

基于胶囊网络及其权重剪枝的 SAR 图像变化检测方法

陈志文¹ 王 坤¹ 周广蕴² 王 旭² 张晓丹² 朱虎明¹

1 西安电子科技大学智能感知与图像理解教育部重点实验室 西安 710071

2 北京机电工程研究所 北京 100074

(18829535531@163.com)

摘 要 基于深度神经网络的 SAR 图像变化检测算法由于精确率高等优点,已被广泛应用于农业检测、城市规划以及森林预警等多个领域。设计了基于胶囊网络的 SAR 图像变化检测算法,针对其模型复杂度高、参数量大等问题,提出了基于权重剪枝的模型压缩方法。该方法对其胶囊网络参数进行逐层分析,针对不同类型的层采取不同的剪枝策略,对网络中冗余的参数进行剪枝,随后对剪枝后的网络进行微调,从而提高了剪枝后模型的检测性能。最后,通过对模型中保留下来的参数进行压缩存储,显著降低了模型所占用的存储空间。在 4 组真实 SAR 图像上的实验结果证明了所提出的模型压缩方法的有效性。

关键词: SAR 图像;变化检测;胶囊网络;模型压缩;权重剪枝

中图法分类号 TP391

SAR Image Change Detection Method Based on Capsule Network with Weight Pruning

CHEN Zhi-wen¹, WANG Kun¹, ZHOU Guang-yun², WANG Xu², ZHANG Xiao-dan² and ZHU Hu-ming¹

1 Key Laboratory of Intelligent Perception and Image Understanding of Ministry of Education, Xidian University, Xi'an 710071, China

2 Beijing Electro-mechanical Engineering Institute, Beijing 100074, China

Abstract The SAR image change detection algorithm based on deep neural network has been widely used in many fields such as agricultural detection, urban planning and forest early warning due to its high accuracy. This paper designs a SAR image change detection algorithm based on capsule network. In view of its high model complexity and large number of parameters, a model compression method based on weight pruning is proposed. This method performs layer-by-layer analysis of its capsule network parameters, adopts different pruning strategies for different types of layers, prunes redundant parameters in the network, and then fine-tunes the pruned network to improve the detection performance of the model. Finally, by compressing and storing the parameters retained in the model, the storage space occupied by the model is significantly reduced. Experiments on four datasets of real SAR images prove the effectiveness of the proposed model compression method.

Keywords SAR image, Change detection, Capsule network, Model compression, Weight pruning

1 引言

近年来,深度神经网络发展迅速。在 SAR 图像变化检测^[1]这个领域,应用深度神经网络的实验结果大大优于传统算法,且随着神经网络结构复杂度以及参数数量的增加,检测效果会更好,但同时其计算复杂度和计算资源开销也会随之增加。SAR 图像变化检测主要应用在军事侦察、灾难检测等方面,对其检测效率有很高的要求。目前性能良好的变化检测模型具有计算量大以及参数多等特点,这些特点在很大程度上制约了其在资源受限的环境和实时在线处理系统中的应用。如果能对模型进行适当的改进,去除冗余的参数或者优化网络结构,那么就可以将它强大的检测能力移植到计算能

力较弱的设备中。减少变化检测模型的计算量和压缩模型体积对于提高其检测效率和加速模型的部署都有很大的促进作用。

本文针对 SAR 图像中相干斑噪声与真实像素点混杂等特点,设计了适用于 SAR 图像变化检测的胶囊网络^[2],并在该胶囊网络模型的基础上,提出了适用于该网络的模型压缩方法,然后在多组真实 SAR 数据集中进行实验,验证了它的有效性。

2 理论概述

2.1 SAR 图像变化检测的基本流程和相关技术

典型的 SAR 图像变化检测流程如图 1 所示。对于两幅

到稿日期:2020-08-31 返修日期:2021-01-24

基金项目:教育部产学研合作协同育人项目(201901159006);国防基础科研计划资助(JCKY2016204A102)

This work was supported by the Cooperative Education Project of the Ministry of Education(201901159006) and National Defense Basic Scientific Research Program Funding(JCKY2016204A102).

通信作者:朱虎明(zhuhum@mail.xidian.edu.cn)

图像数据,先进行一些必要的预处理,然后根据具体问题选择合适的算法,最后对算法的性能进行定量的分析。

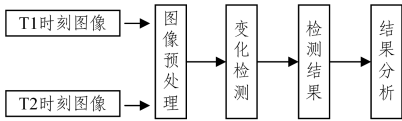


图1 SAR 图像变化检测流程图

Fig. 1 Flow chart of SAR image change detection

2.2 胶囊网络

传统卷积神经网络^[3]在对图像进行特征提取后,为了保留特征并减少计算量,会增加池化层对通道进行降维,同时保留必要的特征。这样做虽然保留了原始特征图中的重要特征,但是会丢失特征中的一些拓扑信息。2017 年,Sabour 等提出了胶囊网络^[2],该网络不同于以往的卷积神经网络,因为其中不仅包含卷积网络部分,还包含了被称为胶囊的神经元。胶囊神经元是一个由一组标量神经元组成的向量神经元,向量的各个分量表示胶囊神经元对于物体不同方面的感知,如大小、位置等。因此,胶囊神经元对于特征的编码体现在向量的方向上,对于特征出现的概率则编码为向量的长度,这使得胶囊神经元具有很大的灵活性,该神经元对于特征的感知不仅仅体现在特征的强度上,而且还包含了特征的各种状态。这种性质为胶囊网络带来了很大的好处,胶囊神经元克服了卷积神经网络只根据局部特征的有无而进行判别的缺陷。因此,胶囊网络对于数据的泛化能力较强,这使得在训练网络时不需要做过多的数据增强,并且不需要刻意增加网络的深度来获得更好的特征学习能力,同时胶囊网络对特征的判别能力较强,能清楚地分辨不同的特征。胶囊网络的这些特性非常适合处理 SAR 图像,因此引入胶囊网络来实现 SAR 图像变化检测是一个可行的新想法。

胶囊网络中的胶囊神经元的结构如图 2 所示。从图 2 可见,上一层的各个胶囊与下一层的各个胶囊都有连接,这类似于全连接层的连接方式。以后面一层的胶囊 v_i 为例,与它连接的前一层的各个胶囊是 $u_1, u_2, \dots, u_i$,那么每个连接都有对应的权重矩阵 $w_{1j}, w_{2j}, \dots, w_{ij}$,以及两个胶囊是否连接的对数先验概率 $b_{1j}, b_{2j}, \dots, b_{ij}$ 。在进行前向传播时,上一层的胶囊先与对应的权重矩阵相乘完成姿态变换:

$$\hat{u}_{j|i} = w_{ij} u_i \tag{1}$$

然后,与后一层胶囊 v_j 连接的所有前层胶囊进行加权求和:

$$s_j = \sum_i c_{ij} \hat{u}_{j|i} \tag{2}$$

其中, c_{ij} 为前一层中的第 i 个胶囊与后一层中的第 j 个胶囊之间的耦合系数,其计算方法如下:

$$c_{ij} = \frac{e^{b_{ij}}}{\sum_k e^{b_{ik}}} \tag{3}$$

在获取所有前层胶囊输入的加权和后,需要激活函数进行变换,胶囊中的激活函数为 squash^[4],它的定义如下:

$$v_j = \frac{\|s_j\|^2}{1 + \|s_j\|^2} \cdot \frac{s_j}{\|s_j\|} \tag{4}$$

从式(4)可以看出,较短的输入经激活之后长度接近于 0,较长的输入经过激活之后长度接近于 1,激活函数不会改变输入的方向。

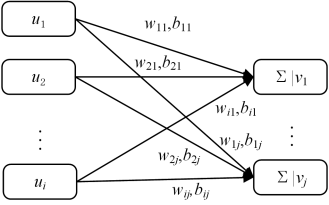


图2 胶囊神经元的连接结构图

Fig. 2 Connection structure of capsule neurons

胶囊网络包含卷积模块和胶囊模块。其中卷积模块和大部分卷积神经网络类似,复杂度较高;胶囊模块的拓扑结构和全连接层相似,有较多的冗余且效率不高。为了在减小模型体积的同时保证检测精度的稳定,有必要研究模型压缩对胶囊网络性能的影响。

2.3 深度神经网络模型压缩方法

伴随深度模型性能的快速提升,已有许多模型压缩相关的算法^[5]被提出,这些方法从不同的角度对模型进行压缩:1)基于新型卷积计算的模型压缩。这类方法是对卷积层的卷积方式进行了合理的优化,如 Howard 等提出的 MobileNets 中的深度可分离卷积^[6]。2)基于剪枝的模型压缩^[7]。训练好的深度神经网络中含有大量参数以及复杂的结构,可以对这些参数或结构进行统计和分析,找出相对较冗余的部分,然后将其删除。3)基于参数量化的模型压缩^[8]。参数量化的基本思想是使用 fp16 等较低位数的数据类型代替 32 位浮点数,从而降低网络的计算量以及存储空间需求。4)基于知识蒸馏的模型压缩^[9]。

权重剪枝属于非结构化剪枝,它直接对模型中的非重要权值进行裁剪,然后进行再训练,该方法具有实施起来较方便且效果良好等优点。因此,本文选择使用基于权重剪枝的非结构化模型压缩方法对用于 SAR 图像变化检测的胶囊网络进行压缩。

3 基于胶囊网络和权重剪枝的 SAR 图像变化检测算法的设计

3.1 深度胶囊网络的设计

本文使用的胶囊网络分为两个模块,两者的结构不同,作用不同,它们分别用于特征提取和特征分类。前者由特征提取能力较强的卷积模块实现,后者由特征判别能力较强的胶囊神经元模块实现。下面对这两个模块分别进行说明。

(1)卷积模块

由于 SAR 图像中固有的斑点噪声,图像特征和噪声掺杂在一起,容易对网络的检测性能造成不利影响,因此本文中的卷积模块同时使用了跨层连接以及通道间加权,以充分提取和利用多层次和同层内不同通道的多样性特征^[10]。卷积模块的结构如图 3 所示。网络的输入取自两幅原始 SAR 图像中对应像素位置的图像块,然后由 3×3 大小的卷积核对图像

中不同维度的特征进行提取。图 3 中虚线箭头代表通道扩充,SE(Squeeze-and-Excitation)^[11]模块对同层中各通道的特

征图进行加权,最后将不同维度的特征进行融合,形成卷积模块的输出。

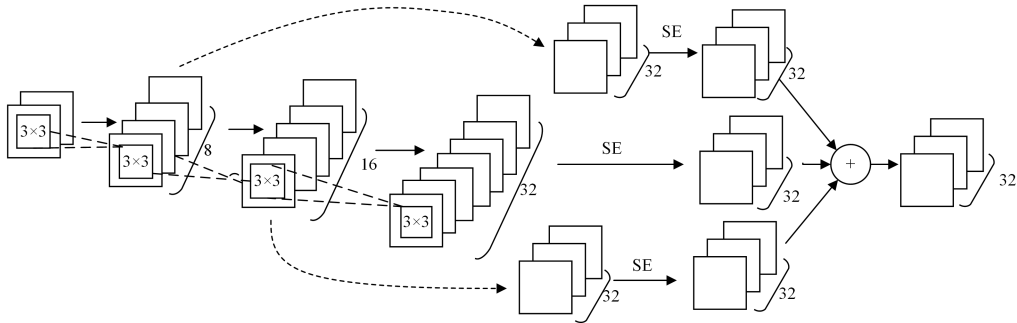


图 3 卷积模块结构图

Fig. 3 Diagram of convolution module structure

(2)胶囊神经元模块

相较于传统深度卷积神经网络,胶囊网络在 SAR 图像变化检测中的优势如下:首先,胶囊神经元是向量神经元,其中各个分量表示不同维度的特征,胶囊神经元的长度用于表示某类特征存在的强度,而卷积神经网络是使用数字表示某一特征的强度,所以胶囊神经元对于特征的提取与判断要强于卷积神经网络。其次,卷积神经网络使用池化操作对特征进行筛选与保留,但池化操作并不能随输入数据的变化而变化,而胶囊网络使用动态路由算法^[12],可以根据数据特征动态改变胶囊神经元之间的耦合系数,其具有更好的灵活性。对于 SAR 图像变化检测,图像中相干斑噪声与真实像素点混杂在一起,网络很容易将相干斑噪声理解为变化特征,从而引起误判率的增加。因此,如果希望网络对噪声进行有效的抑制,就要从多维度对特征进行判断。最后,由于胶囊网络中不含有全连接层,因此胶囊网络相较于其他深度神经网络也更为轻量。为了提高网络对 SAR 图像中噪声的判别能力以及检测效率,使用胶囊神经元对特征进行判别与分类,本文设计的胶囊神经元模块如图 4 所示。

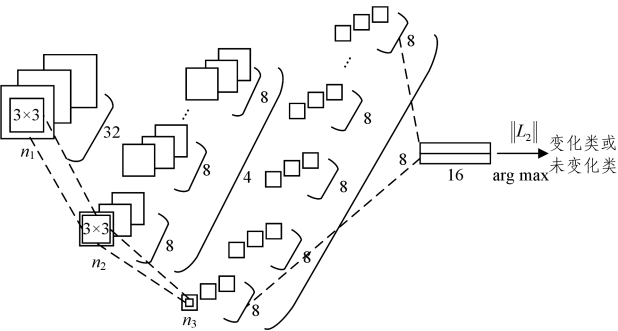


图 4 胶囊神经元模块结构图

Fig. 4 Diagram of capsule neuron module structure

该模块首先将卷积模块提取的特征组合成 4 个胶囊,每个胶囊含有 8 个特征神经元,它们表示该胶囊不同维度的特征。然后通过卷积以及动态路由算法,对该层特征进行更高维度的抽象,形成含有 8 个 8 维胶囊的特征层,该层通过动态路由算法对变化特征进行判断,形成的两个胶囊的长度表示变化以及未变化特征的强度。最后,取长度的较大者作为网络输入对应的变化类别,从而完成对输入图像块的分类。在

训练网络时,使用间隔损失^[13](margin loss)作为网络的损失函数:

$$L = \frac{1}{n} \sum_{k=1}^n (y_k \cdot \max(0, m^+ - \|v_k\|)^2 + \lambda(1 - y_k) \cdot \max(0, \|v_k\| - m^-)^2)$$

其中, n 为输出层胶囊个数; y_k 为该胶囊对应的类别是否为标签的类别,如果是则为 1,否则为 0; λ 为损失的权重系数,用于调节正反例损失,这里取预分类结果图中变化类与未变化类像素总数的比值; m^+ 和 m^- 的值分别为 0.9 和 0.1。因为本文将胶囊网络用于图像像素的分类,所以训练参数与原始网络保持一致,动态路由算法的迭代次数为 3。训练时使用的优化器为 Adam^[14]。

3.2 基于胶囊网络的 SAR 图像变化检测流程

对于 SAR 图像变化检测问题,由于检测结果图与输入的原始图像具有相同的尺寸及空间信息,且变化检测图中只有两种像素值,因此可以将该问题看作是像素分类问题,对于两幅原始图像中的每对像素值,将其分为变化或者未变化。本文采用基于像素点邻域的图像块方法完成对原始图像中像素的分类。考虑到差异算子在提取原始图像中的差异信息时可能会对原始图像中的信息造成破坏,本文的胶囊网络直接以原始图像中的像素块作为网络的输入,对网络的输出进行简单的大小比较,即可获得原始像素点所属的变化类别。算法的整体流程如图 5 所示。

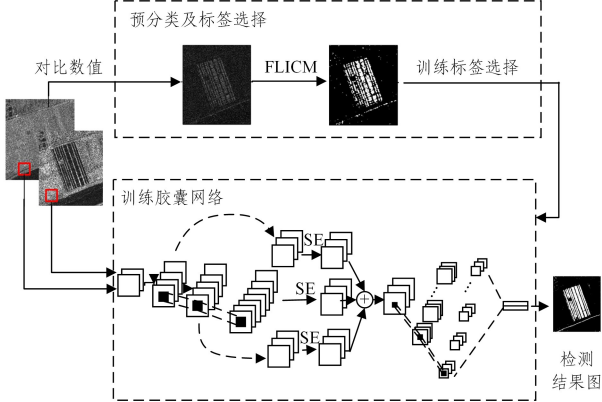


图 5 基于胶囊网络的 SAR 图像变化检测流程图

Fig. 5 Flow chart of SAR image change detection on capsule network

算法的 3 个主要步骤如下。

(1)预分类生成可用于训练胶囊网络的训练集。对于 SAR 图像变化检测,可用于研究的数据往往较少,而且几乎没有对应的真实变化检测标签可用。为了有足够的训练数据,本文采用两幅原始图像中对应像素点的邻域来构造训练集。对于训练标签集,使用局部模糊 C 均值聚类^[15] (Fuzzy Local Information C-Means, FLICM)算法生成的结果图。如图 6 所示,对于两幅原始图像中某一对应位置 (i, j) 处的像素,选取以它为中心、大小为 $n \times n$ 的图像块作为一个训练数据,选取预分类结果图中 (i, j) 处的像素值作为对应的标签。因此,对于 $H \times W$ 大小的原始图像,共可以得到 HW 个训练数据。

(2)训练深度胶囊网络。从上一步得到的训练数据集中,选取一定数量且较为可靠的训练数据以及对应的标签和一定数量的验证数据,对深度胶囊网络进行训练,保存验证集上验证误差最小的网络参数。

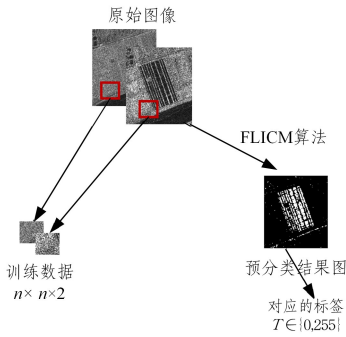


图 6 训练数据生成方法示意图

Fig. 6 Schematic diagram of training data generation method

(3)对差异图进行最终的变化检测。对于原始图像中的所有像素值,都将其为中心的邻域图像块作为上一步训练好的深度胶囊网络的输入,从而获得差异图中每个像素所属的类别。

3.3 胶囊网络各组件的剪枝方法

深度胶囊网络由卷积层对输入图像块进行不同维度的特征提取,为了保留图像块中的位置信息,没有进行池化操作,之后由胶囊层进行特征的路由及分类,该网络的结构如表 1 所列。

表 1 深度胶囊网络结构表
Table 1 Table of deep capsule network structure

Type	Input size	Kernel size	Num of kernels	Output size
input	$n \times n \times 2$	—	—	$n \times n \times 2$
conv_1	$n \times n \times 2$	3×3	8	$n \times n \times 8$
conv_2	$n \times n \times 8$	3×3	16	$n \times n \times 16$
conv_3	$n \times n \times 16$	3×3	32	$n \times n \times 32$
conv_1'	$n \times n \times 8$	1×1	32	$n \times n \times 32$
conv_2'	$n \times n \times 16$	1×1	32	$n \times n \times 32$
conv_1''	$n \times n \times 32$	3	1	$n \times n \times 32$
conv_2''	$n \times n \times 32$	3	1	$n \times n \times 32$
conv_3''	$n \times n \times 32$	3	1	$n \times n \times 32$
concat	$n \times n \times 32 \times 3$	—	—	$n \times n \times 32$
caps_1	$n \times n \times 32$	$n \times n \times 32$	32	$n' \times n' \times 8 \times 4$
caps_2	$n' \times n' \times 8 \times 4$	$3 \times 3 \times 8$	8	$n'' \times n'' \times 8 \times 8$
caps_3	$n'' \times n'' \times 8 \times 8$	—	—	16×2

表 1 中的 conv_1, conv_2, conv_3, conv_1', conv_2', conv_3' 以及 caps_1, caps_2 中的卷积都带有批量正则化, n' 和 n'' 是因为卷积采用有效填充方式并且步长为 2, 所以在输出特征图的尺寸上有所变化。

由表 1 可知,该胶囊网络中含有的基本组件有卷积、批量正则化、激活函数、特征图融合、含有胶囊神经元的胶囊层,在进行剪枝时,要根据各组件的情况采用不同的剪枝方法,具体的剪枝方法如下。

(1)卷积层的剪枝方法

卷积运算是输入图像先和卷积核中的权值进行乘法和加法运算,再将得到的和与该卷积核对应的偏置相加,最后经过激活函数得到输出。对于卷积核中的参数,其大小代表了卷积核对特征的敏感度,而卷积结果中的值的大小代表了特征的强弱。因此,为了仅保留比较重要的参数,可将较小的卷积核参数直接设置为 0。对于卷积核中的偏置,其对于提取的特征具有重要的调节作用,所以不对其进行剪枝。

(2)批量归一化层的剪枝方法

批量归一化是将批量数据归一化为均值为 0、方差为 1 的统计分布,避免数据处于激活函数的饱和区,从而加速网络的训练过程。批量归一化层参数不多,只有缩放和转换两个参数。由于 BN 层对于数据的分布起着非常重要的调节作用,且 BN 层中的参数很少,因此不对 BN 层中的参数进行剪枝。

(3)胶囊层的剪枝方法

胶囊神经元在进行计算时需要先和转移矩阵进行矩阵乘法运算以完成对神经元特征的转换,然后通过胶囊神经元之间的耦合系数进行乘法和加法运算。胶囊层的参数主要是层间的转移矩阵以及耦合系数。对于转移矩阵,由于矩阵乘法由乘加运算完成,较小的参数对结果影响较小,因此将矩阵中较小的元素去除掉;对于耦合系数,它在层间特征传递时起着关键的路由作用,所以不对其进行剪枝。

(4)其余组件的剪枝方法

对于胶囊网络,剪枝的主要对象就是卷积层和胶囊层的参数。对于网络中的其余组件,如特征图融合、激活函数、全局池化,由于这些层只是根据输入数据进行特定的计算,没有保留参数,因此不需要对其进行剪枝。

3.4 基于权重剪枝的模型压缩方法

(1)剪枝流程概述

基于权重剪枝的模型压缩方法首先需要有一个训练好的模型;然后对需要剪枝的层的参数逐层进行分析,找出该层的剪枝阈值并根据该阈值去除掉冗余的参数;最后对于保留下的较重要的权值,需要使用训练数据对模型进行训练微调,使剪枝后的模型能够较好地拟合数据。

(2)剪枝阈值的确定以及参数剪枝

本文使用基于阈值的权重剪枝方法,首先,对于网络中的某一层,需要根据其参数确定出一个阈值。假设该层有 n 个参数 $p = \{a_1, a_2, \dots, a_n\}$, 另有一个剪枝比率 λ , 为了根据剪枝比率计算出剪枝阈值,对该层的 n 个参数取绝对值之后按升

序排序,设排序之后的参数为 $p'=\{a_1',a_2',\cdots,a_n'\}$,然后使用剪枝比率根据式(5)计算出阈值所在的索引。

$$i=\lceil \lambda n \rceil \tag{5}$$

计算出阈值的索引之后,该层参数在该比例下的剪枝阈值可由式(6)求得。

$$\theta=a_{i'} \tag{6}$$

其中, $a_{i'}$ 为该层参数按升序排序后的数组 P' 中的第 i 个元素。在求出该层在该剪枝比率下的剪枝阈值之后,即可按照式(7)对该层参数进行剪枝。

$$a_i=\begin{cases} 0, & |a_i|\leqslant \theta \\ a_i, & |a_i|>\theta \end{cases} \tag{7}$$

按照该方法对各层参数进行处理之后,网络中不大于阈值的参数被设置为 0。因为胶囊网络中对应于权值的运算均为乘法和加法,所以这相当于移除掉了对应的连接和权值,即对网络进行了剪枝。

(3)剪枝后网络的微调

剪枝去除了原始网络中较冗余的连接,保留了更为重要的连接。虽然保留下的权值较为重要,但是仍然需要对剪枝之后的网络进行微调^[16],才能恢复网络的识别能力。为了达到只对保留下的连接进行微调而不对剪掉的连接进行改动,需要记录被剪去的连接,因此本文使用掩码来实现这一目的。对网络进行重新训练的流程如图 7 所示。

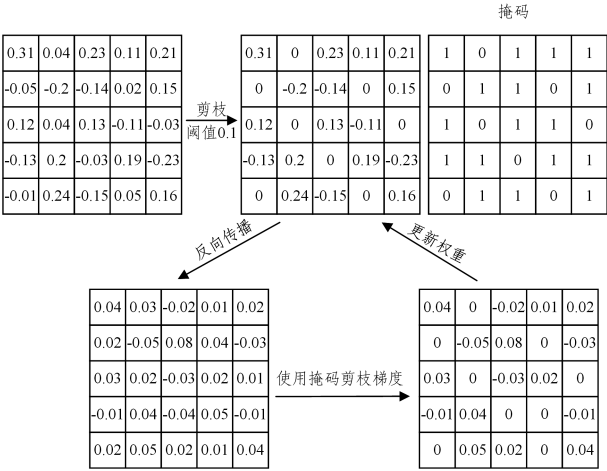


图 7 剪枝及网络微调示意图

Fig. 7 Schematic diagram of pruning and network fine-tuning

图 7 中的掩码用于记录该层被剪去的连接的位置,在反向传播时,先得到所有连接的更新增量,然后使用掩码把被剪去位置的值为 0,其余保持不变,使用掩码处理过的更新增量来更新权重,这样就达到了只对保留下的权值进行微调的目的。

3.5 模型的压缩存储

深度学习框架将神经网络中的权值参数存储到多维数组中^[17],这样可以方便参数的查找并且可以将网络中的计算转换为矩阵运算。对于已经进行了剪枝的模型,其中存在大量权值为 0,如果仍将这些被裁剪掉的权值保存到模型的参数中,那么会极大地浪费存储模型所需要的存储空间。为了压缩剪枝后模型参数所占用的存储空间,可以只存储模型中的非零权值以及它们在数组中的位置,这样原来的多维数组就

可以压缩为 1 个一维的非零权值数组和一个非零权值对应的位置数组。深度卷积神经网络中的权重大多为多维数组,如典型的二维卷积,其参数为一个 4 维数组 (n,n,m,l) ,其中 n 为卷积核大小, m 为卷积核通道数, l 为卷积核个数。为了避免用 4 个整数表示一个非零权重的位置信息,本文使用相对索引以及参数数组的形状来表示非零权重的位置信息。进一步地,神经网络中的权值为 32 位的浮点型数据类型,也就是存储一个模型权值需要 32 位的存储空间,为了压缩网络权值所占用的存储空间,本文使用 8 位整数数据类型以及一个缩放因子来保存模型权值。假设模型中某一层的权值数组为 $P_{float32}$,那么该层权值在 8 位整数数据类型下的缩放因子为:

$$s=\frac{\max(|P_{float32}|)}{127}$$

其中, $|\cdots|$ 表示绝对值操作, $\max$ 表示取最大值,127 是 8 位有符号整数可表示的最大值。该层参数转换为 8 个字节表示的整数:

$$P_{int8}=\text{round}\left(\frac{P_{float32}}{s}\right)$$

其中,round 为四舍五入取整操作。模型压缩存储示意图如图 8 所示。

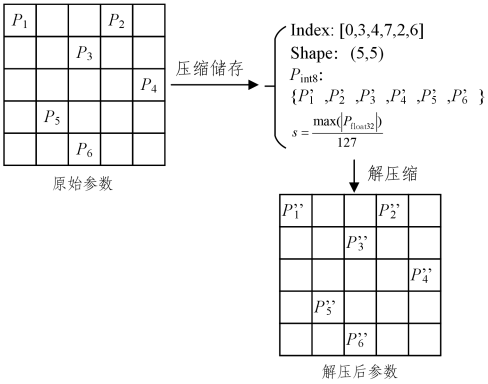


图 8 模型压缩存储示意图

Fig. 8 Schematic diagram of model compression storage

图 8 中原始参数为尺寸为 5×5 的数组,有 19 个参数被剪去,对于保留下来的 6 个权值,将其压缩后保存为 index, shape, Pint8, s。解压缩时,根据保留下来的 4 组数据,在忽略误差的情况下,即可恢复原始参数。

4 实验结果与分析

为验证本文提出的算法,使用 4 个标准数据集来进行实验。实验共分为两组,第一组通过改变网络的压缩比率,使用 SAR 图像变化检测评价指标对不同剪枝比率剪枝后的网络的精确度进行实验;第二组通过对比不使用模型压缩存储和使用模型压缩存储的模型的检测精度,以验证本文模型压缩存储方法的有效性。

4.1 实验环境

实验的硬件平台为 Inter(R) Core(TM) i5-6300HQ CPU@ 2.3GHz, NVIDIA GeForce GTX 950M。软件平台为 Ubuntu 16.04 64 位, MatlabR2019a, Python 3.6.9, Tensorflow 1.13.1, keras2.2.4。

4.2 实验数据

下面介绍本文使用的 4 组 SAR 图像测试数据集。第 1 组数据是我国黄河口地区稻田地的 SAR 图像,图像的尺寸为 289×257 ,如图 9 所示。第 2 组数据到第 4 组数据分别为渥太华,伯尔尼和旧金山数据集^[1],如图 10—图 12 所示。



图 9 黄河稻田地数据集

Fig. 9 Yellow River Estuary dataset

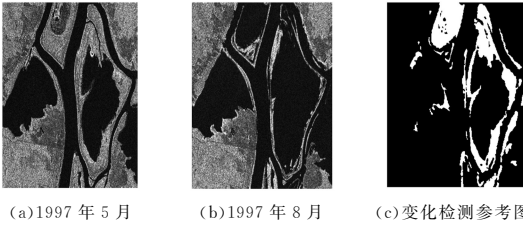


图 10 渥太华数据集

Fig. 10 Ottawa Dataset

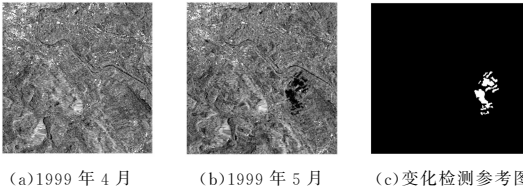


图 11 伯尔尼数据集

Fig. 11 Bern dataset

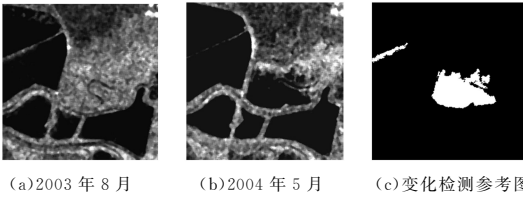


图 12 旧金山数据集

Fig. 12 San Francisco dataset

4.3 实验评价指标

对于 SAR 图像变化检测来说,主要的指标有 PCC 和 Kappa 系数^[18]。

正确检测率(PCC)的公式如下:

$$PCC = \frac{TP + TN}{TP + FP + TN + FN} \tag{8}$$

其中, FN 为漏检数, FP 为错检数, TP 为真正变化数, TN 为真正未变化数。

Kappa 系数(KC)的公式如下

$$KC = \frac{PCC - PRE}{1 - PRE} \tag{9}$$

其中, PRE 的定义如下:

$$PRE = \frac{(TP + FP) \times (TP + FN) + (FN + TN) \times (FP + TN)}{(TP + FP + TN + FN)^2}$$

PCC 是检测正确的像素值的比例,能在一定程度上代表算法的检测性能。但是,在未变化像素点很多的图像中,即便算法不能检测出图像中的任何变化点,根据式(8), TP 为 0,但是 TN 很大,而 TN 与分母几乎相等,这会导致 PCC 接近于 1,很明显此时的 PCC 无法衡量算法对于该图的检测性能。KC 考虑了混淆矩阵中的所有值,在评估检测结果时更全面,更能衡量检测结果图与参考图的一致性。因此,对于 PCC 相同的两个算法, KC 越高,则检测性能越好。

4.4 不同剪枝比率对模型检测精度的影响

由式(5)、式(6)可知,剪枝比率控制着剪枝时的阈值,其值越大,保留下来的参数越少,对网络的剪枝程度越大。图 13 是不同剪枝比率与模型参数数量的对应关系,其中参数数量用 kB(kilobyte)表示。

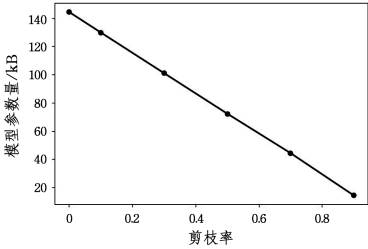


图 13 不同剪枝比率与模型参数数量的关系

Fig. 13 Relationship between different pruning ratios and model parameters

本组实验使用原始网络模型进行剪枝,对比对象为剪枝比率分别为 0.1,0.3,0.5,0.7,0.9 时网络的检测精度。剪枝流程如图 7 所示,剪枝后使用训练网络时使用的训练数据对网络保留下的权值进行微调,微调时观察在验证集上的验证损失,当其不再减小时停止微调。

(1) 黄河稻田地数据集

在黄河稻田地数据集上,剪枝率对模型的影响实验结果如图 14 所示。可以看出,在该数据集上剪枝之后的模型检测性能均有所提升。当 λ 为 0.5 时,模型的检测精度提升最少, PCC 提升约为 0.2%, KC 提升约为 0.9%;当 λ 为 0.9 时,模型的检测精度提升最多, PCC 提升了 0.3%, KC 提升了 1.2%。

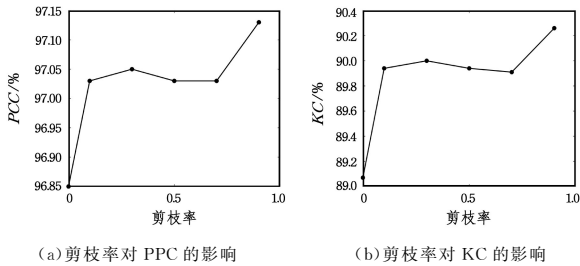


图 14 黄河稻田数据集上剪枝率对模型的影响

Fig. 14 Influence of pruning rate on Yellow River Estuary dataset

(2) 渥太华数据集

由图 15 可以看出,经过剪枝之后的网络在该数据集上的总体检测精度有所下降,当 λ 为 0.3 时,网络检测精度降低最多, PCC 约有 0.35% 的降低, KC 约有 1.2% 的损失;当 λ 从 0.3 增加时,网络的检测精度有所提升;当 λ 为 0.7 时,网络检

测精度损失最小,PCC 约降低了 0.2%,KC 的损失约为 0.8%。

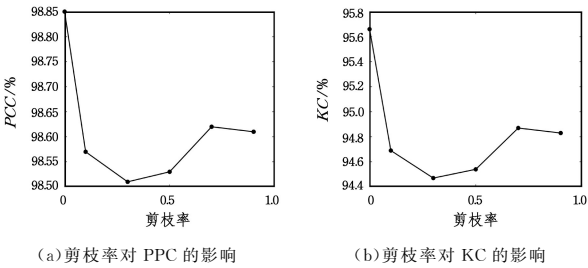


图 15 渥太华数据集上剪枝率对模型的影响
Fig. 15 Influence of pruning rate on Ottawa dataset

(3)伯尔尼数据集

由图 16 可以看出在伯尔尼数据集上,剪枝后模型的检测精度也均有一定程度的损失。

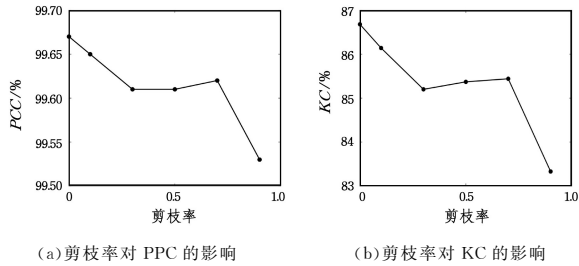


图 16 伯尔尼数据集上剪枝率对模型的影响
Fig. 16 Influence of pruning rate on Bern dataset

当 λ 为 0.1 时检测精度损失最少,PCC 大约损失了 0.02%,KC 约有 0.5%的损失;当 λ 为 0.9 时,模型检测精度损失最多,PCC 大约有 0.1%的降低,KC 的降低约为 3%。

(4)旧金山数据集

图 17 为在旧金山数据集上剪枝比率对算法检测性能的影响,由图中可以看出,经过剪枝的模型的检测精度均有一定程度的损失。当 λ 为 0.3 时,模型检测性能损失最严重,PCC 约降低了 0.1%,KC 约降低了 0.9%;当 λ 为 0.9 时,模型的检测性能损失最少,PCC 约有 0.04%的损失,KC 约有 0.3%的损失。

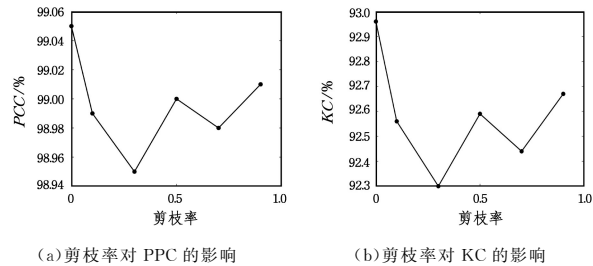


图 17 旧金山数据集上剪枝率对模型的影响
Fig. 17 Influence of pruning rate on San Francisco dataset

4.5 权值剪枝前后模型性能以及模型参数的可视化

第 4.4 节通过调整剪枝比率 λ ,测试了不同剪枝程度对模型检测性能的影响。本节通过模型剪枝前后和微调后检测的结果图,以及剪枝前后和微调后网络中部分结构的权值分布图,对模型性能以及模型参数的变化进行了可视化。

本节以黄河稻田地数据集为例,设置剪枝比率为 0.9,以网络中 conv_2 以及 caps_1 中的参数为对象进行观察。

图 18 为黄河数据集上剪枝前后以及经过微调后模型检测结果图。可以看出,剪枝后模型的检测性能明显下降,但是经过微调后,模型的检测性能得到了很大的修复。图 19 和图 20 分别表示 conv_2 和 caps_1 层参数在原始网络以及微调后网络中的分布,图中横坐标代表权值区间,纵坐标代表权值在某区间上的个数。由图 19 和图 20 可知,原始模型中的参数接近于以 0 为中心的正态分布,剪枝后,模型中的参数量明显减少,主要保留了较大的参数。从通过微调得到的最终网络中可以看出,参数的范围明显比原始网络大。如在层 conv_2 中,原始网络中参数的绝对值的最大值约为 0.2,而在微调后的网络中,参数的绝对值的最大值趋近于 0.4,并且参数的分布也更离散。caps_1 中的参数微调后范围也明显变大。

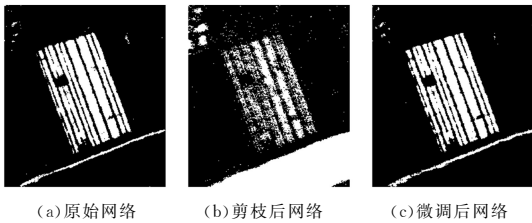


图 18 黄河稻田地数据集上剪枝前后及微调后模型检测结果
Fig. 18 Detection results of Yellow River Estuary dataset before and after pruning and after fine-tuning

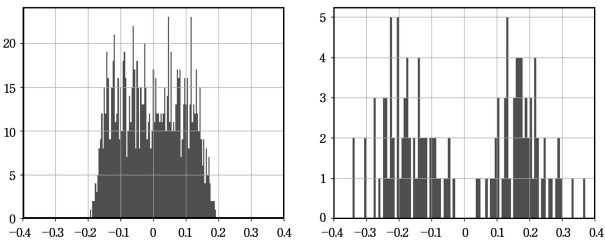


图 19 conv_2 层参数在剪枝前后的分布
Fig. 19 Distribution of conv_2 parameters before and after pruning

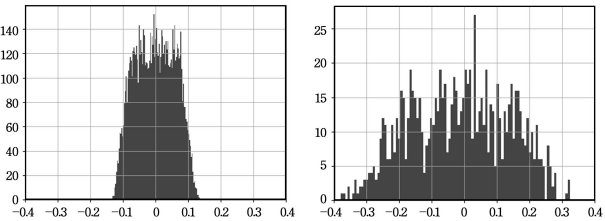


图 20 caps_1 层参数在剪枝前以及微调后的分布
Fig. 20 Distribution of caps_1 parameters before and after pruning

4.6 模型压缩对检测性能的影响

对于剪枝后的模型,在保存模型中的参数时,只对没有被剪去的参数进行保存。在保存参数时,为了压缩模型中的参数,使用 8 位有符号整数以及一个尺度因子对 32 位浮点型模型参数进行近似。本节将对比该模型存储方法对模型检测精度的影响,观察不同数据集在不同剪枝率下,模型压缩前后所占的存储空间大小以及检测精度。

用 PCC,KC 表示没有使用压缩存储时模型的检测精度评价指标,用 PCC',KC' 表示使用压缩存储后模型的检测精度的评价指标, λ 表示剪枝率。在 4 个 SAR 数据集上模型压

缩前后模型检测性能的评价指标实验结果如表 2—表 5 所列。压缩率为进行参数压缩前后模型参数文件大小的比值。

表 2 黄河稻田数据集上模型压缩前后对检测性能的影响
Table 2 Influence of compression on model performance before and after compression on Yellow River Estuary dataset

模型指标	PCC	KC	PCC′	KC′	压缩率
未剪枝	0.9685	0.8907	0.9685	0.8907	1
$\lambda=0.1$	0.9703	0.8994	0.9703	0.8994	2.03
$\lambda=0.3$	0.9705	0.9000	0.9705	0.9001	2.38
$\lambda=0.5$	0.9703	0.8994	0.9703	0.8993	2.72
$\lambda=0.7$	0.9703	0.8991	0.9703	0.8994	3.33
$\lambda=0.9$	0.9713	0.9026	0.9715	0.9035	4.12

表 3 渥太华数据集上模型压缩前后对检测性能的影响
Table 3 Influence of compression on model performance before and after compression on Ottawa dataset

模型指标	PCC	KC	PCC′	KC′	压缩率
未剪枝	0.9885	0.9566	0.9885	0.9566	1
$\lambda=0.1$	0.9857	0.9469	0.9858	0.9474	2.03
$\lambda=0.3$	0.9851	0.9447	0.9853	0.9453	2.37
$\lambda=0.5$	0.9853	0.9454	0.9855	0.9460	2.72
$\lambda=0.7$	0.9862	0.9487	0.9863	0.9490	3.32
$\lambda=0.9$	0.9861	0.9483	0.9862	0.9487	4.12

表 4 伯尔尼数据集上模型压缩前后对检测性能的影响
Table 4 Influence of compression on model performance before and after compression on Bern dataset

模型指标	PCC	KC	PCC′	KC′	压缩率
未剪枝	0.9967	0.8668	0.9967	0.8668	1
$\lambda=0.1$	0.9965	0.8614	0.9965	0.8612	2.03
$\lambda=0.3$	0.9961	0.8520	0.9962	0.8523	2.37
$\lambda=0.5$	0.9961	0.8537	0.9961	0.8510	2.72
$\lambda=0.7$	0.9962	0.8544	0.9962	0.8557	3.33
$\lambda=0.9$	0.9953	0.8333	0.9953	0.8329	4.12

表 5 旧金山数据集上模型压缩前后对检测性能的影响
Table 5 Influence of compression on model performance before and after compression on San Francisco dataset

模型指标	PCC	KC	PCC′	KC′	压缩率
未剪枝	0.9905	0.9296	0.9905	0.9296	1
$\lambda=0.1$	0.9899	0.9256	0.9899	0.9258	2.03
$\lambda=0.3$	0.9895	0.9230	0.9895	0.9226	2.38
$\lambda=0.5$	0.9900	0.9259	0.9899	0.9255	2.72
$\lambda=0.7$	0.9898	0.9244	0.9899	0.9256	3.33
$\lambda=0.9$	0.9901	0.9267	0.9902	0.9269	4.12

根据表 2—表 5,对比使用和未使用模型压缩存储模型参数前后的模型检测精度可以发现,在参数压缩存储前后,模型的检测精度有不同的变化情况。如对于渥太华数据集,当 $\lambda=0.1$ 时,未对模型参数进行压缩存储时,模型的 KC 为 94.69%,对模型参数进行压缩存储之后,模型的 KC 为 94.74%,模型的检测精度有微小提升;在黄河稻田地数据集上,当 $\lambda=0.5$ 时,对模型进行参数压缩存储前后模型的 KC 分别为 89.94%和 89.93%,模型的检测精度有所下降。综合 4 个数据集来看,本文设计的模型压缩存储没有对模型的检测精度造成很大的损失。

结束语 胶囊网络由于其优异的性能已经被应用于各个领域,本文提出了基于胶囊网络的 SAR 图像变化检测算法,取得了良好的检测效果.为降低算法复杂度和存储空间,本文针对胶囊网络设计了参数剪枝以及模型压缩存储算法,在 4 个 SAR 图像检测数据集上验证了算法效果。未来还需要探索更加高效、更加自动化的胶囊网络模型压缩方法。

参 考 文 献

[1] LIU L, JIA Z, YANG J, et al. SAR Image Change Detection Based on Mathematical Morphology and the K-Means Clustering Algorithm[J]. IEEE Access, 2019, 7: 43970-43978.

[2] SABOUR S, FROST N, HINTON G E. Dynamic routing between capsules[C]// Advances in Neural Information Processing Systems. 2017: 3856-3866.

[3] DHILLON A, VERMA G K. Convolutional neural network: a review of models, methodologies and applications to object detection[J]. Progress in Artificial Intelligence, 2020, 9(2): 85-112.

[4] PAOLETTI M E, HAUT J M, FERNANDEZ-BELTRAN R, et al. Capsule Networks for Hyperspectral Image Classification [J]. IEEE Transactions on Geoscience & Remote Sensing, 2019, 57(4): 2145-2160.

[5] CHOI Y, EL-KHAMY M, LEE J. Universal Deep Neural Network Compression[J]. IEEE Journal on Selected Topics in Signal Processing, 2020, 14(4): 715-726.

[6] HOWARD A G, ZHU M, CHEN B, et al. Mobilenets: Efficient convolutional neural networks for mobile vision applications[J]. arXiv:1704.04861, 2017.

[7] ZHANG T Y, YE S K, ZHANG K Q, et al. A Systematic DNN Weight Pruning Framework using Alternating Direction Method of Multipliers[J]. arXiv:1804.03294, 2018.

[8] GUPTA S, AGRAWAL A, GOPALAKRISHNAN K, et al. Deep learning with limited numerical precision[C]// International Conference on Machine Learning. 2015: 1737-1746.

[9] SUN S, CHENG Y, GAN Z, et al. Patient Knowledge Distillation for BERT Model Compression[J]. arXiv:1908.09355, 2019.

[10] HUANG G, LIU Z, WEINBERGER K Q. Densely Connected Convolutional Networks[C]// IEEE Conference on Computer Vision and Pattern Recognition (CVPR). 2017: 2261-2269.

[11] HU J, SHEN L, ALBANIE S, et al. Squeeze-and-Excitation Networks[J]. IEEE Transactions on Pattern Analysis and Machine Intelligence, 2020, 42(8): 2011-2023.

[12] XIONG Y, SU G, YE S, et al. Deeper Capsule Network for Complex Data[C]// 2019 International Joint Conference on Neural Networks (IJCNN). Budapest, Hungary, 2019: 1-8.

[13] KRIZHEVSKY A, SUTSKEVER I, HINTON G E. Imagenet classification with deep convolutional neural networks[C]// Advances in Neural Information Processing Systems. 2012: 1097-1105.

[14] KINGMA D P, BA J. Adam: A method for stochastic optimization[J]. arXiv:1412.6980, 2014.

[15] KRINIDIS S, CHATZIS V. A robust fuzzy local information C-means clustering algorithm[J]. IEEE Transactions on Image Processing, 2010, 19(5):1328-1337.

[16] HE Y, KANG G, DONG X, et al. Soft Filter Pruning for Accelerating Deep Convolutional Neural Networks[J]. arXiv:1808.06866.

[17] PASANDI M M, HAJABDOLLAHI M, KARIMI N, et al. Modeling of Pruning Techniques for Deep Neural Networks Simplification[J]. arXiv:2001.04062.

[18] VU V T, GOMES N R, PETTERSSON M I, et al. Bivariate Gamma Distribution for Wavelength-Resolution SAR Change Detection[J]. IEEE Transactions on Geoscience and Remote

Sensing, 2019, 57(1):473-481.



CHEN Zhi-wen, born in 1995, postgraduate. His main research interests include artificial intelligence computing system and its performance optimization.



ZHU Hu-ming, born in 1978, Ph.D, associate professor. His main research interests include high performance computing and parallel machine learning algorithm.