

A Research on Deep Learning Model for Face Emotion Recognition Based on Swish Activation Function

Lingjiao Wang, Qian Li, Hua Guo

College of Information Engineering, Xiangtan University, Xiangtan Hunan
Email: xtu_wlj@126.com, 645513992@qq.com

Received: Jun. 11st, 2019; accepted: Jun. 27th, 2019; published: Jul. 3rd, 2019

Abstract

In recent years, deep learning model has been developed rapidly. As one of the methods, deep convolution neural network has been widely used in computer vision. There are many factors affecting the performance of deep learning model, among which the selection of activation function and the structure of neural network have important impact on the performance of deep learning model. This paper analyses the advantages and disadvantages of the traditional activation function and the new Swish activation function, introduces Swish function into the deep learning model of facial emotion, proposes an improved back propagation algorithm, and uses multi-layer small-size convolution module instead of large-size convolution module in the convolution neural network to extract refinement features, and constructs a new deep learning model of facial emotion recognition, Swish-FER-CNNs. The experimental results show that the recognition accuracy of deep learning model based on Swish activation function is higher than that of activation functions such as ReLU, L-ReLU and P-ReLU. With the improved network structure, the recognition accuracy of the deep learning model of Swish-FER-CNNs constructed in the paper is improved by 4.02% compared with the existing model.

Keywords

Activation Function, Back Propagation, Convolutional Neural Network, Deep Learning, Computer Vision

基于Swish激活函数的人脸情绪识别的深度学习模型研究

王灵娇, 李 乾, 郭 华

湘潭大学, 信息工程学院, 湖南 湘潭
Email: xtu_wlj@126.com, 645513992@qq.com

收稿日期：2019年6月11日；录用日期：2019年6月27日；发布日期：2019年7月3日

摘要

近年来,深度学习模型得到飞速发展,深度卷积神经网络作为其中一种方法在计算机视觉中得到广泛应用。影响深度学习模型性能的因素众多,其中激活函数的选取和神经网络的结构对深度学习模型的性能有着重要的影响。本文分析了传统激活函数与新型Swish激活函数的优缺点,将Swish函数引入人脸情绪深度学习模型,提出了一种改进的反向传播算法,并在卷积神经网络中使用多层小尺寸卷积模块替代大尺寸卷积模块,提取细化特征,构建了一种新型的人脸情绪识别深度学习模型Swish-FER-CNNs。实验结果表明,相比于ReLU、L-ReLU、P-ReLU等激活函数,基于Swish激活函数的深度学习模型的识别准确率更高。结合改进的网络结构,本文构建的深度学习模型Swish-FER-CNNs相对于现存模型,识别准确率提高了4.02%。

关键词

激活函数, 反向传播, 卷积神经网络, 深度学习, 计算机视觉

Copyright © 2019 by author(s) and Hans Publishers Inc.

This work is licensed under the Creative Commons Attribution International License (CC BY).

<http://creativecommons.org/licenses/by/4.0/>



Open Access

1. 引言

本人类通过眼睛感知光信息洞悉世界,情绪是人类沟通的一个重要组成部分,影响人类的交流。准确分析人脸情绪,对深入理解人类行为至关重要。在文献[1][2]中, Giannopoulos 等人已经提出了使用卷积神经网络的方法分析人脸情绪。

卷积神经网络系统主要由卷积层、池化层、激活函数、全连接层、分类函数等多部分组成[3][4],组成的方式不同导致网络模型的性能差异显著。经典的 AlexNet 深度学习模型使用 5 层卷积层、3 层全连接层、ReLU 激活函数等联接构成的网络,在 ImageNet 分类上准确率提高显著[5][6][7][8][9],但参数较多、计算负载大,在人脸情绪识别数据集上有过拟合的问题。Jeon [10]在 AlexNet 的基础上进行了结构精简,成功地将其引入人脸情绪识别领域,但网络层数较浅,模型拟性能较弱,识别准确率不高。

卷积神经网络自学习机制的关键是反向传播算法,系统产生的误差通过网络结构向后传播,调节系统自身参数[11]。激活函数[12][13][14]作为反向传播算法中重要的组成部分,表示神经元潜在动作的抽象映射,影响系统性能。ReLU 激活函数[5]是线性整流函数(Rectified Linear Unit),又称修正线性单元。加强了模型的非线性并且可以提高模型的收敛速度。但当 $x < 0$ 时,ReLU 的输出皆为 0,导致模型无法迭代,出现神经元死亡和均值偏移问题。之后, Maas 提出的 L-ReLU 激活函数,即带泄露整流函数(Leaky ReLU),能在一定程度上解决了这些问题,但在 $x < 0$ 时为线性函数,系统鲁棒性较差[15][16][17]。文献[18][19]中提出的 P-ReLU 被称为参数线性整流函数(Parametric ReLU),与 L-ReLU 相似,不同之处在于当 $x < 0$ 时,斜率为系统学习所得,函数性能提高,但仍未解决对输入噪声鲁棒性较差的问题。

为更好地拟合数据与系统模型,本文结合 Swish 激活函数,改进反向传播算法,解决了鲁棒性较差的问题,同时改善了神经元死亡和均值偏移问题。针对 AlexNet 和 Jeon 的深度模型(简称 Jeon's)网络层数较浅、参数较多等不足,加深了神经网络层数并使用更小的卷积核,便于感知细化特征,提出了 Swish-FER-CNNs 情绪识别模型,提高了识别准确率。

2. 基于卷积神经网络的情绪识别模型

卷积神经网络使用卷积层提取图像特征，以池化层简化参数，提高计算效率。采用多层卷积和池化级联能进一步感知和提取图像特征，获取高维特征并使用全连接层将学习到的隐式特征映射到样本标记空间，最后通过 Softmax 分类器进行分类，得到分类标签，具体模型如图 1 所示。

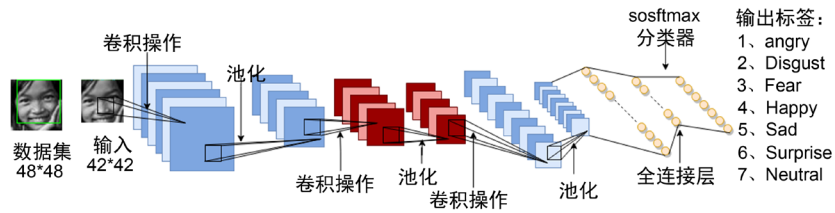


Figure 1. Principle diagram of emotion recognition model
图 1. 情绪识别模型原理图

2.1. 反向传播算法

传统反向传播算法是调整网络模型的参数更迭的依据。已知输入 x 通过前馈传播得到 \hat{y} ，产生的误差为代价函数 J_θ 。来自代价函数 J_θ 的信息通过模型向底层传播，用于计算梯度。反向传播算法是一种计算微积分中链式法则的算法， f 和 t 分别是两个实数集合间的映射函数，令 f 为激活函数， $y = t(x), z = f(t(x)) = f(y)$ ，则链式法则为：

$$\frac{dz}{dx} = \frac{dz}{dy} \frac{dy}{dx} \quad (1)$$

上式可反映上层网络输出 z ，受下层网络输出 x 的影响而变化的数学关系。又有 $\frac{dz}{dy} = f'$ ，故 f 的选取对于深度学习网络的迭代性能有较大影响。下文通过对 Swish 激活函数进行性能分析，提出了一种改进的反向传播算法。

2.1.1. 激活函数分析

传统的 Sigmoid 激活函数，其函数定义如下：

$$f(x) = \frac{1}{1 + e^{-x}} \quad (2)$$

其函数曲线如下图 2 所示：

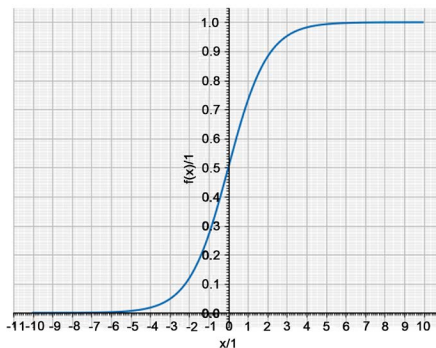


Figure 2. Sigmoid activation function
图 2. Sigmoid 激活函数

从图中可知, Sigmoid 函数具有软饱和性: 函数在定义域范围内, 连续可导, 在输入极大极小时, 函数斜率趋近 0, 导数趋向于 0, 被称为梯度弥散。梯度弥散会造成梯度幅度随传播深度增加而急剧减小的现象, 神经元权重更新放缓, 网络难以训练, 不利于神经网络的发展。

为解决激活函数梯度弥散问题, Krizhevsky 等人采用 ReLU 作为激活函数, 其定义如下:

$$f(x) = \max(0, x) \quad (3)$$

ReLU 激活函数在 $x > 0$ 的定义域内导数恒为定值, 反向传播时可简化计算, 加快收敛速度。在 $x < 0$ 定义域内具有硬饱和特性: 输入落在此区域, 对应的输出皆为 0, 神经元反向传播一阶梯度亦为 0, 神经元不具有激活作用, 即神经元死亡, 导致模型的拟合力下降。此外, ReLU 函数在 $x < 0$ 定义域对应输出为 0 这一特性导致神经元输出均值大于 0, 不利于迭代计算, 此问题被称为均值偏移: 后一个神经元的输入为前一个神经元的输出, 因输出皆为正值, 后一个神经元的输入被限制, 模型的拟合能力下降, 制约深度模型的性能。

Maas 等人引入 L-ReLU 激活函数可有效解决均值偏移问题, 其函数定义为:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha x, & x < 0 \end{cases} \quad (4)$$

L-ReLU 激活函数在 $x \geq 0$ 定义域, 一阶导数恒定, 利于计算, 与 ReLU 性质一致。在 $x < 0$ 定义域内, L-ReLU 图像位于 y 轴的负半轴, 减缓了均值偏移。

He Kaiming 等人引入 P-ReLU 激活函数, 以获取更贴合模型的负轴斜率, 其函数定义为:

$$f(x) = \begin{cases} x, & x \geq 0 \\ \alpha_i x, & x < 0 \end{cases} \quad (5)$$

其函数性质与 L-ReLU 相似, 不同之处在于: 在 $x < 0$ 定义域, 斜率 α_i 需要模型根据输入数据学习而来, 增加了计算复杂度, 给系统计算增添了负载。

本文中使用的 Swish 激活函数的定义如下:

$$f(x) = \alpha * x * \text{sigmoid}(x) \quad (6)$$

其中, α 为常数, 本文将 α 设置为 1。函数图像如图 3 所示。

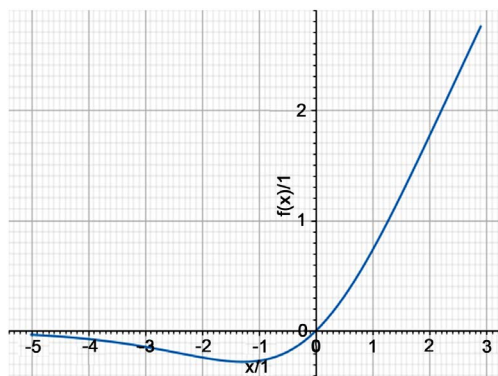


Figure 3. Swish activation function
图 3. Swish 激活函数

当 $x > 0$ 时, Swish 激活函数的一阶导数易于计算, 利于模型训练。当 $x < 0$ 时, 与 ReLU 函数相比, Swish 函数既能够均衡正负轴比重, 减缓了均值偏移现象, 又由于它无硬饱和性, 避免了神经元死亡现

象；与 L-ReLU 函数相比，Swish 函数是非线性的，具有软饱和性，鲁棒性更好；与 P-ReLU 相比，Swish 函数不需要计算参数 α ，减少了计算量，且鲁棒性更好。因此，Swish 激活函数的性能优于 ReLU、L-ReLU 和 P-ReLU 函数。

2.1.2. Swish-FER-CNNs 中的反向传播算法

反向传播算法可广泛应用于深度学习梯度计算，其中包含前馈传播和后馈传播两部分。激活函数是算法的重要组成部分。使用 Swish 激活函数的前馈传播模型可表示为：

$$u^{(i)} = \text{Swish}(A^{(i)}) \quad (7)$$

其中 $u^{(i)}$ 表示 i 节点， $A^{(i)}$ 是 $u^{(i)}$ 所有父节点的集合， $\text{Swish}(\cdot)$ 表示激活函数映射操作。

反向传播算法中为计算单个节点损失函数 $u^{(n)}$ 对 n_i 个输入节点 $u^{(1)}$ 到 $u^{(n)}$ 的梯度，使用图 ξ 储存前向传播的计算过程及节点数值。定义子图 β 在 ξ 上增添额外节点一组且依赖于 ξ ，以实现后向传播过程及节点数值存储。 β 的计算与 ξ 恰好相反， β 中的每个节点导数 $\frac{\partial u^{(n)}}{\partial u^{(i)}}$ 与 ξ 中 $u^{(i)}$ 相关联，可表示为：

$$\frac{\partial u^{(n)}}{\partial u^{(i)}} = \sum_{i,j \in Pa(u^{(i)})} \frac{\partial u^{(n)}}{\partial u^{(i)}} \frac{\partial u^{(i)}}{\partial u^{(j)}} \quad (8)$$

执行内积，其内积因子一为对于子节点 $u^{(i)}$ 已计算的梯度 $\frac{\partial u^{(n)}}{\partial u^{(i)}}$ ，其因子二为对于子节点 $u^{(i)}$ 的偏导数 $\frac{\partial u^{(i)}}{\partial u^{(j)}}$ 。总而言之， ξ 中每一条边中的计算过程都包含激活函数运算，Swish 激活函数的非线性对鲁棒性较强，计算复杂度较低，学习拟合力强，从而提高反向传播算法性能。

改进的反向传播算法使用 $h^{(i)}$ 表示第 i 层隐含层输出值， $a^{(i)}$ 表示激活函数的输入值， θ 包含所有参数(权值和偏置)， $\Omega(\theta)$ 为正则式， J 为总代价函数， g 表示当前位置梯度， ∇ 表示梯度的更迭矢量， \odot 表示矩阵点积。算法对于每一层 k 都生成了对激活函数输入值 a^k 的梯度，从输出层向后计算梯度一直到第一个隐含层，生成的梯度可视为指导每层输出该如何调整以减小模型误差的依据，再根据以上产生的梯度可以计算得到对每层参数的梯度，从而实现代价函数后馈传播，详细算法如下所示。

改进的反向传播算法：

Require: 网络深度, l

Require: $W^{(i)}, i \in \{1, \dots, l\}$

Require: $b^{(i)}, i \in \{1, \dots, l\}$

Require: x , 数据集图像输入

Require: y , 输出标签

前馈传播部分：

$$h^{(0)} = x$$

for $k = 1, \dots, l$ do

$$a^{(k)} = b^{(k)} + W^{(k)} h^{(k-1)}$$

$$h^{(k)} = \text{Swish}(a^{(k)})$$

end for

$$\hat{y} = h^{(l)}$$

$$J = L(\hat{y}, y) + \lambda \Omega(\theta)$$

后馈传播部分:

计算顶层梯度:

$$g \leftarrow -\nabla_{\hat{y}} J = \nabla_{\hat{y}} L(\hat{y}, y)$$

for $k = l, l-1, \dots, 1$ do

 将层输出的梯度转换为Swish激活函数输入前的梯度:

$$g \leftarrow -\nabla_{a^{(k)}} J = g \odot \text{Swish}'(a^{(k)})$$

 计算关于权重和偏置的梯度:

$$\nabla_{b^{(k)}} J = g + \lambda \nabla_{b^{(k)}} \Omega(\theta)$$

$$\nabla_{W^{(k)}} J = g h^{(k-1)\top} + \lambda \nabla_{W^{(k)}} \Omega(\theta)$$

 向下一更低的隐含层传播梯度:

$$g \leftarrow -\nabla_{h^{(k-1)}} J = W^{(k)\top} g$$

end for

2.2. Swish-FER-CNNs 网络模型

为适应 Kaggle 数据 48×48 灰度图像, 并进一步提高模型性能, 本文优化了网络结构, 与改进的反向传播算法相结合, 构建了 Swish-FER-CNNs 深度学习模型。从 48×48 的图像中随机截取 42×42 的灰度图像作为网络的输入。第一层卷积层采用 16 个大小为 3×3 的卷积核对大小为 $42 \times 42 \times 1$ 的输入图像进行卷积操作, 输出大小为 $42 \times 42 \times 16$ 的特征图, 第二层卷积采用 32 个大小为 3×3 的卷积核对上一层输出的特征图进行卷积操作, 并输出大小为 $42 \times 42 \times 32$ 的特征图, 池化操作后可获得大小为 $21 \times 21 \times 32$ 的特征图, 并作为第三层卷积的输入。之后的卷积操作类似于前两层卷积层, 第三层和第五层的卷积层输出需用于后续的池化层操作的输入, 第四层的卷积层输出直接作为第五层卷积层的输入。Swish-FER-CNNs 中使用的池化层全部采用步长为 2、尺寸为 3×3 的最大池化(max pooling)操作。每次池化操作都能减少二分之一的特征参数, 提取有效信息、避免过拟合现象。第三层池化层后输出的特征图大小为 $5 \times 5 \times 64$, 输出连接到第一层全连接层上, 全连接层的输出大小为 $1 \times 1 \times 2048$, 此过程的丢弃率为 0.45, 意味着全连接层中的神经元有 0.45 的概率不参与迭代, 能有效防止过拟合现象。经过两层全连接层, 最后使用 softmax 层分类得到的输出尺寸为 $1 \times 1 \times 7$, 对应于 7 类表情标签。具体的改进的网络结构如表 1 所示。

Table 1. Swish-FER-CNNs network architecture

表 1. Swish-FER-CNNs 网络结构

type	filter	kernel	stride	pad	input	output	dropout
input						$42 \times 42 \times 1$	
convolution 1	16	3×3	1	1	$42 \times 42 \times 1$	$42 \times 42 \times 16$	
convolution 2	32	3×3	1	1	$42 \times 42 \times 16$	$42 \times 42 \times 32$	
pooling 1		3×3	2		$42 \times 42 \times 32$	$21 \times 21 \times 32$	
convolution 3	32	4×4	1	1	$21 \times 21 \times 32$	$20 \times 20 \times 32$	
pooling 2		3×3	2		$20 \times 20 \times 32$	$10 \times 10 \times 32$	
convolution 4	32	3×3	1	1	$10 \times 10 \times 32$	$10 \times 10 \times 32$	

Continued

convolution 5	64	3 × 3	1	1	10 × 10 × 32	10 × 10 × 64	
pooling 3		3 × 3	2		10 × 10 × 64	5 × 5 × 64	
full-connected 1						1 × 1 × 2048	0.45
full-connected 2						1 × 1 × 1024	0.45
output						1 × 1 × 7	

相比 Jeon 的深度学习模型，Swish-FER-CNNs 网络模型多次使用了级联的两层 3 × 3 的卷积层替代 Jeon 使用的 5 × 5 卷积层，加深了网络的层数，尺寸较小的过滤器可提取细化的特征，每层卷积后均激发 Swish 函数，使整体网络的激活函数频数增加，网络的非线性拟合能力增强。

3. 数据集

实验采用的数据集为 kaggle 人脸情绪识别数据集——fer2013，该数据集由 Picere 和 Aaron 在 ICML2013 研讨会上发布，共由 35887 张人脸情绪图片组成。其中，生气的表情有 4953 张，厌恶 547 张，恐惧 5121 张，高兴 8989 张，悲伤 6077 张，惊讶 4002 张，中性 6198 张。数据集由三个部分组成，第一部分为训练集，包含 28,709 张图片。第二部分为验证集，包含 3589 张图片。第三部分为测试集，包含 3589 张图片。

4. 实验分析

实验使用 caffe [20]的深度学习框架，操作系统为 ubuntu16.04，GPU 为 GTX1050Ti。训练的学习率为 0.001，动量为 0.9，学习策略为 SGD。Swish-FER-CNNs 模型开始迭代后，测试准确率开始逐步增加，测试误差逐步下降，之后趋于平稳，这说明模型的准确率达到模型的饱和状态，训练迭代次数选择合理。训练完成后，测试集准确率达到 74.76%，测试误差收敛到 0.3168。训练过程如图 4、图 5 所示：

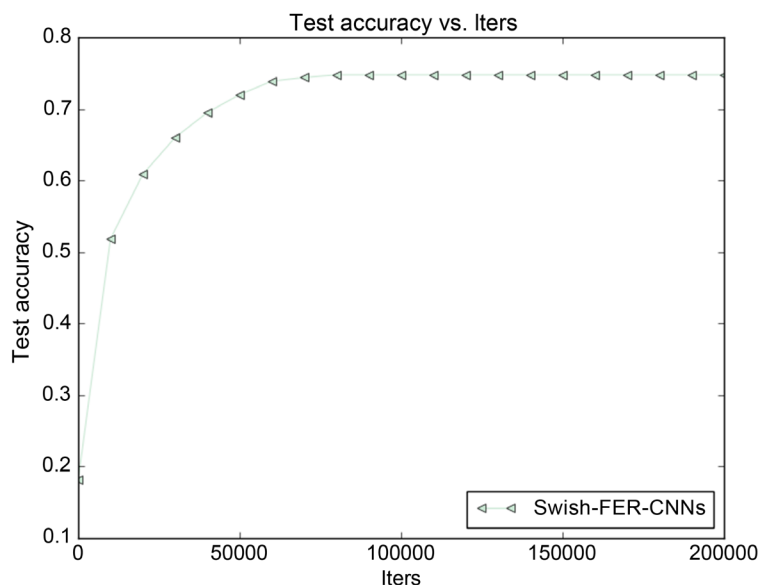


Figure 4. Training accuracy
图 4. 训练准确率

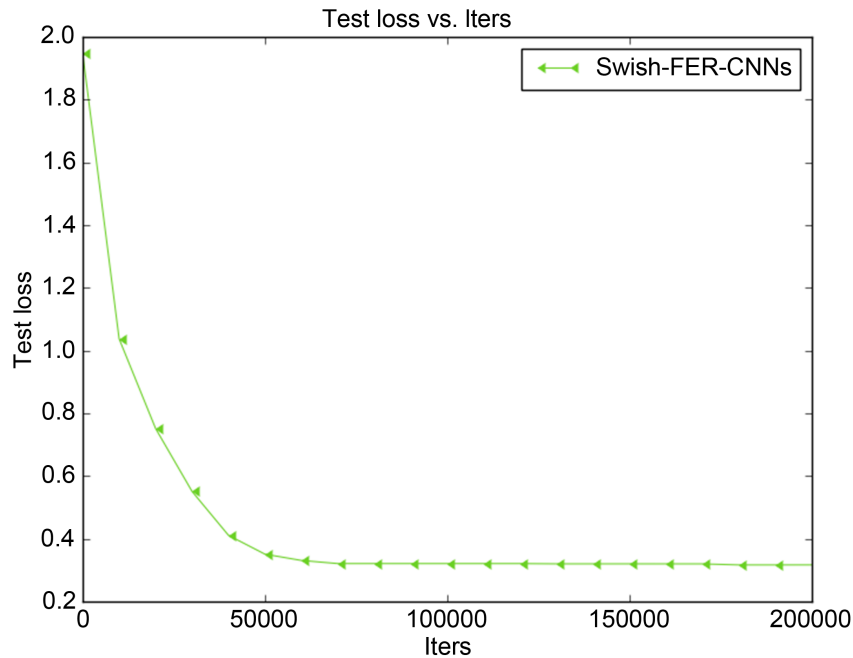


Figure 5. Training loss function
图 5. 训练损失函数

采用训练好的模型对测试数据集中各个类别的情绪图片进行批量识别。可得到第 i 类情绪图片中正确识别的个数，记为 TP_i ，余下的为错误识别个数，记作 FN_i 。 $TP_i + FN_i$ 恒等于测试数据集中该类情绪图片的样本总个数 Sum_i 。7 类情绪的测试准确率 P_i 使用以下公式计算：

$$P_i = \frac{TP_i}{TP_i + FN_i} = \frac{TP_i}{Sum_i} \quad (9)$$

Swish-FER-CNNs 学习模型对各类情绪识别的准确率如表 2 所示。

Table 2. Confusion matrix of emotion recognition accuracy of Swish-FER-CNNs model
表 2. Swish-FER-CNNs 模型情绪识别准确率混淆矩阵

Swish-FER-CNNs	Angry	Disgust	Fear	Happy	Sad	Surprise	Neutral
Angry	65.33	1.01	7.29	4.49	15.29	1.52	5.07
Disgust	20.67	71.41	1.03	0.00	1.91	1.90	3.07
Fear	9.1	1.15	53.31	1.80	19.01	7.05	8.59
Happy	1.60	0.00	1.02	92.88	1.22	0.40	2.88
Sad	6.0	0.00	1.02	5.01	68.42	0.73	18.82
Surprise	2.10	0.00	8.01	1.92	1.92	85.75	0.30
Neutral	2.01	0.90	1.50	7.50	12.11	12.10	73.02

由于 kaggle 数据集中各类别的情绪样本不均匀，导致各类样本在模型中的学习数据量不一致。因此，各个类别的分类效果不同。

将 Jeon's 深度学习模型分别采用 ReLU、L-ReLU、P-ReLU 和 Swish 激活函数，与本文的 Swish-FER-CNNs 深度学习模型进行对比，结果如图 6 所示。

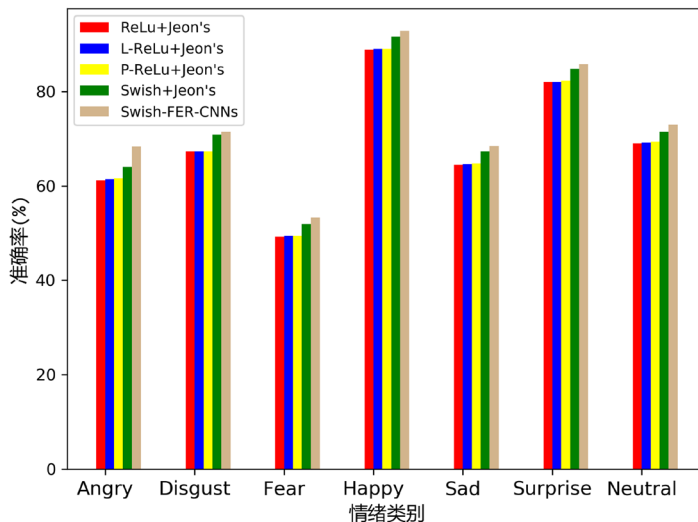


Figure 6. Accuracy of test confusion matrix for each model
图 6. 各模型的测试混淆矩阵精度

如图所示，分别使用 ReLU、L-ReLU、P-ReLU、Swish 激活函数，模型识别准确率依前述顺序依次递增。改进后的 Swish-FER-CNNs 深度模型，相较于使用 Swish 激活函数的 Jeon’s 深度学习模型，取得了更高的识别准确率。

为更直接地分析模型的性能，采用准确率 ACC 表示模型在测试集中识别的准确率，如下所示：

$$ACC = \frac{\sum_{i=1}^s TP_i}{\sum_{i=1}^s (TP_i + FN_i)} \quad (10)$$

测试集中的人脸情绪分为 s 种类别，采用 TP_i 表示某个类别中正确识别情绪的个数， FN_i 表示该类别错误识别的个数。各方法识别情绪的准确率如表 3 所示。

Table 3. Comparison of recognition accuracy
表 3. 识别准确率对比

模型	测试集的准确率	相较 ReLU + Jeon’s 准确率提高度
ReLU + Jeon’s	70.74%	-
L-ReLU + Jeon’s	70.88%	+0.14%
P-ReLU + Jeon’s	70.99%	+0.25%
Swish + Jeon’s	73.54%	+2.80%
Swish-FER-CNNs	74.76%	+4.02%

由表 3 可以看出，经过充分的迭代，在 Jeon’s 深度学习模型下，不同激活函数的识别准确率为：ReLU < L-ReLU < P-ReLU < Swish，因此 Swish 激活函数性能优于前三者。此外，Swish-FER-CNNs 网络模型与基于 Swish 等激活函数的 Jeon’s 深度学习模型相比，具有更高的识别准确率。

5. 结束语

本文研究了激活函数的性能和网络结构的特性，采用性能优良的 Swish 激活函数改进反向传播算法，

并使用能够提取细化特征的多层小尺寸卷积模块, 改进了网络结构, 构建了人脸情绪识别深度学习模型 Swish-FER-CNNs。新模型是一种自动提取人脸情绪特征的深度学习算法, 提高了人脸情绪识别的准确率。模型的训练时间仍然可以通过改进激活函数进行优化, 这将是下一步的研究方向。Swish 激活函数中参数 α 的取值对模型性能有较大的影响, 这可作为优化模型的另一个方向。

基金项目

本研究获得国家自然科学基金(NO.61771414)项目资助。

参考文献

- [1] Giannopoulos, P., Perikos, I. and Hatzilygeroudis, I. (2018) Deep Learning Approaches for Facial Emotion Recognition: A Case Study on FER-2013. In: Hatzilygeroudis, I. and Palade, V., Eds., *Advances in Hybridization of Intelligent Methods, Smart Innovation, Systems and Technologies*, Springer, Cham, 1-16. https://doi.org/10.1007/978-3-319-66790-4_1
- [2] Bruce, V. and Young, A. (1986) Understanding Face Recognition. *British Journal of Psychology*, **77**, 305-327. <https://doi.org/10.1111/j.2044-8295.1986.tb02199.x>
- [3] Ioffe, S. and Szegedy, C. (2015) Batch Normalization: Accelerating Deep Network Training by Reducing Internal Covariate Shift. Arxiv: 1502.03167.
- [4] Simonyan, K. and Zisserman, A. (2014) Very Deep Convolutional Networks for Large-Scale Image Recognition. Arxiv: 1409.1556.
- [5] Krizhevsky, A., Sutskever, I. and Hinton, G.E. (2012) ImageNet Classification with Deep Convolutional Neural Networks. In: Pereira, F., Burges, C.J.C., Bottou, L. and Weinberger, K.Q., Eds., *Advances in Neural Information Processing Systems*, The MIT Press, Cambridge, MA, 1097-1105.
- [6] Zhang, C. and Woodland, P.C. (2016) DNN Speaker Adaptation Using Parameterised Sigmoid and ReLU Hidden Activation Functions. 2016 *IEEE International Conference on Acoustics, Speech and Signal Processing*, Shanghai, 20-25 March 2016, 5300-5304. <https://doi.org/10.1109/ICASSP.2016.7472689>
- [7] Taigman, Y., Yang, M., Ranzato, M.A. and Wolf, L. (2014) Deepface: Closing the Gap to Human-Level Performance in Face Verification. 2014 *IEEE Conference on Computer Vision and Pattern Recognition*, Columbus, OH, 23-28 June 2014, 1701-1708. <https://doi.org/10.1109/CVPR.2014.220>
- [8] Szegedy, C., Ioffe, S., Vanhoucke, V. and Alemi, A.A. (2017) Inception-v4, Inception-Resnet and the Impact of Residual Connections on Learning. *Proceedings of the Thirty-First AAAI Conference on Artificial Intelligence*, San Francisco, CA, 4-9 February 2017, 4278-4284.
- [9] Wang, X., Girshick, R., Gupta, A. and He, K. (2018) Non-Local Neural Networks. 2018 *IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, 18-23 June 2018, 7794-7803. <https://doi.org/10.1109/CVPR.2018.00813>
- [10] Jeon, J., Park, J.C., Jo, Y.J., et al. (2016) A Real-Time Facial Expression Recognizer Using Deep Neural Network. In: *Proceedings of the 10th International Conference on Ubiquitous Information Management and Communication*, ACM, New York, 94:1-94:4. <https://doi.org/10.1145/2857546.2857642>
- [11] LeCun, Y., Boser, B.E., Denker, J.S., et al. (1990) Handwritten Digit Recognition with a Back-Propagation Network. In: Pereira, F., Burges, C.J.C., Bottou, L. and Weinberger, K.Q., Eds., *Advances in Neural Information Processing Systems*, The MIT Press, Cambridge, MA, 396-404.
- [12] Ramachandran, P., Zoph, B. and Le, Q.V. (2017) Searching for Activation Functions. *Computer Science*, ArXiv: 1710.05941.
- [13] Saxe, A.M., McClelland, J.L. and Ganguli, S. (2013) Exact Solutions to the Nonlinear Dynamics of Learning in Deep Linear Neural Networks. *Computer Science*, ArXiv: 1312.6120.
- [14] Eger, S., Youssef, P. and Gurevych, I. (2019) Is It Time to Swish? Comparing Deep Learning Activation Functions Across NLP Tasks. *Computer Science*, arXiv: 1901.02671.
- [15] Maas, A.L., Hannun, A.Y. and Ng, A.Y. (2013) Rectifier Nonlinearities Improve Neural Network Acoustic Models. *International Conference on Machine Learning*.
- [16] Xu, B., Wang, N., Chen, T. and Li, M. (2015) Empirical Evaluation of Rectified Activations in Convolutional Network. *Computer Science*, arXiv: 1505.00853.
- [17] Zhang, X., Zou, Y. and Shi, W. (2017) Dilated Convolution Neural Network with LeakyReLU for Environmental

Sound Classification. 2017 22nd International Conference on Digital Signal Processing, London, 23-25 August 2017, 1-5. <https://doi.org/10.1109/ICDSP.2017.8096153>

- [18] He, K., Zhang, X., Ren, S. and Sun, J. (2015) Delving Deep into Rectifiers: Surpassing Human-Level Performance on ImageNet Classification. 2015 IEEE International Conference on Computer Vision, Santiago, Chile, 7-13 December 2015, 1026-1034. <https://doi.org/10.1109/ICCV.2015.123>
- [19] Gulcehre, C., Moczulski, M., Denil, M. and Bengio, Y. (2016) Noisy Activation Functions. In: *Proceedings of the 33rd International Conference on Machine Learning*, PMLR 48, 3059-3068.
- [20] Jia, Y., Shelhamer, E., Donahue, J., et al. (2014) Caffe: Convolutional Architecture for Fast Feature Embedding. In: Balcan, M.F. and Weinberger, K.Q., Eds., *Proceedings of the 22nd ACM International Conference on Multimedia*, ICML, New York, 675-678.

知网检索的两种方式:

1. 打开知网首页: <http://cnki.net/>, 点击页面中“外文资源总库 CNKI SCHOLAR”, 跳转至: <http://scholar.cnki.net/new>, 搜索框内直接输入文章标题, 即可查询;
或点击“高级检索”, 下拉列表框选择: [ISSN], 输入期刊 ISSN: 2325-6753, 即可查询。
2. 通过知网首页 <http://cnki.net/>顶部“旧版入口”进入知网旧版: <http://www.cnki.net/old/>, 左侧选择“国际文献总库”进入, 搜索框直接输入文章标题, 即可查询。

投稿请点击: <http://www.hanspub.org/Submission.aspx>

期刊邮箱: jisp@hanspub.org