

文章编号: 2095-2163(2021)02-0179-05

中图分类号: TP391.41

文献标志码: A

基于 CNN 池化和进化策略的一般神经网络图像分类研究

高 滔

(上海理工大学 管理学院, 上海 200093)

摘要: 网络的爆炸式发展产生了海量的图像, 图像标签的错误和缺失比较常见, 图像分类研究很有必要。CNN 池化能够提取到输入矩阵的重要特征, 降低数据的维度。进化策略是模仿生物“优胜劣汰”进化方式的一种启发式算法, 能快速找到问题的解。本文基于 CNN 池化提取一组有正确标签的图像的特征, 搭建层数为 3 的神经网络, 进化策略优化初始权重, 通过训练集训练分类模型, 通过测试集来验证模型的优劣, 并使最终的模型实现对未知类别图像的高效分类。实例验证阶段收集 10 类 100 张犬类图片, 按照各研发步骤进行实验, 算法结果验证了进化策略优化权重的必要及神经网络模型的高效。

关键词: CNN 池化; 进化策略; 神经网络; 图像分类

Research on general neural network image classification based on CNN pooling and evolutionary strategy

GAO Tao

(Business School, University of Shanghai for Science and Technology, Shanghai 200093, China)

【Abstract】 The explosive development of the network has produced a large number of images, and the errors and missing of image labels are more common, and the research on image classification is necessary. CNN pooling can extract important features of the input matrix and reduce the dimensionality of the data. Evolutionary strategy is a heuristic algorithm that imitates the evolutionary method of "survival of the fittest" of biology, which can quickly find the solution of the problem. Based on CNN pooling, the paper extracts the features of a set of images with correct labels, builds a neural network with three layers, optimizes the initial weights by evolutionary strategy, trains the classification model through the training set, and verifies the pros and cons of the model through the test set. The final model realizes the efficient classification of unknown category images. In the instance verification stage, 100 dog pictures of 10 categories are collected, and the experiment is carried out according to the above steps. The algorithm results verify the necessity of the evolution strategy to optimize the weight and the efficiency of the neural network model.

【Key words】 CNN pooling; evolutionary strategy; neural network; image classification

0 引言

随着互联网技术的飞速发展, 在对互联网的使用中也随即产生了海量的图像。目前网页上的很多图像都没有或没有正确标签, 并在缺乏先验知识的情况下, 对图像的分类进行识别也仍具有一定难度。很多时候各关键词也不能链接到正确的图像, 即会带来一定误导, 所以良好的图像分类意义重大。

人工神经网络 (Artificial Neural Network, ANN) 是模拟生物神经网络, 由若干个神经网络组成的动态系统。其中, 里程碑式的成就即是由 Rosenblatt 提出的单层感知机模型, Rumelhart 等人^[1]则提出了误差训练传播算法的多层前馈反向传播网络 BP 网络, Hinton 等人^[2]又在《Science》发文剖析了多隐藏层的优点, 至此深度学习的提出就使得神经网络的研究逐渐成为热点。在上世纪末,

已有人开始从事卷积神经网络 (Convolutional Neural Network, CNN) 方面的相关研究, 也取得了一些成果, 但很难拓展, 直到 2012 年大规模视觉识别挑战竞赛 CNN 取得最好的效果, 由此才吸引大量关注^[3]。进化策略是由德国 Rechenberg 和 Schwefe 模拟生物进化的方式提出的算法, 与遗传算法不同的是其存在外形参数, 并认为基因改变无法改变结果的分布^[4]。

现有的分类技术大多属于监督学习, 需要大量标签数据训练模型^[5]。零样本分类是一种跨模态学习, 当前的热门研究领域之一, 其在应用初期的图像分类任务现已扩展到了其他方面^[6-7]。卷积神经网络在图像分类的应用是当前的主流, 取得了众多可观成果^[8-10], 但在一些领域也受到一些挑战, ResNet-50 网络中的 Softmax 分类器不能够很好地区分提取到的特征^[11-12]。目前, 卷积神经网络对图

作者简介: 高 滔(1994-), 男, 硕士研究生, 主要研究方向: 车间调度、智能算法。

收稿日期: 2020-10-16

哈尔滨工业大学主办 ◆ 科技创新与应用

像分类的研究,经常用到的图像去噪方法,也就是本文选用的 NL-means 算法^[13], Softmax-Loss 在单类别分类问题中的应用要更多些^[14-15]。

综上所述,神经网络在图像分类的研究较多,研究成果相当丰富。但大多数都是在特定情况下集中在分类模型的选择和参数的优化上,重点就是算法层面的改进。基于此,本文应用 CNN 的池化提取特征,进化策略优化初始权重,搭建相关神经网络模型并进行训练,最终在具体的实例中验证了模型的效率。

1 研究方法

1.1 池化

池化类似于过滤,用过滤得到输出代替输入(本文中的输入、输出皆是二维矩阵)。相关参数有输入 x 、核 $kernel$ 、输出 y 、Padding 加零、 X 宽高分别是 WX 、 WH ,核的参数有宽 W 、高 H 、步长 $stride$,其宽高分别是 S_width 、 S_height 。池化一般有 3 种: Max Pooling、Min Pooling、Mean Pooling,分别表示最大、最小、均值池化。池化的流程步骤可表述为:

对输入 X 进行加零(Padding = 1, X 外部加一圈零),从左到右,从上到下依次用核去扫描 X ,步长 $stride$ 控制每次移动的距离, $stride$ 有 2 个参数,分别控制右移和下移的幅度,核覆盖超过 X 边界的部分补零,依次对核覆盖的区域取最大值、最小值或均值组成输出 y 。池化的示例如图 1 所示。由图 1 可知,核长宽都是 2,步长的 2 个参数都是 2。

例:Max Pooling



图 1 池化示例

Fig. 1 Example of pooling

这里,研究推得的数学公式可写为:

$$y = f(x, w, h, padding, stride), \quad (1)$$

当 $padding = 0$ 时,输出 y 的宽 $width$,高 $height$ 分别为:

$$width = [(WX - W) / S_width] + 1, \quad (2)$$

$$height = [(HX - H) / S_height] + 1. \quad (3)$$

其中, f 表示池化类型, $[]$ 表示向下取整。

1.2 进化策略

进化策略模拟生物进化中的一种启发式算法,与遗传算法求解过程相似,也有不同的地方。流程步骤具体如下:

(1) 初始解:在解空间初始 n 个 m 维解 (x_1, x_2, \dots, x_n) ,并随机生成每个解。每个解对应 m 维变异强度 (s_1, s_2, \dots, s_m) 。

(2) 交叉:与遗传算法相似,选取 2 个父代 x_i, x_j ,交换某些位置的基因,产生子代 x_k ,不同的是, s_i, s_j 也要进行相应操作,产生 s_k 。

(3) 变异:对于 x_k ,把 x_k m 维度上的每一个元素看作均值 u, s_k 对应的值看作标准差 σ ,正态产生新接近 x_k 的新解,更新 x_k, s_k 按特定情况逐渐变小。

(4) 选择:将子代加入父代,计算所有解的目标函数值,根据目标函数对所有解进行排序,淘汰劣解,留下优解。

不断重复上述步骤,直到找到满意的解。

$(\mu + \lambda)$ -ES 表示父代 DNA 个数为 μ ,每次产生 λ 个子代,合并后种群里 DNA 个数是 $(\mu + \lambda)$,每次淘汰 λ 个个体。

1.3 神经网络

神经网络也对应着输入 X ,输出 Y ,不同的是其引入权重 W 、偏置 b 和激活函数 $F(x)$,激活函数有 $\text{sigmoid}(x), \text{softmax}(x), \text{relu}(x)$ 等等。

按学习方式,神经网络可分为监督式学习和非监督式学习,两者间的区别是有无标签(目标值)。对于分类问题,常用的方法是监督式学习。其大致步骤就是:初始化 W, b ,计算模型输出 Y ,与标签 t 进行比较,通过彼此间的差异来调整 W 和 b ,并引入学习率 LR 表示调整幅度的大小,最终达到训练模型的效果。通常,损失函数 $loss(x)$ 可表示两值间差异。研究中,常见的损失函数有均方差,交叉熵,常用的算法是梯度反向传播算法。下面就本文用到的函数及模型给出对应的数学表示:

$$\text{sigmoid}(x) = \frac{1}{1 + e^{-x}}, \quad (4)$$

$$\text{softmax}(x) = \begin{bmatrix} \frac{e^{x_1}}{\sum_{c=1}^c e^{x_c}} \\ \frac{e^{x_2}}{\sum_{c=1}^c e^{x_c}} \\ \dots \\ \frac{e^{x_c}}{\sum_{c=1}^c e^{x_c}} \end{bmatrix}, \quad (5)$$

$$Y = \begin{cases} W * X + b, & F(x) = None, \\ F(W * X + b), & else. \end{cases} \quad (6)$$

$$loss(x) = \sum_{c=1}^c t * \ln Y, \quad (7)$$

$$W = W - LR * \frac{dloss(x)}{dW}, \quad (8)$$

$$b = b - LR * \frac{dloss(x)}{db}. \quad (9)$$

其中,式(4)和式(5)是本文用到的2个激活函数;式(6)是Y的计算方式;式(7)是交叉熵损失函数;式(8)和式(9)分别是梯度传递下的权重和偏置

更新。

2 实验及结果分析

2.1 数据收集及处理

本文收集了10种犬的图片,分别是:藏獒、哈士奇、柴犬、贵宾犬、杜宾犬、吉娃娃、边境牧羊犬、萨摩耶犬,英国斗牛犬,金毛寻回犬。每种10张,共100张图片。每张图都是jpg格式,并对每张图的像素进行调整,均调整为121×121。对部分图片处理后的缩影见图2。



图2 部分处理图片缩影

Fig. 2 Partial processing of image miniaturization

对于数据处理,研究中拟依次展开的步骤详情见如下:

(1)通过python的图像处理模块把每张图像转为灰度图,并通过自定义函数把每张图读为二维矩阵,因为矩阵上的值在0~255之间,每个矩阵都除255,做归一化处理。返回矩阵的行和列,对应图像的width,height,分别是121,121。现有100个121×121的矩阵。

(2)对这100个矩阵做maxpool池化处理,核的W=H=3,步长,S_width=3,S_height=3,padding=0,由式(2),式(3)知处理后的矩阵行列C=R=[(121-3)/3]+1=40,变为100个40×40的矩阵。其池化的工具是tensorflow模块里面的相关函数。

(3)把矩阵变为100个1600×1的矩阵并垂直合并,变换为100×1600×1的矩阵,降维为100×1600,其矩阵的每一行就表示一张图片。对矩阵的每一行的标签进行编码,采用独热编码(One-Hot

Encoding)。编码完成后与前一个矩阵合并,矩阵变为100×1610,保存在Excel中。

(4)复制一份数据,与之前数据合并,变为200×1610,并随机打乱矩阵的行。读取前面的160行作为训练集,用作模型的训练,读取最后40行用作测试集,用作模型的测试。

处理后的部分数据缩影见图3。

id	姓名	性别	年龄	品种	身高	体重	毛色	眼睛	耳朵	尾巴	脚掌	爪子	牙齿	舌头	鼻子	嗅觉	听力	视力	记忆	智力	性格	健康状况	训练情况	其他
0001	藏獒	♂	3	藏獒	120	60	黑	黑	立	卷	黑	黑	黑	黑	黑	灵敏	好	好	好	好	温顺	健康	未训练	
0002	哈士奇	♀	2	哈士奇	60	25	白	蓝	半立	垂	白	白	白	白	白	灵敏	好	好	好	好	活泼	健康	未训练	
0003	柴犬	♂	4	柴犬	60	25	黄	黄	立	垂	黄	黄	黄	黄	黄	灵敏	好	好	好	好	忠诚	健康	未训练	
0004	贵宾犬	♀	1	贵宾犬	40	15	白	黑	立	垂	白	白	白	白	白	灵敏	好	好	好	好	优雅	健康	未训练	
0005	杜宾犬	♂	2	杜宾犬	70	30	黑	黑	立	垂	黑	黑	黑	黑	黑	灵敏	好	好	好	好	勇敢	健康	未训练	
0006	吉娃娃	♀	1	吉娃娃	20	10	白	黑	立	垂	白	白	白	白	白	灵敏	好	好	好	好	可爱	健康	未训练	
0007	边境牧羊犬	♂	3	边境牧羊犬	50	20	白	黑	立	垂	白	白	白	白	白	灵敏	好	好	好	好	聪明	健康	未训练	
0008	萨摩耶犬	♀	2	萨摩耶犬	60	25	白	黑	立	垂	白	白	白	白	白	灵敏	好	好	好	好	友善	健康	未训练	
0009	英国斗牛犬	♂	4	英国斗牛犬	80	40	黑	黑	立	垂	黑	黑	黑	黑	黑	灵敏	好	好	好	好	强壮	健康	未训练	
0010	金毛寻回犬	♀	3	金毛寻回犬	70	35	黄	黄	立	垂	黄	黄	黄	黄	黄	灵敏	好	好	好	好	友好	健康	未训练	

图3 部分处理数据缩影

Fig. 3 Partial processing of data miniaturization

2.2 模型搭建

本文搭建的是3层神经网络,第一层是输出层,第二层是隐藏层,激活函数是 *sigmoid*,第三层输出层,激活函数是 *softmax*。隐藏层的神经元数采用 Masters(1994),即:

$$NHN \approx (NINP * NOU)^{1/2}. \quad (10)$$

输出神经元 1 600,输出神经元 10(10类),隐藏神经元数 $NHN \approx \sqrt{1\ 600 \times 10} \approx 126$ 。

隐藏层和输出层的权重矩阵的值都从标准正态分布中生成,初始 b 都为 0.1。

其中,隐藏层矩阵是 1 600×126,输出层矩阵是 126×10。损失函数采用交叉熵函数。学习率设为 0.1。

2.3 进化策略生成初始权重

初始权重相当于寻找满意解的起点,好的初始权重能使算法收敛更快,运用进化策略对初始权重进行寻优,算法步骤可简化为2步:

(1) 随机初始化多个输出层矩阵权重当作 DNA,也就是问题的多个解,随机初始化多个变异强度与之对应。

(2) 计算多个解下的输出,因为适应度函数(目标函数)是最值函数,把交叉熵当作适应度函数。对初始矩阵进行进化策略迭代,寻找最优的权重矩阵。

初始的 500 个权重矩阵和对应的变异强度矩阵都是从标准正态函数中生成。本文初始化 500 个 DNA,即 500 个权重矩阵,每次产生 200 个子代,种群变为 700 个 DNA,每次淘汰交叉熵大的 200 个 DNA,经过多次迭代后种群还是 500 个 DNA,最终输出交叉熵最小的权重矩阵。

2.4 算法测试

保持其他神经网络的各参数不变,参见 2.2 节的模型搭建,迭代次数都设为 1 000 次,分别输出不加入进化策略(随机输出层权重矩阵)和加入进化策略(进化策略找到较好的初始矩阵)的结果。该结果表示每次通过模型迭代去计算测试集的正确率。结果正确的依据就是每次测试集的输出与测试标签在对应位置处的误差在较小的范围内,否则认为不正确。研究得到的实验结果如图 4~图 6 所示。3 个结果都是多次实验得到较为平均的结果。其中,图 6 是学习率 $LR = 0.3$ 的未加入进化策略的仿真结果。

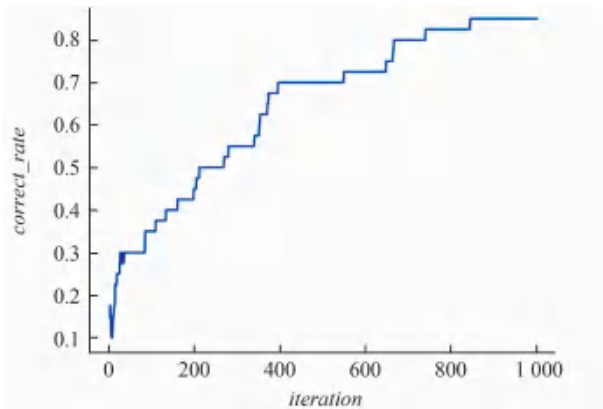


图 4 未考虑进化策略的图片正确率

Fig. 4 Image accuracy without considering evolutionary strategy

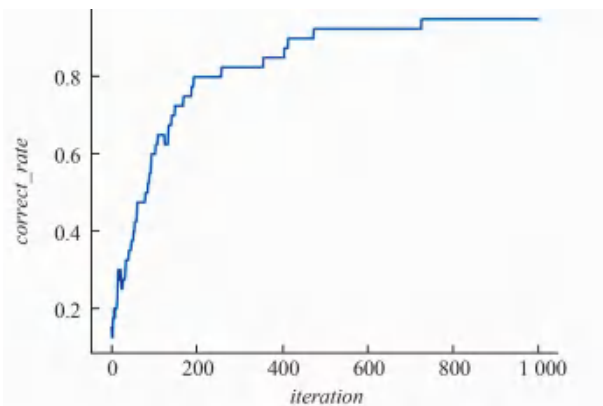


图 5 考虑进化策略的图片正确率

Fig. 5 Image accuracy considering evolutionary strategy

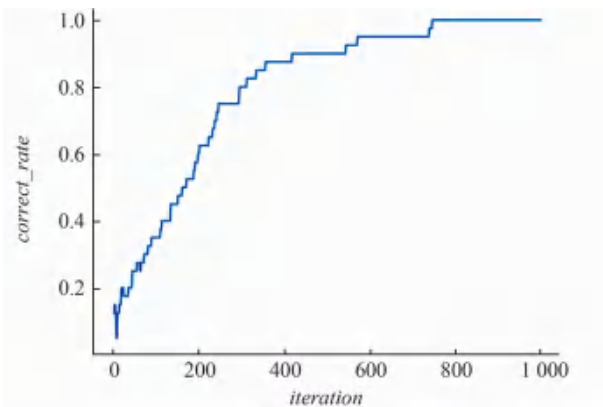


图 6 学习率为 0.3 且考虑进化策略的图片正确率

Fig. 6 The learning rate is 0.3 and the image accuracy is considered with evolutionary strategy

由图 4~图 6 可以看出,考虑 CNN 池化提取图像特征正确率都较高。结果中,加入进化策略的正确率 95%,未加入进化策略正确率为 85%,学习率调整为 0.3 时正确率在迭代次数 800 次左右就达到 100%。

(下转第 186 页)