



(12) 发明专利



(10) 授权公告号 CN 113610227 B

(45) 授权公告日 2023. 11. 21

(21) 申请号 202110838976.6

(22) 申请日 2021.07.23

(65) 同一申请的已公布的文献号
申请公布号 CN 113610227 A

(43) 申请公布日 2021.11.05

(73) 专利权人 人工智能与数字经济广东省实验室(广州)

地址 510330 广东省广州市海珠区新港东路2429号首层自编051房

专利权人 西安电子科技大学广州研究院

(72) 发明人 谢雪梅 石光明 杨建朋 汪振宇

(74) 专利代理机构 陕西电子工业专利中心
61205

专利代理师 王品华

(51) Int.Cl.

G06N 3/082 (2023.01)

G06N 3/086 (2023.01)

G06N 3/0464 (2023.01)

G06N 3/126 (2023.01)

(56) 对比文件

US 2021192352 A1,2021.06.24

CN 111105035 A,2020.05.05

CN 109242142 A,2019.01.18

CN 109683161 A,2019.04.26

CN 110276450 A,2019.09.24

CN 110874631 A,2020.03.10

CN 111368699 A,2020.07.03

高 晗等.深度学习模型压缩与加速综述.《软件学报》.2021,第32卷(第01期),全文.

Ao Ren.et al.ADMM-NN: An Algorithm-Hardware Co-Design Framework of DNNs Using Alternating Direction Method of Multipliers.《Arxiv》.2018,全文.

Zhenyu Wang.et al.Network pruning using sparse learning and genetic algorithm.《Neurocomputing》.2020,第404卷全文.

(续)

审查员 宋亚静

权利要求书3页 说明书8页 附图2页

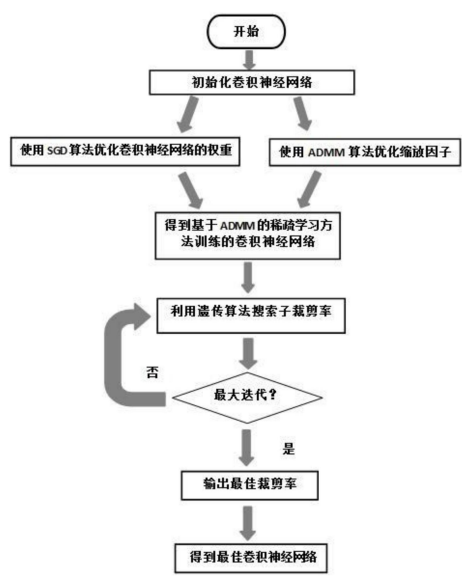
(54) 发明名称

一种用于图像分类的深度卷积神经网络剪枝方法

(57) 摘要

本发明公开了一种高效的深度卷积神经网络剪枝方法,主要解决现有深度卷积神经网络对存储资源和计算资源消耗大的问题,其实现方案是:通过基于ADMM算法的稀疏学习方法优化缩放因子,对深度卷积神经网络进行训练,使网络结构稀疏化;利用遗传算法对训练好的深度卷积神经网络各层所适宜的裁剪率进行搜索,并在适应度函数的引导下自动搜索出符合需求的最佳裁剪率;使用最佳裁剪率对稀疏学习训练后的网络各层进行裁剪,得到效率最佳的卷积神经网络。本发明可大大降低剪枝后卷积神经网络的精度损失,通过减少网络的参数量大大减少了卷积神经网络对存储资源和计算资源消耗,可用于深度

卷积神经网络的压缩。



[转续页]

CN 113610227 B

[接上页]

(56) 对比文件

赖叶静等. 深度神经网络模型压缩方法与进

展.《华东师范大学学报》(自然科学版).2020,
(第05期),全文.

1. 一种用于图像分类的深度卷积神经网络剪枝方法,其特征在于,包括:

(1) 利用基于ADMM的稀疏学习方法训练深度卷积神经网络:

(1a) 对深度卷积神经网络各通道引入缩放因子 $\gamma_{1,i}$;

(1b) 对深度卷积神经网络的损失函数 loss ,添加各通道中缩放因子的0范正则项,得到新的损失函数 loss_{new} ;

(1c) 从公开数据网站下载训练数据集,利用该训练数据集和随机梯度下降算法训练神经网络中缩放因子以外的参数,并利用ADMM算法对缩放因子 $\gamma_{1,i}$ 进行优化,直到(1b)中新的损失函数收敛,得到训练好的深度卷积神经网络模型;

(2) 利用遗传算法搜索深度卷积神经网络模型的最佳子裁剪率:

(2a) 设置遗传算法的最大迭代次数,计算训练后卷积神经网络模型的总参数量 B_0 和计算量 D_0 ;

(2b) 初始化M组裁剪率,每组包括N个不同的裁剪率,N等于网络层数;

(2c) 将各组裁剪率分别编码为二进制编码,并对二进制编码进行交叉和变异操作产生若干新二进制编码,得到新产生的二进制编码数量和原有的二进制编码数量之和为P;

(2d) 对(2c)中P个二进制编码进行解码,并将每条二进制编码解码为一组裁剪率 $p_{i,j}$;

(2e) 根据(2d)中生成的裁剪率,调整网络模型,选择卷积神经网络各层中缩放因子数值较小的通道,使卷积神经网络各层被选择的通道数与该层总通道数之比等于裁剪率;

(2f) 将(2e)中被选择通道从卷积神经网络中删除,获得P个子网络,每个子网络对应一组裁剪率;

(2g) 计算(2f)中的每个子网络的准确率 a_0 ,参数量 b_0 和计算量 d_0 ,并利用 a_0 , b_0 和 d_0 计算出每个子网络的适应度 f_i ;公式如下:

$$f_i = \begin{cases} -a_i \log(\theta \cdot \frac{b_i}{b_0} + (1-\theta) \cdot \frac{d_i}{d_0}), a_i \geq a_0 - \varepsilon; \\ -\frac{a_i}{2} \log(\theta \cdot \frac{b_i}{b_0} + (1-\theta) \cdot \frac{d_i}{d_0}), a_i < a_0 - \varepsilon; \end{cases} \quad <12>$$

其中, a_i , b_i 和 d_i 分别表示每个子网络的准确率,参数量和计算量; θ 是平衡公式各部分的超参数, $-1 \leq \theta \leq 1$, ε 表示精度的允许偏差;

(2h) 通过每个子网络的适应度 f_i 计算出每个网络被选择的概率 q_i ,通过轮盘选择法,从(2f)所获得的P个子网络中筛选出R个子网络,得到每个子网络在(2d)中所对应的当前裁剪率 $p_{i,j}$, $1 \leq R \leq P$;

(2i) 重复(2b)到(2h)过程,当迭代次数达到(2a)中所设置的最大迭代次数,完成搜索,得到最佳适应度所对应的卷积神经网络模型的最佳子裁剪率 p_{best} ;

(3) 选择卷积神经网络各层中缩放因子数值较小的 ω 个通道,使卷积神经网络各层被选择的通道数 ω 与该层总通道数 β 之比 $\frac{\omega}{\beta}$ 等于最佳子裁剪率,以完成对卷积神经网络模型的裁剪,得到卷积神经网络的最佳子网络,用于图像分类。

2. 根据权利要求1所述的方法,其特征在于:(1b)中新的损失函数 loss_{new} 的构造,其实现如下:

向神经网络原来的损失函数 $E(\cdot)$ 添加深度卷积神经网络各通道的缩放因子 $\gamma_{1,i}$ 的 l_0 范数正则项,将约束优化问题转换为如下缩放因子 $\gamma_{1,i}$ 的非约束优化问题:

$$\arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |\Gamma|_0, \lambda > 0 \quad <1>$$

并根据该非约束优化式得到一个新的损失函数 loss_{new} :

$$\text{loss}_{\text{new}} = E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |\Gamma|_0, \quad <2>$$

其中, λ 为超参数, $\lambda > 0$, $E(\cdot)$ 表示原网络的损失函数 loss , Γ 表示缩放因子 $\gamma_{1,i}$ 集合的向量, W_1 和 B_1 分别是网络第1层的权重和偏置集合, $|\Gamma|_0$ 为 l_0 范数, 表示非零因子的数量。

3. 根据权利要求2所述的方法, 其特征在于: (1c) 中利用ADMM算法优化缩放因子的方法, 其实现如下:

(1c1) 为了求解 (1b) 式<1>中的非约束优化问题, 向其加入辅助变量, 并利用增广拉格朗日乘子法进行变换并进行配方, 将公式<1>转换为如下公式:

$$\arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |Z|_0 + \frac{\rho}{2} [(\Gamma - Z + U)^2 - U^2] \quad <3>$$

其中, Γ 表示缩放因子 $\gamma_{1,i}$ 集合的向量, ρ 为常数, U 为用于配方的对偶变量, Z 为辅助变量; $|Z|_0$ 为 l_0 范数, 表示辅助变量的数量;

(1c2) 将 (1c1) 中公式<3>分解为三个子问题, 并通过迭代的方式求解;

(1c3) 将第一个子问题表示为:

$$\arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \frac{\rho}{2} [(\Gamma - Z^{k-1} + U^{k-1})^2 - (U^{k-1})^2] \quad <4>$$

$k = 1, 2, 3, \dots, T,$

其中, k 为迭代次数索引, Z^{k-1} 和 U^{k-1} 分别是第 $k-1$ 次迭代中辅助变量和对偶变量的集合, 可以视为常数;

(1c4) 求解式<4>表示的第一个子问题, 得到第 k 次迭代中缩放因子 $\gamma_{1,i}$ 的集合 Γ^k :

从公开数据网站下载对应任务的训练数据集, 利用该训练数据集和随机梯度下降算法训练该问题, 直到其收敛, 得到第 k 次迭代中缩放因子 $\gamma_{1,i}$ 的集合 Γ^k ;

(1c5) 将第二个子问题表示为:

$$\arg \min_Z \lambda |Z|_0 + \frac{\rho}{2} (\Gamma^k - Z + U^{k-1})^2, k = 1, 2, 3, \dots, T, \quad <5>$$

其中, Γ^k 是第一个子问题在 k 次迭代的结果, U^{k-1} 是第 $k-1$ 次迭代的对偶变量; Z , Γ^k 和 U^{k-1} 均为向量的表示, 可将该公式<5>转换为如下公式:

$$\arg \min_{z_j} \lambda |z_j|_0 + \frac{\rho}{2} (\gamma_j^k - z_j + u_j^{k-1})^2, \quad <6>$$

$j = 1, 2, \dots, C; k = 1, 2, 3, \dots, T,$

其中, z_j , γ_j^k 和 u_j^{k-1} 分别是向量 Z , Γ^k 和 U^{k-1} 的元素, C 表示神经网络总的通道数量, 其值等于向量 Z , Γ^k 和 U^{k-1} 的长度;

(1c6) 根据式<6>中辅助变量 z_j 的取值, 得到第 k 次迭代中辅助变量的集合 Z^k :

若辅助变量 z_j 不为零, 则将公式<6>重写为:

$$\arg \min_{z_j} \lambda + \frac{\rho}{2}(\gamma_j^k - z_j + u_j^{k-1})^2, j=0,1,2,\dots,C; z_j \neq 0 \quad <7>$$

对公式<7>的完全平方项取零,即 $z_j^k = \gamma_j^k + u_j^{k-1}$,得到公式<10>的最小值为 λ , z_j^k 为 z_j 在第k次迭代时的值;

若辅助变量 z_j 为0,即 z_j^k 为0,则公式<6>中的 $|z_j|_0$ 项为0,只剩完全平方项,得到其最小值为 $\frac{\rho}{2}(\gamma_j^k + u_j^{k-1})^2$;

综合以上两种取值情况,得到 z_j^k 的值如下:

$$z_j^k = \begin{cases} \gamma_j^k + u_j^{k-1}, & \frac{\rho}{2}(\gamma_j^k + u_j^{k-1})^2 > \lambda; \\ 0, & \frac{\rho}{2}(\gamma_j^k + u_j^{k-1})^2 \leq \lambda; \end{cases} j=1,2,\dots,C. \quad <8>$$

将式<5>第二个子问题的结果表示为 $Z^k = (z_1^k, z_2^k, \dots, z_C^k)$, Z^k 为第k次迭代中辅助变量集合;

(1c7) 将第三个子问题表示为:

$$\arg \min_U \frac{\rho}{2}(\Gamma^k - Z^k + U^{k-1})^2 - U^2, k=1,2,3,\dots,T, \quad <9>$$

其中, Z^k 为第二个子问题在k次迭代的结果;此子问题为凸二次优化问题,其结果为其极值,表示如下:

$$U^k = \Gamma^k - Z^k + U^{k-1}, \quad <10>$$

其中, U^{k-1} 为第k-1次迭代中对偶变量的集合, Γ^k 和 Z^k 分别为第一个和第二个子问题在k次迭代时的结果;

(1c8) 在第k+1次迭代时,将(1c7)中第三个子问题在第k次迭代中的结果 U^k 输入到(1c3)中的第一个子问题中;

(1c9) 重复(1c3)到(1c8)的过程,直到(1b)中新的损失函数 loss_{new} 收敛,完成使用基于ADMM的稀疏学习方法对缩放因子的优化,得到优化之后的缩放因子集合 Γ 。

4. 根据权利要求1所述的方法,其特征在于:(2b) 裁剪率的初始化方式如下:

初始化M个个体作为遗传算法的种群,每一个个体表示一组通道裁剪率,每组包含N个不同的裁剪率:

$$s_i = \{p_{i,j} | 0 \leq p_{i,j} < 1, j=1,2,\dots,N\}, i=1,2,\dots,M, \quad <11>$$

其中, s_i 是第i个个体, $p_{i,j}$ 代表第 s_i 中第j层的通道裁剪率; M 表示个体的总数, N 表示网络的层数。

一种用于图像分类的深度卷积神经网络剪枝方法

技术领域

[0001] 本发明属于计算机技术领域,主要涉及一种用于图像分类的深度卷积神经网络的剪枝方法,可用于深度卷积神经网络的压缩。

背景技术

[0002] 近年来,神经网络技术在科研和实际应用中取得了很好的效果,但是相较于传统算法,神经网络的计算过程需要消耗大量的存储资源和计算资源,这导致其在部署应用时耗费较多的功耗和成本,限制了神经网络在功耗有限的移动设备上的使用。神经网络剪枝作为一种压缩神经网络的方法,通过去除神经网络中的冗余成分减少神经网络的存储消耗及计算消耗,从而达到降低运行神经网络时功耗的目的。

[0003] 目前,神经网络剪枝方法分为非结构化剪枝和结构化剪枝两大类。

[0004] 非结构化剪枝方法,是通过任意地去除神经网络中的非重要权重达到压缩神经网络的目的,这类方法虽然能够取得很高的压缩率,但由于其所去除权重的位置是无序的,因此破坏了神经网络原有的数据结构,且还需要专门的设备去存储裁剪的权重参数,限制了其在通用设备上的应用。

[0005] 结构化剪枝方法,是通过去除神经网络中的非重要通道或者卷积核达到压缩神经网络的目的,由于去除通道不会破坏神经网络的数据结构,因此这类方法能够很好的运用到现有计算设备上。但是该结构化剪枝方法对于神经网络中冗余成分的定位不精确,对于神经网络的压缩率较低,且对神经网络的性能影响较大。

[0006] 此外,使用非结构化方法和结构化方法进行剪枝后的神经网络模型,大部分需要再次进行训练,耗费大量的时间,效率低。

发明内容

[0007] 本发明的目的在于针对上述现有技术的不足,提出一种用于图像分类的深度卷积神经网络的剪枝方法,以在对剪枝后的模型不再进行训练的情况下,保证良好的压缩率和精度,提高剪枝效率。

[0008] 本发明的技术思路是:通过基于ADMM算法的稀疏学习方法约束调控卷积神经网络的训练,使网络结构稀疏化;通过利用遗传算法对训练好的卷积神经网络各层所适宜的裁剪率进行启发式搜索,并在适应度函数的引导下自动搜索出符合需求的最佳裁剪率;通过最佳裁剪率裁剪网络,得到最佳的卷积神经网络。其实现步骤包括如下:

[0009] (1) 利用基于ADMM的稀疏学习方法训练深度卷积神经网络:

[0010] (1a) 对深度卷积神经网络各通道引入缩放因子 $\gamma_{1,i}$;

[0011] (1b) 对深度卷积神经网络的损失函数loss,添加各通道中缩放因子的0范正则项,得到新的损失函数 $loss_{new}$;

[0012] (1c) 从公开数据网站下载训练数据集,利用该训练数据集和随机梯度下降算法训练神经网络中缩放因子以外的参数,并利用ADMM算法对缩放因子 $\gamma_{1,i}$ 进行优化,直到 (1b)

中新的损失函数收敛,得到训练好的深度卷积神经网络模型;

[0013] (2) 利用遗传算法在稀疏学习训练后的卷积神经网络中搜索最佳子裁剪率;

[0014] (2a) 设置遗传算法的最大迭代次数,计算训练后卷积神经网络模型的总参数量 B_0 和计算量 D_0 ;

[0015] (2b) 初始化M组裁剪率,每组包括N个不同的裁剪率,N等于网络层数;

[0016] (2c) 将各组裁剪率分别编码为二进制编码,并对二进制编码进行交叉和变异操作产生若干新二进制编码,得到新产生的二进制编码数量和原有的二进制编码数量之和为P;

[0017] (2d) 对(2c)中P个二进制编码进行解码,并将每条二进制编码解码为一组裁剪率 $p_{i,j}$;

[0018] (2e) 根据(2d)中生成的裁剪率,调整网络模型,选择卷积神经网络各层中缩放因子数值较小的通道,使卷积神经网络各层被选择的通道数与该层总通道数之比等于裁剪率;

[0019] (2f) 将(2e)中被选择通道从卷积神经网络中删除,获得P个子网络,每个子网络对应一组裁剪率;

[0020] (2g) 计算(2f)中的每个子网络的准确率 a_0 ,参数量 b_0 和计算量 d_0 ,并利用 a_0 , b_0 和 d_0 计算出每个子网络的适应度 f_i ;

[0021] (2h) 通过每个子网络的适应度 f_i 计算出每个网络被选择的概率 q_i ,通过轮盘选择法,从(2f)所获得的P个子网络中筛选出R个子网络,得到每个子网络在(2d)中所对应的当前裁剪率 $p_{i,j}$, $1 \leq R \leq P$;

[0022] (2i) 重复(2b)到(2h)过程,当迭代次数达到(2a)中所设置的最大迭代次数,完成搜索,得到最佳适应度所对应的卷积神经网络模型的最佳子裁剪率 p_{best} ;

[0023] (3) 选择卷积神经网络各层中缩放因子数值较小的 ω 个通道,使卷积神经网络各层被选择的通道数 ω 与该层总通道数 β 之比 $\frac{\omega}{\beta}$ 等于最佳子裁剪率 p_{best} ,以完成对卷积神经网络模型的裁剪,得到最佳的卷积神经网络。

[0024] 本发明具有如下优点:

[0025] 1.精度损失小

[0026] 本发明利用基于ADMM算法的稀疏学习方法训练深度卷积神经网络,抑制了卷积神经网络中冗余参数数值的增大,减小了参数对网络性能影响,并通过遗传算法对裁剪率进行细致的筛选,进一步减小了网络模型剪枝对网络精度造成的影响,使得本发明极大的降低了网络模型剪枝带来的精度损失。

[0027] 2.剪枝效率高

[0028] 本发明利用基于ADMM算法的稀疏学习方法以及基于遗传算法启发式搜索方法,分别在卷积神经网络的训练和剪枝阶段进行参数的粗筛和细筛,极大的减小了剪枝带来的精度损失,使得本发明所获得的剪枝后的网络无需再训练就能够拥有与原网络几乎一致的性能,从而节省了再训练的时间,提高了剪枝的效率。

[0029] 3.剪枝可调节

[0030] 本发明利用遗传算法对适合卷积神经网络的裁剪率进行启发式搜索,通过改变适应度的计算方式改变搜索的方向,进而改变最终搜索结果,使得本发明能根据实际需求调

节剪枝。

[0031] 仿真结果表明,本发明能高效压缩深度卷积神经网络并产生极小的精度损失。在基于CIFAR-10和ImageNet数据集的分类任务中,对卷积神经网络参数量的平均压缩率达到68.0%,对计算量的平均压缩率达到63.4%,对通道数的平均压缩量为63.2%。且精度与压缩前持平甚至超越。

附图说明

[0032] 图1为本发明的实现流程图;

[0033] 图2为本发明的仿真结果图。

具体实施方式

[0034] 下面结合附图对本发明的实施例和效果做进一步的详细描述。

[0035] 参照图1,本实例的实现步骤如下:

[0036] 步骤1,利用基于ADMM的稀疏学习方法训练深度卷积神经网络。

[0037] 所述深度卷积神经网络为包括N层卷积层的现有神经网络,其中第1层的输入表示为 x_1 ,对每层输入 x_1 进行卷积和规范化操作,操作集合表示为 $f(\cdot)$,该深度卷积神经网络的每一个通道的输出表示为:

[0038] $y_{1,i} = f(x_1, w_{1,i}, b_{1,i}), l=1, 2, \dots, N; i=1, 2, \dots, n_1$

[0039] 其中,1是神经网络层数的索引,i是通道的索引, $w_{1,i}$ 和 $b_{1,i}$ 分别是通道的权重和偏置集合, n_1 表示深度卷积神经网络中第1层的总通道数, $f(\cdot)$ 表示深度神经网络中卷积核规范化操作的集合, x_1 表示神经网络中第1层的输入,N表示深度卷积神经网络的总层数,其数值根据不同的任务需求进行调整。

[0040] 本步骤利用基于ADMM的稀疏学习方法对深度卷积神经网络进行训练,其实现如下

[0041] (1.1) 在深度卷积神经网络的各通道引入缩放因子,得到其各通道的输出:

[0042] $y_{1,i} = \gamma_{1,i} \cdot f(x_1, w_{1,i}, b_{1,i}), l=1, 2, \dots, N; i=1, 2, \dots, n_1$

[0043] 其中, $\gamma_{1,i}$ 表示深度卷积神经网络中第1层第i个通道的缩放因子;若 $\gamma_{1,i}$ 为0,则对应通道的输出 $y_{1,i}$ 为0,表示是所对应的通道是无效的,可以被安全地裁剪。

[0044] (1.2) 根据(1.1)中缩放因子 $\gamma_{1,i}$ 的定义,将深度卷积神经网络通道的稀疏性 ξ_c 使用缩放因子的稀疏性 ξ_s 来表示,即将求解通道稀疏性问题转换为对如下缩放因子 $\gamma_{1,i}$ 的约束优化问题:

$$[0045] \quad \arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma), \quad <3>$$

$$s.t. |\Gamma|_0 < C,$$

[0046] 其中, $E(\cdot)$ 表示网络的损失函数loss, Γ 表示缩放因子 $\gamma_{1,i}$ 集合的向量, W_1 和 B_1 分别是第1层的权重和偏置集合,C是通道总数, $|\Gamma|_0$ 为 l_0 范数,表示非零因子的数量。

[0047] (1.3) 向(1.2)中的损失函数 $E(\cdot)$ 添加深度卷积神经网络各通道的缩放因子 $\gamma_{1,i}$ 的 l_0 范数正则项,将约束优化问题转换为如下缩放因子 $\gamma_{1,i}$ 的非约束优化问题:

$$[0048] \quad \arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |\Gamma|_0, \lambda > 0 \quad <4>$$

[0049] 并根据该非约束优化式得到一个新的损失函数 loss_{new} :

$$[0050] \quad \text{loss}_{\text{new}} = E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |\Gamma|_0, \text{其中}\lambda\text{为超参数,}\lambda>0.$$

[0051] (1.4) 为了求解 (1.3) 中的非约束优化问题,向其加入辅助变量,并利用增广拉格朗日乘子法进行变换并进行配方,将公式<4>转换为如下公式:

$$[0052] \quad \arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \lambda |Z|_0 + \frac{\rho}{2} [(\Gamma - Z + U)^2 - U^2] \quad <5>$$

[0053] 其中 ρ 为常数, U 为用于配方的对偶变量, Z 为辅助变量。 $|Z|_0$ 为 l_0 范数,表示辅助变量的数量。

[0054] (1.5) 将 (1.4) 中公式 (5) 再分解为如下三个子问题:

[0055] (1.5.1) 将第一个子问题表示为:

$$[0056] \quad \arg \min_{\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma} E(\{W_l\}_{l=1}^N, \{B_l\}_{l=1}^N, \Gamma) + \frac{\rho}{2} [(\Gamma - Z^{k-1} + U^{k-1})^2 - (U^{k-1})^2] \quad <6>$$

$$k = 1, 2, 3, \dots, T,$$

[0057] 其中, k 为迭代次数索引, Z^{k-1} 和 U^{k-1} 分别是第 $k-1$ 次迭代中辅助变量和对偶变量的集合,可以视为常数。

[0058] (1.5.2) 将第二个子问题表示为:

$$[0059] \quad \arg \min_Z \lambda |Z|_0 + \frac{\rho}{2} (\Gamma^k - Z + U^{k-1})^2, k = 1, 2, 3, \dots, T, \quad <7>$$

[0060] 其中, Γ^k 是第一个子问题在 k 次迭代的结果, U^{k-1} 是第 $k-1$ 次迭代的对偶变量。 Z, Γ^k 和 U^{k-1} 均为向量的表示,可将该公式<7>转换为如下公式:

$$[0061] \quad \arg \min_{z_j} \lambda |z_j|_0 + \frac{\rho}{2} (\gamma_j^k - z_j + u_j^{k-1})^2, \quad <8>$$

$$j = 1, 2, \dots, C; k = 1, 2, 3, \dots, T,$$

[0062] 其中, z_j, γ_j^k 和 u_j^{k-1} 分别是向量 Z, Γ^k 和 U^{k-1} 的元素, C 表示神经网络总的通道数量,其值等于向量 Z, Γ^k 和 U^{k-1} 的长度;

[0063] (1.5.3) 将第三个子问题表示为:

$$[0064] \quad \arg \min_U \frac{\rho}{2} (\Gamma^k - Z^k + U^{k-1})^2 - U^2, k = 1, 2, 3, \dots, T, \quad <9>$$

[0065] 其中, Z^k 为第二个子问题在 k 次迭代的结果。

[0066] (1.6) 通过迭代地求解 (1.5) 中三个子问题来求解 (1.4) 中的公式:

[0067] (1.6.1) 求解 (1.5.1) 中式<6>表示的第一个子问题,得到第 k 次迭代中缩放因子 $\gamma_{1,i}$ 的集合 Γ^k :

[0068] 所述第一个问题由两部分组成,第一部分为卷积神经网络最初的损失函数 loss ;第二部分是凸的和可微的。根据具体任务需求,从公开数据网站下载对应任务的训练数据

集,利用该训练数据集和随机梯度下降算法训练该问题,直到其收敛,得到第k+1次迭代中缩放因子 $\gamma_{1,i}$ 的集合 Γ^k ;

[0069] (1.6.2) 求解(1.5.2)中式<8>表示的第二个子问题,根据式<8>中辅助变量 z_j 的取值,得到第k次迭代中辅助变量的集合 Z^k :

[0070] 若辅助变量 z_j 不为零,则将公式<8>重写为:

$$[0071] \quad \arg \min_{z_j} \quad \lambda + \frac{\rho}{2} (\gamma_j^k - z_j + u_j^{k-1})^2, j = 0, 1, 2, \dots, C; z_j \neq 0 \quad <10>$$

[0072] 对公式<10>的完全平方项取零,即 $z_j^k = \gamma_j^k + u_j^{k-1}$,得到公式<10>的最小值为 λ , z_j^k 为 z_j 在第k次迭代时的值;

[0073] 若辅助变量 z_j 为0,即 z_j^k 为0,则公式<8>中的 $|z_j|_0$ 项为0,只剩完全平方项,得到其最小值为 $\frac{\rho}{2} (\gamma_j^k + u_j^{k-1})^2$;

[0074] 综合以上两种取值情况, z_j^k 的值如下:

$$[0075] \quad z_j^k = \begin{cases} \gamma_j^k + u_j^{k-1}, & \frac{\rho}{2} (\gamma_j^k + u_j^{k-1})^2 > \lambda; \\ 0, & \frac{\rho}{2} (\gamma_j^k + u_j^{k-1})^2 \leq \lambda; \end{cases} j = 1, 2, \dots, C. \quad <11>$$

[0076] 将第二个子问题的结果表示为 $Z^k = (z_1^k, z_2^k, \dots, z_C^k)$, Z^k 为第k次迭代中辅助变量集合。

[0077] (1.6.3) 求解(1.5.3)中式<9>表示的第三个子问题,得到第k次迭代中对偶变量的集合 U^k :

[0078] 由于(1.5.3)中第三个子问题为凸二次优化问题,其结果为其极值,表示如下:

$$[0079] \quad U^k = \Gamma^k - Z^k + U^{k-1} \quad <12>$$

[0080] 其中, U^{k-1} 为第k-1次迭代中对偶变量的集合, Γ^k 和 Z^k 分别为第一个和第二个子问题在k次迭代时的结果;

[0081] (1.7) 在第k+1次迭代时,将(1.6.3)中第三个子问题在第k次迭代中的结果 U^k 输入到(1.6.1)求解的第一个子问题中;

[0082] (1.8) 重复(1.6)到(1.7)的过程,直到(1.3)中新的损失函数 loss_{new} 收敛,完成使用基于ADMM的稀疏学习方法对深度卷积网路的训练,得到训练后卷积神经网络。

[0083] 步骤2,利用遗传算法在稀疏学习训练后的卷积神经网络中搜索最佳子裁剪率。

[0084] (2.1) 设置最大迭代次数,计算训练后神经网络的参数量 B_0 和计算量 D_0 。

$$[0085] \quad B_0 = \sum_{l=1}^N n_l \cdot n_{l-1} \cdot k_w \cdot k_h + 4 \cdot n_l + n_l \cdot (n_{l-1} + 1) \quad <13>$$

$$[0086] \quad D_0 = \sum_{l=1}^N m_w^l \cdot m_h^l \cdot n_l \cdot (n_{l-1} \cdot (2 \cdot k_w \cdot k_h - 1) + 4) + 2 \cdot n_l \cdot n_{l-1} \quad <14>$$

[0087] 其中,N表示神经网络的总层数, n_1 表示神经网络中第1层的总通道数, k_w 和 k_h 分别

表示神经网络通道中二维卷积核的宽度和长度, m_w^l 和 m_h^l 分别表示神经网络第1层的输出特征图的宽度和长度;

[0088] (2.2) 初始化M个个体作为遗传算法的种群, 每一个个体为一组通道裁剪率, 表示为:

[0089] $s_i = \{p_{i,j} | 0 \leq p_{i,j} < 1, j=1, 2, \dots, N\}, i=1, 2, \dots, M, <15>$

[0090] 其中, s_i 是第i组裁剪率, $p_{i,j}$ 代表 s_i 中第j层的通道裁剪率, M表示裁剪率的组数, 每组裁剪率包括N个不同的裁剪率, N等于网络层数;

[0091] (2.3) 为(2.2)中初始化的M组裁剪率生成二进制编码, 其值为0或者1, 表示如下:

[0092] $c_i = \sum_{j=1}^N \lfloor p_{i,j} \times (2^m - 1) \rfloor \times 2^{m(N-j)}, i=1, 2, \dots, M, <16>$

[0093] 其中, c_i 是第i个编码, m是决定二进制码长度的超参数。

[0094] 在式<16>表示的二进制编码的基础上, 通过对其进行交叉和变异操作, 产生若干新的二进制编码, 得到新产生的二进制编码数量和原有的二进制编码数量之和为P。其中, 交叉是指在两个编码之间随机交换一些位元, 变异是指改变编码中的一些位元(0到1或1到0);

[0095] (2.4) 通过对(2.3)中式<16>的逆过程, 将所有P个编码解码为P组裁剪率, 表示如下:

[0096] $p_{i,j} = \lfloor c_j \div 2^{m(N-j)} \rfloor \div (2^m - 1) <17>$

[0097] 其中, $p_{i,j}$ 表示第i组的第j个裁剪率;

[0098] (2.5) 根据(2.4)中生成的裁剪率, 调整网络模型, 选择卷积神经网络各层中缩放因子数值较小的通道, 使卷积神经网络各层被选择的通道数与该层总通道数之比等于裁剪率 $p_{i,j}$;

[0099] (2.6) 将(2.5)中的被选择通道从神经网络中删除, 获得P个子网络, 每个子网络对应一组裁剪率 $p_{i,j}$;

[0100] (2.7) 针对不同的任务, 从公开数据网站下载对应的测试数据集, 并将该测试集输入到(2.6)中的每个子网络中, 根据网络输出结果和标签, 计算其准确率 d_i :

[0101] $d_i = \varphi(x_i, g)$

[0102] 其中, d_i 表示第i个子网络的准确率, x_i 表示第i个子网络的输出结果, g表示测试集样本的标签。

[0103] (2.8) 根据(2.1)中的式<13>和式<14>, 分别计算出每个子网络的参数量 a_i 和计算量 b_i , 并计算第i个子网络的适应度 f_i :

[0104]
$$f_i = \begin{cases} -a_i \log(\theta \cdot \frac{b_i}{B_0} + (1-\theta) \cdot \frac{d_i}{D_0}), a_i \geq A_0 - \varepsilon; \\ -\frac{a_i}{2} \log(\theta \cdot \frac{b_i}{B_0} + (1-\theta) \cdot \frac{d_i}{D_0}), a_i < A_0 - \varepsilon; \end{cases} <18>$$

[0105] 其中, A_0 和 B_0 和 D_0 分别是原网络的准确率, 参数量和计算量; θ 是平衡式<18>各部分的超参数, $-1 \leq \theta \leq 1$, ε 表示精度的允许偏差。

[0106] (2.9) 将(2.8)所获得的适应度 f_i 求和得到一个总的适应度 $\text{sum}(f)$, 然后为每个子

网络计算一个加权适应度 q_i :

$$[0107] \quad q_i = \frac{f_i}{\text{sum}(f)} \quad <19>$$

[0108] 其中, $\text{sum}(f)$ 是所有子网络适应度 f_i 的总和, q_i 是每个子网络的加权适应度, 表示每个子网络被选择的概率。

[0109] (2.10) 根据(2.9)中每个子网络被选择的概率 q_i , 通过轮盘选择方法选择 R 个子网络, 得到每个子网络在(2.6)所对应的裁剪率 $p_{i,j}$, 并将其输入到(2.2)中进行下一次迭代, 其中 $1 \leq R \leq P$ 。

[0110] (2.11) 重复(2.2)到(2.10), 直到达到最大迭代次数后完成搜索, 得到卷积神经网络模型的最佳子裁剪率 p_{best} 。

[0111] 步骤三, 利用最佳子裁剪率对稀疏学习训练后的卷积神经网络进行裁剪。

[0112] 选择卷积神经网络各层中缩放因子数值较小的 ω 个通道, 使卷积神经网络各层被选择的通道数 ω 与该层总通道数 β 之比 $\frac{\omega}{\beta}$ 等于最佳子裁剪率 p_{best} , 删除选出的 ω 个通道, 以完成对卷积神经网络模型的裁剪, 得到最佳的卷积神经网络, 以为后续的深度学习任务和实际场景应用提供计算耗时更短、占用存储资源更少的卷积神经网络模型。

[0113] 本发明的效果通过以下仿真进一步说明。

[0114] 1. 仿真实验条件:

[0115] 本发明仿真实验的操作系统为Ubuntu18.04 LTS, 深度学习软件平台为Pytorch, GPU采用英伟达TITAN Xp;

[0116] 本发明仿真实验所使用的数据集包括: CIFAR-10, ImageNet;

[0117] 本发明仿真实验所涉及的神经网络包括: Vgg-16, ResNet-50, ResNet-56, ResNet-110, GoogLeNet, DenseNet-40。

[0118] 2. 仿真实验内容:

[0119] 仿真1, 对卷积神经网络训练的仿真实验:

[0120] 使用CIFAR-10数据集, 分别使用传统训练方式和本发明中基于ADMM的稀疏学习的训练方式训练分类神经网络Vgg-16, 结果如图2所示。其中, 图2(a)为使用传统训练方法得到的神经网络特征图; 图2(b)为本发明中基于ADMM的稀疏学习训练方式得到的神经网络特征图。图2中, 黑色小格表示无效的特征, 白色纹理小格表示有效的特征。

[0121] 从图2中可以看出, 本发明基于ADMM的稀疏学习训练方式得到的无效特征图明显多于传统训练方式得到的神经网络特征图, 表明本发明对卷积神经网络的稀疏学习效果明显。

[0122] 仿真2, 卷积神经网络的剪枝仿真实验:

[0123] 利用本发明在CIFAR-10数据集的训练集上分别对卷积神经网络Vgg-16, ResNet-56, ResNet-110, GoogLeNet, DenseNet-40进行训练并完成剪枝;

[0124] 利用本发明在ImageNet数据集的训练集上别对卷积神经网络ResNet-50进行训练并完成剪枝。

[0125] 使用剪枝后的卷积神经网络Vgg-16, ResNet-56 ResNet-110, GoogLeNet, DenseNet-40对来自CIFAR-10数据集的测试集的数据进行分类; 使用剪枝后的卷积神经网络

络ResNet-50,对来自ImageNet数据集的测试集的数据进行分类,得到剪枝后的卷积神经网络对来自两种测试集中的数据进行分类的效果,如表1所示。

[0126] 表1.剪枝后的卷积神经网络对测试集的分类效果

数据集	网络类型	模型	准确率	参数量	参数量 压缩率	计算量	计算量 压缩率	通道 压缩率
[0127] CIFAR-10	Vgg-16	原网络 剪枝后网络	94.09% 94.14%	14.73M 1.79M	87.8%	0.314B 0.138B	56.0%	65.6%
	ResNet-56	原网络 剪枝后网络	93.42% 93.41%	0.86M 0.49M	43.0%	0.127B 0.062B	51.2%	41.4%
	ResNet-110	原网络 剪枝后网络	93.72% 93.75%	1.73M 0.60M	65.6%	0.255B 0.077B	69.8%	62.5%
	GoogleNet	原网络 剪枝后网络	94.81% 94.84%	6.17M 0.72M	88.4%	1.542B 0.253B	83.6%	68.1%
	DenseNet-40	原网络 剪枝后网络	94.22% 94.20%	1.07M 0.25M	77.0%	0.288B 0.079B	72.4%	81.6%
ImageNet	ResNet-50	原网络 剪枝后网络	73.31% 73.29%	25.58M 13.87M	45.8%	4.11B 2.14B	47.9%	60.2%

[0128] 表中参数量的计数单位‘M’表示 10^6 个浮点数,计算量的计数单位‘B’表示 10^9 次浮点运算。原网络表示未经过剪枝的卷积神经网络,剪枝后网络表示经过本发明的方法剪枝后的卷积神经网络。

[0129] 从表1中可以看到,本发明对卷积神经网络参数量的最高压缩率为87.8%,最低压缩率为43.0%,平均压缩率为68.0%;对卷积神经网络计算量的最高压缩率为83.6%,最低压缩率为47.9%,平均压缩率为63.4%;对卷积神经网络通道的最高压缩率为81.6%,最低压缩率为41.4%,平均压缩率为63.2%。

[0130] 表1结果表明,本发明可以有效压缩卷积神经网络,且剪枝后所获得的卷积神经网络错误率与原网络基本一致,表明本发明可以有效压缩卷积神经网络,且产生的精度损失很小。通过本发明对卷积神经网络的压缩大大减小了卷积神经网络的存储消耗及计算消耗,从而可提升后续各种使用效果。

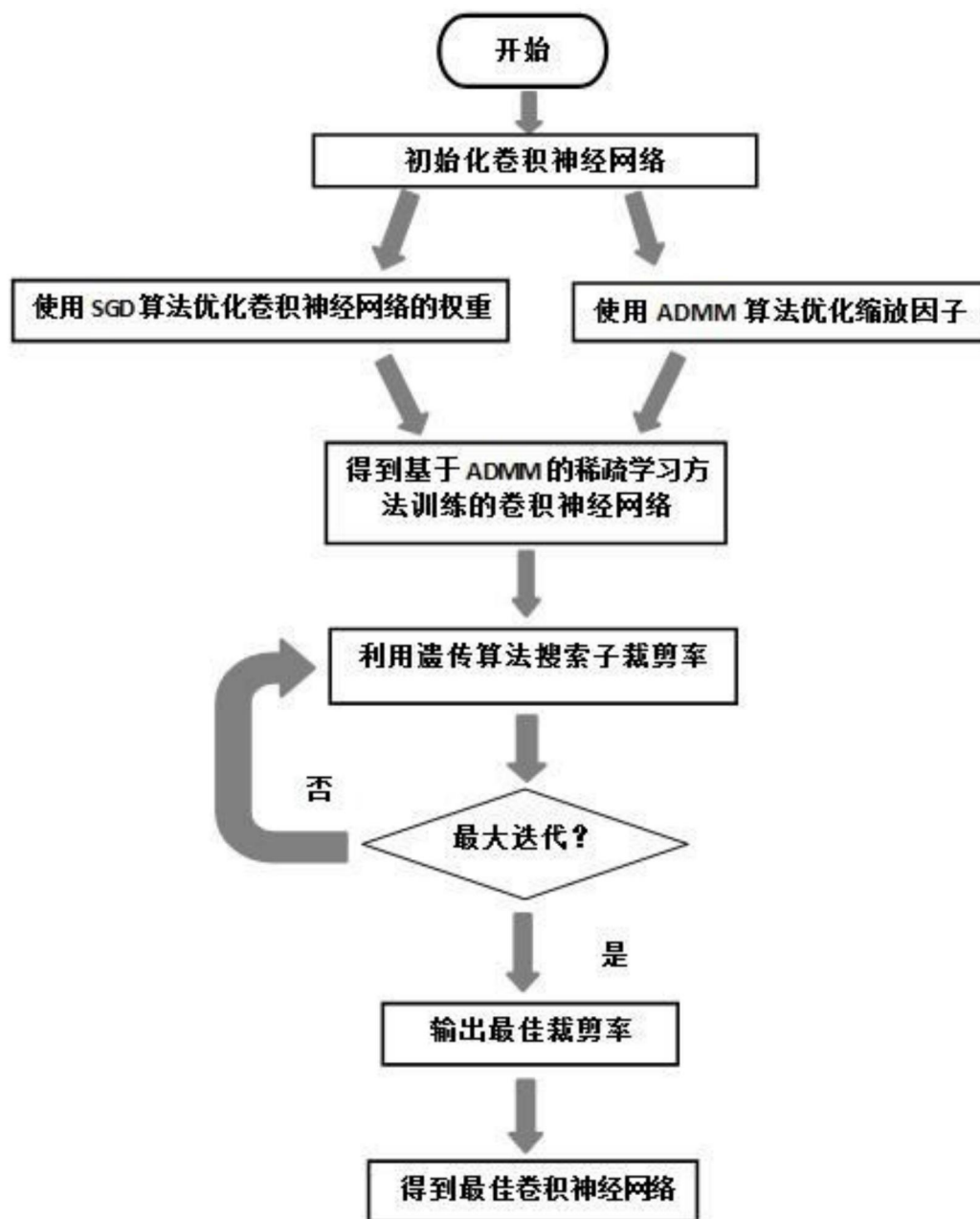
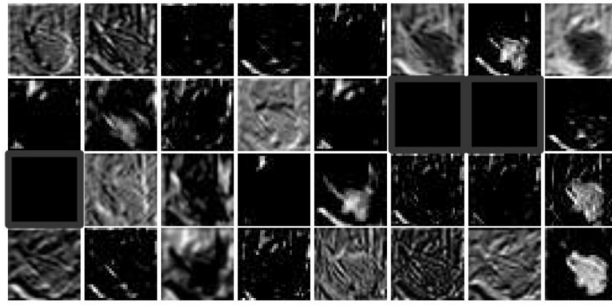
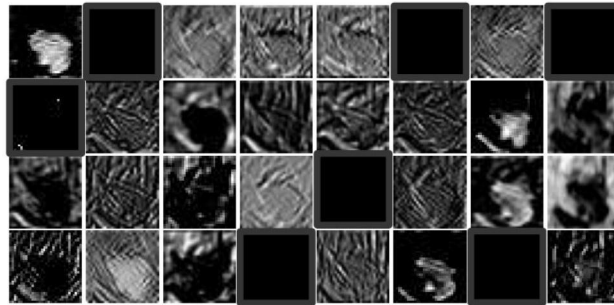


图1



(a)



(b)

图2