

博弈杯:VGG 与 ResNet 在情绪识别中的应用

方桂安, 刘梦莎, 刘玥

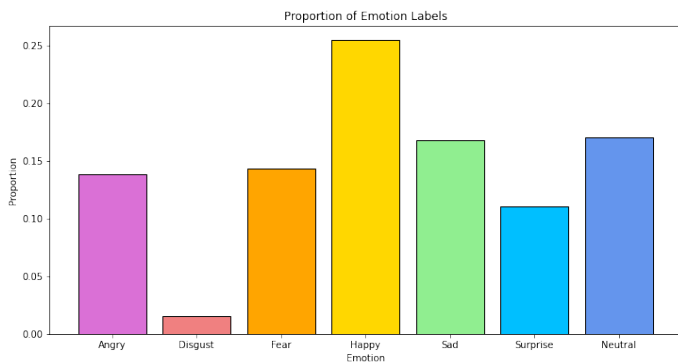
摘要—人脸识别技术越来越多地应用于当今世界的各个场景与产品中。人脸识别是实现自然的人机交互 (HCI) 的基本手段之一。情绪识别是进一步研究人机交互的重中之重。情绪的复杂性使得信息采集的这一任务在这项工作中变得更加困难。在本文中, 我们首先分析了两万张数据集的类别分布情况, 发现七种情绪在数据集的分布并不均衡; 而后, 我们将两万个数据集按 8:1:1 的比例进行划分, 分为 Training & PublicTest & PrivateTest 三部分; 随后, 我们通过比较简单神经网络、VGG19 和 RseNet18 的训练结果, 发现 Rsenet18 的训练结果最佳; 于是, 我们开始对 Rsenet18 进行调参, 得出较优的超参数组合。实验结果表明, 我们训练出的模型对于情绪分类具有较高的准确率。

关键词—情绪识别, 深度学习, VGG19, Rsenet18

I. 数据处理

A. 预处理

FEATURE 为 48x48 的像素值, 读取训练集, 将数据统计并绘制出 label 的分布情况。



显然 7 种情绪在训练集的分布并不均衡。

数据为 48x48 的单通道灰度图, 用像素生成图片展示 7 种情绪, 可以看出仅凭肉眼很难区分。



B. 归一化

0 ~ 255 范围内的强度值是 RGB 色彩模式使用 RGB 模型为图像中每一个像素的 RGB 分量赋予的值。为了加快模型拟合速度, 我们对训练集进行了归一化。

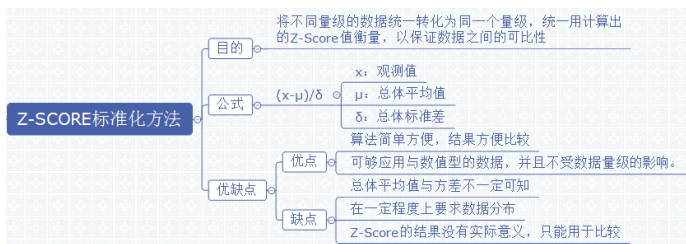


Fig. 1. z-score

本次数据处理使用的是 z-score 标准化, 转换公式为:

$$\begin{cases} Z_i = \frac{Z_i - \mu_i}{\sigma_i} \\ \mu_i = \frac{1}{N} \sum_{j=1}^N Z_j \\ \sigma_i = \sqrt{\frac{1}{N} \sum_{j=1}^N (Z_j - \mu_i)^2} \end{cases} \quad (1)$$

C. 数据集划分

1) 过拟合: 机器学习的目的是使学得模型不仅对已知数据而且对未知数据都能有很好的预测能力。

由于事先并不知道新样本的特征, 我们只能努力使训练误差最小化; 但是当学习器把训练样本学得“太好”, 将训练样本本身的特点当做所有样本的一般性质, 因此而导致泛化性能下降。过拟合 (overfitting) 是指学习时选择的模型所包含的参数过多, 以至出现这一模型对已知数据预测得很好, 但对未知数据预测得很差的现象。

2) 泛化能力: 概括地说, 泛化能力 (generalization ability) 是指机器学习算法对新鲜样本的适应能力。学习的目的是学到隐含在数据背后的规律, 对具有同一规律的数据集以外的数据 (数据集中不曾出现过的数据), 经过训练的模型也能给出合适的输出, 该能力称为泛化能力。

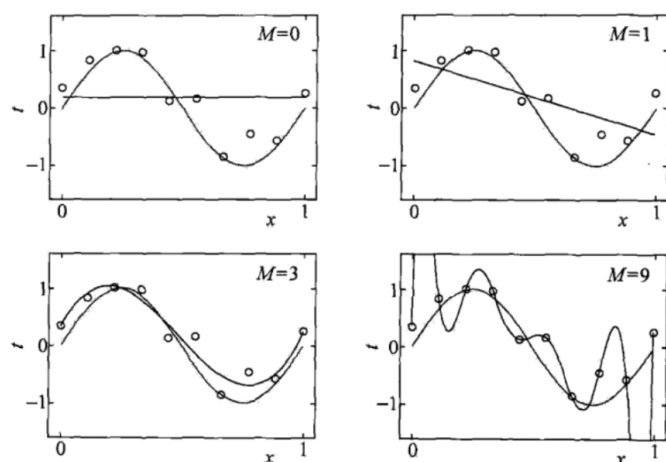


Fig. 2. 过拟合

现实中采用最多的办法是通过测试误差来评价学习方法的泛化能力。显然，这种评价是依赖测试集的，如果测试集不可靠，则评价结果不可靠，而测试集的误差评价也就是模型的性能评估。

泛化误差 (generalization error) 是模型的期望风险，或者说在新样本上的误差。泛化误差 = 偏差 + 方差 + 噪声。

3) **模型选择**: 当假设空间含有不同复杂度 (例如, 不同的参数个数) 的模型时, 就要面临模型选择 (model selection) 的问题。

- 选择或学习一个合适的模型, 并且如果在假设空间中存在“真”模型, 那么所选择的模型应该逼近真模型 (参数个数相同, 参数向量相近)
- 如果过于追求提高对训练数据的预测能力, 所选模型的复杂度则往往会比真模型更高, 则会出现过拟合
- 模型选择旨在避免过拟合并提高模型的泛化能力

由于我们拿到的数据集有限, 为了选择更好的模型, 提高模型的泛化能力, 我们采用交叉验证的方法, 随机地按比例对数据集进行划分。



Fig. 3. 模型选择

4) **具体划分**: 在机器学习算法中, 我们通常将原始数据集划分为三个部分:

- **Training set (训练集)**: 通过设置分类器的参数, 训练模型来拟合数据。
- **Validation set (验证集)**: 当通过训练集训练出多个模型后 (同一算法在不同参数下拟合数据的模型, 如不同参数组合的 SVM), 为了找出效果最佳的模型, 使用各个模型对验证集数据进行预测, 并记录模型的泛化能力。选出效果最佳的模型所对应的参数, 即用来调整模型参数。如 SVM 中的参数 c 和核函数等。
- **Testing set (测试集)**: 通过训练集和验证集得出最优模型后, 使用测试集计算模型的泛化误差, 用来衡量该最优模型的性能。即可以把测试集当做从来不存在的数据集, 当已经确定模型参数后, 使用测试集进行模型性能评价。

首先将数据集划分为训练集和测试集, 由于模型的构建过程中也需要检验模型的配置, 以及训练程度是过拟合还是欠拟合, 所以会将训练数据再划分为两个部分, 一部分是用于训练的训练集, 另一部分是进行检验的验证集。训练集用于训练得到神经网络模型, 然后用验证集验证模型的有效性, 挑选获得最佳效果的模型。验证集可以重复使用, 主要是用来辅助我们构建模型的。最后, 当模型“通过”验证集之后, 我们再使用测试集测试模型的最终效果, 评估模型的准确率, 以及误差等。

在具体实现中, 由于 Leaderboard 分为 Public 和 Private 两部分, 我们将两万个数据集按 8: 1: 1 的比例进行划分, 分为 Training & PublicTest & PrivateTest 三部分, 分别用于模型的训练和选择。

II. 模型构建及训练

A. VGG19 概述

在模型训练之初, 我们选用了 VGG19 模型。

VGG19 包含了 19 个隐藏层 (16 个卷积层和 3 个全连接层)。VGG 网络的结构非常一致, 从头到尾全部使用的是 3×3 的卷积和 2×2 的 maxpooling。

VGG19 采用连续的几个 3×3 的卷积核, 对于给定的感受野 (与输出有关的输入图片的局部大小), 采用堆积的小卷积核是优于采用大的卷积核, 因为多层非线性层可以增加网络深度来保证学习更复杂的模式, 而且代价还比较小 (参数更少)。

在 VGG 中, 使用了 3 个 3×3 卷积核来代替 7×7 卷积核, 使用了 2 个 3×3 卷积核来代替 5×5 卷积核, 这样做的主要目的是在保证具有相同感知野的条件下, 提升了网络的深度, 在一定程度上提升了神经网络的效果。

比如，3 个步长为 1 的 3×3 卷积核的一层层叠加作用可看成一个大小为 7 的感受野（其实就表示 3 个 3×3 连续卷积相当于一个 7×7 卷积），其参数总量为 $3 \times (9 \times C^2)$ ，如果直接使用 7×7 卷积核，其参数总量为 $49 \times C^2$ ，这里 C 指的是输入和输出的通道数。很明显， $49 \times C^2$ 小于 $27 \times C^2$ ，即减少了参数；而且 3×3 卷积核有利于更好地保持图像性质。

5×5 卷积看做一个小的全连接网络在 5×5 区域滑动，我们可以先用一个 3×3 的卷积滤波器卷积，然后再用一个全连接层连接这个 3×3 卷积输出，这个全连接层我们也可以看做一个 3×3 卷积层。这样我们就可以用两个 3×3 卷积级联（叠加）起来代替一个 5×5 卷积。如下图所示。

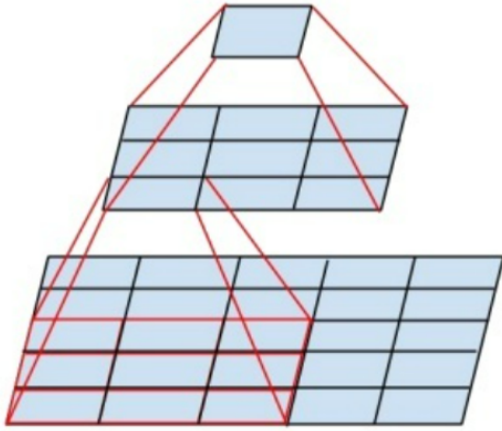


Fig. 4. 5×5 卷积

这里给出 VGG 网络的 5 种模型结构。A-E 模型均是由 5 个 stage 和 3 个全链层和一个 softmax 分类层组成，其中每个 stage 有一个 max-pooling 层和多个卷积层。每层的卷积核个数从首阶段的 64 个开始，每个阶段增长一倍，直到达到 512 个。模型 D 是就是经常说的 VGG16 网络，模型 E 则为 VGG19 网络。

- A: 是最基本的模型，8 个卷基层，3 个全连接层，一共 11 层。
- B: 在 A 的基础上，在 stage1 和 stage2 基础上分别增加了 1 层 3×3 卷积层，一共 13 层。
- C: 在 B 的基础上，在 stage3, stage4 和 stage5 基础上分别增加了一层 1×1 的卷积层，一共 16 层。
- D: 在 B 的基础上，在 stage3, stage4 和 stage5 基础上分别增加了一层 3×3 的卷积层，一共 16 层。
- E: 在 D 的基础上，在 stage3, stage4 和 stage5 基础上分别增加了一层 3×3 的卷积层，一共 19 层。

下面是 VGG 网络的结构（VGG16 和 VGG19 都在）：

ConvNet Configuration					
A	A+LRN	B	C	D	E
11 weight layers	11 weight layers	13 weight layers	16 weight layers	16 weight layers	19 weight layers
input (224 × 224 RGB image)					
conv 3-64	conv 3-64 LRN	conv 3-64	conv 3-64	conv 3-64	conv 3-64
maxpool					
conv 3-128	conv 3-128	conv 3-128	conv 3-128	conv 3-128	conv 3-128
maxpool					
conv 3-256	conv 3-256	conv 3-256	conv 3-256	conv 3-256	conv 3-256
conv 3-256	conv 3-256	conv 3-256	conv 3-256	conv 3-256	conv 3-256
maxpool					
conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512
conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512
maxpool					
conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512
conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512	conv 3-512
maxpool					
FC-4096					
FC-4096					
FC-1000					
soft-max					

VGG 网络结构

Fig. 5. VGG 网络结构

之所以在整个 VGG 网络中都使用 3×3 大小的 filter，是因为：

- 1) 3×3 是最小的能够捕获上下左右和中心概念的尺寸。
- 2) 两个 3×3 的卷基层的有限感受野是 5×5 ；三个 3×3 的感受野是 7×7 ，可以替代大的 filter 尺寸。（感受野表示网络内部的不同位置的神经元对原图像的感受范围大小，神经元感受野的值越大表示其能接触到的原始图像范围就越大，也意味着他可能蕴含更为全局、语义层次更高的特征；而值越小则表示其所包含的特征越趋向于局部和细节。）
- 3) 多个 3×3 的卷基层比一个大尺寸 filter 卷基层有更多的非线性，使得判决函数更加具有判决性。
- 4) 多个 3×3 的卷积层比一个大尺寸的 filter 有更多的参数。

B. VGG 训练过程

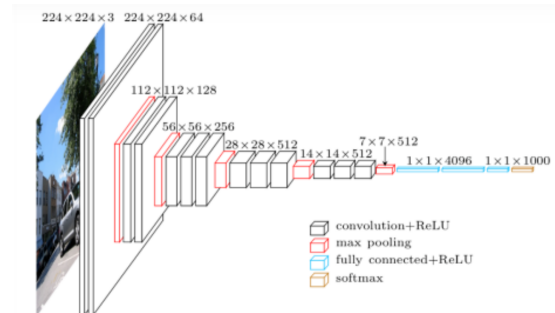


Fig. 6. VGG 训练过程

VGG19 的网络结构图，输入是一张 224×224 大小的 RGB 图片，在本次比赛中我们调整为 48×48 大小的

单通道图片。然后使用 3X3 大小的卷积核进行卷积，接着图片依次经过 5 个 Stage 和 3 层全连层的处理，一直到 softmax 输出分类。卷积核深度从 64 一直增长到 512，更好的提取了图片的特征向量。

- 1) Stage1: 包含两个卷积层，一个池化层，每个卷积层和池化层的信息如下：卷积核 3*3 深度 64 步长 1*1
- 2) Stage2: 包含两个卷积层，一个池化层，每个卷积层和池化层的信息如下：卷积核 3*3 深度 128 步长 1*1
- 3) Stage3: 包含四个卷积层，一个池化层，每个卷积层和池化层的信息如下：卷积核 3*3 深度 256 步长 1*1
- 4) Stage4: 包含四个卷积层，一个池化层，每个卷积层和池化层的信息如下：卷积核 3*3 深度 512 步长 1*1
- 5) Stage5: 包含四个卷积层，一个池化层，每个卷积层和池化层的信息如下：卷积核 3*3 深度 256 步长 1*1

整个网络包含 5 个池化层，分别位于每一个 Stage 的后面，每个池化层的尺寸均一样，如下：池化层过滤器 2*2 步长 2*2

对于其他的隐藏层，整个网络不包含 LRN，因为 LRN 会占用内存和增加计算时间。接着经过 3 个全链层的处理，由 Softmax 输出 7 个类别的分类结果。

层之间的连接关系，实际上就是前向传播的过程，其中主要包含卷积层、池化层和全连接层，前项传播就是将图片数据输入到网络中，经过层层卷积、池化、全连接等各种操作得到输出。卷积层可以起到下采样的作用，提取图片中的特征。池化层可以进一步缩小图像尺寸减少计算量。最后一层全连接层输出的维度为类别的数目，分别对应各类别的概率。

VGGNet 在训练时，先训练级别 A 的简单网络，再复用 A 网络的权重来初始化后面的几个复杂模型，这样训练收敛的速度更快。

在预测时，VGG 采用 Multi-Scale 的方法，将图像 scale 到一个尺寸 Q，并将图片输入卷积网络计算。然后在最后一个卷积层使用滑窗的方式进行分类预测，将不同窗口的分类结果平均，再将不同尺寸 Q 的结果平均得到最后结果，这样可提高图片数据的利用率并提升预测准确率。

在训练中，VGGNet 还使用了 Multi-Scale 的方法

做数据增强，将原始图像缩放到不同尺寸 S，然后再随机裁切 224*224 的图片，这样能增加很多数据量，对于防止模型过拟合有很不错的效果。

C. VGG 缺点

VGG 有 3 个全连接层，耗费更多计算资源，并且使用了更多的参数，导致更多的内存占用。其中绝大多数的参数都是来自于第一个全连接层。且可能存在深度网络的退化问题。

D. 深度网络的退化问题

从经验来看，网络的深度对模型的性能至关重要，当增加网络层数后，网络可以进行更加复杂的特征模式的提取，所以当模型更深时理论上可以取得更好的结果，但是更深的网络其性能一定会更好吗？实验发现深度网络出现了退化问题：网络深度增加时，网络准确度出现饱和，甚至出现下降。

E. 残差学习

深度网络的退化问题至少说明深度网络不容易训练。现在有一个浅层网络，我们想通过向上堆积新层来建立深层网络，一个极端情况是这些增加的层什么也不学习，仅仅复制浅层网络的特征，即这样新层是恒等映射 (Identity mapping)。在这种情况下，深层网络应该至少和浅层网络性能一样，也不应该出现退化现象。目前的训练方法有问题，才使得深层网络很难去找到一个好的参数。

因此提出了残差学习来解决退化问题。对于一个堆积层结构（几层堆积而成）当输入为 x 时其学习到的特征记为 $H(x)$ ，现在我们希望其可以学习到残差 $F(x)=H(x)-x$ ，这样其实原始的学习特征是 $F(x)+x$ 。之所以这样是因为残差学习相比原始特征直接学习更容易。当残差为 0 时，此时堆积层仅仅做了恒等映射，至少网络性能不会下降，实际上残差不会为 0，这也会使得堆积层在输入特征基础上学习到新的特征，从而拥有更好的性能。残差学习的结构如下图所示。这有点类似与电路中的“短路”，所以是一种短路连接 (shortcut connection)。

残差单元可以表示为：

$$\begin{aligned} y_l &= h(x_l) + F(x_l, W_l) \\ x_{l+1} &= f(y_l) \end{aligned} \quad (2)$$

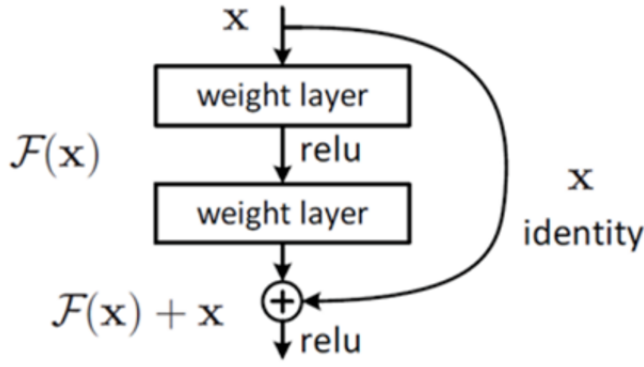


Fig. 7. 残差学习单元

其中 x_l x_{l+1} 分别表示的是第 l 个残差单元的输入和输出, 注意每个残差单元一般包含多层结构。F 是残差函数, 表示学习到的残差, 而 $h(x_l) = x_l$ 表示恒等映射, f 是 ReLU 激活函数。基于上式, 我们求得从浅层 l 到深层 L 的学习特征为:

$$x_L = x_l + \sum_{i=l}^{L-1} F(x_i, W_i) \quad (3)$$

利用链式规则, 可以求得反向过程的梯度:

$$\frac{\partial \text{loss}}{\partial x_l} = \frac{\partial \text{loss}}{\partial x_L} \cdot \frac{\partial x_L}{\partial x_l} = \frac{\partial \text{loss}}{\partial x_L} \cdot \left(1 + \frac{\partial}{\partial x_L} \sum_{i=1}^{L-1} F(x_i, W_i)\right) \quad (4)$$

式子的第一个因子 $\frac{\partial \text{loss}}{\partial x_l}$ 表示的损失函数到达 L 的梯度, 小括号中的 1 表明短路机制可以无损地传播梯度, 而另外一项残差梯度则需要经过带有 weights 的层, 梯度不是直接传递过来的。残差梯度不会那么巧全为-1, 而且就算其比较小, 有 1 的存在也不会导致梯度消失。所以残差学习会更容易。

F. ResNet 网络概述

多次训练之后, 准确率一直无法提升, 我们选择了 ResNet 网络作为改进模型。

ResNet 网络是参考了 VGG19 网络, 在其基础上进行了修改, 并通过短路机制加入了残差单元, 如下图所示。变化主要体现在 ResNet 直接使用 stride=2 的卷积做下采样, 并且用 globalaveragepool 层替换了全连接层。ResNet 的一个重要设计原则是: 当 featuremap 大小降低一半时, featuremap 的数量增加一倍, 这保持了网络层的复杂度。从下图中可以看到, ResNet 相比普通网络每两层间增加了短路机制, 这就形成了残差学习, 其中虚线表示 featuremap 数量发生了改变。图中展示的 34-layer 的 ResNet, 还可以构建更深的网络如表

1 所示。从表中可以看到, 对于 18-layer 和 34-layer 的 ResNet, 其进行的两层间的残差学习, 当网络更深时, 其进行的是三层间的残差学习, 三层卷积核分别是 1×1 , 3×3 和 1×1 , 一个值得注意的是隐含层的 featuremap 数量是比较小的, 并且是输出 featuremap 数量的 $1/4$ 。

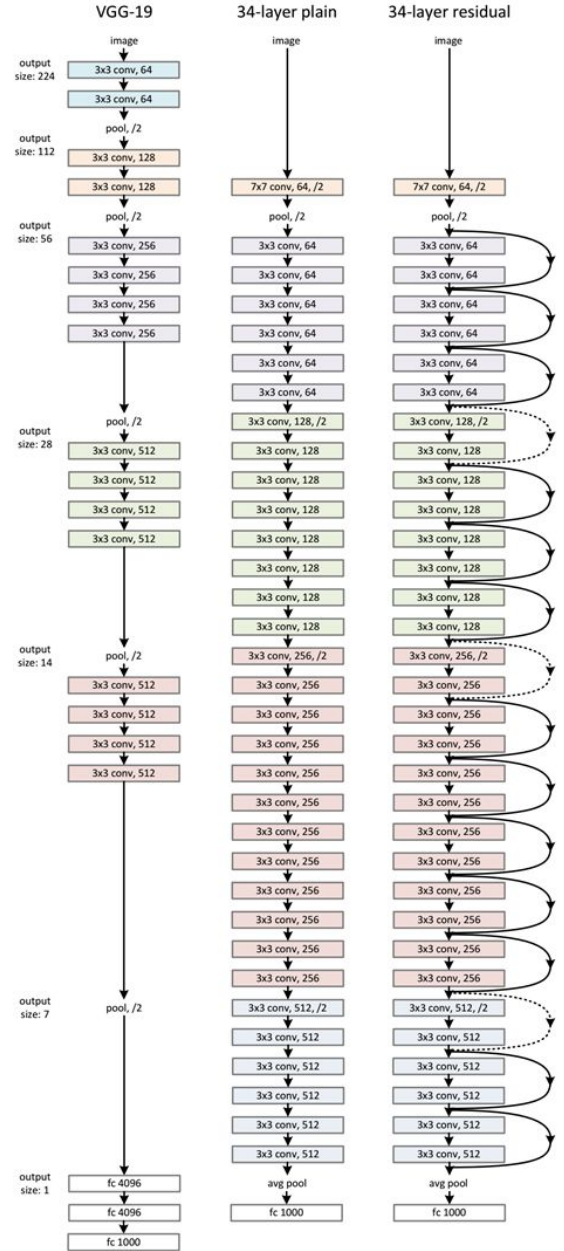


Fig. 8. ResNet 网络结构

G. ResNet 训练过程

ResNet 使用两种残差单元, 如图 10 所示。左图对应的是浅层网络, 而右图对应的是深层网络。对于短路连接, 当输入和输出维度一致时, 可以直接将输入加到输出上。但是当维度不一致时 (对应的是维度增加一倍),

layer name	output size	18-layer	34-layer	50-layer	101-layer	152-layer
conv1	112×112	3×3 max pool, stride 2				
conv2.x	56×56	3×3, 64 ×2				
conv3.x	28×28	3×3, 128 ×2				
conv4.x	14×14	3×3, 256 ×2				
conv5.x	7×7	3×3, 512 ×2				
FLOPs		1.8×10 ⁹	3.6×10 ⁹	3.8×10 ⁹	7.6×10 ⁹	11.3×10 ⁹

Fig. 9. 不同深度的 ResNet

这就不能直接相加。有两种策略：(1) 采用 zero-padding 增加维度，此时一般要先做一个 downsamp，可以采用 stride=2 的 pooling，这样不会增加参数；(2) 采用新的映射 (projection shortcut)，一般采用 1x1 的卷积，这样会增加参数，也会增加计算量。短路连接除了直接使用恒等映射，当然都可以采用 projection shortcut。

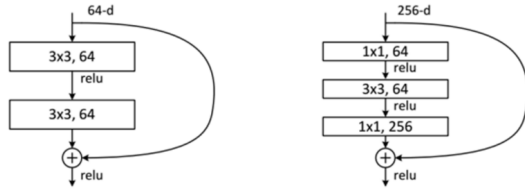


Fig. 10. 不同的残差单元

ResNet 通过残差学习解决了深度网络的退化问题，让我们可以训练出更深的网络，在训练效果上，ResNet18 模型比 VGG19 效果更优。

III. 结果分析

A. 混淆矩阵

混淆矩阵中，横坐标是预测标签，纵坐标是真实标签，正对角线上的数字即为模型预测正确的数量，而用颜色深浅则可以形象反应数据大小。训练过程中我们设置的初始学习率为 0.01，训练过程中学习率会以 0.9 为因子衰减。

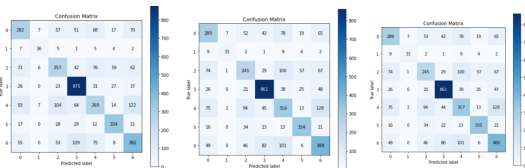


Fig. 11. 三种学习率下的混淆矩阵

1) 学习率为 0.01: 当学习率为 0.01 时，从图 11 可以看出，此模型对序号为 3 的表情识别能力较强，其次是序号 6，而对序号为 1 的情绪图像识别能力较弱。最高能辨别出 875 张序号为 3 的情绪图像。

2) 学习率为 0.009: 当学习率为 0.009 时，从图 12 可以看出，此模型对序号为 3 的表情识别能力较强，其次是序号 6，而对序号为 1 的情绪图像识别能力较弱。最高能辨别出 861 张序号为 3 的情绪图像。

3) 学习率为 0.0081: 当学习率为 0.0081 时，从图 13 可以看出，此模型对序号为 3 的表情识别能力较强，其次是序号 6，而对序号为 1 的情绪图像识别能力较弱。最高也能辨别出 861 张序号为 3 的情绪图像。

4) 两种模型: 综上所述，学习率越大，输出误差对参数的影响就越大，参数更新的就越快，但同时受到异常数据的影响也就越大，很容易发散。对比三个学习率情况下发现，数据波动并不大，序号为 3 的图像情绪在学习率为 0.01 时的识别能力最强，但总体来说，并无太大差距。最终我们得到 public, private 两份模型下的混淆矩阵如图：

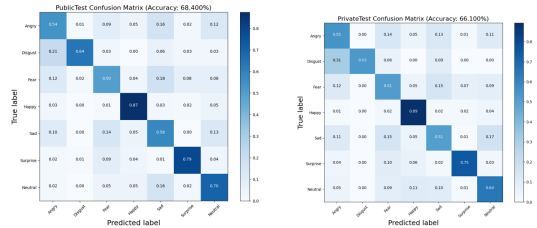


Fig. 12. PublicTest 与 PrivateTest 两份模型下的混淆矩阵

相比较下，public 训练出的模型在辨别 angry, disgust, fear, happy 情绪时的准确率较高，而 private 模型则在 sad, surprise, neutral 情绪图像中识别能力较高。

B. 损失函数和准确率

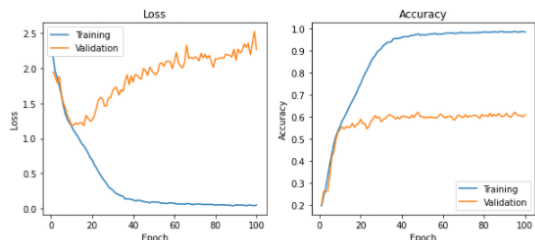
1) 交叉熵损失函数: 在二分类问题中，损失函数为交叉熵损失函数。对于样本 (x,y) 来讲，x 为样本 y 为对应的标签。在二分类问题中，其取值的集合可能为 0, 1，我们假设某个样本的真实标签为 y_t ，该样本的 $y_t=1$ 的概率为 y_p ，则该样本的损失函数为：

$$\log(y_t | y_p) = -(y_t * \log(y_p) + (1 - y_t) \log(1 - y_p)) \quad (5)$$

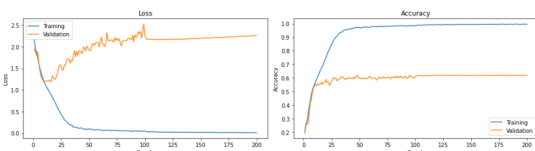
在多分类问题中，损失函数也是交叉熵损失函数，对于样本 (x,y) 来讲，y 是真实的标签，预测标签为所有标签的集合，我们假设有 k 个标签值，第 i 个样本预测为第 K 个标签的概率为 $p_{i,k}$ ，一共有 N 个样本，则总的数据集损失函数为：

$$L_{\log}(Y, P) = -1/N \sum_{i=0}^{N-1} \sum_{k=0}^{K-1} y_{i,k} \log p_{i,k} \quad (6)$$

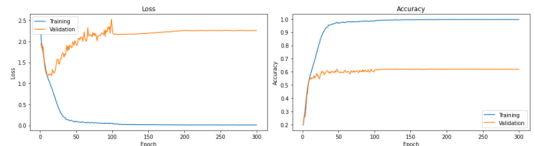
2) 准确率 Accuracy: 《统计学习方法》中指出, 分类准确率 (Accuracy) 的定义: 对于给定的测试数据集, 分类器正确分类的样本数与总样本数比。



当学习率为 0.01 时, 可以看出训练过程中准确率一直都在波动, 无法趋于稳定。



当学习率为 0.009 时, 可以看到训练数据在近 100 次迭代时, 准确率趋于稳定。



当学习率为 0.0081 时, 可以看到训练数据在近 100 次迭代时, 准确率趋于稳定。

相比较而言, 三种学习率下损失函数和准确率最后的收敛值差距不大, 但学习率为 0.00081 时, 收敛较快。

C. 小结

本次比赛的过程正如我们在机器学习这门课的探索, 机器学习这门学科的发展一样。我们首先根据自己掌握的 tensorflow 和 pytorch 深度学习框架的知识搭建了一个简单的卷积神经网络, 然后对网络进行了训练, 并且对训练结果进行了可视化。三次不同学习率下的最高得分约为 0.7。

接下来我们使用了大名鼎鼎的 VGG [1] 作为 backbone, 训练结果更加稳定, 准确率也更加高, 我们的得分约为 0.8。

不断地超参数调优过程中, 得分不再上升, 我们又参考文章改进了网络结构, 使用 ResNet [2] 作为 backbone, 得分终于达到了 0.85 以上。

IV. 比赛收获

- 1) 应用数学, 计算机的能力得到明显提高: 课后一直觉得知识很“虚”, 机器学习中充满了各种公式与推导, 却不清楚其中的奥秘所在。比赛过程中通过对经典网络的实现以及调试, 使我们更加深刻地了解到什么是卷积, 什么是计算机视觉, 要怎么样才能实现对图片的情绪分类。
- 2) 自学能力: 包括查找, 获取, 消化, 吸收并运用新知识的能力的提高。通过建立数学模型求解复杂问题: 实际问题的分析论证约化过程, 运用数学手段建立实际问题的模型, 求解模型并对结果进行分析、检验、解释。并学习使用了 LaTeX 将这一过程形成研究报告。
- 3) 对团队精神的真切体会, 良好的合作是通向成功的钥匙, 每个人都应该倾听他人的意见, 哪怕听起来不顺耳, 这特别是对那些满脑子充满了新鲜主意的队员尤其重要。辩论的目的是为了把题目做好, 争吵的目的是为了把论文写好。当自己错了的时候, 乐意地接受别人的意见; 当别人有道理的时候, 大度地容纳不同的观点。
- 4) 作为智能工程学院的一名学生, 看到学院能支持并举办这样的比赛, 我们非常地自豪。我们在这里获得了更多的知识, 更多的经验, 更多的能力, 更多的自信。希望每年都有许许多多的同学们在博弈杯中像我们一样, 在实践的过程中收获知识, 收获真正的归属感。

参考文献

- [1] Simonyan K, Zisserman A. Very Deep Convolutional Networks for Large-Scale Image Recognition[J]. Computer Science, 2014.
- [2] He K, Zhang X, Ren S, et al. Deep Residual Learning for Image Recognition[J]. IEEE, 2016.