

关于“A Circular-Structured Representation for Visual Emotion Distribution Learning”的实现

0 动机

随着智能通讯设备的不断普及，人们在社交媒体分享自己日常生活的所见所闻所感变得容易，人们分享的社交网络图片包含着丰富的情感信息。近年人工智能在模仿人的感知能力和认知能力取得很大的进展，目前图片的觉情感分析(VEA)也受到越来越多的人的关注。由于人的情感的激发是主观的，对于同一图片不同的人会有不同的感受，一张图片可能包含多种情感，然而不同情感之间也没有明确的边界，对于同一图片所表达的情感也很难给出明确的答案，相对于取最大值作为图片表达的情感来说或许分布式更能表达其存在的模糊性和不明确定，这篇文章所提出的“情绪圆(Emotion Circle)”能够将图片中包含所有情感的合成综合情感，优化情感的分布结果，想通过研究情感分析充分挖掘情绪圆模型的应用。这是我想研究的课题即怎样张成情感空间以及如何使用几个属性的向量来表述情感。

1 方法综述

情感是区分人类与其他动物的重要因素之一，它无时无刻的影响着我们日常生活。随着社交网络中流行，越来越多的人倾向于通过在网上分享图片来表达他们的感受。心理学家认为人有八种基本情绪：愉悦，敬畏，满足，兴奋，愤怒，厌恶，恐惧，悲伤。根据情绪极向性（积极和消极）、强度（情绪的强烈程度）和情感类别（八种基本情感）提出了一种基于圆形结构的视觉情感分布模型表示可以表达在不同的强度，可以是结合起来形成任何情绪状态。在上述研究的启发下，旨在有效地利用情感之间的内在关系及其内在的独特特征。具体地说，作者构造一个情感圈来统一其中的任何情感状态，包括基本情感和复合情感。在情绪圈中，每种情绪状态包含三个属性情感载体（即情绪极性、情绪类型和情绪强度）的两个性质，即相似性和可加性。如图 1 所示，基本情绪用一组在情绪圆（Emotion Circle）(b)上均匀分布的单位向量，对应圆形结构的基本八个情感(例如，愉悦，敬畏，满足，兴奋，愤怒，厌恶，恐惧，悲伤)在普拉切克的情绪轮盘。

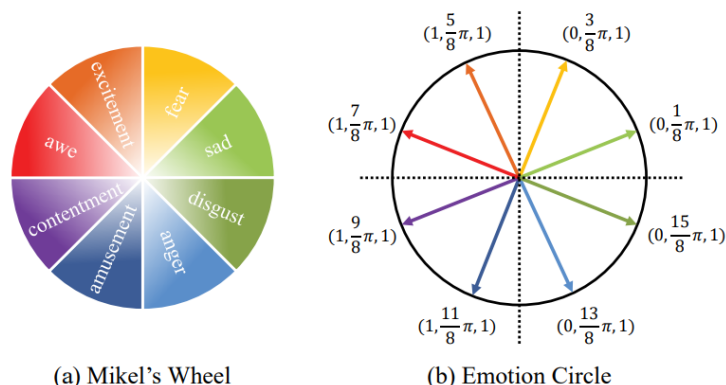


图 1. (a)心理模型中的普拉切克的情绪轮盘, (b)提出的情感圈及按照普拉切克的情绪轮盘均匀分布 8 个基本情感向量

情绪除了存在复杂的情绪状态(如情绪分布), 情绪也是有强度状态的, 文中提出一种新的情绪表达方法, 将情绪分布映射到情绪曲线上的复合情绪向量上。在细节上, 每个情绪分布首先被投射基本情感向量。根据其描述程度的不同, 具有不同的情感强度, 然后将加权的基本情感向量组合成复合情感向量, 可视为给定情感分布式标签。

本文利用情感的分布特性构建了情绪圆将其中的情绪状态统一起来。在所提出的情绪圆中, 每种情绪状态都用情绪向量 e_i 表示, 被描述为

$$e_i = (p_i, \theta_i, r_i) \quad (1)$$

p_i, θ_i, r_i 分别表示情绪极性、情绪类型和情感强度。

2 实施细节

2.1 网络结构构建

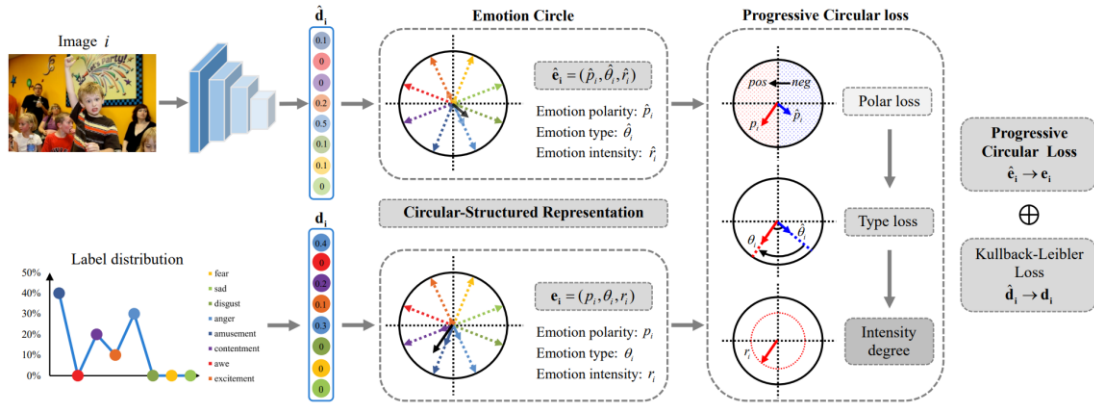


图 2. 环形结构方法框架。在提出的情绪圆上, 两者都预测了情绪通过系统的方法, 用复合情感向量表示分布和标记。然后我们提出从粗到细的循环损耗, 进一步利用循环损耗与 Kullback-Leibler 损耗一起训练网络。

网络的整体框架可以分为两个部分: 第一部分是图像情感训练预测网络 ResNet-50 部分; 第二个部分是情感用情绪极性、情绪类型和情感强度转化为情绪圆中所表示的向量问题并计算每个情感分布结果所对应的综合向量表示信息; 第三部分是损失函数的设计, 加速训练过程。

第一部分网络的搭建:

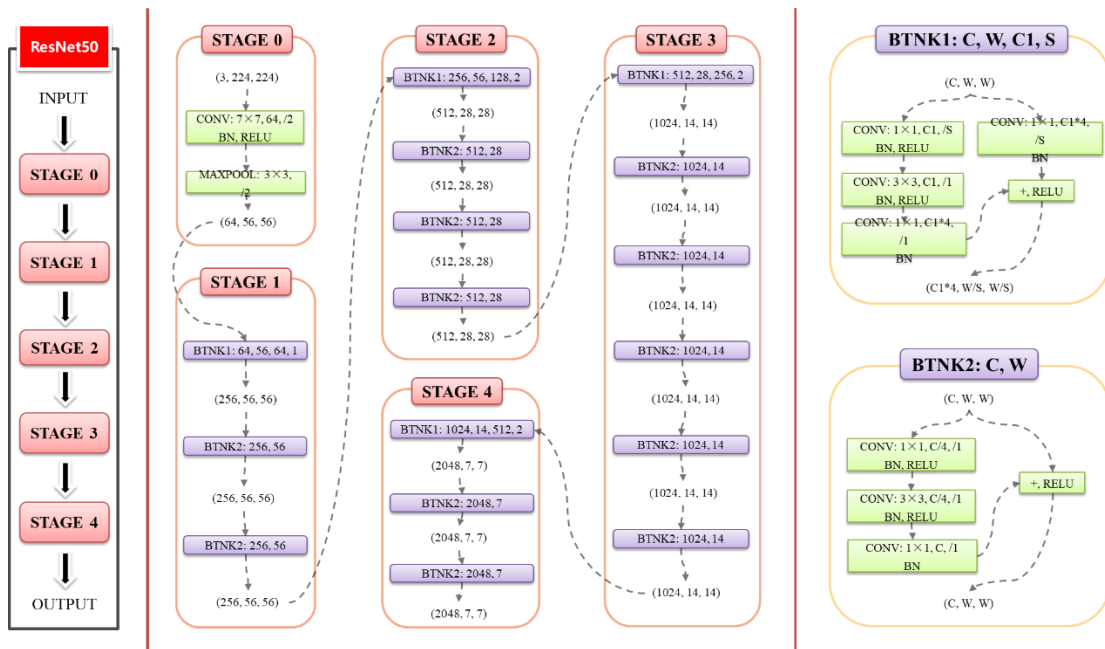


图 3. ResNet 50 网络结构图

这次网络的搭建工作采用之前搭建过的 ResNet18 和 ResNet34 经验和方法，来暴力的搭建 ResNet50，直接累加完成搭建，没采用 BasicBlock 和 BottleNeck，直接使用 forward 再加上前面的 DownSample 模型类函数，指定 ResNet50 所有参数构建模型。因此，工作量相对于采用 BasicBlock 和 BottleNeck 提高很多，后期会研究采用两个模块的方法，对此进行必要的改进。

经过手写 ResNet50 网络模型的暴力搭建，我认识到了要想把 ResNet50 复杂网络的搭建完成，前提必须要把模型整个流程环节全部弄清楚。例如，ResNet50 里面每一次的 shortcut 里面的升维操作的 in_channel, out_channel, kernel_size, stride, padding 的参数大小变化。

2.2 向量及损失函数构建

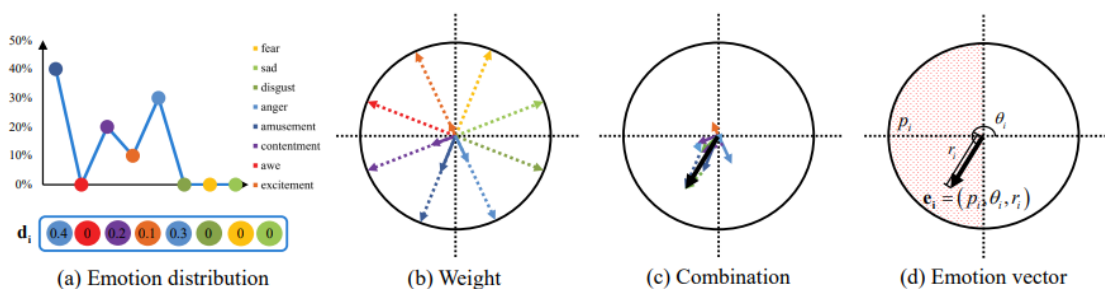


图 4. 映射到情绪圆上的复合情绪向量。(a) 8 种不同描述程度的基本情绪分布；(c)通过向量相加将其组合成复合情绪向量；(d)最终的情绪向量可以被视为给定情绪分布的特定圆形结构表示。

第二部分将概率分布信息转化成向量信息；这一部分主要是为损失的计算提供数据，通过模型优化损失函数提高测试精确度。第三部分损失函数部分的设计，重点放在理解清楚数据之间的计算方式，对数字进行精确的计算，实现文中的复现结果信息。

Kullback-Leibler (KL) [18] loss for visual emotion distribution(视觉情绪分布)的损失(loss for visual emotion distribution):衡量预测分布与标记分布不一致所造成的信息损失:

$$L_{KL} = -\frac{1}{N} \sum_{i=1}^N \sum_{j=1}^C d_i(j) \ln \hat{d}_i(j) \quad (1)$$

$$\hat{d}_i(j|f_i, W) = \frac{\exp(w_{i,j}f_i)}{\sum_{j=1}^C \exp(w_{i,j}f_i)} \quad (2)$$

d_i 表示标记的情绪分布从何而来 \hat{d}_i 表示预测的数据集。 N 表示特定数据集集中的图像数量, C 表示涉及的情感类别。

提出了一种渐进损失(Progressive Circular, PC)来从粗糙中学习情感分布, 在情感圆中是为了惩罚两个情感向量之间的差异, 即被标记的向量 $\mathbf{e}_i = (p_i, \theta_i, r_i)$, 预测结果为 $\hat{\mathbf{e}}_i = (\hat{p}_i, \hat{\theta}_i, \hat{r}_i)$ 。来具体来说, 我们逐步建立了三个约束条件情绪向量的属性, 即情绪极性(p_i), 情绪类型(θ_i)和情绪强度(r_i)。极性是情感的大方向的属性, 我们先设计极性损失来衡量情绪极性之间的差异, 通过实现均方误差(MSE)损失:

$$L_p = \frac{1}{N} \sum_{i=1}^N (p_i - \hat{p}_i)^2 \quad (3)$$

其中 p_i 表示所标记的情绪极性, \hat{p}_i 表示所预测的情绪极性。

情绪极性预测, 这是我们迈向的第一个目标准确的情绪预测。极性角之间的距离可以度量不同情感类型之间的相似性。于是, 分了情绪之后在两个极性中, 我们建立了一个更细粒度的情感类型约束, 称为类型损失:

$$L_t = \frac{1}{N} \sum_{i=1}^N (\theta_i - \hat{\theta}_i)^2 \quad (4)$$

在式 (4) 中, 两个极坐标角越近, 则越相似之处在于他们的情感状态。然而, 仅仅考虑情绪类型是足够的, 因为情绪强度也被认为是实现目标的一个关键因素一种特定的情绪状态。假设有两个具有相同情绪类型的图像, 即 $\theta_{i_1} = \theta_{i_2}$, 而且相当不同的情绪强度, 即 $r_{i_1} = 1$, $r_{i_2} = 0.01$, 它很难把他们归为同一种情绪状态。因此, 作为情绪类型 θ_i 和情绪极性 p_i 的置信度, 我们进一步添加情感强度国际扶轮对所提出的损失函数进行了精确的和求视觉情绪分布学习的详细约束条件:

$$L_{PC} = \frac{1}{N} \sum_{i=1}^N r_i ((p_i - \hat{p}_i)^2 - (\theta_i - \hat{\theta}_i)^2) \quad (5)$$

基于情绪圈, 我们的 PC 损失是最终的由公式 9 中一个从粗到精的人的三个约束构成。我们的损失函数与两个 KL 积分损耗和 PC 损耗加权组合:

$$L = (1 - \mu)L_{KL} + \mu L_{PC} \quad (6)$$

其中 μ 是一个超参数, 平衡两个损失之间的重要性, 本文在训练过程中 $\mu=0.7$ 。

2.3 网络多任务-分类数据集训练

网络多任务学习是网络学习过程新的热点, 能够在同一个网络模型中实现多

种任务。在情感分析任务中网络分布学习任务能够帮助我们理解图片信息中情感的倾向性的数据分布，然而情感分析数据集中也存在较多单类别标签的数据集。在网络任务学习过程中分类数据是分布数据的一种特殊方式，分类数据表示分布数据的较大数量的列别。现有研究中很少考虑将分类数据和分布数据在同一个网络中进行训练，应用同一个情感模型实现分类任务和分布任务的学习。

在本文提出的模型中使用训练训练网络及参数对分类情感数据集 FI-8 数据集在多任务学习过程中心理模型情绪圆环可以对更好的优化分类损失函数，心理模型提出的渐进损失函数可以从情绪的极向和情感类型两个方面优化损失函数，能够帮助网络学习过程更快的收敛。本文从将分类任务在分布网络中应用通过情绪圆心理模型优化测试精度。

2.4 运行参数细节

文中的网络结构部分建立在 ResNet-50，并使用在大尺度视觉识别数据集 ImageNet 预训练参数，输出的最后一个全连接层的尺寸更改为情感数字根据数据集，本文增添了不同的骨干模型对比 ResNet 101 实验结果。也对于除了分布数据集 Twitter LDL，Filcker LDL 进行了训练测试。在分类数据集 FI-8 数据集上进行训练，渐进损失函数帮助网络快速收敛，训练精度相比于分布数据集更加快速。数据集随机分成训练集(80%)和测试集(20%)，对于训练/测试集将每张图像的较短一侧调整为 480，将其裁剪到 448×448，并水平翻转增强数据集。整个网络训练过程使用 Adam 优化器并采用端到端的方式结合 KL 损失和 PC 损耗。当 Adam 的权重衰减为 5e-5 时，Adam 的值由从 1e-5 每 10 个 epoch 缩小 10 倍，最大的 epoch 数值设置为 50。我们的框架使用的是 PyTorch 实现应用操作环境 Ubuntu18.04、GPU 为 NVIDIA GTX 3090。

3 实验结果

3.1 分布数据集实验结果

初步实验训练结果及参数调整后的结果。

表 1. 分布数据集网络训练结果

Dataset	Backbone	Chebyshev↓	Clark↓	Canberra ↓	KL↓	Cosine↑	Inter↓	Acc
Twitter_LDL	ResNet-50	0.235	0.852	0.779	0.444	0.879	0.710	76.705
	ResNet-101	0.234	0.768	0.877	0.668	0.843	0.677	75.561
Filckr_LDL	ResNet-50	0.238	0.778	0.937	0.690	0.843	0.687	65.561
	ResNet-101	0.236	0.775	0.886	0.665	0.873	0.696	66.954

3.2 分类数据集实验结果

提出将分类任务应用到情绪圆环的任务中，在训练过程中主要通过调节超参数 lr_sgd 和网络的骨干模型，骨干网络选用 ResNet-50 和 ResNet-101 两种，从训练结果可以看下出 ResNet-101 相比于 ResNet-50 更快收敛，测试的精度也相对较高。测试精度变化趋势图如图 5：（a）网络骨干模型为 RestNet-50；（b）网

络骨干模型为 RestNet-101。在 lr_sgd 超参数选择中 lr_sgd=1e-3 时收敛更快，精度也更高。

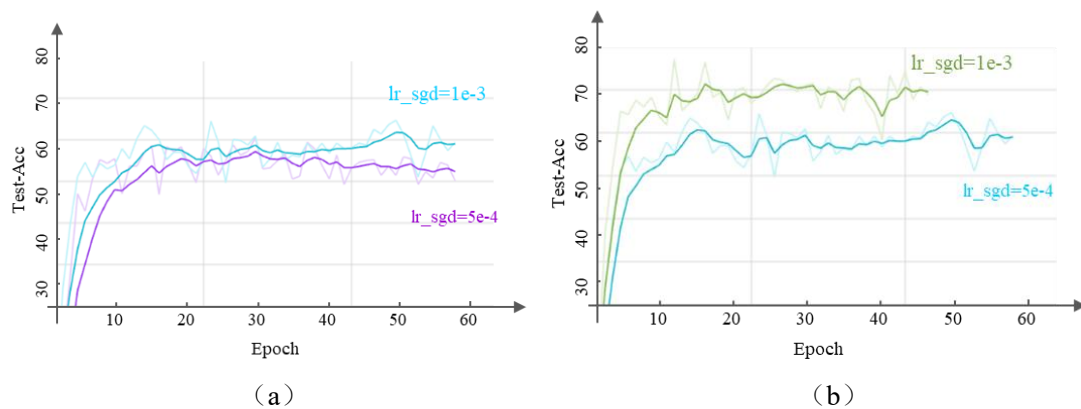


图 5. 分类数据集 FI-8 网络训练过程 Test-Acc 趋势图。(a) 网络骨干模型为 RestNet-50；
(b) 网络骨干模型为 RestNet-101。

表 2. 分类数据集 FI-8 网络训练结果

Backbone	Batch size	lr_sgd	Chebyshev↓	Clark↓	Canberra↓	KL↓	Cosine↑	Inter↓	Acc
ResNet-50	32	5e-4	0.372	0.949	0.913	0.990	0.733	0.628	69.47
		1e-3	0.386	0.949	0.914	0.965	0.734	0.618	69.05
ResNet-101	32	5e-4	0.369	0.949	0.913	0.978	0.736	0.631	69.47
		1e-3	0.346	0.949	0.911	1.005	0.743	0.654	70.71

4 UI 界面制作

UI 界面主要实现三个功能:能够实现 main 文件的运行;可以选择要训练的骨干网络来,包含两个选项 ResNet-50 和 ResNet-101 两个 Backbone,可以实现骨干网络的选择;最后就是把训练的信息打印在文本框中,实现实时的查看训练结果信息。

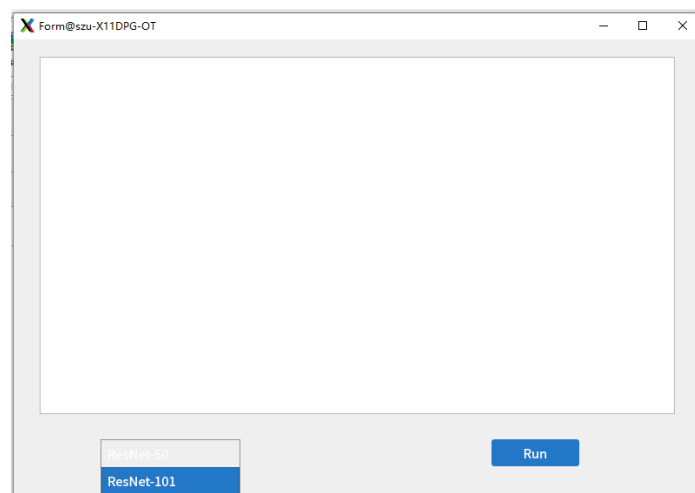


图 6. UI 界面 Backbone 选择框。

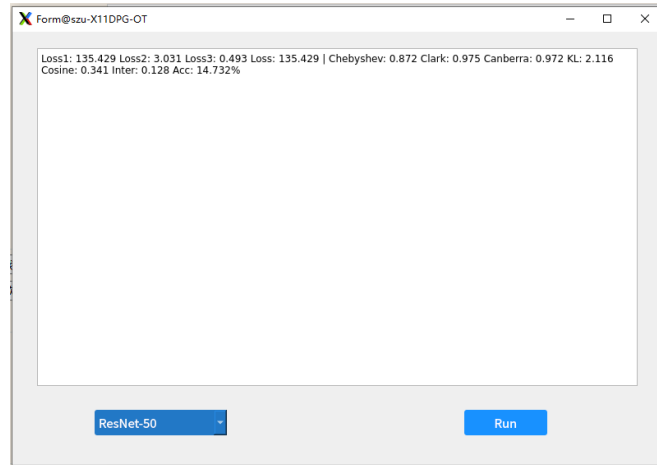


图 7. UI 界面运行后文本显示效果。

总结

这篇复现工作的主要内容分为三个部分：

第一部分：实现基本的复现工作，包括网络的搭建和前期的训练，在训练过程中增添骨干网络 ResNet-101 测试结果，训练结果相对于 ResNet-50 没有明显提高，在网络训练过程中的小尝试。

第二部分：将分类任务引入模型中应用，实现网络模型的多任务学习，训练结果也表现较好，说明基于心理学的模型能够帮助网络训练和学习。

第三部分：创建 UI 界面，能够实现三个基本的功能。

下一步工作计划，延续现有工作及现有问题继续深挖多任务学习的应用，增添更多的训练过程，找到比较优秀的超参数，并增添超参数的变化测试。