Midterm Warp-up

Shan Wang

Lingnan College, Sun Yat-sen University

2020 Data Mining and Machine Learning LN3119

https://wangshan731.github.io/DM-ML/

Course outline

- Supervised learning
  - Linear regression
  - Logistic regression
  - SVM and kernel
  - Tree models
- Unsupervised learning
  - Clustering
  - PCA
  - EM
- Deep learning
  - Neural networks
  - Convolutional NN
  - Recurrent NN
- Reinforcement learning
  - MDP
  - ADP
  - Deep Q-Network

ML THREE elements

Model

Model

- Spaces
  - Input space (feature space) X, output space (labeled space) Y

- Training data
  - Sample S of size N drawn i.i.d. from $X \times Y$ according to distribution D:
  - $S = \{(\boldsymbol{x}_1, y_1), (\boldsymbol{x}_2, y_2), \dots, (\boldsymbol{x}_N, y_N)\}$

- Hypothesis set: $F \subseteq Y^X$ (mappings from X to Y)
  - Space of possible models, e.g. all linear functions
  - Depends on feature structure and prior knowledge about the problem

ML THREE elements

Strategy

Strategy

- Objective
  - Find a good hypothesis $f \in F$
- What is a good f
  - A f with small generalization error
- Loss function: L: $Y \times Y \rightarrow \mathbb{R}$
  - $L(\hat{y}, y)$: loss of predicting $\hat{y}$ when the true output is y
    - Binary classification: $L(\hat{y}, y) = 1_{\hat{y} \neq y}$
    - Regression: $L(\hat{y}, y) = \frac{1}{2}(\hat{y} - y)^2$
- Generalization error
  - $R(f) = \mathbb{E}_{(\boldsymbol{x},y)\sim D}[L(f(\boldsymbol{x}), y)]$
- Empirical error
  - $\hat{R}(f) = \frac{1}{N}\sum_{i=1}^{N} L(f(\boldsymbol{x}_i), y_i)$

Generalization error bound

- Finite hypothesis set F
- Generalization error bound
  - For any function $f \in F$, with probability no less than $1-\delta$, it satisfies

$$R(f) \leq \hat{R}(f) + \epsilon(d, N, \delta)$$

Where

$$\epsilon(d, N, \delta) = \sqrt{\frac{1}{2N}(\log d + \log \frac{1}{\delta})}$$

- N: number of training instances
- d: number of functions in F

Bonus question: How to prove it?

Hint: Hoeffding Inequality

Generalization error bound - Hint

Lemma: Hoeffding Inequality

Let $X_1, X_2, \ldots, X_n$ be bounded independent random variables $X_i \in [a, b]$, the average variable *Z* is

$$Z = \frac{1}{n} \sum_{i=1}^{n} X_i$$

Then the following inequalities satisfy:

$$P(Z - \mathbb{E}[Z] \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

$$P(\mathbb{E}[Z] - Z \geq t) \leq \exp\left(\frac{-2nt^2}{(b-a)^2}\right)$$

Maximum likelihood estimation

- Maximum likelihood estimation
  - We know $x_1, x_2, \dots, x_N \sim N(\mu, \sigma^2)$, how to know μ?
  - Set up likelihood equation: $P(\boldsymbol{x}|\mu, \sigma^2)$, and find μ to maximize it.
  - If $x_1, x_2, \dots, x_n$ are independent

$$\mathcal{L}(\boldsymbol{x}) = P(\boldsymbol{x}|\mu, \sigma^2) = \prod_{i=1}^{N} P(x_i|\mu, \sigma^2) = \prod_{i=1}^{N} \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(x_i-\mu)^2}{2\sigma^2}}$$

  - Taking the log likelihood (we get to do this since log is monotonic) and removing some constants:

$$\log(\mathcal{L}(\boldsymbol{x})) \propto \sum_{i=1}^{N} -(x_i - \mu)^2$$

  - FOC:

$$\mu = \frac{1}{N} \sum x_i$$

About MLE

- Maximum likelihood estimator:

$$\theta = \text{argmax}\, P(\boldsymbol{x}|\theta)$$

where

- $P(\boldsymbol{x}|\theta)$ is the joint **probability density function** of observations $\boldsymbol{x} = (x_1, x_2, \ldots, x_N)$
- **Frequentist**: only believe the data
- It is almost unbiased
- If we have enough data, it is great

Maximum A Posterior

- What if you have some ideas about your parameter?
- Bayes' Rule

$$P(\theta|\boldsymbol{x}) = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{P(\boldsymbol{x})} = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\theta, \boldsymbol{x})} = \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\boldsymbol{x}|\theta)P(\theta)}$$

- Maximum A Posterior

$$\theta = \operatorname{argmax} P(\theta|\boldsymbol{x}) = \operatorname{argmax} \frac{P(\boldsymbol{x}|\theta)P(\theta)}{\sum_{\Theta} P(\boldsymbol{x}|\theta)P(\theta)}$$

- Equivalent to maximize the numerator $P(\boldsymbol{x}|\theta)P(\theta)$
- Different from MLE:
  - Assume there is a **prior** distribution $P(\theta)$
  - We have some knowledge about the parameter

About MAP

- Example
  - There are two bags
    - Bag A: 50% Green balls+ 50% Red balls
    - Bag B: 100% Red balls
  - If you consecutively pick two red balls from one bag, which bag is most likely?
  - MLE
    - A: $P(\boldsymbol{x}|\theta) = 0.25$; B:$P(\boldsymbol{x}|\theta) = 1$; so B
  - MAP – we know get bag A with 0.9, get bag B with 0.1
    - A: $P(\boldsymbol{x}|\theta)P(\theta) = 0.25 * 0.9 = 0.225$
    - B: $P(\boldsymbol{x}|\theta)P(\theta) = 1 * 0.1 = 0.1$
    - So A
- If the prior is uniform distribution
  - We do not have knowledge about the parameter
  - MAP=MLE

ML THREE elements

Algorithm

Algorithm

- Objective
  - Find a good hypothesis $f \in F$ with small generalization error
- Solve $\min_{f} \hat{R}(f)$
  - An optimization problem
    - Analytical solution
    - Gradient method
    - Heuristics

Model evaluation

Confusion matrix

- Confusion Matrix
  - TP – True Positive ; FP – False Positive
  - FN – False Negative; TN – True Negative

	Predicted class		
Actual class		Class=yes	Class=no
	Class=yes	TP	FN
	Class=no	FP	TN

$$Accuracy = \frac{TP + TN}{TP + FN + FP + TN}$$

- Any limitation?

Other measures

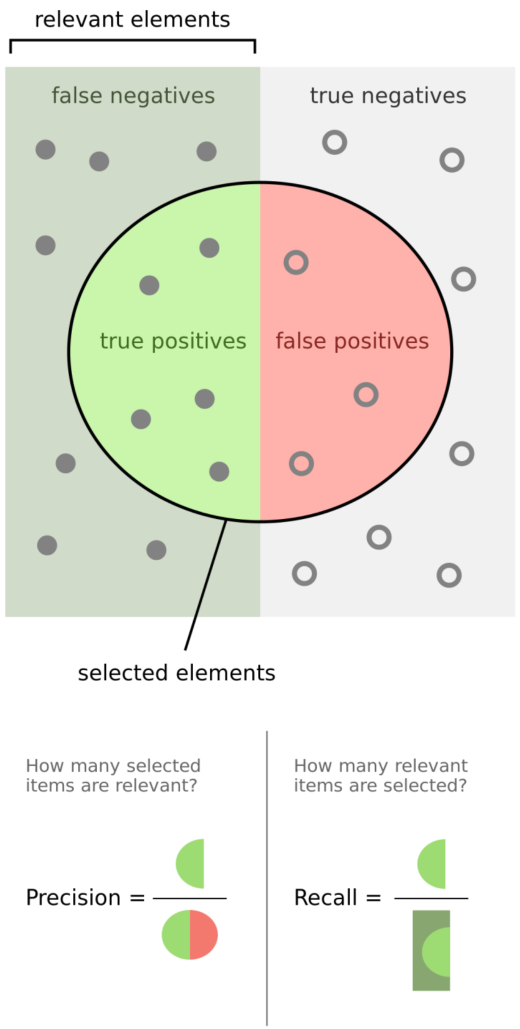
- Cost-sensitive measures

	Predicted class	
Actual class		Class=yes	Class=no
	Class=yes	TP	FN
	Class=no	FP	TN

$$Precision\ (p) = \frac{TP}{TP + FP}$$

$$Recall\ (r) = \frac{TP}{TP + FN}$$

$$F1 - measure\ (F) = \frac{2rp}{r + p} = \frac{2TP}{2TP + FP + FN}$$



- F1 measure is best if there is some sort of balance between precision (p) & recall (r)

R-squared

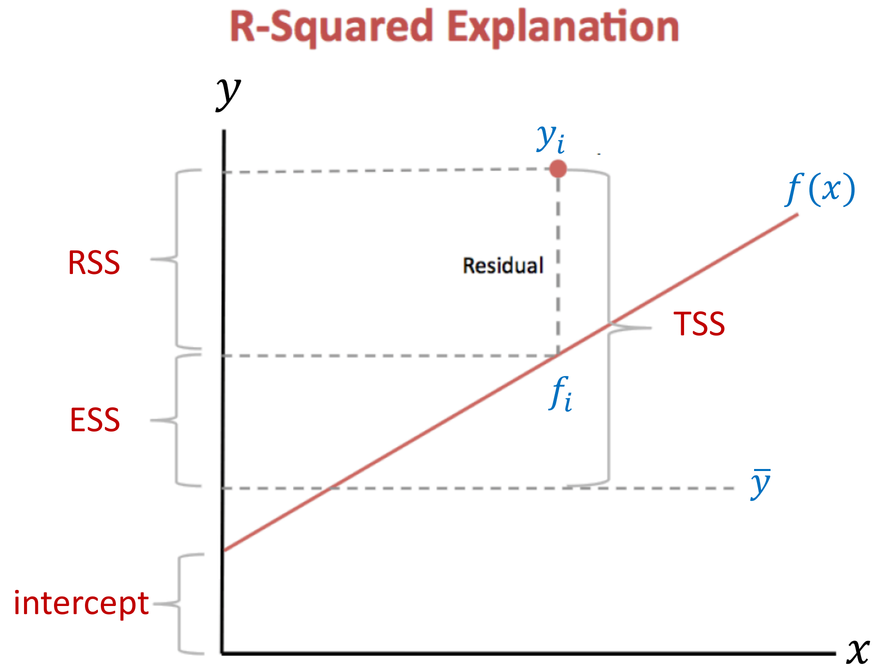
- Coefficient of determination (R^2)
  - measures how much of the residue can be explained by the regression line

Total Sum of Squares $TSS = \sum(y_i - \bar{y})^2$
(Total variance)

Explained Sum of Squares $ESS = \sum(f_i - \bar{y})^2$
(Explained variance)

Residual Sum of Squares $RSS = \sum(f_i - y_i)^2$
(Unexplained variance)

$$R^2 = \frac{explained\ variance}{total\ variance} = \frac{ESS}{TSS} = 1 - \frac{RSS}{TSS}$$



Model selection and regularization

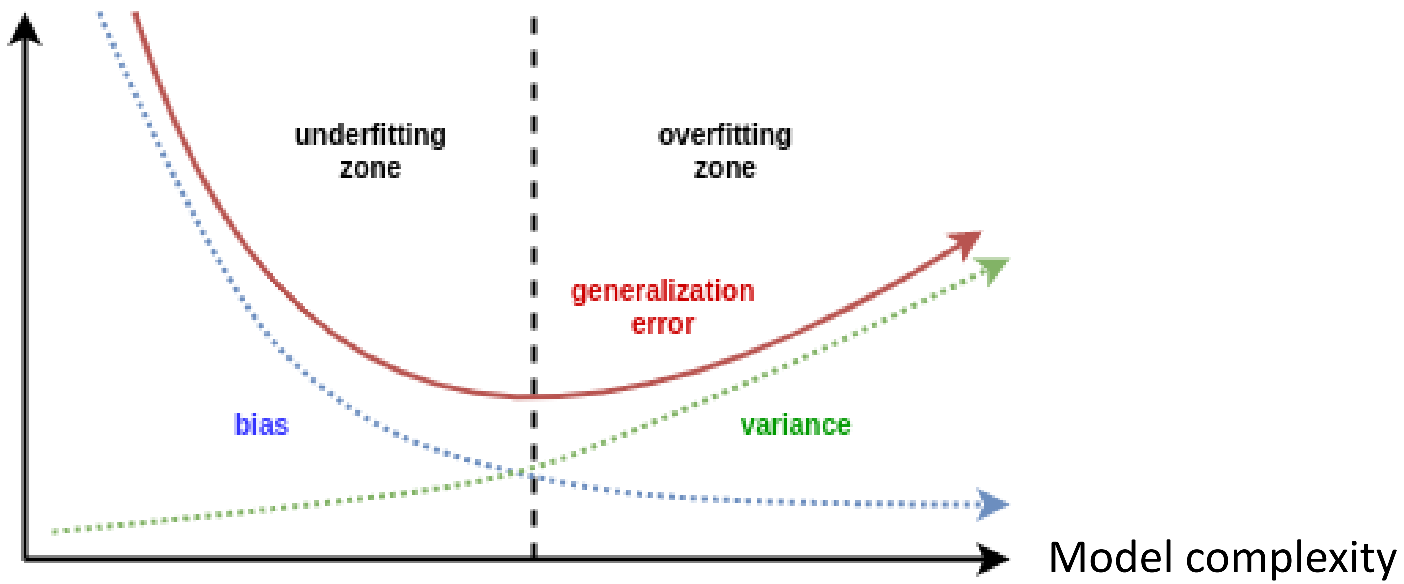
Bias vs. Variance

- Bias – variance decomposition

$$E[S] = E[(y - \hat{y})^2]$$

$$E[(y - \hat{y})^2] = (y - E[\hat{y}])^2 + E[(E[\hat{y}] - \hat{y})^2]$$
$$= [\text{Bias}]^2 + \text{Variance}.$$

- Bias – variance trade-off

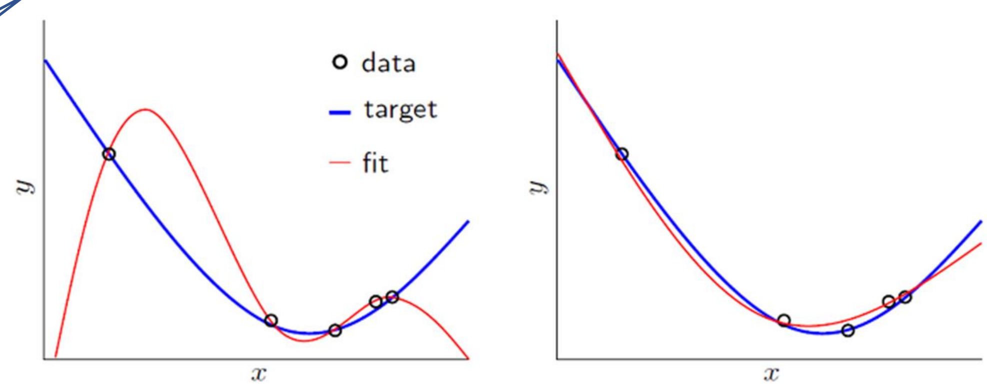


How regularization works

- Regularization
  - Add a penalty term of the parameters to prevent the model from overfitting the data
- Recall empirical risk minimization(ERM):
  - $f = \mathrm{argmin}_{h \in H}\, \hat{R}(f)$
  - It can be over-optimized (overfitting)
- With regularization
  - $f = \mathrm{argmin}_{f \in F}\, \hat{R}(f) + \lambda\Omega(f)$

Complexity of f

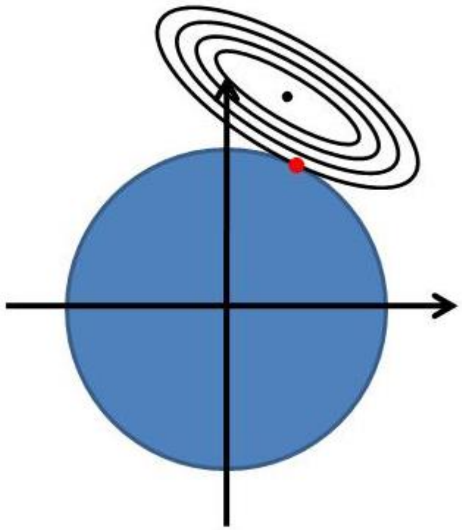
Regularization parameter



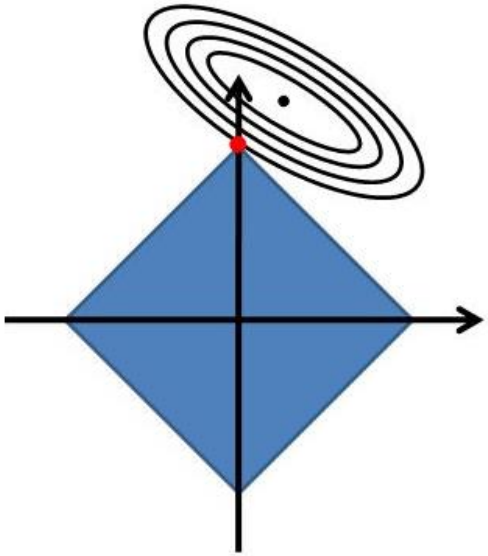
(a) without regularization   (b) with regularization

L1-norm and L2-norm regularization

- L2-norm (Ridge):
  - $\Omega(f = ax + b) = a^2 + b^2$



- L1-norm (Lasso):
  - $\Omega(f = ax + b) = |a| + |b|$



Cross validation

k-fold Cross Validation

- k-fold Cross Validation
  - Given the **training** set, split into k pieces (“folds”)
  - Use $(k-1)$ folds to estimate a model, and test model on remaining one fold (which acts as a validation set) for each candidate parameter value
  - Repeat for each of the k folds
  - For each candidate parameter value, average accuracy over the k folds, or validation sets
- For parameter tuning

- Regression Problem
  - Linear Regression

- Classification Problem
  - Logistic Regression
  - SVM

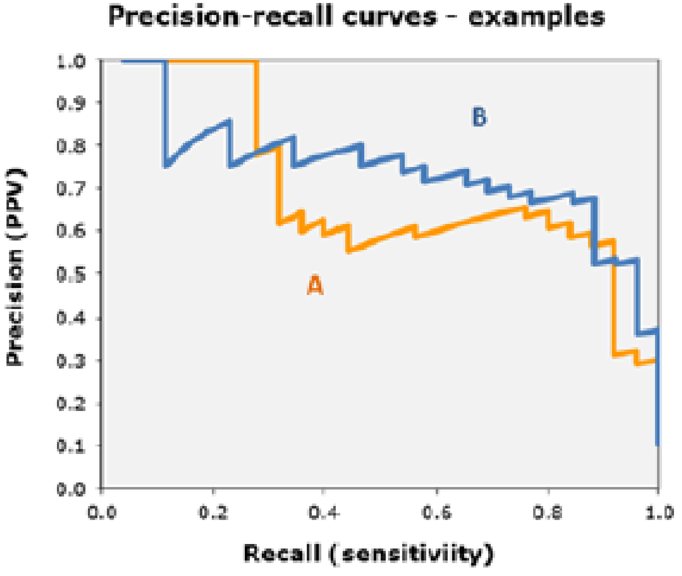
- Decision Tree
- Neural Network

Linear Regression

- $y = \boldsymbol{\theta}'\boldsymbol{x}$
- Minimize squared error ⇔ Maximize likelihood

Logistic Regression

- $p_\theta(y = 1|\boldsymbol{x}) = \sigma(\boldsymbol{\theta}'\boldsymbol{x}) = \frac{e^{\boldsymbol{\theta}'\boldsymbol{x}}}{1+e^{\boldsymbol{\theta}'\boldsymbol{x}}}$
- $p_\theta(y = 0|\boldsymbol{x}) = \frac{1}{1+e^{\boldsymbol{\theta}'\boldsymbol{x}}}$
- Minimize cross entropy
    - Cross entropy$= -\sum_{k=1}^{K} p_k \log q_k$
        - p_k: true label distribution
        - q_k: predicted label distribution
- Threshold and PR curve



SVM

- $y = f_{\boldsymbol{\theta}}(\boldsymbol{x}) = \begin{cases} +1, & \text{if } \boldsymbol{\theta}'\boldsymbol{x} + \theta_0 \geq 0 \\ -1, & \text{if } \boldsymbol{\theta}'\boldsymbol{x} + \theta_0 < 0 \end{cases}$
- Maximize margin
- Prim

$$\min_{\boldsymbol{\theta},\theta_0} \frac{1}{2}\|\boldsymbol{\theta}\|^2$$

$$s.t. \;\; y_i(\boldsymbol{\theta}'\boldsymbol{x_i} + \theta_0) \geq 1, i = 1, \dots, N$$

- Dual

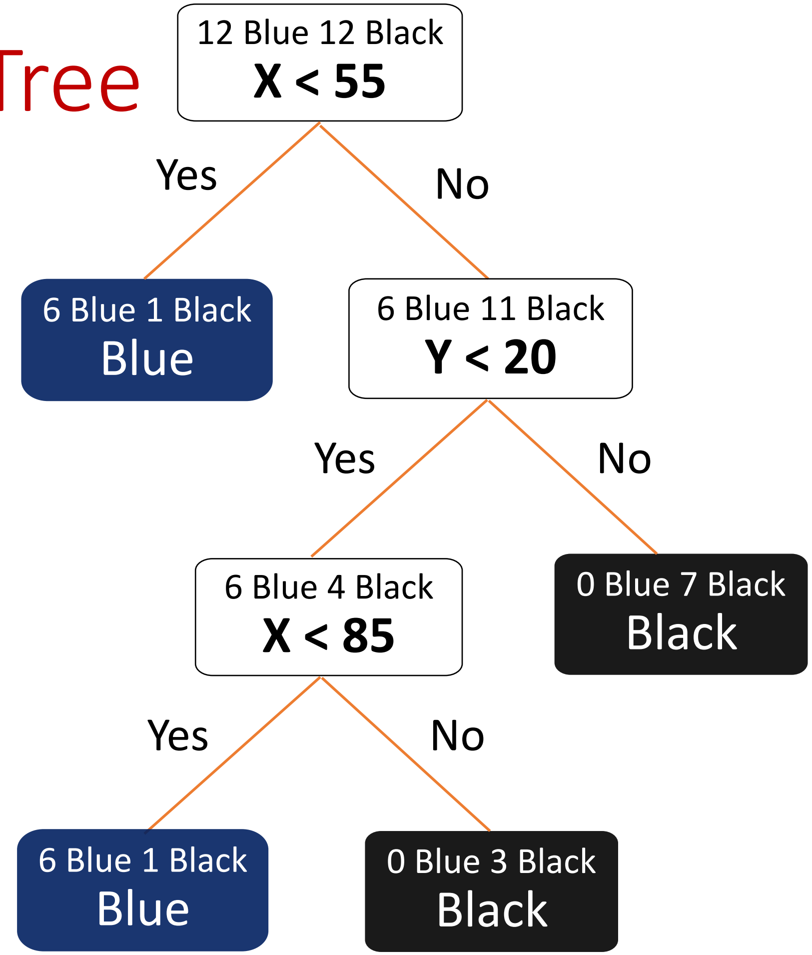
$$\max_{\boldsymbol{\alpha}\geq\mathbf{0}} \sum_{i=1}^{N} \alpha_i - \frac{1}{2}\sum_{i=1}^{N}\sum_{j=1}^{N} \alpha_i\alpha_j y_i y_j \boldsymbol{x_j}'\boldsymbol{x_i}$$

$$s.t. \sum_{i=1}^{N} \alpha_i y_i = 0$$

- Soft margin $C\sum_{i=1}^{N} \xi_i$
- Kernel $\phi(\boldsymbol{x})$
- SMO algorithm

Decision Tree

- Strategy & Algorithm
  - ID.3: information gain
  - C4.5: information gain ratio
  - CART
    - Classification: Gini index
    - Regression: Squared error
- Regularization: pruning
- Random forest: bagging
  - combining the results of multiple parallel models
  - Bootstrapping the data sets



Information gain: with the X, how much the uncertainty of Y decreases

Information gain ration: Information gain divided by uncertainty of X

Neural Network

Update the parameters:

Error of k

$$w_{k,j}^{(2)\prime} = w_{k,j}^{(2)} - \eta \Delta w_{k,j}^{(2)} = w_{k,j}^{(2)} - \boxed{\eta}\boxed{\varepsilon_k}\boxed{h_j^{(1)}}$$

Output of j

Learning rate

$$w_{j,m}^{(1)\prime} = w_{j,m}^{(1)} - \eta \Delta w_{j,m}^{(1)} = w_{j,m}^{(1)} - \boxed{\eta}\boxed{\varepsilon_j}\boxed{x_m}$$

Output of m

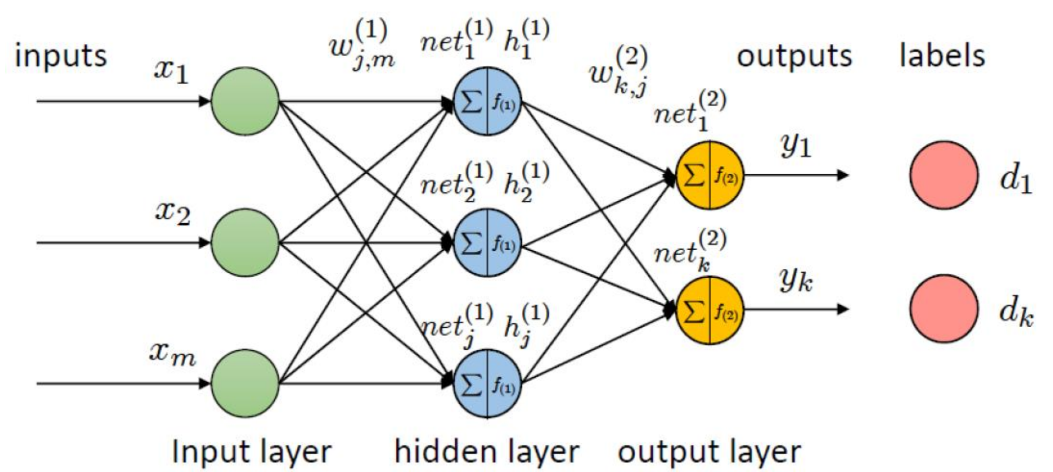
Error of j

If node i is in output layer

$$\varepsilon_i = (o_i - y_i) f'^{i}$$

If node i is in hidden layer

$$\varepsilon_i = \sum_{k=1}^{K} \varepsilon_k \, w_{k,i} f'^{i}$$



Two-layer feedforward neural network

Questions?

Shan Wang (王杉)

https://wangshan731.github.io/