大作业报告

杨雨翔

自91, 2018012382

June 26, 2022

1 问题分析与形式化

本次大作业的题目是商品图片和颜色标签的匹配。和多分类问题不同的是，给定的图片数据没有固定的几个分类类别，而是和每个商品相关。具体来说，每个商品包含几张图片和几个分类标签，需要完成的任务是将商品和标签进行匹配。每个商品的标签各不相同，因此不能简单地作为多分类解决。 另一个要关注的问题是图片和标签分别属于视觉和文本两种模态的数据，因此在处理时也要考虑不同模态数据的特点。综上来看，这次作业主要是一个粗粒度的跨模态的匹配问题，可以形式化如下：

- **输入**：一组图片$I = \{I_1, I_2, \ldots, I_n\}$和一组文本标签$T = \{T_1, T_2, \ldots, T_m\}$
- **输出**：为每个图片$I_i \in I$分配一个标签$T_j \in T$。有前提条件$n \leq m$

2 数据处理流程

首先观察数据的特征。由于使用了model-based的方法，因此对数据的处理较少。这里数据的处理可以从图片和文本两个方面来考虑。

2.1 图片

观察图片的特征，发现由于数据来自淘宝，因此图片的主要特征——商品主要展示在图片的中央。商品的颜色和标签的匹配直接相关，但是只占据图片中央的一小部分。同时商家为了突出产品特征，往往会选择对比强烈的颜色作为背景色，有可能会影响模型的判断。因此为了帮助模型更好地提取图片特征，我使用 `torchvision.transforms.CenterCrop()` 函数对图片进行了裁剪。这里并不是对图片进行数据增强，因此在测试集仍然使用。 另外考虑到数据增强，需要对图片进行随机的处理。考虑到这次任务主要关注的是图片某个区域的颜色，而不是物体的形状和分类，因此使用裁剪、改变色调、`Channel Dropout`等都可能会影响算法的性能。最后选择了高斯模糊作为数据增强的方法，这样可以引导模型更多地关注图片的颜色而不是物体具体的形状。

2.2 文本

观察数据集中提供的文本标签，发现文本标签的风格和信息内容种类非常多。例如既有“黑

色”、“灰色”这种直接形容颜色的词语，也有“绿色上衣+粉色裙子”这种复杂的表达。或者是“杏色(付款后30天内发货)”这种包含大量无关信息的标签。为了提高标签的精确性，首先对结尾多余的信息进行裁剪。观察到文字标签的结尾往往包含一些重复的信息，因此对每个商品里面的标签从末尾开始检查，去除所有标签公共的部分。另外标签中的“色”字不包含任何信息，因此也将其裁剪。

3 算法原理

考虑到实现的简便性，我使用了model-based的方法，通过端到端的神经网络完成匹配任务。下面从特征提取器和匹配器两个部分进行描述。

3.1 特征提取器

由于使用了model-based端到端的方法，一个重要的环节就是寻找合适的特征提取器供下游任务使用。这里将图片和文本分开介绍。

3.1.1 文本特征

文本特征的提取使用了预训练的BERT[1]模型。BERT模型的主干网络类似GPT，都使用Transformer作为编码器，即使用attention代替了RNN对输入数据进行处理。BERT和GPT采用了类似的训练任务，即给定一个序列，让模型预测下一个位置的单词。和GPT不同的是，BERT使用了双向的任务，即给定一个序列，需要预测序列中空白位置的单词。这样双向的结构可以驱动模型学习上下文的关系，而不是单向任务中单纯的上文关系。另外BERT使用了MaskLM的方式进行训练，即使用特殊符号代替空白位置作为输入，这样可以保存待预测词的位置信息。同时考虑到下游任务不会出现这种符号，训练时需要以一定概率加入原单词和随机的单词。另外为了使模型学到句子之间的关系，还加入了一个连续性预测的任务。即给定两个句子，模型需要预测这两个句子是否是连续的上下文。 预训练的权重和词表均使用了Google公开的BERT-base-Chinese模型，它使用了中文维基百科作为语料库进行训练。 BERT的输出维度为($Batch\ Size$, $Seq\ Length$, $Feature\ Size$)。为了得到合适的特征向量，使用了`GRU`对特征进一步编码，使用最后一步的输出作为最后的特征向量。

3.1.2 图片特征

对于图片特征主要使用了`ResNet18`作为特征提取器，代码部分参考了经典的ResNet18[2]的结构并在其基础上加入了一些全连接层和 dropout 操作。ResNet18 包含 17 个卷积层，共分为 4 个 ResNet Layer。每个 Layer 包含两个 basic block，basic block 中又包含基本的卷积层，batch norm 层和 ReLu 层。Basic block 中的 shortcut 即为残差结构，通过 stride 参数可以调整残差的维度。 这里尝试了两种不同的方案。一种是直接使用`ImageNet`上预训练的ResNet18权重，而另一种是从头进行训练。从头训练也分为两种，一种是借鉴supervised contrastive learning，直接训练特征提取器。使用的Loss Function如下：

$$\mathcal{L}_{out}^{sup} = \sum_{i \in I} \mathcal{L}_{out,i}^{sup} = \sum_{i \in I} \frac{-1}{|P(i)|} \sum_{p \in P(i)} \log \frac{\exp\left(z_i \cdot z_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a / \tau\right)}$$

$$\mathcal{L}_{in}^{sup} = \sum_{i \in I} \mathcal{L}_{in,i}^{sup} = \sum_{i \in I} -\log \frac{1}{|P(i)|} \sum_{p \in P(i)} \frac{\exp\left(z_i \cdot z_p / \tau\right)}{\sum_{a \in A(i)} \exp\left(z_i \cdot z_a / \tau\right)}$$

其中$P(i)$是所有同类别的正样本。训练时不同商品的不同颜色作为不同的类别，训练的目标是使得正样本的距离尽可能的近而不同类别的样本距离尽可能地远。Loss Function的代码参考了原论文。 另一种是将ResNet和下游的匹配器一起训练，优化的目标即为总体的`CrossEntropyLoss`。

得到了初步提取出的特征之后，考虑到图片本身很大，而我们关注的只是其中商品占据的一小部分，因此对特征进行了进一步的处理。首先将特征拆分为n段，然后在这些片段之间执行`self-attention`，使得模型可以注意到特征中感兴趣的部分。最后得到的特征向量可以作为下游匹配器的输入。 最后使用两个MLP Layer使得所有的特征向量有相同的维度。为了统一不同模态间特征的数值范围，对所有输入的数据进行了`BatchNorm`处理。

3.2 匹配器

得到文本和图片的特征后，下一步是根据特征信息对其进行匹配。考虑到每次匹配发生在一个商品的图片和标签之内，图片和标签的绝对信息没有那么重要，更重要的是图片和标签之间的相对关系。例如同一个“黑色衣服+白色裤子”的组合，如果售卖的商品是上衣，就应该和黑色匹配。而如果售卖的商品是裤子，就应该和白色匹配。因此一个图片匹配哪个标签很大程度上取决于其他图片的信息。同理，文本的信息也和其他标签的特征有关。

考虑到这一点，在进行标签匹配之前，首先在标签的特征和图片的特征之间进行$self-attetion$操作。具体来说，给定一个文本或者图片向量x，分别计算映射$x_{key} = W_{key}x, x_{query} = W_{query}x, x_{value} = W_{value}x$。其中$W_{key}, W_{query}, W_{value}$均为线性映射。然后根据$x_{key}, x_{query}$计算`attention weights`。

$$Weights^{i,j} = \frac{\exp x_{key}^{j} \cdot x_{query}^{i}}{\sum_k \exp x_{query}^{i} \cdot x_{key}^{k}}$$

最后根据计算的权重，使用带有残差连接的MLP更新所有特征向量,记为:

$$x^{i:(t+1)} = x^{i:t} + f_{node}(x^{i:t}, \sum_k weight^{i,k:t} \cdot x_{value}^{k:t})$$

使用这种方法更新图片特征和文本特征后，下一步需要对二者进行匹配。同样地，这个匹配过程需要考虑到每个标签和每个图片的得分。这里使用图片和文本特征的`cross attention`来计算图片和文本之间的关系。 首先计算`attention weights`，考虑到不同模态特征间的差异，这里没有直接使用Cosine 作为特征向量之间的距离，而是使用了一个MLP Layer进行计算。

$$Weights^{i,j} = \frac{\exp\left(f_{edge}(x_{query}^{i}, x_{key}^{j})\right)}{\sum_k \exp\left(f_{edge}(x_{query}^{i}, x_{key}^{h})\right)}$$

计算得到的`attention weights`可以作为二者匹配的概率。然后使用计算的权重，再次使用带有残差连接的MLP更新双方的特征向量,更新方式和上述类似。

进行过一次`self attention + cross attention`之后，每个图片和文本的特征向量都得到了更新，将其视为一个单元。使用堆叠的三个单元对数据进行处理，即前一级输出的更新过的特征作为下一个单元的输入。注意到`cross attention`计算的`attention weights`可以作为图片特征和文本特征的匹配程度，因此直接取最后一个单元的注意力权重，将其在文本信息的维度做一

次Softmax，即可得到图片对每一个文本标签的分类概率。此时每个商品内可以看作一个多分类问题，使用CrossEntropy+Adam对其进行优化。

这个问题实际上可以建模为二分图的匹配，每个图片或文本的特征向量可以看作图上的一个节点，节点之间可以视为全连接的。这部分代码参考了一个特征匹配项目SuperGlue[4]。

4 实现过程和结果

具体实现时对不同的方法都进行了尝试。所有的实验记录可以参考https://wandb.ai/ethanyang/course_project?workspace=user-ethanyang。

4.1 使用预训练模型

首先尝试使用预训练模型作为特征提取器，直接训练下游的匹配器。在验证集上调参后，模型最佳的训练结果如下：

Figure 1: 使用预训练模型的结果

图中是模型在训练数据上的总体正确率的曲线。后面曲线接近平滑，但是匹配率只有90%左右，说明模型的表达能力不够。因此考虑只使用BERT作为文本的特征提取器，将图像的特征提取器也加入训练。

4.2 预训练图片特征提取器

使用supervised contrastive learning对ResNet18进行训练，得到的训练曲线如下。这里loss funtion参考了原论文的代码[3]

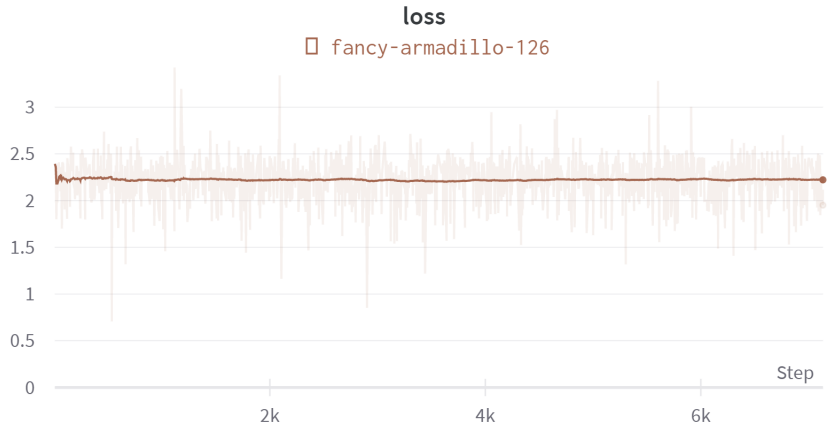

Figure 2: 预训练ResNet的结果

发现训练效果并不好，使用得到的ResNet权重对下游任务finetune，也没有得到较好的效果。因此选择直接对两个模块同时进行优化。

4.3 同时训练特征提取器和匹配器

使用较小的学习率同时对两个模块进行优化，得到的曲线如下：

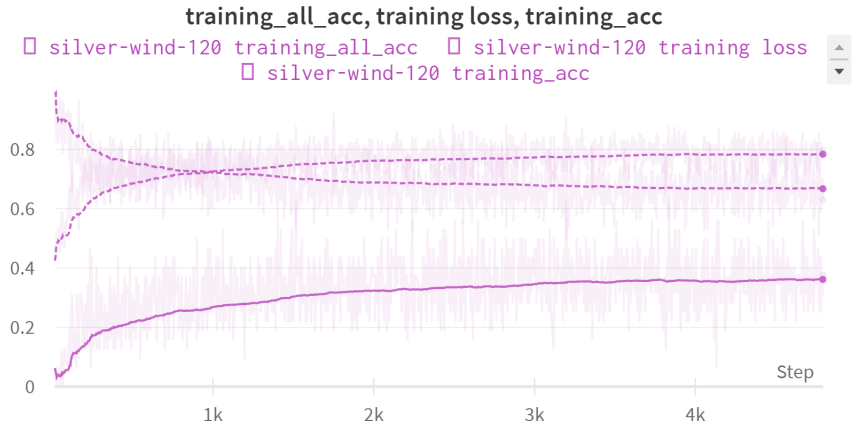

Figure 3: 同时训练两个模块的结果

曲线中展示了总体正确率`train_acc`，全商品匹配率`train_all_acc`和损失函数`loss`。可以看到训练的效果也不是很好。考虑到同时优化两个模块非常困难，因此将此时的ResNet保存为checkpoint，然后固定住ResNet此时的权重，finetune下游的匹配器。得到的曲线如下：

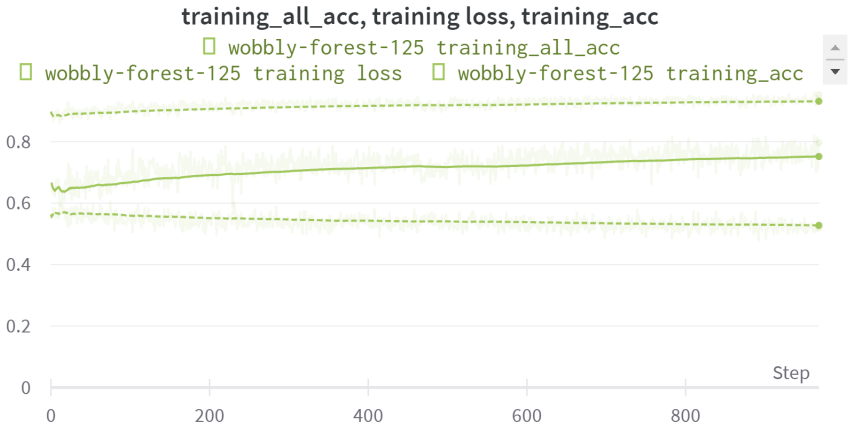

Figure 4: 使用前一步特征提取器finetune的结果

可以看到在训练数据上达到了0.95的总体准确率和0.79的全商品匹配率。

4.4 实验结果

最后使用finetune的匹配器和整体训练得到的ResNet进行推理。在测试集上的结果为整体准确率0.9055，全商品匹配率0.7006。ResNet的权重文件为`https://cloud.tsinghua.edu.cn/f/4c73e11949514252b447/`，匹配器的权重文件为`https://cloud.tsinghua.edu.cn/f/1a6c378bdf7c402ca5f1/`

5 结果分析

观察模型的结果，发现首先模型的表达能力有所欠缺，训练集上的完全匹配率也只在80%左右，因此可以尝试提高模型的复杂度。可能的方法有改变下游的匹配器模型，使用更复杂的GNN结构来完成图匹配的任务。比如有些工作使用了分层的网络结构进行图匹配，从coarse-to-fine的角度来更好地提取两个图之间的关系[5]。另外对于图片特征提取器，我认为使用contrastive learning对其进行预训练是一个很好的办法。由于我不熟悉这方面的内容，因此初次的尝试失败了。不过后续可以尝试使用不同的算法和超参数继续训练ResNet。而且也可以使用一些rule-based的方法对图片进行预处理，例如尽量裁剪掉每个商品的图片之间相似的部分，这样可以更好地帮助网络学习图片的特征。对于文本向量，由于预训练权重来自中文维基百科，因此有关颜色的词语可能出现较少，导致特征提取效率不高。可能的解决方法有使用小样本语料上预训练的权重，或者是使用其他较小的transformer-based的网络自己进行预训练。同时尽管对文本数据进行了清洗，但是文本之间仍然存在一些冗余和干扰的信息。可以进一步对文本的信息进行统计，然后针对性地进行更精细的数据清洗。

另外考虑到测试集上的准确率，模型也发生了一定的过拟合的现象，因此可以对文本和图像的数据进行数据增强。考虑到文本的特征，我们可以随机打乱文本的顺序，这样并不会影响词语的意义。另外也可以人为加入一些冗余的信息，模仿数据集的特征，帮助模型克服干扰。 对于图片，除了高斯模糊以外也可以尝试其他不改变颜色特征的增强方式，例如旋转，裁剪等。也可以尝试自己合成一部分数据集。例如使用几何图形代表商品，然后根据预先定义的颜色集合对图形

进行上色，最后在预先定义的颜色集合中加入一些冗余的文本模拟数据集中不同商品的情况。

References

[1] Jacob Devlin et al. “BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding”. In: 2018. URL: https://arxiv.org/abs/1810.04805.

[2] Kaiming He et al. “Deep Residual Learning for Image Recognition”. In: *CoRR* abs/1512.03385 (2015). arXiv: 1512.03385. URL: http://arxiv.org/abs/1512.03385.

[3] Prannay Khosla et al. “Supervised Contrastive Learning”. In: *CoRR* abs/2004.11362 (2020). arXiv: 2004.11362. URL: https://arxiv.org/abs/2004.11362.

[4] Paul-Edouard Sarlin et al. “Superglue: Learning feature matching with graph neural networks”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2020, pp. 4938–4947.

[5] Jiaming Sun et al. “LoFTR: Detector-free local feature matching with transformers”. In: *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 2021, pp. 8922–8931.