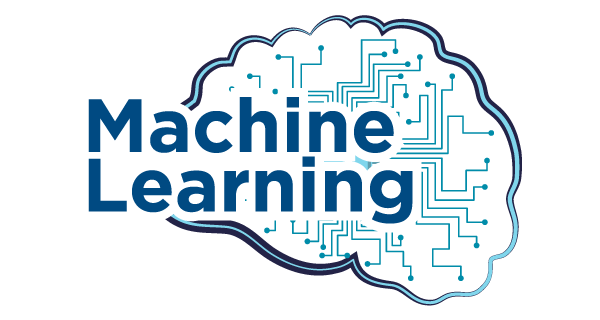
**面试题答案：**

# 机器学习理论类：



## 1.写出全概率公式&贝叶斯公式

全概率公式：

A picture containing drawing

Description automatically generated

贝叶斯公式：

A screenshot of a cell phone

Description automatically generated

## 2.模型训练为什么要引入偏差(bias)和方差(variance)？

优化监督学习=优化模型的泛化误差，模型的泛化误差可分解为偏差、方差与噪声之和 **Err = bias + var + irreducible error**，以回归任务为例，其实更准确的公式为：**Err = bias^2 + var + irreducible error^2**

符号的定义：一个真实的任务可以理解为Y=f(x)+e，其中f(x)为规律部分，e为噪声部分

* 训练数据D训练的模型称之为[](https://camo.githubusercontent.com/33daae26cf384006ca27854f9c6f4c28dd70125f/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c777a32316269396a3330307930307064666c2e6a7067)，当我们使用相同的算法，但使用不同的训练数据D时就会得到多个[](https://camo.githubusercontent.com/33daae26cf384006ca27854f9c6f4c28dd70125f/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c777a32316269396a3330307930307064666c2e6a7067)。则[](https://camo.githubusercontent.com/33268b0cf11976a480935419ec023223707f2fd4/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c777a71696176376a333031713030707132702e6a7067)代表了这个模型的期望，即使用某一算法训练模型所能得到的稳定的平均水平。
* 方差：模型的稳定性：[](https://camo.githubusercontent.com/2ffbeb38e4b54723a689dc2f47513ea1425638d4/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c783361327163336a3330363430306f74386a2e6a7067)
* 偏差：模型的准确性：[](https://camo.githubusercontent.com/d533c50360f85a6a51514e203ff55b10216728cf/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c783672716733346a3330356730306f64666e2e6a7067)
* Err(x) = Err(f,[](https://camo.githubusercontent.com/4ab14733b27037b07fec01b9bea233597f14bcee/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c783864686b706c6a3330306530306d3073362e6a7067))+Err(f,Y)
  + Err(f,[](https://camo.githubusercontent.com/4ab14733b27037b07fec01b9bea233597f14bcee/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c783864686b706c6a3330306530306d3073362e6a7067))为可解释规则误差
  + Err(f,Y) 为噪声e部分，即为[](https://camo.githubusercontent.com/92fcfa87614dadc4ede1682d735682ddcdda50ef/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786336346c35796a3330306730306230706e2e6a7067)
  + [](https://camo.githubusercontent.com/618625b47391ecf63ee00f194ac1f3ed71933f82/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c78796a326736376a3330316b30306d6d77782e6a7067)可推导如下： [](https://camo.githubusercontent.com/517dfbd6ebe9f3cc6b3ca65033a3f51ed65e77ab/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786b76783339676a3330676830316233796a2e6a7067)
  + f为真实值，固定；[](https://camo.githubusercontent.com/33268b0cf11976a480935419ec023223707f2fd4/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c777a71696176376a333031713030707132702e6a7067)代表了这个模型不同数据预测结果的期望，固定；所以f-[](https://camo.githubusercontent.com/33268b0cf11976a480935419ec023223707f2fd4/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c777a71696176376a333031713030707132702e6a7067)固定
  + [](https://camo.githubusercontent.com/7d99c40336a7a57f3c3778baaf78d30db963f3dd/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786c7a6a7039686a3330357330306d74386a2e6a7067)中[](https://camo.githubusercontent.com/0ce6f6a5a560c592938c2e3589d7e33eac7b83f8/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786d38677a6d726a3330317a30306d7132702e6a7067)为常数。所以[](https://camo.githubusercontent.com/7d99c40336a7a57f3c3778baaf78d30db963f3dd/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786c7a6a7039686a3330357330306d74386a2e6a7067)=[](https://camo.githubusercontent.com/642f970e5936f2e16bd71f05b9ee56d9c630876a/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c786f3276396f7a6a3330627730306d30736e2e6a7067) = 0
  + Err(x) = [](https://camo.githubusercontent.com/456156b498852211eb0c33141db8a144a358b292/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f30303679386d4e3667793167386c787675647466676a3330636d30306d6a72612e6a7067)

## 3.CRF/朴素贝叶斯/EM/最大熵模型/马尔科夫随机场/混合高斯模型

这几个模型中都有概率计算的过程，不像knn，svm等都是距离计算一看就知道是判别模型。

* 生成式模型：朴素贝叶斯，混合高斯模型，马尔科夫随机场，EM
  + 仔细看过这些模型细节的朋友都应该知道，他们最后都是判断x属于拟合一个正负样本分布，然后对比属于正负样本的概率
* 判别式模型：最大熵模型，CRF

## 4.如何解决过拟合问题？

* 加到数据量
* 解决数据不平衡的问题
* 加大正则化
* 减小模型复杂度

## 5.One-hot的作用是什么？为什么不直接使用数字作为表示

主要是为了解决categorcial的数据在做模型搭建的时候它的数字表征对函数引起波动

## 6.决策树和随机森林的区别是什么？

随机森林大家可以看做是为了增强模型的稳定性，由若干个决策树组合成的集合型模型，随机森林中的每个决策树，只使用部分features进行预测，然后做投票，这样增强了模型的稳定性。

## 7.朴素贝叶斯为什么“朴素naive”？

朴素贝叶斯模型，全称为：Naive Bayesian Model，Naive 翻译为朴素。朴素贝叶斯模型假设样本特征彼此独立，没有相关关系。这个假设在现实世界中是很不真实的，因此说朴素贝叶斯真的很“朴素”。

## 8.kmeans初始点除了随机选取之外的方法

先层次聚类，再在不同层次上选取初始点进行kmeans聚类。

## 9.LR明明是分类模型为什么叫回归

观测样本中该特征在正负类中出现概率的比值满足线性条件，用的是线性拟合比率值，所以叫回归。

## 10.梯度下降如何并行化

首先需要理解梯度下降的更新公式：θj+1=θj-η1mi=1mhθxi-yixi；

处的并行，不同样本在不同机器上进行计算，计算完再进行合并；

同一条样本不同特征维度进行拆分，hθxi-yixi处并行，把hθxi内的xi和wi拆分成块分别计算后合并，再把外层hθxi-yixi同样拆分成若干块进行计算。

## 11.LR中的L1/L2正则项是啥

L1正则项：为模型加了一个先验知识，未知参数w满足拉普拉斯分布，fw=12λe-|w-u|λ，u为0在lr模型损失函数中新增了|wj|2λ2项。

L2正则项：为模型加了一个先验知识，未知参数w满足0均值正太分布，fw|u,σ=12πσexp⁡(-(w-u)22σ2)，u为0。在lr模型损失函数中新增了wwT2σ2项。

## 12.简述决策树构建过程

* 构建根节点，将所有训练数据都放在根节点
* 选择一个最优特征，按照这一特征将训练数据集分割成子集，使得各个子集有一个在当前条件下最好的分类
* 如果子集非空，或子集容量未小于最少数量，递归1，2步骤，直到所有训练数据子集都被正确分类或没有合适的特征为止

## 13.解释Gini系数

Gini系数二分情况下：GiniP=Pk1-Pk=2p(1-p)

* 对于决策树样本D来说，Ginip=CkD1-CkD=1-(|Ck||D|)2
* 对于样本D，如果根据特征A的某个值，把D分成D1和D2，则在特征A的条件下，D的基尼系数为：GiniD,A=D1DGiniD1+D2DGiniD2

## 14.决策树的优缺点

优点：

* 缺失值不敏感，对特征的宽容程度高，可缺失可连续可离散
* 可解释性强
* 算法对数据没有强假设
* 可以解决线性及非线性问题
* 有特征选择等辅助功能

缺点：

* 处理关联性数据比较薄弱
* 正负量级有偏样本的样本效果较差
* 单棵树的拟合效果欠佳，容易过拟合

## 15.出现估计概率值为 0 怎么处理

拉普拉斯平滑

## 16.随机森林的生成过程

* 生成单棵决策树
* 随机选取样本
* 从M个输入特征里随机选择m个输入特征，然后从这m个输入特征里选择一个最好的进行分裂
* 不需要剪枝，直到该节点的所有训练样例都属于同一类
* 生成若干个决策树

## 17.介绍一下Boosting的思想

* 初始化训练一个弱学习器，初始化下的各条样本的权重一致
* 根据上一个弱学习器的结果，调整权重，使得错分的样本的权重变得更高
* 基于调整后的样本及样本权重训练下一个弱学习器
* 预测时直接串联综合各学习器的加权结果

## 18.gbdt的中的tree是什么tree？有什么特征

Cart tree，但是都是回归树

## 19.xgboost对比gbdt/boosting Tree有了哪些方向上的优化

1. 显示的把树模型复杂度作为正则项加到优化目标中
2. 优化目标计算中用到二阶泰勒展开代替一阶，更加准确
3. 实现了分裂点寻找近似算法
4. 暴力枚举
5. 近似算法（分桶）
6. 更加高效和快速
7. 数据事先排序并且以block形式存储，有利于并行计算
8. 基于分布式通信框架rabit，可以运行在MPI和yarn上
9. 实现做了面向体系结构的优化，针对cache和内存做了性能优化

## 20.什么叫最优超平面

两类样本分别分割在该超平面的两侧

超平面两侧的点离超平面尽可能的远

## 21.什么是支持向量

在求解的过程中，会发现只根据部分数据就可以确定分类器，这些数据称为支持向量。换句话说，就是超平面附近决定超平面位置的那些参与计算锁定平面位置的点。

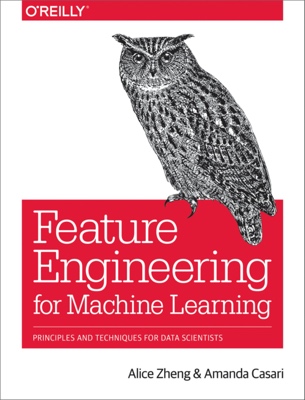
## 22.SVM如何解决多分类问题

对训练器进行组合。其中比较典型的有一对一，和一对多。

## 23.核函数的作用是啥

核函数能够将特征从低维空间映射到高维空间， 这个映射可以把低维空间中不可分的两类点变成高维线性可分的。

# 特征工程类：



## 1.怎么去除DataFrame里的缺失值？

1. DataFrame.dropna(axis=0, how='any', thresh=None, subset=None, inplace=False)函数作用：删除含有空值的行或列
2. axis:维度，axis=0表示index行,axis=1表示columns列，默认为0
3. how:"all"表示这一行或列中的元素全部缺失（为nan）才删除这一行或列，"any"表示这一行或列中只要有元素缺失，就删除这一行或列
4. thresh:一行或一列中至少出现了thresh个才删除。
5. subset：在某些列的子集中选择出现了缺失值的列删除，不在子集中的含有缺失值得列或行不会删除（有axis决定是行还是列）
6. inplace：刷选过缺失值得新数据是存为副本还是直接在原数据上进行修改。

## 2.特征无量纲化的常见操作方法

无量纲化：即nondimensionalize 或者dimensionless，是指通过一个合适的变量替代，将一个涉及物理量的方程的部分或全部的单位移除，以求简化实验或者计算的目的。

常见操作方法（基于sklearn）：

1. 标准化，返回值为标准化后的数据 from sklearn.preprocessing import StandardScaler StandardScaler().fit\_transform(data)
2. 区间缩放，返回值为缩放到[0, 1]区间的数据 from sklearn.preprocessing import MinMaxScaler MinMaxScaler().fit\_transform(data)
3. 归一化，返回值为归一化后的数据from sklearn.preprocessing import Normalizer Normalizer().fit\_transform(data)

## 3.如何对类别变量进行独热编码？

很多时候我们需要对类别变量进行独热编码，然后才可以作为入参给模型使用，独热的方式有很多种，这里介绍一个常用的方法 get\_dummies，这个方法可以让类别变量按照枚举值生成N个（N为枚举值数量）新字段，都是0-1的变量值。

以泰坦尼克号的数据集为例，对数据进行预处理操作，见下：

# 导入相关库

import pandas as pd

import numpy as np

from pandas import Series,DataFrame

import re

# 导入泰坦尼的数据集

data\_train = pd.read\_csv("./data/titanic/Train.csv")

# 提取其中几列

data = data\_train.loc[:,['PassengerId','Name']]

# 提取称谓

data['Title'] = data['Name'].map(lambda x: re.compile(", (.\*?)\.").findall(x)[0])

# 定义一个空字典来收集映射关系

title\_Dict = {}

title\_Dict.update(dict.fromkeys(['Capt', 'Col', 'Major', 'Dr', 'Rev'], 'Officer'))

title\_Dict.update(dict.fromkeys(['Don', 'Sir', 'the Countess', 'Dona', 'Lady'], 'Royalty'))

title\_Dict.update(dict.fromkeys(['Mme', 'Ms', 'Mrs'], 'Mrs'))

title\_Dict.update(dict.fromkeys(['Mlle', 'Miss'], 'Miss'))

title\_Dict.update(dict.fromkeys(['Mr'], 'Mr'))

title\_Dict.update(dict.fromkeys(['Master','Jonkheer'], 'Master'))

data['Title'] = data['Title'].map(title\_Dict)data.Title.value\_counts()

接下来对字段Title进行独热编码，这里使用get\_dummies，生成N个0-1新字段：

# 对字段Title进行独热编码，这里使用get\_dummies，生成N个0-1新字段dummies\_title = pd.get\_dummies(data['Title'], prefix="Title")

data = pd.concat([data,dummies\_title], axis=1)

data.head()

结果为：

## 4.如何把“年龄”字段按照我们的阈值分段？

·截断

连续型的数值进行截断或者对长尾数据进行对数后截断(保留重要信息的前提下对特征进行截断，截断后的特征也可以看作是类别特征)

·二值化

数据分布过于不平衡

空值/异常值过多

·分桶

小范围连续数据内不存在逻辑关系，比如31岁和32岁之间不存在明显的差异，可以归为一类

## 5.如何根据变量相关性画出热力图？

调用corr来实现变量相关性的计算，同时绘制热力图，颜色越深的话，代表相关性越强！

# 人体胸部加速度数据集,标签activity的数值为1-7

'''

1-在电脑前工作

2-站立、走路和上下楼梯

3-站立

4-走路

5-上下楼梯

6-与人边走边聊

7-站立着说话

'''

import pandas as pd

from sklearn.preprocessing import PolynomialFeatures

df = pd.read\_csv('./data/activity\_recognizer/1.csv', header=None)

df.columns = ['index','x','y','z','activity']

x = df[['x','y','z']]

y = df['activity']

# 多项式扩充数值变量

poly = PolynomialFeatures(degree=2, include\_bias=False,interaction\_only=False)

x\_poly = poly.fit\_transform(x)

pd.DataFrame(x\_poly, columns=poly.get\_feature\_names()).head()

# 查看热力图(颜色越深代表相关性越强)

%matplotlib inline

import seaborn as sns

sns.heatmap(pd.DataFrame(x\_poly, columns=poly.get\_feature\_names()).corr())

## 6.如何把分布修正为类正态分布？

数据下载地址：https://www.kaggle.com/c/house-prices-advanced-regression-techniques/data

import pandas as pd

import numpy as np

# Plots

import seaborn as sns

import matplotlib.pyplot as plt

# 读取数据集

train = pd.read\_csv(data')

分布情况，可以调用以下的方法来进行绘制：

sns.set\_style("white")

sns.set\_color\_codes(palette='deep')

f, ax = plt.subplots(figsize=(8, 7))

#Check the new distribution

sns.distplot(train['SalePrice'], color="b");

ax.xaxis.grid(False)

ax.set(ylabel="Frequency")

ax.set(xlabel="SalePrice")

ax.set(title="SalePrice distribution")

sns.despine(trim=True, left=True)

plt.show()

从结果可以看出，销售价格是右偏，而大多数机器学习模型都不能很好地处理非正态分布数据，所以我们可以应用log(1+x)转换来进行修正。

# log(1+x) 转换

train["SalePrice\_log"] = np.log1p(train["SalePrice"])

sns.set\_style("white")

sns.set\_color\_codes(palette='deep')

f, ax = plt.subplots(figsize=(8, 7))

sns.distplot(train['SalePrice\_log'] , fit=norm, color="b");

# 得到正态分布的参数

(mu, sigma) = norm.fit(train['SalePrice\_log'])

plt.legend(['Normal dist. ($\mu=$ {:.2f} and $\sigma=$ {:.2f} )'.format(mu, sigma)],

loc='best')

ax.xaxis.grid(False)

ax.set(ylabel="Frequency")

ax.set(xlabel="SalePrice")

ax.set(title="SalePrice distribution")

sns.despine(trim=True, left=True)

plt.show()

## 7.怎么简单使用PCA来划分数据且可视化呢？

PCA，全称为Principal Component Analysis，也就是主成分分析方法，是一种降维算法，其功能就是把N维的特征，通过转换映射到K维上（K<N），这些由原先N维的投射后的K个正交特征，就被称为主成分。

使用的数据集iris

# 导入相关库

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

from sklearn.decomposition import PCA

%matplotlib inline

#解决中文显示问题，Mac

%matplotlib inline

from matplotlib.font\_manager import FontProperties

# 设置显示的尺寸

plt.rcParams['font.family'] = ['Arial Unicode MS'] #正常显示中文

# 导入数据集

iris = load\_iris()

iris\_x, iris\_y = iris.data, iris.target

# 实例化

pca = PCA(n\_components=2)

# 训练数据

pca.fit(iris\_x)

pca.transform(iris\_x)[:5,]

# 自定义一个可视化的方法

label\_dict = {i:k for i,k in enumerate(iris.target\_names)}

def plot(x,y,title,x\_label,y\_label):

ax = plt.subplot(111)

for label,marker,color in zip(

range(3),('^','s','o'),('blue','red','green')):

plt.scatter(x=x[:,0].real[y == label],

y = x[:,1].real[y == label],

color = color,

alpha = 0.5,

label = label\_dict[label]

)

plt.xlabel(x\_label)

plt.ylabel(y\_label)

leg = plt.legend(loc='upper right', fancybox=True)

leg.get\_frame().set\_alpha(0.5)

plt.title(title)

# 可视化

plot(iris\_x, iris\_y,"原始的iris数据集","sepal length(cm)","sepal width(cm)")

plt.show()

plot(pca.transform(iris\_x), iris\_y,"PCA转换后的头两个正交特征","PCA1","PCA2")

## 8.怎么简单使用LDA来划分数据且可视化呢？

LDA的全称为Linear Discriminant Analysis, 中文为线性判别分析，LDA是一种有监督学习的算法，和PCA不同。PCA是无监督算法，。LDA是“投影后类内方差最小，类间方差最大”，也就是将数据投影到低维度上，投影后希望每一种类别数据的投影点尽可能的接近，而不同类别的数据的类别中心之间的距离尽可能的大。

使用的数据集iris

# 导入相关库

from sklearn.datasets import load\_iris

import matplotlib.pyplot as plt

from sklearn.discriminant\_analysis import LinearDiscriminantAnalysis

%matplotlib inline

#解决中文显示问题，Mac

%matplotlib inline

from matplotlib.font\_manager import FontProperties

# 设置显示的尺寸

plt.rcParams['font.family'] = ['Arial Unicode MS'] #正常显示中文

# 导入数据集

iris = load\_iris()

iris\_x, iris\_y = iris.data, iris.target

# 实例化

lda = LinearDiscriminantAnalysis(n\_components=2)

# 训练数据

x\_lda\_iris = lda.fit\_transform(iris\_x, iris\_y)

# 自定义一个可视化的方法

label\_dict = {i:k for i,k in enumerate(iris.target\_names)}

def plot(x,y,title,x\_label,y\_label):

ax = plt.subplot(111)

for label,marker,color in zip(

range(3),('^','s','o'),('blue','red','green')):

plt.scatter(x=x[:,0].real[y == label],

y = x[:,1].real[y == label],

color = color,

alpha = 0.5,

label = label\_dict[label]

)

plt.xlabel(x\_label)

plt.ylabel(y\_label)

leg = plt.legend(loc='upper right', fancybox=True)

leg.get\_frame().set\_alpha(0.5)

plt.title(title)

# 可视化

plot(iris\_x, iris\_y,"原始的iris数据集","sepal length(cm)","sepal width(cm)")

plt.show()

plot(x\_lda\_iris, iris\_y, "LDA Projection", "LDA1", "LDA2")

# 深度学习类：



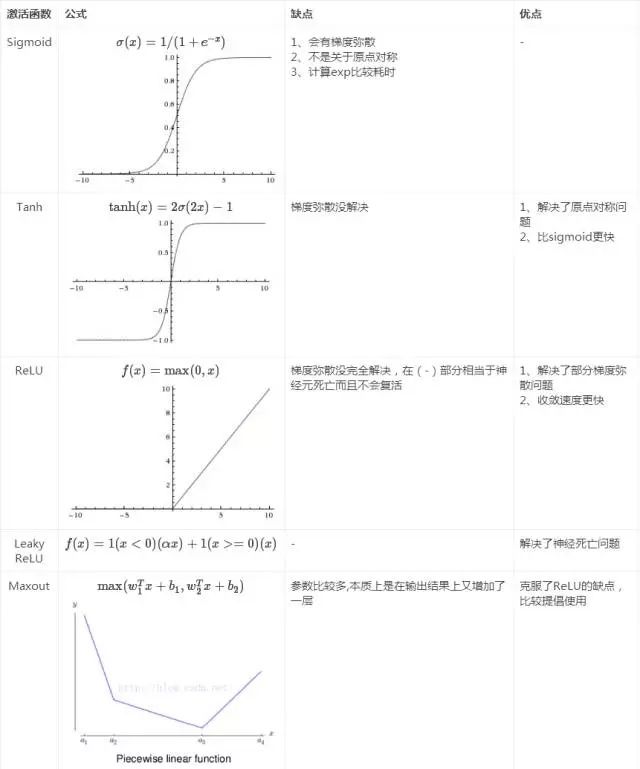
## 1.[你觉得batch-normalization过程是什么样的](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/batch_normalization.md#L164)

* 按batch进行期望和标准差计算
* 对整体数据进行标准化
* 对标准化的数据进行线性变换
* 变换系数需要学习

## 2.激活函数有什么用？常见的激活函数的区别是什么？

激活函数作用：

如果不用激励函数（其实相当于激励函数是f(x) = x），在这种情况下你每一层节点的输入都是上层输出的线性函数，很容易验证，无论你神经网络有多少层，输出都是输入的线性组合，与没有隐藏层效果相当，这种情况就是最原始的感知机（Perceptron）了，那么网络的逼近能力就相当有限。正因为上面的原因，我们决定引入非线性函数作为激励函数，这样深层神经网络表达能力就更加强大（不再是输入的线性组合，而是几乎可以逼近任意函数）。



链接：<https://zhuanlan.zhihu.com/p/32610035>

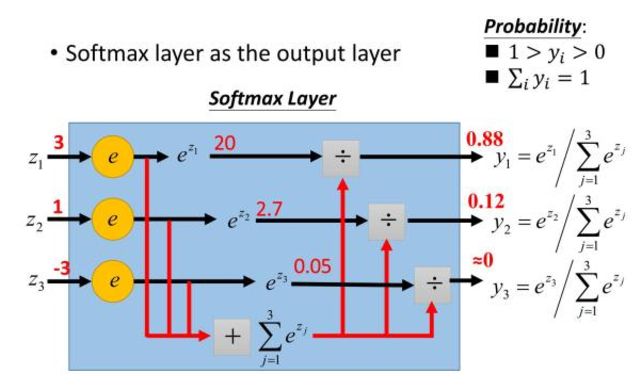
## 3.Softmax的原理是什么？有什么作用？

Softmax用于多分类神经网络输出，目的是让大的更大。函数公式是

A screenshot of a cell phone

Description automatically generated

示意图如下：



Softmax是Sigmoid的扩展，当类别数k＝2时，Softmax回归退化为Logistic回归。

## 4.CNN的平移不变性是什么？如何实现的？

平移不变性（translation invariant）指的是CNN对于同一张图及其平移后的版本，都能输出同样的结果。这对于图像分类（image classification）问题来说肯定是最理想的，因为对于一个物体的平移并不应该改变它的类别。

卷积+最大池化约等于平移不变性。卷积：简单地说，图像经过平移，相应的特征图上的表达也是平移的。在神经网络中，卷积被定义为不同位置的特征检测器，也就意味着，无论目标出现在图像中的哪个位置，它都会检测到同样的这些特征，输出同样的响应。

池化：比如最大池化，它返回感受野中的最大值，如果最大值被移动了，但是仍然在这个感受野中，那么池化层也仍然会输出相同的最大值。这两种操作共同提供了一些平移不变性，即使图像被平移，卷积保证仍然能检测到它的特征，池化则尽可能地保持一致的表达。

链接：<https://zhangting2020.github.io/2018/05/30/Transform-Invariance/>

## 5.AlexNet, VGG，GoogleNet，ResNet等网络之间的区别是什么？

网络结构对比：

A screenshot of a cell phone

Description automatically generated

特点：

AlexNet相比传统的CNN，主要改动包括Data Augmentation（数据增强）、Dropout方法、激活函数用ReLU代替了传统的Tanh或者Logistic、Local Response Normalization（LRN，实际就是利用临近的数据做归一化）、Overlapping Pooling（有重叠，即Pooling的步长比Pooling Kernel的对应边要小）、多GPU并行。

VGG很好地继承了AlexNet的特点，但是网络更深。

GoogLeNet，网络更深，但主要的创新在于他的Inception，这是一种网中网（Network In Network）的结构，即原来的结点也是一个网络。相比于前述几个网络，用了Inception之后整个网络结构的宽度和深度都可扩大，能够带来2-3倍的性能提升。

 ResNet在网络深度上有了进一步探索。但主要的创新在残差网络，网络的提出本质上还是要解决层次比较深的时候无法训练的问题。这种借鉴了Highway Network思想的网络相当于旁边专门开个通道使得输入可以直达输出，而优化的目标由原来的拟合输出H(x)变成输出和输入的差H(x)-x，其中H(X)是某一层原始的的期望映射输出，x是输入。

## 6.[残差网络为什么能解决梯度消失的问题](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/%E6%AE%8B%E5%B7%AE%E7%BD%91%E7%BB%9C.md#L164)

图片包含 游戏机, 物体, 钟表

描述已自动生成



虽然是对[](https://camo.githubusercontent.com/fd80a007be0ae1e790d5e256ff4740601ce4d7d3/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f303036744e6252776c7931673969733872746e78756a3330306c30306b3379392e6a7067)求偏导数，但是存在一项只和[](https://camo.githubusercontent.com/f7d9f54f61675a822e87d139d9709fecb8080c8e/68747470733a2f2f747661312e73696e61696d672e636e2f6c617267652f303036744e6252776c7931673969733938346269306a3330306c30306b3379392e6a7067)相关的项，之间避免了何中间权重矩阵变换导致梯度消失的问题

## 7.LSTM为什么能解决梯度消失/爆炸的问题

LSTM把原本RNN的单元改造成一个叫做CEC的部件，这个部件保证了误差将以常数的形式在网络中流动 ，并在此基础上添加输入门和输出门使得模型变成非线性的，并可以调整不同时序的输出对模型后续动作的影响。

## 8.[Attention对比RNN和CNN，分别有哪点你觉得的优势](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/Attention.md#L164)

对比RNN的是，RNN是基于马尔可夫决策过程，决策链路太短，且单向

对比CNN的是，CNN基于的是窗口式捕捉，没有受限于窗口大小，局部信息获取，且无序

## 9.[写出Attention的公式](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/Attention.md#L164)



## 10.[Attention机制，里面的q,k,v分别代表什么](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/Attention.md#L164)

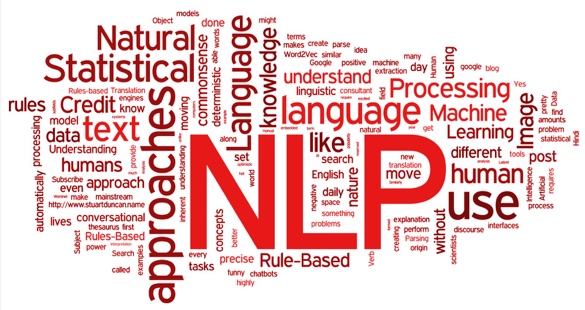
* Q：指的是query，相当于decoder的内容
* K：指的是key，相当于encoder的内容
* V：指的是value，相当于encoder的内容

q和k对齐了解码端和编码端的信息相似度，相似度的值进行归一化后会生成对齐概率值（注意力值）。V对应的是encoder的内容，刚说了attention是对encoder对重编码，qk完成权重重新计算，v复制重编码

## 11.[为什么self-attention可以替代seq2seq](https://github.com/sladesha/Reflection_Summary/blob/master/%E6%B7%B1%E5%BA%A6%E5%AD%A6%E4%B9%A0/Attention.md#L164)

* seq2seq最大的问题在于将Encoder端的所有信息压缩到一个固定长度的向量中，并将其作为Decoder端首个隐藏状态的输入，来预测Decoder端第一个单词(token)的隐藏状态。在输入序列比较长的时候，这样做显然会损失Encoder端的很多信息，而且这样一股脑的把该固定向量送入Decoder端，Decoder端不能够关注到其想要关注的信息。
* self-attention让源序列和目标序列首先“自关联”起来，这样的话，源序列和目标序列自身的embedding表示所蕴含的信息更加丰富，而且后续的FFN层也增强了模型的表达能力，并且Transformer并行计算的能力是远远超过seq2seq系列的模型。

# 自然语言处理（NLP）类：



## 1.解释GolVe的损失函数？

其实，一句话解释就是想构造一个向量表征方式，使得向量的点击和共现矩阵中的对应关系一致。因为共现矩阵中的对应关系证明了，存在i，k，j三个不同的文本，如果i和k相关，j和k相关，那么p(i,j)=p(j,k)近似于1，其他情况都过大和过小。

·如何处理未出现词？

按照词性进行已知词替换，[unknow-n],[unknow-a],[unknow-v]...，然后再进行训练。实际去用的时候，判断词性后直接使用对应的unknown-?向量替代

## 2.为什么GolVe会用的相对比W2V少？

GloVe算法本身使用了全局信息，自然内存费的也就多一些

公现矩阵，NXN的，N为词袋量

W2V的工程实现结果相对来说支持的更多，比如most\_similarty等功能

## 3.层次softmax流程

构造Huffman Tree

最大化对数似然函数

* 输入层：是上下文的词语的词向量
* 投影层：对其求和，所谓求和，就是简单的向量加法
* 输出层：输出最可能的word

沿着哈夫曼树找到对应词，每一次节点选择就是一次logistics选择过程，连乘即为似然函数

对每层每个变量求偏导，参考sgd

## 4.负采样流程

* 统计每个词出现对概率，丢弃词频过低对词
* 每次选择softmax的负样本的时候，从丢弃之后的词库里选择（选择是需要参考出现概率的）
* 负采样的核心思想是：利用负采样后的输出分布来模拟真实的输出分布

## 5.怎么衡量学到的embedding的好坏

从item2vec得到的词向量中随机抽出一部分进行人工判别可靠性。即人工判断各维度item与标签item的相关程度，判断是否合理，序列是否相关

对item2vec得到的词向量进行聚类或者可视化

## 6.阐述CRF原理

首先X,Y是随机变量，P(Y/X)是给定X条件下Y的条件概率分布

如果Y满足马尔可夫满足马尔科夫性，及不相邻则条件独立

则条件概率分布P(Y|X)为条件随机场CRF

## 7.详述LDA原理

从狄利克雷分布α中取样生成文档i的主题分布

多项式分布的共轭分布是狄利克雷分布

二项式分布的共轭分布是Beta分布

从主题的多项式分布中取样生成文档i第j个词的主题

从狄利克雷分布β中取样生成主题对应的词语分布

从词语的多项式分布中采样最终生成词语

文档里某个单词出现的概率可以用公式表示：

采用EM方法修正词-主题矩阵+主题-文档矩阵直至收敛

## 8.LDA中的主题矩阵如何计算

这个问题很难说清楚，一般会揪着细节问，不会在乎你的公式写的是不是完全一致。这部分是LDA的核心，是考验一个nlp工程师的最基础最基础的知识点。

吉布斯采样：

先随机给每个词附上主题；

因为多项式分布的共轭分布是狄利克雷分布，可以根据狄利克雷分布先验分布结合每个词实际的主题满足的多项式分布得到后验狄利克雷分布分布，从而积分得到一文档的主题条件分布，词同理，从而得到每篇文章的主题和词的联合概率分布；

有了联合概率分布，去除词wi后，就可以得到其他词主题条件概率分布；

根据条件概率分布使用坐标轮换的吉布斯采样方法，得到词对应的平稳矩阵及词对应的主题；

收敛后统计文章的词对应的主题，得到文章的主题分布；统计词对应的主题，得到不同主题下词的分布。

通常会引申出如下几个问题：

* 吉布斯采样是怎么做的？（基于MCMC思想，面对多维特征优化一维特征固定其他维度不变，满足细致平稳性，坐标变换以加快样本集生成速度）
* MCMC中什么叫做蒙特卡洛方法？
* 通常用于求概率密度的积分
* 用已知分布去评估未知分布
* reject-acpect过程

## 9.马尔科夫链收敛性质？

非周期性，不能出现死循环

连通性，不能有断点

## 10.MCMC中什么叫做马尔科夫链采样过程？

先得到转移矩阵P在N次迭代下收敛到不变的平稳矩阵

再根据平稳矩阵后的条件概率p(x/xt)得到平稳分布的样本集(xn+1,xn+2...)

## 11.给定平稳矩阵如何得到概率分布样本集？

·M-C采样

给定任意的转移矩阵Q，已知π(i)p(i,j) = π(j)p(j,i)，近似拟合π(i)Q(i,j)a(i,j) = π(j)Q(j,i)a(j,i)

根据Q的条件概率Q(x/xt)得到xt+1

u~uniform

u<π(xt+1)Q(xt+1,xt) 则accept，就和蒙特模拟一样否则xt+1 = xt

(xt,xt+1...)代表着我们的分布样本集

·M-H采样

左右同乘缩放，更新a(i,j)的计算公式，加快收敛速度

·Gibbs采样

同上，差别在固定n−1个特征在某一个特征采样及坐标轮换采样

## 12.什么叫做坐标转换采样？

平面上任意两点满足细致平稳条件π(A)P(A->B) = π(B)P(B->A)

·从条件概率分布P(x2|x(t)1)中采样得到样本x(t+1)2

·从条件概率分布P(x1|x(t+1)2)中采样得到样本x(t+1)1

其为一对样本，有点像Lasso回归中的固定n-1维特征求一维特征求极值的思路

## 13.变分推断EM算法

整体上过程是，LDA中存在隐藏变量主题分布，词分布，实际主题，和模型超参alpha，beta，需要E步求出隐藏变量基于条件概率的期望，在M步最大化这个期望，从而得到alpha，beta

变分推断在于隐藏变量没法直接求，用三个独立分布的变分分步去拟合三个隐藏变量的条件分布

实际去做的时候，用的是kl散度衡量分布之间的相似度，最小化KL散度及相对熵

EM过程

E：最小化相对熵，偏导为0得到变分参数

M：固定变分参数，梯度下降法，牛顿法得到alpha和beta的值

LDA和Word2Vec区别？LDA和Doc2Vec区别

LDA比较是doc，word2vec是词

LDA是生成的每篇文章对k个主题对概率分布，Word2Vec生成的是每个词的特征表示

LDA的文章之间的联系是主题，Word2Vec的词之间的联系是词本身的信息

LDA依赖的是doc和word共现得到的结果，Word2Vec依赖的是文本上下文得到的结果

## 14.Bert的双向体现在什么地方

mask+attention，mask的word结合全部其他encoder word的信息

## 15.Bert的是怎样预训练的

MLM：将完整句子中的部分字mask，预测该mask词

NSP：为每个训练前的例子选择句子 A 和 B 时，50% 的情况下 B 是真的在 A 后面的下一个句子， 50% 的情况下是来自语料库的随机句子，进行二分预测是否为真实下一句

在数据中随机选择 15% 的标记，其中80%被换位[mask]，10%不变、10%随机替换其他单词，原因是什么

mask只会出现在构造句子中，当真实场景下是不会出现mask的，全mask不match句型了

随机替换也帮助训练修正了[unused]和[UNK]

强迫文本记忆上下文信息

为什么BERT有3个嵌入层，它们都是如何实现的

input\_id是语义表达，和传统的w2v一样，方法也一样的lookup

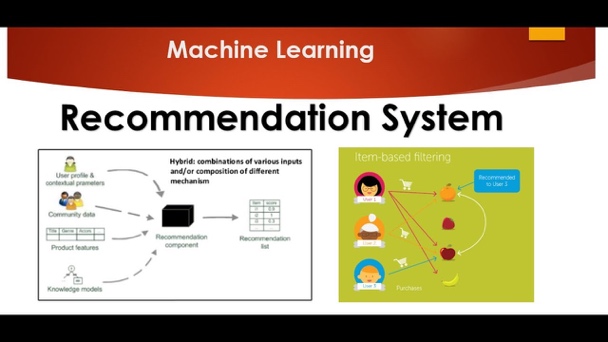
segment\_id是辅助BERT区别句子对中的两个句子的向量表示，从[1,embedding\_size]里面lookup

position\_id是为了获取文本天生的有序信息，否则就和传统词袋模型一样了，从[511,embedding\_size]里面lookup

## 16.手写一个multi-head attention

tf.multal(tf.nn.softmax(tf.multiply(tf.multal(q,k,transpose\_b=True),1/math.sqrt(float(size\_per\_head)))),v)

# 推荐系统类：



## 1.DNN与DeepFM之间的区别

DNN是DeepFM中的一个部分，DeepFM多一次特征，多一个FM层的二次交叉特征

## 2.你在使用deepFM的时候是如何处理欠拟合和过拟合问题的

* 欠拟合：增加deep部分的层数，增加epoch的轮数，增加learning rate，减少正则化力度
* 过拟合：在deep层直接增加dropout的率，减少epoch轮数，增加更多的数据，增加正则化力度，shuffle数据

## 3.deepfm的embedding初始化有什么值得注意的地方吗

* 常规的是Xavier，输出和输出可以保持正态分布且方差相近：np.random.rand(layer[n-1],layer[n])\*np.sqrt(1/layer[n-1])
* relu的情况下通常是HE，保证半数神经元失活的情况下对输出方差影响最小:：np.random.rand(layer[n-1],layer[n])\*np.sqrt(2/layer[n-1])
* 文本项目上也可以用预训练好的特征

## 4.YoutubeNet 变长数据如何处理的

input数据中只拿了近20次的点击，部分用户是没有20次的历史行为的，所以我们记录了每一个用户实际点击的次数，在做embedding的时候，我们除以的是真实的history length

20次点击过去一周内的行为，曾经尝试扩大历史点击次数到40，60没有很明显的效果提升

* 点击行为是处理过的，停留时间过短的click不要
* 点击行为是处理过的，连续多次的重复点击会去重
* 点击行为是处理过的，session内的点击次数需要在约定范围内

## 5.YouTubeNet如何避免百万量级的softmax问题的

使用负采样方法

## 6.推荐系统有哪些常见的评测指标？

按照推荐任务的不同，最常用的推荐质量度量方法可以划分为三类：

* + 1. 对预测的评分进行评估，适用于评分预测任务。
    2. 对预测的item集合进行评估，适用于Top-N推荐任务。
    3. 按排名列表对推荐效果加权进行评估，既可以适用于评分预测任务也可以用于Top-N推荐任务。

1. 评分预测指标：如准确度指标：平均绝对误差（MAE）、均方误差根（RMSE）、标准化平均误差（NMAE）；以及覆盖率（Coverage）
2. 集合推荐指标：如精密度(Precision)、召回(Recall)、 ROC和AUC
3. 排名推荐指标：如half-life和discounted cumulative gain等

## 7.MLR的原理是什么？ 做了哪些优化？

MLR可以看做是对LR的一个自然推广，它采用分而治之的思路，用分片线性的模式来拟合高维空间的非线性分类面，其形式化表达如下：

这里面超参数分片数m可以较好地平衡模型的拟合与推广能力。

MLR算法适合于工业级的大规模稀疏数据场景问题，如广告CTR预估。

优势体现在两个方面：

1）端到端的非线性学习：从模型端自动挖掘数据中蕴藏的非线性模式，省去了大量的人工特征设计，这 使得MLR算法可以端到端地完成训练，在不同场景中的迁移和应用非常轻松。

2）稀疏性：MLR在建模时引入了L1和L2,1范数正则，可以使得最终训练出来的模型具有较高的稀疏度， 模型的学习和在线预测性能更好。

2.3 MLR算法高级特性

1）结构先验。基于领域知识先验，灵活地设定空间划分与线性拟合使用的不同特征结构。例如精准定向 广告中验证有效的先验为：以user特征空间划分、以ad特征为线性拟合。

2）线性偏置。这个特性提供了一个较好的方法解决CTR预估问题中的bias特征，如位置、资源位等。

3）模型级联。MLR支持与LR模型的级联式联合训练，这有点类似于wide&deep learning。在我们的实践经验中，一些强feature配置成级联模式有助于提高模型的收敛性。

4）增量训练。实践证明，MLR通过结构先验进行pretrain，然后再增量进行全空间参数寻优训练，会获得进一步的效果提升。同时增量训练模式下模型达到收敛的步数更小，收敛更为稳定。

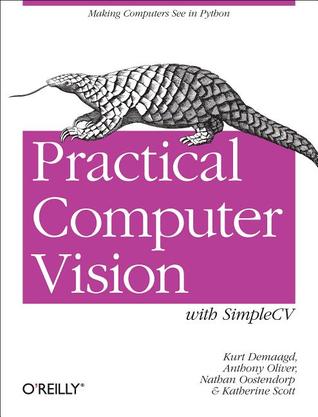
MLR中使用的优化算法是从OWLQN改进过来的，主要有三个地方的变化：

MLR使用方向导数来优化目标函数，而不是OWLQN的次梯度

MLR对更新方向p进行了象限约束：非正定时直接用方向导数作为搜索方向，否则要进行象限约束在方向导数所在象限内。

线性搜索的象限约束不同，当MLR参数不在零点时，line search保持在参数所在象限内搜索，在零点时，参数在方向导数约束的象限内进行line search，给定更新方向，MLR使用了 backtracking line search方法找到合适的步长α

# 计算机视觉（CV）类：



## 1.常见的模型加速方法

模型加速通常是指模型训练完成后，对模型推理速度得提高。

常用思路有：

* 网络剪枝系列思路
* 模型量化系列思路

常用工具：

* tensorTR
* Tensor Comprehension
* Distiller

## 2.目标检测里如何有效解决常见的前景少背景多的问题

常用解决样本数据不平衡思路都可。

1. 硬数据挖掘
2. 类平衡思路改进过得损失函数

## 3.目标检测里有什么情况是SSD、YOLOv3、Faster R-CNN等所不能解决的，假设网络拟合能力无限强

若网络拟合能力无限强，

* 剩下最大得问题数据完备性。
* 目标是3维的，显示在图像是2维的，二维数据库在目标大小，以及方向上。无法满足SSD,YOLOv3,Faster R-CNN得需求。

## 4.ROIPool和ROIAlign的区别

* ROIPool获取到得候选区域相比ROIAlign来说偏差较大
* ROIPool采用以缩小后得featuremap得像素维单位进行候选区域求取
* ROIPool采用直接拟合得方式进行候选区域求取，更加精细化

## 5.介绍常见的梯度下降优化方法

GD 梯度下降

批量梯度下降 BGD

随机梯度下降 SGD

A screenshot of a cell phone

Description automatically generated

Momentum

A screenshot of a cell phone

Description automatically generated

AdaGrad

A screenshot of a cell phone

Description automatically generated

RMSprop

A screenshot of a cell phone

Description automatically generated

Adadelta

A screen shot of a social media post

Description automatically generated

Adam

A close up of text on a white background

Description automatically generated

## 6.Detection你觉的还有哪些可做的点

可以往加速的方向做，可以考虑更结构化得图神经网络。

## 7.mini-Batch SGD相对于GD有什么优点

优点：

* 每次迭代计算量小，对硬件算力要求低
* 可训练更大得数据集

## 8.人体姿态估计主流的两个做法是啥？简单介绍下

* 先整体后局部：第一步先检测到人，得到人身体区域。第二步，在身体区域内检测骨骼关键点。
* 先局部后整体：第一步对图像上所有得骨骼关键点进行检测。第二部，将属于一个身体得骨骼关键点合并。

## 9.卷积的实现原理以及如何快速高效实现局部weight sharing的卷积操作方式

* + 卷积操作在具体实现时有多种方式
  + 可以直接按照公式实现，但是这种方法无法并行计算。
  + 利用高效得并行计算得方式来实现卷积

通常需要牺牲空间。

具体做法：

* 将卷积得第一个滑动窗内得数据取出，做为列向量v1。
* 将卷积得第二个滑动窗内得数据取出，做为列向量v2。
* N 将卷积得第N个滑动窗内得数据取出，做为列向量v\_N。
* 将v1,v2,v\_N组合成NxK的矩阵与KxK得卷积核做矩阵乘法，即实现局部快速weight sharing卷积。

注意到：v1,v2之间数据可能重复。这就多占用了空间。

## 10.CycleGAN的生成效果为啥一般都是位置不变纹理变化，为啥不能产生不同位置的生成效果

总体上来说网络结构与损失函数决定了没有对网络进行任何得拉伸变换或者刚体变换。

卷积神经网络主要是对图像产生滤波操作。 loss得约束也只是改变了卷积核得取值。