Squeeze-and-Excitation Networks

开始之前

简介

- 顶会论文，发表于CVPR
- 2018 年
- 牛津大学 VGG 组
- 是 CVPR 2018 中引用数量最高的一篇论文

前期知识储备

- PyTorch
- ResNet、ResNeXt、VGG、Inception系列

学习目标

- SE Block（也叫 SE Module）
- Role of SE Block
- Ablation Study

论文背景及前置知识

背景、成果和意义

成果：

- ILSVRC 2017 分类冠军，在各种数据集上均超过了主流模型
- CVPR 2018 中引用数量最高的一篇论文

意义：

- 较早的**将注意力机制**放入了卷积神经网络中
- 并且该机制**是一个即插即用的模块**，可以嵌入到任意主流的卷积神经网络中
- 为当时的深度学习领域提供了一个新的思路——即插即用的模块设计

背景：

- 注意力机制

- STNet
- Attention ResNet（带有注意力机制的 ResNet）
- CBAM（也是一个注意力机制的模块）

注意力机制

注意力

注意力指的是人的心理活动指向和**集中于某种事物**的能力

也就是，当我们看向桌子的时候，总是会优先注意到桌子上最显眼的东西，而当我们仔细打量，才会发现有其他的东西

当我们在看向桌子的时候，我们的大脑会对我们感兴趣的区域给予更多的关注

注意力机制

我们想让网络具备「注意力」这一能力，需要通过另一个维度来理解卷积神经网络

注意力机制来源于人类的大脑，一开始是 NLP 领域所使用的，后来被 CV 领域所采用

从数学角度看，注意力机制就是**提供一种权重模式进行运算**

例如，通过设置权重来考察学生的综合能力，例如：$Chinese:Math:English=3:4:3$

小明的考试成绩分别 $120,100,120$，小红的考试成绩分别为 $100,120,120$

则小明的综合得分为 112，小红的综合得分为 114，所以**小红更加优秀**

在上面这个例子中，我们可以发现，权重的设置决定了将**更多的注意力**放在数学这一科目上，这直接导致了小红更加优秀这一事实

而在神经网络，注意力机制即利用**一些网络层计算得到特征图所对应的权重值**，对特征图使用**「注意力机制」**

相关研究：STNet

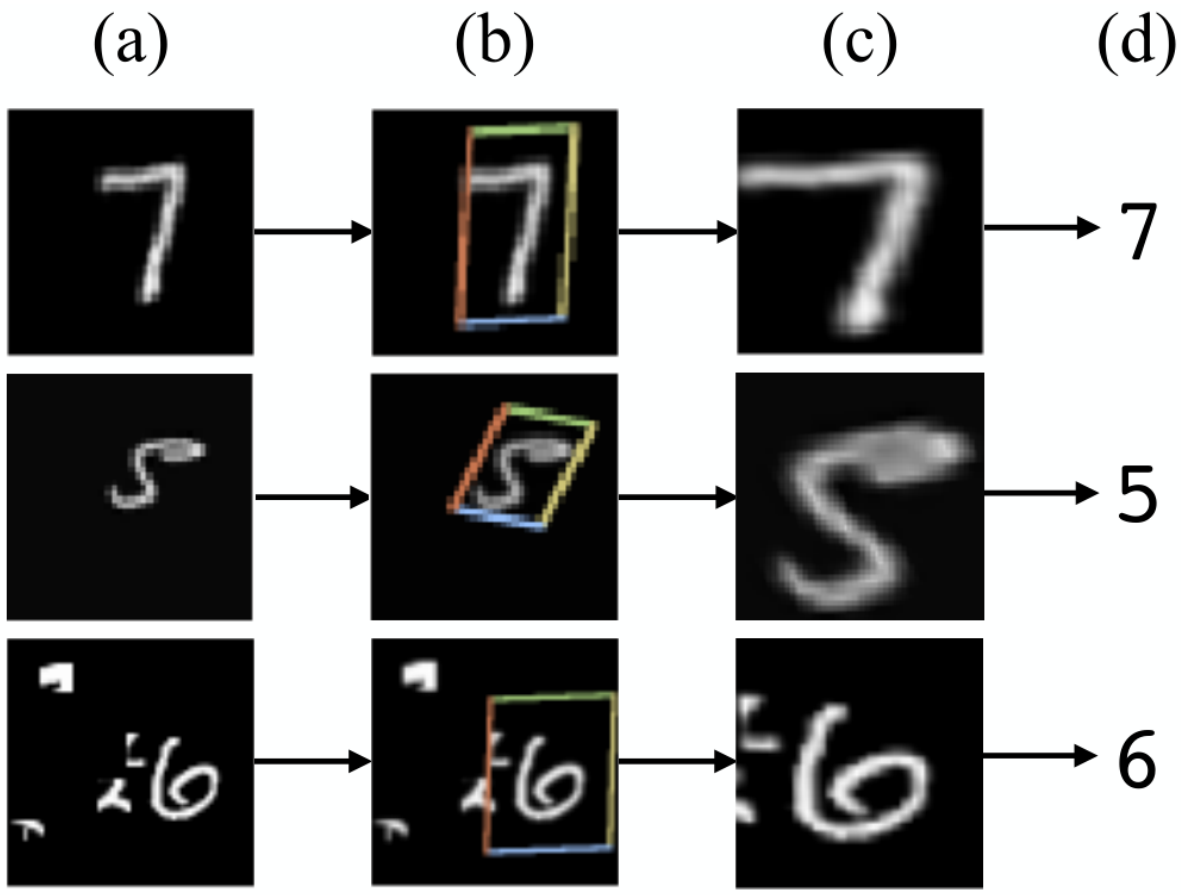
文章： Spatial transformer networks

定义及作用

STNet（Spatial-Transform Net），空间变换网络

提出 Spatial Transformer 模块，用于**增强 CNN 对图像的空间变换的鲁棒性**

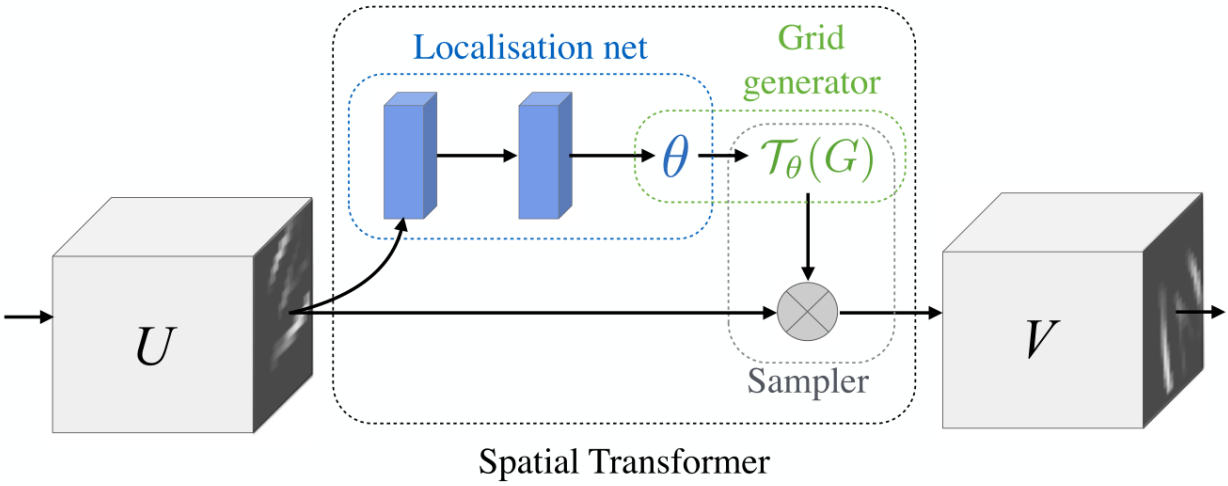
网络效果



如上图所示，STNet 通过 Spatial Transformer 将**已经形变的、位置偏离的**图像变换到图像**正中间**，使得网络对空间变得更加鲁棒

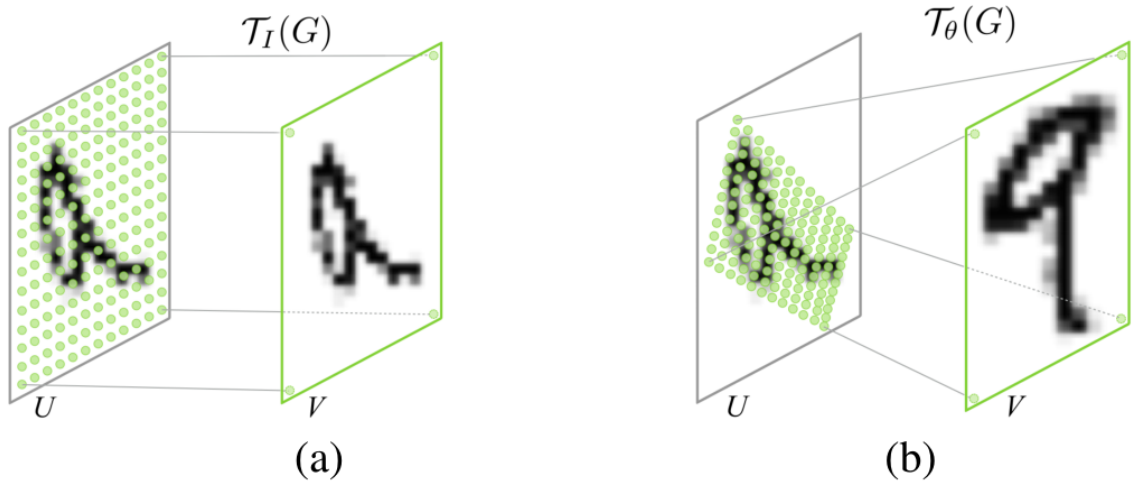
实现方式

STNet 的 Spatial Transformer 对特征图加入了三个子模块：Localisation net、Grid generator、Sampler



Spatial Transformer

前两个模块负责寻找特征矩阵（即建立一个映射关系，原图映射到处理后的图像），第三个模块通过特征矩阵进行「仿射变换」，来实现图像居中的效果



思路来源

STNet 考虑到 CNN 对平移不变性的支持是不足的，因此通过设计一个模块来变换特征图，从而让 CNN 可以适应更多的变换，例如 放大、旋转、平移 等

与 SENet 的不同

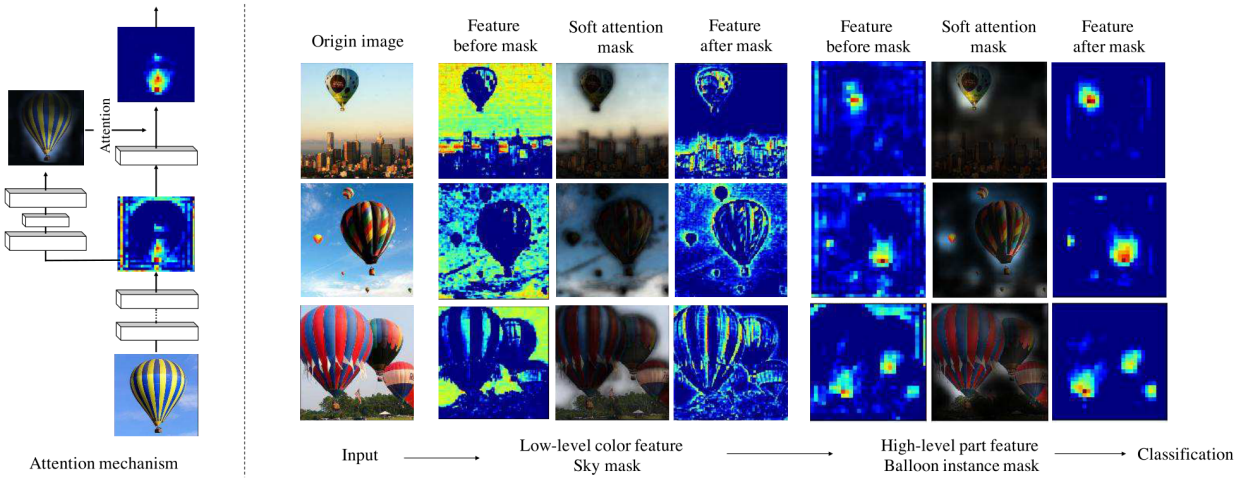
STNet 主要是从 **空间维度** 上着手考虑，而 SENet 是从 **通道维度** 上入手的

相关研究：Attention ResNet

文章： Residual Attention Network for Image Classification

注意力机制在网络中的使用

需要关注一张图：注意力机制示意图



Tips: 越呈现红色的地方，表示网络就越关注这个位置；蓝色相反

左边这部分展示了注意力机制是如何引入到网络中的：

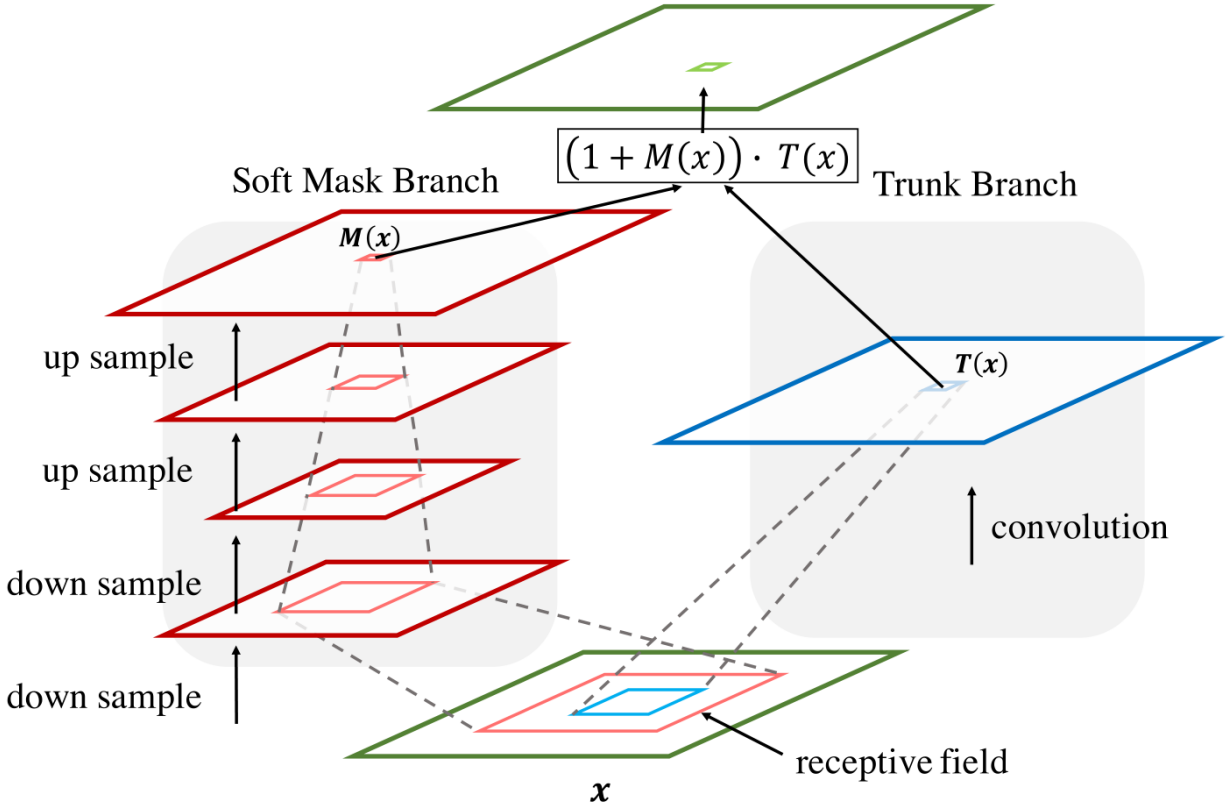
1. 先将 Feature Map 引出另外一个分支
2. 这个分支去计算得到一些权重
3. 这些权重拿回来和原来的 Feature Map 进行一些操作，得到新的特征图

右边这部分展示了注意力机制的应用的可视化过程

可以看到，经过注意力机制模块处理后，Feature Map 的注意力就被增强了

在比较 Low-level 的 Feature Map 上，我们直观上暂时感觉不出来注意力模块的重要性；但是在 High-level 的 Feature Map 上，注意力增强的 **效果还是十分明显** 的

分支结构的实现



如图，一张特征图输入之后，会分成两路：

- 第一路进行几次特征提取，获得 `Soft Mask`，结果记为 $M(x)$
- 第二路进行一次卷积运算，结果记为 $T(x)$
- 最后通过公式 $(1 + M(x)) \cdot T(x)$ 获得最终的输出结果
    - 小细节：将公式展开可以得到 $M(x) \cdot T(x) + T(x)$，这样想更符合逻辑——因为我们想要的结果就是 $M(x) \cdot T(x)$，但是可能会出现二者乘积为 0 或者其他难以预测的情况，所以这里又加了一个 $T(x)$

相关研究：CBAM

文章： CBAM: Convolutional Block Attention Module

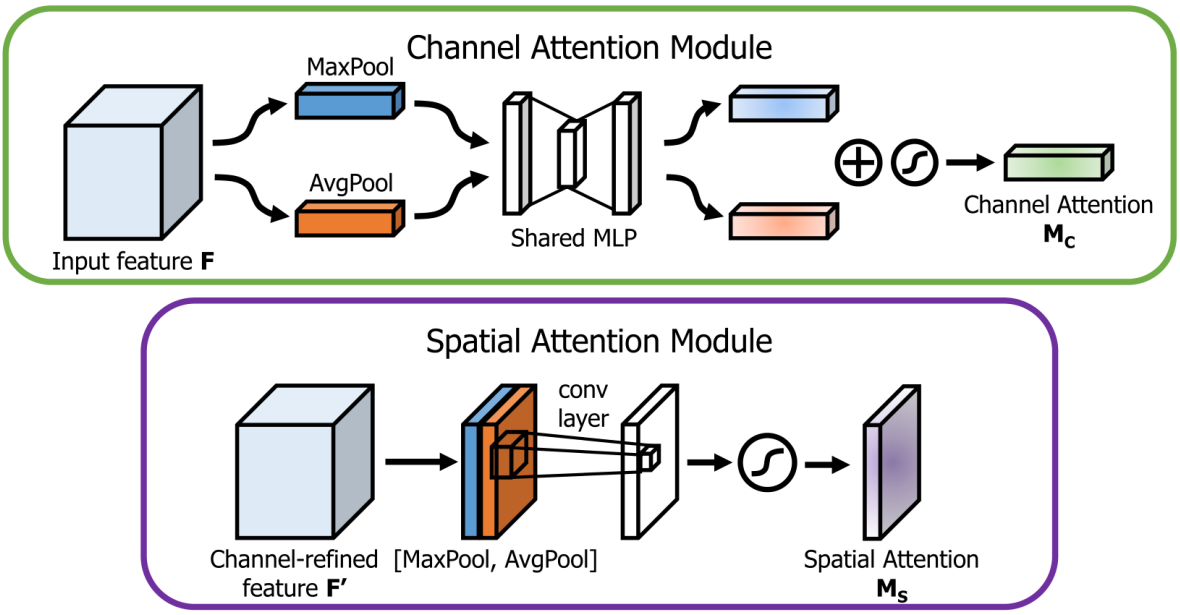
定义及作用

CBAM（Convolutional Block Attention Module）

提出了两阶段注意力机制，一个针对 **通道维度**，一个针对 **空间维度**

从这篇文章可以看出：**注意力机制可以分为不同的维度进行**

注意力提取的实现



^ 核心思想包含于上面这张图中

Tips:

- 上面左边的立方体，正面（我们看起来面积最大的那一面）表示的是 `Channel 维度`，侧面的才是 `宽` 和 `高`
- Shared MLP 指的是 **「一些网络层」**，这里我们不关心具体是什么网络层

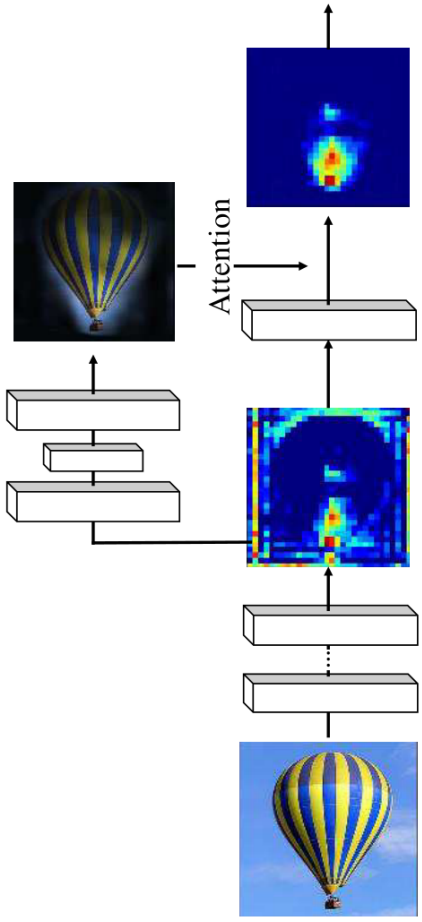
图片的上部展示的是提取 **「通道注意力（`Channel Attention`）」** 的过程，它通过两种池化，将空间信息完全压缩，只保留通道信息，最终得到通道注意力 M_c

图片的下部展示的是提取 **「空间注意力（`Spatial Attention`）」** 的过程，同样是通过两种池化运算，将通道信息完全压缩，只保留空间信息，最终再经过一次卷积运算合并两种池化得到的 `Channel`，最终得到空间注意力 M_s

研究背景总结

注意力机制可以理解为：**设计一些网络层输出「权重」值**，利用权重值与特征图（Feature Map）进行计算，实现特征图的变换，使 **模型加强关注区域** 的特征值

此图是非常有助于理解注意力机制的，多看看：



Attention mechanism

论文

Abstract

- 卷积操作是 CNN 的核心，其可以融合**「空间（Spatial）」和「通道（Channel-wise）」**的特征
  - 注意：**空间**和**通道**是本文强调的重点，二者不是一种东西，本文讨论的是**通道注意力**
- 文章发表时，已经有对**空间特征提取增强**的研究
- 本文针对通道特征，探索通道之间的关系，提出`SE Block`，其可以自适应的校正通道特征
- `SE Block`的堆叠可以形成`SE Net`，并且在多个数据集上取得了很不错的成果，提升了网络的鲁棒性
- `SE Net`与原骨干网络相比，仅增加了少量参数，便实现了大幅度提升网络精度的效果
- 代码开源：SENet——GiHub/hujie-frank

3. Squeeze and Excitation Blocks

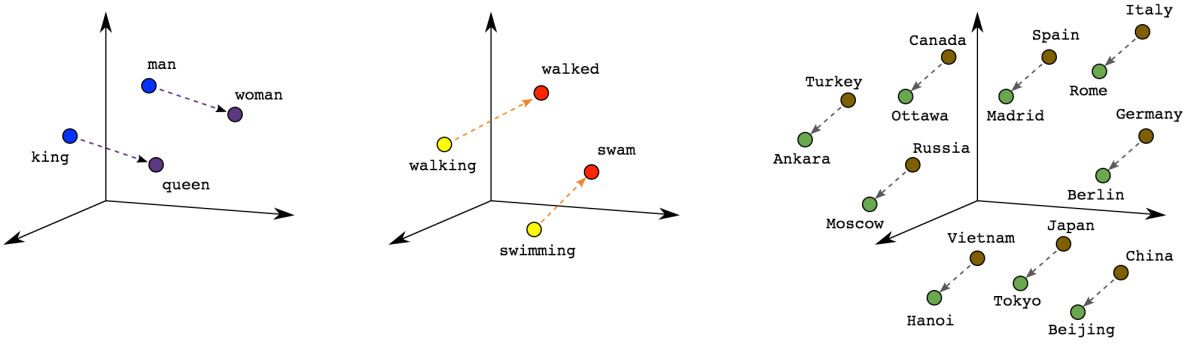
Squeeze

目的：Global Information Embedding（全局信息嵌入）

Squeeze 操作：采用`全局池化`（暴力 (H, W) 转 $(1, 1)$），即通过池化操作压缩输入图片的 $(H, W) \to (1, 1)$，仅保留一个像素信息，其余信息全压缩至通道内，实现信息嵌入

Tips: Embedding 是个较难理解的单词，在此特别说明

- 大多数时候，Embedding 出现往往代表着`「维度降低」`，即一个数据从高维降低到低维的过程
- 在本例中，一个 (H, W, C) 的图像内部所包含的向量无疑是要稀疏于 $(1, 1, C)$ 的，过大的参数，以及过于稀疏的向量，就会导致网络感受域能力不足，难以捕捉到空间尺度上较为稀疏的关联特征，即向量之间缺少有意义的联系
- 而通过 Embedding 操作，就是为了解决上述问题，Embedding 操作可以将大量的稀疏向量（在这里是图像）通过降低维度，将高维数据`映射`到低维空间来解决稀疏输入数据的核心问题
- 假设我们使用 One-hot 来编码语义信息，Embedding 操作的作用更加直观，如图：



Male–Female　　Verb Tense　　Country–Capital

Squeeze 部分的公式：

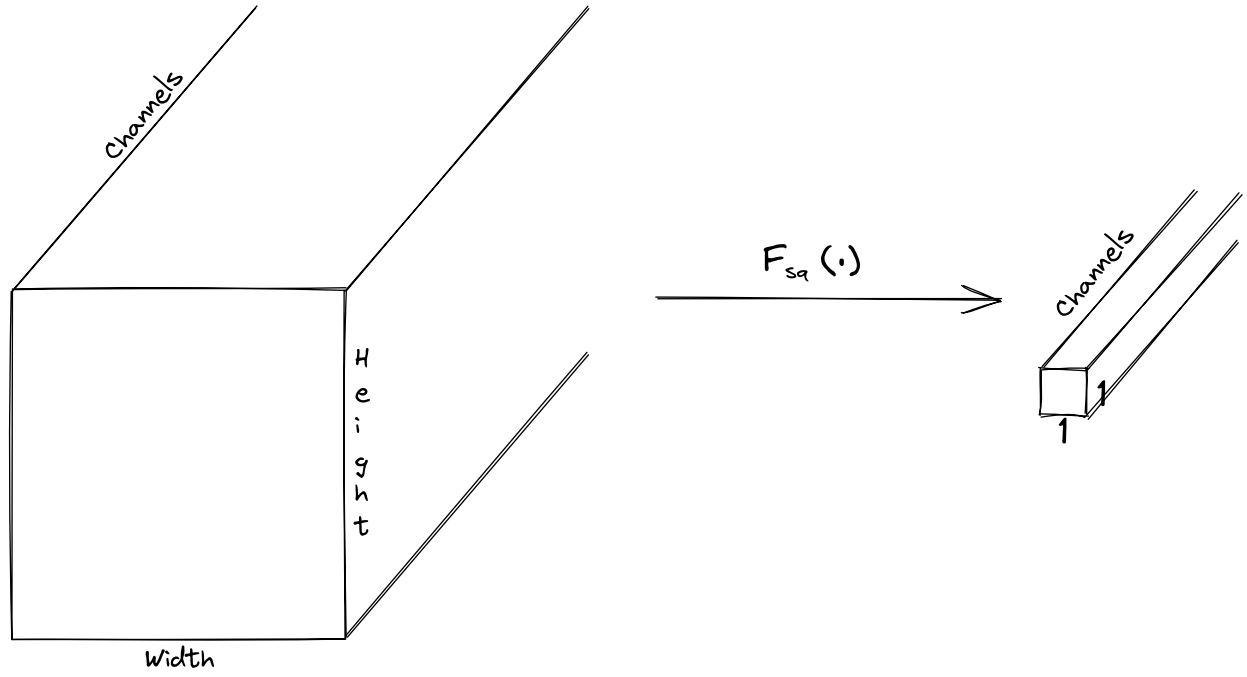
$$z_c = F_{sq}(u_c) = \frac{1}{H \times W} \sum_{i=1}^{H} \sum_{j=1}^{W} u_c(i, j)$$

^ 其中$u_c(x, y)$ 表示的是取 (x, y) 位置的值

从上述公式我们可以明显看出，这是一个`平均池化`操作

为什么不用`最大池化`？答案很简单，作者做实验做出来的

那么 Squeeze 部分的操作我们可以通过下图来很清晰的了解：



Excitation

目的：Adaptive Recalibration （自适应重新校准）

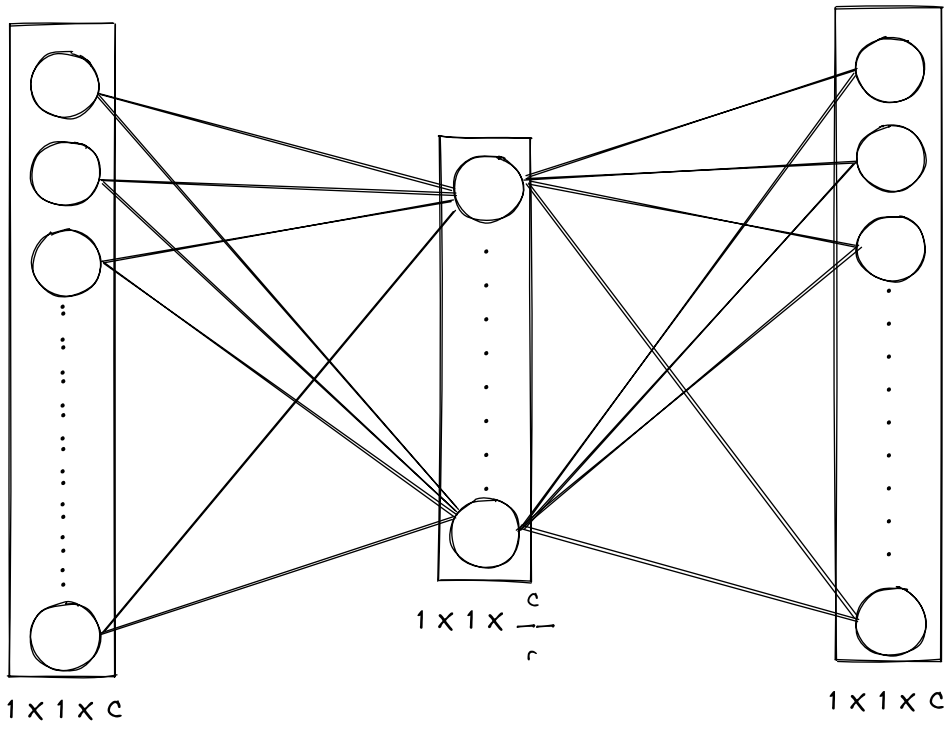
Excitation 操作：采用两个全连接，第一个线形层的神经元由超参数 r 来控制，第二个线性层输出为 $1 \times 1 \times C$ 的权重矩阵

Excitation 部分的公式：

$$s = F_{ex}(z, W) = \sigma(g(z, W)) = \sigma(W_2\delta(W_1 z))$$

^ 其中 δ 代表 ReLU 函数；$W_1 \in R^{\frac{C}{r} \times C}$；$W_2 \in R^{C \times \frac{C}{r}}$

流程图可以简略绘制如下：



如图，其中 $\frac{C}{r}$ 中的 r 为超参数，其目的是**「控制网络参数量」**

论文中有提到相关的对比实验，经过文章作者的实际测试，$r = 16$ 是较好的一个参数取值

Summary

上述的 Squeeze 和 Excitation 过程非常好理解，没有什么困难的地方，看懂上面两幅图即可

注意到其运算过程比较简单，公式再次总结如下（整个过程）：

$$out = F_{ex}(F_{sq}(u_c), W) = \sigma(W_2\delta(W_1(F_{sq}(u_c)))) = \sigma\{W_2\delta[W_1(F_{sq}(u_c))]\} = \sigma\{W_2\delta[W_1(\frac{1}{H \times W}\sum_{i=}^{H}$$