



(21) 申请号 202110610837.8

(22) 申请日 2021.05.28

(65) 同一申请的已公布的文献号

申请公布号 CN 113313175 A

(43) 申请公布日 2021.08.27

(73) 专利权人 北京大学

地址 100871 北京市海淀区颐和园路5号

(72) 发明人 林宙辰 徐鑫

(74) 专利代理机构 北京万象新悦知识产权代理

有限公司 11360

专利代理师 黄凤茹

(51) Int.Cl.

G06V 10/764 (2022.01)

G06V 10/82 (2022.01)

G06N 3/0464 (2023.01)

G06N 3/048 (2023.01)

G06N 3/084 (2023.01)

(56) 对比文件

CN 107480777 A, 2017.12.15

CN 108805167 A, 2018.11.13

CN 111062474 A, 2020.04.24

CN 111652271 A, 2020.09.11

CN 111967528 A, 2020.11.20

CN 112819252 A, 2021.05.18

US 2019340510 A1, 2019.11.07

US 2019392323 A1, 2019.12.26

US 2020184310 A1, 2020.06.11

Shiping Wang. Learning Deep Sparse Regularizers With Applications to Multi-View Clustering and Semi-Supervised Classification.《IEEE》.2021,全文.

邹维宝;于昕玉;麦超.基于Lorentz函数的稀疏约束RBM模型的算法研究.计算机工程与应用.2017,(第07期),全文.

审查员 王明芳

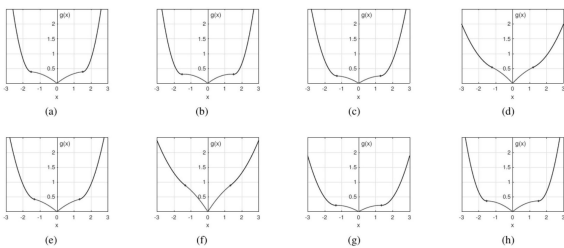
权利要求书2页 说明书8页 附图1页

(54) 发明名称

基于多元激活函数的稀疏正则化神经网络的图像分类方法

(57) 摘要

本发明公布了一种基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,通过学习多元激活函数进一步学习多元稀疏正则化神经网络模型,即稀疏正则化子或稀疏正则化器,用于高效地进行图像分类。采用本发明的多元激活函数将任意一个已有CNN模型中的激活函数进行正则化,得到一个多元正则化的CNN模型,再用该模型进行图像分类,由此达到用更少的模型参数,且能降低图像分类的错误率。



1. 一种基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,通过学习多元激活函数进一步学习多元稀疏正则化神经网络模型,即稀疏正则化子或稀疏正则化器,用于高效地进行图像分类;包括以下步骤:

1) 将图像数据 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 分为训练样本和测试样本;

2) 对训练样本图像进行预处理,包括同态滤波和图像平滑处理;

3) 构建基于多元激活函数的稀疏正则化神经网络模型MAF-SRL并进行训练,具体包括如下步骤:

3-1) 设计多元激活函数,初始化多元激活函数的学习参数: $\hat{\mathbf{A}}_{(0)}, \hat{\mathbf{q}}_{(0)}, \mathbf{b}_{(0)}, \mathcal{U}_{(0)}$,其中 $\hat{\mathbf{A}}_{(0)}$ 表示初始化的稀疏矩阵, $\hat{\mathbf{q}}_{(0)}$ 表示初始化的权重参数向量, $\mathbf{b}_{(0)}$ 表示初始化的偏置参数向量, $\mathcal{U}_{(0)}$ 表示初始化的投影的参数集合;

设置计数器 $l=0$;定义多元激活函数 $\xi(\mathbf{x})$ 为:

$$\xi(\mathbf{x}) = \mathbf{A}^{-T} [\hat{\xi}(\mathbf{A}^T \mathbf{x} + \mathbf{b}) - \mathbf{b}]: \mathbb{R}^n \rightarrow \mathbb{R}^n$$

其中, $\mathbf{A} = (a_1, \dots, a_M)$, $\mathbf{q} = (q_1, \dots, q_M)^T$, $\mathbf{b} = (b_1, \dots, b_M)^T$; $\hat{\xi}(x)$ 为单元激活函数;

3-2) 定义多元正则化器;

定义多元正则化器 $\mathcal{G}(\mathbf{x})$ 为: $\mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{a_i^T \mathbf{x} + b_i} \hat{\xi}^{-1}(y) dy - \frac{1}{2} (a_i^T \mathbf{x} + b_i)^2 \right)$;其中, a_i , q_i , b_i 分别是 \mathbf{A} , \mathbf{q} , \mathbf{b} 中的元素;通过构建一个近端算子: $\mathbf{y} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + \mathcal{G}(\mathbf{y})$ 作为单元激活函数 $\xi(\mathbf{x})$,将学习多元正则化器转换为学习多元激活函数;

3-3) 定义单元激活函数;

定义多元激活函数 $\xi(\mathbf{x})$ 包括的单元激活函数 $\hat{\xi}_{(\mu_1, \mu_2)}(x)$:

$$\hat{\xi}_{(\mu_1, \mu_2)}(x) = \begin{cases} \eta_2(x - \delta_2) + \eta_1(\delta_2 - \delta_1), & \delta_2 \leq x, \\ \eta_1(x - \delta_1), & \delta_1 \leq x < \delta_2, \\ 0, & -\delta_1 \leq x < \delta_1, \\ \eta_1(x + \delta_1), & -\delta_2 \leq x < -\delta_1, \\ \eta_2(x + \delta_2) + \eta_1(\delta_1 - \delta_2), & x < -\delta_2, \end{cases}$$

$\hat{\xi}_{(\mu_1, \mu_2)}(x)$ 包含两个可学习的参数 (μ_1, μ_2) ,其中 $\mu_1 = (\eta_1, \delta_1)$, $\mu_2 = (\eta_2, \delta_2)$, $0 \leq \delta_1 \leq \delta_2$, $\eta_1, \eta_2 > 0$;

3-4) 对输入图像 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 进行训练过程的初始化;

定义初始化的迭代求解 \mathbf{x} 的公式为:

$$\mathbf{x}_{(0)} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T \mathbf{x} - (\mathbf{A} \operatorname{diag}(\mathbf{q}))^{-1} \nabla \phi(\mathbf{x})) + \mathbf{b}) - \mathbf{b}]$$

$\mathbf{x}_{(0)}$ 表示输入图像第0次迭代的向量;

3-5) 迭代求解;

首先通过近端梯度的迭代算法进行求解:

$$\mathbf{x}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}_k + \mathbf{Q}^{-1} \nabla f(\mathbf{x}_k)\|_{\mathbf{Q}}^2 + \mathcal{G}(\mathbf{y})$$

接下来令 $\mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \frac{1}{L} \nabla \phi(\mathbf{x}^{(k)})$,则:

$$\text{Prox}_{\alpha\mathcal{G}}(\mathbf{r}^{(k)}) = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{r}^{(k)}\|_F^2 + \alpha\mathcal{G}(\mathbf{x})$$

得到迭代公式:

$$\mathbf{x}^{(k+1)} = \xi\left(\mathbf{x}^{(k)} - \mathbf{Q}^{-1}\nabla\phi(\mathbf{x}^{(k)})\right)$$

由于 $\mathbf{x}^{(k+1)}$ 是步骤3-1)通过近端梯度算法将学习多元正则化子即多元正则化器转化为学习多元激活函数 $\xi(\mathbf{x})$,故得到:

$$\mathbf{x}_{k+1} = \mathbf{A}^{-T} \left[\hat{\xi} \left(\mathbf{A}^T (\mathbf{x}_k - \mathbf{Q}^{-1}\nabla f(\mathbf{x}_k)) + \mathbf{b} \right) - \mathbf{b} \right]$$

进一步化简得到最后迭代求解的表达式:

$$\mathbf{x}_{k+1} = \mathbf{A}^{-T} \left[\left(\hat{\xi} \left(\mathbf{A}^T \mathbf{x}_k - (\mathbf{A} \operatorname{diag}(\mathbf{q}))^{-1} \nabla f(\mathbf{x}_k) \right) + \mathbf{b} \right) - \mathbf{b} \right];$$

3-6) 进行神经网络的学习训练

对神经网络每一层 $i, i=1, 2, 3, \dots, N$,逐层学习激活函数进而迭代求解 \mathbf{x} :

$$\mathbf{x}_{(i)} = \mathbf{A}_{(i)}^{-T} \left[\left(\hat{\xi} \left(\mathbf{A}_{(i)}^T \mathbf{x}_{(i-1)} - (\mathbf{A}_{(i)} \operatorname{diag}(\mathbf{q}_{(i)}))^{-1} \nabla \phi(\mathbf{x}_{(i-1)}) \right) + \mathbf{b}_{(i)} \right) - \mathbf{b}_{(i)} \right]$$

3-7) 计算神经网络每一层的损失函数: N 代表第 N 层的 \mathbf{x} 图像向量;

$$\mathcal{L}(\hat{\mathbf{A}}, \hat{\mathbf{q}}, \mathbf{b}, \mathbf{u}) = \frac{1}{2} \|\mathbf{x}_{(N)} - \mathbf{x}\|_F^2$$

直到网络收敛为止;

将网络最后一层的输出,作为经迭代优化的最优图像输出 \mathbf{x}^* ;卷积神经网络全连接层参数初始化方式采用Xavier初始化方式;经过全连接层之后输出最优图像的分类标签;

4) 利用构建并训练好的基于多元激活函数的稀疏正则化神经网络模型,对测试图像样本进行识别,得到预测的图像分类标签,由此实现图片分类识别,其中测试样本无需经过图像预处理。

2.如权利要求1所述基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,卷积神经网络全连接层参数的初始化具体采用Xavier初始化方式。

3.如权利要求2所述基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,权重衰减因子为0.001,动量为0.85。

4.如权利要求1所述基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,具体在深度学习框架Tensorflow下实现构建所述基于多元激活函数的稀疏正则化神经网络模型MAF-SRL;将构建并初始化的多元激活函数代入卷积神经网络模型,即得到基于多元激活函数的稀疏正则化神经网络模型。

5.如权利要求1所述基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,具体采用反向传播算法,用带Nesterov动量的随机梯度下降算法进行模型训练。

6.如权利要求5所述基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,其特征是,具体训练过程迭代1000轮;图像批大小为256,即每256张图片为一个批次;学习速率初始值为0.1。

基于多元激活函数的稀疏正则化神经网络的图像分类方法

技术领域

[0001] 本发明属于模式识别、机器学习、人工智能技术领域,涉及图像分类方法,具体涉及一种基于多元激活函数的稀疏正则化神经网络的图像分类方法。

背景技术

[0002] 稀疏正则化是解决深度神经网络中图像分类问题的一种功能强大、应用广泛的策略。稀疏正则化的有效性在实践和理论上都得到了验证,如LASSO算子。稀疏正则化同时进行参数估计和特征选择,通过识别信息变量提供可解释的结果,通过舍弃冗余变量有效避免过拟合。由于这些特性,稀疏正则化在图像分类领域取得了巨大的成功。此外,已经开展了一些理论研究来支持稀疏正则化的有效性,并提出了高效的优化方法,使稀疏学习易于应用。

[0003] 许多稀疏正则化被提出来产生稀疏解。L0范数是稀疏正则化最直观的形式,用于计算非零元素的个数,其目的是促进最稀疏解的实现。然而,L0范数的最小化问题通常是NP-hard的。L1范数是最常用的范数,它是凸的,相应的L1范数正则化问题可以很容易地解决。虽然L1范数具有很好的性质,但可能不会得到足够稀疏的解,并可能导致严重的估计偏差。目前已有其他的稀疏正则化器包括smoothly clipped absolute deviation (SCAD), log penalty, capped L1, minimax concave penalty (MCP), Lp penalty等被提出以克服这些缺陷。但是,上述正则化器大多是可分离的,它们可能不能充分利用向量项之间的交互来产生最佳性能。

[0004] 另一方面,现有的稀疏正则化器都是手工制作的。它们要么不能与控制模型很好地合作以促进稀疏解决方案,要么不能很好地适合数据以实现良好的图像分类性能。因此,现有技术一般采用尝试所有现有的稀疏正则化器,并选出最有效的一个的做法。但是,这种做法效率低下,难以实际应用。

发明内容

[0005] 为了克服上述现有技术的不足,本发明提供一种基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,采用基于多元激活函数的不可分离稀疏正则化器进行学习训练,用训练好的模型进行图像分类,降低图像分类的错误率。

[0006] 本发明通过设计多元激活函数,将任意一个已有CNN模型中的激活函数采用该多元激活函数进行正则化,得到一个多元正则化的CNN模型,即基于多元激活函数的稀疏正则化神经网络模型(MAF-SRL);利用近端梯度算法,将不可分离稀疏正则化学习转化为多元激活函数的学习,得到训练好的基于多元激活函数的稀疏正则化神经网络模型;再用该模型进行图像分类。

[0007] 本发明提供的技术方案是:

[0008] 一种基于多元激活函数的稀疏正则化神经网络模型的图像分类方法,通过学习多元激活函数进一步学习多元稀疏正则化神经网络模型(也称稀疏正则化子或稀疏正则化

器),用于高效的进行图像分类,包括以下步骤:

[0009] 1) 将图像数据 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 分为训练样本和测试样本;

[0010] 2) 对训练样本图像进行预处理,包括同态滤波和图像平滑处理;

[0011] 3) 构建基于多元激活函数的稀疏正则化神经网络模型并进行训练,具体包括如下步骤:

[0012] 3-1) 设计多元激活函数。

[0013] 初始化多元激活函数的学习参数: $\hat{\mathbf{A}}_{(0)}, \hat{\mathbf{q}}_{(0)}, \mathbf{b}_{(0)}, \mathbf{u}_{(0)}$,其中 $\hat{\mathbf{A}}_{(0)}$ 表示初始化的稀疏矩阵, $\hat{\mathbf{q}}_{(0)}$ 表示初始化的权重参数向量, $\mathbf{b}_{(0)}$ 表示初始化的偏置参数向量, $\mathbf{u}_{(0)}$ 表示初始化的投影的参数集合。

[0014] 设置计数器 $l=0$ 。定义多元激活函数 $\xi(\mathbf{x})$ 为:

[0015] $\xi(\mathbf{x}) = \mathbf{A}^{-T} [\xi(\mathbf{A}^T \mathbf{x} + \mathbf{b}) - \mathbf{b}]: \mathbb{R}^n \rightarrow \mathbb{R}^n$

[0016] 其中, $\mathbf{A} = (\mathbf{a}_1, \dots, \mathbf{a}_M)$, $\mathbf{q} = (q_1, \dots, q_M)^T$ and $\mathbf{b} = (b_1, \dots, b_M)^T$; $\xi(x)$ 为单元的激活函数;

[0017] 3-2) 定义多元正则化器。

[0018] 本发明定义多元正则化器 $\mathcal{G}(\mathbf{x})$ 为:

[0019]
$$\mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \xi^{-1}(y) dy - \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} + b_i)^2 \right),$$

[0020] 其中, \mathbf{a}_i, q_i, b_i 分别是 $\mathbf{A}, \mathbf{q}, \mathbf{b}$ 中的元素;

[0021] 通过构建一个近端算子: $\mathbf{y} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + \mathcal{G}(\mathbf{y})$ 作为单元激活函数 $\xi(\mathbf{x})$,将学习多元正则化器转换为学习多元激活函数。

[0022] 3-3) 定义单元激活函数。

[0023] 接下来定义多元激活函数 $\xi(\mathbf{x})$ 包括的单元激活函数 $\hat{\xi}_{(\mu_1, \mu_2)}(x)$,它包含两个可学习的参数 (μ_1, μ_2) ,其中 $\mu_1 = (\eta_1, \delta_1)$, $\mu_2 = (\eta_2, \delta_2)$, $0 \leq \delta_1 \leq \delta_2$, $\eta_1, \eta_2 > 0$ 。 $\eta_1, \delta_1, \eta_2, \delta_2$ 都是训练过程中学习到的参数。 η_1, η_2 分别为第一个和第二个单变量权重参数, δ_1 和 δ_2 分别为第一个和第二个单变量偏置参数。 $\hat{\xi}_{(\mu_1, \mu_2)}(x)$ 是 $\mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \xi^{-1}(y) dy - \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} + b_i)^2 \right)$ 中的 $\xi^{-1}(y)$ 。单元激活函数的形式是

[0024]
$$\hat{\xi}_{(\mu_1, \mu_2)}(x) = \begin{cases} \eta_2(x - \delta_2) + \eta_1(\delta_2 - \delta_1), & \delta_2 \leq x, \\ \eta_1(x - \delta_1), & \delta_1 \leq x < \delta_2, \\ 0, & -\delta_1 \leq x < \delta_1, \\ \eta_1(x + \delta_1), & -\delta_2 \leq x < -\delta_1, \\ \eta_2(x + \delta_2) + \eta_1(\delta_1 - \delta_2), & x < -\delta_2, \end{cases}$$

[0025] 3-4) 对输入图像 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 进行训练过程的初始化。

[0026] 定义初始化的迭代求解 \mathbf{x} 的公式为

[0027] $\mathbf{x}_{(0)} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T \mathbf{x} - (\mathbf{A}_{\text{diag}}(\mathbf{q}))^{-1} \nabla \phi(\mathbf{x})) + \mathbf{b}) - \mathbf{b}]$ 。

[0028] \mathbf{x} 表示输入图像向量; $\mathbf{x}_{(0)}$ 表示输入图像第0次迭代的向量。

[0029] 3-5) 迭代求解

[0030] 本发明首先通过近端梯度的迭代算法进行求解,

$$[0031] \quad \mathbf{x}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}_k + \mathbf{Q}^{-1} \nabla f(\mathbf{x}_k)\|_{\mathbf{Q}}^2 + \mathcal{G}(\mathbf{y})$$

[0032] 接下来令 $\mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \frac{1}{L} \nabla \phi(\mathbf{x}^{(k)})$, 则 $\operatorname{Prox}_{\alpha \mathcal{G}}(\mathbf{r}^{(k)}) = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{r}^{(k)}\|_F^2 + \alpha \mathcal{G}(\mathbf{x})$,

[0033] 得到迭代公式:

$$[0034] \quad \mathbf{x}^{(k+1)} = \xi(\mathbf{x}^{(k)} - \mathbf{Q}^{-1} \nabla \phi(\mathbf{x}^{(k)}))$$

[0035] 由于 $\mathbf{x}^{(k+1)}$ 是步骤3-1) 通过近端梯度算法将学习多元正则化子 (多元正则化器) 转化为学习多元激活函数 $\xi(\mathbf{x})$, 所以得到:

$$[0036] \quad \mathbf{x}_{k+1} = \mathbf{A}^{-T} [\hat{\xi}(\mathbf{A}^T (\mathbf{x}_k - \mathbf{Q}^{-1} \nabla f(\mathbf{x}_k)) + \mathbf{b}) - \mathbf{b}]$$

[0037] 进一步化简得到最后迭代求解的表达式:

$$[0038] \quad \mathbf{x}_{k+1} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T \mathbf{x}_k - (\mathbf{A}_{diag}(\mathbf{q}))^{-1} \nabla f(\mathbf{x}_k)) + \mathbf{b}) - \mathbf{b}]$$

[0039] 3-6) 进行神经网络的学习训练

[0040] 对神经网络每一层 i ($i=1, 2, 3, \dots, N$), 逐层学习激活函数进而迭代求解 \mathbf{x} :

$$[0041] \quad \mathbf{x}_{(i)} = \mathbf{A}_{(i)}^{-T} [(\hat{\xi}(\mathbf{A}_{(i)}^T \mathbf{x}_{(i-1)} - (\mathbf{A}_{(i)diag}(\mathbf{q}_{(i)}))^{-1} \nabla \phi(\mathbf{x}_{(i-1)})) + \mathbf{b}_{(i)}) - \mathbf{b}_{(i)}]$$

[0042] 3-7) 计算神经网络每一层的损失函数: N 代表第 N 层的 \mathbf{x} 图像向量。

$$[0043] \quad \mathcal{L}(\hat{\mathbf{A}}, \hat{\mathbf{q}}, \mathbf{b}, \mathbf{u}) = \frac{1}{2} \|\mathbf{x}_{(N)} - \mathbf{x}\|_F^2$$

[0044] 直到网络收敛为止, 将网络最后一层的输出, 作为经迭代优化的最优图像向量输出 \mathbf{x}^* 。本发明中, 卷积神经网络模型全连接层参数初始化方式采用Xavier初始化方式。经过全连接层之后就输出了最优图像的分类标签。

[0045] 本发明的训练参数设置为: 权重衰减因子为0.001, 动量为0.85。该网络模型可以在深度学习框架Tensorflow下实现。将上述构建并初始化的多元激活函数代入CNN卷积神经网络模型, 即得到基于多元激活函数的稀疏正则化神经网络模型。再利用训练样本图像对构建的模型进行训练。本发明具体实施时, 均采用反向传播算法, 用带Nesterov动量的随机梯度下降算法进行模型训练。训练过程迭代1000轮, 图像批大小为256, 即每256张图片为一个批次, 学习速率初始值为0.1。实验重复20次, 报告20次实验的平均结果。

[0046] 4) 利用步骤3) 构建并训练好的基于多元激活函数的稀疏正则化神经网络模型, 对测试图像样本进行识别, 网络输出最优图像的标签, 即得到预测的图像分类标签, 由此实现图片分类识别, 其中测试样本无需经过图像预处理。

[0047] 与现有技术相比, 本发明的有益效果是:

[0048] 本发明提供了一种基于多元激活函数的稀疏正则化神经网络模型的图像分类方法, 通过训练多元激活函数的参数隐式地计算多元正则化器。在具体实施中, 可以将任意一个已有CNN模型中的激活函数采用本发明设计的多元激活函数进行正则化, 得到一个多元正则化的CNN模型, 最后用该模型进行图像分类。

[0049] 本发明利用近端梯度算法, 将不可分离稀疏正则化学习问题转化为多元激活函数的学习问题。与现有的图像分类模型相比, 本发明取得了更好的效果, 可以用更少的模型参数, 在图像分类数据集上达到更低的图像分类错误率。

附图说明

[0050] 图1是基于多元激活函数的稀疏正则化神经网络模型处理图像的单变量正则化器 $g(x)$ 的示意图;

[0051] 其中, (a) - (h) 分别表示FashionMNIST, MNIST, DIGITS, CIFAR-10, CIFAR-100, SDD, PENDIGITS, Caltech-101数据集上学习而得到的单变量的正则化器 $g(x)$ 。

[0052] 图2是本发明提供的神经网络模型MAF-SRL不同层数N对学习稀疏正则化器性能的影响图;

[0053] 其中, (a) 为MAF-SRL不同层数N在FashionMNIST, MNIST, DIGITS, CIFAR-10上的实施效果; (b) 为MAF-SRL不同层数N在CIFAR-100, SDD, PENDIGITS, Caltech-101上的实施效果。

具体实施方式

[0054] 下面结合附图, 通过实施例进一步描述本发明, 但不以任何方式限制本发明的范围。

[0055] 本发明提供一种基于多元激活函数的稀疏正则化神经网络模型MAF-SRL的图像分类方法, 训练多元激活函数的参数隐式地计算多元正则化器, 用于高效的进行图像分类。包括以下步骤:

[0056] 步骤1: 将图像数据分为训练样本和测试样本, 本实施例所有数据集为FASHION-MNIST, MNIST, DIGITS, CIFAR-10, CIFAR-100, SDD, PENDIGITS和Caltech-10数据集, 其中CIFAR-10和CIFAR-100训练数据50,000张, 测试数据10,000张, 其他数据集训练集和测试集比例为8:2。

[0057] 步骤2: 对训练样本图像进行图像预处理。图像预处理操作包括: 同态滤波和图像平滑。

[0058] 图像 $g(i, j)$ 是由光源产生的照度场 $g_1(i, j)$ 和目标的反射系数场 $g_2(i, j)$ 共同作用下产生的, 三者的关系是:

[0059] $g(i, j) = g_1(i, j) g_2(i, j)$

[0060] 照度场 $g_1(i, j)$ 变化缓慢, 在频谱上其能量集中于低频, 反射系数场 $g_2(i, j)$ 包含了所需要的图像信息, 在空间中的变化较快, 能量集中在高频部分。采用同态滤波方法, 把这两个分量分开处理, 处理步骤如下:

[0061] (1) 对上式两边取对数, 得到两个加性分量, 这是同态滤波方法中的第一步:

[0062] $\ln g(i, j) = \ln g_1(i, j) + \ln g_2(i, j)$

[0063] (2) 对上式两边同时进行傅立叶变换, 得到:

[0064] $G(u, v) = F\{\ln g(i, j)\} = F\{\ln g_1(i, j)\} + F\{\ln g_2(i, j)\} = G_1(u, v) + G_2(u, v)$

[0065] (3) 为了消除照度的起伏, 应该压缩 $G_1(u, v)$ 分量, 为了增加图像本身的对比度, 应该增强 $G_2(u, v)$ 分量。因此使用同态滤波器来处理 $G(u, v)$, 其中频域中的点 (u, v) 与频率中心的距离 $D(u, v) = \sqrt{u^2 + v^2}$, 低频增益 $H_L = 1$, 高频增益 $H_H = 3$ 。

[0066] $G^*(u, v) = H(u, v) G(u, v) = H(u, v) G_1(u, v) + H(u, v) G_2(u, v)$

[0067] (4) 取傅立叶反变换和指数运算, 得到图像 $g_3(i, j)$:

[0068] $g_3(i, j) = \exp(F^{-1}\{G^*(u, v)\})$

[0069] 为了减少图像中的噪声,需要对图像进行平滑处理,本发明实施例采用的图像平滑是空域的局部平均法,这种方法易于实现、效果也比较好。经局部平均法处理后的 $g_4(i, j)$ 为:

$$[0070] \quad g_4(i, j) = \frac{1}{N_S} \sum_{(i, j) \in S} [g_3(i, j)]$$

[0071] 其中, S 是点 (i, j) 邻域内的点集,由于本发明实施例选取的卷积模板为:

$$\frac{1}{16} \begin{bmatrix} 1 & 2 & 1 \\ 2 & 4 & 2 \\ 1 & 2 & 1 \end{bmatrix}, \text{所以邻域内的点的个数为 } N_S = 16。$$

[0072] 步骤3:将图像预处理过的训练样本通过本发明得到的模型进行训练。本发明可以选用任意一个已有的CNN架构,然后将其中的激活函数采用本发明构建的多元激活函数,得到一个基于多元激活函数的稀疏正则化神经网络模型。在本实施例中,我们选用ResNet作为网络基本架构。其中,训练多元激活函数的稀疏正则化神经网络模型的步骤如下:

[0073] 3-1) 设计多元激活函数。

[0074] 输入图像 x ,初始化CNN卷积神经网络中多元激活函数的学习参数 $\hat{\mathbf{A}}_{(0)}, \hat{\mathbf{q}}_{(0)}, b_{(0)}, \mathcal{U}_{(0)}$,计数器 $l=0$ 。

[0075] 本发明中,多元激活函数被定义为: $\xi(\mathbf{x}) = \mathbf{A}^{-T} [\hat{\xi}(\mathbf{A}^T \mathbf{x} + \mathbf{b}) - \mathbf{b}] : \mathbb{R}^n \rightarrow \mathbb{R}^n$

[0076] 其中, $\mathbf{A} = (a_1, \dots, a_M)$, $\mathbf{q} = (q_1, \dots, q_M)^T$, $\mathbf{b} = (b_1, \dots, b_M)^T$ 。

[0077] 本发明定义多元正则化器为: $\mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \hat{\xi}^{-1}(y) dy - \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} + b_i)^2 \right),$

通过构建一个近端算子: $\mathbf{y} = \operatorname{argmin}_{\mathbf{y}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + \mathcal{G}(\mathbf{y}) = \xi(\mathbf{x})$,学习多元正则化器能被转换为学习多元激活函数。接下来定义多元激活函数 $\xi(\mathbf{x})$ 包括的单元激活函数 $\hat{\xi}(\mathbf{x})$,它包含两个可学习的参数 (μ_1, μ_2) ,其中 $\mu_1 = (\eta_1, \delta_1)$, $\mu_2 = (\eta_2, \delta_2)$, $0 \leq \delta_1 \leq \delta_2$, $\eta_1, \eta_2 > 0$

$$[0078] \quad \hat{\xi}_{(\mu_1, \mu_2)}(x) = \begin{cases} \eta_2(x - \delta_2) + \eta_1(\delta_2 - \delta_1), & \delta_2 \leq x, \\ \eta_1(x - \delta_1), & \delta_1 \leq x < \delta_2, \\ 0, & -\delta_1 \leq x < \delta_1, \\ \eta_1(x + \delta_1), & -\delta_2 \leq x < -\delta_1, \\ \eta_2(x + \delta_2) + \eta_1(\delta_1 - \delta_2), & x < -\delta_2, \end{cases}$$

[0079] 3-2) 定义多元正则化器。

[0080] 本发明定义多元正则化器 $\mathcal{G}(\mathbf{x})$ 为:

$$[0081] \quad \mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \hat{\xi}^{-1}(y) dy - \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} + b_i)^2 \right),$$

[0082] 其中, a_i, q_i, b_i 分别是 $\mathbf{A}, \mathbf{q}, \mathbf{b}$ 中的元素;

[0083] 通过构建一个近端算子: $\mathbf{y} = \operatorname{argmin}_{\mathbf{y}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}\|^2 + \mathcal{G}(\mathbf{y})$ 作为单元激活函数 $\xi(\mathbf{x})$,将学习多元正则化器转换为学习多元激活函数。

[0084] 3-3) 定义单元激活函数。

[0085] 接下来定义多元激活函数 $\xi(\mathbf{x})$ 包括的单元激活函数 $\hat{\xi}_{(\mu_1, \mu_2)}(x)$,它包含两个可学习

的参数 (μ_1, μ_2) , 其中 $\mu_1 = (\eta_1, \delta_1)$, $\mu_2 = (\eta_2, \delta_2)$, $0 \leq \delta_1 \leq \delta_2$, $\eta_1, \eta_2 > 0$. $\eta_1, \delta_1, \eta_2, \delta_2$ 都是训练过程中学习到的参数。 η_1, η_2 分别为第一个和第二个单变量权重参数, δ_1 和 δ_2 分别为第一个和第二个单变量偏置参数。 $\hat{\xi}_{(\mu_1, \mu_2)}(x)$ 是 $\mathcal{G}(\mathbf{x}) = \sum_{i=1}^n q_i \left(\int_0^{\mathbf{a}_i^T \mathbf{x} + b_i} \xi^{-1}(y) dy - \frac{1}{2} (\mathbf{a}_i^T \mathbf{x} + b_i)^2 \right)$ 中的 $\xi^{-1}(y)$ 。单元激活函数的形式是

$$[0086] \quad \hat{\xi}_{(\mu_1, \mu_2)}(x) = \begin{cases} \eta_2(x - \delta_2) + \eta_1(\delta_2 - \delta_1), & \delta_2 \leq x, \\ \eta_1(x - \delta_1), & \delta_1 \leq x < \delta_2, \\ 0, & -\delta_1 \leq x < \delta_1, \\ \eta_1(x + \delta_1), & -\delta_2 \leq x < -\delta_1, \\ \eta_2(x + \delta_2) + \eta_1(\delta_1 - \delta_2), & x < -\delta_2, \end{cases}$$

[0087] 3-4) 对输入图像 $\mathbf{x} = \{x_1, x_2, \dots, x_n\}$ 进行训练过程的初始化。

[0088] 定义初始化的迭代求解 \mathbf{x} 的公式为

$$[0089] \quad \mathbf{x}_{(0)} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T \mathbf{x} - (\mathbf{A} \text{diag}(\mathbf{q}))^{-1} \nabla \phi(\mathbf{x})) + \mathbf{b}) - \mathbf{b}]。$$

[0090] $\mathbf{x}_{(0)}$ 表示输入图像第0次迭代的向量。

[0091] 3-5) 迭代求解

[0092] 本发明首先通过近端梯度的迭代算法进行求解,

$$[0093] \quad \mathbf{x}_{k+1} = \underset{\mathbf{y}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{y} - \mathbf{x}_k + \mathbf{Q}^{-1} \nabla f(\mathbf{x}_k)\|_{\mathbf{Q}}^2 + \mathcal{G}(\mathbf{y})$$

[0094] 接下来令 $\mathbf{r}^{(k)} = \mathbf{x}^{(k)} - \frac{1}{L} \nabla \phi(\mathbf{x}^{(k)})$, 则 $\operatorname{Prox}_{\alpha \mathcal{G}}(\mathbf{r}^{(k)}) = \underset{\mathbf{x}}{\operatorname{argmin}} \frac{1}{2} \|\mathbf{x} - \mathbf{r}^{(k)}\|_F^2 + \alpha \mathcal{G}(\mathbf{x})$, 得到迭代公式:

$$[0095] \quad \mathbf{x}^{(k+1)} = \xi(\mathbf{x}^{(k)} - \mathbf{Q}^{-1} \nabla \phi(\mathbf{x}^{(k)}))$$

[0096] 由于 $\mathbf{x}^{(k+1)}$ 是步骤3-1)通过近端梯度算法将学习多元正则化子(多元正则化器)转化为学习多元激活函数 $\xi(\mathbf{x})$, 所以得到:

$$[0097] \quad \mathbf{x}_{k+1} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T (\mathbf{x}_k - \mathbf{Q}^{-1} \nabla f(\mathbf{x}_k)) + \mathbf{b}) - \mathbf{b})]$$

[0098] 进一步化简得到最后迭代求解的表达式:

$$[0099] \quad \mathbf{x}_{k+1} = \mathbf{A}^{-T} [(\hat{\xi}(\mathbf{A}^T \mathbf{x}_k - (\mathbf{A} \text{diag}(\mathbf{q}))^{-1} \nabla f(\mathbf{x}_k)) + \mathbf{b}) - \mathbf{b}]$$

[0100] 3-6) 进行神经网络的学习训练

[0101] 对神经网络每一层 i ($i=1, 2, 3, \dots, N$), 逐层学习激活函数进而迭代求解 \mathbf{x} :

$$[0102] \quad \mathbf{x}_{(i)} = \mathbf{A}_{(i)}^{-T} \left[\left(\hat{\xi}(\mathbf{A}_{(i)}^T \mathbf{x}_{(i-1)} - (\mathbf{A}_{(i)} \text{diag}(\mathbf{q}_{(i)}))^{-1} \nabla \phi(\mathbf{x}_{(i-1)})) + \mathbf{b}_{(i)} \right) - \mathbf{b}_{(i)} \right]$$

[0103] 3-7) 计算神经网络每一层的损失函数: N 代表第 N 层的 \mathbf{x} 图像向量。

$$[0104] \quad \mathcal{L}(\hat{\mathbf{A}}, \hat{\mathbf{q}}, \mathbf{b}, \mathbf{u}) = \frac{1}{2} \|\mathbf{x}_{(N)} - \mathbf{x}\|_F^2$$

[0105] 直到网络收敛为止, 输出网络最后一层的输出图像 $\mathbf{x}_{(N)}$, 作为最优图像 \mathbf{x}^* 输出。本发明中, 卷积神经网络全连接层参数初始化方式采用Xavier初始化方式。经过全连接层之后就输出了最优图像的分类标签。

[0106] 步骤4: 将测试样本(无需进行图像预处理)分别通过训练好的网络进行识别, 得到预测的分类标签。

[0107] 本发明中的模型均使用反向传播算法, 利用带Nesterov动量的随机梯度下降算法

进行训练,动量为0.85。该网络模型在深度学习框架Tensorflow下实现。将上述构建并初始化的多元激活函数代入CNN模型。再利用训练样本图像对该模型进行训练。本发明具体实施时,均采用反向传播算法,用带Nesterov动量的随机梯度下降算法进行模型训练。训练过程迭代1000轮,批大小为256,即每256张图片为一个批次,学习速率初始值为0.1。实验重复20次,报告20次实验的平均结果。

[0108] 表1给出了本发明模型MAF-SRL和其他正则化子在不同数据集上的图像分类准确率和稀疏度比较,较高的准确率意味着训练后的网络更适合图像分类任务。非零参数比例越小,稀疏正则化能力越好。

[0109] 表1本发明模型和其他正则化子在不同数据集上的图像分类准确率和稀疏度

Dataset	Measure	ℓ_1	ℓ_{1-2}	SGL	CGES	SCAD	capped- ℓ_1	LSP	MCP	MAF-SRL (ours)
Fashion-MNIST	accuracy	0.9124	0.9281	0.8924	0.8873	0.8671	0.8982	0.9031	0.9127	0.9421
	weight sparsity	0.2398	0.4363	0.4218	0.2819	0.5728	0.6629	0.2763	0.3397	0.1537
MNIST	accuracy	0.9642	0.9538	0.9863	0.9837	0.9824	0.9563	0.9563	0.9623	0.9921
	weight sparsity	0.1727	0.2735	0.1029	0.2013	0.1197	0.1126	0.0928	0.3328	0.0629
DIGITS	accuracy	0.8638	0.8837	0.8542	0.8837	0.8682	0.8538	0.8772	0.8831	0.9028
	weight sparsity	0.3387	0.2928	0.2901	0.4283	0.4419	0.2765	0.5319	0.4019	0.1774
CIFAR-10	accuracy	0.8238	0.8188	0.8092	0.8542	0.8452	0.8562	0.8458	0.8229	0.8759
	weight sparsity	0.6784	0.5829	0.5429	0.4492	0.5186	0.6294	0.5529	0.3165	0.2396
CIFAR-100	accuracy	0.7329	0.7219	0.6872	0.7239	0.6549	0.7129	0.7278	0.7362	0.7769
	weight sparsity	0.5587	0.4982	0.8829	0.7623	0.4927	0.6549	0.5498	0.4892	0.3225
SDD	accuracy	0.9829	0.9669	0.9539	0.9827	0.9567	0.9632	0.9862	0.9685	0.9941
	weight sparsity	0.3092	0.4294	0.2397	0.4962	0.2981	0.3982	0.5729	0.4839	0.1703
PENDIGITS	accuracy	0.9852	0.9902	0.9762	0.9683	0.9719	0.9629	0.9739	0.9827	0.9958
	weight sparsity	0.6931	0.3397	0.6791	0.3018	0.2973	0.7538	0.5392	0.4492	0.1778
Caltech-101	accuracy	0.9733	0.9758	0.9883	0.9901	0.9632	0.9857	0.9683	0.9775	0.9949
	weight sparsity	0.3679	0.4133	0.5582	0.6271	0.3036	0.2279	0.3864	0.4272	0.1762

[0110] 从表1可以看出,与其他基线相比,本发明构建的MAF-SRL模型具有最好的性能。在所有数据集上,MAF-SRL既具有最高的精度,又具有最小的稀疏性(非零权值百分比),表明本发明学习的多元稀疏正则化方法确实有效,能够更好地适应数据。

[0112] 图1展示了学习到的不同数据集的单变元正则化器 $g(x)$,其中各子图参数的选值分别是:

[0113] (a) $\eta_1=1.37, \eta_2=0.22, \delta_1=0.46, \delta_2=1.57$.

[0114] (b) $\eta_1=1.46, \eta_2=0.24, \delta_1=0.44, \delta_2=1.48$.

[0115] (c) $\eta_1=1.35, \eta_2=0.34, \delta_1=0.36, \delta_2=1.31$.

[0116] (d) $\eta_1=1.41, \eta_2=0.62, \delta_1=0.62, \delta_2=1.49$.

[0117] (e) $\eta_1=1.33, \eta_2=0.36, \delta_1=0.48, \delta_2=1.47$.

[0118] (f) $\eta_1=1.51, \eta_2=0.64, \delta_1=0.89, \delta_2=1.77$.

[0119] (g) $\eta_1=1.34, \eta_2=0.45, \delta_1=0.33, \delta_2=1.33$.

[0120] (h) $\eta_1=1.44, \eta_2=0.27, \delta_1=0.47, \delta_2=1.53$.

[0121] 我们可以看到, $g(x)$ 不是凸的,并且在不同的数据集上差异显著,表明我们学习的稀疏正则化器能够很容易地适应数据。

[0122] 本发明进一步研究了层数N对学习的稀疏正则化器性能的影响,结果如图2所示。层数N的取值范围为2-30,学习率固定为0.1。从图2中可以发现,当层数 $N>16$ 时,精度基本稳定。

[0123] 需要注意的是,公布实施例的目的在于帮助进一步理解本发明,但是本领域的技术人员可以理解:在不脱离本发明及所附权利要求的范围内,各种替换和修改都是可能的。因此,本发明不应局限于实施例所公开的内容,本发明要求保护的范围以权利要求书界定的范围为准。

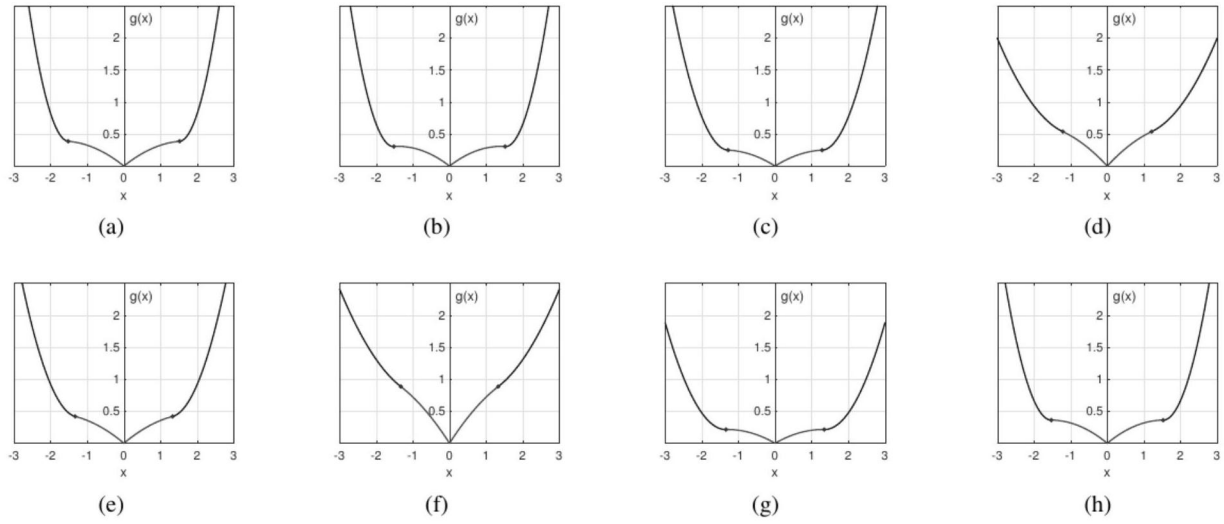


图1

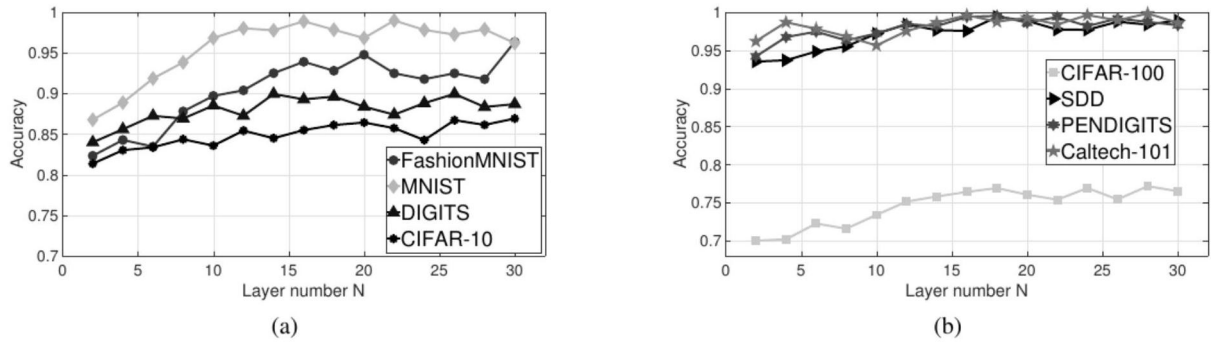


图2