**机器学习考试题目答案**

# 简描述机器学习概念？

# Tom Mitchell：“对于某类任务T和性能度量P，如果一个计算机程序在T上以P衡量的性能随着经验E而自我完善，那么我们称这个计算机程序在从经验E学习。”

我们遇到的大部分事情一般包括分类问题与回归问题。如房价的预测，股价的预测等属于分类问题。一般的处理过程是：首先，1）获取数据；2）提取最能体现数据的特征；3）利用算法建模；4）将建立的模型用于预测。如人脸识别系统，首先我们获取到一堆人脸照片，首先，对数据进行预处理，然后提取人脸特征，最后用算法如SVM或者NN等。这样，我们就建立了一个人脸识别系统，当输入一张人脸，我们就知道这张面孔是否在系统中。这就是机器学习的整个流程，其次还包括寻找最优参数等。

**机器学习主要分为：**

监督学习：数据集是有标签的，大部分机器学习模型都属于这一类别，包括线性分类器、支持向量机等等；

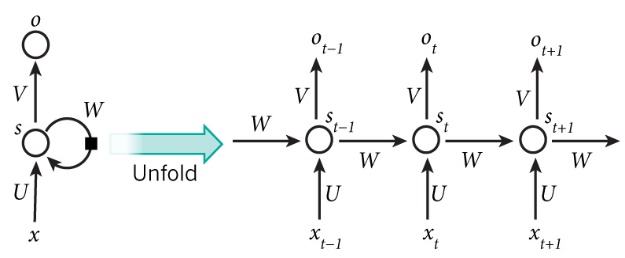
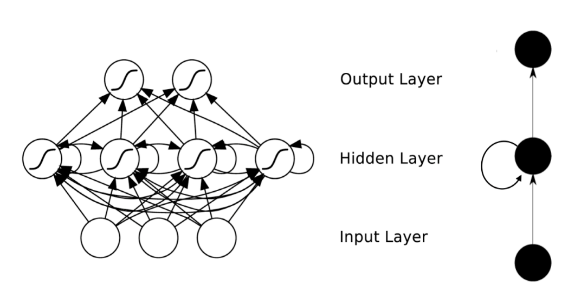
无监督学习：跟监督学习相反，数据集是完全没有标签的，主要的依据是相似的样本在数据空间中一般距离是相近的，这样就能通过距离的计算把样本分类，这样就完全不需要label，比如著名的kmeans算法就是无监督学习应用最广泛的算法；

半监督学习：半监督学习一般针对的问题是数据量超级大但是有标签数据很少或者说标签数据的获取很难很贵的情况，训练的时候有一部分是有标签的而有一部分是没有的；

强化学习：一直激励学习的方式，通过激励函数来让模型不断根据遇到的情况做出调整；

# 循环神经网络的基本原理？

RNNs的目的是用来处理序列数据。在传统的神经网络模型中，是从输入层到隐含层再到输出层，层与层之间是全连接的，每层之间的节点是无连接的。但是这种普通的神经网络对于很多问题却无能无力。例如，你要预测句子的下一个单词是什么，一般需要用到前面的单词，因为一个句子中前后单词并不是独立的。RNNs之所以称为循环神经网路，即一个序列当前的输出与前面的输出也有关。具体的表现形式为网络会对前面的信息进行记忆并应用于当前输出的计算中，即隐藏层之间的节点不再无连接而是有连接的，并且隐藏层的输入不仅包括输入层的输出还包括上一时刻隐藏层的输出。理论上，RNNs能够对任何长度的序列数据进行处理。但是在实践中，为了降低复杂性往往假设当前的状态只与前面的几个状态相关，下图便是一个典型的RNNs： （注：下面两张图片仅为了帮助你理解RNN，考试不必作答）



RNNs包含输入单元(Input units)，输入集标记为{x0,x1,...,xt,xt+1,...}，而输出单元(Output units)的输出集则被标记为{y0,y1,...,yt,yt+1.,..}。RNNs还包含隐藏单元(Hidden units)，我们将其输出集标记为{s0,s1,...,st,st+1,...}，这些隐藏单元完成了最为主要的工作。你会发现，在图中：在RNN中有一条单向流动的信息流是从输入单元到达隐藏单元的，与此同时另一条单向流动的信息流从隐藏单元到达输出单元。在某些情况下，RNNs会打破后者的限制，引导信息从输出单元返回隐藏单元，这些被称为“Back Projections”，并且隐藏层的输入还包括上一隐藏层的状态，即隐藏层内的节点可以自连也可以互连。   
 上图将循环神经网络进行展开成一个全神经网络。例如，对一个包含5个单词的语句，那么展开的网络便是一个五层的神经网络，每一层代表一个单词。对于该网络的计算过程如下：

* xt表示第t,t=1,2,3...步(step)的输入。比如，x1为第二个词的one-hot向量(根据上图，x0为第一个词)；
* st为隐藏层的第t步的状态，它是网络的记忆单元。 st根据当前输入层的输出与上一步隐藏层的状态进行计算。st=f(Uxt+Wst−1)，其中f一般是非线性的激活函数，如[tanh](https://reference.wolfram.com/language/ref/Tanh.html" \t "_blank)或[ReLU](https://en.wikipedia.org/wiki/Rectifier_%28neural_networks%29" \t "_blank)，在计算s0时，即第一个单词的隐藏层状态，需要用到s−1，但是其并不存在，在实现中一般置为0向量；
* ot是第t步的输出，如下个单词的向量表示，ot=softmax(Vst).   
  需要注意的是：
* 你可以认为隐藏层状态st是网络的记忆单元. st包含了前面所有步的隐藏层状态。而输出层

的输出ot只与当前步的st有关，在实践中，为了降低网络的复杂度，往往st只包含前面若干步而不是所有步的隐藏层状态；

* 在传统神经网络中，每一个网络层的参数是不共享的。而在RNNs中，每输入一步，每一层各自都共享参数U,V,W。其反应者RNNs中的每一步都在做相同的事，只是输入不同，因此大大地降低了网络中需要学习的参数；这里并没有说清楚，解释一下，传统神经网络的参数是不共享的，并不是表示对于每个输入有不同的参数，而是将RNN是进行展开，这样变成了多层的网络，如果这是一个多层的传统神经网络，那么xt到st之间的U矩阵与xt+1到st+1之间的U是不同的，而RNNs中的却是一样的，同理对于s与s层之间的W、s层与o层之间的V也是一样的。
* 上图中每一步都会有输出，但是每一步都要有输出并不是必须的。比如，我们需要预测一条语句所表达的情绪，我们仅仅需要关系最后一个单词输入后的输出，而不需要知道每个单词输入后的输出。同理，每步都需要输入也不是必须的。RNNs的关键之处在于隐藏层，隐藏层能够捕捉序列的信息。

# 卷积神经网络的基本原理？

受Hubel和Wiesel对猫视觉皮层电生理研究启发，有人提出卷积神经网络（CNN），Yann Lecun 最早将CNN用于手写数字识别并一直保持了其在该问题的霸主地位。近年来卷积神经网络在多个方向持续发力，在语音识别、人脸识别、通用物体识别、运动分析、自然语言处理甚至脑电波分析方面均有突破。

卷积神经网络与普通神经网络的区别在于，卷积神经网络包含了一个由卷积层和子采样层构成的特征抽取器。在卷积神经网络的卷积层中，一个神经元只与部分邻层神经元连接。在CNN的一个卷积层中，通常包含若干个特征平面(featureMap)，每个特征平面由一些矩形排列的神经元组成，同一特征平面的神经元共享权值，这里共享的权值就是卷积核。卷积核一般以随机小数矩阵的形式初始化，在网络的训练过程中卷积核将学习得到合理的权值。共享权值（卷积核）带来的直接好处是减少网络各层之间的连接，同时又降低了过拟合的风险。子采样也叫做池化（pooling），通常有均值子采样（mean pooling）和最大值子采样（max pooling）两种形式。子采样可以看作一种特殊的卷积过程。卷积和子采样大大简化了模型复杂度，减少了模型的参数。

下面介绍几个重要操作：

**局部感受野**

卷积神经网络有两种神器可以降低参数数目，第一种神器叫做局部感知野。一般认为人对外界的认知是从局部到全局的，而图像的空间联系也是局部的像素联系较为紧密，而距离较远的像素相关性则较弱。因而，每个神经元其实没有必要对全局图像进行感知，只需要对局部进行感知，然后在更高层将局部的信息综合起来就得到了全局的信息。网络部分连通的思想，也是受启发于生物学里面的视觉系统结构。视觉皮层的神经元就是局部接受信息的（即这些神经元只响应某些特定区域的刺激）。

在一个全连接网络中假如有1000000个神经元，则参数共有10^12个，假如每个神经元只和10×10个像素值相连，那么权值数据为1000000×100个参数，减少为原来的万分之一。而那10×10个像素值对应的10×10个参数，其实就相当于卷积操作。

**权值共享**

 但其实这样的话参数仍然过多，那么就启动第二级神器，即权值共享。在上面的局部连接中，每个神经元都对应100个参数，一共1000000个神经元，如果这1000000个神经元的100个参数都是相等的，那么参数数目就变为100了。

 怎么理解权值共享呢？我们可以这100个参数（也就是卷积操作）看成是提取特征的方式，该方式与位置无关。这其中隐含的原理则是：图像的一部分的统计特性与其他部分是一样的。这也意味着我们在这一部分学习的特征也能用在另一部分上，所以对于这个图像上的所有位置，我们都能使用同样的学习特征。

 更直观一些，当从一个大尺寸图像中随机选取一小块，比如说 8x8 作为样本，并且从这个小块样本中学习到了一些特征，这时我们可以把从这个 8x8 样本中学习到的特征作为探测器，应用到这个图像的任意地方中去。特别是，我们可以用从 8x8 样本中所学习到的特征跟原本的大尺寸图像作卷积，从而对这个大尺寸图像上的任一位置获得一个不同特征的激活值。

**多卷积核**

上面所述只有100个参数时，表明只有1个10\*10的卷积核，显然，特征提取是不充分的，我们可以添加多个卷积核，比如32个卷积核，可以学习32种特征。

**Down-pooling**

在通过卷积获得了特征 (features) 之后，下一步我们希望利用这些特征去做分类。理论上讲，人们可以用所有提取得到的特征去训练分类器，例如 softmax 分类器，但这样做面临计算量的挑战。例如：对于一个 96X96 像素的图像，假设我们已经学习得到了400个定义在8X8输入上的特征，每一个特征和图像卷积都会得到一个 (96 − 8 + 1) × (96 − 8 + 1) = 7921 维的卷积特征，由于有 400 个特征，所以每个样例 (example) 都会得到一个 7921 × 400 = 3,168,400 维的卷积特征向量。学习一个拥有超过 3 百万特征输入的分类器十分不便，并且容易出现过拟合 (over-fitting)。

为了解决这个问题，首先回忆一下，我们之所以决定使用卷积后的特征是因为图像具有一种“静态性”的属性，这也就意味着在一个图像区域有用的特征极有可能在另一个区域同样适用。因此，为了描述大的图像，一个很自然的想法就是对不同位置的特征进行聚合统计，例如，人们可以计算图像一个区域上的某个特定特征的平均值 (或最大值)。这些概要统计特征不仅具有低得多的维度 (相比使用所有提取得到的特征)，同时还会改善结果(不容易过拟合)。这种聚合的操作就叫做池化 (pooling)，有时也称为平均池化或者最大池化 (取决于计算池化的方法)。

子采样有两种形式，一种是均值子采样（mean-pooling），一种是最大值子采样（max-pooling）。两种子采样看成特殊的卷积过程：

 (1)均值子采样的卷积核中每个权重都是0.25，卷积核在原图inputX上的滑动的步长为2。均值子采样的效果相当于把原图模糊缩减至原来的1/4。

 (2)最大值子采样的卷积核中各权重值中只有一个为1，其余均为0，卷积核中为1的位置对应inputX被卷积核覆盖部分值最大的位置。卷积核在原图inputX上的滑动步长为2。最大值子采样的效果是把原图缩减至原来的1/4，并保留每个2\*2区域的最强输入。

# 深度神经网络（循环神经网络或卷积神经网络）训练主要难点在哪里？如何解决？

**1 、神经网络的难点**

使用层数较深的神经网络会遇到许多困难，比如：容易过拟合、参数难以调试、梯度弥漫等，针对这些问题有很多trick解决。

**2、过拟合**

过拟合是机器学习中经常遇到的问题，它是指模型预测准确率在训练集上升高，但是在测试集上反而下降了，这意味着泛化性不好，模型只是记忆了当前数据的特征，不具备推广能力。

Hinton教授团队提出了一个思路简单但非常有效的方法：Dropout。其思路为：在训练时，将神经网络某一层的输出节点数据随机丢弃一部分，这样做法的实质等于创造出了很多新的随机样本，通过增大样本量、减少特征数量来防止过拟合。

PS:在测试时，不要Dropout。

**3、参数难以调试**

尤其是SGD的参数，对SGD设置不同的学习速率，最后得到的结果可能差异巨大，因为不同的学习速率可能导致神经网络落入截然不同的局部最优之中。

因此，有很多像Adagradm,Adam，Adadelta等自适应的方法可以减轻调试参数的压力，对于这些优化算法，通常我们使用它们默认的参数就可取得较好的结果。

**4、梯度弥散**

在ReLU激活函数出现之前，都是采用Sigmoid激活函数的。但是当网络层数较多时，Sigmoid函数在反向传播中梯度值会逐渐减小，经过多层的传递后会呈指数级急剧减小。

ReLU能够完美地解决梯度弥散问题，它是一个非常简单的非线性函数y=max(0,x)，经过多层的反向传播，梯度依旧不会大幅减小，该函数从正面解决了梯度弥散的问题，而不需要通过无监督的逐层训练初始化权重来绕行。

ReLU相比于Sigmoid函数的主要变化有如下3点：

（1）单侧抑制

（2）相对宽阔的兴奋边界

（3）稀疏激活性

# 在工业应用场景中，面对一个要解决的问题（分类、回归或者结构化预测），在给出解决方案前，你会考虑哪些因素？

**针对数据：**

* + 1. 该领域提供的历史数据集容量大小
    2. 相关数据的可区分性
    3. 相关数据是否为时序相关的
    4. 数据的质量如何（有无噪声数据或者数值缺失情况）
    5. 给定历史数据中是否存在私密信息
    6. 数据是否符合某些已知的分布

**针对应用场合：**

1. 考虑项目的背景，分析希望算法的准确率高还是召回率高
2. 考虑目前的设备情况，尽量选择设备能够承受的计算复杂算法
3. 查阅文献，是否有应用于相关领域的较好的算法

# 如果实验数据集中部分样本带有噪声，部分数值缺失，你如何处理？

对于本题目的两个内容属于机器学习中的数据清洗部分，在数据收集的过程中，由于人为失误或者机器的异常导致收集到的数据集中参杂不完整数据，以上两类数据都属于不完整数据，因此在实验正式开始前需要对数据集进行清洗，以去掉上述的不完整数据。针对上述的两个问题，具体的解决方案如下所述：

**一、数据集中部分样本带有噪声**

对于包含噪声的数据进行数据清洗，主要包含以下两步：

（1）发现那些数据是噪声

（2）对已发现的噪声数据进行修复

* 1. **发现噪声数据的方法：**

**1.简单的统计分析**

拿到数据后可以对数据进行一个简单的描述性统计分析，譬如最大最小值可以用来判断这个变量

的取值是否超过了合理的范围，如客户的年龄为-20岁或200岁，显然是不合常理的，为异常值。

**2.3∂原则**

　　如果数据服从正态分布，在3∂原则下，异常值为一组测定值中与平均值的偏差超过3倍标准差的值。如果数据服从正态分布，距离平均值3∂之外的值出现的概率为P(|x-u| > 3∂) <= 0.003，属于极个别的小概率事件。如果数据不服从正态分布，也可以用远离平均值的多少倍标准差来描述。

**3.箱型图分析**

　　箱型图提供了识别异常值的一个标准：如果一个值小于QL01.5IQR或大于OU-1.5IQR的值，则被称为异常值。QL为下四分位数，表示全部观察值中有四分之一的数据取值比它小；QU为上四分位数，表示全部观察值中有四分之一的数据取值比它大；IQR为四分位数间距，是上四分位数QU与下四分位数QL的差值，包含了全部观察值的一半。箱型图判断异常值的方法以四分位数和四分位距为基础，四分位数具有鲁棒性：25%的数据可以变得任意远并且不会干扰四分位数，所以异常值不能对这个标准施加影响。因此箱型图识别异常值比较客观，在识别异常值时有一定的优越性。

**4.基于模型检测**

　　首先建立一个数据模型，异常是那些同模型不能完美拟合的对象；如果模型是簇的集合，则异常是不显著属于任何簇的对象；在使用回归模型时，异常是相对远离预测值的对象

优缺点：1.有坚实的统计学理论基础，当存在充分的数据和所用的检验类型的知识时，这些检验可能非常有效；2.对于多元数据，可用的选择少一些，并且对于高维数据，这些检测可能性很差。

**5.基于距离**

　　通常可以在对象之间定义邻近性度量，异常对象是那些远离其他对象的对象

优缺点：1.简单；2.缺点：基于邻近度的方法需要O(m2)时间，大数据集不适用；3.该方法对参数的选择也是敏感的；4.不能处理具有不同密度区域的数据集，因为它使用全局阈值，不能考虑这种密度的变化。

**6.基于密度**

　　当一个点的局部密度显著低于它的大部分近邻时才将其分类为离群点。适合非均匀分布的数据。

优缺点：1.给出了对象是离群点的定量度量，并且即使数据具有不同的区域也能够很好的处理；2.与基于距离的方法一样，这些方法必然具有O(m2)的时间复杂度。对于低维数据使用特定的数据结构可以达到O(mlogm)；3.参数选择困难。虽然算法通过观察不同的k值，取得最大离群点得分来处理该问题，但是，仍然需要选择这些值的上下界。

**7.基于聚类**

　　基于聚类的离群点：一个对象是基于聚类的离群点，如果该对象不强属于任何簇。离群点对初始聚类的影响：如果通过聚类检测离群点，则由于离群点影响聚类，存在一个问题：结构是否有效。为了处理该问题，可以使用如下方法：对象聚类，删除离群点，对象再次聚类（这个不能保证产生最优结果）。

优缺点：1.基于线性和接近线性复杂度（k均值）的聚类技术来发现离群点可能是高度有效的；2.簇的定义通常是离群点的补，因此可能同时发现簇和离群点；3.产生的离群点集和它们的得分可能非常依赖所用的簇的个数和数据中离群点的存在性；4.聚类算法产生的簇的质量对该算法产生的离群点的质量影响非常大。

**数据集中的异常点的处理方法：**

1.删除异常值----明显看出是异常且数量较少可以直接删除

2.不处理---如果算法对异常值不敏感则可以不处理，但如果算法对异常值敏感，则最好不要用，如基于距离计算的一些算法，包括kmeans，knn之类的。

3.平均值替代----损失信息小，简单高效。

4.视为缺失值----可以按照处理缺失值的方法来处理

**二．样本部分数据缺失**

1. 直接删除----适合缺失值数量较小，并且是随机出现的，删除它们对整体数据影响不大的情况

2. 使用一个全局常量填充---譬如将缺失值用“Unknown”等填充，但是效果不一定好，因为算法可能会把它识别为一个新的类别，一般很少用

3. 使用均值或中位数代替----优点：不会减少样本信息，处理简单。缺点：当缺失数据不是随机数据时会产生偏差.对于正常分布的数据可以使用均值代替，如果数据是倾斜的，使用中位数可能更好。

4. 插补法

　　1）随机插补法----从总体中随机抽取某个样本代替缺失样本

　　2）多重插补法----通过变量之间的关系对缺失数据进行预测，利用蒙特卡洛方法生成多个完整的数据集，在对这些数据集进行分析，最后对分析结果进行汇总处理

　　3）热平台插补----指在非缺失数据集中找到一个与缺失值所在样本相似的样本（匹配样本），利用其中的观测值对缺失值进行插补。

　　　　优点：简单易行，准去率较高

　　　　缺点：变量数量较多时，通常很难找到与需要插补样本完全相同的样本。但我们可以按照某些变量将数据分层，在层中对缺失值使用均值插补

　　4)拉格朗日差值法和牛顿插值法（简单高效，数值分析里的内容）

5. 建模法

可以用回归、使用贝叶斯形式化方法的基于推理的工具或决策树归纳确定。例如，利用数据集中其他数据的属性，可以构造一棵判定树，来预测缺失值的值。

　　以上方法各有优缺点，具体情况要根据实际数据分分布情况、倾斜程度、缺失值所占比例等等来选择方法。一般而言，建模法是比较常用的方法，它根据已有的值来预测缺失值，准确率更高。