燕山大学

Python 机器学习实验指导书

Python machine learning Experiment Instruction Book

实验六：神经网络

教 务 处

2021 年 2 月

实验六　神经网络

一、实验目的

1.理解并掌握神经网络模型。

2.能够基于神经网络模型实现手写数字识别。

3.能够举一反三，基于神经网络模型实现肿瘤预测与分析。

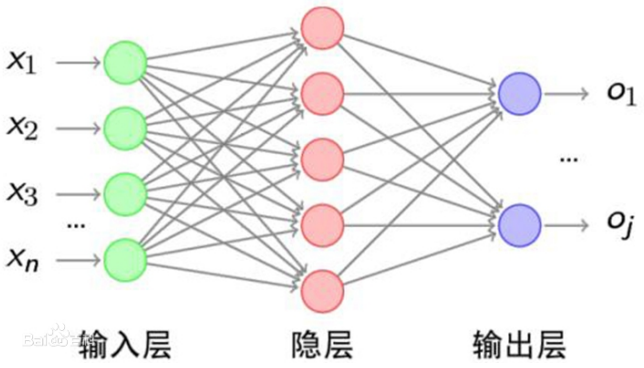
二、实验原理

（一）神经网络

人工神经网络（Artificial Neural Networks，简写为 ANN）也简称为神经网络（NN）或称作连接模型（Connection Model），它是一种模仿动物神经网络行为特征，进行分布式并行信息处理的算法数学模型。这种网络依靠系统的复杂程度，通过调整内部大量节点之间相互连接的关系，从而达到处理信息的目的。

人工神经网络是 20 世纪 80 年代以来人工智能领域兴起的研究热点。它从信息处理角度对人脑神经元网络进行抽象， 建立某种简单模型，按不同的连接方式组成不同的网络。在工程与学术界也常直接简称为神经网络或类神经网络。

神经网络是一种运算模型，由大量的节点（或称神经元）之间相互联接构成。每个节点代表一种特定的输出函数，称为激励函数（activation function）。每两个节点间的连接都代表一个对于通过该连接信号的加权值，称之为权重，这相当于人工神经网络的记忆。网络的输出则依网络的连接方式，权重值和激励函数的不同而不同。而网络自身通常都是对自然界某种算法或者函数的逼近，也可能是对一种逻辑策略的表达。



神经网络模型

人工神经网络无需事先确定输入输出之间映射关系的数学方程，仅通过自身的训练，学习某种规则，在给定输入值时得到最接近期望输出值的结果。作为一种智能信息处理系统，人工神经网络实现其功能的核心是算法。

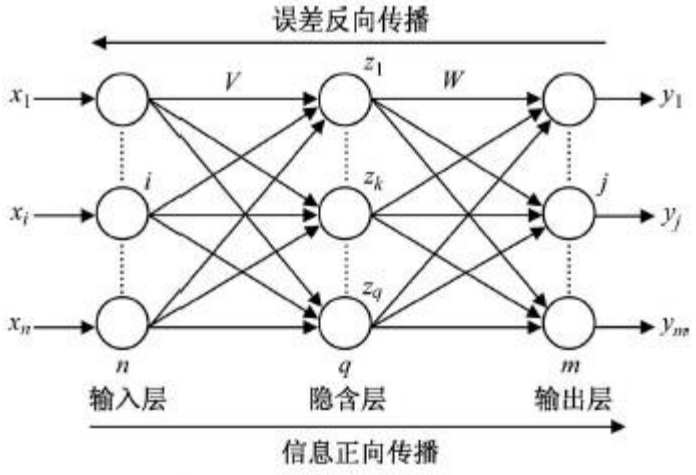
最近十多年来，人工神经网络的研究工作不断深入，已经取得了很大的进展，其在模式识别、智能机器人、自动控制、预测估计、生物、医学、经济等领域已成功地解决了许多现代计算机难以解决的实际问题，表现出了良好的智能特性。

（二）BP 神经网络

BP(back propagation)神经网络是 1986 年由 Rumelhart 和 McClelland 为首的科学家提出的概念，是应用最广泛的神经网络。

BP 神经网络是一种按误差反向传播(简称误差反传)训练的多层前馈网络，其算法称为 BP 算法。它的基本思想是梯度下降法，利用梯度搜索技术，以期使网络的实际输出值和期望输出值的误差均方差为最小。

基本 BP 算法包括信号的前向传播和误差的反向传播两个过程。即计算误差输出时按从输入到输出的方向进行，而调整权值和阈值则从输出到输入的方向进行。正向传播时，输入信号通过隐含层作用于输出节点，经过非线性变换，产生输出信号，若实际输出与期望输出不相符，则转入误差的反向传播过程。误差反传是将输出误差通过隐含层向输入层逐层反传，并将误差分摊给各层所有单元，以从各层获得的误差信号作为调整各单元权值的依据。通过调整输入节点与隐层节点的联接强度和隐层节点与输出节点的联接强度以及阈值，使误差沿梯度方向下降，经过反复学习训练，确定与最小误差相对应的网络参数(权值和阈值)，训练即告停止。此时经过训练的神经网络即能对类似样本的输入信息，自行处理输出误差最小的经过非线形转换的信息。



3 层 BP 神经网络模型

结构：BP 网络是在输入层与输出层之间增加若干层(一层或多层)神经元，这些神经元称为隐单元，它们与外界没有直接的联系，但其状态的改变，则能影响输入与输出之间的关系，每一层可以有若干个节点。

计算过程：由正向计算过程和反向计算过程组成。正向传播过程，输入模式从输入层经隐单元层逐层处理，并转向输出层，每一层神经元的状态只影响下一层神经元的状态。如果在输出层不能得到期望的输出，则转入反向传播，将误差信号沿原来的连接通路返回，通过修改各神经元的权值，使得误差信号最小。

我们可以使用 sklearn 中的神经网络模块 MLPClassifier 处理分类问题。下面分别介绍一下该函数的参数、属性、方法。

参数说明:

1. hidden_layer_sizes ：例如 hidden_layer_sizes=(50, 50)，表示有两层隐藏层，第一层隐藏层有 50 个神经元，第二层也有 50 个神经元。

2. activation：激活函数,{‘identity’, ‘logistic’, ‘tanh’, ‘relu’}, 默认 relu。

- identity：f(x) = x
- logistic：其实就是 sigmod,f(x) = 1 / (1 + exp(-x)).
- tanh：f(x) = tanh(x).
- relu：f(x) = max(0, x)

3. solver：{‘lbfgs’, ‘sgd’, ‘adam’}, 默认 adam，用来优化权重。

- lbfgs：quasi-Newton 方法的优化器。
- sgd：随机梯度下降。
- adam：Kingma, Diederik, and Jimmy Ba 提出的机遇随机梯度的优化器。

注意：默认 solver。‘adam’在相对较大的数据集上效果比较好（几千个样本或者更多），对小数据集来说，lbfgs 收敛更快效果也更好。

4.alpha：float,可选的，默认 0.0001,正则化项参数。

5.batch_size：int ,可选的，默认’auto’,随机优化的 minibatches 的大小 batch_size=min(200,n_samples)，如果 solver 是’lbfgs’，分类器将不使用 minibatch。

6.learning_rate：学习率,用于权重更新,只有当 solver 为’sgd’时使用，{‘constant’, ’ invscaling’, ‘adaptive’}，默认 constant。

-‘constant’：有’learning_rate_init’给定的恒定学习率

-‘incscaling’：随着时间 t 使用’power_t’的逆标度指数不断降低学习率 learning_rate_,effective_learning_rate= learning_rate_init / pow(t, power_t)

-‘adaptive’：只要训练损耗在下降，就保持学习率为’learning_rate_init’不变，当连续两次不能降低训练损耗或验证分数停止升高至少 tol 时，将当前学习率除以 5。

7. power_t：double，可选，default 0.5，只有 solver=’sgd’时使用，是逆扩展学习率的指数.当 learning_rate=’invscaling’，用来更新有效学习率。

8. max_iter：int，可选，默认 200，最大迭代次数。

9. random_stat：int 或 RandomState，可选，默认 None，随机数生成器的状态或种子。

10. shuffle：bool，可选，默认 True,只有当 solver=’sgd’或者‘adam’时使用，判断是否在每次迭代时对样本进行清洗。

11. tol：float，可选，默认 1e-4，优化的容忍度。

12. learning_rate_int：double,可选，默认 0.001，初始学习率，控制更新权重的补偿，只有当 solver=’sgd’ 或’adam’时使用。

13. verbose：bool，可选，默认 False,是否将过程打印到 stdout。

14. warm_start：bool，可选，默认 False,当设置成 True，使用之前的解决方法作为初始拟合，否则释放之前的解决方法。

15. momentum：float，默认 0.9,动量梯度下降更新，设置的范围应该 0.0-1.0. 只有 solver=’sgd’时使用。

16.nesterovs_momentum：boolean，默认 True，Whether to use Nesterov’s momentum. 只有 solver=’sgd’并且 momentum > 0 使用。

17. early_stopping:bool，默认 False,只有 solver=’sgd’或者’adam’时有效，判断当验证效果不再改善的时候是否终止训练，当为 True 时，自动选出 10%的训练数据用于验证并在两步连续迭代改善，低于 tol 时终止训练。

18. validation_fraction：float，可选，默认 0.1,用作早期停止验证的预留训练数据集的比例，早 0-1 之间，只当 early_stopping=True 有用。

19. beta_1：float，可选，默认 0.9，只有 solver=’adam’时使用，估计一阶矩向量的指数衰减速率，[0,1)之间。

20. beta_2：float,可选，默认 0.999，只有 solver=’adam’时使用估计二阶矩向量的指数衰减速率[0,1)之间。

21. epsilon：float,可选，默认 1e-8,只有 solver=’adam’时使用数值稳定值。

22.n_iter_no_change：nt，optional，默认值 10,不符合改进的最大历元数。仅在 solver =’sgd’或’adam’时有效。

参数	备注
hidden_layer_sizes	tuple，length = n_layers - 2，默认值（100，）第i个元素表示第i个隐藏层中的神经元数量。
激活	{'identity'，'logistic'，'tanh'，'relu'}，默认'relu' 隐藏层的激活函数：'identity'，无操作激活，对实现线性瓶颈很有用，返回f（x）= x；'logistic'，logistic sigmoid函数，返回f（x）= 1 /（1 + exp（-x））；'tanh'，双曲tan函数，返回f（x）= tanh（x）；'relu'，整流后的线性单位函数，返回f（x）= max（0，x）
slover	{'lbfgs'，'sgd'，'adam'}，默认'adam'。权重优化的求解器：'lbfgs'是准牛顿方法族的优化器；'sgd'指的是随机梯度下降。'adam'是指由Kingma，Diederik和Jimmy Ba提出的基于随机梯度的优化器。注意：默认解算器"adam"在相对较大的数据集（包含数千个训练样本或更多）方面在训练时间和验证分数方面都能很好地工作。但是，对于小型数据集，"lbfgs"可以更快地收敛并且表现更好。
alpha	float，可选，默认为0.0001。L2惩罚（正则化项）参数。
batch_size	int，optional，默认'auto'。用于随机优化器的minibatch的大小。如果slover是'lbfgs'，则分类器将不使用minibatch。设置为"auto"时，batch_size = min（200，n_samples）
learning_rate	{'常数'，'invscaling'，'自适应'}，默认'常数"。 用于权重更新。仅在solver ='sgd'时使用。'constant'是'learning_rate_init'给出的恒定学习率；'invscaling'使用'power_t'的逆缩放指数在每个时间步't'逐渐降低学习速率learning_rate_， effective_learning_rate = learning_rate_init / pow（t，power_t）；只要训练损失不断减少，"adaptive"将学习速率保持为"learning_rate_init"。每当两个连续的时期未能将训练损失减少至少tol，或者如果'early_stopping'开启则未能将验证分数增加至少tol，则将当前学习速率除以5。
learning_rate_init	double，可选，默认为0.001。使用初始学习率。它控制更新权重的步长。仅在solver ='sgd'或'adam'时使用。
power_t	double，可选，默认为0.5。反缩放学习率的指数。当learning_rate设置为"invscaling"时，它用于更新有效学习率。仅在solver ='sgd'时使用。
max_iter	int，optional，默认值200。最大迭代次数。solver迭代直到收敛（由'tol'确定）或这个迭代次数。对于随机解算器（'sgd'，'adam'），请注意，这决定了时期的数量（每个数据点的使用次数），而不是梯度步数。
shuffle	bool，可选，默认为True。仅在solver ='sgd'或'adam'时使用。是否在每次迭代中对样本进行洗牌。
random_state	int，RandomState实例或None，可选，默认无随机数生成器的状态或种子。如果是int，则random_state是随机数生成器使用的种子;如果是RandomState实例，则random_state是随机数生成器;如果为None，则随机数生成器是np.random使用的RandomState实例。
tol	float，optional，默认1e-4 优化的容忍度，容差优化。当n_iter_no_change连续迭代的损失或分数没有提高至少tol时，除非将learning_rate设置为'adaptive'，否则认为会达到收敛并且训练停止。
verbose	bool，可选，默认为False 是否将进度消息打印到stdout。
warm_start	bool，可选，默认为False，设置为True时，重用上一次调用的解决方案以适合初始化，否则，只需擦除以前的解决方案。请参阅词汇表。
momentum	float，默认0.9，梯度下降更新的动量。应该在0和1之间。仅在solver ='sgd'时使用。
nesterovs_momentum	布尔值，默认为True。是否使用Nesterov的势头。仅在solver ='sgd'和momentum> 0时使用。
early_stopping	bool，默认为False。当验证评分没有改善时，是否使用提前停止来终止培训。如果设置为true，它将自动留出10％的训练数据作为验证，并在验证得分没有改善至少为n_iter_no_change连续时期的tol时终止训练。仅在solver ='sgd'或'adam'时有效
validation_fraction	float，optional，默认值为0.1。将训练数据的比例留作早期停止的验证集。必须介于0和1之间。仅在early_stopping为True时使用
beta_1	float，optional，默认值为0.9，估计一阶矩向量的指数衰减率应为[0,1)。仅在solver ='adam'时使用
beta_2	float，可选，默认为0.999,估计一阶矩向量的指数衰减率应为[0,1)。仅在solver ='adam'时使用
epsilon	float，optional，默认值1e-8, adam稳定性的价值。 仅在solver ='adam'时使用
n_iter_no_change	int，optional，默认值10,不符合改进的最大历元数。 仅在solver ='sgd'或'adam'时有效

属性说明：

- classes_：每个输出的类标签。
- loss_：损失函数计算出来的当前损失值。
- coefs_：列表中的第 i 个元素表示 i 层的权重矩阵。
- intercepts_：列表中第 i 个元素代表 i+1 层的偏差向量。
- n_iter_：迭代次数。
- n_layers_：层数。
- n_outputs_：输出的个数。
- out_activation_：输出激活函数的名称。

属性	备注
classes_	array or list of array of shape （n_classes，）每个输出的类标签。
loss_	float,使用损失函数计算的当前损失。
coefs_	list，length n_layers - 1,列表中的第i个元素表示对应于层i的权重矩阵。
intercepts_	list，length n_layers - 1,列表中的第i个元素表示对应于层i + 1的偏置矢量。
n_iter_	int，迭代次数。
n_layers_	int,层数。
n_outputs_	int,输出的个数。
out_activation_	string，输出激活函数的名称。

方法说明：

- fit(X,y)：拟合。
- get_params([deep])：获取参数。
- predict(X)：使用 MLP 进行预测。
- predic_log_proba(X)：返回对数概率估计。
- predic_proba(X)：概率估计。
- score(X,y[,sample_weight])：返回给定测试数据和标签上的平均准确度。
- set_params(**params)：设置参数。

方法	备注
fit（X，y）	使模型适合数据矩阵X和目标y。
get_params（[deep]）	获取此估算器的参数。
predict（X）	使用多层感知器分类器进行预测
predict_log_proba（X）	返回概率估计的对数。
predict_proba（X）	概率估计。
score（X，y [，sample_weight]）	返回给定测试数据和标签的平均准确度。
set_params（** params）	设置此估算器的参数。

（三）MINST 数据集

MNIST 数据集是一个专门用来训练各种图像处理系统的庞大数据集，实际上它是从 NIST 原始数据及中提取的，其训练集和测试集有一半是来自 NIST 数据集的训练集，而另一半是来自 NIST 的测试集。

MNIST 数据集是机器学习领域中非常经典的一个数据集，由 60000 个训练样本和 10000 个测试样本组成，每个样本都是一张 28 * 28 像素的灰度手写数字图片。

一共 4 个文件，训练集图片、训练集标签、测试集图片、测试集标签。图片是 28*28 的像素矩阵，标签为 0~9 共 10 个数字。

文件名称	大小	内容
train-images-idx3-ubyte.gz	9,681 kb	55000张训练集，5000张验证集
train-labels-idx1-ubyte.gz	29 kb	训练集图片对应的标签
t10k-images-idx3-ubyte.gz	1,611 kb	10000张测试集
t10k-labels-idx1-ubyte.gz	5 kb	测试集图片对应的标签

本次实验把该数据集用于手写数字识别，属于典型的图像多分类问题。

（四）威斯康星乳腺癌数据集

威斯康星乳腺癌数据集来自美国威斯康星州的乳腺癌诊断数据集。医疗人员采集了患者乳腺肿块经过细针穿刺 (FNA) 后的数字化图像，并且对这些数字图像进行了特征提取，这些特征可以描述图像中的细胞核呈现。

该数据集中肿瘤是一个非常经典的用于医疗病情分析的数据集，包括 569 个病例的数据样本，每个样本具有 30 个特征。样本共分为两类：恶性(Malignant)和良性(Benign)。

字段	含义
ID	ID标识
diagnosis	M/B（M：恶性，B：良性）
radius_mean	半径（点中心到边缘的距离）平均值
texture_mean	文理（灰度值的标准差）平均值
perimeter_mean	周长 平均值
area_mean	面积 平均值
smoothness_mean	平滑程度（半径内的局部变化）平均值
compactness_mean	紧密度（=周长*周长/面积-1.0）平均值
concavity_mean	凹度（轮廓凹部的严重程度）平均值
concave points_mean	凹缝（轮廓的凹部分）平均值
symmetry_mean	对称性 平均值
fractal_dimension_mean	分形维数（=海岸线近似-1）平均值
radius_se	半径（点中心到边缘的距离）标准差
texture_se	文理（灰度值的标准差）标准差
perimeter_se	周长 标准差
area_se	面积 标准差
smoothness_se	平滑程度（半径内的局部变化）标准差
compactness_se	紧密度（=周长*周长/面积-1.0）标准差
concavity_se	凹度（轮廓凹部的严重程度）标准差
concave points_se	凹缝（轮廓的凹部分）标准差
symmetry_se	对称性标准差
fractal_dimension_se	分形维数（=海岸线近似-1）标准差
radius_worst	半径（点中心到边缘的距离）最大值
texture_worst	文理（灰度值的标准差）最大值
perimeter_worst	周长 最大值
area_worst	面积 最大值
smoothness_worst	平滑程度（半径内的局部变化）最大值
compactness_worst	紧密度（=周长*周长/面积-1.0）最大值
concavity_worst	凹度（轮廓凹部的严重程度）最大值
concave points_worst	凹缝（轮廓的凹部分）最大值
symmetry_worst	对称性 最大值
fractal_dimension_worst	分形维数（=海岸线近似-1）最大值

威斯康辛乳腺癌数据集特征

属性信息：

ID——身份证号码

diagnose——诊断结果（M=恶性，B=良性）

其中'B'代表良性，包含 357 例；'M'代表恶性，包含 212 例。

计算每个细胞核的 10 个实值特征：

1） radius_mean：半径（从中心到周界各点的平均距离）

2） texture_mean：纹理（灰度值的标准偏差）

3） perimeter_mean：周长

4） area_mean：面积

5） smoothness_mean ：平滑度（半径长度的局部变化）

6） compactness_mean：密实度/紧密度（周长^2/面积-1.0）

7） concavity_mean：凹度（轮廓凹陷部分的严重程度）

8） concave points_mea ：凹点（轮廓凹面部分的数量）

9） symmetry_mean：对称性

10）fractal_dimension_mean：分形维数（“海岸线近似值”-1）

Mean、se、worst：为每个图像计算这些特征，产生了 30 个特征。所有特征值用四个有效数字重新编码。

- 包含 mean 的数据——平均值。
- 包含 se 的数据——标准误差。
- 包含 worst 的数据——最差值或最大值（三者中的平均值最大值），是最严重的数据样例(最坏值)。

三、实验环境

计算机；网络环境。

四、实验内容及步骤

（一）实验内容

1.对于给定的例题，基于 BP 神经网络算法进行手写数字识别的练习。

2.对于给定的项目，自行编写程序，使用 BP 神经网络实现肿瘤预测与分析。

3.扩展内容：学习扩展实验 4、5、6 中的深度神经网络（DNN）、卷积神经网络（CNN）、循环神经网络（RNN）内容，对手写数字识别、猫狗分类、机器阅读理解项目进行练习。

（二）实验步骤

1.在熟悉了实验原理的基础上，进行“实验六：神经网络”例题部分的实操练习。

- 例题：手写数字识别

2.在步骤 2 例题练习的基础上，对给定的实操项目，自行编写程序，使用 BP 神经网络实现肿瘤预测与分析。

- 实操项目——肿瘤预测与分析（神经网络）

3.扩展内容（选做）：学习扩展实验四、五、六中的深度神经网络（DNN）、卷积神经网络（CNN）、循环神经网络（RNN）内容，结合视频讲解，对手写数字识别、猫狗分类、机器阅读理解项目进行练习。（此部分内容选做，练习后不用写实验报告）

4.完成实验报告。

请严格基于实验报告的模板撰写实验报告。

本课程所有实验全部结束后再统一打印。

附：实操项目要求

- **实操项目——肿瘤预测与分析（神经网络）**

基于威斯康星乳腺癌数据集，搭建BP神经网络，实现肿瘤预测与分析。

【实验要求】

1.加载sklearn自带的数据集，探索数据。

2.划分训练集与测试集。

3.建立BP模型（评估后可进行调参，从而选择最优参数）。

4.进行模型训练。

5.进行模型预测，对真实数据和预测数据进行可视化（用Axes3D绘制3d散点图）。

6.进行模型评估，并进行预测结果指标统计（统计每一类别的预测准确率、召回率、F1分数）。

7.计算混淆矩阵，并用热力图显示。

注：混淆矩阵（confusion matrix）衡量的是一个分类器分类的准确程度。

混淆矩阵的每一列代表了预测类别 ,每一列的总数表示预测为该类别的数据的数目;每一行代表了数据的真实归属类别 ，每一行的数据总数表示该类别的数据实例的数目。

封面设计： 贾丽

地 址：中国河北省秦皇岛市河北大街 438 号
邮 编：066004
电 话：0335-8057068
传 真：0335-8057068
网 址：http://jwc.ysu.edu.cn