

The background of the slide is a photograph of the Shandong University campus, featuring several large, modern buildings with a mix of traditional and contemporary architectural styles. The image is overlaid with a semi-transparent red filter. The main title is centered in the upper half of the image.

卷积神经网络

2023年9月

C 目录

CONTENTS



山东大学
SHANDONG UNIVERSITY

- 1 机器学习相关基础知识
- 2 深度神经网络



山东大学
SHANDONG UNIVERSITY

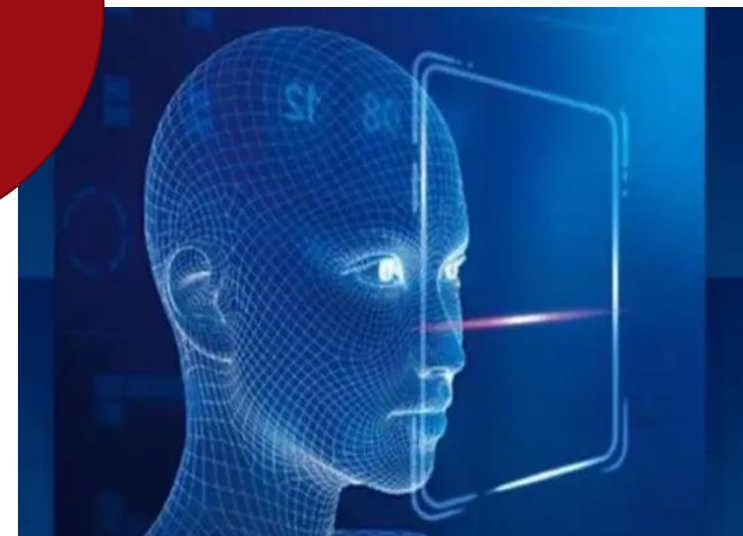
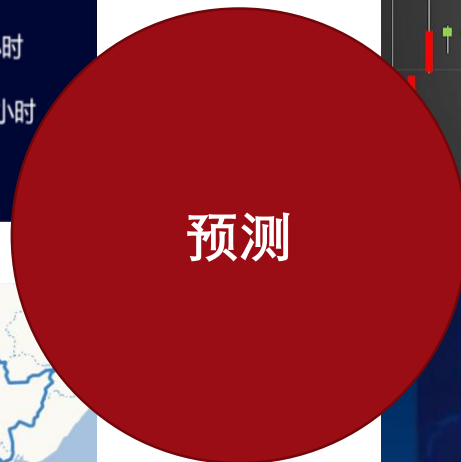
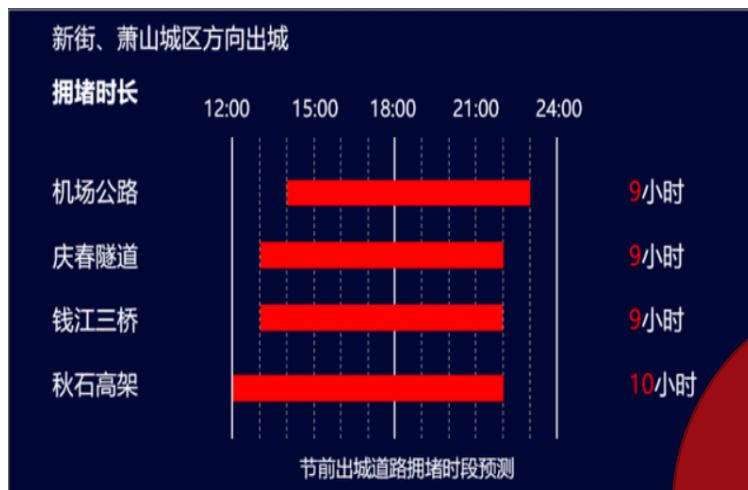
基础知识

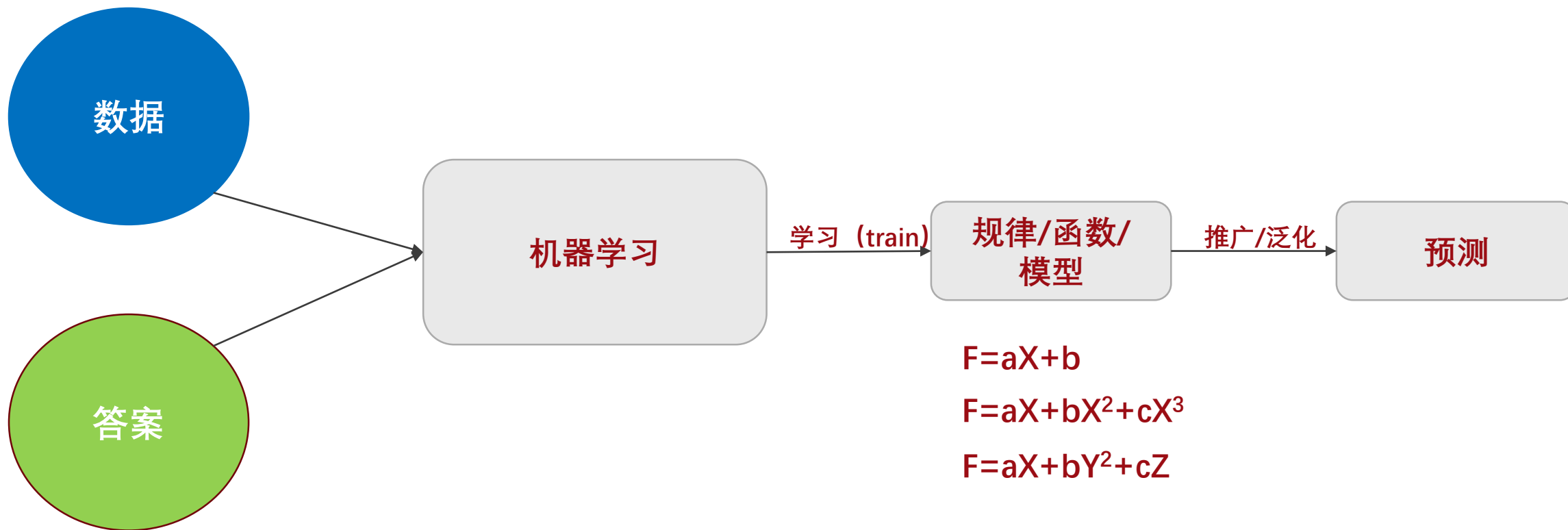


人工智能包括神经网络、决策树、支持向量机、随机森林、深度学习等多种方法。



人工智能技术蓝图

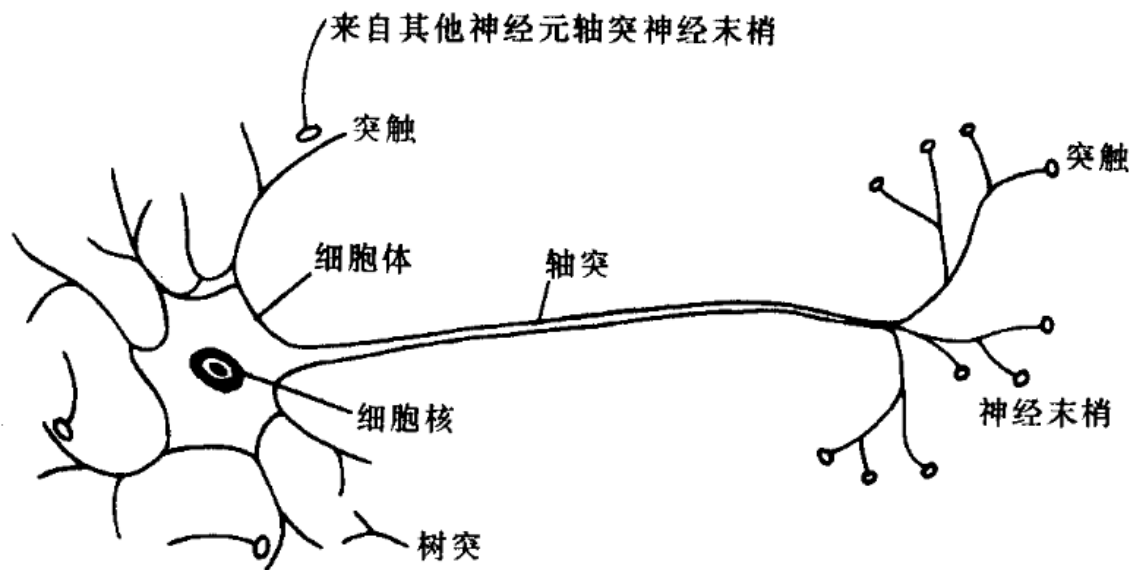






山东大学
SHANDONG UNIVERSITY

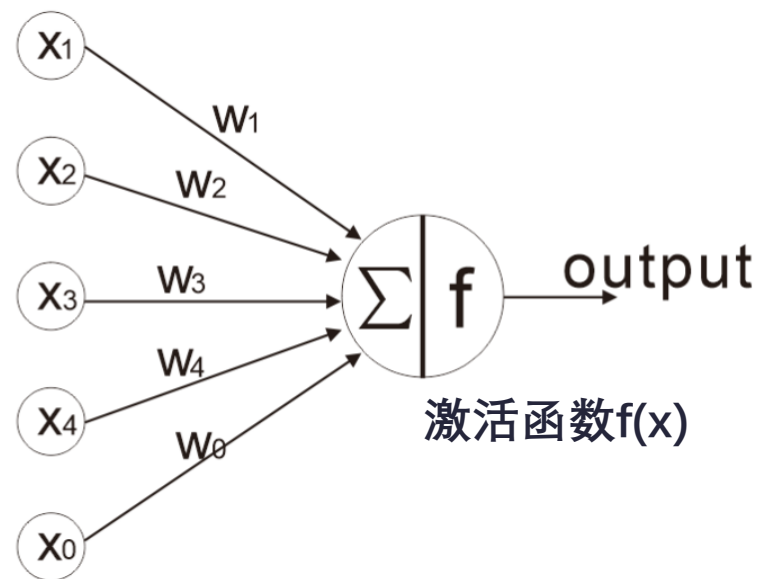
深度神经网络



人体神经元



input

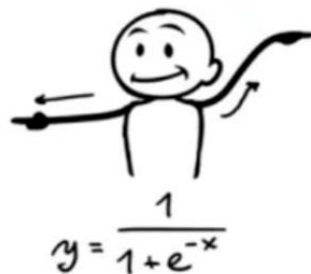


https://blog.csdn.net/qq_38299170

人工神经元



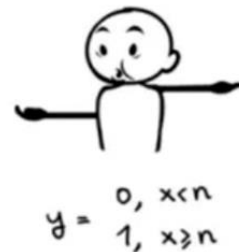
Sigmoid



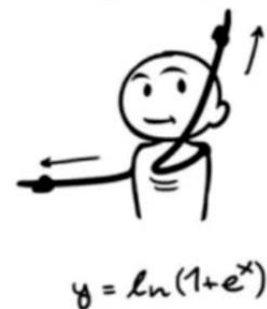
Tanh



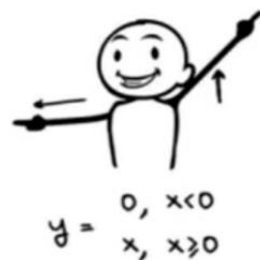
Step Function



Softplus



ReLU



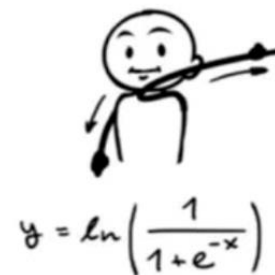
Softsign



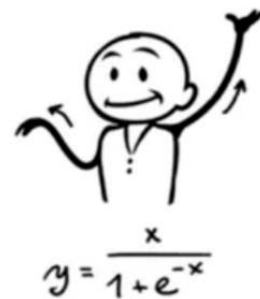
ELU



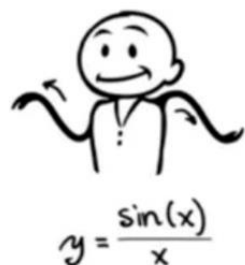
Log of Sigmoid



Swish



Sinc

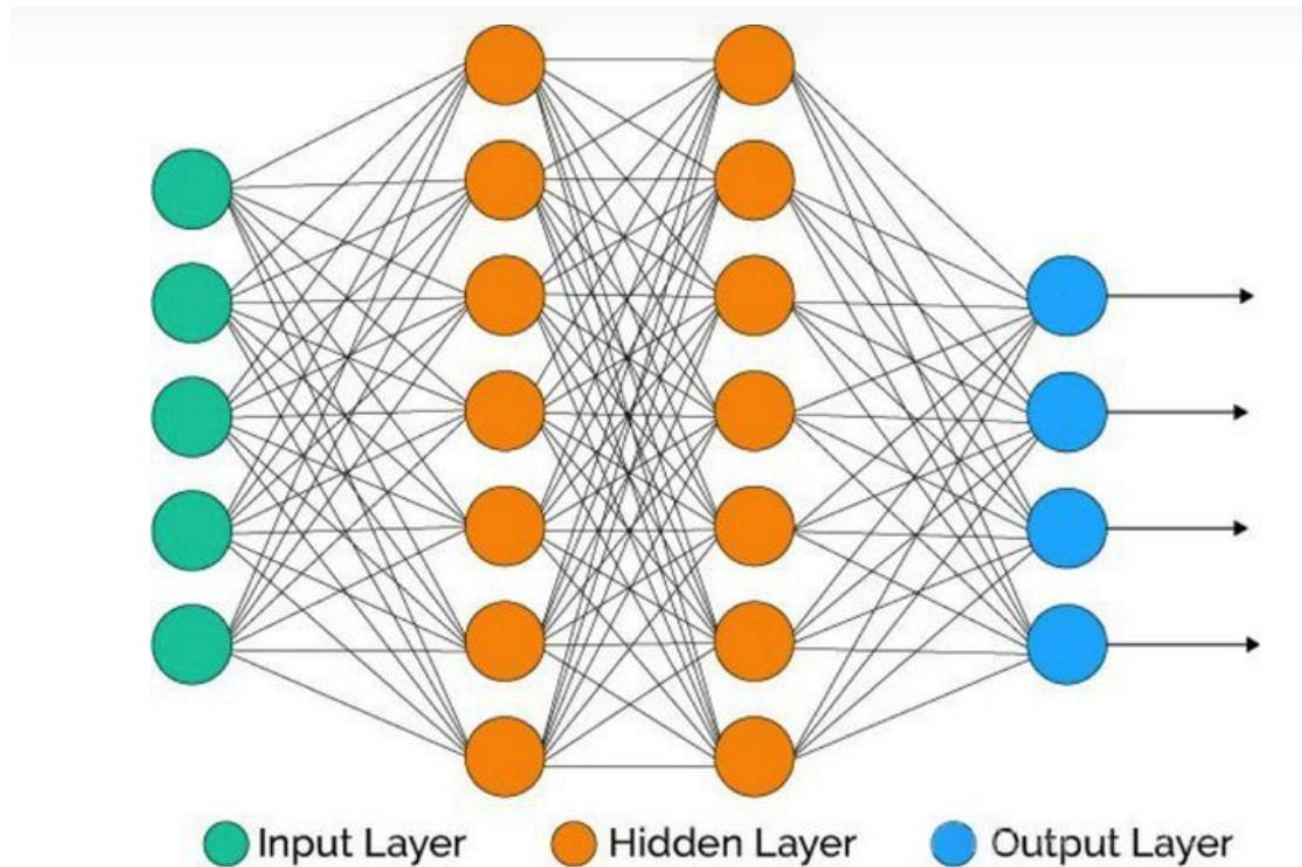


Leaky ReLU



Mish





多层神经网络



Colour Image

=



Red

+



Green

+



Blue

					141	142	143	144	145
					151	152	153	154	155
					161	162	163	164	165
	35	36	37	38	39	173	174	175	
	45	46	47	48	49	183	184	185	
	55	56	57	58	59	193	194	195	
	65	66	67	68	69				
31	32	33	34	35	6	77	78	79	
41	42	43	44	45	16	87	88	89	
51	52	53	54	55					
61	62	63	64	65					
71	72	73	74	75					
81	82	83	84	85					

R

G

B

141	142	143	144	145
151	152	153	154	155
161	162	163	164	165
171	172	173	174	175
181	182	183	184	185
191	192	193	194	195

R

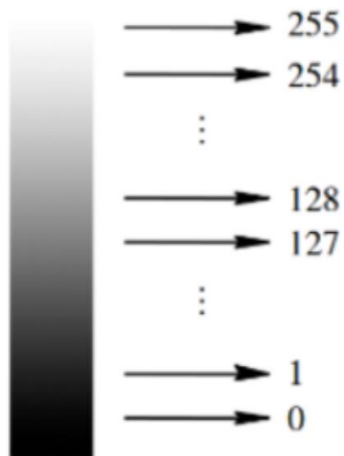
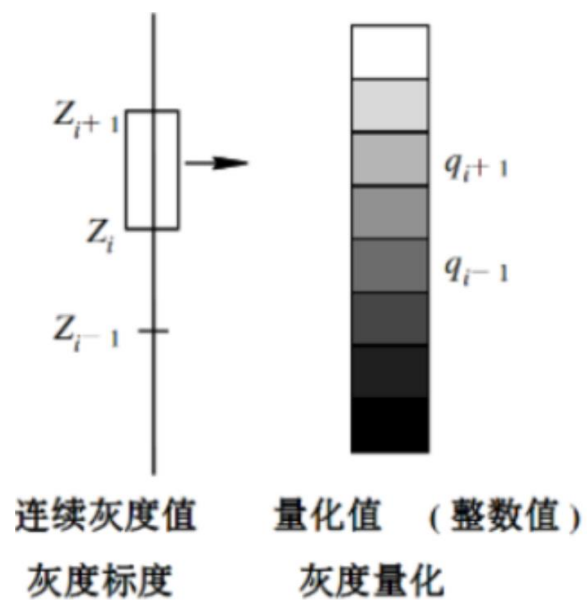


What We See

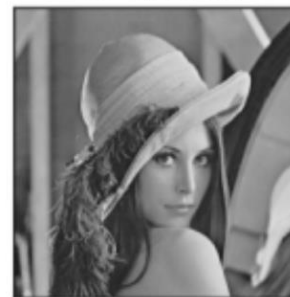
```
08 02 22 97 38 15 00 40 00 75 04 05 07 78 32 12 50 77 91 08
49 49 99 40 17 81 18 57 60 87 17 40 98 43 69 48 04 56 62 00
81 49 31 73 55 79 14 29 93 71 40 67 53 88 30 03 49 13 36 65
52 70 95 23 04 60 11 42 69 24 68 56 01 32 56 71 37 02 36 91
22 31 16 71 51 67 63 83 41 92 36 54 22 40 40 28 66 33 13 60
24 47 32 60 99 03 45 02 44 75 33 53 78 36 84 20 35 17 12 50
32 98 81 28 64 23 67 10 26 38 40 67 59 54 70 66 18 38 64 70
67 26 20 68 02 62 12 20 95 63 94 39 63 08 40 91 66 49 94 21
24 55 58 05 66 73 99 26 97 17 78 78 96 83 14 88 34 89 63 72
21 36 23 09 75 00 76 44 20 45 55 14 00 61 33 97 34 31 33 95
78 17 53 28 22 75 31 67 15 94 03 80 04 62 16 14 09 53 56 92
16 39 05 42 96 35 31 47 55 58 88 24 00 17 54 24 36 29 85 57
86 56 00 48 55 71 89 07 05 44 44 37 44 40 21 58 51 54 17 58
19 80 81 68 05 94 47 69 28 73 92 13 86 52 17 77 04 89 55 40
04 52 08 83 97 35 99 16 07 97 57 32 16 26 26 79 33 27 98 66
88 36 68 87 57 62 20 72 03 46 33 67 46 55 12 32 63 93 53 69
04 42 16 73 38 25 39 11 24 94 72 18 08 46 29 32 40 62 76 36
20 69 36 41 72 30 23 88 34 62 99 69 82 67 59 85 74 04 36 16
20 73 35 29 78 31 90 01 74 31 49 71 48 86 81 16 23 97 05 54
01 70 54 71 83 51 54 69 16 92 33 48 61 43 52 01 89 19 67 48
```

What Computers See

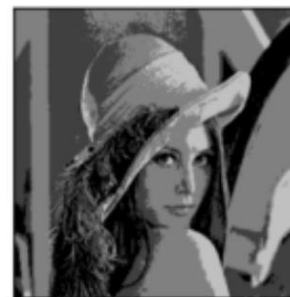
有一张 JPG 格式的 480×480 大小的彩色图片，那么它对应的数组就有 $480 \times 480 \times 3$ 个元素。其中每个元素的数值从 0 到 255 不等，其描述了对应那一点的像素灰度



(a) 原始图像



(c) 量化-L4



(b) 量化-L2



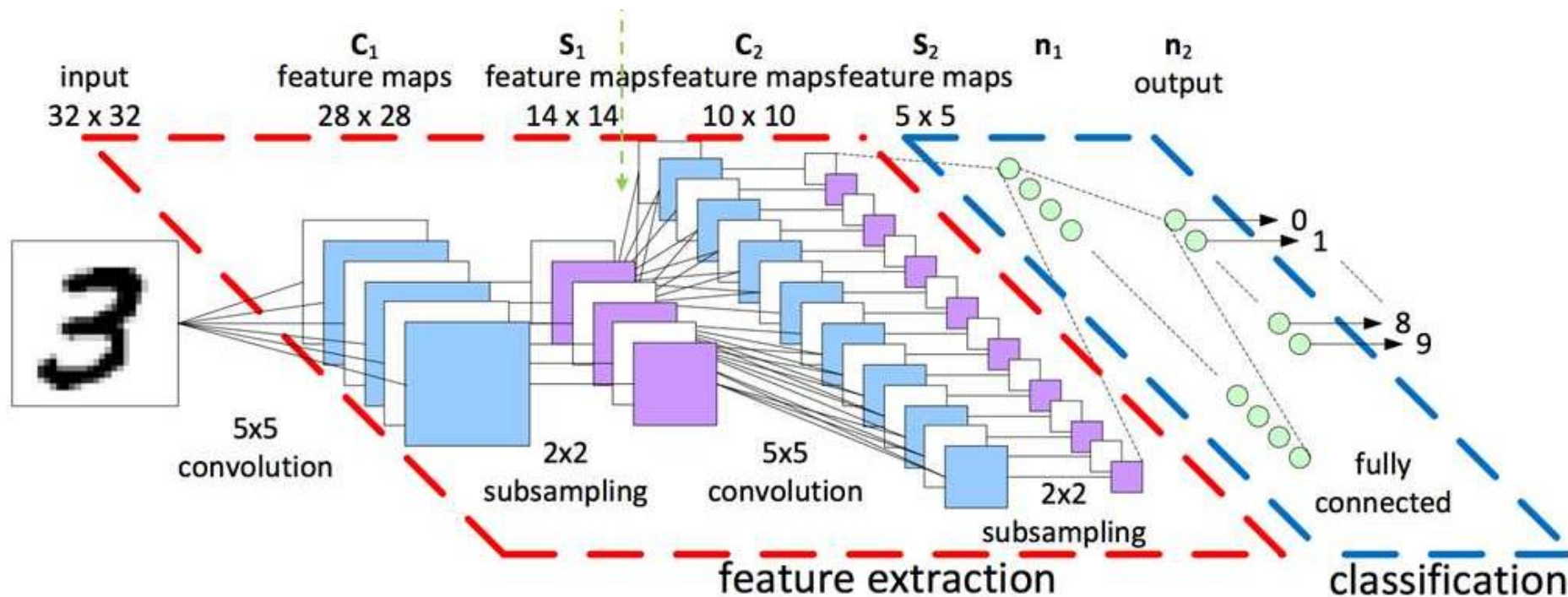
(d) 量化-L8





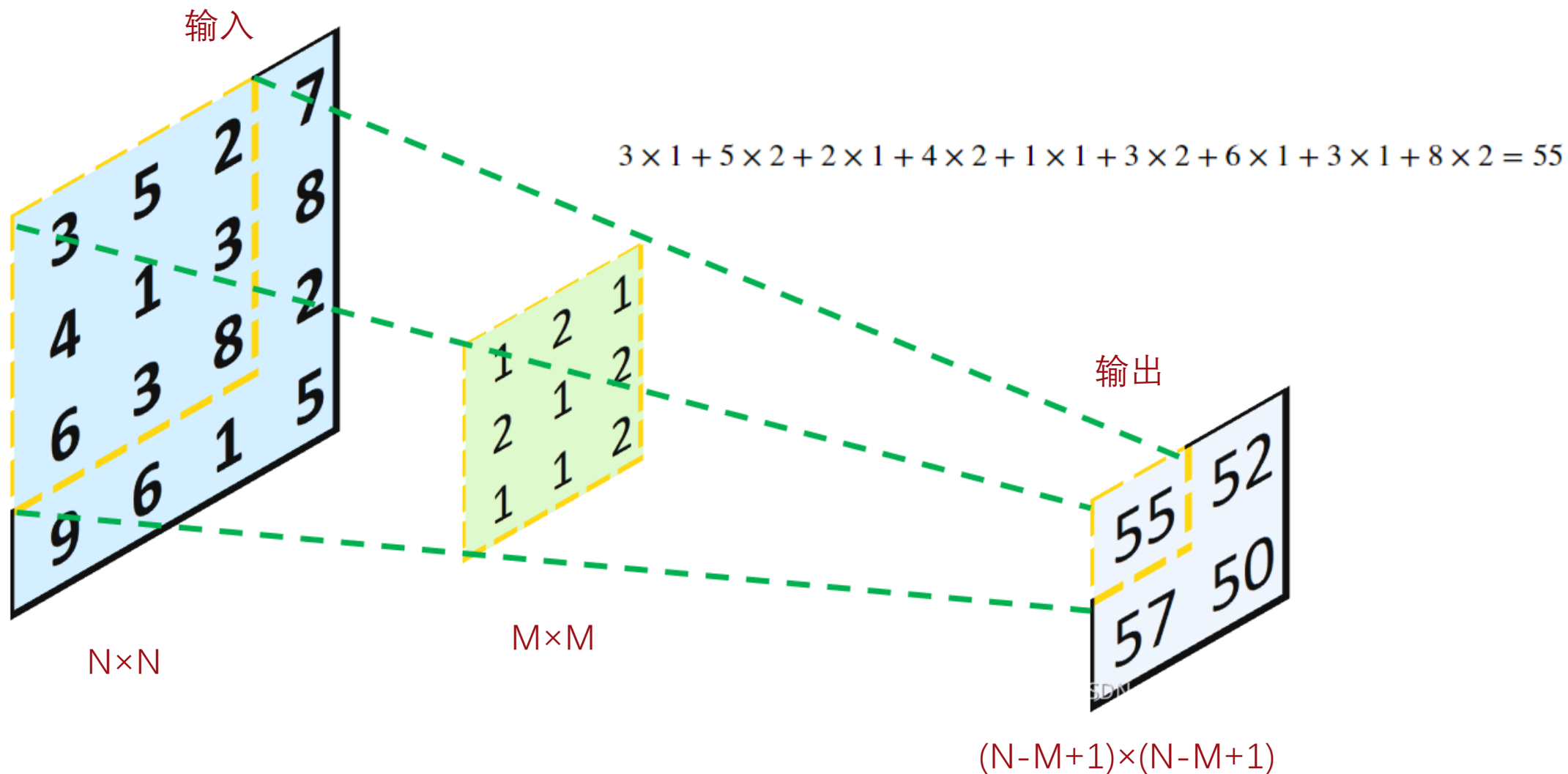
一个典型的卷积神经网络(Convolutional Neural Network,简称CNN)由多个卷积层、池化层和全连接层组成:

- **卷积层**: 卷积层(Convolution layer)是CNN的核心, 它负责提取输入数据的局部特征。每个卷积层都由多个卷积核(也称为滤波器)组成, 每个卷积核都可以捕捉到输入数据的一种特定特征。例如, 一个卷积核检测边缘, 另一个卷积核检测纹理。
- **池化层**: 池化层(pooling layer)位于卷积层之后, 它的主要功能是降低数据的维度, 从而减少计算量。同时, 池化层也可以提高模型的空间不变性, 使模型对输入数据的小变化更加鲁棒。
- **全连接层**: 全连接层(fully connected layer)位于网络的最后, 它将所有的特征信息整合在一起, 用于做出最终的预测。全连接层的神经元与前一层的所有神经元都有连接, 因此被称为“全连接”。



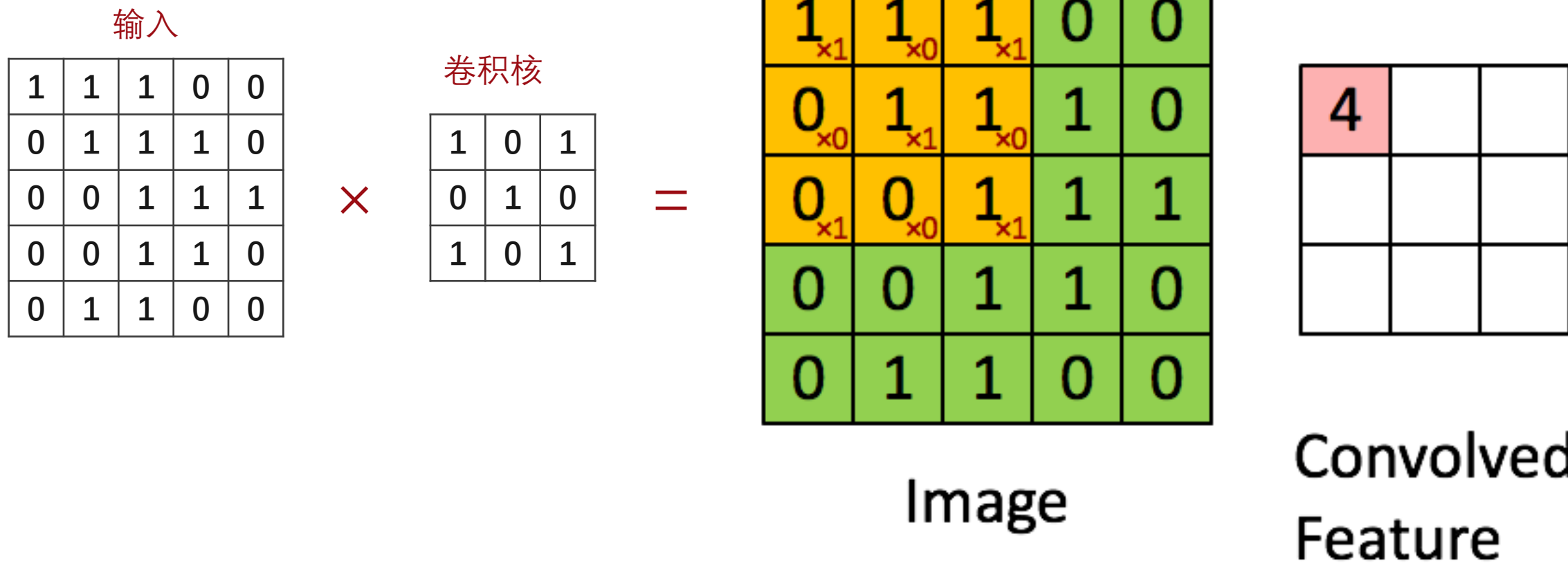


卷积 $\text{output}[i, j] = \text{sum}(\text{input}[i:i+k, j:j+k] * \text{kernel})$



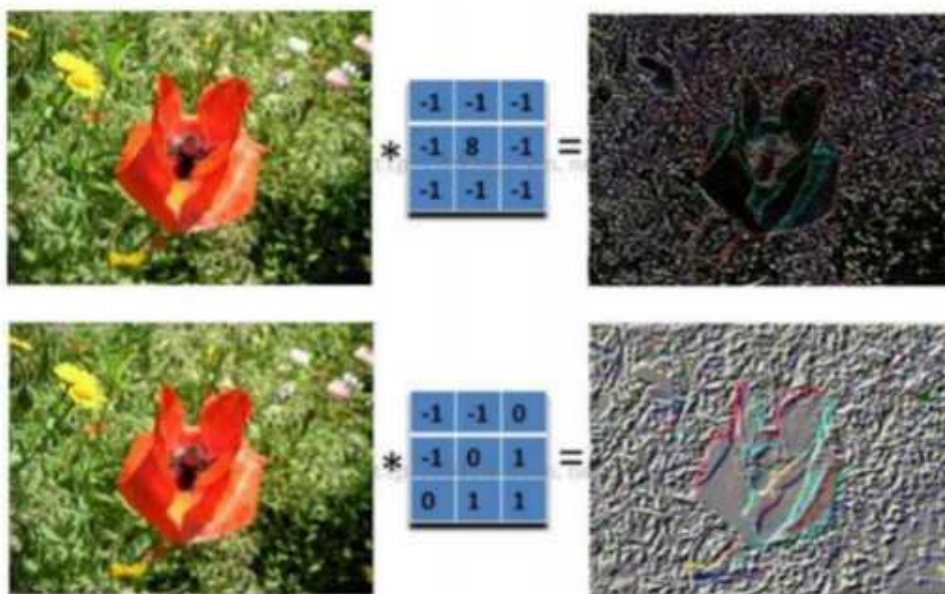


普遍使用的卷积核大小为 3×3 、 5×5 等;





1. 特征提取：卷积可以通过滑动一个卷积核（也称为滤波器）来提取输入信号的局部特征。卷积核的大小和形状不同，可以提取不同类型的特征。例如，在图像处理中，可以使用边缘检测卷积核来提取图像中的边缘特征。
2. 降噪：卷积可以通过滤波器对输入信号进行平滑处理，从而去除噪声。例如，在图像处理中，可以使用高斯滤波器来对图像进行平滑处理，从而去除图像中的噪声。
3. 压缩：卷积可以通过降低信号的维度来实现数据压缩。例如，在语音处理中，可以使用卷积将语音信号压缩成更小的维度，从而减少存储空间和计算成本。





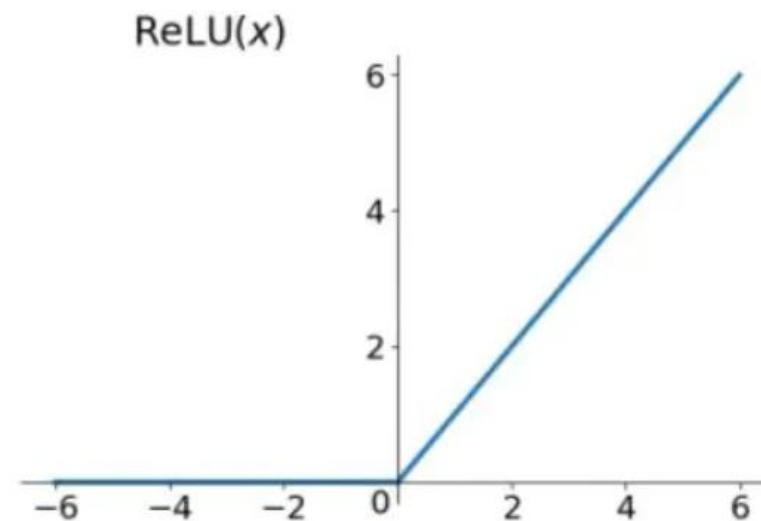
在计算完卷积之后，我们通常会使用一个激活函数来增加模型的非线性。

激活函数的作用是将卷积的结果映射到一个非线性的空间，这使得模型可以学习到更复杂的特征。

激活函数层能够增加模型的灵活性、加速收敛过程、避免梯度消失等。

常用ReLU (Rectified Linear Unit) 激活函数

$$y = \begin{cases} x & (x > 0) \\ 0 & (x \leq 0) \end{cases}$$



ReLU函数：如果正向传播时的输入 x 大于0，则反向传播会将上游的值原封不动地传给下游。反过来，如果正向传播时的 x 小于等于0，则反向传播中传给下游的信号将停在此处。

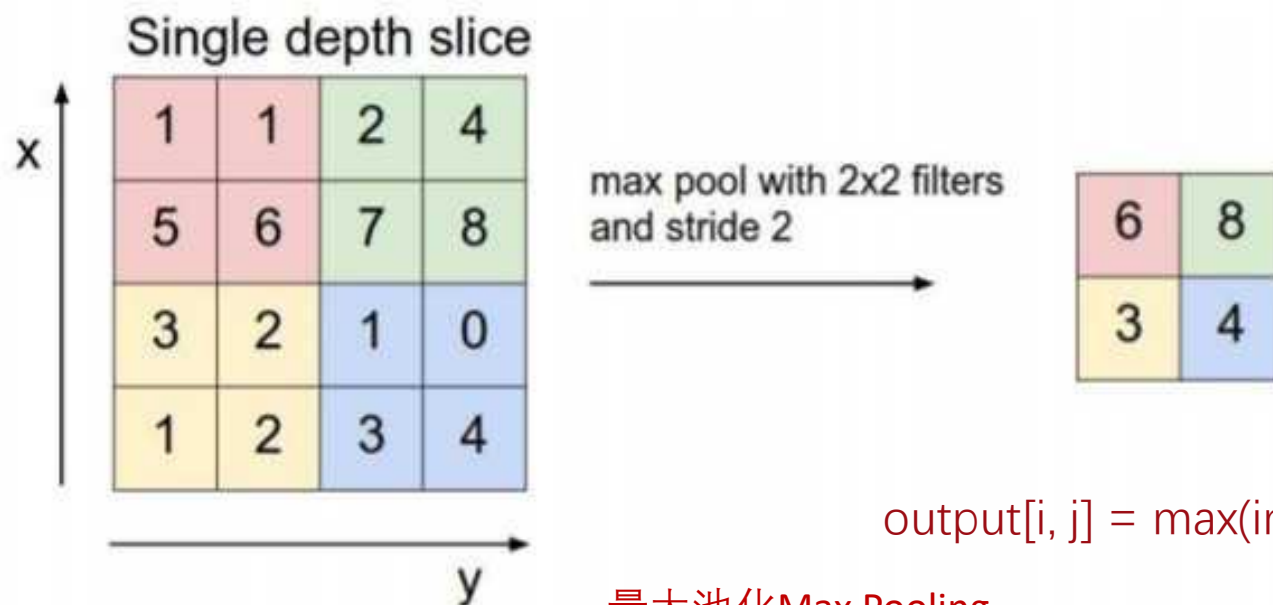


池化层 (pooling layer) 将输入数据划分为多个小的区域，然后在每个区域上计算一个统计值，如最大值或者平均值。这个过程可以看作是对输入数据进行降维，同时保留最重要的信息，从而提高计算效率和模型泛化能力。

具体来说，池化层可以分为最大池化 (max pooling) 和平均池化 (average pooling) 两种操作

- 最大池化将每个特征图划分为不重叠的区域，在每个区域内选择最大的值作为该区域的输出；
- 平均池化则是选择每个区域内的平均值

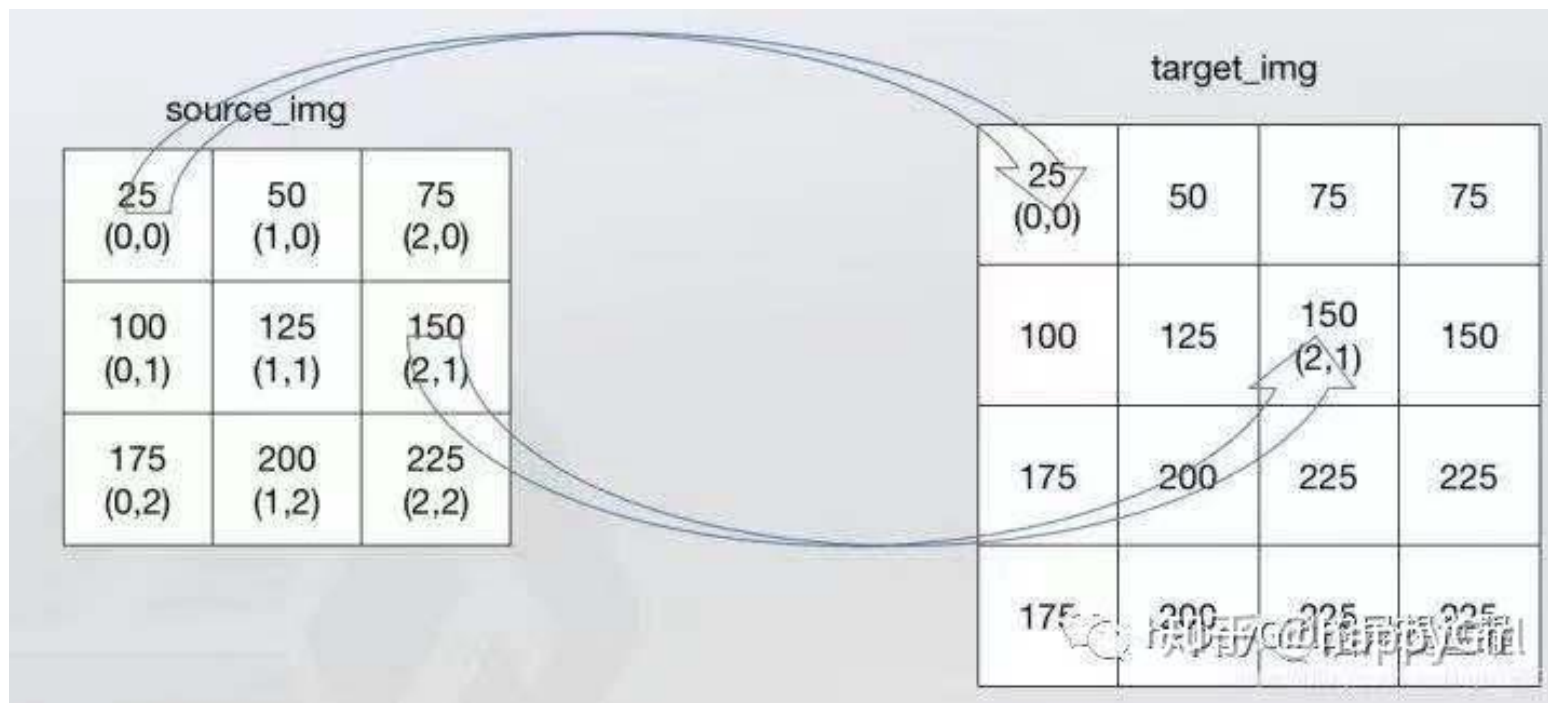
在实际应用中，**最大池化更为常用**，因为它能更好的保留图像中的边缘信息。



最大池化Max Pooling

上采样 (upsampling)：又名放大图像、图像插值，将 $N \times M$ 大小图像改为 $W \times H$ ，其中 $(W > N, H > M)$
主要目的是放大原图像，从而可以显示在更高分辨率的显示设备上

上采样常见的方法：最近邻插值法、双线性插值(bilinear)、反卷积(Transposed Convolution)、反池化(Unpooling);



最近邻插值法

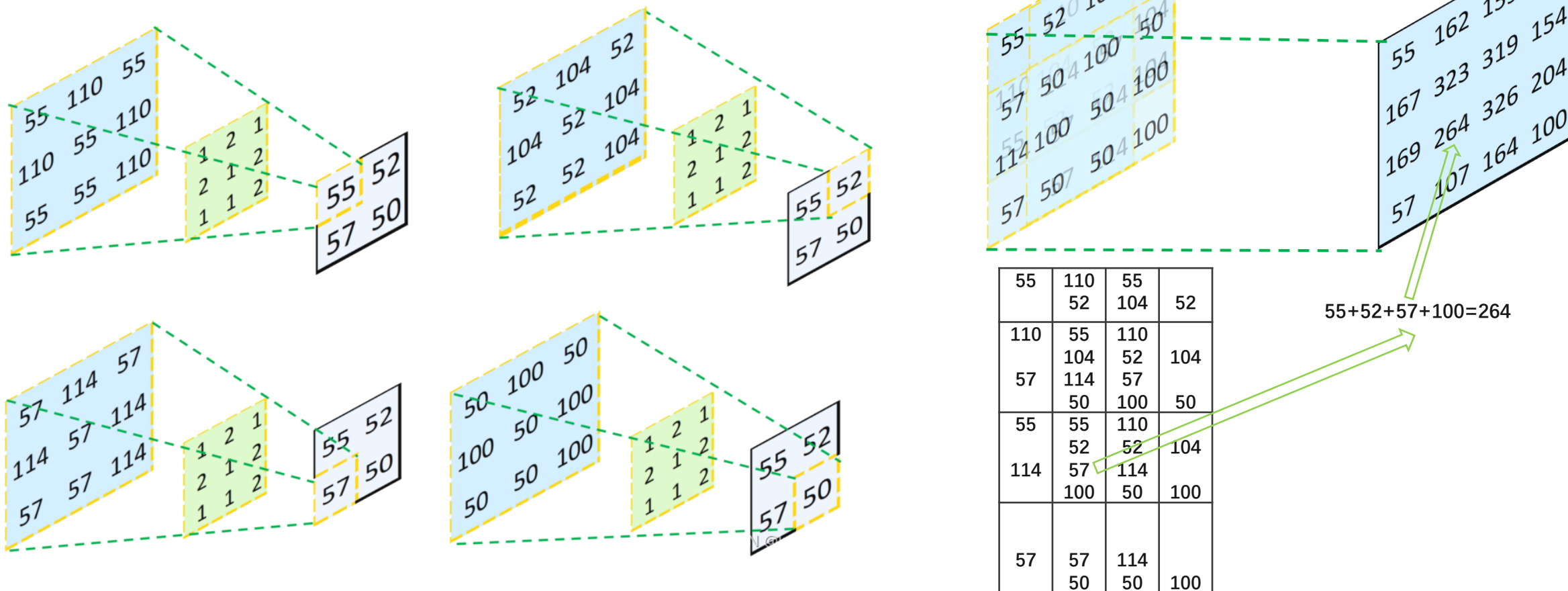


上采样——反卷积(Transposed Convolution)



山东大学
SHANDONG UNIVERSITY

反卷积（也称为转置卷积）是卷积的逆操作

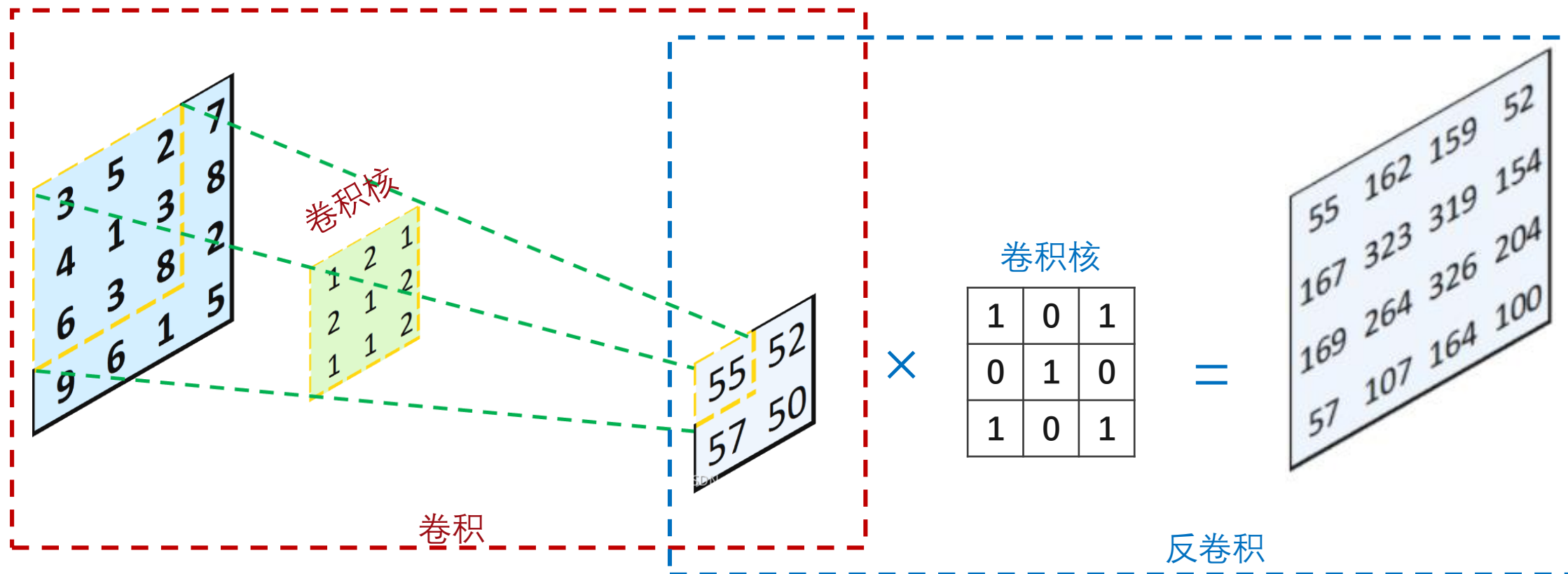


左上角的图，将输出的55按照绿色的线的逆方向投影回去，可以得到[[55,110,55],[110,55,110],[55,55,110]]的结果；

在声学、结构动力学等领域，通过反卷积可以知道噪声源、振动源的位置、强度，甚至分离出源信号的数量，获得类型、频率等信息



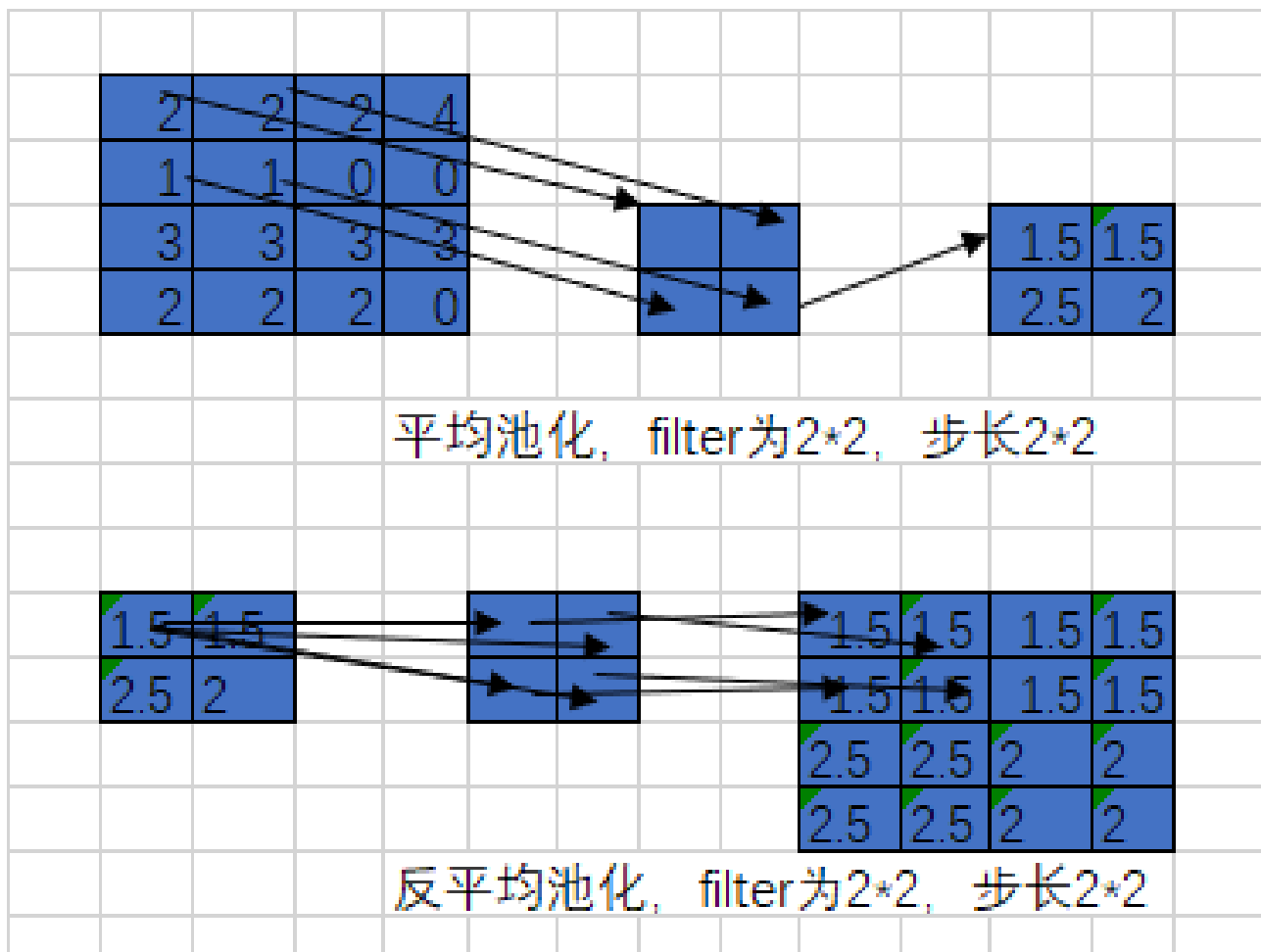
卷积 vs 反卷积



说明：卷积和反卷积并不是完全对等的可逆操作（即使采用相同的卷积核，卷积和反卷积得到的输入输出不同）
反卷积只能恢复尺寸，不能恢复数值



反池化(Unpooling)是池化的逆操作，是无法通过池化的结果还原出全部的原始数据。因为池化的过程就只保留了主要信息，舍去部分信息。如果想从池化后的这些主要信息恢复出全部信息，则存在信息缺失，这时只能通过补位来实现最大程度的信息完整。

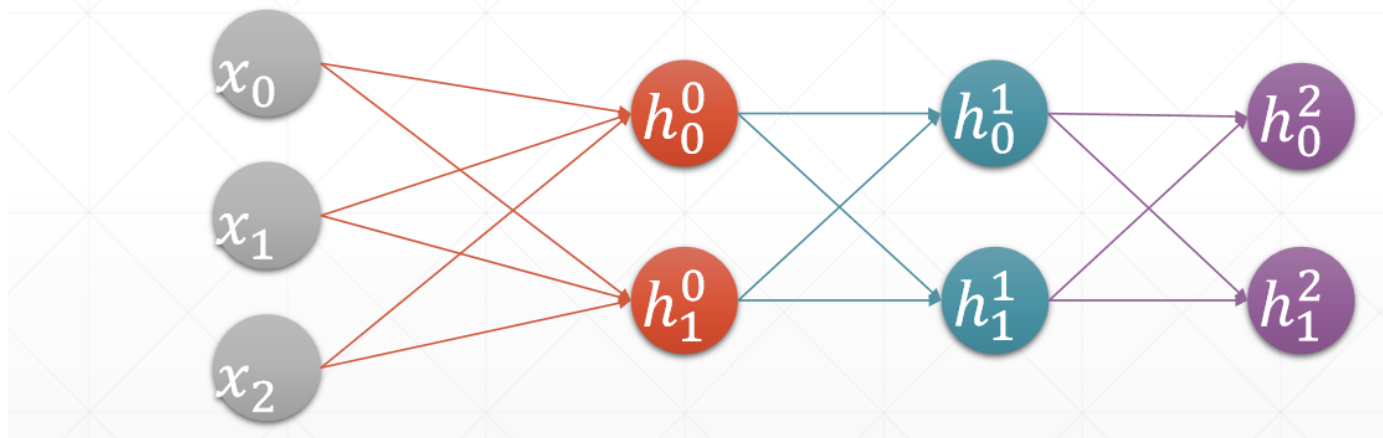




全连接层(Fully Connected Layer)把卷积层和池化层提取出来的所有局部特征重新通过权值矩阵组装成一个完整的图，因为用到了所有的局部特征，所以叫全连接。

全连接层作用是将输入特征映射到输出结果，通常在神经网络的最后一层使用，用于分类、回归等任务。全连接层的输出结果可以看作是对输入特征的一种非线性变换，这种变换可以将输入特征空间映射到输出结果空间，从而实现模型的复杂性和非线性拟合能力。

- `keras.Sequential([layer1, layer2, layer3])`



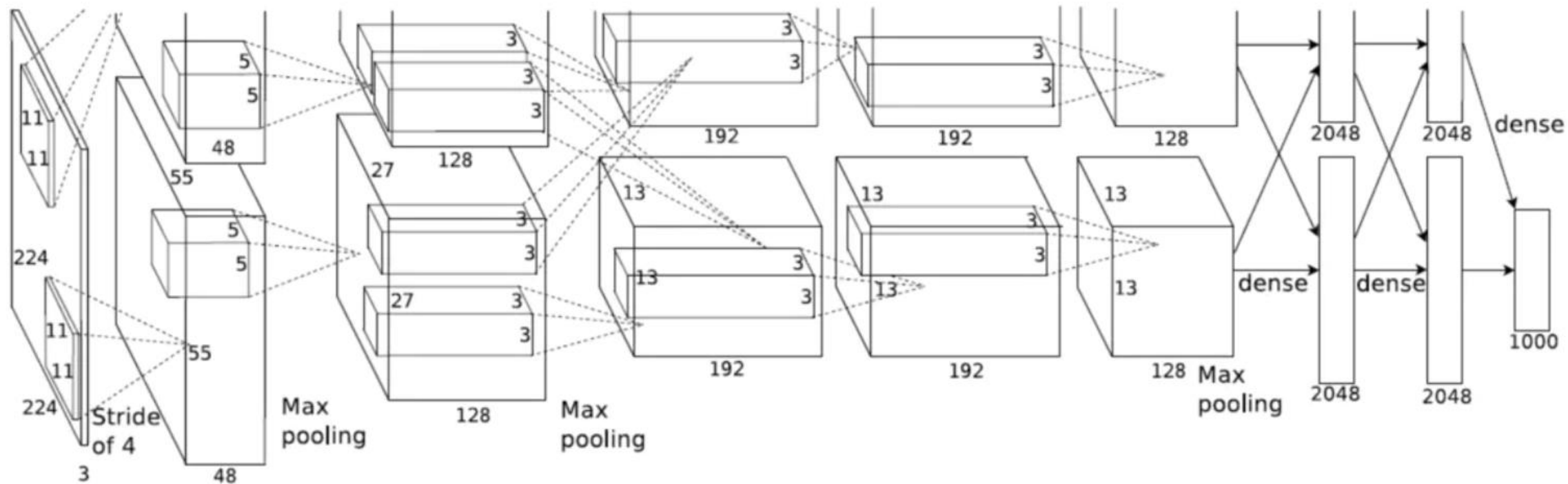
过拟合 (overfitting) 是指机器学习模型在训练数据集上表现得很好, 但是在新的、未见过的数据集上表现得很差的现象。

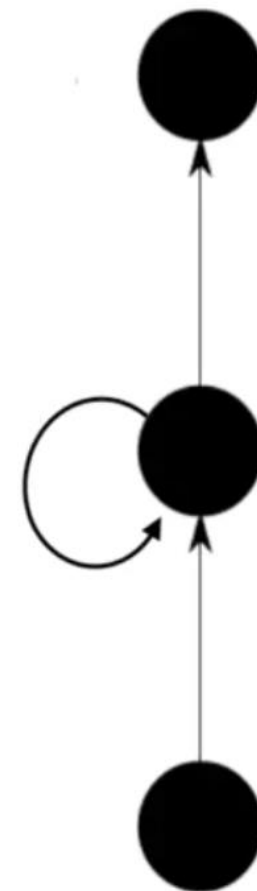
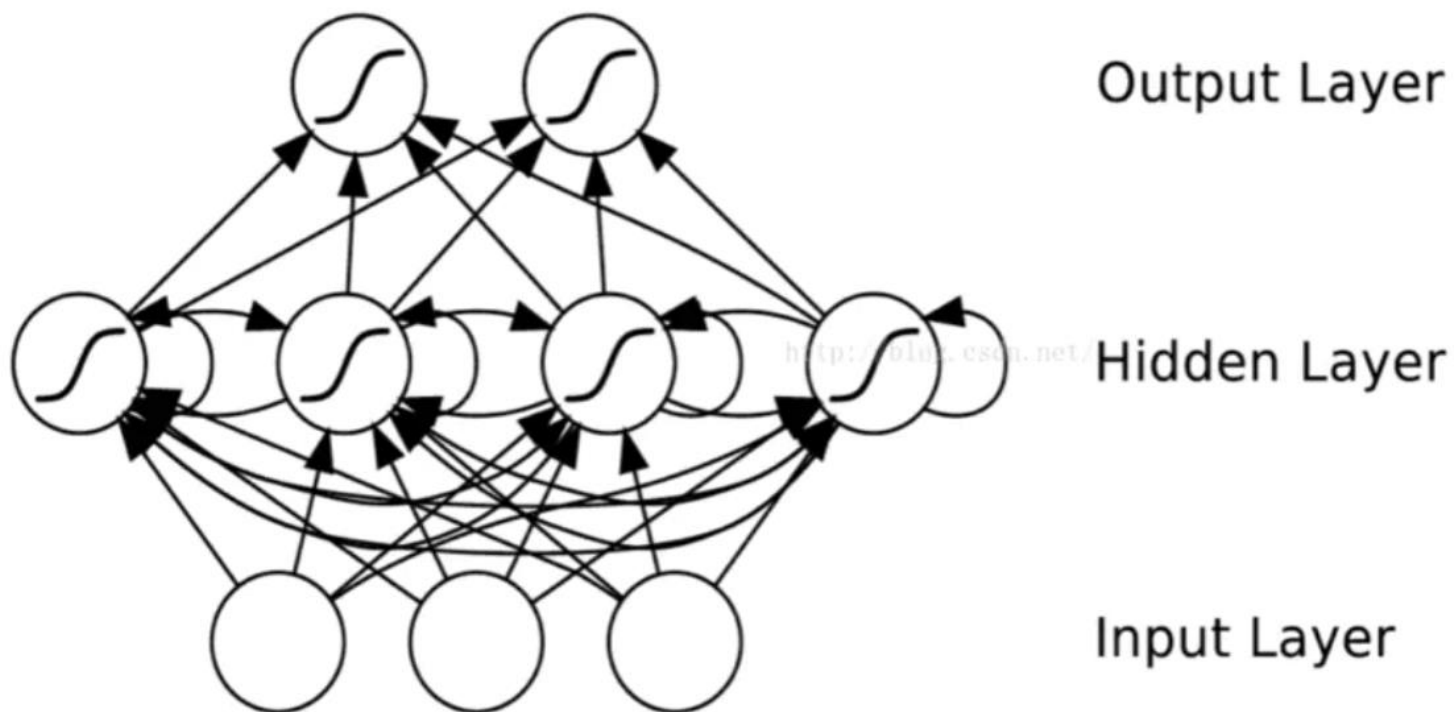
过拟合的原因一般是模型过于复杂, 导致模型在训练数据上过于准确地拟合了噪声和细节, 从而失去了泛化能力。过拟合的表现包括模型在训练集上的损失 (或错误率) 很小, 但在测试集或验证集上的损失很大, 或者模型在训练集上的预测效果很好, 但是在测试集或验证集上的预测效果很差。

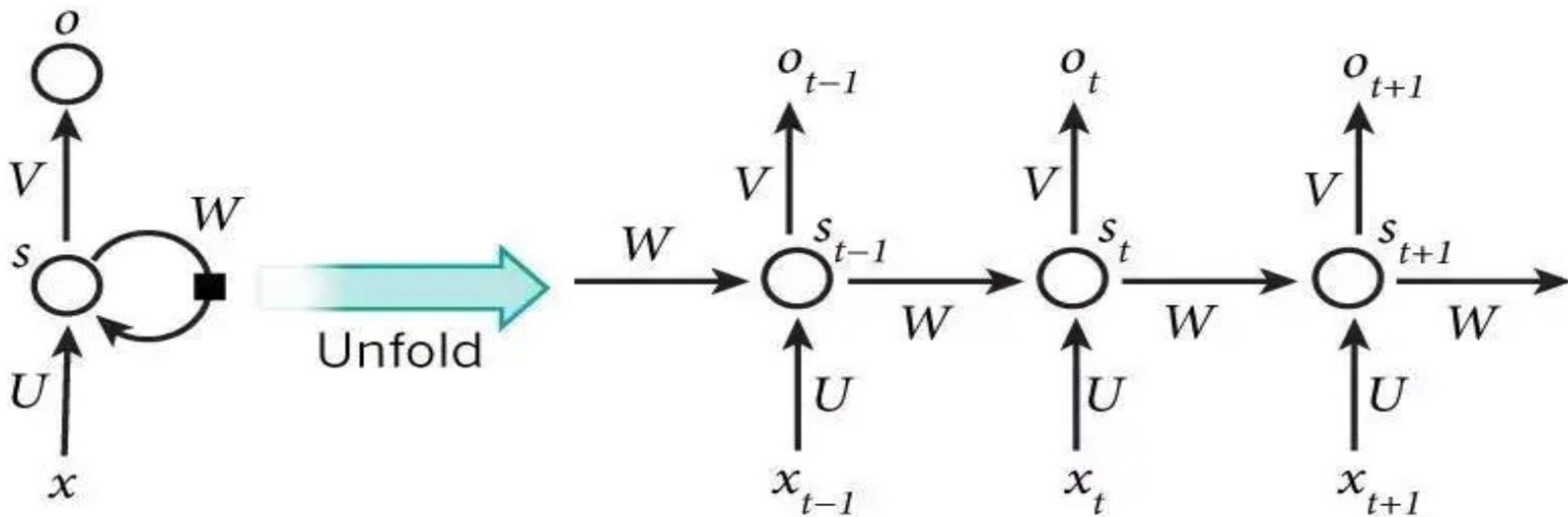
解决过拟合的方法包括:

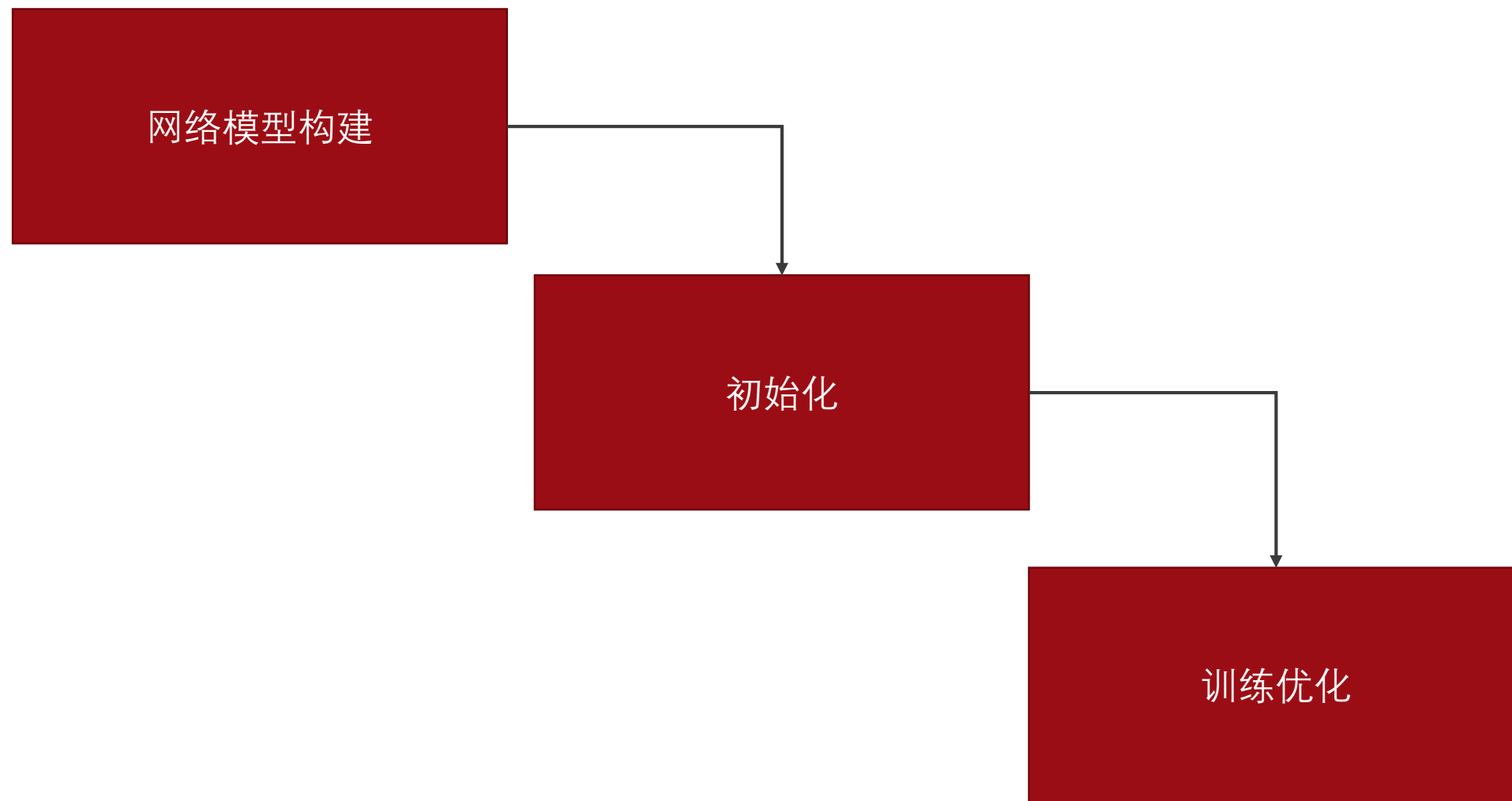
- 简化模型结构, 如减少模型参数、降低网络层数等;
- 增加数据量, 如采集更多的数据或使用数据增强技术;
- 添加正则化, 如L1、L2正则化、dropout等;
- 早停法 (early stopping), 即在模型在验证集上表现开始下降时停止训练

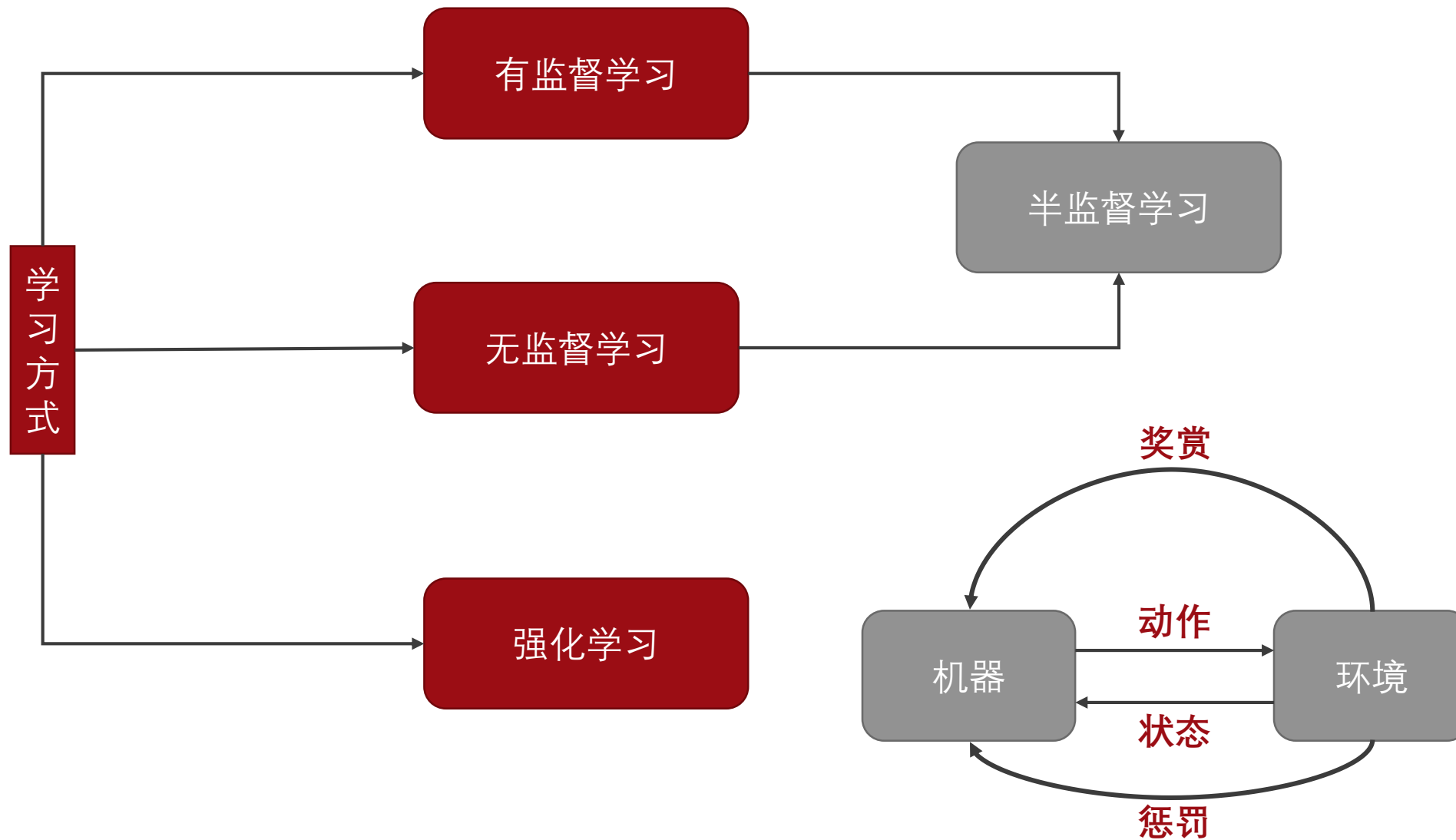
Dropout是一种常用的正则化方法, 用于减少神经网络的过拟合现象。它的基本思想是在训练神经网络的过程中, 随机地将一部分神经元的输出值置为0, 从而使得神经网络的结构变得不稳定, 从而强制网络学习到更加鲁棒的特征表示。

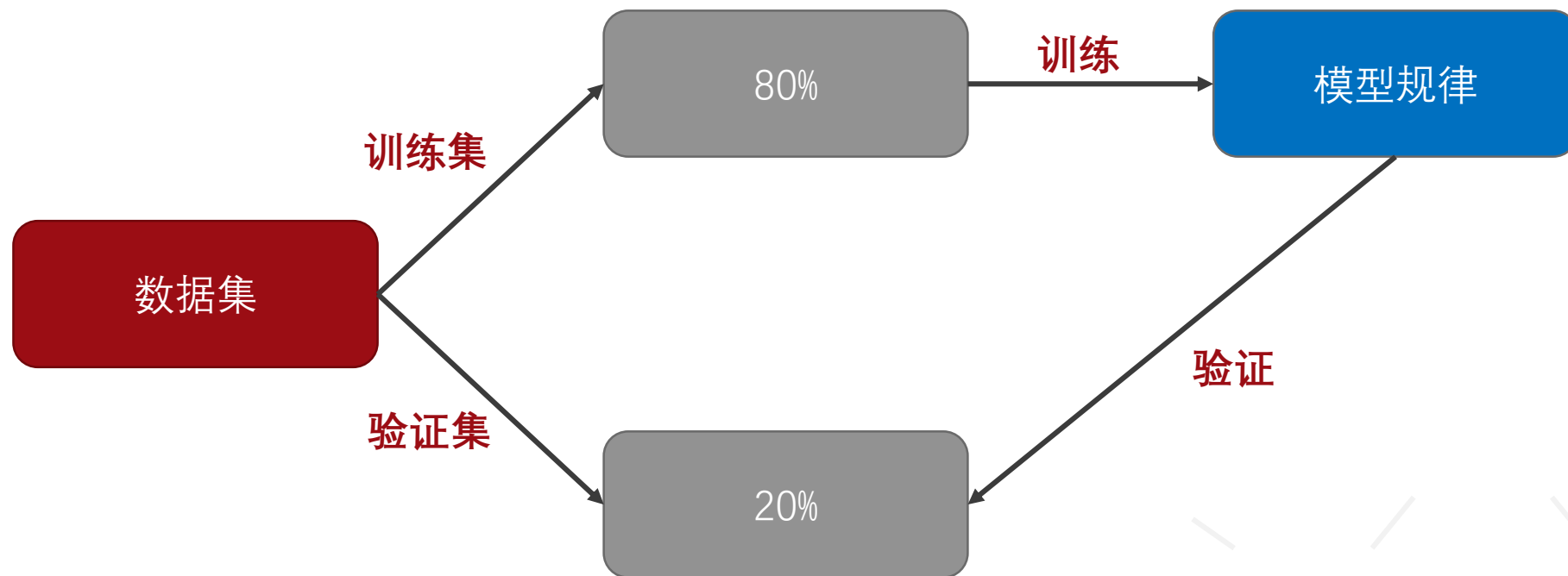






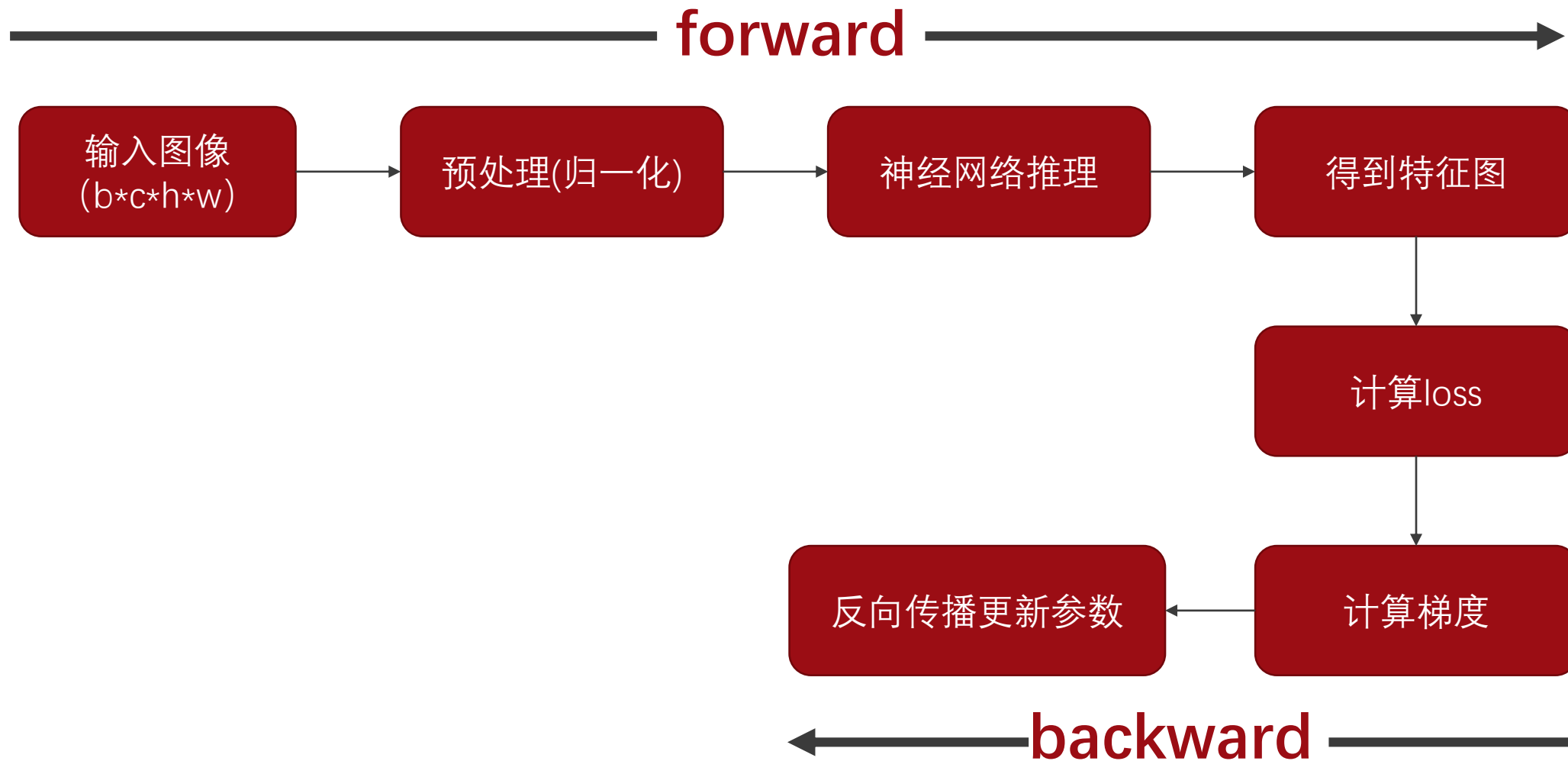






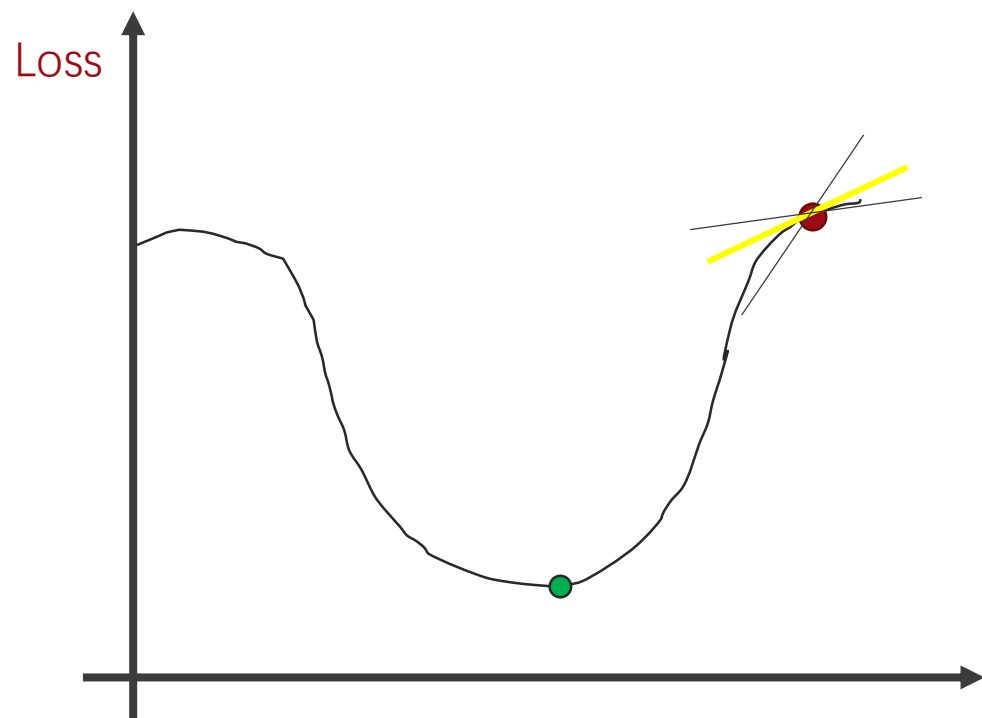


BP(forward and backward)



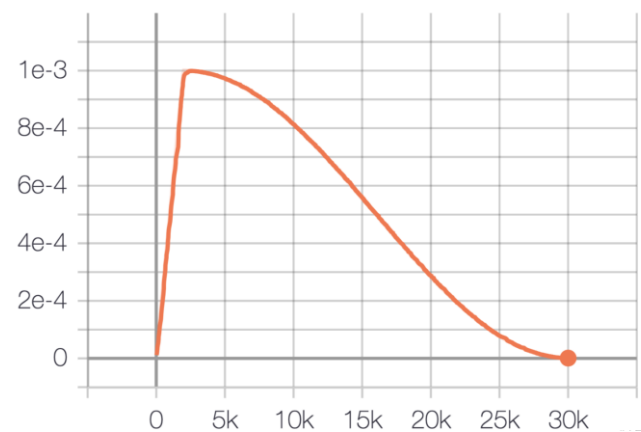


loss、学习率与梯度下降



梯度:loss下降最快的方向

lr:移动的步长



余弦退火



山东大学
SHANDONG UNIVERSITY

谢谢