

基于 TensorFlow 手写数字识别模型设计与实现

杜俊

(陇东学院,甘肃 庆阳 745000)

摘要:介绍了图像识别和机器学习基本原理和流程,深入分析了 softmax 回归算法,设计了手写数字识别模型,在 TensorFlow 上使用 softmax 回归算法实现了手写数字图像的识别,实验表明,使用 TensorFlow 能够快速实现图像识别、语音识别、自然语言处理等相关的机器学习和深度学习算法,可靠性和准确性较高。

关键词:图像识别;机器学习;TensorFlow;softmax 回归

中图分类号:TP391

图像识别是通过计算机对图像信息进行处理,理解图像的内容,以达到分类识别的效果,对图像进行精确识别,是实现目标跟踪、机器人导航、辅助驾驶、产品检测、车牌定位、物种识别等技术领域的关键环节。机器学习作为图像识别实现的主要方法,其实现方法、工具、性能要求仍难以满足当前的需要,应用时也存在一定的困难,在信息化、智能化的时代背景下,本文主要利用 Google 的第二代人工智能平台 TensorFlow,使用 softmax 回归算法实现手写数字图片的识别。

1 图像识别技术

图像识别是模式识别技术的一种。模式识别是通过计算机用数学技术方法来对观测到的物理对象自动进行处理、分类和识别。而图像识别特指对图像类模式的识别。图像识别过程主要包括训练和测试两个阶段。训练阶段对图像特征进行训练得到分类模型,测试阶段可以利用已训练模型得到识别结果。以上两个阶段主要由图像预处理、特征提取及分类三个环节组成。

在图像识别中,首先要对获取的图像进行加工处理,即预处理。预处理的目的是去除图像中的噪声,把它变成一幅清晰的点线图,以便于提取正确的图像特征。预处理主要对图像进行像素亮度变换、几何变换、图像滤波、图像复原。得到满足人的视觉和心理需要的改进形式。其次要对图像中的目标进行分析,提取图像特征。把预处理后的图像分割成不同目标物和背景不同的区域,提取正确代表不同目标物特点的特征参数(图像特征)。图像特征是图像区别于其他图像的本质特点或特性或是这些特点

和特性的集合。对图像进行特征提取后,将具备将一个目标或场景的不同影像进行匹配。最后要对图像中目标物进行识别和解释。设计分类器,建立分类模型,对图像中目标物进行识别和分类,理解不同目标物,分析其相互关系,从而知道和规划进一步的行动。

2 机器学习及其实现流程

机器学习能够通过先验学习获取经验,自动分析数据获取规律,重新组织已有知识、结构,并不断完善自身性能,让计算机在无事先明确编程的情况下做出正确的响应。机器学习研究的是如何让计算机模拟人类学习活动,是研究计算机获取新知识、新技能、识别现有知识、不断改善性能、实现自我完善的方法。

机器学习主要从已知数据学习获取正确经验或规律,然后利用该经验或规律推理其中未知的、潜在的概率分布等重要信息,提示数据样本中变量(特征)之间的关系。

机器学习主要有两个方向。一是监督学习,它主要解决分类和回归问题。分类的主要任务是将实例数据划分到合适的分类中,而回归主要用于预测数值型数据。目前有很多用于执行分类和回归的算法,例如 k-近邻算法、朴素贝叶斯算法、支持向量机算法、决策树算法、线性回归、局部加权线性回归、Ridge 回归、lasso 最小回归系数估计等,其中前四种用于分类,而后面的用于预测数值型数据。二是无监督学习。它既不对数据分类、也不预测数值数据。它是将数据集分成由类似对象组成的多个类的过程,即聚类;或者将寻找描述数据统计值的过程,即

密度估计。用于无监督学习的算法主要有 k-均值算法、最大期望算法、DBSCAN 算法、Parzen 窗设计等。在构建机器学习模型选择算法时,首先要明确机器学习算法的目的。如果想要预测,则选择监督学习算法,再考虑目标变量类型,如果是预测值是离散的,则选择分类算法,如果预测值是连续的,则选择回归算法。如果不想预测,则选择无监督学习,如果需要将数据划分为离散的组,则使用聚类算法。然后要分析数据集。由于机器学习的同类算法很多,为了缩小算法的选择范围,需要充分了解数据,对实际数据了解得越充分,才能选出最切近该问题算法。

机器学习应用程序开发过程中,首先需要收集数据。如制作网络爬虫从网站爬取数据、传感节点发送的实测数据等,也可以使用公开可用的数据源;接着要格式化数据源,确保数据格式符合要求。如采用 Python 的 List 格式,这种格式可以融合算法和数据源,方便匹配操作;然后将格式化的数据输入到算法,即训练算法,从中抽取知识或信息;算法训练好后,测试算法的工作效果,检验算法的成功率,如不满意,可修改前面步骤;最后将机器程序转换为应用程序,执行实际任务。

3 手写数字识别模型设计与实现

3.1 Tensorflow

2015 年 9 月,谷歌发布了其第二代人工智能系统—Tensorflow,一个开源的机器学习软件资源库。它既是一个实现机器学习算法的接口,也是执行机器学习算法的框架。它能够支持多种开发语言,可在异构的系统上进行移植,支持异构分布式系统部署。TensorFlow 既支持深度学习算法,也实现了很多其他算法,例如线性回归、逻辑回归、随机森林等。

3.2 数据采集

本模型主要用 MNIST 数据集进行训练、验证和测试。MNIST 数据集在 TensorFlow 官网下载。MNIST 数据集由训练集、验证集和测试集三部分构成,其中训练集有 55000 个样本,验证集有 5000 个样本,测试集有 10000 个样本。每一个样本都有它对应的标签信息(label),用来描述样本表示的数字。训练集用来训练模型,然后在验证集上检验效果,决定何时完成训练,最后在测试集上测试模型的准确率。

MNIST 样本描述的是一个 28×28 像素的灰度

图片,可用一个长度 784 的数组来表示一张图片。图片样本的这种方式丢弃了图片二位结构的信息,但由于分类任务比较简单,这种数据简化,可以选择比较简单的分类算法实现模型。

训练样本的数据特征是一个 55000×784 的 Tensor。第一维是图片编号,第二维是对应图片的像素点,其值表示某个像素点的灰度值。训练样本的标签是一个 55000×10 的 Tensor,用来描述每个样本图片代表的数字,每个样本的 label 是一个 10 维向量,只有 1 个值为 1,其余为 0,那个数字,其样本图片的 label 向量对应位置为 1,其余为 0。例如数字 0 的样本 label 就是[1,0,0,0,0,0,0,0,0,0]。

由于 MNIST 数据集所有样本描述的数字图片值为 0~9,所有样本就分为 10 类,即 0,1,2,3,4,5,6,7,8,9,手写数字图像识别问题就是计算测试集中第 i 样本是数字 j(j=0,1,⋯,9)的概率问题,基本以上特点,可采用 softmax 回归算法实现。

3.3 softmax 回归算法

Softmax 回归是 Logistic 回归的一般化,将只能解决二分类问题的 Logistic 回归扩展至能够解决多分类问题。假设 softmax 回归模型的样本来自 k 个类,共有 m 个,则由这些样本组成的训练集为

$\{(x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \dots, (x^{(m)}, y^{(m)})\}$ 。其中 $x^{(i)} \in R^{(n+1)}$, 标

签: $y^{(i)} \in \{1, 2, \dots, k\}$ 。对于给定的输入 x, 假设函数针对每一个类别 j 估算出概率值 $p(y=j | x)$, 用来估计 x 的每一种分类结果出现的概率。因此,假设函数将输出一个 k 维向量(向量元素和为 1)来表示这个 k 个估计的概率。假设函数 $h_\theta(x)$ 形式如下:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)}=1 | x^{(i)}; \theta_1) \\ p(y^{(i)}=2 | x^{(i)}; \theta_2) \\ \vdots \\ p(y^{(i)}=k | x^{(i)}; \theta_k) \end{bmatrix} = \frac{1}{\sum_{j=1}^k e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

其中 $\theta_1, \theta_2, \dots, \theta_k \in R^{n+1}$ 是模型参数, $p(y^{(i)}=j | x^{(i)}; \theta_j)$ 表示样本 $x^{(i)}$ 属于第 j 类的概率。

该模型的代价函数 $J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)}=j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right]$ 。

其中 $1\{\cdot\}$ 是显性函数,其取值规则为 $1\{\text{值为真表达式}\}=1, 1\{\text{值为假的表达式}\}=0$ 。

此模型具有参数冗余的问题,可以给代价函数加入权重衰减,代价函数变为:

$$J(\theta) = -\frac{1}{m} \left[\sum_{i=1}^m \sum_{j=1}^k 1\{y^{(i)} = j\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^k e^{\theta_l^T x^{(i)}}} \right] + \frac{\lambda}{2} \sum_{i=1}^m \sum_{j=0}^n \theta_{ij}^2$$

有了这个权重衰减项,代价函数就变成了严格的凸函数,保证能够得到唯一解,有效解决了参数冗余问题。

对 $J(\theta)$ 求偏导数:

$$\nabla_{\theta_j} J(\theta) = -\frac{1}{m} \sum_{i=1}^m \left[x^{(i)} \left(1\{y^{(i)} = j\} - p(y^{(i)} = j | x^{(i)}; \theta) \right) \right] + \lambda \theta_j$$

通过梯度下降法的计算参数 $\theta_1, \theta_2, \dots, \theta_k$, 得到训练好的 softmax 回归模型。然后将测试集中的样本数据代入假设函数计算其所属类别的概率,选择概率最大的类别作为预测结果。

3.4 TensorFlow 实现手写数字识别

3.4.1 实现模型

1) 首先载入 TensorFlow 库, 创建一个新的 InteractiveSession, 然后创建一个 Placeholder, 即输入数据的地方, Placeholder 的第一个参数是数据类型, 第二个参数 [None, 784] 代表 tensor 的 shape, 也就是数据的尺寸, None 代表不限条数的输入, 784 代表每条输入时一个 784 维的向量。

```
Import tensorflow as tf
```

```
Sess=tf.InteractiveSession()
```

```
X=tf.placeholder(tf.float32,[None,784])
```

2) 创建 softmax 回归模型的 weights 和 biases 的 Variable 对象。

```
W = tf.Variable(tf.zeros([784, 10]))
```

```
b = tf.Variable(tf.zeros([10]))
```

3) 实现 softmax 回归算法

```
y = tf.nn.softmax(tf.matmul(x, W) + b)
```

3.4.2 训练模型

1) 定义、计算损失函数。损失函数描述模型对问题的分类精度。损失越小, 代表模型的分类结果与真实值的偏差越小, 模型越精确。初始时将模型参数全部设置为零, 随着训练 loss 值不断减小, 直到达到一个全局最优或局部最优解。对于多分类问题, 通常使用 cross-entropy 作为损失函数。在 TensorFlow 中定义、计算 corss-entropy 如下:

```
y_ = tf.placeholder(tf.float32,[None,10])
```

```
cross_entropy=tf.reduce_mean(-tf.reduce_sum(y_
```

```
*tf.log(y), reduction_indices=[1]))
```

2) 定义优化算法, 采用随机梯度下降。

```
rain_step = tf.train.GradientDescentOptimizer
```

```
(0.5).minimize(cross_entropy)
```

3) 迭代执行训练操作。每次随机抽取 100 条样本构成一个 mini-batch, 并 feed 给 placeholder, 然后调用 train_step 对样本训练。

```
for _ in range(1000):
```

```
batch_xs, batch_ys = mnist.train.next_batch
```

```
(100)
```

```
sess.run (train_step, feed_dict={x: batch_xs,  
y_: batch_ys})
```

3.4.3 评估模型

训练完成后, 需要测试模型的准确率。tf.argmax 是从 tensor 中寻找最大值得序号, tf.argmax(y, 1) 就是求各个预测的数字总的概率最大的那个, tf.argmax(y_, 1) 则是寻找样本的真是数字类别。tf.equal 方法则用来判断预测数字类别是否就是正确类别, 最后返回计算分类是否正确操作。

```
correct_prediction = tf.equal(tf.argmax(y,1),
```

```
tf.argmax(y_,1))
```

```
accuracy = tf.reduce_mean(tf.cast
```

```
(correct_prediction, tf.float32))
```

4 结论

目前, 图像识别、语音识别、自然语言处理是人工智能应用的重点领域。这些应用主要通过机器学习和深度学习算法实现, 应用时在构建模型、优化参数、模型分析方面存在一定的困难。TensorFlow 是谷歌近年推出的开源人工智能系统, 具有编程简单、优化算法集成度高、使用灵活的特性, 利用 TensorFlow, 将大幅提高建模、编程、分析的效率。本文在 TensorFlow 平台上, 使用 softmax 回归算法构建手写数字图像识别模型, 实现手写数字图像的识别。实验表明该模型识别精度较高, 该模型的实现对在 TensorFlow 上构建更复杂机器学习应用相关模型具有较大的参考意义。

参考文献:

- [1] Martin Abadi, A Agarwal, P Barham, E Brevdo, Z Chen, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015, software available from tensorflow.org [Online]. Available: <http://tensorflow.org/>. (下转第6页)

3) 增强社会主义意识形态网络阵地的吸引力和知名度。社会主义网络意识形态建设需要打造坚如磐石的网络阵地。网络阵地建设中首先应该保证网站的专业性与严谨性,这是取得访问者信任的基础^[4]。网民面对专业的网站,才会有可能驻足浏览,而不太专业的网站则会受到网民的无情拒绝。相较内容丰富且制作技术精良的专业网站,不太专业的网站甚至被认为是病毒网站。要确保社会主义意识形态网络阵地的专业性,必须要有坚信、真懂社会主义意识形态的专业人士参与网络阵地的建设中来,从社会主义经典理论和社会主义伟大实践两个方面出发为网络阵地撰稿、组稿和审稿,以增加网络阵地的信任度。此外,除了重视网络阵地的内容建设外,还需加强网络阵地的形式建设。要采取公众喜闻乐见的形式展现社会主义主流意识形态,增加网站的趣味性与交互性。社会主义意识形态网络阵地应该紧扣当今社会热点与关键点更新网站栏目与内容,设置多样化、多层次的阅读栏目,吸引不同文化水平的网民参与网站互动。

4) 创新网络意识形态传播体制,增强网络意识形态建设的时效性。创新网络意识形态的传播体制,增强网络意识形态建设的时效性是必要的。在遵守党管媒体的原则下,我们要充分利用出现的互联网新技术,建立和完善网络意识形态传播的新机制,针对热点与重大事件建立快速反应机制。我们要充分利用网页、论坛、聊天室、群聊等网络工具加强党的意识形态交流,促进意识形态传播的针对性与有效性。除此之外,还需要引入专家评价机制,通过遴选兼具丰富网络知识与深厚社会主义意识形态思想的专家组成专家委员会,实时、动态地对网络意识形态的状况进行监控与指导,以提高社会主

义主流意识形态的传播效果。再次,切实加强和改进网络思想政治工作。在网络空间建设社会主义意识形态应当加强和改进网络思想政治工作。网络时代的思想政治工作者应该学习现代信息技术,掌握网络通信的特点与动态,创新内容,扩大思想政治工作的成效。例如,可以通过举办各种网络文化活动,开展理想的信仰教育、政策教育、时事教育,了解网民的困惑,交流认识,澄清网民关注的热点与难点问题。

4 结语

全面建设小康社会,建设幸福美好新中国离不开网络意识形态建设。在信息时代,意识形态的冲突已经对全面建设小康社会产生了不容忽视的影响,我们要充分认识网络意识形态建设在全面建设小康社会中的重要作用。从技术的角度来看,网络并不存在好与坏的价值判断,但网络的运作总离不开人的参与,网络天然地将现实社会中的各种复杂关系通过数字的方式进行了另一种表征。网络对于意识形态建设的影响就像一个硬币的两面,既有好的一面也有不好的一面,既有机遇又有挑战,我们应该迎难而上,建设好社会主义网络意识形态。

参考文献:

- [1] 魏晓文,邵芳强.论网络背景下的高校意识形态安全建设[J].思想教育研究,2014(06):29-33.
- [2] 张静,周三胜.论网络传播条件下党的意识形态建设[J].毛泽东邓小平理论研究,2005(06):61-66.
- [3] 张骥,方晓强.论网络文化对我国社会主义意识形态建设的影响[J].求实,2009(02):40-43.
- [4] 杨静娴,刘路路.马克思主义意识形态网络阵地建设中存在的问题及对策[J].理论导刊,2014(08):61-64.

(上接第16页)

- [2] Y. LeCun, C. Cortes, and C. J. Burges. The MNIST database of handwritten digits, 1998. yann.lecun.com/exdb/mnist/.
- [3] 吴忠,朱国龙,黄葛峰,等.基于图像识别技术的手写数字识别方法[J].计算机技术与发展,2011,21(12):48-51.
- [4] 龚声蓉.数字图像处理与分析[M].北京:清华大学出版社,2014.
- [5] 闫友彪,陈元琰.机器学习的主要策略综述[J].计算机应用研究,2004,21(7):4-10.
- [6] 章敏敏,徐和平,王晓洁,等.谷歌 TensorFlow 机器学习框架及应用[J].微型机与应用,2017,36(10):58-60.
- [7] I.H. Witten, E. Frank. Data Mining: Practical Machine Learning Tools and Techniques[M].机械工业出版社,2012,31(1):76-77.
- [8] 周志华,王钰.机器学习及其应用[M].北京:清华大学出版社,2007.