

文章编号: 2095-2163(2020)11-0144-04

中图分类号: TP311.1

文献标志码: A

基于 TensorFlow 手写体数字识别系统的研究

李建新

(广州工商学院 计算机科学与工程系, 广州 528138)

摘要: 手写体数字识别是人工智能识别系统中的重要组成部分,本文基于 TensorFlow 深度学习模型,完成了手写体数字识别及应用。首先建立 TensorFlow 深度学习模型,分析了卷积神经网络(CNN)模型结构及 Softmax 模型结构,对手写体数据集 MNIST 中的 60 000 个样本进行深度学习,并对 10 000 个样本的测试进行对比。通过深度学习得到手写体数字的识别模型参数,进而使用模型参数识别用户手写体数字。实测数据验证,相对于传统的 Softmax 模型,基于 TensorFlow 深度学习的 CNN 模型准确率高达 100%,提升了 7%。该研究为人工智能数字识别系统应用提供了一定的科研价值。

关键词: 机器学习;深度学习;卷积;TensorFlow

Research on handwritten digit recognition system based on Tensorflow

LI Jianxin

(Dept of Computer Science and Engineering, Guangzhou College of Technology and Business, Guangzhou 528138, China)

[Abstract] Handwritten numeral recognition is an important part of artificial intelligence recognition system. Handwritten numeral recognition and application are completed based on tensorflow deep learning model. Firstly, tensorflow deep learning model is established, convolutional neural network (CNN) model structure and softmax are analyzed, and then 60 000 samples of handwritten dataset MNIST are deeply learned, and 10 000 samples are processed Through deep learning, we can get the recognition model parameters of handwritten numbers, and use the model parameters to recognize the handwritten numbers of users. Compared with the traditional softmax model, the accuracy of CNN model based on tensorflow in-depth learning is up to 100%, 7% higher. The research provides a certain research value for the application of AI digital recognition system.

[Key words] machine learning; deep learning; Convolutional Neural Network(CNN); TensorFlow

0 引言

目前,手写体数字识别是人工智能机器学习领域众多研究者关注的一个热点,广泛应用于公安、税务、交通、金融、教育等行业的实践活动中。在实际应用中,对手写数字单字识别率的要求比手写普通文字识别要苛刻许多,识别精度需要达到更高的水平。由于手写数字识别没有文字识别那样的上下文,不存在语意的相关性,而数据中的每一个数字又至关重要。因此,提高手写数字识别率,成为人工智能的重要研究领域。

人工神经网络是一种类似于大脑神经突触联接的结构,进行信息处理的数学模型,具有学习和记忆功能,可以让神经网络学习各个模式类别中的大批量训练样本,记忆各个模式类别中的样本特征。在识别样本时,将逐个与记忆中各模式类别的样本特征相比较,从而确定样本所属类别。人工神经网络通过学习调整内部大量节点之间相互连接的关系,可获得众多优化的模型参数。

1 手写体数字识别系统

1.1 系统组成

手写体数字识别系统由输入、预处理、识别等环节组成,系统架构如图 1 所示。

输入环节包括 MNIST 手写体数据集输入和用户预测图片里的手写体数字;预处理环节主要指用户手写体数字预测图片的预处理;识别环节包括卷积神经网络、分类识别器、损失函数和优化器。卷积神经网络实现数字手写体 0~9 特征提取,分类识别是手写体数字识别最关键的部分。经过卷积神经网络提取来的特征被送到分类器中,对其进行分类识别。分类识别器由 softmax 完成,softmax 得到当前样例中属于不同种类的概率分布情况。

优化器在神经网络训练和测试过程中,根据计算结果与损失函数要求,调节模型可变参数,让神经网络参数尽快达到理想误差范围。

1.2 MNIST 数据集训练与测试

训练一个卷积神经网络时,需要大量的数据、设置卷积的层数、卷积核大小、池化的方式(最大、平均)、损失函数、设置目标(如:手写数字的识别,输

作者简介:李建新(1954-),男,硕士,教授,高级工程师,主要研究方向:计算机智能制造。

收稿日期:2020-08-23

出一个 10 维向量)等,让卷积神经网络通过不断的训练更新参数,向设置的目标靠近,最后可以通过这些参数来预测样本。

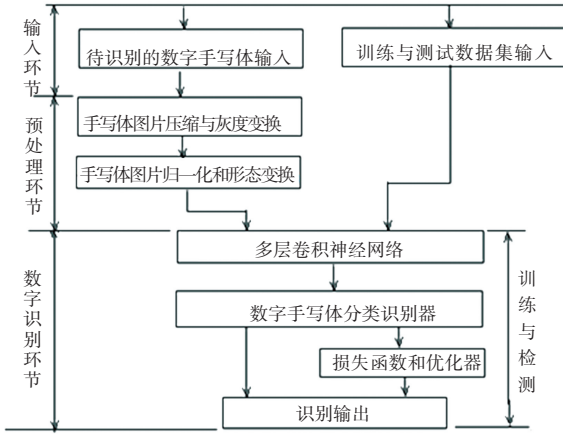


图 1 手写体数字识别系统组成

Fig. 1 Composition of handwritten digit recognition system

通常情况下,假设样本由 K 个类组成,共有 m 个,则数据集可由式(1)表示:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}. \quad (1)$$

式中, $x^{(i)} \in R(n + 1)$; $y^{(i)} \in \{1, 2, \dots, K\}$, $n + 1$ 为特征向量 x 的维度。

MNIST 是机器学习中的一个数据集,包含 0~9 的不同人手写数字图片。数据集由 60 000 个手写数字训练样本和 10 000 个手写数字测试样本组成,每个样本是一张 $28 * 28$ 像素的灰度手写数字图片。则 MNIST 数据集的训练集 $m = 60\ 000, n + 1 = 784, K = 10$, MNIST 数据集的测试集 $m = 10\ 000, n + 1 = 784, K = 10$;分别由式(2)、式(3)表示为:

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(60\ 000)}, y^{(60\ 000)})\}, \quad (2)$$

$$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(10\ 000)}, y^{(10\ 000)})\}. \quad (3)$$

机器学习模型设计时,将提供大量的数据以训练一个手写体数字的机器学习,所得到的模型参数,最终用于预测用户图片里面的数字。

1.3 卷积神经网络与分类器

卷积神经网络是系统的核心部分,是由输入层、多个卷积层、多个池化层、多个全连接层以及输出组成的多层感知器。卷积层为局部连接,用于提取图像特征;池化层用于进一步降低输出参数数量,提高了模型的泛化能力;全连接层进行特征关系和权重值计算,完成分类处理。手写体数字识别卷积神经网络的组成示意如图 2 所示。

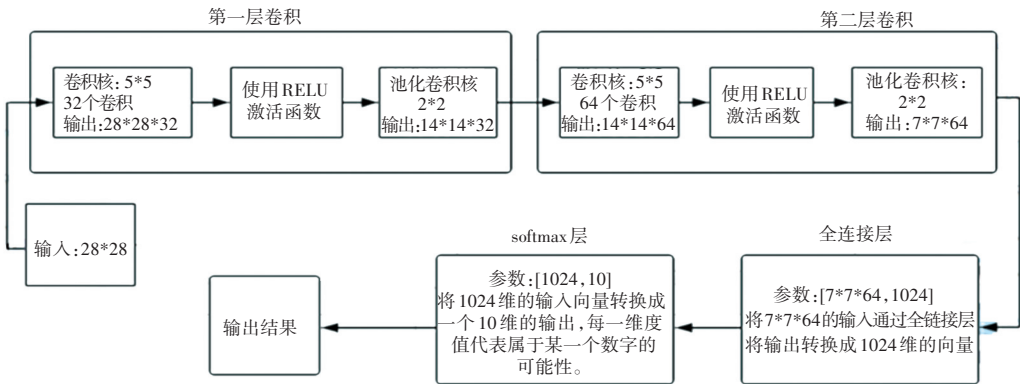


图 2 手写体数字识别卷积神经网络组成

Fig. 2 Composition of convolution neural network for handwritten digit recognition

在 TensorFlow 中卷积层、池化层和全连接层部分代码如下:

```
input = tf.placeholder(tf.float32, [None, 784]) # 定义数据集扁平化手写数字图片
input_image = tf.reshape(input, [-1, 28, 28, 1]) # 扁平化数据集形态调整 28x28, 单输入通道
y = tf.placeholder(tf.float32, [None, 10]) # 定义数据集监督标签
..... # 函数定义省略
filter = [5, 5, 1, 32] # 卷积核形态, 5x5, 单输入
```

```
通道、32 输出通道
filter_conv1 = weight_variable(filter) # 生成卷积核
b_conv1 = bias_variable([32]) # 生成与卷积核 32 输出通道相应偏移
h_conv1 = tf.nn.relu(conv2d(input_image, filter_conv1) + b_conv1) # 卷积并激活函数
h_pool1 = max_pool(h_conv1) # 池化运算
h_flat = tf.reshape(h_pool1, [-1, 14 * 14 * 32]) # 卷积层输出形态调整, 适应全连接
```

```

W_fc1 = weight_variable([14 * 14 * 32, 768])
#全连接权值矩阵形态设置
b_fc1 = bias_variable([768])#生成与全连接
层相应偏移矩阵
h_fc1 = tf.matmul(h_flat, W_fc1) + b_fc1#生成
全连接层
W_fc2 = weight_variable([768, 10]) #softmax
层权值矩阵形态设置
b_fc2 = bias_variable([10])#生成 softmax 层
相应偏移矩阵
y_hat = tf.nn.softmax(tf.matmul(h_fc1, W_fc2)
+ b_fc2)#生成 softmax 层

```

为了提高卷积神经网络对手写体数字识别的准确率,可以通过多种途径来实现。如:增加卷积层数量,以便进行更高层次的特征提取;改变各卷积层的卷积核大小,以便调整特征提取范围,或改变卷积步长;增加卷积层中卷积核的数量,意味着提取特征图的数量(一个卷积核卷积出来一个特征图),也就意味着可以从更多的角度提取原图的特征,避免视角单一,卷积出来多个特征图后要重新组织输出形态,用以适应下一层的输入要求。

分类器采用 Softmax 回归算法,该算法能将二分类的 Logistic 回归问题扩展至多分类。Softmax 回归算法对于给定的输入值 $x^{(i)}$, 输出的 K 个估计概率可由式(4)^[1]表示:

$$h_w(\vec{x}) = \frac{1}{\sum_{i=1}^k e^{\vec{w}_i \vec{x} + b_i}} \begin{bmatrix} e^{\vec{w}_1 \vec{x} + b_1} \\ e^{\vec{w}_2 \vec{x} + b_2} \\ \vdots \\ e^{\vec{w}_k \vec{x} + b_k} \end{bmatrix}. \quad (4)$$

其中, K 为类别的个数, \vec{w}_i 和 b_i 为第 i 个类别对应的权重向量和偏移向量,在 TensorFlow 中用张量表示。

式中,分量 $\frac{e^{\vec{w}_j \vec{x} + b_j}}{\sum_{i=1}^k e^{\vec{w}_i \vec{x} + b_i}} = P(y = j | \vec{x})$ 是样本 $x^{(i)}$ 的标签为第 j 个类别的概率,且有 $\sum_j P(y = j | \vec{x}) = 1$ 。

上述 Softmax 回归算法在 TensorFlow 中的相应代码如下:

```

y_model = tf.nn.softmax(tf.matmul(x_data,
weight) + bias)

```

1.4 损失函数和优化器

通常采用的损失函数“交叉熵”(cross -

entropy),如式(5)^[2]所示:

$$H_y(y) = - \sum_i y_i' \log(y_i). \quad (5)$$

式中, y 为预测概率的分布, y' 为实际样本的标签值。

TensorFlow 中提供了 8 种优化器,即 BGD、SGD、Momentum、Nesterov Momentum、Adagrad、Adadelata、RMSprop 和 Adam,模型训练一般采用梯度下降优化器。

在 TensorFlow 中,损失函数、学习率为 learning_rate 的优化器代码^[2]如下:

```

cross_entropy = -tf.reduce_sum(y_ * tf.log(y_
conv))

```

```

train_step = tf.train.AdamOptimizer(learning_
rate).minimize(cross_entropy)

```

```

train_setp = tf.train.GradientDescentOptimizer
(learning_rate).minimize(cross_entropy)

```

在 TensorFlow 中,求取数据准确率的代码如下:

```

correct_prediction = tf.equal(tf.argmax(y_conv,
1), tf.argmax(y_, 1))

```

```

accuracy = tf.reduce_mean(tf.cast(correct_
prediction, "float"))

```

由以上系统组成可知, MNIST 数据集、卷积神经网络与分类器、损失函数和优化器是人工神经网络学习模型训练机制的组成部分。

2 用户手写体数字识别

通过对 MNIST 数据集的训练和测试,模型参数达到要求并予以保存。在用户实际应用时可以导入模型参数对用户手写数字识别。由于 MNIST 数据集手写数字以张量形式存在,其形态为 1 维[784],而用户手写数字可以各种像素的数字图像存在,故用户手写数字在识别前需要将其图像进行预处理,需与 MNIST 数据集 28×28 像素达到一致。

2.1 数字预处理

预处理对于系统设计非常重要,经过预处理后的图像才能进行特征提取。因此,预处理的好坏直接影响着识别结果。因导入的点阵图形大小各异,需将点阵图形压缩成与 MNIST 数据集中的大小相一致,调整用户手写数字灰度图像形态为[28, 28];随后将用户手写数字图像转换为灰度图像。灰度图像像素值纯黑色的灰度级别为 0,纯白色的灰度级别为 255,介入两者之间还有许多灰度级别。由于 MNIST 数据集图片像素纯白色~纯黑色采取正规化的 0~1 区间,故用户手写数字灰度图像也须进一步

正规化到 0~1 区间。用户手写数字预处理流程如图 3 所示。

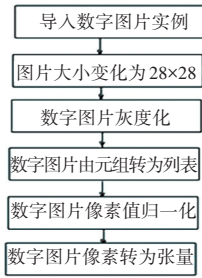


图 3 用户手写数字预处理流程

Fig. 3 Preprocessing process and code of handwritten digits

经处理后的数字图片格式与 MNIST 数据集的图片格式相同,则可将图片输入到网络中进行识别。

2.2 检测用户手写数字图像的神经网络

对用户手写数字图像进行检测的卷积神经网络,与 MNIST 数据集的训练和测试的卷积神经网络结构相同。模型参数为 MNIST 数据集的训练与测试得到并保存模型参数,主要是卷积神经网络的各层参数。只要恢复、使用之前保存的模型参数,用这些已知的参数构建卷积神经网络,通过其将用户手写数字图像从卷积神经网络输入,以完成到输出的映射,获得相应数字。

2.3 数字识别实现

TensorFlow 是一个开源的机器学习框架,在图形分类、音频处理、推荐系统和自然语言处理等场景下有着丰富的应用,并提供了多种编程语言的接口。本系统建立在 TensorFlow 平台上,分两部分完成:

2.3.1 建立模型参数

- (1) 导入 MNIST 数据集,并构建卷积神经网络学习模型;
- (2) 对学习模型进行训练和测试;
- (3) 评价学习模型是否为最佳模型参数,是则保存模型参数,否则修改学习模型,继续执行(2)、(3)步骤。

2.3.2 用户手写数字识别

- (1) 数字预处理,以使用户手写数字与 MNIST 数据集一致;
- (2) 使用保存的模型参数构建用户手写数字识别模型;
- (3) 用户手写数字输入手写数字识别模型,得到用户手写数字输出。

3 实验数据分析

实验分为无卷积神经网络和有卷积神经网络。实验以损失函数、训练数据准确率和测试数据准确

率为参考,见表 1。

表 1 训练损失值与数据准确率

Tab. 1 Training loss and data accuracy

| 类别 | 学习率 | 总的损失值 | 训练准确率 | 测试准确率 |
|---------|-------|---------------|-------|-------|
| 无卷积神经网络 | 0.002 | 0.395 069 083 | 0.929 | 0.901 |
| | 0.01 | 0.306 353 631 | 0.939 | 0.919 |
| | 0.025 | 0.277 722 566 | 0.970 | 0.922 |
| | 0.05 | 0.260 918 177 | 0.970 | 0.923 |
| | 0.1 | 0.245 686 171 | 0.970 | 0.925 |
| | 1.0 | 0.178 545 930 | 0.939 | 0.908 |
| 有卷积神经网络 | 1.5 | 0.173 448 963 | 0.929 | 0.919 |
| | 0.002 | 0.113 675 268 | 0.960 | 0.967 |
| | 0.01 | 0.010 198 373 | 0.980 | 0.981 |
| | 0.025 | 0.031 697 357 | 1.000 | 0.983 |
| | 0.05 | 0.003 000 124 | 1.000 | 0.981 |
| | 0.1 | 0.001 001 291 | 1.000 | 0.981 |
| | 0.75 | 2.296 184 935 | 0.111 | 0.097 |
| | 1.0 | 2.294 518 451 | 0.111 | 0.097 |
| | 1.5 | nan | 0.121 | 0.098 |

由表 1 中上半部数据可知,随着学习率的增加,损失函数值的减少,训练准确率和测试准确率均增加,但学习率增加过大。如,学习率为 1.0 和 1.5 时,导致数据准确率发生振荡。

由表 1 中下半部数据可知,当学习率小于 0.65 时,前 20 轮训练损失函数收敛很快,训练准确率和测试准确率很快达到 0.98 以上,最高为 1。学习率过大,如 0.75,损失函数找不到最小点,无法收敛,甚至出现未定义或不可表示的值。

4 结束语

通过实验比较了无卷积和卷积在神经网络中的作用,对学习训练参数、优化器也做了比较,取得了相应数据,学习率大小对无卷积神经网络影响为引起数据准确率发生振荡,对有卷积神经网络影响为引起损失函数找不到最小点,数据准确率大幅度下降。在学习率大小合适条件下有卷积神经网络的损失值和学习率均明显优于无卷积神经网络,实验结果挖掘了神经网络对图形内在的特性。

本研究是手写体数字识别系统开发的组成部分,其成果对整个系统开发工作奠定了很好的基础。

参考文献

- [1] 赵卫东. 机器学习[M]. 北京:人民邮电出版社.2018.
- [2] 李嘉璇. TensorFlow 技术解析与实践[M]. 北京:人民邮电出版社.2017.
- [3] 陈昊,郭海,刘大全,等. 基于 TensorFlow 的手写数字识别系统[J].2018, 183(3):108-110.
- [4] 杨颖,杨磊. 自由手写体数字表格自动识别系统[J]. 计算机工程与应用, 2002, 38(13):251-253.
- [5] 徐胤,袁浩巍,李智. 基于卷积神经网络和 TensorFlow 的手写数字识别研究[J]. 上海电气技术, 2018, 11(1):31-34,61.