**PPT上补充内容问题：**

1. 自动编码可以干什么？

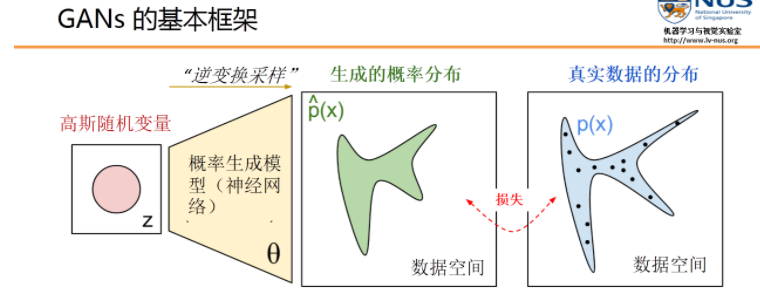
目前自编码器的应用主要有两个方面，第一是数据去噪，第二是为进行可视化而降维。配合适当的维度和稀疏约束，自编码器可以学习到比PCA等技术更有意思的数据投影。

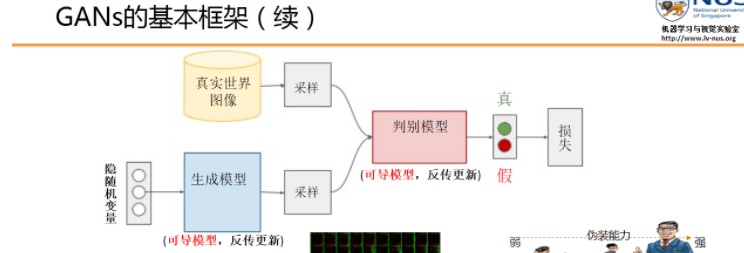
**自动编码器就是一种尽可能复现输入信号的神经网络。**为了实现这种复现，自动编码器就必须捕捉可以代表输入数据的最重要的因素，就像PCA那样，找到可以代表原信息的主要成分。自动编码器是一种数据的压缩算法，其中数据的压缩和解压缩函数是数据相关的、有损的、从样本中自动学习的。在大部分提到自动编码器的场合，压缩和解压缩的函数是通过神经网络实现的。这意味着自动编码器只能压缩那些与训练数据类似的数据。

1. GAN是一个怎样的模型？什么流程？怎么做？

**生成模型GAN**就是一种在拟合一张图像数组分布的一种模型，是概率统计结合深度学习之后的一次升级。它包括生成模型+判别模型两部分。

**流程如下图**：将一个随机变量（可以是高斯分布，或0到1之间的均匀分布），通过参数化的概率生成模型（通常是用一个神经网络模型来进行参数化），进行概率分布的逆变换采样，从而得到一个生成的概率分布（图中绿色的分布模型）。其训练目标，就是要最小化判别模型D的判别准确率。



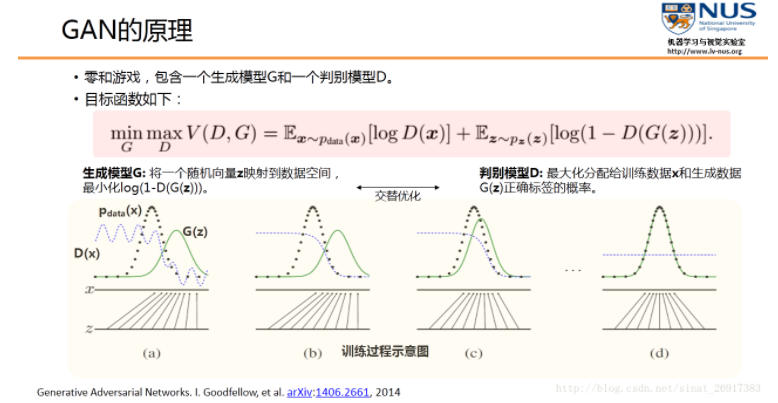


**具体训练**：

GAN采用了一种非常直接的交替优化方式，它可以分为两个阶段：

• 第一个阶段：固定判别模型D，然后优化生成模型G，使得判别模型的准确率尽量降低。

• 第二个阶段：固定生成模型G，来提高判别模型的准确率。



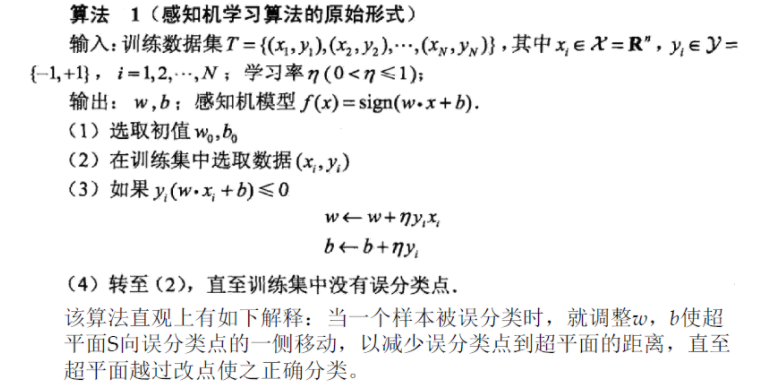
图(a)中黑色大点虚线P(x)是真实的数据分布，绿线G(z)是通过生成模型产生的数据分布（输入是均匀分布变量z，输出是绿色的曲线）。蓝色的小点虚线D(x)代表判别函数。

在图(a)中，我们可以看到，绿线G(z)分布和黑色P(x)真实分布，还有比较大的差异。这点也反映在蓝色的判别函数上，判别函数能够准确的对左面的真实数据输入，输出比较大的值。对右面虚假数据，产生比较小的值。但是随着训练次数的增加，图（b）和图（c）反映出，绿色的分布在逐渐靠近黑色的分布。到图（d），产生的绿色分布和真实数据分布已经完全重合。这时，判别函数对所有的数据（无论真实的还是生成的数据），输出都是一样的值，已经不能正确进行分类。G成功学习到了数据分布，这样就达到了GAN的训练和学习目的。

1. 感知机模型是什么？怎么训练？过程是怎样的？学习准则是什么？

**感知机**由Rosenblatt在1957年提出,是一种二类线性分类模型。输入一个实数值的n维向量（特征向量），经过线性组合，如果结果大于某个数，则输出1，否则输出-1.

**感知机训练过程**：



**学习准则**：寻找最佳**W**的一种方法是：首先给权向量的每个分类赋一个随机的权值，然后将这个感知机（此时模型参数已经有了）反复应用到所有的样本，如果样本分类错误（当前权向量的感知机的输出结果），则修改权向量，直到所有样本被正确分类。即误分类点到划分超平面S(w\*x+b=0)的总距离最小。

1. 传统机器学习的挑战？如何过度到深度学习的？

答：传统的机器学习算法的性能在很大程度上依赖给定数据的表示，表示的选择会对机器学习算法的性能产生巨大的影响。，对于许多任务，我们很难知道应该提取那些特征深度学习发觉表示本身，通过其他较简单的表示来表达复杂的表示，解决了表示学习中的核心问题

**书上内容相关问题：**

1. 深度学习历史趋势？为什么要用它？

答：20世纪40年代到60年代深度学习的雏形出现在控制论中

20世纪80年代到90年代深度学习以联结主义为代表

2006年开始，以深度学习之名复兴

为什么用它：

1）与日俱增的数据量

2）与日俱增的模型规模

3）与日俱增的精度、复杂度和对现实世界的冲击。

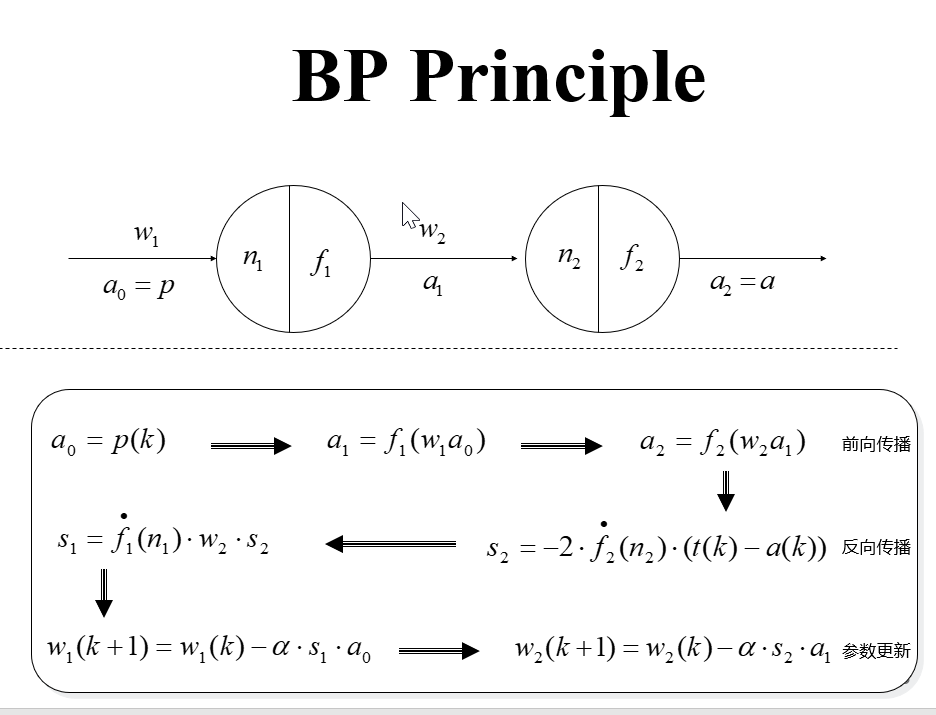
1. 前向传播后向传播的算法？

答：整个神经网络的训练过程可以简要描述如下

（1）将训练集数据输入到神经网络的输入层，经过隐藏层，最后达到输出层并输出结果，这是神经网络的前向传播过程；

（2）由于神经网络的输出结果与实际结果有误差，则计算估计值与实际值之间的误差，并将该误差从输出层向隐藏层反向传播，直至传播到输入层

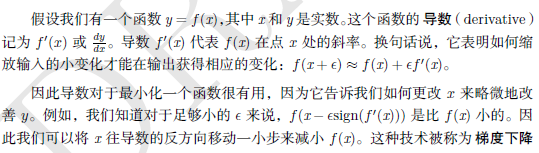
（3）在反向传播的过程中，根据误差调整各种参数的值；不断迭代上述过程，直至收敛。

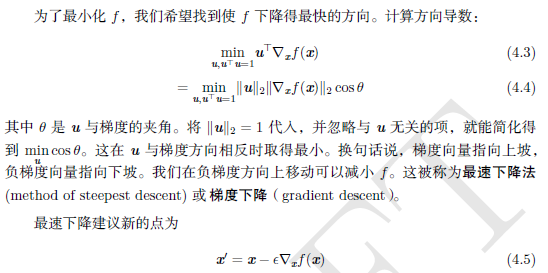
在正向传播结束之后，会根据预测值和标签计算Loss，反向传播是按照链式法则计算loss对每个参数的偏导数，然后乘以学习率，更新参数。

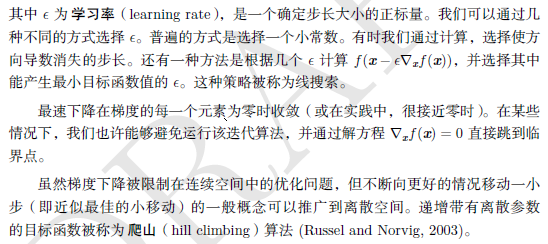
1. 梯度算法（随机梯度和梯度下降）？

答：梯度下降算法包括随机梯度下降算法和批量梯度下降算法。

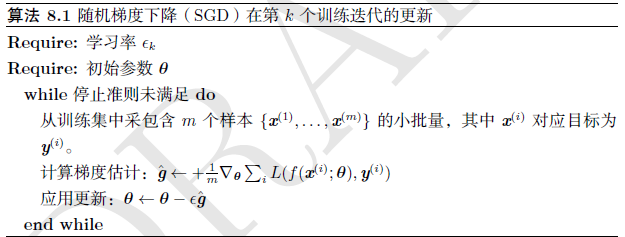
(1) 梯度下降算法：







(2) 随机梯度下降(SGD)按照数据生成分布抽取m个小批量（独立同分布的）样本，通过计算他们的梯度均值，可以得到梯度的无偏估计。



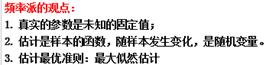
SGD算法中学习率是变化的，由于SGD中梯度估计引入的噪声源(m个训练样本的随机采样)并不会在极小点处消失，因此有必要随着时间的推移逐渐降低学习率(第k步迭代的学习率记作 )。但是批量梯度下降可以使用固定的学习率，因为使用批量梯度下降算法到达极小点时，整个代价函数的真实梯度会变得很小直到0。

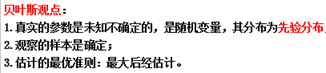
保证SGD收敛的充分条件是：,并且

实践中，一般会线性衰减学习率直到第次迭代：

学习率可通过试验和误差来选取，最好的选择方法是监测目标函数值随时间变化的学习曲线。使用线性策略时，需要选择参数，其中设为需要反复遍历训练集几百次的迭代次数；设为大约的1%；选择一个比在效果上表现最佳的学习率更大的学习率，但又不能太大导致严重的振荡（若太大，学习曲线会剧烈振荡，代价函数值会明显增加；若太小，则学习过程会很缓慢，可能会卡在一个相当高的代价值）。

1. 频率派和贝叶斯观点，两个派别的估计（最大似然和MAP）是怎样处理系统的不确定性？不确定性分观察的数据不全面和系统本身不确定





最大似然和最大后验的区别：(1)最大似然使用参数的点估计，MAP使用参数的概率分布；(2)最大似然方法用均方误差评估估计的不确定问题，MAP使用积分处理不确定；(3)MAP中加入了先验分布，先验分布实现了对参数的选择和偏好，因此有正则化作用。

1. 偏差、方差与欠过拟合的关联

答：(1) 偏差：，其中是模型预测输出的期望，是的真实值。方差：，随机变量是训练集。低偏差低方差时，预测值最接近真实值，且比较集中。低偏差高方差时，预测值接近真实值，但很分散，此时方差较大，模型的稳定性不够好。高偏差低方差时，预测值与真实值有较大距离，但此时值很集中，方差小，模型的稳定性较好，但预测准确率不高。高偏差高方差时，模型不仅预测不准确，而且还不稳定，每次预测的值都差别比较大。

(2) 欠拟合是指模型不够复杂或者训练数据过少时，模型均无法捕捉训练数据的基本(或者内在)关系，会出现偏差。这样一来，模型一直会错误地预测数据，从而导致准确率降低。过拟合是指模型过于复杂或者没有足够的数据支持模型的训练时，模型含有训练集的特有信息，对训练集过于依赖，训练误差和测试误差之间的差距太大。

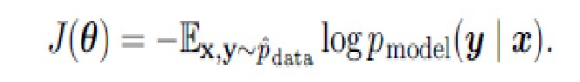
(3) 偏差、方差与欠过拟合的关联：一般来说，欠拟合的模型偏差比较大，而过拟合的模型方差比较大。模型欠拟合时，预测结果不准，偏差较大；但对于不同训练集，训练得到的模型都差不多(对训练集不敏感)，此时的预测结果差别不大，方差小。模型过拟合时，模型含有训练集的信息，预测的准确度一般不高，偏差较大；模型对训练集敏感，在与总体同分布的相同大小的不同训练样本上训练得到的模型，在验证集上的表现不一，预测结果相差大，方差大；由于模型含有训练集的信息，此时的训练得分很高，但验证得分不高(偏差大)。上述关系如下表所示：



1. 深度前馈网络用什么样的激活函数？代价函数？什么loss

（1）常见的代价函数：

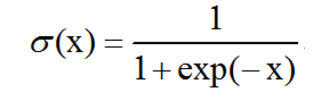
交叉熵（负的最大似然函数）



（2）激活函数：整流线性单元使用激活函数g(z)=max(0,z)，P108页

在引入整流线性单元之前，大多数神经网络使用logistic sigmoid激活函数

g(z)=σ(z)



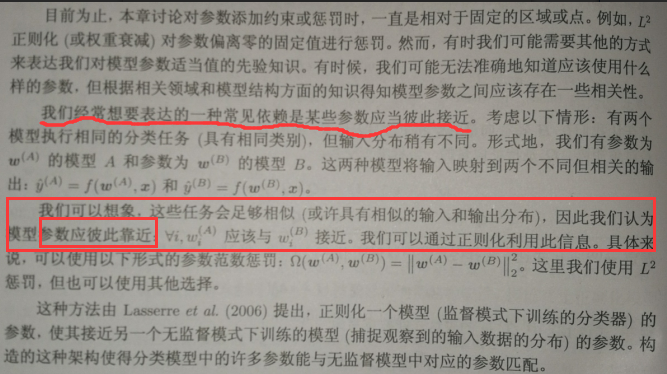
或者是双曲正切激活函数

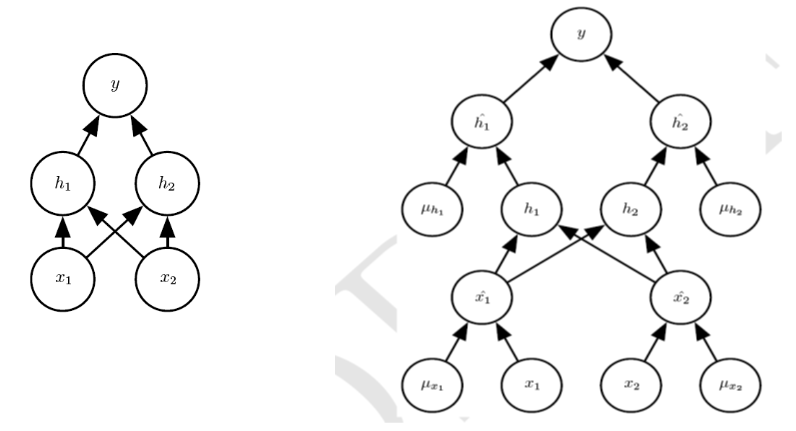
g(z)=tanh(z)

与分段线性单元不同，sigmoid单元在其大部分定义域内都饱和——当z取绝对值很大的正值时，它们饱和到一个高值，当z取绝对值很大的负值时，它们饱和到一个低值，并且仅仅当z接近0时它们才对输入强烈敏感。 sigmoid单元的广泛饱和性会使得基于梯度的学习变得非常困难。 因为这个原因，现在不鼓励将它们用作前馈网络中的隐藏单元。

（3）什么loss这个什么意思不懂

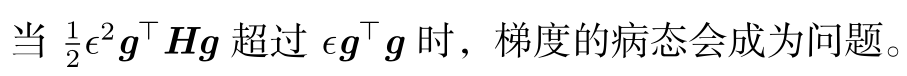
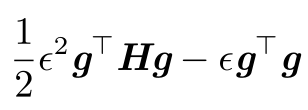
1. 正则化有哪些手段？什么时候使用？效果如何？
2. 正则化策略： 参数范数惩罚、数据集增强、增加噪声、半监督学习、多任务学习、提前终止、参数绑定与参数共享、稀疏表示、集成方法、dropout、对抗训练。
3. 正则化策略什么时候使用，效果如何：
4. 参数范数惩罚：限制模型（如神经网络、线性回归或逻辑回归）的学习能力。
5. 数据集增强：训练数据集不足时，创建假数据集并添加到数据集中。
6. 增加噪声：增强模型鲁棒性，使模型更健壮。
7. 半监督学习：使想同类中的样本有类似的表示。
8. 多任务学习：提高泛化
9. 提前终止：将数据集分为训练集和测试集，测试集用来探测最优的模型。当探测到最优模型，则训练终止。
10. 参数绑定与参数共享：



1. 稀疏表示：**隐藏层被激活的神经元数据较少**
2. 集成方法：结合几个模型降低泛化误差。Bagging方法：从训练集有重复采样k个子集，每个子集独立训练一个分类器。
3. Dropout：在模型上通过训练很多的子模型来实现近似Bagging方法。
4. 对抗训练：生成对抗样本实现对抗训练，通过对抗训练减少原有独立同分布的测试机的错误率。
5. 第八章深度模型的问题？如何应用

深度模型的优化问题。

1）深度模型训练的优化算法和传统的优化算法不同。前者通过降低代价函数的方式间接地优化性能度量P，机器学习中是最小化经验风险，这样容易导致过拟合。深度模型为了解决这个问题，采用了代理损失函数（比原函数学到的更多）和提前终止的方法。同时采用小批量训练，减少过拟合。

2）Hessian矩阵的病态。随机梯度下降会卡在某些情况，即使是很小的训练步长也会导致loss的上升。此时只能收缩学习率或者采用牛顿法。

3）局部极小值。优化算法可能会收敛到loss较高的局部极小值。

4）高原、鞍点和其他平坦区域。鞍点附近的梯度通常会非常小，应用到牛顿法也成为了一个问题（牛顿法会跳进鞍点）。通过扩展与修改，使用二阶的无鞍算法。

5）悬崖与梯度爆炸。可以使用启发式梯度截断，减少更新的步长。

6）长期依赖。计算图变得极深时，模型丧失了学习到先前信息的能力，有梯度消失和梯度爆炸问题。

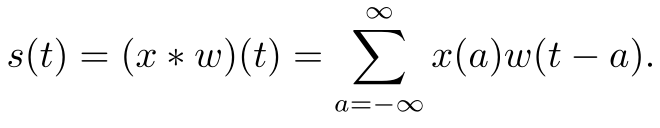
7）非精确梯度。梯度有噪声，通过代理损失函数解决这个问题。

8）局部与全局结构间的弱对应。局部能够到达最优，但无法指向全局的低loss区域。方法是选择良好的初始化点。

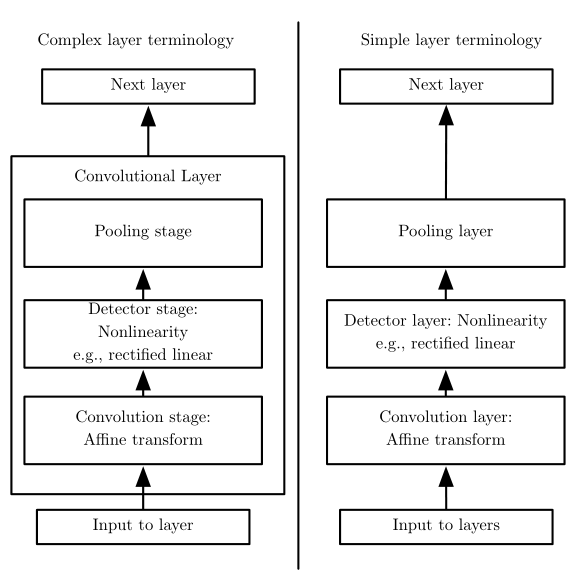
9）优化的理论限制。所以学术界在研究优化算法更现实的性能上界。

1. 卷积网络是怎么样的？有什么应用？

卷积：

离散形式

二维卷积核

卷积网络

应用：

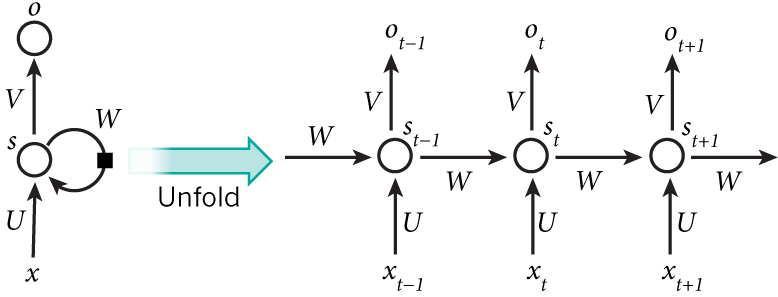
1）图像识别、目标检测、图像特征提取、图像自编码、图像主题生成、内容生成

2）文本生成、机器翻译（Facebook）、问答系统

1. 循环和递归网络 几种网络模型

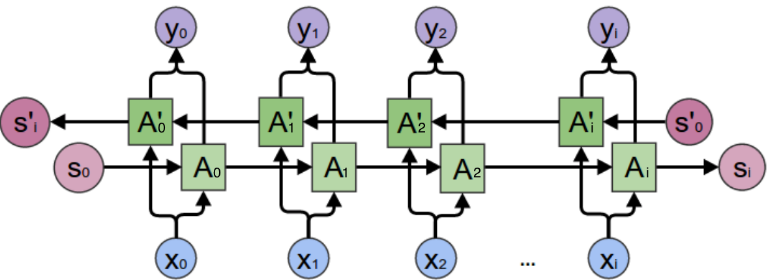
**循环神经网络**是一类用于处理序列数据的神经网络，包括三种重要的设计模式：

* 每个时间步都有输出，并且隐藏层之间有循环连接的循环网络；（重点讨论）
* 每个时间步都产生一个输出，只有当前时刻的输出到下个时刻的隐藏单元之间有循环连接的循环网络
* 隐藏单元之间存在循环连接，但读取整个序列后产生单个输出的循环网络



其中x是一个向量，它表示**输入层**的值；s是一个向量，它表示**隐藏层**的值（这里隐藏层面画了一个节点，你也可以想象这一层其实是多个节点，节点数与向量s的维度相同）；U是输入层到隐藏层的**权重矩阵**；o也是一个向量，它表示**输出层**的值；V是隐藏层到输出层的**权重矩阵。循环神经网络的隐藏层**的值s不仅仅取决于当前这次的输入x，还取决于上一次**隐藏层**的值s**。权重矩阵**W就是**隐藏层**上一次的值作为这一次的输入的权重。

**双向RNN**结合时间上从序列起点开始移动的RNN和另一个时间上从序列尾端开始移动的RNN，即不仅需要上一时刻隐藏层单元的值，还需要下一时刻隐藏层单元的值。



双向RNN的隐藏层需要保存两个值，一个值A参与正向计算，另一个值A’参与反向计算，输出值取决于A和A’。计算方法为：







**深度循环神经网络**简单说就是不止一个隐藏层

**递归神经网络**是循环神经网络的另一个扩展，它被构造为深的**树状结构**而不是RNN的链状结构，因此是不同类型的计算图。

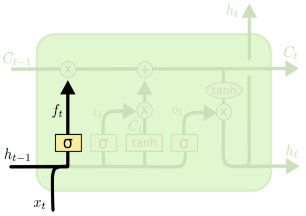
**递归神经网络**可以把一个树/图结构信息编码为一个向量，也就是把信息映射到一个语义向量空间中。这个语义向量空间满足某类性质，比如语义相似的向量距离更近。也就是说，如果两句话（尽管内容不同）它的意思是相似的，那么把它们分别编码后的两个向量的距离也相近；反之，如果两句话的意思截然不同，那么编码后向量的距离则很远。

递归网络的一个明显优势是，对于具有相同长度t的序列，深度（通过非线性操作的组合数量来衡量）可以急剧的从t减小为O(logt)。

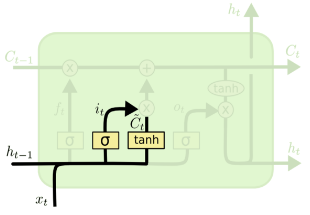
**LSTM**主要是为了解决RNN会梯度爆炸和梯度消失而产生的，LSTM通过门控制器和新的记忆单元，在RNN原有的短期记忆之上保留了长期记忆。如果一个事件非常重要，则输入门就按重要程度将短期记忆合并到长期记忆，或者通过遗忘门忘记部分长期记忆，按比例替换为现在的新记忆。在最后，输出门会基于长期记忆和短期记忆综合判断到底该有什么样的输出。

LSTM是通过精心设计的称作为“门”的结构来去除或者增加信息到细胞状态的能力。门是一种让信息选择式通过的方法。他们包含一个 sigmoid 神经网络层和一个 pointwise 乘法操作。Sigmoid 层输出 0 到 1 之间的数值，描述每个部分有多少量可以通过。0 代表“不许任何量通过”，1 就指“允许任意量通过”！

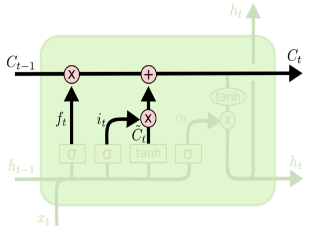
第一层是遗忘层，决定细胞状态中丢弃什么信息。把和拼接起来，传给一个sigmoid函数，该函数输出0到1之间的值，这个值乘到细胞状态上去。sigmoid函数的输出值直接决定了状态信息保留多少。



上一步的细胞状态已经被忘记了一部分，接下来本步应该把信息新加到细胞状态中。这里又包含2层：一个tanh层用来产生更新值的候选项，tanh的输出在[-1,1]上，说明细胞状态在某些维度上需要加强，在某些维度上需要减弱；还有一个sigmoid层（输入门层），它的输出值要乘到tanh层的输出上，起到一个缩放的作用，极端情况下sigmoid输出0说明相应维度上的细胞状态不需要更新。



现在可以让旧的细胞状态与（f是forget忘记门的意思）相乘来丢弃一部分信息，然后再加个需要更新的部分（i是input输入门的意思），这就生成了新的细胞状态。



最后该决定输出什么了。输出值跟细胞状态有关，把输给一个tanh函数得到输出值的候选项。候选项中的哪些部分最终会被输出由一个sigmoid层来决定。

