基于 PB 人工神经网络的数字手写体识别模型

冯唐正
南京理工大学
电子工程与光电技术学院

版本：0.10
日期：2024 年 8 月 8 日

摘　要

人工神经网络是实现人工智能的关键技术。反向传播是一种通过最小化预测误差来优化神经网络参数的算法，其核心思想是利用梯度下降法调整网络权重，以提高预测准确性。本文讨论了反向传播人工神经网络的原理，阐述了该模型的前向传播与反向传播以及参数更新的过程，并用 MNIST 数字手写体数据集进行模型验证。该模型识别准确率在测试集中达到 97.64%。

关键词: 人工智能；人工神经网络；bp 人工神经网络；图像识别；分类任务

1 引言

在本文中，首先介绍了 BP 神经网络的基本结构，包括输入层、隐含层和输出层。我们具体说明了每个神经元的工作机制及其数学描述。网络的设计包括输入层的维度设定、隐含层的激活函数（ReLU）和输出层的激活函数（Softmax）。交叉熵损失函数用于度量模型的预测与实际标签之间的差距。

通过前向传播和反向传播的详细推导，本文展示了如何计算从损失函数到每一层权重和偏置的梯度。特别是在输出层，Softmax 函数的导数和交叉熵损失函数的梯度被用来优化模型的参数。反向传播算法通过计算梯度并更新参数，从而逐步提高网络的性能。

本研究中的 BP 神经网络模型展示了人工智能（AI）领域中如何应用基础的神经网络理论解决实际问题。该模型的有效性在于其通过优化学习算法和调整网络参数来不断提升分类精度。这些原理不仅适用于数字识别，还为其他复杂任务中的 AI 模型提供了理论基础。

2 网络结构

2.1 神经元结构

一个神经元从多个输入端感受信号刺激，通过线性系统和非线性的激活函数输出一个激活值。假设信号的输入矢量是 $\mathbf{X}_{in} \in \mathbb{R}^n$，其中 n 是输入信号的维度，若该神经元的参数特性有权重矢量 $\mathbf{W} \in \mathbb{R}^n$，偏置量 b，则该神经元的信号输入 $\mathbf{X}_{in}$ 与激活值 x_{out} 满足

$$x_{out} = f(\mathbf{W}^T \cdot \mathbf{x}_{in} + b) \tag{1}$$

其中，$f(\cdot)$ 为激活函数。

2.2 层结构

BP 神经网络有三种层，分别是**输入层（Input）**、**隐含层（Hidden）**、**输出层（Output）**。隐含层可以有若干个。有关研究表明，一个隐含层的神经网络，只要神经元足够多，就可以以任意精度逼近一个非线性函数。因此，通常采用含有一个隐层的三层多输入单输出的 BP 神经网络建立预测模型。若是神经元数目过少，则会影响网络性能，达不到预期效果；若隐层神经元数目过多，会加大网络计算量并容易产生过度拟合问题。

设若某个层有 m 个神经元，其前一层有 n 个神经元，则这两层之间的信号传递满足

$$\mathbf{x}_m = f(\mathbf{W}_{m\times n} \cdot \mathbf{x}_n + \mathbf{b}_m) \tag{2}$$

其中，$\mathbf{x}_n \in \mathbb{R}^n$ 与 $\mathbf{x}_m \in \mathbb{R}^m$ 分别为当前层输入与输出矢量、$\mathbf{W} \in \mathbb{R}^{m\times n}$ 为当前层的权重矩阵，$\mathbf{b} \in \mathbb{R}^m$ 为偏置矢量。

3 网络设计

3.1 输入层与输出层维度

将图像的二值化矩阵从 (m, n) 展平成 $(m*n, 1)$ 作为特征输入层，则输入层的维度为 $m*n$；设计输出层维度为十个节点，每个节点输出表征特征输入信号在当前节点的分类置信度。将每个图像的标签写成one-hot 编码，则神经网络的学习目标就是输出层输出信号尽可能接近标签 one-hot 编码。

3.2 激活函数

3.2.1 隐含层激活函数

ReLU 函数可以很有效解决 [梯度消失] 的问题。

$$ReLU(x) = \begin{cases} 0 & x \le 0 \\ x & x > 0 \end{cases} \tag{3}$$

3.2.2 输出层激活函数

Softmax 用于表征输出层的分类置信度（概率）。给定 $\forall \mathbf{z} = (z_1\ z_2\ ...\ z_n)^T$，Softmax 输出

$$y_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{4}$$

且

$$\sum_{i=1}^{n} y_i = 100\% \tag{5}$$

3.3 损失函数

采用交叉熵函数作为损失函数。给定 $\forall \mathbf{y} = (y_1\ y_2\ ...\ y_n)^T$ 与对应 $\hat{\mathbf{y}} = (\hat{y}_1\ \hat{y}_1\ ...\ \hat{y}_n)^T$，有

$$H(\mathbf{y}, \hat{\mathbf{y}}) = -\sum_{i=1}^{n} y_i \ln(\hat{y}_i) \tag{6}$$

其中，$\mathbf{y}$ 为标签真值 one-hot 编码矢量，$\hat{\mathbf{y}}$ 为模型置信度矢量。

3.4 权重矩阵与偏置矢量

3.4.1 隐含层

令隐含层权重矩阵为 $\mathbf{W}_h \in \mathbb{R}^{k \times m*n}$，$\mathbf{b}_h \in \mathbb{R}^k$ 为偏置矢量，其中，k 为隐含层维度。

3.4.2 输出层

令输出层权重矩阵为 $\mathbf{W}_o \in \mathbb{R}^{10 \times k}$，$\mathbf{b}_o \in \mathbb{R}^{10}$ 为偏置矢量。

4 前向传播

4.1 输出层到隐含层

设若输入信号的矢量为 $\mathbf{x}_{in} \in \mathbb{R}^{m*n}$，令

$$\mathbf{z_h} = \mathbf{W}_h \cdot \mathbf{x}_{in} + \mathbf{b}_h \tag{7}$$

其中，$\mathbf{z}_h \in \mathbb{R}^k$。则隐含层的激活值 $\mathbf{x}_h$ 有

$$\mathbf{x}_h = f(\mathbf{z}_h) \tag{8}$$

$$= ReLU(\mathbf{z}_h) \tag{9}$$

其中，激活函数 $f(\cdot)$ 为 ReLU。对 $\forall \mathbf{z}_h = (z_1\ z_2\ ...\ z_k)^T$，对应 $\mathbf{x}_h = (x_{h1}\ x_{h2}\ ...\ x_{hk})^T$ 有

$$x_{hi} = \begin{cases} 0 & if\ z_i \leq 0 \\ z_i & otherwise \end{cases} \tag{10}$$

4.2 隐含层到输出层

令

$$\mathbf{z}_o = \mathbf{W}_o \cdot \mathbf{x}_h + \mathbf{b}_o \tag{11}$$

其中，$\mathbf{z}_o \in \mathbb{R}^{10}$。则输出层的激活值 $\hat{\mathbf{y}}$ 有

$$\hat{\mathbf{y}} = f(\mathbf{z}_o) \tag{12}$$

$$= Softmax(\mathbf{z}_o) \tag{13}$$

$$\tag{14}$$

其中，激活函数 $f(\cdot)$ 为 Softmax。对 $\forall \mathbf{z}_o = (z_1\ z_2\ ...\ z_{10})^T$，对应 $\hat{\mathbf{y}}_h = (\hat{y}_1\ \hat{y}_2\ ...\ \hat{y}_{10})^T$ 有

$$\hat{y}_i = \frac{e^{z_i}}{\sum_{j=1}^{n} e^{z_j}} \tag{15}$$

5 反向传播

5.1 损失函数到输出层的梯度

交叉熵 H 对 $\hat{y}_i$ 求偏导，有

$$\frac{\partial H}{\partial \hat{y}_i} = -\frac{y_i}{\hat{y}_i} \tag{16}$$

5.2 损失函数到隐含层的梯度

Softmax 函数值 $\hat{y}_i$ 对 z_{oi} 求偏导，有

$$\frac{\partial \hat{y}_i}{\partial z_{oi}} = \frac{e^{z_{oi}}(\sum - e^{z_{oi}})}{\sum^2} \tag{17}$$

$$= \hat{y}_i \cdot (1 - \hat{y}_i) \tag{18}$$

Softmax 函数值 $\hat{y}_i$ 对 $z_{oj}, i \neq j$ 求偏导，有

$$\frac{\partial \hat{y}_i}{\partial z_{oj}} = \frac{-e^{z_{oi}+z_{oj}}}{\sum^2} \tag{19}$$

$$= -\hat{y}_i \hat{y}_j \tag{20}$$

综上所述，由链式法则，损失函数到隐含层的梯度有

$$\frac{\partial H}{\partial z_{oi}} = \sum_{j=1}^{10} \frac{\partial H}{\partial \hat{y}_j} \cdot \frac{\partial \hat{y}_j}{\partial z_{oi}} \tag{21}$$

$$= -\frac{y_i}{\hat{y}_i}(\hat{y}_i(1 - \hat{y}_i) + \sum_{j \neq i} \hat{y}_i \hat{y}_j) \tag{22}$$

$$= -y_i + y_i \hat{y}_i + \sum_{j \neq i} y_i \hat{y}_j \tag{23}$$

$$= -y_i + \hat{y}_i \sum \hat{y}_j \tag{24}$$

又

$$\sum_{j=1}^{10} \hat{y}_j = 1 \tag{25}$$

所以有

$$\frac{\partial H}{\partial z_{oi}} = \hat{y}_i - y_i \tag{26}$$

令

$$\delta_o = \hat{\mathbf{y}} - \mathbf{y} \tag{27}$$

5.2.1 输出层权重矩阵与偏置矢量的梯度

- **权重矩阵**

$$\nabla_{\mathbf{W}_o} H = \delta_o \cdot \mathbf{x}_h^T \tag{28}$$

- **偏置矢量**

$$\nabla_{\mathbf{b}_o} H = \delta_o \tag{29}$$

5.3 损失函数到输入层的梯度

对 ReLU 函数值 x_{hi} 对 z_{hi} 求偏导，有

$$\frac{\partial x_{hi}}{\partial z_{hi}} = \begin{cases} 0 & if\ z_{hi} \leq 0 \\ 1 & otherwise \end{cases} \tag{30}$$

令

$$\delta_h = (\mathbf{W}_o^T \delta_o) \odot (\nabla_{\mathbf{z}_h} f) \tag{31}$$

其中，$f(\cdot)$ 为 ReLU。

5.3.1 隐含层权重矩阵与偏置矢量的梯度

- **权重矩阵**

$$\nabla_{\mathbf{W}_h} H = \delta_h \cdot \mathbf{x}_{in}^T \tag{32}$$

- **偏置矢量**

$$\nabla_{\mathbf{b}_h} H = \delta_h \tag{33}$$

5.4 参数更新

$$\mathbf{W}_o' = \mathbf{W}_o - \eta \nabla_{\mathbf{W}_o} H \tag{34}$$

$$\mathbf{b}_o' = \mathbf{b}_o - \eta \nabla_{\mathbf{b}_o} H \tag{35}$$

$$\mathbf{W}_h' = \mathbf{W}_h - \eta \nabla_{\mathbf{W}_h} H \tag{36}$$

$$\mathbf{b}_h' = \mathbf{b}_h - \eta \nabla_{\mathbf{b}_h} H \tag{37}$$

$$\tag{38}$$

6 数字手写体识别

6.1 数据集预览

MNIST 数据集包括 60,000 张数字手写体图片训练集与 10,000 张数字手写体图片的测试集。每张图片的大小为 28*28。

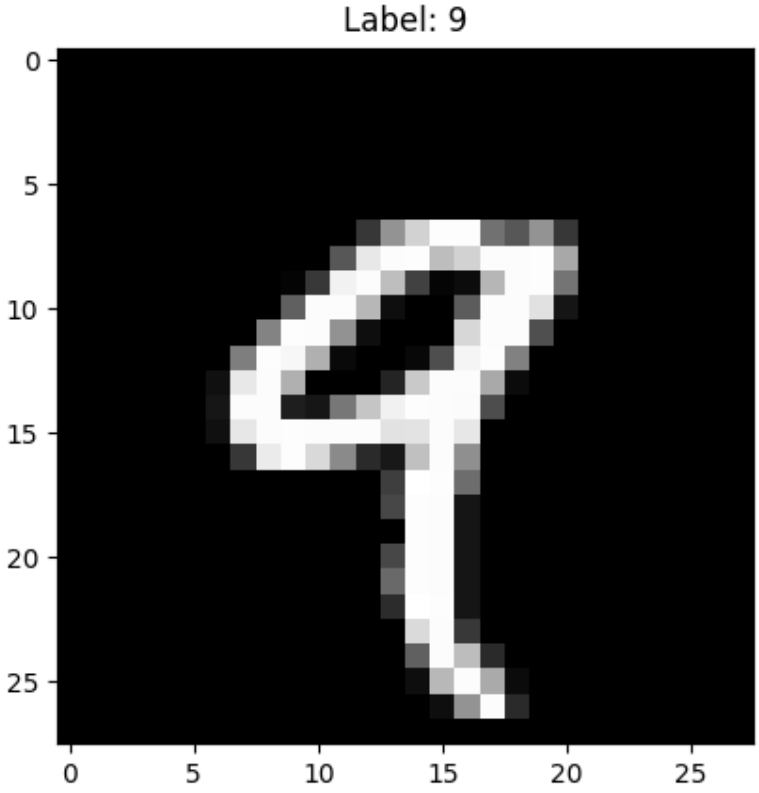

图 1: 手写体样例

6.2 网络结构参数

在3中的模型，取 $m = n = 28$，则输入层神经元维度为 $\mathbf{x}_{in} \in \mathbb{R}^{784}$；取隐含层维度 $k = 256$。输出层维度为 $\hat{\mathbf{y}} \in \mathbb{R}^{10}$。

6.3 训练结果

如图 2 所示，在 20 次 epoch 之后，模型的交叉熵损失在 10^{-10} 的数量级，识别准确率达到 97.64%，训练时间为 21min43s。

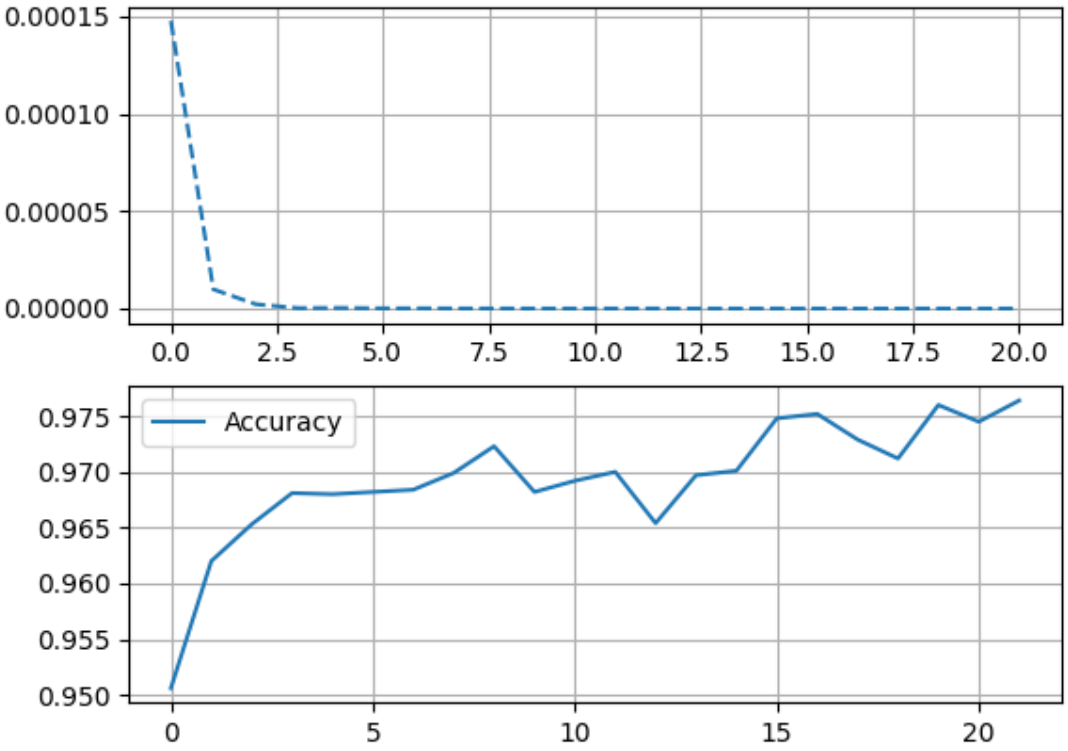

图 2: 损失函数与识别准确率曲线

7 结论

反向传播（Back Propagation, BP）人工神经网络是一种重要的深度学习模型，通过模拟生物神经网络的结构和功能来处理复杂的计算任务。BP 算法是用于训练神经网络的核心技术，其通过计算损失函数的梯度并更新网络中的权重和偏置，使得模型能够逐渐优化性能。

7.1 网络结构

BP 神经网络通常由输入层、一个或多个隐含层和输出层组成。每一层的神经元通过权重和偏置相连接，使用激活函数（如 ReLU 和 Softmax）来引入非线性，从而增强模型的表达能力。

7.2 前向传播

在前向传播过程中，输入数据通过网络层层传递，并在每一层计算加权和偏置后的激活值，最终输出结果。输出层的激活值用于计算损失函数，从而衡量模型的预测与实际结果之间的差距。

7.3 反向传播

反向传播是 BP 神经网络的核心，主要通过以下步骤进行：

- **损失计算**：首先计算输出层的损失（如交叉熵损失），并通过链式法则逐层计算损失相对于每个参数的梯度。
- **梯度计算**：通过梯度下降法，根据梯度值调整每一层的权重和偏置，使得损失函数逐渐减小。
- **参数更新**：利用学习率来控制更新步长，逐步优化模型的参数，使其更好地拟合训练数据。

7.4 应用与挑战

BP 神经网络广泛应用于图像分类、语音识别、自然语言处理等领域。然而，训练 BP 神经网络也面临一些挑战，如梯度消失、过拟合和计算资源消耗等问题。为此，通常采用适当的正则化方法、早停法以及加深网络结构来缓解这些问题。

参考文献

[1] Toshitaka Hayashi, Dalibor Cimr, and Richard Cimler. “Machine Learning Could be Easier if All Data Were MNIST”. In: *New Trends in Intelligent Software Methodologies, Tools and Techniques: Proceedings of the 22nd International Conference on New Trends in Intelligent Software Methodologies, Tools and Techniques (SoMeT_23).Naples, Italy.20-22 September 2023*. Fcuy of Science, Universiy of Hrdec Krove, Fcuy of Science, Universiy of Hrdec Krove, and Fcuy of Science, Universiy of Hrdec Krove. Naples, 2023, pp. 185–194.

[2] Hailong Xi, Haiyan Liu, and Yu Zhang. “Recognition and Optimization Algorithm of MNIST Dataset Based on LeNet5 Network Structure”. In: *International Conference on Transportation & Logistics, Information & Communication, Smart City: TLICSC 2018, Chengdu City, China, 30-31 October 2018*. Chengdu, 2019, pp. 322–328.

[3] 孟雪. “基于 BP 神经网络的图像识别”. In: 软件 43.7 (July 2022), pp. 137–141.

[4] 崔海霞, 杨红, and 刘佐濂. “MNIST 邮政编码手写数字识别的研究”. In: 广州大学学报（自然科学版）8.4 (Jan. 2009), pp. 14–18.

[5] 施强. “基于 BP 神经网络的计算机图像智能识别方法”. In: 电脑编程技巧与维护 10 (Oct. 2022), pp. 134–137.

[6] 曾琪. “基于 BP 神经网络的图像自动识别技术”. In: 自动化应用 11 (Nov. 2022), pp. 73–75, 80.

[7] 郭梦洁, 杨梦卓, and 马京九. “基于 Keras 的 MNIST 数据集识别模型”. In: 现代信息科技 3.14 (Jan. 2019), pp. 18–19, 23.

[8] 韩雷. “基于 BP 神经网络的学术论文评价模型研究”. In: 现代情报 44.2 (Feb. 2024), pp. 170–177.