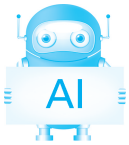
第 5 章



注意力机制

注意力机制(Attention Mechanism)是人类特有的大脑信号处理机制。例如,人类视觉通过快速扫描全局图像获得需要重点关注的目标区域,也就是一般所说的注意力焦点,而后对这一区域投入更多注意力资源,获取更多需要关注目标的细节信息,抑制其他无用信息,人类的听觉也具有同样的功能。

近几年,注意力机制在各种深度神经网络中已广泛使用,成为提升深度神经网络的重要手段。从本质上讲,深度神经网络中采用的注意力机制和人类的选择性视觉、听觉注意力机制类似,其核心目的也是从众多信息中选择出对当前任务更关键的信息。

在深度神经网络中,一般而言,模型的参数越多,则模型的表达能力越强,模型所存储的信息量也越大,但这会带来信息过载的问题。通过引入注意力机制,在众多的输入信息中聚焦对当前任务更为关键的信息,降低对其他信息的关注度,甚至过滤掉无关信息,就可以解决信息过载问题,并提高任务处理的效率和准确性。

注意力机制,最早是 20 世纪 90 年代视觉图像领域提出来的,但是真正火起来始于 2014 年 Google Mind 团队的论文 *Recurrent Models of Visual Attention*,他们在 RNN 模型上使用了 Attention 机制来进行图像分类。也是 2014 年,Bahdanau 等在论文 *Neural Machine Translation by Jointly Learning to Align and Translate* 中,将注意力机制引入神经机器翻译的研究,使用类似的注意力机制在机器翻译任务上将翻译和对齐同时进行,主要用于对整个句子的特征向量进行加权,从而选取当前时间最重要的特征向量的子集。随后注意力机制被广泛应用在基于 RNN/CNN 等神经网络模型的各种任务中。2017 年,Google 机器翻译团队发表的 *Attention is All You Need* 论文中提出的 Transformer 大量使用了自注意力(Self-Attention)机制来学习文本表示。自此,自注意力机制也成为了研究热点,并在各种任务上应用,取得了良好效果。

在深度神经网络中使用的注意力机制有两类:硬注意力(Hard Attention)和软注意力(Soft Attention)。

硬注意力机制是指选择输入图像或序列一些位置上的信息(关注点),比如随机选择一些信息或概率高的信息。通常对硬注意力,选取概率高的特征这一操作是不可微的,很难在深度神经网络中通过训练得到,因此实际应用并不多。

软注意力机制是指在选择信息的时候,不是从 N 个信息中只选择几个,而是计算 N 个输入信息的加权平均,再输入到神经网络中进行处理。它是当前深度神经网络中应用最多的注意力机制。软注意力更关注区域或通道,它是确定性的注意力,学习完成后直接可以通

过网络生成。最关键的是软注意力是可微的，可微分的注意力可以通过神经网络算出梯度，并且利用前向传播和反向传播来学习得到注意力的权重。

本章将主要介绍应用中最常用的软注意力机制。首先介绍软注意力的原理和计算方法，然后介绍在处理静态图像数据的 CNN 中应用的软注意力机制，之后介绍自注意力机制，最后介绍在处理动态时序数据的 RNN 中应用的互注意力机制。

5.1 软注意力机制的原理及计算过程

给定一组输入信息 $\boldsymbol{X}=[x_1,x_2,\cdots,x_N]$和一个查询向量 $\boldsymbol{q}$，通过寻找 $\boldsymbol{q}$ 与每一个输入信息 $\boldsymbol{x}_i$ 的相关性来选择 $\boldsymbol{x}_i$ 中的部分信息，然后组合起来形成新的 $\boldsymbol{x}_i$，送入神经网络进行处理。这就是软注意力机制的工作原理。这里所谓的“软性”选择机制，不是从存储的多个信息中只挑出一条信息来，而是雨露均沾，从所有的信息中都抽取一些，只不过最相关的信息抽取得就多一些。

软注意力机制的计算过程包括 3 个步骤。

(1) 计算相似度。

使用相关系数 $S(\boldsymbol{x}_i,\boldsymbol{q})$表示相似度。相关系数 $S(\boldsymbol{x}_i,\boldsymbol{q})$也称注意力打分函数，可以采用以下几种方式计算。

$$\begin{aligned}
&\text{加性模型} && S(\boldsymbol{x}_i,\boldsymbol{q})=\boldsymbol{V}^{\mathrm{T}}\mathrm{Tanh}(\boldsymbol{W}\boldsymbol{x}_i+\boldsymbol{U}\boldsymbol{q})\\
&\text{点积模型} && S(\boldsymbol{x}_i,\boldsymbol{q})=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{q}\\
&\text{缩放点积模型} && S(\boldsymbol{x}_i,\boldsymbol{q})=\frac{\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{q}}{\sqrt{d}}\\
&\text{双线性模型} && S(\boldsymbol{x}_i,\boldsymbol{q})=\boldsymbol{x}_i^{\mathrm{T}}\boldsymbol{W}\boldsymbol{q}
\end{aligned}\tag{5-1}$$

显然，式(5-1)的加性模型是一个隐层采用 Tanh 作为激活函数，输出层采用线性激活函数的 3 层前向神经网络，$\boldsymbol{U}$、$\boldsymbol{W}$ 和 $\boldsymbol{V}$ 是网络的权值参数，当然也可以采用卷积神经网络来实现；缩放点积模型中的参数 d 是输入信息 $\boldsymbol{x}_i$ 的维度；双线性模型中的 $\boldsymbol{W}$ 是加权参数。点积模型和缩放点积模型由于计算简单，没有另需确定的参数，最为常用。

(2) 计算注意力分布。

定义一个注意力变量 $z\in[1,N]$来表示被选择信息 $\boldsymbol{x}_i$ 的索引位置，即 $z=i$ 来表示选择了第 i 个输入信息 $\boldsymbol{x}_i$。通过计算 $\boldsymbol{q}$ 和 $\boldsymbol{x}_i$ 的相关性来确定选择第 i 个输入 $\boldsymbol{x}_i$ 的概率 α_i。$\alpha_1,\alpha_2,\cdots,\alpha_N$ 构成的概率向量 $\boldsymbol{\alpha}$ 称为注意力分布(Attention Distribution)。

$$\begin{aligned}
\alpha_i&=p(z=i\mid\boldsymbol{X},\boldsymbol{q})\\
&=\mathrm{Softmax}(s(\boldsymbol{x}_i,\boldsymbol{q}))\\
&=\frac{\exp\left(s(\boldsymbol{x}_i,\boldsymbol{q})\right)}{\sum_{j=1}^{N}\exp(s(\boldsymbol{x}_j,\boldsymbol{q}))}
\end{aligned}\tag{5-2}$$

其中，Softmax 函数用于将相关系数转换为概率值。

(3) 计算注意力输出。

将注意力分布 α_i 与相应的输入信息 $\boldsymbol{x}_i$ 相乘，汇总求和就得到了注意力输出，如式(5-3)所示。

$$\mathrm{att}(\boldsymbol{X},\boldsymbol{q})=\sum_{i=1}^{N}\alpha_i\boldsymbol{x}_i \tag{5-3}$$

实际应用中并不一定直接使用 $\boldsymbol{X}$ 和 $\boldsymbol{q}$，而是它们的线性变换，这称为键值对方式，也可以称为广义的软注意力机制。

在广义的软注意力机制中用键值对(Key-Value Pair)来表示输入信息，那么 N 个输入信息就可以表示为 $(\boldsymbol{K},\boldsymbol{V})=[(\boldsymbol{k}_1,\boldsymbol{v}_1),(\boldsymbol{k}_2,\boldsymbol{v}_2),\cdots,(\boldsymbol{k}_N,\boldsymbol{v}_N)]$，其中“键”用来计算注意力分布 α_i，“值”用来计算聚合信息。这就相当于由 Query 与 Key 的相似性来计算每个 Value 值的权重，然后对 Value 值进行加权求和。加权求和得到最终的 Value 值，也就是注意力值。

广义的软注意力值的计算过程与前述普通的软注意力值计算过程一样，仅是将式(5-1)和式(5-2)中的 $\boldsymbol{x}_i$ 换为 $\boldsymbol{k}_i$，式(5-3)中的 $\boldsymbol{x}_i$ 换为 $\boldsymbol{v}_i$。如式(5-4)～式(5-6)所示。

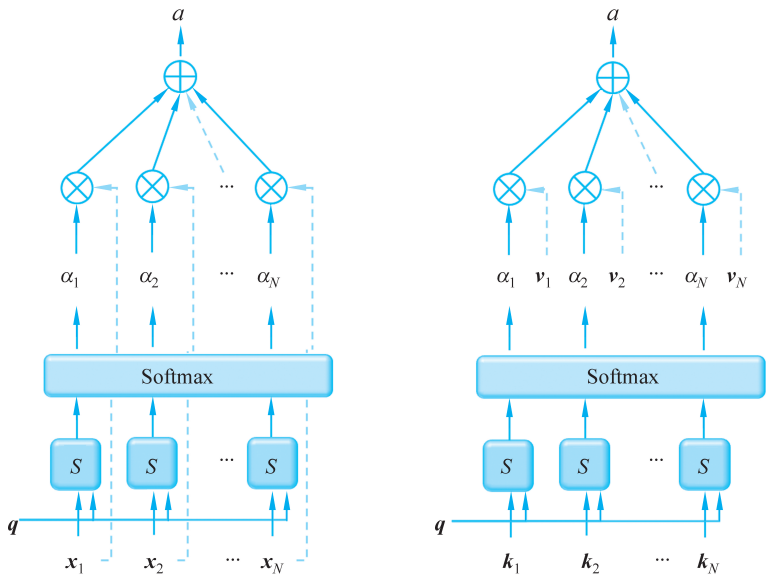
$$s_i=F(\boldsymbol{Q},\boldsymbol{k}_i) \tag{5-4}$$

$$\alpha_i=\mathrm{Softmax}(s_i)=\frac{\exp(s_i)}{\sum_{j=1}^{N}\exp(s_j)} \tag{5-5}$$

$$\mathrm{att}((\boldsymbol{K},\boldsymbol{V}),\boldsymbol{Q})=\sum_{i=1}^{N}\alpha_i\boldsymbol{v}_i \tag{5-6}$$

需要说明的是，广义软注意力机制，由于引入线性变换求取 $\boldsymbol{K}$、$\boldsymbol{Q}$ 和 $\boldsymbol{V}$，实际上这些变换可以代表输入 $\boldsymbol{X}$ 和查询量 $\boldsymbol{Q}$ 的不同特性，且可以看作一层线性神经网络层，也就是在软注意力机制中引入了可学习训练的参数，使其具有了记忆能力。

软注意力值计算过程可由图 5-1 表示。



(a) 普通模式　　(b) 键值对模式

图 5-1　软注意力值的计算过程

图 5-2 给出了另一种在文献中常见的广义软注意力(键值对方式)值的计算过程。

分析式(5-1)～式(5-6)可知，软注意力机制的核心要素是查询向量 $\boldsymbol{q}$。它的有无和来自何处，形成了实际应用中的不同类型的软注意力机制。

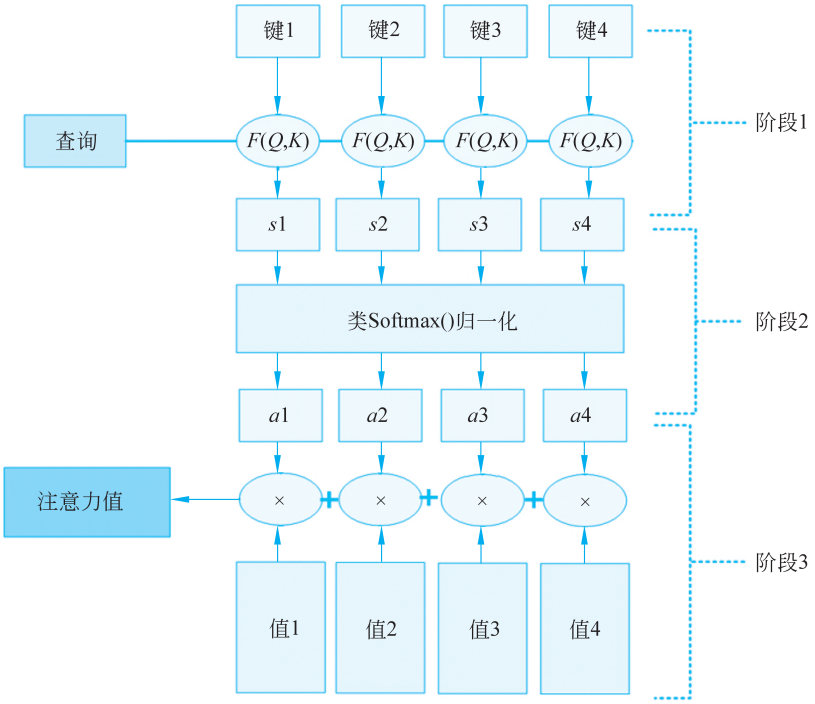


图 5-2 广义软注意力(键值对方式)值计算过程

(1) 当不存在查询量 $\boldsymbol{q}$ 时,式(5-1)的相似性计算只有加性模型可用。它显然是一个三层前向网络,只需通过网络训练确定参数,然后计算输入信息的权值,给其加权。这时的软注意力就演变成了卷积神经网络中常用的通道注意力和空间注意力模型。

(2) 当查询量 $\boldsymbol{q}$ 来自输入信息,软注意力就变成了近几年最有影响力的自注意力。

(3) 当查询量 $\boldsymbol{q}$ 来自其他,例如输出信息,软注意力就变成了在 Encoder-Decoder 框架(编码器—译码器)中处理时序数据的互注意力。

5.2 通道注意力和空间注意力

卷积神经网络是最常用的针对图像数据的前向神经网络,它的每个隐层都包含大量的特征图,每张特征图代表了原始图像数据的某种特征属性。CNN 每层中的特征图组有 3 个维度,针对各张特征图的通道维度和空间维度,形成了通道注意力和空间注意力。由于通道注意力和空间注意力的权值计算利用了输入的所有信息,因此属于软注意力机制概念范畴,是软注意力机制,按式(5-1)的加性模型计算注意力权值。

5.2.1 通道注意力

在卷积神经网络中,特征图代表了原始图像数据的特征,在同一层中,不同的特征图代表了不同的属性。显然,不同属性对于卷积神经网络要完成的工作贡献程度不同,应该给予不同的重视程度。由于在卷积神经网络中,特征图所在的位置称为通道,因此,反映对通道重视程度的给通道加权的方法称为通道注意力。

3.3.4 节介绍的 SE-Net 正好反映了各通道的重要程度，因此，它被称为最早和最基本的通道注意力。关于 SE-Net 的详细介绍见 3.3.4 节，这里不再重复。

SE-Net 也存在不足，它仅仅针对每张特征图使用了全局平均池化(GAP)，不能全面反映特征图的特性。因此，有研究人员提出将特征图进行全局最大值池化(GMP)，并与 GAP 的结果串接在一起进行后续处理，这一方式目前已成为通道注意力最常用的手段。当然，按照这一思路也可对特征图求方差，单独或与 GAP、GMP 的结果串接后求取通道注意力。

SE-Net 提出后，针对 SE-Net 的不足，研究人员提出了一些新的通道注意力方法，下面介绍两种方法：ECA-Net 和 SK-Net。

1. ECA-Net

SE-Net 对全局池化后形成的特征矢量先降维，然后再升维，有研究表明降维会对通道关注度的预测产生副作用，而且对所有通道的相关性进行捕获是低效且不必要的。基于上述研究提出了改进的通道注意力方法 ECA-Net(Efficient Channel Attention Networks)。

ECA-Net 是在 SE-Net 的基础上使用一维卷积代替全连接，提出的一种无降维的局部跨通道交互策略。在没有降维的情况下，通过考虑每个通道及其 k 个邻居，捕获本地跨通道交互。k 的选取与通道个数有关，其函数表达如下：

$$k=\varphi(C)=\left|\frac{\log_2(C)}{\gamma}+\frac{b}{\gamma}\right|_{\text{odd}} \tag{5-7}$$

其中，C 表示通道数；b 和 γ 已知，为设定的超参数；$|\cdot|_{\text{odd}}$ 表示最近的奇数。

图 5-3 为 ECA-Net 的结构。

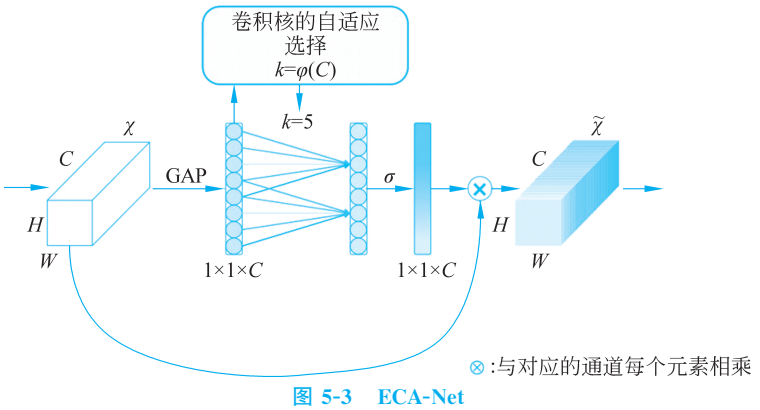


图 5-3　ECA-Net

2. SK-Net

SE-Net 是直接针对一层中的特征图进行的，但是人的视觉皮层神经元接受域大小是会根据看不同尺寸、不同远近的物体来调节的。受这一思想启发，研究人员提出了一种可以根据输入信息的多个尺度调节接受域大小的多分支通道注意力 SK-Net。图 5-4 展示了有两个分支的 SK-Net。

在 SK-Net 中，Split 操作如下：将输入划分为 N 个分支、不同卷积核大小的完整卷积操作(卷积、BN、ReLU)。Fuse 操作如下：将 N 个分支获得的特征图相加以后，与 SE-Net 类似，先通过全局平均池化将相加后的特征图压缩为一维向量，以收集全局上下文信息，然后通过 N 个分支的全连接层得到权重向量。与 SE-Net 不同的是，SK-Net 使用的非线性函

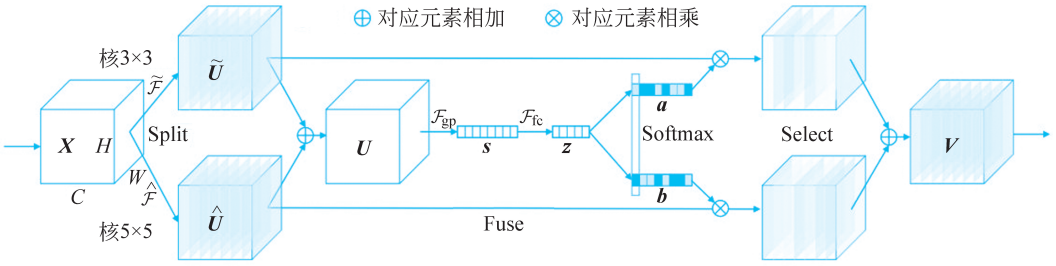


图 5-4 多分支通道注意力 SK-Net

数是 Softmax，并且对于每个分支都输出一个权重向量，N 个分支对应位置之和为 1，图 5-5 给出了 Fuse 的操作过程。Select 操作如下：对 Fuse 操作得到的 N 个权重向量，分别通过乘法逐通道加权到先前 N 个分支的特征图上，然后相加。

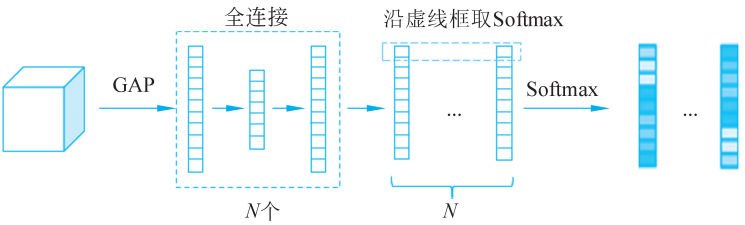
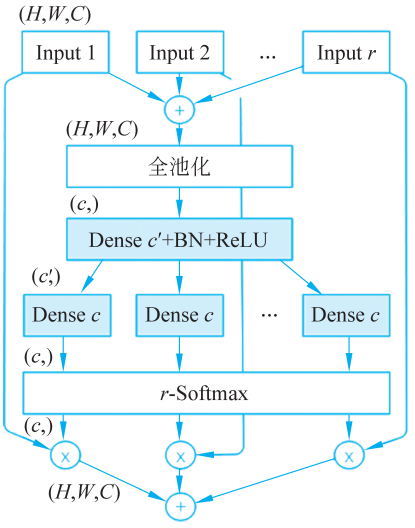


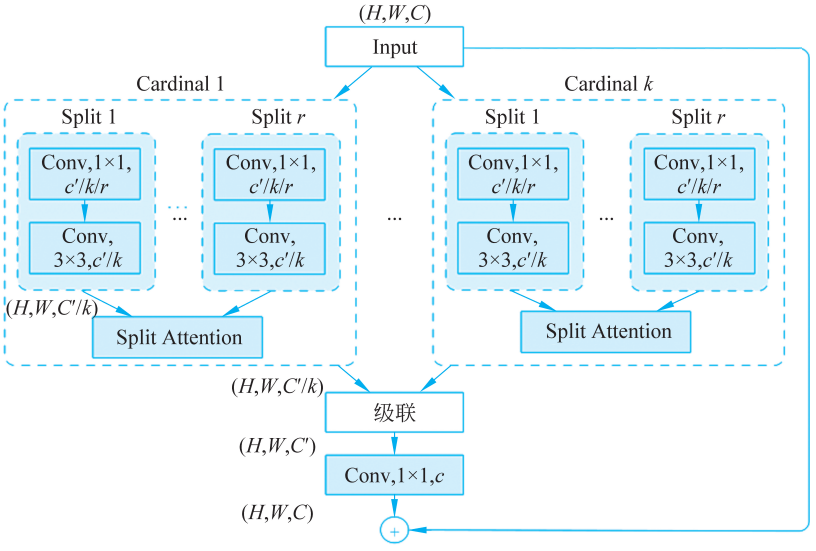
图 5-5 SK-Net 中 Fuse 的操作过程

SK-Net 中的 Fuse 操作对于将通道注意力用于深度可分离网络有实际意义。ResNeSt 受启发于 ResNeXt(ResNet 的一种改进，将 ResNet 的主干 Block 分成多个分支 Cardinal，进行分组卷积操作，提高网络工作效率)中分组卷积的思想，将输入(c，h，w)降维后(c'，h，w)分别通过 k 个 Cardinal 支路，每个 Cardinal 支路里都使用了 SK-Net。图 5-6 中的 Split-Attention 就是 SK-Net 的 Fuse 操作，Dense 代表全连接操作。



(a) Split-Attention

图 5-6 SK-Net 在 ResNeSt 模块中的应用



(b) ResNeSt模块

图 5-6　（续）

5.2.2　空间注意力

卷积神经网络处理图像数据中的每个像素对于所要完成的任务重要性不完全相同，同样，隐层特征图中每个像素对所完成任务的重要性也不相同。显然，给特征图的每一个像素加权有利于提高卷积神经网络的性能，由于这种加权是针对特征图像素空间位置进行的，因此称为空间注意力。

空间注意力类似通道注意力，不同在于它的全局平均池化（GAP）和全局最大值池化（GMP）不是针对通道（特征图），而是针对同一层内的所有特征图中相同位置的像素进行 GAP 和 GMP，并将得到的均值特征图和最大值特征图并在一起进行卷积操作，卷积生成特征图的每一个像素对应的神经元激活函数选取 Sigmoid 函数，得到针对每个像素加权值构成的特征图，最后将这一特征图的像素与原所有特征图的对应像素相乘，为所有特征图的每个像素加权。图 5-7 给出了空间注意力图的生成和为像素加权的过程。

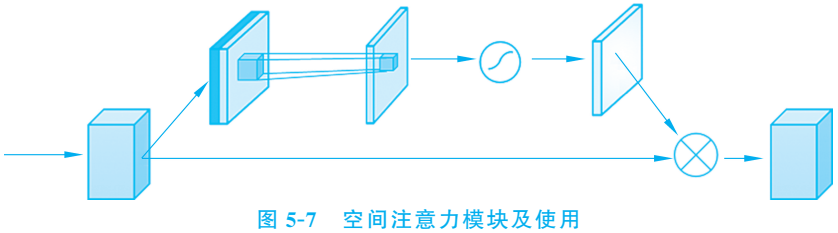


图 5-7　空间注意力模块及使用

当然，空间注意力也可以按类似于多分支通道注意力的方式改进，将原特征图进行多尺度卷积变换，形成多尺度融合的空间注意力。

5.2.3　混合注意力

通道注意力和空间注意力都能提高卷积神经网络的性能，可以同时应用于卷积神经网

络之中，称为混合注意力。

CBAM(Convolutional Block Attention Module)是著名的混合注意力模块，它实质上是通道注意力和空间注意力的串行使用，图 5-8 给出 CBAM 模块及其组成。CBAM 的使用方法与 SE-Net 一样，图 5-9 给出了它的使用方式。

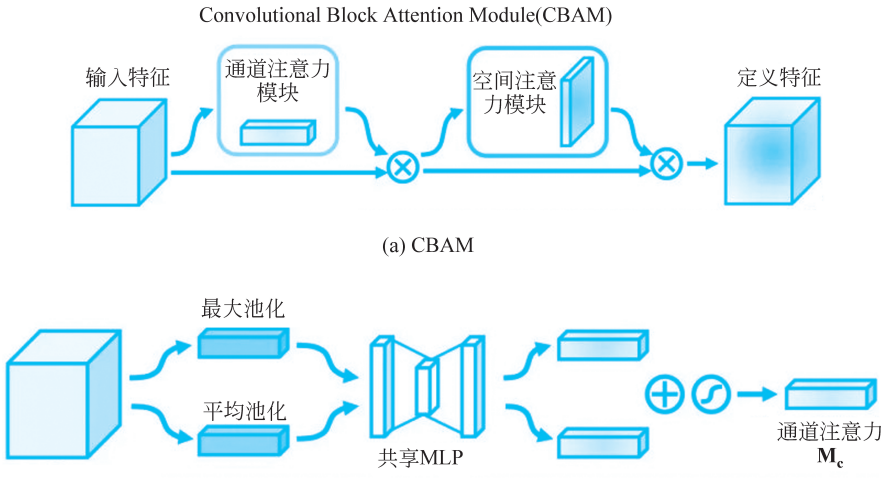


图 5-8 CBAM 模块及其组成

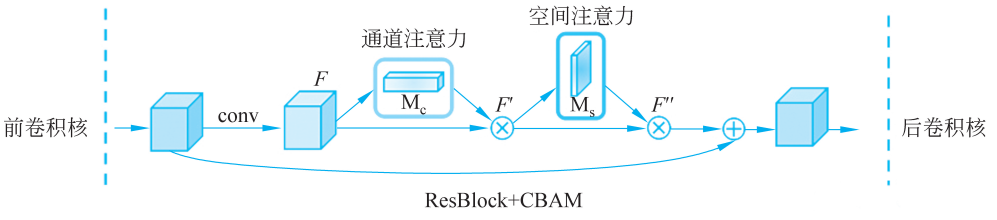


图 5-9 CBAM 的使用

通道注意力和空间注意力既然可以串行构成混合注意力，当然也可以并行构成混合注意力，如图 5-10 所示。其使用方式与图 5-9 的 CBAM 相同。

前述的混合注意力都是通道注意力和空间注意力的串接或并行形成的，实质上并未融合在一起，而是独立运行的。2021 年的论文 *Coordinate Attention for Efficient Mobile Network Design* 针对轻量化网络设计提出的 CA 注意力就是融合了通道和位置信息的混合注意力，它与通道注意力 SE-Net、CBAM 的区别如图 5-11 所示。

与通过二维全局池化将特征张量转换为单个特征向量的通道注意力不同，CA 注意力

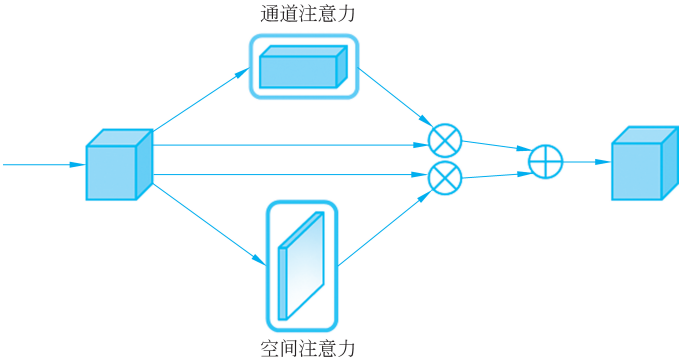
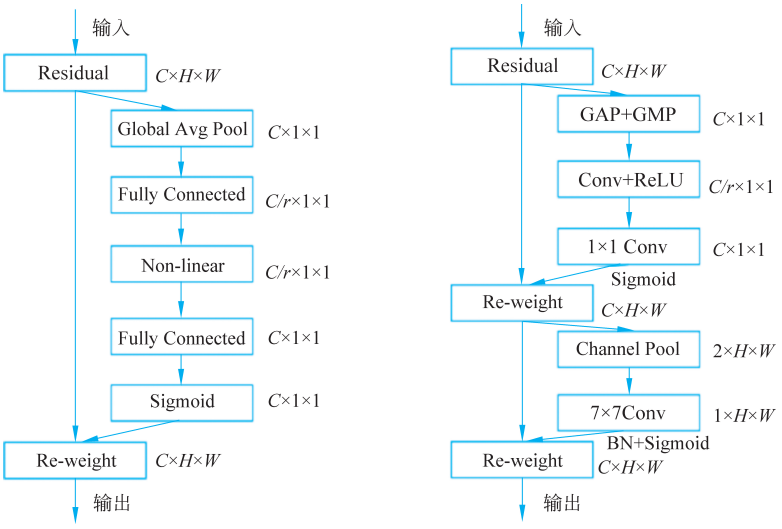
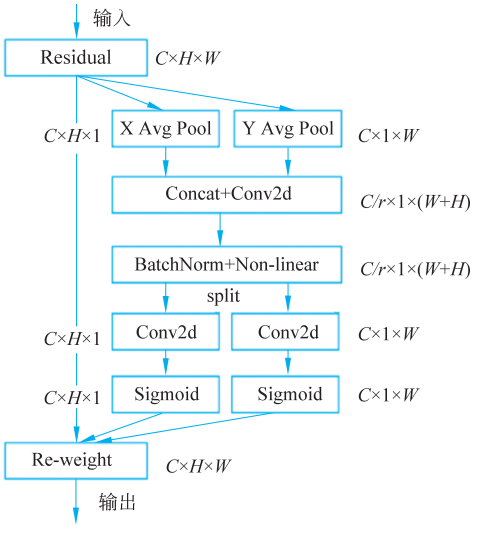


图 5-10　通道注意力和空间注意力并行构成的混合注意力



(a) SE　　(b) CBAM



(c) CA

图 5-11　CA 模块与 SE 模块、CBAM 模块的区别

将通道注意力分解为两个一维特征编码过程，分别沿 2 个空间方向聚合。这样，可以沿一个空间方向捕获远程依赖关系，也可以沿另一空间方向保留精确的位置信息。然后将生成的特征图分别编码为一对方向感知和位置敏感的注意力图(Attention Map)，并将其互补地应用于输入特征图，增强所关注对象的特征表示。

5.3 自注意力机制

如果软注意力机制中的 $\boldsymbol{K}$、$\boldsymbol{Q}$ 和 $\boldsymbol{V}$ 均来自输入信息，软注意力机制就被称为自注意力机制。

自注意力机制的数学描述如下：假设一个神经网络层中的输入序列为 $\boldsymbol{X}=[\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N]$，输出序列为同等长度的 $\boldsymbol{H}=[\boldsymbol{h}_1,\boldsymbol{h}_2,\cdots,\boldsymbol{h}_N]$，首先通过线性变换得到 3 组向量序列：$\boldsymbol{K}=\boldsymbol{W}_K^{\mathrm{T}}\boldsymbol{X}=[\boldsymbol{k}_1,\boldsymbol{k}_2,\cdots,\boldsymbol{k}_N]$，$\boldsymbol{Q}=\boldsymbol{W}_Q^{\mathrm{T}}\boldsymbol{X}=[\boldsymbol{q}_1,\boldsymbol{q}_2,\cdots,\boldsymbol{q}_N]$，$\boldsymbol{V}=\boldsymbol{W}_V^{\mathrm{T}}\boldsymbol{X}=[\boldsymbol{v}_1,\boldsymbol{v}_2,\cdots,\boldsymbol{v}_N]$，然后用式(5-4)、式(5-5)和式(5-6)计算自注意力输出。

自注意力可以建立输入序列的长程关系，在深度神经网络中应用自注意力机制有效提高了深度神经网络的性能。本小节将首先介绍自注意力的输入方式及自注意力机制的特性，然后讨论自注意力机制与 RNN 的区别，最后介绍几种自注意力机制在深度神经网络中的应用方案。

5.3.1 自注意力机制的输入方式及特性

在应用中，自注意力机制存在两种输入方式：全输入和逐项输入(掩膜输入)。

1. 全输入

全输入指的是按顺序输入 $\boldsymbol{X}$，经变换产生 $\boldsymbol{K}$、$\boldsymbol{Q}$ 和 $\boldsymbol{V}$ 序列后，再进行自注意力处理，即

$$\boldsymbol{h}_i=\mathrm{att}((\boldsymbol{K},\boldsymbol{q}_i),\boldsymbol{V})=\sum_{j=1}^{N}\alpha_{ji}\boldsymbol{v}_j=\sum_{j=1}^{N}\mathrm{Softmax}(\boldsymbol{S}(k_j,\boldsymbol{q}_i))\boldsymbol{v}_j \quad j=1,2,\cdots,N \tag{5-8}$$

其计算过程如图 5-12 所示。

由式(5-8)和图 5-12 可知，全输入自注意力机制的任何一个注意力输出都与全部输入序列相关，能够建立最长程的双向输出输入关系。

全输入的自注意力可以按输入采用矢量相乘的方式由式(5-8)逐项计算，也可以使用矩阵方式求取注意力输出，用矩阵方式的效率更高。

使用矩阵运算的全输入自注意力方法如下。

设每个输入序列信息的长度为 n，那么 N 个序列 $\boldsymbol{x}_1,\boldsymbol{x}_2,\cdots,\boldsymbol{x}_N$ 输入构成 $n\times N$ 维的输入矩阵；3 个变换矩阵 $\boldsymbol{W}_K$、$\boldsymbol{W}_Q$ 和 $\boldsymbol{W}_V$ 将长度 n 的向量变换为长度 d 的向量，3 个变换可以看作有 n 个输入节点、d 个线性输出神经元的 3 个单层全连接线性神经网络，如图(5-12)所示，即 3 个变换阵的维度为 $n\times d$。

第一步：将 X 送入 3 个线性神经网络层，分别得到维度为 $d\times N$ 的 $\boldsymbol{K}$、$\boldsymbol{Q}$ 和 $\boldsymbol{V}$ 矩阵。

第二步：以最简单的点积模型 $\boldsymbol{K}^{\mathrm{T}}\boldsymbol{Q}$ 求维度为 $N\times N$ 的相似度矩阵，再对其逐行求 Softmax。

第三步：将经过 Softmax 的相似度矩阵与 $\boldsymbol{V}$ 阵相乘，就得到了 $d\times N$ 维的注意力输出矩阵 $\boldsymbol{H}$。