

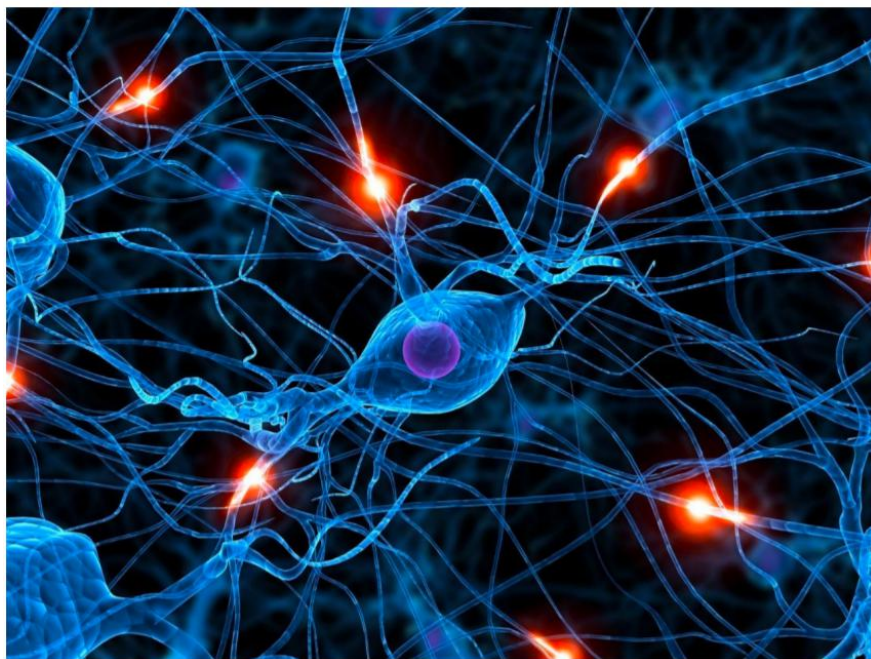
第二节：

基于TensorFlow实现CNN项目

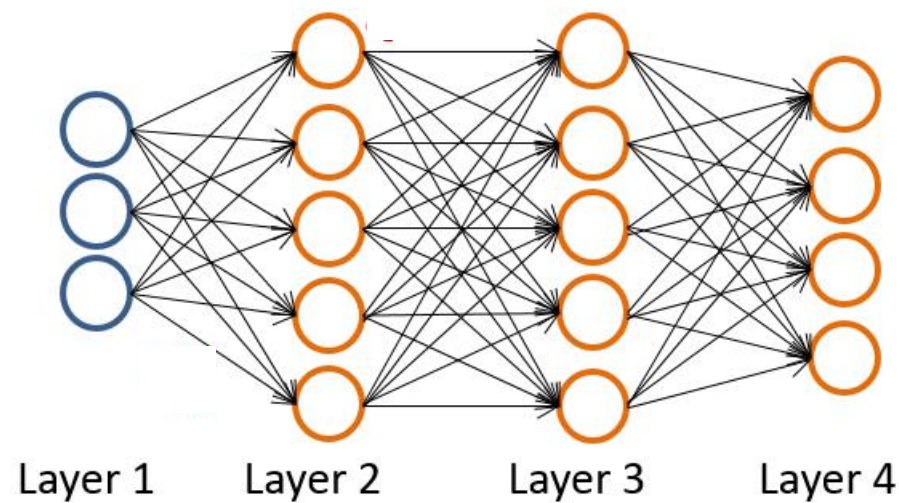
主讲：门徒

- 了解神经网络的基础元素如神经元等
- 深入了解卷积神经网络的架构
 - 神经网络的定义
 - 卷积层、激活函数、池化层和全连接层的功能及原理
- 基于TensorFlow的卷积神经网络实战

了解卷积神经网络



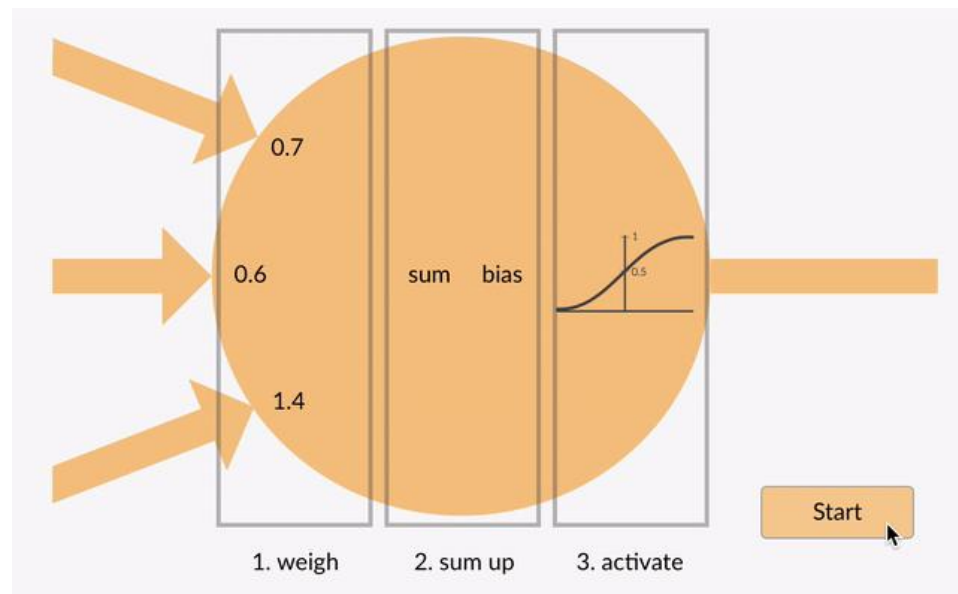
人类的神经结构



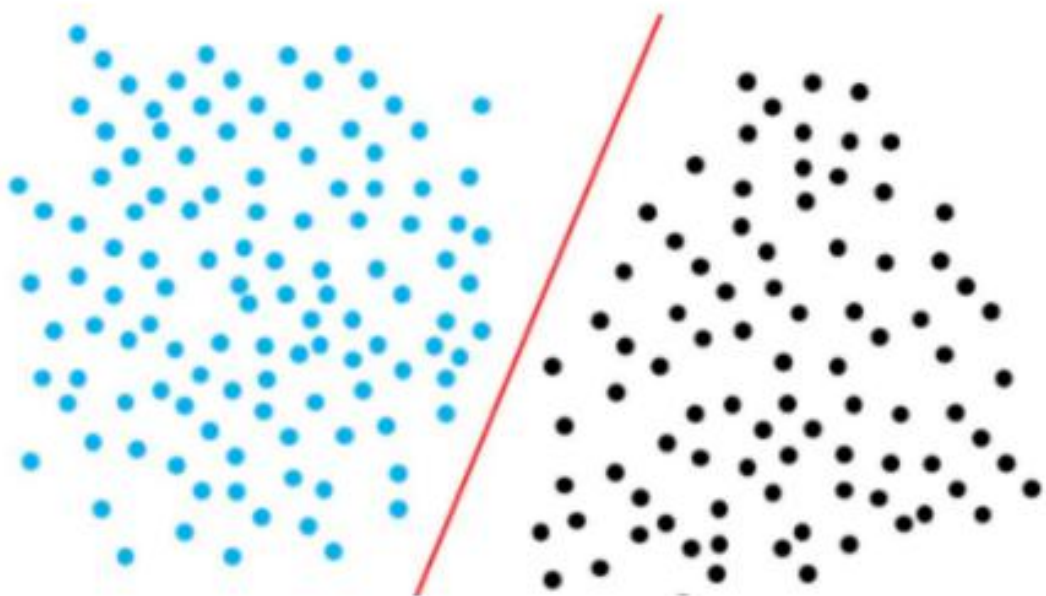
人造的神经网络

感知机 (Perceptron) 在1957年由Rosenblatt提出，是最早的人工智能模型之一。
感知机至今仍然是深度学习的重要组成部分。

感知机是二分类的线性模型，其输入是实例的特征向量，输出的是事例的类别，分别是1和0，属于判别模型。



$$Y = W^T X + B$$

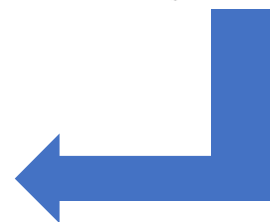
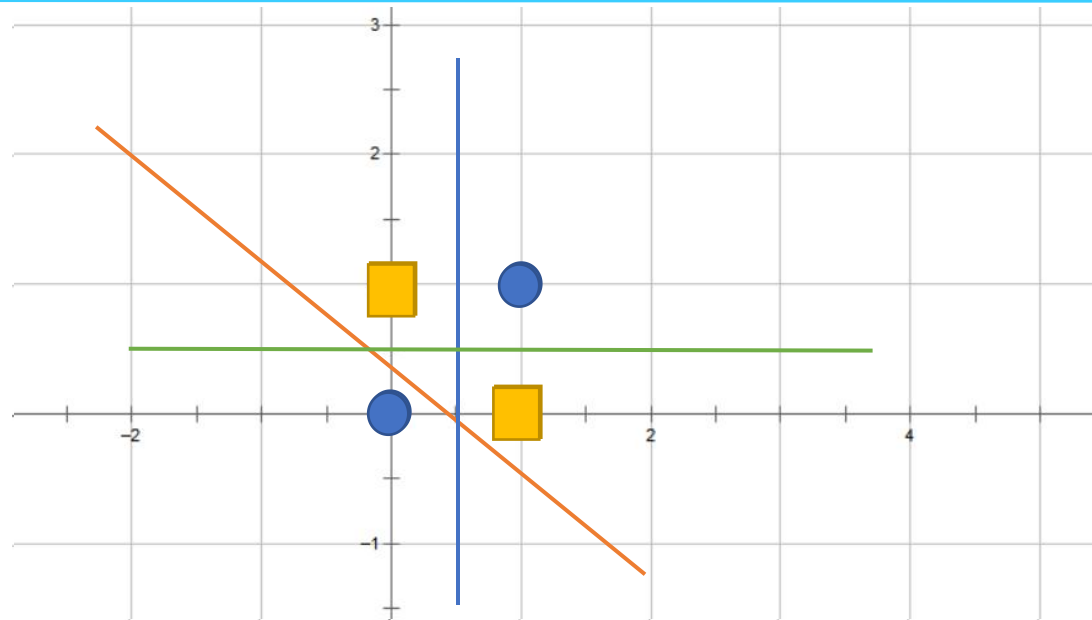
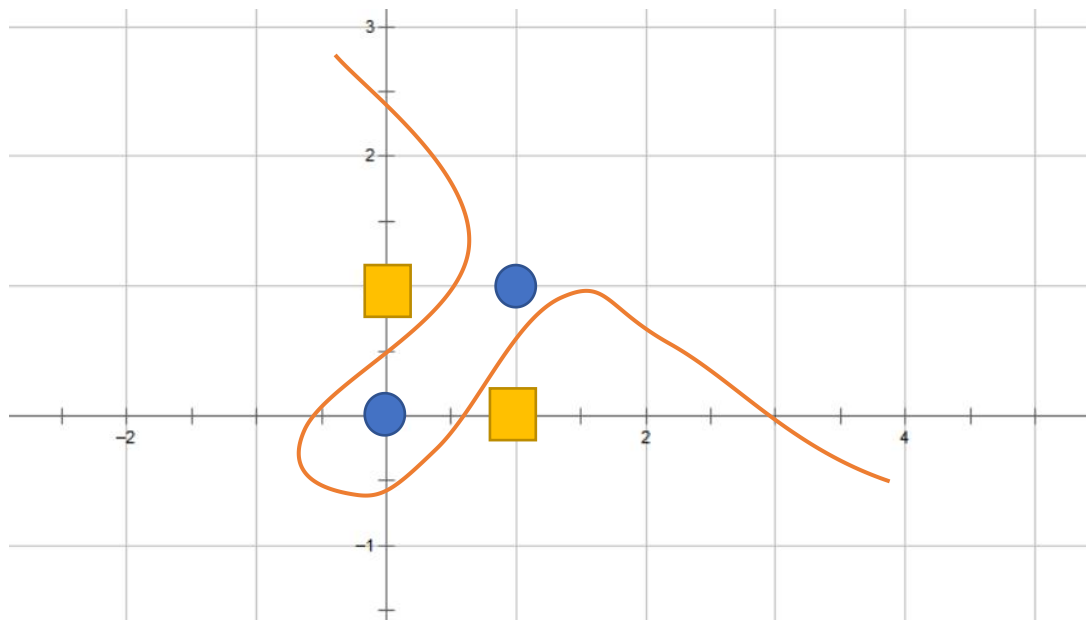


$$f(x) = \text{sign}(w \cdot x + b)$$

$$\text{sign}(x) = \begin{cases} -1 & x < 0 \\ 1 & x \geq 0 \end{cases}$$

对于一个平面来说，就是用一条直线将样本分成两类

对于异或问题来说，单个感知机永远无法找到一条合适的直线将所有样本点分隔开，这个问题曾导致人工智能陷入了长达10年的寒冬



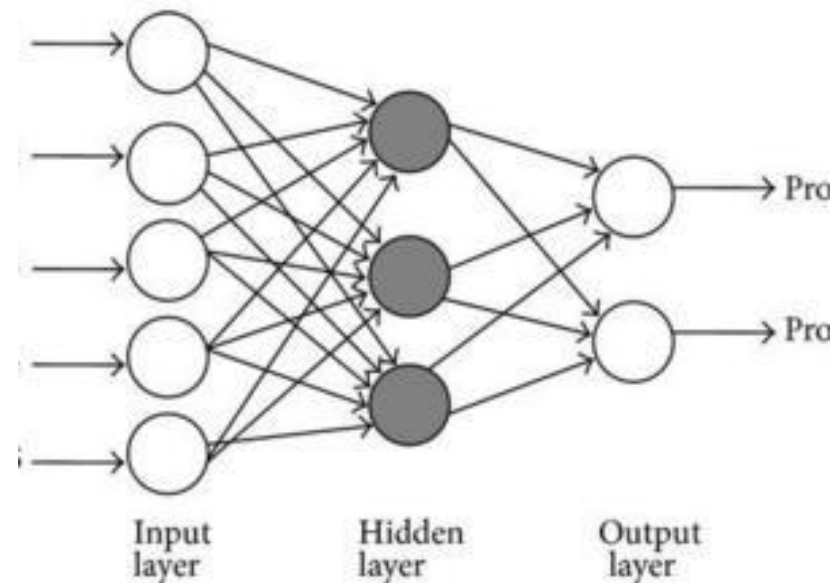
解决异或问题的唯一方法就是用一条曲线来分割平面

多层感知机（Mutli-Layer Perceptron, MLP）的出现使得神经网络具备了解决异或问题的能力。

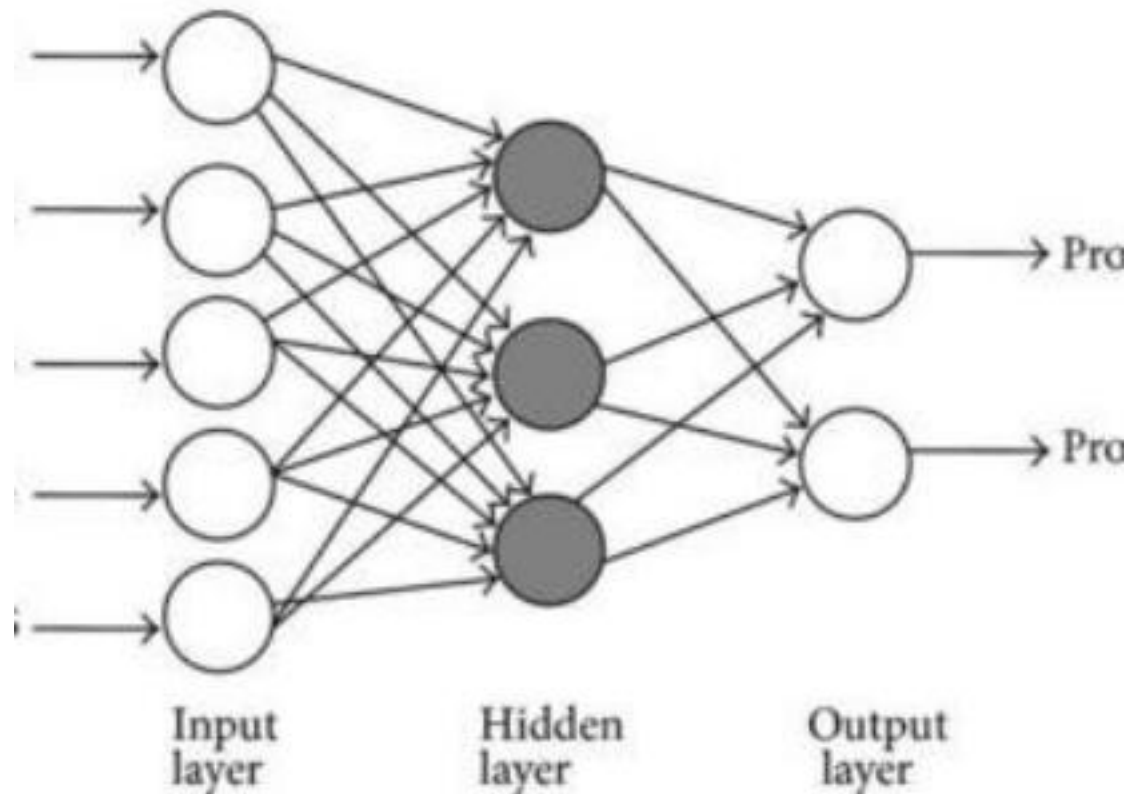
隐层神经元越多，MLP的拟合能力就越强。理论上MLP可以拟合任意连续函数

MLP各个层之间是全连接（Fully Connected）的

MLP也被称之为前馈式神经网络（Feed-Forward Neural network）



- 设定/调整权重非常困难
- 参数极多，容易过拟合



前馈型BP（反向传播）神经网络

为了解决MLP存在的问题，Rumelhart和Hinton等人于1986年提出了反向传播（Back Propagation）算法，解决了神经网络的权重更新问题。



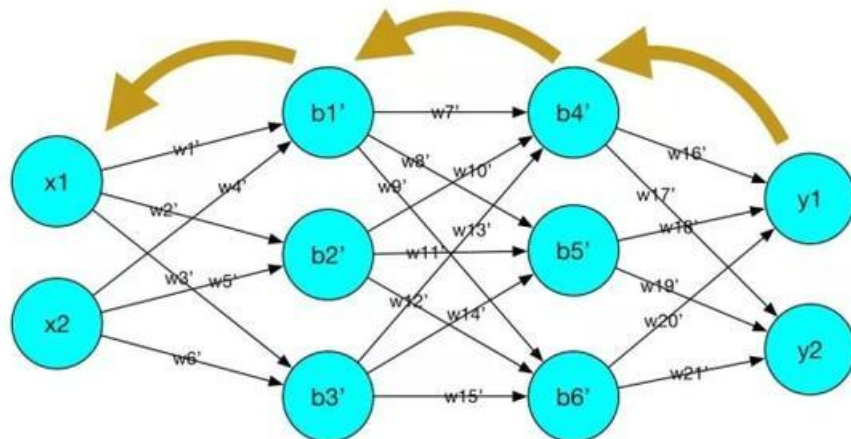
David Rumelhart
1942——2011



Geoffrey Hinton
1947——

什么是反向传播（BP）：

反向传播是一种适合于多层神经元网络的机器学习方法。在正向传播（也就是做出预测）完成后，如果预测结果和真实值存在误差，则通过反向传播的方式对网络中的权重进行调整以修正误差。



反向传播的基础：链式求导法则

考虑函数 $z = f(x, y)$ ，其中 $x = g(t)$ ， $y = h(t)$ ， $g(t)$ 和 $h(t)$ 是可微函数，那么：

$$\frac{dz}{dt} = \frac{\partial z}{\partial x} \frac{dx}{dt} + \frac{\partial z}{\partial y} \frac{dy}{dt}.$$

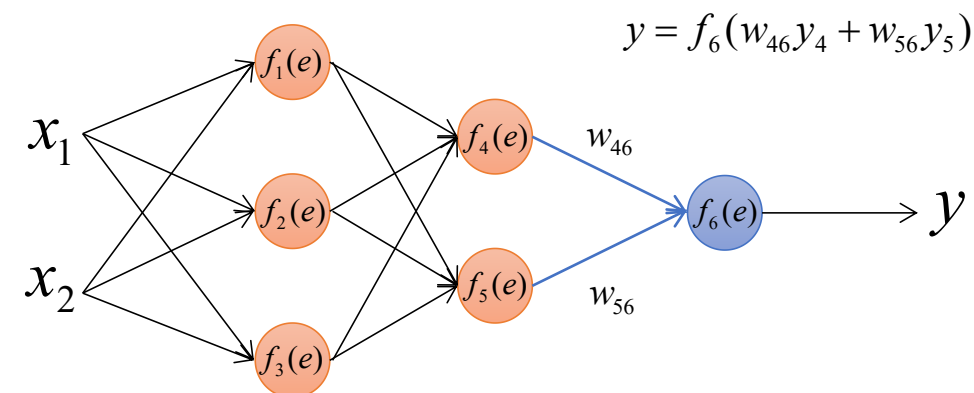
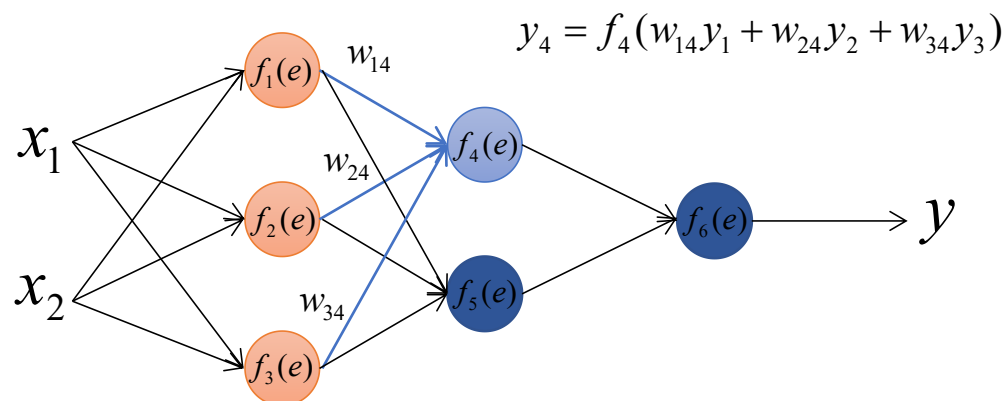
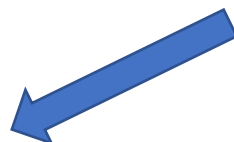
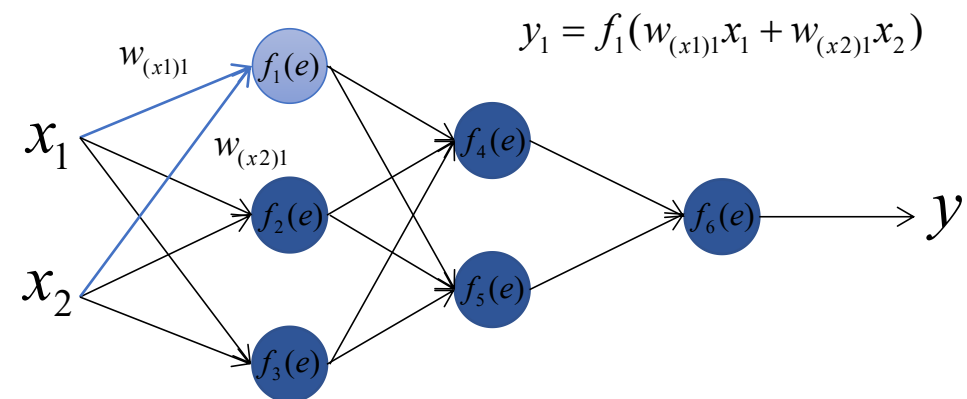
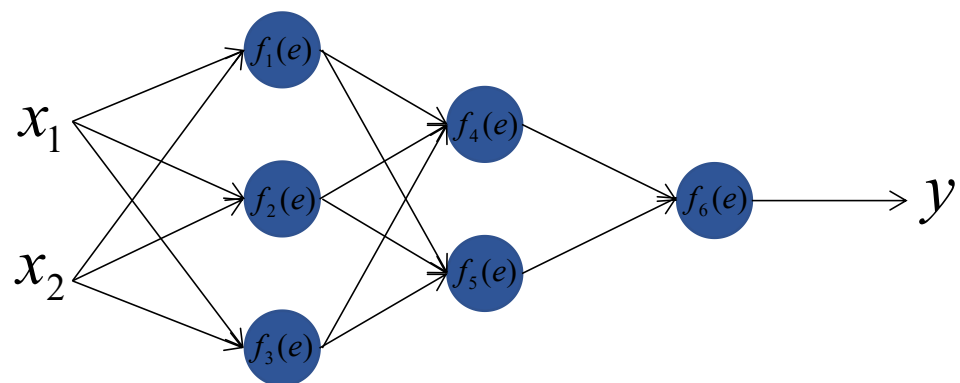
假设 $z = f(u, v)$ 的每一个自变量都是二元函数，也就是说， $u = h(x, y)$ ， $v = g(x, y)$ ，且这些函数都是可微的。那么， z 的偏导数为：

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial u} \frac{\partial u}{\partial x} + \frac{\partial z}{\partial v} \frac{\partial v}{\partial x}$$

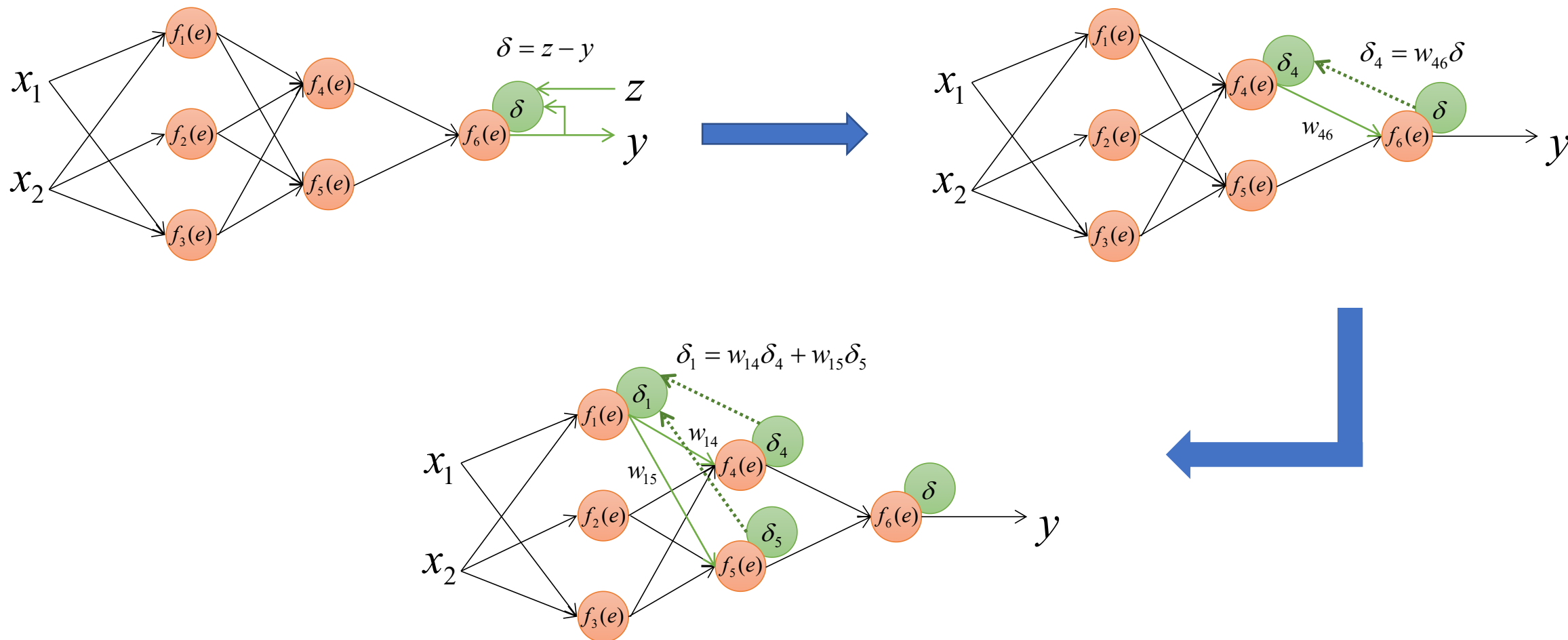
$$\frac{\partial z}{\partial y} = \frac{\partial z}{\partial u} \frac{\partial u}{\partial y} + \frac{\partial z}{\partial v} \frac{\partial v}{\partial y}.$$

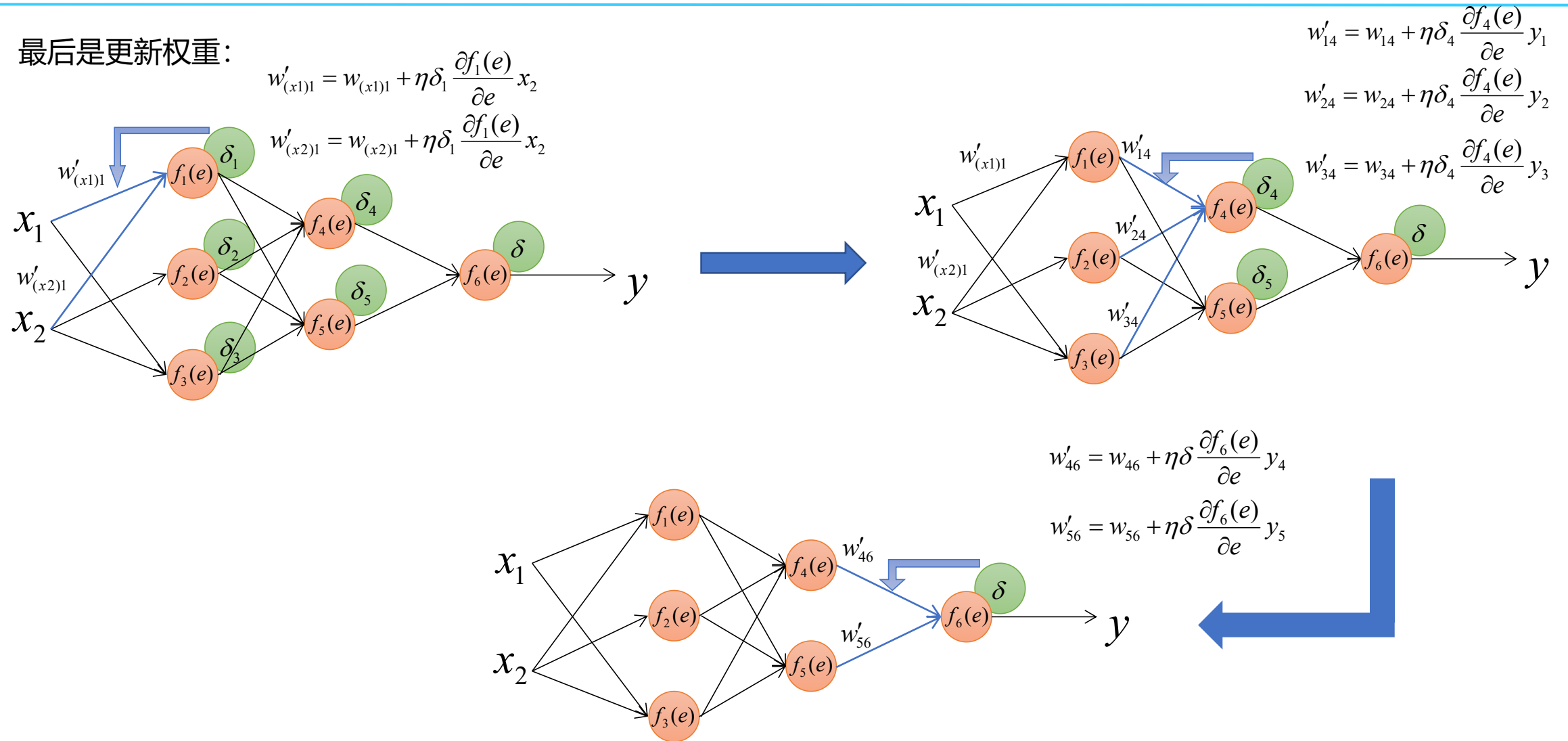
反向传播

首先是前向传播：



接下就是反向传播：





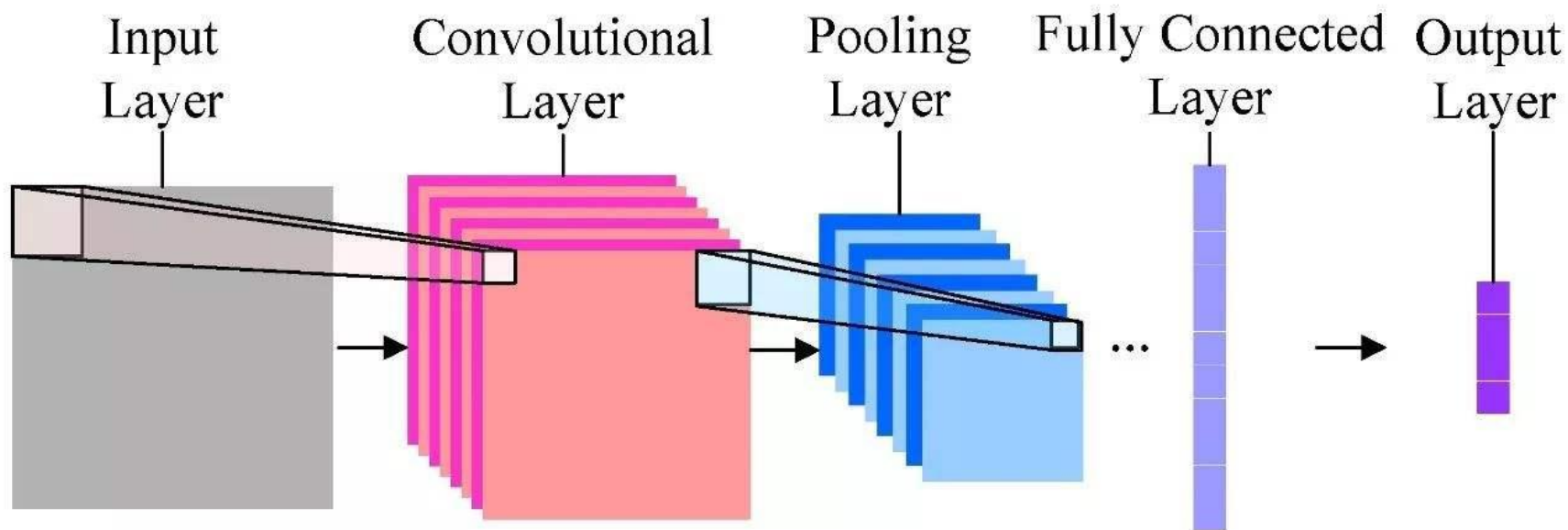
卷积神经网络的架构

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

- 卷积神经网络是深度神经网络的一种特殊形式，通过“卷积”而不是“全连接”提取特征

如何定义一个卷积网络

- 卷积层
- 池化层
- 输出层 (全连接层)

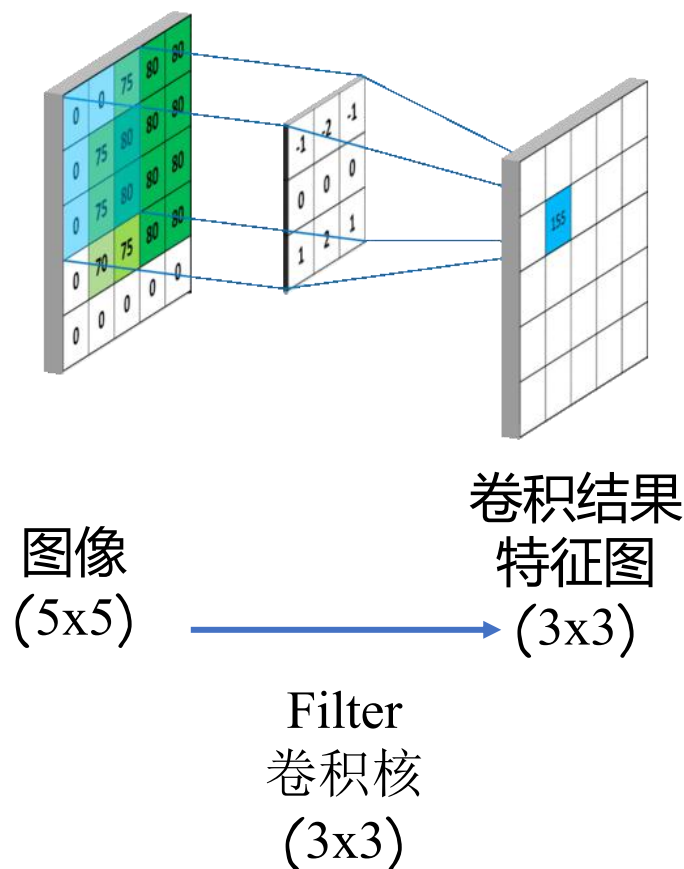


通俗来说，你在过去不同时刻惹女朋友生气的叠加，对女朋友现在坏心情的贡献就是卷积（convolution）。

$$h(t - t_1) = \int_{-\infty}^{\infty} \delta(\tau - t_1) h(t - \tau) d\tau .$$

这里假设惹女朋友生气的操作（作死行为）是输入函数 $\delta(x)$ ，女朋友的心情（也就是不开心）用响应函数表示

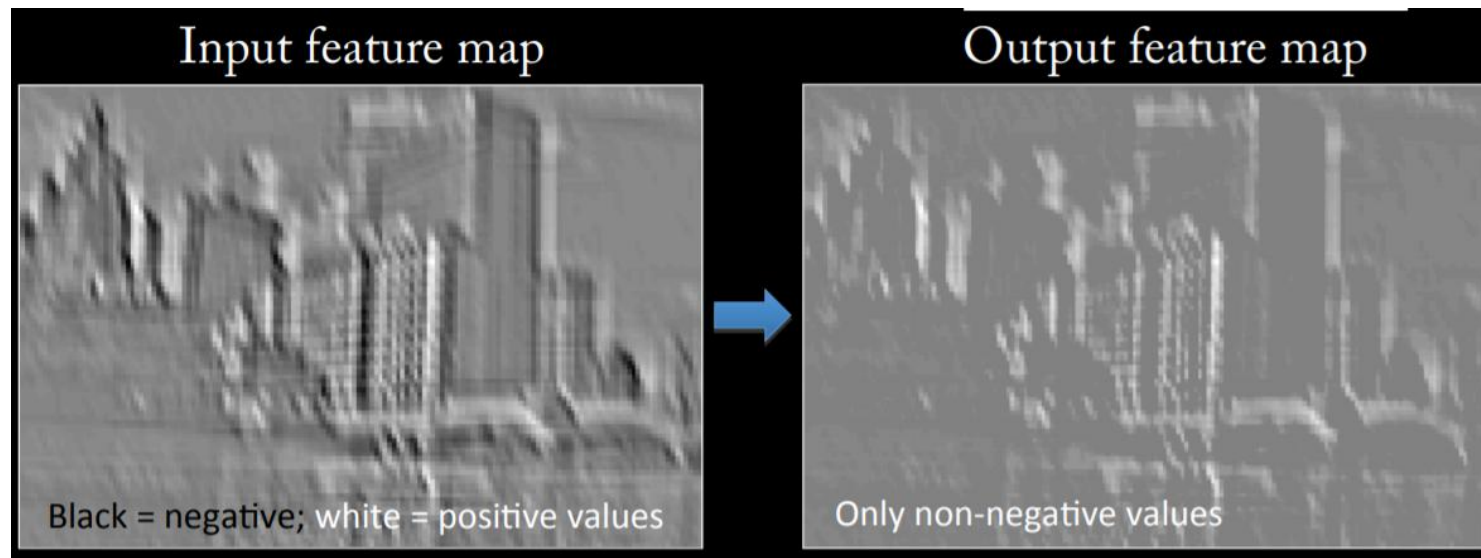
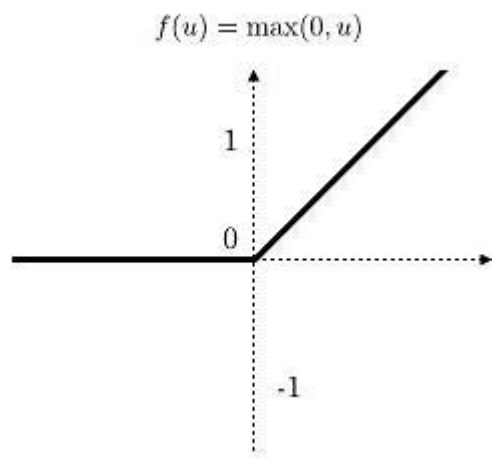
实际上对图像进行的卷积操作如下所示：





激活函数（Activation function）是一种非线性映射函数。神经网络的卷积操作实际上是一种线性变换，而线性函数经过线性变换后得到的仍然是线性函数。因此需要添加激活函数来使神经网络具备拟合非线性问题的能力。

目前激活函数中最常用的就是ReLU函数，可以将卷积层中所有的负像素值替换成零。

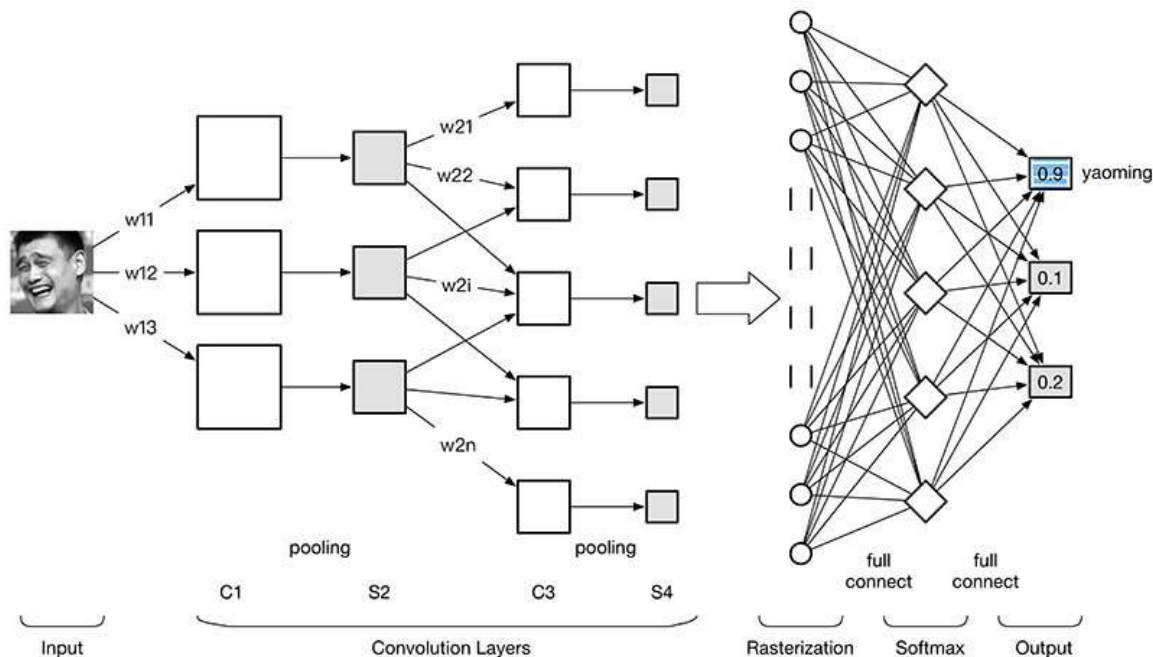


池化（pooling）的作用就是在尽可能保留特征的情况下减少需要处理的数据量



全连接层的功能及原理

全连接层（Fully Connected Layer）通常被用在神经网络的最后几层，用于完成最后的分类。在卷积层中获得的特征图（feature map）在全连接层中被摊平展开，成为一个一维的向量。这个向量经过激活函数（通常是softmax或sigmoid）处理后会转变成属于某个类的概率，取概率最高的一类作为最终结果。



卷积神经网络的实战应用

The screenshot shows the Kaggle competition page for 'Kannada MNIST'. The left sidebar contains navigation links: Home, Compete, Data, Notebooks, Discuss, Courses, and More. Below these are 'Recently Viewed' items and a 'View Active Events' button. The main content area displays the competition title, a search bar, and a grid of handwritten Kannada digits. Below the grid, there are tabs for Overview, Data, Notebooks, Discussion, Leaderboard, Rules, and Team. The 'Overview' tab is selected, showing a description of the competition and a table of Kannada digits.

Kannada MNIST
MNIST like dataset for Kannada handwritten digits

Kaggle · 1,214 teams · a year ago

Overview Data Notebooks Discussion Leaderboard Rules Team My Submissions **Late Submission**

Overview

Description

Bored of MNIST?

The goal of this competition is to provide a simple extension to the classic [MNIST competition](#) we're all familiar with. Instead of using Arabic numerals, it uses a recently-released dataset of Kannada digits.

Kannada is a language spoken predominantly by people of Karnataka in southwestern India. The language has roughly 45 million native speakers and is written using the Kannada script. [Wikipedia](#)

0	1	2	3	4	5	6	7	8	9
ಒಂದು	ಎರಡು	ಮೂರು	ನಾಲ್ಕು	ಐದು	ಆರು	ಏಳು	ಎಂಟು	ಒಂಬತ್ತು	ಹತ್ತು
omdu	eradu	muru	naluku	aidu	aru	elu	emtu	ombattu	hattu
1	2	3	4	5	6	7	8	9	10

This competition uses the same format as the [MNIST competition](#) in terms of how the data is structured,

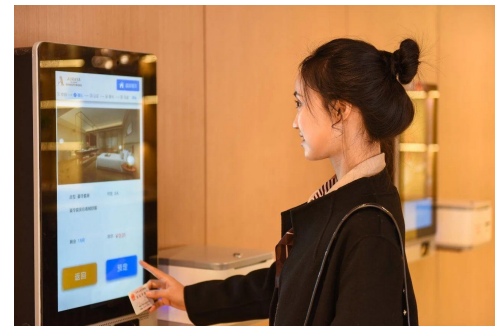
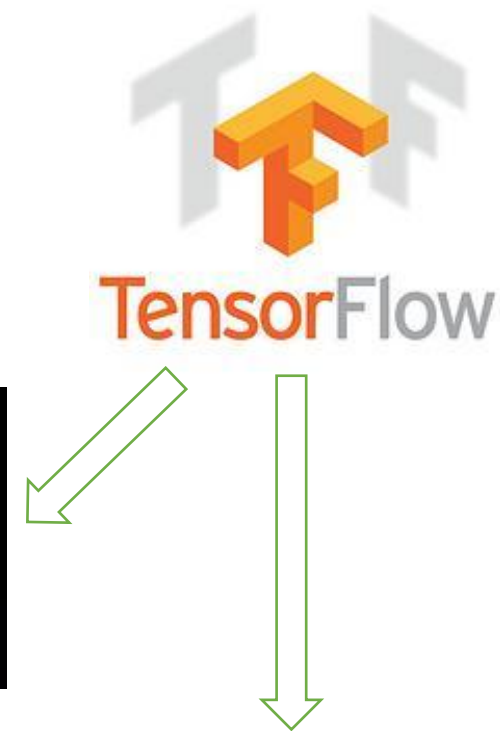
网址: <https://www.kaggle.com/c/Kannada-MNIST>

TensorFlow是Google的人工智能团队谷歌大脑（Google Brain）开发并维护的一个机器学习框架，尤其适用于各种深度学习项目中。

TensorFlow的应用场景包括：

- AlphaGo
- 人脸识别
- 语音识别
- etc....

总结来说：在深度学习领域有着广泛的应用



在开始前我们来做一个小问答：

开始一个项目的第一步是什么呢？

还记得我们上节课的数据吗？每张图像都是用一行向量来代替的。然而实际上一个行向量是不能被称之为“图像”的

In [28]: `train.head(4)`

Out[28]:

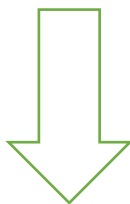
	label	pixel0	pixel1	pixel2	pixel3	pixel4	pixel5	pixel6	pixel7	pixel8	...	pixel774	pixel775	pixel776	pixel777	pixel778	pixel779	pixel780	pixel781	p
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
1	1	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
2	2	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	
3	3	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	0	0	

4 rows × 785 columns



我们现在要把行向量重新变成图像

1	1	0	0	1	0	1	1	1
---	---	---	---	---	---	---	---	---

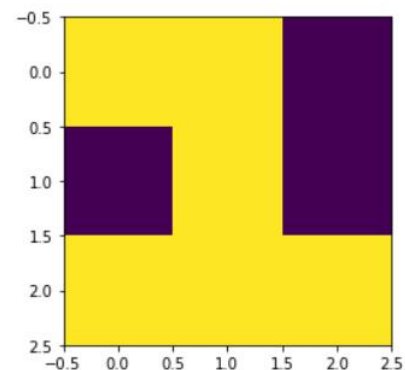


1	1	0
0	1	0
1	1	1

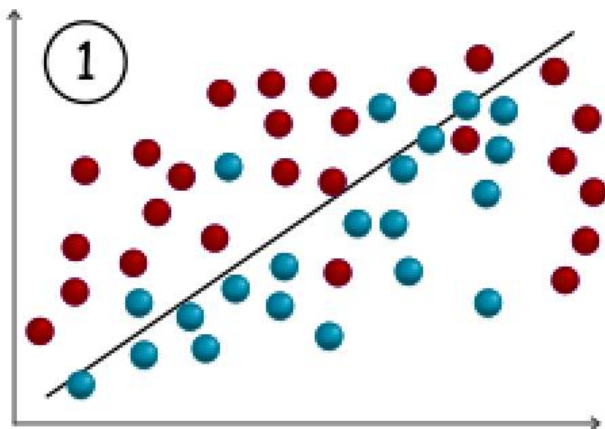


```
In [33]: a = np.matrix([1, 1, 0, 0, 1, 0, 1, 1, 1])  
a = a.reshape(3, 3)  
plt.imshow(a)
```

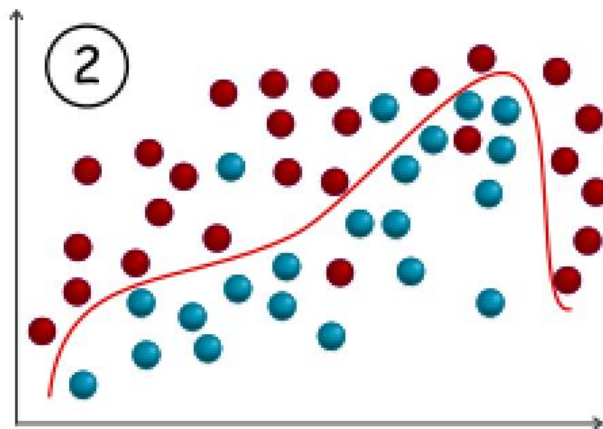
```
Out[33]: <matplotlib.image.AxesImage at 0x2a0b498a9e8>
```



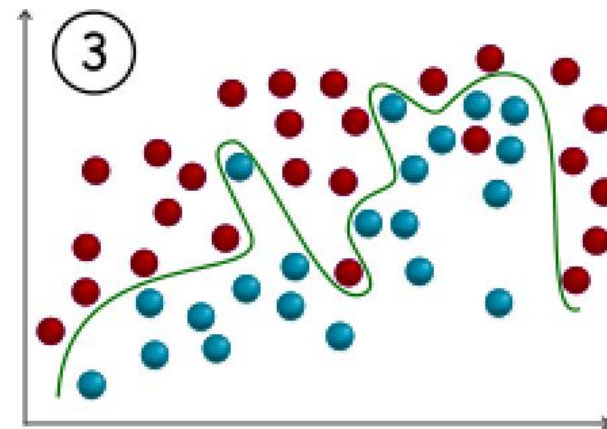
underfitting



good fit



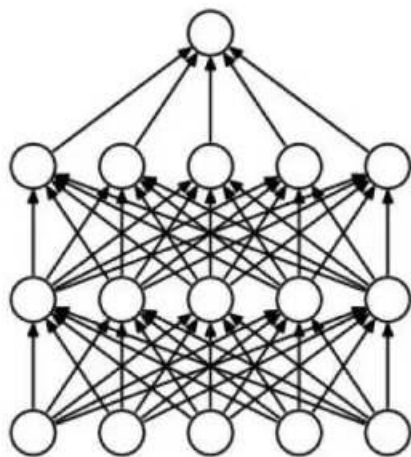
overfitting



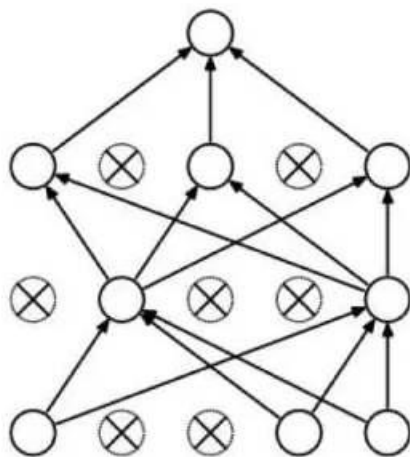
过拟合就是由于过于追求完美拟合现有数据集，以至于会将单个数据点的特征认为是一般特征。结果就是预测新数据的能力极差，表现特征就是在训练集上准确度很高，但是在验证集上准确度很低

目前已经有了很多方案可以解决过拟合问题：

- Early Stopping
- 数据扩增 (Data Augmentation)
- Dropout



(a) 标准神经网络 (左)



(b) 应用Dropout后的神经网络 (右)

现在让我们进入代码环节

- 了解了感知机/多层感知机的优点及缺陷
- 对反向传播（BP）有了基本的认识
- 深入了解卷积神经网络的架构
- 基于TensorFlow的卷积神经网络实战

在下节课我们将主要集中在下面几个内容：

- 针对一个新的项目（猫狗识别）讨论CNN模型的进一步改进方向。
- 根据机器学习技能图谱深入讲解后续发展方向
- 面试中深度问题的详解
- 学习资料推荐

Q & A

自由问答时间