recall = TP / (TP + FN)

To fully evaluate the effectiveness of a model, must examine both precision and recall, but these two measure are often in tension.

Type I & Type II Errors