人工智能
——人工神经网络 II

Fear Surprise Sadness Anger Disgust Joy

饶洋辉
数据科学与计算机学院,
中山大学
raoyangh@mail.sysu.edu.cn

Neural Network

- The study of artificial neural networks was inspired by attempts to simulate biological neural systems.

神经

- The human brain consists of nerve cells called neurons (神经元) primarily .
- Neurons are linked together via strands of fiber called axons (轴突).

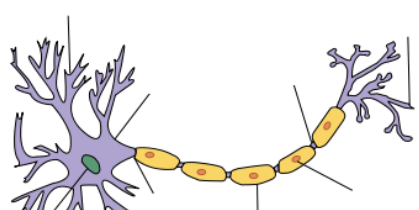
冲动

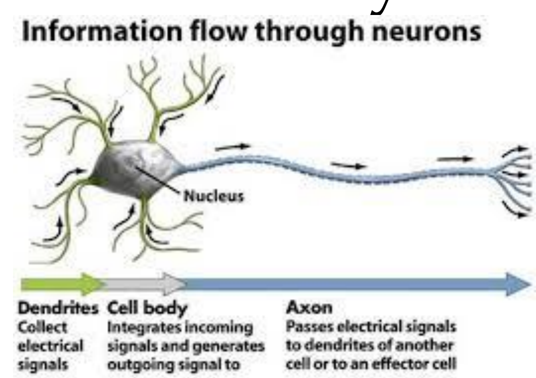
- Axons are used to transmit nerve impulses from one neuron to another whenever the neurons are stimulated.

刺激

Neural Network

- A neuron is connected to the axons of other neurons via dendrites (树突), which are extensions from the cell body of the neuron.
- The contact point between a dendrite and an axon is called a synapse (突触).
- The human brain learns by changing the strength of the synaptic connection between neurons upon repeated stimulation by the same impulse.





Neural Network

- A neural network consists of a large number of simple and interacting nodes (artificial neurons).
相互作用
- Knowledge is represented by the strength of connections between these nodes.
获得 调整
- Knowledge is acquired by adjusting the connections through a process of learning.
处理
- All the neurons process their inputs simultaneously and independently.
同时地

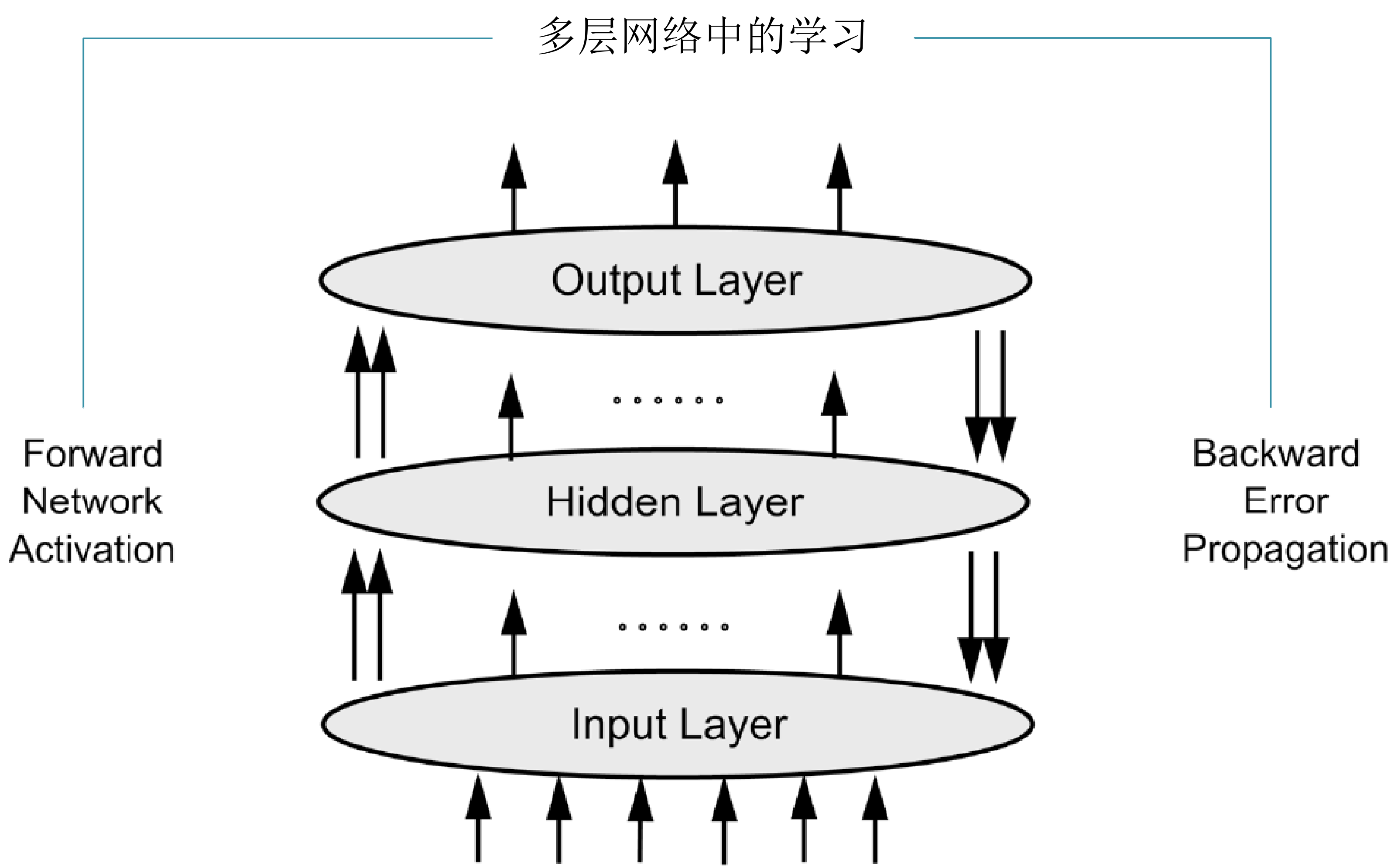
Neural Network

- The additional layers in between the input and output nodes are called hidden layers.
- 嵌入的
  The nodes embedded in these layers are called hidden nodes.
- We focus on feedforward neural networks, in which the nodes in one layer are connected 一层中的节点仅连接到下一层中的节点 only to the nodes in the next layer.
- 反向传播
  The **backpropagation** learning algorithm is specifically designed for neural networks with multiple layers.

Neural Network

- There are two phases in each iteration of the training algorithm
  阶段
  - The forward phase
  - The backward phase

Neural Network



多层前馈神经网络

Neural Network

在前向阶段，利用前一次迭代得到的权重计算每个神经元的输出值

- During the forward phase, the weights obtained from the previous iteration are used to compute the output value of each neuron.
- Outputs of the neurons at level l are computed prior to computing the outputs at level l+1.

在计算L+1层神经元的输出之前，先计算L层神经元的输出

Neural Network

在向后阶段，权重更新方程的应用方向相反

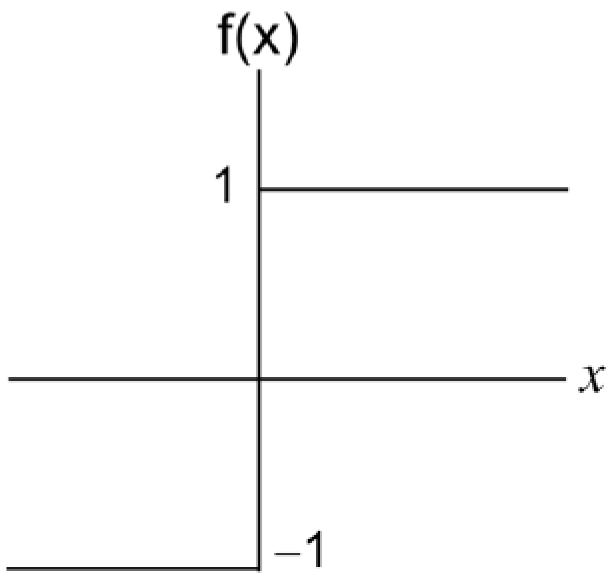
- During the backward phase, the weight update equation is applied in the reverse direction.

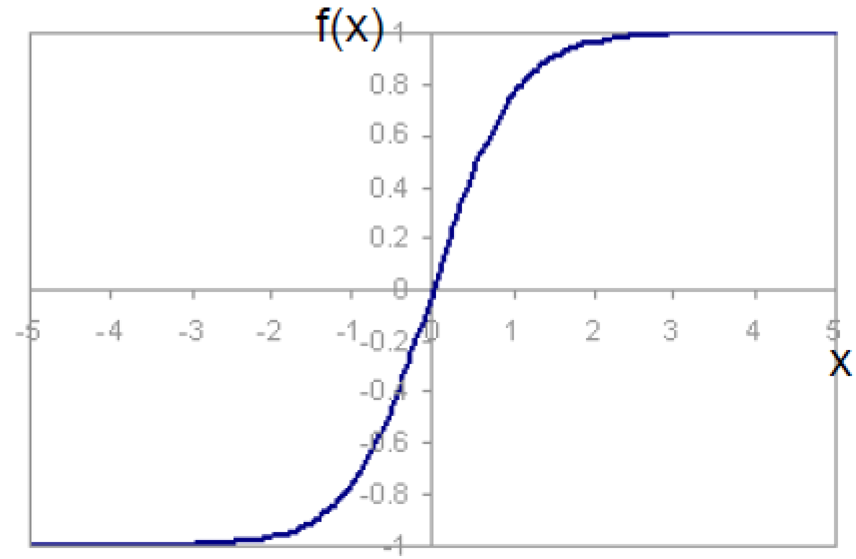
在更新L层的权重前更新L+1层的权重

- In other words, the weights at level l+1 are updated before the weights at level l are updated.
- The learning algorithm allows us to use the errors for neurons at layer l+1 to estimate the errors for neurons at layer l. 使用L+1层神经元的误差来估计L层神经元的误差

Neural Network

- For this type of network, instead of the threshold function, another activation function is used.





Neural Network

- A common activation function is the hyperbolic tangent function

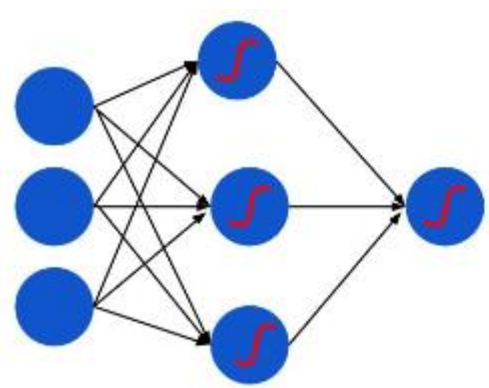
双曲正切函数

$$f(x) = \frac{e^{x} - e^{-x}}{e^{x} + e^{-x}} = \frac{e^{2x} - 1}{e^{2x} + 1}$$ 取值范围为[-1,1]

- An important property of the function is that it is differentiable

可微的

$$f'(x) = 1 - f(x)^2$$



Neural Network

- Another common activation function is the **sigmoid** function

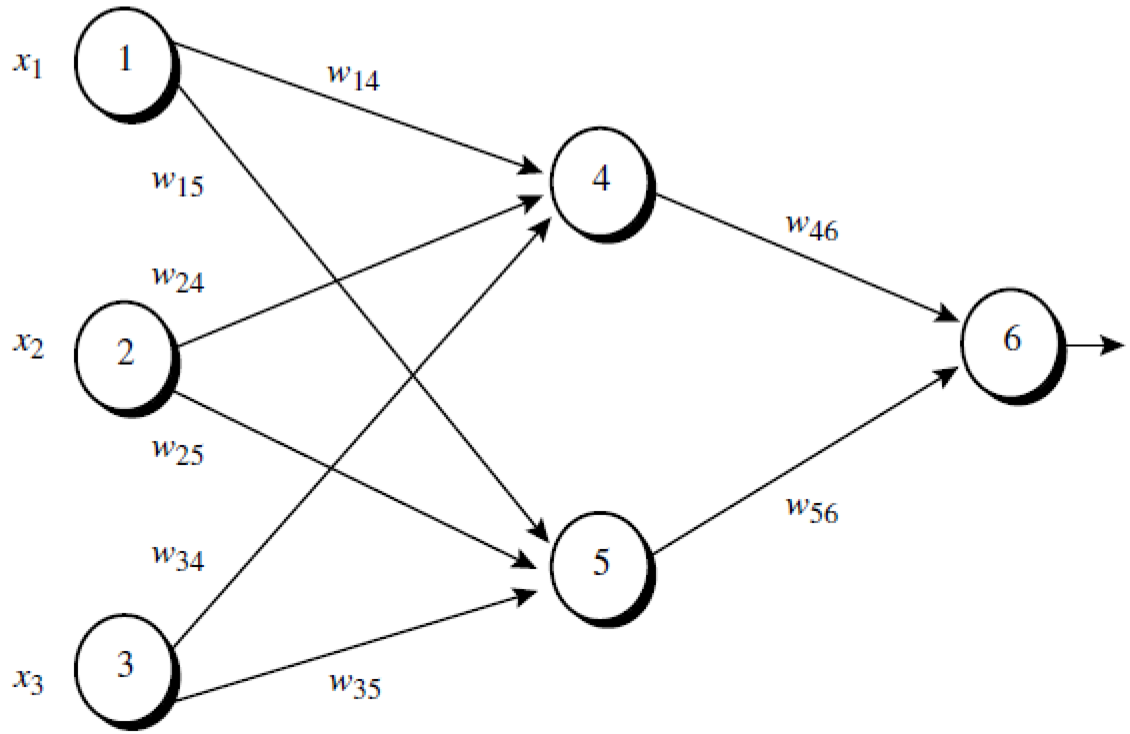
$$f(x) = \frac{1}{1+e^{-x}}$$

- This function is also differentiable

$$f'(x) = f(x)(1 - f(x))$$

Example

- One training tuple **X**=(1,0,1), whose class label is 1.



Example

偏移量

Initial input, weight, and bias values.

x_1	x_2	x_3	w_{14}	w_{15}	w_{24}	w_{25}	w_{34}	w_{35}	w_{46}	w_{56}	θ_4	θ_5	θ_6
1	0	1	0.2	−0.3	0.4	0.1	−0.5	0.2	−0.3	−0.2	−0.4	0.2	0.1

The net input and output calculations.

Unit j	*Net input, I_j*	*Output, O_j*
4	$0.2+0-0.5-0.4=-0.7$	$1/(1+e^{0.7})=0.332$
5	$-0.3+0+0.2+0.2=0.1$	$1/(1+e^{-0.1})=0.525$
6	$(-0.3)(0.332)-(0.2)(0.525)+0.1=-0.105$	$1/(1+e^{0.105})=0.474$

Example

Calculation of the error at each node.

Unit j	Err_j
6	$(0.474)(1-0.474)(1-0.474) = 0.1311$
5	$(0.525)(1-0.525)(0.1311)(-0.2) = -0.0065$
4	$(0.332)(1-0.332)(0.1311)(-0.3) = -0.0087$

Calculations for weight and bias updating.

Weight or bias	*New value*
w_{46}	$-0.3+(0.9)(0.1311)(0.332) = -0.261$
w_{56}	$-0.2+(0.9)(0.1311)(0.525) = -0.138$
w_{14}	$0.2+(0.9)(-0.0087)(1) = 0.192$
w_{15}	$-0.3+(0.9)(-0.0065)(1) = -0.306$
w_{24}	$0.4+(0.9)(-0.0087)(0) = 0.4$
w_{25}	$0.1+(0.9)(-0.0065)(0) = 0.1$
w_{34}	$-0.5+(0.9)(-0.0087)(1) = -0.508$
w_{35}	$0.2+(0.9)(-0.0065)(1) = 0.194$
θ_6	$0.1+(0.9)(0.1311) = 0.218$
θ_5	$0.2+(0.9)(-0.0065) = 0.194$
θ_4	$-0.4+(0.9)(-0.0087) = -0.408$

Neural Network

Propagate the inputs forward

- Given a unit j in a hidden or output layer, the net input, I_j, to unit j is $I_j = \sum_i w_{ij} O_i + \theta_j$ 前一层节点的输出乘以相应的权重再求和最后加上偏移量

  where w_{ij} is the weight of the connection from unit i in the previous layer to unit j; O_i is the output of unit i from the previous layer; and θ_j is the bias of the unit.
- Given the net input I_j to unit j, then O_j, the output of unit j, is computed as $O_j = \dfrac{1}{1+e^{-I_j}}$

Backpropagate the error

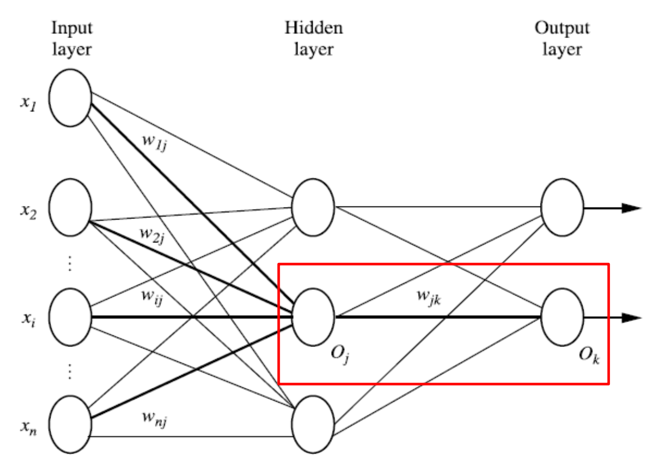
- For a unit k in the output layer, the error Err_k is computed by

  $Err_k = O_k(1-O_k)(T_k - O_k)$ T_K表示实际结果
- The error of a hidden layer unit j is

  $Err_j = O_j(1-O_j)\sum_k Err_k w_{jk}$ 求和表示该节点有多个输出 k为下一层节点
- Weights are updated by

  $w_{jk} = w_{jk} + \eta Err_k O_j$

  $\theta_k = \theta_k + \eta Err_k$



Neural Network

- Minimize the error of node O_k
- We define it as $E = \frac{1}{2}e^2 = \frac{1}{2}(T-O)^2$

与真实数据的误差的二分之一方差比较常用

- To adjust weight w_{jk}, we first calculate the partial derivation of E on w_{jk}

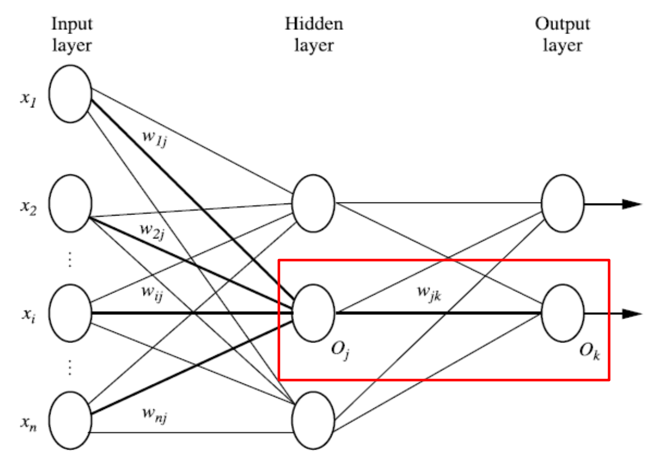
偏微分

$$\frac{\partial E}{\partial w_{jk}} = \frac{\partial E}{\partial e} \times \frac{\partial e}{\partial O_k} \times \frac{\partial O_k}{\partial w_{jk}}$$

$$= -(e) \times \left(O_k(1-O_k)\right) \times \left(O_j\right)$$

$$= -(T_k - O_k)O_k(1-O_k)O_j$$

- and then use the "gradient decent"



Neural Network

- Backpropagation learning is based on the idea of an error surface.
- The surface represents cumulative 累计 error over a data set as a function of network weights.
- Each possible network weight configuration 配置 is represented by a point on the surface.

Neural Network

- The goal of the learning algorithm is to determine a set of weights that minimize the error.
- The learning algorithm should be designed to find the direction on the surface which most rapidly reduces the error.
- This can be achieved by moving in the 实现 opposite direction of the gradient vector at each surface point (i.e., by employing the gradient descent learning method).
通过在每个表面点沿梯度向量的相反方向移动来实现

Neural Network

- **Weakness**
  - 很长的训练时间
  - Long training time
  - 需要一系列参数，通常由经验确定，例如，网络拓扑或结构
  - Require a number of parameters typically best determined empirically, e.g., the network topology or "structure".
  - Poor interpretability 解释能力差
    - Difficult to interpret the symbolic meaning behind the learned weights and of "hidden units" in the network
    - 很难解释所学的权重背后的含义和网络中的 " 隐藏单元 "

Neural Network

- **Strength**
  - 高噪声数据耐受性
    High tolerance to noisy data
  - 非常适合连续值输入和输出
    Well-suited for continuous-valued inputs and outputs
  - 在大量真实数据上取得成功
    Successful on a wide array of real-world data
  - 最近已开发出从训练过的神经网络中提取规则的技术。
    Techniques have recently been developed for the extraction of rules from trained neural networks

Neural Network

从网络中提取规则

- **Rule extraction from networks**: network pruning
  - Simplify the network structure by removing weighted links that have the least effect on the trained network

  研究一组输入和激活值来推导出描述输入和隐藏单元层之间关系的规则

  - The set of input and activation values are studied to derive rules describing the relationship between the input and hidden unit layers

敏感性分析

- **Sensitivity analysis**: assess the impact that a given input variable has on a network output. The knowledge gained from this analysis can be represented in rules

评估给定输入变量对网络输出的影响，从这个分析中获得的知识可以用规则表示