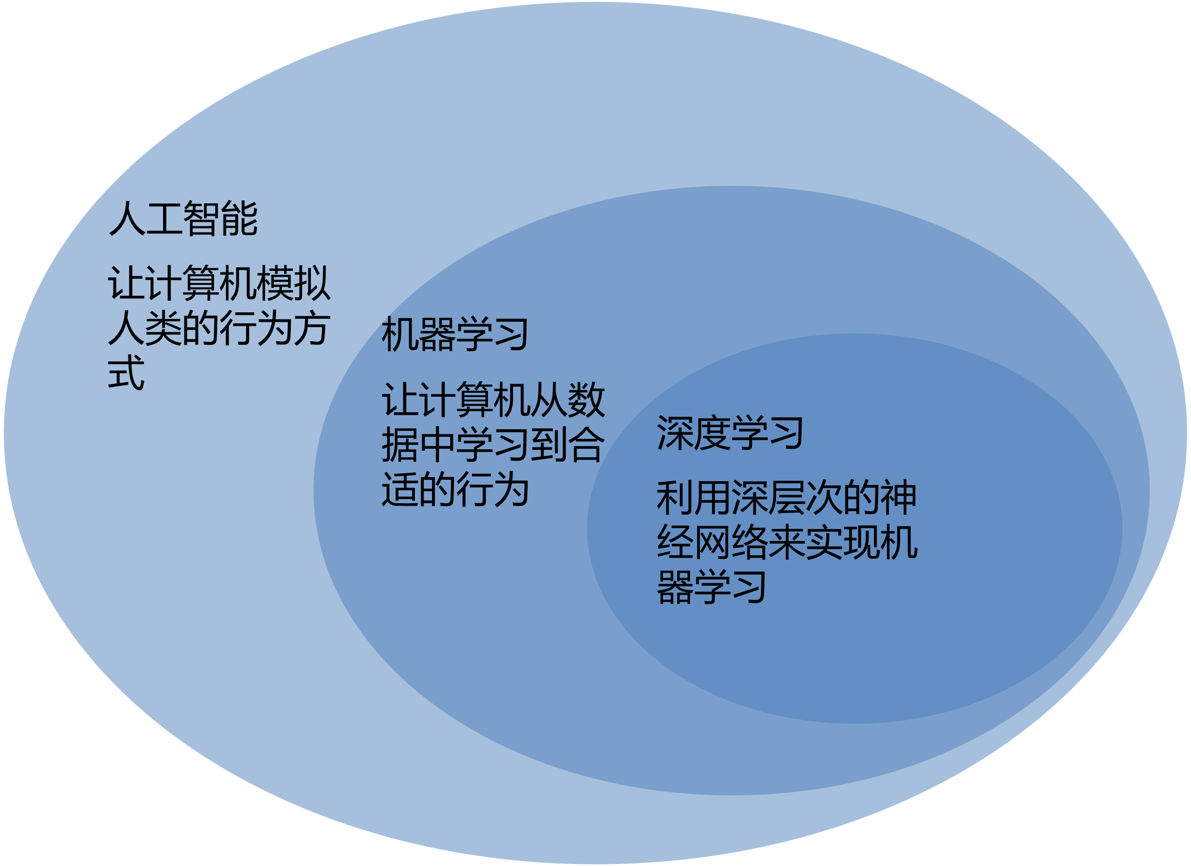
深蓝学院
shenlanxueyuan.com

深度学习基础

目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

什么是深度学习


人工智能
让计算机模拟人类的行为方式
机器学习
让计算机从数据中学习到合适的行为
深度学习
利用深层次的神经网络来实现机器学习

深度学习简史

1943

人工神经元

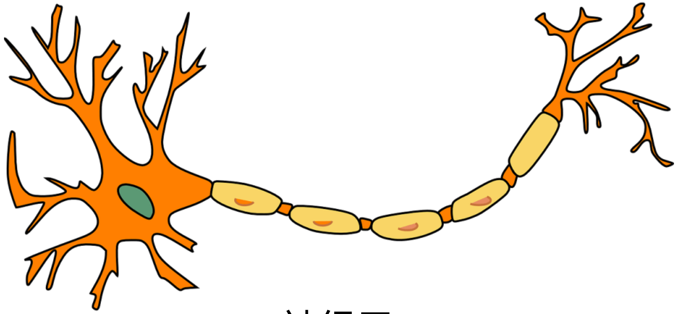
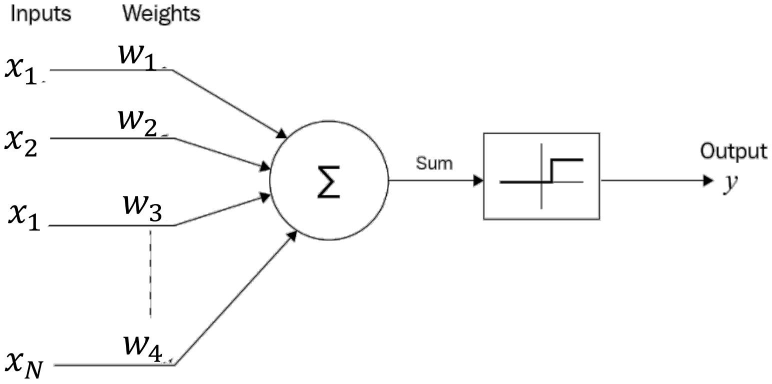


Image credit

神经元



McCulloch-Pitts 神经元

深度学习简史

1943

人工神经元

1958

感知机

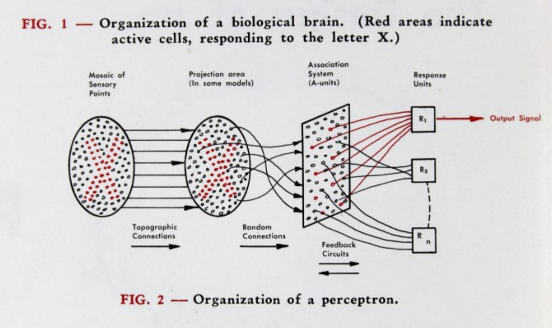


Image credit

感知机（Perceptron）

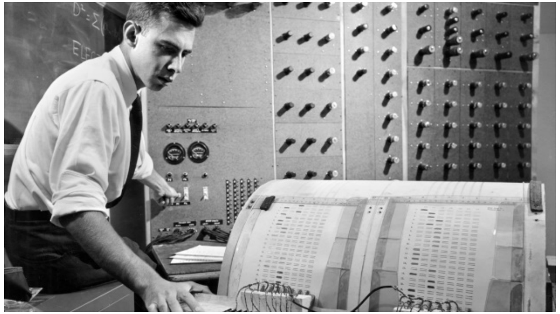
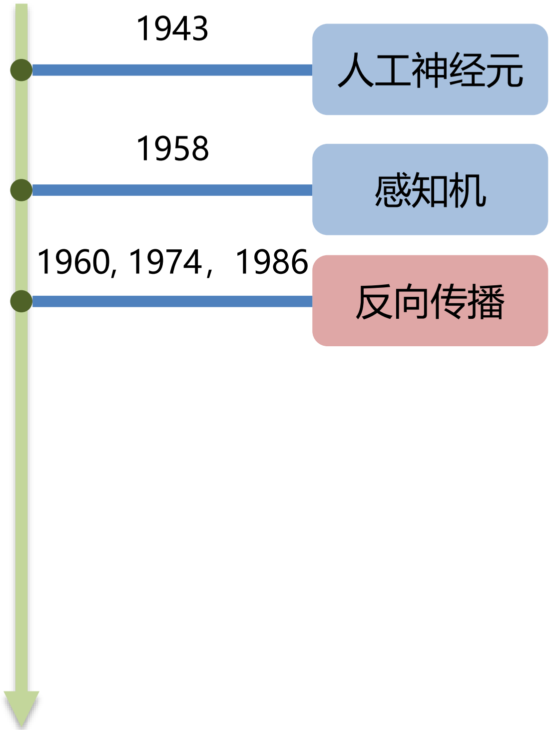


Image credit

Frank Rosenblatt

深度学习简史



Gradient Theory of Optimal Flight Paths

HENRY J. KELLEY[1]

Grumman Aircraft Engineering Corp.
Bethpage, N. Y.

An analytical development of flight performance optimization according to the method of gradients or "method of steepest descent" is presented. Construction of a minimizing sequence of flight paths by a stepwise process of descent along the local gradient direction is described as a computational scheme. Numerical application of the technique is illustrated in a simple example of orbital transfer via solar sail propulsion. Successive approximations to minimum time planar flight paths from Earth's orbit to the orbit of Mars are presented for cases corresponding to free and fixed boundary conditions on terminal velocity components.

Beyond regression : new tools for prediction and analysis in the behavioral sciences

Author:	Paul J Werbos
Dissertation:	Ph. D. Harvard University 1975
Edition/Format:	Thesis/dissertation : Thesis/dissertation : English View all editions and formats

Learning representations by back-propagating errors

David E. Rumelhart*, Geoffrey E. Hinton†
& Ronald J. Williams*

* Institute for Cognitive Science, C-015, University of California, San Diego, La Jolla, California 92093, USA
† Department of Computer Science, Carnegie-Mellon University, Pittsburgh, Philadelphia 15213, USA

深度学习简史

- 1943 人工神经元
- 1958 感知机
- 1960, 1974，1986 反向传播
- 1980~2000 CNN, RNN, LSTM

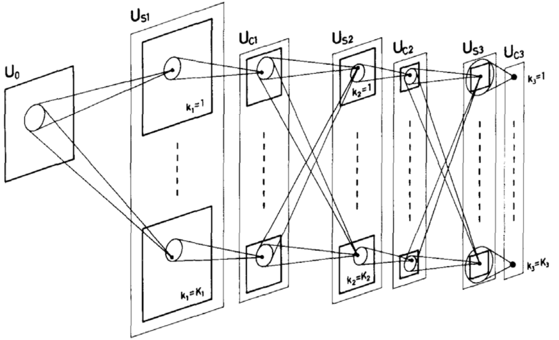


Image credit

Neocognitron,1980 (CNN的雏形)

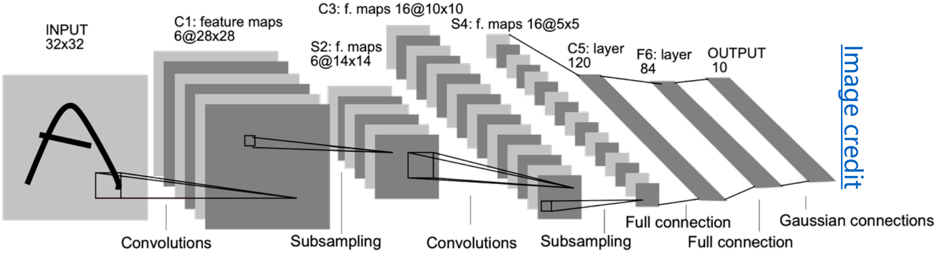


Image credit

LeNet-5, 1989

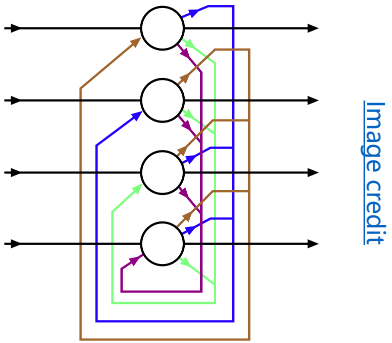
深度学习简史

1943 人工神经元

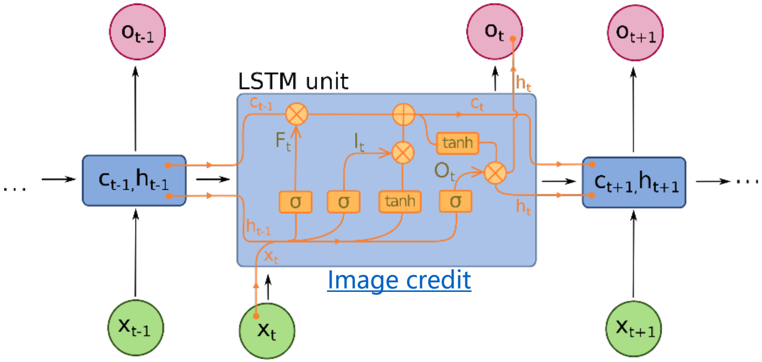
1958 感知机

1960, 1974， 1986 反向传播

1980~2000 CNN, RNN, LSTM

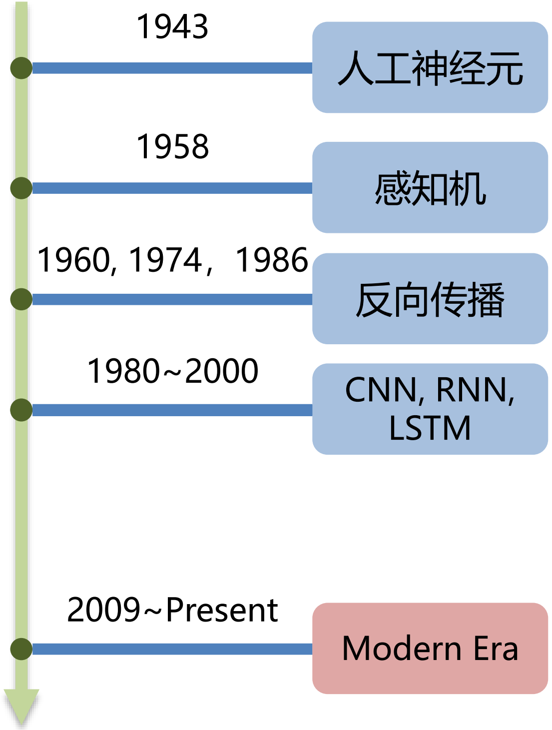


早期的RNN: Hopfield Network, 1982



LSTM,1997

深度学习简史



这些技术已经存在数十年，
但是深度学习直到近些年
才真正受到大家的关注。

深度学习简史

2009-

大数据

Image credit

ImageNet 数据集, 2009

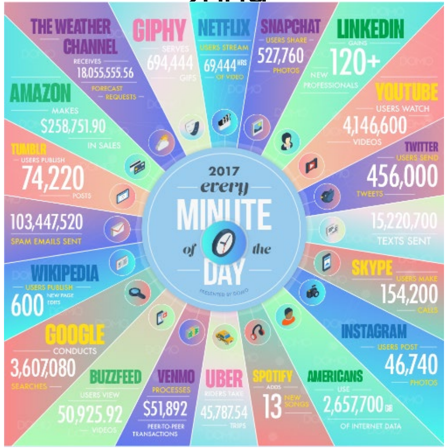


image credit

我们每分每秒都在产生大量的数据

深度学习简史

2009-

大数据

2009-

计算资源

Image credit

GPU开始被用来训练神经网络

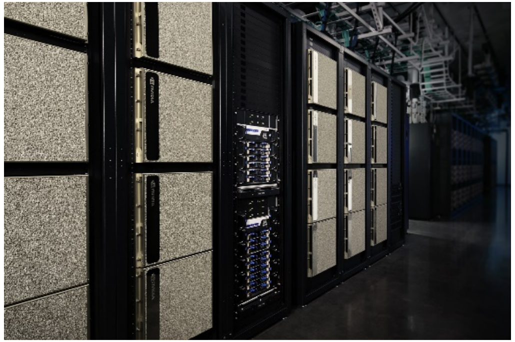


Image credit

GPU集群

深度学习简史

2009- 大数据

2009- 计算资源

2012 AlexNet

ImageNet Large Scale Visual Recognition Challenge

深度学习简史

2009- 大数据

2009- 计算资源

2012 AlexNet

2014 GANs

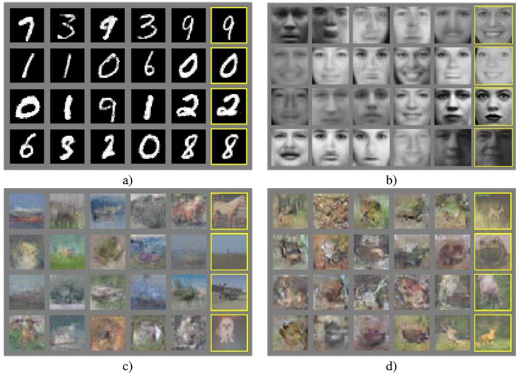


Image credit

由GANs生成的图片

Image credit

基于GANs的风格转换

深度学习简史

2009-	大数据
2009-	计算资源
2012	AlexNet
2014	GANs
2016	AlphaGo

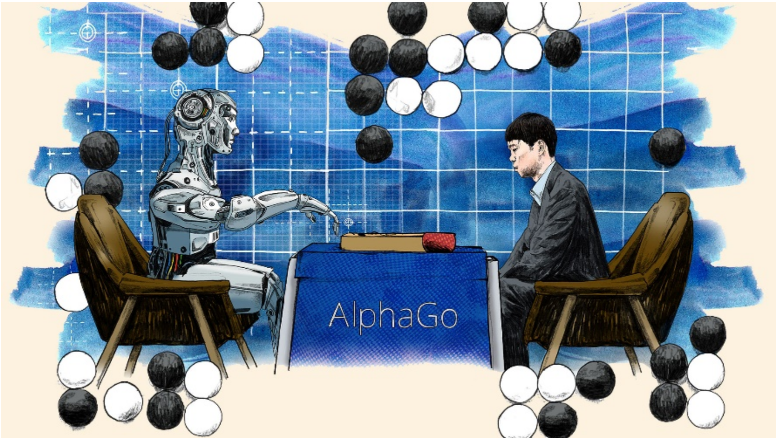
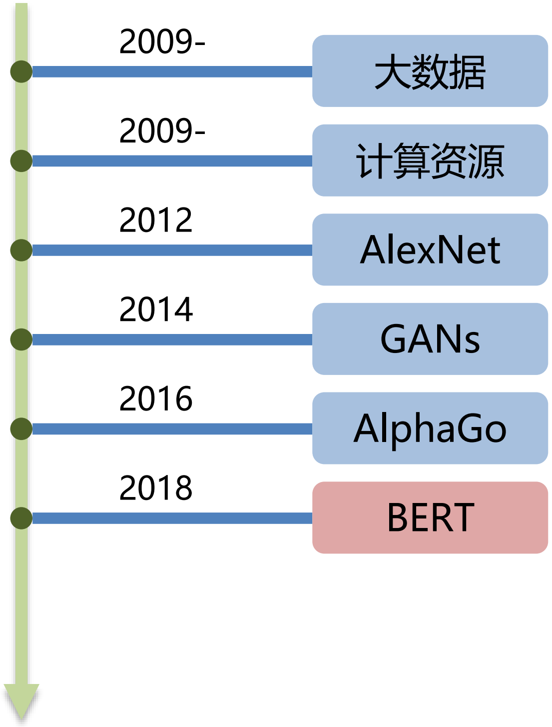


Image credit

AlphaGo 战胜最顶尖的人类围棋手

深度学习简史



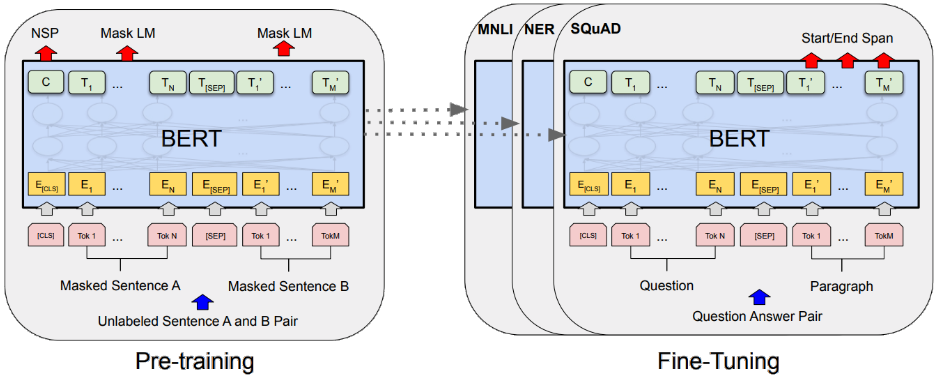


Image credit

System	MNLI-(m/mm)	QQP	QNLI	SST-2	CoLA	STS-B	MRPC	RTE	**Average**
	392k	363k	108k	67k	8.5k	5.7k	3.5k	2.5k	-
Pre-OpenAI SOTA	80.6/80.1	66.1	82.3	93.2	35.0	81.0	86.0	61.7	74.0
BiLSTM+ELMo+Attn	76.4/76.1	64.8	79.8	90.4	36.0	73.3	84.9	56.8	71.0
OpenAI GPT	82.1/81.4	70.3	87.4	91.3	45.4	80.0	82.3	56.0	75.1
$\text{BERT}_{\text{BASE}}$	84.6/83.4	71.2	90.5	93.5	52.1	85.8	88.9	66.4	79.6
$\text{BERT}_{\text{LARGE}}$	**86.7/85.9**	**72.1**	**92.7**	**94.9**	**60.5**	**86.5**	**89.3**	**70.1**	**82.1**

Table 1: GLUE Test results, scored by the evaluation server (`https://gluebenchmark.com/leaderboard`). The number below each task denotes the number of training examples. The "Average" column is slightly different than the official GLUE score, since we exclude the problematic WNLI set.[8] BERT and OpenAI GPT are single-model, single task. F1 scores are reported for QQP and MRPC, Spearman correlations are reported for STS-B, and accuracy scores are reported for the other tasks. We exclude entries that use BERT as one of their components.

深度学习简史

- 2009- 大数据
- 2009- 计算资源
- 2012 AlexNet
- 2014 GANs
- 2016 AlphaGo
- 2018 BERT
- 2018 图灵奖

Image credit

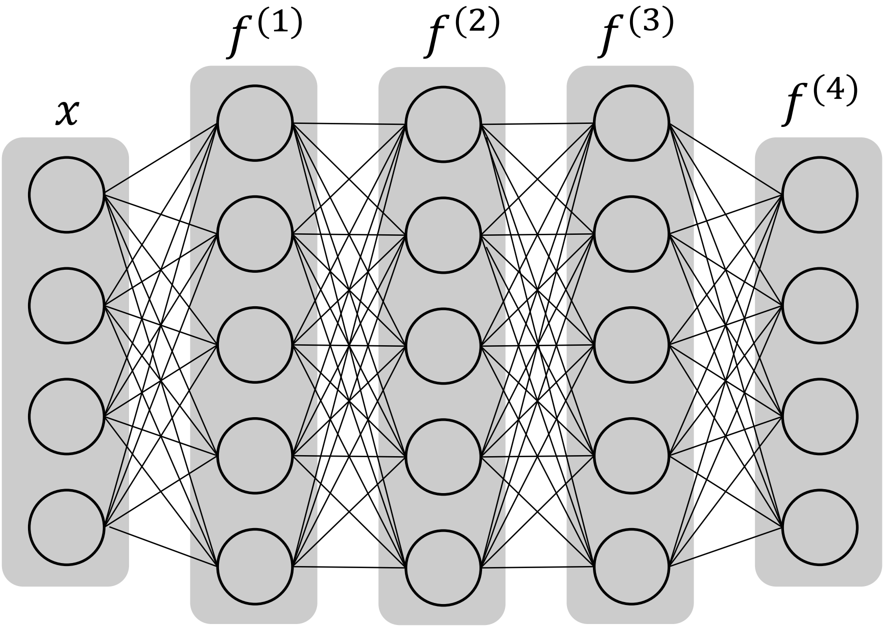
"Fathers of the Deep Learning Revolution"

ACM A.M. Turing Award

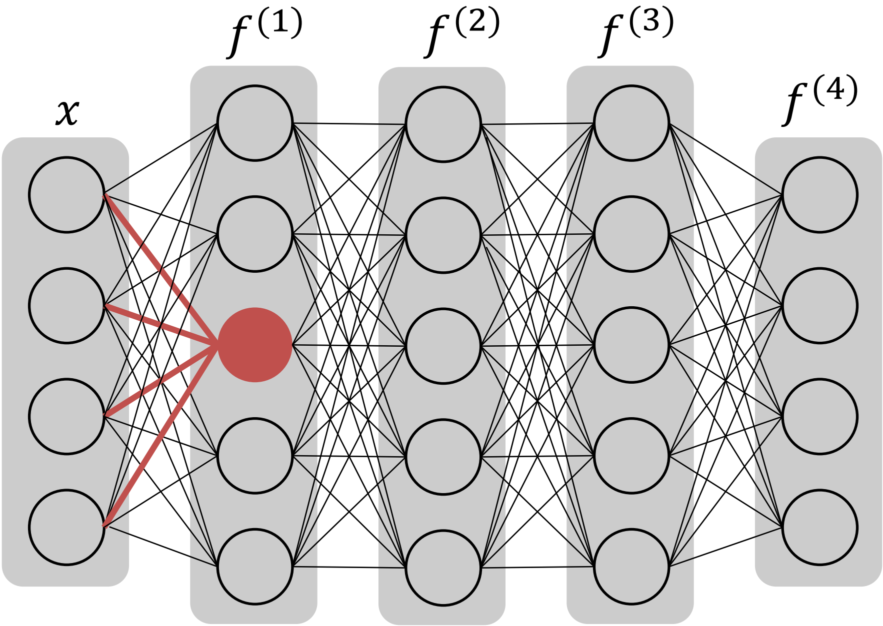
目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

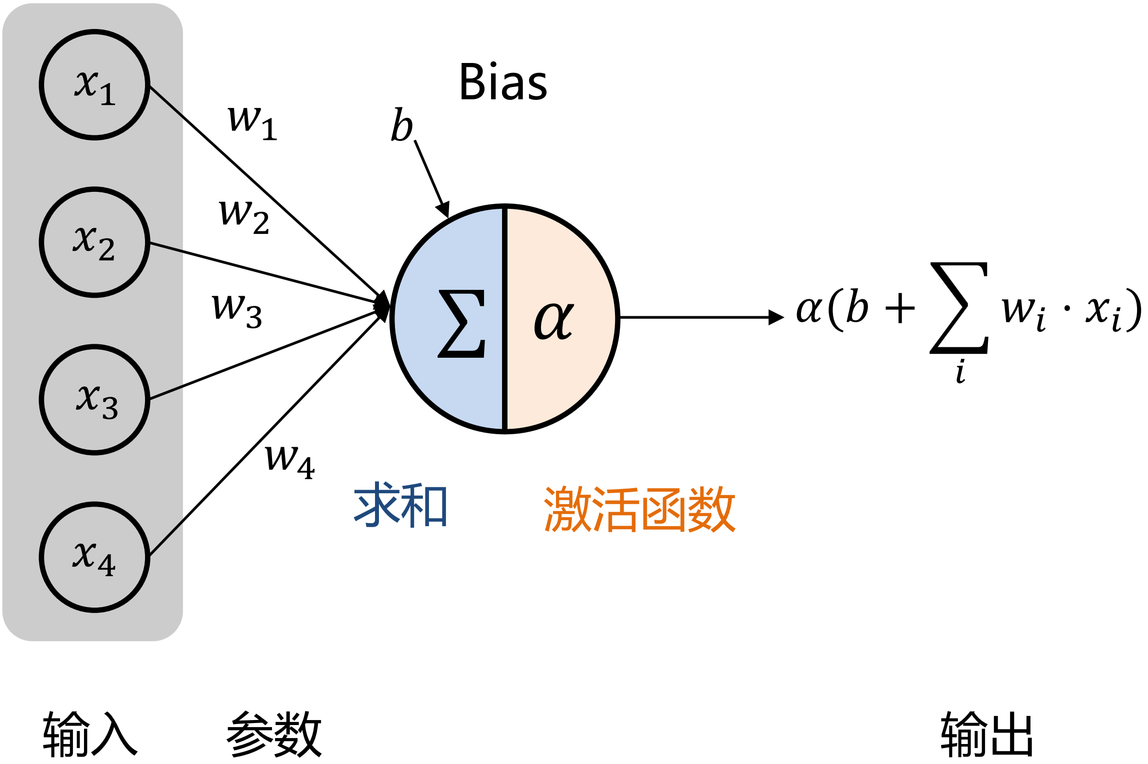
前馈神经网络



前馈神经网络

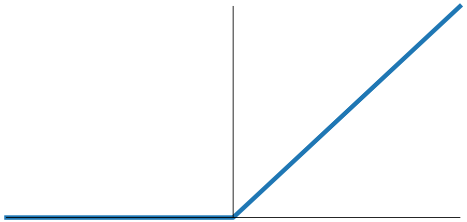


人工神经元

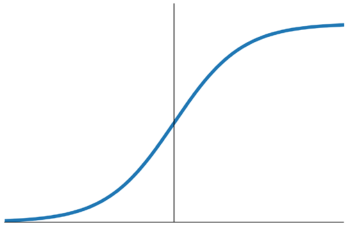


x_1
x_2
x_3
x_4
w_1
w_2
w_3
w_4
Bias
b
Σ
α
$\alpha(b + \sum_i w_i \cdot x_i)$
求和
激活函数
输入
参数
输出

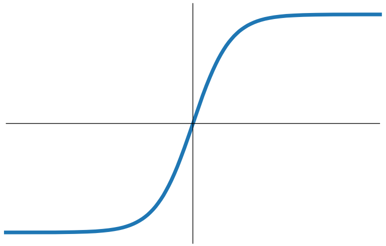
激活函数



$$\mathrm{ReLU}(z) = \max\{0, z\}$$

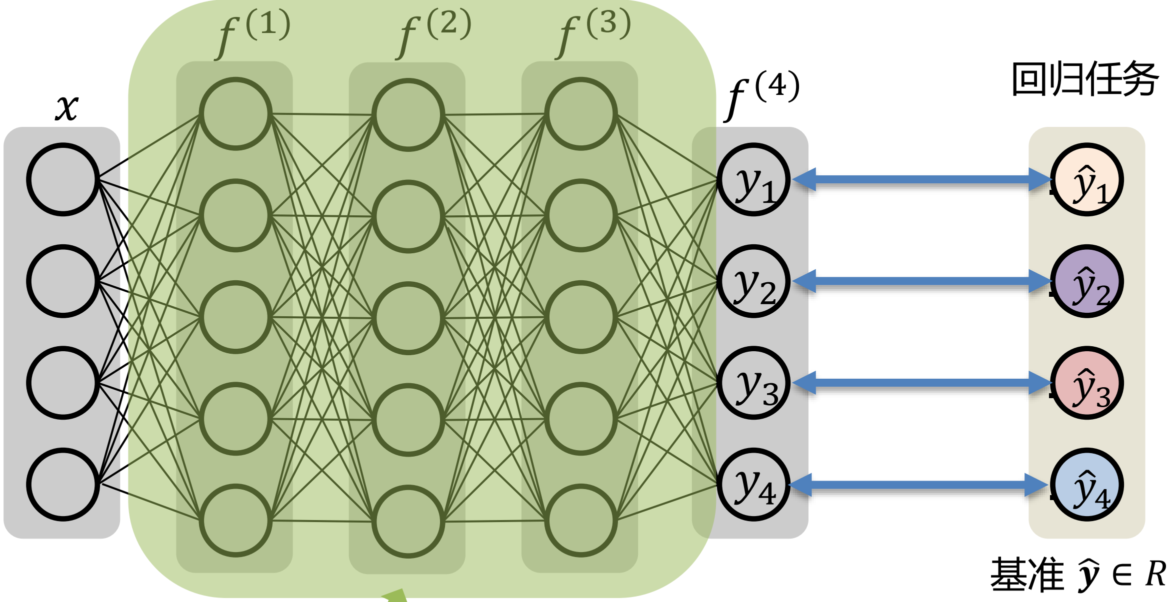


$$\sigma(z) = \frac{1}{1 + \exp(-z)}$$



$$\tanh(z) = \frac{2}{1 + \exp(-2z)} - 1$$

输出层以及损失函数



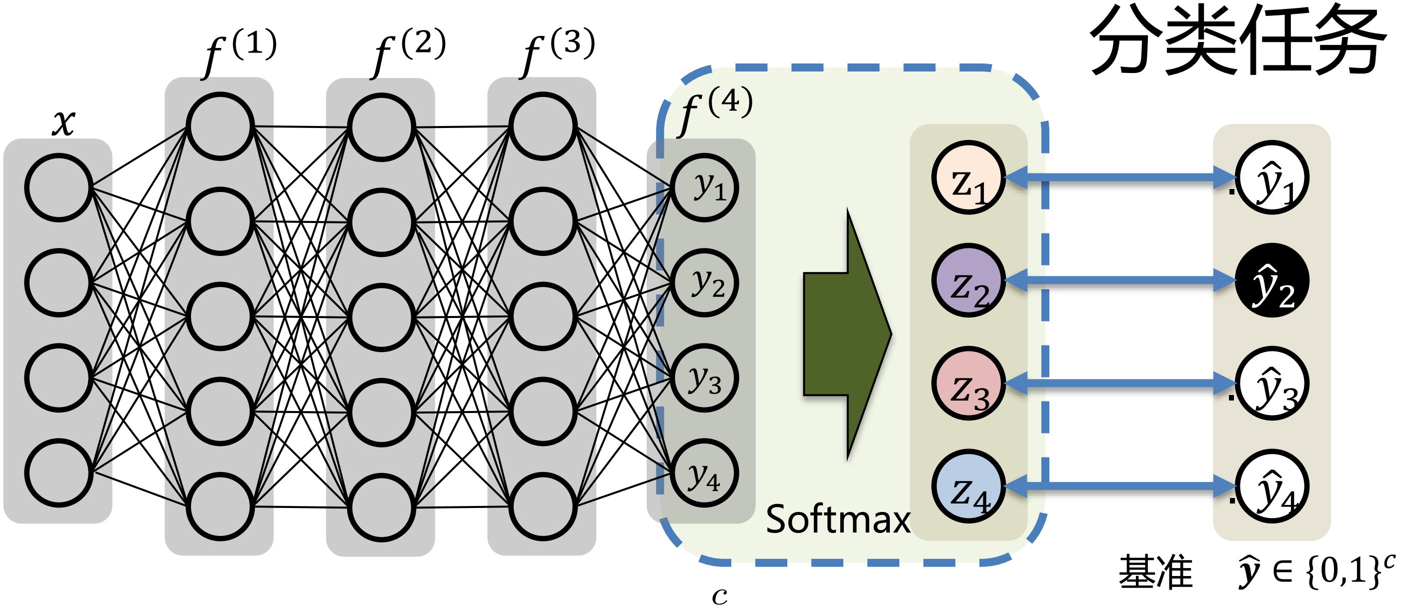
$$\ell(\mathbf{y}, \hat{\mathbf{y}}) = \|\mathbf{y} - \hat{\mathbf{y}}\|_2^2$$

损失函数

$$\mathcal{L}(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \ell(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)})$$

第i个训练样本

输出层以及损失函数



损失函数

$$\ell(\mathbf{z}, \hat{\mathbf{y}}) = -\sum_{j=1}^{c} \hat{\mathbf{y}}_j \log \mathbf{z}_j$$

$$\mathcal{L}(\mathbf{W}) = \frac{1}{n} \sum_{i=1}^{n} \ell\left(\mathbf{z}^{(i)}, \hat{\mathbf{y}}^{(i)}\right)$$

目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

优化的目标

$$\mathcal{L}(\mathbf{W}) = \frac{1}{n}\sum_{i=1}^{n} \ell\left(\mathbf{y}^{(i)}, \hat{\mathbf{y}}^{(i)}\right)$$

梯度下降法

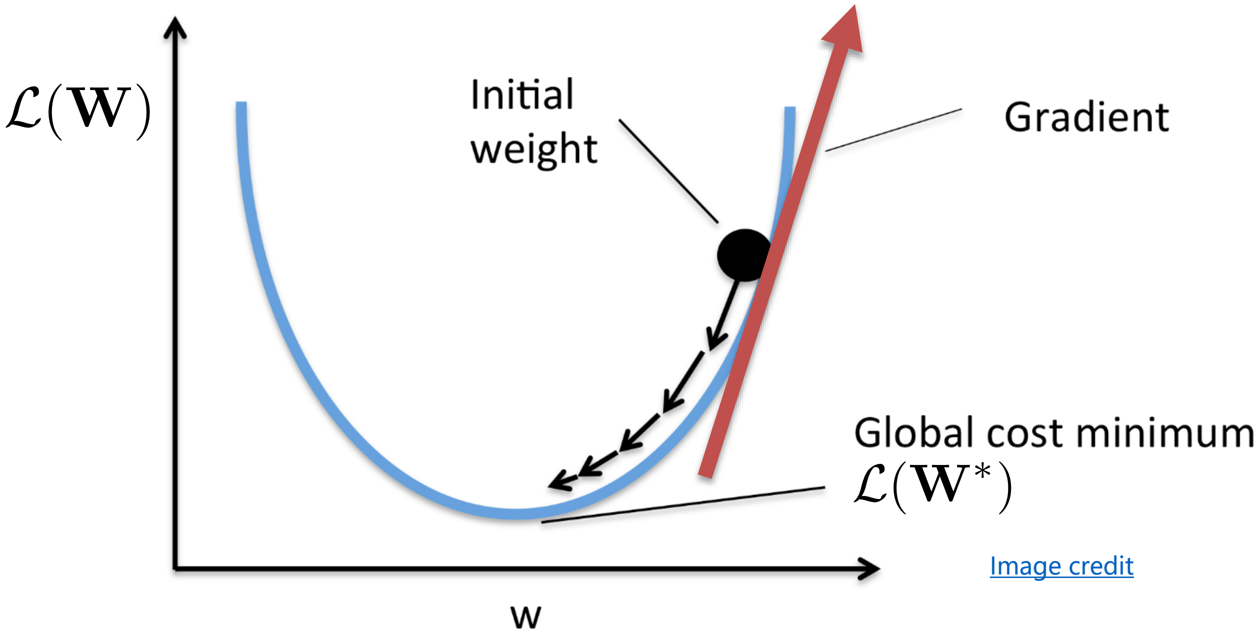


Image credit

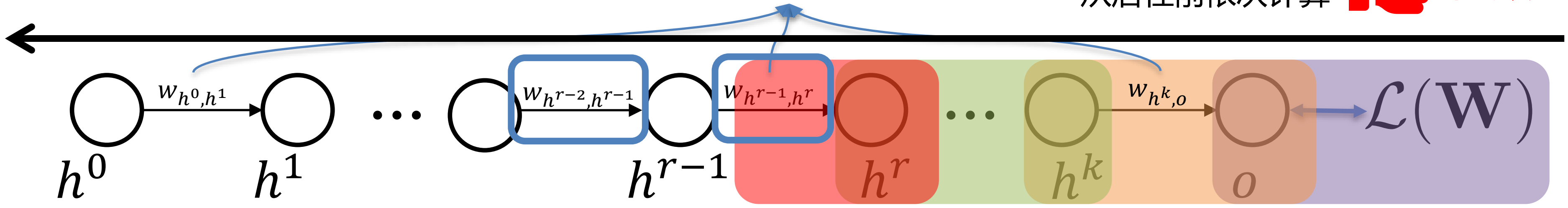
$$\mathbf{W}_{n+1} \leftarrow \mathbf{W}_n - \eta \nabla \mathcal{L}(\mathbf{W}_n)$$

反向传播

考虑每一层只有一个神经元的情况

参数

从后往前依次计算 高效



输入

输出

$$\frac{\partial \mathcal{L}}{\partial w_{(h^{r-1},h^r)}} = \frac{\partial \mathcal{L}}{\partial o} \cdot \left[\frac{\partial o}{\partial h^k} \prod_{i=r}^{k-1} \frac{\partial h^{i+1}}{\partial h^i} \right] \cdot \frac{\partial h^r}{\partial w_{(h^{r-1},h_r)}}$$

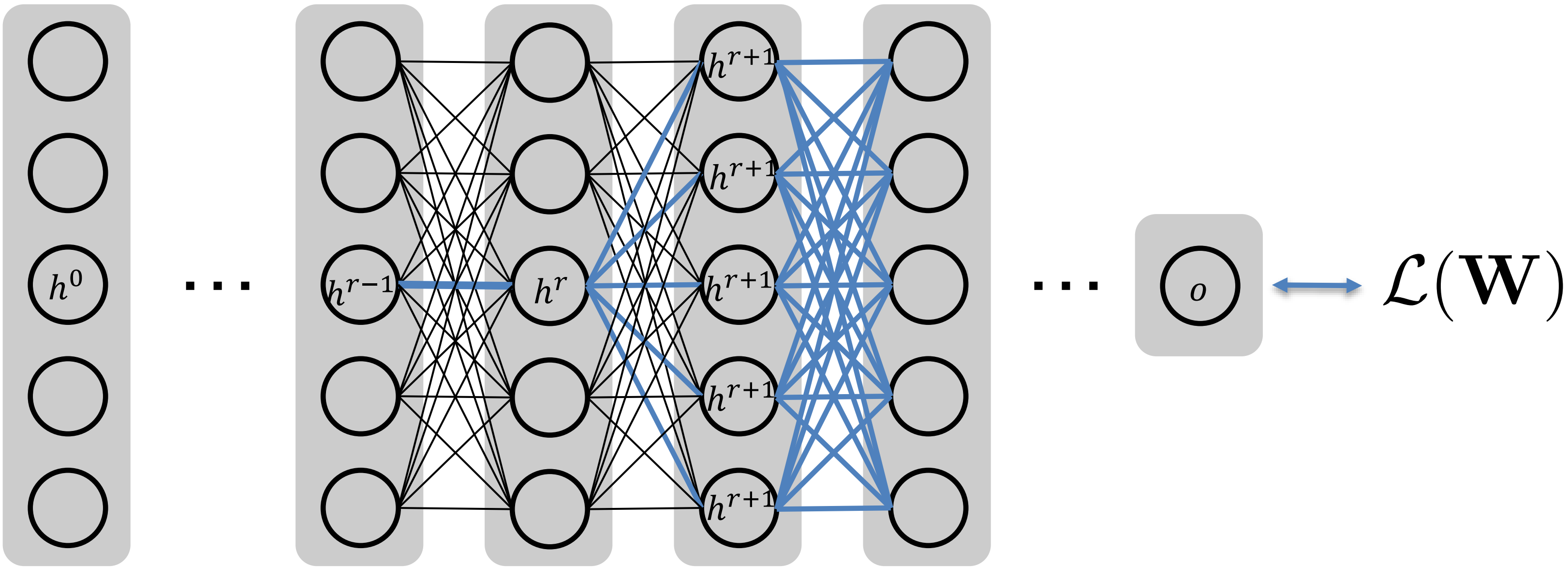
$$\frac{\partial \mathcal{L}}{\partial w_{(h^{r-2},h^{r-1})}} = \frac{\partial \mathcal{L}}{\partial o} \cdot \left[\frac{\partial o}{\partial h^k} \prod_{i=r}^{k-1} \frac{\partial h^{i+1}}{\partial h^i} \right] \cdot \frac{\partial h^r}{\partial h^{r-1}} \cdot \frac{\partial h^{r-1}}{\partial w_{(h^{r-2},h_{r-1})}}$$

不需要重复计算

需要额外计算

反向传播

考虑每一层都有多个神经元的情况

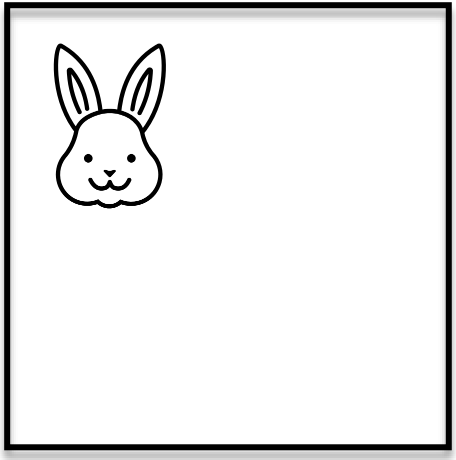


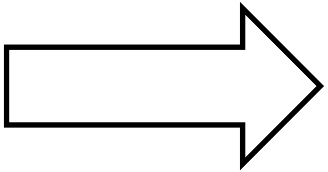
更为复杂，但是原理和单个神经元的情况类似

目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

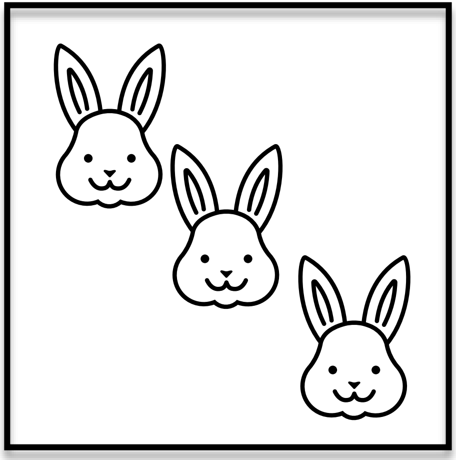
图像分类

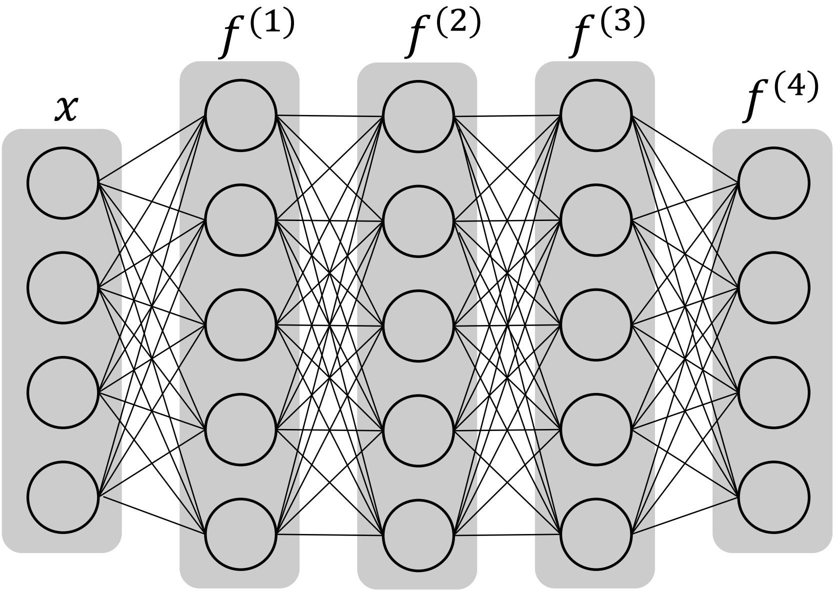




兔子?

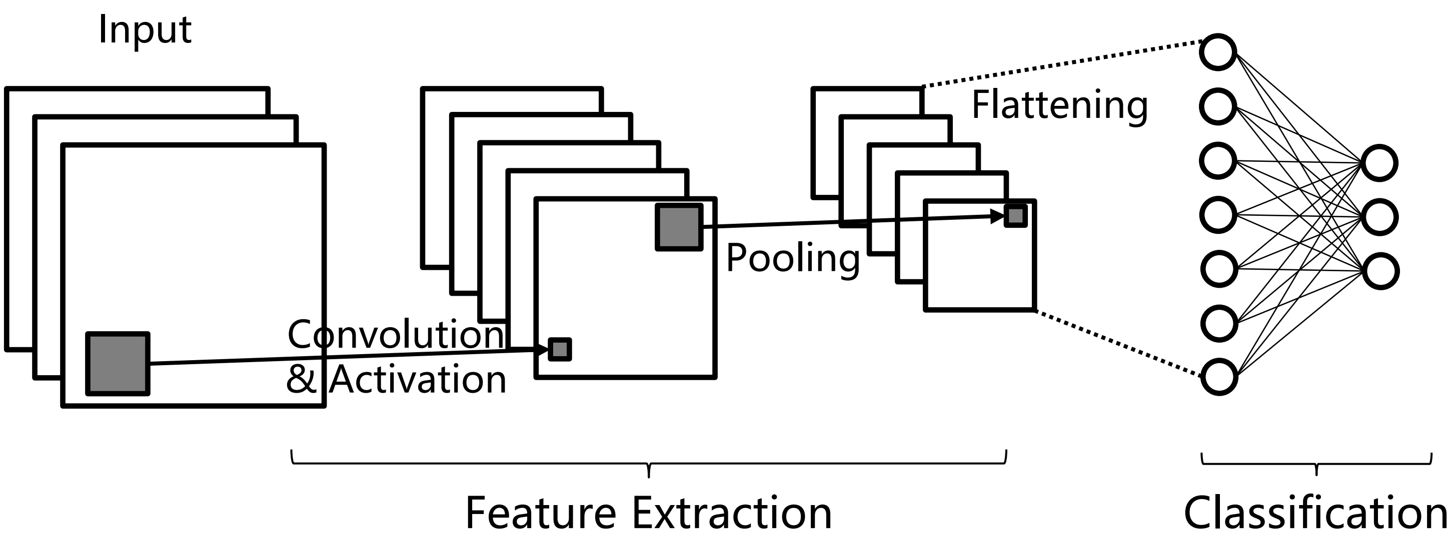
图像分类





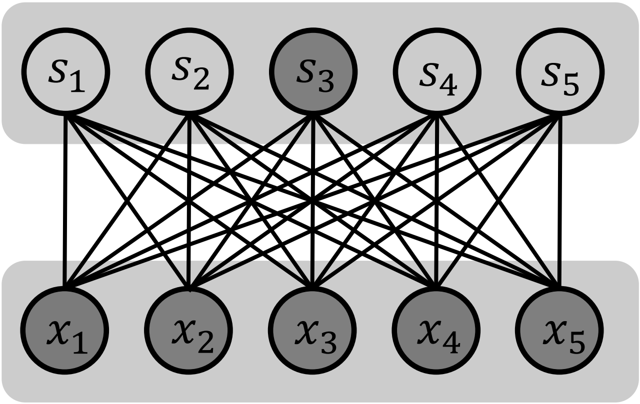
兔子?

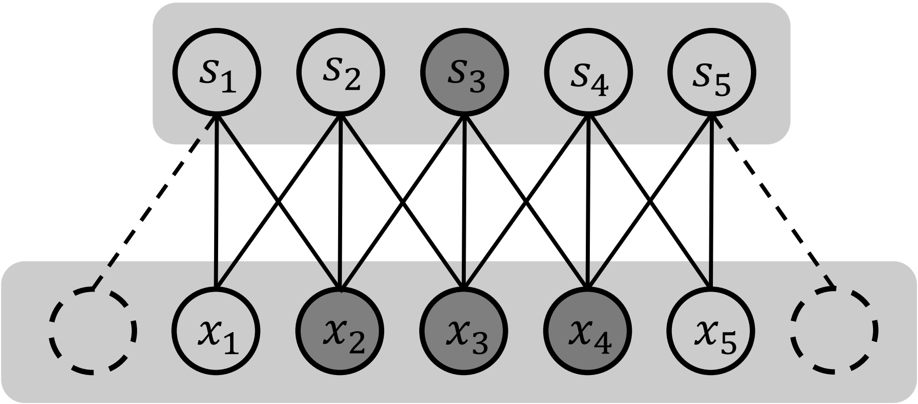
卷积神经网络



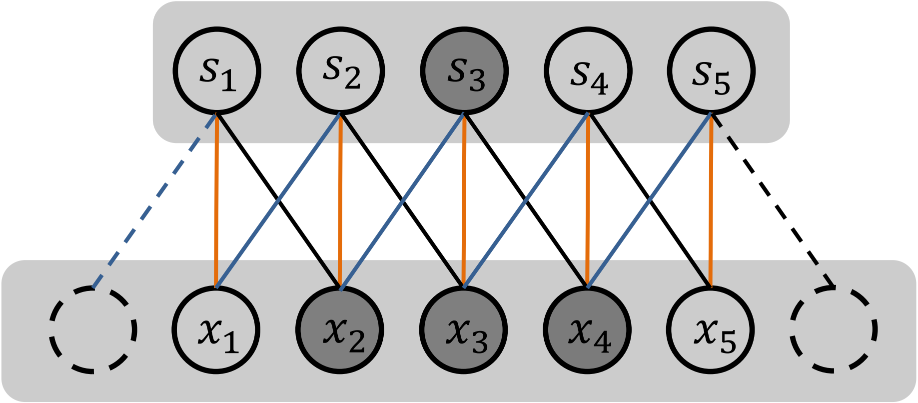
- ❑ 卷积操作
- ❑ 池化操作

卷积操作：稀疏连接

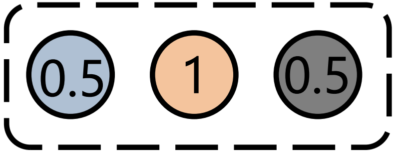




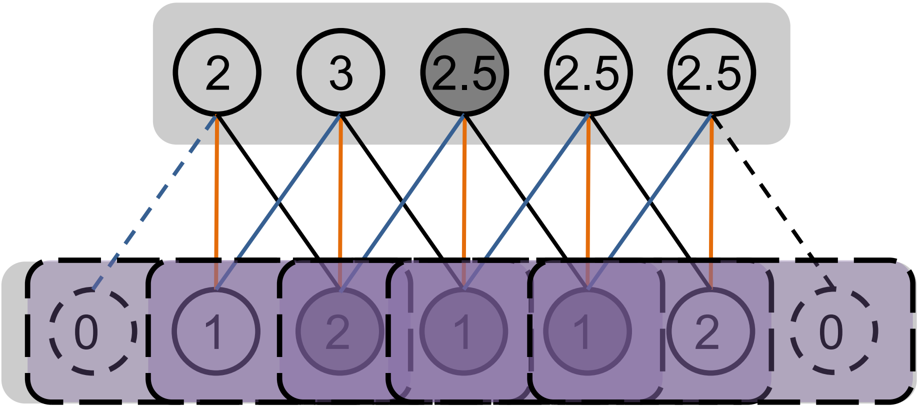
卷积操作：参数共享

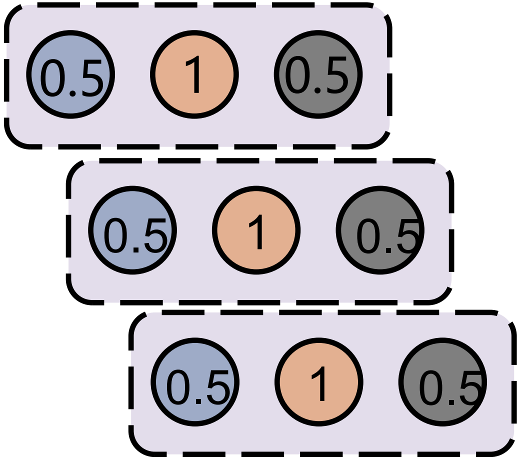


卷积核

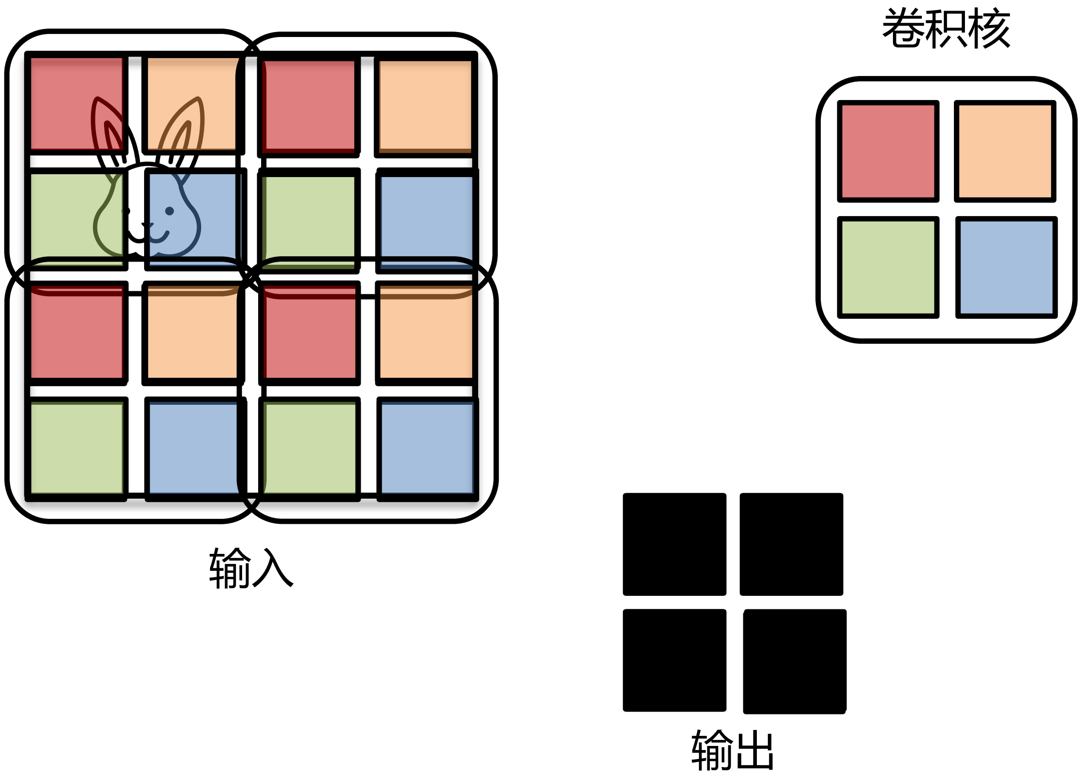


卷积操作

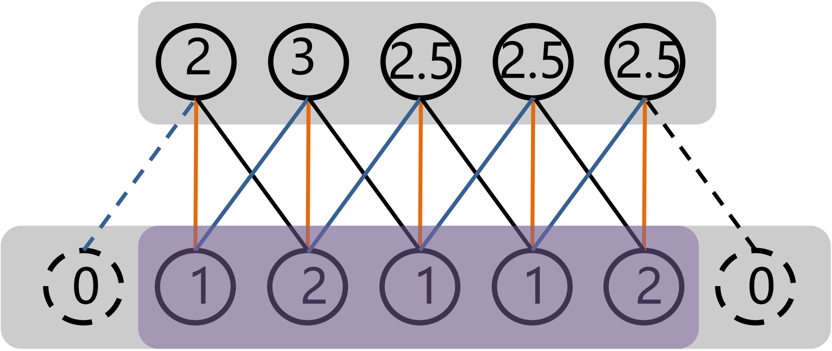


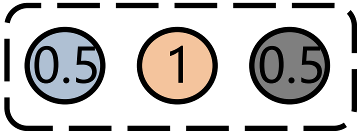


二维卷积操作

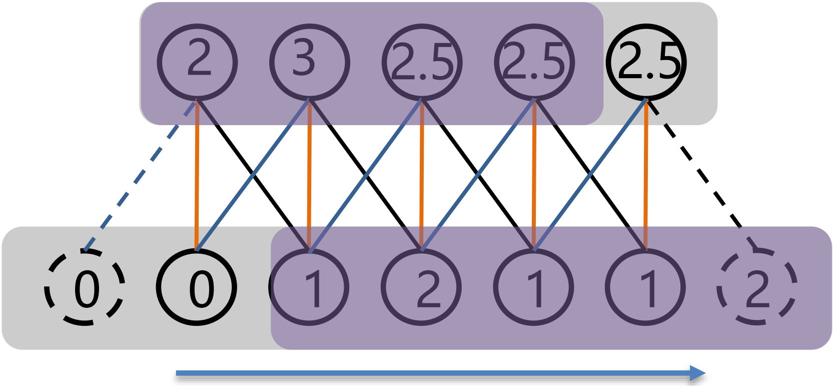


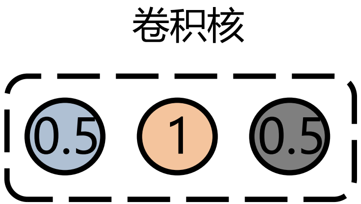
卷积操作：平移等变



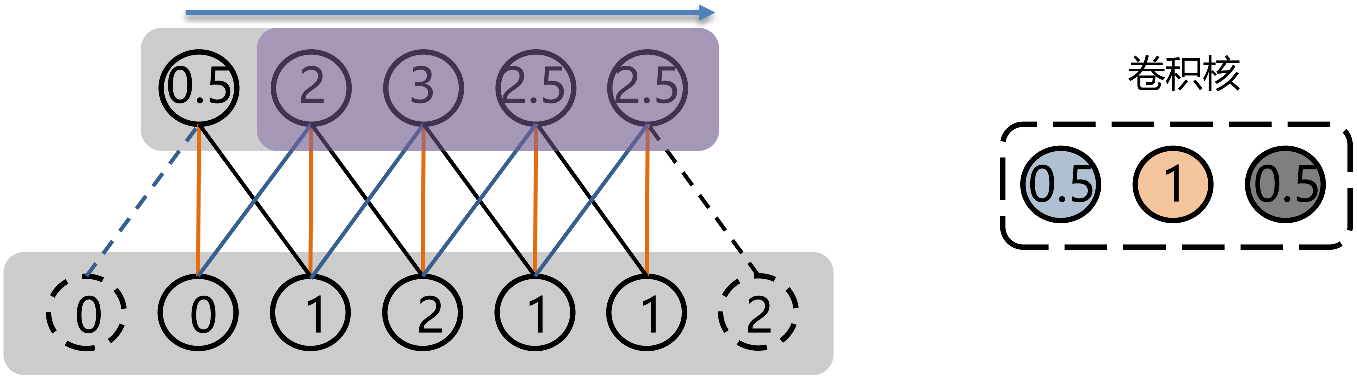


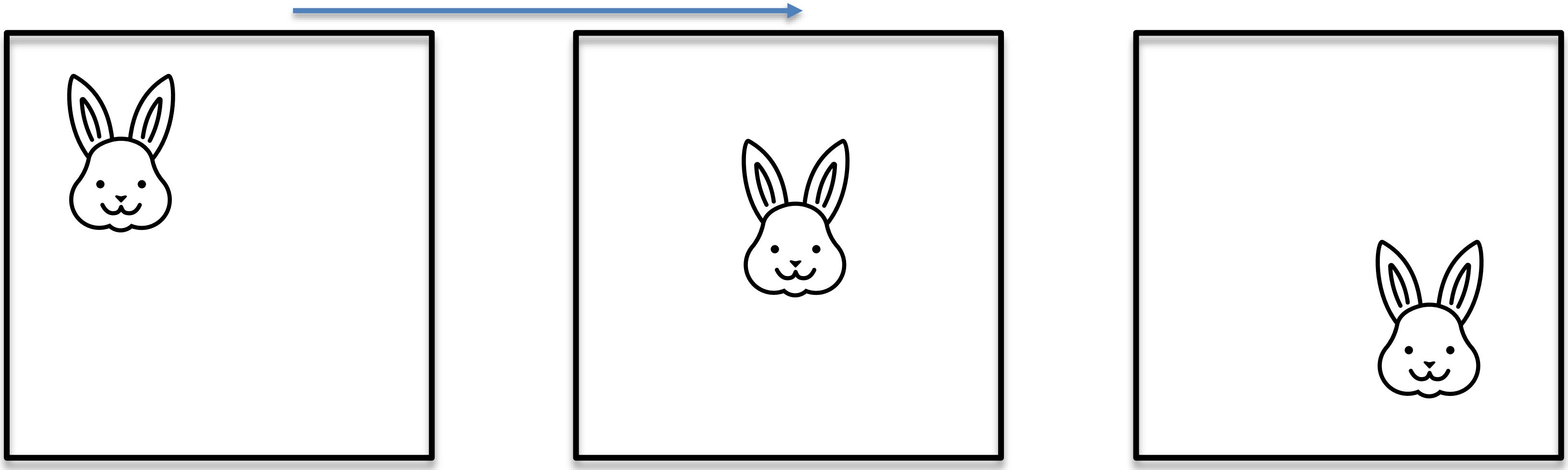
卷积操作：平移等变



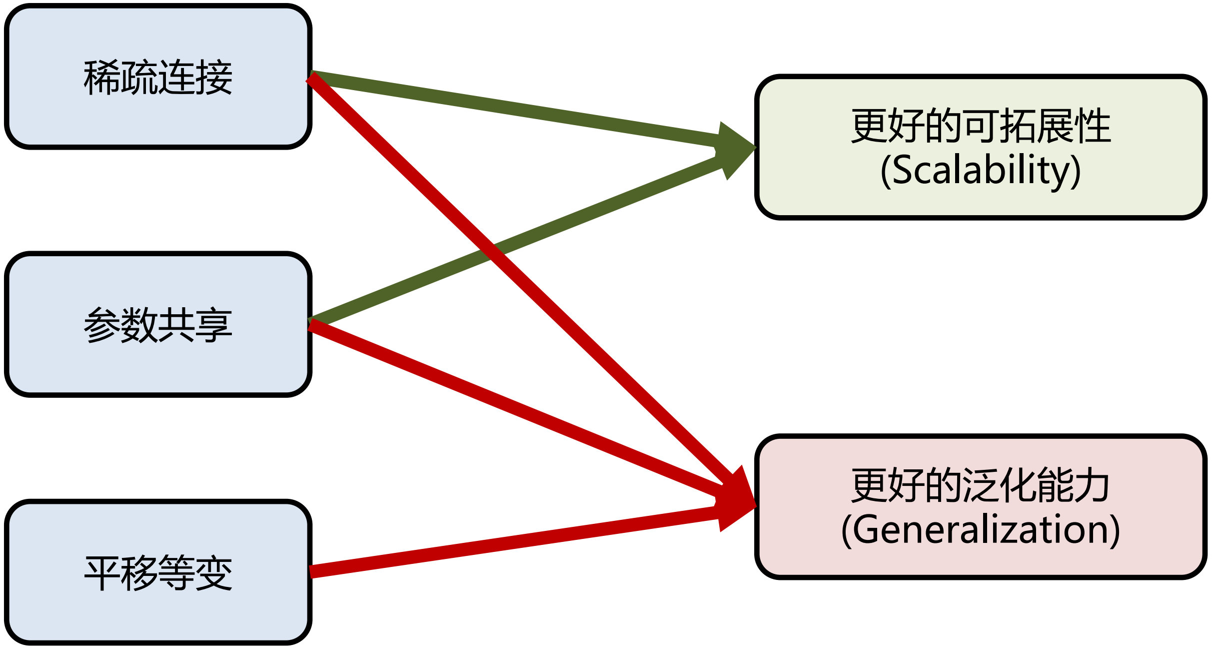


卷积操作：平移等变

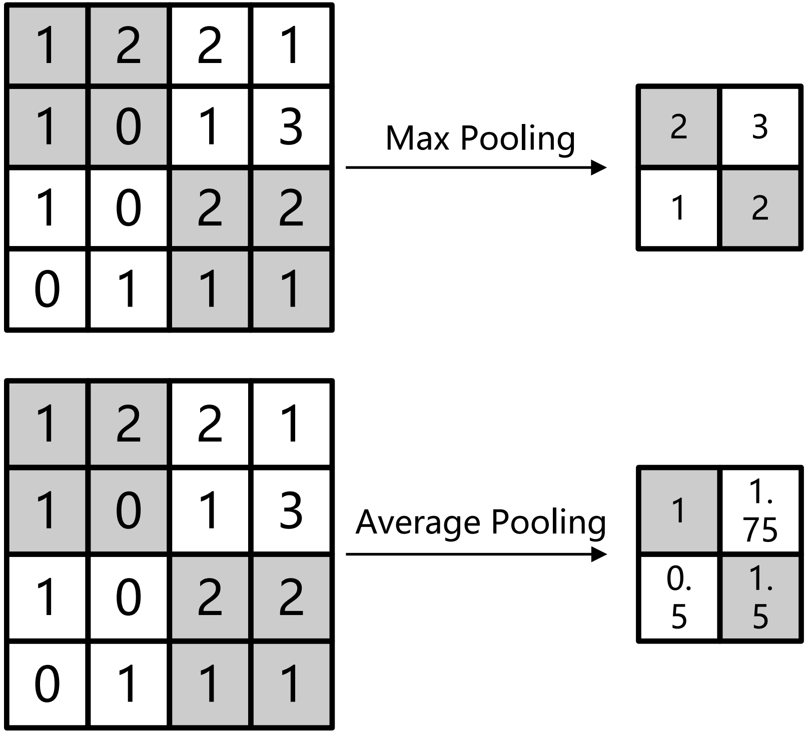




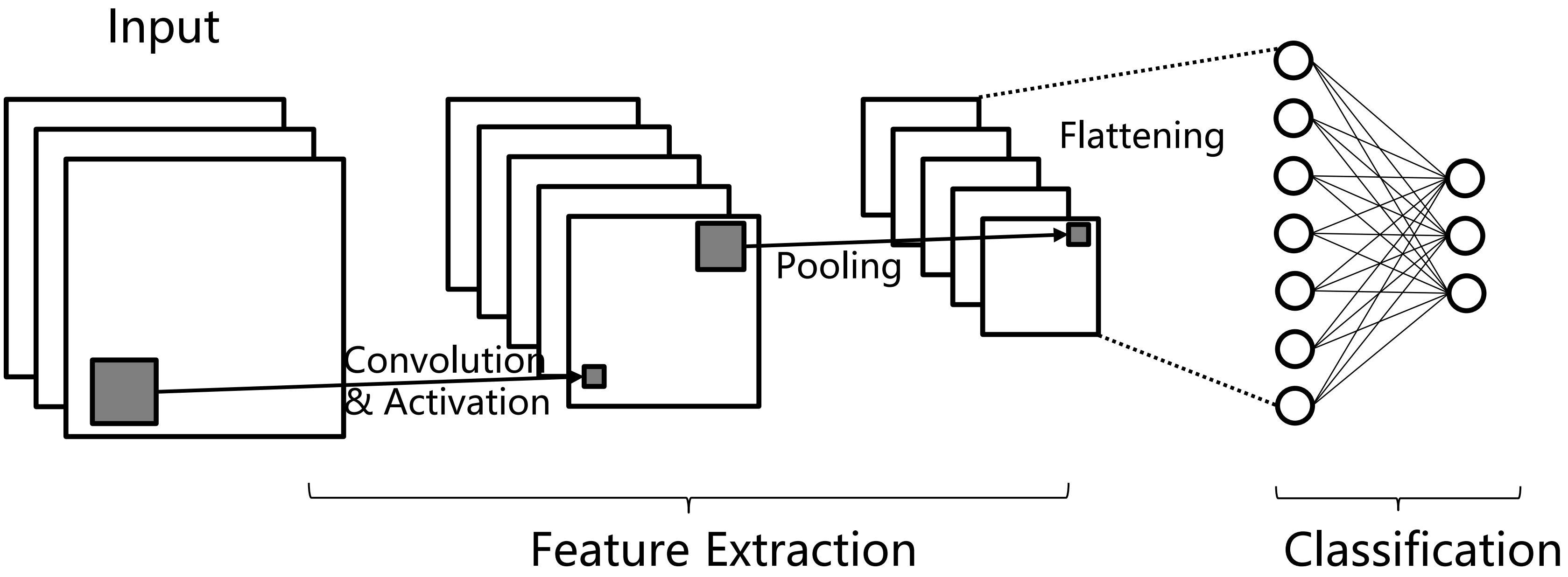
卷积操作的性质总结


稀疏连接
参数共享
平移等变
更好的可拓展性
(Scalability)
更好的泛化能力
(Generalization)

池化操作


1 2 2 1
1 0 1 3
1 0 2 2
0 1 1 1
Max Pooling
2 3
1 2
1 2 2 1
1 0 1 3
1 0 2 2
0 1 1 1
Average Pooling
1 1.75
0.5 1.5

卷积神经网络的整体框架



目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

序列数据

I went to Yellowstone National Park last summer...

Image credit

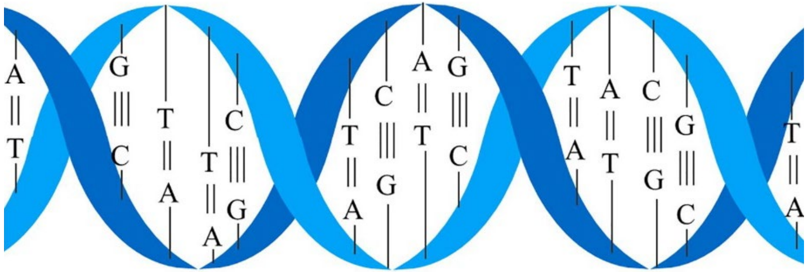
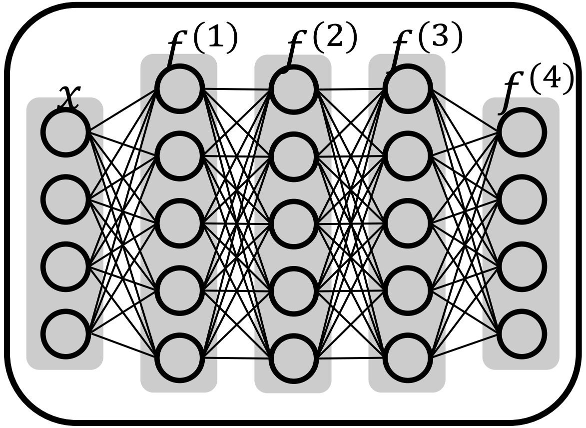


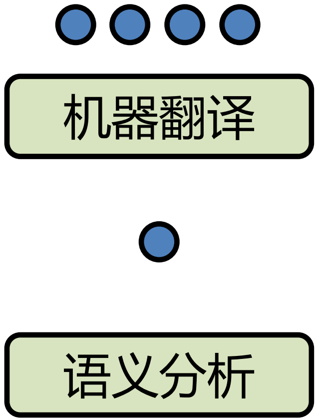
Image credit

序列数据上的任务

可以用前馈网络处理序列数据吗

序列可能长短不一





输入

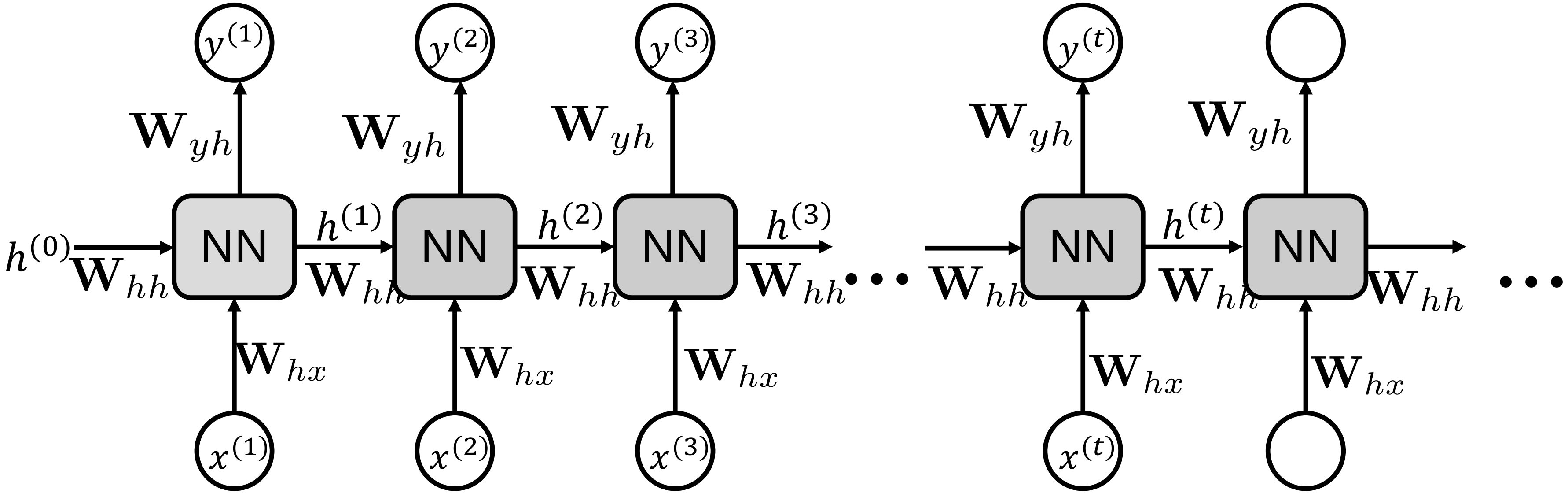
输出

同一信息可能出现在序列的不同位置

Last summer, I went to Yellowstone National Park.

I went to Yellowstone National Park last summer.

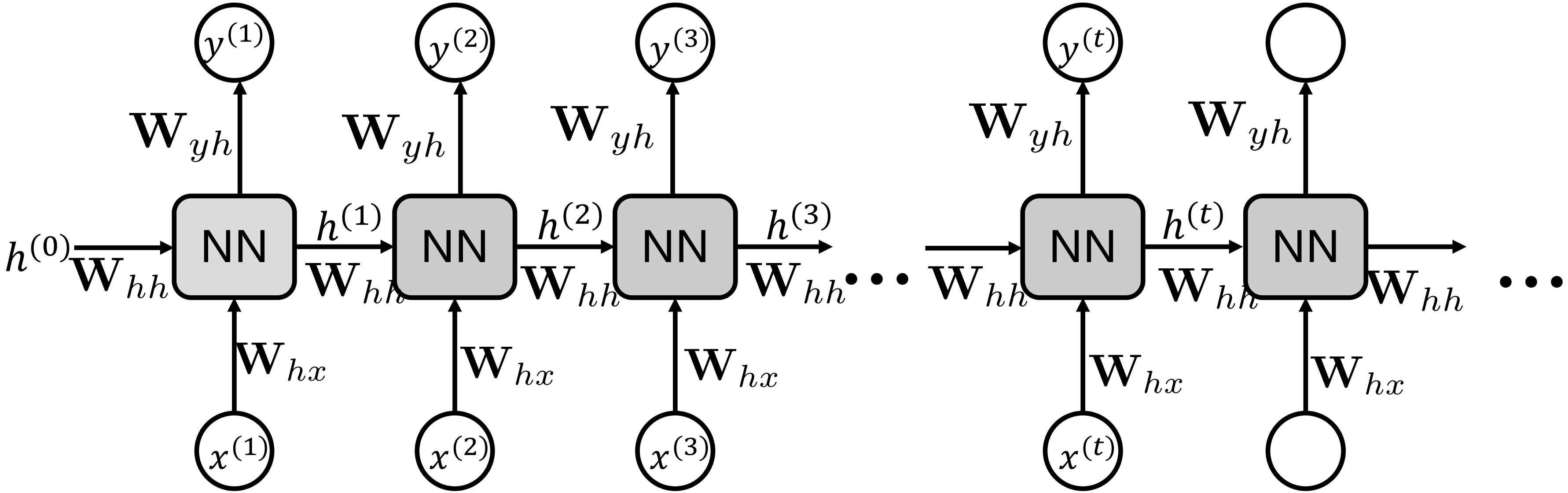
循环神经网络



$$\boldsymbol{h}^{(i)} = \alpha_h \left(\boldsymbol{W}_{\mathrm{hh}} \cdot \boldsymbol{h}^{(i-1)} + \boldsymbol{W}_{\mathrm{hx}} \boldsymbol{x}^{(i-1)} + \boldsymbol{b}_h \right)$$

$$\boldsymbol{y}^{(i)} = \alpha_y \left(\boldsymbol{W}_{\mathrm{yh}} \boldsymbol{h}^{(i)} + \boldsymbol{b}_y \right)$$

循环神经网络



Last summer, I went to Yellowstone National Park.

I went to Yellowstone National Park last summer.

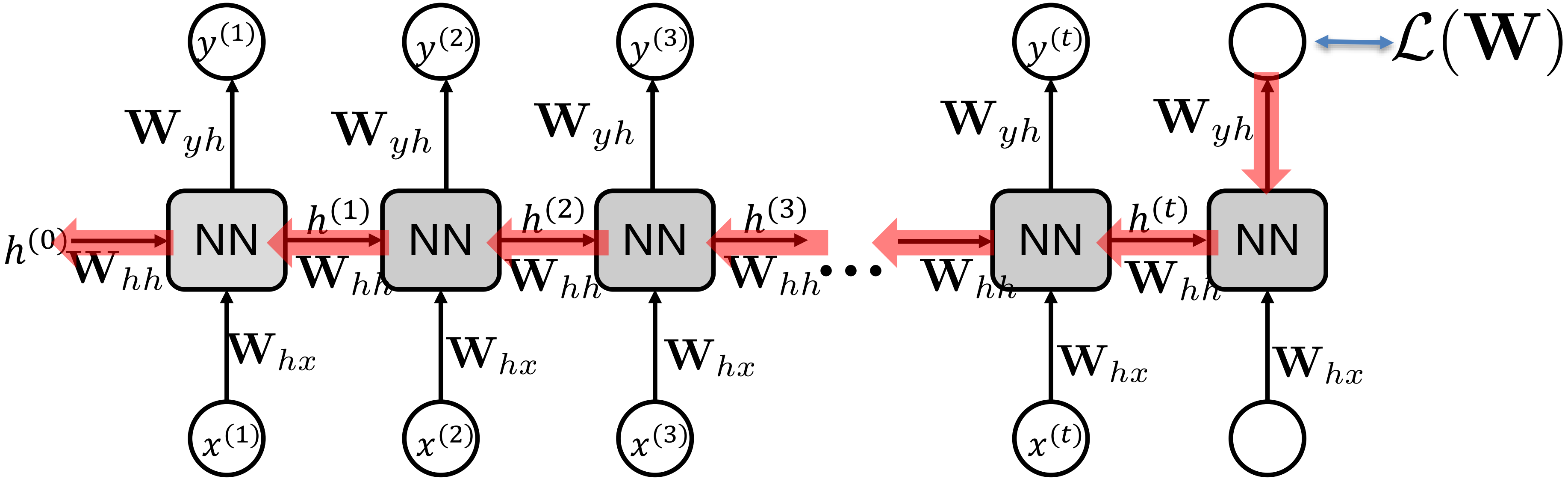
循环神经网络的性质

处理不同长度的序列

捕捉序列的顺序信息

参数共享

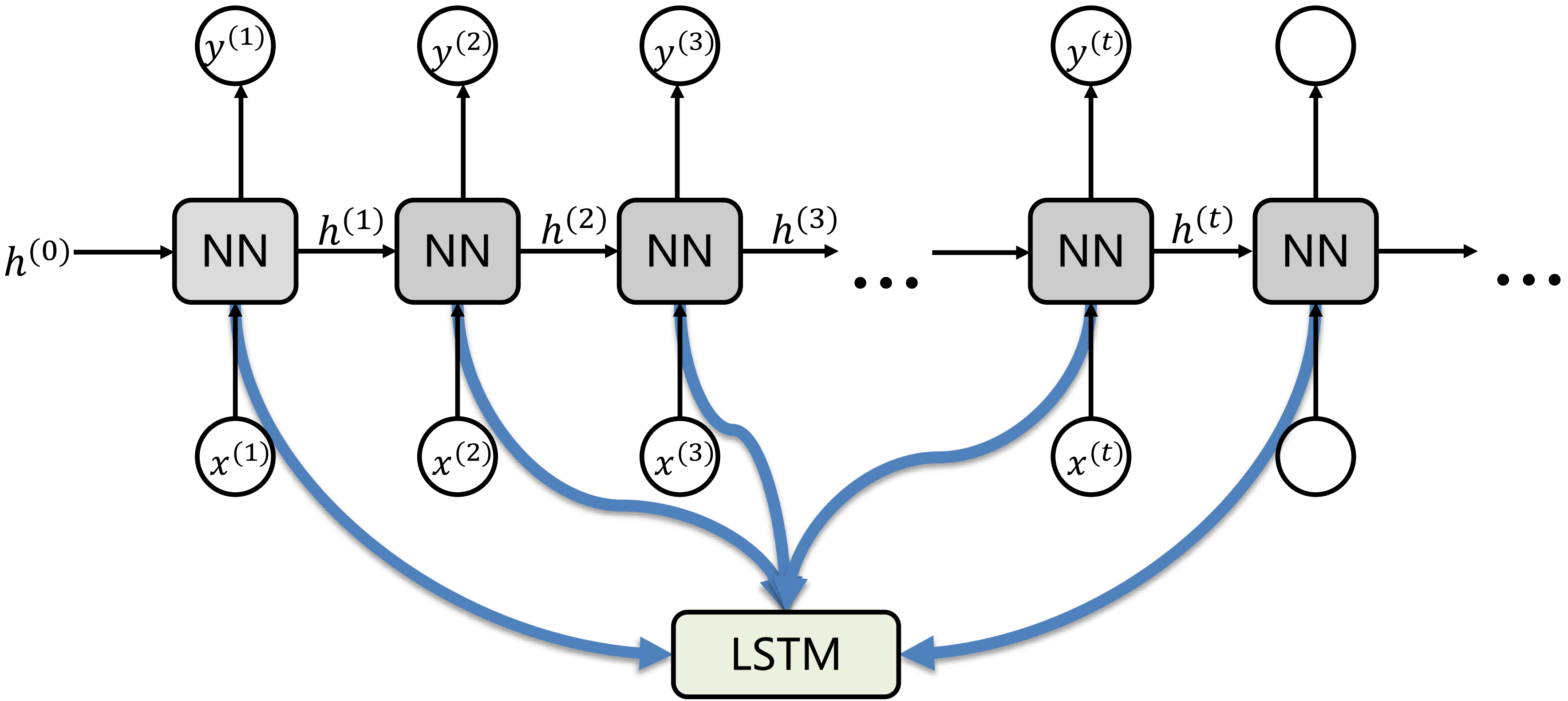
梯度消失与梯度爆炸



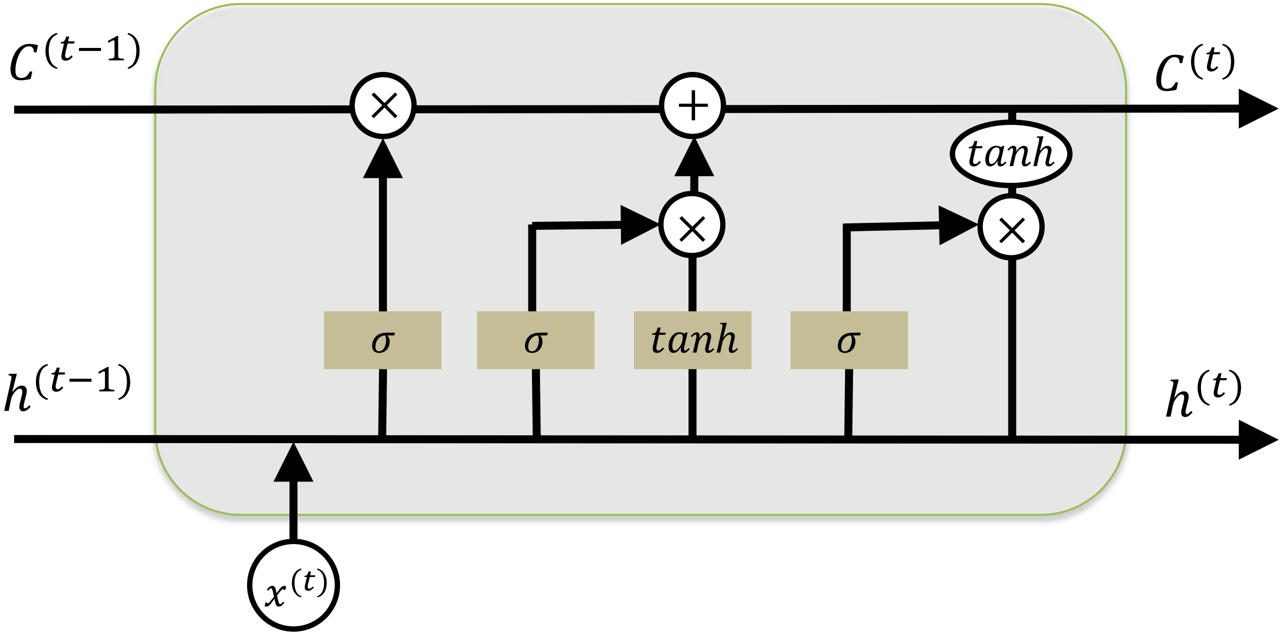
$$\boldsymbol{h}^{(i)} = \alpha_h \left(\boldsymbol{W}_{\text{hh}} \cdot \boldsymbol{h}^{(i-1)} + \boldsymbol{W}_{\text{hx}} \boldsymbol{x}^{(i-1)} + \boldsymbol{b}_h \right)$$

算梯度的过程涉及到了W_{hh}的连续相乘

Long Shor Term Meory (LSTM)



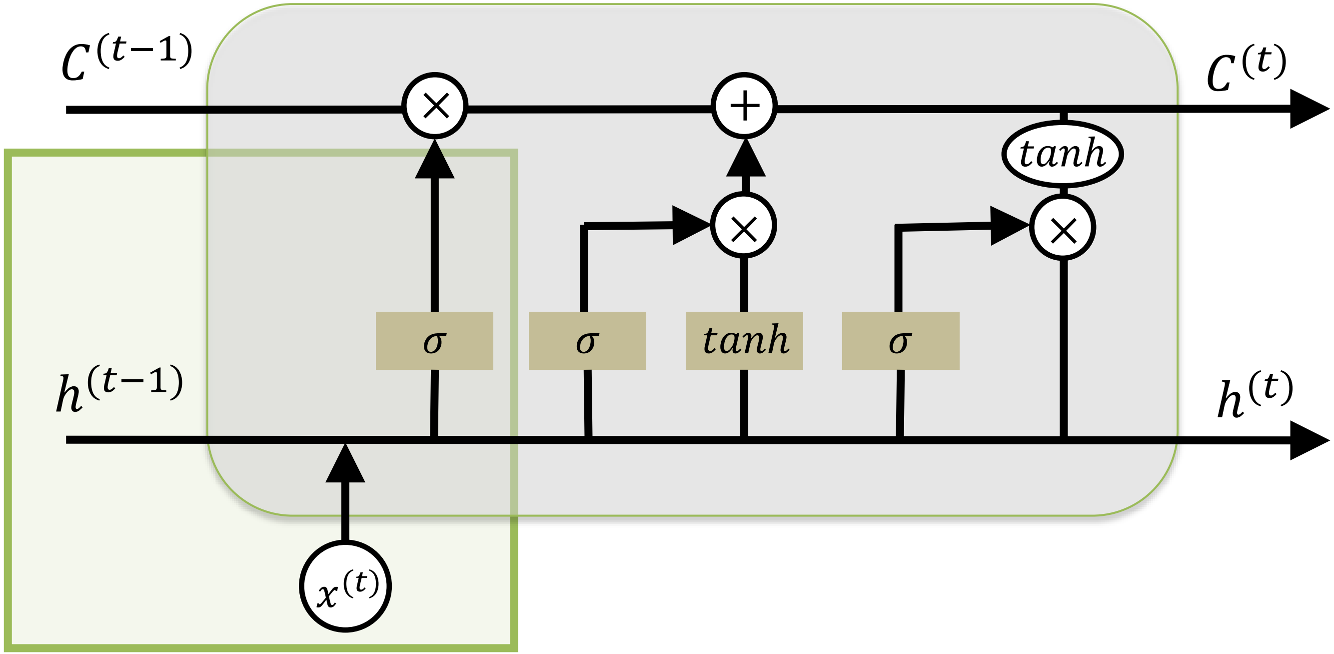
LSTM



单元状态 （Cell State）： $C^{(t)}$

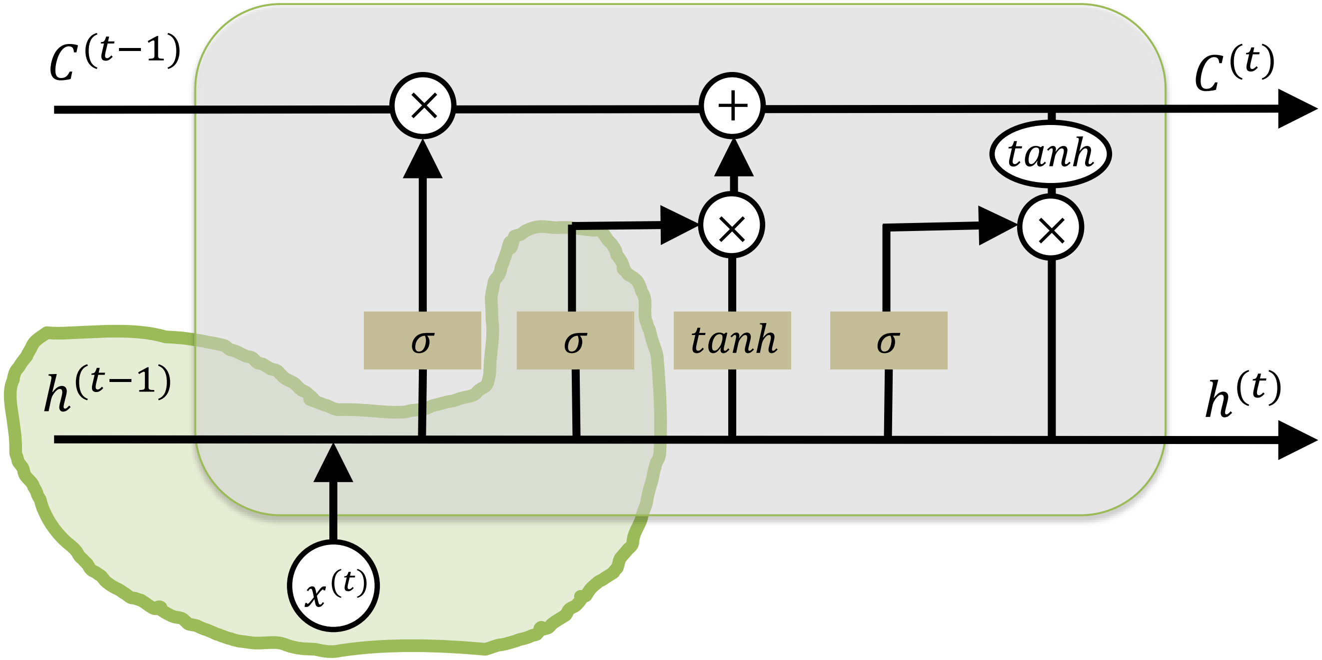
隐藏状态 (Hidden State)： $h^{(t)}$

LSTM：遗忘门



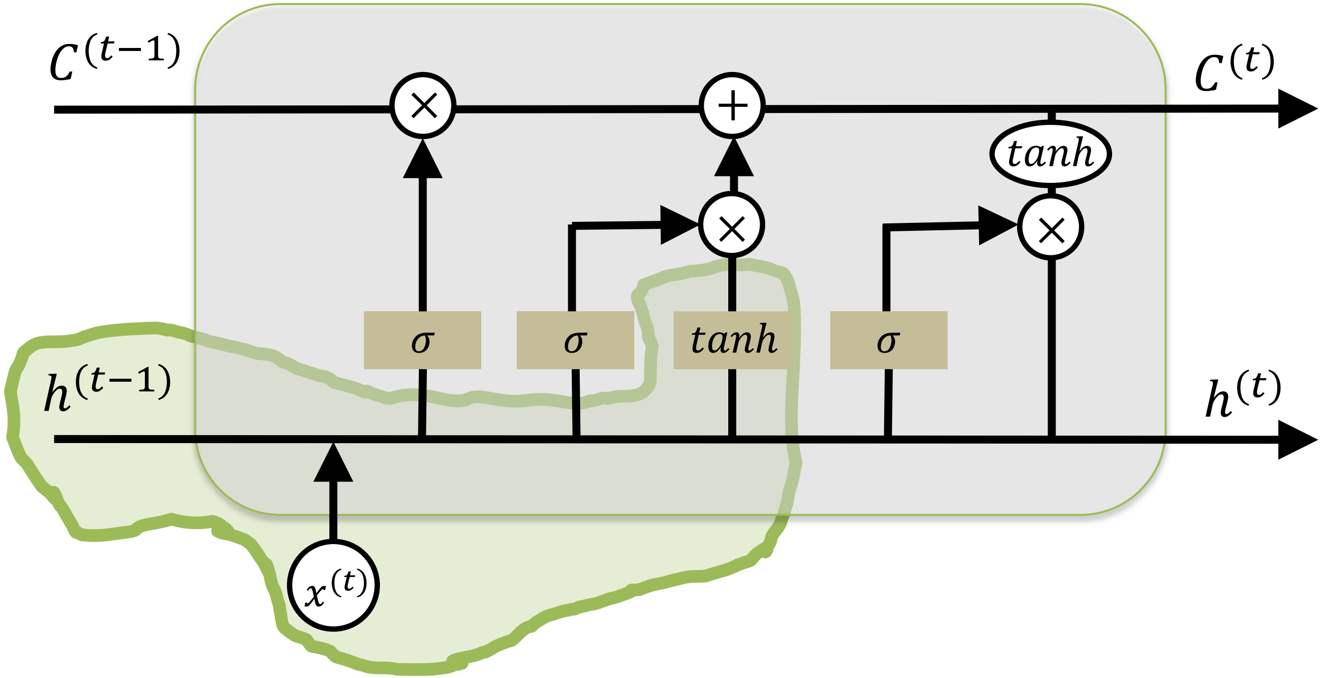
$$\boldsymbol{f}_t = \sigma\left(\boldsymbol{W}_f \cdot \boldsymbol{x}^{(t)} + \boldsymbol{U}_f \cdot \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_f\right)$$

LSTM：输入门



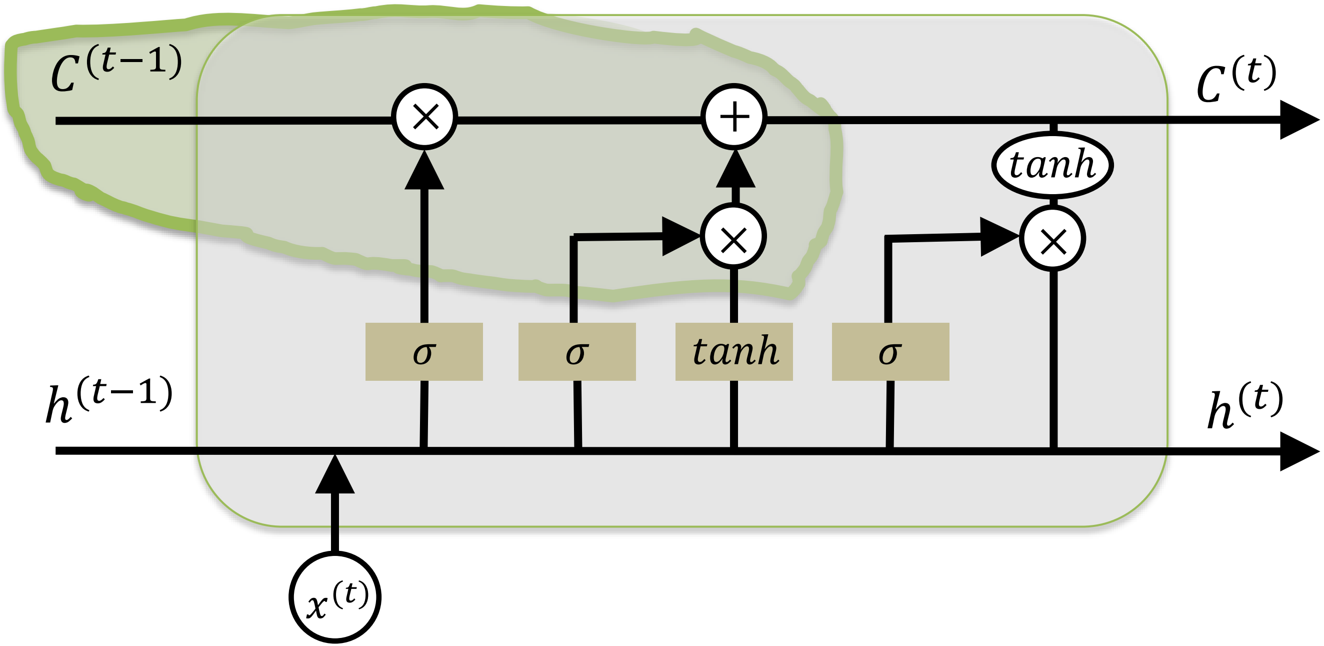
$$\boldsymbol{i}_t = \sigma\left(\boldsymbol{W}_i \cdot \boldsymbol{x}^{(t)} + \boldsymbol{U}_i \cdot \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_i\right)$$

LSTM：候选值



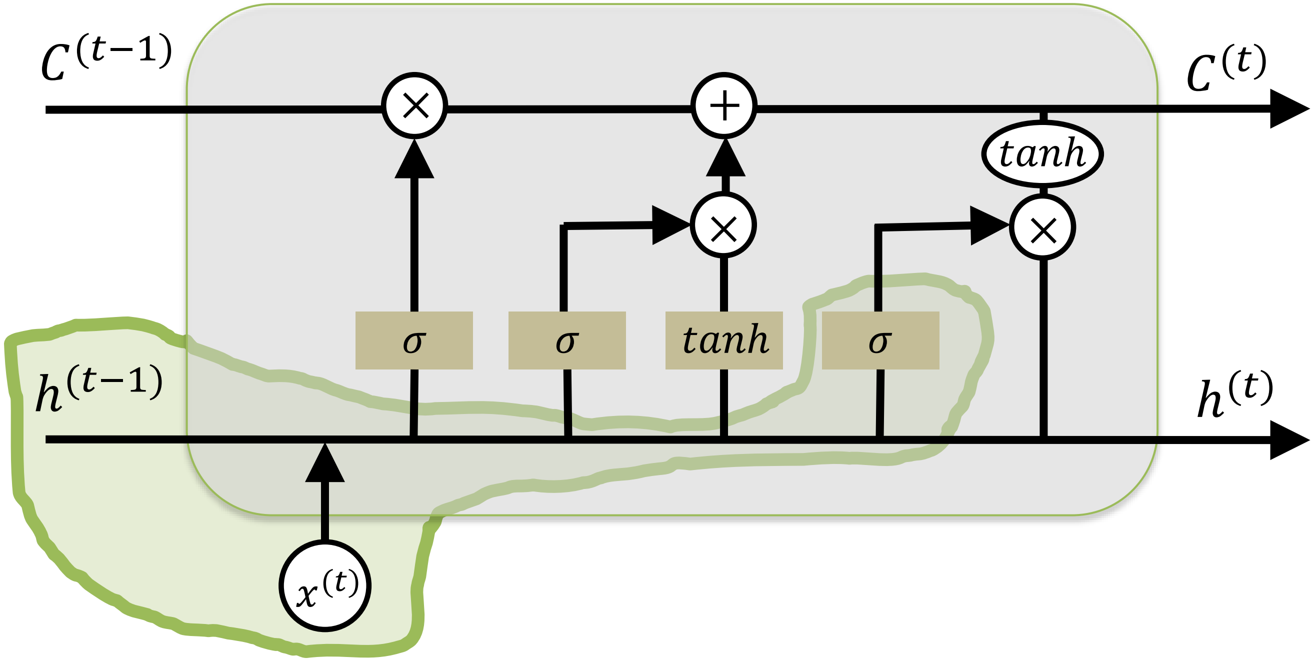
$$\tilde{C}^{(t)} = \tanh\left(\boldsymbol{W}_c \cdot \boldsymbol{x}^{(t)} + \boldsymbol{U}_c \cdot \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_c\right)$$

LSTM：新的单元状态



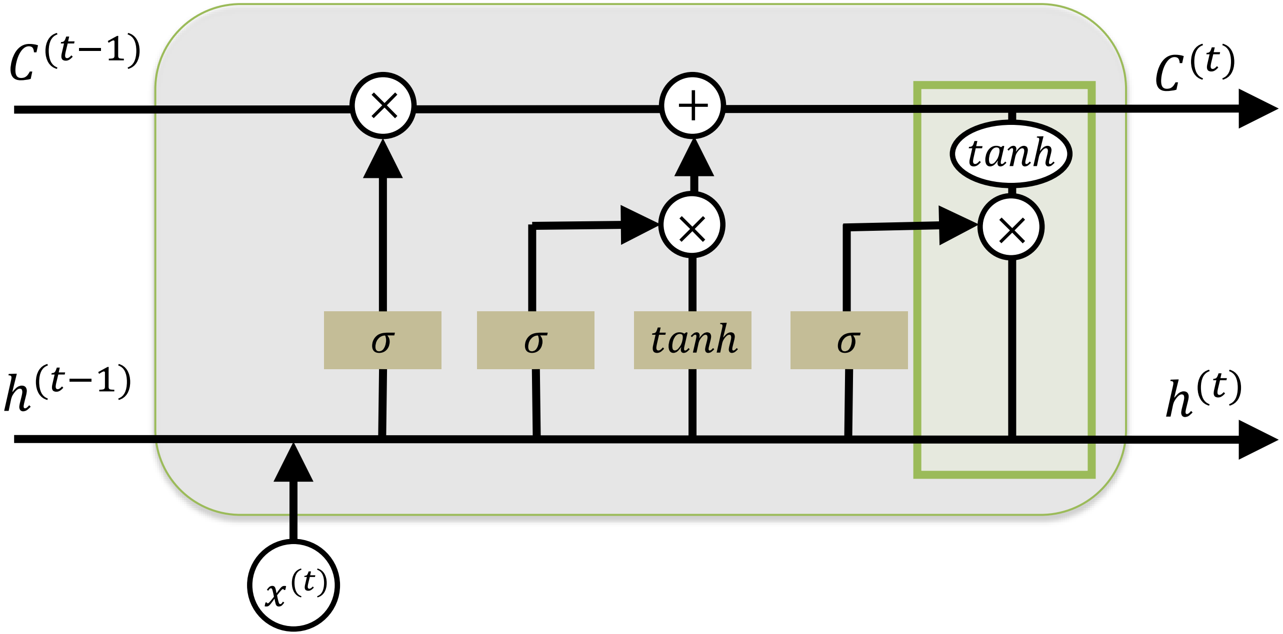
$$\boldsymbol{C}^{(t)} = \boldsymbol{f}_t \odot \boldsymbol{C}^{(t-1)} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{C}}^{(t)}$$

LSTM：输出门



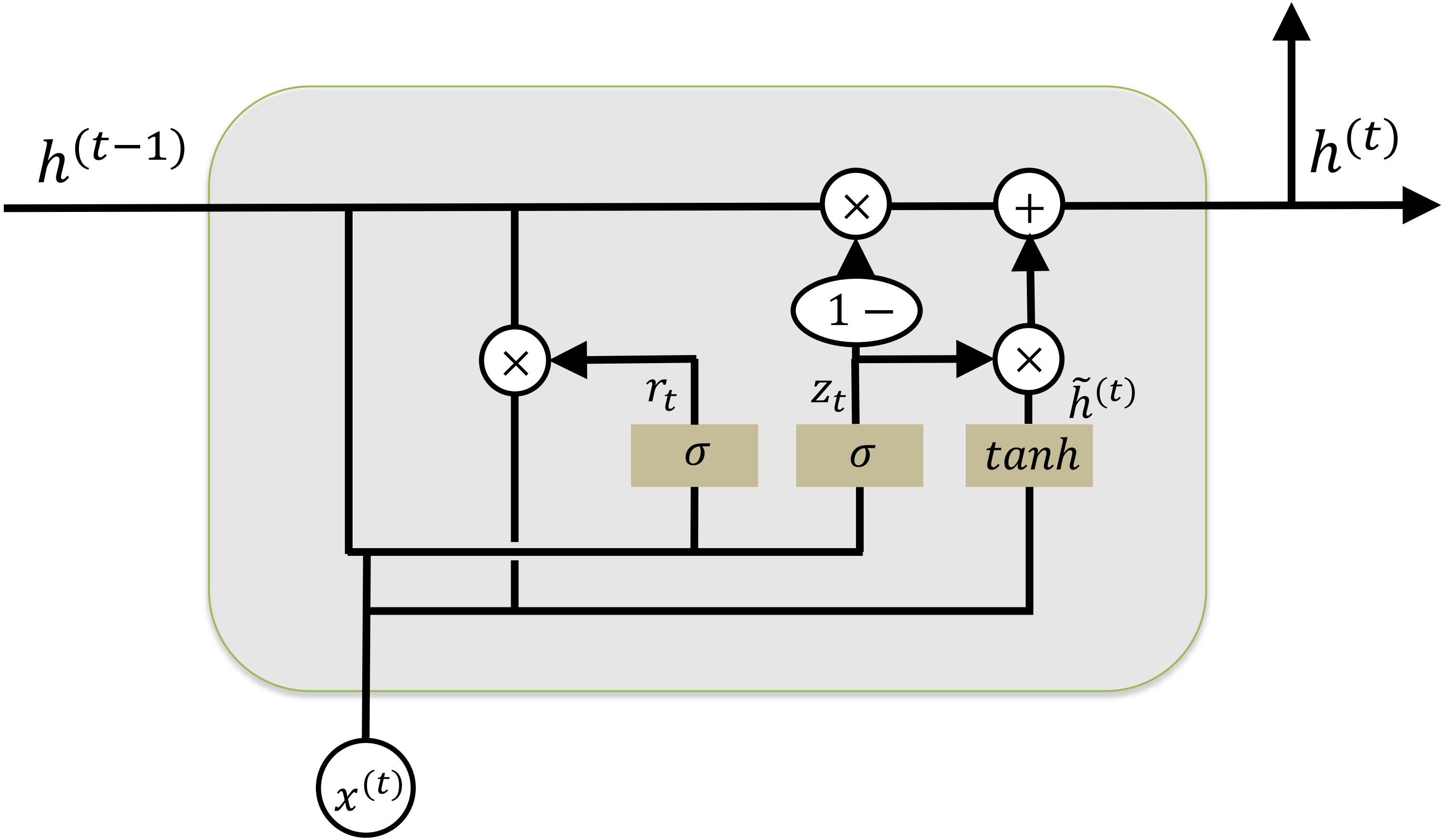
$$\boldsymbol{o}_t = \sigma\left(\boldsymbol{W}_o \cdot \boldsymbol{x}^{(t)} + \boldsymbol{U}_o \cdot \boldsymbol{h}^{(t-1)} + \boldsymbol{b}_o\right)$$

LSTM



$$\boldsymbol{h}^{(t)} = \boldsymbol{o}_t \odot \tanh\left(\boldsymbol{C}^{(t)}\right)$$

Gated Recurrent Unit (GRU)



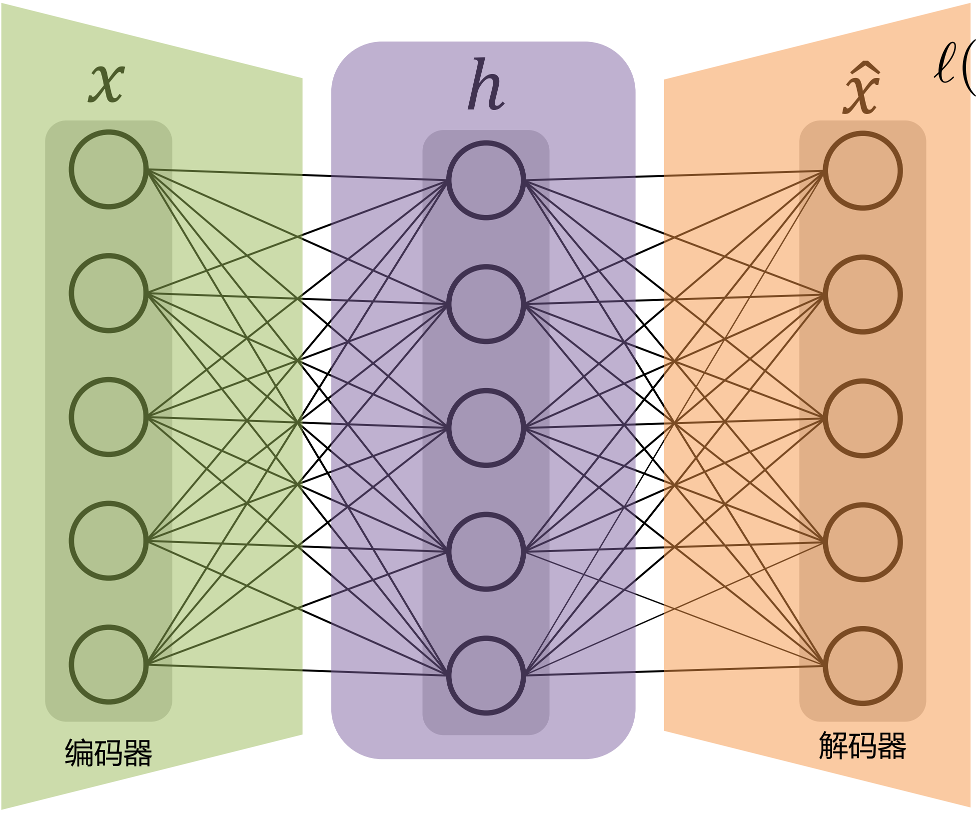
目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

自编码器

无监督学习

编码器和解码器都可以用神经网络建模



$$\ell(\mathbf{x}, \hat{\mathbf{x}}) = \|\mathbf{x} - \hat{\mathbf{x}}\|_2^2$$

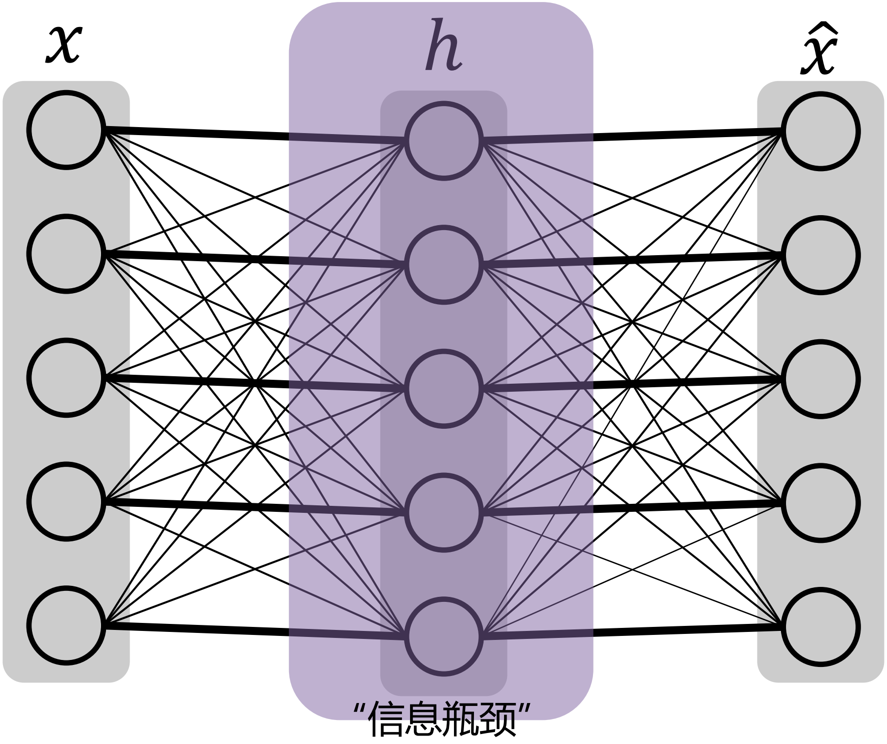
损失函数的一个例子

重建

目标：h能很好地编码x中的信息

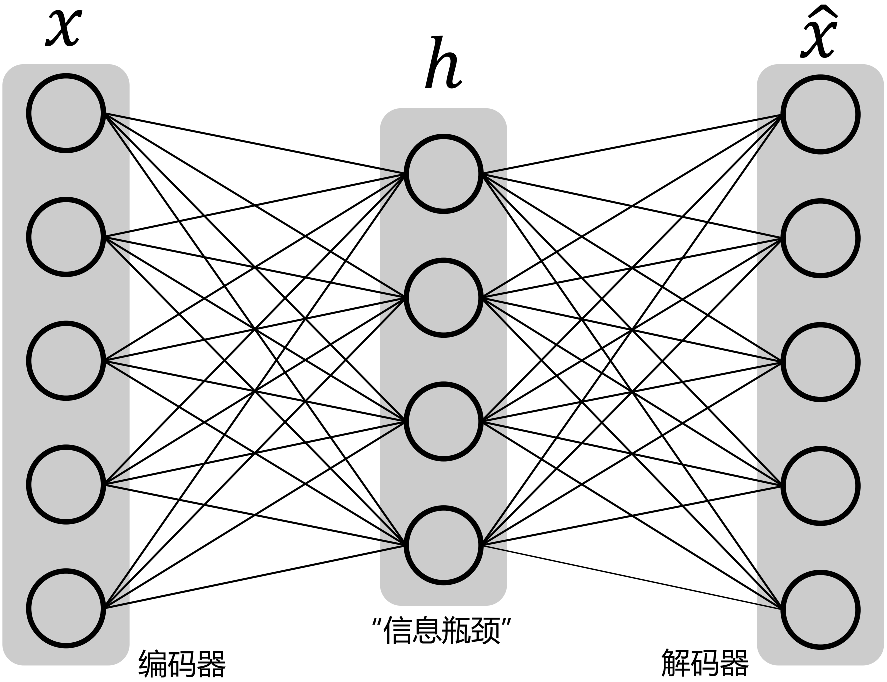
自编码器

无监督学习

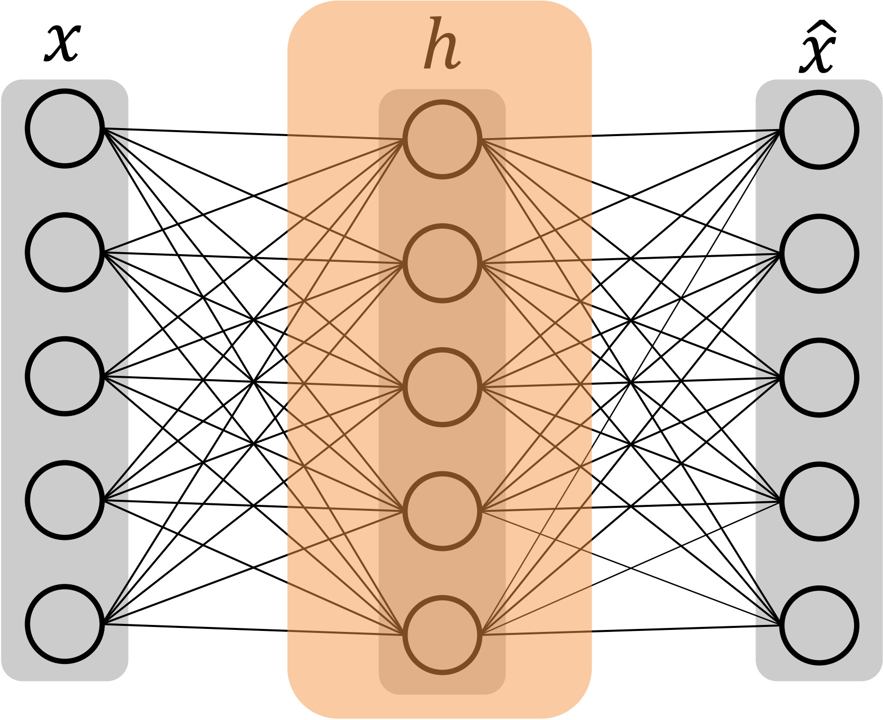


目标：h能很好地编码x中的信息

欠完备自编码器



正则化自编码器



$$\Omega(\mathbf{h}) = \|\mathbf{h}\|_2$$

目录

- 深度学习简史
- 前馈神经网络
- 神经网络的训练
- 卷积神经网络
- 循环神经网络
- 自编码器

深蓝学院
shenlanxueyuan.com

感谢聆听！

Thanks for Listening