基于 HardSoftmax 的并行选择核注意力

朱　梦[1]，闵卫东[2,3]，张　煜[1]，段静雯[1]

1. 南昌大学 信息工程学院，南昌 330031
2. 南昌大学 软件学院，南昌 330047
3. 江西省智慧城市重点实验室，南昌 330047

摘　要：注意力被广泛地运用在卷积神经网络中，并有效地提升了卷积神经网络的性能。同时，注意力是非常轻量的，且几乎不需要改变卷积神经网络原来的架构。本文提出了基于 HardSoftmax 的并行选择核注意力。首先，针对 Softmax 包含指数运算，对于较大的正输入很容易发生计算溢出的问题，本文提出了计算更安全的 HardSoftmax 来替换 Softmax。然后，不同于选择核注意力将全局特征的提取和转换放在特征融合之后，并行选择核注意力将全局特征的提取和转换单独放在一个分支，与具有不同核大小的多个分支构成并行结构。同时，并行选择核注意力的全局特征转换使用分组卷积，进一步减少参数量和计算量。最后，并行选择核注意力通过 HardSoftmax 注意来关注不同核大小的多个分支。一系列的图像分类实验表明，只是简单地用 Hardsoftmax 替换 Softmax，也能保持或提升原注意力的性能。HardSoftmax 的运行速度在实验中也比 Softmax 更快速。并行选择核注意力能够以更少的参数量和计算量追平或超越选择核注意力。

关键词：卷积神经网络；HardSoftmax；并行选择核注意力
文献标志码:A　中图分类号:TP183　doi：10.3778/j.issn.1002-8331.2010-0085

Parallel Selective Kernel Attention Based on HardSoftmax

ZHU Meng[1], MIN Weidong[2,3], ZHANG Yu[1], DUAN Jingwen[1]

1. School of Information Engineering, Nanchang University, Nanchang 330031, China
2. School of Software, Nanchang University, Nanchang 330047, China
3. Jiangxi Key Laboratory of Smart City, Nanchang 330047, China

Abstract: Attention has been widely used in Convolutional Neural Networks (CNNs), and effectively improves the performance of CNNs. At the same time, attention is very lightweight, and almost does not need to change the original architecture of CNNs. This paper proposes parallel selective kernel (PSK) attention based on HardSoftmax. Firstly, to solve the problem that Softmax contains exponential operation, which is easy to occur computational overflow for large positive inputs, this paper proposes computationally safer HardSoftmax to replace Softmax. Then, different from selective kernel (SK) attention which puts the extraction and transformation of global features after feature fusion, PSK attention puts it in one branch alone, thus being in parallel connection with multiple branches with different kernel sizes. Meanwhile, the transformation of global features uses group convolution to further reduce the number of parameters and multiply adds (MAdds). Finally, multiple branches with different kernel sizes

基金项目：国家自然科学基金（62076117, 61762061）；江西省自然科学基金（20161ACB20004）；江西省智慧城市重点实验室（20192BCD40002）。

作者简介：朱梦，男，硕士研究生，研究领域为计算机视觉、自然语言处理和强化学习，E-mail：mengzhu@email.ncu.edu.cn；闵卫东，通信作者，男，博士，教授，CCF 杰出会员，研究领域为图像和视频处理、人工智能、大数据、分布式系统和智慧城市信息技术，E-mail：minweidong@ncu.edu.cn；张煜，女，硕士研究生，研究领域为计算机视觉、图像和视频处理、行为识别和人工智能，E-mail：530092719@qq.com；段静雯，女，硕士研究生，研究领域为图像理解和计算机视觉，E-mail：jingwen_duan@163.com。

are fused using HardSoftmax attention that is guided by the information in these branches. A wide range of image classification experiments show that just simply replacing Softmax with HardSoftmax can maintain or improve the performance of original attention. HardSoftmax also runs faster than Softmax in the experiments of this paper. PSK attention can match or outperform SK attention with less parameters and MAdds.

Key words：Convolutional Neural Networks (CNNs); HardSoftmax; parallel selective kernel (PSK) attention

作为下游网络，卷积神经网络 (Convolutional Neural Networks, CNNs) 在计算机视觉中发挥了重要的作用，比如目标检测、语义分割、图像生成等。更好的、更快的卷积神经网络架构一直是研究的热点。而且，卷积神经网络已经被应用到许多实际项目中，比如交通管理[1,2,3]、摔倒检测[4,5,6,7]、人脸识别[8,9,10]等。网络深度已经被许多工作[11,12,13]证明是非常重要的。但是深层网络存在退化的问题：随着网络深度的增加，准确率变得饱和，然后急速下降。为了解决深层网络退化的问题，ResNet[14]提出了残差连接。ResNet 开启了卷积神经网络架构设计的新纪元，以致于后来的卷积神经网络都开始借鉴残差连接的思想。

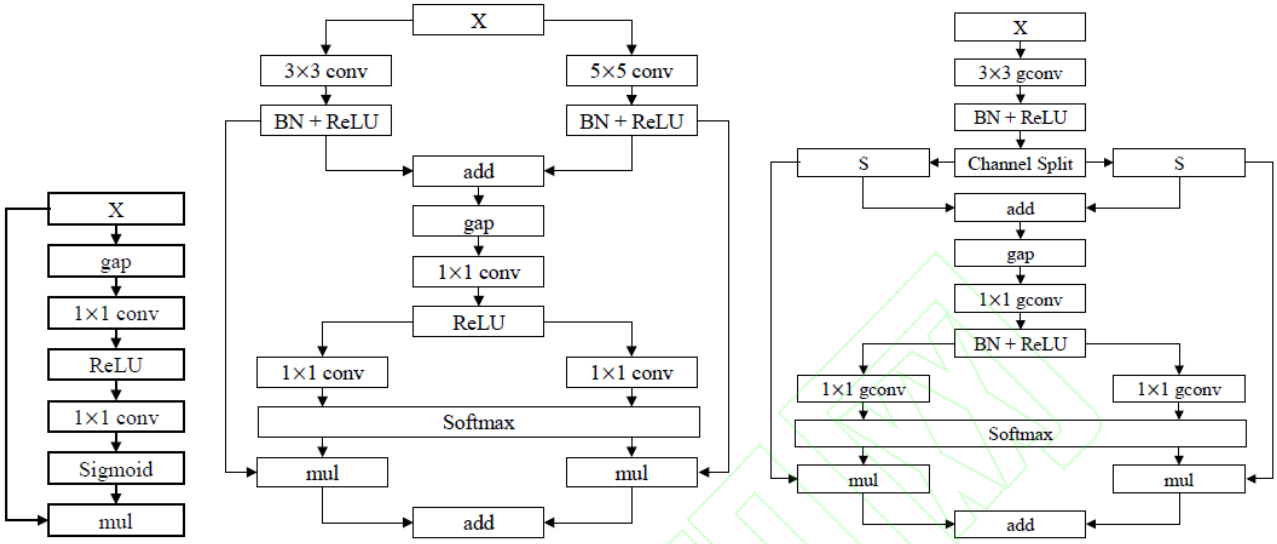
PreResNet[15]证明了残差连接的重要性，并提出了预激活残差模块。Wide ResNet[16]通过加 ResNet 的宽度，从而更加有效地提升 ResNet 的性能。ResNeXt[17]将分组卷积融合到残差瓶颈模块中，提出了多路信息传输的结构。Inception 系列[18,19,20]提出了多路的、多核的级联连接模块。DenseNet[21]提出了密集级联连接模块，有效地减少了网络参数量。DPN[22]通过结合残差连接和密集级联连接，使特征既能重利用，又能再发现。

最近，各种不同形式的注意力被应用到卷积神经网络中，有效地提升了网络的性能。图 1 列举了卷积神经网络中三种经典的注意力。SENet[23]开创性地提出了 Squeeze-and-Excitation (SE)通道注意力，自适应地重标定通道特征响应。MobileNetV3[24]考虑到 SE 通道注意力中的 Sigmoid 计算代价是昂贵的，提出了计算更轻量的 HardSigmoid 来替换 Sigmoid。SKNet[25]提出了选择核 (Selective Kernel, SK) 注意力，自适应地选择不同卷积核尺寸的分支。ResNeSt[26]提出了分割 (Split) 注意力，自适应地选择分割分支。在这三种注意力中，SE 通道注意力通 Sigmoid 计算通道权重。SK 注意力和 Split 注意力则通过 Softmax 计算不同分支的通道权重。

随着计算算力的提高，网络架构的设计已经从手工设计转移到自动搜索。MnasNet[27]在 MobileNetV2[28]结构的基础上，引入了 SE 通道注意力。MobileNetV3 通过引入平台神经网络适配 (platform-aware neural network adaptation) 和 HardSwish 激活函数，扩展 MnasNet。EfficientNet[29]仍然在 MobileNetV2 结构的基础上，引入了模型复合压缩方法，在网络效率和准确率之间达到了很好的平衡。这些方法主要基于强化学习[30,31,32]、进化搜索[33]、可微搜索[34]或其他学习算法[31,35]。

然而，不管是手工设计的网络架构，还是自动搜索的网络架构，它们彼此不同，这会使得下游网络难以建立。但是，注意力可以在几乎不改变原来网络架构的同时，以额外的、非常轻量的参数量和计算复杂度，有效地提升网络性能。另外，注意力还可以扩大神经网络架构自动搜索的空间，并潜在地提高整体性能。因此，研究更好的、更轻量的注意力是非常重要的。本文提出了基于 HardSoftmax 的并行选择核 (Parallel Selective Kernel, PSK) 注意力。首先，针对 Softmax 包含指数运算，对于较大的正输入很容易发生计算溢出的问题，本文提出了计算更安全的 HardSoftmax 来替换 Softmax。然后，不同于 SK 注意力将全局特征的提取和转换放在特征融合之后，PSK 注意力将全局特征的提取和转换单独放在一个分支，与具有不同核大小的多个分支构成并行结构。同时，PSK 注意力的全局特征转换使用分组卷积，进一步减少参数量和计算量。最后，PSK 注意力通过 HardSoftmax 注意来关注不同核大小的多个分支。一系列的图像分类实验表明，简单地用 Hardsoftmax 替换 Softmax，也能保持或提升原注意力的性能。Hard-Softmax 的运行速度在实验中也比 Softmax 更快速。PSK 注意力能够以更少的参数量和计算量追平或超越 SK 注意力。

本文剩余的内容安排如下：第 2 节提出了计算更安全的 HardSoftmax 和基于 HardSoftmax 的并行选择核注意力。第 3 节展示和分析了实验结果。第 4 节陈述了结论和未来的工作。



(a) SE 通道注意力[23] (b) 选择核注意力[25] (c) 分割注意力[26]

图 1 卷积神经网络中三种经典的注意力。gap 表示全局平均池化。gconv 表示分组卷积。

Fig.1 Three classical attentions in CNNs. gap denotes global average pooling. gconv denotes group convolution

1 HardSoftmax 定义和基于 HardSoftmax 的并行选择核注意力

本节首先定义了 HardSoftmax，然后介绍了基于 HardSoftmax 的并行选择核注意力。

1.1 HardSoftmax 定义

Softmax 的定义如公式(1)所示：

$$\mathrm{Softmax}(X_i)=\frac{e^{X_i}}{\sum_j^J e^{X_j}}\#(1)$$

其中X_i为第i个节点的输出值，J为输出节点的个数。通过 Softmax 可以将输出值转换为范围在$[0,1]$及和1的概率分布。众所周知，对于较大的正输入，指数运算是很容易发生计算溢出的。为了解决这个问题，常用的方法是将每一个输出值减去输出值中最大的值，从而让输出值小于或等于0,那么进行指数运算就不会发生计算溢出，如公式(2)所示：

$$\begin{cases}M=\max(X)\\ \mathrm{Softmax}(X_i)=\dfrac{e^{X_i-M}}{\sum_j^J e^{X_j-M}}\end{cases}\#(2)$$

不同于公式(2)所示的解决方法，本文的想法是寻找计算更安全的$E(X_i)$，来模拟指数函数e^{X_i}的形状，从而保留 Softmax 相似的分布特性。为了设计$\mathrm{E}(X_i)$来模拟e^{X_i}，本文首先提出了一个新颖的激活函数，被称为幂线性单元 (Power Linear Unit, PLU)，定义如公式(3)所示：

$$\mathrm{PLU}(X_i)=\max\left(X_i,\frac{\alpha X_i}{1+|X_i|}\right)=\begin{cases}X_i & if\ X_i>0\\ \dfrac{\alpha X_i}{1+|X_i|} & if\ X_i\le 0\end{cases}\#(3)$$

其中，α为一个预设的固定值，满足$\alpha\in(0,1]$，通常$\alpha=0.5$。

幂线性单元的一阶导函数如公式(4)所示：

$$\mathrm{PLU}'(X_i)=\begin{cases}1 & if\ X_i>0\\ \dfrac{\alpha}{(1+|X_i|)^2} & if\ X_i\le 0\end{cases}\#(4)$$

显然，$\forall X_i\in\boldsymbol{R},\mathrm{PLU}'(X_i)>0$，所以$\mathrm{PLU}(X_i)$是严格地单调递增。当$X_i\to-\infty,\mathrm{PLU}(X_i)=\frac{\alpha}{-1+\frac{1}{X_i}}\to-\alpha$。

为了更好地模拟e^{X_i}的形状，这里令$\alpha=1$。那么有：$\mathrm{PLU}'(0^+)=\mathrm{PLU}'(0^-)=1$，以及$X_i\to-\infty,\mathrm{PLU}(X_i)\to-1$。也就是说，当$\alpha=1$时，$\mathrm{PLU}(X_i)$处处可导，严格地单调递增，以$Y_i=-1$为下界，无上界。图 2 绘制了当$\alpha=1$时，$\mathrm{PLU}(X_i)$的函数图像。

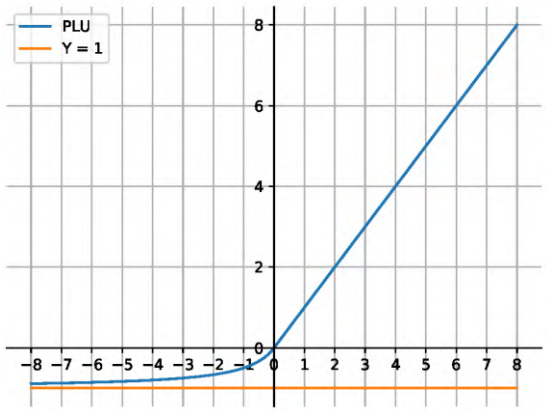


图 2　当$\alpha = 1$时，PLU(X_i)的函数图像

Fig.2　The shape of PLU(X_i) when $\alpha = 1$

然后，本文让PLU(X_i)向上平移一个单位，满足PLU(X_i) + 1 > 0。最后，本文用E(X_i) = PLU(X_i) + 1替换 Softmax 中的e^{X_i}，从而构造了计算更安全的HardSoftmax，定义如公式(5)所示：

$$\begin{cases} X \leftarrow \max\left(X_i, \dfrac{\alpha X_i}{1+|X_i|}\right)+1 \\ \text{Softmax}(X_i) = \dfrac{X_i}{\sum_j^J X_j} \end{cases} \#(5)$$

1.2 基于 HardSoftmax 的并行选择核注意力

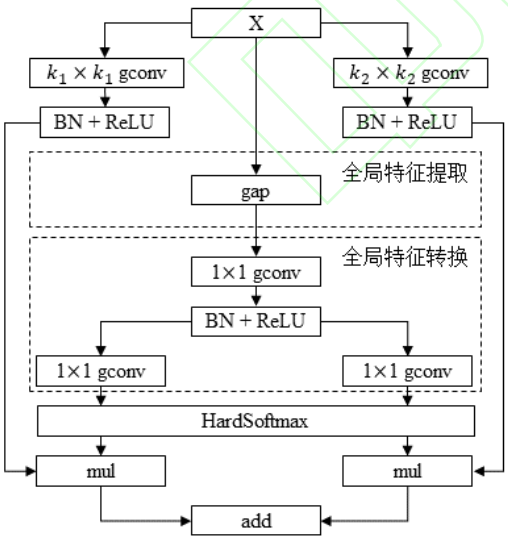


图 3　基于 HardSoftmax 的并行选择核注意力

Fig.3　Parallel Selective Kernel Attention Based on Hard-Softmax

基于 HardSoftmax 的并行选择核注意力如图 3 所示，它可以抽象成公式(6)：

$$\tilde{X} = \sum_{k=1}^{M} X_k G_k^{C_{out}} \#(6)$$

公式(6)中C_{out}表示不同核大小分组卷积(如图 3 中的$k_1 \times k_1$和$k_2 \times k_2$的分组卷积，其中k_1通常等于3，k_2通常等于5) 的输出通道数；M表示不同核大小的分支数量(如图 3 中的2个分支)。其中，X_k满足公式(7)，f_k表示$GConv_k$[17]→ BN_k[18]→ $ReLU_k$[36]。$G_k^{C_{out}}$又满足公式(8)和(9)，g表示$GConv \rightarrow BN \rightarrow ReLU \rightarrow GConv$。公式(9)先通过本文提出的 HardSoftmax 计算，再进行通道分割。

$$X_k = f_k(X) \#(7)$$

$$G^{C_{out} \times M} = g(X) \#(8)$$

$$\sum_{k=1}^{M} G_k^{C_{out}} = 1 \#(9)$$

在全局特征转换模块中，1 × 1分组卷积的分组数等于不同核大小分组卷积的分组数。并且在全局特征转换模块中，第一个1 × 1分组卷积的输出通道数 (即第二个1 × 1分组卷积的输入通道数) 的按公式(10)计算：

$$\tilde{C} = \max(\max(C_{in}, C_{out} \times M) \div R, G) \#(10)$$

这里C_{in}为不同核大小分组卷积的输入通道数；R为一个正整数，通常为4；G表示不同核大小分组卷积的分组数。如果采用标准的卷积，那么按公式(11)计算：

$$\tilde{C} = \max(C_{in}, C_{out} \times M) \div R \#(11)$$

与图 1b 显示的 SK 注意力相比，本文提出的 PSK 注意力有如下不同：

1）全局特征的提取和转换被单独放在一个分支，与具有不同核大小的多个分支构成并行结构。那么，这些分支可以并行地运行，从而降低整体的计算延迟。

2）全局特征的转换，使用1 × 1的分组卷积(group convolution, gconv)[17]。与标准的卷积相比，分组卷积已经被证明[37]，可以有效地降低参数量和计算量，同时只会降低很少的性能。

3）全局特征的转换，第一个1 × 1的分组卷积后面(即 ReLU 前面) 使用了 BN 算法，帮助权重学习更加稳定。

4）使用 HardSoftmax 注意，来关注不同卷积核大小的多个分支。HardSoftmax 保留了 Softmax 相似的分布特性，但它的计算更安全。

2 实验及结果分析

本节展示了将 SE 通道注意力、SK 注意力、Split 注意力和基于 HardSoftmax 的 PSK 注意力分别融合到不同骨干网络的实验结果。本节也展示了 SE 通道注意力分别使用 Sigmoid 和HardSigmoid的对比结果，以及 SK 注意力和 Split 注意力分别使用 Softmax 和 HardSoftmax 的对比结果。

2.1 实验环境

本文的所有实验结果都在配置如表 1 描述的计算机上完成的：

表 1　实验环境配置

Table 1　Hardware and software setups.

中央处理器	一块英特尔(R) 酷睿(TM) i5-8500 CPU
显卡	一块英伟达 Quadro RTX 4000 8GB
内存	32GB
Python	3.8.3
PyTorch[38]	1.6.0
Cuda	10.2
Cudnn	8.0.3

2.2 数据集

Fashion-MNIST. Fashion-MNIST 数据集[39]由 10 个类别、70,000 幅时尚产品图像组成。每幅图像是 28 × 28像素的灰度图像。每个类别包含 7,000 幅图像。训练集包含 60,000 幅图像，评估集包含 10,000 幅图像。

CIFAR. CIFAR 数据集[40]由32 × 32像素的 RGB 图像组成。CIFAR-10 数据集包含 10 个类别，CIFAR-100 包含 100 个类别。训练集和评估集分别包含 50,000 和 10,000 幅图像。

2.3 模型细节

本文采用两种骨干网络：ResNeXt[17] 和 MobileNetV3[24]。然后将 SE 通道注意力、SK 注意力、Split 注意力和基于 HardSoftmax 的 PSK 注意力别融合到这两种骨干网络中。值得注意的是，MobileNetV3-Small 中的某些线性反转瓶颈模块本身就包含了 SE 通道注意力。因此，本文将 MobileNetV3-Small 中原来包含的 SE 通道注意力全部删除，再作为骨干网络。为了适合28 × 28和32 × 32像素的图像，ResNeXt 和 MobileNetV3 都只保留最后三次下采样。另外，ResNeXt 的宽度变为原来的一半，即所有层的通道数变为原来的一半。

表 2　不同模型在不同数据集上的分类准确率(%)。

Table 2　Classification accuracy rate (%) of different models on different datasets.

	FM[39]	C10[40]	C100[40]	Params	MAdds
ResNeXt50 $(16 \times 4d)$[17]	94.09	93.10	74.49	6.22	726.63
SENet50 $(16 \times 4d)$[23]	94.01	93.17	74.18	8.74	731.66
SENet50 $(16 \times 4d)$ + HardSigmoid[24]	94.07	92.83	75.20	8.74	731.66
SKNet50 $(16 \times 4d)$[25]	94.23	93.20	75.02	10.09	941.49
SKNet50 $(16 \times 4d)$ + HardSoftmax (本文)	**94.32**	**93.79**	74.94	10.09	941.49
ResNeSt50 $(16 \times 4d \times 2r)$[26]	94.29	93.13	74.56	7.06	803.75
ResNeSt50 $(16 \times 4d \times 2r)$ + HardSoftmax (本文)	94.17	93.41	74.62	7.06	803.75
ResNeXt50 $(16 \times 4d)$ + PSK (本文)	94.19	93.52	75.00	8.32	937.93
ResNeXt50 $(16 \times 4d)$ + PSK (本文) + 标准1 × 1卷积	94.25	93.71	**75.78**	10.09	941.47
MobileNetV3-Small[24]	92.93	91.02	70.93	1.16	35.54
MobileNetV3-Small + SE	93.08	90.98	71.53	1.62	36.46
MobileNetV3-Small + SE + HardSigmoid	92.58	90.71	71.15	1.62	36.46
MobileNetV3-Small + SK	92.97	91.98	71.63	2.57	42.97
MobileNetV3-Small + SK + HardSoftmax (本文)	93.08	**92.14**	71.48	2.57	42.97
MobileNetV3-Small + Split (r=2)	93.34	91.19	**72.09**	1.24	40.85
MobileNetV3-Small + Split (r=2) + HardSoftmax (本文)	**93.49**	91.67	71.77	1.24	40.85
MobileNetV3-Small + PSK (本文)	93.32	91.61	71.66	1.20	40.21
MobileNetV3-Small + PSK (本文) + 标准1 × 1卷积	93.10	91.77	71.59	2.57	42.95

2.4 训练细节

输入到模型的图像采用减去均值，再除以标准差的方式进行预处理。本文还使用了数据增强，包括随机旋转、随机平移和随机水平翻转。所有的模型都采用交叉熵 (categorical cross entropy) 损失函数和跟随文献[41]进行初始化。在 Fashion-MNIST 数据集上，所有的模型都使用 AdamW[42]进行训练；在 CIFAR 数据集上，所有的模型都使用 SGDM[43]进行训练。对于骨干网络为 MobileNetV3-Small 的模型，批处理大小为 256；对于骨干网络为 ResNeXt 的模型，批处理大小为 32。在 Fashion-MNIST 数据集上，训练周期为 60，在前 20 个周期内，初始学习率为 0.001，接下来的 40

个周期，每隔 20 个周期，学习率衰减为原来的 0.1 倍。在 CIFAR 数据集上，训练周期为 150，在前 80 个周期内，初始学习率为 0.1，接下来的 70 个周期，每隔 35 个周期，学习率衰减为原来的 0.1 倍。本文还采用了 L2 权重衰减，从而帮助训练过程更加稳定。对于所有卷积层和全连接层的权重，权重衰减率为 5×10^{-4}。

2.5 HardSoftmax 的实验结果及分析

表 2 显示了不同模型在 Fashion-MNIST、CIFAR-10 和 CIFAR-100 数据集上的分类准确率。Params 的单位为百万(million)。MAdds 表示先做乘法再做加法的运算次数，单位为百万(million)。对于骨干网络相同的不同模型，加粗表示每列的最优结果。表中的 Params 和 multiply adds (MAdds) 是跟随 torchstat 计算的，输入为32×32像素的 RGB 图像，类别数为 100。不管是标准的 SE 通道注意力还是使用 Hard-Sigmoid 的 SE 通道注意力，当它们融合到骨干网络中，不能一致地提升原骨干网络的分类准确率。即 SENet50$(16 \times 4d)$和[SENet50 $(16 \times 4d)$ + Hard-Sigmoid] 的分类准确率并不是总能优于 NeXt50$(16 \times 4d)$。同理，[MobileNetV3-Small + SE] 和 [MobileNetV3-Small + SE + HardSigmoid] 也是如此。但是，不管是使用 Softmax 的 SK 注意力和 Split 注意力，还是使用 HardSoftmax 的 SK 注意力和 Split 注意力，当它们融合到骨干网络中，不能一致地提升原骨干网络的分类准确率。

表 3 显示了 HardSoftmax 和 Softmax 的比较结果。表 3 的结果是通过比 SK 注意力和 Split 注意力分别使用 HardSoftmax 和 Softmax 的准确率汇总而来的。结果表明，使用 HardSoftmax 的注意力在大多数情况下都能够超越使用 Softmax 的注意力。

表 3 HardSoftmax 优于或劣于使用 Softmax 的数量。

Table 3 The number of HardSoftmax outperforming or underperforming Softmax.

HardSoftmax (本文) > Softmax	8
HardSoftmax (本文) < Softmax	4

计算复杂度是评价 Softmax 和 HardSoftmax 优劣的另一个重要指标。在实际评估中，本文实现的 HardSoftmax 是慢于 PyTorch 标准实现的 Softmax。本文猜测 PyTorch 标准实现，对 Softmax 进行了并行加速。为了最大的公平，表 4 列举了 PyTorch 标准实现的 Softmax、本文实现的 Softmax 和本文实现的 Hard-Softmax 的速度比较结果。速度是通过在 CIFAR-100 数据集上训练和评估 SKNet50$(16 \times 4d)$一个周期的时间进行衡量的。结果表明，本文实现的 HardSoft-max 略快于本文实现的 Softmax。结果也表明，本文实现的 Softmax 明显比 PyTorch 标准实现的 Softmax 更慢。所以，本文有理由相信，如果 HardSoftmax 也是 PyTorch 的标准函数，那么 HardSoftmax 的实际运行速度一定会获得更大的收益。

表 4 HardSoftmax 和 Softmax 的速度比较 单位:秒

Table 4 Comparison of speed between HardSoftmax and Softmax. Speed is in second

	训练	评估
Softmax (PyTorch)	**154.30**	**6.00**
Softmax (本文)	161.51	6.01
HardSoftmax (本文)	160.47	**6.00**

2.6 基于 HardSoftmax 的并行选择核注意力的实验结果及分析

基于 HardSoftmax 的 PSK 注意力的实验结果仍然列举在表 2 中。从结果可知，当 PSK 注意力融合到骨干网络中，依然能够稳定地提高骨干网络的准确率。与[SK 注意力+ HardSoftmax] 相比，PSK 额外增加的 Params 和 MAdds 更少，同时准确率也能几乎不变或提升。当骨干网络是 ResNeXt50 $(16 \times 4d)$时，PSK 注意力额外增加的 Params 和 MAdds 更多，但它的准确率总是优于[Split 注意力+ HardSoftmax] 的准确率。然而，当骨干网络是 MobileNetV3-Small 时，PSK 注意力额外增加的 Params 和 MAdds 更少，但它的准确率总是劣于[Split 注意力+ HardSoftmax] 的准确率。本文认为，MobileNetV3-Small 中的3×3或 5×5卷积核尺寸是通过自动搜索出来的，所以不同层的卷积核尺寸已经是最优的结果，所以当骨干网络是 MobileNetV3-Small 时，不管是 SK 注意力，还是本文提出的 PSK 注意力，准确率都会劣于 Split 注意力的准确率。

表 2 还显示了 PSK 注意力的消融实验结果。这里的消融实验是指，在 PSK 注意力的全局特征转换模块中，使用标准1×1卷积，再和 [SK 注意力 + HardSoftmax] 进行比较。由结果可知，随着参数的增加，[PSK 注意力+ 标准1×1卷积] 明显优于 PSK 注意力。表 5 显示了 [PSK 注意力 + 标准1×1卷积] 和[SK 注意力 + HardSoftmax] 的比较结果。表 5 的结果是通过比较 [PSK 注意力 + 标准1×1卷积] 和 [SK 注意力 + HardSoftmax] 的准确率汇总而来的。结果表明，[PSK 注意力+ 标准1×1卷积] 是和[SK 注意力 + HardSoftmax] 持平的。再结合表 2 中 MAdds 分析，[PSK 注意力 + 标准1×1卷积] 的 MAdds 是

略低于 [SK 注意力 + HardSoftmax] 的。这说明，将全局特征的提取和转换被单独放在一个分支，与具有不同核大小的多个分支构成并行结构，并不会带来准确率的损失，却可以带来信息多路传输的优势，充分利用显卡的并行性，从而降低整体的计算延迟。

表 5　消融实验，PSK 注意力优于或劣于 SK 注意力的数量

Table 5　Ablation experiment, the number of PSK attention outperforming or underperforming SK attention

PSK (本文) + 标准1 ×1卷积> SK + HardSoftmax	3
PSK (本文) + 标准1 ×1卷积< SK + HardSoftmax	3

3　结束语

本文提出了计算更安全的 HardSoftmax 和基于 HardSoftmax 的并行选择核注意力。一系列的图像分类实验表明，简单地用 Hardsoftmax 替换 Softmax，也能保持或提升原注意力的性能。HardSoftmax 的运行速度在实验中也比 Softmax 更快速。基于 HardSoftmax 的并行选择核注意力能够以更少的参数量和计算量追平或超越选择核注意力。未来的工作包括在更多场景下比较 HardSoftmax 和 Softmax 的性能，以及测试并行选择核注意力的性能。

参考文献

[1] Min W, Fan M, Guo X, et al. A new approach to track multiple vehicles with the combination of robust detection and two classifiers[J]. IEEE Transactions on Intelligent Transporta-tion Systems, 2018, 19(1): 174-186

[2] Min W, Guo X, Han Q. An improved vibe algorithm and its application in traffic video processing[J]. Guangxue Jingmi Gongcheng/Opt. Precis. Eng., 2017, 25(3): 806-811.

[3] Zhou L, Min W, Lin D, et al. Detecting motion blurred vehicle logo in iov using filterdeblurgan and vl-yolo[J]. IEEE Transactions on Vehicular Technology, 2020, 69(4): 3604-3614.

[4] Yao L, Min W, Lu K. A new approach to fall detection based on the human torso motion model[J]. Applied Sciences, 2017

[5] Min W, Cui H, Rao H, et al. Detection of human falls on furniture using scene analysis based on deep learning and activity characteristics[J]. IEEE Access, 2018, 6(): 9324-9335

[6] Xiong X, Min W, Zheng W, et al. S3d-cnn: Skeleton-based 3d consecutive-low-pooling neural network for fall detection[J]. Applied Intelligence, 2020, 1-14

[7] 赵中堂，陈继光，马倩。摔倒检测中的样本失衡问题研究 [J]。计算机工程与应用，2017，53(23)：142-146
(Zhao Z, Chen J, Ma Q. Research on sample imbalance in fall detection[J]. Computer Engineering and Applications, 2017,53 (23): 142-146)

[8] Min W, Shi J, Han Q. A distributed face recognition method and performance optimization[J]. Optics and Precision Engineering, 2017, 25(3): 779-785

[9] Zou F, Li J, Min W. Distributed face recognition based on load balancing and dynamic prediction[J]. Applied Sciences, 2019.

[10] 董艳花，张树美，赵俊莉。有遮挡人脸识别方法综述[J]。计算机工程与应用，2020，56(9)：1-12
(Dong Y, Zhang S, Zhao J. Overview of occluded face recognition methods[J]. Computer Engineering and Applications, 2020,56 (9): 1-12).

[11] Krizhevsky A, Sutskever I, Hinton G E. Imagenet classification with deep convolutional neural networks[C]. Advances in Neural Information Processing Systems, 2012, 1097-1105.

[12] Szegedy C, Liu W, Jia Y, et al. Going deeper with convolutions[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2015, 1-9.

[13] Simonyan K, Zisserman A. Very deep convolutional networks for largescale image recognition[C]. International Conference on Learning Representations, 2015.

[14] He K, Zhang X, Ren S, et al. Deep residual learning for image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016, 770-778.

[15] He K, Zhang X, Ren S, et al. Identity mappings in deep resid-ual networks[C]. European Conference on Computer Vision, 2016, 630-645.

[16] Zagoruyko S, Komodakis N. Wide residual networks[C]. Proceedings of the British Machine Vision Conference, 2016, 8701-8712.

[17] Xie S, Girshick R, Dollar P, et al. Aggregated residual transformations for deep neural networks[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017, 1492-1500.

[18] Ioffe S, Szegedy C. Batch normalization: Accelerating deep network training by reducing internal covariate shift[C]. International Conference on Machine Learning, 2015, 448-456.

[19] Szegedy C, Vanhoucke V, Ioffe S, et al. Rethinking the inception architecture for computer vision[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2016, 2018-2826.

[20] Szegedy C, Ioffe S, Vanhoucke V, et al. Inception-v4, inception-resnet and the impact of residual connections on learning[C]. AAAI Conference on Artificial Intelligence, 2017, 2018-2826.

[21] Huang G, Liu Z, Maaten L, et al. Densely connected convolutional networks[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2017, 4700-4708.

[22] Chen Y, Li J, Xiao H, et al. Dual path networks[C]. Advances in Neural Information Processing Systems, 2017, 4467-4475.

[23] Hu J, Shen L, Sun G. Squeeze-and-excitation networks[C].

IEEE Conference on Computer Vision and Pattern Recognition, 2018, 7132-7141.
[24] Howard A, Sandler M, Chu G, et al. Searching for mobilenetv3[C]. IEEE International Conference on Computer Vision, 2019, 1314-1324.
[25] Li X, Wang W, Hu X, et al. Selective kernel networks[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2019, 510-519.
[26] Zhang H, Wu C, Zhang Z, et al. Resnest: Split-attention-networks[OL],2020. https://arxiv.org/pdf/2004.08955.pdf.
[27] Tan M, Chen B, Pang R, et al. Mnasnet: Platform-aware neural architecture search for mobile[C]. IEEE/CVF Conference on Computer Vision and Pattern Recognition, 2019, 2820-2828
[28] Sandler M, Howard A, Zhu M, et al. Mobilenetv2: In-verted residuals and linear bottlenecks[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018, 4510-4520
[29] Tan M, Le Q. Efficientnet: Rethinking model scaling for convolutional neural networks[C]. Proceedings of the 36th International Conference on Machine Learning, 2019, 6105-6114
[30] Baker B, Gupta O, Naik N, et al. Designing neural network architectures using reinforcement learning[C]. International Conference on Learning Representations, 2017
[31] Chen X, Xie L, Wu J, et al. Progressive differentiable architecture search: Bridging the depth gap between search and evaluation[C]. IEEE/CVF International Conference on Com-puter Vision, 2019, 294-303
[32] Zoph B, Vasudevan V, Shlens J, et al. Learning transferable architectures for scalable image recognition[C]. IEEE Conference on Computer Vision and Pattern Recognition, 2018, 8697-8710
[33] Real E, Aggarwal A, Huang Y, et al. Regularized evolution for image classifier architecture search[J]. Proceedings of the AAAI Conference on Artificial Intelligence, 2019, 33(1): 4780-4789
[34] Liu H, Simonyan K, Yang Y. Darts: Differentiable architecture search[C]. International Conference on Learning Representations, 2019
[35] Kandasamy K, Neiswanger W, Schneider J, et al. Neural architecture search with bayesianoptimisation and optimal transport[C]. Advances in Neural Information Processing Systems, 2018, 2016-2025
[36] Nair V, Hinton G E. Rectified linear units improve restricted boltzmann machines[C]. International Conference on Machine Learning, 2010, 807–814
[37] Haward A G, Zhu M, Chen B, et al. MobileNets: Efficient convolutional neural networks for mobile vision applications[OL]. http://arxiv.org/abs/1704.04861
[38] Paszke A, Gross S, Massa F, et al. Pytorch: An imperative style, highperformance deep learning library[C]. Advances in Neural Information Processing Systems, 2019, 8026-8037
[39] Xiao H, Rasul K, Vollgraf R. Fashion-mnist: A novel image dataset for benchmarking machine learning algorithms[OL], 2017. http://arxiv.org/abs/1708.07747
[40] Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images[OL], 2009. http://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.222.9220&rep=rep1&type=pdf.
[41] He K, Zhang X, Ren S, et al. Delving deep into rectifiers: Surpassing human-level performance on imagenet classification[C]. IEEE International Conference on Computer Vision, 2015, 1026-1034.
[42] Loshchilow H, Hutter F. Decoupled weight decay regularization[C]. International Conference on Learning Representations, 2019.
[43] Sutskever I, Martens J, Dahl G, et al. On the importance of initialization and momentum in deep learning[C]. International Conference on Machine Learning, 2013, 1139-1147.