**成绩考核**

平时成绩：40%， 三次作业: 30%（第三、五、七周），课堂表现、出勤：10%

课程论文：60%（第十周交）

**课程内容**

第一周：线性模型、非线性模型、神经网络

第二周：神经网络的激活函数与损失函数

第三周：神经网络实现分类与回归

第四周：神经网络的发展---深度学习

第五周：卷积神经网络、图片分类与文本分类

第六周：基于卷积神经网络的图片、文本分类文献阅读

第七周：递归神经网络、时间序列预测

第八周：迁移学习、模型微调

第九周：基于递归神经网络和迁移学习的文献阅读

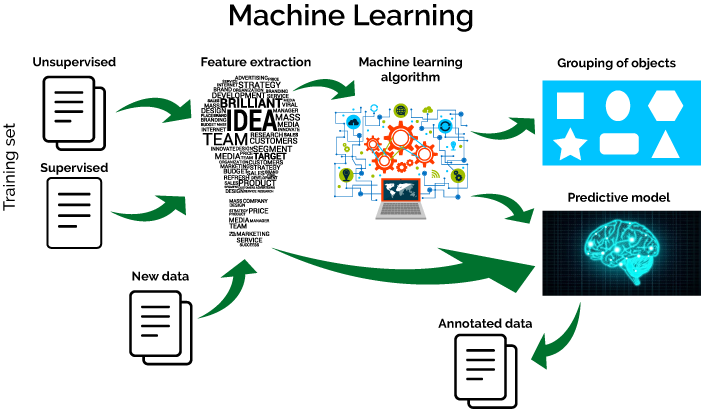
第十周：论文写作指导

**第一课 线性模型、非线性模型、神经网络**

# 机器学习做什么



# 机器学习流程



# 机器学习原理

1、输入x：一般以m表示输入样本数量，n表示每个样本具有的属性个数。

：表示第i个样本的x向量



：表示i个样本的第j个维度的值



2、输出y：目标值/标签值。

3、假设函数（Hypothesis）：https://img-blog.csdn.net/2018052522442518?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

4、需要求解的参数（Parameters）：https://img-blog.csdn.net/20180525224437273?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

5、损失函数（Loss/Error function）：单个样本的误差。

https://img-blog.csdn.net/20180525224543119?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

6、代价函数（Cost function）：训练集所有样本损失函数之和的平均值。

https://img-blog.csdn.net/2018052522455617?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

https://img-blog.csdn.net/2018052522460870?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70为https://img-blog.csdn.net/20180525224625387?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70的预测值； Goal：https://img-blog.csdn.net/20180525224639459?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

7、目标函数（Objective function）：代价函数加上正则项

https://img-blog.csdn.net/20180525224703878?watermark/2/text/aHR0cHM6Ly9ibG9nLmNzZG4ubmV0L2xpdXk5ODAz/font/5a6L5L2T/fontsize/400/fill/I0JBQkFCMA==/dissolve/70

8、鲁棒性（Robustness）：表示系统对特性或参数扰动的不敏感性，即系统的健壮性、稳定性，当存在部分异常数据时算法也会很好的拟合数据集。

9、拟合：构建的算法符合给定的数据集的特征程度。

欠拟合（Underfitting）：high bias 算法不太符合给定数据集的特征。

过拟合（Overfitting）：high variance 算法太符合给定数据集的特征，但对新数据集特征的拟合程度差。



# 机器学习实例

一个完整的端到端机器学习项目

<https://mp.weixin.qq.com/s?__biz=MjM5ODkzMzMwMQ==&mid=2650408766&idx=1&sn=7500a60a7eec39b0e52482543186e195&chksm=becd816489ba0872fe37b15c66dbd427d719e4d2bed63468e2ca17ffdeac9b335db5123e8053&scene=0&xtrack=1#rd>

实例分析

# 怎样区分线性和非线性

1、线性linear指量与量之间按比例、成直线的关系，在数学上可以理解为一阶导数为常数的函数；

　 非线性non-linear则指不按比例、不成直线的关系，一阶导数不为常数。

2、线性的可以认为是1次曲线，比如y=ax+b ，即成一条直线；

　 非线性的可以认为是2次以上的曲线，比如y=ax^2+bx+c，（x^2是x的2次方），即不为直线的即可。

3、两个变量之间的关系是一次函数关系的——图象是直线，这样的两个变量之间的关系就是“线性关系”；

　 如果不是一次函数关系的——图象不是直线，就是“非线性关系。

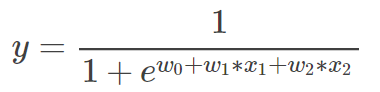
4、“线性”与“非线性”，常用于区别函数 y = f(x) 对自变量x的依赖关系。

　 线性函数即一次函数，其图像为一条直线。其它函数则为非线性函数，其图像不是直线。

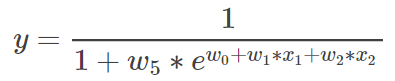
# 线性模型和非线性模型区别

　　1、线性模型可以是用曲线拟合样本，但是分类的决策边界一定是直线的，例如logistics模型。

　　2、区分是否为线性模型，主要是看一个乘法式子中自变量x前的系数w，如果w只影响一个x，那么此模型为线性模型。或者判断决策边界是否是线性的。

　　3、举例：

　此模型画出y和x是曲线关系，但是它是线性模型，因为x1\*w1中可以观察到x1只被一个w1影响



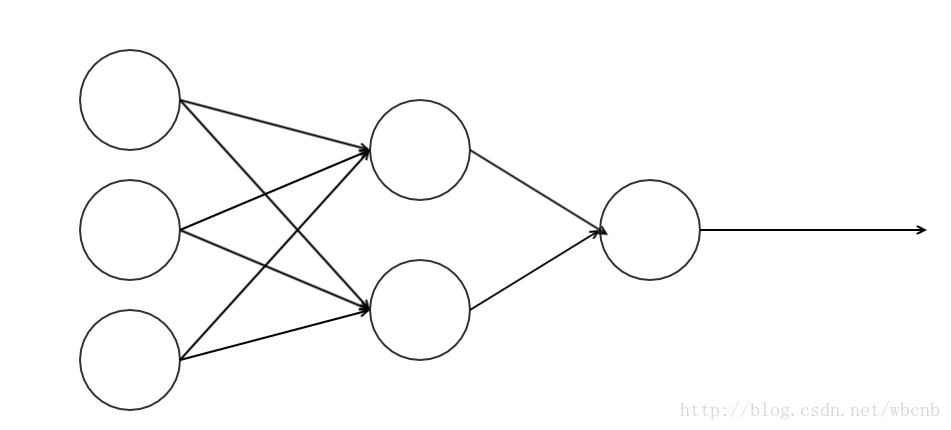
　　此模型是非线性模型，观察到x1不仅仅被参数w1影响，还被w5影响，如果自变量x被两个以上的参数影响，那么此模型是非线性的！

　　4、其实最简单判别一个模型是否为线性的，只需要判别决策边界是否是直线，也就是是否能用一条直线来划分。

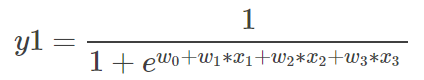
　　举例：

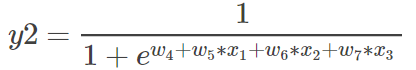
神经网络是非线性，虽然神经网络的每个节点是一个logistics模型，但是组合起来就是一个非线性模型。

　　此处我们仅仅考虑三层神经网络

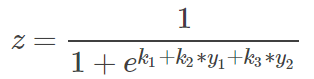


第一层的表达式：



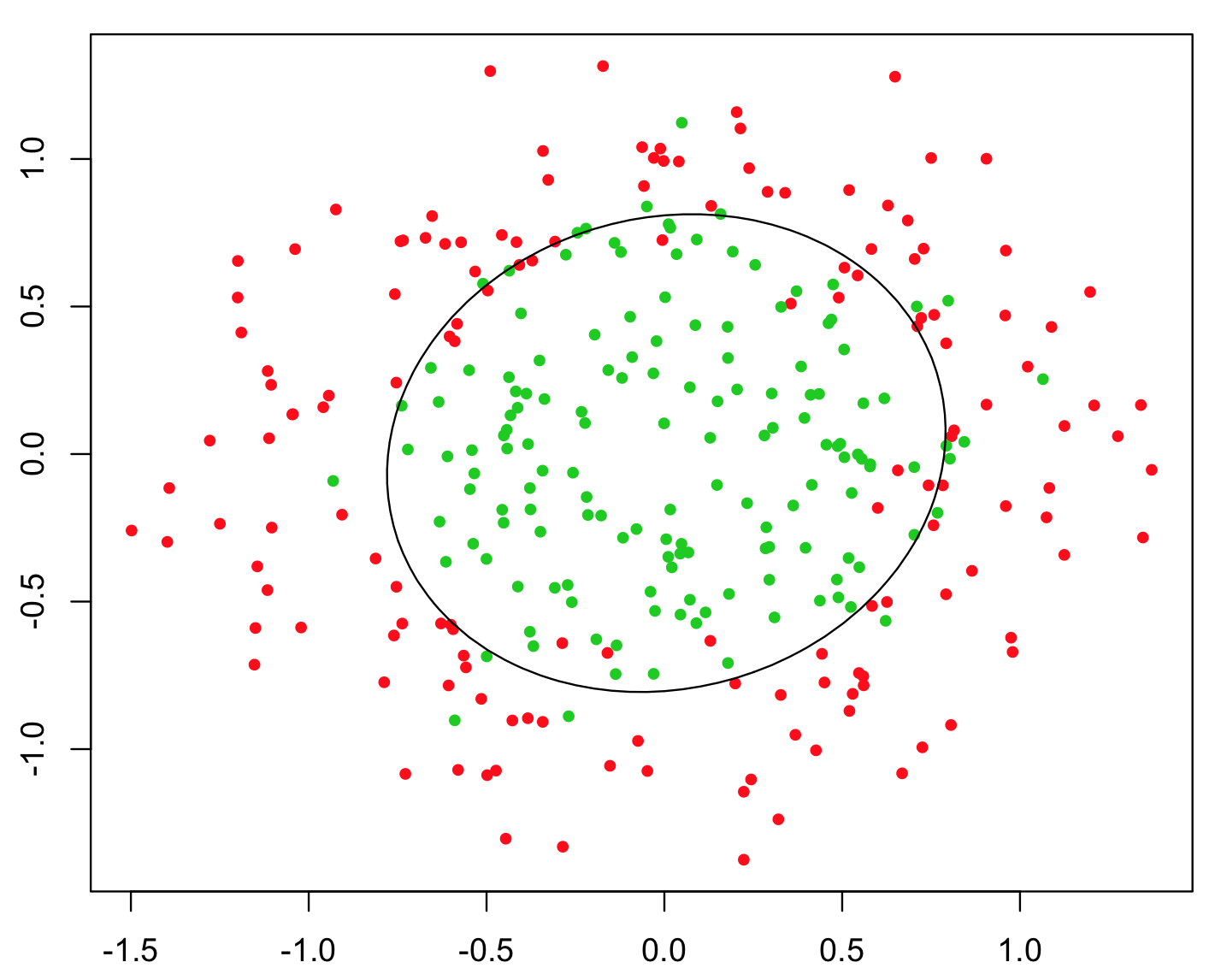
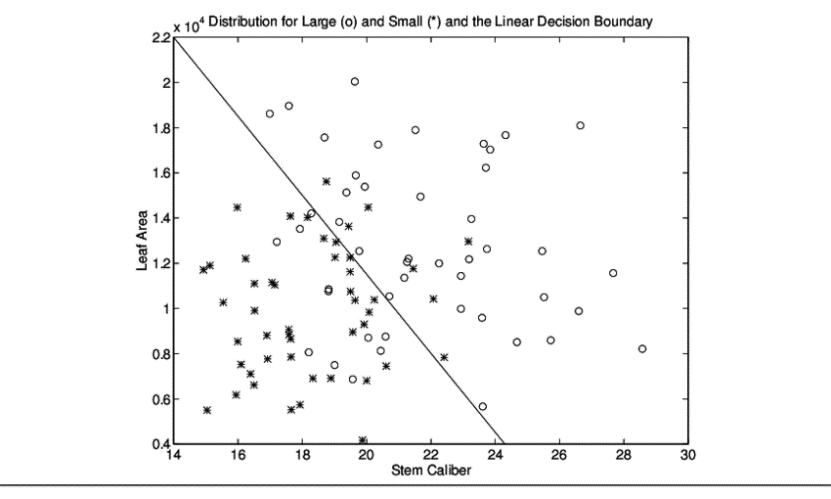


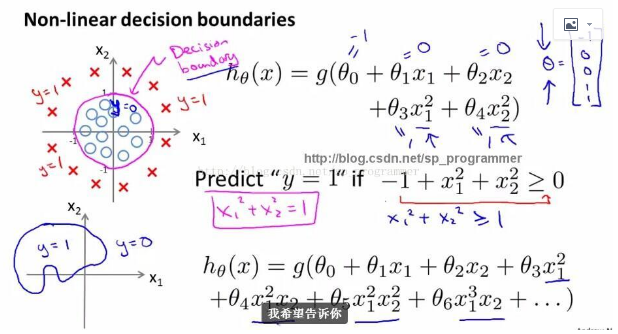
　　第二层的表达式：



　　将第一层的表达式带入第二层表达式中，可以观察到x1变量不仅仅被w1影响还被k2影响，所以此模型不是一个线性模型，是个非线性模型。

线性模型与非线性模型 图像





常见的线性分类器有：LR, 贝叶斯分类，单层感知机、线性回归

常见的非线性分类器：决策树、RF、GBDT、多层感知机SVM两种都有(看线性核还是高斯核)

**普通神经网络与其他机器学习方法的效果比较**

对于结构化的数值数据来说，普通神经网络没有SVM、随机森林、xgboost、lightGBM效果好，因为数据特征简单、人为加了一些更有效的特征（数值数据特征、文本TF-IDF特征、图像SIFT特征）

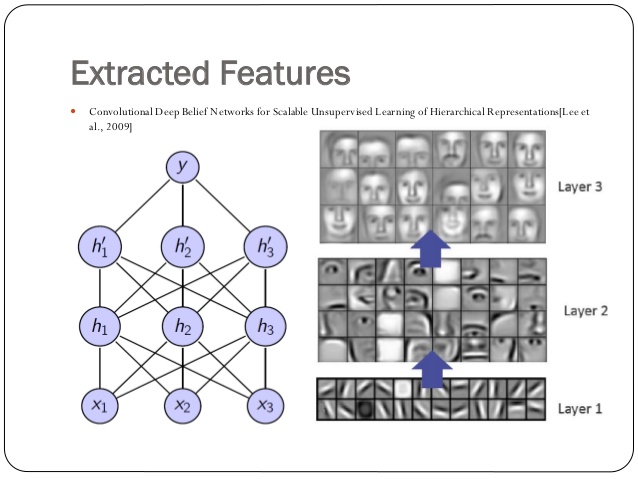
**实验验证**

**为什么要学习神经网络**

深度学习（深度神经网络）

对于非结构化数据（文本、图像），人为创建特征困难，没有了创建好的特征，其他机器学习方法也没办法取得好的效果

深度神经网络可以实现特征的自动提取—这是它最大的优势



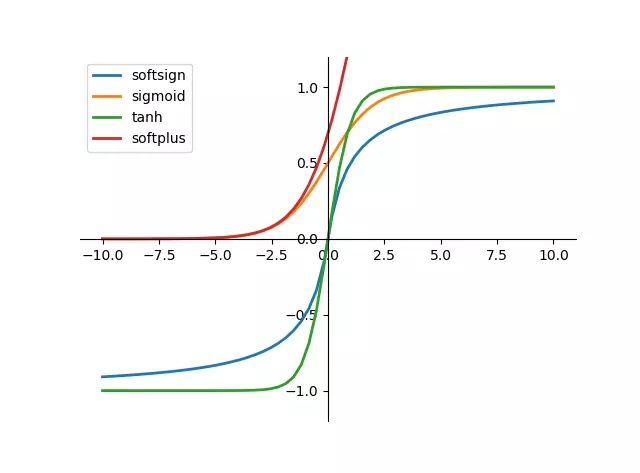
# 神经网络原理

小白都能看懂的神经网络教程

<https://mp.weixin.qq.com/s?__biz=MzU0MDk0MjA4NA==&mid=2247484002&idx=1&sn=b9972c53b140a08e4bd95236e522133f&chksm=fb30c809cc47411fd12f45d26647cccbb783f463154bf02035a31f41879fb3bff10344bb4a98&mpshare=1&scene=1&srcid=&key=4c038a0a0f8a008b46bcfc3419f9a1ca003573d5ba2e3a2ec834d0b0ea151d1a004cdb615018493eeda7f0a01439200daed2d80e8a32b81e69f9acb7c6d38e38f465d754c8e0f9a9deb14beeb0ca5a3a&ascene=1&uin=MTYwOTEyNjk4Mw%3D%3D&devicetype=Windows+10&version=62060739&lang=zh_CN&pass_ticket=qy119xNTHEJ%2B%2FTMGDvez4f%2BSWwgMXNvczPA%2FvOcWyNhT9ec5KoKY4TXv1zm3K7rd>

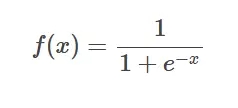
# 激活函数

常见的传统激活函数主要有两个：sigmoid和tanh。



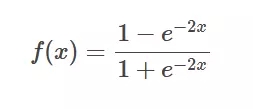
代码运行figure1

**sigmoid函数**



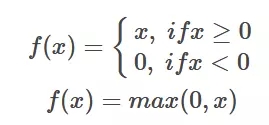
它是使用范围最广的一类激活函数，具有指数函数形状，在物理上最接近神经元。它的输出范围在（0,1）之间，可以被表示成概率，或者用于数据的归一化。但是它有两个严重的缺陷：       1. 软饱和性——导数 f'(x)=f(x)(1-f(x))，当x趋于无穷时，f(x)的两侧导数逐渐趋于0。       在后向传递时，sigmoid向下传递的梯度包含了一个f'(x)因子，因此，一旦落入饱和区f'(x)就变得接近于0，导致了向后传递的梯度也非常小。此时，网络参数很难得到有效训练，这种现象被称为梯度消失。一般在5层以内就会产生梯度消失的现象。        2. sigmoid函数的输出均大于0，这就使得输出不是0均值，这称为偏置现象。这将会导致后一层的神经元将得到上一层输出的非0均值的信号作为输入。

**tanh函数**



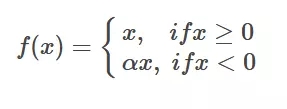
    tanh函数与sigmoid函数相比，输出均值为0，这就使得其收敛速度要比sigmoid快，从而可以减少迭代次数。    缺点就是同样具有软饱和性，会造成梯度消失。针对sigmoid和tanh的饱和性，产生了ReLU函数。

**ReLU函数**

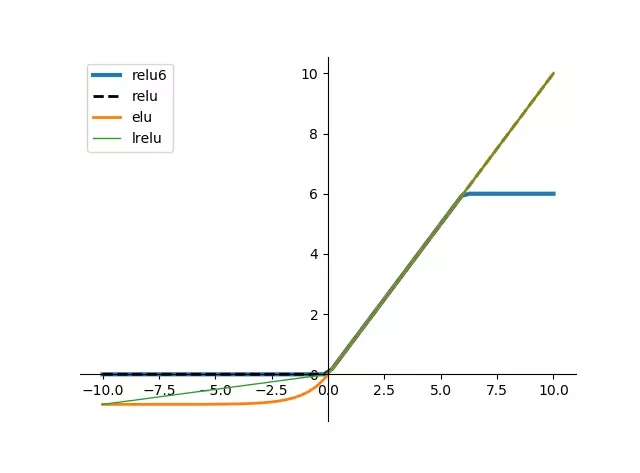


    ReLU全称为Rectified Linear Units，可以翻译成线性整流单元或者修正线性单元。    它在x>0时不存在饱和问题，从而使保持梯度不衰减，从而解决了梯度消失问题。这让我们能够直接以监督的方式训练深度神经网络，而无需依赖无监督的逐层预训练。然而，随着训练的推进，部分输入会落入硬饱和区，导致对应权重无法更新，这种现象称为“神经元死亡”    与sigmoid类似，ReLU的输出均值也大于0，所以偏移现象和神经元死亡共同影响网络的收敛性。Leaky-ReLU

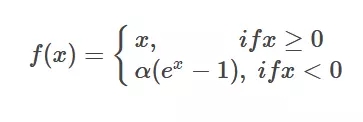
**Leaky-Relu函数**



    为了避免ReLU在x<0时的神经元死亡现象，添加了一个参数。



**ELU函数**



它结合了sigmoid和ReLU函数，左侧软饱和，右侧无饱和。    右侧线性部分使得ELU能缓解梯度消失，而左侧软饱和能让对ELU对输入变化或噪声更鲁棒。ELU的输出均值接近于0，所以收敛速度更快。

<https://www.cnblogs.com/lliuye/p/9486500.html> 解释更详细

<https://www.jianshu.com/p/dc4e53fc73a0>

<https://blog.csdn.net/c123_sensing/article/details/81531519>

# 梯度消失和梯度爆炸

详解机器学习中的梯度消失、爆炸原因及其解决方法

<https://blog.csdn.net/qq_25737169/article/details/78847691>

# 损失函数

神经网络Loss损失函数总结

<https://blog.csdn.net/willduan1/article/details/73694826>