By：计算机科学与技术2019-03 王杰永

机器学习基础(2021-2022-2)

1 问答题（15分）

1.1 监督学习与无监督学习

摘自西瓜书与统计学习方法

监督学习是指从标注数据中学习预测模型的问题。标注数据表示输入输出的对应关系，预测模型对给定的输入产生相应的输出。监督学习的本质是学习输入到输出的映射的统计规律。

无监督学习是指从无标注数据中学习预测模型的问题。无标注数据是自然得到的数据，预测模型表示数据的类别、转换或规律。无监督学习的本质是学习数据中的统计规律或潜在结构。

监督学习与无监督学习最本质的区别是训练数据是否带有标签。

1.2 欠拟合与过拟合

摘自西瓜书

过拟合：学习器把训练样本“学的太好”，将训练样本自身的一些特点当作了所有潜在样本都会具有的一般性质，从而导致泛化性能下降。

欠拟合：与过拟合相对，指学习器对训练样本的一般性质尚未学好。

1.3 机器学习系统包括哪三个部分

摘自PPT

- 预处理
- 特征提取和特征选择

  图像特征方法有：sift、hog；文本特征方法有：tf-idf
- 分类器(的设计或者分类决策过程的设计)

1.4 分类与回归

摘自西瓜书与统计学习方法

分类：输出变量的取值为离散值时，预测问题/学习任务称为分类问题。

回归：输出变量的取值为连续值时，预测问题/学习任务称为回归问题。

1.5 特征选择与特征提取

摘自PPT

- 特征选择：从一组特征中挑选出一些最有效的特征以达到降低特征空间维数的目的，这个过程叫特征选择。
- 特征提取：特征提取是一种变换，将原始数据变换成坐标个数减少了的数据。

2 Python程序设计（10分）

2.1 求出1000以内能被3整除且至少有一个数字为5的所有整数，比如15、51、513等。

```python
def main():
    ans = []
    for num in range(1, 1001):
        if num % 3 == 0 and '5' in str(num):
            ans.append(num)
    print(ans)
```

2.2 求阶乘$n!$，比如$5! = 120$

```python
def factorial(n):
    ans = 1
    for factor in range(1, n + 1):
        ans *= factor
    return ans
```

2.3 求阶乘的累加和

```python
def factorial_sum(n):
    ans = 0
    cur = 1
    for factor in range(1, n + 1):
        cur *= factor
        ans += cur
    return ans
```

2.4 编写生成斐波那契数列的程序 $1, 1, 2, 3, 5, 8, 13...$

```python
def fib(n):
    if n == 1 or n == 2:
        return 1
    pre = 1
    cur = 1
    for i in range(3, n + 1):
        temp = cur
        cur = pre + cur
        pre = temp
    return cur
```

2.5 编写函数实现$exp(x) = 1 + \frac{x}{1!} + \frac{x^2}{2!} + ...$(取前30项)

```
def calculate_exp(x, n):
    ans = 1
    factorial = 1
    for i in range(1, n):
        factorial *= i
        ans += pow(x, i) / factorial
    return ans
```

2.6 编写判别素数的程序

```
def judge(x):
    if x == 1 or x == 0:
        return False
    i = 2
    while i * i <= x:
        if x % i == 0:
            return False
        i += 1
    return True
```

3 计算题(15分)

最优化

很简单

4 阅读程序题(50分)

4.1 PCA

- pca及计算
- 混淆矩阵，precision recall F1的计算

4.2 sklearn API

- sklearn数据集划分函数(划分成训练集和测试集)
- 分类方法函数(支持向量机、K近邻、线性判别分析)及个参数的意义(比如，支持向量机中和函数的名称及作用)

SVM:

来自

```
model=svm.SVC(C=2,kernel='rbf',gamma=10,decision_function_shape='ovo')
model.fit(train_data,train_label.ravel())
```

- C越大代表惩罚程度越大，越不能容忍有点集交错的问题，但有可能会过拟合（defaul C=1）；
- kernel常规的有‘linear’，‘poly’，‘rbf’，‘sigmoid’，‘precomputed’ ，默认的是rbf;

- gamma是核函数为‘rbf’，‘poly’ 和 ‘sigmoid’时的参数设置，其值越小，分类界面越连续，其值越大，分类界面越“散”，分类效果越好，但有可能会过拟合，默认的是特征个数的倒数；
- decision_function_shape='ovr'时，为one v rest（一对多），即一个类别与其他类别进行划分，等于'ovo'时，为one v one（一对一），即将类别两两之间进行划分，用二分类的方法模拟多分类的结果。

KNN：

来自

```
from sklearn.neighbors import KNeighborsClassifier
knn = KNeighborsClassifier(n_neighbors=3, weights='distance')
knn.fit(X_train, y_train)

y_predict = knn.predict(X_test)
```

- n_neighbors：整数，默认为5，决定考虑几个点来投票
- weights：'distance'为投票时是否考虑距离的远近，越近的权重越大；'uniform'则为等权重投票。
- p：整数，考虑使用哪种距离考量，默认值为2，即欧式距离

LDA：

LDA的思想是最大化类间均值，最小化类内方差。降维后（将数据投影至低维度上），同类别投影点尽可能近，不同类别投影点尽可能远。

LDA与PCA均用于数据降维。但LDA是监督学习方法，利用到了训练集类别的先验知识，PCA这种无监督学习方法则无法利用先验知识。同时，LDA主要用于降维，但还可以用于分类。

LDA选择分类性能最好的投影方向，而PCA选择样本点投影具有最大方差的方向。

来自

```
from sklearn.discriminant_analysis import LinearDiscriminantAnalysis
clf = LinearDiscriminantAnalysis()
clf.fit(X_train, y_train)

y_predict = clf.predict(X_test)
```

- n_components：默认值为min(n_class-1, n_features)，且该值为可以设置的最大值。

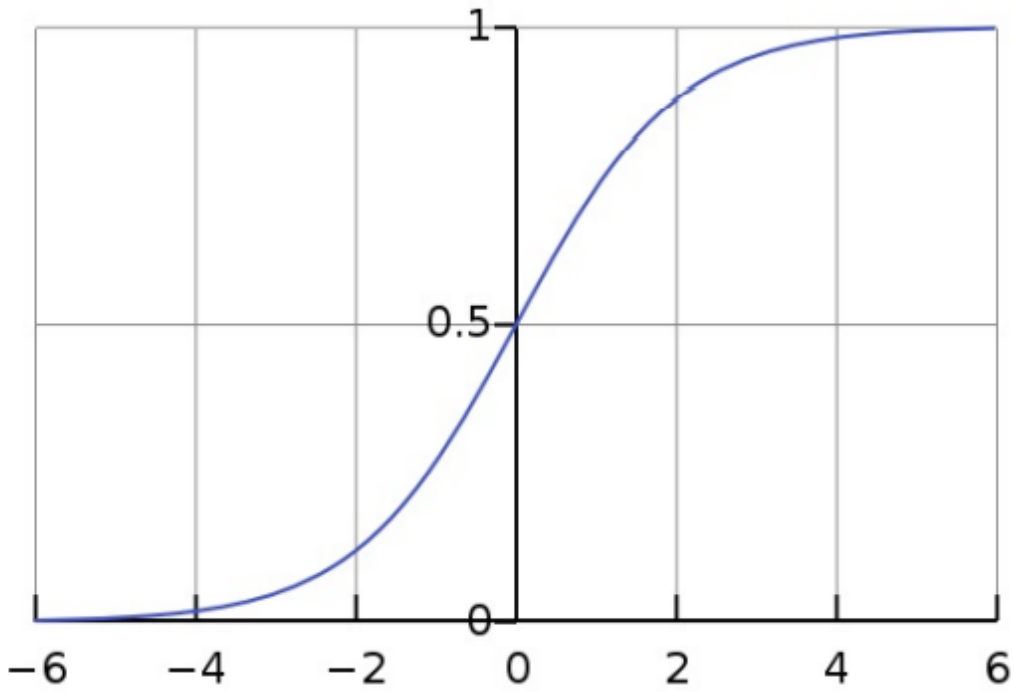
4.3 Pytorch

- 网络定义：卷积层、全连接层，几层网络；
- 卷积的计算
- 激活函数的种类和作用
- 什么优化方法: SGD、Adam
- 损失函数在何处定义、是什么（比如交叉熵损失）

4.3.1 激活函数

Sigmoid：

$$\sigma(x) = \frac{1}{1+e^{-x}}$$



优点：

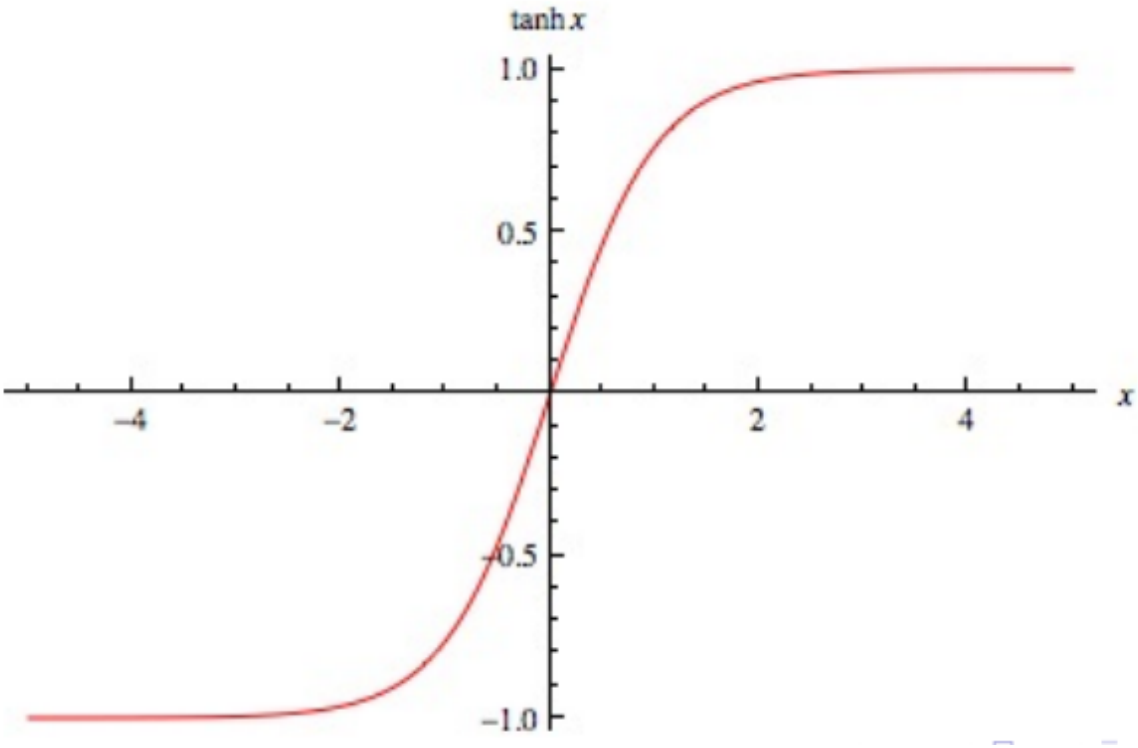
- 引入非线性
- 容易求导
- 可以将实数压缩至$(0,1)$，进而被视作概率。

主要缺点：

- Sigmoid神经元在值为0或1时接近饱和，梯度几乎为0；同时，其导数的最大值为0.25，层数较深时累计梯度几乎为0。从而导致反向传播时，这些接近0的梯度会使参数难以更新，即梯度对模型的更新没有任何贡献
- 函数并非关于原点中心对称，导致后续网络层的输入也不是零中心的，进而影响梯度下降的运作（和梯度消失比起来，小问题）

Tanh：

$$tanh(x) = \frac{e^{x}-e^{-x}}{e^{x}+e^{-x}}$$



tanh相当于一个放大的sigmoid函数，解决了sigmoid输出非零中心问题，但同样存在饱和而导致的梯度消失。

ReLU:

相较于sigmoid/tanh，ReLU对SGD的收敛有巨大的加速作用。且不存在饱和现象，不需要进行指数等复杂运算，计算速度较快。

但ReLU在训练时对学习率敏感，比较脆弱可能“死掉”，即可能导致流过ReLU单元的全部梯度都变为0。

5 展望(10分)