Learning without Memorizing

论文动机

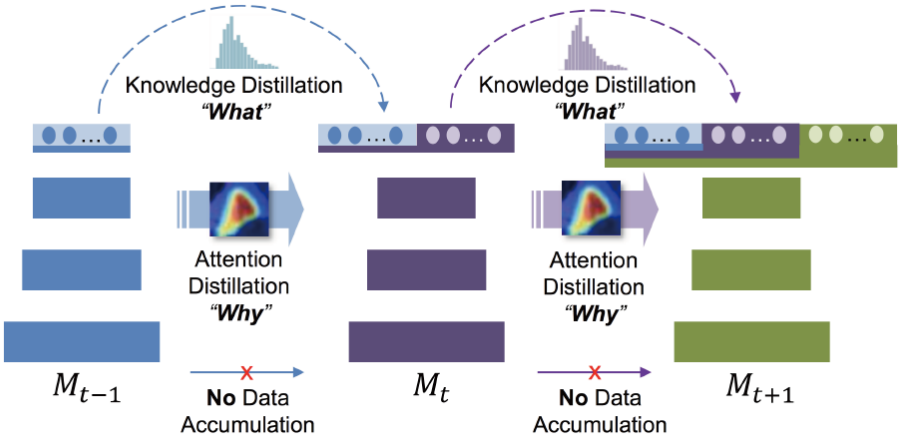
本篇文章关注的是对于增量学习目前存在的问题的解决与优化。以往的论文中，增量学习的解决方法一般是通过存储一小部分先前的旧样本类，但是这样的方法也有很大的缺点，比如说：如果数据量过大，我们尽管只存储一部分的旧类别的内容，但是依然会消耗很多的内存空间，不适合$life-long\ learning$；关于隐私问题，在工业中，数据的拥有者一般不会将先前的数据交给终端使用者，所以终端使用者是无法获得先前的旧类别的数据的；而且这种学习方式与人类的学习模式相违背。人类一般在学习新的知识的时候，不会去反复地学习与观察旧数据，所以我们急需一种新的方法来完成这种学习任务。

为了解决这一问题，本文提出了一种全新的方法，这种增量学习方法不需要接触我们的旧类别的样本数据，根据先前的论文，我们可以得到之前的解决方法是：采取传统的知识蒸馏的约束，意义是：对于一张新的图片I_n，我们尽可能去学习或者保留这张图片在旧模型中可能会被预测的旧类别。但是这种传统的方式的结果也不是特别好。

我们观察到，前面的图片与后面的图片并不是完全没有关联的。我们通过分析，发现在图中有一部分的区域是对于预测的结果有着很大的影响，比如说我们通过下面的图就可以发现，我们利用我们的方法，用神经网络准确定位到与类别相关的区域后，提取的特征才更具有区分性，常见的$distiilation\ loss$存在一个问题，当神经网络对某类别的注意力区域发生转移后，其大小并不会发生太大的改变，如下图所示。

挂盘电话的有区分度的区域为挂盘号码，挂盘电话下方的提示符并不能显示这是一个挂盘电话，分类器很有可能把这张图片划分其他类别，而常见的$knowledge\ distillation\ loss$对此并不敏感，因此我们提出了$attention\ distillation\ loss$，即注意力蒸馏损失，通过加入这一类别的损失来更好地完成我们的预测。这个值就是我们定义的注意力蒸馏损失，即L_{AD}，这一部分是本篇论文较新的一个点。

为什么我们加入这一部分的*loss*值就可以更好的预测呢？这是因为在新类别进行训练的时候，原方法相当于没有抓住首次训练时获得的信息，我们使用先前的*knowledge distillation loss*只能获得“*what*”，就是说，两张图片为什么像，但是却找不出两者相像的原因“*why*”。但是通过我们新加入的*attention distillation loss*就可以很好的预测出为什么两张图相似，下图就很好地说明了这一点。



方法

本文有两个很关键的问题需要解决，第一个就是如何定义或者生成注意力区域；第二个就是如何对注意力区域进行约束和限制，即对注意力区域的知识蒸馏。在这之前我们先了解一下该方法的历史背景。

第一个就是原先就被广泛使用的知识蒸馏，也就是L_D。作者给出了计算公式，其中$\mathbf{y}$和$\hat{\mathbf{y}}$是增长步长为t的M_{t-1}和M_t的基类预测向量。

$$L_D(\mathbf{y},\hat{\mathbf{y}}) = -\sum_{i=1}^{N} y_i' \cdot \log\left(\hat{y_i'}\right)$$

然后我们就开始去生成我们所需要的注意力区域，本文采用了$Grad-CAM$方法来生成注意力区域。首先，图片会输入到模型中进行前向传播，得到每一个类别的置信度y_c；然后我们对其进行反向传播，计算出每一个卷积层的梯度图；然后我们对获得的梯度图做*global average pooling*操作，得到我们每一层的置信值α_k；最后，我们记$\alpha = [\alpha_1, \alpha_2, \dots \alpha_k]$，$A = [A_1, A_2, \dots A_K]$，文章定义注意力区域的特征为

$$Q = ReLU(A\alpha^T)$$

此处我认为原文的公式有些许的错误，因为根据矩阵乘法的运算，原公式应该不能做矩阵乘法，所以此处我进行了修改；还有一个就是为什么要使用$ReLU$函数来进行运算呢？我认为是此处我们需要消除掉负元素。梯度值是正的的时候，随着X_j的变大，y_c才会变大，所以我们只需要关注那些正向变化的值就可以了。

第二个关键问题就是，在生成注意力区域后，我们如何计算注意力区域的知识蒸馏，即所需要的L_{AD}。根据文章中的方法，我们首先计算出输入照片的前后两张图对应时刻的注意力区域Q，分别记为$Q_{t-1}^{i,c}$和$Q_t^{i,c}$，具体公式如下所示：

$$\begin{aligned} Q_{t-1}^{i,c} &= \text{vector}\left(\text{GradCAM}\left(i, M_{t-1}, c\right)\right) \\ Q_t^{i,c} &= \text{vector}\left(\text{GradCAM}\left(i, M_t, c\right)\right) \end{aligned}$$

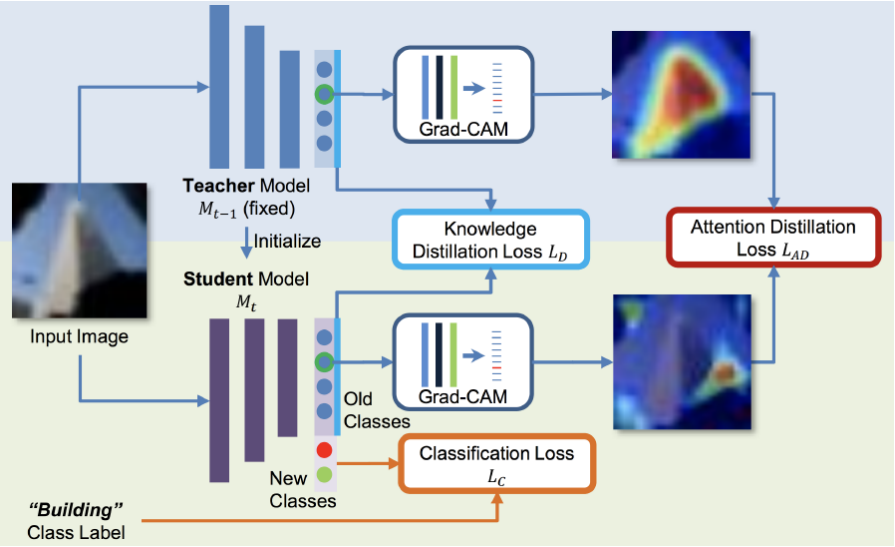
然后我们计算两者之间的权值差值，就可以得出L_{AD}，就是两者之间的注意力区域的蒸馏，使用以下的公式：

$$L_{AD} = \sum_{j=1}^{l} || \frac{Q_{t-1,j}^{I_n,b}}{||Q_{t-1}^{I_n,b}||_2} - \frac{Q_{t,j}^{I_n,b}}{||Q_t^{I_n,b}||_2} ||$$

最后，我们最终的*loss*的组成部分有三个，第一个是分类的损失，即L_C；第二个是传统的知识蒸馏损失，即L_D；第三个是本文提出的基于注意力区域蒸馏的损失，即L_{AD}。最后的*loss*的公式如下所示：

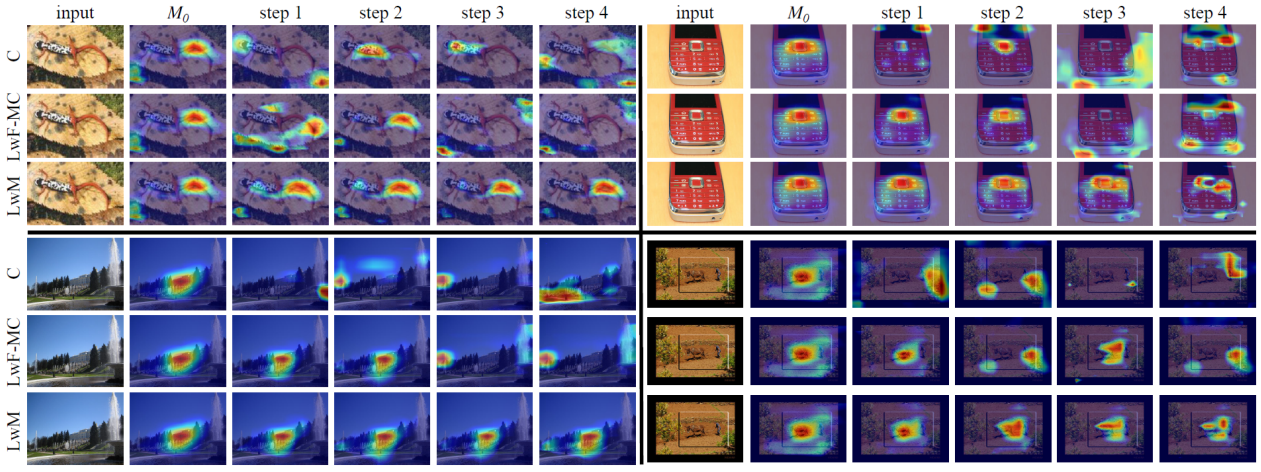
$$L_{Lwm} = L_C + \beta L_D + \gamma L_{AD}$$

实验总体的流程大概如下图所示。我们先用一些固定数量的类来训练出一个*teacher Model*，然后再分别加入新的类来进行训练，在原有的模型与新模型的对比中，计算出三个损失值，然后去计算总损失值，从而能够更好地定位图片的特征，可以使最后预测的结果相比于先前的方法更加准确！



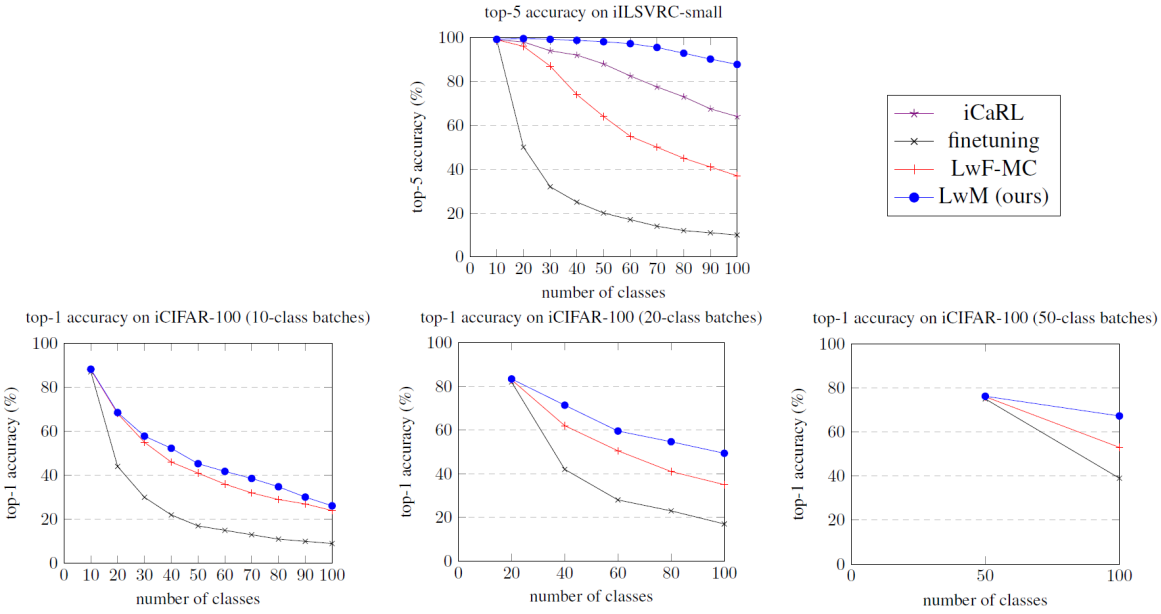
实验结果

作者通过使用自己的*loss*值和先前的方法计算出的*loss*值进行对比，发现确实是采用新方法获得的预测准确率要高一些，如下图所示。



从上图我们可以看出，面对不同的图片，采用这种新方法，都能够较好地对特征值的部分进行预测。

#Classes/Config	$L_C + L_{AD}$	LwM(ours)
20	84.95	**99.55**
30	55.82	**99.18**
40	43.46	**98.72**
50	36.36	**98.10**
60	26.78	**97.22**



由上面的表格和测试结果，我们可以看出，使用我们的模型来进行预测，准确率会比原先的方法高出不少。