



**课 程 报 告**

|  |  |
| --- | --- |
| 课程名称 | 模式识别 |
| 题目名称 | 性别识别 |
| 专业班级 | 自动化卓越班 |
| 学号姓名 | 陈汉杰3119000837 |
|  | 郑宇凡 3119000874 |
|  | 刘敏伟3119000855 |
|  | 陈冠希3119000836 |
|  | 谢锋3119000867 |
| 指导教师 | 邢延 |

2021年7月 2日

目录

[1 模式识别系统设计 1](#_Toc106369016)

[2 全卷积神经网络（FCN）算法 2](#_Toc106369017)

[2.1 卷积神经网络基本原理 3](#_Toc106369018)

[*2.1.1CNN* 3](#_Toc106369019)

[*2.1.2FCN* 5](#_Toc106369020)

[*2.1.3深度残差网络* 6](#_Toc106369021)

[2.3 神经网络模型 7](#_Toc106369022)

[*2.3.1原始模型* 7](#_Toc106369023)

[*2.3.2修改模型* 10](#_Toc106369024)

[*2.3.3*训练策略 14](#_Toc106369025)

[*2.4训练结果* 14](#_Toc106369026)

[3 KNN最邻近分类算法 14](#_Toc106369027)

[3.1 KNN算法原理 14](#_Toc106369028)

[3.2 KNN算法流程 17](#_Toc106369029)

[3.3 KNN改进算法 17](#_Toc106369030)

[4 SVM支持向量机算法 18](#_Toc106369031)

[4.1 SVM算法原理 18](#_Toc106369032)

[4.2 RBF核函数参数最优化选择 20](#_Toc106369033)

[5决策树算法 22](#_Toc106369034)

[5.1决策树算法流程 22](#_Toc106369035)

[5.2决策树算法流程 25](#_Toc106369036)

[6 BP神经网络 25](#_Toc106369037)

[结论 29](#_Toc106369038)

[参考文献 29](#_Toc106369039)

[附录 30](#_Toc106369040)

# 1 模式识别系统设计

内容要求：

1. 说明识别或者分类的目标，例如：基于人脸图像的性别（男、女）识别、基于短文本的情感（正面、负面）分类等

用有监督学习机制设计并实现模式识别方法，对性别（男性、女性）基于人脸图像进行模式分类

1. 数据来源及数据特点（数据量、特征维度、类别是否均衡、数据量是否足够等）

数据量：

总数居4000张人脸图片，男女比例大约6：4，如下:



图 1.1数据集样例

特征维度：

128\*128

类别均衡：

由于样本分配稍微有点不均匀，我们采取重采样方式，增加女性的数据，从而使得样本比例基本为1：1； 经过样本的重采样我们最终得到4860个数据。但是存在数据标签缺失的情况，我们实际可使用数据为4500个左右。

4500个数据中，存在相同的人的不同表情等的人脸数据，而且数据相似度较高，于是我们采用了数据增强的方式来提高训练效果。 在实践中发现在数据增强前，数据容易存在过拟合现象，进行了数据增强后，过拟合现象明显减缓。

数据集处理：

训练集：验证集：测试集 = 8：1：1

训练集和验证集进行k=10的k折交叉验证

1. 采用的开发工具（编程语言必须用Python、集成开发环境等（Jupyter notebook, 或者Pycharm））

使用的集成开发环境为Pycharm。

1. 采用的模式识别方法（注意要与数据特点相匹配）
2. 特征提取算法（经典、改进）（必做）
3. 特征选择算法（经典、改进）（此项可根据需要和兴趣选做）
4. 采用的分类算法（含经典算法、集成算法/深度学习算法/改进算法）
5. 分类器性能评价方法（K折交叉验证，分类准确率/错分率/AUC,是否考虑算法的时间复杂度等）

# 2 全卷积神经网络（FCN）算法

内容要求：

1. 经典算法的原理、参数设定与调整、实验结果的分析比较、小结。
2. 集成算法/深度学习算法/改进算法的原理、参数设定与调整、实验结果的分析比较（必须要有算法之间的结果对比）、小结。

## 2.1 卷积神经网络基本原理

### *2.1.1CNN*

普通的神经网络模型是由多层数量不一的神经元组成的一种分层结构，并且每两层之间的神经元都互相连接，如图2.1.1.1所示。这个神经网络模型只包含一个隐含层，其中每个神经元输出的计算公式可表示为:

式中x是神经元的输出，b为偏置，WT是每个神经元与上一层连接的参数向量，该向量长度和上一层神经元的数目相同，在上图中前两层网络都只有3个神经元，那么这两层之间的参数只有3\*3=9个。但是在实际中一个输入图像不可能只有三个像素，以一张200\*200的图像为例，对应的输入层就相当于有40000个神经元;如果隐含层与输入层神经元的数目相等，那么它们之间的参数数量为

40000\*40000=16亿，这样太过庞大的参数数目，基本无法进行神经网络的训练。

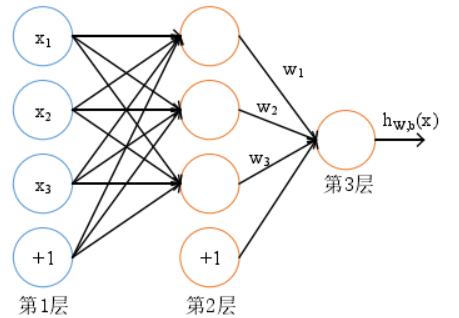


图2.1.1.1仅有三层的BP神经网络模型

为了减少参数数量、加快训练速度，卷积神经网络主要使用了两种方法:局部感知和参数共享。局部感知就是将神经元只与前一层的一小块区域连接，而不是与所有的神经元都进行计算，如图2.1.1.2所示。其中，左图为全连接，右图为局部连接示意图。

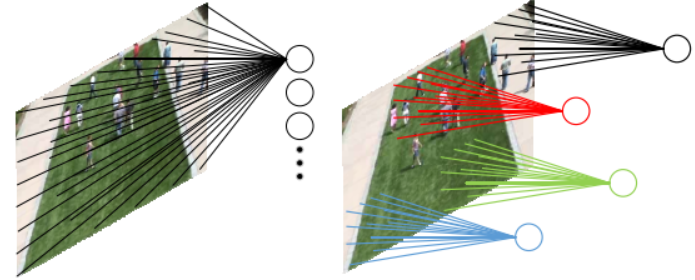


图2.1.1.2 局部感知原理图

假设在神经网络中，每个神经元只与10\*10的像素区域进行连接计算，那么

同样是输入200\*200的图像，一层的参数数量就是40000\*100，与之前相比减少了400倍。像这样每10\*10个像素值与参数计算实际上就是神经网络中的卷积操作，这些参数就是1010的卷积核。局部感知的策略，大大减少了神经网络中需要训练的参数数量，但是其规模仍旧十分庞大，所以为了更大程度的优化训练速度，卷积神经网络又采取了参数共享的方法，即让不同神经元的参数相同。还是以上述局部感知的神经网络为例，一共40000个神经元，每个神经元都对应不同的100个参数，这时参数的数量为40000\*100个。卷积神经网络为了极大程度的减少参数数量，将每个神经元的参数进行共享，使参数的数量只有100个。这100个参数被称为一个10\*10的卷积核，每一种卷积核提取输入图像的一种特征，如果有上百种卷积核，就能学习到图像的上百种特征。再加上卷积神经网络的深层结构，这些卷积特征将包含图像丰富的局部细节信息以及更全局的轮廓形状等信息。因此，卷积神经网络能充分的提取图像特征，在图像处理上有极大的优势。

通过多层卷积得到特征之后，是利用这些特征进行分类，也就是基于所有提取到的卷积特征训练分类器。但是这样做的计算量相当大，例如，我们用300种11\*11的卷积核提取一张200\*200的图像的特征，每一种卷积核和图像进行计算将得到一个(200-11+1)\*(200-11+1)= 36100维的特征。那么有300种特征，就是说一张图像通过卷积层会生成36100\*300=10830000维特征，如果同时对大量图像进行分类识别，在如此高维度下训练分类器是很不科学的，不仅计算量巨大，而且容易造成过拟合(over-fitting)的现象。因此，为了解决这一问题，卷积神经网络通常在卷积层之后加入池化层(pooling layer)。池化操作是对特征进行聚合统计，以降低特征维度和提高泛化能力，简单来说就是用一个特征区域中的平均值或最大值来表示该区域，其示意图如图2.1.1.3所示。

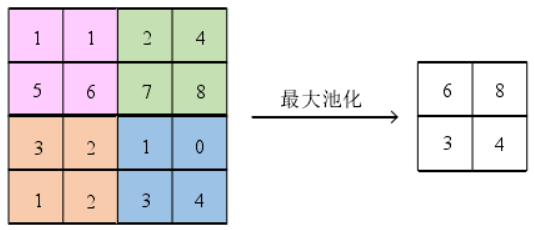


图2.1.1.3 最大池化原理图

池化的计算方法有很多种，常用的有最大池化、平均池化和最小池化等。

### *2.1.2FCN*

因为CNN存在的缺陷，人们便提出全卷积神经网络FCN的概念，FCN顾名思义是该网络中全是卷积层链接，它实际上是由卷积神经网络变换而来，CNN和FCN之间的最大区别就是FCN将卷积神经网络的全连接层都转换为了卷积层，如图2.1.2.1所示。

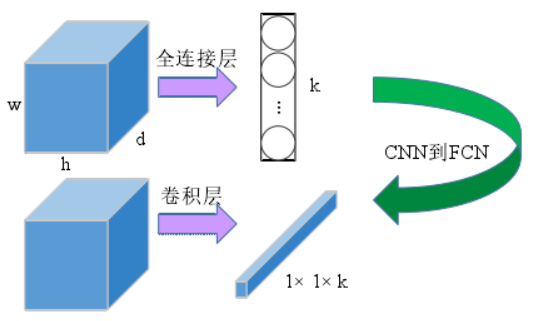


图2.1.2.1 全连接层转换为卷积层的示意图

卷积层与全连接层的不同之处在于，卷积层有卷积核，与上一层的输入数据局部感知，而全连接层是将输入数据编码为一个特征向量，但是它们的函数形式是一样的，所以它们两者能相互转化。假设输入图像经过几层卷积层后，得到大小为w\*h\*d的数据体，其中w, h分别是数据的宽度和高度，d则是数据的维度或者说通道的数量。在CNN中该数据体通过全连接层输出的是一个长为k的向量，将全连接层转化为卷积层，即使用数量为d\*k，尺寸为w\*h的卷积核与数据体进行卷积，得到l\*l\*k的新卷积层的输出。

### *2.1.3深度残差网络*

随着网络深度的加深，误差不断积累在反向传播过程中，最初几层的梯度值可能几乎为0,使网络无法收敛，这种现象称为“梯度弥散”由于“梯度弥散”现象的存在，20层以上的深度网络随着层数的增加，效果不升反降，网络出现退化，即当网络变得越来越深时，训练的准确率趋于平缓，但训练误差变大

为解决该问题，文献[20]提出了ResNet，其原理如下：由于神经网络的本质是将某空间维度的向量a经过非线性变换H(a)映射到另一空间维度，网络训练过程就是优化H(a)；由于H(a)难以优化，文献[20]提出不再用多个堆叠的层直接拟合期望的特征映射，而是用它们拟合一个残差映射，引入更容易学习的H(a)的残差形式

假设最优化残差映射比最优化期望的映射更容易，即比更容易优化，则在极端情况下，期望的特征映射是恒等映射，此时，残差网络的任务是拟合,而普通网络要拟合的是,显然前者更容易优化深度残差网络由多个残差单(residual unit)组成，在每个残差单元中输入直接连接到输出，类似一个“高速通道”，两部分相加形成最后的输出，使输出层的残差传递得更远，在训练更深层的网络时网络收敛速度更快。

ResNet 借鉴了高速网络的思想，在网络中加入 shortcut 连接形成残差单元. 每层输出的不仅是输入的映射，还是映射和输入的叠加. 残差单元的引入增加了当前层以前的先验特征， 加强了网络中信息的传递. 因此，网络训练阶段模型收敛速度更快，训练所得模型整体效果更优.

迁移残差思想改造基于FCNs（FCN8，FCN16，FCN32）的网络结构， 将部分卷积层改造成残差单元的结构形成残差块. 图 2 细化展示了残差优化的 FCNs。

从图 2.1.3.1可以看出，FCN32、FCN16、FCN8 都包含1个卷积块、4个残差块和1个反卷积块. 在3种网络中，残差块的结构完全相同，即每个残差块包含3个卷积层，前2个卷积层后紧跟一个激活层。为了加强网络中特征的传递，在第 1 个卷积层后引入 shortcut 连接，将其激活后的输出与第 2 个卷积层的输出相加，再进行激活。

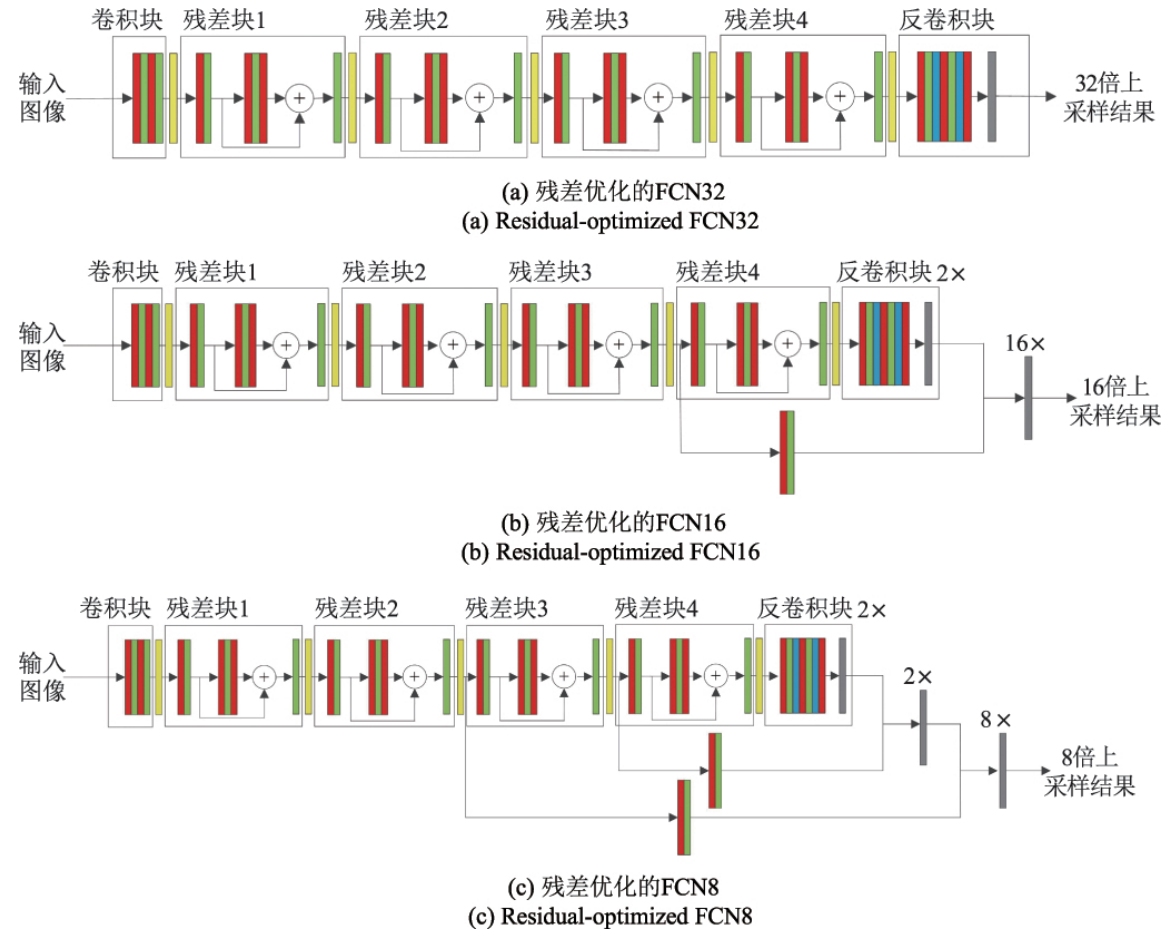


图 2.1.3.1 残差优化的FCNs（FCN8，FCN16，FCN32）网络结构图（红色为卷积层，绿色为激活层，黄色为池化层，蓝色为 50% dropout 层，灰色为反卷积层）

## 2.3 神经网络模型

### *2.3.1原始模型*

我们开始想法是通过深度较深的卷积模型来识别图片，为了防止深度神经网络（DNN）隐藏层过多时的网络退化问题，我们使用了残差网络。 并且使用输出层使用(1\*1)的卷积核来实现降维，从而代替全连接层来减少模型的参数。

综上所述，我们使用了全卷积深度残差网络，模型输入为（n\*128\*128\*1），输出为（n\*2）。

具体模型结果如下：

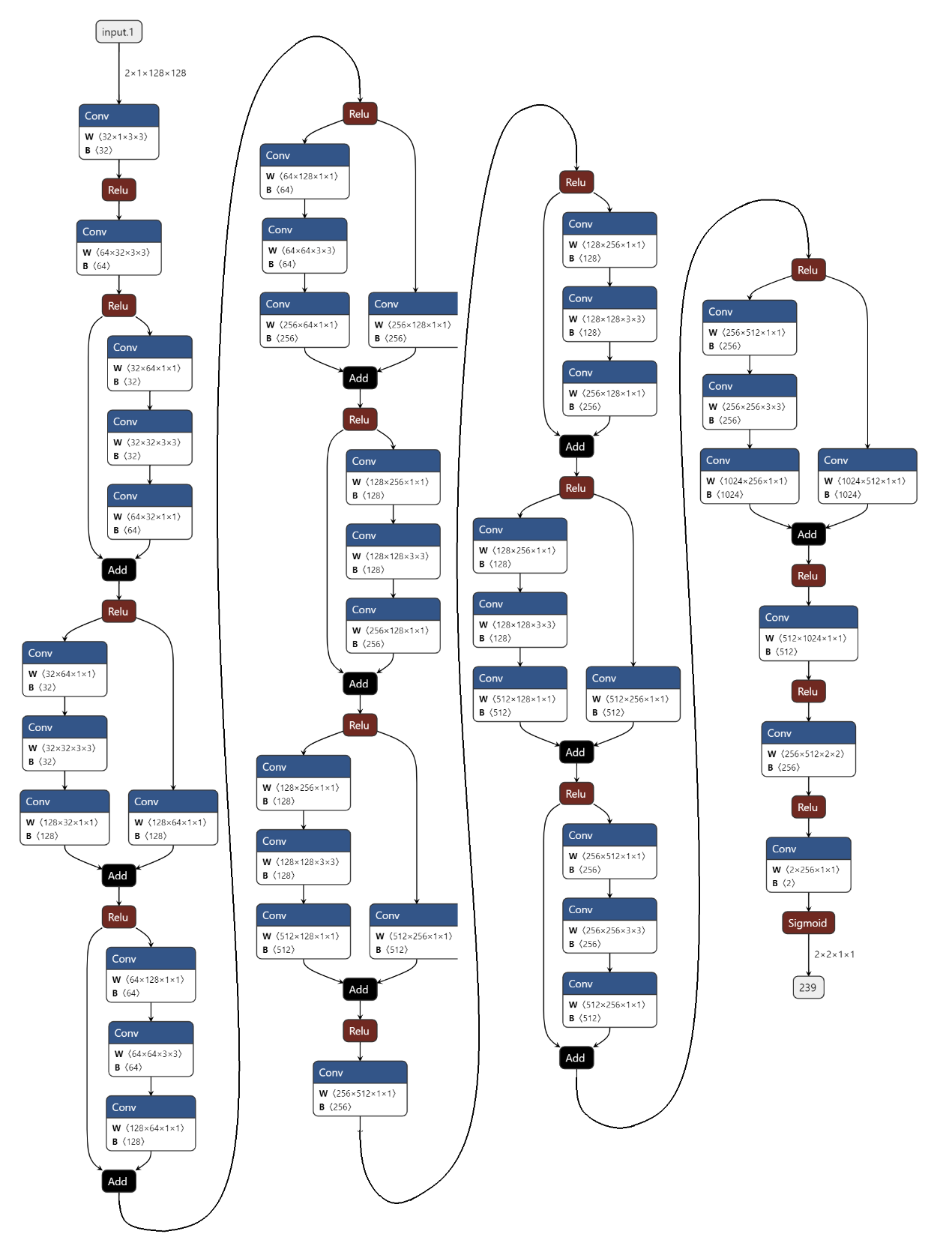


图 2.3.1.1 Model的模型示意图

### *2.3.2修改模型*

为了增加模型的鲁棒性，我们将原始模型的残差通道拓展至三条。

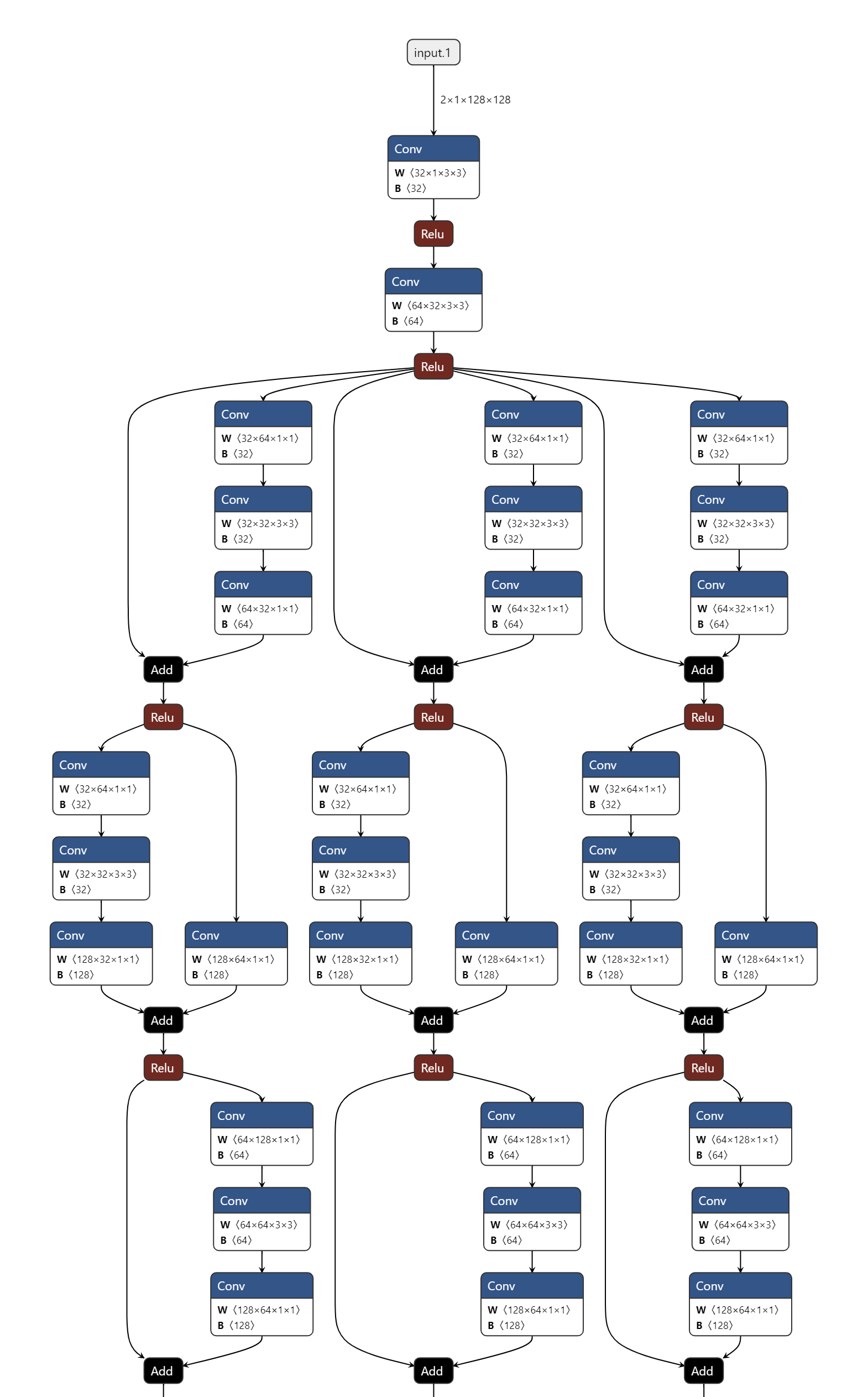


图 2.3.1.2 Model\_wise的模型示意图1

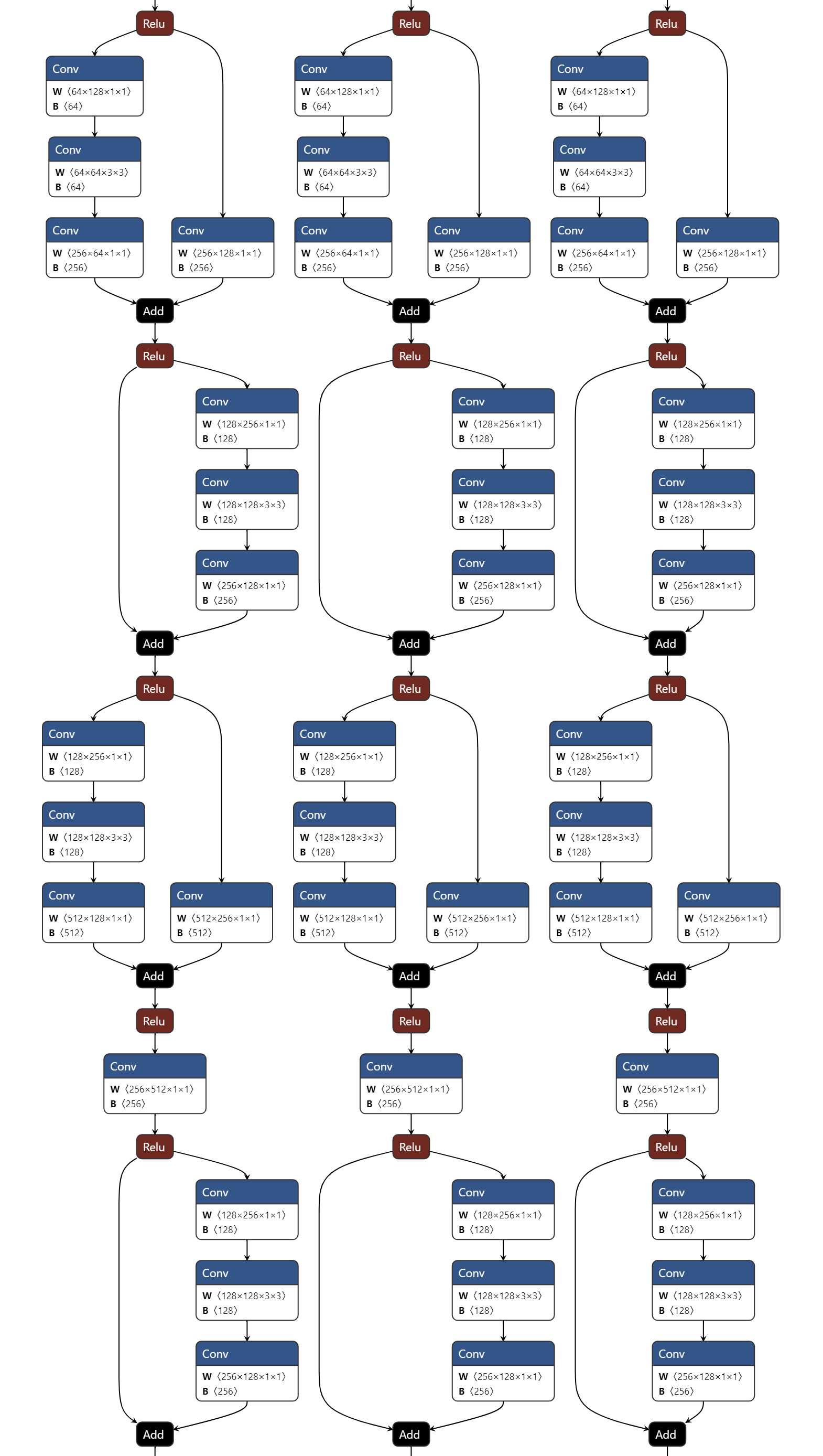


图 2.3.1.3 Model\_wise的模型示意图2

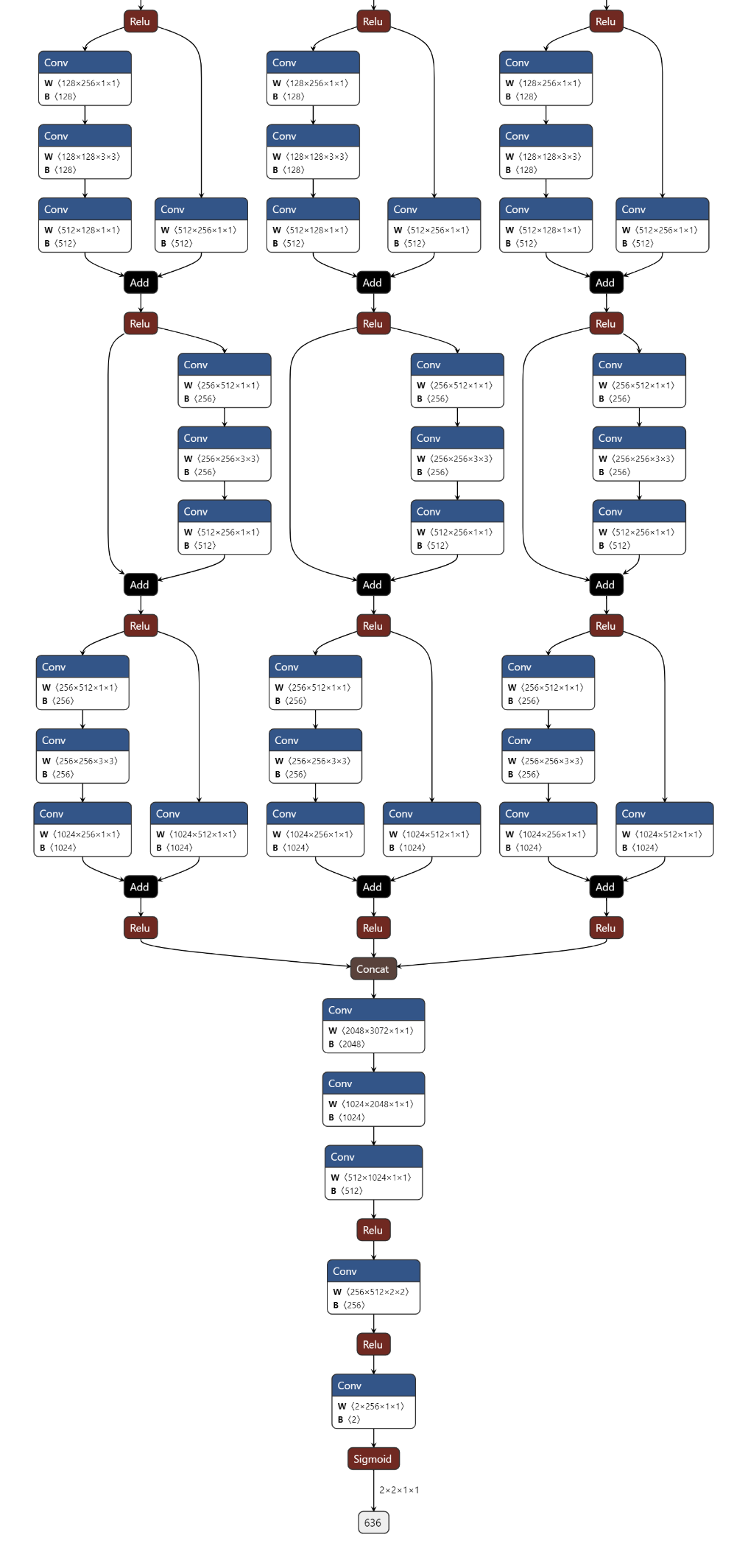


图 2.3.1.4 Model的模型示意图3

## *2.3.3*训练策略

1.损失函数选择：

本任务为二分类任务，故选择交叉熵损失函数。

2.优化器选择：

在比较了adam和sgd后，发现使用adam优化时，模型经常难以收敛，而使用sgd收敛比较稳定，不会存在模型不收敛的情况。

## *2.4训练结果*

为了显示差别，我们分别对每个模型使用原始数据和增强数据来训练1000epochs，得到下面的结果；

## *2.4.1Model wise*

在数据增强且训练了1000epochs后，得到下面两图，可知并没有发生过拟合，模型对增强数据还不能完全拟合，但是由于采用了交叉验证，val的数据是没有被增强的，所以准确率很高。

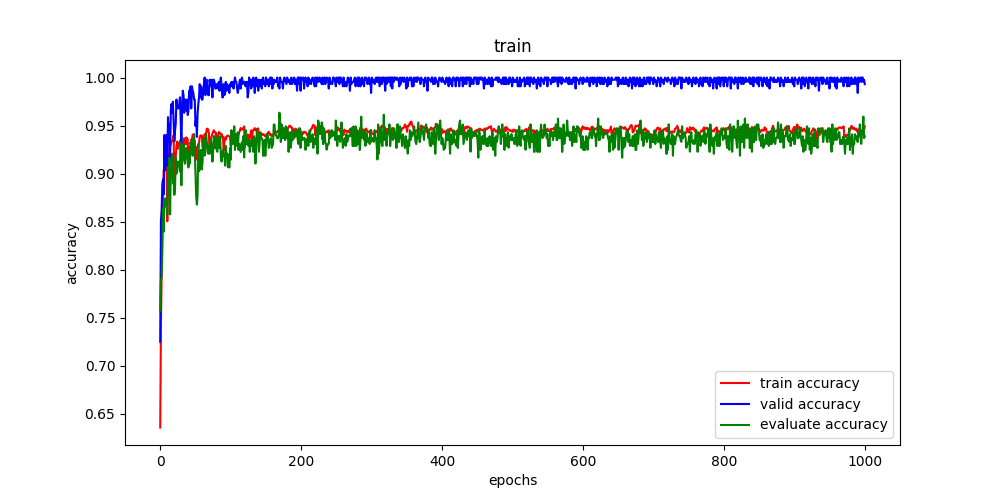


图 2.4.1.1 Model\_wise训练过程精确率可视化

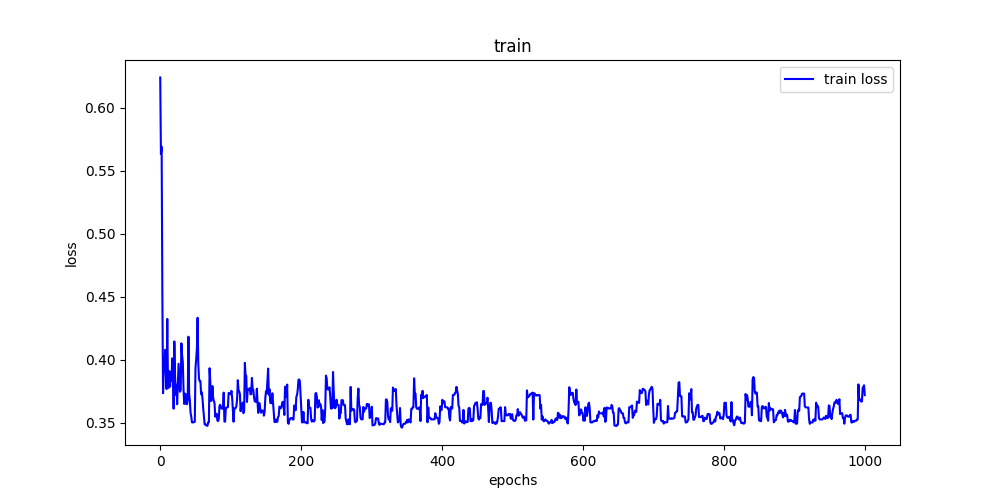


图 2.4.1.1 Model\_wise训练过程loss可视化

# 3 KNN最邻近分类算法

## 3.1 KNN算法原理

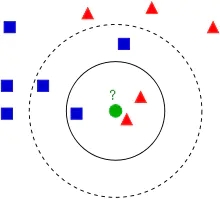
K最近邻(K-Nearest Neighbor,KNN)算法，是著名的模式识别统计学方法，在机器学习分类算法中占有相当大的地位。它是一个理论上比较成熟的方法。既是最简单的机器学习算法之一，也是基于实例的学习方法中最基本的，又是最好的文本分类算法之一。

如果一个实例在特征空间中的K个最相似（即特征空间中最近邻）的实例中的大多数属于某一个类别，则该实例也属于这个类别。所选择的邻居都是已经正确分类的实例。

该算法假定所有的实例对应于N维欧式空间Ân中的点。通过计算一个点与其他所有点之间的距离，取出与该点最近的K个点，然后统计这K个点里面所属分类比例最大的，则这个点属于该分类。

由其思想可以看出，KNN是通过测量不同特征值之间的距离进行分类，而且在决策样本类别时，只参考样本周围k个“邻居”样本的所属类别。因此比较适合处理样本集存在较多重叠的场景，主要用于聚类分析、预测分析、文本分类、降维等，也常被认为是简单数据挖掘算法的分类技术之一。

举个🌰，看下面这幅图：



KNN的算法过程是是这样的： 从上图中我们可以看到，图中的数据集是良好的数据，即都打好了label，一类是蓝色的正方形，一类是红色的三角形，那个绿色的圆形是我们待分类的数据。 如果K=3，那么离绿色点最近的有2个红色三角形和1个蓝色的正方形，这3个点投票，于是绿色的这个待分类点属于红色的三角形 如果K=5，那么离绿色点最近的有2个红色三角形和3个蓝色的正方形，这5个点投票，于是绿色的这个待分类点属于蓝色的正方形 我们可以看到，KNN本质是基于一种数据统计的方法！其实很多机器学习算法也是基于数据统计的。 KNN是一种memory-based learning，也叫instance-based learning，属于lazy learning。即它没有明显的前期训练过程，而是程序开始运行时，把数据集加载到内存后，不需要进行训练，就可以开始分类了。 具体是每次来一个未知的样本点，就在附近找K个最近的点进行投票。

KNN算法的关键是要比较待分类数据与样本数据之间的距离，机器学习中通常的做法是：提取数据的特征值，根据特征值组成一个n维实数向量空间（这个空间也被称作特征空间），然后计算向量之间的空间距离。空间之间的距离计算方法有很多种，常用的有欧氏距离、余弦距离等。

对于数据 和 ，若其特征空间为n维实数向量空间 ，即 ，

3.1.1 欧几里得距离:

  n维实数向量空间，欧氏距离计算公式为：

3.1.2 余弦值：

几何中，夹角余弦可用来衡量两个向量方向的差异；机器学习中，借用这一概念来衡量样本向量之间的差异。

  n维实数向量空间，余弦值为：

余弦相似度的值越接近1表示其越相似，越接近0表示其差异越大，使用余弦相似度可以消除数据的某些冗余信息，某些情况下更贴近数据的本质。

3.1.3相关系数

相关系数：是衡量随机变量X与Y相关程度的一种方法，相关系数的取值范围是[-1,1]。相关系数的绝对值越大，则表明X与Y相关度越高。当X与Y线性相关时，相关系数取值为1（正线性相关）或-1（负线性相关）：

相关距离：

3.1.4 曼哈顿距离

n维空间点，曼哈顿距离：

## 3.2 KNN算法流程

下面对 KNN 算法的流程做简单介绍。KNN 分类算法主要包括以下 4 个步骤：

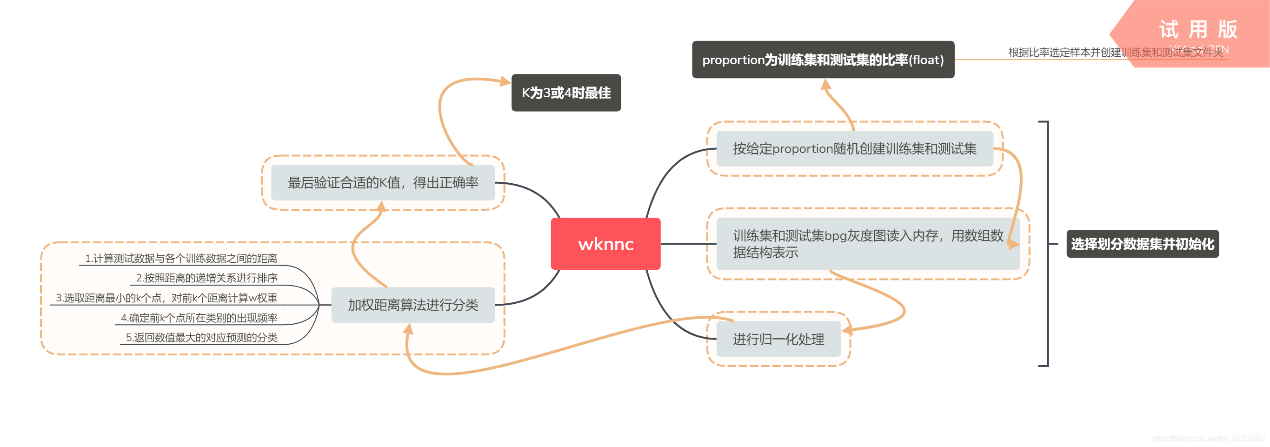
1. 准备数据，对数据进行预处理 。
2. 计算测试样本点（也就是待分类点）到其他每个样本点的距离（选定度量距离的方法）。
3. 对每个距离进行排序，然后选择出距离最小的 K 个点。
4. 对 K 个点所属的类别进行比较，按照少数服从多数的原则（多数表决思想），将测试样本点归入到 K 个点中占比最高的一类中。

## 3.3 KNN改进算法

加权欧氏距离KNN。

通过结合KNN本身的分类算法以及对前k个距离加权，来达到分类的目的 wk-nnc算法是对经典knn算法的改进，这种方法是对k个近邻的样本按照他们距离待分类样本的远近给一个权值w

w(i)是第i个近邻的权值，其中1<i<k，h(i)是待测样本距离第i个近邻的距离。



# 4 SVM支持向量机算法

## 4.1 SVM算法原理

支持向量机(SVM)理论[1]主要是针对二类模式识别问题提出的。对于二类模式识别问题,设给定的训练集为 ,其中 为输入向量,输出向量为 ,如果该训练集可被一个超平面线性划分,则该超平面为 ,其中 和 是决定了超平面的位置, 为两个向量的内积。为了得到最优化的划分, 则该问题就转化为求最优化超平面的问题。

其中： 是特征空间中分类超平面的系数向量； 是分类面的阈值； 是考虑分类误差而引入的松弛因子； 是对于错分样本的惩罚因子。这样的话所构造出来的最优化超平面为：。式优化问题可以转化为其对偶问题。

对于大多数样本来讲, =0,对应≠0的样本称为支持向量(Support Vector ,SV)。解出式的最优化函数为

其中

上式求和实际上由支持向量，即 ≠0的样本决定. 从这一点可以得出支持向量决定超平面(图4.1.1所示)的划分.

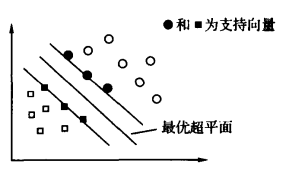


图4.1.1 最优化超平面示意图

对于非线性可分的情况,可以通过一个映射函数 (在SVM称核函数)，将低维的输入空间 映射到高维的特征空间 H 使线性可分(图4.1.2所示)，则低维的线性不可分问题就变成高维空间的线性可分问题。

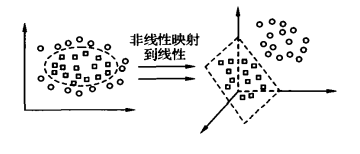


图4.1.2 核函数原理图

这样问题就可以表述为：输入向量 X 通过映射 影射到高维空间 H 中,则核函数 ,则优化问题转化为

解出式的最优化函数为：

从式的最小化问题可以看出，不需要知道H 和,只需要选择合适的核函数 K(·)和 C 就可以确定SV M。现在，应用较常见的核函数有以下四种：

线性核：

多项式核：

径向基(RBF)核：

Sigmoid 核：

在这四种核函数中,应用最广泛的是 RBF 核, 无论是低维、高维、小样本、大样本等情况，RBF核函数均适用，具有较宽的收敛域，是较为理想的分类依据函数[2] 。

RBF 核为：则对应式的最优化问题就转化就下面最小化问题

其中

这样求式的最小值就取决于参数的选择。这样选择最佳的参数就可以使SVM分类器性能最好,即推广能力最强.

## 4.2 RBF核函数参数最优化选择

对于一个基RB核函数SVM ,其性能是由参数决定,选取不同的 C 和 γ就会得到不同的SVM .我们的目的是为了寻找最佳的参数组合使该 SVM 的性能最好,即推广错误率最低.

最简单的方法是分别选取不同的参数组合，得出不同的错误率；分别比较这些错误率选取其中错误率最小的参数组合作为最优化选择，这种方法也叫做“穷举法”。参数 C 和 γ分别取 N 个值和 M 个值,对 个 的组合分别训练不同的SV M ,再估计其推广错误率，从而在 N × M 个组合中得到错误率最低的一个组合作为最优参数。如图4.2.1所示参数为最优化参数,此时的错误率最低。虽然用这种方法最终能找出最优化参数，但是其复杂度为 ,显然运算量非常大，花费时间很大,特别对大样本数据来讲是不切实际的。

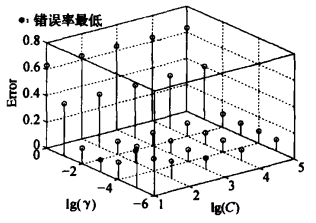


图4.2.1 “穷举法”得到不同参数不同的错误率

从以上图4.2.1分析知道,对于在一定区域内的组合得到错误率都非常低,即SVM推广识别能力都非常高。对这个区域叫“好区” ,如图4.2.2(a)所示。为了确定“好区”内最优化参数组合，该文的思想如图4.2.2 (b)所示，通过对曲线 上的点 来估计最优化参数，则用该曲线得到的最优化参数为作 为 SVM的最优化参数组合,其中是常数,用该思想得到复杂度为 .而且，且已知当C取得一定值时，使 SVM最优化, 此时用线性 SVM 来得到最优化参数,把该参数作为常数，以此来确定曲线 ，使错误率最低的参数组合集中出现在“好区”中该曲线的附近。

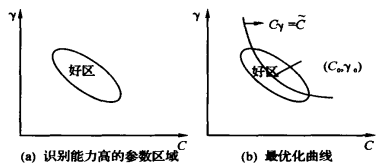


图4.2.2 识别能力高的参数区域内的最优化曲线

基于该思想得出本文优化参数的算法步骤：

(1)用线性SVM求解最优化参数 ,使之该参数的 SVM推广识别错误率最低；

(2)对RBF的SVM ,固定 ,取满足的 ,训练SVM ,根据对推广识别错误率的估计, 取错误率最低的参数，把该参数作为SVM的最优化参数。

# 5决策树算法

## 5.1决策树算法流程

数据集

所有样本为同一类别？

选择类别

生成叶节点

提取特征集A

A为空？

选择数据集中占比最多的类

所有特征取值唯一？

选择最优划分特征a生成结点(算法:ID3,C4.5,CART...)

特征值遍历结束？

生成数据子集

剔除特征a

数据子集

上图是决策树生成算法流程图

以下介绍CART（classification and regression tree），CART的分类效果一般要优于其他决策树。

**CART算法**

**基本介绍**

CART算法是一种决策树算法，主要用于分类与回归。CART算法将概率论与统计学的知识引入到决策树的研究中，既可以用于分类，也可以用于回归。不同于C4.5算法，CART算法的本质是对特征空间进行二元划分（CART生成的决策树是一颗二叉树），并能够对标量属性与连续属性进行分裂。CART算法采用地即是递归地对每个特征进行二元切分，然后根据输入的特征值预测输入样本的结果。

优点

[1] 计算简单，易于理解，可解释性强；

[2] 比较适合处理有缺失属性的样本；

[3] 不仅能够处理不相关的特征，还能在相对短的时间内对大型数据源得出可行且效果良好的结果。

缺点

[1] 不支持在线学习，在有新的样本产生后，决策树模型要重建；

[2] 容易出现过拟合的现象，生成的决策树可能对训练数据有很好的分类能力，但对未知的测试数据却未必有很好的分类能力。

**特征选择**

特征选择也即选择最优划分属性，从当前数据的特征中选择一个特征作为当前节点的划分标准。 随着划分过程不断进行，希望决策树的分支节点所包含的样本尽可能属于同一类别，即节点的“纯度”越来越高。

**熵（entropy）**

熵表示事务不确定性的程度，也就是信息量的大小（一般说信息量大，就是指这个时候背后的不确定因素太多），熵的公式如下：

其中，是分类出现的概率，n是分类的数目。可以看出，熵的大小只和变量的概率分布有关。

对于在X的条件下Y的条件熵，是指在X的信息之后，Y这个变量的信息量（不确定性）的大小，计算公式如下

当只有A类或只有B类时，

所以当Entropy最大为1的时候，是分类效果最差的状态，当它最小为0的时候，是完全分类的状态。因为熵等于零是理想状态，一般实际情况下，熵介于0和1之间 。

熵的不断最小化，实际上就是提高分类正确率的过程。

**信息增益（information gain）**

信息增益：在划分数据集之前之后信息发生的变化，计算每个特征值划分数据集获得的信息增益，获得信息增益最高的特征就是最好的选择。

定义属性A对数据集D的信息增益为，它等于D本身的熵，减去 给定A的条件下D的条件熵，即：

其中 ,K个值

信息增益的意义：引入属性A后，原来数据集D的不确定性减少了多少。

计算每个属性引入后的信息增益，选择给D带来的信息增益最大的属性，即为最优划分属性。一般，信息增益越大，则意味着使用属性A来进行划分所得到的的“纯度提升”越大。

**步骤**

1从根节点开始，计算所有可能的特征的信息增益，选择信息增益最大的特征作为节点的划分特征；

2由该特征的不同取值建立子节点；

3再对子节点递归1-2步，构建决策树；

4直到没有特征可以选择或类别完全相同为止，得到最终的决策树。

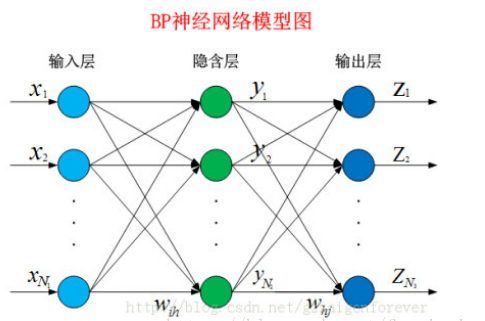
# 6 BP神经网络

**基本原理**

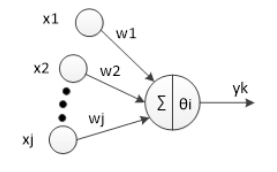
人工神经网络无需事先确定输入输出之间映射关系的数学方程，仅通过自身的训练，学习某种规则，在给定输入值时得到最接近期望输出值的结果。作为一种智能信息处理系统，人工神经网络实现其功能的核心是算法。BP神经网络是一种按误差反向传播(简称误差反传)训练的多层前馈网络，其算法称为BP算法。

BP算法会每次根据训练得到的结果与预想结果进行误差分析，进而修改权值和阈值，一步一步得到能输出和预想结果一致的模型。

BP网络由输入层、隐藏层、输出层组成。给定训练集D={(x1,y1),(x2,y2…(xn,yn)},其中,表示输入示例由d个属性组成，输出l维实值变量。



神经元是以生物研究及大脑的响应机制而建立的拓扑结构网络，模拟神经冲突的过程，多个树突的末端接受外部信号，并传输给神经元处理融合，最后通过轴突将神经传给其它神经元或者效应器。神经元的拓扑结构如图：



对于第i个神经元，X1、X2、…、Xj为神经元的输入，输入常为对系统模型关键影响的自变量，W1、W2、…、Wj为连接权值调节各个输入量的占重比。将信号结合输入到神经元有多种方式，选取最便捷的线性加权求和可得neti神经元净输入:

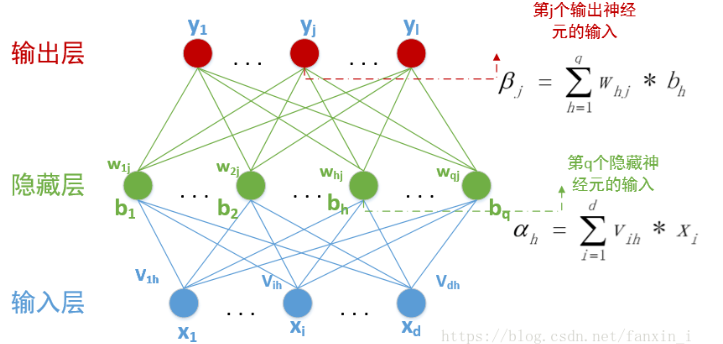
表示该神经元的阈值，根据生物学中的知识，只有当神经元接收到的信息达到阈值是才会被激活。因此，我们将和进行比较，然后通过激活函数处理以产生神经元的输出。

激活函数：激活函数这里我们不多重述。如果输出值有一定的范围约束，比如用来分类，一般我们用的最多的是Sigmod函数，它可以把输入从负无穷大到正无穷大的信号变换成0到1之间输出。如果没有约束的话，我们可以使用线性激活函数(即权值相乘之和)。这样我们得到的输出为：

我们可以将公式化简一下，设第一个输入永远值为θ权值为-1，则我们可以得到公式：

其中=-1,=,其中f为选择的激活函数。

已经知道在BP神经网络模型中，我们有三层结构，输入层、隐藏层、输出层,因此输入层到隐藏层的权值，设为​ ,隐藏层第h个神经元的阈值我们设为。隐藏层到输出层的权值，设为,输出层第j个神经元的阈值我们用表示。在下面这张图里，有d输入神经元,q个隐藏神经元，隐藏有q个隐藏神经元阈值，l个输出神经元，因此有l个输出神经元阈值。



其中中的。隐藏层和输出层的激活函数，在这里我们暂时全部用Sigmod函数。在某个训练示例中，假设神经网络的训练输出为输出为l维向量，其中

优劣势

BP神经网络无论在网络理论还是在性能方面已比较成熟。其突出优点就是具有很强的非线性映射能力和柔性的网络结构。网络的中间层数、各层的神经元个数可根据具体情况任意设定，并且随着结构的差异其性能也有所不同。但是BP神经网络也存在以下的一些主要缺陷。

①学习速度慢，即使是一个简单的问题，一般也需要几百次甚至上千次的学习才能收敛。

②容易陷入局部极小值。

③网络层数、神经元个数的选择没有相应的理论指导。

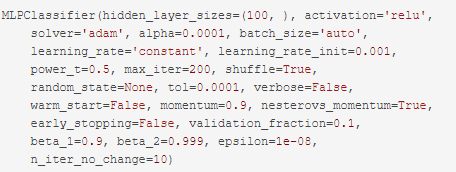
④网络推广能力有限。对于上述问题，已经有了许多改进措施，研究最多的就是如何加速网络的收敛速度和尽量避免陷入局部极小值的问题。

**MLP分类器**

接下来我们着重介绍MLP分类器

MLP分类器是基于神经网络的动态分类器，是一种多层感知器分类器。

sklearn.neural\_network的参数说明



我们着重介绍以下几个参数

hidden\_layer\_sizes元组，长度 = 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。

solver{'lbfgs'， 'sgd'， 'adam'}， default='adam'

用于权重优化的求解器。

“lbfgs”是准牛顿方法系列中的一个优化器。

“sgd”是指随机梯度下降。

“adam”是指由 Kingma、Diederik 和 Jimmy Ba 提出的基于随机梯度的优化器

注意：在训练时间和验证分数方面，默认求解器“adam”在相对较大的数据集（具有数千个或更多训练样本）上效果很好。然而，对于小型数据集，“lbfgs”可以更快地收敛并更好地执行。

alpha：float，默认值 = 0.0001

L2正则化项的强度。L2正则化项除以添加到损失时的样本数量。

random\_stateint:int,RandomState instance，默认值 = None

确定权重和偏差初始化的随机数生成，如果使用早期停止，则确定训练测试拆分，以及当 solver='sgd' 或 'adam' 时进行批量采样。传递一个 int 以获得跨多个函数调用的可重现结果。请参见词汇表。

# 结论

内容要求：

1. 做了什么，取得了什么结论（同种算法之间？不同种算法之间？）
2. 需要改进的方面

# 参考文献

[1] 邓乃扬,田英杰.数据挖掘中的新方法：支持向量机[ M].北 京： 科学出版社,2004.

[2] 李盼池,许 少 华.支 持 向 量 在 模 式 识 别 中 的 核 函 数 特 性 分 析 [ J].计算机工程与设计,2005,26(2)：302-304.

[20] He K, Zhang XY, Ren S Q, et al. Deep residual learning for image recognition[C]MEEE Conference on Computer Vision and Pattern Recognition. IEEE,2016:603-607.

# 附录

表 A1 成员分工及贡献度自评

| 姓名 | 个人分工 | 小组任务贡献度 |
| --- | --- | --- |
| 张三 | 具体内容，课程报告第1、2节 | 60% |
| 李四 | 具体内容，课程报告第3节 | 40% |
|  |  |  |
|  |  |  |

课程体会与建议

具体内容