1. **神经网络**

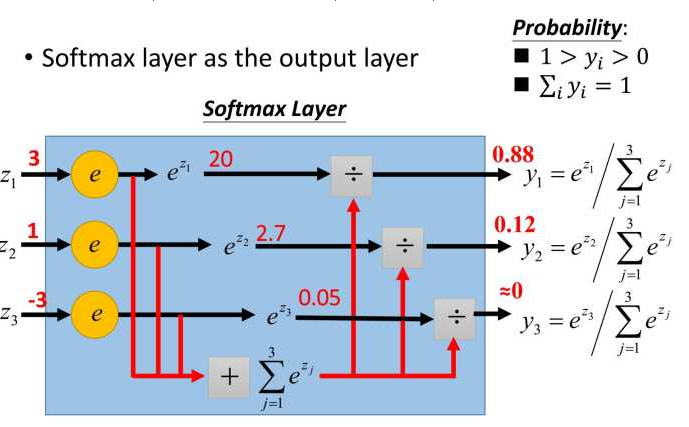
**1.1 Softmax**

<https://blog.csdn.net/bitcarmanlee/article/details/82320853>

<http://www.cnblogs.com/yjmyzz/p/7822990.html>

在多分类问题中，如果希望得到目标对象属于每个类别的概率，而我们通过分类器得到的一般是一系列数值，这里就可以通过Softmax将它们转换成0-1直接的概率，并且概率和为1

Softmax的计算：



Softmax分类和k个二元分类的比较

如果希望将样本分为k类，选择Softmax还是k个二元分类器，取决于是否类别之间是否互斥。如果互斥，则应该选择Softmax，如果不互斥，则可以选择k个二元分类器。

**1.2 极大似然估计(maximum-likelihood)**

<https://blog.csdn.net/zengxiantao1994/article/details/72787849>

先看贝叶斯分类：



其中，为先验概率，表示某种类别的概率分布。为类条件概率，表示属于某个类别的前提下，事件发生的概率。为后验概率，表示事件发生的前提下，属于某个类别的概率。

例子：

已知：在夏季，某公园男性穿凉鞋的概率为1/2，女性穿凉鞋的概率为2/3，并且该公园中男女比例通常为2:1，问题：若你在公园中随机遇到一个穿凉鞋的人，请问他的性别为男性或女性的概率分别为多少？

男性，为女性，为穿凉鞋。

先验概率：

，

类条件概率：

，

穿凉鞋的概率：



则：





在实际问题中，我们往往只能获取有限数目的样本，也无法知晓先验概率和类条件概率。

先验概率可以通过以下方式获取：1、每个样本所属的自然状态都是已知的（有监督学习）；2、依靠经验；3、用训练样本中各类出现的频率估计。

类条件概率：将估计类条件概率转换为参数估计，并采用极大似然估计来解决。

极大似然估计的重要前提：

1. 训练样本的分布能代表样本的真实分布。
2. 每个样本集中的样本都是所谓独立同分布的随机变量。
3. 有充分的训练样本。

极大似然估计：利用已知的样本结果，反推最有可能（最大概率）出现这种结果的参数值。

设样本集为：



我们来估计参数。

似然函数（likelihood function）：联合概率密度函数称为相对于的的似然函数。



其中，最后一个=之所以成立，是因为样本是独立同分布的。

如果能使似然函数最大，则就是最可能的值。



为了便于分析，定义对数似然函数：



对似然函数求导，极大似然估计量为：



极大似然估计的一般步骤：

1. 写出似然对数
2. 对似然对数求导
3. 解似然方程，求出。

极大似然估计的优点：

1. 简单易实现
2. 收敛性好，如果样本趋近总体，则收敛性会变好

极大似然估计的缺点：

1. 如果假设的概率模型正确，则结果会较好；假如模型出现偏差，则效果可能会比较差。
2. **深度神经网络**

**2.1 Keras**

1. 序列模型

from keras.models import Sequential

#Create the Sequential model

model = Sequential()

1. 层

**from** keras.models **import** Sequential

**from** keras.layers.core **import** Dense, Activation, Flatten

*#创建序列模型*

model = Sequential()

*#第一层 - 添加有128个节点的全连接层以及32个节点的输入层*

model.add(Dense(128, input\_dim=32))

*#第二层 - 添加 softmax 激活层*

model.add(Activation('softmax'))

*#第三层 - 添加全连接层*

model.add(Dense(10))

*#第四层 - 添加 Sigmoid 激活层*

model.add(Activation('sigmoid'))

创建32x128x10的神经网络，输入为32结点，中间层为128结点，激活函数为softmax，输出为10个结点，激活函数为sigmoid。

1. 模型编译：损失函数，优化器，评估指标

model.compile(loss="categorical\_crossentropy", optimizer="adam", metrics = ['accuracy'])