北京交通大学
BEIJING JIAOTONG UNIVERSITY




《大数据概论》
大数据分析与挖掘

鲍鹏
软件学院

目录

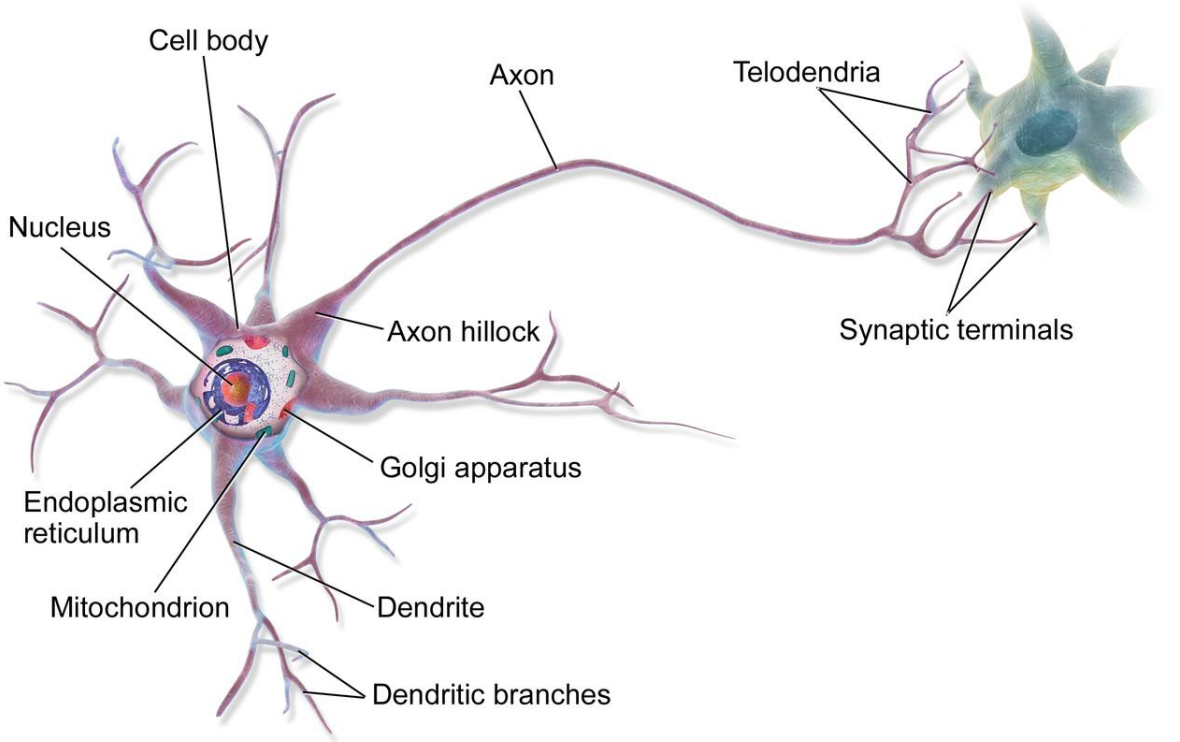
- 数据理解与特征工程
- 常用数据挖掘算法
- 高级数据建模技术
  - 从人工神经网络到深度神经网络
  - 卷积神经网络
  - 循环神经网络
  - 生成对抗网络
- 数据可视化技术

从人工神经网络到深度神经网络

- 人类大脑中的神经网络是由一个个神经元组成的，神经元由细胞体、轴突、动作电位、突出末端、突触前神经元的轴突和树突等部分组成。
- 身体的不同部位产生的信息通过不同的路径到达神经元，神经元处理它并产生一个输出。
- 神经元可能被连接到另一个神经元。

从人工神经网络到深度神经网络

- 单个神经细胞只有两种状态：兴奋和抑制



生物神经元

从人工神经网络到深度神经网络

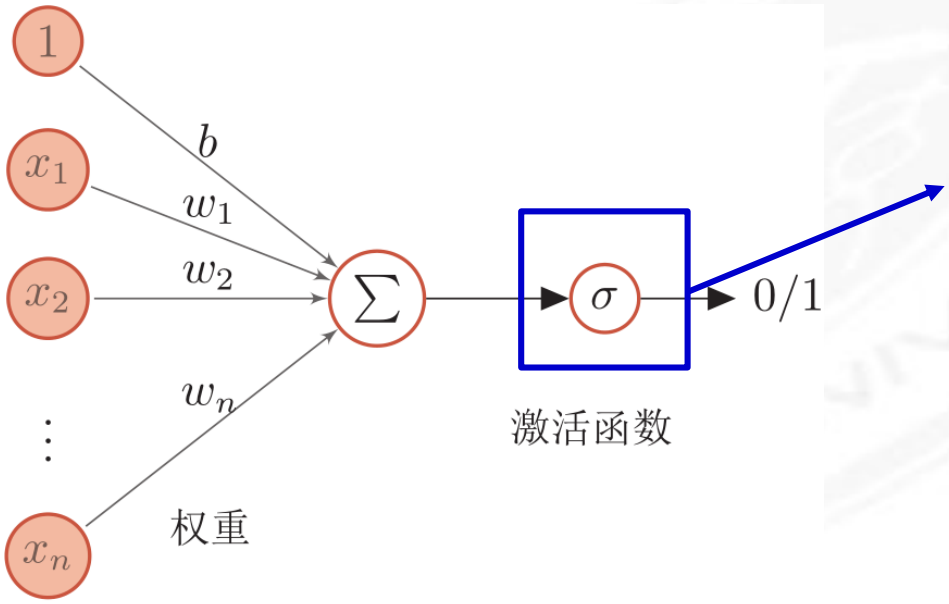
- 根据神经网络的基本生物学模型，建立如下数学模型：

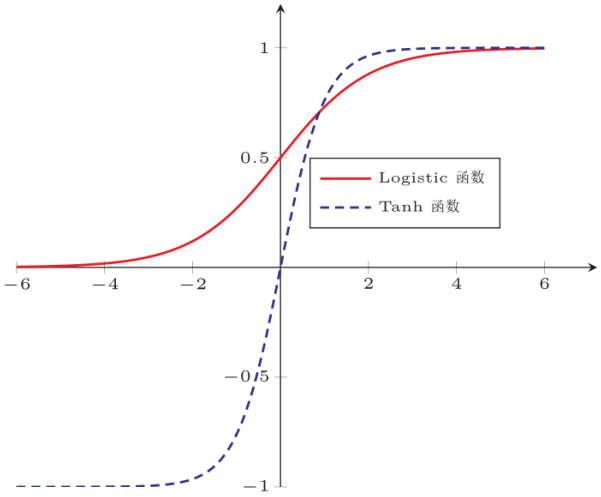
$$y = f\left(\sum_{i=1}^{n} x_i w_i - \theta\right)$$

- 其中，x_i代表着从不同路径到达的信号值，w_i代表不同路径上的权重，θ是设置的阈值，f是激励函数，决定最后的响应内容。

从人工神经网络到深度神经网络

- 常用的激活函数：阈值函数、双向阈值函数、S型函数、双曲正切函数、高斯函数。





从人工神经网络到深度神经网络

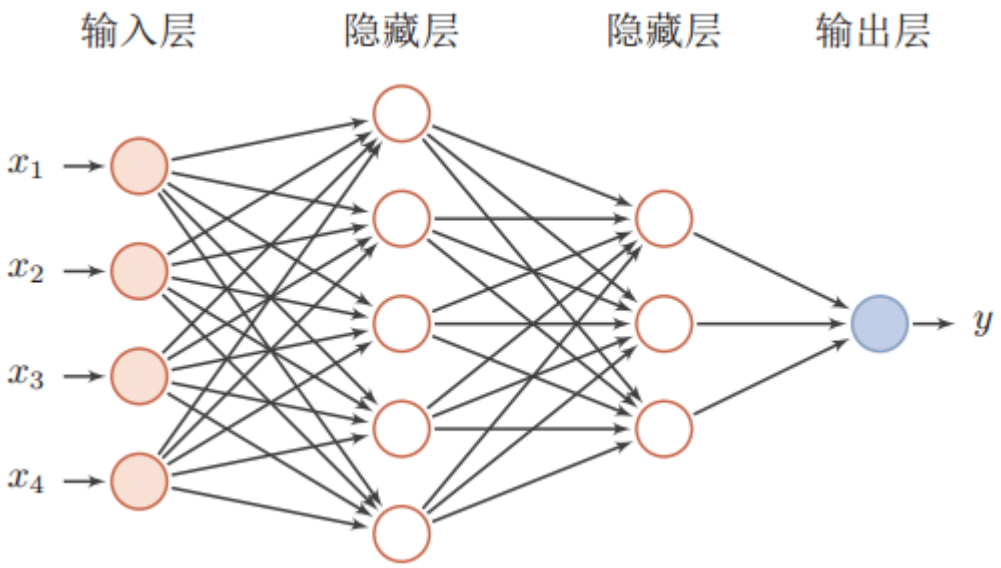
- 人工神经网络主要由大量的神经元以及它们之间的有向连接构成。
- 最早的人工神经网络是单层神经网络，由输入层和输出层组成。
  - 输入层里的输入单元只负责传输数据，不做计算。
  - 输出层里的输出单元则需要对前面一层的输入进行计算。

从人工神经网络到深度神经网络

- BP神经网络
  - BP神经网络由输入层、隐藏层和输出层组成，其中隐含层和输出层都实现计算功能，属于两层神经网络。
  - 输入值从输入层单元通过连接权重加权激活逐层向前传播经过隐藏层最后到达输出层得到输出。在信号向前传递的过程中，网络的权值是固定不变的，每一层神经元的状态只影响下一层神经元的状态。
  - 神经网络的本质是通过参数与激励函数来拟合特征与目标之间的真实函数关系。
  - 两层神经网络可以无限逼近任意连续函数。

从人工神经网络到深度神经网络

- **BP神经网络**
  - BP网络是一种前馈神经网络，各神经元分层排列。
  - 每个神经元只与不同层神经元相连，接收前一层的输出，并输出给下一层。



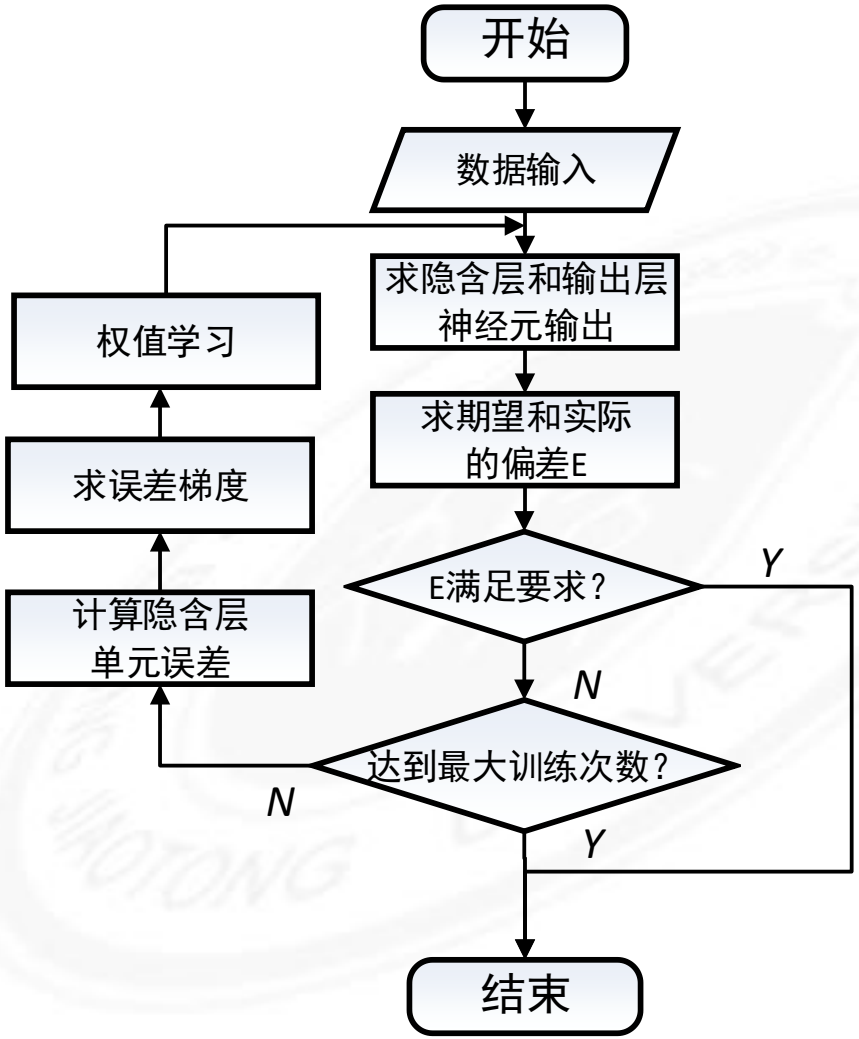
从人工神经网络到深度神经网络

- BP神经网络——反向传播算法
  - 反向传播是指从输出层开始沿着相反的方向来逐层调整参数的过程。
  - 网络的实际输出与期望输出之间的差值即为误差信号。
  - 误差信号的反向传播过程由输出端开始逐层向前传播。
  - 在反向传播中，网络的权值由误差反馈进行调节，通过权值的不断修正使网络的实际输出更加接近期望输出。

从人工神经网络到深度神经网络

- BP神经网络——代价函数
  - 在回归问题中，指定代价函数以使目标变量的真实值和预测值的距离最小。
  - 代价函数描述了网络输出与真实值之间的误差。
  - 通过随机梯度下降的方法最小化代价函数以提高网络精度。
  - 在代价函数中引入其他约束以满足设定要求。

从人工神经网络到深度神经网络



BP神经网络的流程

从人工神经网络到深度神经网络

- 深度学习
  - 深度学习算法是基于生物学对人脑的进一步认识而衍生的方法，将神经-中枢-人脑的工作原理设计成一个不断迭代、不断抽象的过程，以便得到最优数据特征表示的机器学习算法。
  - 从原始信号开始，先做低层抽象，然后逐渐向高层抽象迭代，从而组成深度学习算法的基本框架。
  - 一般特点：

    （1）使用链式结构非线性变换对数据进行多层抽象。

    （2）以寻求更适合待解决的问题的概念表示方法为目标。

    （3）形成一类具有代表性的特征表示学习方法。

从人工神经网络到深度神经网络

- 深度学习的优点
  - 概念提取可以由简单到复杂。深度学习中的“深度”指神经网络包含较多的隐藏层。
  - 每一层中非线性处理单元的构成方式取决于要解决的问题。每一层中学习模式可按需求灵活调整为有监督或者无监督学习，有利于调整学习策略，从而提高效率。
  - 有利于学习无标签数据。

从人工神经网络到深度神经网络

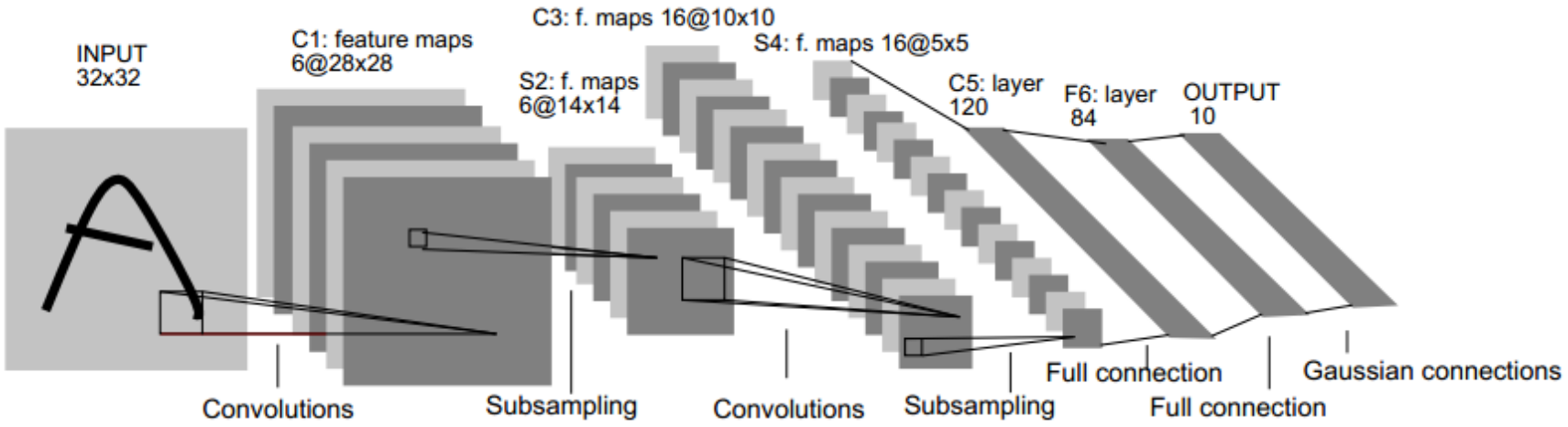
- 常用的深度学习框架
  - 简易和快速的原型设计
  - 自动梯度计算
  - 无缝CPU和GPU切换

从人工神经网络到深度神经网络

- 卷积神经网络
- 循环神经网络
- 生成对抗神经网络

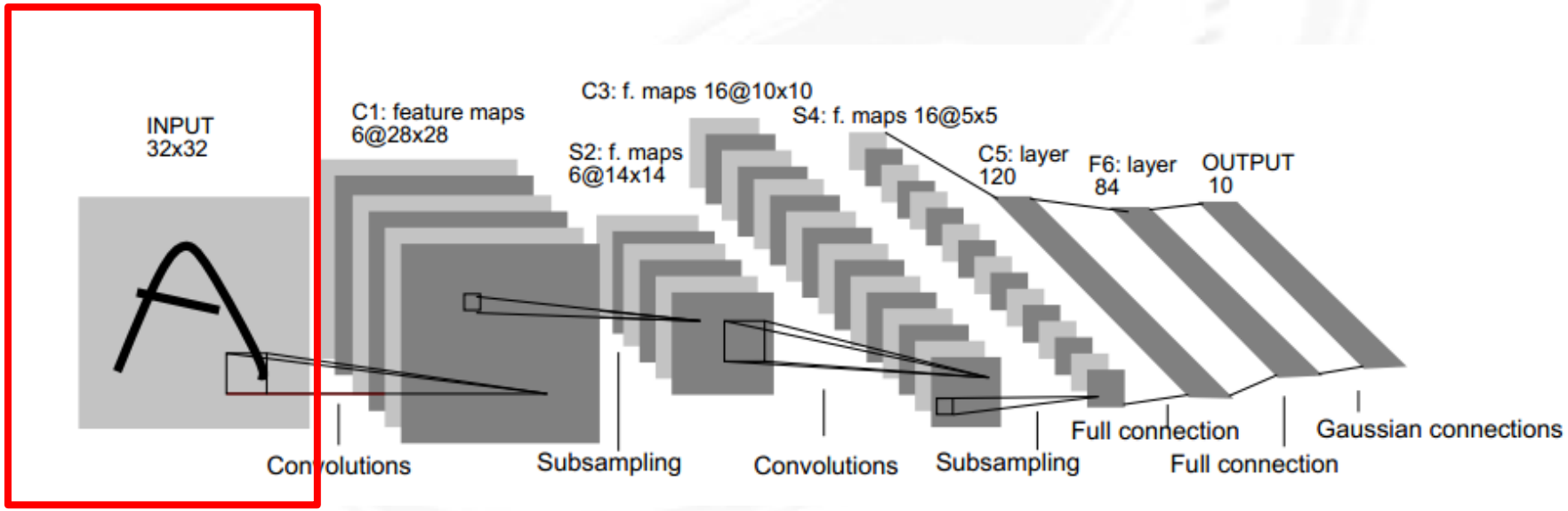
卷积神经网络

- 卷积神经网络由卷积层、池化层、全连接层交叉堆叠，加上输入层和输出层构成。
  - 趋向于小卷积、大深度
  - 趋向于全卷积



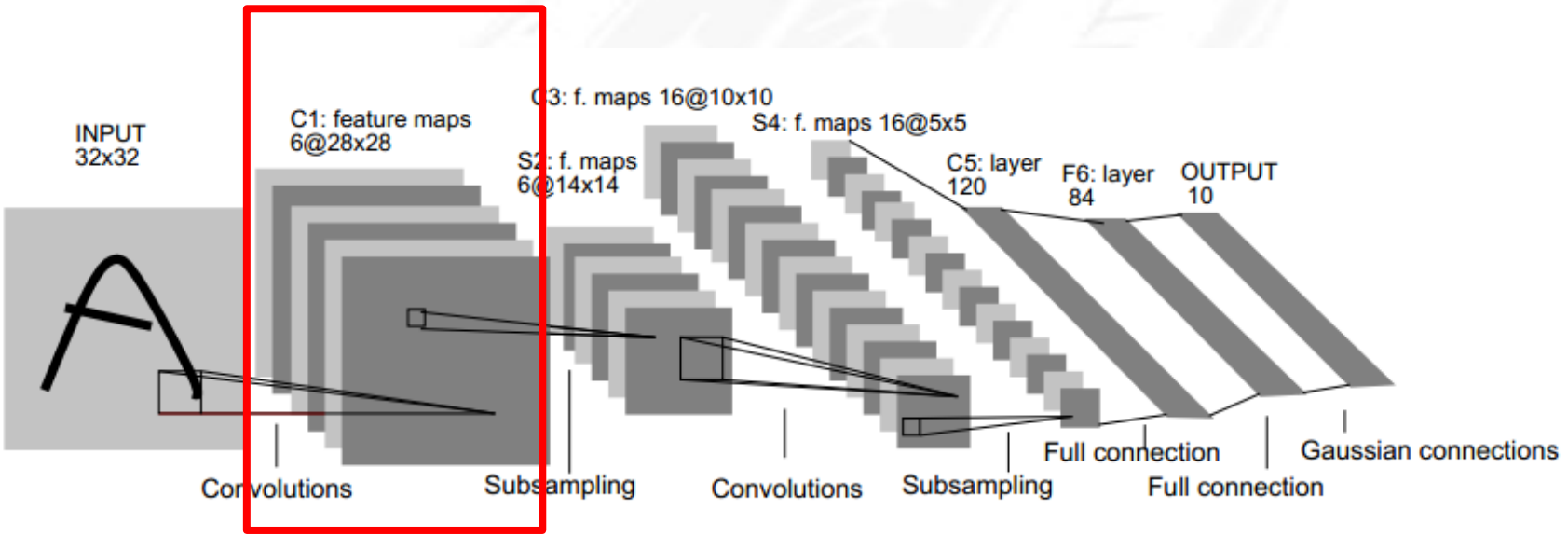
卷积神经网络

- 输入层：CNN的第一层，输入数据通常为原始图像矩阵或者向量化的文本矩阵等形式。



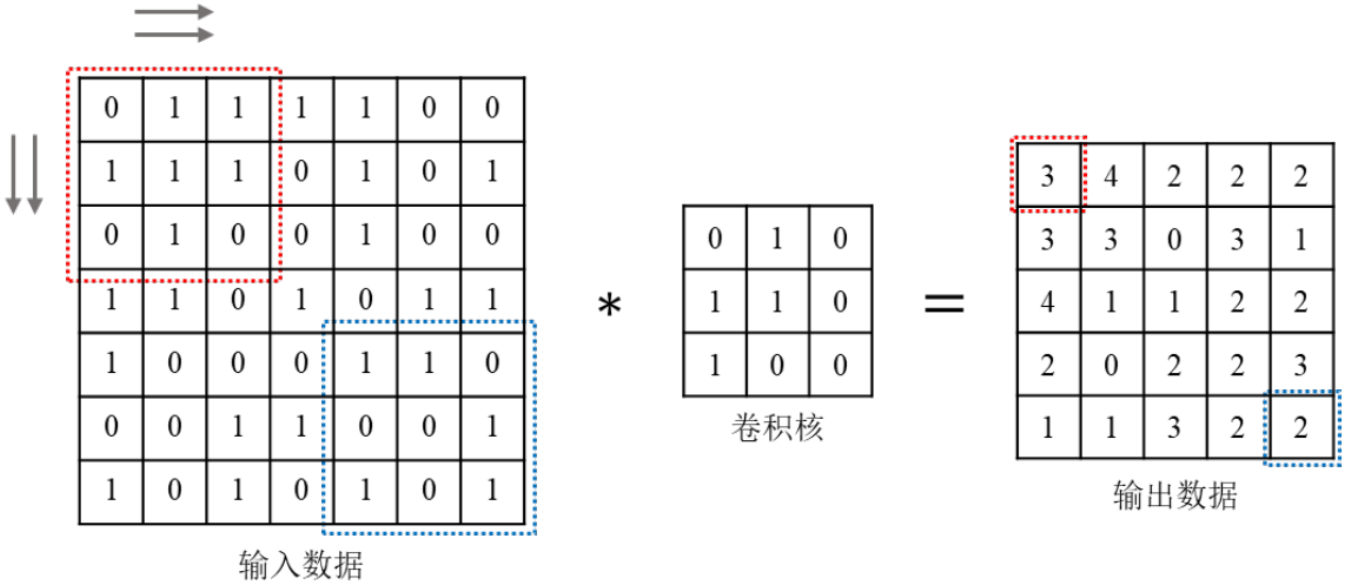
卷积神经网络

- 卷积层：CNN的核心结构，每层卷积层由若干卷积单元组成。卷积运算的目的是提取输入的不同特征，利用多层卷积从低级特征中迭代提取更复杂的特征。



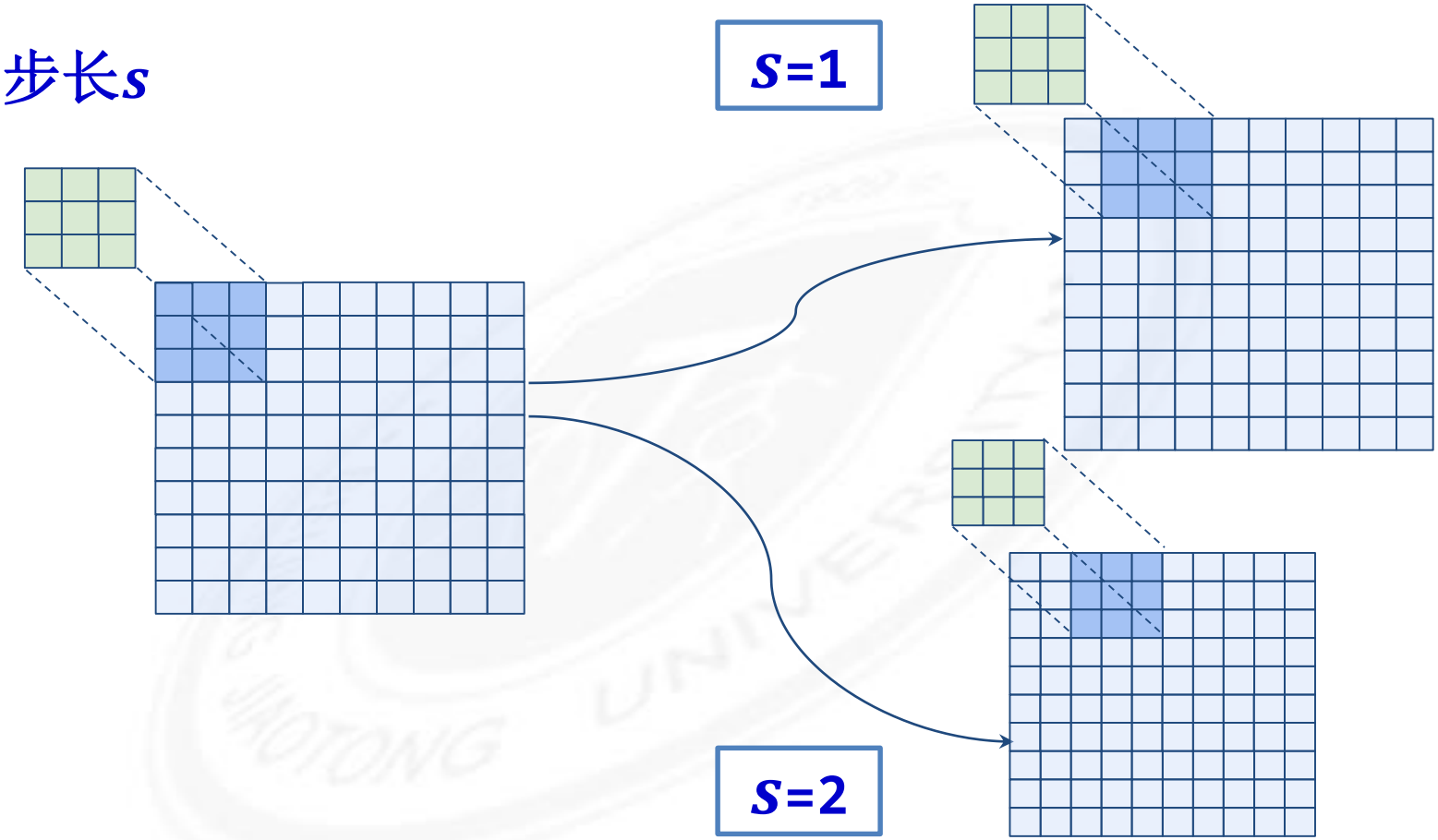
卷积神经网络

- 卷积计算利用卷积核逐步对输入数据进行区域扫描。
  - 卷积步长 s 和图像尺寸 n 控制卷积结果的尺寸。
  - 当步长为3时，水平方向剩余区域不足，利用“零填充”。

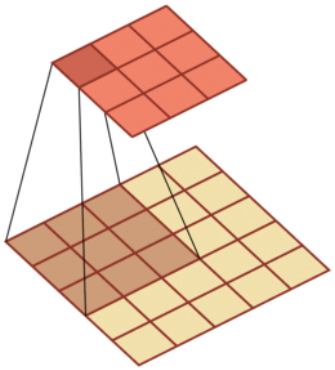


卷积神经网络

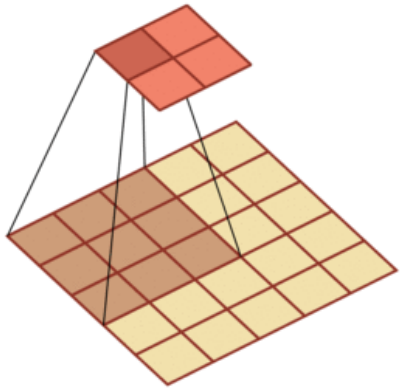
卷积步长s



卷积神经网络

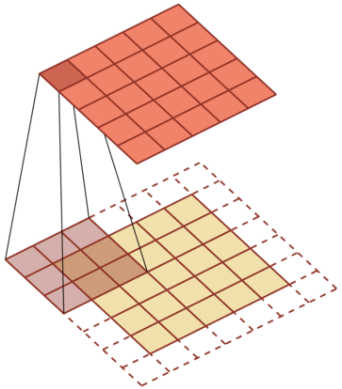


步长1，零填充0

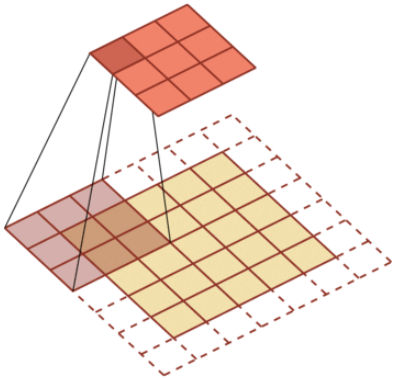


步长2，零填充0

参数共享



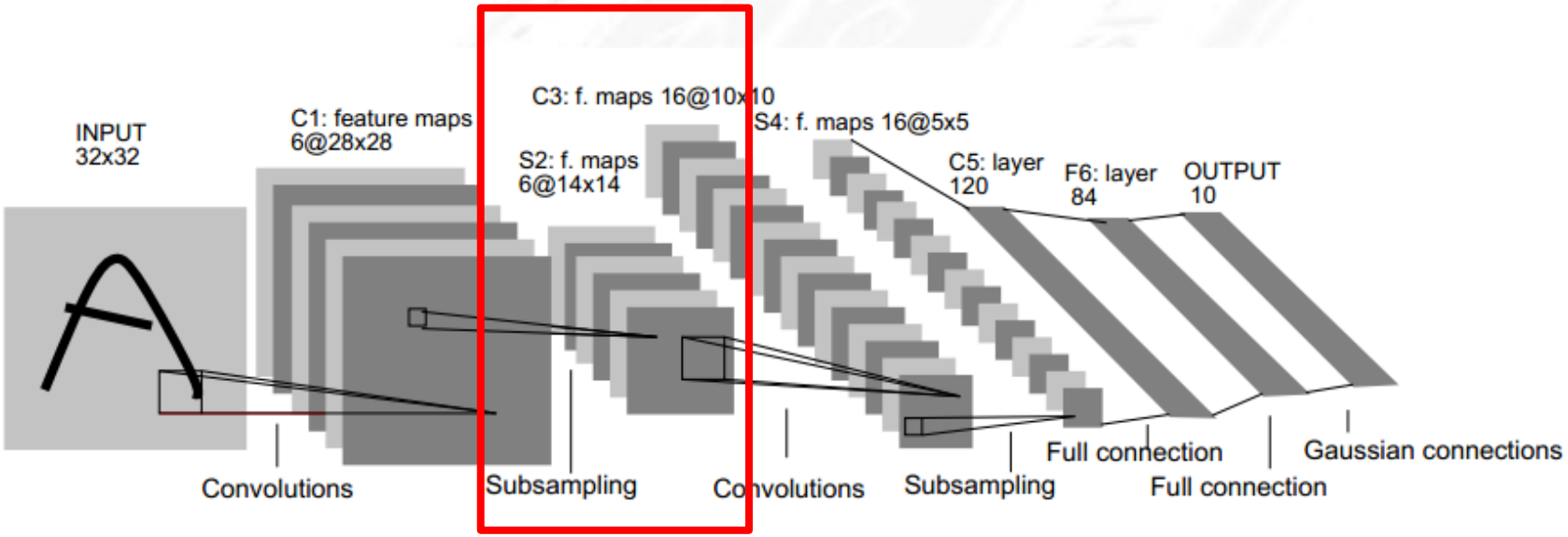
步长1，零填充1



步长2，零填充1

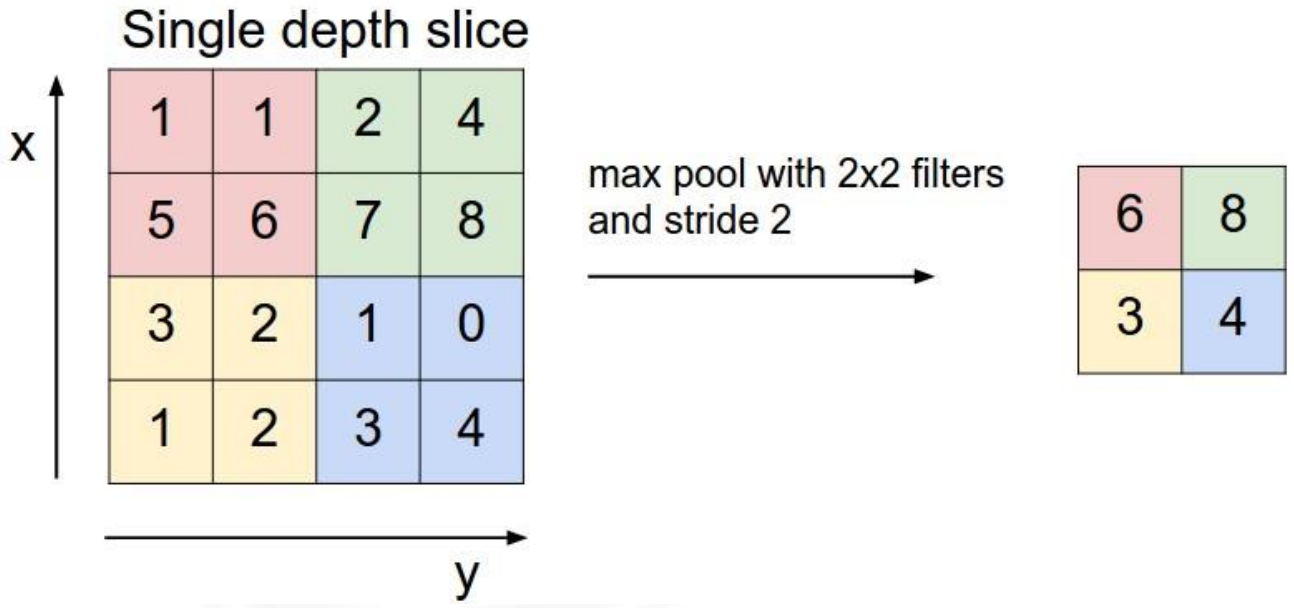
卷积神经网络

- 池化层：通常在卷积层之后，将得到维度很大的特征切成几个区域，取其最大值或平均值，得到新的维度较小的特征。
- 常用的池化方法：均值池化、最大化池化、重叠采样、均方采样、归一化采样。



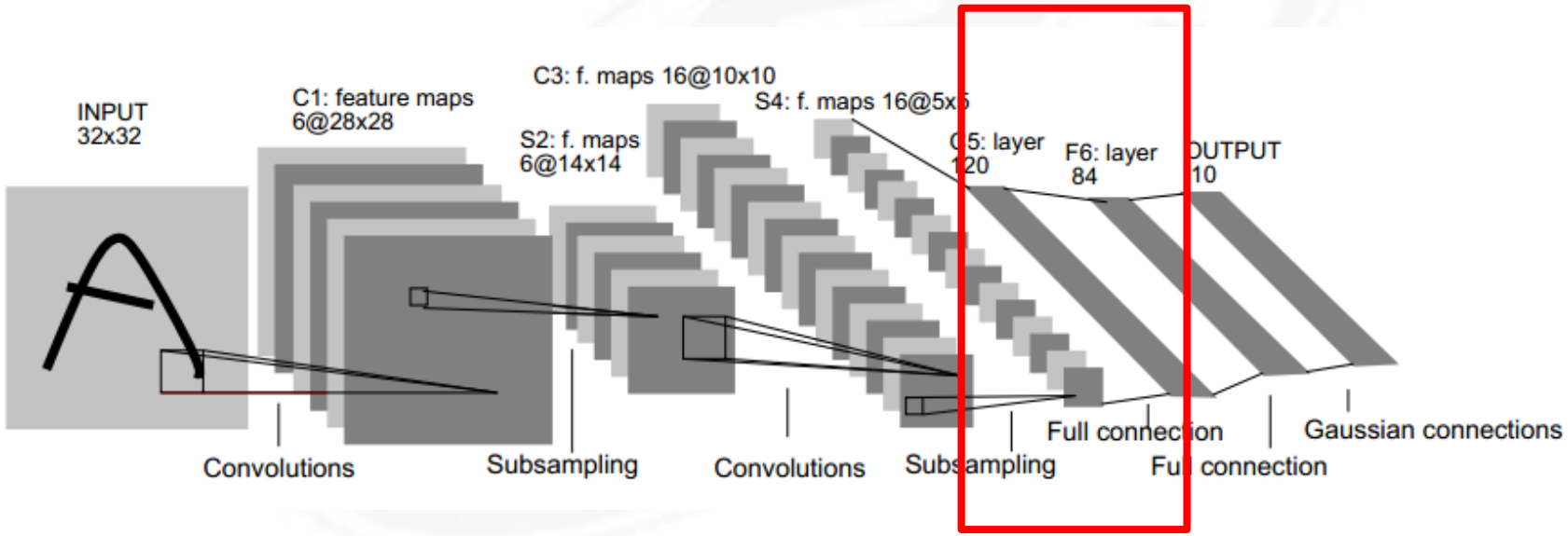
卷积神经网络

- 最大化池化(max pooling)
  - 选取扫描区域的最大值作为池化后的输出结果。



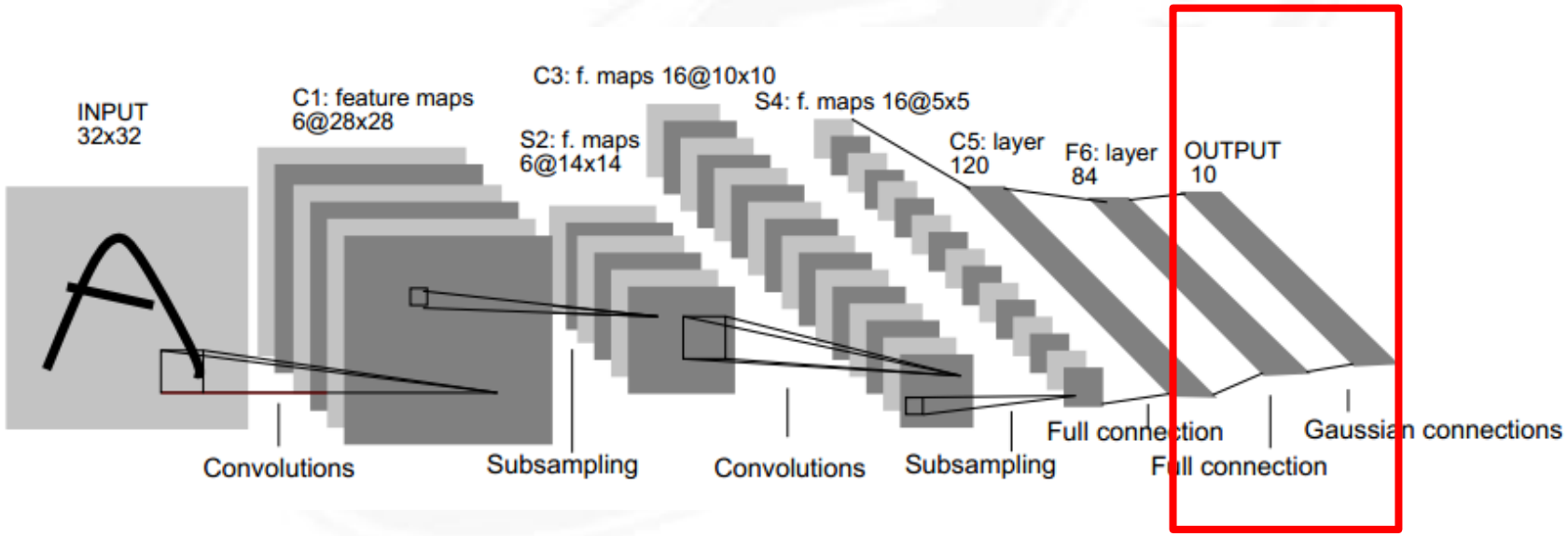
卷积神经网络

- 全连接层：将所有局部特征结合变成全局特征，用来计算最后每一类的得分。卷积层常采用**RELU**作为激活函数，全连接层通常采用**softmax**作为激活函数，进行多分类输出。



卷积神经网络

- 输出层：利用前面得到的特征计算并输出分类结果。
- 实际应用中，多个卷积层和池化层进行堆叠，一层卷积只能捕获局部特征，层数越高，得到的特征越全局化。



卷积神经网络

- 卷积神经网络的核心思想：将局部感受野、权值共享以及降采样三种结构思想结合起来，以此来达到简化网络参数并使得网络具有一定程度的位移、尺度、缩放、非线性形变稳定性。
- 局部感受野
  - 由于图像的空间联系是局部的，每个神经元无需对全部的图像做感受，只需要感受局部特征，然后在更高层将所得的不同的局部神经元综合起来得到全局信息，减少了连接的数目。
- 权重共享
  - 权值共享是对图像用同样的卷积核进行卷积操作，第一个隐藏层的所有神经元能检测到处于图像不同位置的完全相同的特征，能够适应图像的小范围的平移性，具有较好的平移不变性。

循环神经网络

- 循环神经网络（recurrent neural network, RNN）
  - RNN是一种人工神经网络，除不同层之间的连接，同层之间的神经元之间连接构成了一个时间序列。
  - RNN利用带自反馈的神经元，能够处理任意长度的序列。
  - 循环神经网络比前馈神经网络更加符合生物神经网络的结构。
  - 循环神经网络已被广泛应用于语音识别、语言模型以及自然语言生成等任务。

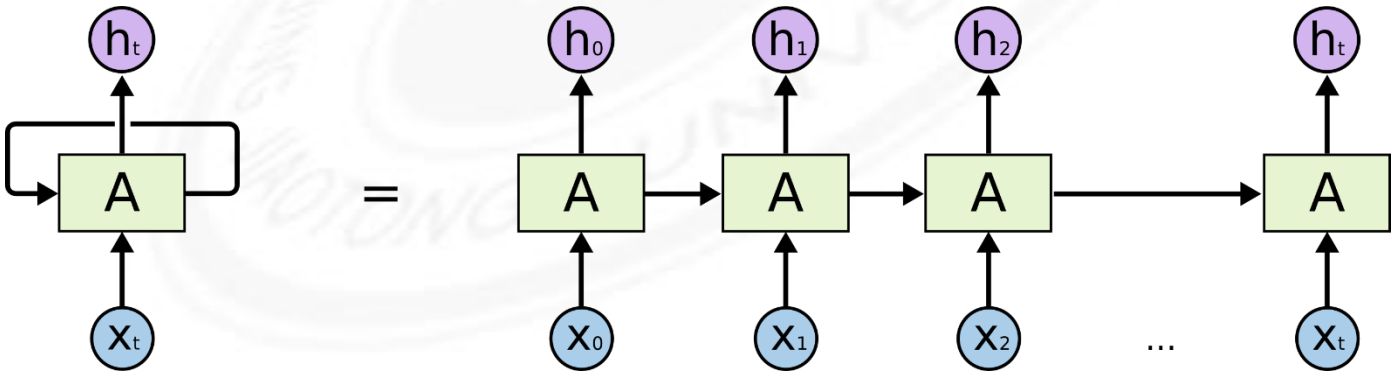
循环神经网络

- RNN与传统神经网络的不同之处在于其允许对向量的序列进行操作，输入输出都可为序列形式。

- 由于一个序列当前的输出与前面的输出也有关，RNN需要有记忆特性，具体表现为RNN会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再是无连接的而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。

循环神经网络

- RNN与传统神经网络的不同之处在于其允许对向量的序列进行操作，输入输出都可为序列形式。
- 由于一个序列当前的输出与前面的输出也有关，RNN需要有记忆特性，具体表现为RNN会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再是无连接的而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。



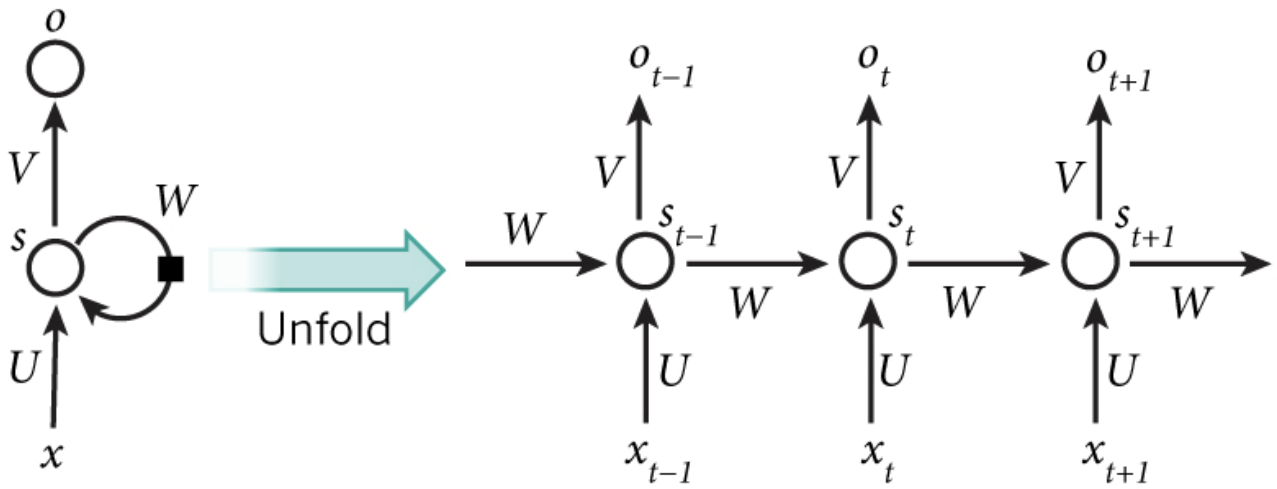
循环神经网络

- x_t 表示 t 时刻的输入，该时间序列的长度为 T；
- s_t是 t 时刻的隐藏层状态，基于上一时刻的隐藏层状态 s_{t-1} 和当前的输入 x_t 计算得到。$f(\cdot)$为非线性的激活函数。

$$s_t = f(U x_t + W s_{t-1})$$

- o_t 表示 t 时刻的输出，计算公式如下：

$$o_t = softmax(V s_t)$$

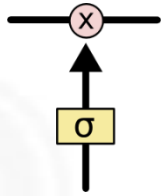


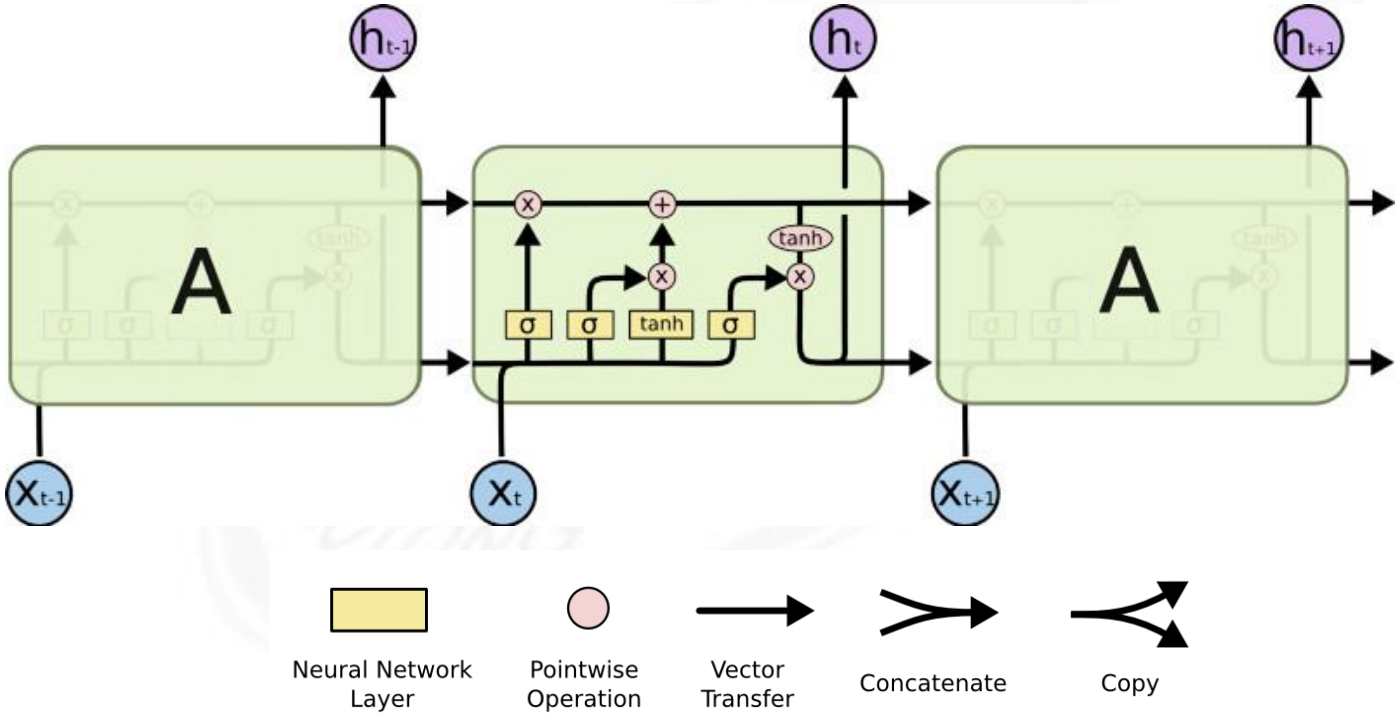
循环神经网络

- **LSTM**
  - 长短期记忆网络（**Long-Short Time Memory，LSTM**）是一个基于RNN进行扩展的循环神经网络。
  - 解决问题：传统的RNN模型在反向传播过程中容易出现梯度消失和梯度爆炸，不能够很好地实现参数训练。
  - LSTM引入“细胞状态”（cell state），在序列的整个处理过程中存储相关信息，使得后面的时间步也能利用来自较早时间步的信息，减少了短期记忆的影响。

循环神经网络

- LSTM由遗忘门、输入门和输出门构成，利用三个门控单元控制序列中的信息流。
  - sigmoid层输出0-1之间的数字，确定每个组件应通过的量。
  - 右图中的粉色门代表逐点乘法。



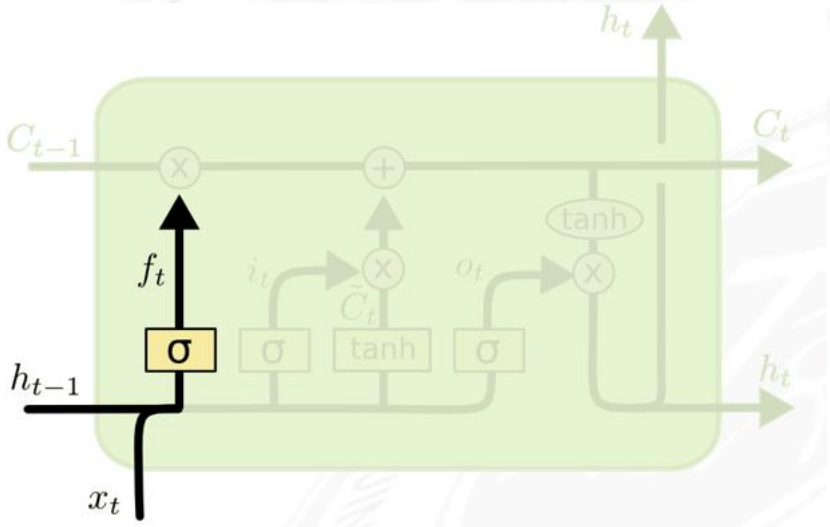


循环神经网络

- **LSTM——遗忘门（forget gate）**
  - 遗忘门用于判断细胞状态中应当丢弃的信息。

$$f_t = \sigma(W_f h_{t-1} + U_f x_t + b_f)$$

  - f_t 中每个值的范围为 [0, 1]，值越接近1表明细胞状态 c_{t-1}中对应位置的值应该被记住，值越接近0表明对应位置的值更应该被遗忘。



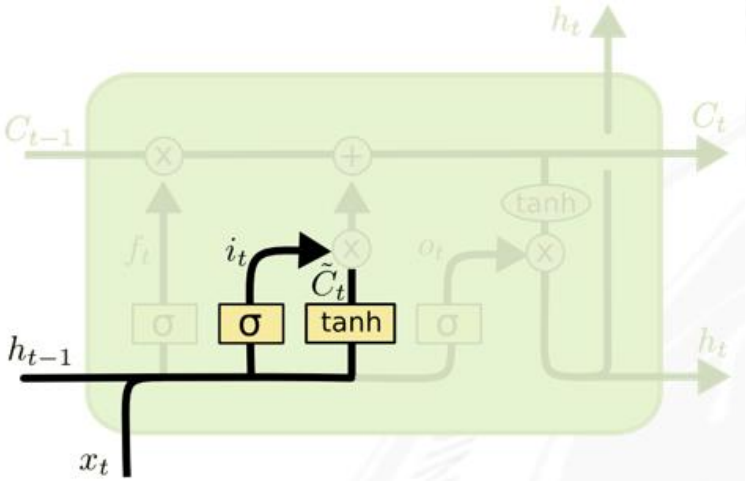
循环神经网络

- **LSTM——输入门（input gate）**
  - 输入门用于判断应加入细胞状态中的信息。

$$i_t = \sigma(W_i h_{t-1} + U_i x_t + b_i)$$

  - 候选的细胞状态 $\tilde{C}_t$ 提供更新的输入信息。

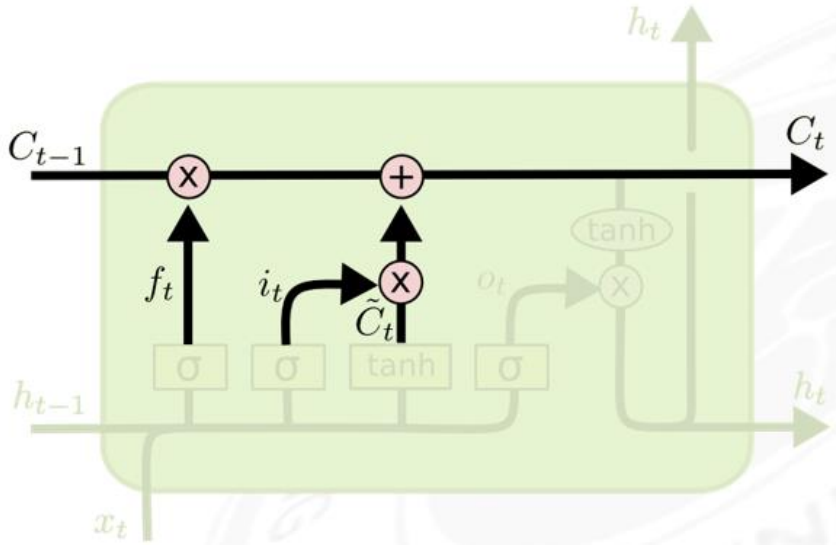
$$\tilde{C}_t = \tanh(W_C h_{t-1} + U_C x_t + b_C)$$



循环神经网络

- **LSTM——细胞状态更新（input gate）**
  - 利用遗忘门和输入门可更新得到 t 时刻的细胞状态。

$$C_t = f_t \odot C_{t-1} + i_t \odot \tilde{C}_t$$



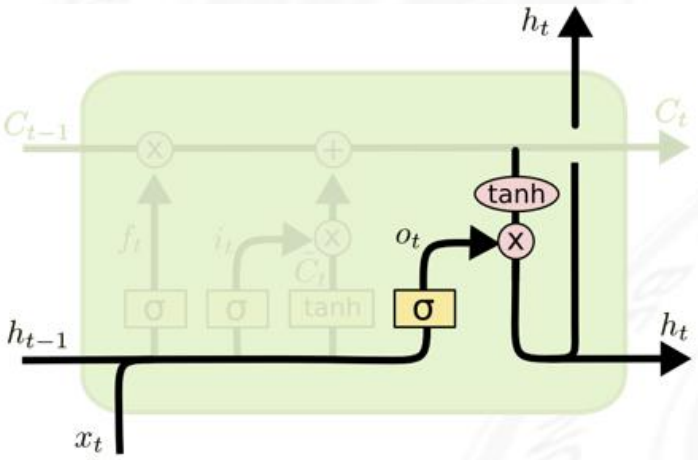
循环神经网络

- **LSTM——输出门（output gate）**
  - 输入门用于控制应从细胞状态输出的信息。

$$o_t = \sigma(W_o h_{t-1} + U_o x_t + b_o)$$

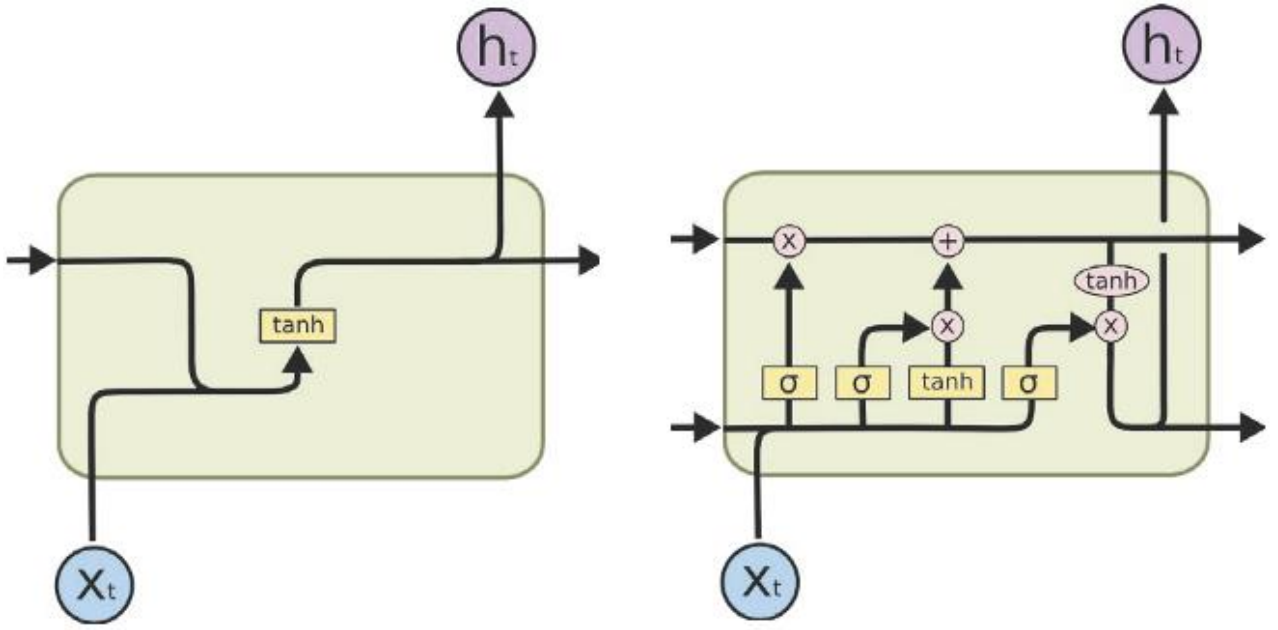
  - 利用输入门得到更新后的 t 时刻的隐藏层状态。

$$h_t = o_t \odot \tanh(C_t)$$



RNN VS LSTM

- LSTM实现长期记忆，在考虑最近时刻的状态的同时允许模型获取之前的长期状态。



(a) RNN　　(b) LSTM

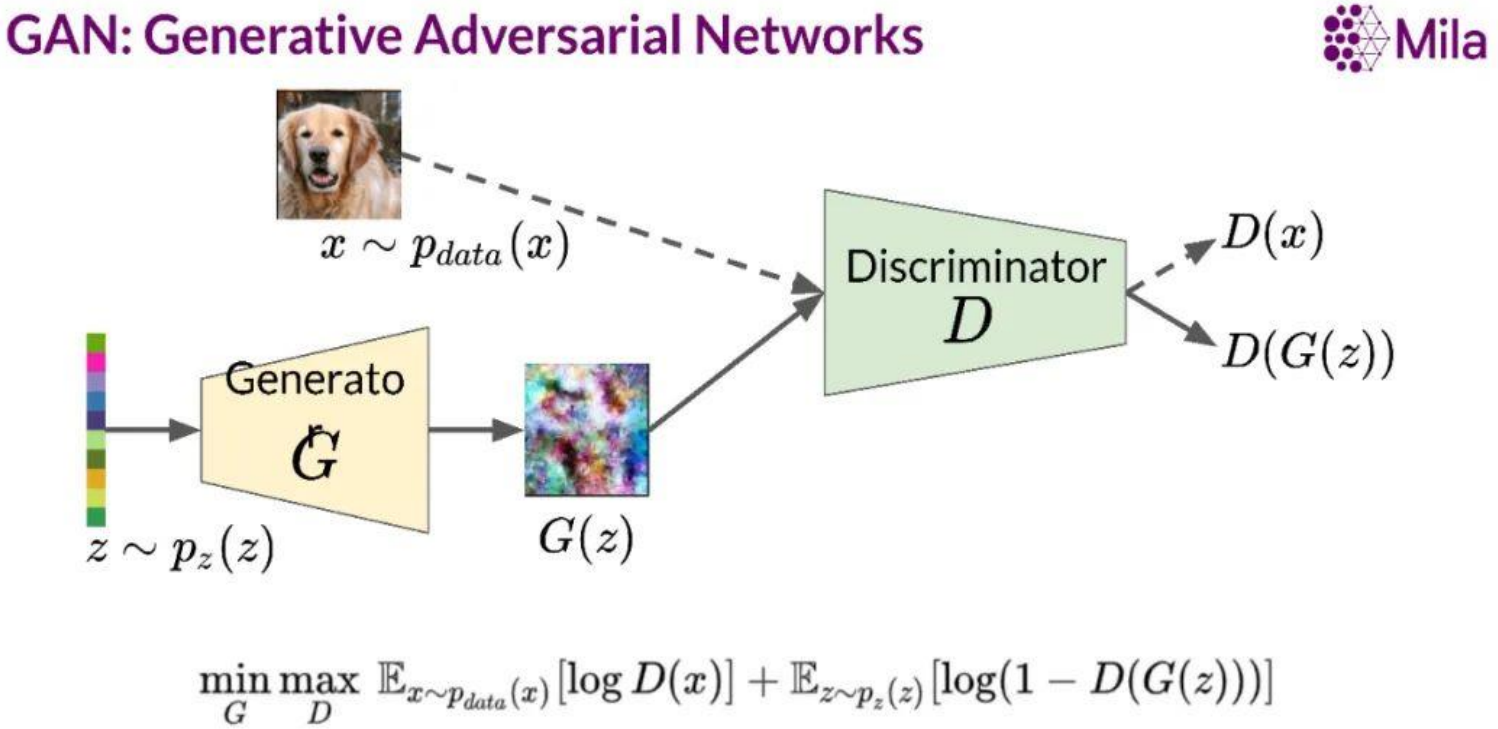
生成对抗网络

- 生成对抗网络(Generative Adversarial Networks, GAN)
  - 生成对抗网络由一个生成器与一个判别器组成。
  - 生成器从潜在空间中随机采样作为输入，其输出结果需尽量模仿训练集中的真实样本。
  - 判别器的输入为真实样本或生成器的输出，其目的是将生成器的输出从真实样本中尽可能分辨出来。

生成对抗网络

- 生成对抗网络(GAN)
  - 生成器 G 作为一个生成图片的网络,接收一个随机的噪声z,利用该噪声生成图片,记作G(z)。
  - 判别器 D 作为一个判别网络,判别一张图片是否“真实”。输入参数为 x, x代表一张图片,输出D(x)代表x为真实图片的概率。若D(x)为1,表示100%是真实的图片,输出为0则代表其不是真实图片。
  - 在训练过程中,G的目标是尽量生成真实的图片去欺骗D, 而D的目标是尽量把G生成的图片和真实的图片区分开。G和D构成了一个动态的“博弈过程”。

生成对抗网络



- 其中，G代表生成器，D代表判别器，训练时分别对D和G进行交互迭代，固定G，优化D，一段时间后，固定D再优化G，直到模型收敛。

生成对抗网络—训练过程

算法 13.1: 生成对抗网络的训练过程

输入: 训练集 $\mathcal{D}$，对抗训练迭代次数 T，每次判别网络的训练迭代次数 K，小批量样本数量 M

随机初始化 θ, ϕ;

for $t \leftarrow 1$ **to** T **do**

  // 训练判别网络 $D(\mathbf{x}, \phi)$

  for $k \leftarrow 1$ **to** K **do**

    // 采集小批量训练样本

    从训练集 $\mathcal{D}$ 中采集 M 个样本 $\{\mathbf{x}^{(m)}\}, 1 \leq m \leq M$;

    从分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采集 M 个样本 $\{\mathbf{z}^{(m)}\}, 1 \leq m \leq M$;

    使用随机梯度上升更新 ϕ，梯度为

$$\frac{\partial}{\partial \phi}\left[\frac{1}{M}\sum_{m=1}^{M}\left(\log D(\mathbf{x}^{(m)}, \phi) + \log\left(1 - D\left(G(\mathbf{z}^{(m)}, \theta), \phi\right)\right)\right)\right];$$

  end

  // 训练生成网络 $G(\mathbf{z}, \theta)$

  从分布 $\mathcal{N}(\mathbf{0}, \mathbf{I})$ 中采集 M 个样本 $\{\mathbf{z}^{(m)}\}, 1 \leq m \leq M$;

  使用随机梯度上升更新 θ，梯度为

$$\frac{\partial}{\partial \theta}\left[\frac{1}{M}\sum_{m=1}^{M} D\left(G(\mathbf{z}^{(m)}, \theta), \phi\right)\right];$$

end

输出: 生成网络 $G(\mathbf{z}, \theta)$