第五章　非线性方法

2. 人工神经网络

模式识别基础，2019春

引言

- 贝叶斯分类需要知道样本的概率分布，但估计样本分布有时并不容易

- 直接对数据进行划分的方法：
  - 线性方法：简单、实用、经济，但数据不满足线性可分条件时错误可能大
  - 非线性方法：解决线性不可分问题

张学友演唱会抓了多少逃犯了?46名!看看歌神对这一数字怎么说

2018年10月3日 - 张学友,四大天王中的歌神,不过随着无数逃犯在张学友演唱会上被抓,张学友现在又有了另外一个称号:"逃犯克...

半日娱乐 - 百度快照

"歌神"张学友是如何成为"逃犯克星"的?

2019年1月12日 - 元旦期间,江苏苏州警方在张学友举行演唱会的3天时间里,抓获22名在逃犯。" 歌迷感叹 听说已有将近60个在逃...

新民晚报 - 百度快照

天网系统

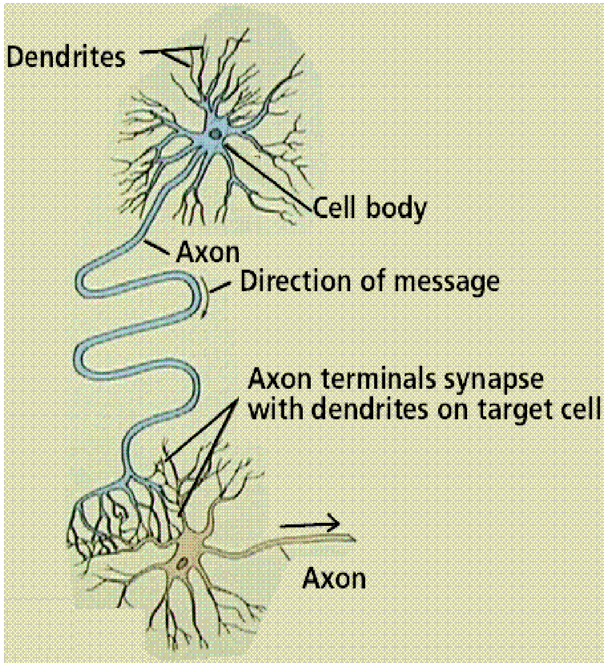
“准确锁定、捕捉到他们的，是‘天网工程’人脸识别系统，” 南湖区公安分局技术与数据服务中心综合管理室主任沈月光介绍说。

“演唱会在正对着检票口的地方增设了几个摄像头，能够对进出检票口所有人员姓名、身份证号、穿着、相貌等进行精确识别把控，它在非常短的时间内便可将数据库筛选一遍。”

......

神经元

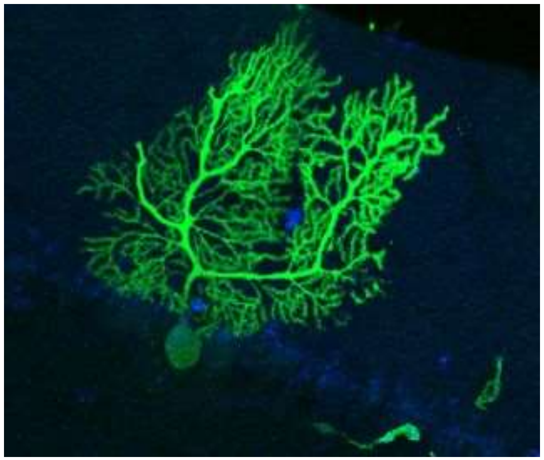
- 神经元（neuron）：
  细胞体（cell）、树突（dentrite）、轴突（axon）、突触（synapses）

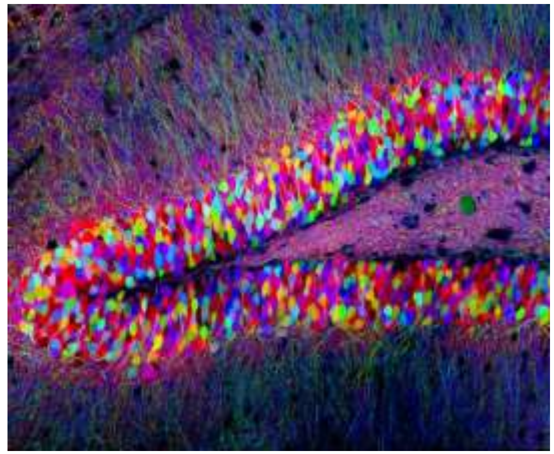


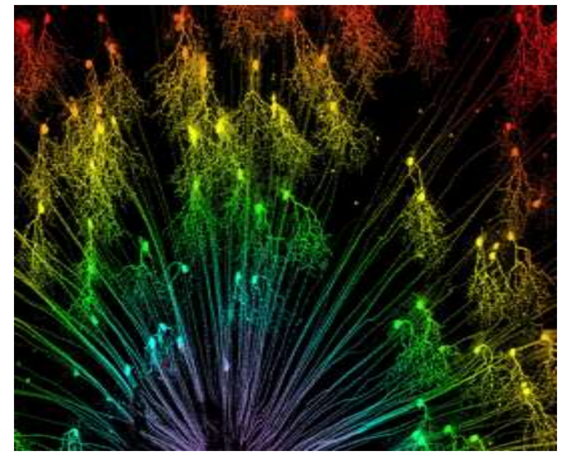
- 神经元的作用：加工、传递信息（电脉冲）
- 神经系统：神经网：大量神经元的复杂连接
- 通过大量简单单元的广泛、复杂的连接而实现各种智能活动。

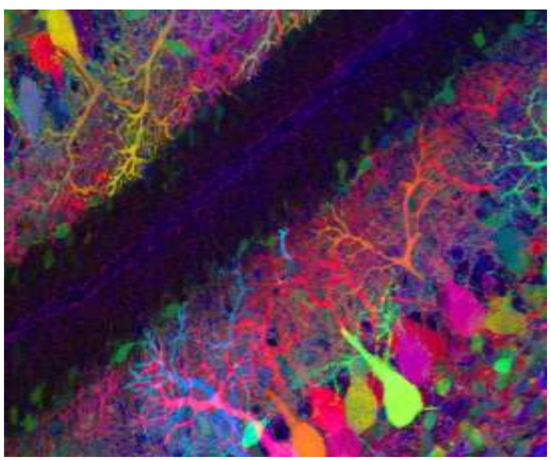
神经细胞

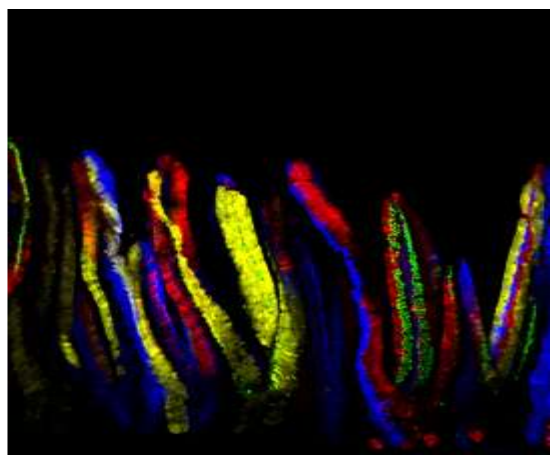
Purkinje cells　　Hippocampus　　Retinal ganglion cells











http://www.cell.com/pictureshow/brainbow

人工神经网络
Artificial Neural Network

- 自然神经网络的某种模型（数学模型）
- 两路研究方向
  - 探索智能活动机理（类脑）
  - 建立可用的高级机器（AI）
- 简史：

➢始于1940～1950年代，以Perceptron为代表

➢1960～1970年代几乎中止研究（AI、符号主义、ES大发展）

➢1980年代　重掀热潮、达到顶峰

➢1990年代　趋于平稳，理论进展不大、多应用　（混合系统）
统计学习理论与支持向量机兴起。

➢2010年代，伴随着大数据热潮，以深度学习(deep learning)之名卷土重来

（人工）神经网络的基本结构

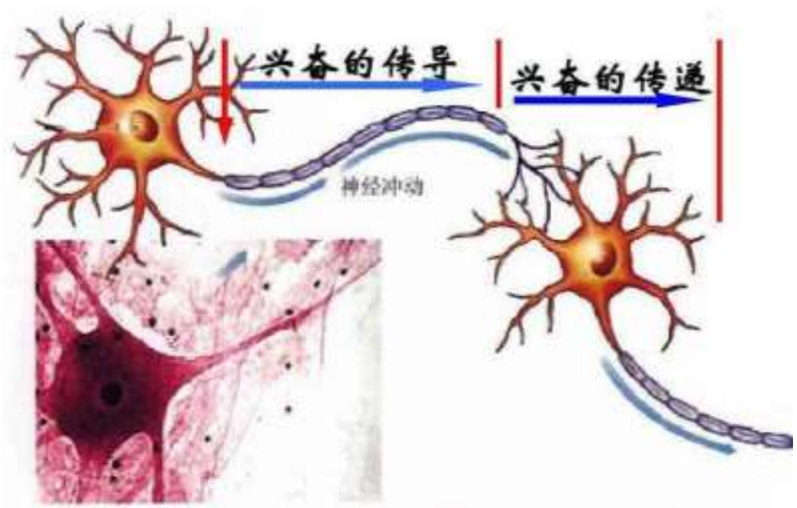
大量简单的计算单元（结点）以某种形式相连接，形成一个网络，其中的某些因素，如连接强度（权值）、结点计算特性甚至网络结构等，可依某种规则随外部数据进行适当的调整（学习），最终实现某种功能。

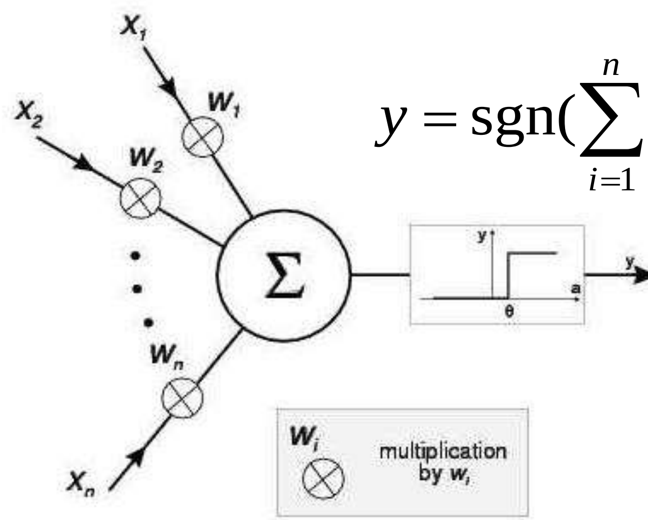
- 三个要素：
  - 神经元的计算特性（传递函数）
  - 网络的结构（连接形式）
  - 学习规则
- 三要素的不同形成了各种各样的神经网模型

神经网络的基本分类

- 基本可分为三大类：
  - 前馈网络　　以Multi-layer Perceptron (MLP)为代表
  - 反馈网络　　以Hopfield网为代表
  - 自组织网络（竞争学习网络） 以SOM为代表

- 基本的神经元模型　　McCulloch-Pitts Model (1943)
  (Threshold Logic Unit - TLU)





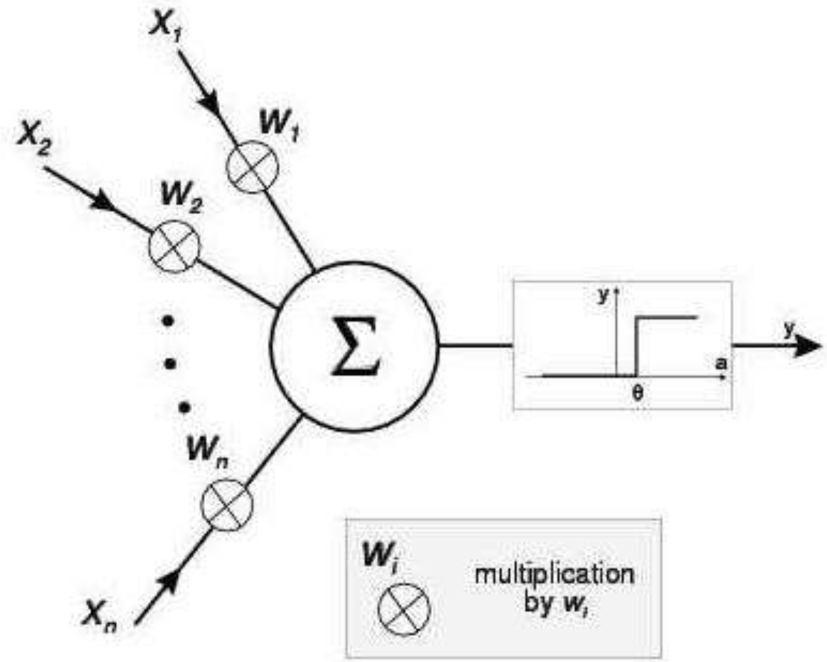
$$y = \mathrm{sgn}\left(\sum_{i=1}^{n} w_i x_i + w_0\right)$$

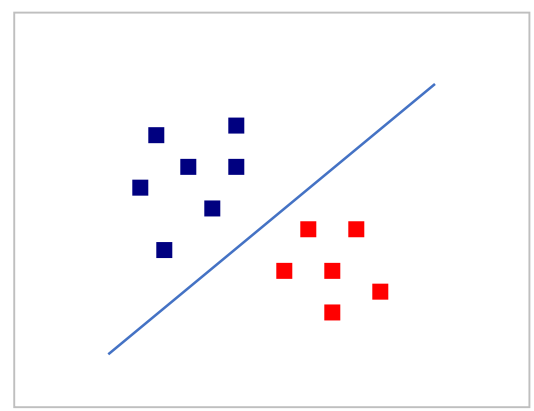
感知器　Perceptron

- Rosenblatt于1950s末提出。

$$y = f_n\left(\sum_{i=1}^{n} w_i x_i - w_0\right)$$

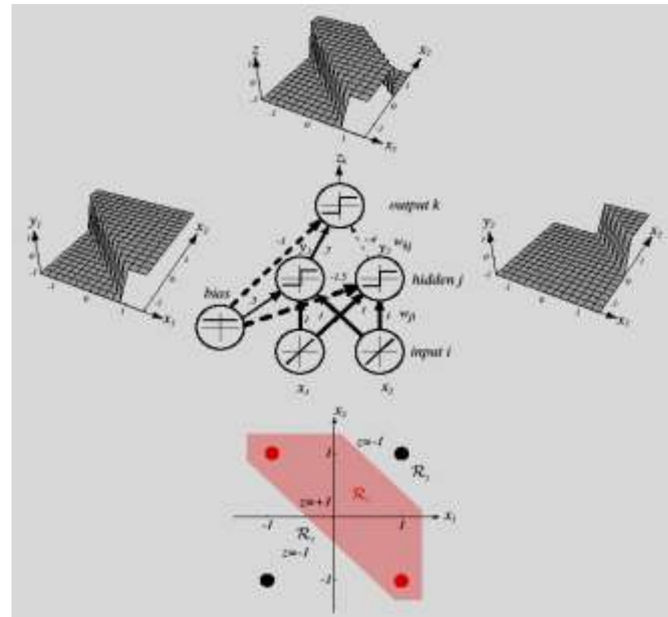
$$y = \begin{cases} +1 & \Rightarrow \text{class } A \\ 0 & \Rightarrow \text{class } B \end{cases}$$

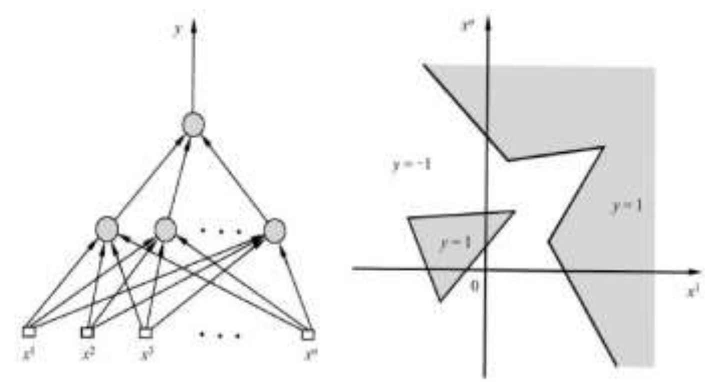




多层感知器

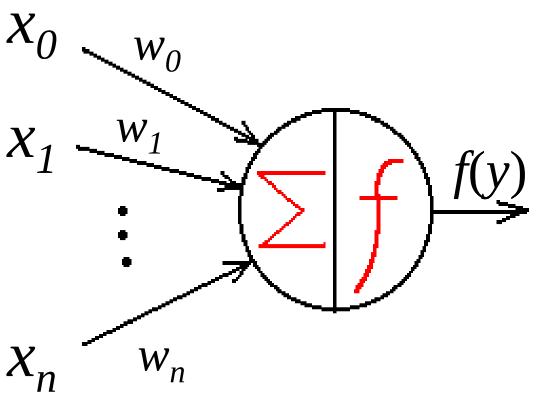
- 当两类线性可分时，此算法收敛
- 问题：Minsky等发现并证明(1969)，感知器只能解决一阶谓词逻辑问题，不能解决高阶问题，如不能解决XOR问题。
- 出路：多个感知器结点结合，引入隐节点，如右图的结构可实现XOR。
- ----- 多层感知器





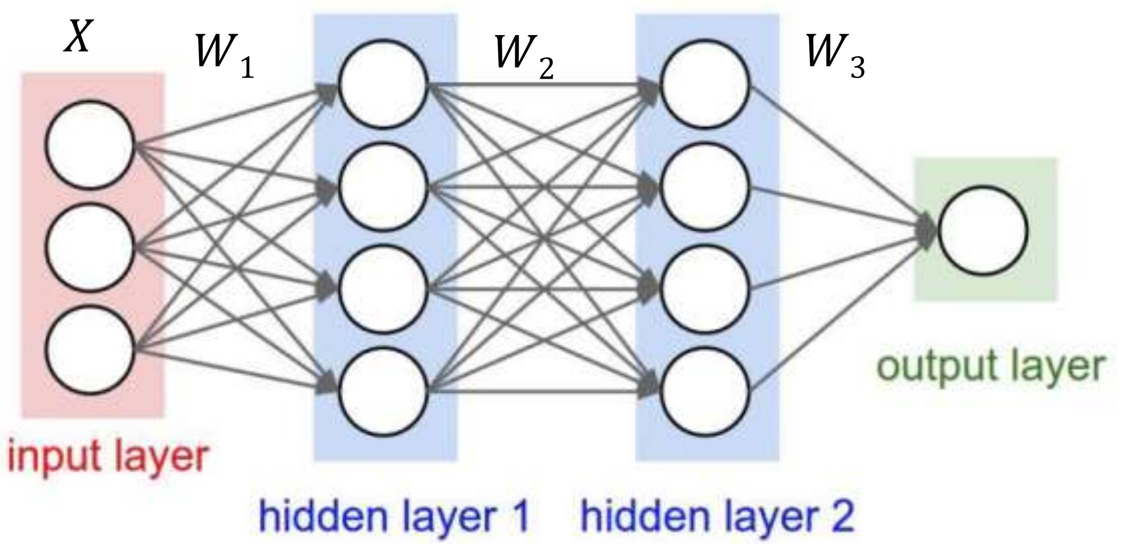
(a) 感知器模型　(b) 输入空间中的两个区域

多层神经网络



$$y=f(W_3f(W_2f(W_1X)))$$

W_i是权重矩阵　　如何求解？



梯度下降法

最小化损失函数：

$$J(\Theta)=\frac{1}{2}\sum_{i=1}^{n}\left(y^{(i)}-f_{\Theta}(\mathbf{x}^{(i)})\right)^2$$

对模型中任意参数迭代更新：

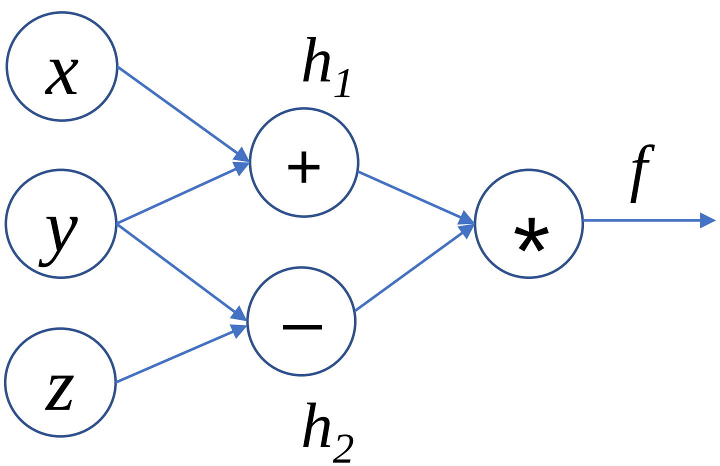
$$w(t)=w(t-1)+\Delta w$$

$$\Delta w=-\alpha*\frac{\partial J}{\partial \mathrm{w}}$$ 其中α为学习率

问题：当模型复杂时，如何求解$\frac{\partial J}{\partial \mathrm{w}}$?

计算图（Computational Graph）

$$f = (x + y)(y - z)$$



$f = h_1 * h_2$ $\quad \dfrac{\partial f}{\partial h_1} = h_2 \quad \dfrac{\partial f}{\partial h_2} = h_1$

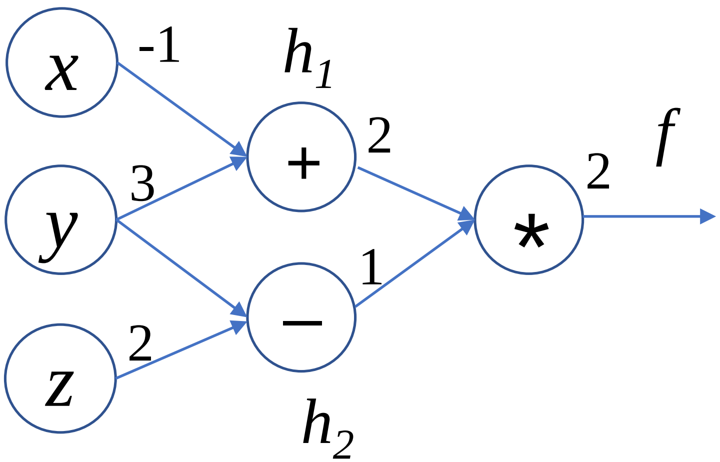
$h_1 = x + y$ $\quad \dfrac{\partial h_1}{\partial x} = 1 \quad \dfrac{\partial h_2}{\partial y} = 1$

$h_2 = y - z$ $\quad \dfrac{\partial h_1}{\partial y} = 1 \quad \dfrac{\partial h_2}{\partial z} = -1$

目标：求解 $\dfrac{\partial f}{\partial x}$，$\dfrac{\partial f}{\partial y}$，$\dfrac{\partial f}{\partial z}$

计算图（Computational Graph）

$f = (x+y)(y-z)$



$f = h_1 * h_2$ $\quad \frac{\partial f}{\partial h_1} = h_2 \quad \frac{\partial f}{\partial h_2} = h_1$

$h_1 = x + y \quad \frac{\partial h_1}{\partial x} = 1 \quad \frac{\partial h_2}{\partial y} = 1$

$h_2 = y - z \quad \frac{\partial h_1}{\partial y} = 1 \quad \frac{\partial h_2}{\partial z} = -1$

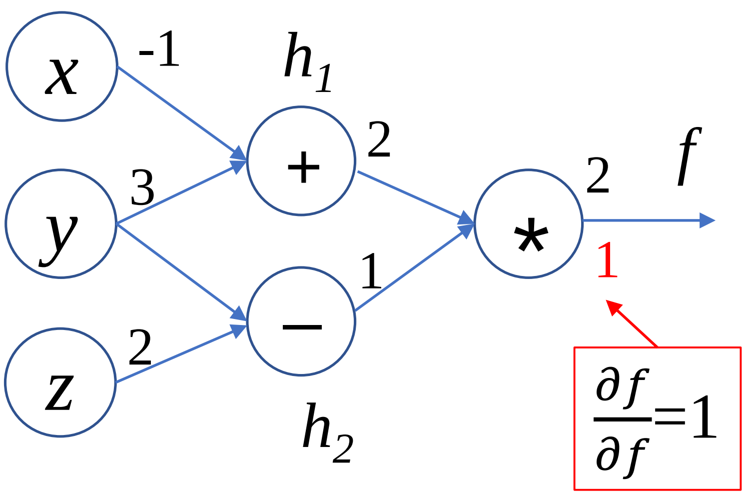
给定初值：

$(x,y,z)=[-1, 3, 2]$

目标：求解 $\frac{\partial f}{\partial x}$，$\frac{\partial f}{\partial y}$，$\frac{\partial f}{\partial z}$

计算图（Computational Graph）

$f = (x+y)(y-z)$



(x,y,z)=[-1, 3, 2]

$f = h_1 * h_2$ $\quad \frac{\partial f}{\partial h_1} = h_2 \quad \frac{\partial f}{\partial h_2} = h_1$

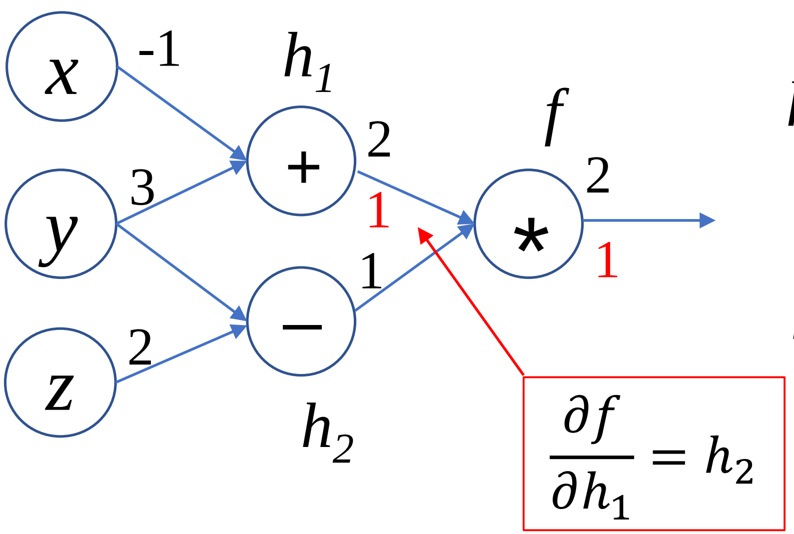
$h_1 = x+y$ $\quad \frac{\partial h_1}{\partial x} = 1 \quad \frac{\partial h_2}{\partial y} = 1$

$h_2 = y-z$ $\quad \frac{\partial h_1}{\partial y} = 1 \quad \frac{\partial h_2}{\partial z} = -1$

目标：求解 $\frac{\partial f}{\partial x}$，$\frac{\partial f}{\partial y}$，$\frac{\partial f}{\partial z}$

计算图（Computational Graph）

$$f=(x+y)(y-z)$$



(x,y,z)=[-1, 3, 2]

$f= h_1 * h_2$ $\quad \frac{\partial f}{\partial h_1}=h_2$ $\quad \frac{\partial f}{\partial h_2}=h_1$

$h_1=x+y$ $\quad \frac{\partial h_1}{\partial x}=1$ $\quad \frac{\partial h_2}{\partial y}=1$

$h_2=y-z$ $\quad \frac{\partial h_1}{\partial y}=1$ $\quad \frac{\partial h_2}{\partial z}=-1$

目标：求解 $\frac{\partial f}{\partial x}$， $\frac{\partial f}{\partial y}$， $\frac{\partial f}{\partial z}$

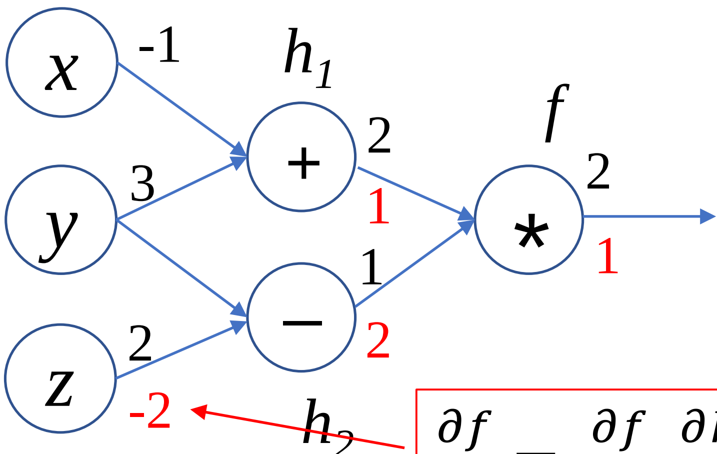
计算图（Computational Graph）

$f = (x+y)(y-z)$

$f = h_1 * h_2 \quad \frac{\partial f}{\partial h_1} = h_2 \quad \frac{\partial f}{\partial h_2} = h_1$

$h_1 = x+y \quad \frac{\partial h_1}{\partial x} = 1 \quad \frac{\partial h_1}{\partial y} = 1$

$h_2 = y-z \quad \frac{\partial h_2}{\partial y} = 1 \quad \frac{\partial h_2}{\partial z} = -1$



$$\boxed{\frac{\partial f}{\partial z} = \frac{\partial f}{\partial h_2}\frac{\partial h_2}{\partial z}}$$

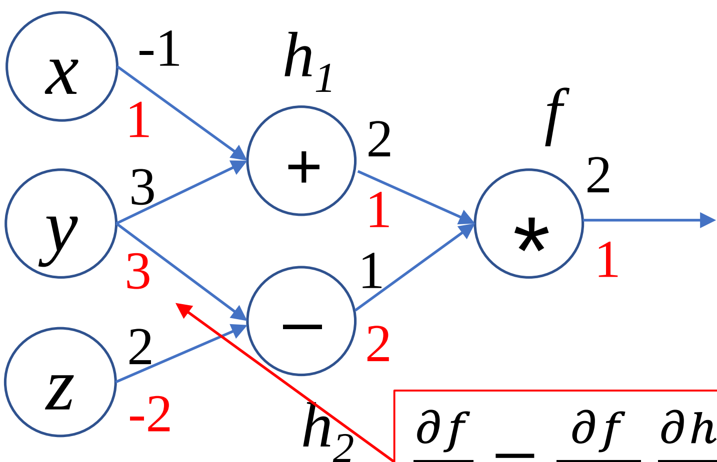
$(x,y,z)=[-1, 3, 2]$

目标：求解 $\frac{\partial f}{\partial x}$，$\frac{\partial f}{\partial y}$，$\frac{\partial f}{\partial z}$

计算图（Computational Graph）

$f=(x+y)(y-z)$

$f=h_1 * h_2 \quad \frac{\partial f}{\partial h_1}=h_2 \quad \frac{\partial f}{\partial h_2}=h_1$



$h_1=x+y \quad \frac{\partial h_1}{\partial x}=1 \quad \frac{\partial h_1}{\partial y}=1$

$h_2=y-z \quad \frac{\partial h_2}{\partial y}=1 \quad \frac{\partial h_2}{\partial z}=-1$

$$\boxed{\frac{\partial f}{\partial y}=\frac{\partial f}{\partial h_2}\frac{\partial h_2}{\partial y}+\frac{\partial f}{\partial h_1}\frac{\partial h_1}{\partial y}}$$

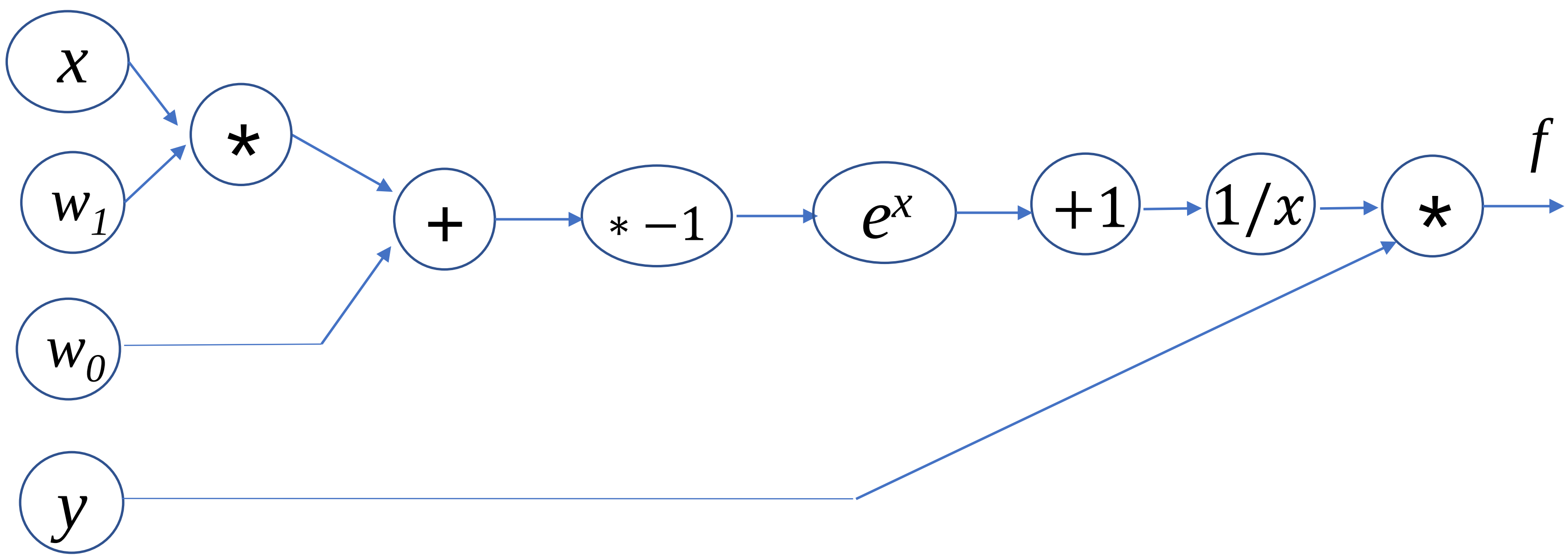
$(x,y,z)=[-1, 3, 2]$

目标：求解 $\frac{\partial f}{\partial x}$， $\frac{\partial f}{\partial y}$， $\frac{\partial f}{\partial z}$

又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

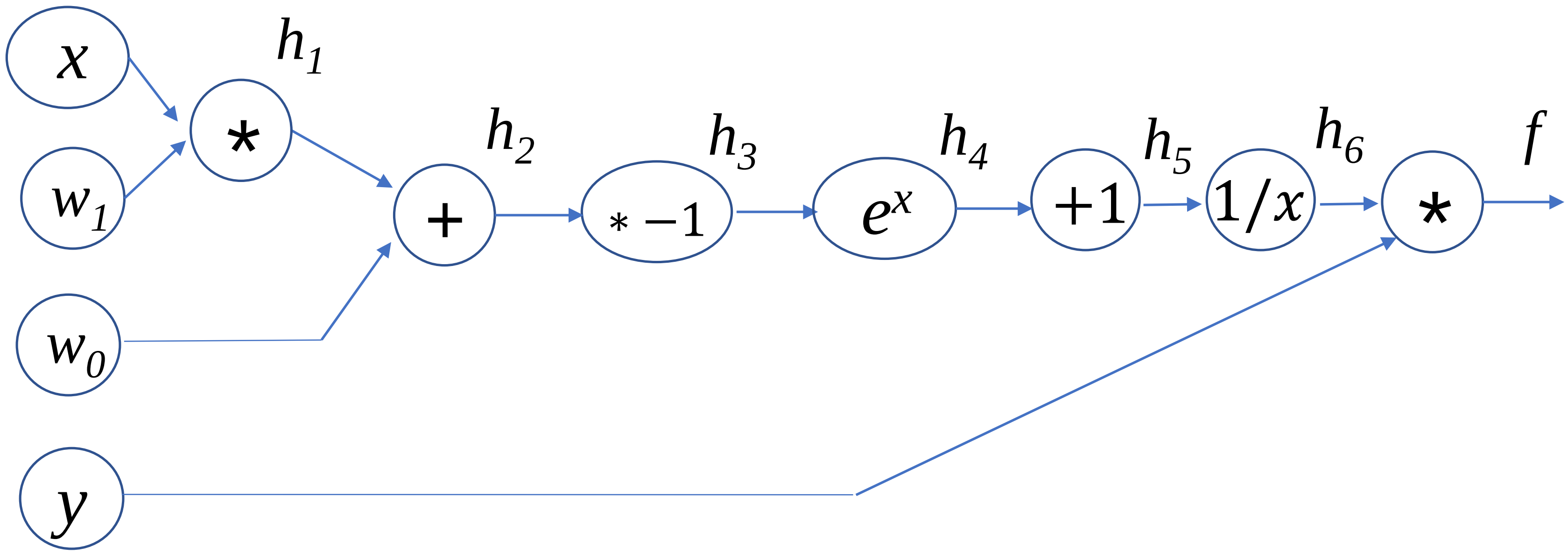
初值： x=-2　w_0=1
y=1　w_1=3



又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值： x=-2　w_0=1
y=1　w_1=3



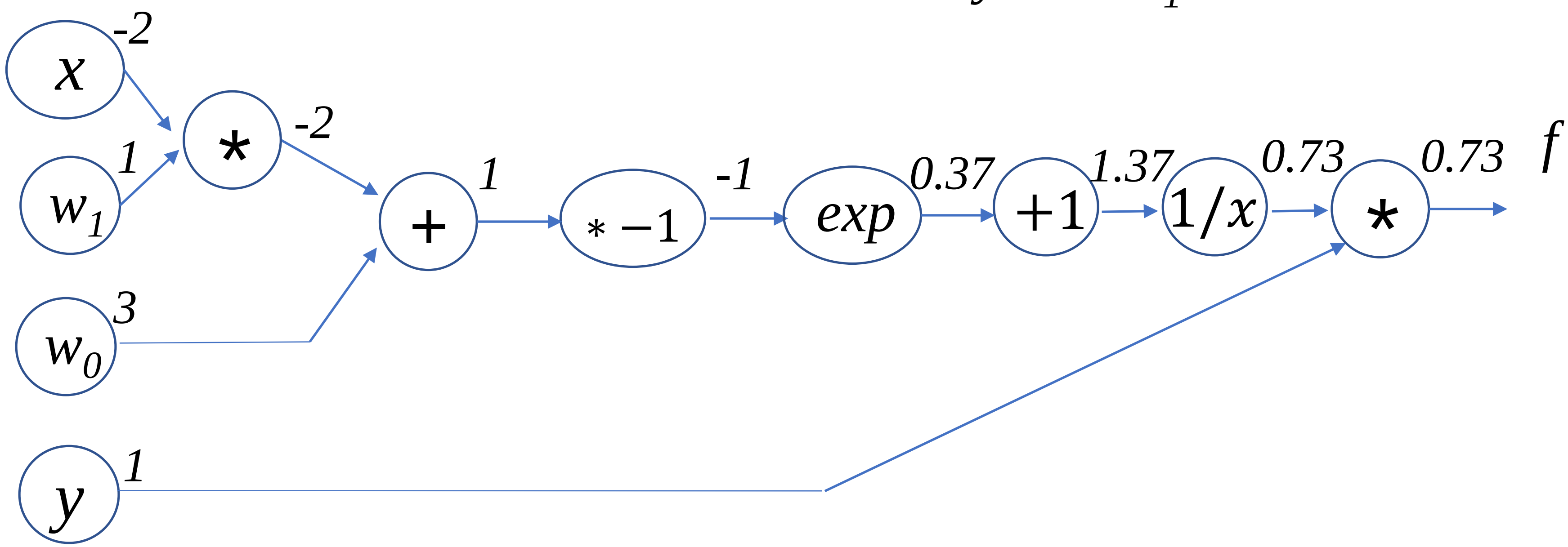
$f=h_6*y$　$h_6=1/h_5$　$h_5=1+h_4$　$h_4=exp(h_3)$

$h_3=(-1)*h_2$　$h_2=w_0+h_1$　$h_1=x+w_1$

又一个例子

$$f=\frac{y}{1+e^{-(w_0+w_1x)}}$$

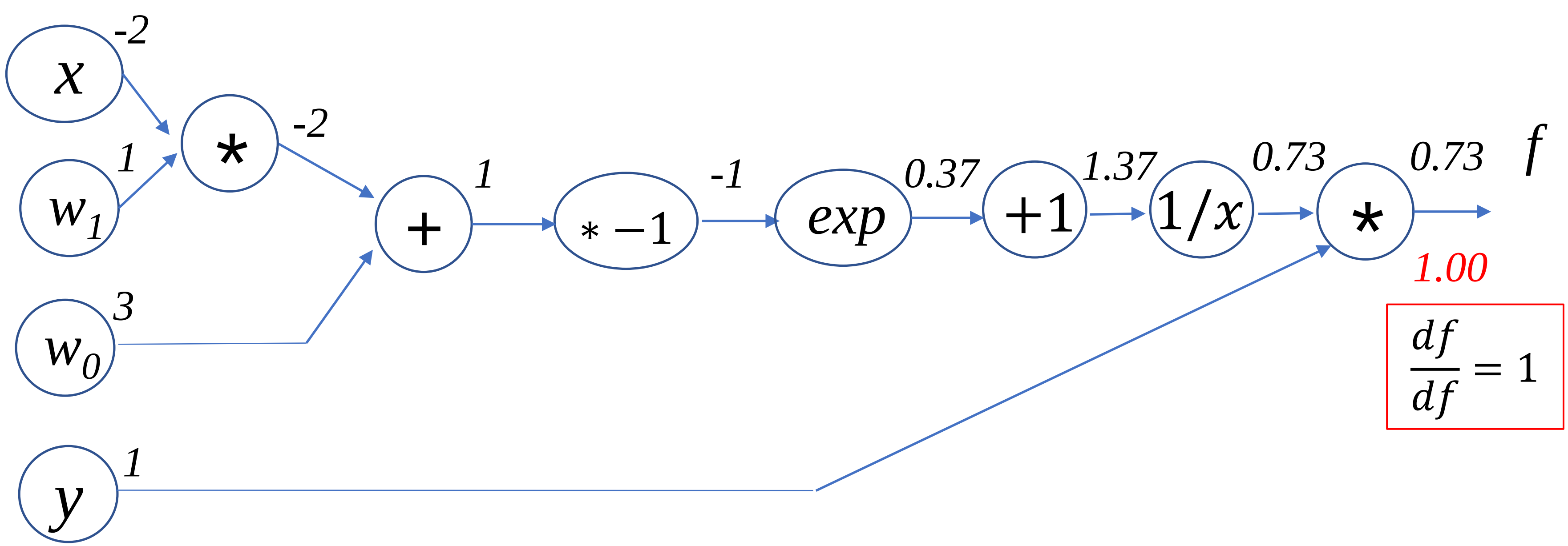
初值： x=-2 w_0=1
y=1 w_1=3



又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值：x_0=-2　w_0=3　w_1=1　y=1



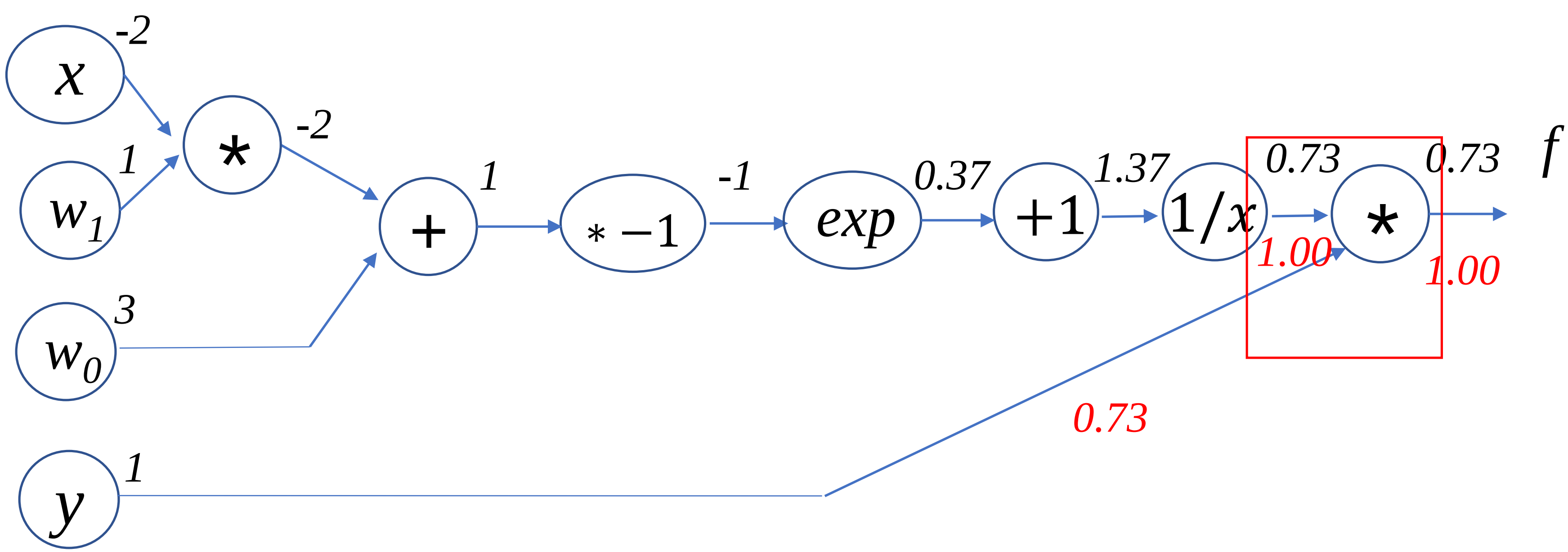
$$f(x) = ex \rightarrow \frac{df}{dx} = ex \qquad f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -\frac{1}{x^2}$$

$$f(x) = ax \rightarrow \frac{df}{dx} = a \qquad f(x) = a + x \rightarrow \frac{df}{dx} = 1$$

又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值：x=-2　w_0=3　w_1=1　y=1



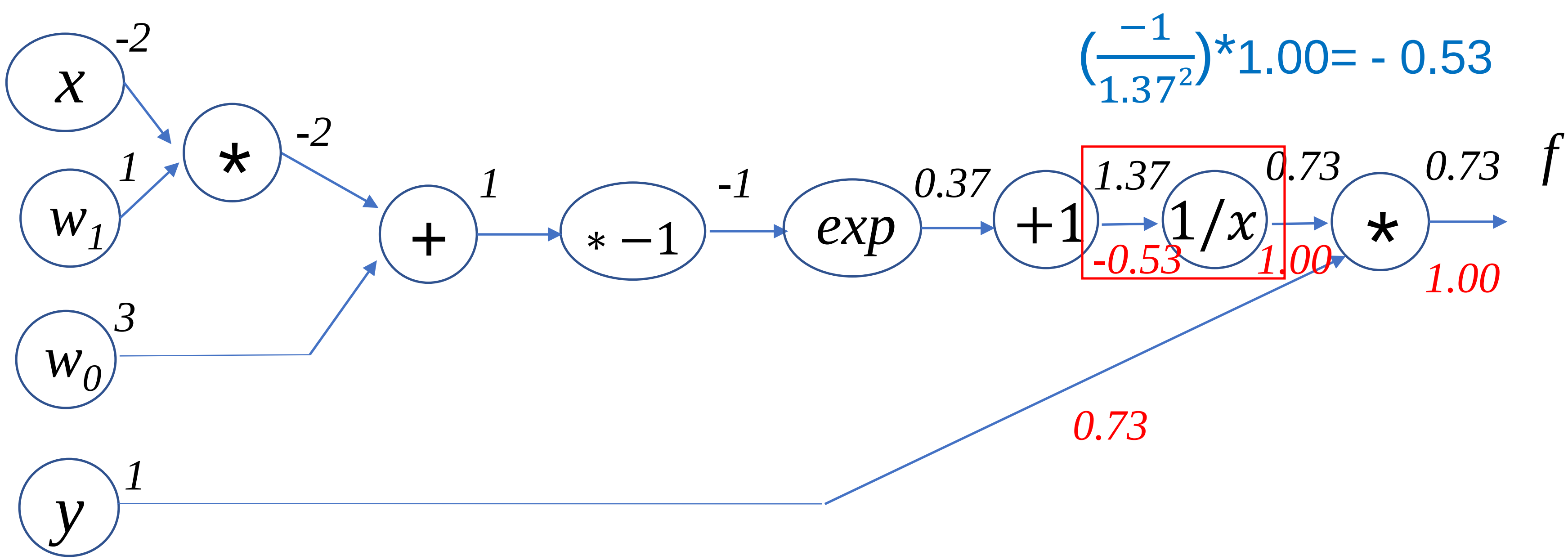
$$f(x) = ex \to \frac{df}{dx} = ex \qquad f(x) = \frac{1}{x} \to \frac{df}{dx} = -\frac{1}{x^2}$$

$$\boxed{f(x) = ax \to \frac{df}{dx} = a} \qquad f(x) = a + x \to \frac{df}{dx} = 1$$

又一个例子

$$f=\frac{y}{1+e^{-(w_0+w_1x)}}$$

初值： $x=-2 \quad w_0=3 \quad w_1=1 \quad y=1$



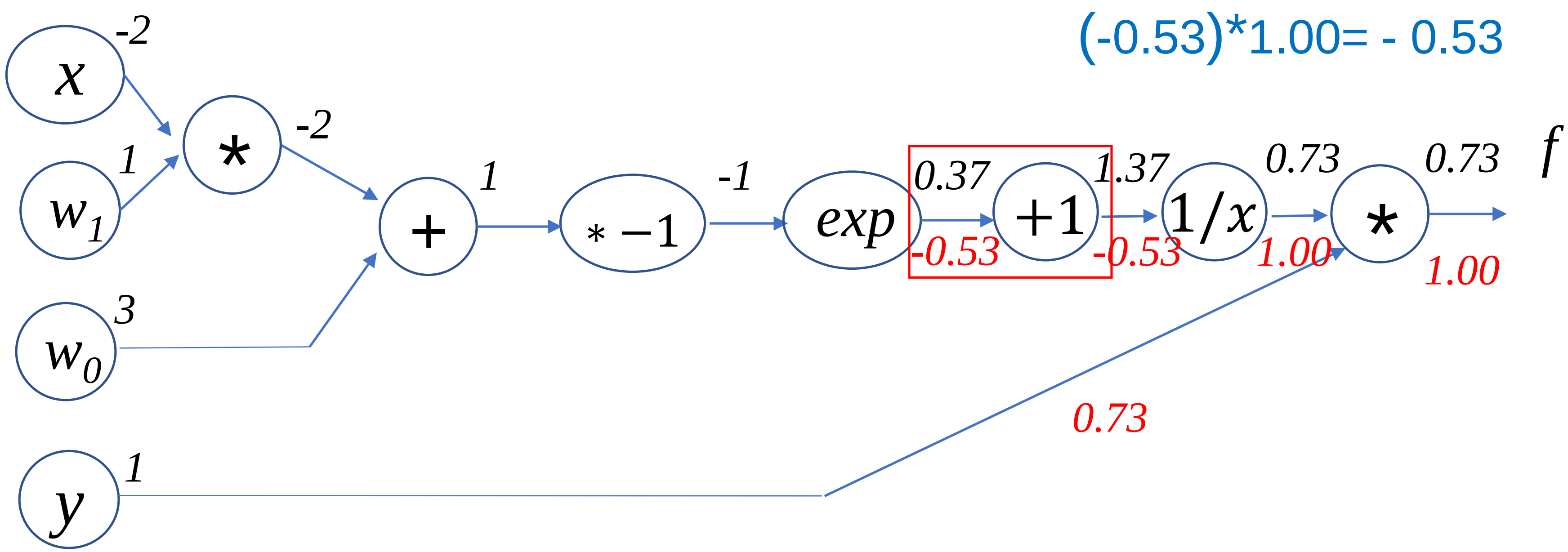
$$f(x)=ex \rightarrow \frac{df}{dx}=ex \qquad \boxed{f(x)=\frac{1}{x} \rightarrow \frac{df}{dx}=-\frac{1}{x^2}}$$

$$f(x)=ax \rightarrow \frac{df}{dx}=a \qquad f(x)=a+x \rightarrow \frac{df}{dx}=1$$

又一个例子

$$f=\frac{y}{1+e^{-(w_0+w_1x_0)}}$$

初值：x_0=-2 w_0=3 w_1=1 y=1



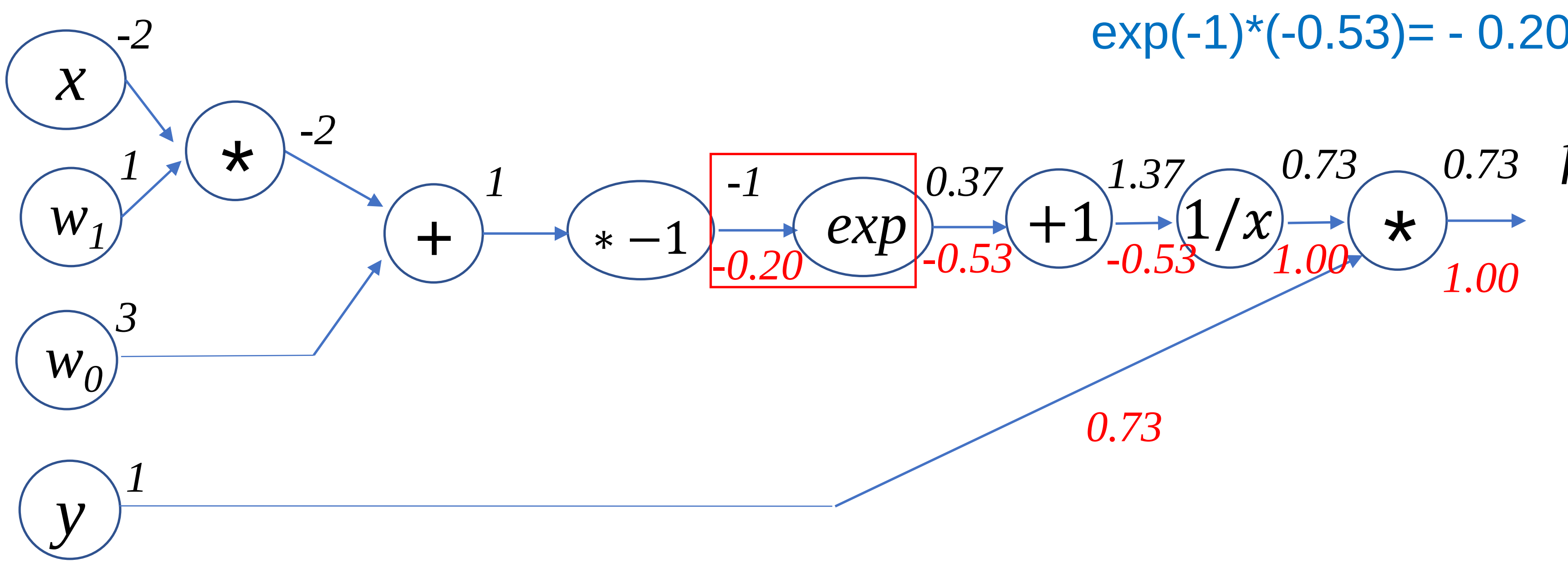
$$f(x)=ex\rightarrow\frac{df}{dx}=ex \qquad f(x)=\frac{1}{x}\rightarrow\frac{df}{dx}=-\frac{1}{x^2}$$

$$f(x)=ax\rightarrow\frac{df}{dx}=a \qquad \boxed{f(x)=a+x\rightarrow\frac{df}{dx}=1}$$

又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值：x=-2　w_0=3　w_1=1　y=1



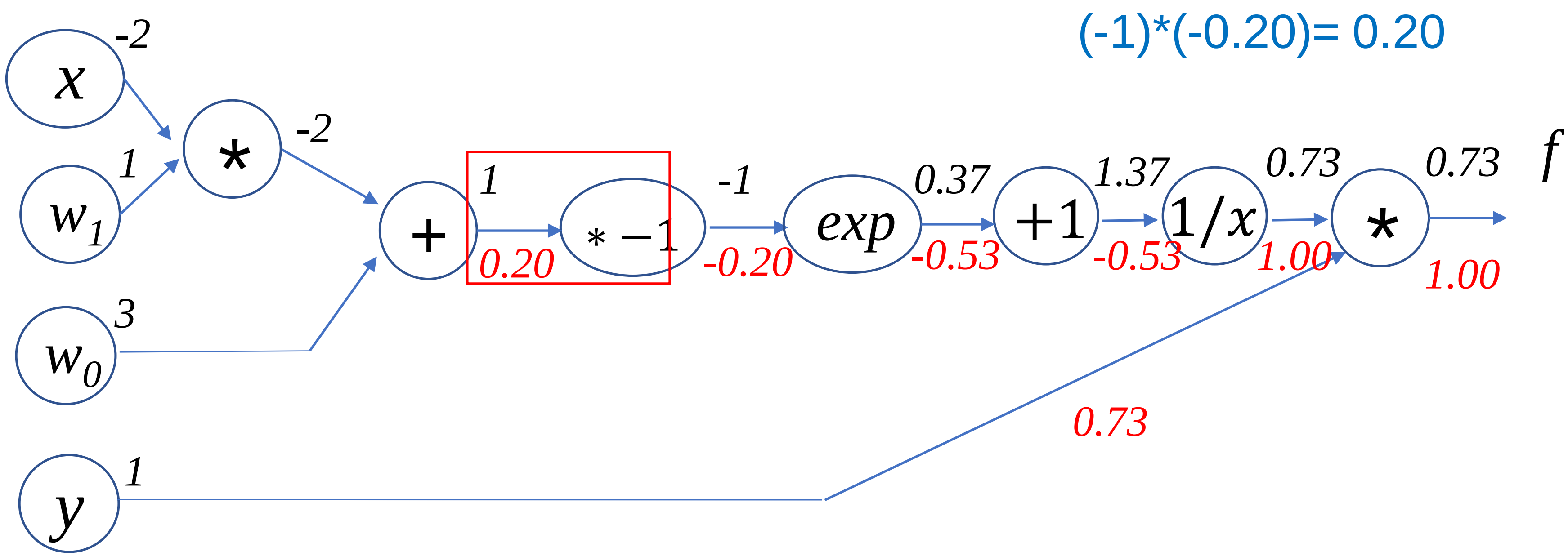
$$\boxed{f(x) = ex \rightarrow \frac{df}{dx} = ex} \qquad f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -\frac{1}{x^2}$$

$$f(x) = ax \rightarrow \frac{df}{dx} = a \qquad f(x) = a + x \rightarrow \frac{df}{dx} = 1$$

又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值： x=-2 w_0=3 w_1=1 y=1



$$f(x) = ex \rightarrow \frac{df}{dx} = ex \qquad f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -\frac{1}{x^2}$$

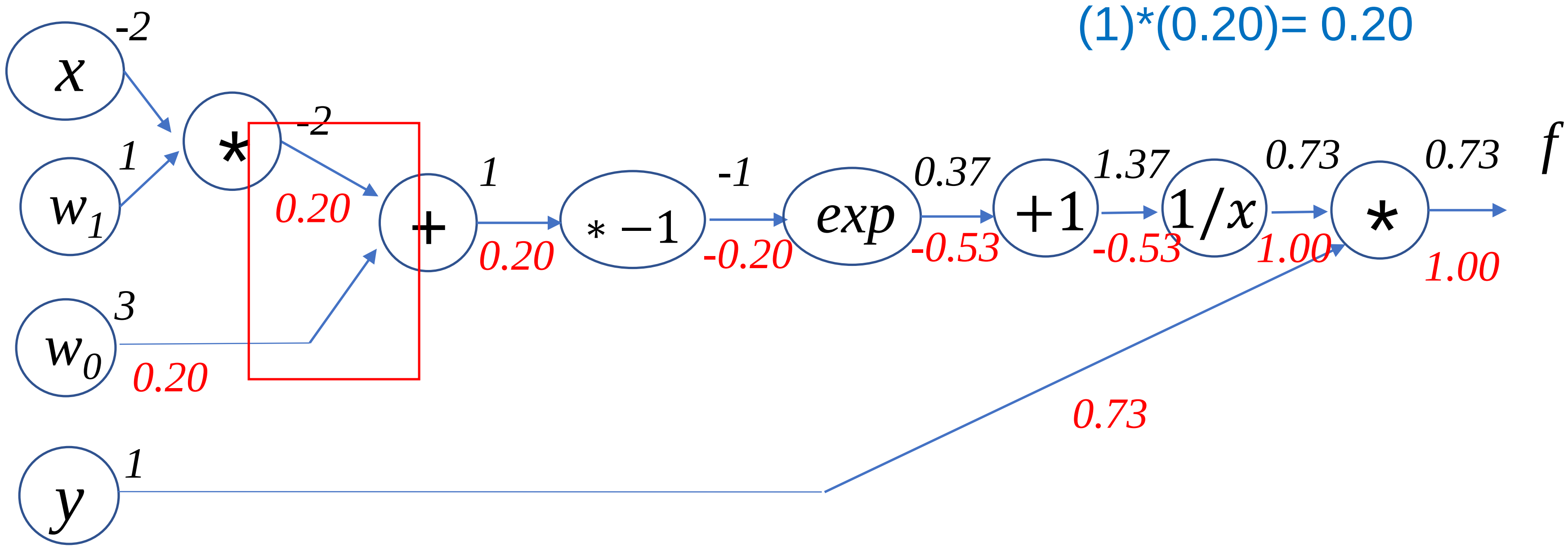
$$\boxed{f(x) = ax \rightarrow \frac{df}{dx} = a} \qquad f(x) = a + x \rightarrow \frac{df}{dx} = 1$$

又一个例子

$$f = \frac{y}{1 + e^{-(w_0 + w_1 x)}}$$

初值： x=-2 w_0=3 w_1=1 y=1

(1)*(0.20)= 0.20



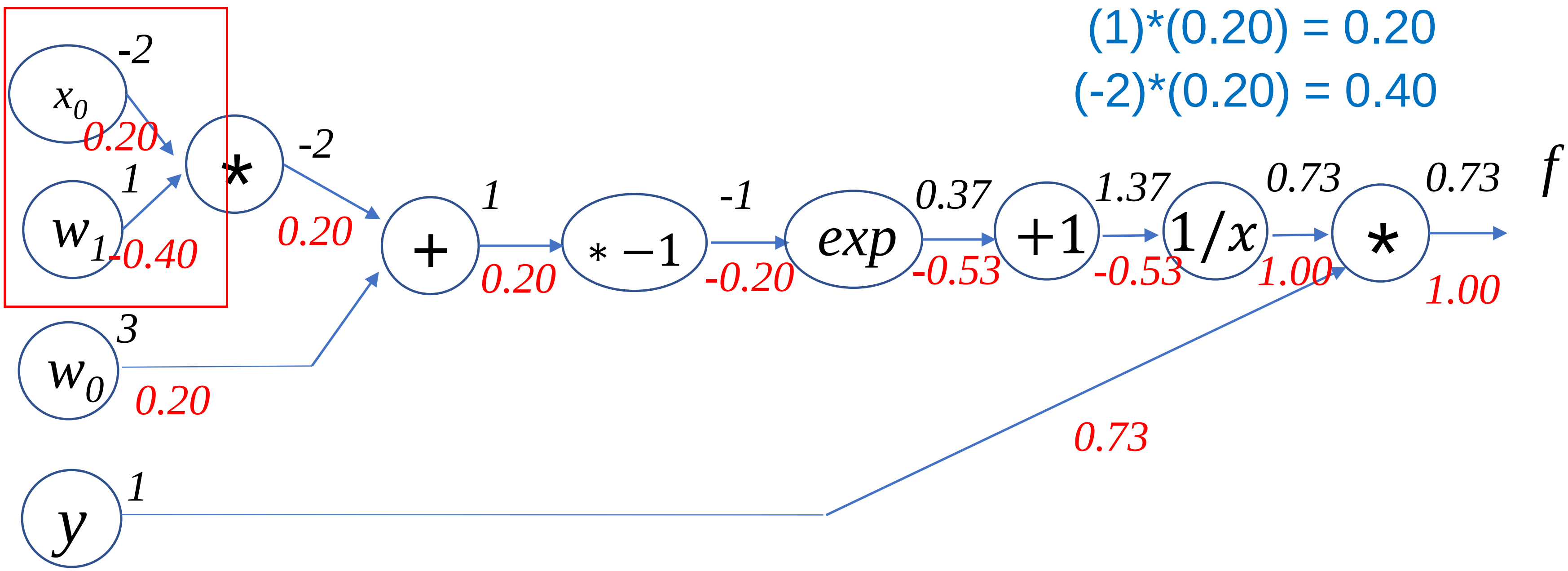
$$f(x) = ex \rightarrow \frac{df}{dx} = ex \qquad f(x) = \frac{1}{x} \rightarrow \frac{df}{dx} = -\frac{1}{x^2}$$

$$f(x) = ax \rightarrow \frac{df}{dx} = a \qquad \boxed{f(x) = a + x \rightarrow \frac{df}{dx} = 1}$$

又一个例子

$$f=\frac{y}{1+e^{-(w_0+w_1x)}}$$

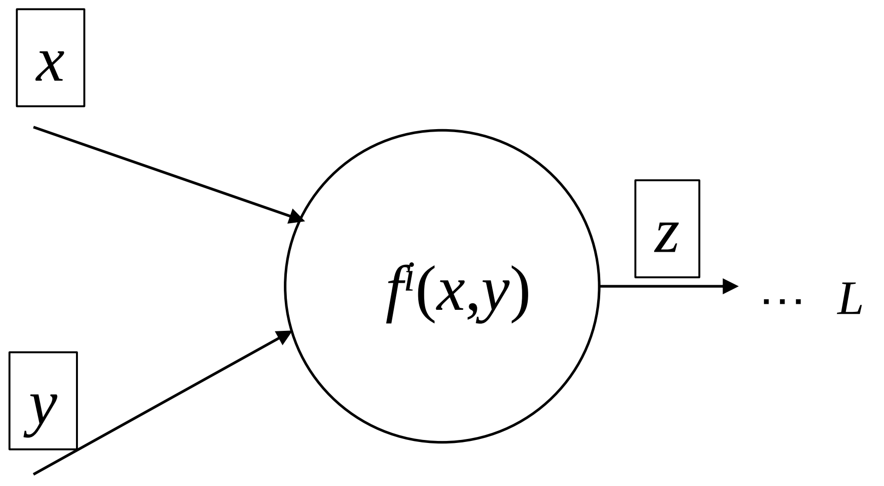
初值： x=-2 w_0=3 w_1=1 y=1



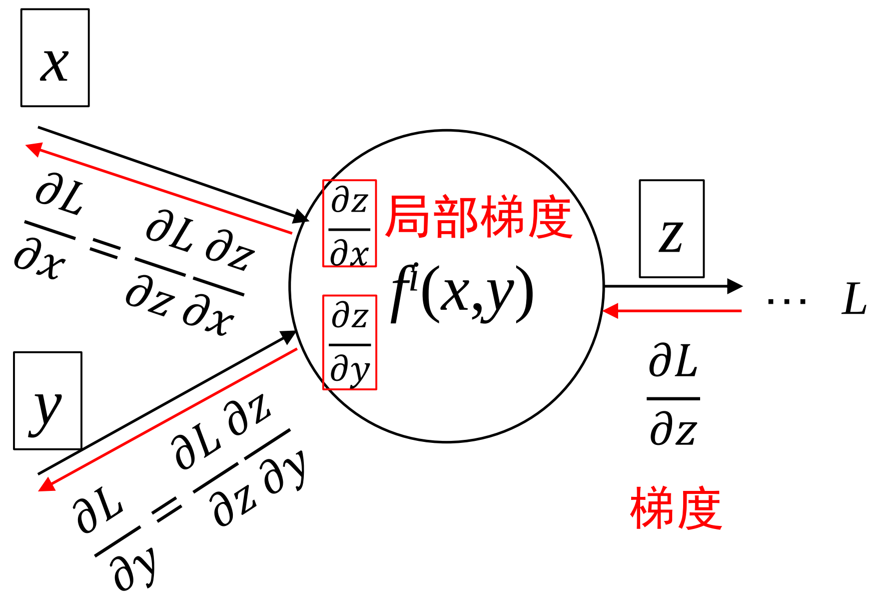
$$f(x)=ex \rightarrow \frac{df}{dx}=ex \qquad f(x)=\frac{1}{x} \rightarrow \frac{df}{dx}=-\frac{1}{x^2}$$

$$\boxed{f(x)=ax \rightarrow \frac{df}{dx}=a} \qquad f(x)=a+x \rightarrow \frac{df}{dx}=1$$

对任意的神经元

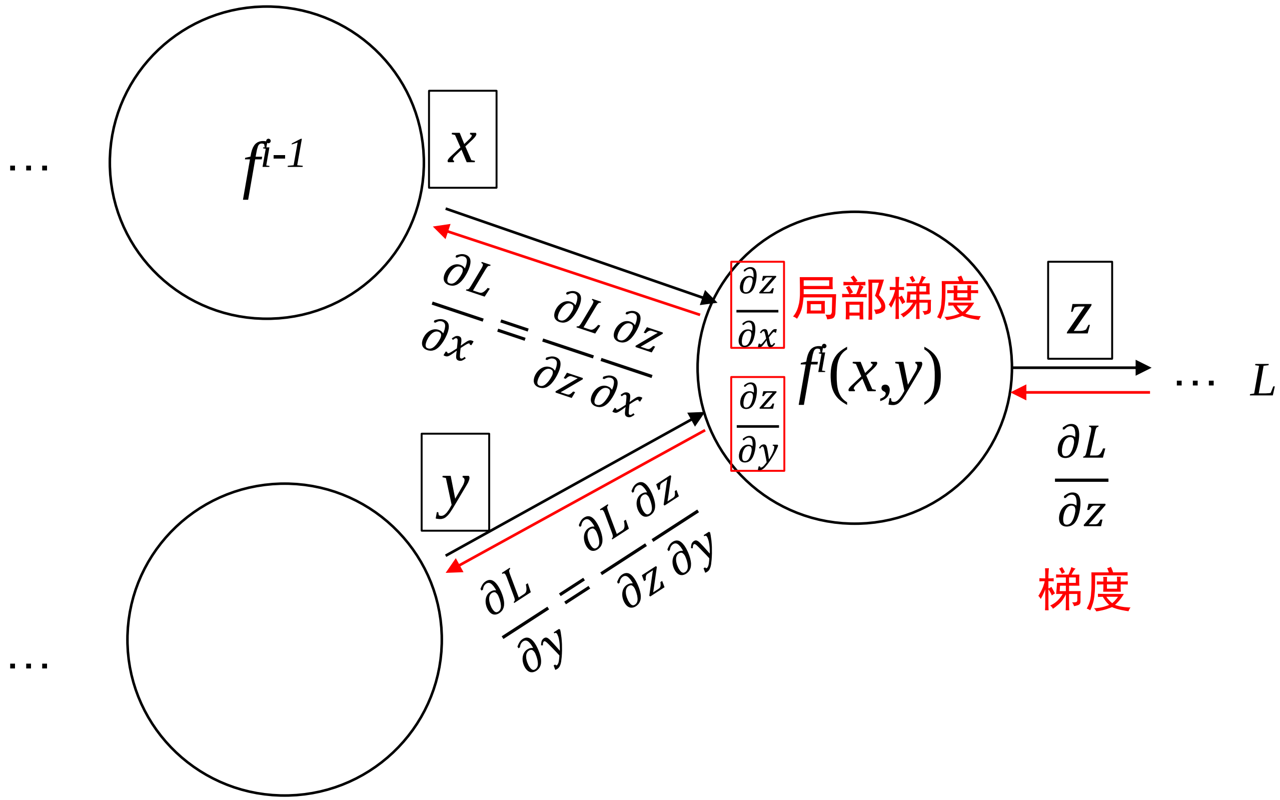


对任意的神经元



x
y
z
$\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial y}$
局部梯度
$f^i(x,y)$
$\cdots\ L$
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
$\frac{\partial L}{\partial z}$
梯度

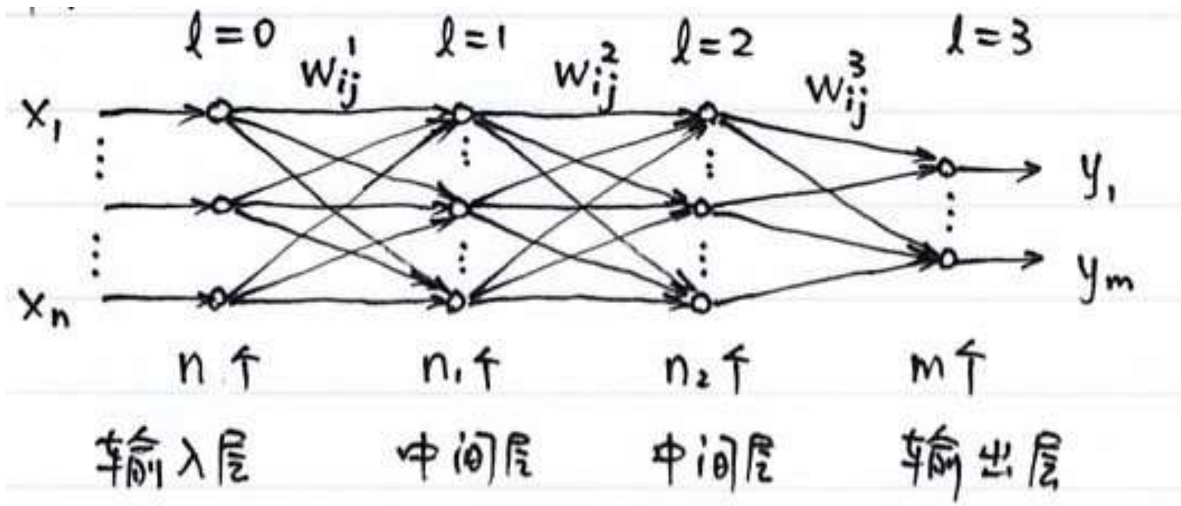
对任意的神经元



…
f^{i-1}
x
$\frac{\partial L}{\partial x} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial x}$
$\frac{\partial z}{\partial x}$
局部梯度
$f^i(x,y)$
$\frac{\partial z}{\partial y}$
z
… L
$\frac{\partial L}{\partial z}$
梯度
y
$\frac{\partial L}{\partial y} = \frac{\partial L}{\partial z}\frac{\partial z}{\partial y}$
…

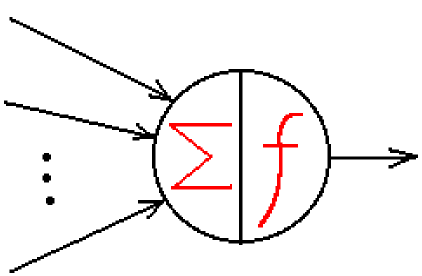
多层感知器与BP算法
(MLP & the Back-Propagation Algorithm)

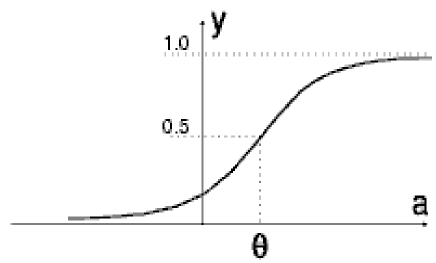
MLP结构



定义误差损失函数： $$E=\frac{1}{2}\sum_{k=1}^{m}(\hat{y}_m-y_{\mathrm{m}})^2$$

f 为节点的**激活函数**，例如采用Sigmoid函数





sigmoid

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

BP算法：

LeCun, 1986; Rumelhart, Hinton & Williams, 1986; Parker, 1985

(1) 权值初始化，$t=0$（用小随机数）

(2) 给出一个训练样本 $x=[x_1,\cdots,x_n]^T\in R^n$ 和期望输出 $D=[y_1,\cdots,y_m]^T\in R^m$

(3) 计算在 x 输入下的实际输出 $Y=[y_1,\cdots,y_m]^T$

(4) 从输出层开始，调整权值，对第 l 层，有

$$w_{ij}^l(t+1)=w_{ij}^l(t)-\eta\delta_j^l x_i^{l-1}\quad,\quad j=1,\cdots,n_l,\quad i=1,\cdots,n_{l-1}$$

其中其中 η 为学习步长，δ_j^l 计算如下：

对输出层：$\delta_j^l=f'(z_j^l)(\hat{y}_j-y_j)$，$j=1,\cdots,m$

对中间层：$\delta_j^l=f'(z_j^l)\sum_{k=1}^{n_{l+1}}\delta_k^{l+1}w_{jk}^{l+1}(t)$，$j=1,\cdots,n_l$

(5) 重新计算输出，考查误差指标（或其它终止条件）
如达到终止条件则终止，否则置 $t=t+1$，转（2）。

说明：

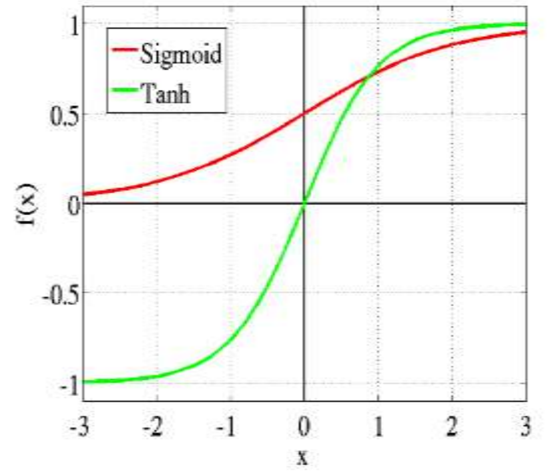
- 算法可能收敛于局部极小点（梯度算法）
- 与初值、步长等的选择有关
- 更与网络结构（结点数目）有关

多凭经验或试验选择

改进：如步长时变，网络结点可剪裁，等等。

关于激活函数

- Sigmoid函数： $f(z)=\frac{1}{(1+e^{-z})}$

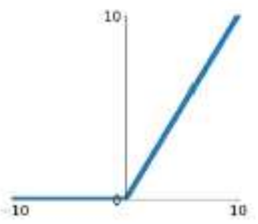


- 双曲正切函数（hyperbolic tangent, Tanh）：

$$f(z)=\frac{(e^{z}-e^{-z})}{(e^{z}+e^{-z})}=\frac{2}{(1+e^{-2z})}-1$$

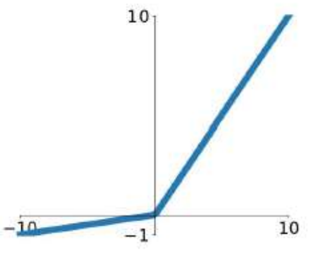
- Rectified Linear Unit (ReLU):

$$f(z)=\max(0,\ z)$$



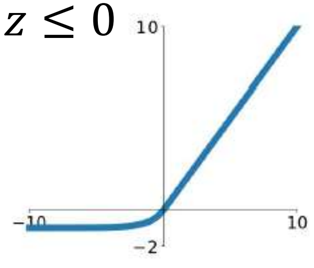
- Leaky-ReLU:

$$f(z)=\begin{cases} z & (z>0) \\ az & (z<0) \end{cases}$$



- Exponential Linear Units （ELU）

$$f(z)=\begin{cases} z & if\ z>0 \\ a(e^{z}-1) & if\ z\le 0 \end{cases}$$

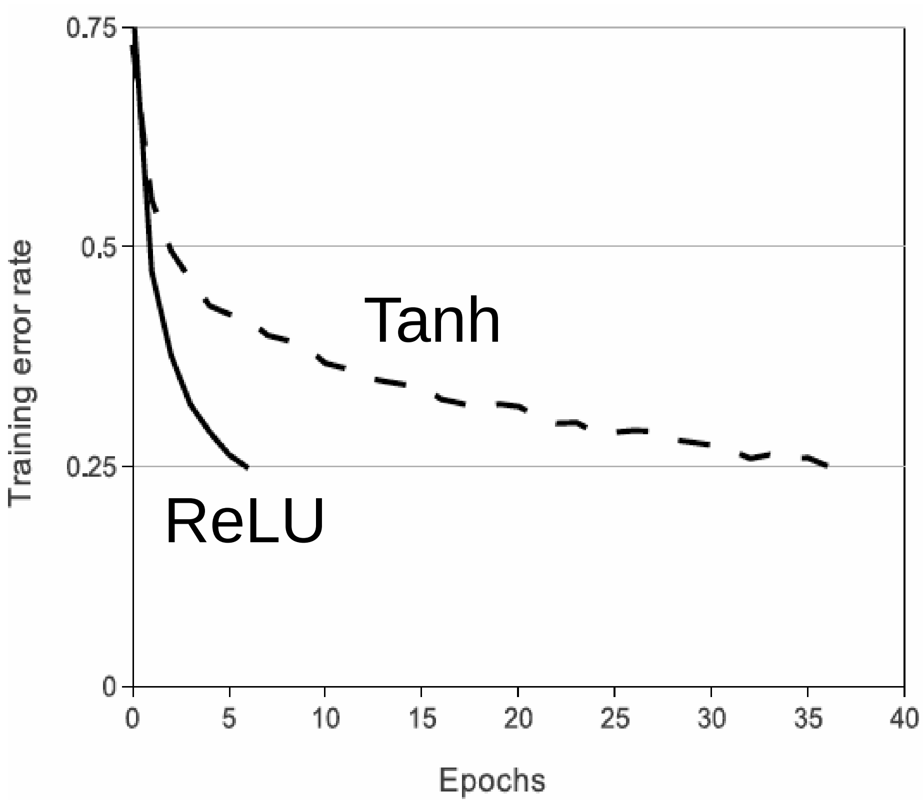


- Maxout:

$$f(z)=\max(w_1^T z+b_1,\ w_2^T z+b_2)$$

ReLU is a good starting point

使用Relu函数通常有较快的收敛速度



Alex Krizhevsky, ICONIP , (2012)

MLP特性：可以实现复杂的非线性映射关系

用于分类：

- 两层网（一个隐层）可实现空间内任意的凸形成区域的划分。
- 三层网（两个隐层）可实现任意形状（连续或不连续）区域划分。

---- 属存在性性质，问题是如何找到这样的**网络结构**？

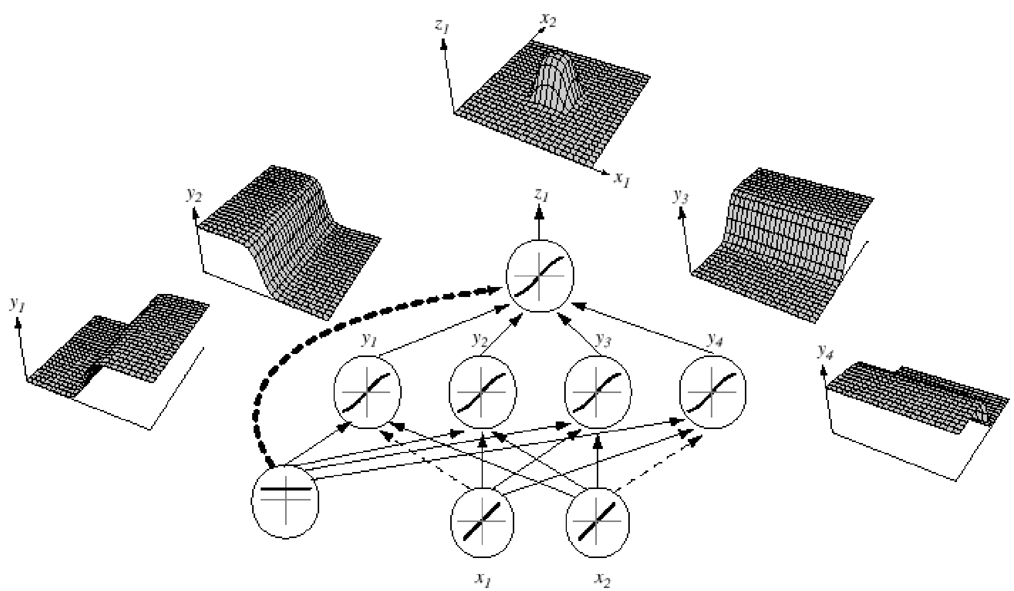


FIGURE 6.2. A 2-4-1 network (with bias) along with the response functions at different units; each hidden output unit has sigmoidal activation function $f(\cdot)$. In the case shown, the hidden unit outputs are paired in opposition thereby producing a "bump" at the output unit. Given a sufficiently large number of hidden units, any continuous function from input to output can be approximated arbitrarily well by such a network. From: Richard O. Duda, Peter E. Hart, and David G. Stork, *Pattern Classification*. Copyright © 2001 by John Wiley & Sons, Inc.

用多层感知器网络实现模式识别

- 输入x—— 样本特征向量（必要时归一化）
- 输出Y—— 类别编码

- 常用输出编码：

  1-of-C编码：

  - *c*类则*c*个输出结点，第*i*类则 $y_i = 1$ 其它 $y_j = 0 \quad i \neq j$

  - 两类：一个输出结点，0、1各代表一类。
  - 也可用*c*个网络解决*c*类问题，每个网络只分一类（是与否）。

其它应用：函数拟合、时间序列预测、数据压缩，……

神经网络研究的“教父”
---加拿大多伦多大学Geoffrey E. Hinton教授

坚持人工神经网络研究40余年

深度学习的开创者

http://www.cs.toronto.edu/~hinton/

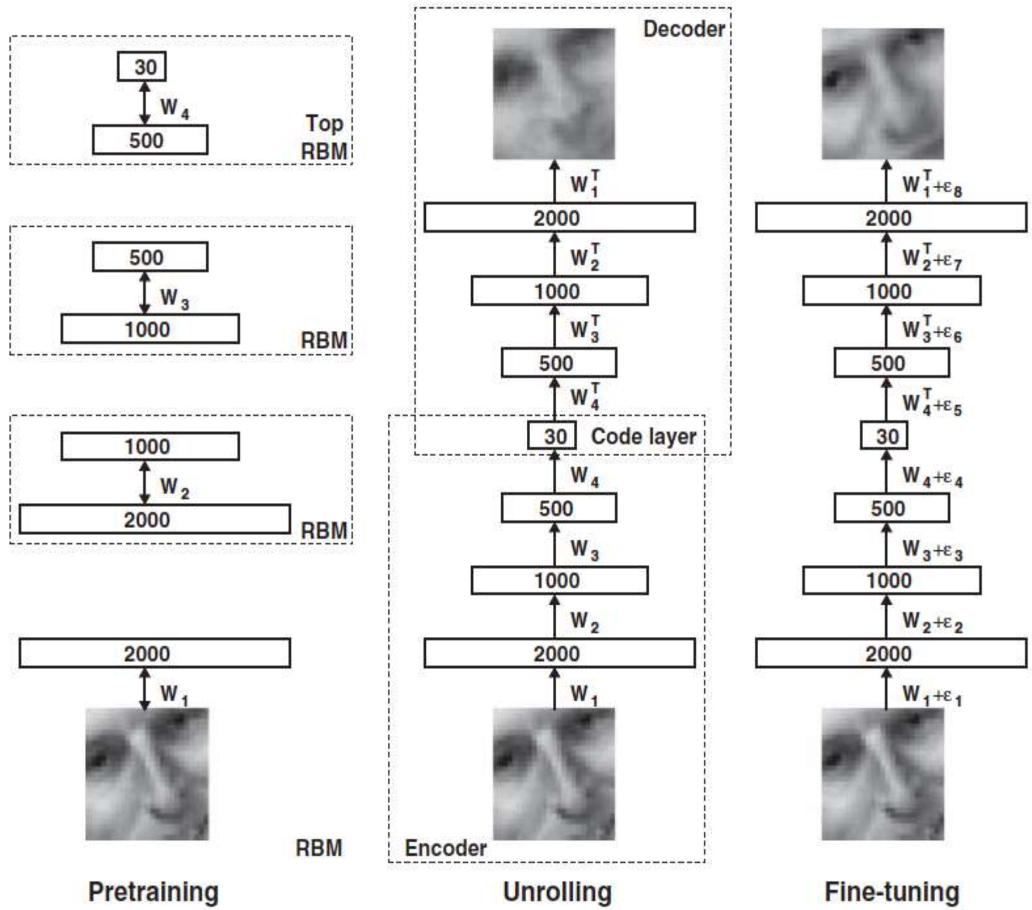
深度神经网络的特征表示

Reducing the Dimensionality of Data with Neural Networks

G. E. Hinton* and R. R. Salakhutdinov

2006 Science

“Autoencoder”

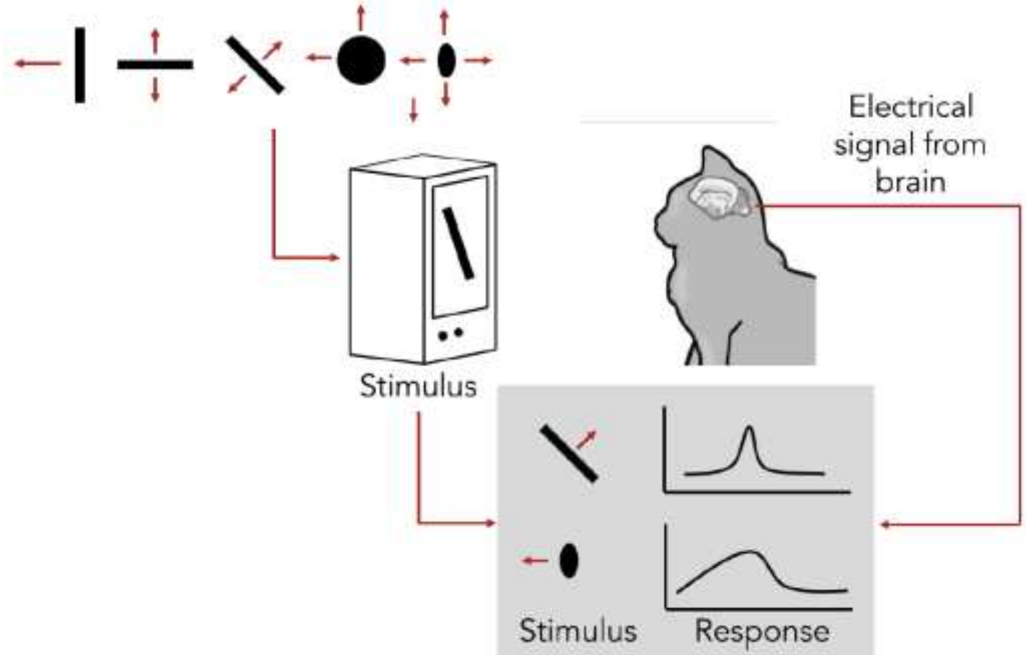


RBM：受限玻尔兹曼机
(Restricted Boltzmann Machine)

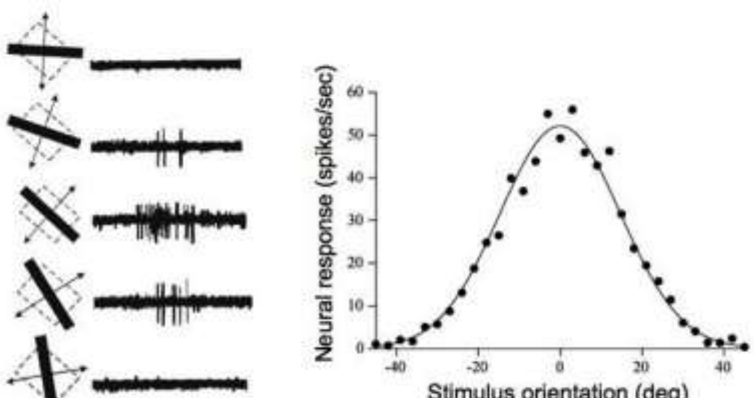
RBM不区分前向和反向，可理解为编码和解码过程

视觉神经系统模型

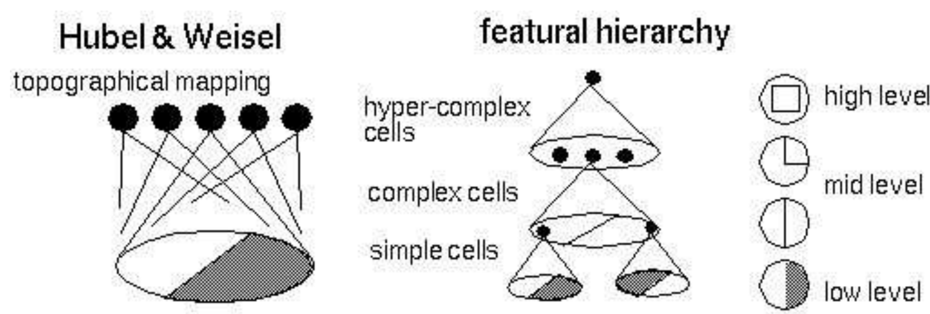
Hubel & Wiesel,
1981 Nobel Prize in Physiology or Medicine



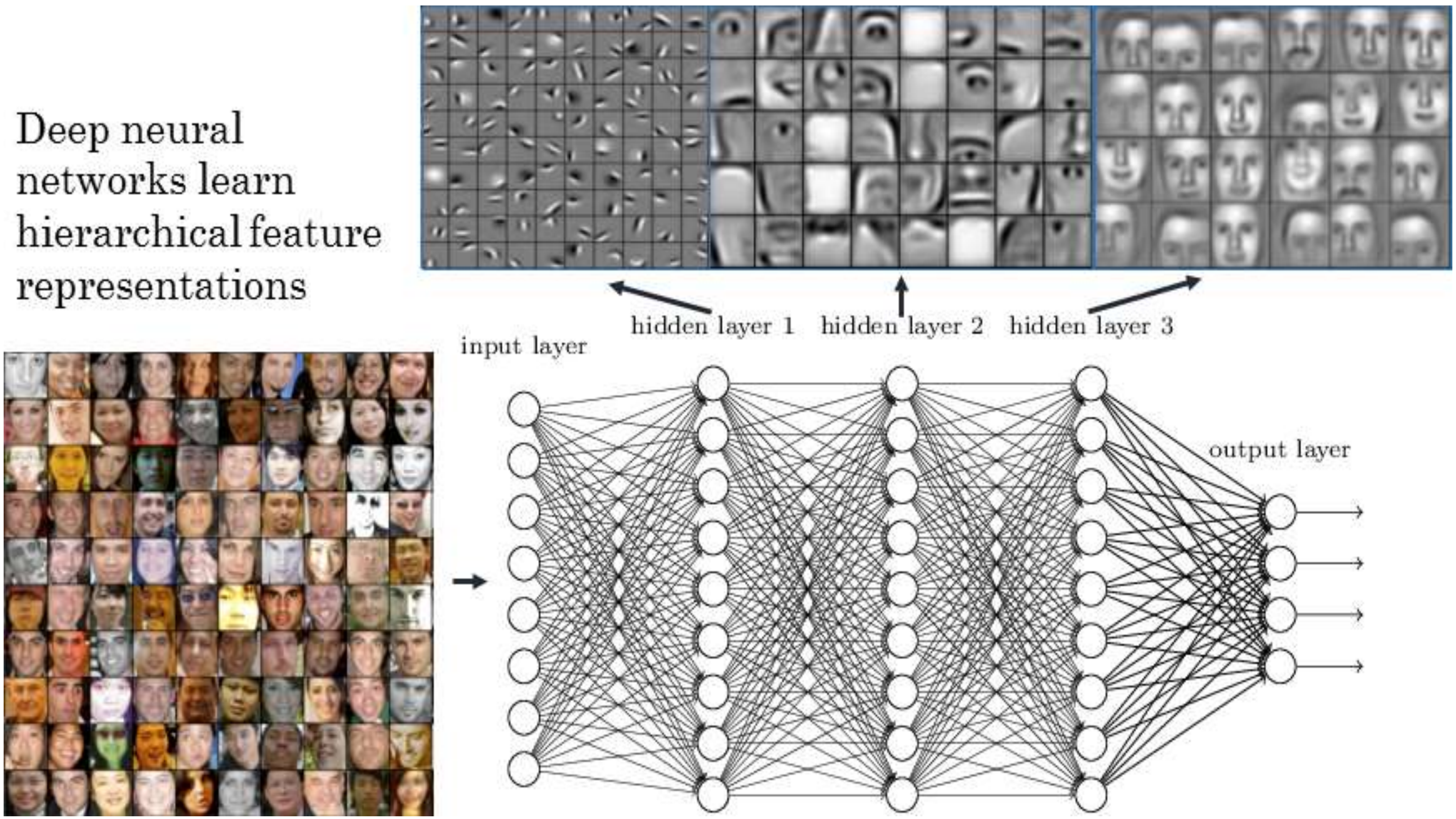
slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**



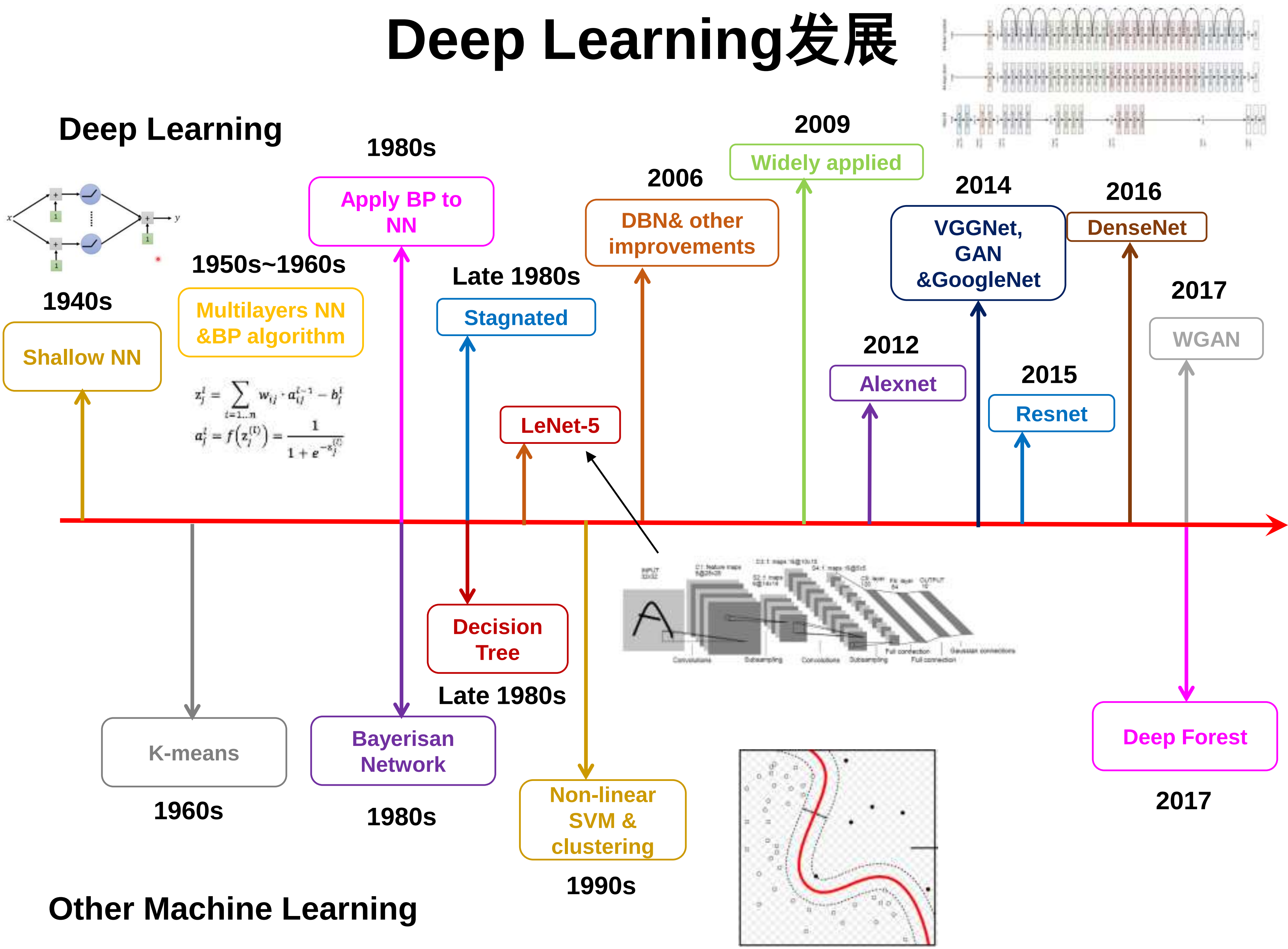
Hubel & Wiesel, 1968

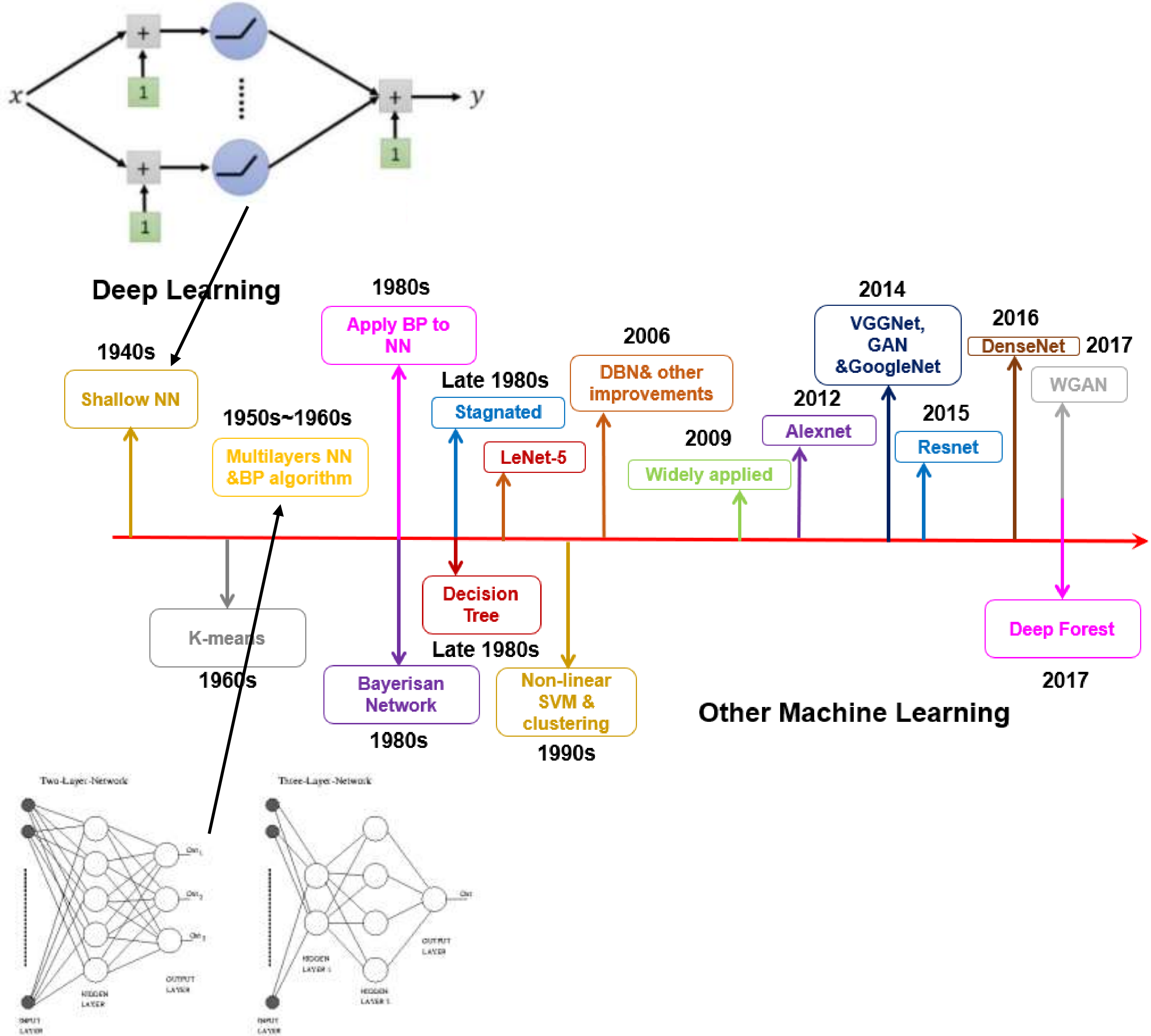


层次化的特征表示

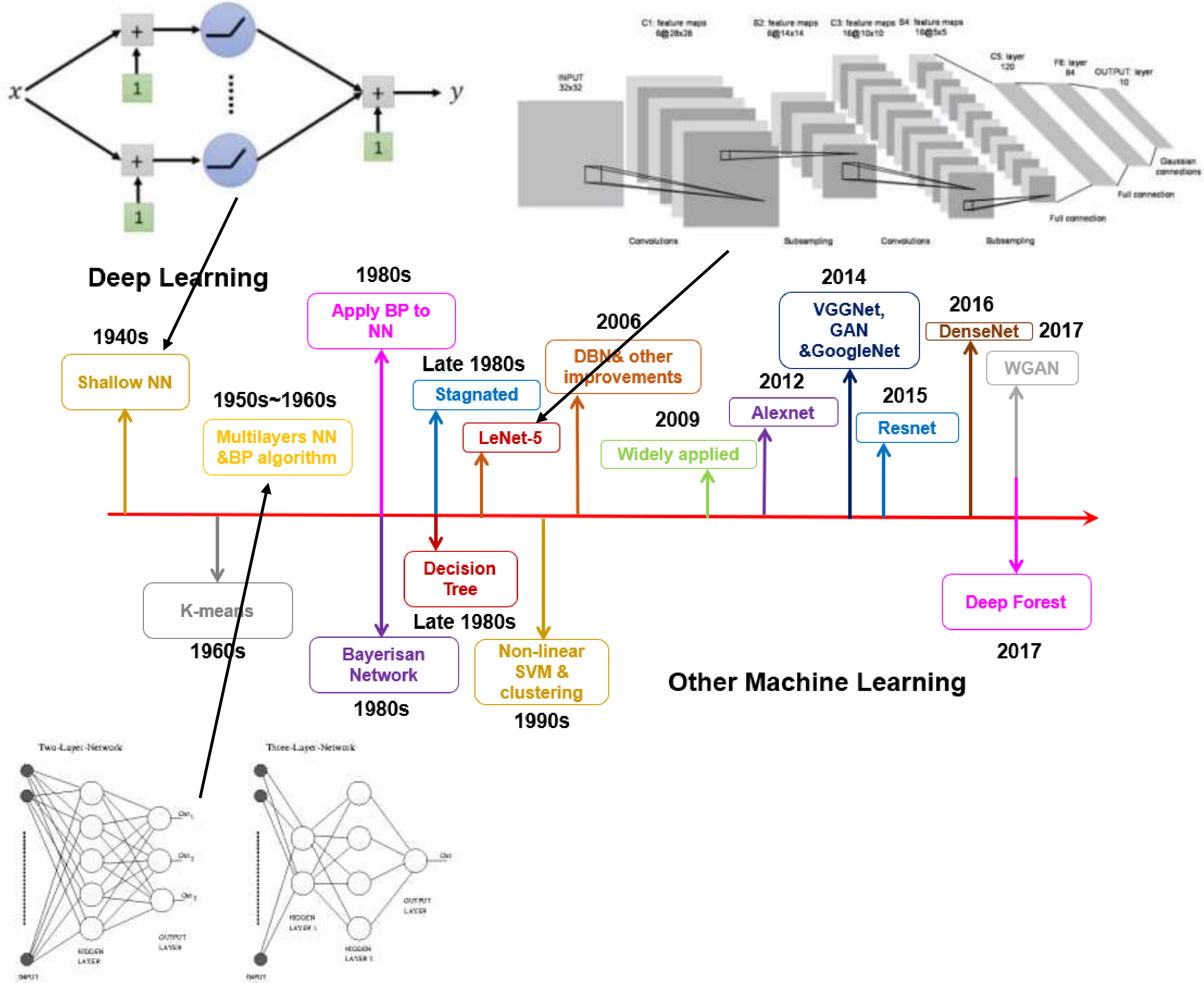

Deep neural networks learn hierarchical feature representations
hidden layer 1
hidden layer 2
hidden layer 3
input layer
output layer

Deep Learning发展

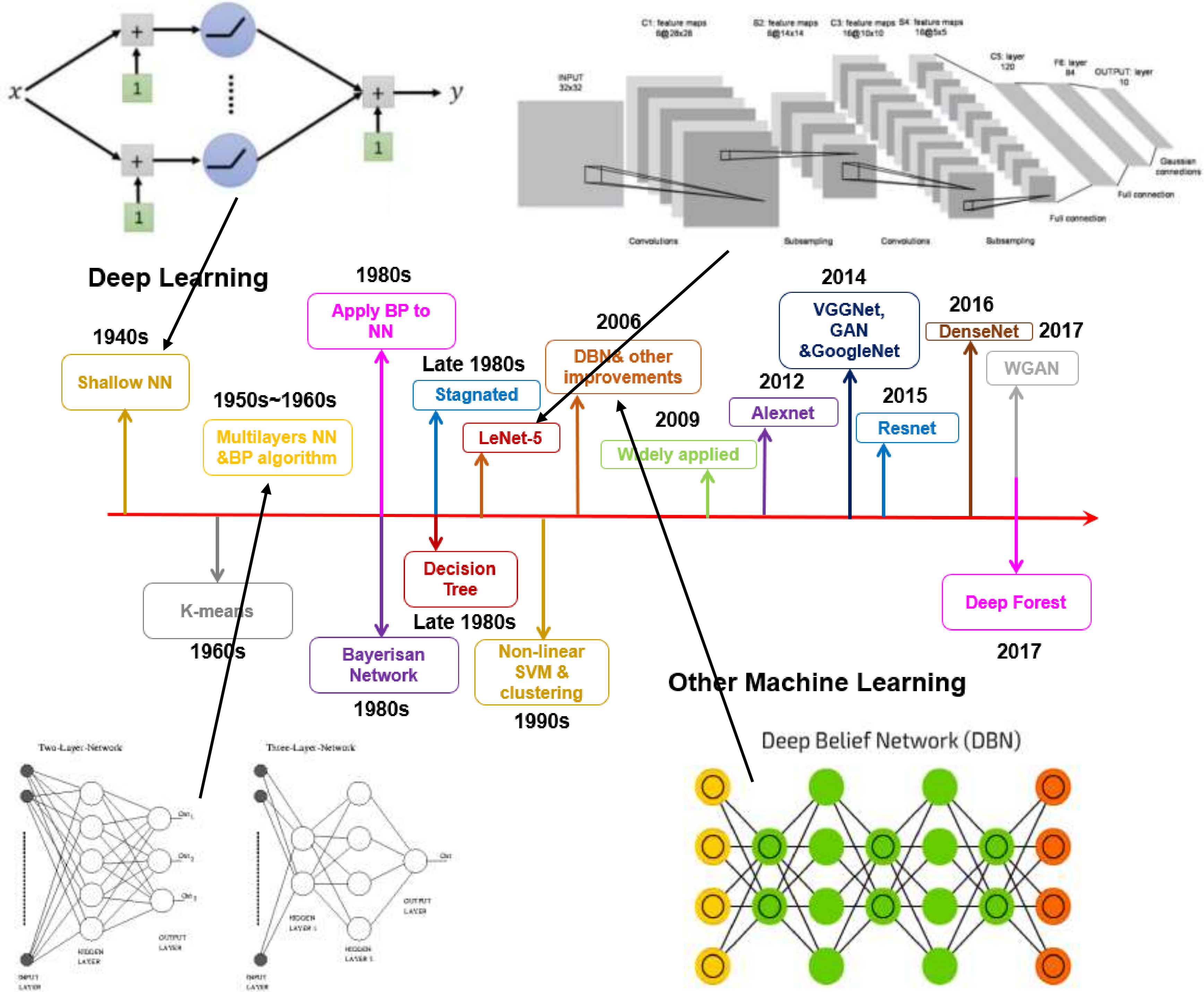


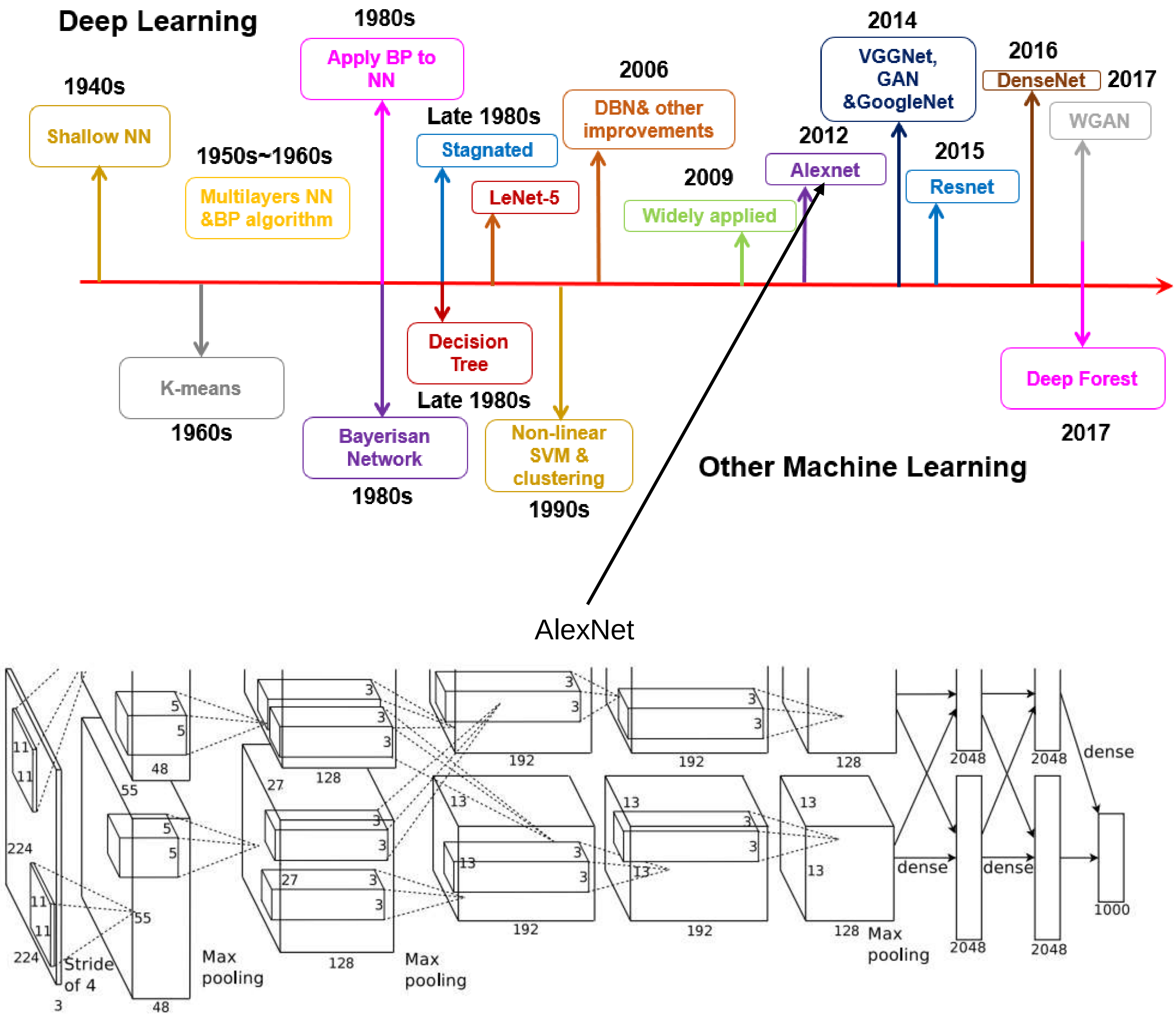


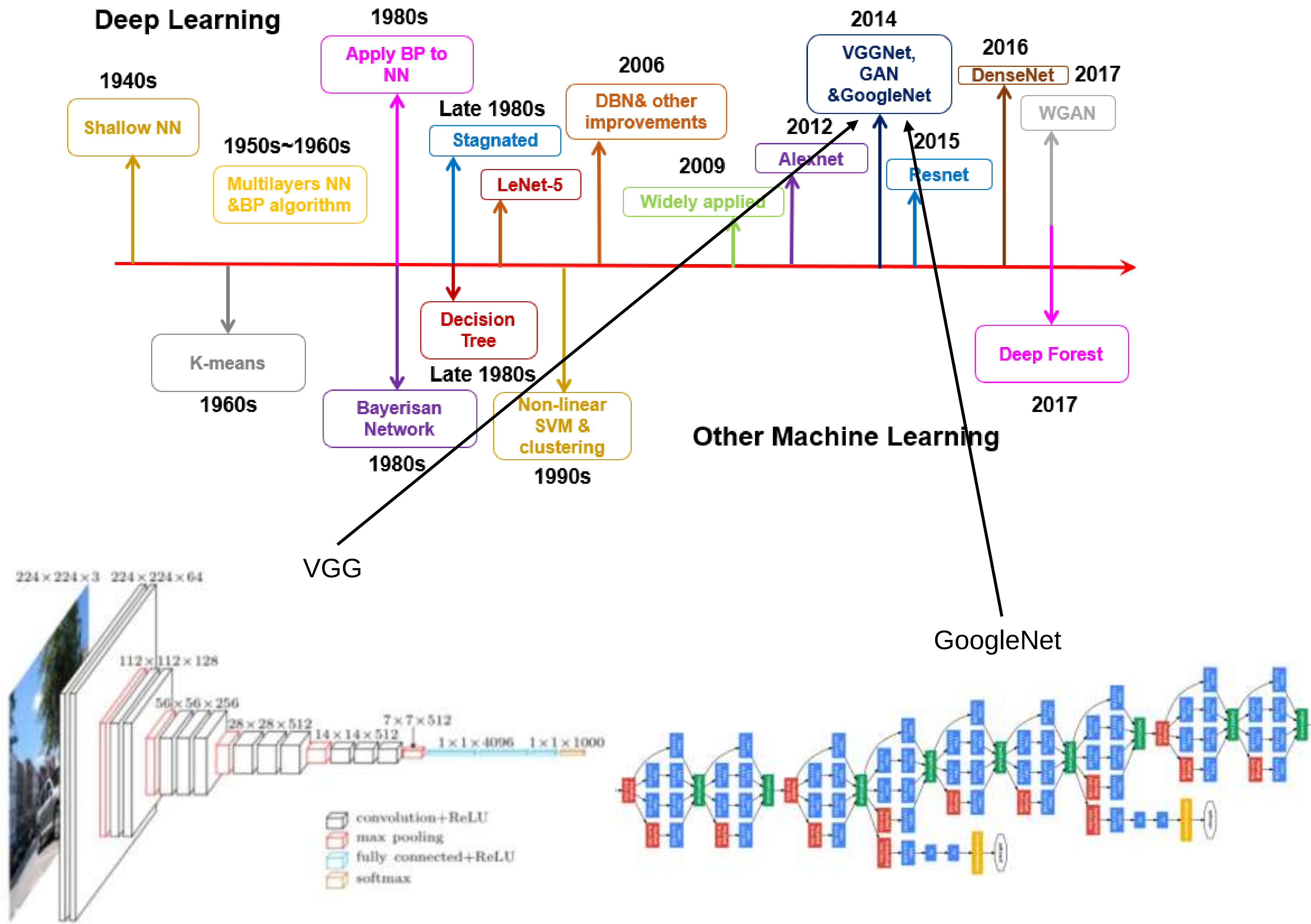
x
y
1
Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN
&BP algorithm
1980s
Apply BP to
NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other
improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet,
GAN
&GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan
Network
1980s
Decision
Tree
Late 1980s
Non-linear
SVM &
clustering
1990s
Deep Forest
2017
Other Machine Learning
Two-Layer-Network
Three-Layer-Network



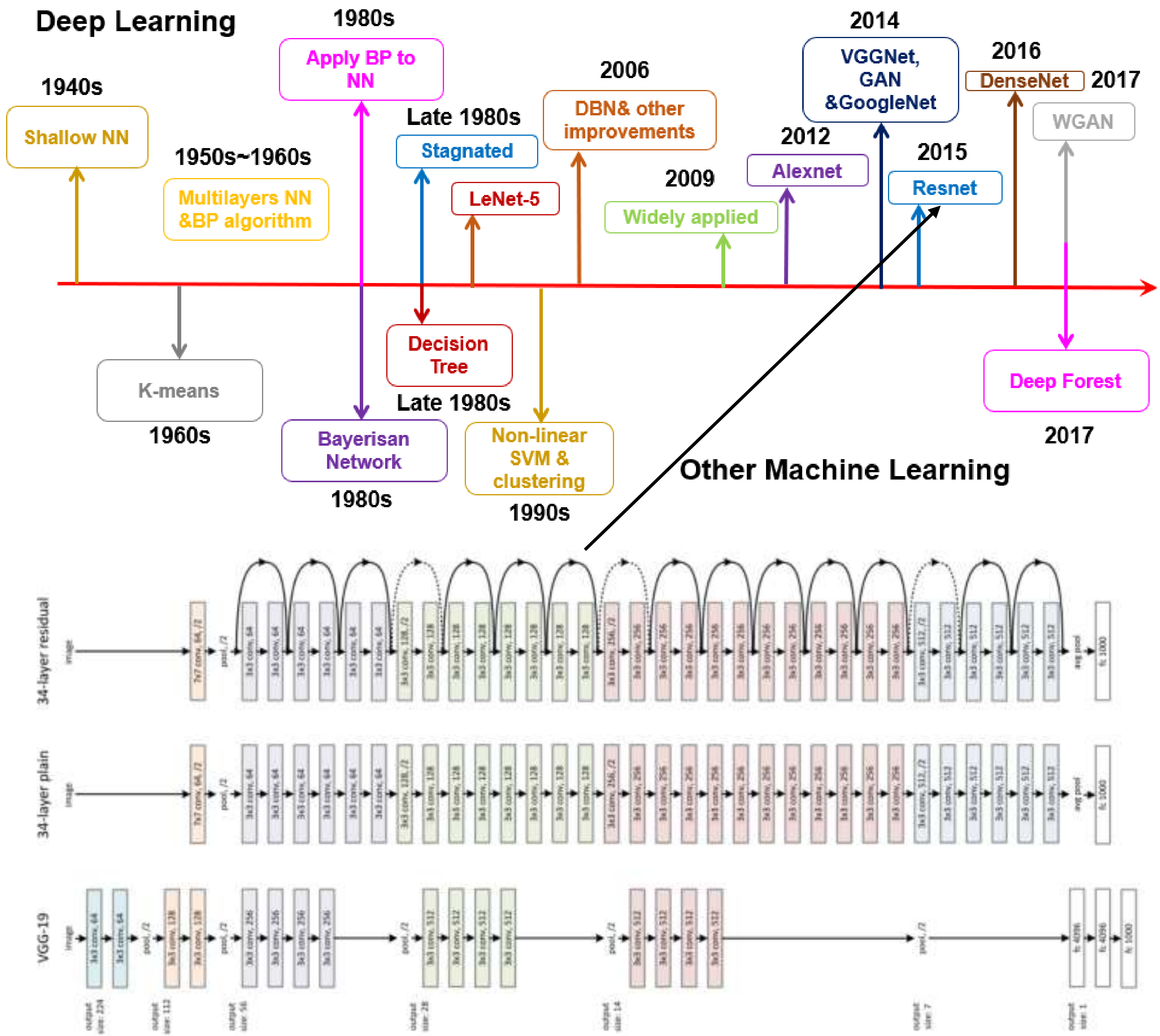
Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN &BP algorithm
1980s
Apply BP to NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet, GAN &GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan Network
1980s
Decision Tree
Late 1980s
Non-linear SVM & clustering
1990s
Deep Forest
2017
Other Machine Learning
Two-Layer-Network
Three-Layer-Network


Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN &BP algorithm
1980s
Apply BP to NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet, GAN &GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan Network
1980s
Decision Tree
Late 1980s
Non-linear SVM & clustering
1990s
Deep Forest
2017
Other Machine Learning
Deep Belief Network (DBN)
Two-Layer-Network
Three-Layer-Network
INPUT 32x32
C1: feature maps 6@28x28
S2: feature maps 6@14x14
C3: feature maps 16@10x10
S4: feature maps 16@5x5
C5: layer 120
F6: layer 84
OUTPUT layer 10
Convolutions
Subsampling
Convolutions
Subsampling
Full connection
Full connection
Gaussian connections

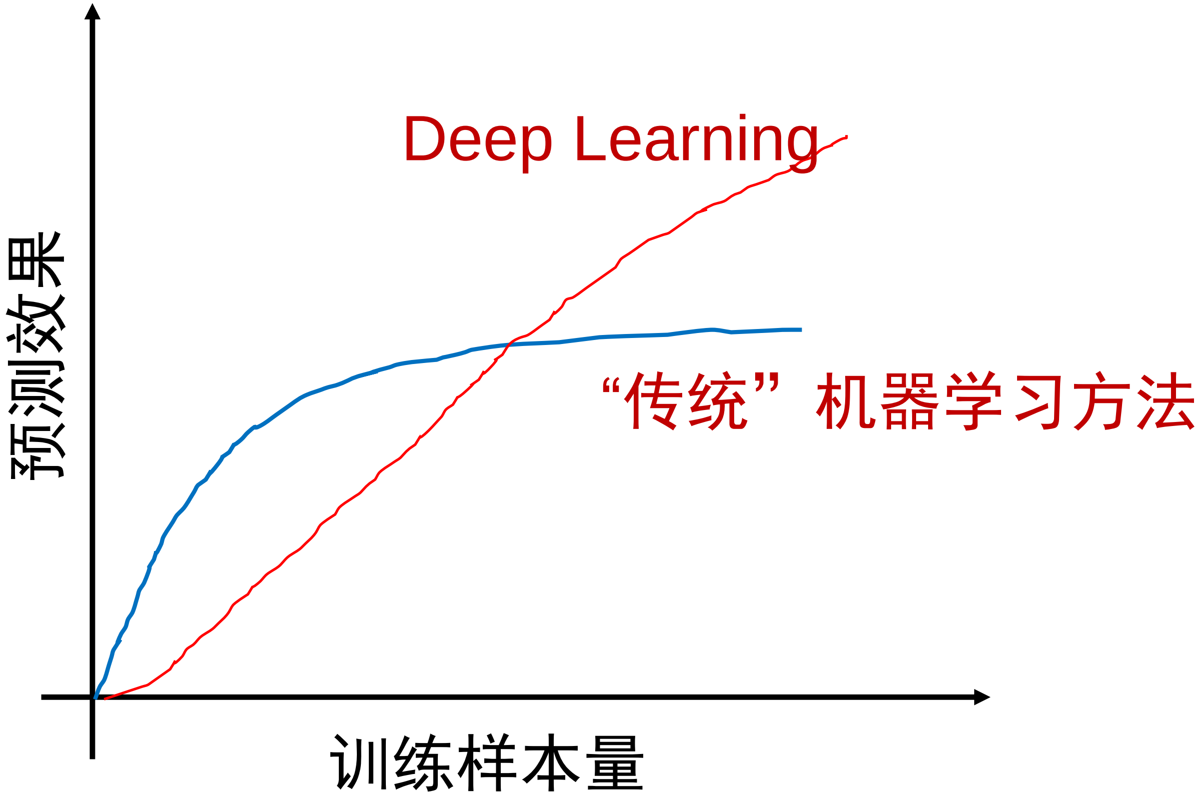

Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN &BP algorithm
1980s
Apply BP to NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet, GAN &GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan Network
1980s
Decision Tree
Late 1980s
Non-linear SVM & clustering
1990s
Deep Forest
2017
Other Machine Learning
AlexNet
224
224
3
11
11
Stride of 4
55
55
48
48
5
5
27
27
128
128
3
3
Max pooling
13
13
192
192
128
128
2048
2048
2048
2048
dense
dense
dense
1000
Max pooling
Max pooling



Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN &BP algorithm
1980s
Apply BP to NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet, GAN &GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan Network
1980s
Decision Tree
Late 1980s
Non-linear SVM & clustering
1990s
Deep Forest
2017
Other Machine Learning
VGG
GoogleNet
224 × 224 × 3
224 × 224 × 64
112 × 112 × 128
56 × 56 × 256
28 × 28 × 512
14 × 14 × 512
7 × 7 × 512
1 × 1 × 4096
1 × 1 × 1000
convolution+ReLU
max pooling
fully connected+ReLU
softmax

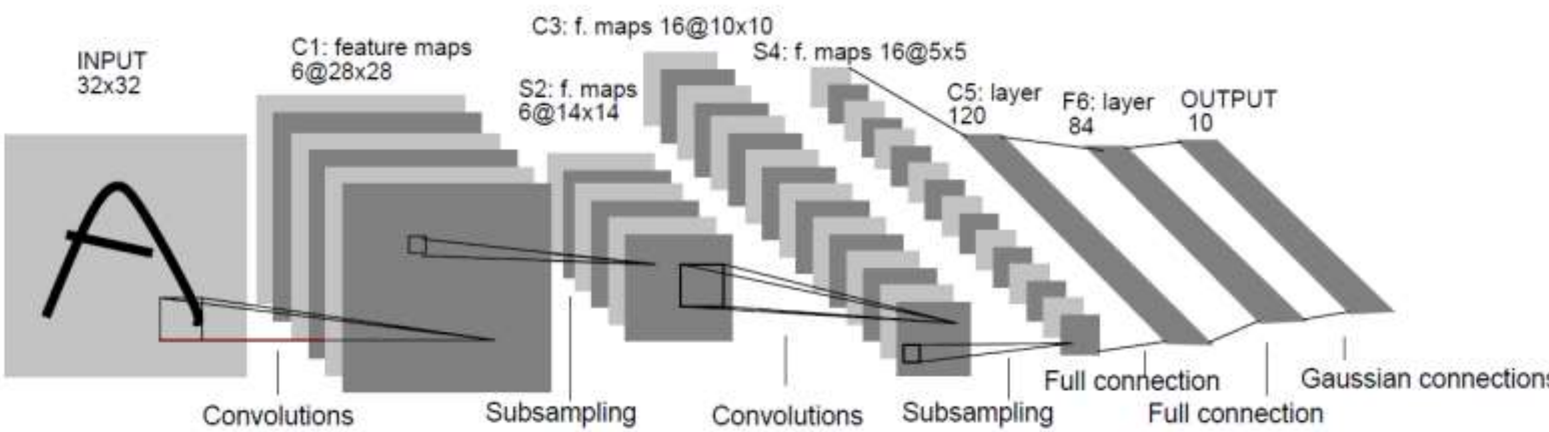

Deep Learning
1940s
Shallow NN
1950s~1960s
Multilayers NN &BP algorithm
1980s
Apply BP to NN
Late 1980s
Stagnated
LeNet-5
2006
DBN& other improvements
2009
Widely applied
2012
Alexnet
2014
VGGNet, GAN &GoogleNet
2015
Resnet
2016
DenseNet
2017
WGAN
K-means
1960s
Bayerisan Network
1980s
Decision Tree
Late 1980s
Non-linear SVM & clustering
1990s
Deep Forest
2017
Other Machine Learning
34-layer residual
34-layer plain
VGG-19

Deep Learing v.s. Classical methods



卷积神经网络

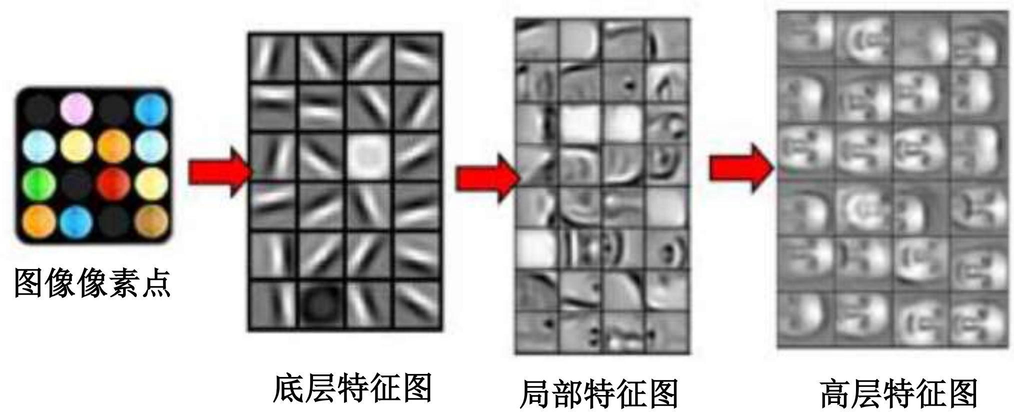
- 卷积神经网络（Convolutional Neural Network，CNN）是一种前馈神经网络，近年来被大量应用于计算机视觉、自然语言处理等领域。
- CNN的研究始于二十世纪末，时间延迟网络和LeNet-5是最早出现的卷积神经网络，近年来卷积神经网络的规模逐渐增大，例如AlphaGo有超过40层的卷积神经网络。



LeNet-5结构，LeCun, et al. *Proceedings of the IEEE* (1998).

卷积神经网络

- 卷积神经网络被广泛使用在图像分类、物体识别等图像领域任务中；
- 卷积神经网络使用卷积核(kernel)捕捉图像特征；
- 卷积核可以自底向上，层级提取图像特征，如下图：



Convolutional neuron networks (CNN)

- LeNet：convolutional network designed for handwritten and machine-printed character recognition

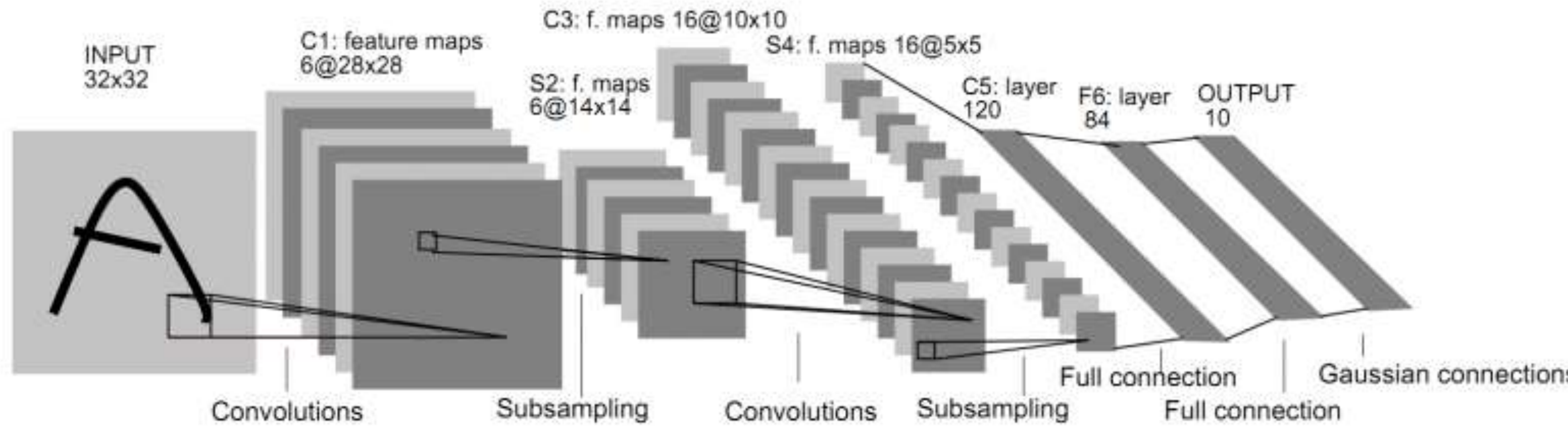
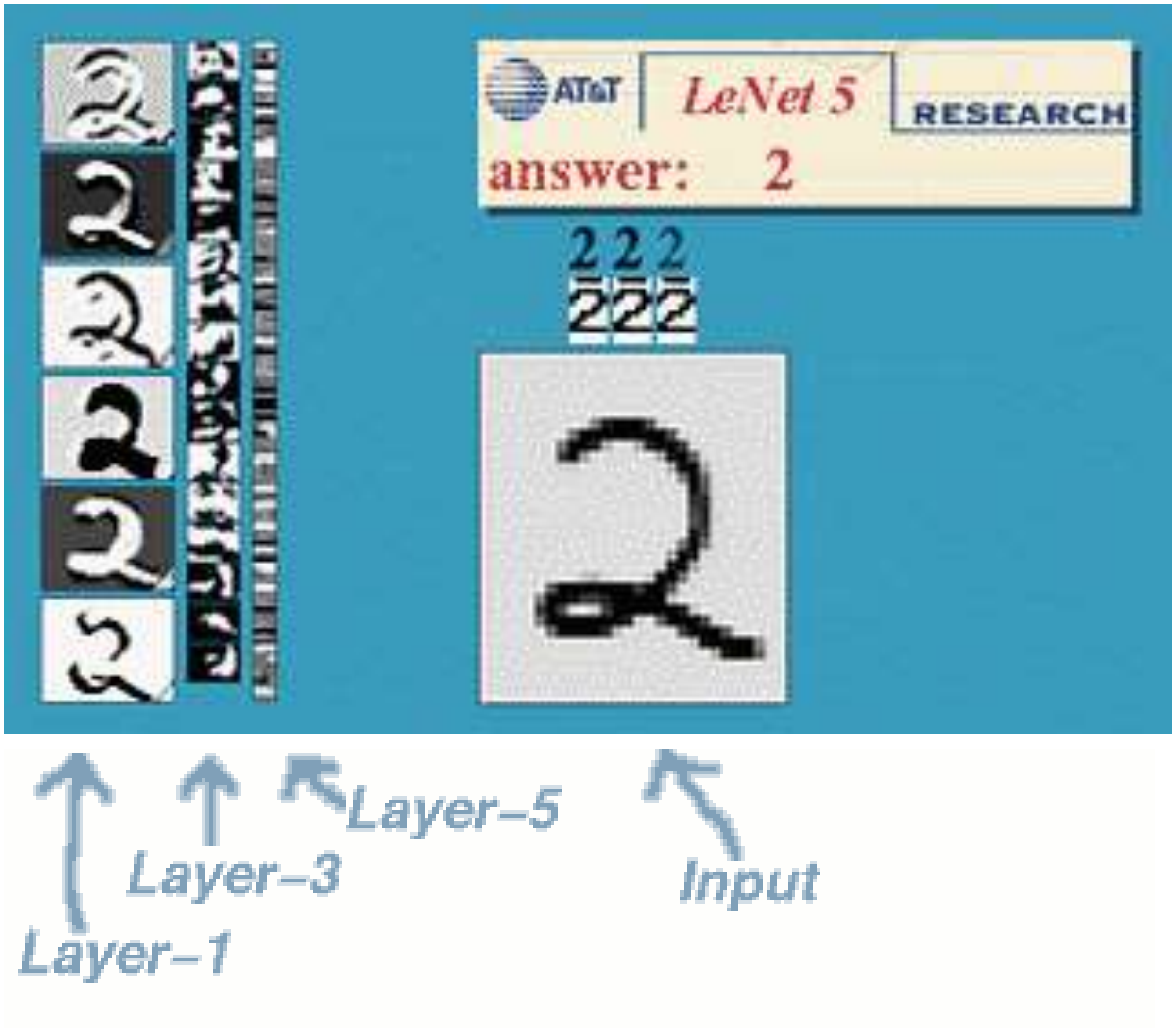


Fig. 2. Architecture of LeNet-5, a Convolutional Neural Network, here for digits recognition. Each plane is a feature map, i.e. a set of units whose weights are constrained to be identical.

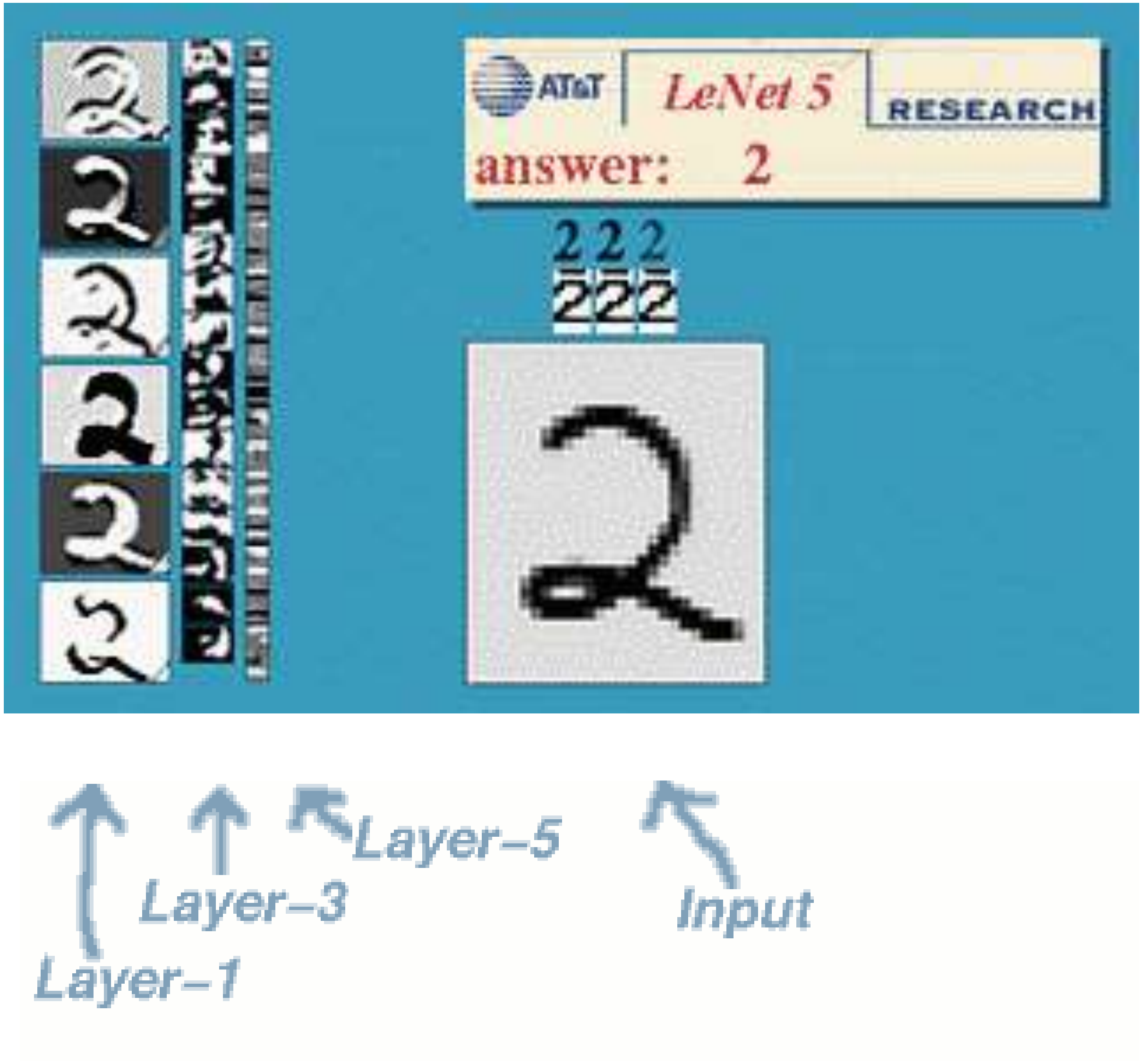
LeCun, Y., Bottou, L., Bengio, Y., and Haffner, P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE, 86(11), 2278–2324.

LeNet：convolutional network designed for handwritten and machine-printed character recognition



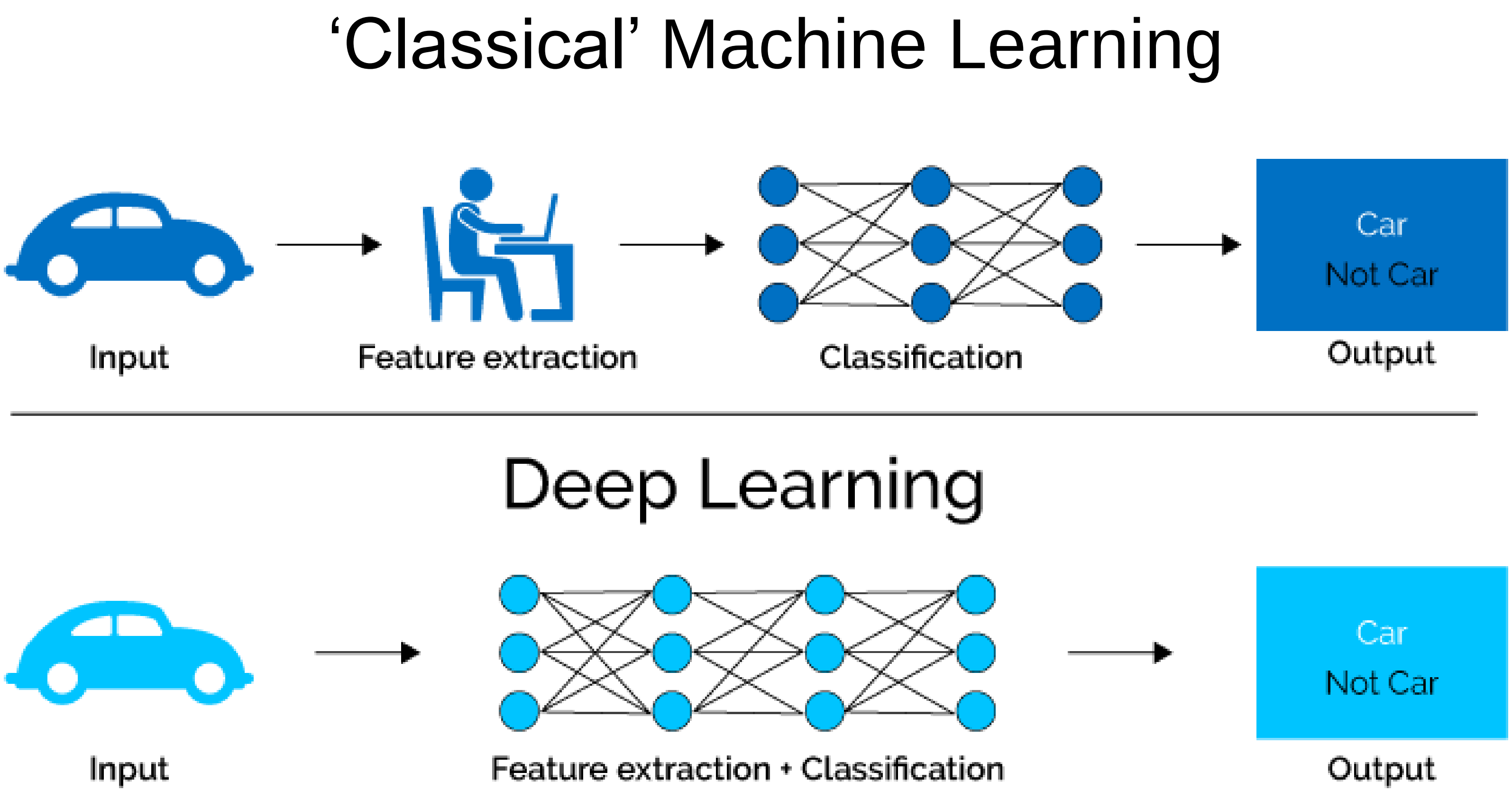
http://yann.lecun.com/exdb/lenet/index.html

LeNet：convolutional network designed for handwritten and machine-printed character recognition



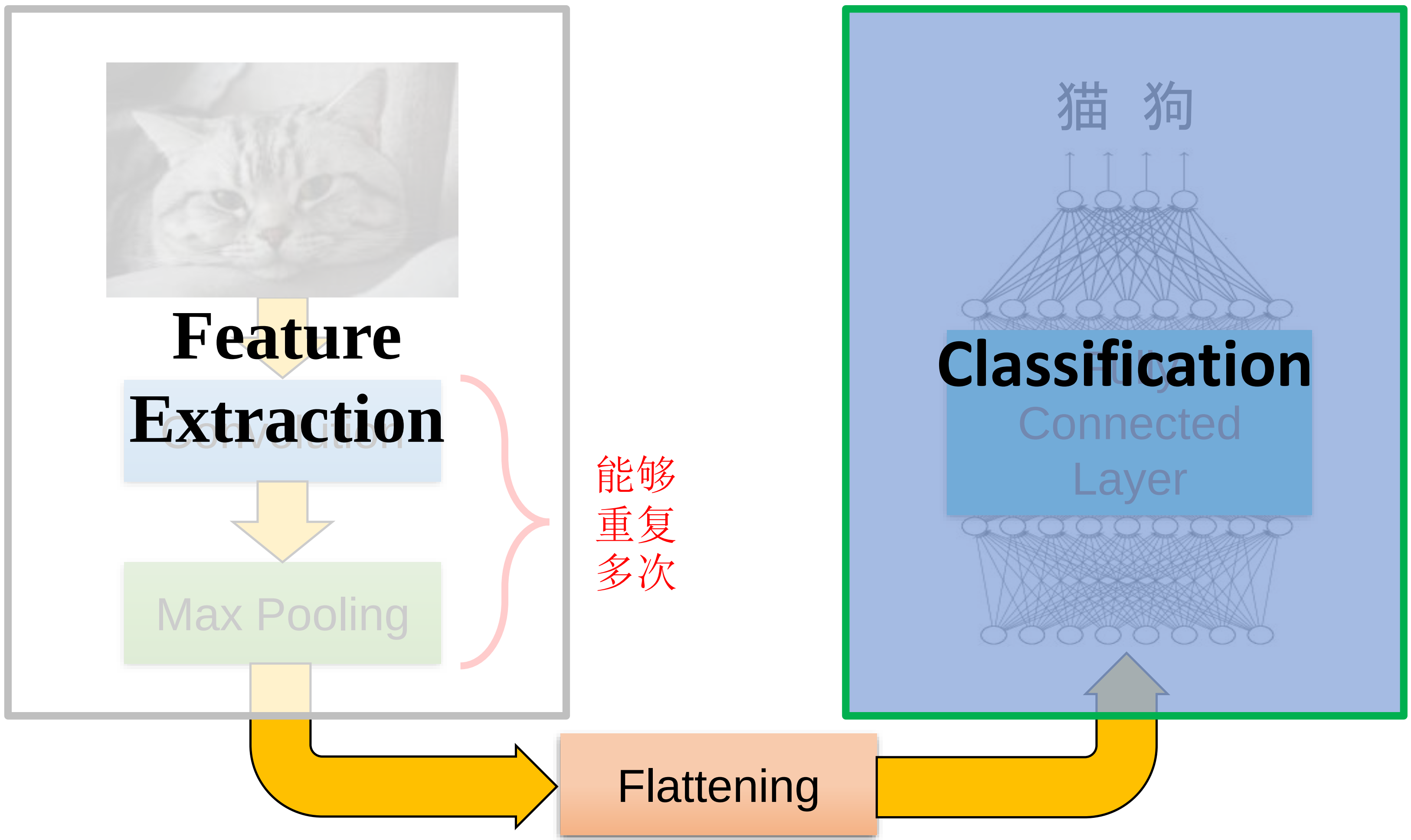
http://yann.lecun.com/exdb/lenet/index.html

端到端（end-to-end）



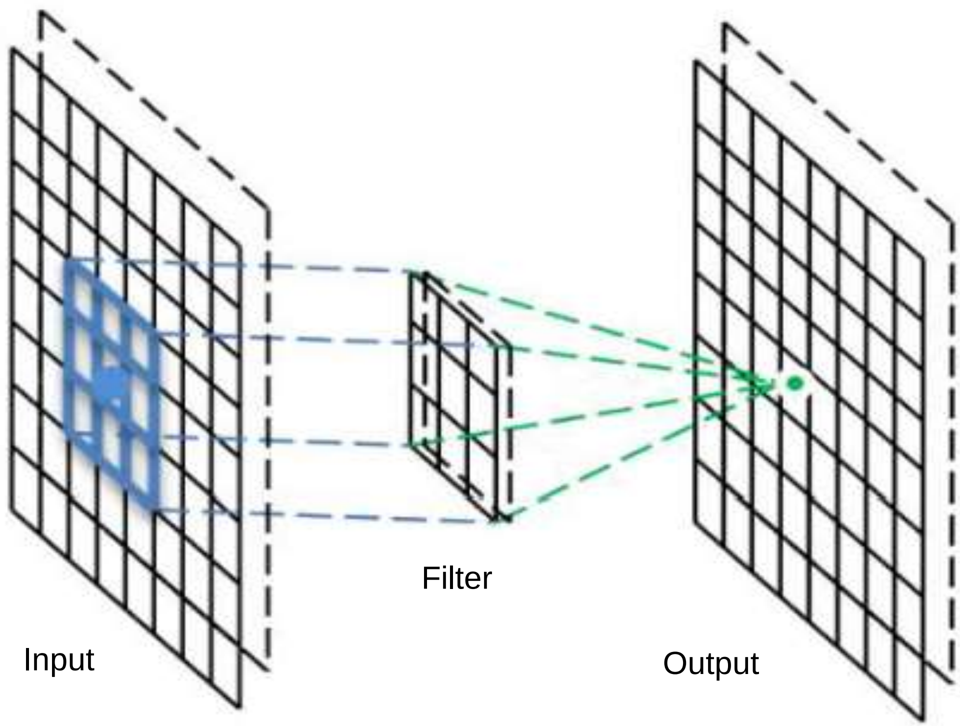
图片来源：https://towardsdatascience.com/why-deep-learning-is-needed-over-traditional-machine-learning-1b6a99177063

CNN在图像识别中的应用



卷积层（Convolutional layer）

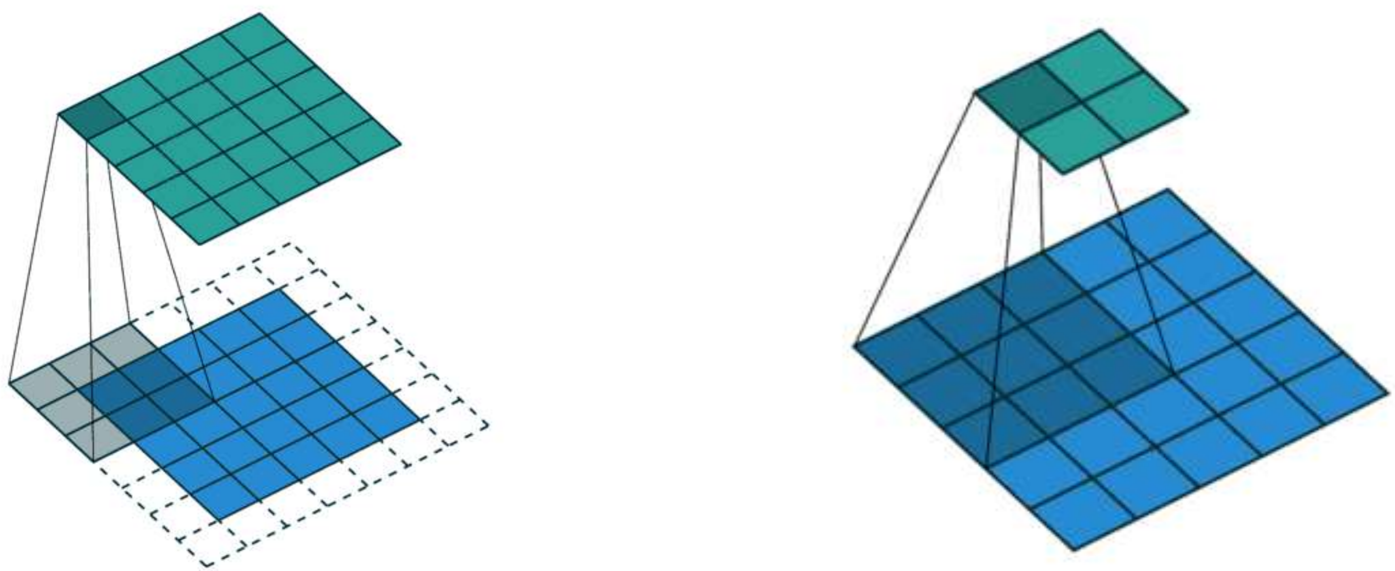
- 卷积层是卷积神经网络的核心部分，一个卷积层通常含有数个卷积核（filters）来执行卷积操作。



CNN 示意图，Poletaev, et al. Journal of Physics: Conference Series(2016).

卷积层

- CNN中常用Padding和Strides来控制输出图像的大小。
  - Padding: 用额外的像素点来填充边缘，使得输出大小等于输入大小。
  - Strides: 用来调整卷积的步长，产生不同尺寸的输出。



左：Padding， 右：strides=2，Dumoulin, et al. arXiv:1603.07285(2016).

卷积层

- 卷积层通过卷积操作来捕获filters关注的信息。

1	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	0	1
1	0	1	0	1	1
0	0	0	0	1	1
0	0	1	0	1	1

6X6图像

1	0	0
0	1	0
1	0	0

0	0	1
0	0	1
0	0	1

⋮ ⋮

每个filter检测3X3的pattern。

卷积层

- 卷积层通过卷积操作来捕获filters关注的信息。

Filter1

1	0	0
0	1	0
1	0	0

1	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	0	1
1	0	1	0	1	1
0	0	0	0	1	1
0	0	1	0	1	1

6X6图像

3	0	0	1
1	1	1	0
1	1	0	1
1	0	2	1

卷积层

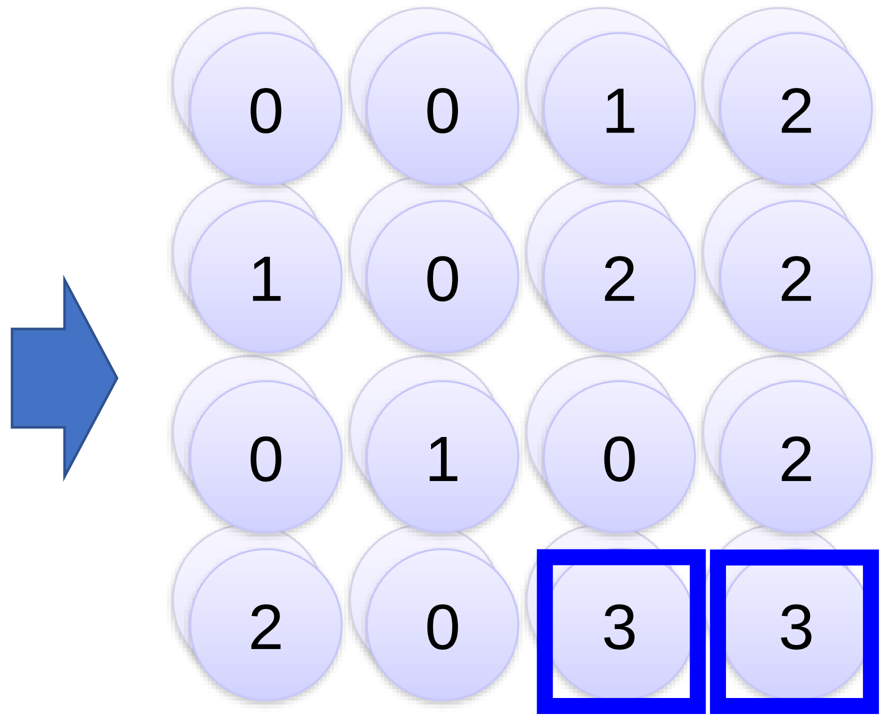
- 卷积层通过卷积操作来捕获filters关注的信息。

Filter2

0	0	1
0	0	1
0	0	1

1	0	0	0	0	1
0	1	0	0	1	0
1	0	0	0	0	1
1	0	1	0	1	1
0	0	0	0	1	1
0	0	1	0	1	1

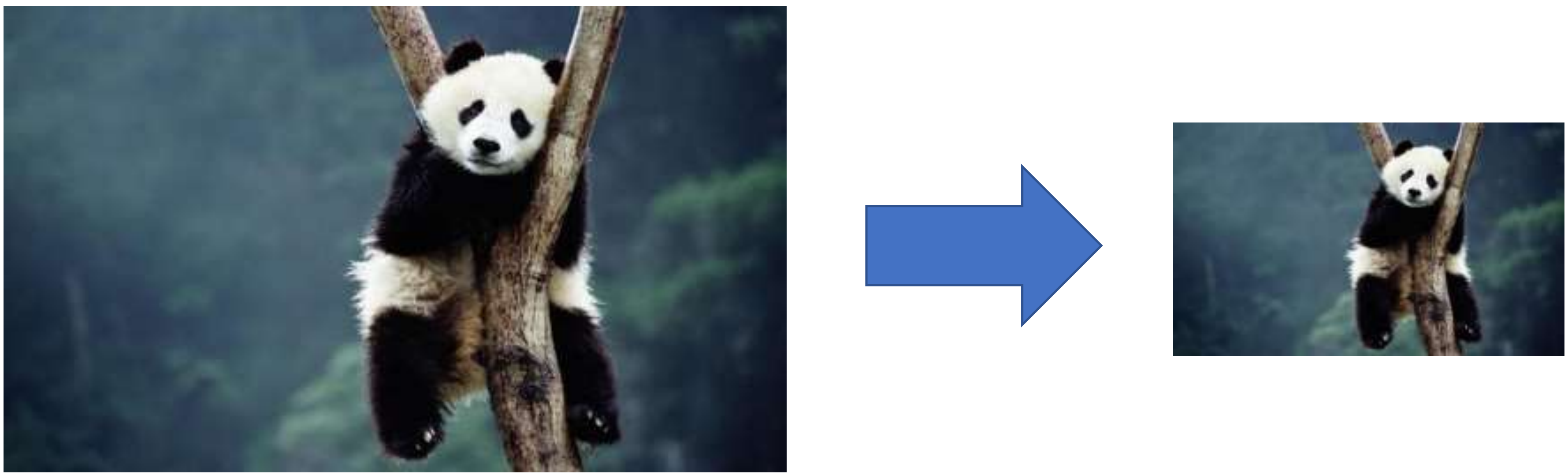
6X6图像



2个4X4的图像构成了
一个2X4X4的图像

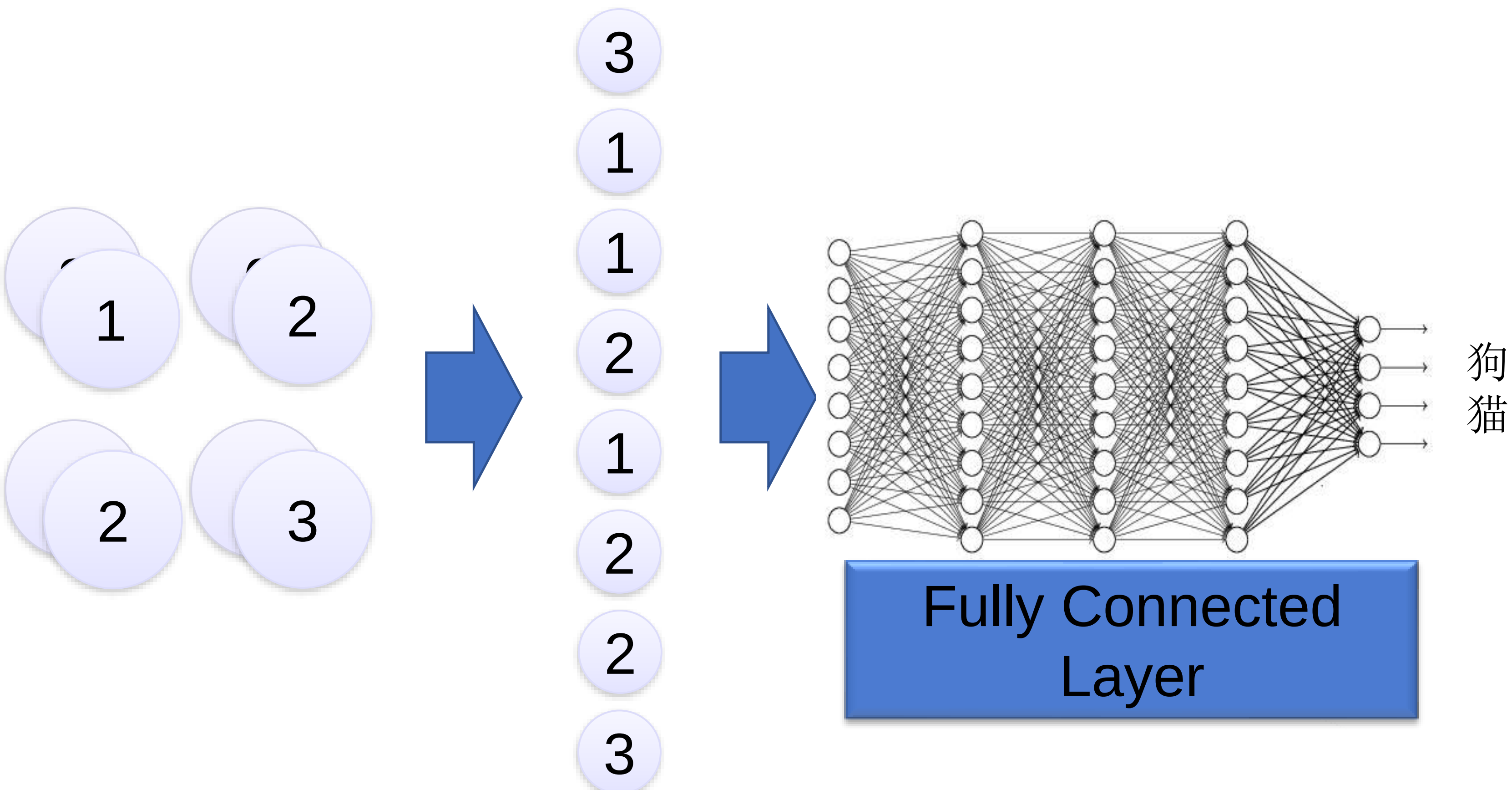
池化层（Pooling layer）

- 对像素的抽样可以在一定程度上保持原始对象，并且减少描述对象所需要的参数。



对图片进行抽样

Flattening和全连接网络



天网系统

“准确锁定、捕捉到他们的，是‘天网工程’人脸识别系统，” 南湖区公安分局技术与数据服务中心综合管理室主任沈月光介绍说。

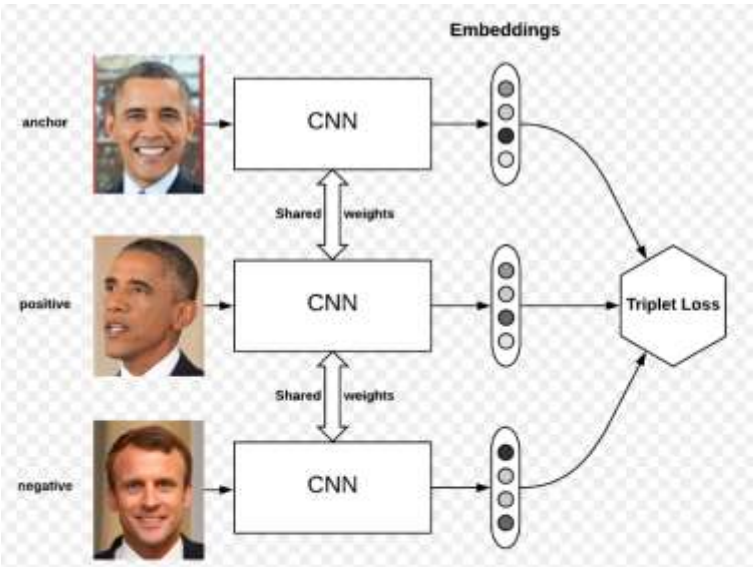
“演唱会在正对着检票口的地方增设了几个摄像头，能够对进出检票口所有人员姓名、身份证号、穿着、相貌等进行精确识别把控，它在非常短的时间内便可将数据库筛选一遍。”

……

人脸识别技术 - Facenet

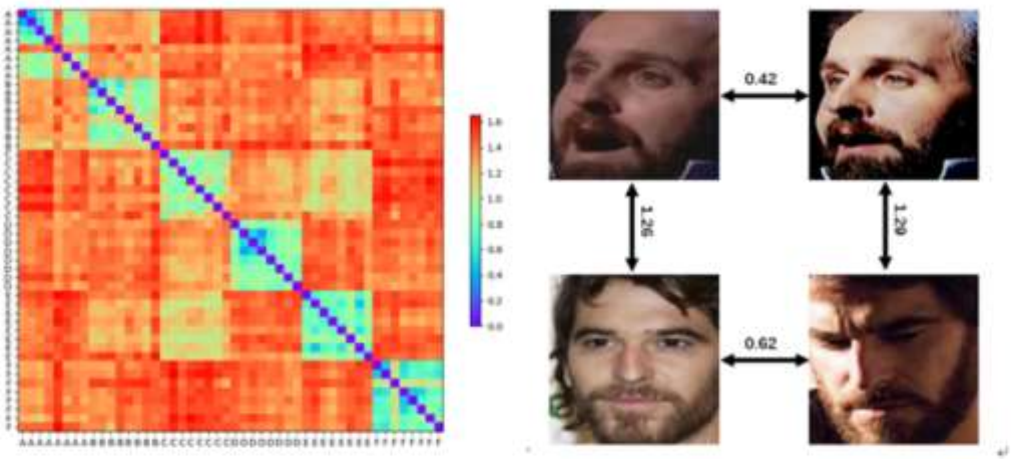
- Face embedding + optimize Triplet Loss

利用CNN将照片投影到表示向量空间



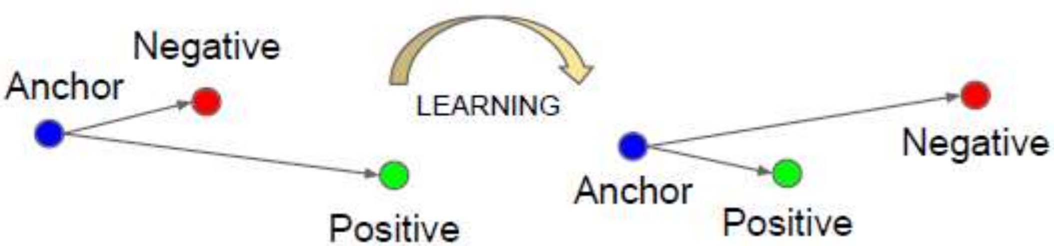
- 在表示空间用欧式距离度量不同人脸照片间的相似度

Person #1



人脸照片相似度矩阵

Person #2



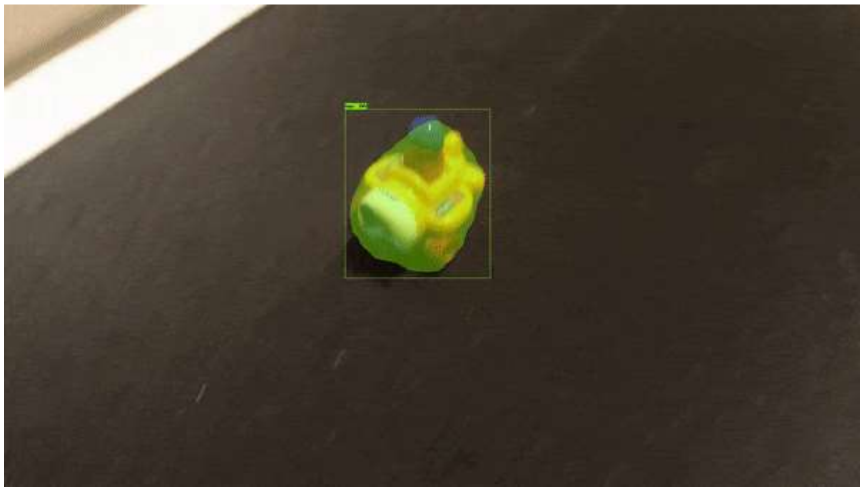
优化Triplet Loss

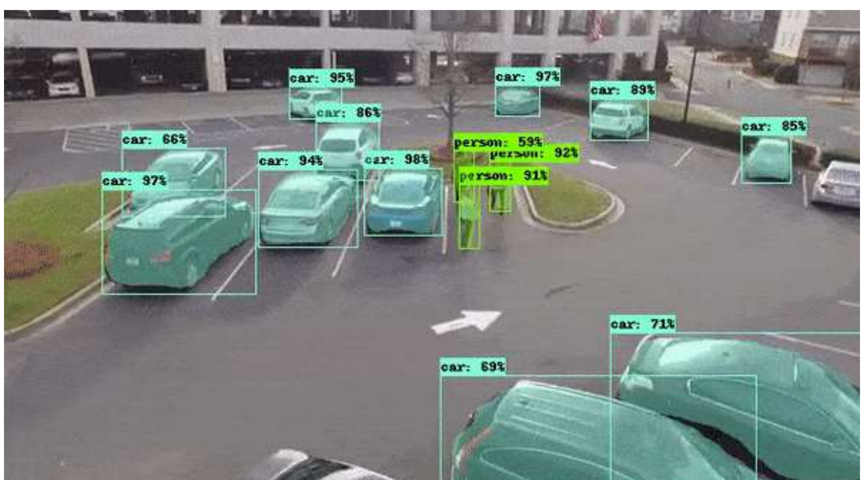
https://infoq.cn/article/how-to-achieve-face-recognition-using-mtcnn-and-facenet

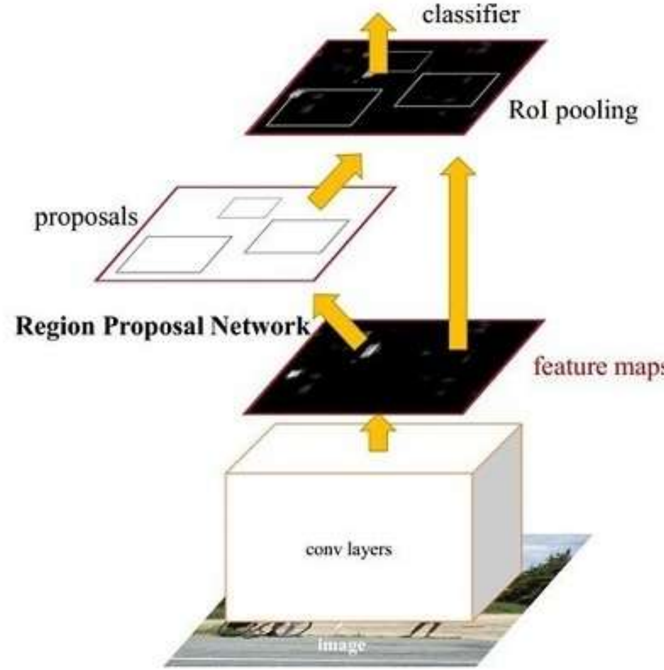
F Schroff, CVPR, 2015

物体识别技术 – Faster R-CNN

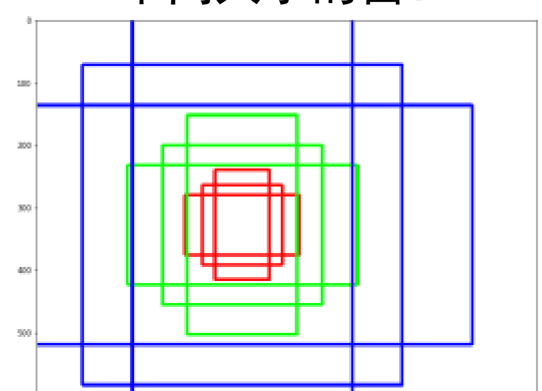
S Ren, CVPR, 2016







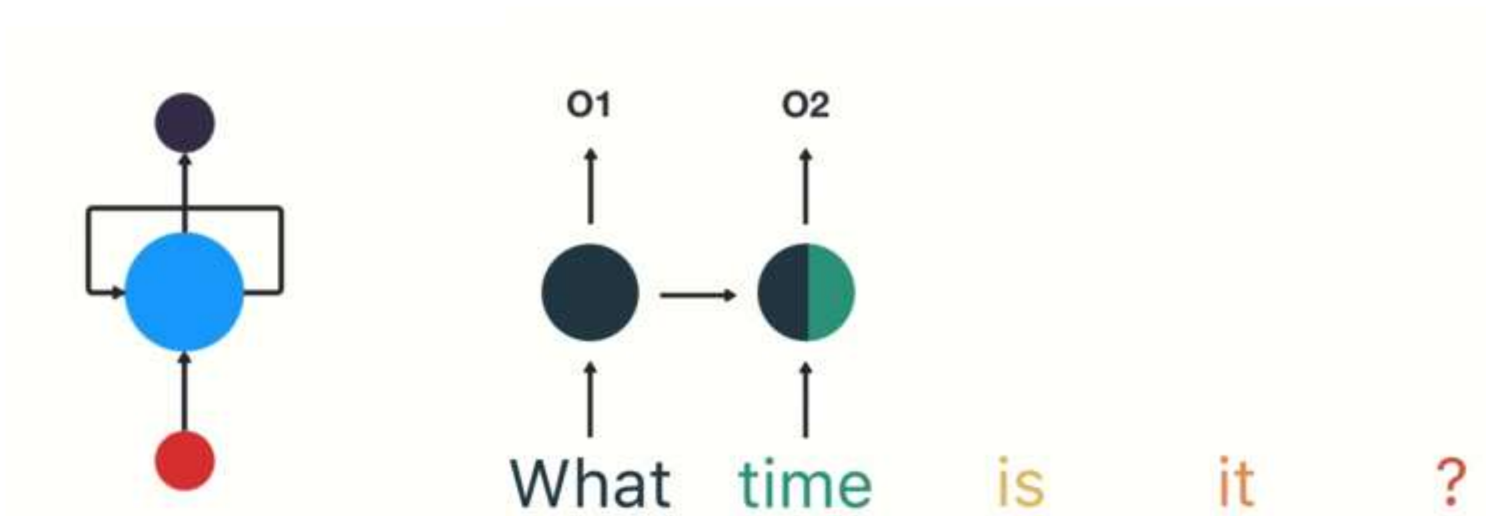
不同大小的窗：



循环神经网络

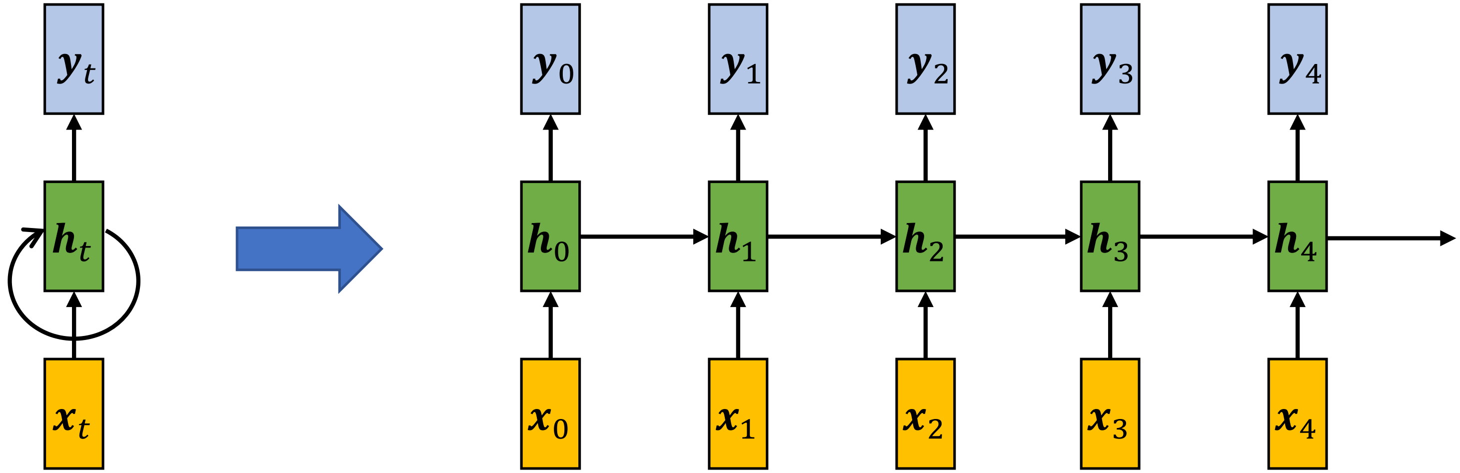
- 循环神经网络(Recurrent Neural Network，RNN)是一类用于处理序列数据的神经网络，它广泛的用于自然语言处理中的语音识别、机器翻译、故障诊断等领域。

- RNN可以将之前的输入“记忆”在神经网络中，从而输出也与之前输入有关，如下图的例子：



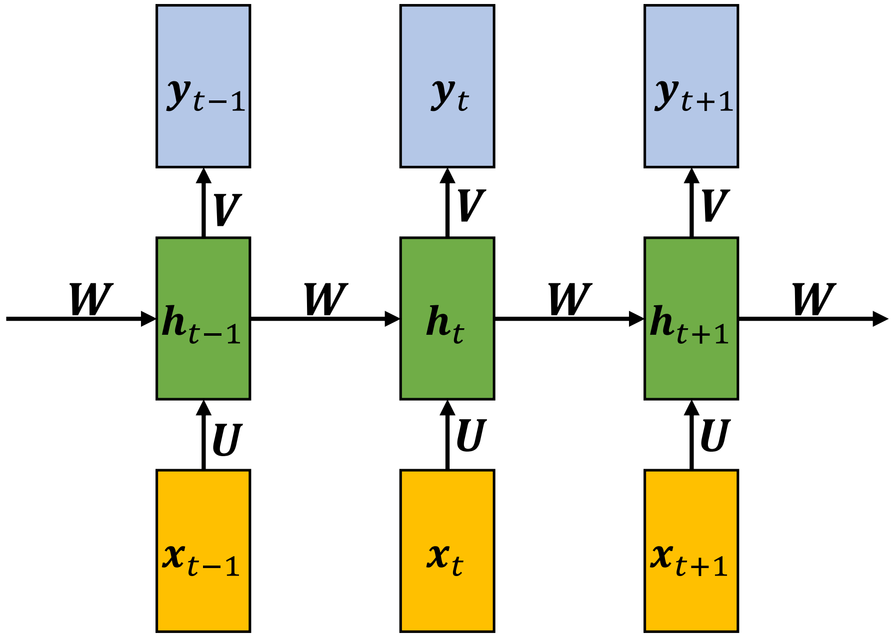
循环神经网络

- RNN是包含有循环的网络，可以使用展开计算图的方式表示。
  - $\boldsymbol{x}_t$：输入，$\boldsymbol{h}_t$：隐藏层，$\boldsymbol{y}_t$：输出



循环神经网络

- RNN每一层的计算需要考虑到上一层的影响，假设使用双曲正切激活函数(tanh)的前提下，用RNN来预测词或者字符，其推导过程如下所示。



$$\boldsymbol{h}_t = \tanh(\boldsymbol{b} + \boldsymbol{W}\boldsymbol{h}_{t-1} + \boldsymbol{U}\boldsymbol{x}_t)$$

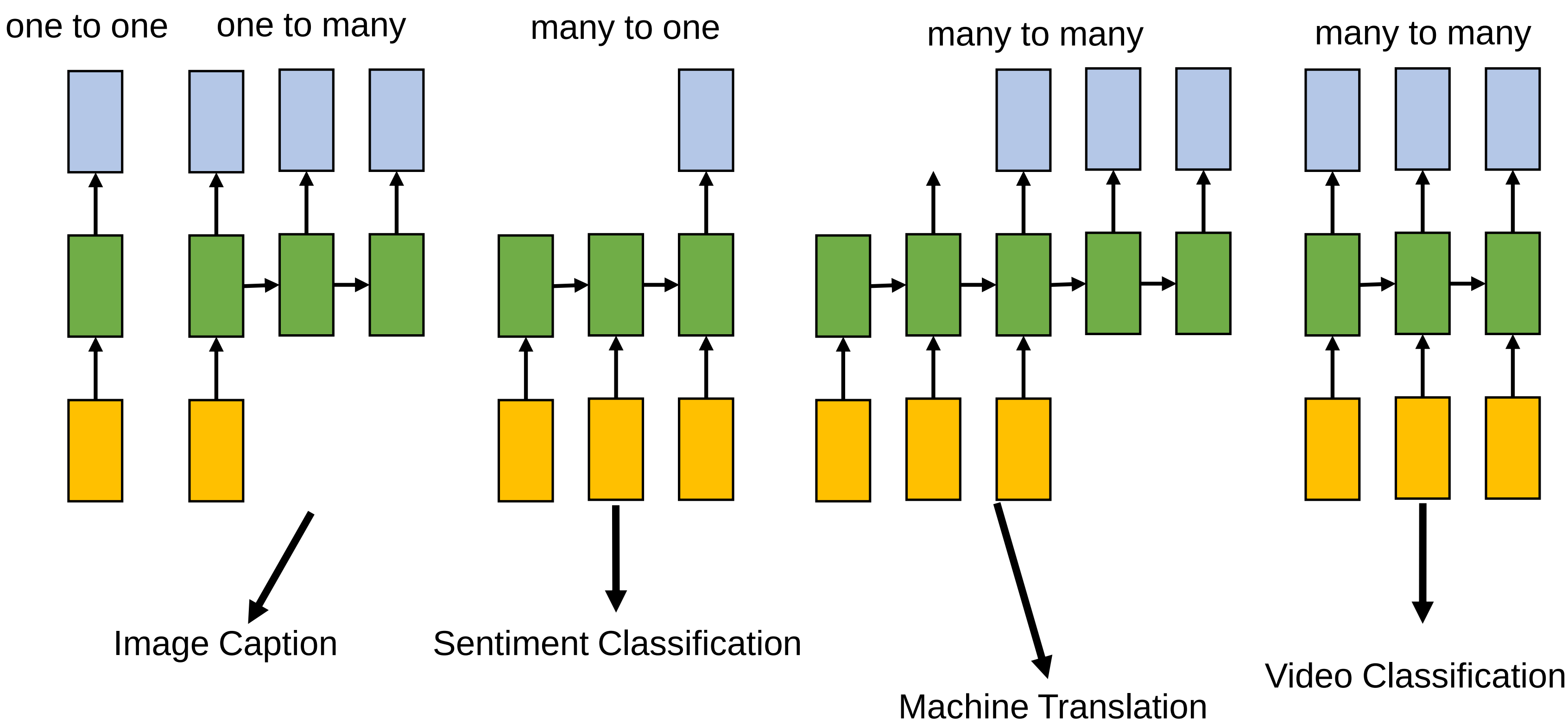
$$\boldsymbol{y}_t = \mathrm{softmax}(\boldsymbol{c} + \boldsymbol{V}\boldsymbol{h}_t)$$

$\boldsymbol{b}, \boldsymbol{c}$为偏置向量

其它RNN模型：LSTM，GRU等。

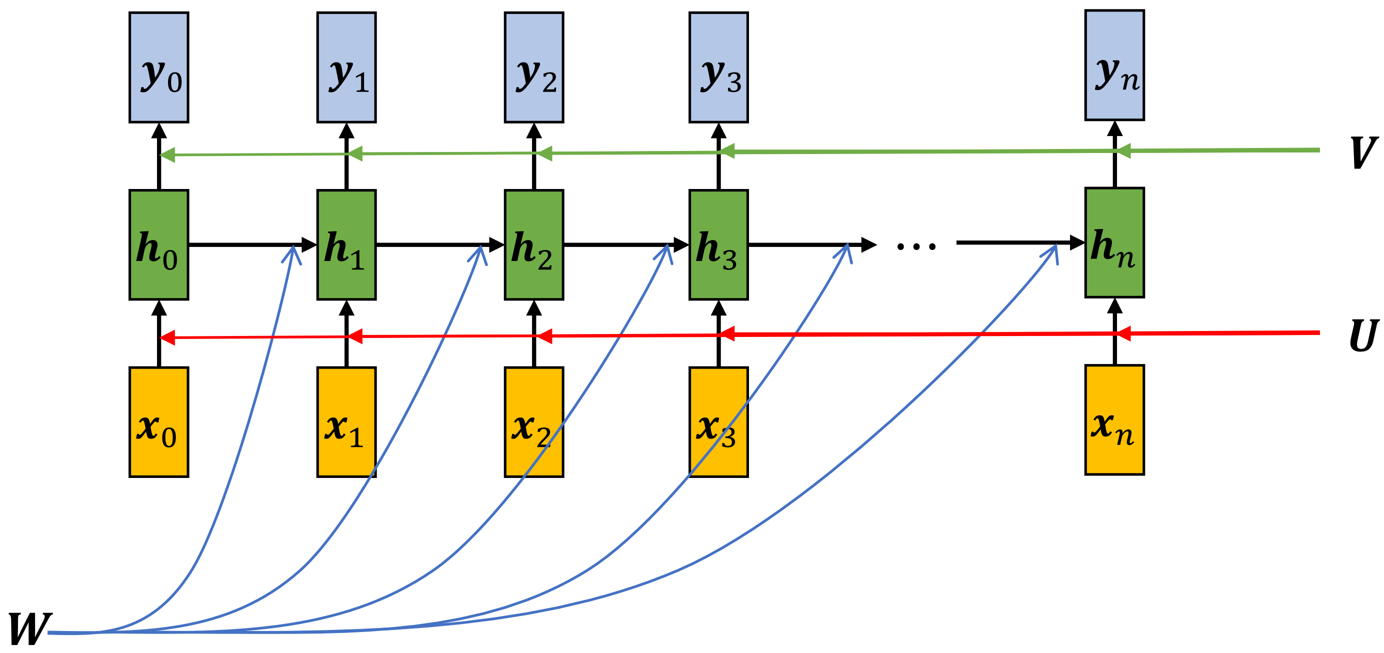
循环神经网络

- RNN按照输入输出可以分为以下四种。



循环神经网络：参数共享

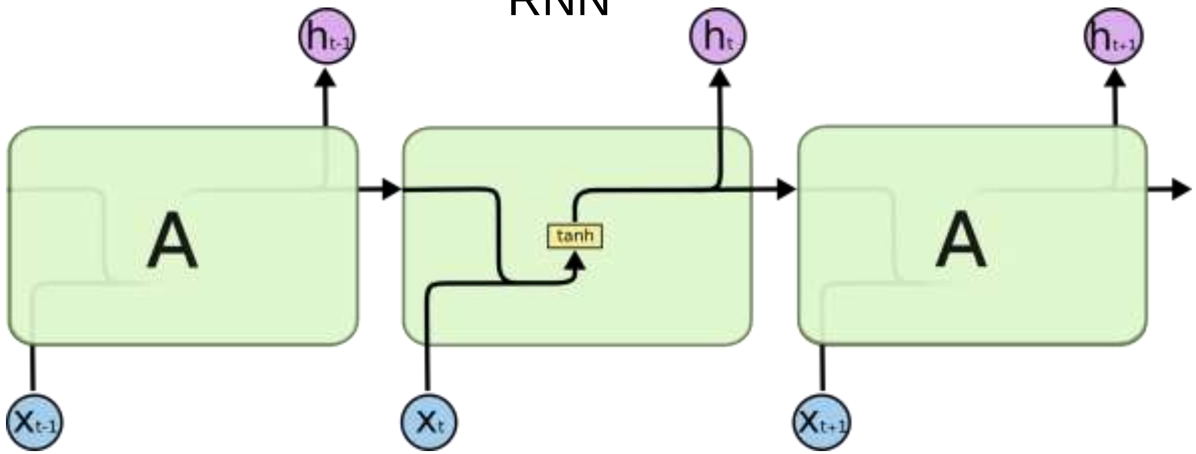
- RNN的一个特点是所有的隐层共享参数($\boldsymbol{W}$，$\boldsymbol{U}$，$\boldsymbol{V}$)，这样极大地缩小了参数空间。



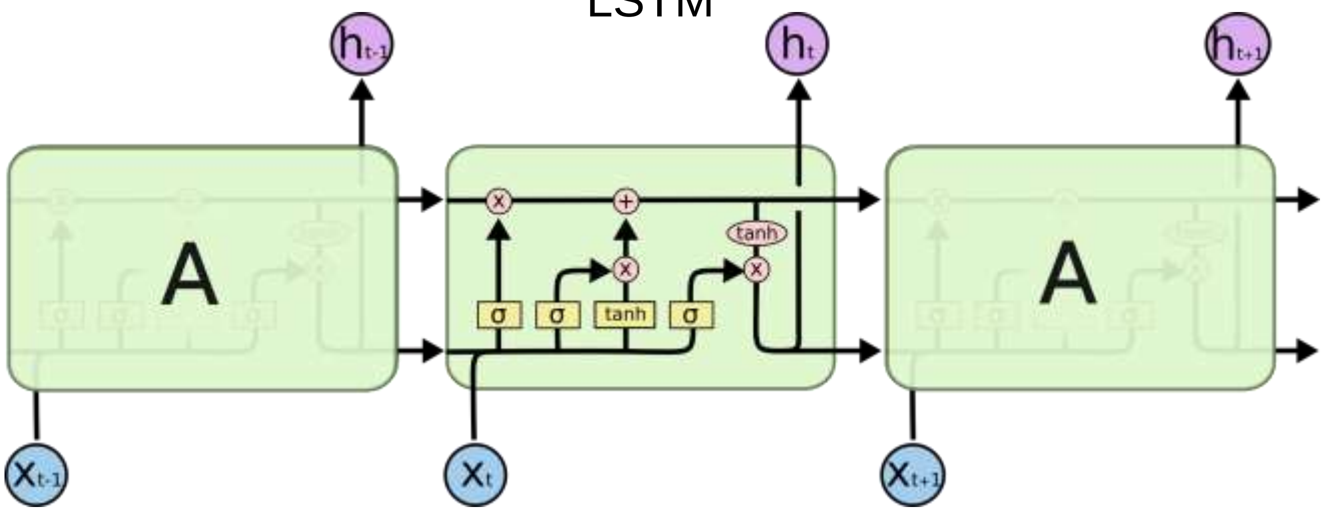
Long short-term memory (LSTM)

更好地处理长程记忆

RNN



LSTM

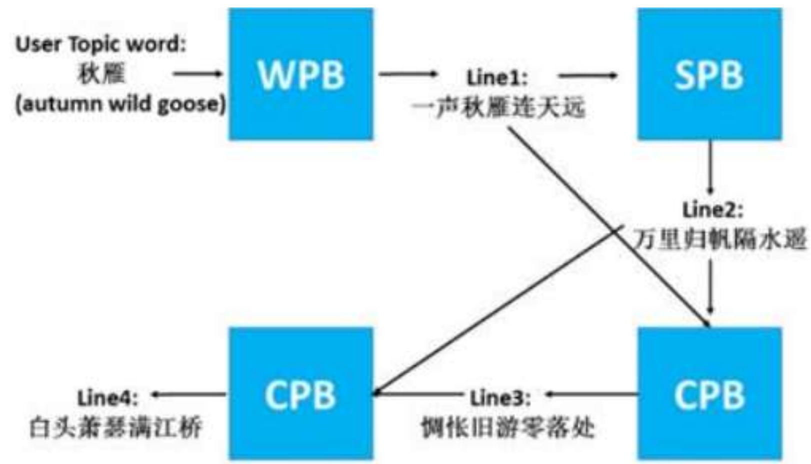


http://colah.github.io/posts/2015-08-Understanding-LSTMs/

RNN诗歌创作

一声秋雁连天远，(*P*ZPPZ)
The twitter of a wild goose
comes from the distant horizon.
万里归帆隔水遥。(*ZPPZZP)
The homebound ships are still ten
thousand miles away from the destination
惆怅旧游零落处，(*Z*PPZZ)
I am so sad to be the place where I
said goodbye to my travelling companions.
白头萧瑟满江桥。(*P*ZZPP)
There is nothing here, but a gloomy
spectacle and the old me in the bridge.

X Yi , CCL, 2016

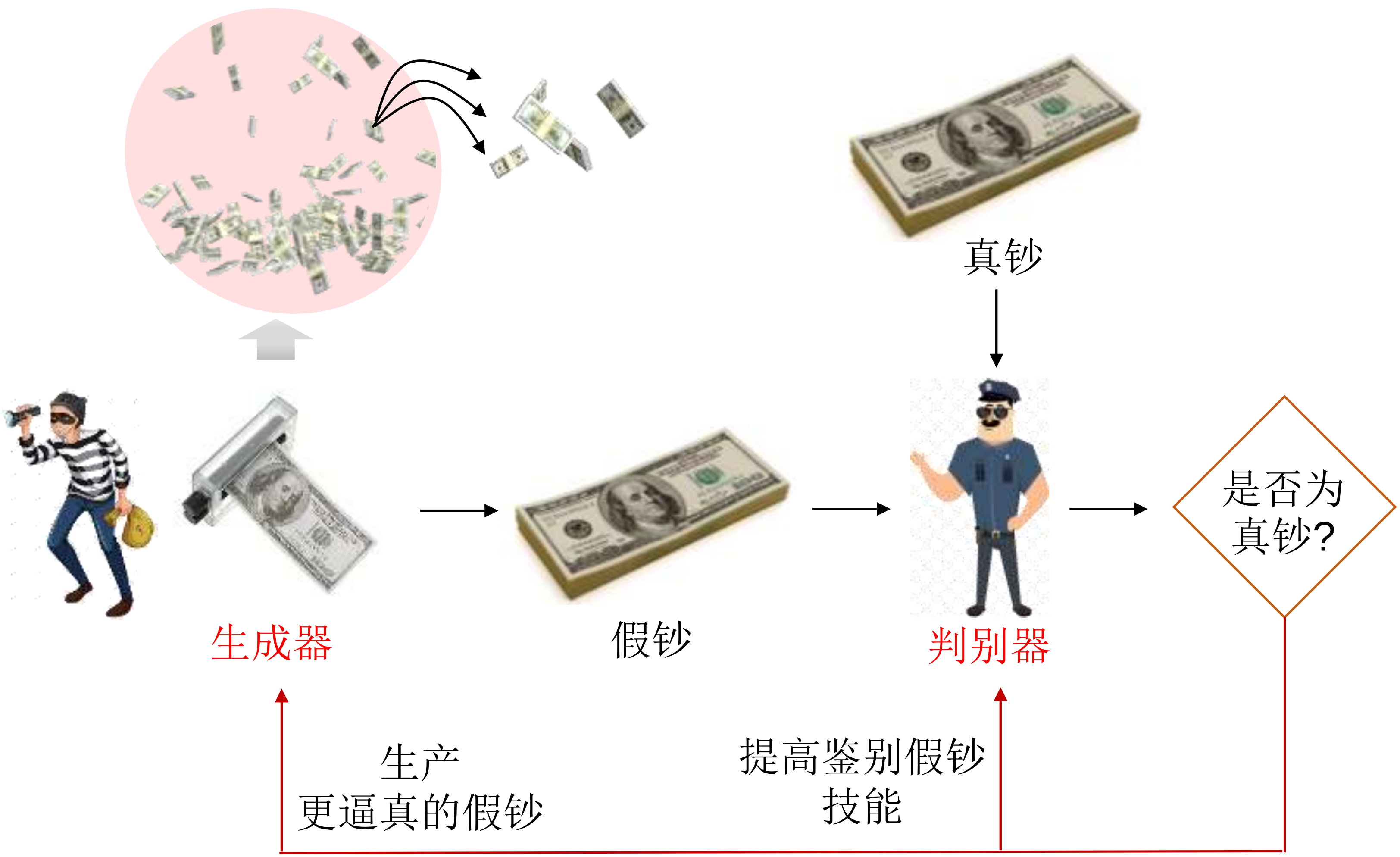


Generating Chinese Classical Poems with RNN Encoder-Decoder

利用深度学习做分布估计：

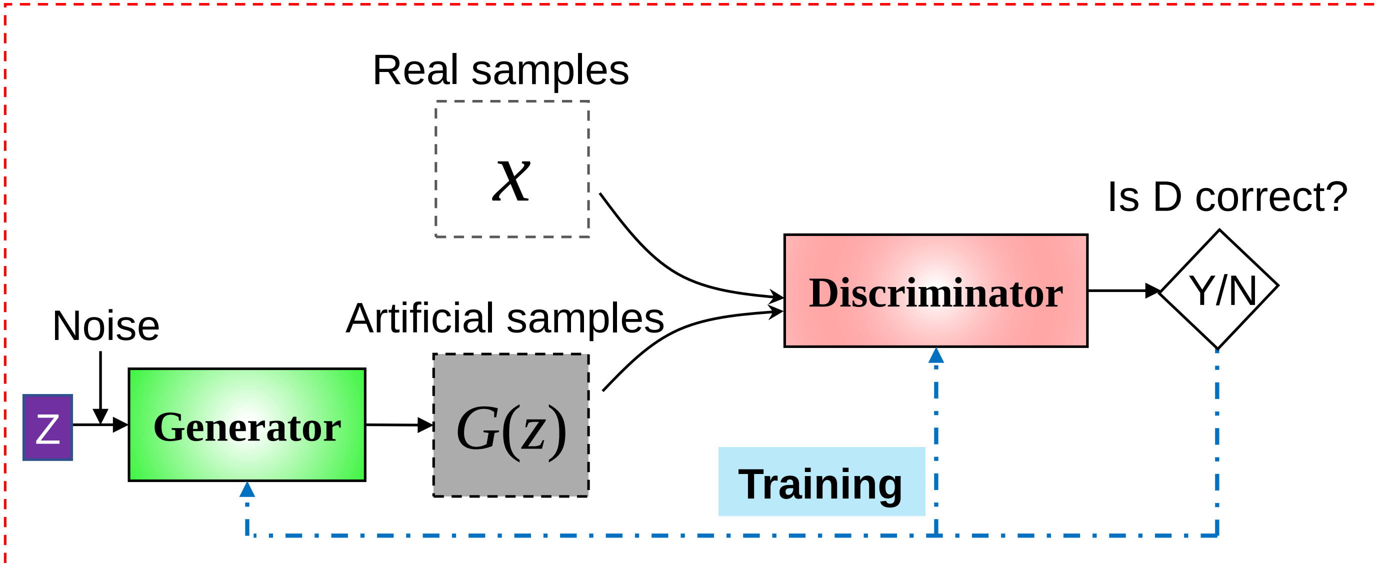
对抗生成网络（Generative Adversarial Nets, GAN）

- 生成器与判别器的自我博弈



利用深度学习做分布估计：

对抗生成网络（Generative Adversarial Nets, GAN）



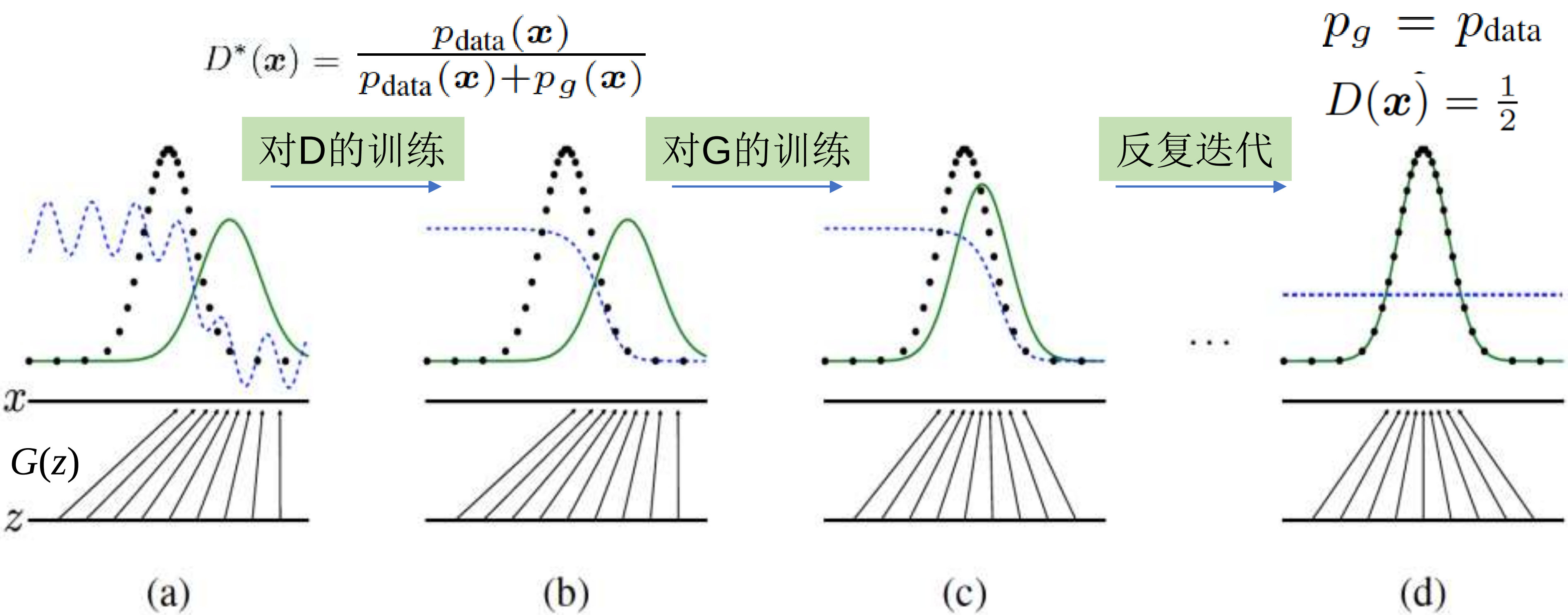
目标函数：

$$\min_G \max_D V(D,G) = \mathbb{E}_{\boldsymbol{x}\sim p_{\text{data}}(\boldsymbol{x})}[\log D(\boldsymbol{x})] + \mathbb{E}_{\boldsymbol{z}\sim p_{\boldsymbol{z}}(\boldsymbol{z})}[\log(1 - D(G(\boldsymbol{z})))]$$

- max$D$ 使得分类器尽量准确的区分真实样本和生成样本，即最大化log(D(x))和log(1-D(G(z))
- min$G$ 使得生成器尽量骗过分类器，即log(1-D(G(z))最小

生成对抗网络的训练过程示意

初始化的D和G



这里的Z表示一个均匀分布, $G(z)$是从Z到X的函数映射

GAN算法流程

Algorithm 1 Minibatch stochastic gradient descent training of generative adversarial nets. The number of steps to apply to the discriminator, k, is a hyperparameter. We used $k = 1$, the least expensive option, in our experiments.

for number of training iterations **do**

for k steps **do**

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Sample minibatch of m examples $\{\boldsymbol{x}^{(1)}, \dots, \boldsymbol{x}^{(m)}\}$ from data generating distribution $p_{\text{data}}(\boldsymbol{x})$.
- Update the discriminator by ascending its stochastic gradient:

$$\nabla_{\theta_d} \frac{1}{m} \sum_{i=1}^{m} \left[\log D\left(\boldsymbol{x}^{(i)}\right) + \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right) \right].$$

end for

- Sample minibatch of m noise samples $\{\boldsymbol{z}^{(1)}, \dots, \boldsymbol{z}^{(m)}\}$ from noise prior $p_g(\boldsymbol{z})$.
- Update the generator by descending its stochastic gradient:

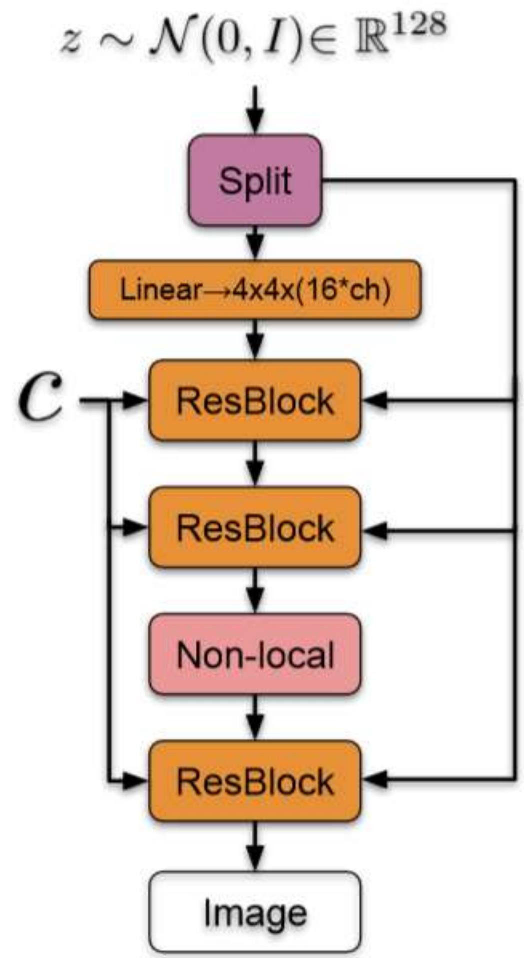
$$\nabla_{\theta_g} \frac{1}{m} \sum_{i=1}^{m} \log\left(1 - D\left(G\left(\boldsymbol{z}^{(i)}\right)\right)\right).$$

end for

The gradient-based updates can use any standard gradient-based learning rule. We used momentum in our experiments.

GAN应用 – 图像生成

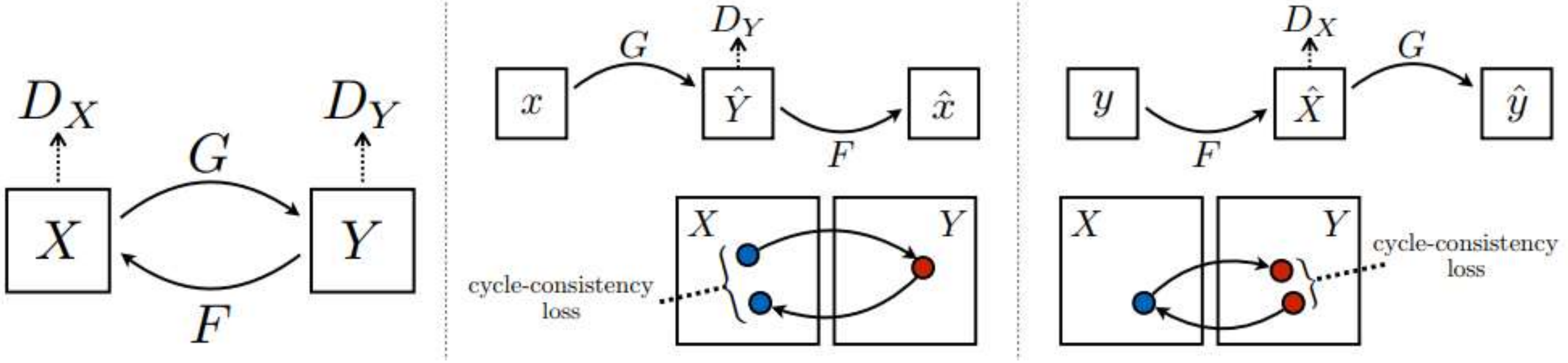
A Brock, ICLR, 2018 – Big-GAN



Generator Structure

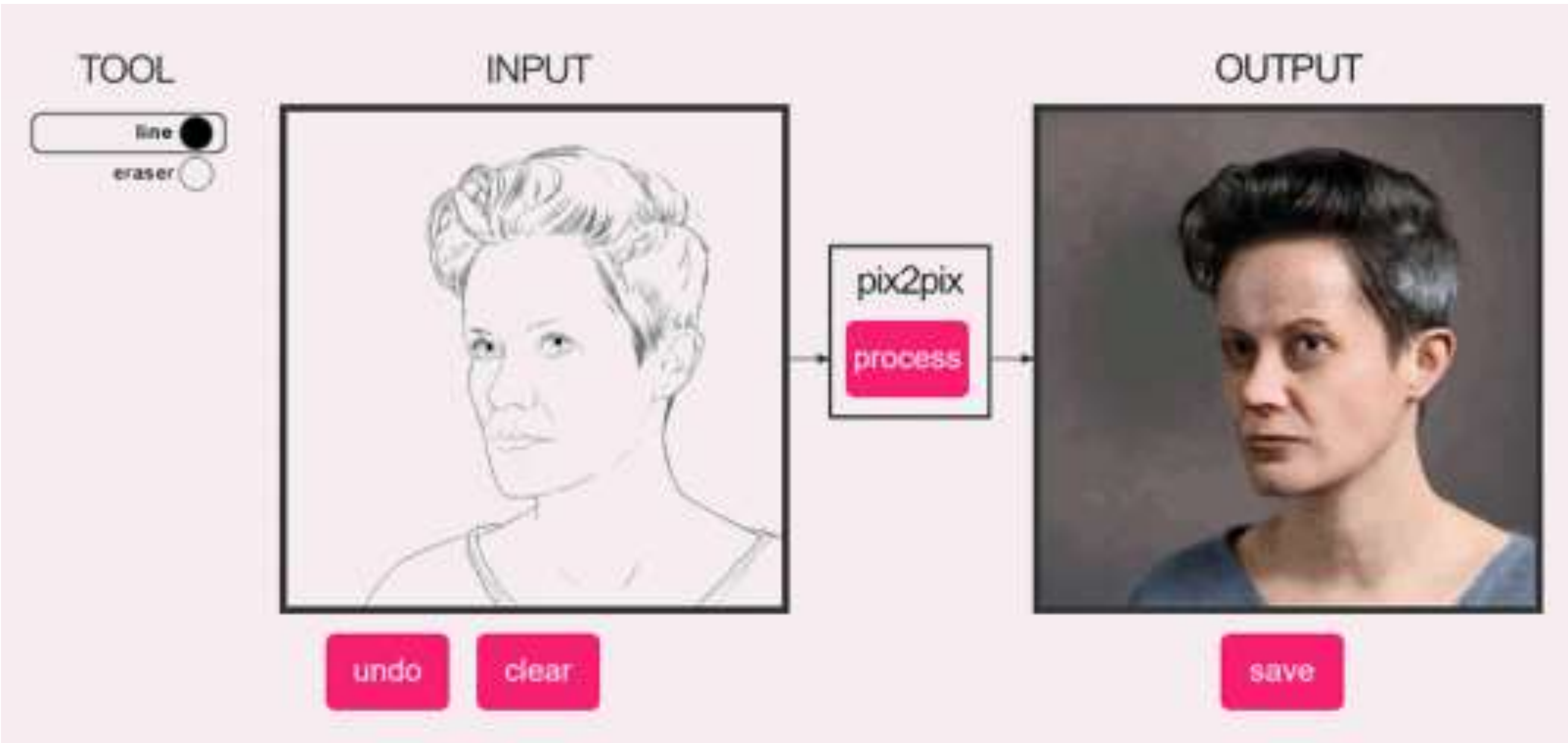
GAN应用 – 风格迁移

JY Zhu, CVPR, 2017 - cycleGAN



一个有趣的例子

Pix2pix – 图像翻译



<u>甚至可以自己尝试：</u>
<u>https://affinelayer.com/pixsrv/</u>

课外阅读：

- LeCun, Y., Bengio, Y. and Hinton, G. E.
  Deep Learning
  Nature, (2015), Vol. 521, pp 436-444

- Ian Goodfellow, Yoshua Bengio, Aaron Courville
  Deep Learning
  MIT press, 2016, https://www.deeplearningbook.org