化工应用数学 第七章 人工智能简介 讲义

计算机的历史：

最早的能够用于计算的机器是我国北宋时期发明的算盘，其只能是人工手动拨动算珠进行四则运算；

之后是约 200 年前，英国数学家巴贝奇设计了第一台能计算二次多项式的计算机器，叫做差分机，通过摇动左侧的手柄计算二次多项式，注意由于时代限制，尽管巴贝奇消耗的资金足够制造好几艘军舰，但他最终也没完成差分机的制造。图上展示的是后来依照巴贝奇的设计图纸制造的，并且计算速度很慢，计算 4^2+2 大约需要摇上十几秒；

真正奠定现代计算机理论基础的是库尔特.哥德尔，他正式提出可以把人类的全部认知归结为无数条定理，并且这些定理都可以用数学的模式进行表示和逻辑推导；

冯.诺依曼被称为现代计算机之父，他设计了经典的冯.诺依曼结构，就是将软件命令和数据素材都存在一起，整个设备由中央处理器、内存、硬盘、输入接口、输出设备组合而成，程序命令按照顺序执行，其次再考虑时间，我们现在几乎所有计算机、笔记本、智能手机都是基于冯诺依曼结构制造和运行的；

图中展示的是 1945 年制造的 ENIAC，世界上第一台通用电子计算机。由于命令仍然需要人工输入和调整，所以经常为了计算某个问题，需要专门人员拔掉或接入上千个插口；

最后这里是现代的大型超级计算机天河二号，通过网络将很多计算刀片连接而成。

人工智能历史：

1956 年夏，麦卡锡、明斯基等科学家在美国达特茅斯学院开会研讨“如何用机器模拟人的智能”，首次提出“人工智能（Artificial Intelligence，简称 AI）”这一概念，标志着人工智能学科的诞生。

人工智能是研究开发能够模拟、延伸和扩展人类智能的理论、方法、技术及应用系统的一门新的技术科学，研究目的是促使智能机器会听（语音识别、机器翻译等）、会看（图像识别、文字识别等）、会说（语音合成、人机对话等）、会思考（人机对弈、定理证明等）、会学习（机器学习、知识表示等）、会行动（机器人、自动驾驶汽车等）。

人工智能充满未知的探索道路曲折起伏。如何描述人工智能自 1956 年以来 60 余年的发展历程，学术界可谓仁者见仁、智者见智。我们将人工智能的发展历程划分为以下 6 个阶段：

一是起步发展期：1956 年—20 世纪 60 年代初。人工智能概念提出后，相继

取得了一批令人瞩目的研究成果，如机器定理证明、跳棋程序等，掀起人工智能发展的第一个高潮。

二是反思发展期：20 世纪 60 年代—70 年代初。人工智能发展初期的突破性进展大大提升了人们对人工智能的期望，人们开始尝试更具挑战性的任务，并提出了一些不切实际的研发目标。然而，接二连三的失败和预期目标的落空（例如，无法用机器证明两个连续函数之和还是连续函数、机器翻译闹出笑话等），使人工智能的发展走入低谷。

三是应用发展期：20 世纪 70 年代初—80 年代中。20 世纪 70 年代出现的专家系统模拟人类专家的知识和经验解决特定领域的问题，实现了人工智能从理论研究走向实际应用、从一般推理策略探讨转向运用专门知识的重大突破。专家系统在医疗、化学、地质等领域取得成功，推动人工智能走入应用发展的新高潮。

四是低迷发展期：20 世纪 80 年代中—90 年代中。随着人工智能的应用规模不断扩大，专家系统存在的应用领域狭窄、缺乏常识性知识、知识获取困难、推理方法单一、缺乏分布式功能、难以与现有数据库兼容等问题逐渐暴露出来。

五是稳步发展期：20 世纪 90 年代中—2010 年。由于网络技术特别是互联网技术的发展，加速了人工智能的创新研究，促使人工智能技术进一步走向实用化。1997 年国际商业机器公司（简称 IBM）深蓝超级计算机战胜了国际象棋世界冠军卡斯帕罗夫，2008 年 IBM 提出“智慧地球”的概念。以上都是这一时期的标志性事件。

六是蓬勃发展期：2011 年至今。随着大数据、云计算、互联网、物联网等信息技术的发展，泛在感知数据和图形处理器等计算平台推动以深度神经网络为代表的人工智能技术飞速发展，大幅跨越了科学与应用之间的“技术鸿沟”，诸如图像分类、语音识别、知识问答、人机对弈、无人驾驶等人工智能技术实现了从“不能用、不好用”到“可以用”的技术突破，迎来爆发式增长的新高潮。

我们上面说的这些都称作人工智能，但本门课我们主要给大家讲述近十年来迅速发展的深度神经网络。

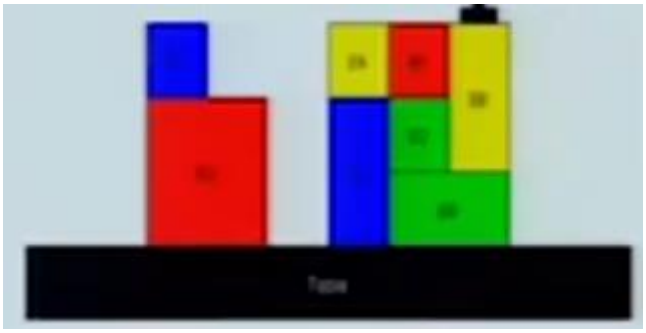
神经网络与专家系统的对比：

传统人工智能多为上述提到的专家系统，它们需要事先找相关领域的专家，针对特定问题，首先进行相关的特征工程，抽取出专业知识及相关领域的常识性知识，采用推理等方式，按照事先编辑好的程序解决相关问题。这些要求大大的限制了专家系统使用范围，使得其虽然在不少领域能够取得很好的效果，但是在一些复杂的领域，专家系统往往不能够胜任。

相比于专家系统，近年来崛起的神经网络不需要事先的详细的专业知识，对于特征工程等的需求也不高，所以其在图像识别、翻译、智能控制等领域获得了很大的成功。

专家系统案例介绍：

（1）码头集装箱摆放



如图所示，有不同的集装箱，现在需要使用 ai 完成任意将某一箱子放到另一箱子上方的任务，基本思想就是将这一任务分解为：找到可放箱子的位置、将下方的箱子放到空位上、将上方的箱子放到下方箱子上方；在这一过程中，我们移动某一箱子的时候，还需要首先进行将其上方清空操作，这里就又涉及到需要将其上方箱子移动到其他位置。。。

（2）动物识别

与我们后面要将的使用神经网络的图像识别技术不同，使用专家系统的动物识别需要我们输入相关动物的一些具体的信息，比如是否有毛发、有几只脚、牙齿形状等，具体的识别过程在程序中是通过这些具体的信息再一步步的去分类，最终识别到是什么动物

（3）字符积分

专家系统进行字符积分的过程实际上与我们人工积分的过程类似。首先，将积分中可能涉及的操作分为了三类：第一类是我们已知结果的一些积分，比如对指数函数、三角函数等；第二类是一些“安全”的转换，包括 cf(x)、f(x)+g(x)等，通过这些操作，我们是一定能将积分进行简化的；最后一种是一些可能能够简化问题的操作，包括三角函数可使用 tan 函数进行化简、遇到 1-x2 时可尝试使用三角函数进行换元等，这些操作可以在遇到的时候分别进行尝试，看是否能够化简问题。最终，通过上述三种方式，在 32k 内存的计算机上，我们可以解决大部分的积分问题。

（4）深蓝（选讲）

97 年击败人类国际象棋冠军的深蓝其实还是在传统人工智能范畴，也可以说是一种专家系统。其主要使用了“最大/最小+αβ+剪枝技术+并行+开局/残局+。。。”等不同的技术，但深蓝还是基于现有知识进行的应编程，本质来说就是针对不同的策略，进行最优解搜索的一种算法，其本身对于棋局并不能理解，可以说深蓝只是在计算或搜索数学上的最优解，其本身并不会下国际象棋。这一点，和我们后续会讲到的 alpha Go 有着很大的区别。

计算机视觉（CV）简单介绍：

深度神经网络有很长的历史、也有很多的分支，我们从计算机视觉（CV）方面为插入点来介绍深度神经网络。计算机视觉简单来说就是要赋予计算机自然视觉能力，让它像是我们一样能够看到东西。

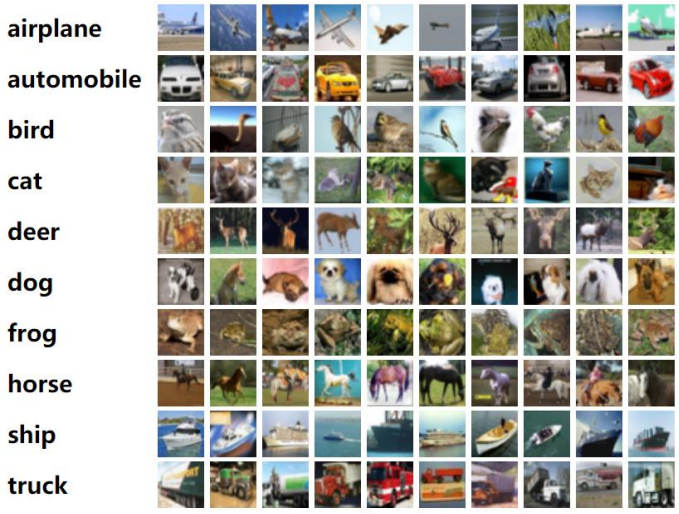
地球上的生命诞生于约 35 到 46 亿年前，而生物的视觉产生的却要晚很多，但生物的视觉是很神奇的，有一种说法：在视觉产生之前的几十亿年里，生物很简单，漂浮着在海洋里，等待食物漂过嘴边，进化十分缓慢；正是因为生物进化出了视觉，5 亿 4 千万年前生物的进化猛然加速、物种开始大爆发，进而一步步的产生了更加复杂的高级生物。

而人类在赋予机器视觉方面也做了很多的努力，从最早的文艺复兴时期的暗箱开始，人类给与了机器记录下视觉的能力，但是这离机器真正有视觉还很远。而真正的计算机视觉的起源可以追溯到 1966 年，当时麻省理工的学者们发起了一个项目：The summer vision project，他们计划用一个暑假的时间来完全解决计算机的视觉问题，当然最终是失败了，因为哪怕到今天计算机视觉方面的研究离完全解决还有很远的路程，但这却是人类对计算机视觉发起研究的开始，这之后计算机视觉也逐步迅速发展。

这里我们不去详述计算机视觉的发展历史，我们更关注的是深度神经网络，计算机视觉在本门课程里只是作为一个切入点。

图像分类-线性分类：

图像分类即按图片内容的不同给图片分别贴上不同的标签，表明图片中出现了哪一种事物。图中列出的是 CIFAR-10 数据集，该数据集共有 60000 张彩色图像，这些图像是 32*32，分为 10 个类，每类 6000 张图，这些数据可以用作之后神经网络的训练和测试使用，其在现在的 CV 领域已经算是规模比较小的数据集了。而图片分类的问题就变成了：在已知一些分类图片（数据集）之后，新得到一张图片，应该把新图片放到哪个分类中？

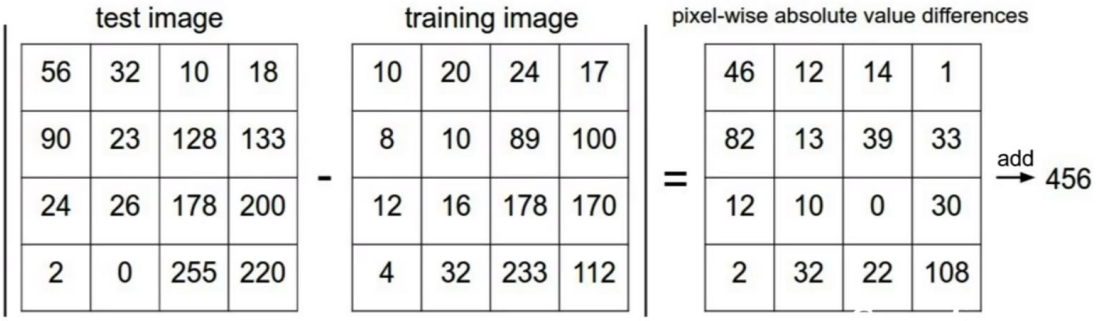


思考：计算机看到的图像是什么样的？怎么用计算机去分类图片？

图像在计算机中的现实和存储是按像素来进行的，对于黑白图片，每个像素上的灰度值被用 0~255 的一个整数代表，从而一张图片就是这样的整数矩阵；而彩色图片则是分别存储了其 R、G、B 三原色的三个通道的数据。

为简单起见，我们现在不考虑三通道的问题，而是认为一个图片就是一个数据矩阵。那么数据分类的最直接的一个思路就是一个像素、一个像素的去对比，类似于我们玩的大家来找茬的游戏那样。

如下图所示那样，我们逐个像素的将同样位置的数值相减，然后将绝对值相加，将相加的结果作为要测图片和我们已知图片的误差，最终将和我们所测图片误差最小的那个种类作为所测图片的种类。



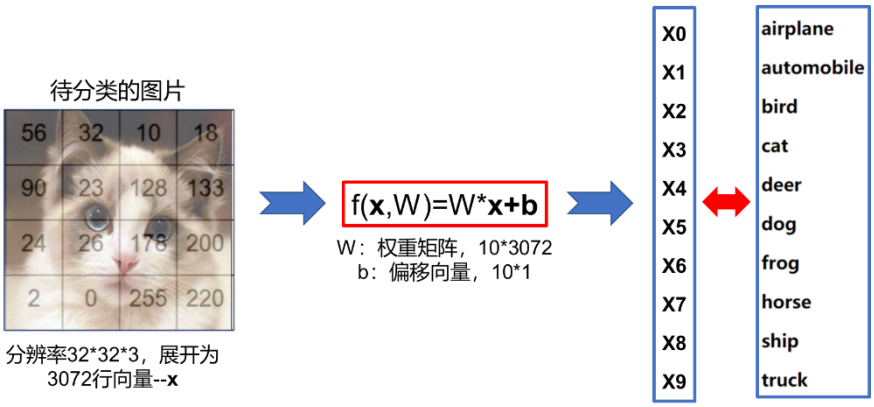
但这种处理方式需要对数据库中每一张图片进行一次处理，太复杂、麻烦，所以我们考虑针对不同类别进行处理。

为了构建这样的算法，我们首先观察上面的算法，可以看出这种算法实际上是对原图的所有像素进行了某种线性变化，所以我们可以构造下面的方式：

1.将图片所有像素 32*32*3 展开为 3072*1 的向量 x；

2.构建权重矩阵 W（10*3072）和偏移向量 b（10*1）；

3.用 W 乘以 x，得到 10*1 的行向量，根据行向量不同行的值确定图片的分类。



这里的 W 和 b 是根据数据集中的图片来训练的，因为数据集中的图片都已经有了分类，那么比如分类为猫的图片，最终结果应该为 x3 值最大而其他值尽量趋于 0，根据这一点构造损失函数 L，我们需要 W 和 b 的取值能让 L 在数据集上尽量最小，根据数据集来计算 W 和 b 的过程就叫做网络的训练；而训练所

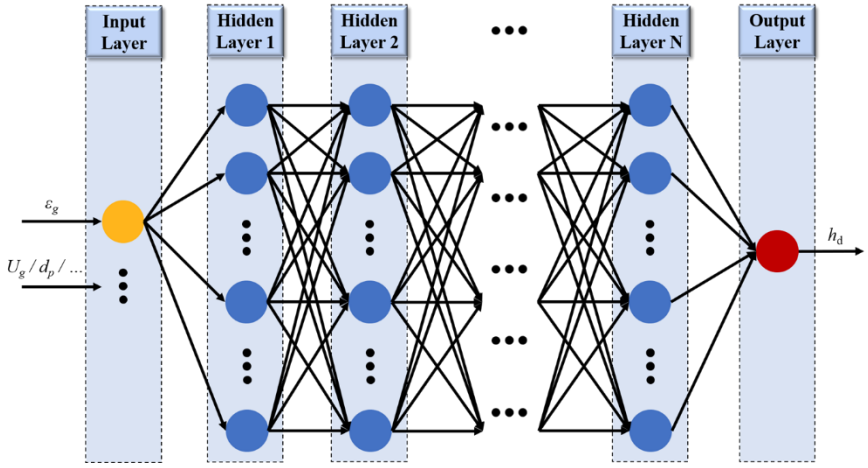
使用的方法现在常用的就是我们之前讲过的梯度下降法及其变种算法。

神经元及全连接网络：

上面的线性分类算法其实就是简单的神经网络，下面我们从最简单的全连接网络开始介绍。

首先是神经网络的基本单元——神经元，神经元其实可以看作上面的线性分类中输出为标量的情况。

然后，我们可以把这些神经元连接起来组成一个神经层，再把不同的神经层连接起来组成神经网络。

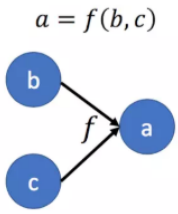


思考：这种神经元与神经网络有什么问题？

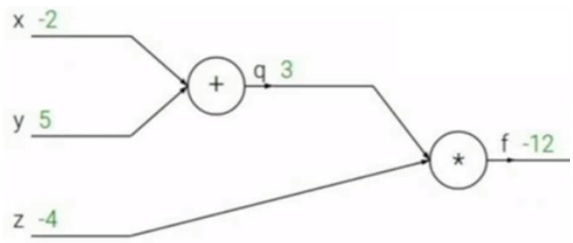
上述这种网络，不管有多少层、多复杂，最终结果也只能是与输入成线性关系，所以我们还需要在神经元中加入非线性成分——激活函数。激活函数有很多种，比如最初的 sigmoid 函数，后来的 tanh 函数等，但现在使用最多的是 Relu 函数，加入激活函数之后，就能组成一种常见的神经网络——全连接网络（如上图）。

计算图与反向传播：

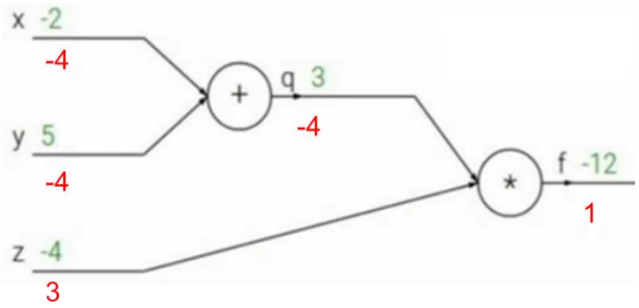
计算图是用图像的形式来描述函数的一种语言，其包括两种基本要素：变量（Node）和边（Edge），这里边指各种简单函数操作，比如下图：



比如，针对 f(x,y,z)=(x+y)z，其中 x=-2、y=5、z=-4，我们有：



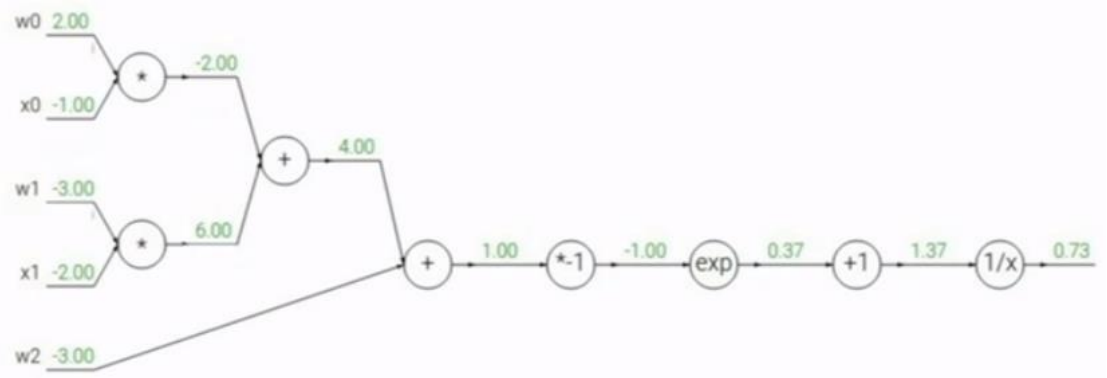
这种图的方式的好处是我们很容易的能够求解相关的导数：



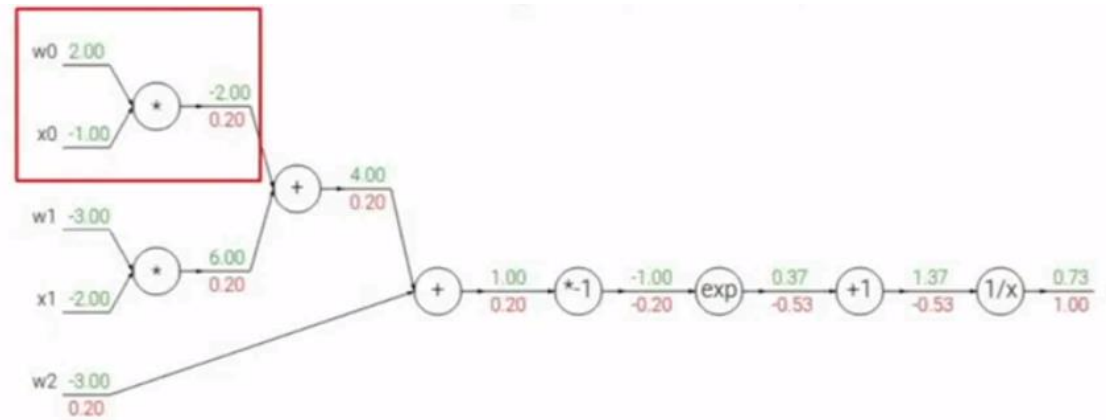
对于上面的计算可能你会觉得使用计算图是没有必要的，但是当计算变得复杂的时候，计算图就会显得清晰、容易很多，如：

$$f(w,x)=\frac{1}{1+e^{-(w_0x_0+w_1x_1+w_2)}}$$

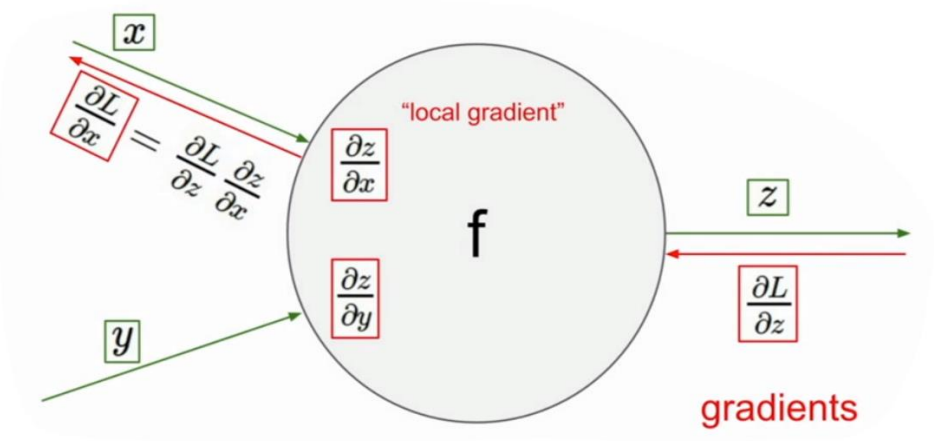
对这种公式我们就没那么容易直接计算导数了，这时就可以构建计算图如下：



相应的我们可以逐步计算相关的导数：



基于上述原理，我们就能够对神经网络中的参数求解梯度：

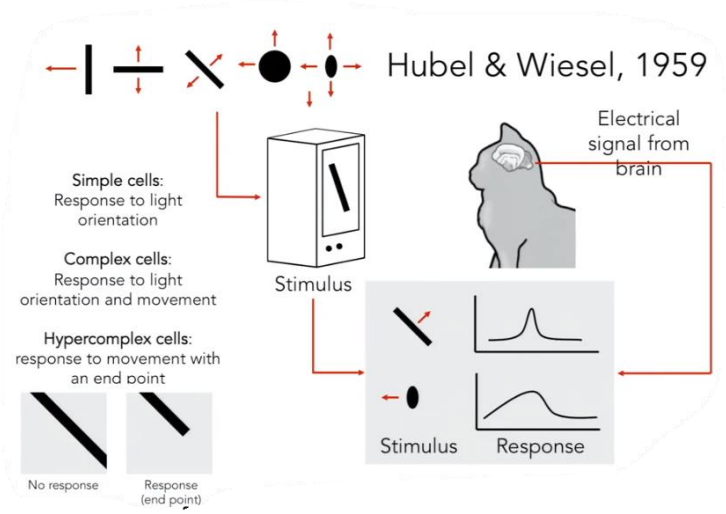


将上图中的计算流程运用到整个神经网络，就可以实现自动求导过程。

卷积神经网络：

前面所讲的全连接网络在图像领域虽然也能够使用，但仅局限于小分辨率的图片，对于大分辨率的图片，所需的网络参数太多、不现实；同时全连接网络也没有考虑图片本身的局部空间关联性，比如眼睛鼻子耳朵的相对位置等。所以，实际使用中，全连接网络往往是在拟合或者一些小规模图片等的问题上使用的，而对于图片等的处理往往会使用名为卷积神经网络的深度学习网络。

在继续后面的讲解之前，先让我们看一下 Hebel&Wiesel（1959）关于生物大脑如何处理视觉的实验：

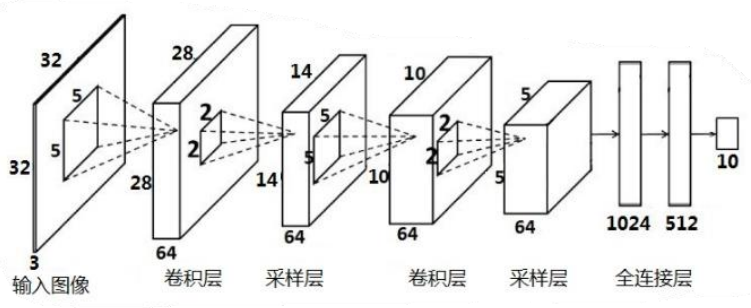


对一只意识清晰但是被麻醉的猫进行实验，把一根针插入它脑部的基础视觉皮层。给猫播放各种图片，看它的神经元有没有反应，结果没有反应。

但是在切换幻灯片的时候，猫有了反应，说明切换幻灯片的动作刺激的猫的神经。

他们发现：基础视觉区的神经元是按一列一列的组织起来，每一列神经元只“喜欢”某一种特定的形状，例如条纹，边界等，这说明视觉的最初处理，不是

整个形状，而是边缘和排列。



卷积神经网络与上述原理类似，如上图。

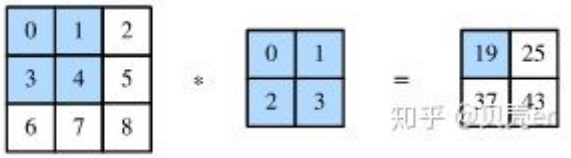
图像首先会经过卷积层，卷积层的作用于上面说的基础视觉皮层类似，是用来人别边界、条纹等简单几何特征的；之后通过采样层，将处理区域缩小，从而提高卷积时的视野区域；前两个过程会不断重复，从而能够识别更加复杂的特征；最终得到的结果会输入到几个全连接层，从而实现图像的识别等需求。

这里的卷积层实际上是进了了输入与卷积核的“卷积运算”，这里要注意和严格数学上的卷积运算是有一定的区别的：

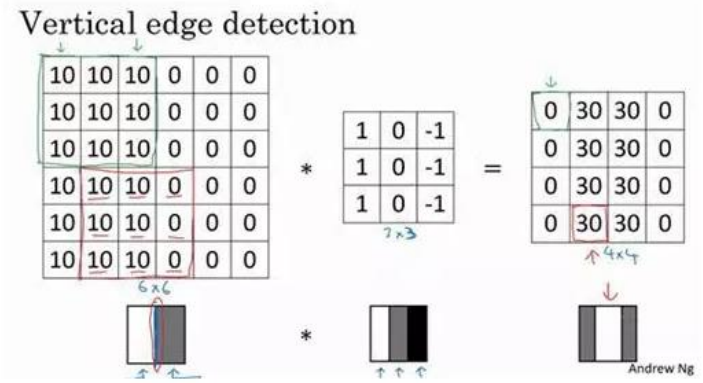
数学上卷积定义为：

$$(f*g)(n)=\int_{-\infty}^{\infty}f(\tau)g(n-\tau)d\tau \quad \text{或者} \quad (f*g)(n)=\sum_{\tau=-\infty}^{\infty}f(\tau)g(n-\tau)$$

数学上的卷积是需要先翻转然后进行滑动积分/求和的，而在神经网络这里我们不进行翻转，而是直接进行滑动求和，如下图：



可以看出，与严格数学意义上的卷积相比，我们这里没有对卷积核进行水平和竖直方向的翻转，实际上我们进行的是数学上的数学上的互相关计算，但是对于图像处理需求，翻转与否没有影响，所以我们一般还是称这种操作为卷积操作。



这里的卷积核就可以用来进行边界等的检测，如上图所示，可以看出通过这样的卷积核我们能够检测出相应的由明转暗的竖直边界，同样的也能够用不同的卷积核去检测由暗转明或者水平的边界等；而在更深层的卷积运算实际上就是检测这些基本结构的组合——不同的特征。

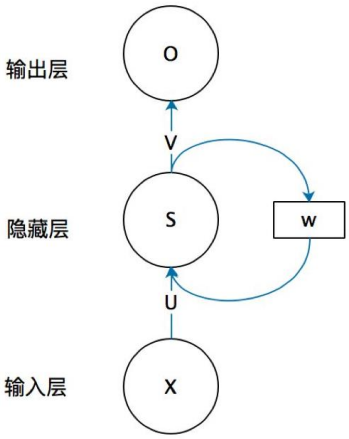
池化层简单来说就是把输入按某种方式（最大值或者平均值）转化为缩小的输出，其最直接的作用是降低下一层需要处理的数据量；同时其也能够提高下一层卷积核的视野。

我们可以把卷积神经网络中学习到的特征通过某种方式展示出来，通过这些图我们可以发现：CNN 学习到的特征，是具有辨别性的特征，比如要我们区分猫和手机，那么通过 CNN 学习后，背景部位的激活度基本很少，我们通过可视化就可以看到我们提取到的特征忽视了背景，而是把关键的信息给提取出来了。从 layer 1、layer 2 学习到的特征基本上是颜色、边缘等低层特征；layer 3 则开始稍微变得复杂，学习到的是纹理特征，比如上面的一些网格纹理；layer 4 学习到的则是比较有区别性的特征，比如猫头；layer 5 学习到的则是完整的，具有辨别性关键特征。

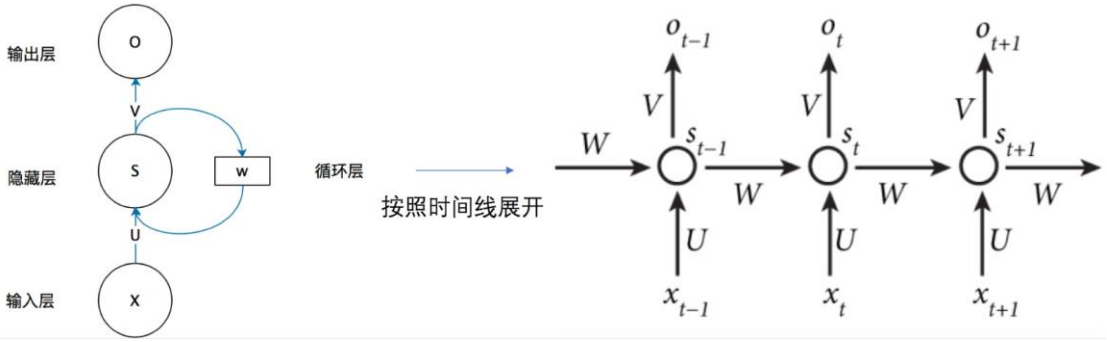
因为时间有限，我们这里也只能是简单的介绍一下相关的概念等。

循环神经网络：

前面所讲的全连接网络还有卷积神经网络处理的都是相互之间无关的数据，但对于有前后关联性的数据其处理效果往往不好，比如人说话的语音，每个语音之间并不是独立的，而是相关有关的，在理解的时候必须同时考虑到前后的语境才能更好的理解整段话，这时候我们前面讲的那些网络结构就显得不够用了，这时就需要使用到循环神经网络（RNN）。一个最基本的 RNN 网络如下图：

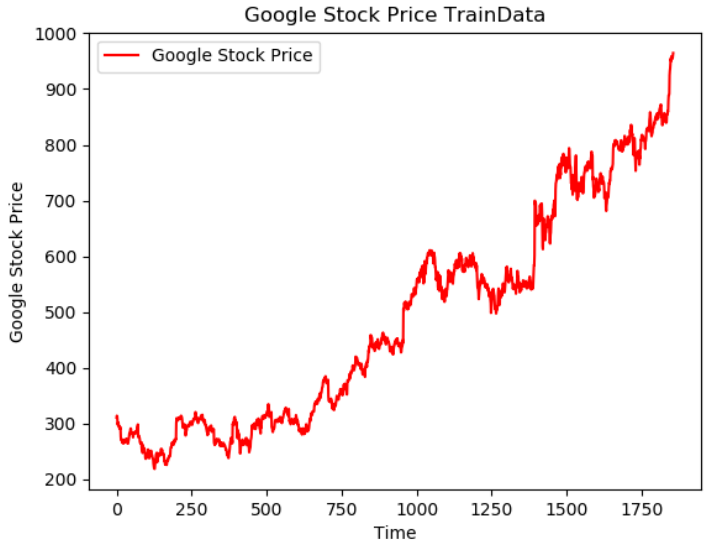


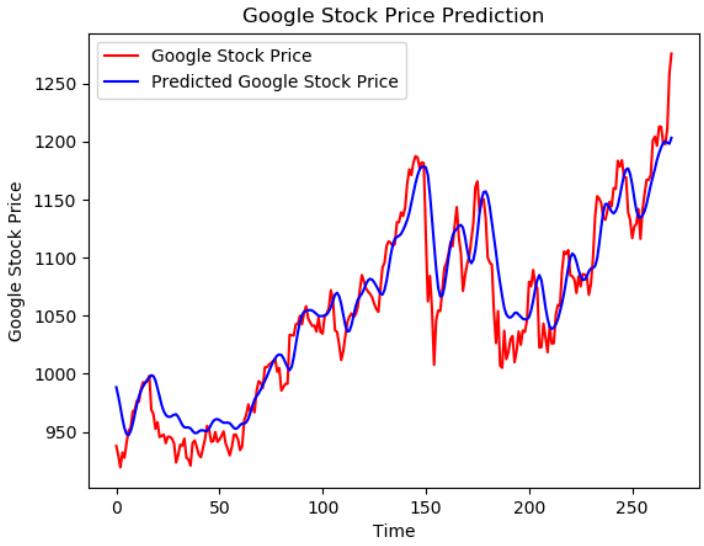
与前面所讲的网络不同，RNN 的输出结果不仅与当前的输入有关，还与之前时刻的输入有关，为了更清楚的展示 RNN 结构，我们可以把上述 RNN 单元按时间线来展开：



限于时间的问题，我们这里不再对 RNN 中的隐藏层的具体结构进行讲解，有兴趣的同学可以去具体了解相关的 LSTM、GRU 等常见结构。

回到本章开始时展示给大家的谷歌股价预测问题，对于那个问题咱们就是使用的 RNN 进行计算的，计算过程中使用 5 年的股价数据输入 RNN 中，这里我们把每一个月的股价作为一个数据 x，然后将其后一天的股价作为估测值 y，对于 RNN 模型进行训练，然后对于后续一年的股价进行预测，相关结果如下两图所示。





强化学习：

虽然强化学习的名称大家可能没有听说过，但强化学习的一个案例应该大家都是耳熟能详——AlphaGo。

阿尔法围棋（AlphaGo）是第一个击败人类职业围棋选手、第一个战胜围棋世界冠军的人工智能机器人，由谷歌（Google）旗下 DeepMind 公司戴密斯·哈萨比斯领衔的团队开发。其主要工作原理是“强化学习”。

2016 年 3 月，阿尔法围棋与围棋世界冠军、职业九段棋手李世石进行围棋人机大战，以 4 比 1 的总比分获胜；2016 年末 2017 年初，该程序在中国棋类网站上以“大师”（Master）为注册账号与中日韩数十位围棋高手进行快棋对决，连续 60 局无一败绩；2017 年 5 月，在中国乌镇围棋峰会上，它与排名世界第一的世界围棋冠军柯洁对战，以 3 比 0 的总比分获胜。围棋界公认阿尔法围棋的棋力已经超过人类职业围棋顶尖水平，在 GoRatings 网站公布的世界职业围棋排名中，其等级分曾超过排名人类第一的棋手柯洁。

2017 年 5 月 27 日，在柯洁与阿尔法围棋的人机大战之后，阿尔法围棋团队宣布阿尔法围棋将不再参加围棋比赛。

2017 年 10 月 18 日，DeepMind 团队公布了最强版阿尔法围棋，代号 AlphaGo Zero，它从零开始训练，仅 8 小时就击败了与李世石对战的 AlphaGo v18。然而 AlphaZero 带来的冲击远不止如此！在 AlphaZero 的封神之战上，面对当时世上最强的国际象棋引擎 Stockfish，AlphaZero 以 28 胜 72 平的百局不败战绩，将冠军 Stockfish 斩于马下。这样的结果不免令人震惊，此前大家都认为 Stockfish 已趋于完美，它的代码中有无数人类精心构造的算法技巧。论速度，Stockfish 以每秒 6 千万个位置的计算能力也足以完爆每秒 6 万的 AlphaZero。可现实情况却是——Stockfish 永远不可能战胜 AlphaZero。AlphaZero 拥有一种更加聪明的思维

模式，这使得它更明智，知道该思考什么，该忽略什么。这种更聪明的思维就来源于强化学习。

强化学习是一类算法, 是让计算机实现从一开始什么都不懂, 脑袋里没有一点想法, 通过不断地尝试, 从错误中学习, 最后找到规律, 学会了达到目的的方法，这就是一个完整的强化学习过程。

与前面我们所讲的所有算法不同，强化学习没有已知的标签或者数据库，计算机只能通过外界反馈所做决定的好与坏来自我评判所做决定，进而尽量去采取能够取得好结果的行为，从而达到自主学习。这里我们不去涉及具体的强化学习算法，而是带大家看一下实际的强化学习的例子。