白话深度学习与TensorFlow

# 机器学习是什么

算法：表示的是一种确定性的机器指令执行序列。

基本完整的有监督学习的过程：



机器学习从学习的种类来说，可以分为：无监督学习（unsupervised learning） 和 有监督学习(supervised learning)

a. 无监督学习：在获得训练的向量数据后在没有标签的情况下尝试找出其内部蕴含关系的一种挖掘工作，这个过程中使用者除了可能要设置一些必要的超参数（hyper-parameter）以外，不用对这些样本做任何的标记甚至是过程干预；

b. 有监督学习：每一个样本都有着明确的标签，最后我们只是要总结出这些训练样本向量与标签的映射关系。

## 聚类

聚类（Clustering），是一种典型的“无监督学习”。是把物理对象或抽象对象的集合分组为由彼此类似的对象组成的多个类的分析过程。

比较常用的聚类算法有K-Means, DBSCAN 等，基本思路都是利用每个向量之间的“距离” ----空间中的欧式距离或者曼哈顿距离，从远近判断是否从属与同一类别。

欧式距离(Euclidean Metric)：以空间为基准的两点之间的最短距离



曼哈顿距离：



切比雪夫距离：



余弦距离： 使用向量空间中两个向量夹角的余弦值作为衡量两个个体差异的大小的度量。

## 回归

回归：regression。就是一种“由果索因”的过程，是一种归纳的思想。当看到大量的事实所呈现的样态，可以推断出原因或客观蕴含的关系；当大量观测而来的向量是某种样态，可以设计一种假说来描述它们之间蕴含的关系。回归常用有：线性回归和非线性回归。

1. 线性回归：在观测和归纳样本的过程中认为向量和最终的函数值呈现线性的关系。



其中和分别是和的矩阵。则变为一个n元一次的方程，对矩阵的内容和偏置的内容求出一个最合适的解，而所谓的合适是得到一个全局范围内由映射得到的y与真实观测值y的差距加和：

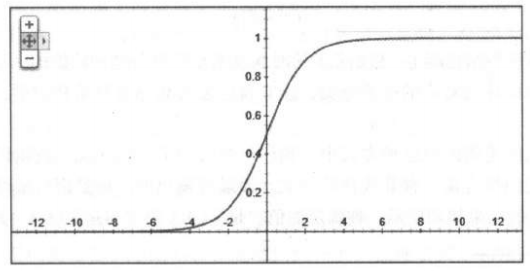


1. 非线性回归：在机器学习领域中，应用最多的属于逻辑回归。逻辑回归在观察者假设的前提是y只有两种值：1和0.



其中等同于线性回归中的。如果设，那么有





逻辑回归的表达式的定义本源来自于伯努利分布.

## 分类

分类的训练过程和回归的训练过程一样，都是极为套路化的程序。

1. 输入样本和分类标签；
2. 建立映射假说的某个的模型；
3. 求解出全局的损失函数和待定系数的映射关系，
4. 通过迭代优化逐步降低，最终找到一个能使召回率和精确率满足当前场景的要求。尤其指在验证数据集上的表现。

# 深度学习是什么

## 2.1 神经网络是什么

A. 深度学习 （Deep Learning）

B. 人工神经网络 (Artificial Neural Network , ANN)

### 2.1.1 神经元

神经元通常有两部分组成：线性模型和激励函数。

线性模型：输入是一个维向量。建立一个有个输入，而在处理过程中可以写成：



其中x为一个完整的特征向量,是一个的矩阵，表示的是一个权重的概念。为实数值。

### 2.1.2 激励函数

激励函数（Activation Function）。在一个神经元中，激励函数跟随在之后，用于加入非线性因素。

#### Sigmoid 函数

，或者



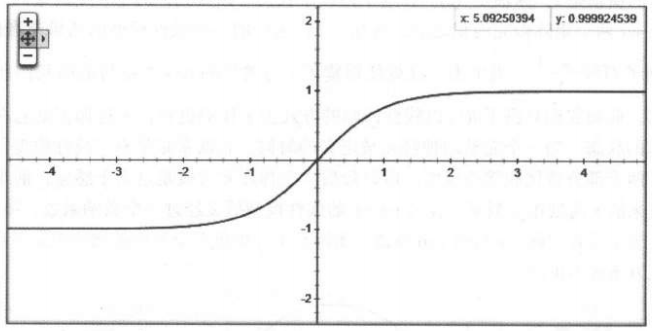
当一个完整的神经元被定义的时候，它通常是带有“线性模型”和“激励函数”两个部分首尾相接而成。前半部分接受外界进来的x向量作为刺激，经过的线性模型后又经过一个激励函数。通过这种方式引入非线性因素。

#### Tanh函数

又称为双曲正切函数



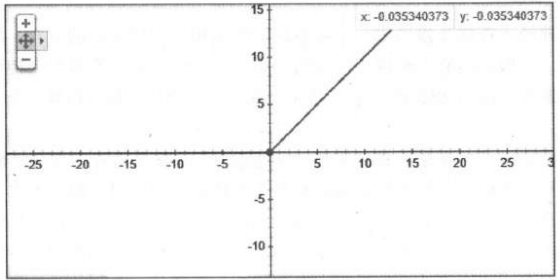
函数曲线如下：



Tanh函数和Sigmoid函数长相相近。只不过Tabh函数把输入值投射到-1到1上。其中“-1”表示完全不激活，“1”表示完全激活，中间其他值表示不同的激活程度。除了映射区域与Sigmoid不同之外，Sigmoid在之后就非常平缓的贴近0或1，Tanh在之后会平缓贴近-1或1。会在一定程度上影响收敛问题。

#### ReLU函数

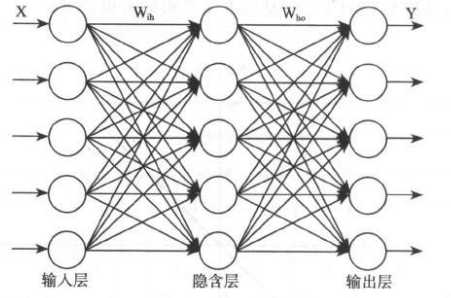
ReLU函数是目前大部分卷积神经网络CNN（Convolutional Neural Networks）中喜欢使用的激励函数，全名是rectified linear units



这个函数的形式为，在这个函数的原点左侧部分斜率为0，在右侧则是一条斜率为1的直线。

### 2.1.3神经网络

当多个神经元首位连接形成一个类似网络的结构来协同工作的时候，可以被称为神经网络。神经网络的结构通常分为：输入层(Input Layer)，隐藏层(Hidden Layer)，输出层(Output Layer)



## 2.2 深度神经网络

深度学习实际指的是基于深度神经网络(Deep Neural Networks ,DNN)的学习。

浅度学习的工作场景：

贝叶斯概率进行学习：使用统计的方法解释不同事件先验概率和后验概率的量化关系计算问题；

使用决策树归纳一个分类模型：用维度的引入把分类模型向信息熵降低的方向做引导，即统计过程；

使用支持向量机SVM做分类：寻找超平面来保证分类的置信度最大，也就是让样本点距离超平面尽可能的远，即统计过程；

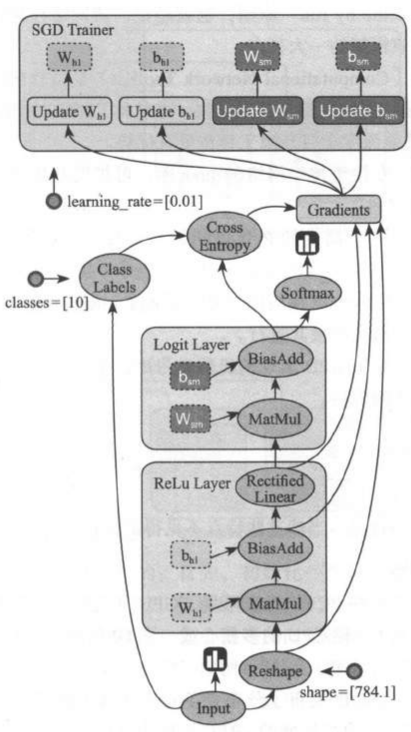
上述过程大部分依靠线性模型或者统计学概率模型给出清晰的物理含义解释，局限性也较为明显。

# 第三章 TensorFlow框架特性与安装

## 3.1 简介

TensorFlow是一个采用数据流图（data flow graphs），用于数值计算的开源软件库。节点（nodes）在图中表示数学操作，图中的线（edges）则表示在节点之间相互联系的多维数据数组，即张量（tensor）.

数据流图用节点和线的有向图来描述数学计算。“节点”一般用来表示施加的数学操作，但也可以表示数据输入（feed in）的起点/输出（push out）的终点，或者表示读取/写入持久变量（persistent variable）的终点。线表示节点之间的输入/输出关系。这些数据“线”可以运输“size可动态调整”的多维数据数组，即张量。



# 第四章 前馈神经网络

最简单朴素的神经网络：叫做前馈神经网络（feedforward neural network）。这种神经网络，各个神经元从输入层开始，接收前一级输入，并输出到下一级，直至输出层。整个网络中无反馈，可以用一个有向无环图（directed acyclic graph DAG）表示。

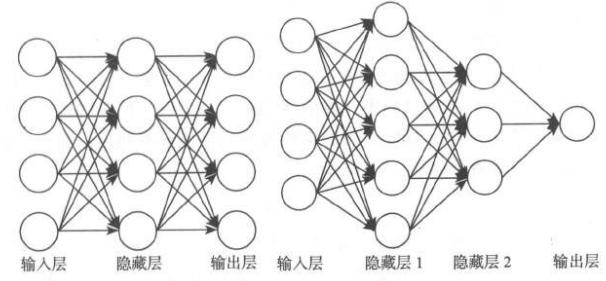
通常所说的前馈神经网络有两种：

***Back Propagation Networks: 反向传播网络（简称BP网络）***

***RBF Network：径向基函数神经网络***

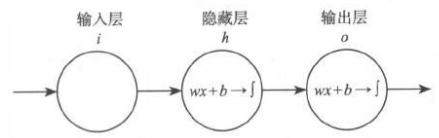
## 网络结构

BP网络是所有的神经网络中结构最为单纯的一种。



第一个神经网络，共两层，每层4个节点。

第二个神经网络，共三层，输出层只有一个神经元。



隐藏层和输出层这两层都是和。这个网络一旦输入了x和y，就可以开始训练过程了。

## 线性回归的训练

1 样本

基础数据对。

2 牛顿法

通过多次“试探性”的计算并比对与这个真实值之间的差距是否缩小来得到解。是一种以有限成本的“次优”取代无限成本的“最优”的思想。

## 神经网络的训练

整个套路如下：

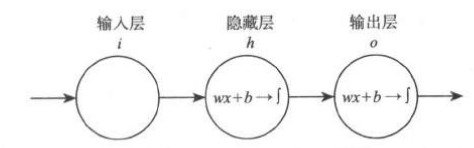
1. 准备样本

需要准备一定数量的用来训练的x向量，当然只要是输入到网络中作为训练样本的一定是向量化的。

1. 清洗处理

为了提高收敛效率等，需要对数据进行清理

1. 正式训练



根据网络中两个神经元的表达式描述：

，

将输入数据带入，于是又关系如下：

，

于是，由和带来的误差值便可以定义为：

而由n个训练数据共同带来的误差值就变成了



因为损失函数始终都是恒为非负数，通过不断的调整的值，这时候便需要通过求梯度的方法来寻找最快的收敛，寻找更合理的值来对进行更新：

,

,

求解过程如下：



可以使用求导技巧如下：









依次迭代求值。

# 卷积神经网络

## 6.1 与全连接网络的对比

每一个神经元节点的输入都来自于上一层的每一个神经元的输出。这种连接方式被称为全连接网络(Full Connect Network)。全连接网络的优势是每个输入维度的信息都会传播到其后任何一个节点中，会最大程度的让整个网络中的节点都不会“漏掉”这个维度所贡献的因素。缺点为：权重极多，收敛速度极其缓慢。

**卷积神经网络**(Convolutional Neural Network, CNN)。同样是一种前馈神经网络。

特点：

1. 卷积网络至少有一个卷积层，用于提取特征；
2. 卷积网络的卷积层通过权值共享的方式进行工作，大大减少权值w的数量，使得在训练中，在达到同样识别率的情况下收敛速度明显快于全连接BP网络。

卷积神经网络主要用来识别位移、缩放及其他形式扭曲不变性的二维图形。由于它的特征检测层通过训练数据进行学习，所以在使用卷积网络时，避免了显式的特征提取，而隐式地从训练数据中进行学习；再者由于同一特征映射面上的神经元权值相同，所以网络可以并行学习。

## 6.2 什么是卷积

**卷积**(Convolution)是一种函数的定义。它是通过两个函数和生成第三个函数的一种数学算子，表征函数与经过翻转和平移的重叠部分的面积。

卷积的数学定义为:



## 6.3 卷积核

## 6.4 卷积层其他参数

Padding（填充）：指用多少个像素单位来填充输入图像（向量）的边界；

Striding（步幅）：在卷积层工作的时候，表示为每次滑动的单位；

## 6.5 池化层

池化层（Pooling Layer）。池化层的作用实际上对Feature Map所做的数据处理又进行了一次所谓的池化处理。

常见的池化层处理有两种方式：

Max Pooling：最大化处理；

Mean Pooling：平均化处理；

还有例如Chunk Max Pooling; Stochastic Pooling

目的：

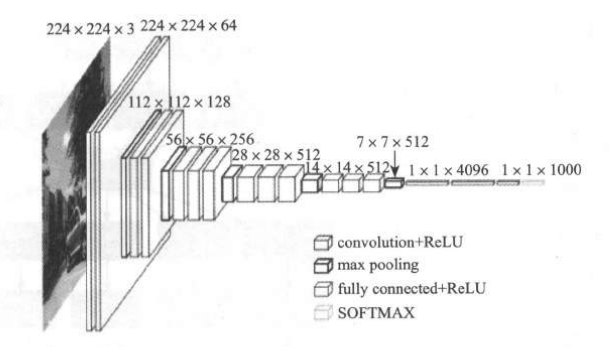
1. 它又进行了一次特征提取，所以肯定是能够减少下一层数据的处理量；
2. 由于这个特征的提取，能够有更大的可能性进一步获取更为抽象的信息，从而防止过拟合，或者说提高一定的泛化性；
3. 由于这种抽象性，能够对输入的微小变化产生更大的容忍，也就是保持其不变性。这里的容忍包括图形的少量平移，旋转以及缩放等变化。

池化层在CNN网络中不是有一个必须的组件。视情况而出现。

## 6.6 典型CNN网络

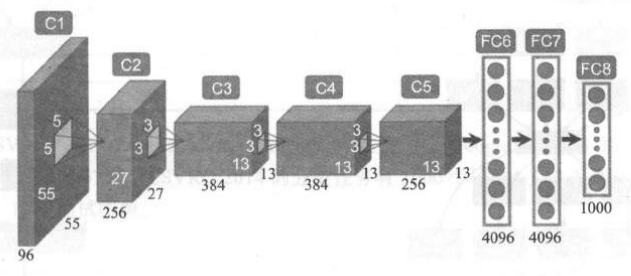
一个比较经典的卷积网络。

由Visual Geometry Group 在2015年发布的卷积网络。Visual Geometry Group隶属于牛津大学。该网络命名为VGG-16，VGG是Visual Geometry Group的缩写，16指的是其中有16个带有参数的网络层。VGG-16是一个带有完整的卷积层、池化层、全连接层的神经网络。



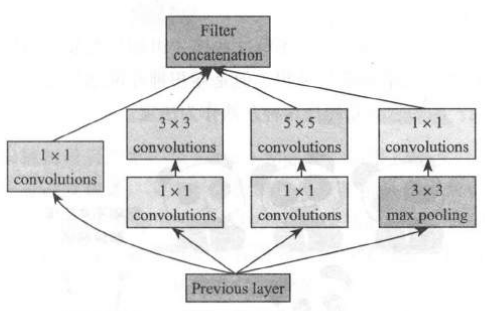
上图为VGG-16的流程概要。一张图从左侧输入后，经过64个不同的的卷积核，每次Stride=1的挪移步长，生成了64个小尺寸的“图片”（Feature Map），把这64个小尺寸的图片“拼接”在一起，又通过64个的卷积核生成了后一层的Feature Map，然后经过一个Max Pooling层来做池化 。之后这个Max Pooling中的信息又被当做一个“图片”向后输入，通过128个的卷积核进一步提取特征。在这个过程中卷积核的数量实在逐步增多的，64个、128个、256个、512个，而每个卷积核提取过后的信息占用的空间越来越少。在最后有一个1000个节点的SOFTMAX层来做分类使用。训练的过程中同样把一种损失函数描述成为各个卷积核中权重的函数，然后进行凸优化来找到极值点。

如下图为VGG-19



2014年发表的AlexNet，前面3个卷积层，2个池化层，2个全连接层，一个1000个节点的SOFTMAX层。

2014年发表的GoogleNet，深度22层，引入了一个新的结构Inception



在这个结构中，可以看到前面输入过来的向量会在这一层上展开成为多个不同卷积核处理的并列结构，这样可以在一定程度上加大下一层输入的信息量。其中和的卷积层会大大提升分类识别的抽象能力（在一定的程度上，卷积核越大这个特点就相对越明显）。Inception结构的引入大大增加了网络的宽度和深度，使得网络的信息容纳能力变强，性能要大大的提高。

## 6.7 图片识别

人类在认知的过程中，有一种对于细微变化的免疫作用，或者说是对细微变化的不敏感性。

卷积网络中所使用的卷积核在对输入向量进行特征提取的过程是一个把高纬向量映射成为低纬向量的过程，也是一种有损压缩，这种特点来自于卷积核的工作原理——卷积核提供了一种前一层输入向量（样本或Feature Map）到后一层输出向量（Feature Map）的刺激能力。而在卷积核滑动的过程中，可以发现个别向量值的变化对于刺激的结果影响极为有限。通过量化的手段去表示敏感程度的过程，而这个量化的程度是通过训练获得。

当一副图被卷积核扫描后产生一张Feature Map A，如果改变这张图的少量像素，无论是颜色还是线条或者涂鸦，少量的旋转等，重新经过卷积核扫描产生的Feature Map B，向量值的改变少。差异不大的部分都会被抹去，进而对后面层的网络产生近似的刺激。

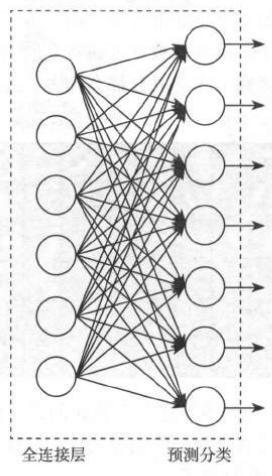
这种效果意味着：

1. 少量的噪声、讹误对深度卷积神经网络的分类影响是非常有限的，具有更强大的容错能力
2. 由于卷积神经网络的这样一种特性，也使得其泛化性更好，因为即便分类对象跟训练样本库的特征有一定的差异，这种“刺激化”处理的结果会使得它们在较深的网络中有类似的刺激结果。

## 6.8 输出层激励函数——SOFTMAX

### 6.8.1 SOFTMAX

该激励函数的输出层：



看起来这种方式跟普通的全连接层没啥区别，但是激励函数的形式却大不一样。

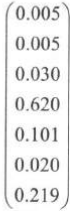
后面一层作为预测分类的输出节点，每一个节点就代表一个分类，那么这7个节点就最多能够表示7个分类的模型。任何一个节点的激励函数都是



其中 节点的下标次序，而，也就是说这是一个线性分类模型的输出作为自然常数e的指数。最后一层有这样一个特性：



也就是说最后一层每个节点的输出值加和是1。一张待分类的图片输出的时候，会产生类似的形式：



### 6.8.2 交叉熵

SOFTMAX这种激励函数使用的损失函数，叫做交叉熵（cross entropy）

信息熵:，其中有

解释：前面的当成一个向量，若干个把每个可能项产生的概率乘以该可能性的信息量，然后各项做加和。

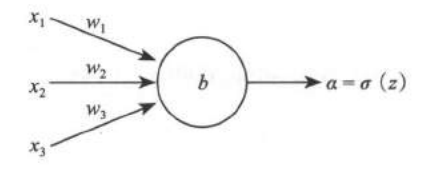
消息产生的种类越多，概率越均等则信息熵越大；反之，消息产生的种类越单一，概率产生越偏向其中某一个信息，那么熵值就越小。最极端情况：只有一种消息，而且概率100%的情况，这种时候熵值为0

交叉熵：









对于最后一层SOFTMAX的每一个输出节点来说，都是上面这个样子，有多个x输入的向量，有节点上的w矩阵做内积，加上偏置b，通过Sigmoid函数输出一个0到1之间的概率值。

Sigmoid函数是一个有趣的函数，它对自变量z求导会到的自身与的乘积









对式子的理解：



在整个训练进行的过程中，是把样本向量和标签向量同时放入模型。在放入的时候自然样本没有特殊理由的情况下都是会将期望的分类维度设置为1，其余的维度设置为0，用这样的形式来标记分类标签向量,而在拟合过程中会有这样的一个事实，当为0的时候，由于失效而这一项是有效的，所以的大小就是损失值了。因为本来就不应该分为这一类，但是却成了分作这一类的概率，是负数。

所谓的交叉熵：就是一种当前拟合出来的模型分类所产生的信息熵和这种“客观上”的信息熵的差距。

举个例子：在一个网络空间中，有10000张图片，分成4类，无论具体情况如何，从统计上都会产生一个信息熵，也就是随便来一张图片，即便不通过这个网络，也会有一个先验概率。这个先验概率来自于这些图片“天然”分布的一个统计比例。这种针对“客观世界”的统计产生的分类的“信息熵”就是分类目标的熵，而在拟合过程中产生的熵实际上跟这个熵值是有差距的，这个差距就是在使用交叉熵损失函数的情况下所定义的熵的差距，那么优化方向就是向着调整待定系数减少“熵差”的方向去运动。而这个差值越大，导数就越大，学习的效率就越高。通过训练不断调整众多卷积核中w的大小，来决定一个合适的特征提取的量化值，这就是卷积网络训练的基本原理。

**独热编码**：（one-hotencoding）。一种用一个向量的每一个维度来标识一种性质有无的方式。例如：性别用向量做标识，1标识为男性，0标识为女性。作为普通向量的输入可以是

1. 和[1]，但是使用独热编码的方式会表示为[1,0]和[0,1]

## 6.10 小结

卷积的网络最重要的核心就是卷积核，卷积神经网络的特征就是卷积层，进而带来的好处就是收敛速度快而且泛化能力会显得比较好。

# 综合问题

## 7.1 并行计算

## 7.2 随机梯度下降

梯度下降算法(Gradient decent)。这种算法，将待定权值w各项之间的系数先算出来，再沿着梯度下降的方向移动各个w的每个维度值。例如，在过程中求偏导数，其损失函数，计算量极大，这时候使用随机梯度下降算法(Stochastic gradient decent)

随机梯度下降算法：可以认为所有的这些作为训练样本的输入向量，从中提取一部分样本来，这些样本中的特征已经在一定程度上代表整个完整训练集的特征。

TensorFlow中是通过制定Batch的Size来规定每次被随机选择参与归纳的样本数量。

## 7.3 梯度消失问题

梯度消失(Vanishing Gradient)。现象为：训练不再收敛——Loss过早的不再下降，而精准度也过早的不再提高。

## 7.4 归一化

归一化(Normalization)。计算机体系中，数字没有量纲或者单位，只有一个具体的浮点数或者定点数。数据归一化就是这样的一个操作过程——把数据的大小分布压缩或框定在一个比例协调的范围内。

线性函数归一化(min-max scaling)，还有0均值标准化(z-score standardization)。目的都是为了让各维度的数据分布经过“拉伸”投射到一个相近的尺度范围。

线性函数归一化：



每一个数值经过归一化的投射，就都变成0~1之间的数值，这个值表示自己再整个样本中该维度所处的位置比例。

批归一化(Batch Normalization)

## 7.5参数初始化问题

业内比较认可的说法是把整个网络中所有的w初始化成以0为均值，以一个很小的值为标准差的正态分布效果会比较好。

最常见的是0为均值，1为方差的分布来随机初始化，记作，对应代码：



也有其他多种初始化方法，不过大多数都是高斯分布（正态分布）的类似或者变种的方式

## 7.6 正则化

正则化(Regulization)。正则化是找到更为简洁的描述方式的量化过程。使用正则化改造损失函数为：

 , (L1正则化项)

上式为正则化的损失函数(cost的简写)。之前提到的损失函数只有前半部分，后半部分为加上了的正则化项，能够在一定程度上避免过拟合。从学术上讲，前半部分的损失函数称为“经验风险”，后半部分称为（结构风险）。

经验风险：指由于拟合结果和样本标签之间的残值总和所产生的这种经验性差距所带来的风险——比较差距越大模型拟合失效的可能性就越大，“欠拟合”的风险；

结构风险：希望这种描述能够简洁来保证其泛化性的良好，所以加入一个权重因子。该因子把整个模型中所有的权重的绝对值加起来除以样本数量。也是一个权重（可称为正则化系数，或惩罚系数或叫学习率）。用以表示对这个部分有多“重视”，如果重视结构风险，那就加大，迫使整个损失函数向着权值减少的方向快速运动。

 ,(L2正则化项)

L1 与 L2区别在于前者取得是绝对值项，后者是平方后才做的加和。

L1的正则项导数如下：







L2的正则项导数如下：









## 7.7 其他超参数

超参数（hyper parameter）：通常指的是那些在机器学习算法训练的步骤开始之前设定的一些参数值，这些参数通常是没办法通过算法本身来学会的。

例如K-Means算法中的簇数N，深度学习中涉及到的

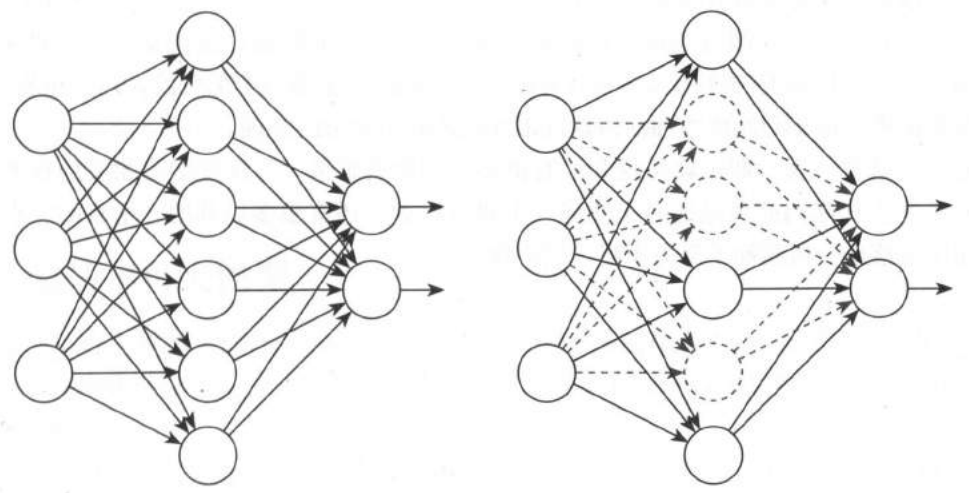
## 7.8 不唯一的模型

## 7.9 Drop Out

Drop out是在深度学习训练中较为常用的方法。主要也是用于克服过拟合现象。

由于整个网络的所蕴含的VC维非常高，导致它的记忆能力非常强，很多个体上没有泛化能力的特征也会被记忆。使得整个网络的参数过多过大。

Drop out：在一轮训练阶段丢弃一部分网络节点，比如可以在其中的某些层上临时关闭一些节点，让它们不输入也不输出，这样相当于整个网络的结构发生了改变。而在下一轮训练的过程中再选择性临时关闭一些节点，原则上都是随机性的。



# 循环神经网络

循环神经网络，Recurrent Neural Networks, RNN。循环神经网络与前馈神经网络最大的不同是有一些“记忆暂存”功能，可以把过去输入的内容所产生的远期影响量化后与当前时间输入的内容一起反应到网络中去参与训练。

尤其是在自然语言处理(Natural Language Processing，NLP)应用广泛。

## 8.1 隐马尔科夫模型

隐马尔可夫模型， Hidden Markov Model, HMM。

马尔科夫链：在给定当前知识或信息的情况下，观察对象过去的历史状态对于将来的预测来说预测是无关的。