Generative Multi-modal Models are Good Class-Incremental Learners

论文动机

方法

在该文章中，作者将我们的自然语言处理(NLP)与图像识别结合在了一起，通过生成式多模态模型的方法来解决我们的增量学习问题。优点在于，该方法可以利用文本和图像之间的丰富的语义对应关系，而且还不需要为每一个新的识别任务扩展新的分类器。

那为什么我们要使用我们的分类器去对图片先进行分类，而不是直接提取我们图片中的特征再进行分类呢？这是因为，如果我们选择使用特征进行学习的话，我们的这些特征就会加剧我们的训练中的遗忘性，导致模型的不稳定。但是使用分类器直接对文本分类就减少了这种影响，因为我们的识别结果是基本不与其他的物品重复的，这样就可以减轻我们的遗忘。

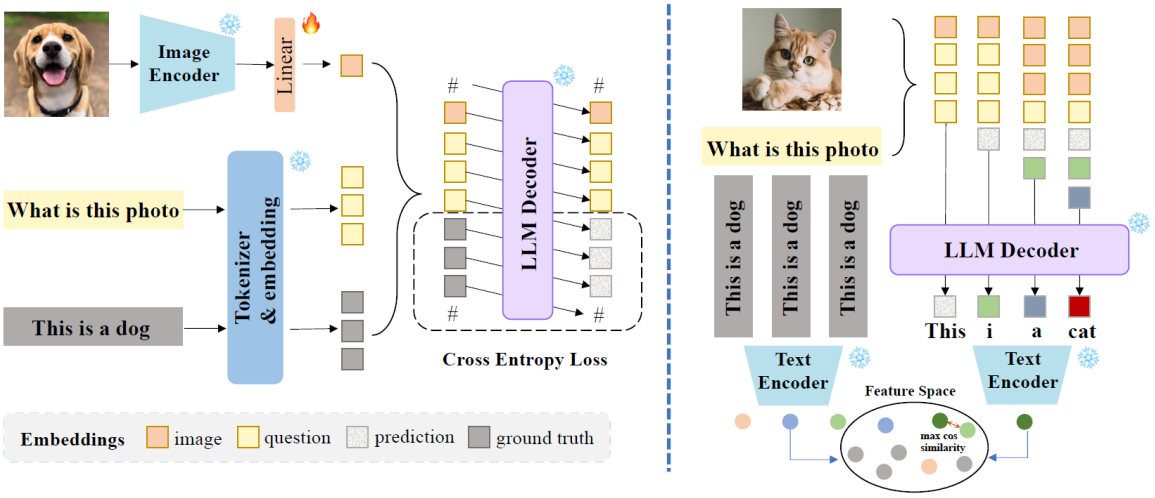
所以本篇论文作者解决的就是，如何将类增量学习和生成式多模态模型结合在一起，并且取得一个较不错的训练结果。下面我们详细地解释一下该篇文章使用的方法。

我们所需要解决的最重要的部分有两个。

第一个问题就是，我们如何将GMM用于分类。在训练过程中，我们使用真实标签的文本，鼓励模型以“This is a photo of [CLS]”的格式，用简洁准确的句子来预测图像的标签，避免对图像中的所有内容进行详细描述。这样我们就不会将一些物品的描述重合过多，就可以减少我们的遗忘性问题。在测试期间，模型遵循该格式输出给定图像的类别文本。我们提取“[CLS]”中的内容，然后使用我们的文本编码器f_{text}获得其文本特征，并计算到目前为止所看到的所有类别的文本特征的距离。然后将最接近的类视为生成模型的最终预测。

第二个问题就是，我们还需要转化我们的类增量学习去适应我们的基准。类增量学习通常在ImageNet、CIFAR-100和ImageNetR数据集上进行评估。这些数据集通常由图像和相应的单热标签$\{X_t, Y_t\}$组成。以CIFAR-100数据集为例，我们使用模板将每个图像与一个句子配对，形成图像-文本对格式$\{X_t, S_t\}$:“这是一张[CLS]的照片”，其中“[CLS]”是该类别的标签名称，例如苹果，狗等。

下图是该论文的主要流程，我们根据流程来介绍其中的过程。



首先，我们需要将我们的图片导入到我们的图像编码器Image Encoder中，被非线性化为特征矩阵Linear。下面的公式完成了该部分任务，我们的函数f_{enc}就是我们的图像编码器；我们的e_i就是我们的提取出来的特征值；我们的x_i就是我们需要训练的图片。

$$\mathbf{e}_i = f_{enc}(\mathbf{x}_i; \theta_{enc})$$

然后，我们需要将对该图像的提问与回答，放入我们的分词器Tokenizer和我们的嵌入层embedding中，也提取出我们的文本特征。下面的公式就完成了这一步。其中，bos代表的是句子开头的符号；e_i就是我们上面所说的提取出来的特征值；q就是我们的训练的图片；s就是我们的Tokenizer提取出的一个一个的特征词块；eos就是我们的句子末尾的符号。

$$\hat{\mathbf{e}}_i = \mathrm{CONCATE}(bos, \mathbf{e}_i, \mathbf{q}, \mathbf{s}, eos).$$

在提取完之后，我们将三者的特征值均放入我们的LLM Decoder，也就是大语言模型编码器中，计算交叉熵损失，然后就完成了我们的训练过程。

接下来讲一下如何进行测试，我们如何使用它来推测出图片的内容呢?

首先，我们提取出我们需要测试的图像的特征值，和我们的问题中包含的特征词块，将他们的特征值先嵌入在一起，导入到我们的LLM Deccoder中。我们根据先前训练出的Cross Entropy Loss，通过寻找cos值相差最小的特征词块，就可以输出我们的第一个词块的结果。

下面是我们计算交叉损失函数的过程，可以看出这跟我们课本中的交叉熵函数十分相似。

$$\mathcal{L}_{\mathrm{CE}} = -\frac{1}{m}\sum_{j=1}^{m} s_j \cdot \log \hat{s}_j$$

下面是我们寻找最接近的词块的公式:

$$pred = \mathrm{argmax} < f_{\mathrm{text}}(\mathbf{s}), f_{\mathrm{text}}(\hat{\mathbf{s}}) >$$

我们使用了我们的f_{text}函数，通过对文本进行特殊的特征化，再去计算我们最大的cos值，也就是向量角相差的最小($cos\ 0 = 1$)，这样就可以找出我们最接近的文本了!

再找完第一个文本之后，我们需要做的就是循环进行上面的流程，只不过我们中间需要稍微调整一些过程，我们需要将前面预测出的文本的特征值也加入到我们的预测模型中去，具体的实现方式如下公式:

$$P(\hat{s}_1, \hat{s}_2, \ldots, \hat{s}_m | \mathbf{x}_i, \mathbf{q}, \mathbf{s}) = \prod_{j=1}^{m-1} P(s_j | \mathbf{e}_i, \mathbf{q}, s_1, s_2, \ldots, s_{j-1})$$

这个公式的意思就是，我们计算每一个词块的条件概率，他由我们的前面的所有词汇的特征值所共同决定，也就是说，随着m的增长，式子后面的概率累加值就会受越来越多的变量所影响，这也与我们上面的原理图所一一对应，我们前面所预测出的每一个词块都会影响到我们后面的词块的生成。

然后我们就按照这个步骤一一进行预测，遍历到最后一个词块，就可以完成我们整个预测了。

实验结果

联系与区别

我们在认真阅读这四篇有关于连续学习的论文后，总结出了这几篇论文的一些联系与区别。

首先是联系。这四篇论文，其中有两篇都是出自南开大学刘夏雷老师实验室下的，在经过仔细的阅读之后，感触颇深。四篇论文都围绕着连续学习中最火的一个方向——增量学习来展开的，内容都是增量学习中的类增量学习(Class-Incremental Learning,CIL)。这几篇论文都致力于解决我们的连续学习中常见的一些问题，比如说灾难性遗忘和连续学习中存储空间不足等类似的问题。

这四篇论文的研究重点也是不太相同的，首先根据年份来看，我们的第一篇论文是提出了我们的类增量学习模型iCaRL，并且将其与当时的常规模型进行了对比；第二篇论文和第三篇论文都是在解决我们的类增量学习中容易遇到的问题，Learning without Memorizing这一篇主要解决的问题是，在学习的过程中，我们不需要去存储过去的一些特征值，而是直接通过我们的三个损失值去计算与前面的模版类的差距；Masked Autoencoders are Efficient Class Incremental Learners这一篇就是，通过遮盖我们的图像的部分，去减少我们的存储空间，利用我们带分类标签的训练目标去提高我们训练的鲁棒性和准确性；第四篇是近几个月刚刚发表的顶刊论文，主要的贡献是，将我们在视觉领域的连续学习与我们的大语言模型结合在一起，将图片的特征提取出文本，再进一步进行分类与标签，从而大大降低我们的特征重合性，很好地降低了连续学习中经常出现的灾难性遗忘问题。

问题描述

尽管近年来深度学习在很多单项任务上取得了相当或超过人类水平的成绩，这些深度学习模型都是为固定的任务所设计，无法动态地根据环境而更新。这意味着每当有来自新的分布的数据输入的时候，此类模型都需要同时在整个历史数据上重新进行训练。显然，在持续变化的现实情境中，进行这种数量级的训练是不现实的。

如果在加入来自新分布的数据时不在旧数据上做上述的重新训练，那么这些针对单一任务的模型很可能会在最近输入的数据上发生过拟合。这种现象被称为**灾难性遗忘 (catastrophic forgetting)**。如果数据本身就存在分布漂移，同样也会发生灾难性遗忘。事实上，灾难性遗忘是深度学习模型所面临的一个更基本的问题的结果：**“stability-plasticity” dilemma**。此处 plasticity 指的是模型融合新的知识的能力，而 stability 即为模型在学习新知识的同时保持就知识的能力，这两者我们在一般的深度学习中是不可兼得的。

连续学习 (Lifelong Learning)，即是一个模型在一个连续序列不同任务上学习的能力。该序列中的新任务通常与旧任务相关联。连续学习旨在防止模型的灾难性遗忘，能够在有效学习新任务的同时维持在历史任务上的表现。连续学习是一个模型适应快速变化的现实情景的关键，对于实现真正的人工智能十分重要。

在目前，连续学习主要分为以下三个方向。

一个是**Task-Incremental Learning**，也就是任务增量学习。在这个情景下，模型在评估时能够得知当前的输入来自于哪一个任务，是最简单的连续学习情景。一个针对 Task-IL 的模型通常会有 **“multi-headed”** 的结构，意味着每一个任务都有独立的输出层，同时其余的网络结构不随任务改变。

第二个就是**Domain-Incremental Learning**，也就是域增量学习。与 Task-IL 不同，模型无法在此情景下得知任务的类别，同时模型也不需要判断当前任务所属的具体类别。此情景下的任务都具有同样的结构，但有着变化的输入分布。一个相关实际例子是目标为学习在不同的环境下生存的模型，每一个环境下的结果都是生存和死亡两种，而模型不需要知道自己面对的具体是何种环境。

第三个就是**Class-Incremental Learning**，也就是我们本文主要研究的类增量学习。在 Class-IL 下，模型需要既能够成功完成给定任务，同时也要判别出任务的具体类别。常见的连续学习问题，如持续加入并学习新的任务类别，就属于这一情形。应用于 Class-IL 的模型使用 **“single-headed”** 的结构，即所有的任务都会使用一个统一的输出层。

这三个类型的学习都发展的十分迅速，本文主要对类增量学习进行深度的了解与研究。