工程优化方法大作业

姓名：陈佳硕
学号：S230200181

1.CNN 卷积神经网络的搭建

1.1 卷积层

自然图像有其固有特性：在从某一图像子块上学习到一些特征后，可将这些特征作为探测器，应用到所有子块中去，获得不同子块的激活值。CNN 中的卷积也是利用图像的这种固有特性，具体做法是：卷积层中一个可训练的卷积核与上一层中不同组合的特征图进行卷积，加上偏置得到当前层的特征图。其基本原理如图 1 所示。

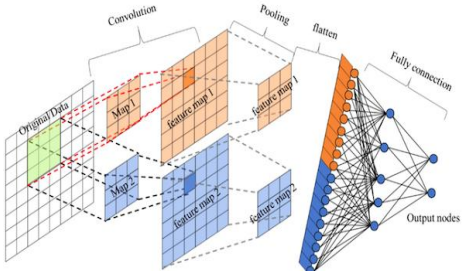


图 1　卷积神经网络基本原理图

该过程可以用下式表示

$$x_j^l = \sum_{i \in M_j} y_i^{l-1} \otimes k_{ij}^l + b_j^l \qquad (1)$$

式中：x_j^l 为第 l 层第 j 个特征图的输入；y_i^{l-1} 为第 $l-1$ 层第 i 个特征图的输出；k_{ij}^l 为前一层第 i 个特征图与当前层第 j 个特征图之间的卷积核；b_j^l 为第 l 层第 j 个特征图的偏置；$i \in M_j$ 为前一层中与当前层第 j 个特征图有连接的所有特征图。

本文共设置了 4 个卷积层，设置为 1×1 的卷积层，使输入增加了非线性的表示、加深了网络、提升了模型的表达能力，同时基本不增加计算量。第 2 个和第 4 个卷积层的核的大小为 5×5，第 3 个卷积层的核的大小为 3×3，且步长均为 1。

1.2 池化层

卷积神经网络中，一般在卷积层后会跟着池化层。在通过卷积获得了特征之后，就可以利用这些特征去分类。虽然在理论上，可以用所有提取到的特征去训练分类器，但是这样面临着巨大的计算量的挑战。为了解决这个问题，则需要降低特征的维数。对于图像不同位置的特征，可以进行合并统计，比较常用的是用图像区域上的某个特定特征的最大值（或平均值）来对图像进行描述。这种操作方法被称为池化。根据池化方法可以分为平均池化、最大池化和随即池化。从以上可以看出，局部感受野、权值共享和空间亚采样是卷积神经网络的三个特性，通过这三者的结合可以得到某种程度上的尺度、位移和形变不变性。

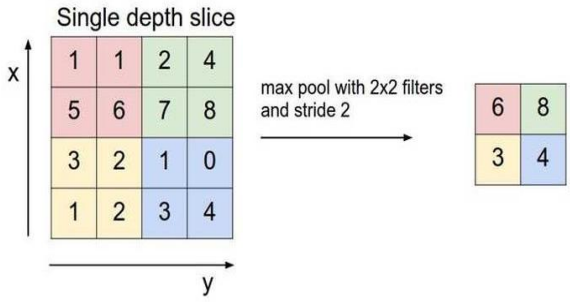


图 2　最大池化过程

本文采用的是最大池化（max-pooling），它是一种非线性的下采样。Max-pooling 的基本思想是，将图像分割成相互之间没有交集的矩形区域，对于每个区域，输出最大值。Max-pooling 有两个优点使其称为一种比较好的下采样方法：（1）减少上层计算的复杂性；（2）提供了平移不变性。图 2 为最大池化过程。

本文在除了第一个卷积层外，每个卷积层后面都设置了一个池化层，且每个池化层的核的大小为 2×2，步长为 2。

1.3　激活函数

“激活函数”，又称“非线性映射函数”，是深度卷积神经网 络中不可或缺的关键模块，其模拟了生物神经元特性，接受一组输入信号并产生输出，并通过一个阈值来控制神经元的兴奋与抑制状态。图 3 为当下神经网络常用的 5 种激活函数。

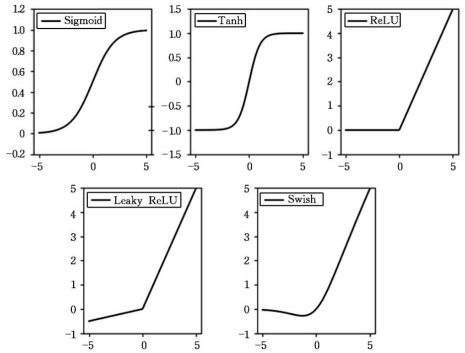


图 3　激活函数

首先尝试使用了 Sigmoid 和 Tanh。

Sigmoid 函数的数学公式为：

$$f\left(x\right)=\frac{1}{1+e^{x}} \qquad (2)$$

其中 $f(x)$ 的取值范围是 0~1。因此 Sigmoid 函数的缺点是其值不能取到 0。

Tanh 激活函数虽然解决了 Sigmoid 函数不能取到 0 的问题，但是任然存在梯度消失的问题。Tanh 的数学公式为：

$$f\left(x\right)=\frac{e^{x}-e^{-x}}{e^{x}+e^{-x}} \qquad (3)$$

在训练后期，二者都产生了因没有进行归一化而梯度消失训练困难的问题。

为了克服该问题，本文采用修正线性单元 Relu（Rectified linear unit）[10]作为激活函数。Relu 是目前最成功和广泛使用的激活函数。其是一个分段线性函数：当输入为负时，Relu 函数的输出为 0；当输入为正时，Relu 函数的输出为 x 。相比于 Sigmoid 和 Tanh，Relu 拥有更快的收敛速度，同时还在隐藏层引入稀疏性而使网络容易获得稀疏表示。尽管 Relu 在 0 处的不连续性可能影响反向传播的表现，但实验证明，Relu 拥有比 Sigmoid 和 Tanh 激活函数更出色的性能，能更好地解决梯度消失问题，二者的对比如图 7 所示。Relu 函数的数

学公式为：

$$f\left(x\right)=\max\left(0,x\right) \tag{4}$$

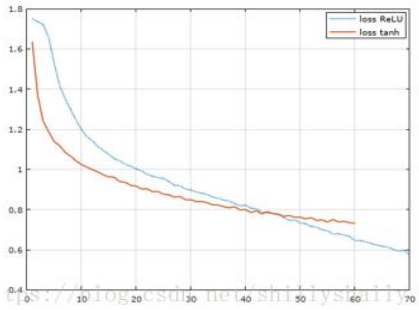


图 4　Relu 和 tanh 的损失函数变化对比

1.4　全连接层和 Softmax 回归

全连接层上的每一个神经单元，均与上一层特征图中的所有神经单元互相连接。每一个神经单元的输出可以用下式表示

$$h_{W,b}\left(x\right)=f\left(W^{T}x+b\right) \tag{5}$$

式中：x 为神经元的输入；$h_{W,b}\left(x\right)$ 为神经元的输 出；W 为连接权重；b 为偏置；$f\left(*\right)$ 为非线性激活函数，在前文中我们已经确定。

本文在卷积神经网络最后一层使用 Sofmax 来进行分类，将逻辑斯蒂回归模型再多分类问题上推广就能得到 Softmax 回归。对于多分类任务，训练集为 $\left\{\left(x^{(1)},y^{(1)}\right),...,\left(x^{(m)},y^{(m)}\right)\right\}$，输出 $y^{(i)}\in\left\{1,2,...,k\right\}$，其中 k 表示类别个数，本文中 Softmax 分类器将表情数据分为 7 类，即 $k=7$。

设 x 表示输入，测试时针对每一个类别 j，先用假设函数估算出其概率值 $p\left(y=j\middle|x\right)$。在这一步中，根据需要估算出所有结果可能出现的概率。在后面的计算中，假设函数会输出一个 k 维向量来表示 k 个估计的概率值，向量相加和为 1。假设函数为 $h_{\theta}\left(x\right)$，形式如下：

$$h_{\theta}\left(x^{(i)}\right)=\begin{bmatrix}p\left(y^{(i)}=1\middle|x^{(i)};\theta\right)\\p\left(y^{(i)}=2\middle|x^{(i)};\theta\right)\\\vdots\\p\left(y^{(i)}=k\middle|x^{(i)};\theta\right)\end{bmatrix}=\frac{1}{\sum_{j=1}^{k}e^{\theta_j^{T}x^{(i)}}}\begin{bmatrix}e^{\theta_1^{T}x^{(i)}}\\e^{\theta_2^{T}x^{(i)}}\\\vdots\\e^{\theta_k^{T}x^{(i)}}\end{bmatrix} \tag{6}$$

其中 $\theta_1,\theta_2,\ldots,\theta_k\in\Re^{n+1}$ 为其参数。而 $\frac{1}{\sum_{j=1}^{k}e^{\theta_j^{T}x^{(i)}}}$ 这一项，则是进行归一化，这样所有概率值相加和为 1。最后进行测试分类时，将前一层的输出传入 $h_{\theta}\left(x\right)$，最大分量所在维度对应的类别即为 x 所属。

在本文的表情识别任务中，使用交叉熵（Cross Entropy，CE）作为度量损失函数[13]，公式如下：

$$CE=-\ln p_n \tag{7}$$

其中 p_n 为模型预测结果对应标签的概率，即 $\frac{e^{\theta_n^{T}x^{(i)}}}{\sum_{j=1}^{k}e^{\theta_j^{T}x^{(i)}}}$。

在误差的反向传播过程中，神经网络计算交叉熵，将其由输出端反向传播至输入端，用梯度下降法更新网络的权值和阈值。网络的训练过程持续不断地进行，直到输出误差降低到可接受的程度，或满足其他终止条件才停止训练。输出层神经元的连接权值和阈值的更新过程如公式（8）-（11）所示，其他层参数的更新过程与其类似，即

$$\Delta w_m = -\alpha \frac{\partial CE}{\partial w_m} \tag{8}$$

$$\Delta b_k = -\alpha \frac{\partial CE}{\partial b_k} \tag{9}$$

$$w_m = w_m + \Delta w_m \tag{10}$$

$$b_k = b_k + \Delta b_k \tag{11}$$

式中 α 为学习率，采用自适应的学习率，在接近极值点时，使用较小的学习率，反之则使用较大的学习率。

2.模型效果

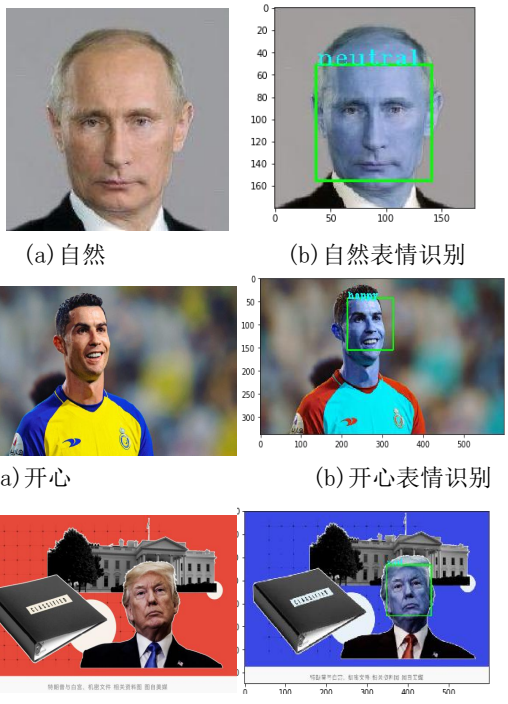
在模型训练好之后，使用该模型进行同时预测，对多个处理过的脸部预测结果进行线性加权融合，最后得出预测结果。这种方法可以提高表情识别分类器在空间上对局部位移和轻微形变的鲁棒性，可以有效提高表情识别系统分类的准确率。

表 1　模型准确率

模型	模型 1	模型 2	模型 3（当前）	模型 4
处理前准确率（%）	65.5	62.3	68.4	67.37
处理后准确率（%）	67.6	64.1	70.2	69.4

最后在 FER2013 的训练集上，该模型表情识别的准确率最高为 70.2%。

为了更清晰地呈现表情识别的结果，在表情识别模型的最后添加了人脸画框功能，最后将识别到的表情显示到方框上。在网络中随机找到了几个人脸图片，最后呈现的效果如下图所示。



(a) 自然　(b) 自然表情识别

(a) 开心　(b) 开心表情识别

(a) 伤心　(b) 伤心表情识别

图 5　人脸表情识别效果展示