

Propaedeutics and Overview of Deep Learning

www.huawei.com



Objectives

- ◆ After completing this course, you will be able to:
 - Understand learning algorithms and common machine learning algorithms.
 - Understand the concepts of the hyperparameter and validation set.
 - Master maximum likelihood estimation and Bayes estimation.
 - Understand the definition and development of neural networks.
 - Master the types of neural networks.



Contents

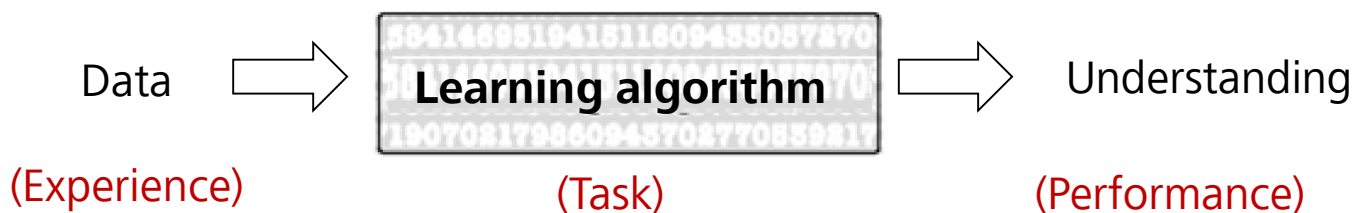
1. Propaedeutics of Deep Learning

- **Learning algorithms**
- Common Machine Learning Algorithms
- Hyperparameter and Validation Set
- Parameter Estimation
- Maximum Likelihood Estimation
- Bayes Estimation

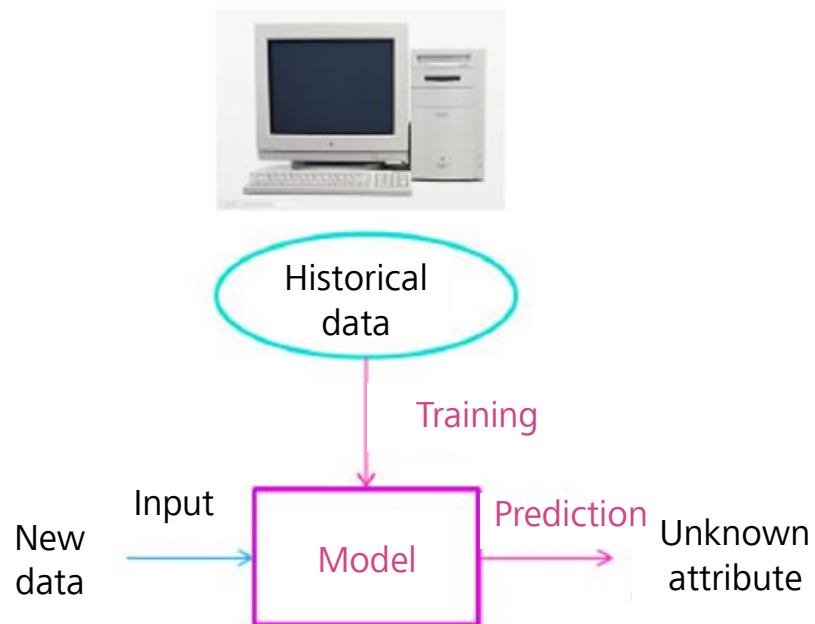
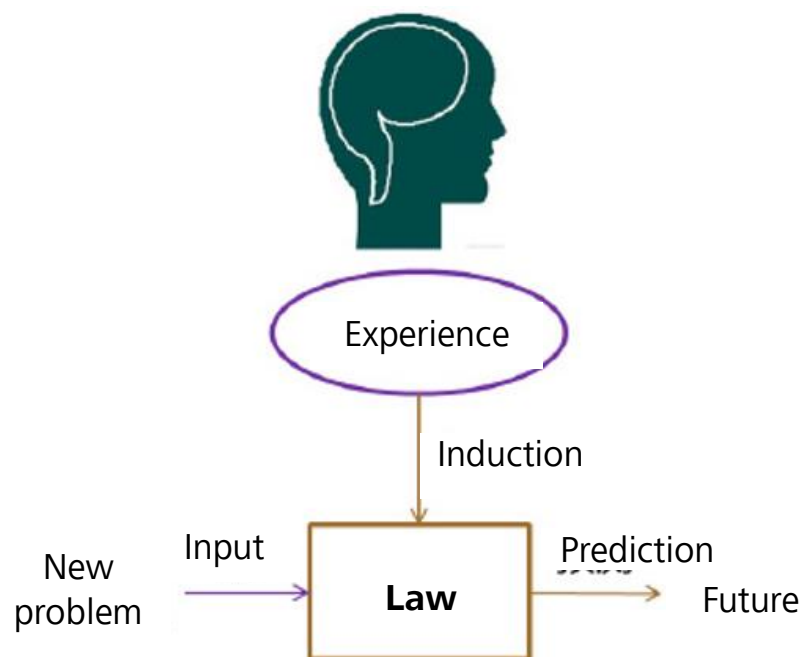
2. Overview of Deep Learning

Learning Algorithms (1)

- Machine learning (including deep learning) is a study of learning algorithms. A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P if its performance at tasks in T , as measured by P , improves with experience E .



Learning Algorithms (2)



Basic Terms and Concepts (1)

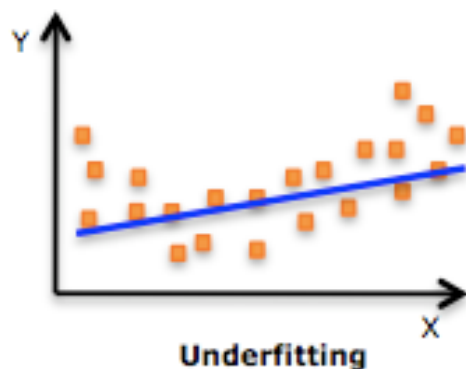
- ◆ **Dataset:** refers to a set of data used in machine learning tasks. Each piece of data is called a sample. The event or attribute that reflects the performance or nature of a sample in a certain aspect is called a **feature**.
- ◆ **Training set:** refers to a dataset used in the training process, where each sample is referred to as a training sample. The process of learning a model from data is called **learning (training)**.
- ◆ **Test set:** Test refers to the process of using the learnt model for prediction. The dataset used is called a test set, and each sample is called a test sample.

Basic Terms and Concepts (2)

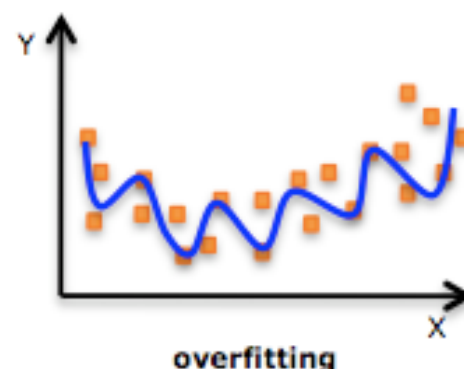
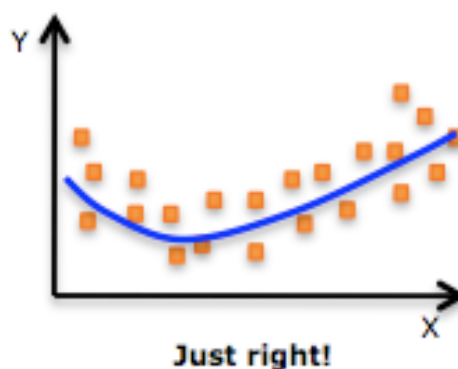
- ◆ **Generalization capability:** The goal of machine learning is that the learnt model should perform well on new samples, not just those on which the model has been trained. The ability to perform well on new samples is called generalization capability.
- ◆ **Error:** refers to the difference between the sample result predicted by the learnt model and the actual sample result.
 - Training error: error of the model on the training set
 - Generalization error: error on the new sample. Obviously, we prefer a model with a smaller generalization error.
- ◆ **Underfitting:** occurs when the training error is too large.
- ◆ **Overfitting:** occurs when the training error of the learnt model is small but the generalization error is large (weak generalization capability).

Basic Terms and Concepts (3)

- ◆ **Capacity of a model:** refers to the ability to fit a wide variety of functions. Machine learning algorithms will generally perform best when their capacity is appropriate for the true complexity of the task they need to perform and the amount of training data they are provided with. Models with an insufficient capacity are unable to solve complex tasks. Models with a high capacity can solve complex tasks, but when their capacity is higher than that is needed to solve the present task, they may overfit.



Failing to learn features



Learning noise

Basic Terms and Concepts (4)

- ◆ Many kinds of tasks can be solved with machine learning. The following describes the most typical and common types.
 - **Classification:** The computer program is asked to specify which of k categories some input belongs to. To solve this task, the learning algorithm is usually asked to produce a function $f: R^n \rightarrow (1, 2, \dots, k)$. For example, the image classification algorithm in computer vision is to solve a classification task.
 - **Regression:** In this type of task, the computer program is asked to predict a numerical value given some input. The learning algorithm is usually asked to produce a function $f: R^n \rightarrow R$. An example of a regression task is the prediction of the expected claim amount that an insured person will make (used to set insurance premiums), or the prediction of future prices of securities.
- ◆ **Classification and regression are two major types of prediction tasks. The outputs of classification are discrete type values while the outputs of regression are consecutive values.**

Basic Terms and Concepts (5)

◆ Terms:

- P : positive examples, involving major interesting classes
- N : negative examples (other examples)
- TP : true positive examples (positive examples that are correctly classified by the classifier)
- TN : true negative examples (negative examples that are correctly classified by the classifier)
- FP : false positive examples (negative examples that are incorrectly marked as positive examples)
- FN : false negative examples (positive examples that are incorrectly marked as negative examples)

◆ Confusion matrix: a table (at least $m \times m$). $CM_{i,j}$, element of the first i -th line and j -th column is the number of examples known to be in class i but marked as j by the classifier.

- Ideally, for a highly accurate classifier, most data samples should be represented by the elements on the diagonal from $CM_{1,1}$ and $CM_{m,m}$. Other elements are 0 or close to 0. In other words, FP and FN are close to 0.

Predicted Observed	Yes	No	Total
Yes	TP	FN	P
No	FP	TN	N
Total	P'	N'	$P + N$

Confusion matrix

Basic Terms and Concepts (6)

Measure	Formula
Accuracy or correct classification rate	$\frac{TP + TN}{P + N}$
Error rate or false classification rate	$\frac{FP + FN}{P + N}$
Sensitivity, true positive rate, or <i>recall</i>	$\frac{TP}{P}$
Specificity or true negative rate	$\frac{TN}{N}$
<i>Precision</i>	$\frac{TP}{TP + FP}$
<i>F</i> score: harmonic mean of precision or recall	$\frac{2 \times \textit{precision} \times \textit{recall}}{\textit{precision} + \textit{recall}}$
F_{β} (β is a non-negative real number)	$\frac{(1 + \beta^2) \times \textit{precision} \times \textit{recall}}{\beta^2 \times \textit{precision} + \textit{recall}}$

Example of Machine Learning Performance Assessment

- ◆ We have trained a machine learning model to identify that whether the object in an image is a cat. Now we use 200 images to test its performance. Among 200 images, 170 contain cats and 30 do not contain cats. The identification result of the model is that 160 images contain cats and 40 do not contain cats.

$$\text{Precision: } P = \frac{TP}{TP+FP} = \frac{140}{140+20} = 87.5. \%$$

$$\text{Recall: } R = \frac{TP}{P} = \frac{140}{170} = 93.3\%.$$

$$\text{Accuracy: } ACC = \frac{TP+TN}{P+N} = \frac{140+10}{170+30} = 85\%$$

Predicted Observed	Yes	No	Total
Yes	140	30	170
No	20	10	30
Total	160	40	200



Contents

1. Propaedeutics of Deep Learning

- Learning algorithms
- **Common Machine Learning Algorithms**
- Hyperparameter and Validation Set
- Parameter Estimation
- Maximum Likelihood Estimation
- Bayes Estimation

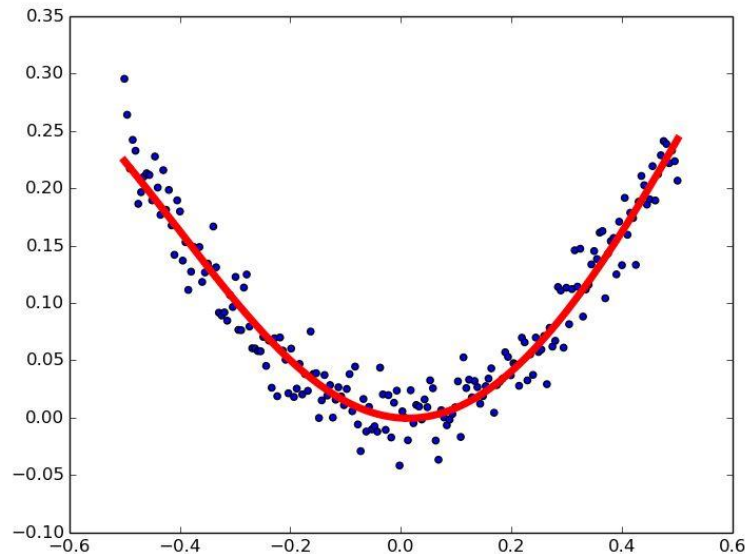
2. Overview of Deep Learning

Types of Machine Learning

- ◆ **Supervised learning:** Based on the samples of known classes, obtain an optimal model with required performance through training and learning. Then, the model is used to map all inputs to outputs and perform simple judgment on outputs (classification). This model classifies unknown data.
- ◆ **Unsupervised learning:** For unlabeled samples, the learning algorithm directly models the input datasets, such as clustering. We only need to put highly similar samples together, calculate the similarity of new samples, and classify them by similarity.
- ◆ **Semi-supervised learning:** Enable the learner to automatically use a large amount of unlabeled data to assist learning of a small amount of labeled data.
- ◆ **Reinforcement learning:** Learning systems learn mappings from environments to actions to maximize the value of the reward signal (reinforcement signal) function. The difference between reinforcement learning and supervised learning is the teacher signal. The reinforcement signal provided by the environment in reinforcement learning is an assessment on the action (scalar signal) rather than telling the learning system how to perform correct actions.

Common Machine Learning Algorithm: Linear Regression

- ◆ Linear regression: a statistical analysis method to determine the quantitative relationships between two or more variables through regression analysis in mathematical statistics.



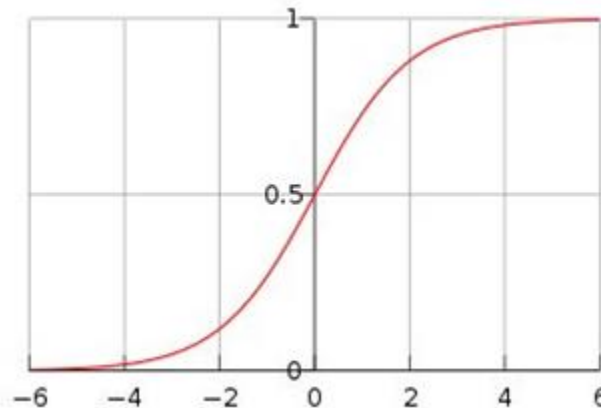
Common Machine Learning Algorithm: Logistic Regression

- ◆ Logistic regression: The logistic regression model is used to solve classification problems. Model definition:

$$P(Y = 1|x) = \frac{e^{wx+b}}{1 + e^{wx+b}}$$

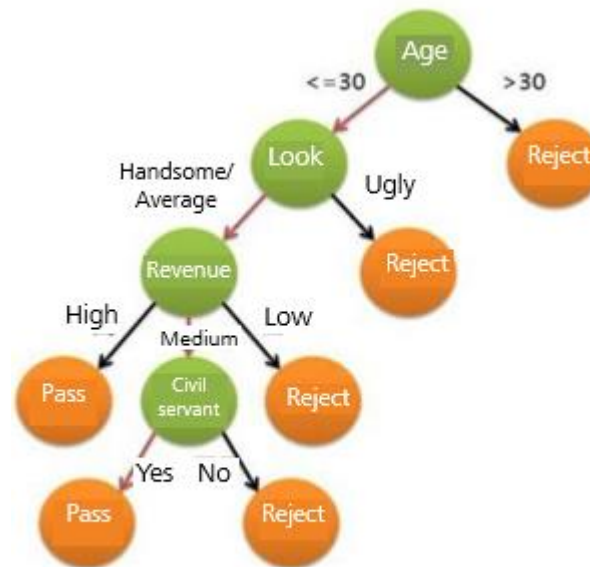
$$P(Y = 0|x) = \frac{1}{1 + e^{wx+b}}$$

where w is the weight, b is the bias, $wx + b$ is the linear function of x . Compare the two probability values. x belongs to the class with a larger probability value.



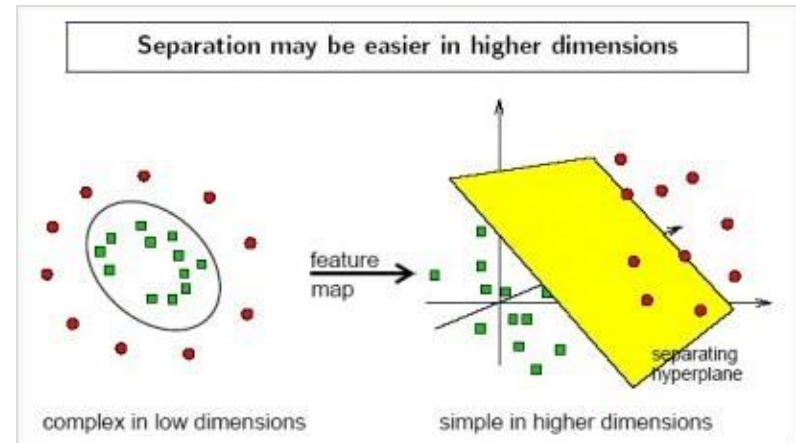
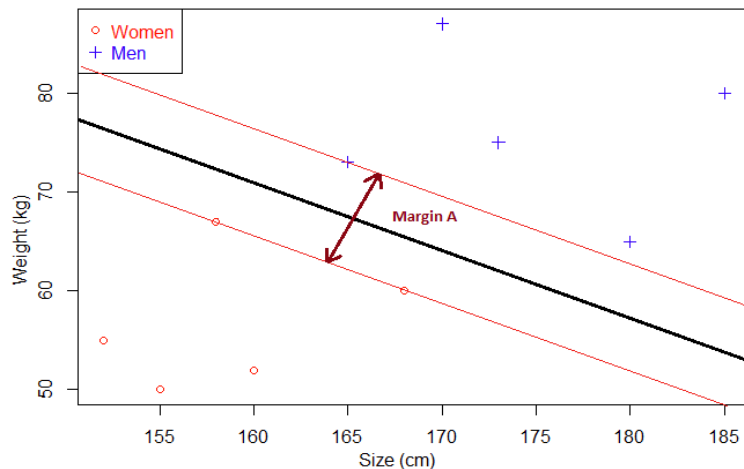
Common Machine Learning Algorithm: Decision Tree

- ◆ Decision tree: A decision tree is a tree-like structure (a binary tree or a non-binary tree). Each non-leaf node represents a test on a feature attribute. Each branch represents the output of a feature attribute at a certain value range, and each leaf node stores a class. To use the decision tree, start from the root node, test the feature attributes of the items to be classified, select the output branches, and use the class stored on the leaf node as the final result.



Common Machine Learning Algorithm: Support Vector Machine

- ◆ A support vector machine (SVM) is a two-class classification model, and its basic model is a linear classifier defined in the feature space with the largest interval. SVM also includes kernel tricks that make it a non-linear classifier. The SVM learning algorithm is the optimum solution to convex quadratic programming.



Common Machine Learning Algorithm: Naive Bayes

- ♦ **Naive Bayes algorithm:** A classification method based on **Bayes' theorem** and **attribute conditional independence assumption**. For a given training set, the algorithm learns the joint probability distribution of inputs/outputs based on the attribute conditional independence assumption, and then based on this model, obtain the output y with the maximum posterior probability for a given input x using Bayes' theorem. According to Bayes' theorem, in a classification problem, for a given sample feature X , the probability that the sample belongs to the class H is:

$$P(H|X) = \frac{P(X|H)P(H)}{P(X)}$$

X is the data sample and generally described with the measurement value of n attribute sets. H is a hypothesis, for example, data sample X belonging to one certain class C . $P(H|X)$ is the posterior probability or H 's posterior probability under the condition X . $P(H)$ is the prior probability or H 's prior probability and is independent of X . $P(X)$ is the prior probability of X .

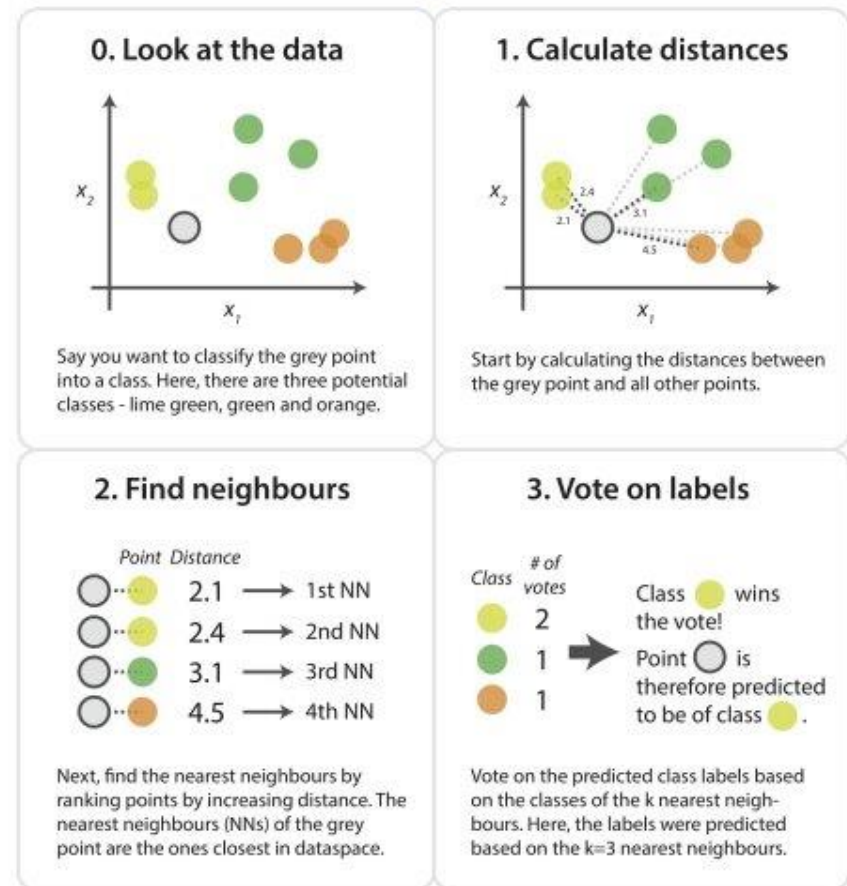
- ♦ Based on the naive assumption, namely attribute conditional independence assumption, and the law of total probability, the basic formula of Naive Bayes classification is as follows:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)} = \frac{\prod_{k=1}^M P(X_i|C_k)P(C_k)}{\sum_k P(C_k) \prod_{k=1}^M P(X_i|C_k)}$$

Common Machine Learning Algorithm: KNN

- ◆ The k-nearest neighbors (KNN) classification algorithm is a theoretically mature method and one of the simplest machine learning algorithms. If the majority of k samples that are most similar to one sample (nearest neighbors in the feature space) belongs to one class, the sample also belongs to this class.

kNN Algorithm





Contents

1. Propaedeutics of Deep Learning

- Learning algorithms
- Common Machine Learning Algorithms
- **Hyperparameter and Validation Set**
- Parameter Estimation
- Maximum Likelihood Estimation
- Bayes Estimation

2. Overview of Deep Learning

Hyperparameter and Validation Set (1)

- ◆ There are two types of parameters in the learning model. One can be obtained from learning and the other cannot be obtained from data but is set based on human's experience. The latter is called hyperparameter.
- ◆ A model hyperparameter is a configuration that is external to the model. It has the following features:
 - It is often used in processes to help estimate model parameters.
 - It is often specified by the practitioner.
 - It can often be set using heuristics.
 - It is often tuned for a given predictive modeling problem.
- ◆ Examples:
 - Learning rate for training a neural network, number of iterations, batch size, activation function, number of neurons
 - C and σ hyperparameters of support vector machines
 - K in KNN

Hyperparameter and Validation Set (2)

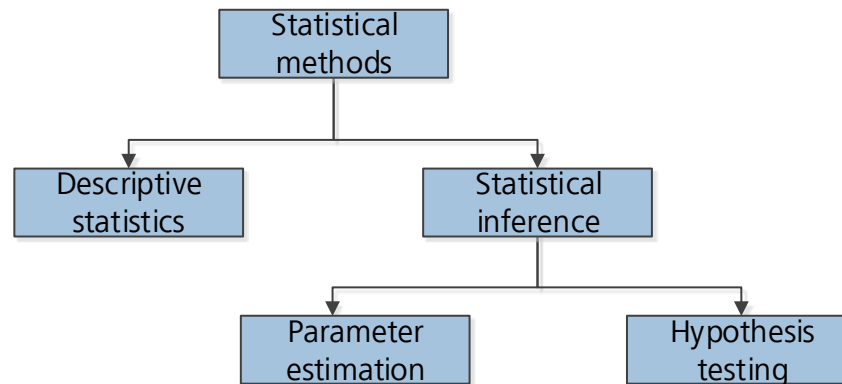
- ◆ The procedure for searching for hyperparameters:
 - Divide a dataset into a training set, verification set, and test set.
 - Optimize the model parameters in the training set based on the performance indicators of the model.
 - Search for the model hyperparameters in the training set based on the performance indicators of the model.
 - Step 2 and step 3 are performed alternately. Finally, obtain parameters and hyperparameters of the model and assess the model in the test set.
- ◆ The search process in step 3 requires search algorithms, including:
 - Grid search
 - Random search
 - Heuristic intelligent search
 - Bayesian search

Hyperparameter and Validation Set (3)

- ◆ **Cross validation:** It is a statistical analysis method used to validate the performance of a classifier. The basic idea is to divide the original dataset into two parts: training set and validation set. Train the classifier on the training set and test the model on the validation set. Then obtain a performance indicator of the classifier.
- ◆ **K-fold cross validation (K-CV):**
 - Partition the original training data set into k (equal) subsets.
 - Each time, one subset is used as a validation set and the remaining $k - 1$ subsets as the training set. Then we obtain k models.
 - Use the mean classification accuracy of the validation set of k models as the performance indicator of the K-CV classifier.

Parameter Estimation

- **Parameter estimation:** Suppose that there is a statistical population and its distribution function is $F(x, \theta)$, where θ is an unknown parameter. Now we sample the population and obtain the sample X_1, X_2, \dots, X_n . We need to find an estimator $\hat{\theta}$ of the parameter θ or estimate θ 's function $g(\theta)$. This problem is called **parameter estimation**, including **point estimation** and **interval estimation**.
- **The position of parameter estimation in statistical methods:**





Contents

1. Propaedeutics of Deep Learning

- Learning algorithms
- Common Machine Learning Algorithms
- Hyperparameter and Validation Set
- **Parameter Estimation**
- Maximum Likelihood Estimation
- Bayes Estimation

2. Overview of Deep Learning

Point Estimator

- ◆ In one area, the fetal weight is $X \sim N(\mu, \sigma^2)$ (μ and σ^2 are unknown). Randomly select 100 babies and obtain their weight data

10, 7, 5, 6, 6, 5.2, ...

We have 100 figures and how do we estimate μ and σ ?

- ◆ Point estimator: We need to construct a function of an appropriate sample: $T(X_1, X_2, \dots, X_n)$. Once we have a sample, we can input the value into the function to obtain a result which can be used as the estimator of μ . $T(X_1, X_2, \dots, X_n)$ is the parameter's point estimator.
- ◆ Problems: Which estimator can be used to estimate μ ? How do we assess the estimators?
 - Mean of the samples
 - Median of the samples
 - Other statistics

Statistics Assessment Standard: Unbiased

- **Unbiased:** The bias of an estimator is defined as

$$\text{bias}(\hat{\theta}) = E(\hat{\theta}) - \theta$$

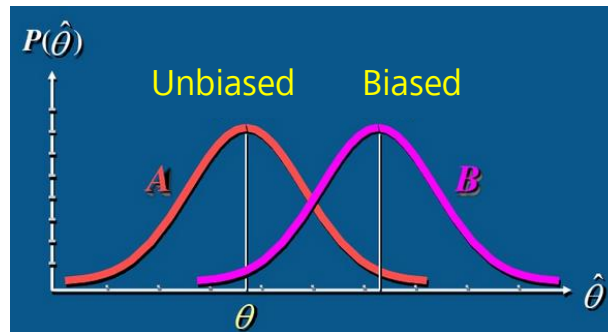
An estimator $\hat{\theta}$ is said to be unbiased if $\text{bias}(\hat{\theta}) = 0$, which implies that $E(\hat{\theta}) = \theta$.

An estimator $\hat{\theta}$ is said to be asymptotically unbiased if $\lim \text{bias}(\hat{\theta}) = 0$, which implies that $\lim E(\hat{\theta}) = \theta$.

- **Example:** Suppose (X_1, X_2, \dots, X_n) is a sample from the population X . $EX = \mu$, $DX = \sigma^2$. Prove:

(1) $\bar{X} = \frac{1}{n} \sum_{i=1}^n X_i$ is an unbiased estimator of μ .

(2) $S^2 = \frac{1}{n-1} (\sum_{i=1}^n X_i^2 - n\bar{X}^2)$ is an unbiased estimator of σ^2 .



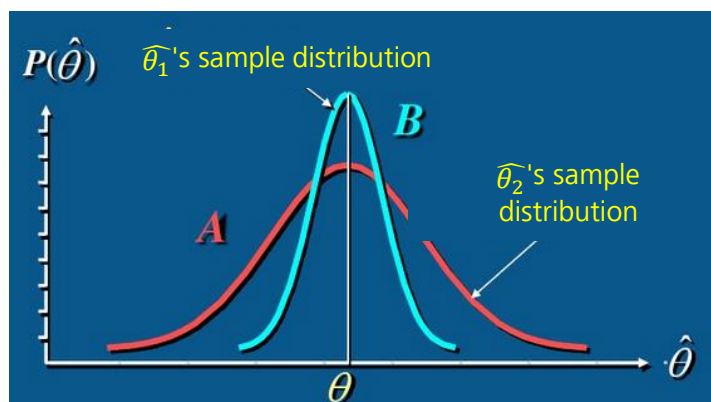
Statistics Assessment Standard: Valid

- ◆ Example: X_1, X_2 , and X_3 is an sample of the population X . Are the following statistics the unbiased estimators of the mean μ ? Which one is optimal?

$$\widehat{\mu}_1 = \frac{2}{5}X_1 + \frac{1}{10}X_2 + \frac{1}{2}X_3, \quad \widehat{\mu}_2 = \frac{1}{3}X_1 + \frac{3}{4}X_2 - \frac{1}{12}X_3,$$

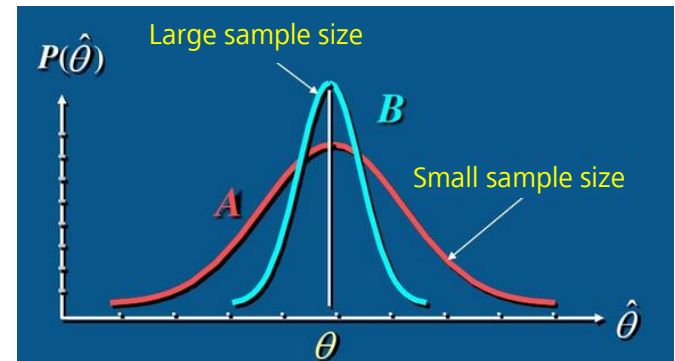
$$\widehat{\mu}_3 = \frac{1}{2}X_1 + \frac{1}{3}X_2 + \frac{1}{6}X_3, \quad \widehat{\mu}_4 = \frac{1}{5}X_1 + \frac{1}{10}X_2 + \frac{7}{10}X_3.$$

- ◆ **Valid:** Suppose $\widehat{\theta}_1$ and $\widehat{\theta}_2$ are two unbiased estimators of θ . If $D(\widehat{\theta}_1) < D(\widehat{\theta}_2)$, $\widehat{\theta}_1$ is more valid than $\widehat{\theta}_2$.



Statistics Assessment Standard: Consistent

- ♦ **Consistent:** With probability P , $\hat{\theta} \rightarrow \theta (n \rightarrow \infty)$, i.e. for $\forall \epsilon > 0$, $\lim_{n \rightarrow \infty} P(|\hat{\theta} -$





Contents

1. Propaedeutics of Deep Learning

- Learning algorithms
- Common Machine Learning Algorithms
- Hyperparameter and Validation Set
- Parameter Estimation
- **Maximum Likelihood Estimation**
- Bayes Estimation

2. Overview of Deep Learning

Maximum Likelihood Estimation (1)

- ◆ Suppose that an urn contains a number of black balls and a number of white balls. It is known that the ratio is 1:3. However, it is not known whether the urn has more black balls or more white balls. If sampling is performed for n times, the probability of the number of black balls as x is:

$$p(x; p) = \binom{n}{x} p^x q^{n-x},$$

where $q = 1 - p$. According to the assumption, $p = \frac{1}{4}$ or $\frac{3}{4}$. If $n = 3$, x 's probability under p is:

x	0	1	2	3
$P(x, \frac{3}{4})$	$\frac{1}{64}$	$\frac{9}{64}$	$\frac{27}{64}$	$\frac{27}{64}$
$P(x, \frac{1}{4})$	$\frac{27}{64}$	$\frac{27}{64}$	$\frac{9}{64}$	$\frac{1}{64}$

$$\hat{p}(x) = \begin{cases} \frac{1}{4}, & x = 0, 1 \\ \frac{3}{4}, & x = 2, 3 \end{cases}$$

Maximum Likelihood Estimation (2)

- ◆ **Maximum likelihood estimation principle:** Suppose that X_1, X_2, \dots, X_n is a sample from the population X . Its joint density or joint distribution function is $f(x_1, x_2, \dots, x_n; \theta)$. Define the likelihood function as:

$$L(\theta) = f(x_1, x_2, \dots, x_n; \theta),$$

where x_1, x_2, \dots, x_n is the observed value of the sample. $L(\theta)$ is a function of parameter θ and can be used to assess the possibility of θ generating x_1, x_2, \dots, x_n .

- ◆ Maximum likelihood estimation uses $\hat{\theta}$ that maximizes $L(\theta)$ to estimate θ .

$$L(\hat{\theta}) = \max_{\theta} L(\theta),$$

$\hat{\theta}$ is the maximum likelihood estimate of θ . The statistic $\hat{\theta}(X_1, X_2, \dots, X_n)$ is the maximum likelihood estimator of θ .

Maximum Likelihood Estimation Example

- ◆ Example: Suppose that X_1, X_2, \dots, X_n is a sample from the population $X \sim B(1, p)$. Find the maximum likelihood estimator of the parameter p .

SOLUTION: Likelihood function:

$$L(p) = f(x_1, x_2, \dots, x_n; p) = \prod_{i=1}^n p^{x_i} (1-p)^{1-x_i} = p^{\sum_{i=1}^n x_i} (1-p)^{n - \sum_{i=1}^n x_i}.$$

Log likelihood function:

$$\ln L(p) = \sum_{i=1}^n x_i \ln(p) + (n - \sum_{i=1}^n x_i) \ln(1-p).$$

Derive the function with respect to p and let it be 0:

$$\frac{d \ln L(p)}{dp} = \frac{1}{p} \sum_{i=1}^n x_i - \frac{1}{1-p} (n - \sum_{i=1}^n x_i) = 0.$$

$\hat{p} = \bar{x}$. Therefore the maximum likelihood estimator of p is:

$$\hat{p}(X_1, X_2, \dots, X_n) = \frac{1}{n} \sum_{i=1}^n X_i = \bar{X}.$$



Contents

1. Propaedeutics of Deep Learning

- Learning algorithms
- Common Machine Learning Algorithms
- Hyperparameter and Validation Set
- Parameter Estimation
- Maximum Likelihood Estimation
- **Bayes Estimation**

2. Overview of Deep Learning

Bayes' Theorem

- ◆ Bayes' theorem calculates the posterior probability $P(h|D)$ with $P(D)$, $P(h)$, and $P(D|h)$. The formula is as follows:

$$P(h | D) = \frac{P(D | h)P(h)}{P(D)}$$

D is the data sample and generally described using measurement values of n attribute sets. h is a hypothesis, for example, data sample D belonging to one certain class C . $P(h | D)$ is the posterior probability or h 's posterior probability given that D is true. $P(h)$ is the prior probability or h 's prior probability and is independent of D . $P(D)$ is D 's prior probability.

Maximum a Posteriori Hypothesis

- ◆ $P(h|D)$ is the posterior probability of the hypothesis h . The hypothesis h that maximizes $P(h|D)$ is the maximum a posteriori (MAP) hypothesis. Formally, MAP supposes that h_{MAP} satisfies:
$$h_{MAP} = \operatorname{argmax}_{h \in H} P(h|D) = \operatorname{argmax}_{h \in H} \frac{P(D|h)P(h)}{P(D)} = \operatorname{argmax}_{h \in H} P(D|h)P(h)$$
- ◆ In practice, we usually cannot obtain the prior probability of each hypothesis. We only assume that all hypotheses in the hypothesis space are equally likely. $P(h)$ is a constant and
$$h_{MAP} = \operatorname{argmax}_{h \in H} P(D|h)P(h) = \operatorname{argmax}_{h \in H} P(D|h) = h_{ML}$$
where $P(D|h)$ is the likelihood of data D given h . Therefore the hypothesis that maximizes $P(D|h)$ is the maximum likelihood hypothesis h_{ML} .
- ◆ If hypotheses in the hypothesis space are equally likely, $h_{MAP} = h_{ML}$.

Challenges Motivating Deep Learning

- ◆ Curse of dimensionality
- ◆ Local constancy and smoothness regularization
- ◆ Manifold learning



Contents

1. Propaedeutics of Deep Learning

2. Deep Learning Overview

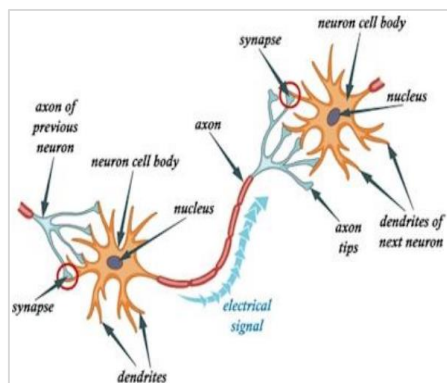
- **Definition and Development of Neural Networks**
- Perceptron and Training Rules
- Activation Functions
- Types of Neural Networks
- Regularization in Deep Learning
- Optimizer
- Applications of Deep Learning

Neural Network Definition

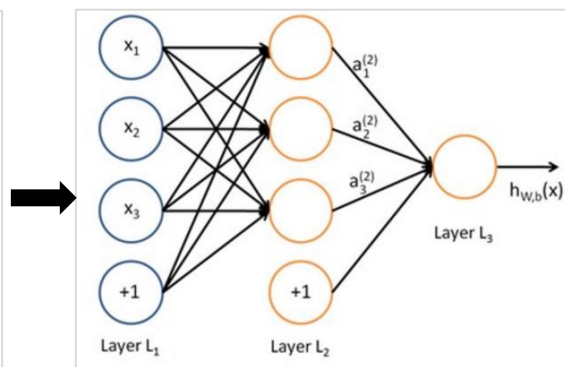
- ◆ Currently, the definition of the neural network is not determined yet. Hecht Nielsen, a neural network researcher in the US, defines a neural network as "a computing system made up of a number of simple, highly interconnected processing elements, which process information by their dynamic state response to external inputs."
- ◆ Based on the origin, features, and interpretations of the neural network, it can be simply defined as **an information processing system designed to simulate human brain's structure and functions.**
- ◆ **Artificial neural network (neural network for short):** refers to a network composed of artificial neurons. It abstracts and simplifies a human brain based on its microscopic structure and functions. It is an important way to simulate human intelligence and reflects some basic features of human brain functions, such as parallel information processing, learning, association, pattern classification, and memory.

Deep learning

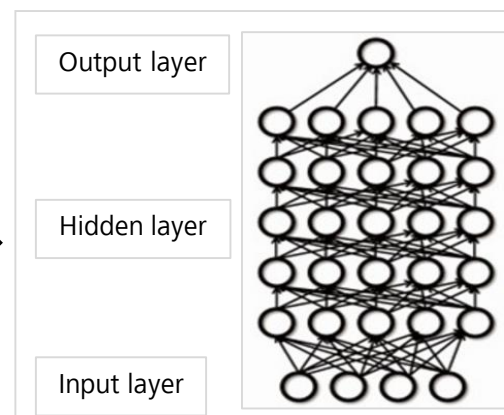
- ◆ Deep learning generally involves a deep neural network, where the depth refers to the number of layers of the neural network.



Artificial neural network

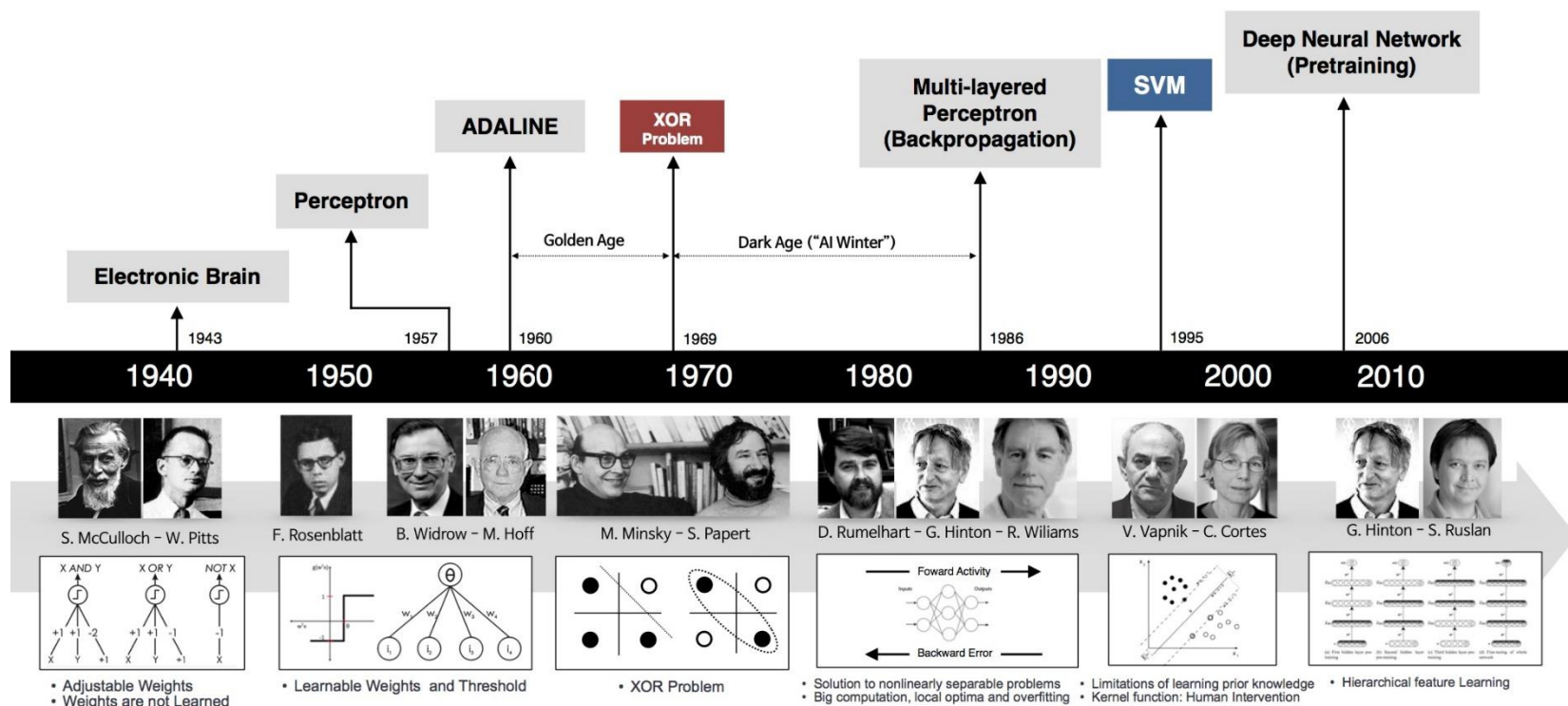


Perceptron



Deep neural network

Deep Learning Milestones





Contents

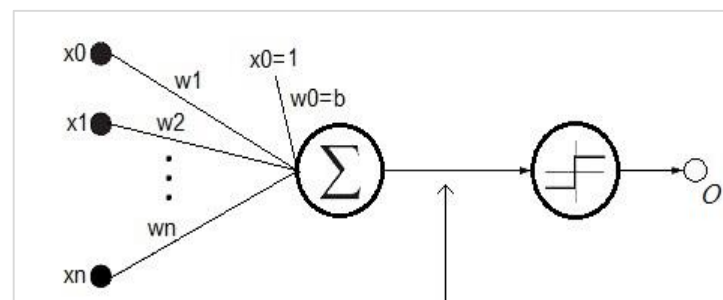
1. Propaedeutics of Deep Learning

2. Deep Learning Overview

- Definition and Development of Neural Networks
- **Perceptron and Training Rules**
- Activation Functions
- Types of Neural Networks
- Regularization in Deep Learning
- Optimizer
- Applications of Deep Learning

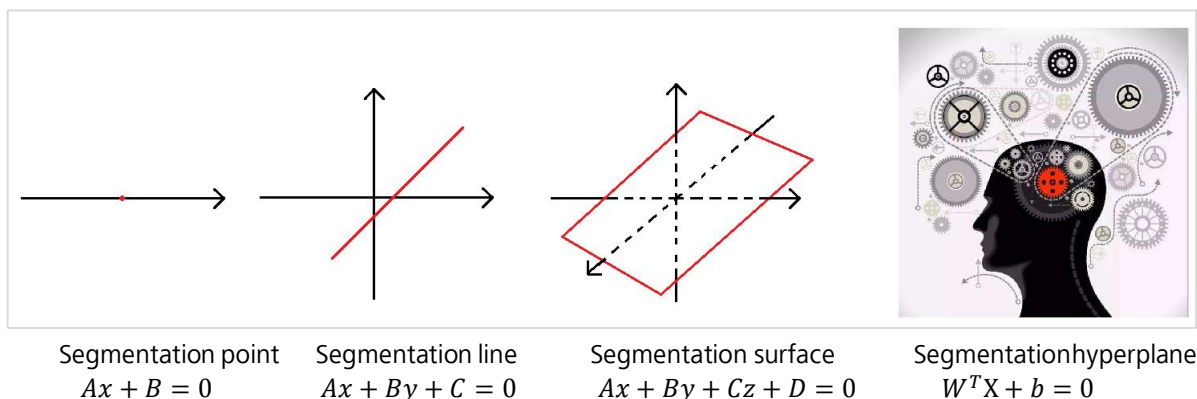
Perceptron

- ◆ **Input vector:** $X = [x_0, x_1, \dots, x_n]^T$.
- ◆ **Weight:** $W = [\omega_0, \omega_1, \dots, \omega_n]^T$, where ω_0 is the bias.
- ◆ **Activation function:** $O = \text{sign}(\text{net}) = \begin{cases} 1, & \text{net} > 0, \\ -1, & \text{otherwise.} \end{cases}$



$$\text{net} = \sum_{i=0}^n \omega_i x_i = \mathbf{W}^T \mathbf{X}$$

- ◆ The perceptron is equivalent to a classifier. Its input is the high-dimensional vector X and it performs binary classification on input samples in the high-dimensional space. If $\mathbf{W}^T \mathbf{X} > 0$, $o = 1$ and the sample is classified into one class. Otherwise, $o = -1$ and the sample is classified into the other class. What is the boundary? $\mathbf{W}^T \mathbf{X} = 0$. This is a high-dimensional hyperplane.



Perceptron Training Rules

- ◆ Perceptron training rules: For each training sample $\langle X, t \rangle$
 - Use the current weights to calculate the perceptron output o .
 - Each weight is updated as follows:

$$\omega_i \leftarrow \omega_i + \Delta\omega_i$$

$$\Delta\omega_i = \eta[t - o]x_i$$

where X is the input vector, t is the target value, o is the output under the current weights, η is the learning rate, x_i and ω_i are the i -th elements of vectors X and W .

- ◆ When the training sample is linearly separable, the preceding method is repeatedly used. After finite times of training, the perceptron converges to a classifier that can correctly classify all training samples.
- ◆ When the training sample is not linearly separable, the training may be unable to converge. Therefore, another rule is designed to overcome this deficiency, namely the **delta rule**. It uses the **gradient descent** method to search for the possible weight vectors in the hypothesis space to find the weight vectors that best fit the training example.

Gradient Descent and Loss Function

- ◆ For the multivariable function $o = f(x) = f(x_0, x_1, \dots, x_n)$, its gradient at $X' = [x_0', x_1', \dots, x_n']^T$ is:

$$\nabla f(x_0', x_1', \dots, x_n') = \left[\frac{\partial f}{\partial x_0}, \frac{\partial f}{\partial x_1}, \dots, \frac{\partial f}{\partial x_n} \right]^T \Big|_{X=X'}.$$

Direction of the gradient vector is the steepest ascent direction of the function. Therefore, the negative gradient vector $-\nabla f$ points to the steepest descent direction of the function.

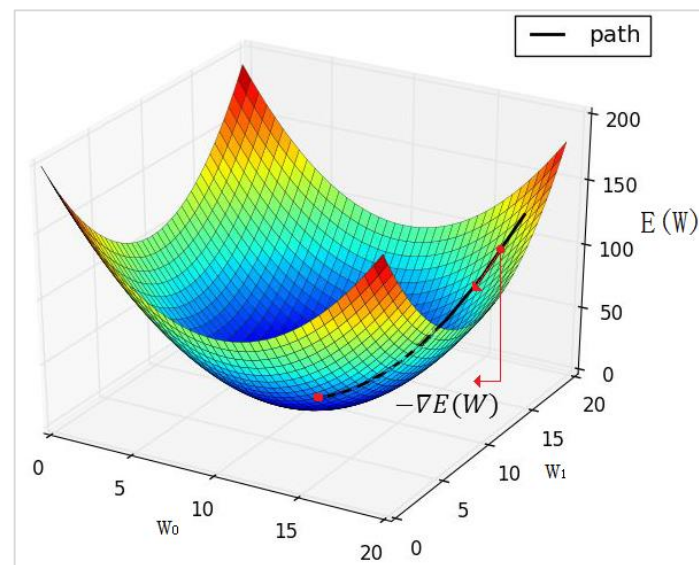
- ◆ When training samples are not linearly separable, we cannot find a hyperplane to enable the perceptron to perfectly classify training samples. However, we can classify them approximately and allow some minor classification errors. To minimize the errors, we need to first parameterize the error using the **loss function (error function)**. The function reflects the error between the target output and the actual output of the perceptron. The most common error function is **L2 loss function**:

$$E(w) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2,$$

where d is the training example, D is the training example set, t_d is the target output, and o_d is the actual output.

Extrema of the Loss Function

- ◆ The independent variable of the loss function $E(W)$ is a weight so the function is defined in a weight space. Then we need to search for a weight vector W that minimizes $E(W)$ in the weight space. Unfortunately, $E(W) = \frac{1}{2} \sum_{d \in D} (t_d - o_d)^2$ defines a very complex high-dimensional curved surface and **there is no effective mathematical method to calculate the extrema of a high-dimensional curved surface**. Considering that the negative gradient direction is the steepest descent direction, we can start from a point and go along the $-\nabla E(W)$ direction to find the minimum point of $E(W)$. This is the core idea of the gradient descent method.



Gradient descent on the binary paraboloid

Global Gradient Descent Algorithm for Linear Units

- ◆ Each sample in the training sample set D is denoted as $\langle X, t \rangle$. X is the input vector, t is the target output, and η is the learning rate.
 - Initialize each w_i to a random value with a small absolute value.
 - Before the termination condition is met, do:
 - ▣ Initialize each Δw_i to 0.
 - ▣ For each $\langle X, t \rangle$ in D , do:
 - Enter X in the unit and calculate the output o .
 - For each w_i in the unit, do $\Delta w_i += \eta(t - o) x_i$
 - ▣ For each w_i in the unit, do $w_i += \Delta w_i$
- ◆ This gradient descent algorithm is not commonly used because it has the following defects:
 - The convergence process is very slow because all training samples need to be calculated every time the weight is updated.
 - If the error surface has multiple local minima, the process is easily trapped in local extrema.

SGD Algorithm and Online Learning

- ◆ To address the defects of the original gradient descent algorithm, a common variant, incremental gradient descent algorithm, is used, which is also called stochastic gradient descent (SGD) algorithm. One implementation is called online learning, which updates the gradient based on each sample:

$$\Delta w_i = \eta \sum_{d \in D} (t_d - o_d) x_{id} \Rightarrow \Delta w_i = \eta (t_d - o_d) x_{id}.$$

- ◆ Online-Gradient-Descent (D, η)
 - Initialize each w_i to a random value with a small absolute value.
 - Before the termination condition is met, do:
 - ▣ For each $\langle X, t \rangle$ in D , do:
 - Enter X in the unit and calculate the output o .
 - For each w_i in the unit, do $\Delta w_i \leftarrow \eta(t - o) x_i$.

Mini-Batch Gradient Descent

- ◆ To address the defects of the previous two gradient descent algorithms, another algorithm, mini-batch gradient descent, has been proposed. It is the most widely used one. The idea is to use samples of a fixed batch size (BS) to calculate Δw_i and update the weight.
- ◆ Batch-Gradient-Descent (D, η, BS)
 - Initialize each w_i to a random value with a small absolute value.
 - Before the termination condition is met, do:
 - ▣ Initialize each Δw_i to 0.
 - ▣ Select samples (BS) from D . For each $\langle X, t \rangle$ among these samples, do:
 - Enter X in the unit and calculate the output o .
 - For each w_i in the unit, do $\Delta w_i += \eta(t-o) x_i$.
 - ▣ For each w_i in the unit, do $w_i += \Delta w_i$
 - ▣ If it is the last batch, disrupt the training sample sequence.



Contents

1. Propaedeutics of Deep Learning

2. Deep Learning Overview

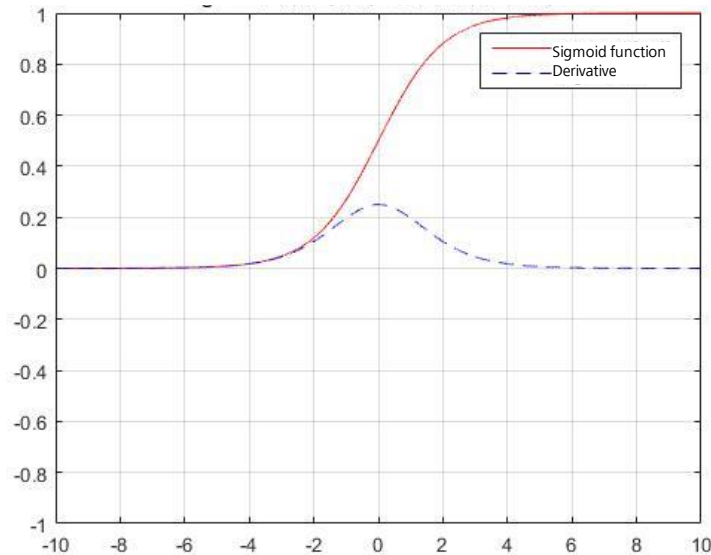
- Definition and Development of Neural Networks
- Perceptron and Training Rules
- **Activation Functions**
- Types of Neural Networks
- Regularization in Deep Learning
- Optimizer
- Applications of Deep Learning

Activation Function: Sigmoid

◆ Sigmoid function:

$$f(x) = \frac{1}{1 + e^{-x}}$$

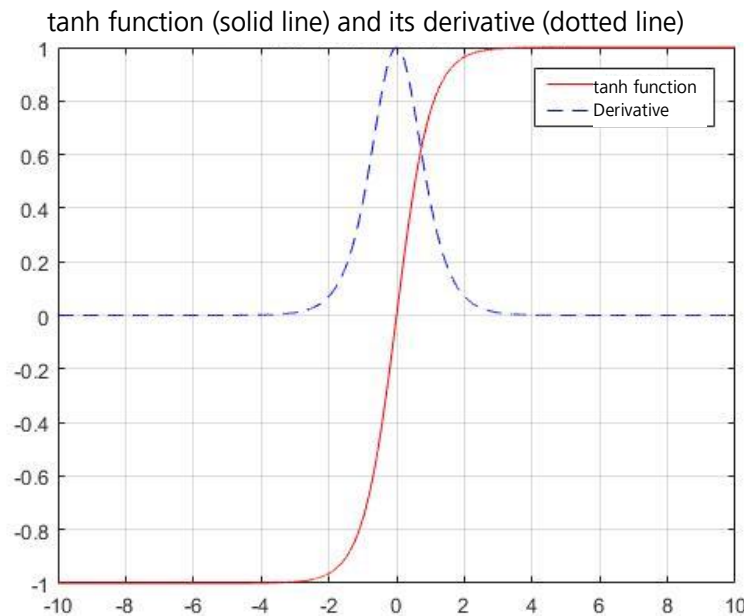
Sigmoid function (solid line) and its derivative (dotted line)



Activation Function: tanh

◆ tanh function:

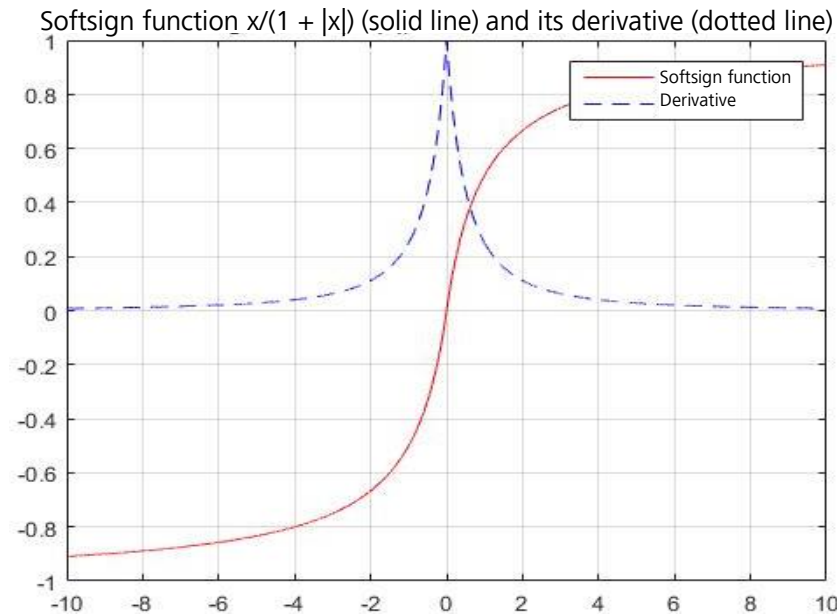
$$\tanh x = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



Activation Function: Softsign

◆ Softsign function:

$$f(x) = \frac{x}{|x| + 1}$$

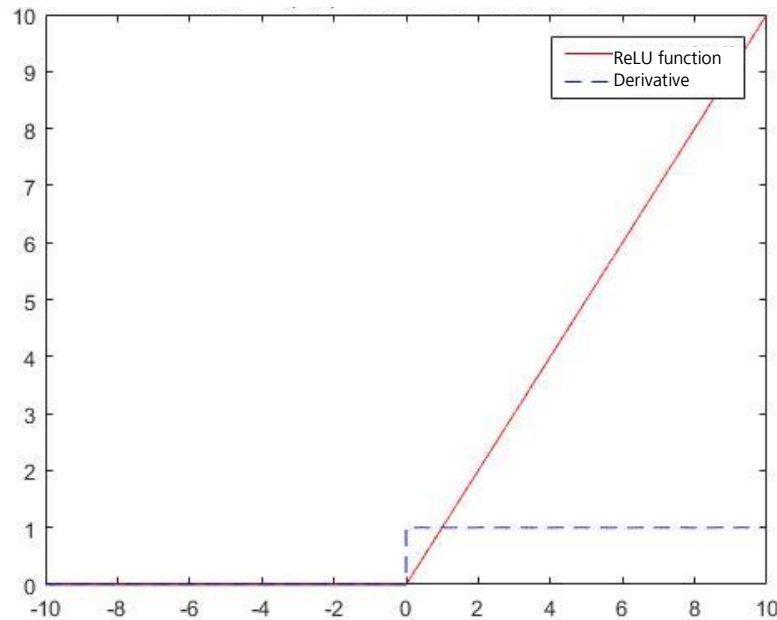


Activation Function: ReLU

◆ Rectified Linear Unit (ReLU) function:

$$y = \begin{cases} x, & x \geq 0 \\ 0, & x < 0 \end{cases}$$

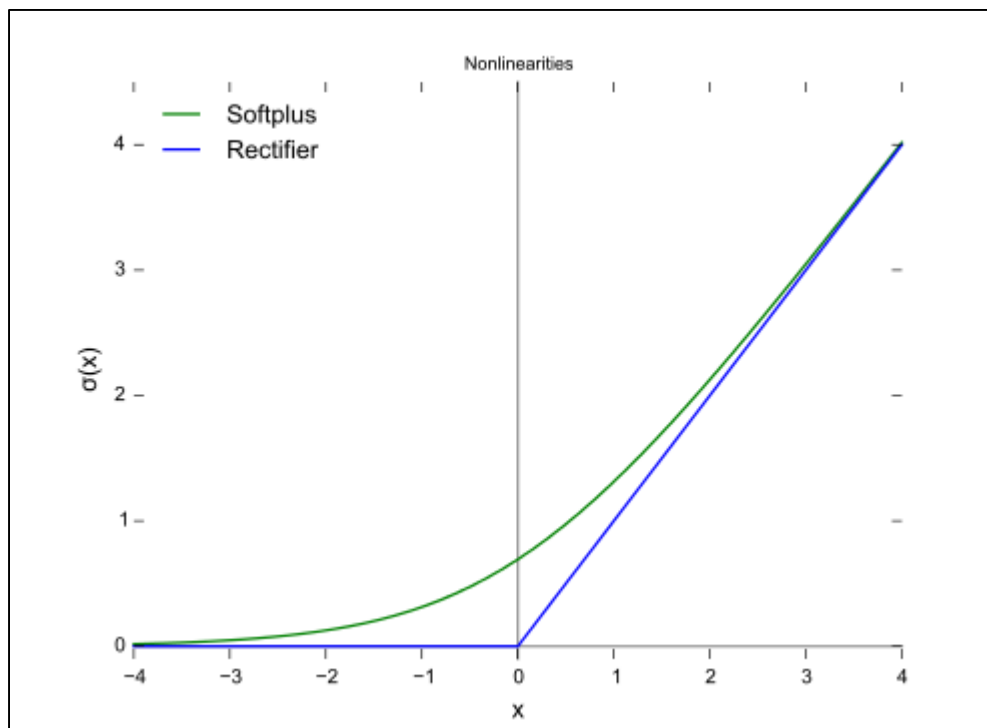
ReLU function $\max(0, x)$ (solid line) and its derivative 0, 1 (dotted line)



Activation Function: Softplus

◆ Softplus function:

$$f(x) = \ln(e^x + 1)$$



Factors to Be Considered in Activation Function Design

- ◆ **Non-linear:** When the activation function is non-linear, a two-layer neural network can be proved to be a common function approximation. If the function is linear, the entire network is equivalent to a single-layer linear model.
- ◆ **Continuously differentiable:** This attribute is desirable for enabling gradient-based optimization methods. If some functions that are locally non-differentiable are selected, the derivative must be defined forcibly.
- ◆ **Range:** When the range of the activation function is finite, gradient-based training methods tend to be more stable. When the range is infinite, training is generally more efficient but it is easy to diverge. Smaller learning rates are typically necessary.
- ◆ **Monotonicity:** When the activation function is monotonic, the loss function associated with a single-layer model is convex.
- ◆ **Smooth:** A smooth function with a monotonic derivative has been proved to generalize better in some cases.
- ◆ **Approximating identity near the origin:** When activation functions have this property, the neural network will learn efficiently when its weights are initialized with small random values. When the activation function does not approximate identity near the origin, special care must be used when initializing the weights.



Contents

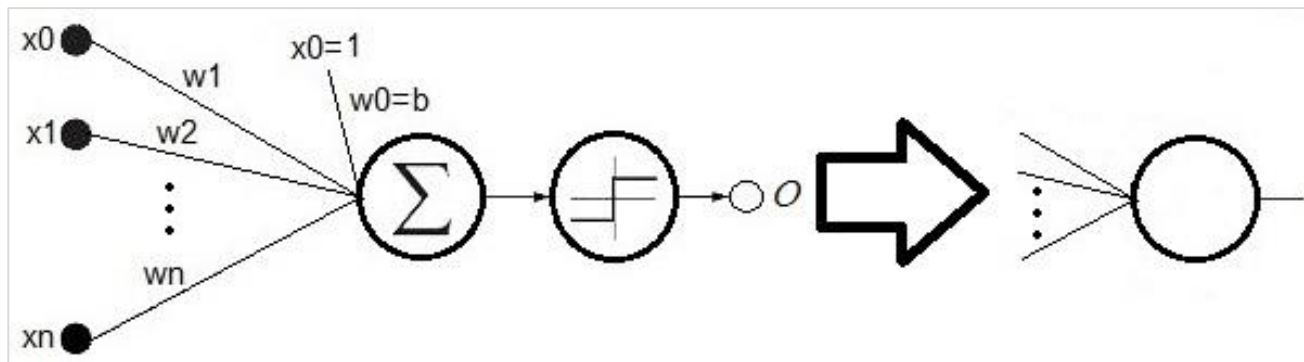
1. Propaedeutics of Deep Learning

2. Deep Learning Overview

- Definition and Development of Neural Networks
- Perceptron and Training Rules
- Activation Functions
- **Types of Neural Networks**
- Regularization in Deep Learning
- Optimizer
- Applications of Deep Learning

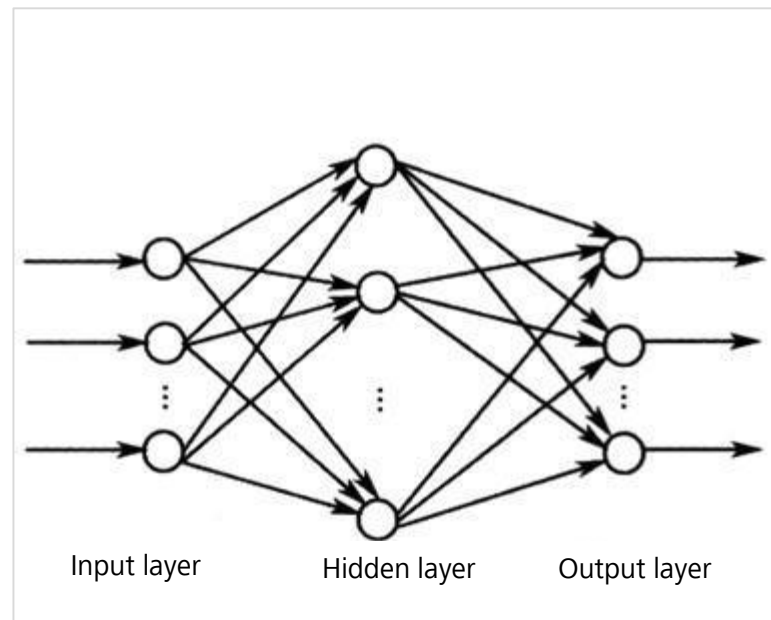
Multi-layer Fully Connected Artificial Neural Network

- ◆ A single perceptron has limited representation capability and can only represent the linear decision surface (hyperplane). If we connect many perceptrons like a human brain does and then replace the activation function with a non-linear function, we can express a wide range of non-linear surfaces.



Feedforward Neural Network

- ◆ **The feedforward neural network** is one of the simplest neural networks in which neurons are arranged hierarchically. It is the most widely neural network with the fastest development.
- ◆ The input node does not support computation, but is merely used to represent each element value of the input vector.
- ◆ Each node represents a neuron that supports computation, which is called a computational unit. Each neuron is connected only to the neurons at the previous layer.
- ◆ A hidden layer receives the output from the previous layer and sends the results to the next layer. A unidirectional multi-layer structure is adopted. Each layer contains several neurons. The neurons at the same layer are not connected to each other. Inter-layer information is transmitted only in one direction.



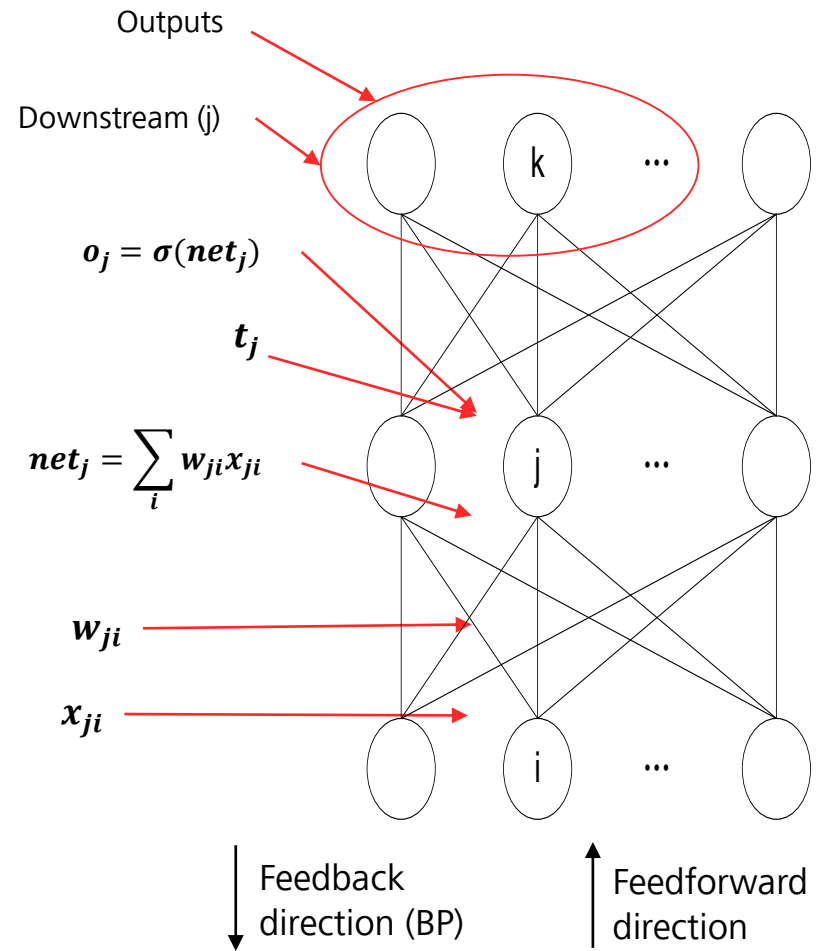
Backpropagation Algorithm

- ◆ We can train the neural network with this formula:

$$\Delta w_{ji} = \eta \delta_j x_{ji}$$

$$\delta_j = \begin{cases} o_j(1-o_j)(t_j - o_j), & j \in \text{outputs} & (1) \\ o_j(1-o_j) \sum_{k \in DS(j)} \delta_k w_{kj}, & \text{otherwise} & (2) \end{cases}$$

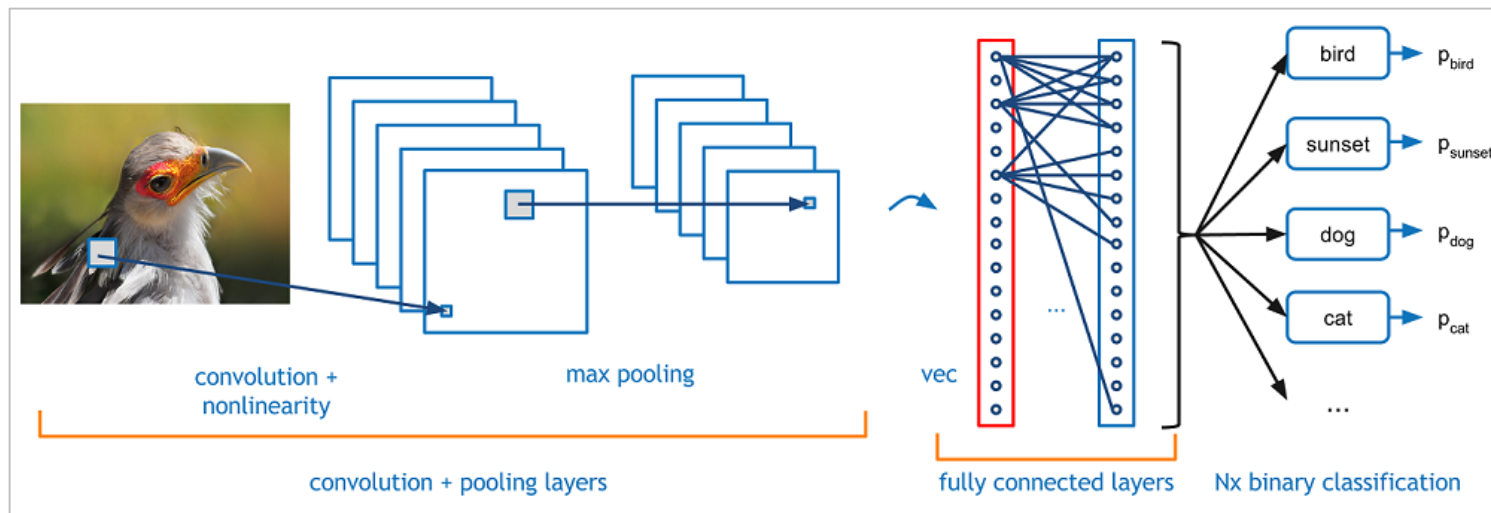
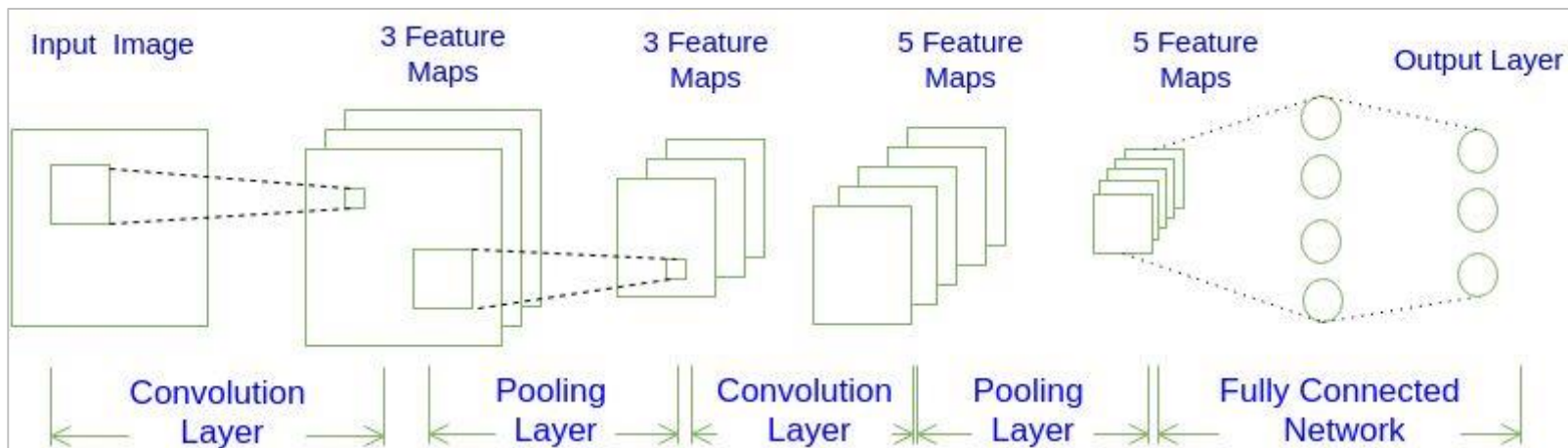
- ◆ The procedure for using the BP algorithm to train the network is as follows:
 - Take out the next training sample $\langle X, T \rangle$, and enter X to the network to obtain the actual output O .
 - Calculate δ of the output layer based on the output layer error formula (1) and update the weight.
 - For the hidden layer, according to the hidden layer error propagation formula (2), the δ of each layer is calculated from the output direction to the input direction, layer by layer with iteration. Once the δ of a layer is calculated, update the weight of the layer until all weights are updated.
 - Return to 1 to continue.



Convolutional Neural Network (1)

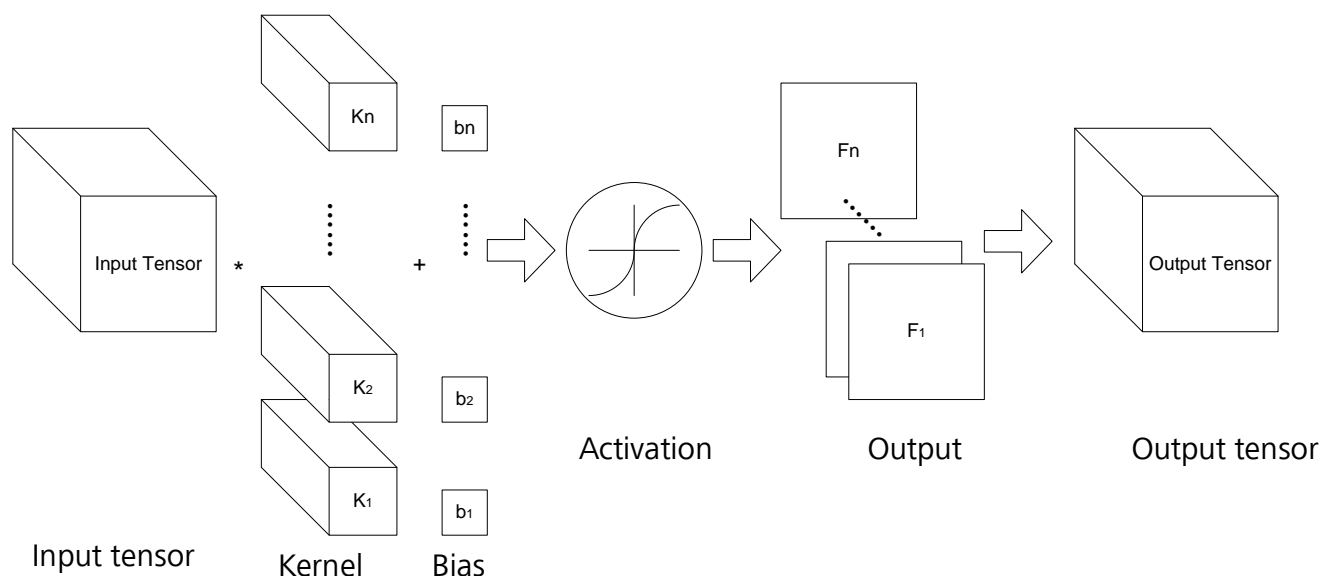
- ◆ A convolutional neural network (CNN) is a feedforward neural network. Its artificial neurons may **respond to surrounding units within the coverage**. CNN excels at image processing. It includes a **convolutional layer**, a **pooling layer**, and a **fully connected layer**.
- ◆ In the 1960s, Hubel and Wiesel studied the cat's cortex neurons used for local sensitivity and direction selection and found that their unique network structure could simplify feedback neural networks. They then proposed the CNN.
- ◆ Now, CNN has become one of the research hotspots in many scientific fields, especially in the pattern classification field. The network is widely used because it can avoid complex pre-processing of images and directly input original images.

Convolutional Neural Network (2)



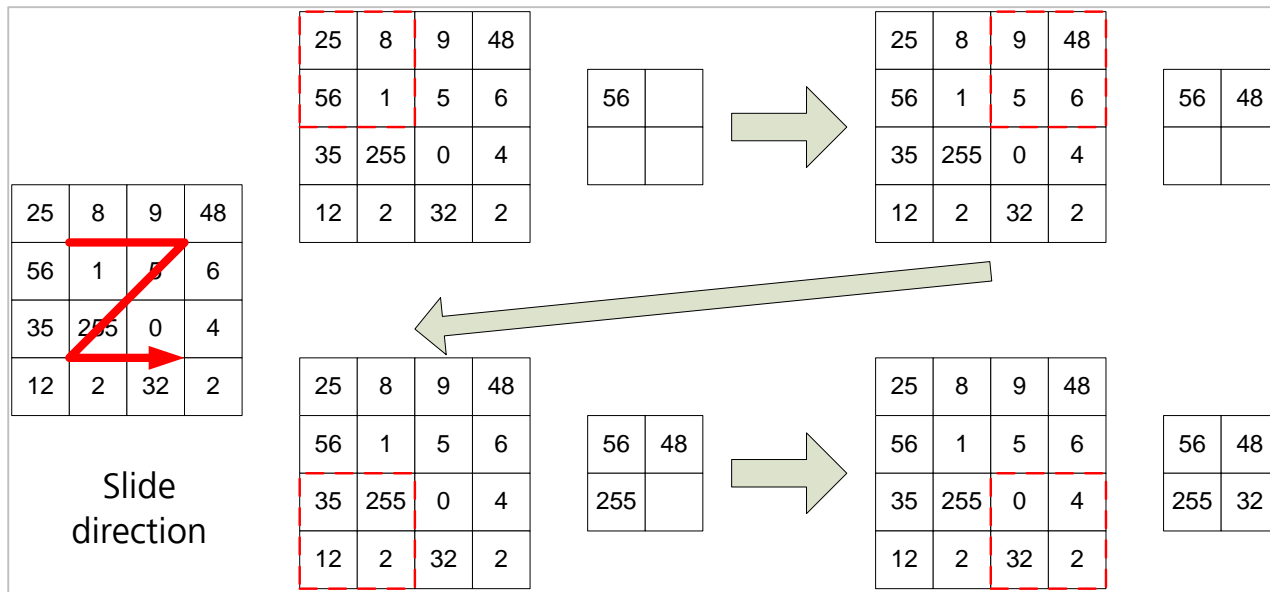
Convolutional Layer

- ◆ The basic structure of the convolutional neural network is the convolutional layer, namely multi-channel convolution that has been mentioned before. The output of the previous layer (or the original image of the first layer) is used as the input of the current layer. It is then convolved with the convolution kernel of the layer and serves as the output of this layer. The convolution kernel of each layer is the weight to be learnt. Similar to FCN, after the convolution is complete, the result should be biased and activated through activation functions before being input to the next layer.



Pooling Layer

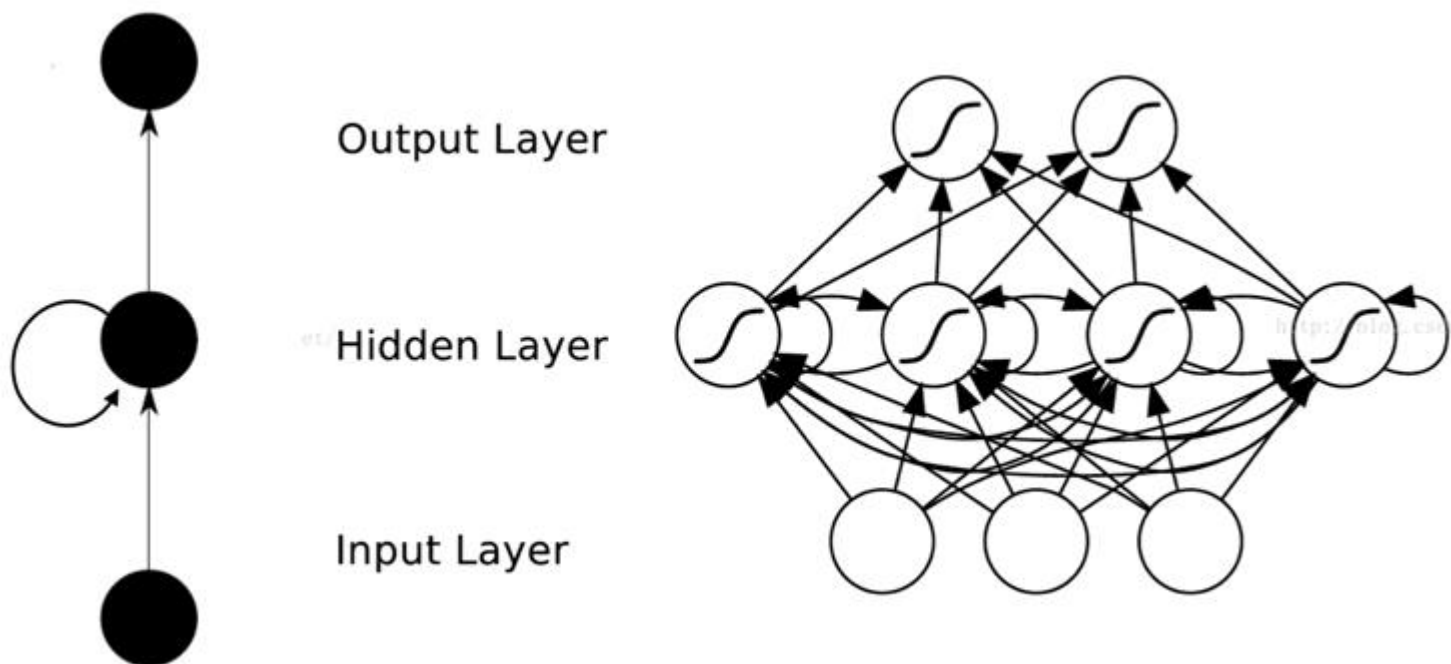
- ◆ Pooling combines the nearby units to reduce the size of the input for the next layer. It includes max pooling and average pooling. The former selects the maximum value in a small square area as the representative of this area while the latter uses the mean. The side of this small area is the pool window size. The following figure shows the max pooling operation whose pooling window size is 2.



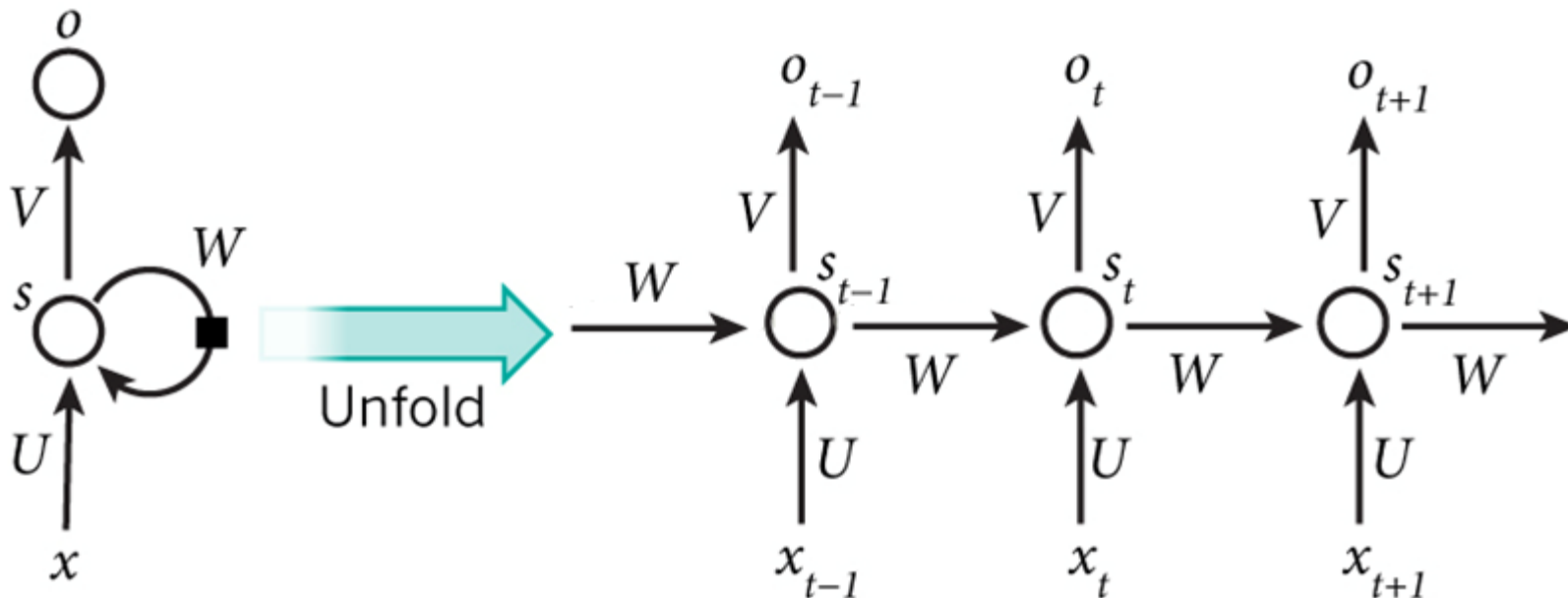
Recurrent Neural Network (1)

- ◆ The recurrent neural network (RNN) is a neural network that captures dynamic information in sequential data through periodical connections of hidden layer nodes. It can classify sequential data.
- ◆ Unlike other forward neural networks, the RNN can keep a context state and even store, learn, and express related information in context windows of any length. Different from traditional neural networks, it extends in space and time sequences. In other words, the hidden layers of the current moment and the next moment are related.
- ◆ The RNN is widely used in scenarios related to sequences, such as videos consisting of image frames, audio consisting of clips, and sentences consisting of words.

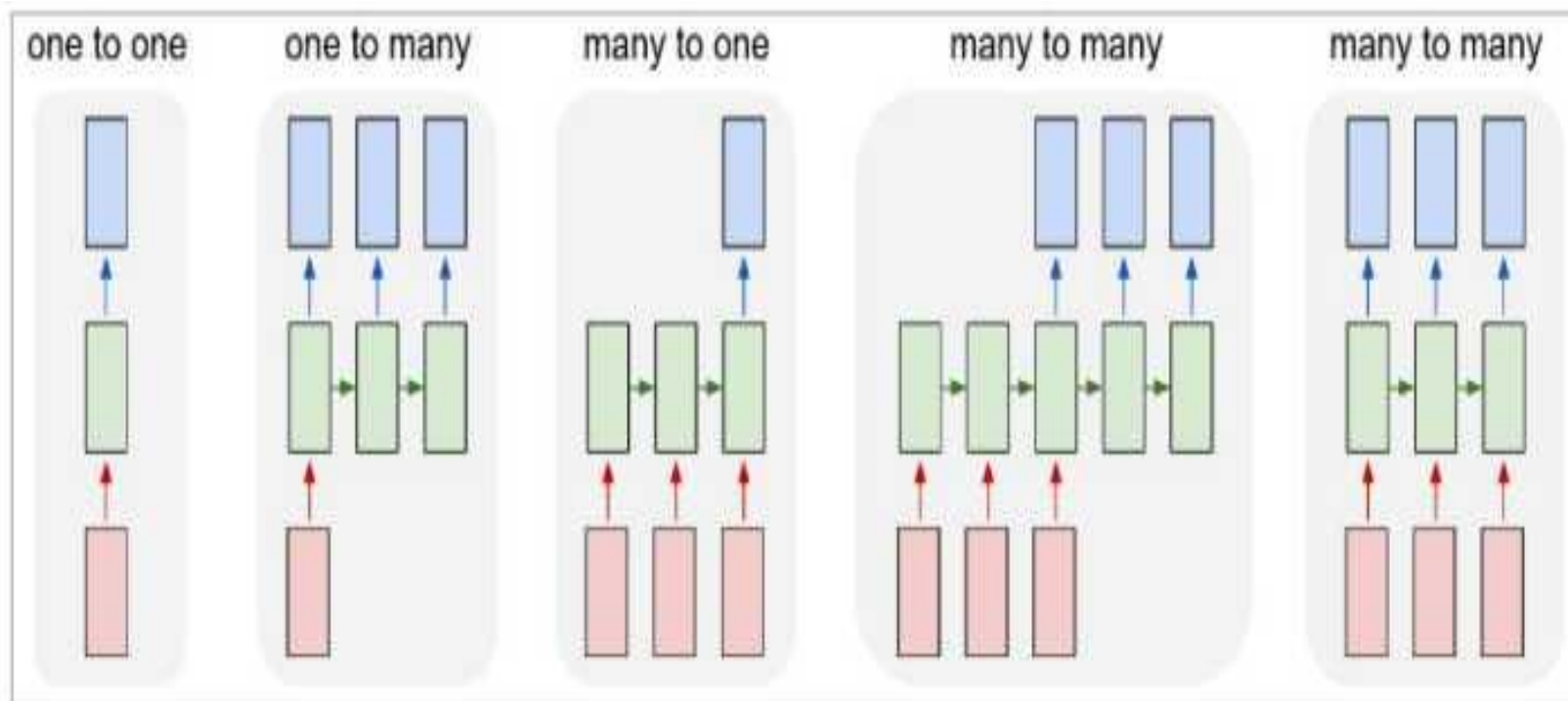
Recurrent Neural Network (2)



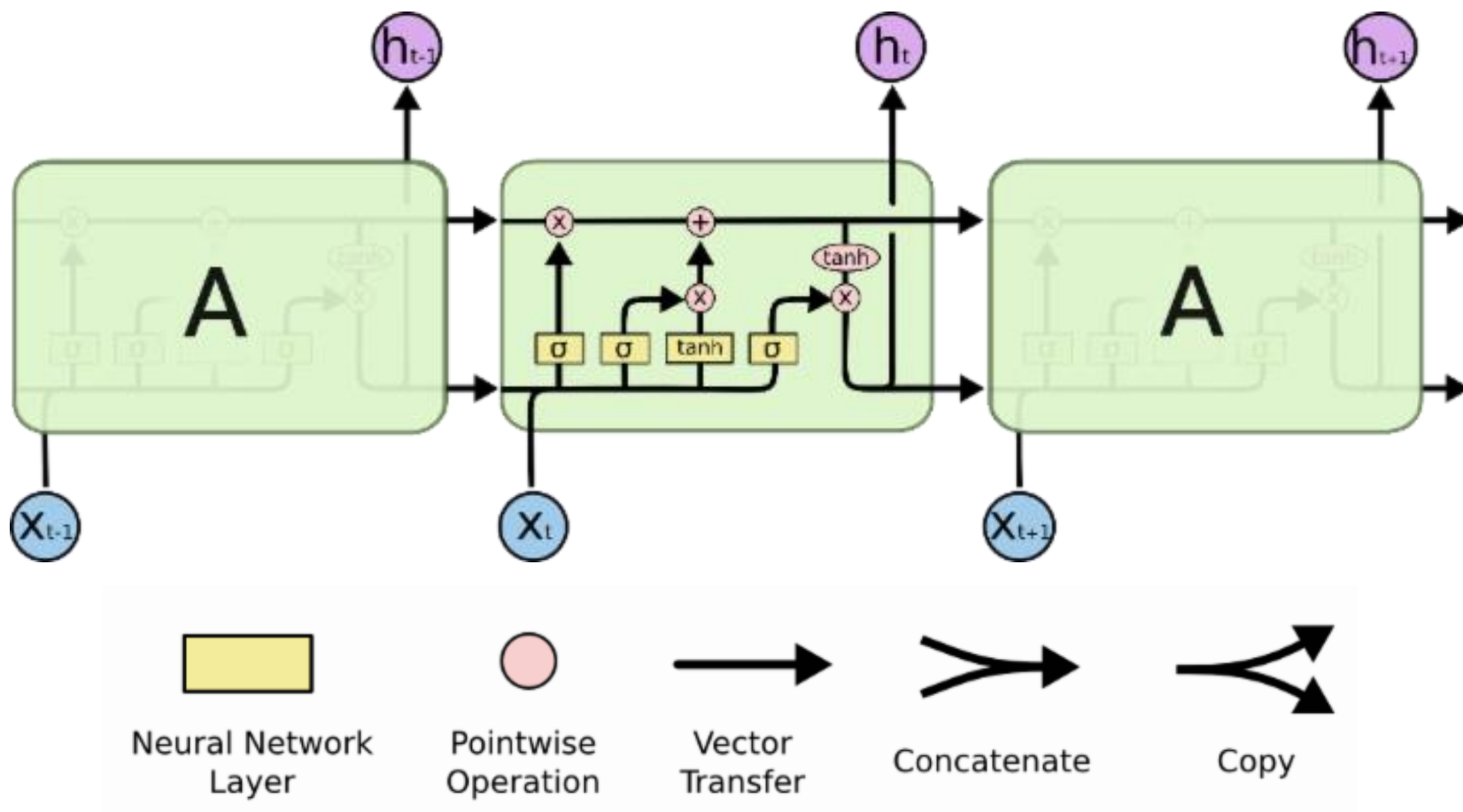
Unfolded Recurrent Neural Network



Types of Recurrent Neural Networks



Standard LSTM Structure





Contents

1. Propaedeutics of Deep Learning

2. Deep Learning Overview

- Definition and Development of Neural Networks
- Perceptron and Training Rules
- Activation Functions
- Types of Neural Networks
- **Regularization in Deep Learning**
- Optimizer
- Applications of Deep Learning

Regularization in Deep Learning

- ◆ Regularization is a very important and effective technology to reduce generalization errors in machine learning. It is especially useful for deep learning models which tend to have overfitting due to diverse parameters. Researchers have also proposed many effective technologies to prevent overfitting, including:
 - Adding constraints to parameters, such as L_1 and L_2 norms
 - Expanding the training set, such as adding noise and transforming data
 - Dropout

Parameter Penalties

- ◆ Many regularization approaches add a parameter penalty $\Omega(\theta)$ to the objective function J , limiting the learning capability of models. We denote the regularized objective function as \tilde{J} :

$$\tilde{J}(\theta; X, y) = J(\theta; X, y) + \alpha\Omega(\theta),$$

where $\alpha \in [0, \infty)$ is a hyperparameter that weights the relative contribution of the norm penalty term, Ω , relative to the standard objective function $J(X; \theta)$. Setting α to 0 results in no regularization. Larger values of α correspond to more regularization.

L_2 Regularization

- ◆ Add L_2 parameter norm penalty to prevent overfitting.

$$\tilde{J}(w; X, y) = J(w; X, y) + \frac{1}{2} \alpha \|w\|^2,$$

The parameter optimization method can be inferred using the optimization technology (such as gradient correlation method):

$$w = (1 - \varepsilon \alpha) \omega - \varepsilon \nabla J(w),$$

where ε is the learning rate. This multiplies parameters by a reduction factor compared with the common gradient optimization function.

L_1 Regularization

- ◆ Add L_1 norm constraint to model parameters:

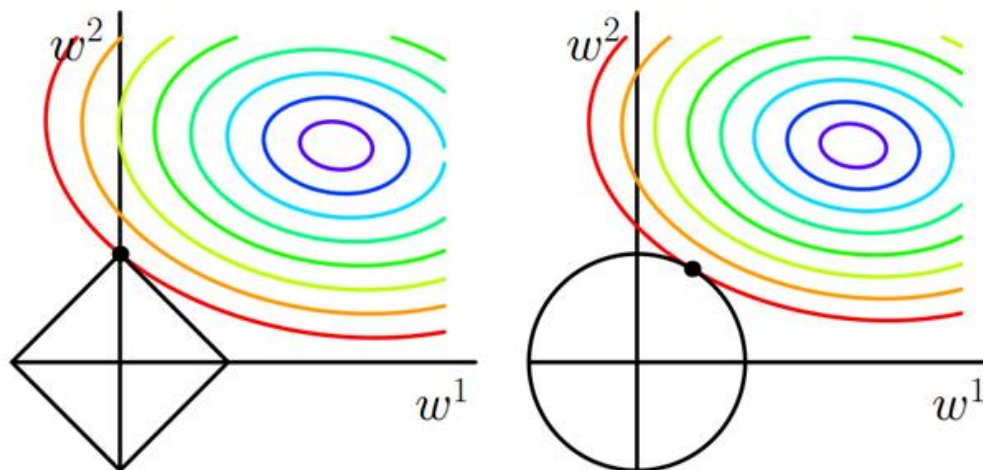
$$\tilde{J}(w; X, y) = J(w; X, y) + \alpha \|w\|_1,$$

- ◆ If the gradient method is used to solve the problem, the parameter gradient is

$$\nabla \tilde{J}(w) = \alpha \operatorname{sign}(w) + \nabla J(w).$$

L_2 VS L_1

- ♦ Major differences between L_2 and L_1 :
 - According to the preceding analysis, L_1 can generate a more sparse model than L_2 . When the parameter w is small, L_1 regularization can directly reduce the parameter to 0, which can be used for feature selection.
 - From the perspective of probability, many norm constraints are equivalent to adding prior distribution to parameters. L_2 norm equals to Gaussian distribution while L_1 norm equals to Laplace distribution.

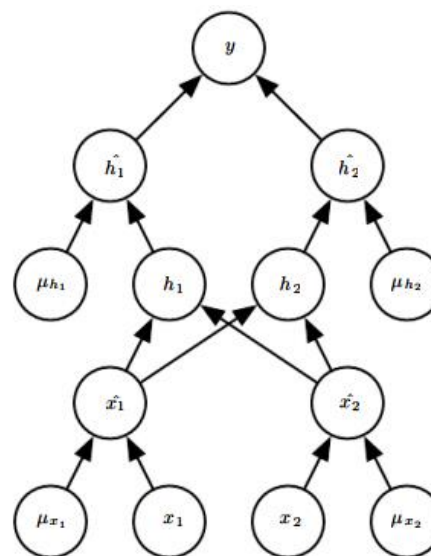
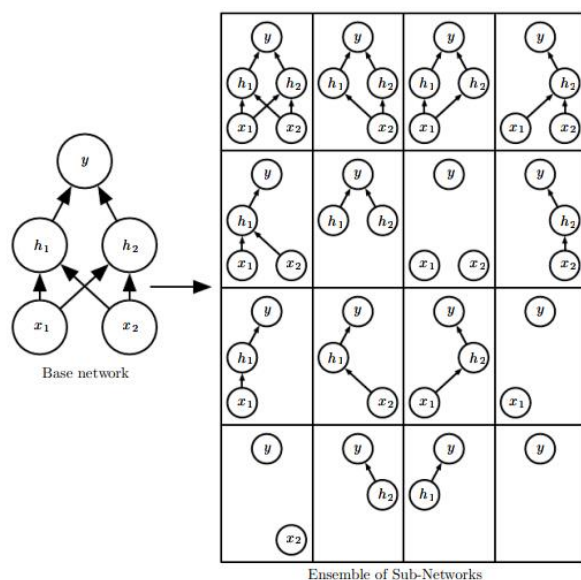


Dataset Expansion

- ◆ The most effective way to prevent overfitting is to add a training set. A larger training set has a smaller overfitting probability. Dataset expansion is a time-saving method, but it varies in different fields.
 - A common method in the object recognition field is to rotate or scale images. (The prerequisite to image transformation is that the type of the image cannot be changed through transformation. For example, for handwriting digit recognition, category 6 and 9 can be easily changed after rotation).
 - Random noise is added to the input data in speech recognition.
 - The common idea of NLP is to replace words with their synonyms.
 - Noise injection can add noise to the input or to the hidden layer or output layer. For example, for the softmax classification problem, noise can be added by using the label smoothing technology. If noise is added to the 0-1 category, the corresponding probability is changed to $\frac{\varepsilon}{k}$ and $1 - \frac{k-1}{k} \varepsilon$.

Dropout

- ◆ Dropout is a common and simple regularization method, which has been widely used since 2014. Simply put, dropout randomly discards some inputs during the training process. In this case, the parameters corresponding to the discarded inputs are not updated. Dropout is an integration method. It combines all sub-network results and obtains sub-networks by randomly dropping inputs. For example:





Contents

1. Propaedeutics of Deep Learning

2. Deep Learning Overview

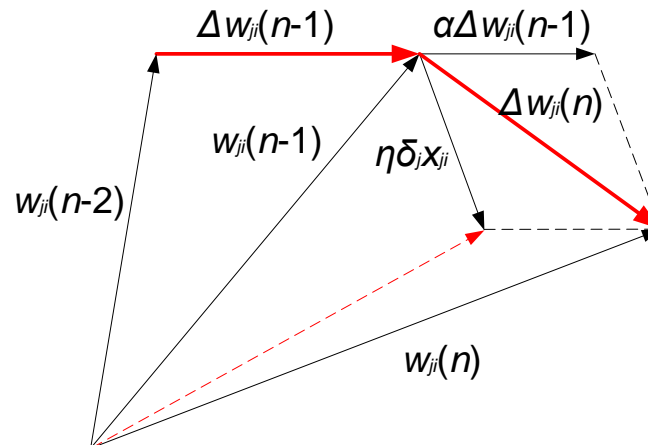
- Definition and Development of Neural Networks
- Perceptron and Training Rules
- Activation Functions
- Types of Neural Networks
- Regularization in Deep Learning
- **Optimizer**
- Applications of Deep Learning

Optimizer

- ◆ There are various improved versions of gradient descent algorithms. In object-oriented language implementation, different gradient descent algorithms are often encapsulated into an object which is called an **optimizer**.
- ◆ The **purpose** of algorithm improvement includes but is not limited to:
 - Accelerates algorithm convergence.
 - Avoids or overshoots local extrema.
 - Simplifies manual parameter setting, especially the learning rate.
- ◆ Common optimizers: common GD optimizer, momentum optimizer, Nesterov, **Adagrad**, **Adadelta**, **RMSprop**, Adam, AdaMax, and Nadam

Momentum Optimizer

- ◆ One basic improvement is to add the momentum term for Δw_{ji} . Denote the weight correction of n -th iteration as $\Delta w_{ji}(n)$ and then the weight correction rule is:
- ◆ $\Delta w_{ji}(n) = \eta \delta_j x_{ji} + \alpha \Delta w_{ji}(n-1)$
where α is a constant ($0 \leq \alpha < 1$) and called momentum and $\alpha \Delta w_{ji}(n-1)$ is a momentum term.
- ◆ Imagine a small ball rolls down the error surface starting from a random point. The introduction of the momentum term is equivalent to giving the small ball inertia.



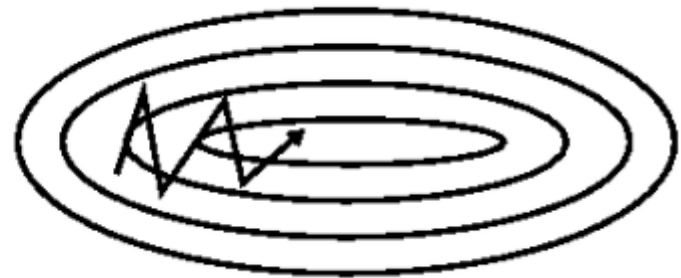
Advantages and Disadvantages of Momentum Optimizer

◆ Advantages:

- Enhances the stability of the gradient correction direction and reduces mutations.
- In areas where the gradient direction is stable, the ball rolls faster and faster (there is a speed upper limit because $\alpha < 1$), which helps the ball quickly overshoot the flat area and accelerates convergence.
- A small ball with inertia is more likely to roll over some narrow local extrema.

◆ Disadvantage:

- The learning rate η and momentum α need to be manually set, which often requires more experiments to determine the appropriate value.



Adam Optimizer (1)

- ◆ Adaptive Moment Estimation (Adam): Developed based on Adagrad and Adadelta, Adam maintains two additional variables m_t and v_t for each variable to be trained:

$$m_t = \beta_1 m_{t-1} + (1 - \beta_1) g_t$$

$$v_t = \beta_2 v_{t-1} + (1 - \beta_2) g_t^2$$

where t represents t -th iteration and g_t is the calculated gradient. m_t and v_t are moving averages of the gradient and the squared gradient. From the statistical perspective, m_t and v_t are estimates of the first moment (the mean) and the second moment (the uncentered variance) of the gradients respectively, hence the name of the method.

Adam Optimizer (2)

- ◆ As m_t and v_t are initialized as vectors of 0's, they are biased towards 0, especially during the initial steps, and especially when β_1 and β_2 are close to 1. To solve this problem, we use \hat{m}_t and \hat{v}_t :

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}$$
$$\hat{v}_t = \frac{v_t}{1 - \beta_2^t}$$

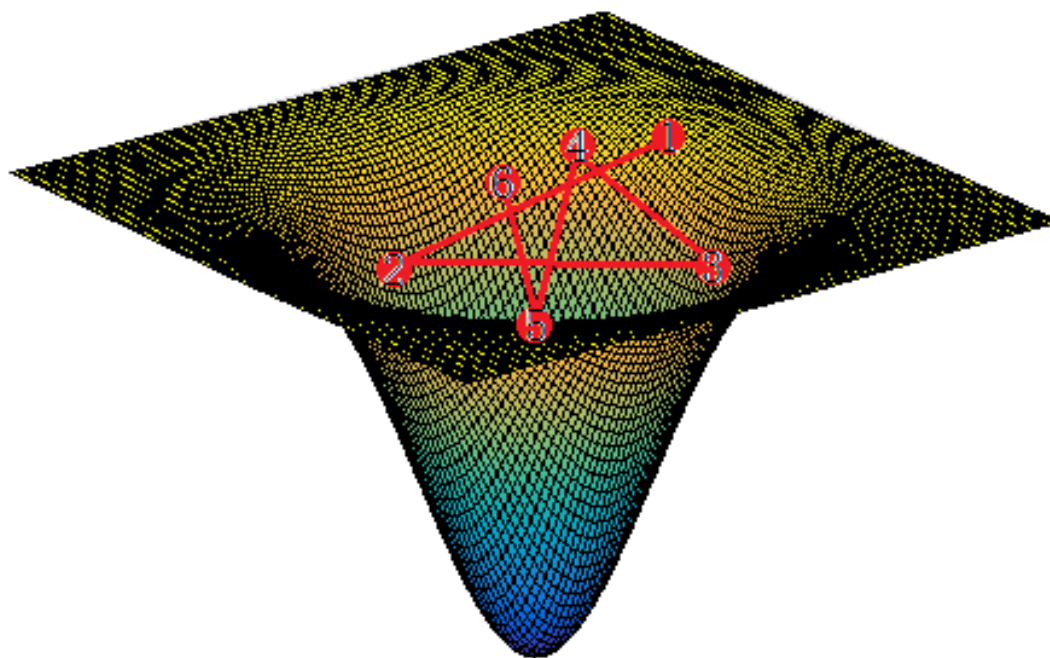
- ◆ Adam's weight update rule:

$$w_{t+1} = w_t - \frac{\eta}{\sqrt{\hat{v}_t} + \epsilon} \hat{m}_t$$

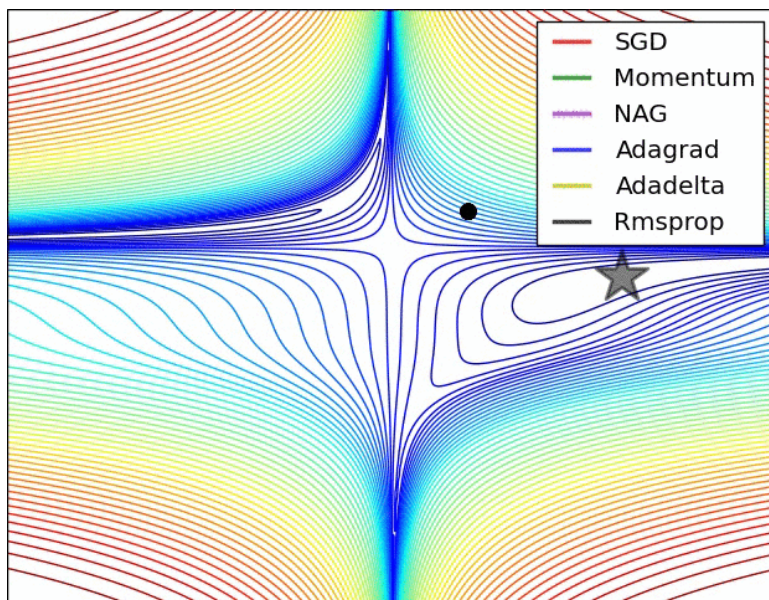
- ◆ Although the rule involves manual setting of η , β_1 , and β_2 , this is much simpler. According to experiments, the default setting are $\beta_1 = 0.9$, $\beta_2 = 0.999$, $\epsilon = 10^{-8}$, and $\eta = 0.001$. In practice, Adam will converge quickly. For convergence saturation, reduce η . After several times of reduction, the satisfying local extrema will be converged. Other parameters do not need adjustment.

Adam Optimizer (3)

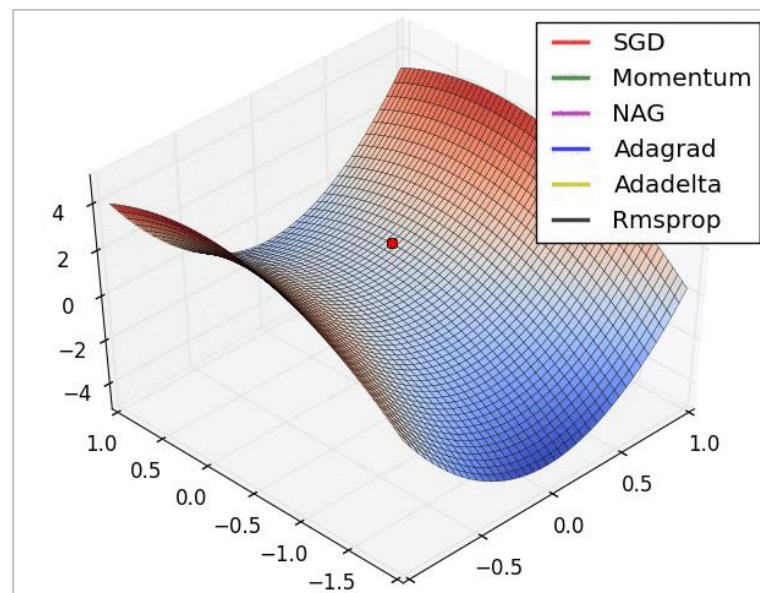
- ◆ \hat{m}_t is the current gradient with a momentum term and $\frac{\eta}{\sqrt{\hat{v}_t + \epsilon}}$ is equivalent to the current learning rate. If the gradient model is too large, the weight cannot jump out of the extrema or converge. Then we need to reduce the learning rate to facilitate convergence.



Visualized Comparison of Optimizers



Comparison of optimization algorithms in contour maps of loss functions



Comparison of optimization algorithms at the saddle point



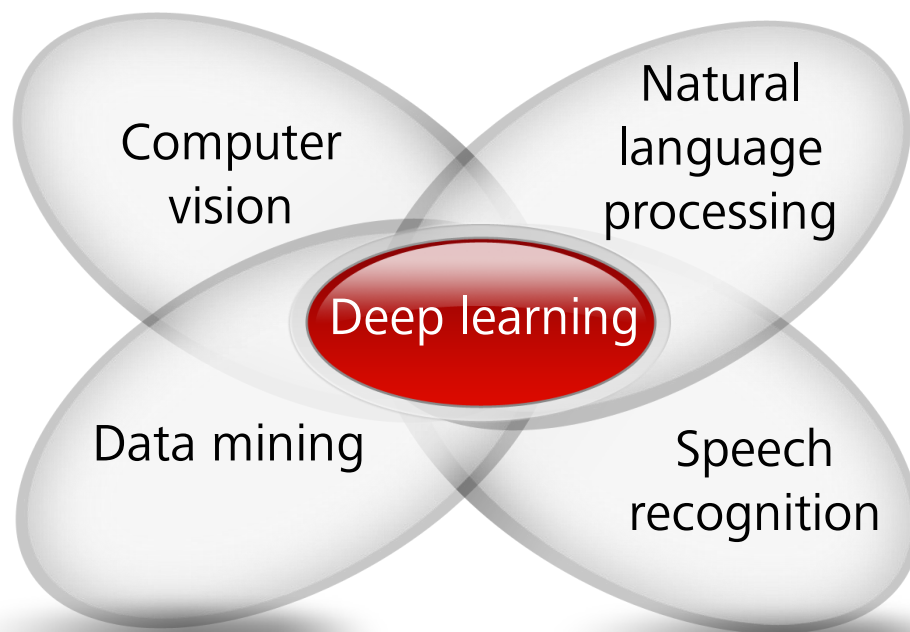
Contents

1. Propaedeutics of Deep Learning

2. Deep Learning Overview

- Definition and Development of Neural Networks
- Perceptron and Training Rules
- Activation Functions
- Types of Neural Networks
- Regularization in Deep Learning
- Optimizer
- **Applications of Deep Learning**

Applications of Deep Learning



Computer vision

- ◆ Image classification
- ◆ Scene recognition
- ◆ Face detection and recognition
- ◆ Object detection and recognition
- ◆ Semantic segmentation
- ◆ Stylization
- ◆ Super-resolution
- ◆ OCR

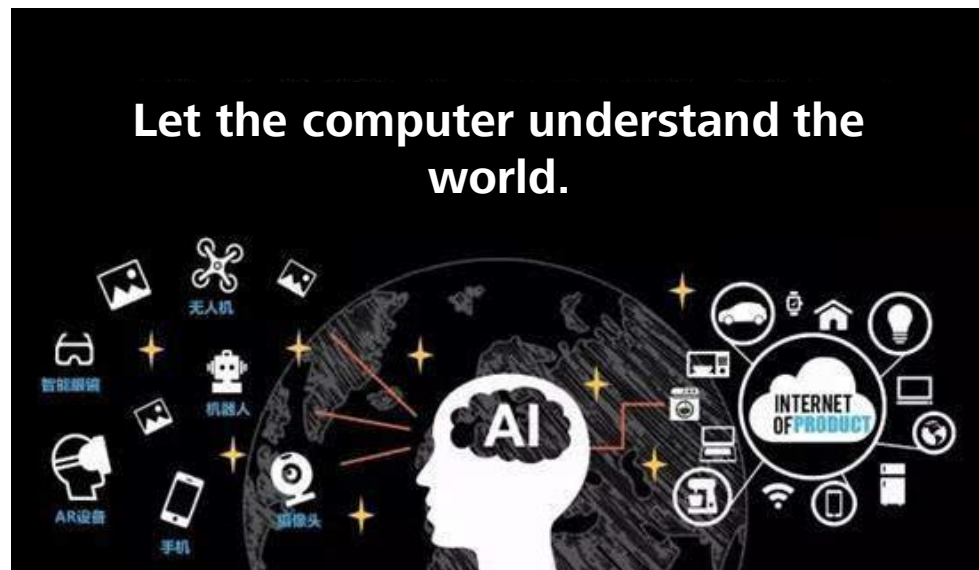


Image Classification

- ◆ Maps images to different class sets, which can be used for image retrieval and image archiving.



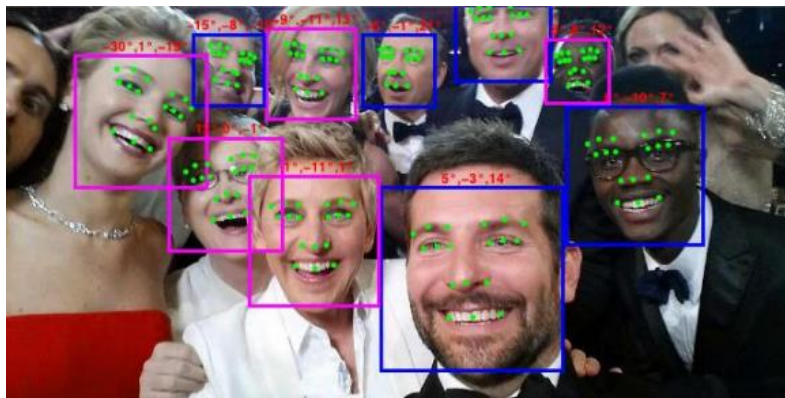
Scene Recognition

- ◆ Classifies images based on scenes and environments, which can be used for situation awareness and intelligent 3A for photographing.

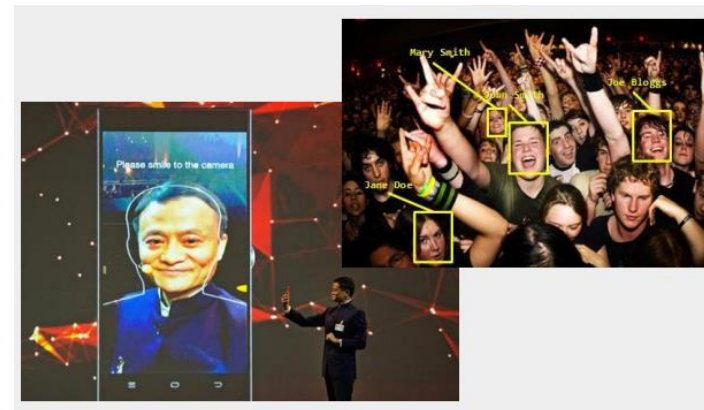


Face Detection and Recognition

- ◆ Face detection: discovers and locates faces and facial features in images, which can be used for face focusing, polishing up, and augmented reality.
- ◆ Facial recognition: differentiates people, which can be used for identity authentication and privacy protection.



Face detection



Facial recognition

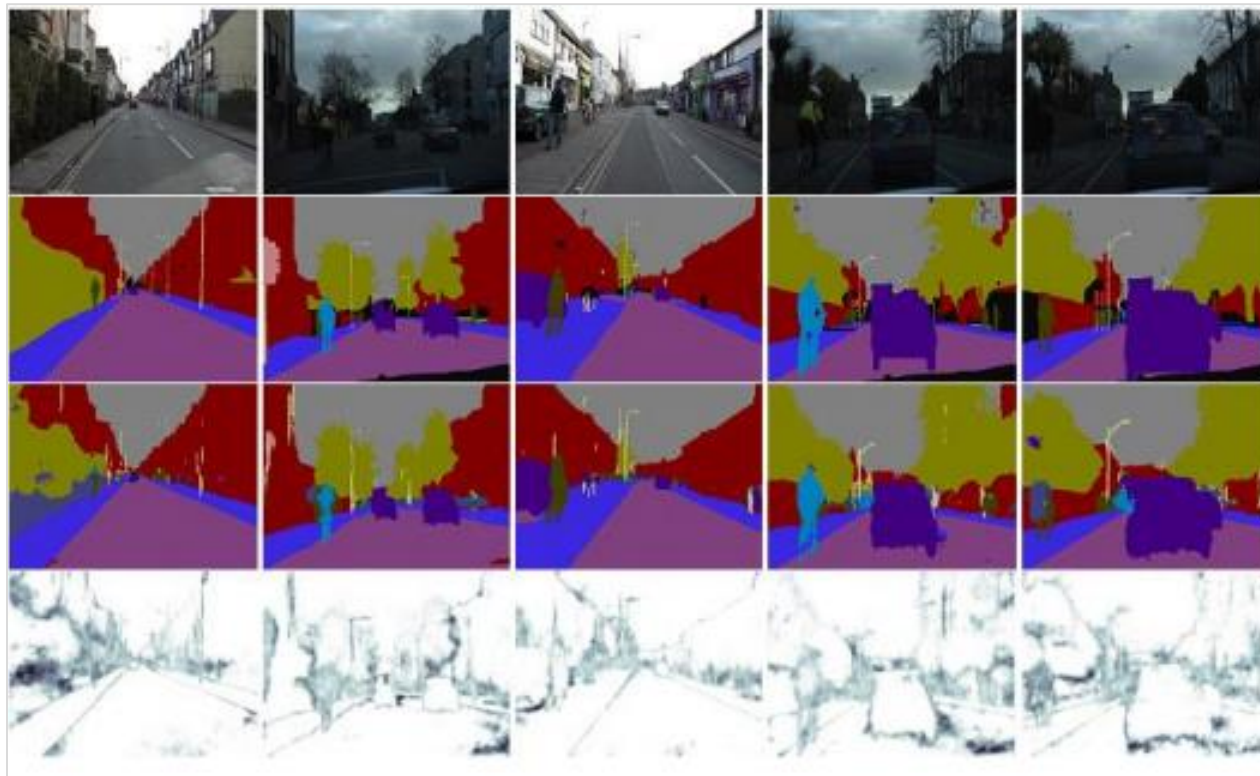
Object Detection and Recognition

- ◆ Detects, locates, and identifies different objects in images, including digital, text, and pedestrian detection. This function can be used for OCR, unmanned driving, and intelligent image tailoring.



Semantic Segmentation

- ◆ Predicts the label of each pixel in the image, including segmentation and recognition. It can be used for unmanned driving, augmented reality, and situation awareness.



Stylization

- ◆ Changes the image style while retaining the content of the image, which can be used for stylized image processing.



Super-resolution

- Generates high-resolution images from low-resolution images, which can be used for image processing, security surveillance, and medical imaging.



OCR

- ◆ Identifies information such as numbers and texts in images to digitize images or paper documents, which can be used to automatically identify business cards or scan invoices.



Natural Language Processing

- ◆ Chinese word segmentation
- ◆ Knowledge exploration
- ◆ Machine translation
- ◆ Sentiment analysis



Chinese Word Segmentation Algorithms

Chinese word
segmentation algorithms

1 Dictionary-based: dictionary and lexicon matching

Character string match and mechanical segmentation
Different scanning directions: forward matching and reverse matching;
different lengths: maximum matching and minimum matching

1. Forward maximum matching (MM)
2. Reverse maximum matching (RMM)
3. Minimum segmentation

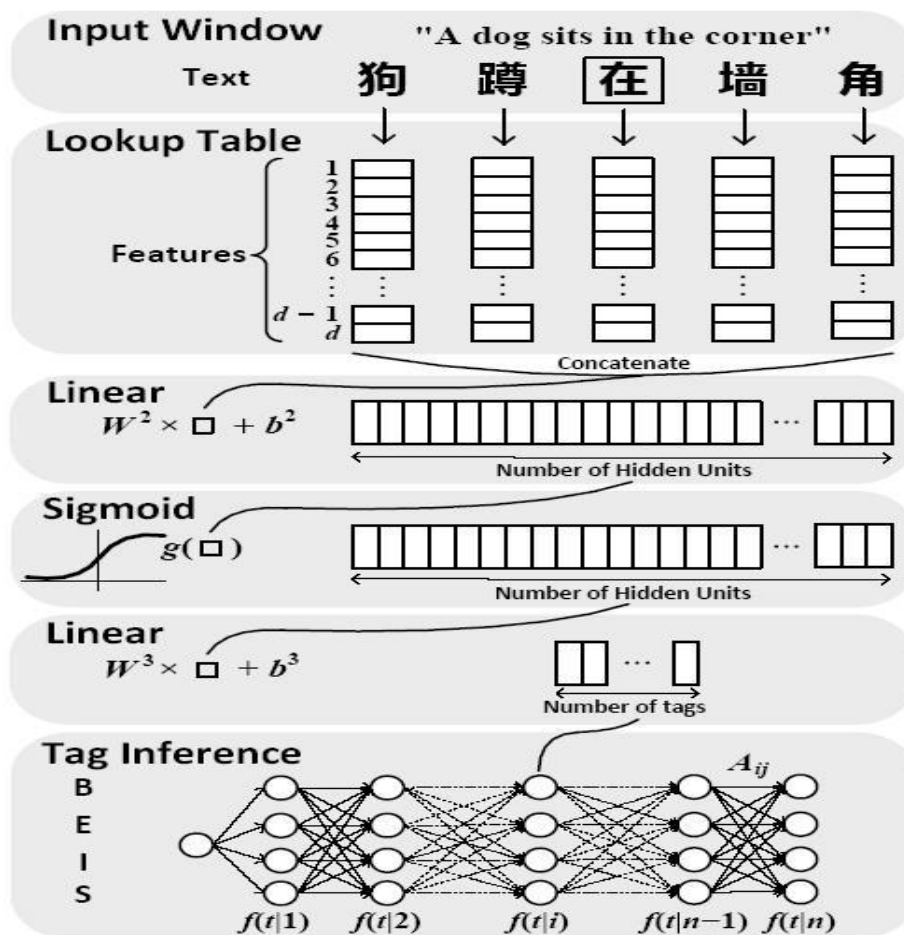
2 Statistics-based: word frequency statistics

Non-dictionary segmentation and full segmentation
Major statistical models: n-gram model and hidden Markov model (HMM)

3 Rule-based: knowledge

Based on syntactic and grammatical analysis and semantic analysis
Three parts: segmentation subsystem, syntax and grammar subsystem,
and general control

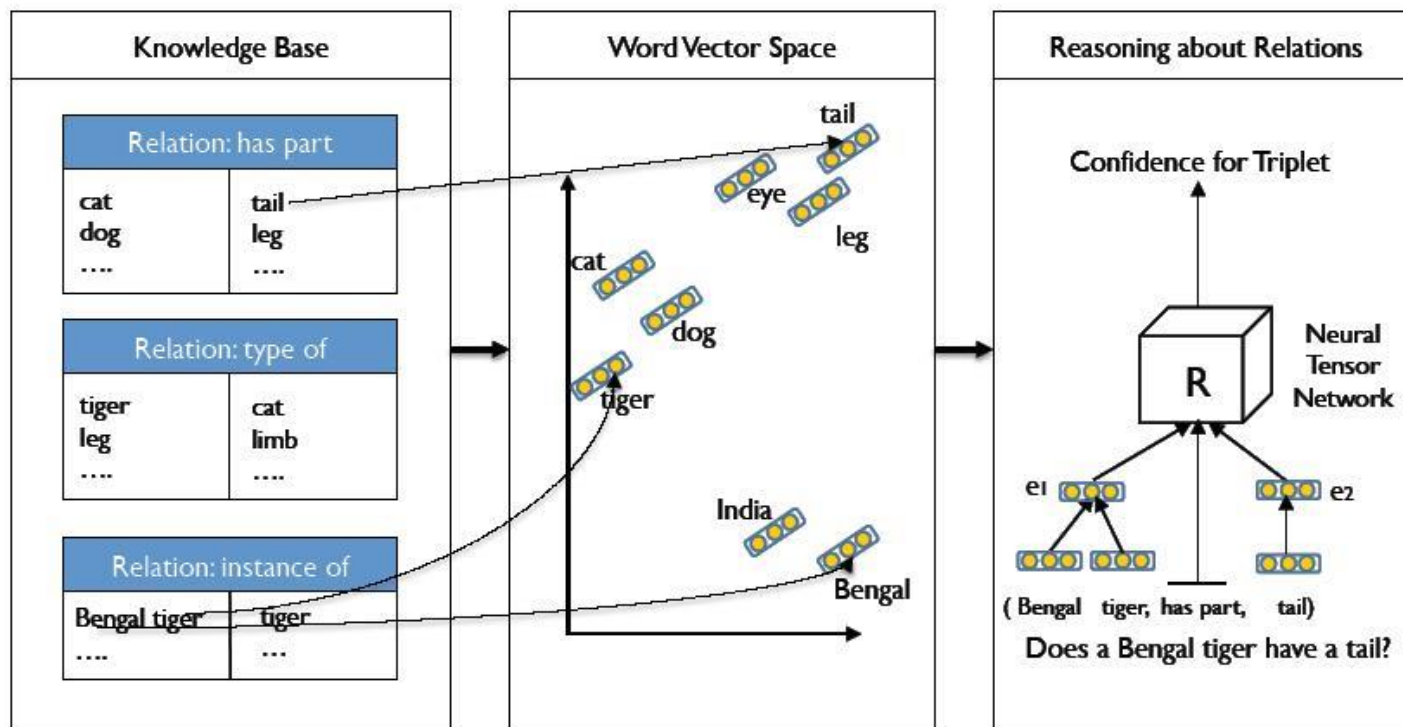
Chinese Word Segmentation



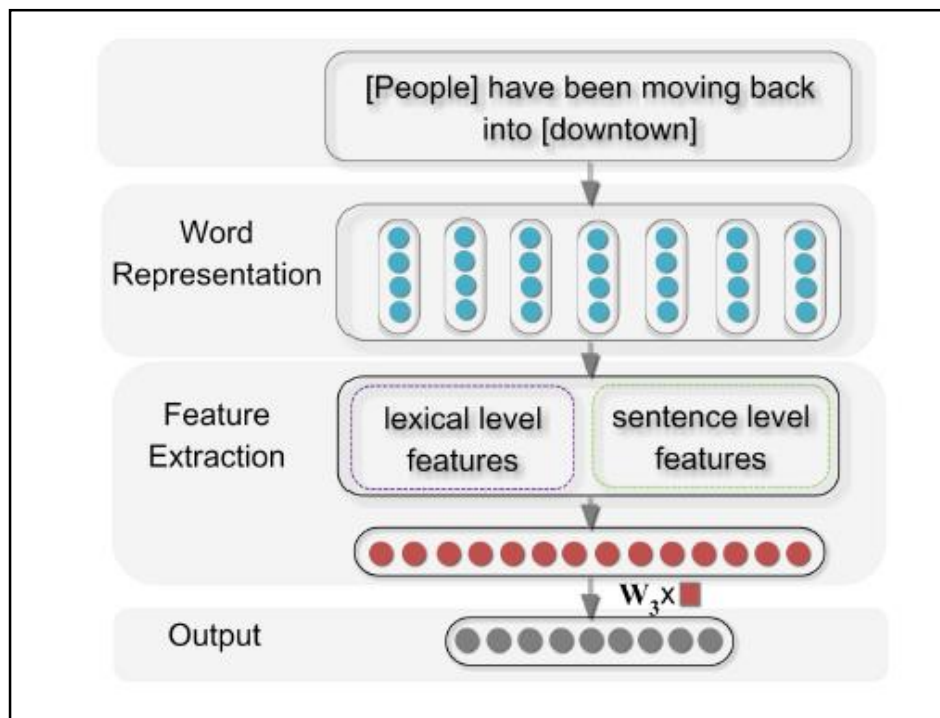
Knowledge Exploration

- ◆ Two types of problems
 - New knowledge reasoning in the existing knowledge base
 - ▣ CYC, WordNet, FreeNet, etc.
 - ▣ Current studies use similar approaches.
 - Known entities are represented by word embeddings.
 - The entity relationship is modeled using tensor networks.
 - Backpropagation + SGD training
 - ◆ Mining structured knowledge from free text
 - ▣ Text mining
 - ▣ TF-IDF

New Knowledge Reasoning in the Existing Knowledge Base



Mining Structured Knowledge from Free Text (1)

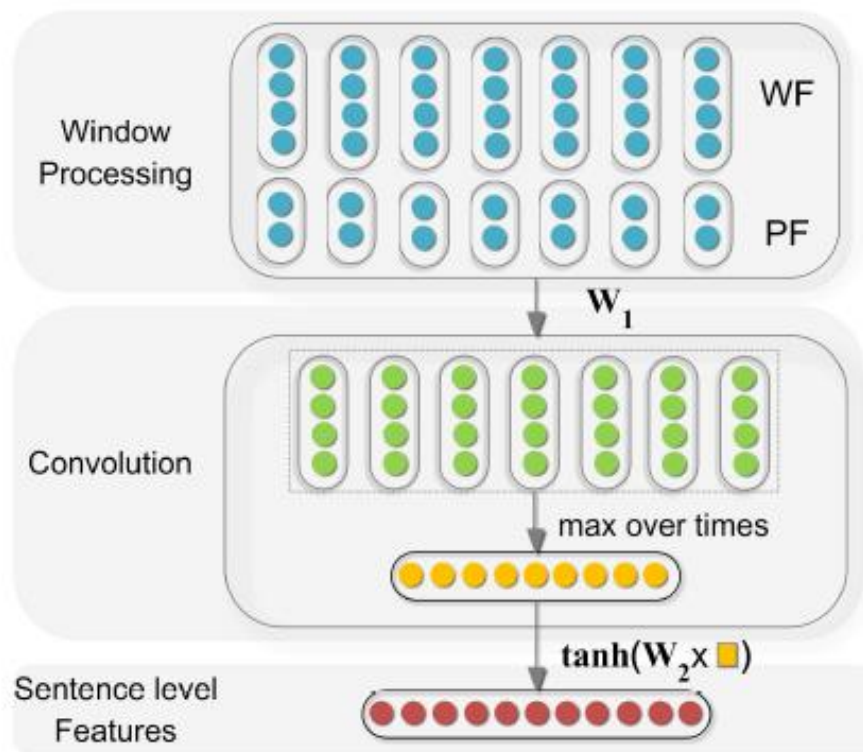


Overall structure

Features	Remark
L1	Noun 1
L2	Noun 2
L3	Left and right tokens of noun 1
L4	Left and right tokens of noun 2
L5	WordNet hypernyms of nouns

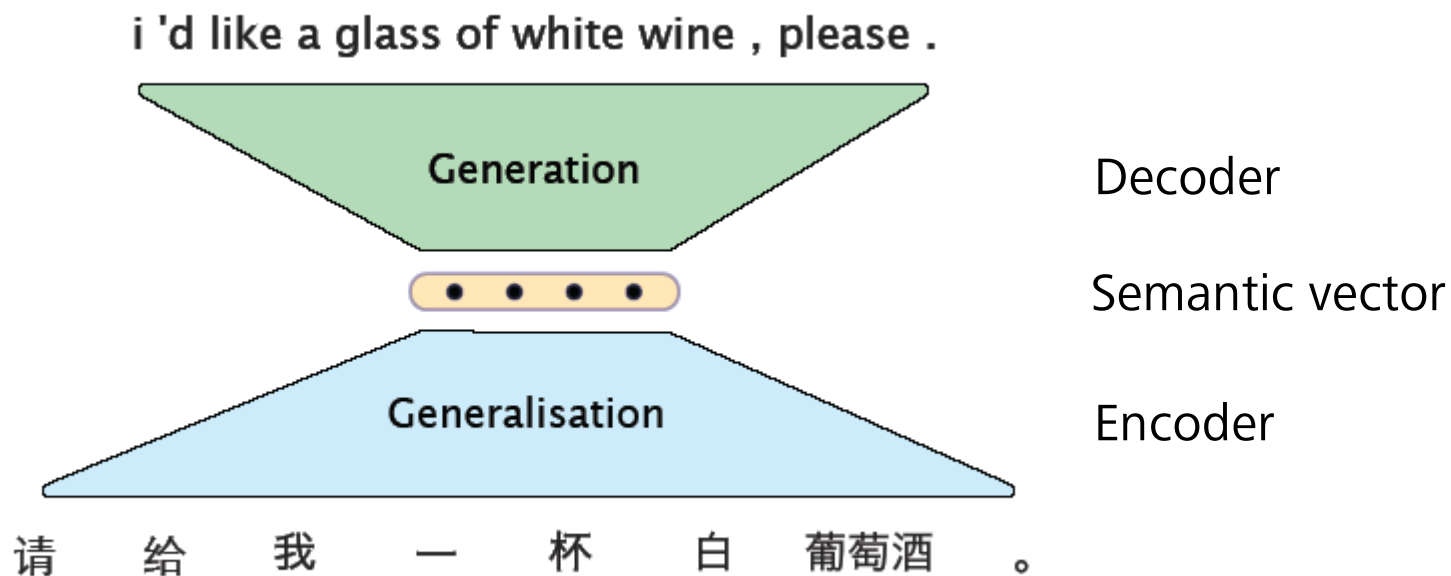
Lexical feature

Mining Structured Knowledge from Free Text (2)



Sentence-level feature extraction: convolutional network

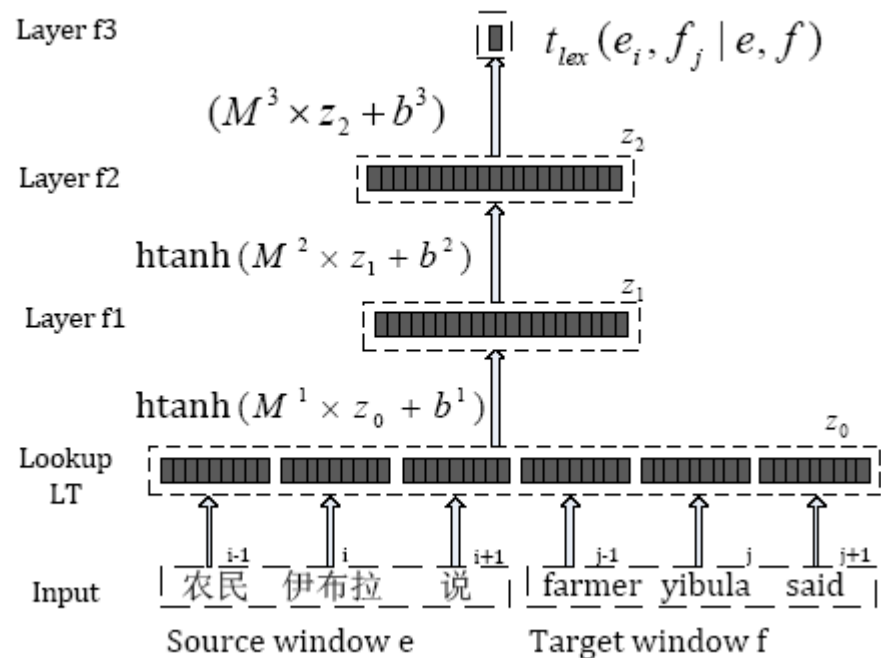
Machine Translation (Common Model)



The most common model: encoder-decoder model

Machine Translation: Deep Learning Applied to Multiple Areas

- ◆ Word alignment
- ◆ Phrase alignment
- ◆ Phrase reordering
- ◆ Language model
- ◆ Translation model
- ◆ Conjunctive model
- ◆ Translation result reordering
- ◆ ...



Sentiment Analysis

- ◆ Two core issues:
 - Sentence-level word embedding representation
 - How to encode emotional tendencies to word embeddings at all levels
 - Semi-supervised or supervised learning: The emotion tendency is encoded into the WE structure through the training process.

What is Sentiment Analysis?

A linguistic analysis technique that identifies opinion early in a piece of text.

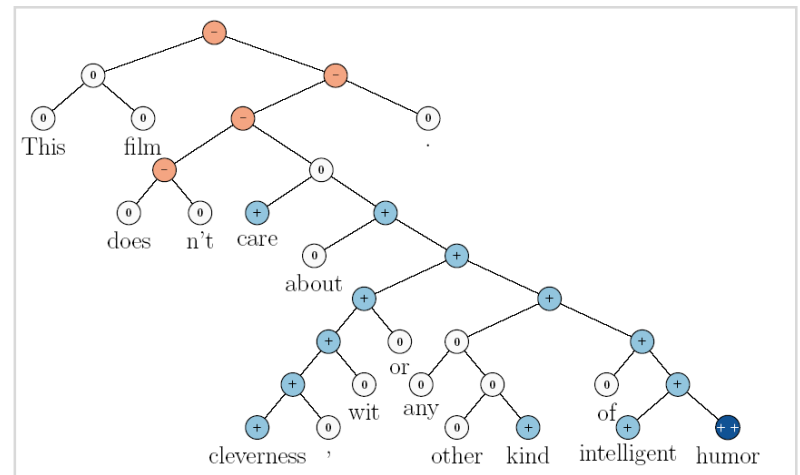
The movie is great.



The movie stars Mr. X



The movie is horrible.



Speech Recognition

Speech can have many roles in mobile apps



Recommendation System





Summary

- ◆ This chapter introduces the propaedeutics of deep learning, including learning algorithms, common machine learning algorithms, hyperparameters, verification sets, parameter estimation, and estimation methods. Then it illustrates the definition, types, and development of neural networks as well as perceptron and its training rules.



Quiz

1. (Multiple choice) Which of the following is not a deep learning development framework? ()

A. CNTK

B. Keras

C. BAFA

D. MXNet

2. (True or false) GAN is a deep learning model, which is one of the most promising methods for unsupervised learning of complex distribution in recent years. ()

A. True

B. False

Quiz

3. (Multiple answer question) Training errors reduce the model accuracy and generate underfitting. How can we improve the model fitting? ()
- A. Increasing data volume
 - B. Feature engineering
 - C. Reducing regularization parameters
 - D. Adding features
4. (True or false) Compared with the recurrent neural network, the convolutional neural network is more suitable for image recognition. ()
- A. True
 - B. False



Recommendations

- ◆ Huawei E-learning website:
 - <http://support.huawei.com/learning/Index!toTrainIndex>
- ◆ Huawei Support case library:
 - <http://support.huawei.com/enterprise/servicecenter?lang=zh>

Thank You

www.huawei.com