**《人工智能概论》大作业**

**作 业 名 称：基于ANN与KNN的图像分类**

**专 业 软件工程**

**组 长 姓 名 \*\*\*\*\*\*\***

**学 号 \*\*\*\*\*\*\*\*\*\*\*\***

**联 系 方 式 \*\*\*\*\*\*\*\*\*\*\*\*\*\*\***

|  |  |  |  |
| --- | --- | --- | --- |
| **小组成绩** |  | **评阅人** |  |
| |  |  |  |  |  |  | | --- | --- | --- | --- | --- | --- | | 贡献排序 | 1 | 2 | 3 | 4 | 5 | | 姓名 | | 学号 | | 权值 | | 成绩 | | | | |

**\*\*\*\*\*大学软件学院**

## 摘要

### 摘要

当下人工智能行业发展迅猛，在各个领域都有涉及。人工智能的一大用处就是分类操作，在智能分类领域中，人工智能可以进行科学的模拟操作，以达到人们的相关需求。

在本次大作业中，需要使用相关的分类器，对200类比的图片进行分类。首先我们选择了ANN模型即人工神经网络模型，这个模型使用神经网络技术模拟人体神经网络进行相关的训练，是对人脑组织结构和运行机制的某种抽象、简化和模拟。因此与传统的数据处理方法相比，神经网络技术在处理模糊数据、随机性数据、非线性数据方面具有明显优势，对规模大、结构复杂、信息不明确的系统尤为适用。

然后我们选择了KNN模型，即K最近邻（KNN，K-NearestNeighbor）分类算法。是数据挖掘分类技术中最简单的方法之一。是使用纯数学方法的适用于样本容量比较大的类域的自动分类方法。

因为图片样本的相关特征已经提取出来，而且标签也已经得到了标注，所以我们只需要进行构建分类器并训练的操作即可。我们使用以上两种模型进行构建分类，并观察相关参数对结果指标的影响，最后得出结论。

### 1.2 模型评判参数

在介绍模型评判参数之前，首先要定义一些变量。

混淆矩阵：

True Positive(真正，TP)：将正类预测为正类数

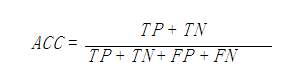
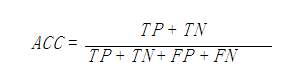
True Negative(真负，TN)：将负类预测为负类数

False Positive(假正，FP)：将负类预测为正类数误报 (Type I error)

False Negative(假负，FN)：将正类预测为负类数→漏报 (Type II error)

准确率：分类正确的比例，其值为分类正确的样本数/样本总数，计算公式为

准确率：分类正确的比例，其值为分类正确的样本数/样本总数，计算公式为



错误率：被错分的比例，其值可为1-准确率，也可通过error rate = (FP+FN)/(TP+TN+FP+FN)求出。

灵敏度：正例中被分对的比例，sensitive = TP/P。

特效度：负例中被分对的比例，specificity = TN/N。

精确度：被分为正例的示例中实际为正例的比例，P=TP/（TP+FP）。

召回率：有多少个正例被分为了正例，R=TP/（TP+FN）。

1. Macro f1: 指将所有混淆矩阵的P和R都算出来，求得的平均P和R求出的F1.公式为：

IMG_256

1. Micro f1: 指将所有混淆矩阵中的数都算出来，求得的平均混淆矩阵计算出来的P和R计算出来的F1;
2. 训练模型准确率：用训练数据求出的准确率;
3. 测试模型准确率：用测试数据求出的准确率;
4. 训练时间：指完成一整次训练需要消耗的时间。

## ANN说明

### 2.1 ANN模型原理与介绍

ANN是指由大量的处理单元(神经元) 互相连接而形成的复杂网络结构，是对人脑组织结构和运行机制的某种抽象、简化和模拟。人工神经网络（Artificial Neural Network，简称ANN ），以数学模型模拟神经元活动，是基于模仿大脑神经网络结构和功能而建立的一种信息处理系统。

人工神经网络有多层和单层之分，每一层包含若干神经元，各神经元之间用带可变权重的有向弧连接，网络通过对已知信息的反复学习训练，通过逐步调整改变神经元连接权重的方法，达到处理信息、模拟输入输出之间关系的目的。它不需要知道输入输出之间的确切关系，不需大量参数，只需要知道引起输出变化的非恒定因素，即非常量性参数。因此与传统的数据处理方法相比，神经网络技术在处理模糊数据、随机性数据、非线性数据方面具有明显优势，对规模大、结构复杂、信息不明确的系统尤为适用。

由Minsley和Papert提出的多层前向神经元网络(也称多层感知器)是目前最为常用的网络结构

人工神经网络是由大量处理单元互联组成的非线性、自适应信息处理系统。它是在现代神经科学研究成果的基础上提出的，试图通过模拟大脑神经网络处理、记忆信息的方式进行信息处理。人工神经网络具有四个基本特征：

（1）非线性 非线性关系是自然界的普遍特性。大脑的智慧就是一种非线性现象。人工神经元处于激活或抑制二种不同的状态，这种行为在数学上表现为一种非线性关系。具有阈值的神经元构成的网络具有更好的性能，可以提高容错性和存储容量。

（2）非局限性 一个神经网络通常由多个神经元广泛连接而成。一个系统的整体行为不仅取决于单个神经元的特征，而且可能主要由单元之间的相互作用、相互连接所决定。通过单元之间的大量连接模拟大脑的非局限性。联想记忆是非局限性的典型例子。

（3）非常定性 人工神经网络具有自适应、自组织、自学习能力。神经网络不但处理的信息可以有各种变化，而且在处理信息的同时，非线性动力系统本身也在不断变化。经常采用迭代过程描写动力系统的演化过程。

（4）非凸性 一个系统的演化方向，在一定条件下将取决于某个特定的状态函数。例如能量函数，它的极值相应于系统比较稳定的状态。非凸性是指这种函数有多个极值，故系统具有多个较稳定的平衡态，这将导致系统演化的多样性。

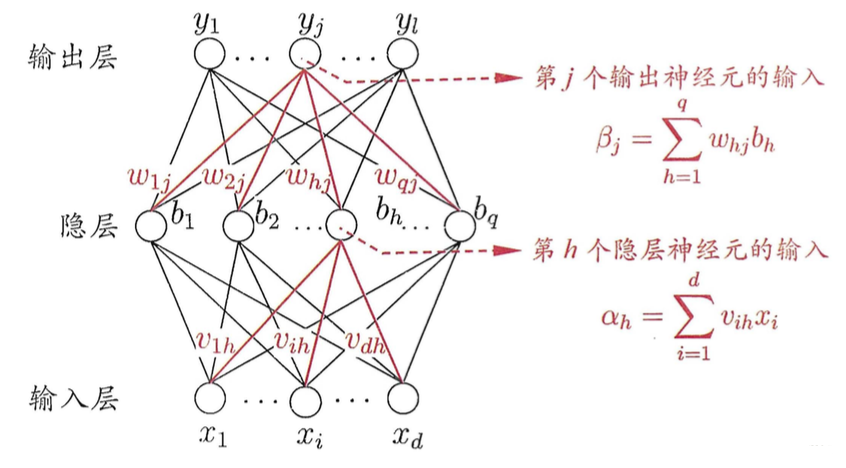
人工神经网络的特点和优越性，主要表现在三个方面：

第一，具有自学习功能。例如实现图像识别时，只在先把许多不同的图像样板和对应的应识别的结果输入人工神经网络，网络就会通过自学习功能，慢慢学会识别类似的图像。自学习功能对于预测有特别重要的意义。预期未来的人工神经网络计算机将为人类提供经济预测、市场预测、效益预测，其应用前途是很远大的。

第二，具有联想存储功能。用人工神经网络的反馈网络就可以实现这种联想。

第三，具有高速寻找优化解的能力。寻找一个复杂问题的优化解，往往需要很大的计算量，利用一个针对某问题而设计的反馈型人工神经网络，发挥计算机的高速运算能力，可能很快找到优化解。

**● 神经元输入输出的计算方法**：假设一个训练样本为，对应的输出向量为，l为类别个数，即输出向量是类别的独热编码。隐藏层第h个节点的输入权重为，对应的偏移量为。第j个输出层节点的输入权重为，对应的偏移量为。q为隐藏层节点个数。令为sigmoid函数，如图所示，



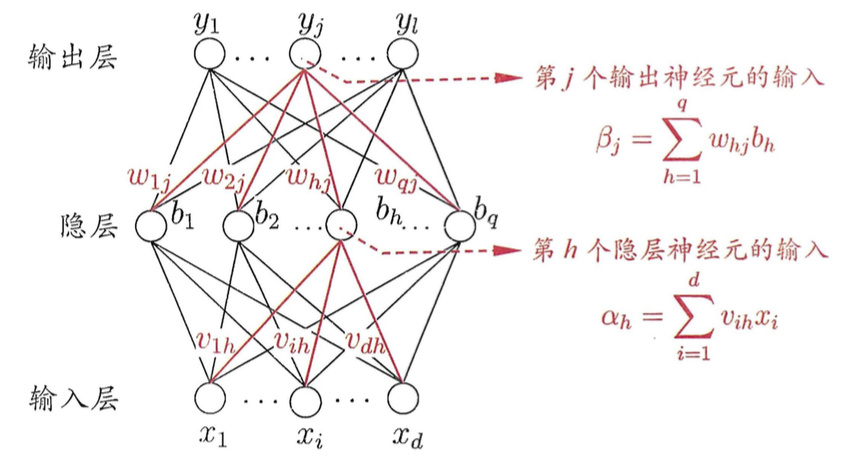
第j个输出神经元的输入为

第j个输出神经元的输出为

第h个隐层神经元的输入为

第h个隐层神经元的输出为

**● 训练算法的计算方法**：如下图所示，假设一个三类别分类问题，



对一个训练样本，l为类别个数，设神经网络的输出为。即，那么网络在输出节点上的均方误差为。BP算法的本质就是梯度下降，在训练神经网络的时候，任意参数的迭代更新公式为：。

1. 隐藏层到输出层的权值的更新过程如下，构成了，影响了，最终影响了E，因此由链式法则可知：









因此。

1. 输出层的阈值更新如下：







此时，误差被逆向传播到隐藏层，同理可得，和。

### 2.2 ANN模型参数介绍

1. 隐藏层层数：指网络中输入层与输出层之间的非线性层数，多个隐藏层可以用于拟合非线性函数。层数越深，理论上拟合函数的能力增强，但可能发生过拟合问题，同时也会增加训练难度，使模型难以收敛。在训练过程中，每个隐藏层中的神经元将前一层的值进行加权线性求和转换IMG_256，再通过非线性激活函数IMG_257。 输出层接收到的值是最后一个隐藏层的输出经过变换而来的。

2. 学习率种类：指使用的学习率的步长。学习率(Learning rate)作为监督学习以及深度学习中重要的超参，其决定着目标函数能否收敛到局部最小值以及何时收敛到最小值。合适的学习率能够使目标函数在合适的时间内收敛到局部最小值。

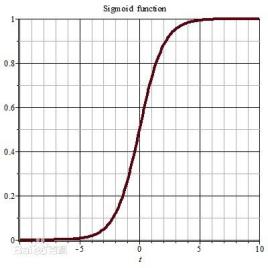
3. 迭代次数：指迭代运算中循环次数的上限，当循环次数达到这一值时停止训练。一般来说，学习次数越多，学习效果越好，但过多的迭代次数会消耗大量时间，还可能产生过拟合，所以应选用合适的迭代次数。

4. 窗口大小：指一次训练所选取的样本数。窗口大小影响模型的优化程度和速度。同时其直接影响到GPU内存的使用情况。无窗口训练的梯度是准确的，但由于训练速度较慢，只适合小样本的数据；随着窗口的增大，梯度会越来越准确，但梯度非常准确后，再增加窗口大小就无太大意义了。

5. 单层隐藏层神经元个数：指一个隐藏层中有多少个神经元。神经元太少将导致欠拟合(underfitting)，但使用过多的神经元可能导致过拟合，且会增加训练时间。神经元的最佳数量需要经过不断实验来获得。

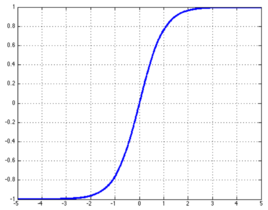
6. 激活函数：指神经网络中用于计算输入和偏差的加权和的函数，用于确定神经元是否可以释放。如果不加激活函数，则无论由多少层隐含层，最终结果仍会是原始输入的线性变化，所以ANN中每个隐含层都会配一个激活函数，用来提供非线性变化。激活函数具有改善数据学习模式的能力，从而实现了特征检测的自动化，并证明它们在神经网络中的合理性，且对跨领域分类很有帮助。比较常见的几种激活函数由Sigmoid函数，Tanh函数以及ReLU函数。

* **Sigmoid函数**：这是一个在生物学中常见的S型函数，也称为S型生长曲线。在信息科学中，由于其单增以及反函数单增等性质，Sigmoid函数常被用作神经网络的阈值函数，将变量映射到0,1之间。公式如下：。



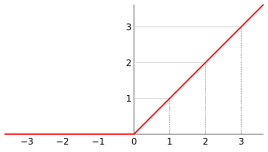
Sigmoid方法图像

* **Tanh函数**：这双曲正切函数，由基本双曲函数双曲正弦和双曲余弦推导而来。公式如下：。



Tanh方法图像

* **ReLU函数**：用于隐层神经元输出，公式如下： 。



Relu方法图像

### 2.3 相关指标介绍

训练停止条件

迭代完成且呈现收敛趋势

(迭代法的收敛性：x(0)是人为地给定的初始迭代向量，G和d分别为n×n矩阵和n维实向量，且满足QA=I-G，Qb=d，此处Q是某一非奇异的n阶方阵.若存在x\*∈Rn使对任意的初值x(0)，由表达式产生的序列{x(k)}都收敛到x\*，则称表达式是收敛的)

训练时间：训练花费时间。

迭代次数：迭代运算过程中循环的次数。

识别率：被正确识别的输入模式的数量占被识别的所有输入模式的总数的百分比。是衡量模式识别系统性能的重要指标，其值越高则识别率越高，系统性能越好。

训练集：确定模型后，用于训练参数，训练的是普通参数（每多加入一个数据对模型进行训练，模型中就会受到影响的参数，通过多次迭代不断更新，是一个梯度下降的过程）而不是超参数（超参数是指训练开始之前设置的参数，超参数的选择与训练过程实际上是独立的，训练过程不会影响超参数。但是训练结束后可以根据训练结果考虑超参数是否可优化，可优化的话就调整超参数的值开始下一次训练）。

测试集：虽然验证集没有对模型的参数产生影响，但是我们却根据验证集的测试结果的准确度来调整参数，所以验证集对结果还是有影响的，即使得模型在验证集上达到最优。就是在很多个模型中，验证集选择了代价函数最小的一个模型。虽然在这个模型上代价很小，但并不代表在其他数据上代价也小。所以需要一个完全没有经过训练的测试集来再最后测试模型的准确率。

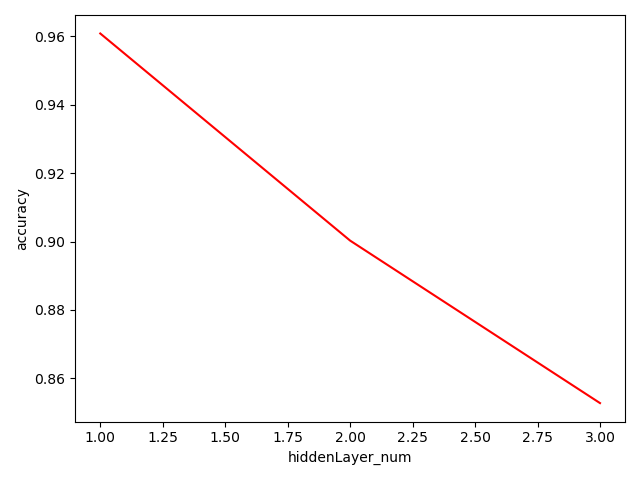
### 2.4 实验结果展示

我们知道，模型训练的最终终止条件是在我们所设定的迭代次数内，模型的效果呈现收敛趋势，因此，迭代次数过小，模型的效果可能是欠拟合的，迭代次数过多，模型的效果是过拟合且会浪费过多时间，经过试验，我们在训练的过程中，迭代次数在5000左右时，拟合效果良好，因此，在以下的分析中，将不再对迭代次数的影响进行讨论。

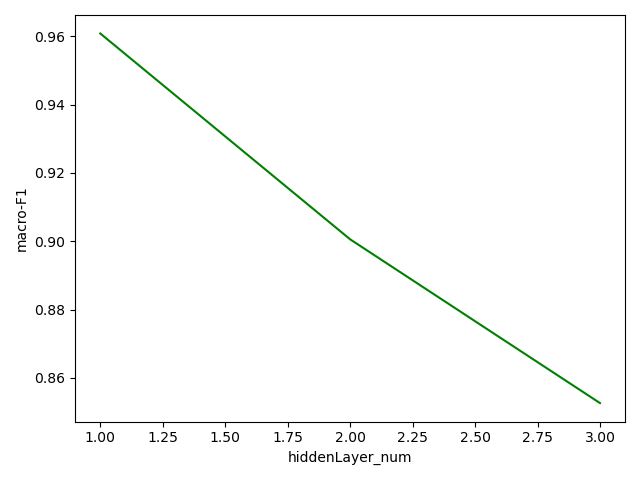
因为需要考虑的参数很多，因此我们在对某一种参数进行讨论时，其他变量的参数将取相同的数值。

#### 2.4.1 隐藏层的层数

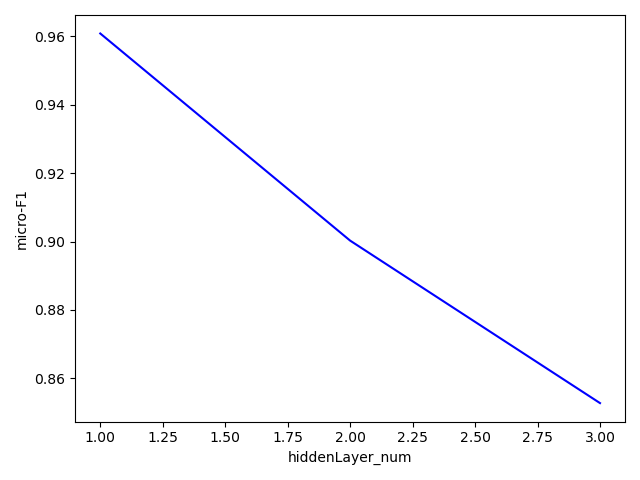
每层固定50个，1层->2层->3层



准确率随隐藏层的变化曲线



Macro f1随隐藏层的变化曲线



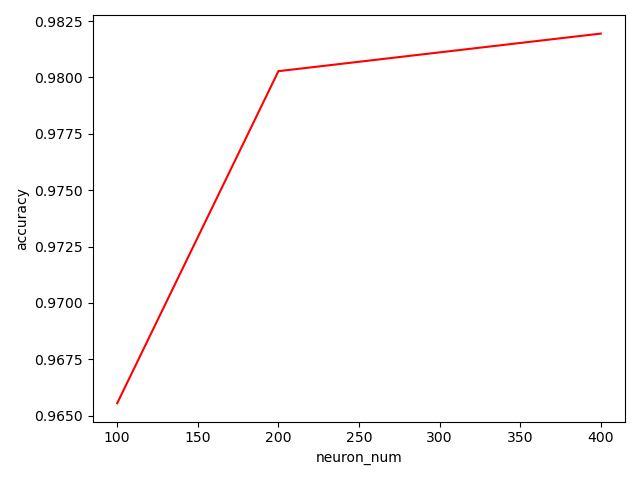
Micro f1随隐藏层的变化曲线

在这里，我们能够发现当我们的隐藏层层数增加时，模型的效果反而是衰退了。

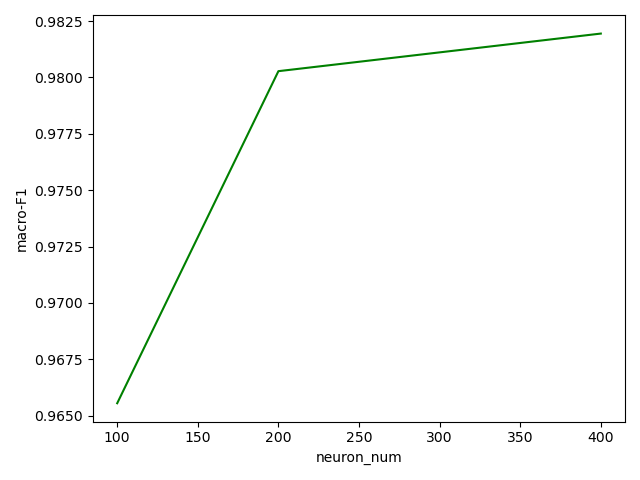
|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Macro-F1 | Micro-F1 |
| 1层 | 0.9608333333333332 | 0.9608537368332261 | 0.9608333333333332 |
| 2层 | 0.9002777777777776 | 0.9005720715926264 | 0.9002777777777776 |
| 3层 | 0.8527777777777776 | 0.8526058902872711 | 0.8527777777777776 |

#### 2.4.2 隐藏层神经元节点的个数

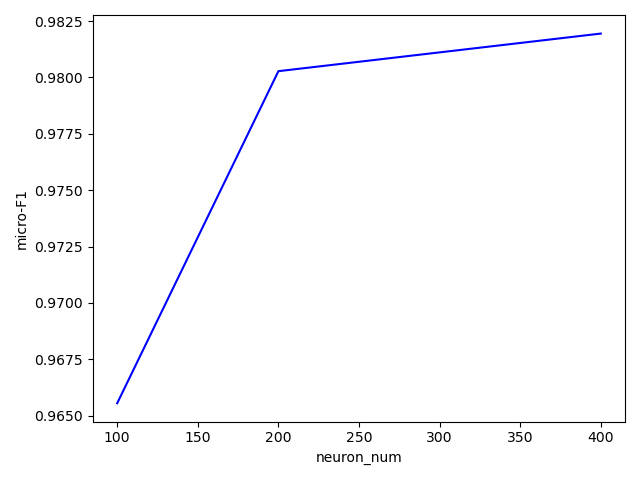
100->200->400



准确率随隐藏层神经元节点个数的变化曲线



Macro f1随隐藏层神经元节点个数的变化曲线



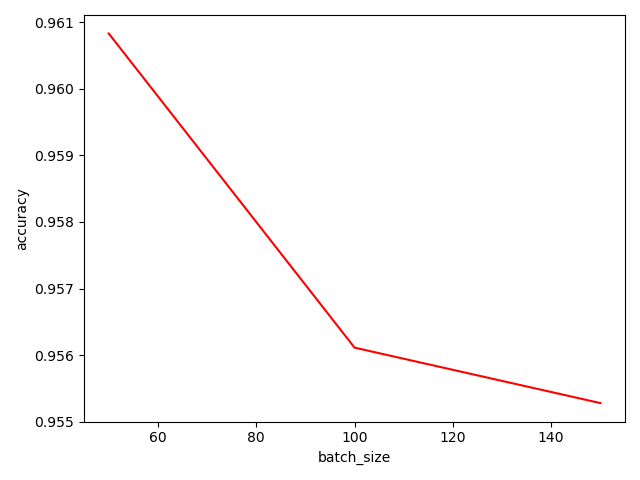
Micro f1随隐藏层神经元节点个数的变化曲线

我们发现，当隐藏层的神经元节点增加的时候，模型的性能是越来越好的，但其增长的速率是不断变小的。

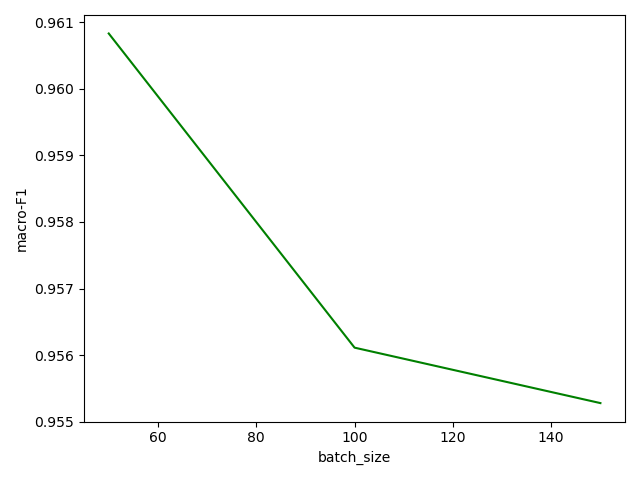
|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Macro-F1 | Micro-F1 |
| 100个 | 0.9655555555555555 | 0.9657414691187194 | 0.9655555555555555 |
| 200个 | 0.9802777777777778 | 0.980292180327784 | 0.9802777777777778 |
| 400个 | 0.9819444444444444 | 0.981951361824427 | 0.9819444444444444 |

#### 2.4.3 窗口大小

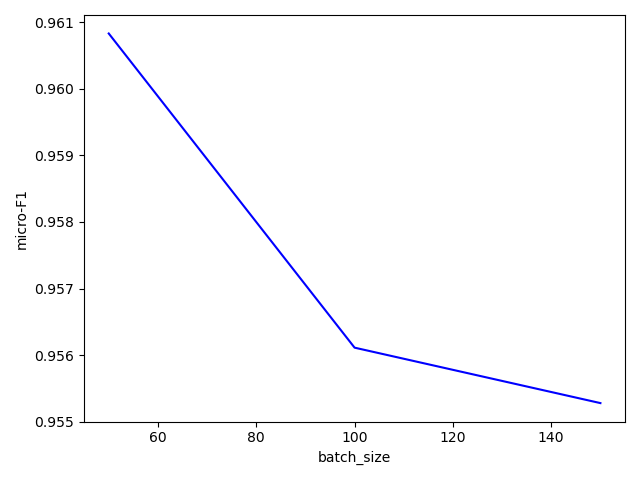
50->100->150



准确率随窗口大小的变化曲线



Macro f1随窗口大小的变化曲线

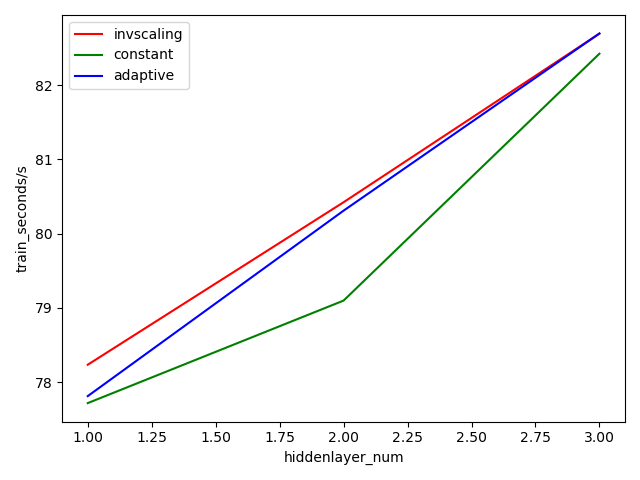


Micro f1随窗口大小的变化曲线

通过图像可以观察到，对于我们的ANN模型来说，窗口大小越大，模型的性能越差。

|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Macro-F1 | Macro-F1 |
| 50 | 0.9608333333333332 | 0.9608537368332261 | 0.9608333333333332 |
| 100 | 0.9561111111111111 | 0.9560975986343635 | 0.9561111111111111 |
| 150 | 0.9552777777777777 | 0.9555034520767116 | 0.9552777777777777 |

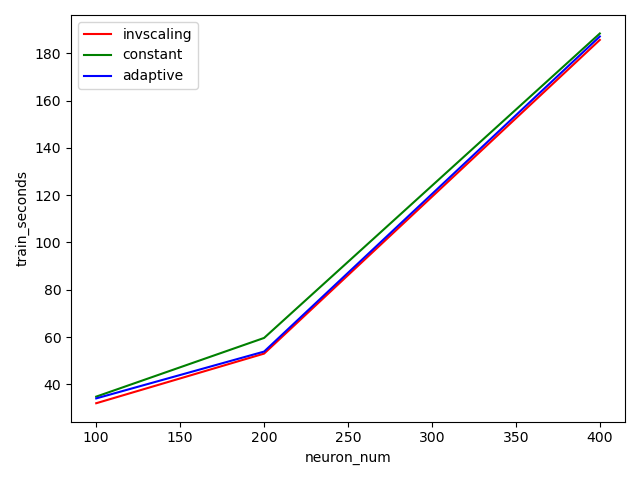
#### 2.4.4 学习率



随着层数的增加，不同学习率训练时间的变化

可以观察到，随着层数的不断叠加，各种学习率的训练时间是不断增加的，且constant方法的表现是最好的。

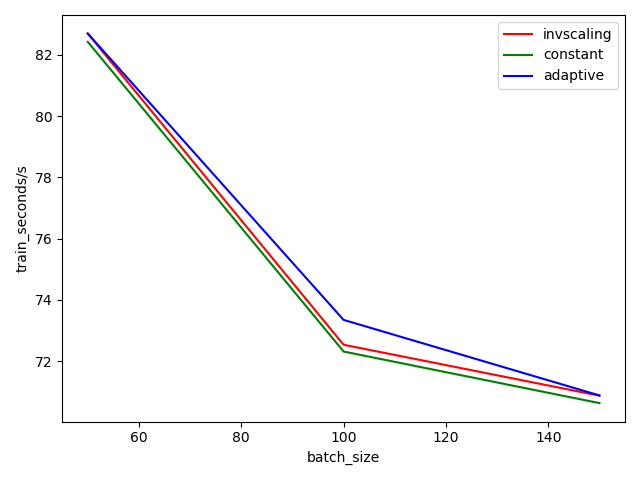
|  |  |  |  |
| --- | --- | --- | --- |
|  | invscaling | constant | adaptive |
| 1层 | 78.23547966000001 | 77.7188933999998 | 77.81292566000047 |
| 2层 | 80.345678454938895 | 80.0234578478455 | 79.64839478932734 |
| 3层 | 82.69218050000018 | 82.42171543999939 | 82.69385183999984 |



随着单层神经元个数的增加，不同学习率训练时间的变化

可以观察到，随着隐藏层中神经元个数的不断增加，各学习率的其训练时间是不断增加的，其中invscaling相对较好。

|  |  |  |  |
| --- | --- | --- | --- |
|  | invscaling | constant | adaptive |
| 100个 | 32.01377033999997 | 34.805954499999984 | 34.05963100000008 |
| 200个 | 52.995126360000086 | 59.619636000000014 | 53.854412799999864 |
| 400个 | 185.63600004000017 | 188.34518989999992 | 187.09286480000037 |



随着窗口大小的增加，不同学习率训练时间的变化

可以观察到，随着batch\_size的增加，其单次训练时间是呈现出减少的趋势的，其中constant的表现最好。

|  |  |  |  |
| --- | --- | --- | --- |
|  | invscaling | constant | adaptive |
| 50 | 82.69218050000018 | 82.42171543999939 | 82.69385183999984 |
| 100 | 72.53692752000043 | 72.31512332000057 | 73.34816916000128 |
| 150 | 70.87542775999991 | 70.63270916000037 | 70.88269793999935 |

在这里，需要对学习率进行说明，每种学习率（这里的学习率是最终一定能使得收敛的）最终到达的终点是相近的，也就是学习率种类对模型的效果影响是不明显的，它影响的是我们到达收敛趋势阈值的快慢程度，也就是我们的训练时间。

### 2.5 实验结论

通过进行 ANN 模型的建立、训练和结果分析，我们发现，使用 ANN 模型进行分类处理，模型的训练数据集的精确度最高可以达到 98.20%，测试数据集的精确度可以达到 95.53%，说明使用 ANN 模型进行此场景的学习过程相对较好，但是要注意选取激活函数和权值优化策略。本例中，最好使用 relu 激活函数和 sgd 权值优化策略，可以获得客观的训练结果和较短的训练时间。

## KNN说明

### 3.1 KNN模型原理与介绍

KNN算法的核心思想是如果一个样本在特征空间中的k个最相邻的样本中的大多数属于某一个类别，则该样本也属于这个类别，并具有这个类别上样本的特性。该方法在确定分类决策上只依据最邻近的一个或者几个样本的类别来决定待分样本所属的类别。

KNN是一种即可用于分类又可用于回归的机器学习算法。对于给定测试样本，基于距离度量找出训练集中与其最靠近的K个训练样本，然后基于这K个“邻居”的信息来进行预测。

在分类任务中可使用投票法，选择这K个样本中出现最多的类别标记作为预测结果；在回归任务中可使用平均法，将这K个样本的实值输出标记的平均值作为预测结果。当然还可以基于距离远近程度进行加权平均等方法。

闵可夫斯基距离（minkowski）：



欧氏距离（euclidean，p=2）：



曼哈顿距离（manhattan，p=1）：



切比雪夫距离（chebyshev，p=∞）:



下面分析k值过大和过小造成的影响：

k值较小，就相当于用较小的领域中的训练实例进行预测，训练误差近似误差小（偏差小），泛化误差会增大（方差大），换句话说，K值较小就意味着整体模型变得复杂，容易发生过拟合；

k值较大，就相当于用较大领域中的训练实例进行预测，泛化误差小（方差小），但缺点是近似误差大（偏差大），换句话说，K值较大就意味着整体模型变得简单，容易发生欠拟合；一个极端是k等于样本数m，则完全没有分类，此时无论输入实例是什么，都只是简单的预测它属于在训练实例中最多的类，模型过于简单。

对于k值的选择，没有一个固定的经验（sklearn默认为5），一般根据样本的分布，选择一个较小的值，可以通过交叉验证选择一个合适的k值。

KNN 算法一般是用多数表决方法，即由输入实例的K个邻近的多数类决定输入实例的类。这也是经验风险最小化的结果。

接下来对KNN算法的思想总结一下：就是在训练集中数据和标签已知的情况下，输入测试数据，将测试数据的特征与训练集中对应的特征进行相互比较，找到训练集中与之最为相似的前K个数据，则该测试数据对应的类别就是K个数据中出现次数最多的那个分类，其算法的描述为：

1）计算测试数据与各个训练数据之间的距离；

2）按照距离的递增关系进行排序；

3）选取距离最小的K个点；

4）确定前K个点所在类别的出现频率；

5）返回前K个点中出现频率最高的类别作为测试数据的预测分类。

### 3.2 模型相关参数介绍

1. 迭代次数：同上文，同一个模型进行多次训练，结果取平均值，目的是减小误差。

2. 邻居个数：超参数，指基于每个查询点的最近邻实现选取个数，其值高度依赖数据。通常较大的值可以抑制噪声的影响，但会使得分类界限不明显。

3. 半径：暴力求解法的可选邻居距离上限，用来筛选近邻样本。

4. 叶子节点数：计算树算法的叶子节点数目，作为参数传递给球树与KD树，影响训练的的资源消耗与训练效果，常见的妥协值为30。

5. 数据点间距离度量法：用来计算数据点间的距离，默认使用欧式距离，但也可以使用KD树法与球树法。方法的选择取决于样本数量、维度、数据结构、近邻数、查询点数。

### 3.3 相关指标介绍

训练停止条件

迭代完成且呈现收敛趋势

(迭代法的收敛性：x(0)是人为地给定的初始迭代向量，G和d分别为n×n矩阵和n维实向量，且满足QA=I-G，Qb=d，此处Q是某一非奇异的n阶方阵.若存在x\*∈Rn使对任意的初值x(0)，由表达式产生的序列{x(k)}都收敛到x\*，则称表达式是收敛的)

训练时间：训练花费时间。

迭代次数：迭代运算过程中循环的次数。

识别率：被正确识别的输入模式的数量占被识别的所有输入模式的总数的百分比。是衡量模式识别系统性能的重要指标，其值越高则识别率越高，系统性能越好。

训练集：确定模型后，用于训练参数，训练的是普通参数（每多加入一个数据对模型进行训练，模型中就会受到影响的参数，通过多次迭代不断更新，是一个梯度下降的过程）而不是超参数（超参数是指训练开始之前设置的参数，超参数的选择与训练过程实际上是独立的，训练过程不会影响超参数。但是训练结束后可以根据训练结果考虑超参数是否可优化，可优化的话就调整超参数的值开始下一次训练）。

测试集：虽然验证集没有对模型的参数产生影响，但是我们却根据验证集的测试结果的准确度来调整参数，所以验证集对结果还是有影响的，即使得模型在验证集上达到最优。就是在很多个模型中，验证集选择了代价函数最小的一个模型。虽然在这个模型上代价很小，但并不代表在其他数据上代价也小。所以需要一个完全没有经过训练的测试集来再最后测试模型的准确率。

### 3.4 实验结果介绍

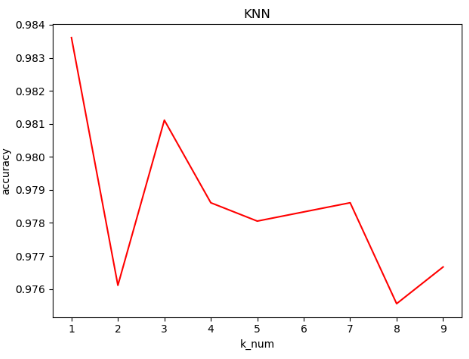
这里使用控制变量法：

控制迭代次数number=1：

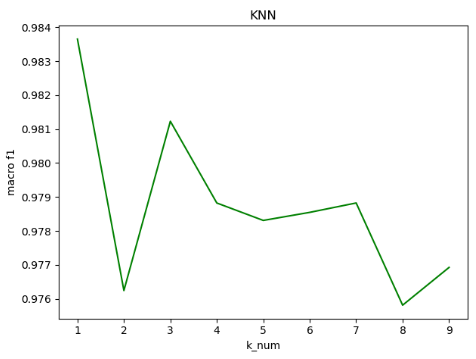
#### 3.4.1 邻居个数变化

变量是邻居个数k\_num(range:1->9,step:1)

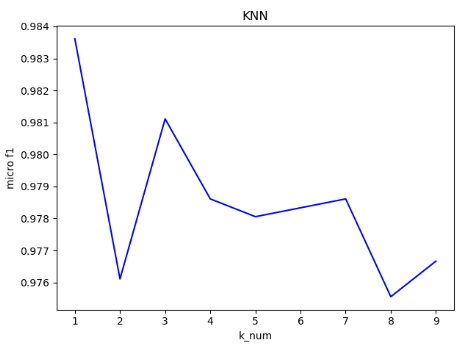
|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Macro f1 | Micro f1 |
| 1 | 0.9836111111111111 | 0.9836526557145753 | 0.9836111111111111 |
| 2 | 0.9761111111111112 | 0.9762436019980573 | 0.9761111111111112 |
| 3 | 0.9811111111111112 | 0.9812310598084594 | 0.9811111111111112 |
| 4 | 0.9786111111111111 | 0.9788235617504288 | 0.9786111111111111 |
| 5 | 0.9780555555555556 | 0.978309935852902 | 0.9780555555555556 |
| 6 | 0.9783333333333334 | 0.9785472937125508 | 0.9783333333333334 |
| 7 | 0.9786111111111111 | 0.9788243872282233 | 0.9786111111111112 |
| 8 | 0.9755555555555555 | 0.9758129289967621 | 0.9755555555555555 |
| 9 | 0.9766666666666667 | 0.9769256358759116 | 0.9766666666666667 |



准确率随邻居个数的变化图像



Macro f1随邻居个数的变化图像

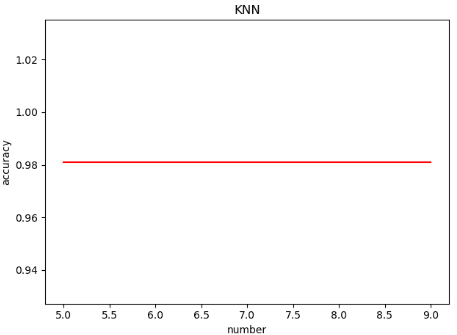


Micro f1随邻居个数的变化图像

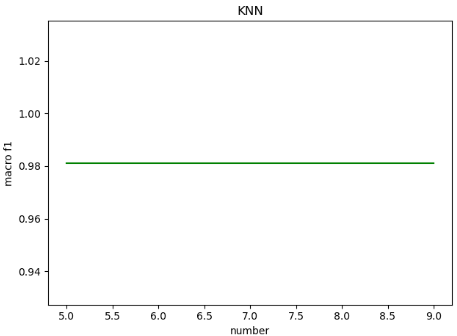
#### 3.4.2 迭代次数

变量是迭代次数number(range:5->9,step:1)

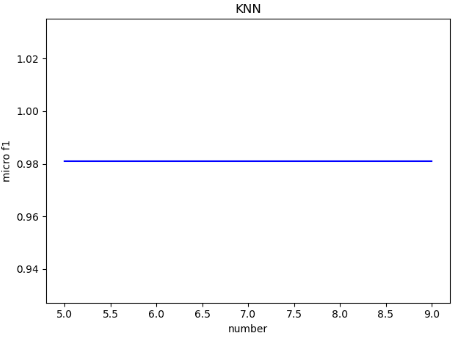
|  |  |  |  |
| --- | --- | --- | --- |
|  | Accuracy | Macro f1 | Micro f1 |
| 5 | 0.9811111111111112 | 0.9812310598084594 | 0.9811111111111112 |
| 6 | 0.9811111111111112 | 0.9812310598084594 | 0.9811111111111112 |
| 7 | 0.981111111111111 | 0.9812310598084595 | 0.981111111111111 |
| 8 | 0.981111111111111 | 0.9812310598084595 | 0.981111111111111 |
| 9 | 0.9811111111111112 | 0.9812310598084595 | 0.9811111111111112 |



准确率随迭代次数的变化图像



Macro f1随迭代次数的变化图像



Micro f1随迭代次数的变化图像

通过图像和表中数据可以明显看出5轮迭代之后，模型的效果已经呈现收敛趋势，指标基本一致，可见对于k\_num=3的情况我们尝试了过多的迭代，浪费时间。接下来数据可以一定程度减少number，减少过拟合可能和所花时间。

### 3.5 实验结论

通过进行KNN模型的建立、训练和结果分析，我们发现，使用 KNN 模型进行分类处理，模型的训练数据集精度最高可以达到100%，比ANN 高，模型的测试数据及精度最高可以到达98.3%，也比ANN的模型高，说明使用KNN模型在分类处理问题中比ANN有更好的分类效果，但是也需要注意对于权重函数和查找的邻居数的选择。