学　　号　2017301500335

武汉大学
高级语言程序设计大作业报告

基于多层感知机网络的手写数字数据集分类算法及其优化

院 (系) 名 称：　计算机学院

专 业 名 称：　计算机类

学 生 姓 名：　张永康

指 导 教 师：　谭成予　副教授

二〇一八年五月

摘　　要

手写数字分类问题是机器学习与模式识别领域的基础问题之一，也是 OCR(光学字符识别) 的核心技术，具有重要的现实意义。本文针对 MNIST 手写数字数据集，通过 C++ 实现了基于多层感知机网络的手写数字数据集分类算法，并采用交叉验证、小批量随机梯度下降算法、扩充训练数据集的方法优化算法，最终达到了 98.042% 的分类准确率。

关键词: 多层感知机; 手写数字; 分类; 优化;

ABSTRACT

The classification of handwritten digits is one of the basic problems of machine learning and pattern recognition, which is also the key technology of optical character recognition(OCR). In this paper, according to the MNIST handwritten digits dataset, we implement a classifying algorithm based on multi-layer perceptron(MLP) by C++, and optimize it by cross-validation, mini-batch stochastic gradient descent algorithm and expanding the training dataset. The accuracy of our algorithm reaches 98.042%.

Key words: Multi-layer Perceptron; Handwritten Digits; Classification; Optimization;

目　录

1　MNIST 手写数字数据集

MNIST(Mixed National Institute of Standards and Technology database) 是一个包含 70000 个 28×28 像素大小的手写阿拉伯数字灰度图片的大型数据集[1]。MNIST 数据库被广泛用于测试机器学习算法的分类能力。

本文使用 Kaggle 在线大数据测试平台的 MNIST 测试系统[2] 来检验分类算法的分类正确率，该系统将 MNIST 数据库中的 42000 个数字作为训练集，28000 个数字作为测试集。其中，每个图片被转换为 28×28=784 维的向量。训练集为 42000 行、785 列的矩阵，矩阵中第 i 行第 1 列代表第 i 张图片对应的数字，之后的 784 列分别代表第 i 张图片中每个像素点的灰度值。测试集训练集为 28000 行、784 列的矩阵，矩阵中第 i 行第 j 列代表第 i 张图片中像素点 j 的灰度值。

2　MNIST 数据集的分类算法比较

MNIST 数据集的分类算法有很多，如 K 近邻算法、支持向量机[3]、集成学习方法、多层感知机网络、卷积神经网络等。目前分类准确率最高的是基于胶囊网络 (Capsule Network) 的分类算法[4]，其准确率为 99.75%。

K 近邻算法 (K Nearest Neighbors,KNN) 是一种度量学习方法，它基于某种距离度量，在训练集中找到与测试对象最近的 K 个训练数据，并在这 K 个数据中选取出现频率最高的数据类别作为测试对象的所属类别，具有算法容易实现的特点。但其无训练过程，在对每个数据对象进行分类时需要遍历整个数据集，耗时过长。

支持向量机 (Support Vector Machine,SVM) 将数据映射到高维空间，使用核函数 (Kernel Function) 寻找合适的超平面划分数据集，与线性回归算法相比，SVM 可以划分原始空间中线性不可分的数据集，具有计算量小、不易过拟合等优点，但数学推导过程过于复杂。

集成学习方法 (如 Adaboost、随机森林) 是通过构建并结合多个分类器来完成训练、分类任务。集成学习通过一定的策略训练每个子分类器、结合每个子分类器的分类结果，常常能获得比单个子分类器更好的分类准确率，具有泛化能力强的优点，缺点是算法难以理解、实现。

多层感知机网络 (Multi Layer Perceptron,MLP) 是由若干层感知机神经元组成的全连接网络。其中每个神经元可以解决一个线性可分问题。MLP 通过多层神经元的共同作用，可以解决大多数的线性不可分问题，其结构简单，算法易于实现，缺点是训练速度略慢，无法提取图片中的空间关系，有时需要通过特征工程对输入数据进行特征提取。

卷积神经网络 (Convolutional Neural Network,CNN) 是由若干卷积层、下采样层组成的神经网络，通过卷积层的卷积核提取图片中的特征信息，具有学习图片中一定特征的能力，分类准确率高，缺点是结构复杂、反向传播算法难以实现、训练时间过长。

综合考虑算法实现难度、训练时间成本等因素，本文采用由两层感知机神经

元组成的多层感知机网络实现分类算法。

3　基于多层感知机网络的分类算法

3.1　感知机

感知机神经元 (Perceptron) 是多层感知机网络的基本组成单位，由输入向量 X、权重向量 W、偏置 b、激励函数 σ 和输出值组成。输入向量经权重向量加权求和后输入到感知机中，然后经激励函数得到输出值。

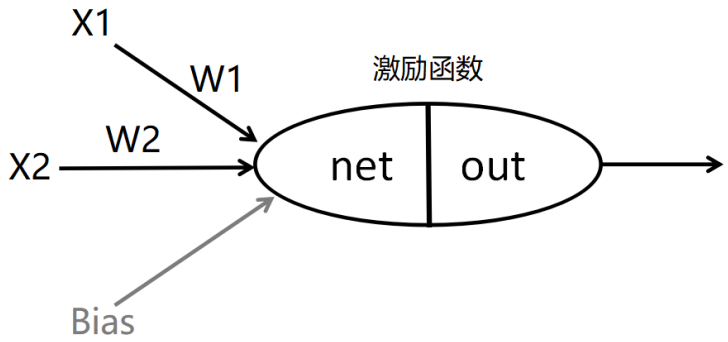

图 3.1　感知机神经元结构

感知机的结构如图 3.1所示。其中，激励函数一般为非线性函数，将输入值映射到一个较小区间中，以提高感知机对非线性函数的拟合能力。激励函数有很多，如 $Sigmoid$、$tanh$、$ReLU$ 等。

本文采用最常见的 Sigmoid 函数作为感知机的激励函数。Sigmoid 函数及其导函数可表示为：

$$Sigmoid(x) = \frac{1}{1+e^{-x}} \tag{3.1}$$

$$Sigmoid'(x) = Sigmoid(x)(1 - Sigmoid(x)) \tag{3.2}$$

3.2　多层感知机网络

单个感知机已被证明不能解决线性不可分问题 (如异或问题)，但多层感知机网络可以解决大多数此类问题。本文采用由两层感知机神经元组成的多层感知机

网络，该网络包含输入层、隐含层与输出层。

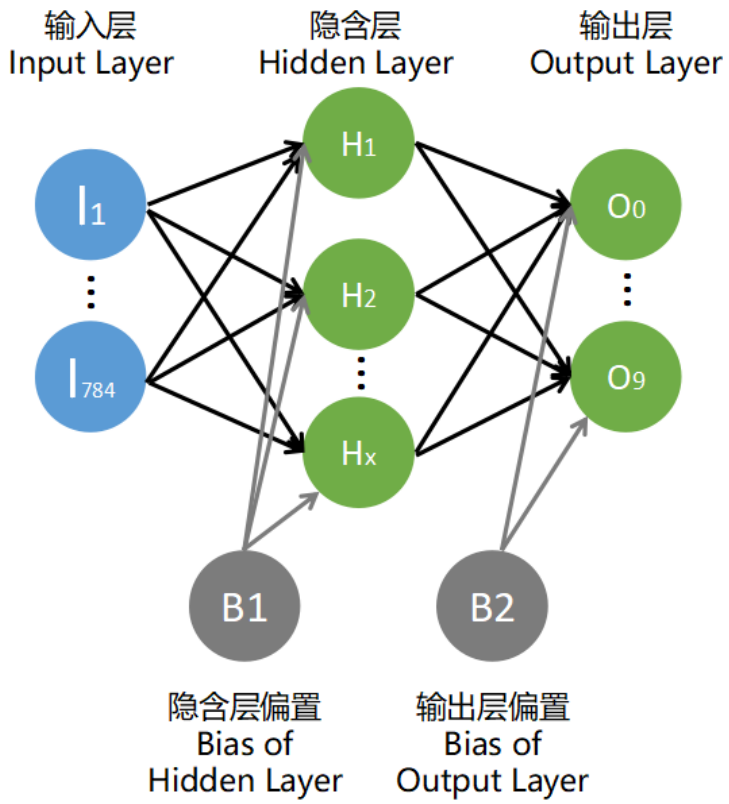

图 3.2　多层感知机网络结构

该网络结构如图3.2所示。其中输入层由 784 个结点构成，分别代表经归一化处理后的输入数据的每个像素点的灰度值，隐含层由 n_h 个感知机神经元组成，输出层由 10 个感知机神经元组成，它们的输出分别代表输入图片为数字 0 到 9 的概率。

3.3　正向传播算法

3.3.1　输入数据的预处理

为了提高训练的收敛速率、最终的分类准确率，需要对输入数据进行归一化 (Normalization) 处理，使得输入数据由较大的区间 $[0, 255]$ 映射到区间 $[0, 1]$ 上。本文采用 minmax 归一化方法。

设输入数据的 784 维中灰度值最小值为 min，最大值为 max，则该数据的第

i 维灰度值归一化为：

$$input_i \leftarrow \frac{input_i - min}{max - min} \tag{3.3}$$

3.3.2 输入数据的正向传播

输入数据通过隐含层、输出层正向传播后得到输出结果。

对于第 i 个隐含层神经元 h_i，它接受来自输入层的 784 个输入值 $input_j$，对它们加权求和并加上偏置 b_h 得到 net_{h_i}：

$$net_{h_i} = \sum_{j=1}^{784} w_j input_j + b_h \tag{3.4}$$

然后经激励函数处理后得到隐含层神经元的输出值 out_{h_i}：

$$out_{h_i} = \sigma(net_{h_i}) \tag{3.5}$$

对于第 i 个输出层神经元 h_i，它接受来自隐含层的 n_h 个输入值 out_{h_j}，对它们加权求和并加上偏置 b_o 得到 net_{o_i}：

$$net_{o_i} = \sum_{j=1}^{n_h} w_j out_{h_j} + b_h \tag{3.6}$$

然后经激励函数处理后得到输出层神经元的输出值 out_{o_i}：

$$out_{o_i} = \sigma(net_{o_i}) \tag{3.7}$$

3.3.3 误差估计

在训练过程中，输入数据经正向传播得到输出数据后，需要通过误差估计，将其与期望得到的输出数据作比对。

误差估计的方法有很多，常见的有均方误差 (Mean Squared Error,MSE)、交叉熵误差 (Cross Entropy Error,CEE)。本文采用 MSE 作为误差估计方法。

对于输出层的第 i 个神经元，其输出 out_{o_i} 与期望输出 $output_i$ 之间的 MSE 为：

$$E_i = \frac{1}{2}(out_{o_i} - output_i)^2 \tag{3.8}$$

则总的 MSE 可表示为：

$$E = \sum_{i=0}^{9} E_i \tag{3.9}$$

3.4 反向传播算法

多层感知机网络中包含很多边权与偏置权重，因此需要高效的算法来调整这些权重，使得网络的 MSE 尽可能小，这一过程称之为训练 (Train)。

训练算法有很多，如反向传播算法 (Error Backpropagation Algorithm,BP)、神经进化算法 (Neuro Evolution) 等，本文采用最经典的反向传播算法。

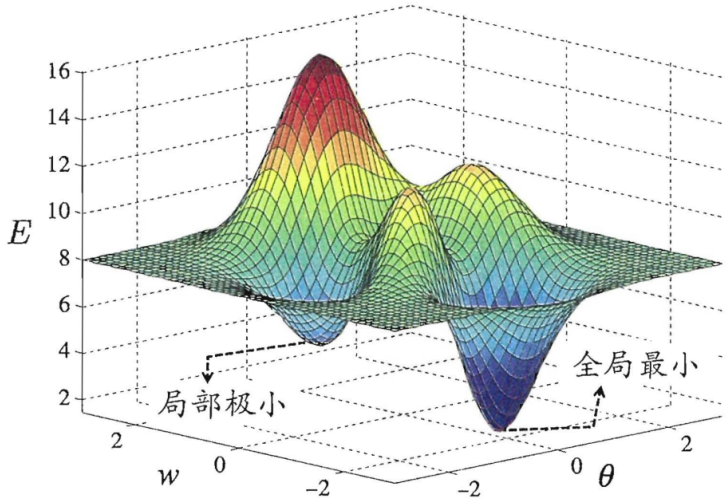

图 3.3　全局最小与局部最小

BP 算法的本质是对于每个权重，求出总 MSE 对该权重的偏导数，并根据偏导数将该权重沿使 MSE 减小的方向调整。

两层结构的多层感知机网络的 BP 算法具体如下[5]。

3.4.1 隐含层到输出层边权的权值更新

设隐含层神经元 i 到输出层神经元 j 的边权为 $w_{i,j}$，则总的 MSE 对其的偏导数为：

$$\frac{\partial E}{\partial w_{i,j}} = \frac{\partial E}{\partial out_{o_j}} \frac{\partial out_{o_j}}{\partial net_{o_j}} \frac{\partial net_{o_j}}{\partial w_{i,j}} \tag{3.10}$$

其中，

$$\frac{\partial E}{\partial out_{o_j}} = out_{o_j} - output_j \tag{3.11}$$

$$\frac{\partial out_{o_j}}{\partial net_{o_j}} = \sigma'(net_{o_j}) = out_{o_j}(1 - out_{o_j}) \tag{3.12}$$

$$\frac{\partial net_{o_j}}{\partial w_{i,j}} = out_{h_i} \tag{3.13}$$

将式3.11、3.12、3.13代入式3.10：

$$\frac{\partial E}{\partial w_{i,j}} = (out_{o_j} - output_j)out_{o_j}(1 - out_{o_j})out_{h_i} \tag{3.14}$$

定义隐含层到输出层边权的学习率为 $\eta_{h,o}$，则 $w_{i,j}$ 的更新公式为：

$$w_{i,j} \leftarrow w_{i,j} - \eta_{h,o}\frac{\partial E}{\partial w_{i,j}} \tag{3.15}$$

3.4.2 输出层偏置的权值更新

设输出层偏置为 b_o，则总的 MSE 对其的偏导数为：

$$\frac{\partial E}{\partial b_o} = \frac{\partial E}{\partial out_{o_j}}\frac{\partial out_{o_j}}{\partial net_{o_j}}\frac{\partial net_{o_j}}{\partial b_o} \tag{3.16}$$

其中，

$$\frac{\partial E}{\partial b_o} = 1 \tag{3.17}$$

将式3.11、3.12、3.17代入式3.16：

$$\frac{\partial E}{\partial b_o} = (out_{o_j} - output_j)out_{o_j}(1 - out_{o_j}) \tag{3.18}$$

定义输出层偏置权值的学习率为 η_{b_o}，则 b_o 的更新公式为：

$$b_o \leftarrow b_o - \eta_{b_o}\frac{\partial E}{\partial b_o} \tag{3.19}$$

参考文献

[1] Lecun Y, Cortes C. The mnist database of handwritten digits[DB/OL].

[2] Digit Recognizer[EB/OL]. (2018-03-18)[3/18]. https://www.kaggle.com/c/digit-recognizer.

[3] Burges C J C. A Tutorial on Support Vector Machines for Pattern Recognition[J]. Data Mining and Knowledge Discovery, 1998,2(2):121-167.

[4] Sabour S, Frosst N, Hinton G E. Dynamic Routing Between Capsules[J]. CoRR, 2017,abs/1710.09829.

[5] 周志华. 机器学习 [M]. 清华大学出版社, 2016.

致　　谢

感谢你, 感谢他和她, 感谢大家.

附录 A　测试

A.1　第一个测试

测试公式编号

$$1+1=2. \tag{A.1}$$

表格编号测试

表 A.1　测试表格

11	13	13	13	13
12	14	13	13	13

附录 B　附录测试

测试

附录 C　附录测试

测试