

HW1-3 拼图模型

520030910393 马逸川

3.0 目录结构

- `Model.py`: 定义了拼图模型和 Resnet 模型。
- `Train.py`: 训练和评测主函数，定义了训练流程。
- `dataset.py`: 定义了切分后的 Cifar_10 数据集。
- `Args.py`: 集成了训练流程中的各项参数。
- `Pretrained_Resnet`: 存放预训练的Resnet模型参数

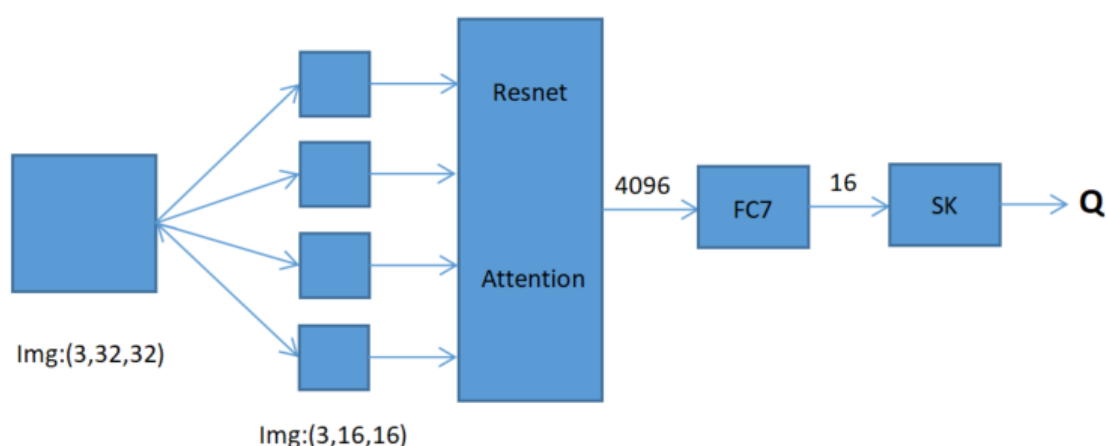
3.1 切分设计

切分设计较为简单，将图片均分为4份打乱排序，作为模型输入。

具体实现中将切分的部分添加在数据集中，通过打乱单位对角矩阵实现标签的随机分配。数据集的实现参考 paddle 框架内置实现的 Cifar_10 数据集。

3.2 模型结构

模型结构借鉴了参考论文的设计。



将切分后的四张图片通过 Resnet，得到长为256的维度特征，记录特征并对保存的(4*256*256)维度特征做 Self_Attention，得到的结果与原特征进行 Concat，最终得到2048维的特征向量。将特征向量输入 FC7 全连接层模型(7层全连接层)综合处理，输出16维预测值，reshape 为4*4大小的矩阵后作 sinkorn 操作，得到预测标签。

其中，Attention层的加入是考虑到不同部分之间存在对应关系，将这样的对应特征输入网络能够使拼图模型更好的识别不同图片特征。

3.2 训练参数和结果

训练参数在 `Args.py` 中定义，如下所示。

```
arg_parser.add_argument('--max_epoch', type=int, default=20, help='Type in the max epoch')
arg_parser.add_argument('--batch_size', type=int, default=256, help='Data batch_size')
arg_parser.add_argument('--seed', default=999, type=int, help='Random seed')
arg_parser.add_argument('--device', type=int, default=0, help='Use which device: -1 -> cpu ; the index of gpu o.w.')
arg_parser.add_argument('--lr', type=float, default=0.00005, help='learning rate')
arg_parser.add_argument('--data_path', type=str, default='./cifar/cifar-10-batches-py', help='Path of data')
arg_parser.add_argument('--hidden_size', type=int, default=256, help='CNN hidden size')
arg_parser.add_argument('--use_attention', type=bool, default=True, help='Use attention in Stitching Net')
```

是否使用Attention层	learning rate	Epoch	Test Acc
Yes	1e-5	20	84.95%
No	1e-5	20	83.60%

对比发现，添加Attention层确实带来了一定的性能提升。

3.3 使用预训练参数训练Resnet

将预训练部分的参数保存并加载到Resnet模型中，在Cifar_10数据集上训练Resnet完成分类任务。保存并加载训练代码的函数如下所示：

```
paddle.save(model.B1k_1.state_dict(), './Pretrained_Resnet/pretrain.pdparams')

.....

if use_pretrained:
    layer_state_dict = paddle.load('./Pretrained_Resnet/pretrain.pdparams')
```

结果与未使用与训练参数的结果对比如下：

是否使用预训练的参数	learning rate	Epoch	Test Acc
Yes	1e-3	30	82.78%
No	1e-3	30	82.13%

对比发现，使用预训练参数带来了一定的提升，因此可以认为使用拼图任务预训练Resnet模型能够使模型学习到一些特征，从而辅助其完成分类任务。

3.4 总结

总结来看，第一部分中，参考论文实现的模型结构在拼图任务上达到了较好的表现。更进一步地，通过对特征作Attention操作能够使模型更好的识别输入图像之间的关系，从而进一步提升性能。

在第二部分中，通过保存并加载第一部分的模型中提取特征的Resnet模型参数，训练得到的Resnet表现也有了一定的提升。

3.5 参考

- Paddle Documents
- [DeepPermNet: Visual Permutation Learning](#)
- [Attention is all you need](#)