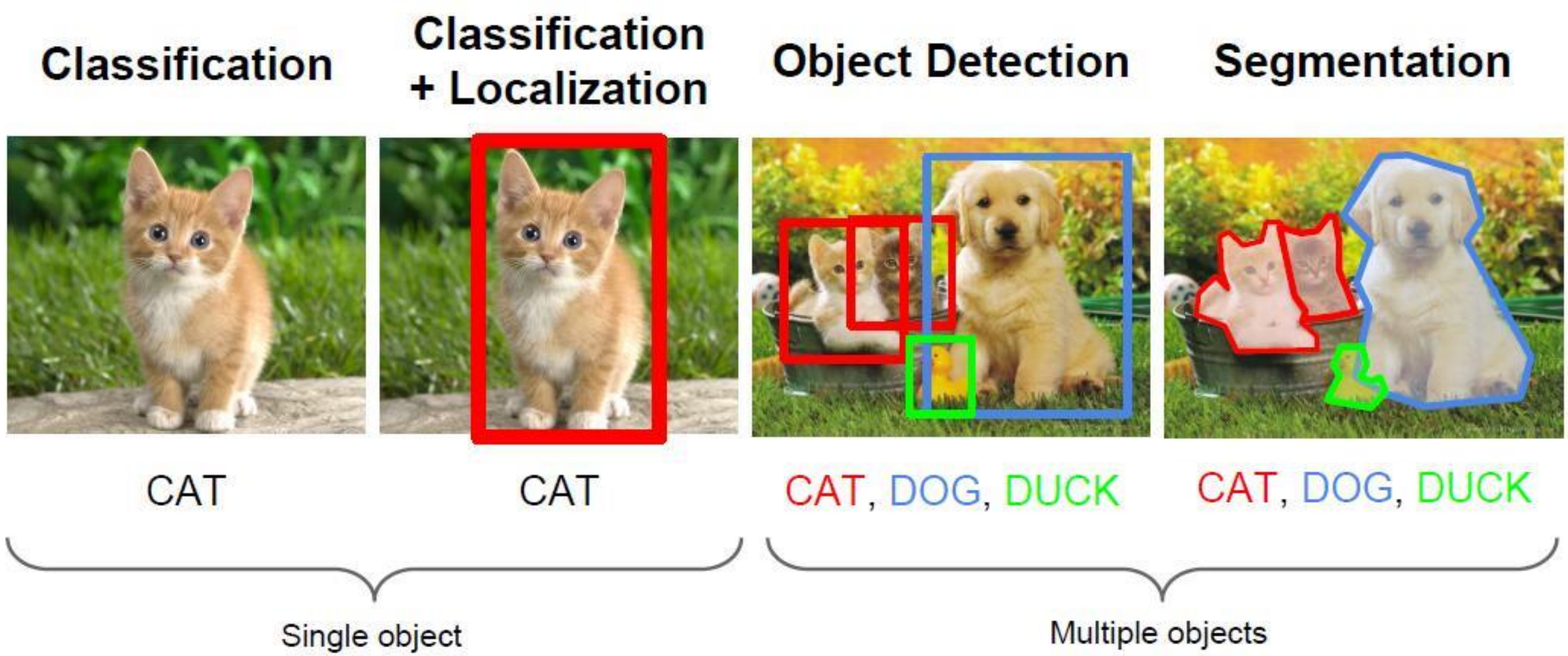
本科生《计算机视觉》
基于深度学习的视觉理解与生成

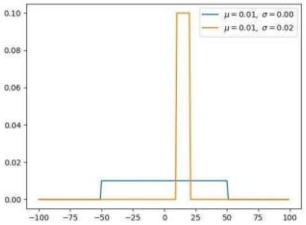
黄雷
人工智能研究院
huangleiAI@buaa.edu.cn

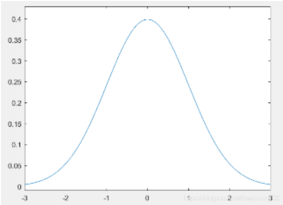
2023年10月10日

计算机视觉任务

- 数据集
- 判别任务
  - 分类
  - 目标检测
  - 分割

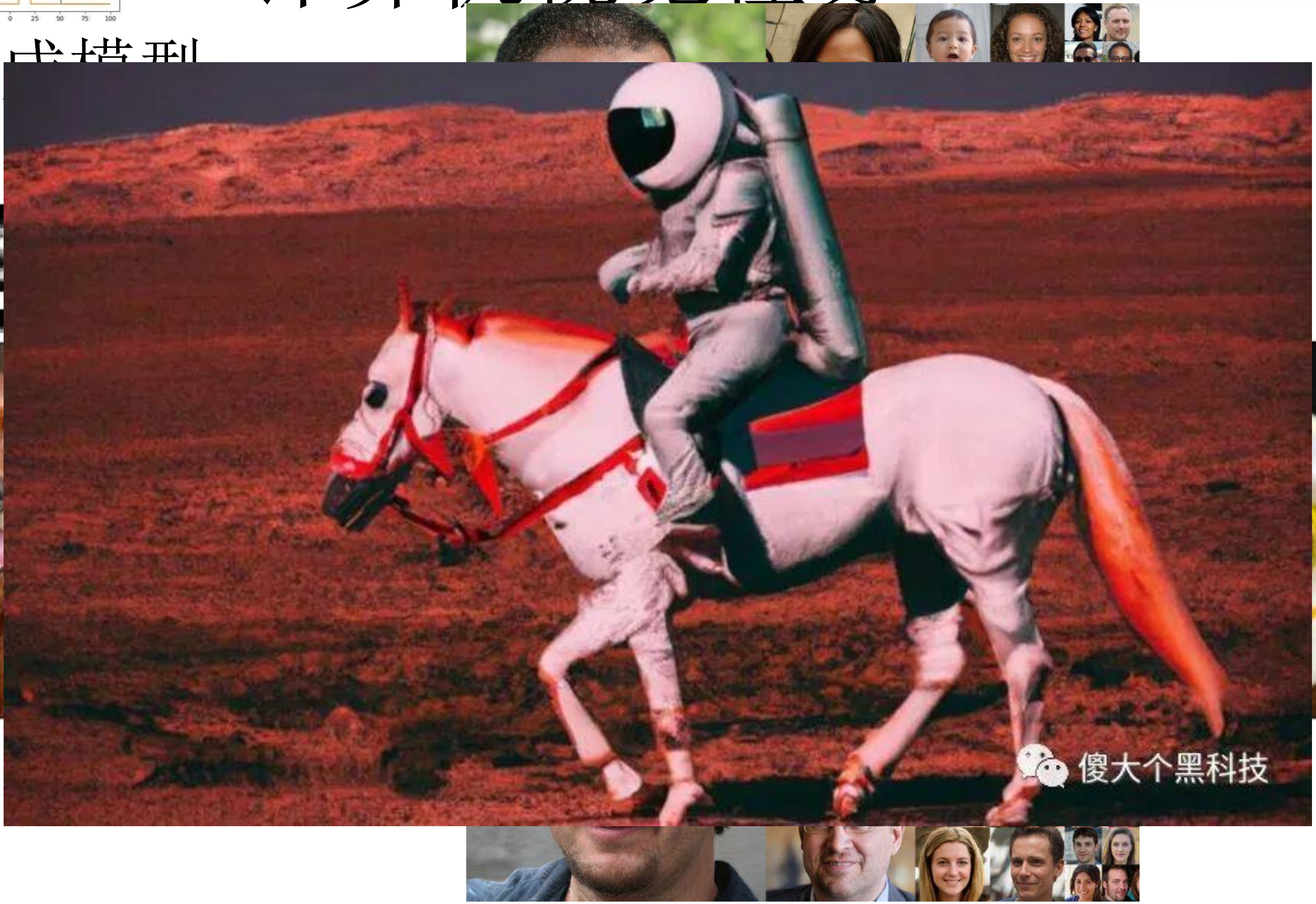






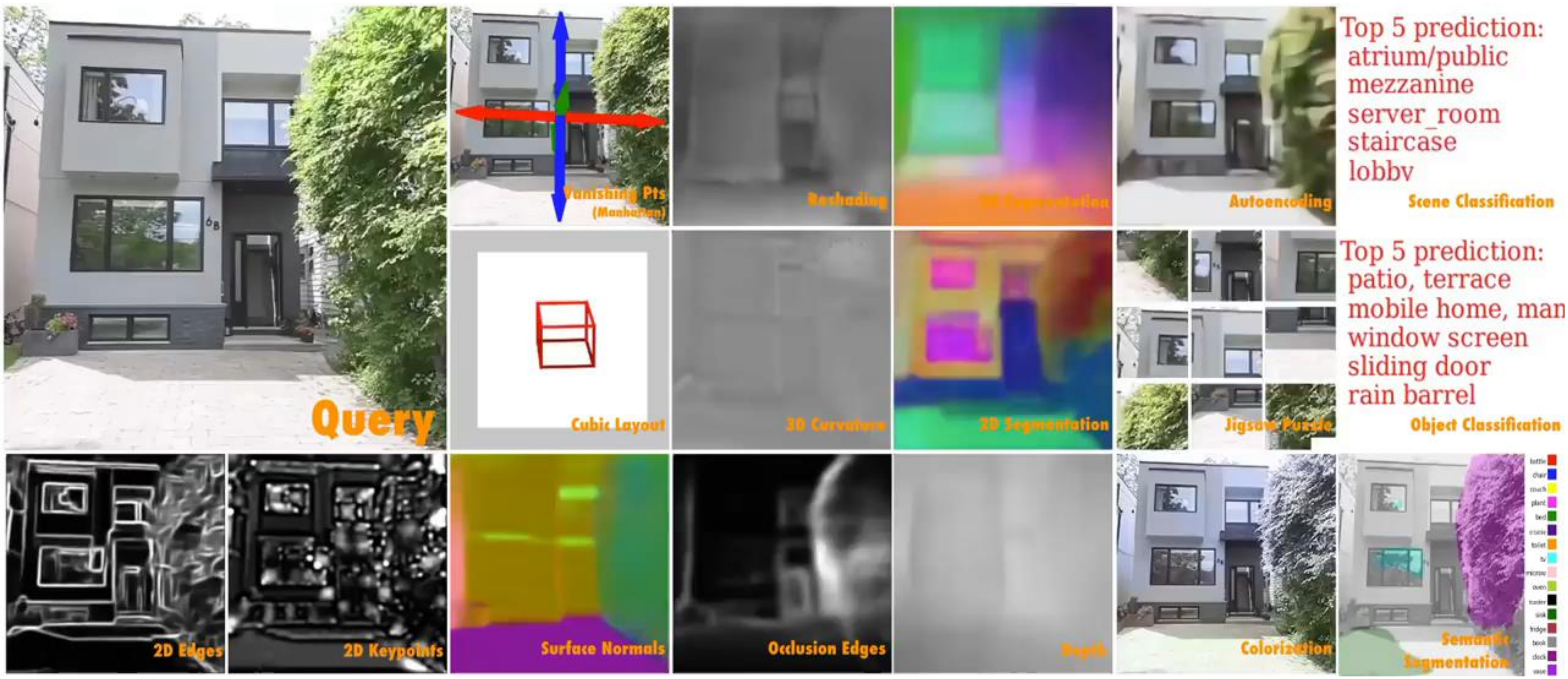
计算机视觉任务

- 生成模型

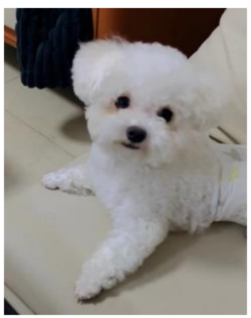


计算机视觉任务

- 多任务学习（大模型-统一模型）



计算机视觉-任务



图像分类
（Image
Classification）

Dog

计算机视觉-任务

输入（X）

F(·)

输出（Y）

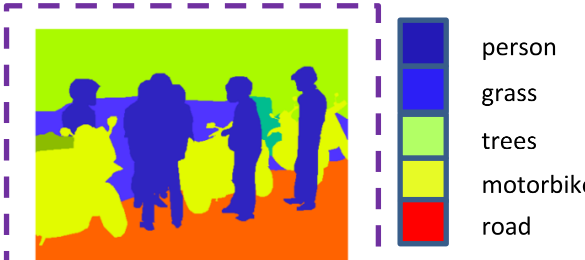
图像摘要生成（Image Caption）

A large clock in a large room with a glass ceiling

计算机视觉-任务

输入（X） F(·) 输出（Y）

图像分割
（Image Segmentation）



计算机视觉-任务

输入（X） → F(·) → 输出（Y）

the train is on the tracks in the station

图像生成
（Image Synthesis）

计算机视觉-任务

输入（X） → F(·) → 输出（Y）

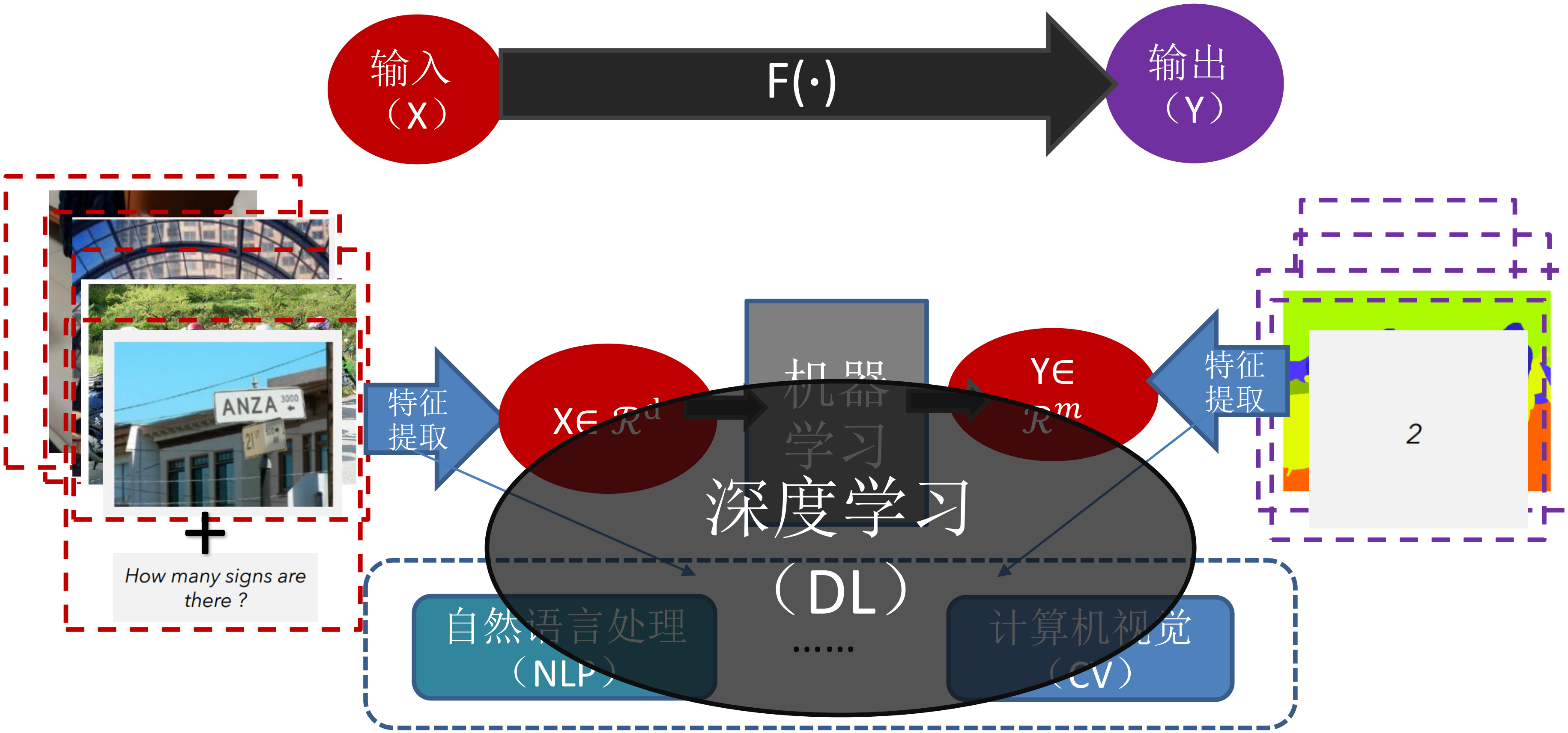
+

How many signs are there ?

视觉问答（Visual Question Answer, VQA）

2

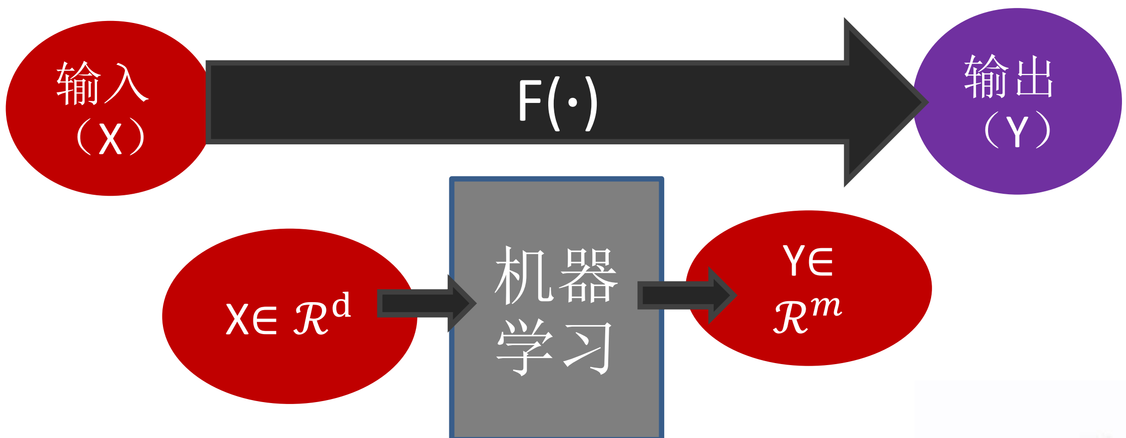
深度学习：端到端统一


输入（X）
F(·)
输出（Y）
特征提取
X∈$\mathcal{R}^d$
机器学习
Y∈$\mathcal{R}^m$
特征提取
ANZA
How many signs are there ?
2
深度学习（DL）
自然语言处理（NLP）
……
计算机视觉（CV）

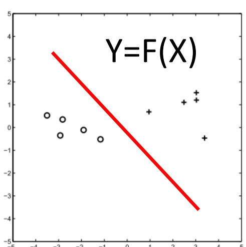
主要内容

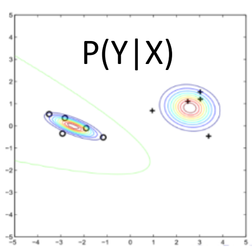
- 深度学习基础
  - 神经网络及反向传播算法
  - 卷积神经网络中的视觉表示思想
- 视觉理解任务
  - 目标检测
    - R-CNN系列；DERT系列；Pix2Seq系列
  - 分割
    - 语义分割，实例分割，全景分割，Prompt-based分割
- 视觉生成
  - 深度生成模型
    - VAE, AR, GAN, Diffusion
  - 图像翻译
- 深度神经网络训练技巧

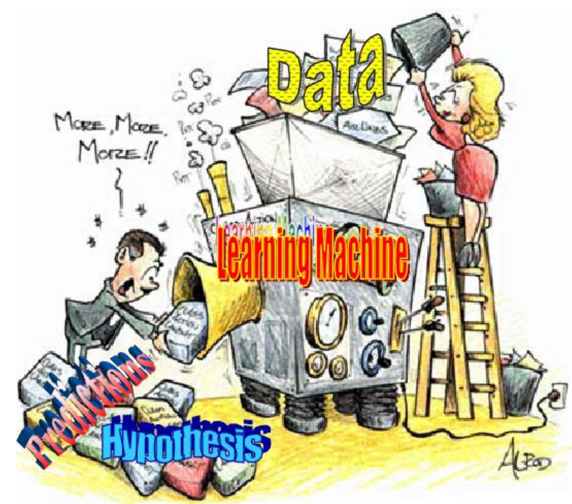
数据驱动的学习



- 目标
  - 拟合: 记住训练数据
  - 泛化: 推广未见数据

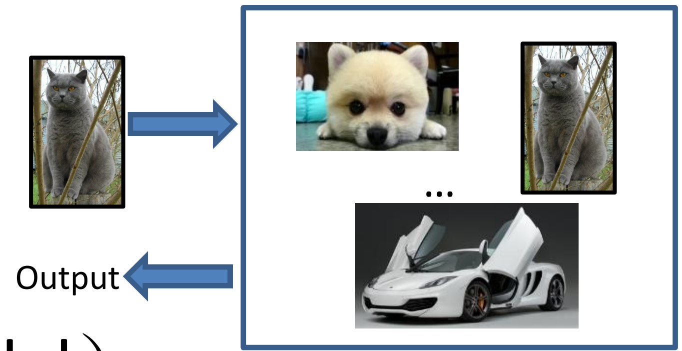




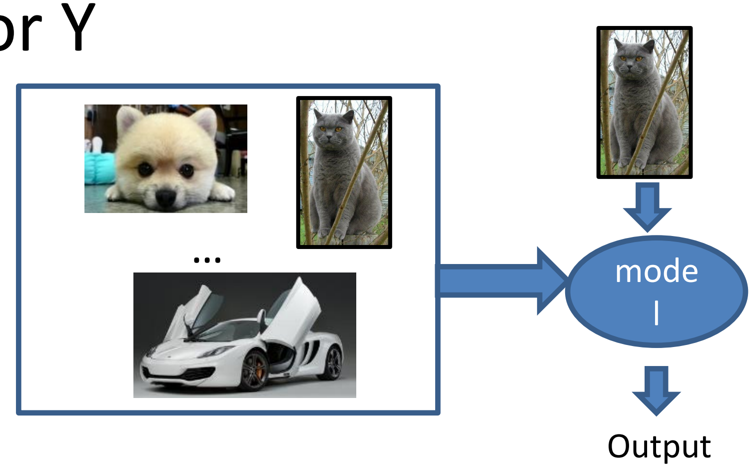


Non-parametric vs parametric model

- 非参模型（Non-parametric model）
  - Y=F(X; $x_1, x_2 \ldots x_n$)



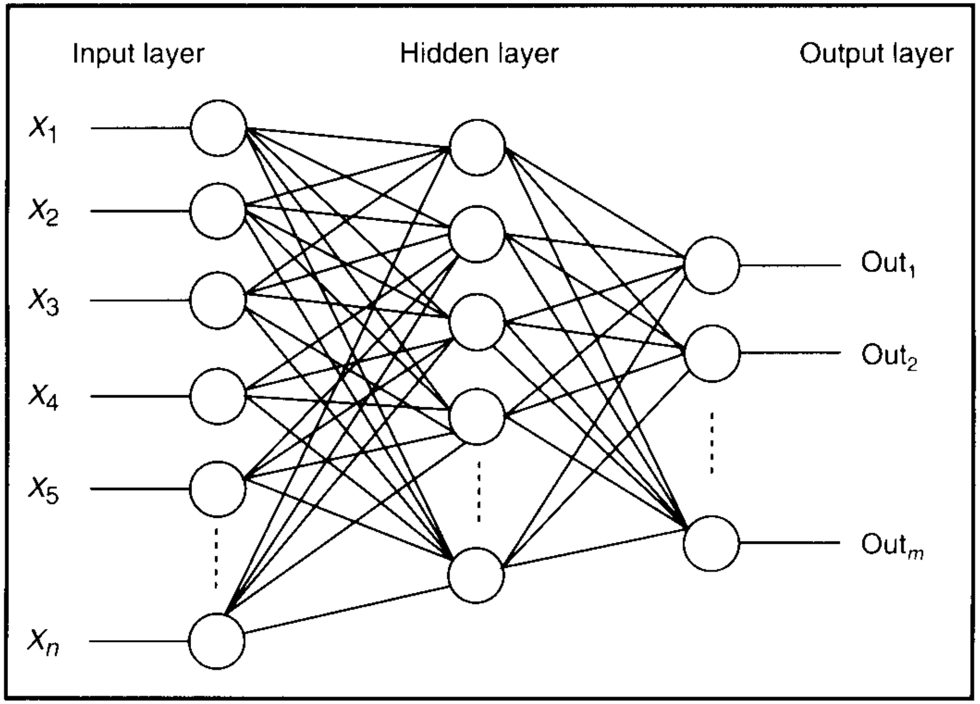
- 参数化模型（Parametric model）
  - Y=F(X; θ)
  - Training for θ; Inference for Y



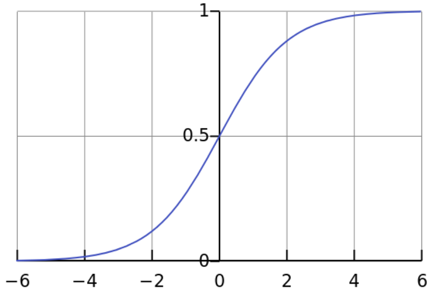
- 记忆 v.s. 泛化

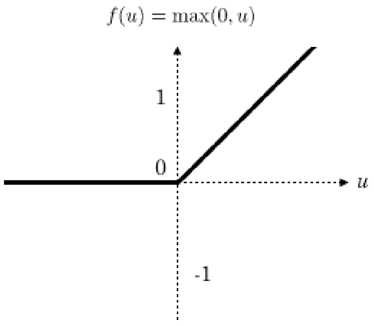
Neural network

- Neural network
  - Y=F(X)=$f_T(f_{T-1}(\dots f_1(X)))$
  - $f_i(x) = g(Wx + b)$



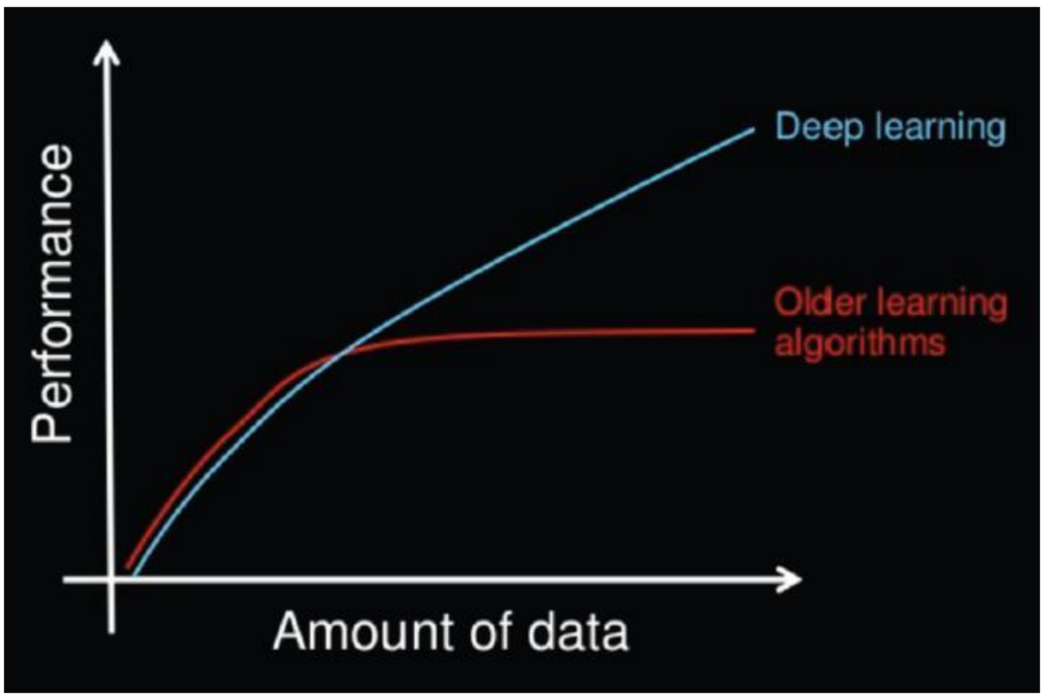
- Nonlinear activation
  - sigmod
  - Relu





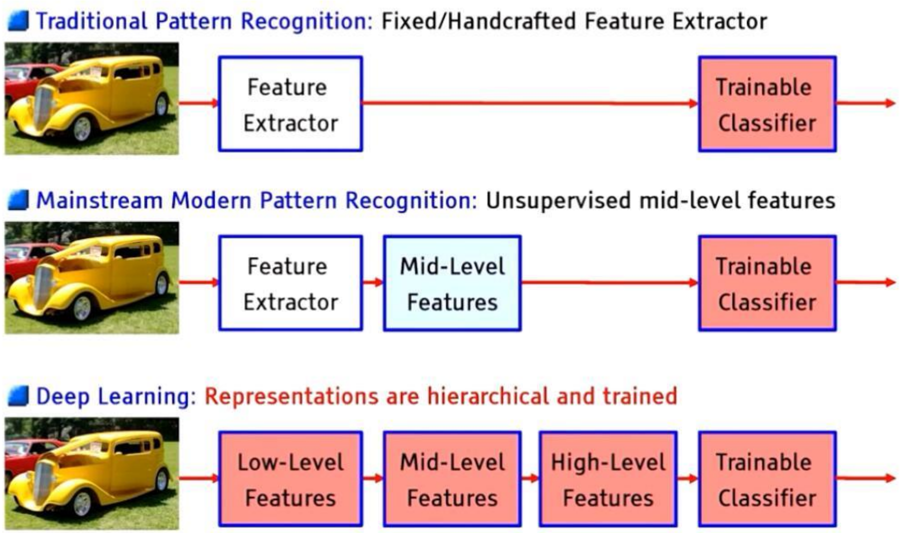
Deep neural network

- Why deep?
  - Powerful representation capacity(函数表达能力)



Key properties of Deep learning

- End to End learning （端到端学习）
  - no distinction between feature extractor and classifier



- “Deep” architectures:
  - cascade of simpler non-linear modules

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

One example(一个贯穿全文的例子)

- Classification tasks
  - Binary classification(二分类)： is cat？
  - Multiple classification（多分类）： is cat, dog, others?
- Assumption
  - The feature vectors of the images are provided， 3-dimensinal vectors

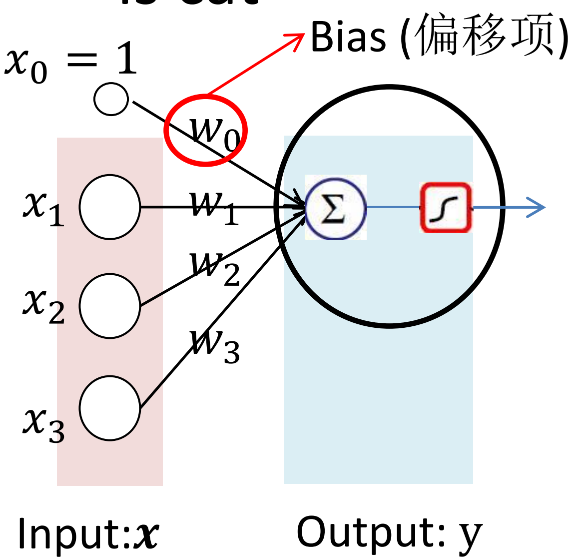
$$(x_1 , x_2 , x_3)^T$$

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

Linear Classifier (线性分类器)

- One neuron
  - Binary classification (二分类问题) is cat？



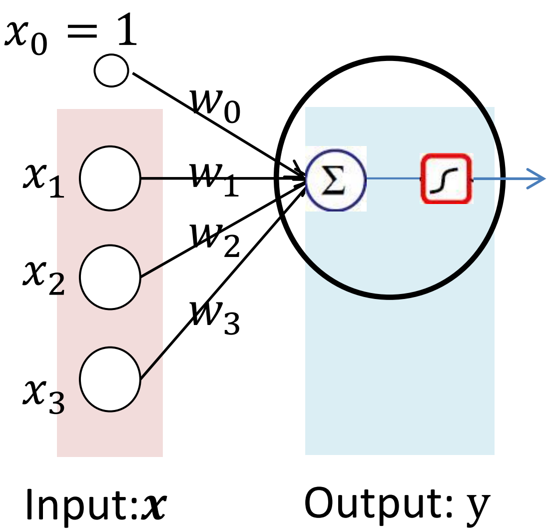
$$(x_1, x_2, x_3)^T$$

$$a = \sum_{i=0}^{3} w_i x_i = \mathbf{w} \cdot \boldsymbol{x}$$

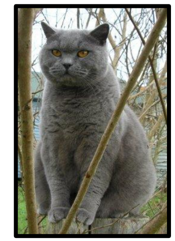
$$y = \sigma(a) = \frac{1}{1 + e^{-a}}$$

Linear Classifier

- One example



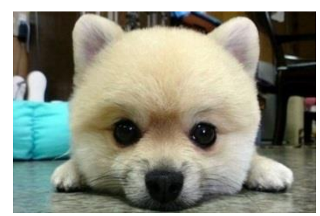
Model: $\mathbf{w}$= （$w_0, w_1, w_2, w_3)^T$
=(2, 0,0, 4)^T



$(x_1 , x_2 , x_3)^T$=(2,2,3)^T

a =*2*1+0*2+0*2+4*3=14*

$$y = \varphi(a) = \frac{1}{1 + e^{-14}} > 0.5$$



$(x_1 , x_2 , x_3)^T$=(1,2,-3)^T

a =*2*1+0*1+0*2+4*(-3)=-10*

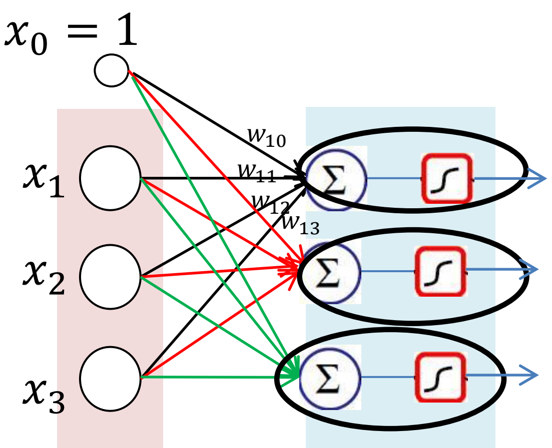
$$y = \varphi(a) = \frac{1}{1 + e^{10}} < 0.5$$

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

Linear Classifier (线性分类器)

- Multiple neurons
  - Multiple classification： is cat？ dog？ others？



Input:$\boldsymbol{x}$ Output: $\mathbf{y}$

(x_1, x_2, x_3)

w_{ij}: 第 i 个输出神经元，连接第j 个输入单元

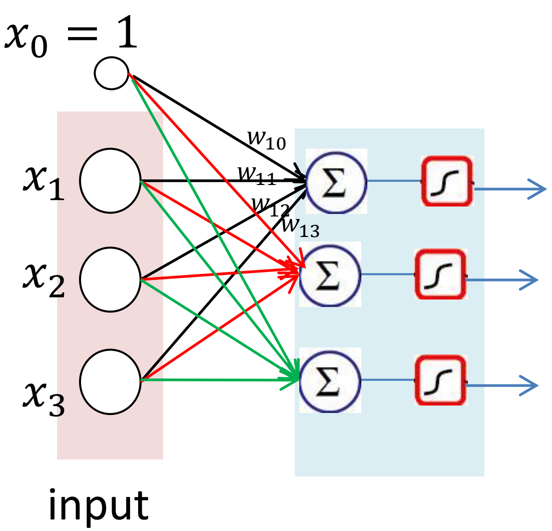
$$a_i = \sum_{j=0}^{3} w_{ij} x_i = \boldsymbol{w}_{i.} \cdot \boldsymbol{x}, \qquad i = (1,2,3)$$

$$y_i = \sigma(a_i) = \frac{1}{1 + e^{-a_i}}$$

$$\boldsymbol{a} = \boldsymbol{W} \cdot \boldsymbol{x}$$
$$\mathbf{y} = \boldsymbol{\sigma}(\boldsymbol{a})$$

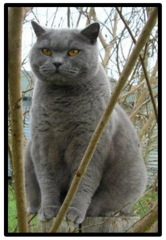
Linear Classifier (线性分类器)

- One example



Model: $\boldsymbol{W}$=

2	0	0	4
0	0	-2	-3
1	-1	0	0



$(x_1, x_2, x_3)^T$=(2,2,3)^T

$\boldsymbol{a} =$

2	0	0	4
0	0	-2	-3
1	-1	0	0

*(1,2,2,3)^T

$$= (14, -13, -1)$$

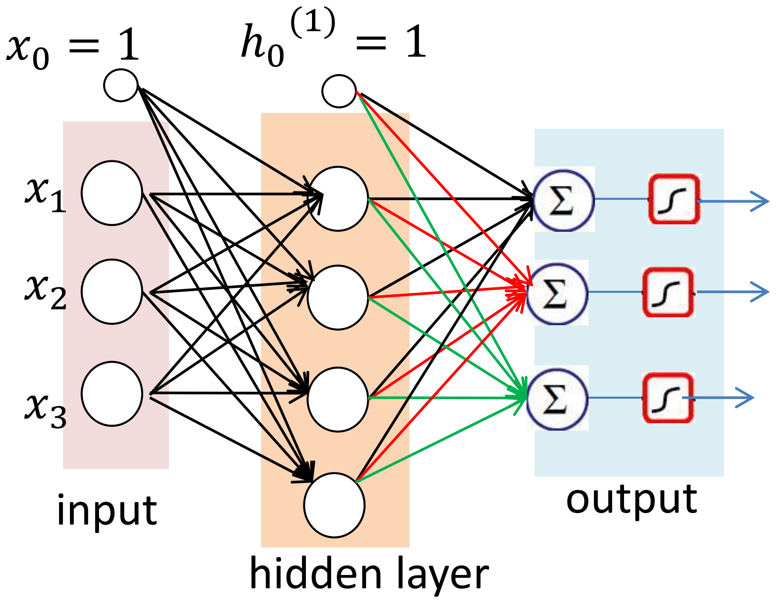
$$\mathbf{y} = \left(\frac{1}{1+e^{-14}}, \frac{1}{1+e^{13}}, \frac{1}{1+e^{1}}\right)^T$$

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

Multi-layer perceptron （多层感知机）

- **Multi-layer perceptron or feed-forward neural network**



x_i: 第i 个输入节点

$h_i^{(k)}$: 第 k 层隐藏层的第i个节点

$w_{ij}^{(k)}$:第 k 层隐藏层, 第 i 个输出神经元，连接第j 个输入神经元

y_i: 第i 个输出节点

Pre-activation → $\boldsymbol{a}^{(1)} = \boldsymbol{W}^{(1)} \cdot \boldsymbol{x}$

activation → $\boldsymbol{h}^{(1)} = \sigma(\boldsymbol{a}^{(1)})$

$$\boldsymbol{a}^{(2)} = \boldsymbol{W}^{(2)} \cdot \boldsymbol{h}^{(1)}$$
$$\boldsymbol{y} = \sigma(\boldsymbol{a}^{(2)})$$

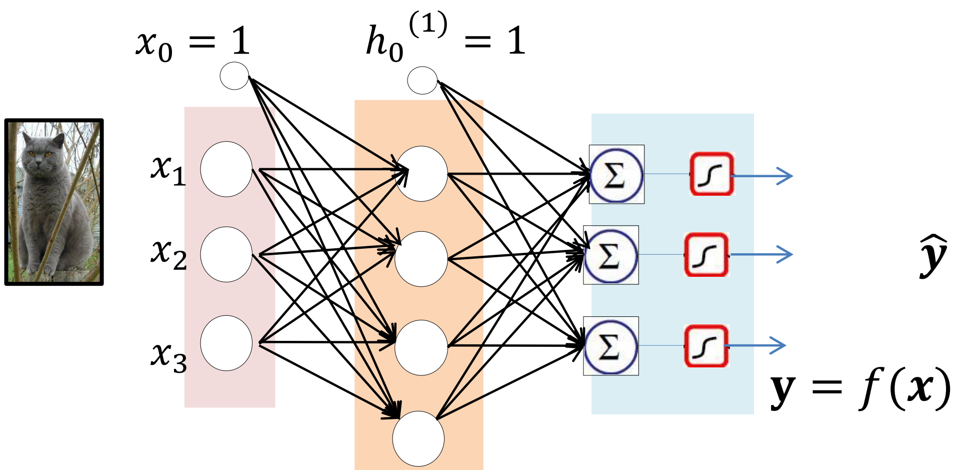
$$\boldsymbol{a}^{(i)} = \boldsymbol{W}^{(i)} \cdot \boldsymbol{h}^{(i-1)}$$
$$\boldsymbol{h}^{(i)} = \sigma(\boldsymbol{a}^{(i)})$$
$$(\boldsymbol{h}^{(0)} = \boldsymbol{x}, \boldsymbol{h}^{(L)} = \boldsymbol{y})$$

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

Target of learning: Loss function

- 损失函数（Loss function）



Cat	dog	car
$(1, 0, 0)^T$	$(0, 1, 0)^T$	$(0, 0, 1)^T$

$$L = (\mathbf{y} - \hat{\boldsymbol{y}})^2$$
$$= (1 - y_1)^2 + y_2{}^2 + y_3{}^2$$

均方误差（Mean Squared Error）： L=$(\mathbf{y} - \hat{\boldsymbol{y}})^2$, 其中 $\mathbf{y} = f(\boldsymbol{x})$

- 优化目标函数: min L=$(\mathbf{y} - \hat{\boldsymbol{y}})^2$

目标函数

- 求解目标函数

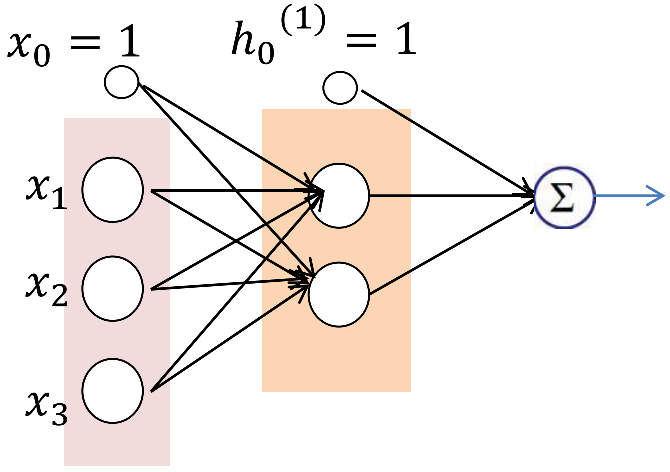
$$\min_{\mathbf{W}} E_{(X,\hat{Y})\sim D} \ \mathrm{L}(\mathrm{F}(\mathrm{X};\mathbf{W}),\hat{Y}))$$

目标函数

- 求解目标函数

$$\min_{W} E_{(X,\hat{Y})\sim D}\ \mathrm{L}(\mathrm{F}(\mathrm{X};\mathrm{W}),\hat{Y}))$$

- 方案一：令 $\frac{\partial \mathrm{L}}{\partial W}=0$，求解方程组



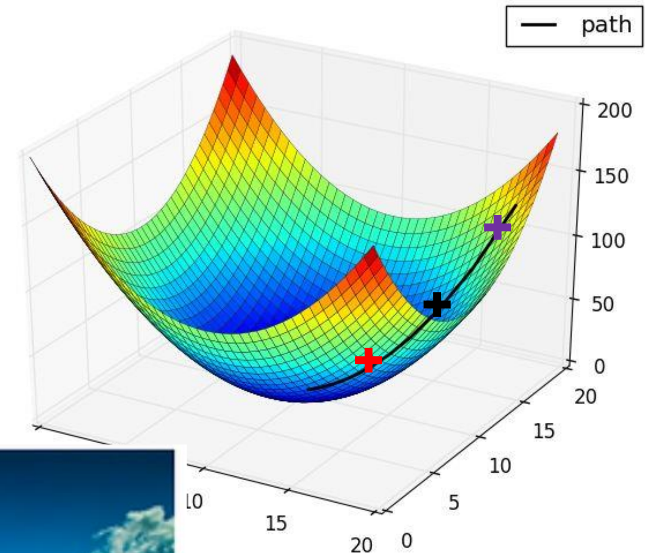
$$\mathrm{L}=\Big(\frac{w_{21}^{(2)}}{1+\mathrm{e}^{-(x_1 w_{11}^{(1)}+x_2 w_{12}^{(1)}+x_3 w_{13}^{(1)}+w_{10}^{(1)})}}+\frac{w_{22}^{(2)}}{1+\mathrm{e}^{-(x_1 w_{11}^{(1)}+x_2 w_{12}^{(1)}+x_3 w_{13}^{(1)}+w_{10}^{(1)})}}+w_{20}^{(2)}-\hat{y}\Big)^2$$

outline

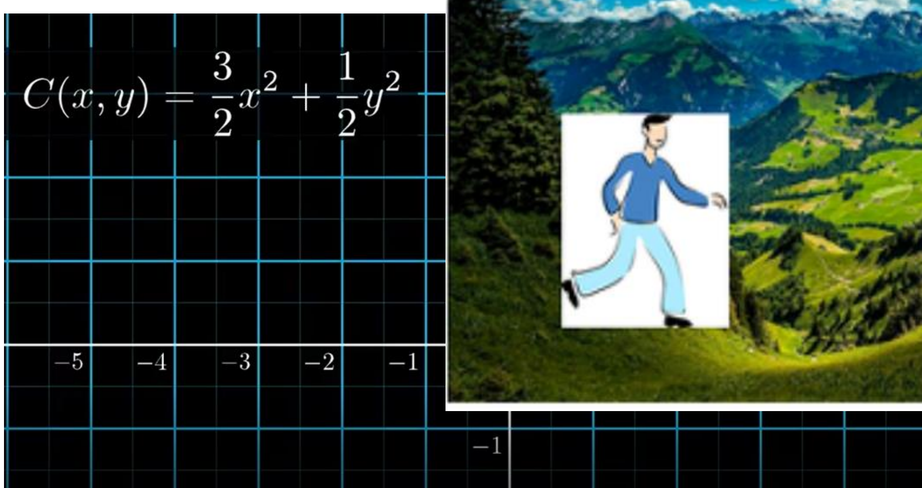
- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

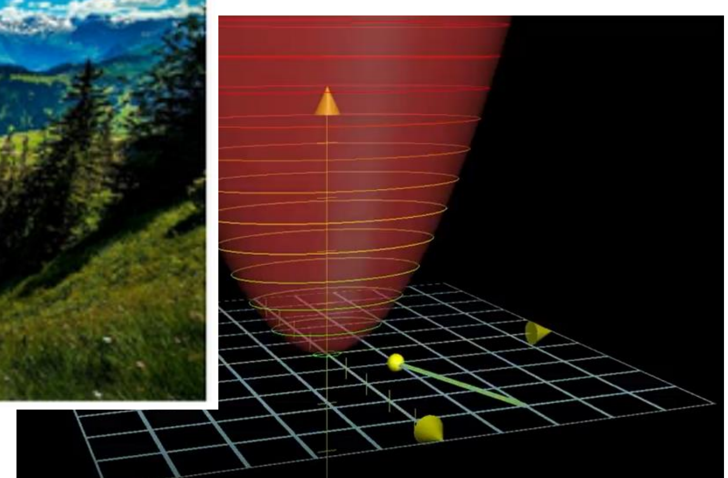
基于迭代的训练方式

$$\min_{W} E_{(X,\hat{Y})\sim D} \; L(F(X;W),\hat{Y}))$$



- 局部下降搜索
  - 基于目前的参数W^{t}，给其多个扰动ΔW ，确保存在某个ΔW，使得$L(W^{t}+\Delta W) < L(W^{t})$,
  - 更新 $W^{t+1} = W^{t} + \Delta W$
- 更高效的下降搜
  - 基于梯度的下

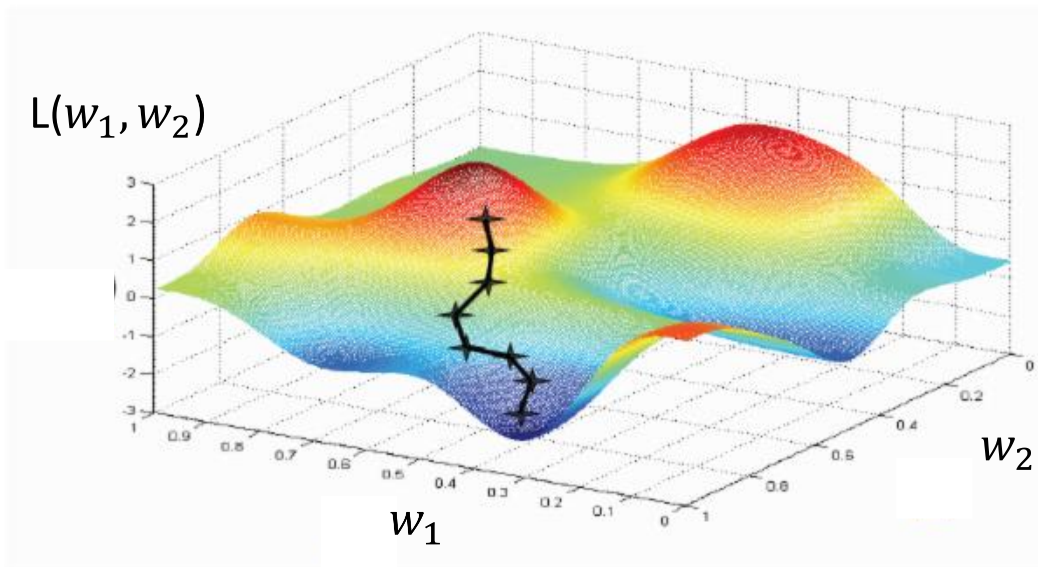




梯度下降算法

- 0.初始化权重 $\boldsymbol{W}^{(0)}$
- 1. 前向过程：
  - 1.1根据输入，计算输出值 $\boldsymbol{y}$
  - 1.2.计算损失函数值L($\mathbf{y}, \hat{\mathbf{y}}$)。
- 2.计算梯度$\frac{d\,L}{\boldsymbol{d\,W}}$
- 3.更新梯度

$$\boldsymbol{W}^{(t+1)} = \boldsymbol{W}^{(t)} - \eta \frac{d\,\mathrm{L}}{\boldsymbol{d\,W}^{(t)}}$$



gradient：$\left(\frac{\mathrm{dL}(w_1,w_2)}{w_1}, \frac{\mathrm{dL}(w_1,w_2)}{w_2}\right)$

outline

- Linear classifier (简单线性分类器)
  - One neuron (一个神经元)
  - Multiple neurons (多个神经元)
- Multi-layer perceptron (多层感知机)
  - Model representation (模型表示)
  - Loss function: the goal for learning
  - Training
    - Gradient based optimization
    - Back-propagation

计算梯度：反向传播

- 求导基础知识回顾

➢实值函数对一维实值变量的导数:

$$\frac{df(x)}{dx} = \lim_{h \to 0} \frac{f(x+h) - f(x)}{h}$$

➢实值函数对多维向量变量的梯度为向量（偏导数）：

$$\frac{\partial \mathrm{f}(\boldsymbol{\theta})}{\partial \boldsymbol{\theta}} = \left(\frac{\partial f(\theta_0,\theta_1)}{\partial \theta_0}, \frac{\partial \mathrm{f}(\theta_0,\theta_1)}{\partial \theta_1}\right), \quad \boldsymbol{\theta} = (\theta_0, \theta_1)$$

计算梯度：反向传播

- 神经网络中的基本操作

加法： $f(x,y)=x+y$ $\quad \frac{\partial f}{\partial \boldsymbol{x}}=1 \quad \frac{\partial f}{\partial \boldsymbol{y}}=1$

乘法： $f(x,y)=xy$ $\quad \frac{\partial f}{\partial \boldsymbol{x}}=y \quad \frac{\partial f}{\partial \boldsymbol{y}}=x$

非线性变换： $\sigma(x)=\frac{1}{1+e^{-x}}$

$$\frac{d\sigma(x)}{dx}=\frac{e^{-x}}{(1+e^{-x})^2}=\left(\frac{1+e^{-x}-1}{1+e^{-x}}\right)\left(\frac{1}{1+e^{-x}}\right)=(1-\sigma(x))\sigma(x)$$

反向传播（Back-Propagation）

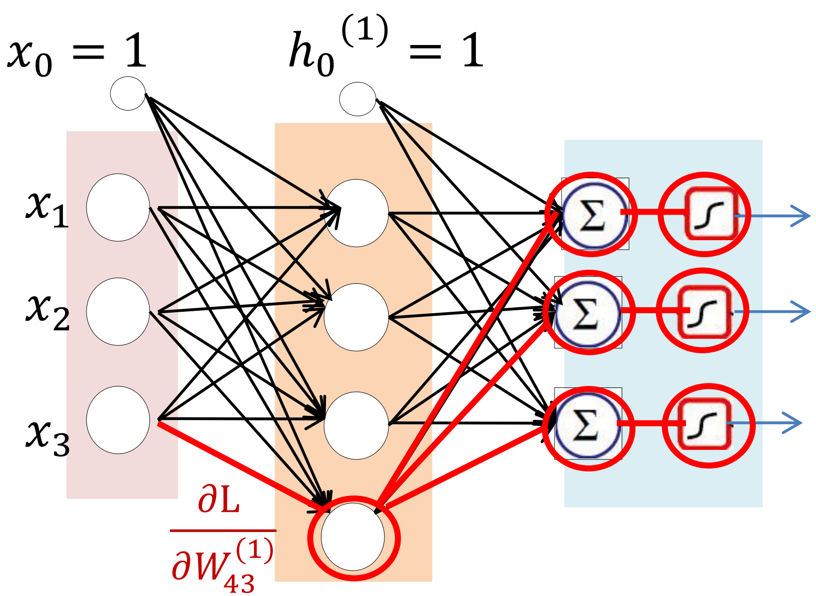
- 链式法则（Chain rule）

$f=q(x)$ $\qquad \dfrac{\partial f}{\partial x} = \dfrac{\partial f}{\partial q}\dfrac{\partial q}{\partial x}$

复合表达式: $f(x,y,z) = (x+y)z$

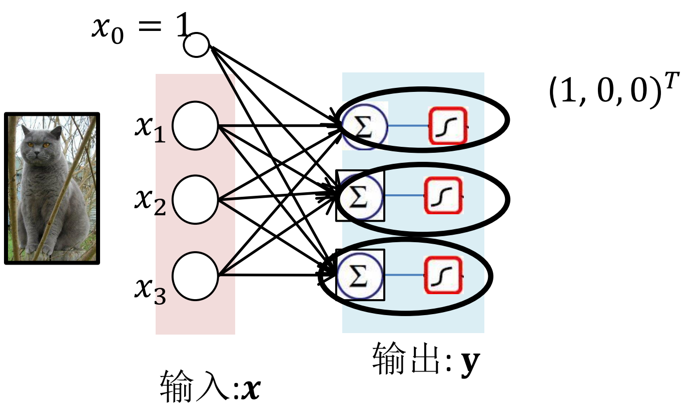
$q=x+y \qquad \dfrac{\partial q}{\partial x}=1 \qquad \dfrac{\partial q}{\partial y}=1$

$f=qz \qquad \dfrac{\partial f}{\partial q} = \mathrm{z} \qquad \dfrac{\partial \mathrm{f}}{\partial \mathrm{z}} = \mathrm{q}$



反向传播

- 一层神经网络（线性模型）



➢ 1.给定输入，计算输出值：

$$a_i = \sum_{j=0}^{3} w_{ij} x_i = \boldsymbol{w}_{i.} \cdot \boldsymbol{x},$$

$$i = (1,2,3)$$

$$y_i = \sigma(a_i) = \frac{1}{1 + e^{-a_i}}$$

MSE Loss: L=$(\mathbf{y} - \hat{\mathbf{y}})^2$ =$(1 - y_1)^2$+$y_2{}^2 + y_3{}^2$

➢ 2.根据链规则，计算梯度$\frac{\partial L}{\partial \boldsymbol{w}}$：

$$\frac{\partial \mathrm{L}}{\partial y_1} = 2(y_1\text{-}1)$$

$$\frac{\partial \mathrm{L}}{\partial y_i} = 2y_i, \ (\mathrm{i}=2,3)$$

$$\frac{\partial \mathrm{L}}{\partial a_i} = \frac{\partial \mathrm{L}}{\partial y_i}\frac{\partial y_i}{\partial a_i} = \frac{\partial \mathrm{L}}{\partial y_i}\sigma(a_i)(1\text{-}\sigma(\mathrm{a_i}))$$

$$\frac{\partial \mathrm{L}}{\partial w_{ij}} = \frac{\partial \mathrm{L}}{\partial a_i}\frac{\partial a_i}{\partial w_{ij}} = \frac{\partial \mathrm{L}}{\partial a_i} x_{ij}$$

$$= \frac{\partial \mathrm{L}}{\partial y_i}\sigma(a_i)(1\text{-}\sigma(a_i))$$

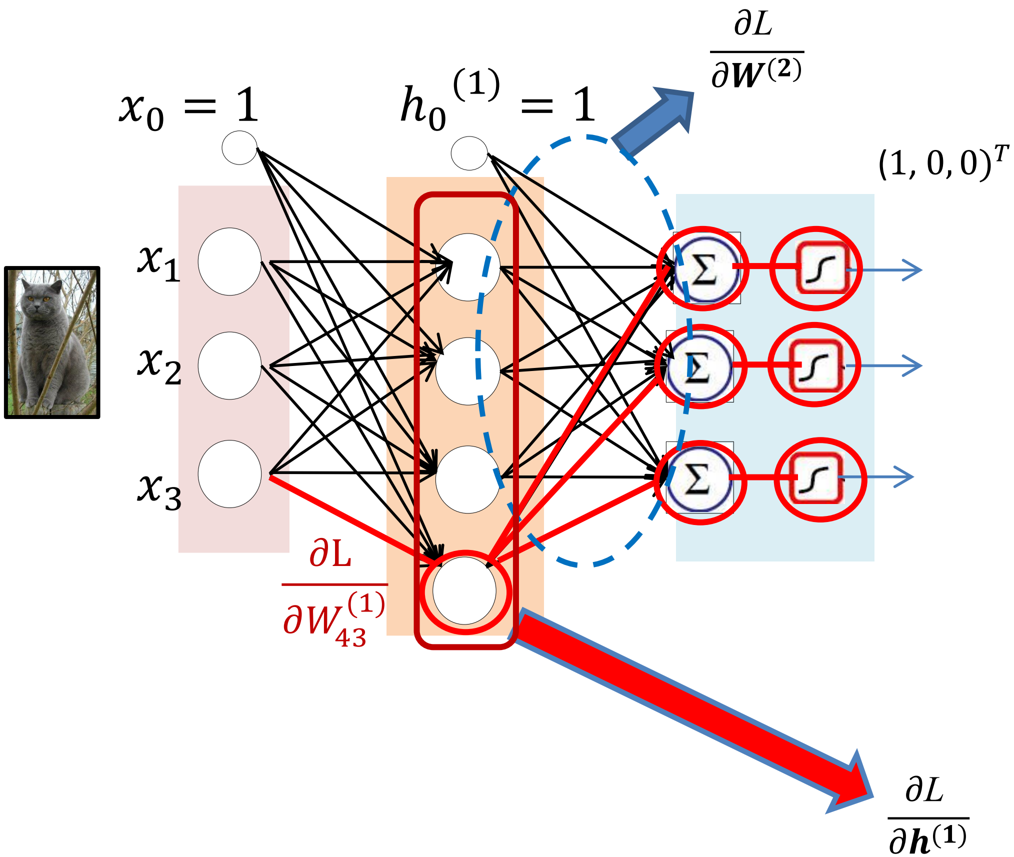
$$\frac{\partial \mathrm{L}}{\partial \mathbf{y}} = 2(\mathbf{y} - \hat{\mathbf{y}})$$

$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{a}} = 2[(\mathbf{y} - \hat{\mathbf{y}}) \cdot \ \sigma(\boldsymbol{a}) \cdot (1 - \sigma(\boldsymbol{a}))]^T$$

$$\frac{\partial L}{\partial \boldsymbol{W}} = 2[(\mathbf{y} - \hat{\mathbf{y}}) \cdot \ \sigma(\boldsymbol{a}) \cdot (1 - \sigma(\boldsymbol{a}))] \ \boldsymbol{x}^T$$

反向传播

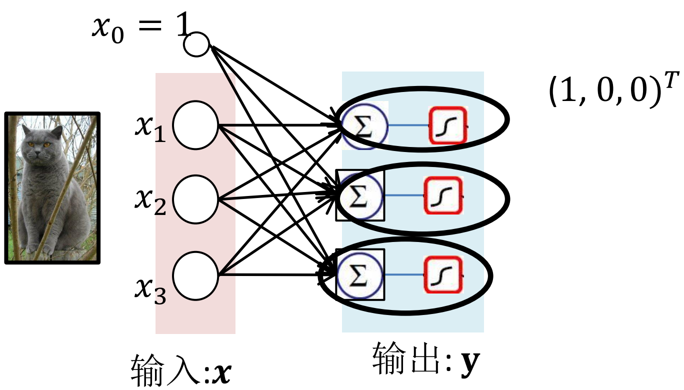
- 两层的网络



MSE Loss: L=$(\mathbf{y} - \hat{\boldsymbol{y}})^2$

反向传播

- 一层神经网络（线性模型）



➢ 1.给定输入，计算输出值：

$$a_i = \sum_{j=0}^{3} w_{ij}x_i = \boldsymbol{w}_{i.} \cdot \boldsymbol{x}, \qquad i = (1,2,3)$$

$$y_i = \sigma(a_i) = \frac{1}{1+e^{-a_i}}$$

MSE Loss: L=$(\mathbf{y} - \hat{\mathbf{y}})^2$ =$(1-y_1)^2$+${y_2}^2 + {y_3}^2$

➢ 2.根据链规则，计算梯度$\frac{\partial L}{\partial \boldsymbol{x}}$：

$$\frac{\partial \mathrm{L}}{\partial y_1}=2(1-y_1)$$

$$\frac{\partial \mathrm{L}}{\partial y_i} = 2y_i, \text{ (i=2,3)}$$

$$\frac{\partial \mathrm{L}}{\partial a_i} = \frac{\partial \mathrm{L}}{\partial y_i}\frac{\partial y_i}{\partial a_i} = \frac{\partial \mathrm{L}}{\partial y_i}\sigma(a_i)(1\text{-}\sigma(a_i))$$

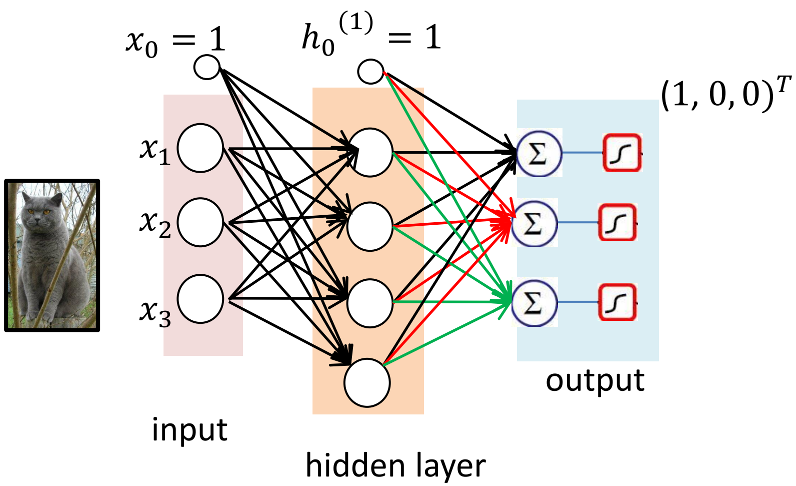
$$\frac{\partial \mathrm{L}}{\partial x_i} = \frac{\partial \mathrm{L}}{\partial a_i}\frac{\partial a_i}{\partial x_i} = \frac{\partial \mathrm{L}}{\partial a_i}\sum_{j=0}^{3} w_{ij}$$

$$\Downarrow$$

$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{x}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{a}}\boldsymbol{W}$$

反向传播

- 两层的网络



➢1给定输入，计算输出值：

$$\boldsymbol{a}^{(1)} = \boldsymbol{W}^{(1)} \cdot \boldsymbol{x}$$
$$\boldsymbol{h}^{(1)} = \sigma\left(\boldsymbol{a}^{(1)}\right)$$
$$\boldsymbol{a}^{(2)} = \boldsymbol{W}^{(2)} \cdot \boldsymbol{h}^{(1)}$$
$$\boldsymbol{y} = \sigma\left(\boldsymbol{a}^{(2)}\right)$$

➢MSE Loss: L=$(\mathbf{y} - \hat{\mathbf{y}})^2$

➢2根据链规则，计算梯度$\frac{\partial L}{\partial \boldsymbol{a}^{(i)}}$,$\frac{\partial L}{\partial \boldsymbol{x}}$：

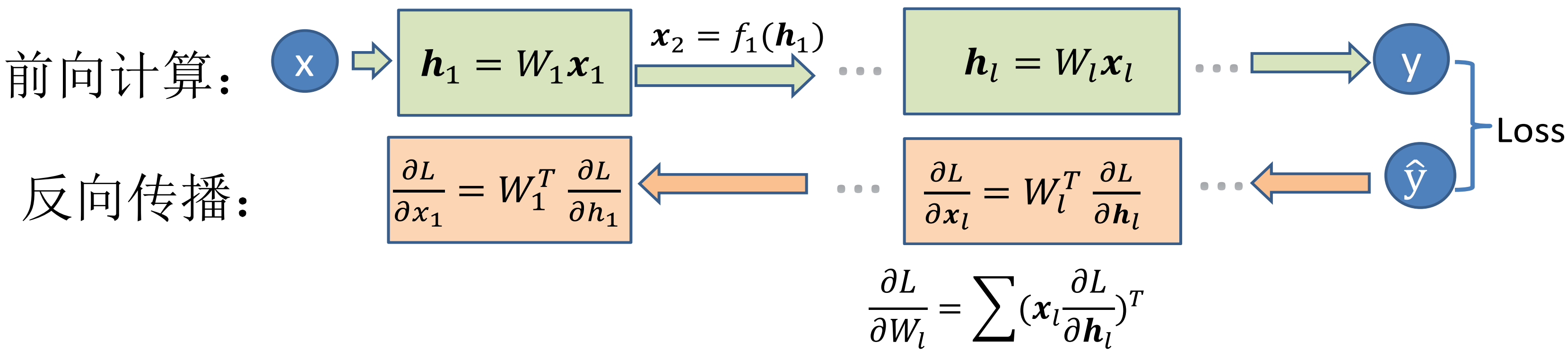
$$\frac{\partial \mathrm{L}}{\partial \mathbf{y}} = 2(\mathbf{y} - \hat{\mathbf{y}})$$
$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(2)}} = \frac{\partial \mathrm{L}}{\partial \mathbf{y}} \cdot \sigma\left(\boldsymbol{a}^{(2)}\right) \cdot \left(1 - \sigma\left(\boldsymbol{a}^{(2)}\right)\right)$$
$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{h}^{(1)}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(2)}} \boldsymbol{W}^{(2)}$$
$$\frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(1)}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{h}^{(1)}} \cdot \sigma\left(\boldsymbol{a}^{(1)}\right) \cdot \left(1 - \sigma\left(\boldsymbol{a}^{(1)}\right)\right)$$
$$\frac{\partial L}{\partial \boldsymbol{x}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(1)}} \boldsymbol{W}^{(1)}$$

➢3.根据链规则，计算梯度$\frac{\partial L}{\partial \boldsymbol{W}^{(i)}}$：

$$\frac{\partial L}{\partial \boldsymbol{W}^{(2)}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(2)}} \boldsymbol{h}^{(1)}$$
$$\frac{\partial L}{\partial \boldsymbol{W}^{(1)}} = \frac{\partial \mathrm{L}}{\partial \boldsymbol{a}^{(1)}} \boldsymbol{x}$$

反向传播

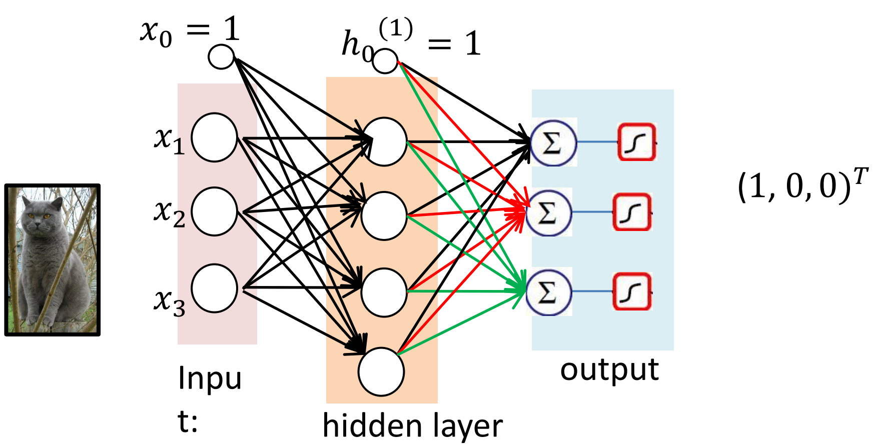
- 利用链式法则计算梯度（数学基础）
- 利用了动态规划的思想（算法设计）



前馈神经网络梯度下降训练算法

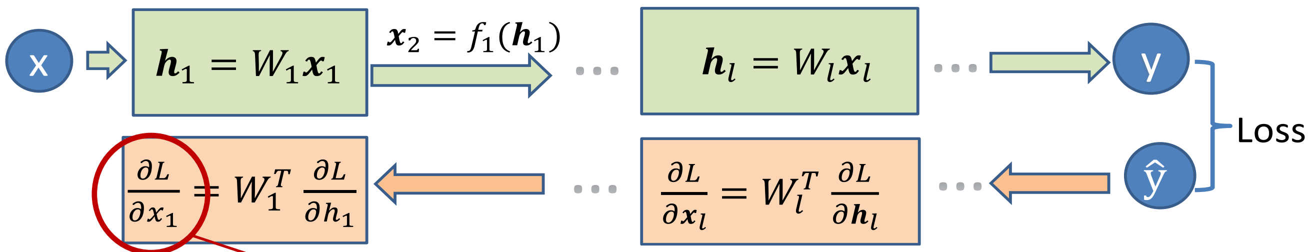
- 0.初始化权重$\boldsymbol{W}_0$
- 1. 前向过程：
  - 1.1根据输入，计算输出值 $\boldsymbol{y}$
  - 1.2.计算损失函数值L($\mathbf{y}, \hat{\boldsymbol{y}}$)。
- 2.反向传播
  - 计算$\frac{\partial \mathrm{L}}{\partial \mathbf{y}}$
  - 后向传播直到计算$\frac{\partial \mathrm{L}}{\partial \mathbf{x}}$
- 3.计算梯度$\frac{\partial L}{\partial \boldsymbol{W}}$
- 4.更新梯度

$$\boldsymbol{W}_{t+1} = \boldsymbol{W}_t - \eta \frac{\partial \mathrm{L}}{\partial \boldsymbol{W}_t}$$

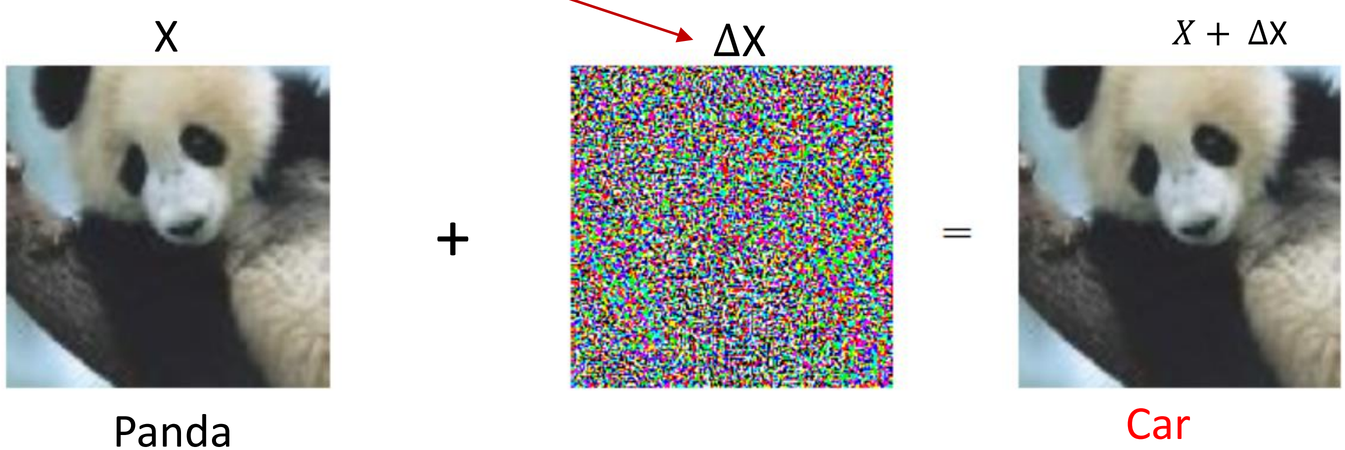


操纵输入空间

- 例如：对抗样例（Adversarial example ）

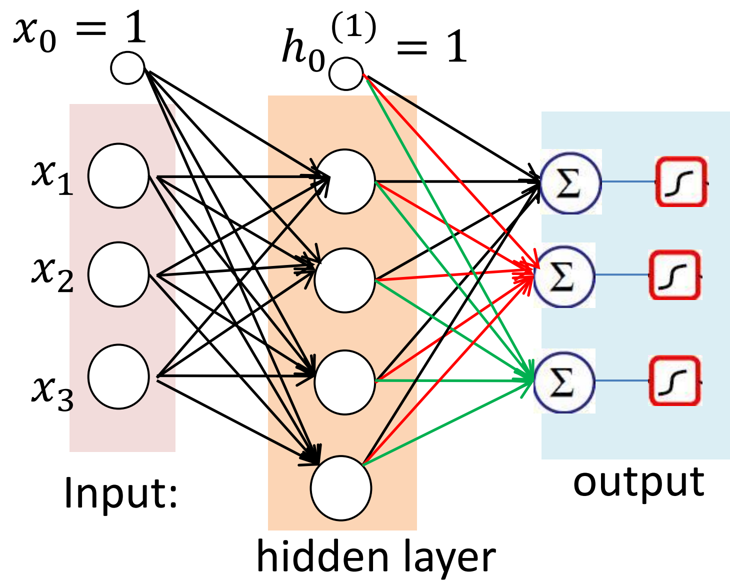


$f(X + \Delta X) \neq f(X)$, ΔX should be imperceptible by human



Some analyses

- Feature extraction
  - Pixel-wise input
  - High dimension
  - Correlation between features



$(1, 0, 0)^T$

(x_1, x_2, x_3)

Convolutional Neural Network(CNN),卷积神经网络