

# 深度学习笔记

陈鸿峥

2020.03\*

## 目录

1 简介	1
2 人工神经网络基础	3
2.1 神经元与多层神经网络	3
2.2 卷积	4
2.3 池化	5

本课程主要选用Ian Goodfellow, Yoshua Bengio, Aaron Courville的《深度学习》(Deep Learning)一书。一些众所周知的概念会在笔记中略过，而着重在一些容易忽略的知识点上面。

## 1 简介

图 1反映了**深度学习**与其他几个常见概念之间的关系。传统的**机器学习**（如决策树、SVM、随机森林等）常需要人工提取特征，这一步经常涉及到**特征工程**(feature engineering)，如果特征没有进行一定处理，直接丢进去让其学习，往往会产生非常糟糕的结果。在一种表示下可能可以对数据进行线性二分，而另一种表示下则没有办法。因此，为了避免对特征的强依赖性，一种方法是利用机器学习来学习**表示(representation)**本身，再将新的表示送入到后面的学习器中让它学习**表示到输出的映射**，此即**表示学习**。再到后来，深度学习则更加将这种思想发扬光大，表示学习只能学习到**浅层简单的特征**，那深度学习则尝试去学习**深层复杂的特征**。

事实上现在**图神经网络(GNN)**也是遵循这样的发展过程，最开始尝试在图上做机器学习[1, 2, 3]；然后又开始在图上以各种随机游走的方式做图表示学习-图嵌入(embedding)[4, 5]；后来发现图嵌入能够获得的特征依然太浅层了，因此现在更多则采用图神经网络[6, 7, 8, 9]的方式来做图相关的工作。

---

\*Build 20200323

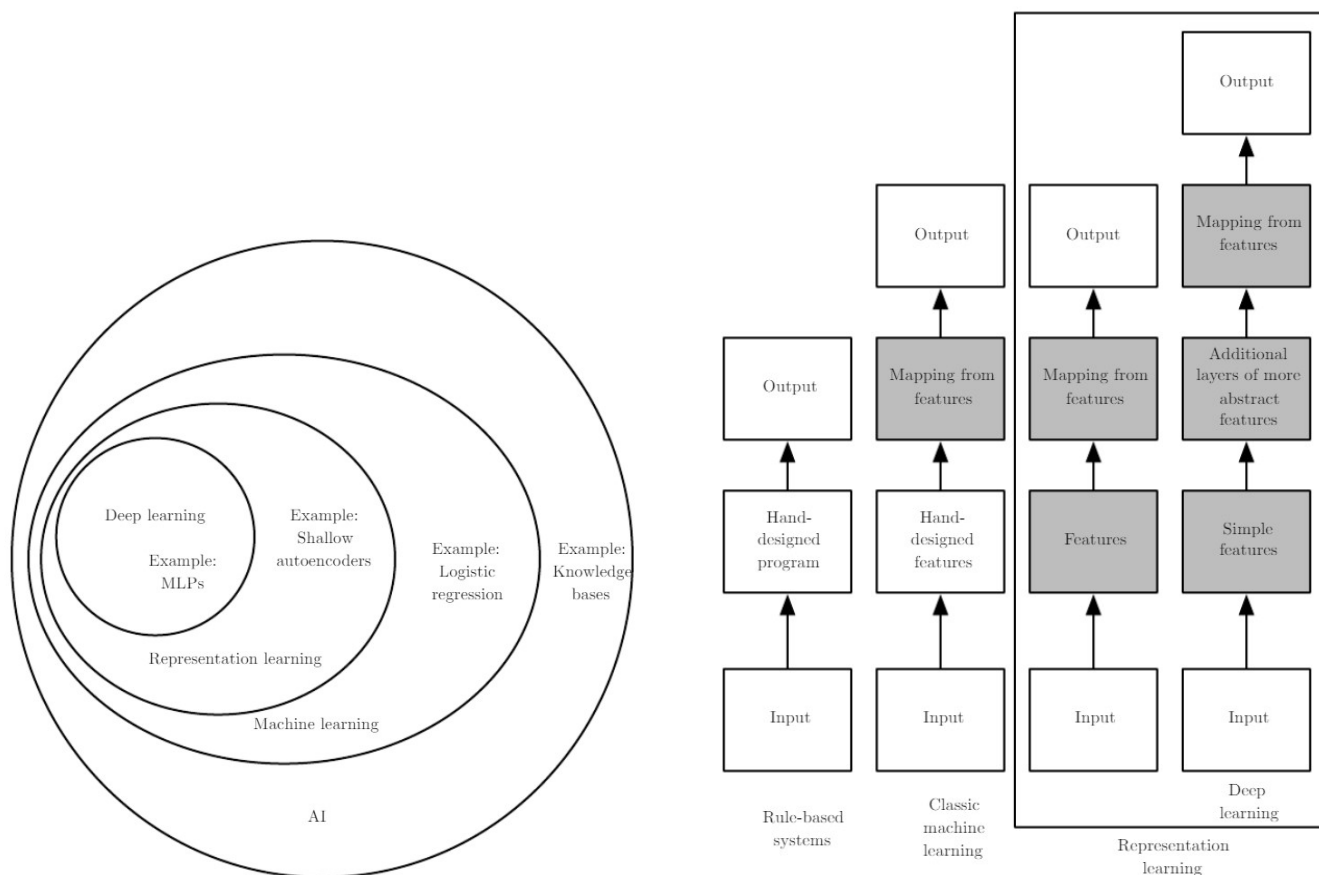


图 1: 深度学习Venn图

深度学习在发展过程中也起过几个名字：在1940年代到1960年代被称为**控制论**(cybernetics)，之后1980到1990年代则被称为**连接主义**(connectionism)，而后从2006年到现在才被称为**深度学习**。2018年的图灵奖正式颁发给深度学习三巨头—Geoffrey Hinton, Yoshua Bengio和Yann LeCun，也奠定了深度学习在学术界的地位。

我也一直在思考是什么造成了深度学习在2010年代的兴起，使得如今我们快速进入软件2.0时代[10]。总结来讲有以下几点：

- **数据：**我们每天产生的数据量越来越多，能够处理的数据量越来越大，名副其实地进入了大数据时代。其中的一些数据经过处理能够变得很干净，2010年拥有大量标注图片的数据集ImageNet就是这样的例子，它的提出使得有监督学习迎来了一波兴起。
- **算法：**我们有更多更优秀的模型，从AlexNet的dropout，再到后来ResNet的残差模块。ImageNet的发展历程，也是深度学习算法/模型改进和发展的历程。
- **软件系统：**早年的深度学习框架Caffe、Theano、MXNet等使得研究人员可以方便地编写神经网络模型，而2015年前后诞生的TensorFlow和PyTorch则是成为了现在深度学习框架的主流范式，程序员不需知道底层的实施细节，只用“调库”和“调参”也可以实现很高效的模型，这些软件系统的诞生极大程度推动了深度学习的遍地开花。再到现在XLA和TVM等深度学习编译器的出现，更是进一步解放程序员，使上层模型可以方便快捷地部署到后端不同硬件平台上。
- **硬件：**GPU为深度学习做出了不可磨灭的贡献，没有GPGPU的发展，很多深度学习任务根本没有办法完成。现在的TPU及各种神经网络加速器也都是在拓宽这一层面，以进一步提升深度学习的能力。

上面提到的这几点都不可或缺，它们共同造就了整个深度学习栈，从而带来现在深度学习的繁荣。

## 2 人工神经网络基础

### 2.1 神经元与多层神经网络

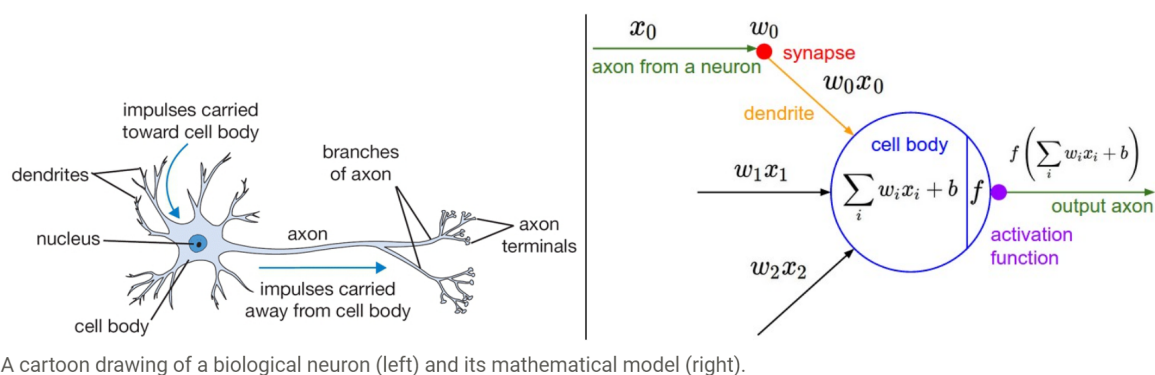


图 2: 神经元(neuron)

常见的激活函数：

- Sigmoid:  $g(x) = \frac{1}{1+e^{-x}}$
- Tanh:  $g(x) = \tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$

- ReLU:  $g(x) = \max(0, x)$

之所以要采用非线性激活函数，是因为它可以使神经网络也变成非线性的，进而捕获到更加复杂的特征。

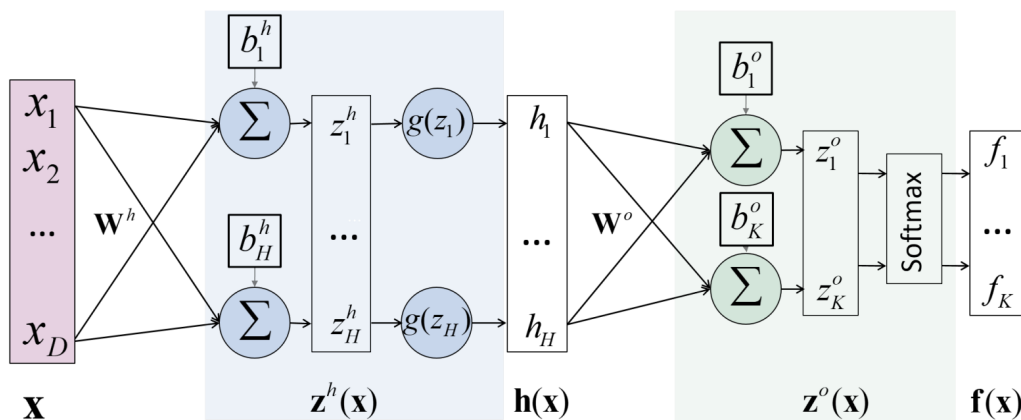


图 3: 多层神经网络(MLP)

但全连接层一个问题在于参数量巨大，当网络变大时计算量将爆炸，故要采用更加好的方法，既能提取出特征，同时计算量也能维持在一个合理的程度，因此就有了**卷积**(convolution)层。

## 2.2 卷积

$f$ 为原图， $g$ 为**卷积核**(kernel)或**滤波器**(filter)

$$(f * g)[i, j] = \sum_{m=-M}^M \sum_{n=-N}^N f[i - m, j - n]g[m, n]$$

即对应元素相乘后相加（但注意卷积与相关操作的不同，**卷积要先取反**）。卷积在边界时可用0填充(padding)。

上面的公式只是2维卷积，现实我们训练的图片通常采用4维张量表示，即NCHW格式

$$\text{n\_samples} \times \text{n\_channels} \times \text{height} \times \text{width}$$

因此对于每一张多通道的图片，卷积核也应表示为3维，且确保卷积核通道数与图片的通道数相同，这样就可以在3维空间做卷积，但出来的图片只剩1个通道了。故可以采用 **$k$ 个权重不同的卷积核分别**对图片做卷积，那么就可以提取到 **$k$ 种不同的特征**，出来的特征图(feature map)也会有 **$k$ 个通道**。

在具体实施中，卷积层也是可训练的，卷积核的权重和偏置就可以通过网络学习得到。

卷积具有以下三个特征：

- **稀疏交互**(sparse interactions): 卷积层不像全连接层，是对整个图像进行权重计算，它只选择与卷积层交叠的部分进行计算。参数少了，自然计算量也少了。
- **参数共享**(parameter sharing): 由于卷积核是滑过整个图片进行计算，因此对于卷积核的参数，都是被**多次**运用在不同位置的；而传统的全连接层的参数在图片的每一个位置只会被使用**一次**，因此

捕获细节特征的能力也会相对弱一些。

- **平移等变(equivariant representation)**: 哪怕图片进行一定的平移变换, 卷积依然有办法将对应特征提取出来。

在图卷积神经网络(GCN)[6]中的卷积也是类似的道理, 在一层中的所有图结点采用**相同**的神经网络进行聚集(aggregate)和更新(update)计算, 这样子可以确保神经网络的参数共享, 从而关注到图中的局部特征。

## 2.3 池化

普通的小卷积只能捕获到低层的特征, 想要获得高层的语义特征则卷积核应该有更大的**感受野**(receptive field), 但大卷积又会使图片的细节部分被忽略。为解决这两者之间的矛盾, 可以在**缩小后的特征图做高层次的卷积**。那么问题就变成了怎么缩小特征图, 常见有两种方法:

- 改变步长(stride)
- **池化(pooling)/降采样(subsampling)**: 对卷积核覆盖的范围取最大值或平均

池化可以使得表示对于输入的微小变换保持近似不变(invariant), 这也是好理解的, 因为池化考虑的是一片区域的整体特征, 对于取最大值或是取平均来说, 当局部区域一并改变时, 池化后的结果确实不会有太多变化。

看到这好像也就能明白经典的LeNet为什么这么设计了。先做卷积是为了提取图片中的小的/低层细节特征, 然后做激活使得网络非线性, 接着做池化, 则是将图片进行缩放, 方便后续的卷积提取出更大/更高层的语义特征。

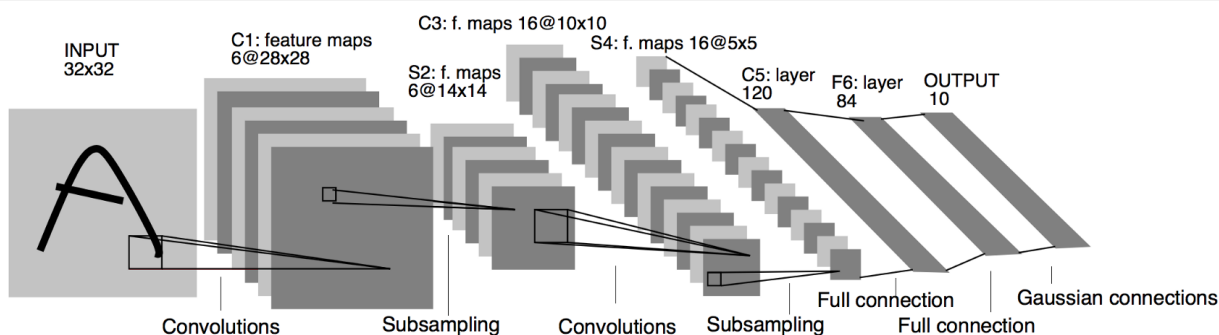


图 4: LeNet5

## 参考文献

- [1] Limin Yao, David Mimno, and Andrew McCallum. Efficient methods for topic model inference on streaming document collections. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2009.
- [2] Mu Li, David G Andersen, and Alexander Smola. Distributed delayed proximal gradient methods. In *NIPS Workshop on Optimization for Machine Learning*, 2013.
- [3] Rainer Gemulla, Erik Nijkamp, Peter J. Haas, and Yannis Sismanis. Large-scale matrix factorization with distributed stochastic gradient descent. In *Proceedings of the 17th ACM SIGKDD international conference on Knowledge discovery and data mining*, 2011.
- [4] Bryan Perozzi, Rami Al-Rfou, and Steven Skiena. DeepWalk: Online learning of social representations. 2014.
- [5] Aditya Grover and Jure Leskovec. node2vec: Scalable feature learning for networks. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining - KDD '16*, 2016.
- [6] Thomas N. Kipf and Max Welling. Semi-supervised classification with graph convolutional networks. In *Proceedings of the International Conference on Learning and Representation (ICLR)*, 2017.
- [7] William L. Hamilton, Rex Ying, and Jure Leskovec. Inductive representation learning on large graphs. In *Advances in neural information processing systems (NeurIPS)*, 2017.
- [8] Yujia Li, Daniel Tarlow, Marc Brockschmidt, and Richard Zemel. Gated graph sequence neural networks. In *Proceedings of the International Conference on Learning and Representation (ICLR)*, 2016.
- [9] Petar Veličković, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro Liò, and Yoshua Bengio. Graph attention networks. In *Proceedings of the International Conference on Learning and Representation (ICLR)*, 2018.
- [10] Kunle Olukotun. Designing computer systems for software 2.0. In *Keynote of the 45th International Symposium on Computer Architecture (ISCA)*, 2018.