# 1卷积神经网络

## 1.1使用场景

卷积主要用于模式识别（包括语音分析等）、图像处理、计算机视觉。卷积的主要目的是为了从输入图像中提取特征。卷积可以通过从输入的一小块数据中学到图像的特征，并可以保留像素间的空间关系。CNN是局部感受和参数共享，比较适合用于图像、时序这方面

卷积可以对图片进行锐化、模糊、边缘检测、浮雕等。边缘检测的图像可以认为是一种特征

## 1.2优缺点

优点：卷积可以从空间角度（通常是二维小块）对数据进行特征提取等。使用CNN进行特征提取，优点就是使用者完全不用关心具体的特征是哪些，即实现了特征提取的封装。

缺点：需要调参，需要大数据量，需要大计算量。给网络性能的改进罩上了一层黑盒子。

# 2循环、递归神经网络（RNN），LSTM网络

## 2.1使用场景

[多层感知器](http://blog.csdn.net/aws3217150/article/details/46316007)和[卷积神经网络](http://blog.csdn.net/aws3217150/article/details/46405095)，这两种结构有一个特点，就是假设输入是一个独立的没有上下文联系的单位，比如输入是一张图片，网络识别是狗还是猫。但是对于一些有明显的上下文特征的序列化输入，比如预测视频中下一帧的播放内容，那么很明显这样的输出必须依赖以前的输入， 也就是说网络必须拥有一定的”记忆能力”。为了赋予网络这样的记忆力，一种特殊结构的神经网络——递归神经网络(Recurrent Neural Network)便应运而生了。

## 2.2优缺点

优点：从时间角度考虑问题。这非常适合解决上下文敏感的问题。

缺点：

## 2.3LSTM长短时记忆神经网络

使用场景：

基于LSTM的系统可以学习翻译语言、控制机器人、图像分析、文档摘要、语音识别图像识别、手写识别、控制聊天机器人、预测疾病、点击率和股票、合成音乐等等任务。

优点：

是RNN的一个优秀的变种模型，继承了大部分RNN模型的特性。同时解决了梯度反传过程由于逐步缩减而产生的Vanishing Gradient问题。具体到语言处理任务中，LSTM非常适合用于处理与时间序列高度相关的问题，例如机器翻译、对话生成、编码\解码等。目前LSTM网络或者相关的门控单元同样用于编码和解码网络，并且在机器翻译中表现良好。

缺点：

目前在图像分类不如CNN，训练数据不够的话过拟合会很严重，效果堪忧。对于t时刻来说，输出层y\_t受到隐层h\_t（包含上下文信息）和输入层x\_t（当前的输入）的影响，但是y\_t和其他时刻的y\_t`是相互独立的，感觉像是一种point wise，对当前t时刻来说，我们希望找到一个概率最大的y\_t，但其他时刻的y\_t`对当前y\_t没有影响，如果y\_t之间存在较强的依赖关系的话（例如，形容词后面一般接名词，存在一定的约束），LSTM无法对这些约束进行建模，LSTM模型的性能将受到限制。

## 2.4循环神经网络其它变种

双向rnn，门控 RNN（包括长短时记忆和基于门控循环单元的网络）等。

# 3BP神经网络（反馈神经网络）

## 3.1使用场景

## 3.2优缺点

优点：

缺点：容易形成局部极小值而得不到全局最优值。神经网络中极小值比较多，所以很容易陷入局部极小值，这就要求对初始权值和阀值有要求，要使得初始权值和阀值随机性足够好，可以多次随机来实现。

训练次数多使得学习效率低，收敛速度慢。

隐含层的选取缺乏理论的指导。

训练时学习新样本有遗忘旧样本的趋势。

BP算法的改进：

增加动量项

自适应调节学习率

引入陡度因子

通常BP神经网络在训练之前会对数据**归一化处理**，即将数据映射到更小的区间内，比如[0,1]或[-1,1]

Rnn（Recurrent）循环神经网络

循环网络与多层网络相比，会共享每层的权重，从而能够扩展和应用网络于不同长度的序列案例，以及泛化这些案例。当**1片特定的信息可以在序列中多处出现时**，这样的共享尤为重要。比如，有两句话：“I went to Nepal in 2009”和“In 2009, I went to Nepal.”如果要求模型提取叙述者哪一年去的 Nepal，不管是输入句子的第2处还是第6处，2009 都是应识别的信息片。传统全连接前向网络对每处输入的特征都有**单独的参数**，所以要单独学习输入句子每一处的所有语法。相比之下，循环网络在不同时间步间**共享权重**。

Rnn（Recursive）递归神经网络

递归网络是循环网络的另1种泛化，但计算图的结构为1个深层树，而不是 RNN 的链式结构。

1990 年 Pollack 提出递归网络。在 NLP 和计算机视觉方面，递归网络成功用于处理送入神经网络的**数据结构**。

对序列长度为*τ*的循环网络，递归网络的深度 (构成非线性操作的数目) 从*τ*大幅减少至*O*(*logτ*)，有助于处理长期依赖 (long-term dependencies)。**最好的树结构**的设计问题尚未解决。1种选择为采用独立于数据的树结构 (如平衡二叉树)。某些领域中，外在方法能表现出合适的树结构。比如处理自然语句时，递归网络的树结构可固定为由自然语言解析器提供的语句解析树。学习器根据给定输入发现和推断出的树结构是理想的树结构。

递归网络的变种很多。1997 和 1998 年 Frasconi 等人用数据关联树结构，且用输入和输出关联树的节点。每个节点的计算不必是传统的人工神经计算 (所有输入的仿射变换，紧随1个单调的非线性)。比如，2013 年 Socher 发现建模概念 (连续向量表示) 间的关系时张量操作和双线性形式有用。

循环网络中，学习长期依赖的基本问题为**许多层上传播的梯度往往会消失 (常见) 或爆炸 (少见，优化伤害大)。**即使参数稳定 (可保存记忆，且梯度不爆炸)，与短期作用相比，长期作用的权重指数衰减，依然难以解决问题。其它方法用**更深的处理** (Hochreiter, 1991; Bengio 等, 1994; Pascanu 等, 2013)。

循环网络的多个时间步相同函数组合产生了极强的非线性。

有人希望通过保留梯度在不消失和不爆炸的参数空间区域内，不幸的是，为保存对小扰动鲁棒的区域，RNN 一定会进入梯度消失的参数空间区域。具体地，长期作用的梯度指数级地小于短期作用的梯度。这并不意味着梯度不可学，只是要花很长的时间去学长期作用，因为短期作用出现的波动常会隐藏长期作用的信号。1994 年 Bengio 等人实验表明扩大捕获的依赖范围会使基于梯度的优化难度加大，对长度为10或20的序列，用 SGD 的 传统 RNN 训练权重的成功率快速减为0。

**回响状态网络 (Echo State Networks)**

实际应用中本文最有效的序列模型为门控 RNN，包括长短时记忆和基于门控循环单元的网络。

类似泄漏单元，门控 RNN 基于使随时间的路径上的导数不消失或爆炸。泄漏单元用手动选择的常数或参数作为连接权重。门控 RNNs 将连接权重泛化成每个时间步都可能改变。

泄漏单元使网络能够在很长的时间内累积信息 (如特定的特征或类型)。然而，用到累积的信息后，网络会遗忘过去的状态。比如，当子序列组成序列，且需要通过设置过去的状态为0来遗忘它。除了手动决定何时清除状态，**希望网络能学习到何时去决定清除。**

# 4GAN生成式对抗网络

使用场景：

异常检测

优点：

与其他生成式模型相比，GAN这种竞争的方式不再要求一个假设的数据分布，即不需要formulate p(x)，而是使用一种分布直接进行采样sampling，从而真正达到理论上可以完全逼近真实数据

缺点：

不收敛问题：目前面临的基本问题是：所有的理论都认为 GAN 应该在纳什均衡（Nash equilibrium）上有卓越的表现，但梯度下降只有在凸函数的情况下才能保证实现纳什均衡。当博弈双方都由神经网络表示时，在没有实际达到均衡的情况下，让它们永远保持对自己策略的调整是可能的【OpenAI Ian Goodfellow的Quora】。

崩溃问题：难以训练：崩溃问题（collapse problem）

模型过于自由不可控：无需预先建模，模型过于自由不可控，不稳定。

变种：

条件生成式对抗网络，Conditional Generative Adversarial Networks

## GAN的优势与缺陷

    与其他生成式模型相比较，生成式对抗网络有以下四个优势[【OpenAI Ian Goodfellow的Quora问答】](https://mp.weixin.qq.com/s?__biz=MzA3MzI4MjgzMw==&mid=2650718178&idx=1&sn=6144523762955325b7567f7d69a593bd&scene=1&srcid=0821xPdRwK2wIHNzgOLXqUrw&pass_ticket=uG39FkNWWjsW38Aa2v5b3cfMhixqsJ0l1XLhNr5mivWEaLyW5R1QED0uAKHOwuGw" \l "rd" \t "_blank)：

* 根据实际的结果，它们看上去可以比其它模型产生了更好的样本（图像更锐利、清晰）。
* 生成对抗式网络框架能训练任何一种生成器网络（理论上-实践中，用 REINFORCE 来训练带有离散输出的生成网络非常困难）。大部分其他的框架需要该生成器网络有一些特定的函数形式，比如输出层是高斯的。重要的是所有其他的框架需要生成器网络遍布非零质量（non-zero mass）。生成对抗式网络能学习可以仅在与数据接近的细流形（thin manifold）上生成点。
* 不需要设计遵循任何种类的因式分解的模型，任何生成器网络和任何鉴别器都会有用。
* 无需利用马尔科夫链反复采样，无需在学习过程中进行推断（Inference），回避了近似计算棘手的概率的难题。

    与PixelRNN相比，生成一个样本的运行时间更小。GAN 每次能产生一个样本，而 PixelRNN 需要一次产生一个像素来生成样本。   
    与VAE 相比，它没有变化的下限。如果鉴别器网络能完美适合，那么这个生成器网络会完美地恢复训练分布。换句话说，各种对抗式生成网络会渐进一致（asymptotically consistent），而 VAE 有一定偏置。   
与深度玻尔兹曼机相比，既没有一个变化的下限，也没有棘手的分区函数。它的样本可以一次性生成，而不是通过反复应用马尔可夫链运算器（Markov chain operator）。   
与 GSN 相比，它的样本可以一次生成，而不是通过反复应用马尔可夫链运算器。   
与NICE 和 Real NVE 相比，在 latent code 的大小上没有限制。

### GAN目前存在的主要问题：

* 解决不收敛（non-convergence）的问题。   
      目前面临的基本问题是：所有的理论都认为 GAN 应该在纳什均衡（Nash equilibrium）上有卓越的表现，但梯度下降只有在凸函数的情况下才能保证实现纳什均衡。当博弈双方都由神经网络表示时，在没有实际达到均衡的情况下，让它们永远保持对自己策略的调整是可能的【OpenAI Ian Goodfellow的Quora】。
* 难以训练：崩溃问题（collapse problem）   
      GAN模型被定义为极小极大问题，没有损失函数，在训练过程中很难区分是否正在取得进展。GAN的学习过程可能发生崩溃问题（collapse problem），生成器开始退化，总是生成同样的样本点，无法继续学习。当生成模型崩溃时，判别模型也会对相似的样本点指向相似的方向，训练无法继续。【[Improved Techniques for Training GANs](https://arxiv.org/abs/1606.03498" \t "_blank)】
* 无需预先建模，模型过于自由不可控。   
      与其他生成式模型相比，GAN这种竞争的方式不再要求一个假设的数据分布，即不需要formulate p(x)，而是使用一种分布直接进行采样sampling，从而真正达到理论上可以完全逼近真实数据，这也是GAN最大的优势。然而，这种不需要预先建模的方法缺点是太过自由了，对于较大的图片，较多的 pixel的情形，基于简单 GAN 的方式就不太可控了。在GAN[Goodfellow Ian, Pouget-Abadie J] 中，每次学习参数的更新过程，被设为D更新k回，G才更新1回，也是出于类似的考虑。