# 一、机器学习过程

## 1、数据预处理

### 1.1、数据清洗

### 1.2、数据采样

### 1.3、数据集拆分

#### 1.3.1、训练集（进行训练4/5）

#### 1.3.2、测试集（对已经训练好的模型进行测试1/5）

#### 1.3.3、验证集（调整超参数变量0-1/5）

### 1.4、交叉验证（N-1块数据训练，1块数据测试，进行N次循环）

## 2、特征工程

### 2.1、特征提取

### 2.2、特征预处理

#### 2.2.1、归一化（适合传统精确小数据场景）

#### 2.2.2、标准化（适合大数据场景）

### 2.3、特征降维

含义：特征矩阵过大，计算量过大，所以要降维;2\*4维a矩阵乘4\*4维b矩阵得到2\*4维c矩阵，即b-->c为降维过程

### 2.4、规范化

## 3、数据建模

### 3.1、分类问题（鲍鱼年龄预测）

#### 3.1.1、近邻算法（你的邻居来判断你的类别）

算法原理（给定一个训练数据集，输入新的实例，在训练数据中找到与实例最近的k个实例，这k个实例属于哪个类，输入实例就属于哪个类）

优缺点：简单易实现不需训练；k值的指定对精度影响大

#### 3.1.2、决策树与随机森林（类似于对所有类别做思维导图）

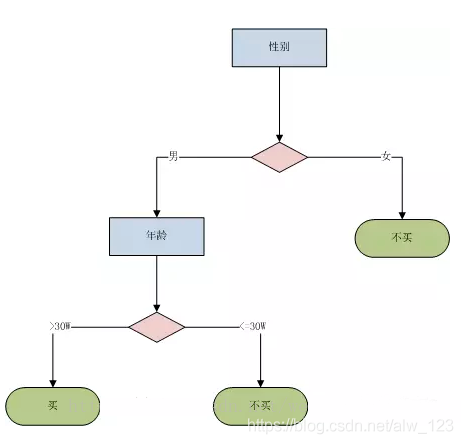
算法原理（类似与人的脑回路：通过信息增益来构建）

（信息增益(ID3算法,通俗理解将信息熵由大到小排列，即不确定性越大到越小排列)<--信息熵(随机变量的不确定性,越不确定，信息熵越大)）

例：猜你明星-->男的(不是女的)-->内地(其他地区)-->演员(爱豆)-->肖战

优缺点：单独决策树可视化易过拟合

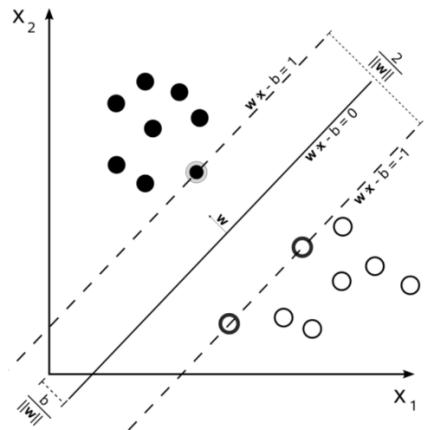
随机森林:使用m颗决策树，产生m个样本训练每颗树，解决决策树过拟合问题



#### 3.1.3、贝叶斯原理（（P(B|A)=P(B)P(A|B)/P(A),P(B|A)表示在A发生的情况下B发生的概率，即A的后验概率）

含义：朴素贝叶斯（上面的B有多种情况，B1和B2）

#### 3.1.4、支持向量积（用线性来分开两个区域类别）

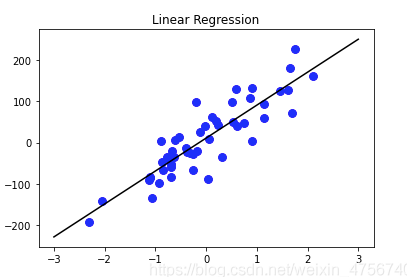


#### 3.1.5、集成学习（结合多个机器学习方法）

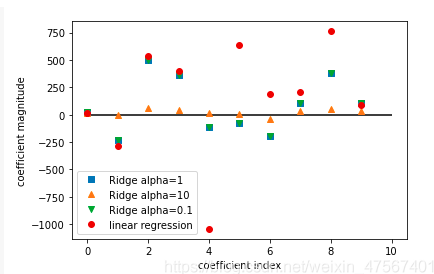
#### 3.1.6、逻辑回归（sigmoid，通过已知数据，求得相应的参数，给定，得到，带入sigmod函数得到对应值，f(x)=1/(1+e-x)）

### 3.2、回归问题（房价预测）

#### 3.2.1、线性回归（找到训练数据集中的和真实值的平方差最小，数据量需多）



#### 3.2.2、岭回归（避免过拟合的线性模型，数据量少）



#### 3.2.3、套索回归

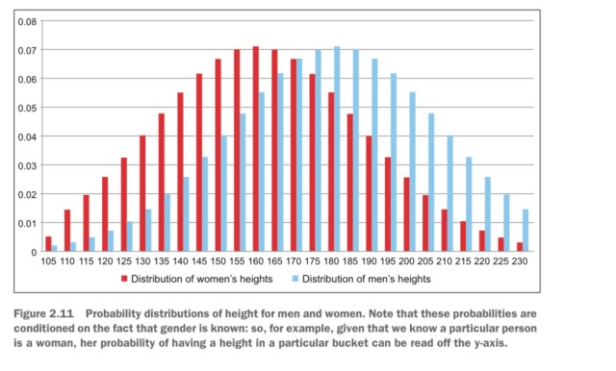
### 3.3、聚类问题（把数据分好后，判断属于哪一类；分类是先确定类别，再判断）

#### 3.3.1、k-means聚类算法（两个目标距离越近，相似度越大）

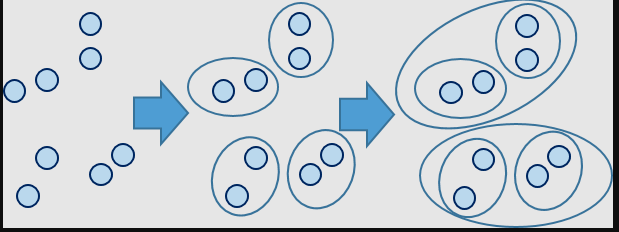
例：名牧师选址开讲座，先随机选两个位置，所有村名按距离远近分给两个牧师；每名牧师收集到他这听课的村民地址，重新选择最中间的位置， 所有村名按距离远近重新选择两个牧师；当所有村民没有改变时就确定了位置。

#### 3.3.2、密度聚类

#### 3.3.3、高斯混合模型（常见EM算法）（多个高斯分布模型）



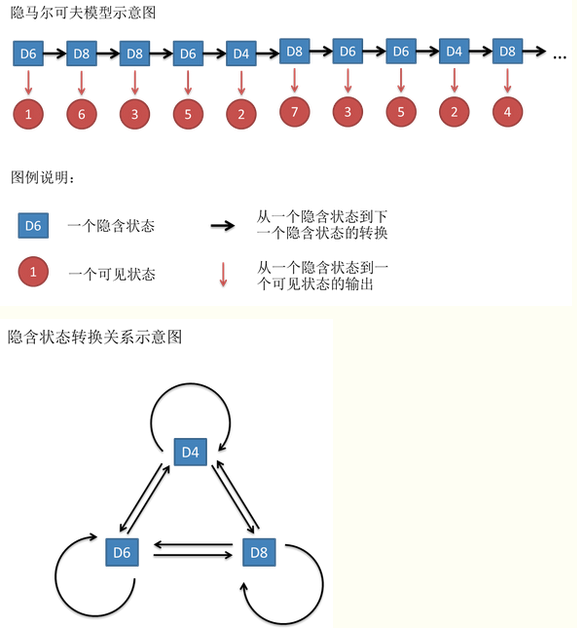
#### 3.3.4、层次聚类（通过计算不同类别数据间的相似度来创建一颗有层次的嵌套聚类树）



#### 3.3.5、谱聚类（基于图论的聚类方法）

### 3.4、其他问题

#### 3.4.1、隐马尔科夫模型（HMM）



#### 3.4.2、条件随机场（CRF）（是马尔科夫随机场的特例，可解决序列标注问题，考虑邻近标签的类别）

#### 3.4.3、LDA主题模型

4、结果评估

### 4.1、过拟合（原始特征过多，模型过于复杂尝试去兼顾各个测试数据点）

解决方式：正则化使w=0(L1范数)或者趋于0(L2范数)

### 4.2、欠拟合（学习到数据的特征过少）

解决方式：增加数据的特征数量

### 4.3、准确率

### 4.4、召回率

# 二、机器学习

## 1、有监督学习

定义：从成对已经标记好的输入输出数据作为模型输入，来预测输出结果，即从正确答案的例子中学习，已知(x1,y1)、(xi,yi)，构建出模型，计算当x为某值时，y？

### 1.1、分类问题（类别）

### 1.2、回归问题（连续型的数据）

## 2、无监督学习

定义：在数据中发现一些规律

### 2.1、聚类问题

### 2.2、降维问题

## 3、半监督学习

# 三、python

## 1、Numpy（存储处理大型矩阵）

### 1.1、Pandas（数据分析工具）

### 1.2、Scipy（科学工程领域软件包：统计、线性代数、傅里叶变换）

## 2、Matplotlib（数据可视化）

## 3、sklearn（机器学习算法库，封装了常用的机器学习方法：回归、降维、分类、聚类）

四、深度学习

1、过程

(1) 数据预处理-词向量（文本数值化）本项目实现了两种数据预处理方式，一种是使用pytorch提供的dataset和dataloader实现数据预处理，一种是直接使用torchtext（强烈推荐上手试试）。两种方法的实现过程大体都是一致的，分词、数值化（word --> id）、填充、封装。最后一步是为了在训练过程中直接从迭代器中取出可输入到网络中的数据格式。TF-IDF、BOW、One-Hot、分布式的表示方式（word2vec、Glove）等。

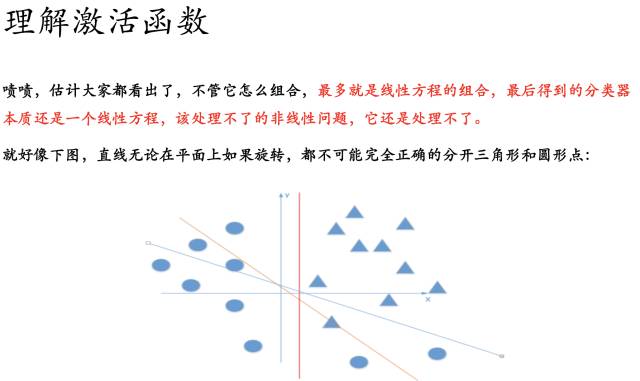
1. 初始化参数
2. 前向传播
3. 计算代价函数
4. 反向传播
5. 更新参数
6. 模型评估

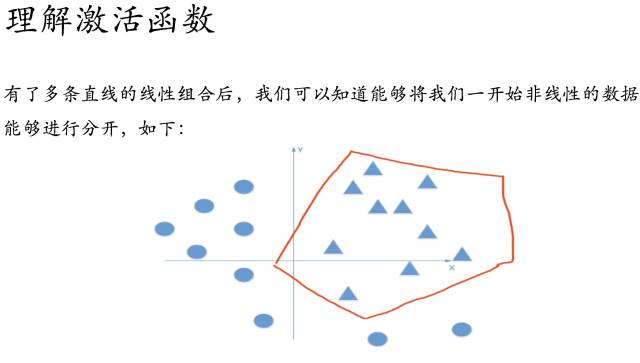
# 五、一些名词

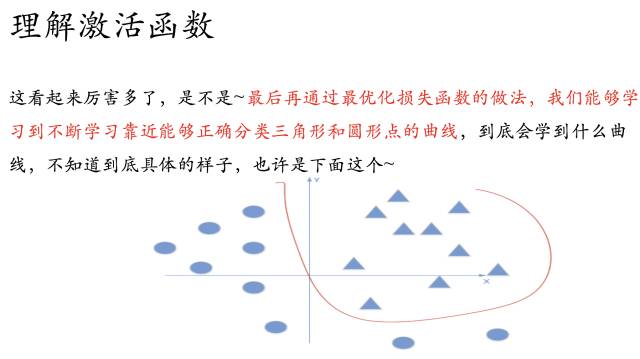
## 1、激活函数

https://blog.csdn.net/UFv59to8/article/details/78884788

非线性，结果控制在0-1之间，数据达到收敛效果







## 2、损失函数、代价函数、目标函数

\*\*损失函数（Loss Function）：\*\*定义在单个样本上的，指一个样本的误差

\*\*代价函数（Cost Function）：\*\*定义在整个训练集上，是所有样本误差的平均，也就是所有损失函数值的平均

\*\*目标函数（Object Function）：\*\*指最终需要优化的函数，一般来说是经验风险+结构风险（代价函数+正则化项）。

## 前向传播、反向传播

<https://blog.csdn.net/weixin_38347387/article/details/82936585>

前向传播：给定输入值，输出值，初始权重，计算第一次输出值与原始输出值的误差

反向传播：利用前向传播求出的损失函数，分别求所有权重的偏导，给定学习率a，调整权重，获得新的w\_new

w\_new = w\_old -a (E\_total/w\_old)’

根据迭代次数继续前向传播--->后向传播，当调整到损失函数非常小，即值与正确的y接近就停止。

## 预训练模型

含义：假设已有A训练集，先用A对网络进行预训练，在A任务上学会网络参数，然后保存以备后用，当来一个新的任务B，采取相同的网络结构，网络参数初始化的时候可以加载A学习好的参数，其他的高层参数随机初始化，之后用B任务的训练数据来训练网络，当加载的参数保持不变时，称为"frozen"，当加载的参数随着B任务的训练进行不断的改变，称为“fine-tuning”，即更好地把参数进行调整使得更适合当前的B任务。

优点：当任务B的训练数据较少时，很难很好的训练网络，但是获得了A训练的参数，会比仅仅使用B训练的参数更优

bert：

含义：训练通用语言模型的工具。

核心思想：随机 mask 句子中的单词，然后通过句子全部上下文（即无方向性）尝试预测它们。（LSTM只可以通过从左往右或者从右往左）

工作原理：依附于’transformer’（一种标注机制，用来学习文本中单词之间的上下文关系；编码器【读取文本输入】+解码器【产生关于任务的预测】）

输入：token embeddings（词嵌入、分词）+segment embeddings（段嵌入、按句子标点符号区分、只有01）+positive embeddings（正嵌入、让BERT学习到输入的顺序属性

）

Masked Language Model：用[MASK]代替句子中的15%的词，通过预测这部分的词语学习句子的语义。

Next Sentence Prediction：（一个预训练中的任务）

该任务是指将两个句子作为输入，做一个判断任务，即第二句话是不是第一句话的下一个任务。

## 5、迁移学习

含义：为了对迁移学习产生一个直观的认识，不妨拿老师与学生之间的关系做类比。

一位老师通常在ta所教授的领域有着多年丰富的经验，在这些积累的基础上，老师们能够在课堂上教授给学生们该领域最简明扼要的内容。这个过程可以看做是老手与新手之间的“信息转移”。

这个过程在神经网络中也适用。我们知道，神经网络需要用数据来训练，它从数据中获得信息，进而把它们转换成相应的权重。这些权重能够被提取出来，迁移到其他的神经网络中，我们“迁移”了这些学来的特征，就不需要从零开始训练一个神经网络了 。

## CNN(卷积神经网络)

通俗理解：<https://blog.csdn.net/u014106644/article/details/84916710>；卷积核去拓印特征点（降维）---》rule---》卷积---》rule---》池化（降维）

卷积（权重）：将实际问题分成许多特征点，即要匹配一张图片是否是猫，首先得确认猫的特征点有：猫头(其中眼睛、耳朵)，拿特征点(眼睛、耳朵)去对测试图片做卷积运算，得到测试图片的卷积结果A

池化（降维）：对卷积测试结果A提取最大像素值，即缩小原始图像，得到池化的结果B（即猫头）

全连接：将池化后的结果去匹配测试图片，判断测试图片是否包含池化特征（即测试图片是否是猫）

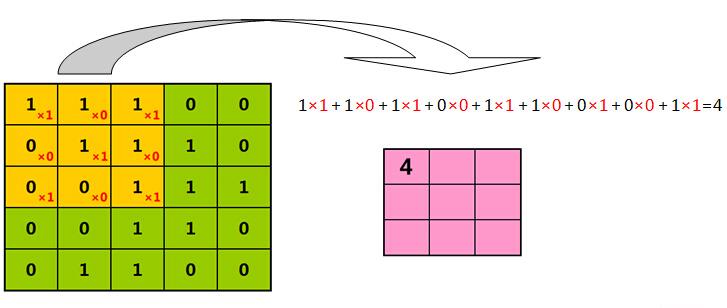
1. 过程：

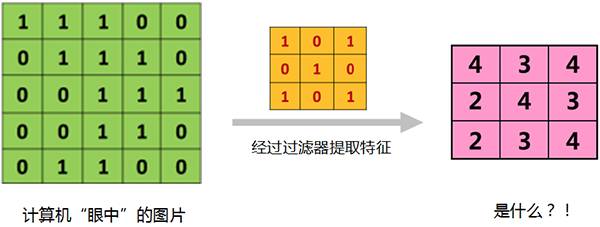


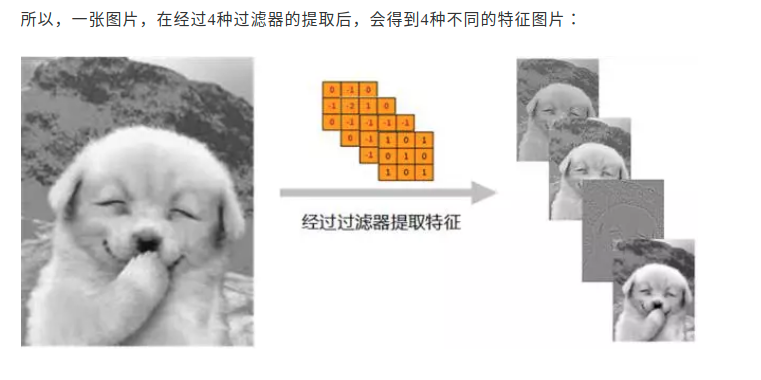
1. 特征提取--卷积

https://mp.weixin.qq.com/s/x6NIkzQSgvl0\_rfGe7Cjqg?

通过过滤器(下图黄色)，找到图片中的特征(下图粉色)

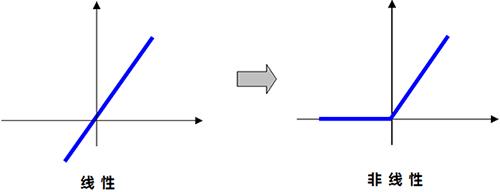


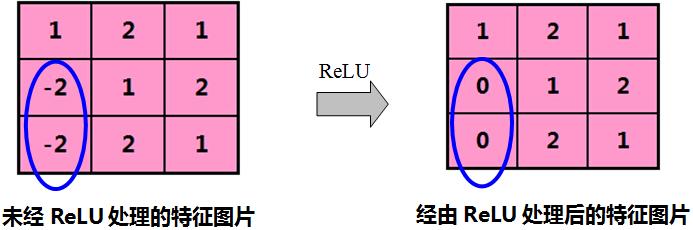




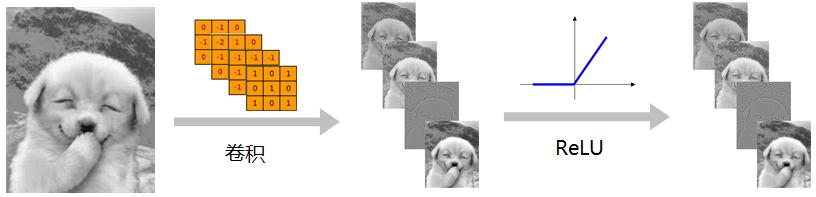
1. 池化--将上述线性问题转化成非线性问题

pianshen.com/article/5504672187/





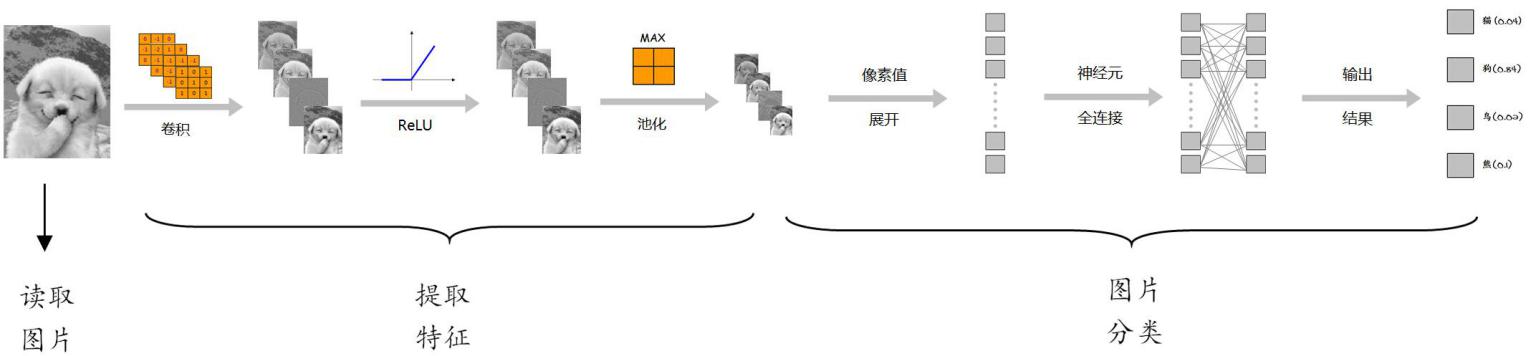
回到我们之前的例子中，在“卷积”的后面，再加上一步“ReLU计算”，即为：



**所谓“池化”，就是在保留图片主要信息的前提下，将图片的尺寸缩小。**



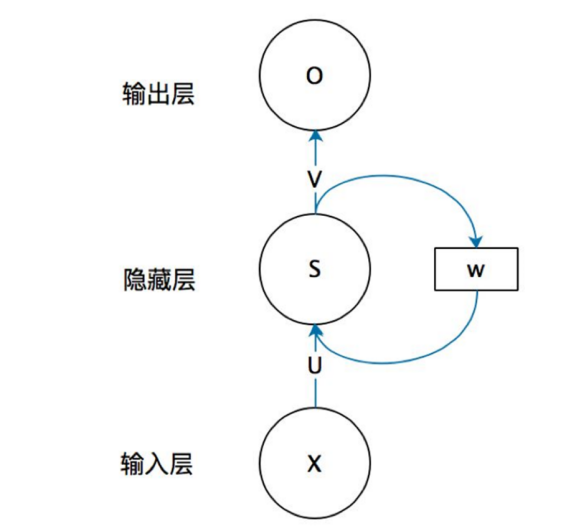




## 循环神经网络(rnn)

https://blog.csdn.net/qq\_39422642/article/details/78676567

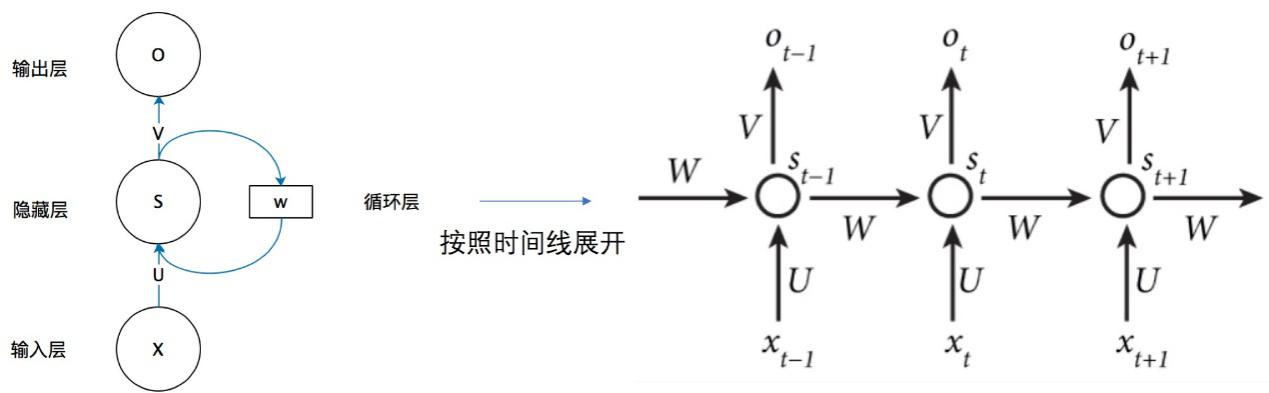
最为原始的循环神经网络，本质就是全连接网络，只是为了考虑过去的信息，输出不仅取决于当前输入，还取决于之前的信息，也就是输出由之前的信息(也就是状态state)和此时的输入决定。

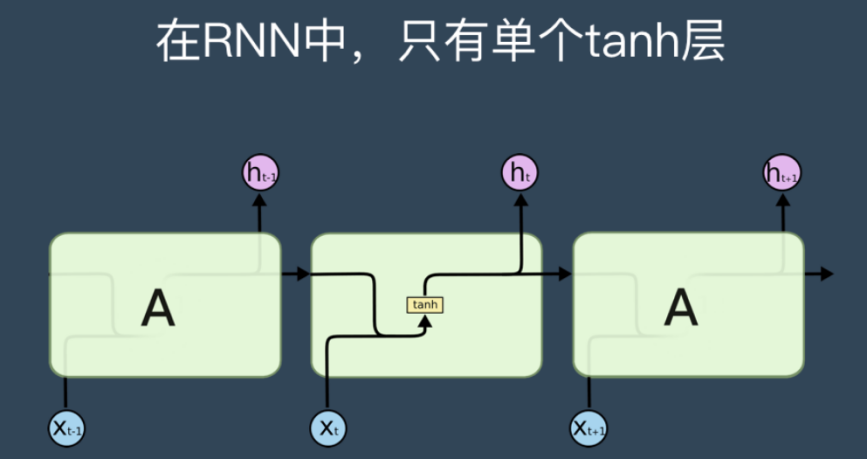


如果把上面有W的那个带箭头的圈去掉，它就变成了最普通的全连接神经网络。x是一个向量，它表示输入层的值；s是一个向量，它表示隐藏层的值（这里隐藏层面画了一个节点，你也可以想象这一层其实是多个节点，节点数与向量s的维度相同）；

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。

那么，现在我们来看看W是什么。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。





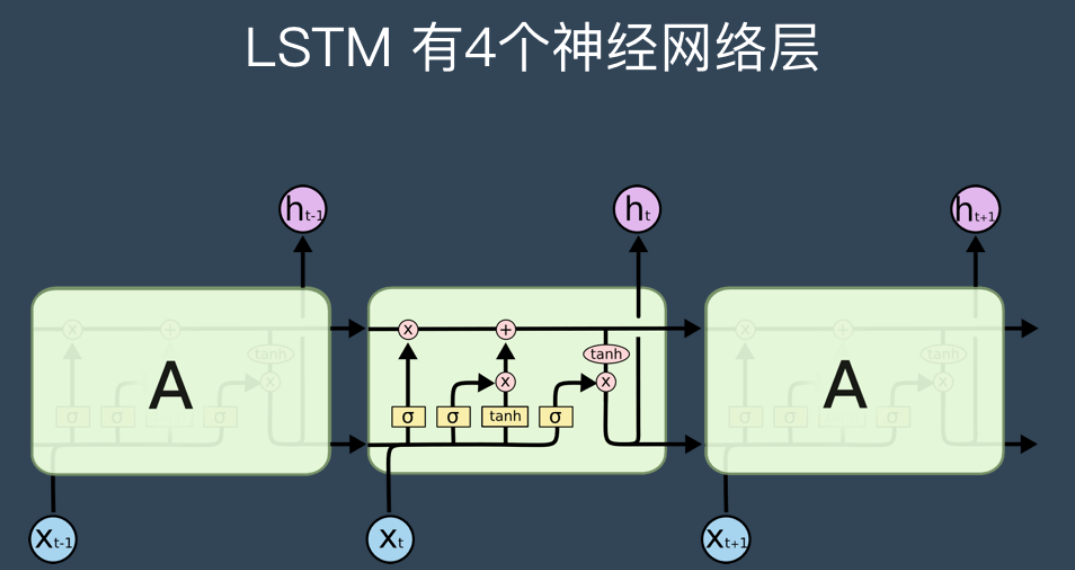
① 梯度消失、梯度爆炸

② rnn较其他cnn和全连接要用更多的显存空间，更难训练

③ 如果采用tanh、relu为激活函数，没法处理太长的序列

## 7、长短期记忆神经网络(lstm)

输入门、遗忘门和输出门



## 8、门控循环单元(GRU)

因为LSTM的训练比较慢，而GRU在其上稍微修改，速度可以快很多，而精度基本不变，所以GRU也十分流行。

更新门和重置门

## 9、 文本对抗网络(GAN)

GAN全称对抗生成网络，顾名思义是生成模型的一种，而他的训练则是处于一种对抗博弈状态中的。

GAN的主要结构包括一个生成器G（Generator）和一个判别器D（Discriminator）。

10、

RoFormer-Sim

RoFormer-Sim