(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 119625407 A
(43)申请公布日 2025.03.14

(21)申请号 202411710062.1

(22)申请日 2024.11.27

(71)申请人 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 10/26*(2022.01)
*G06V 10/82*(2022.01)
*G06N 3/0464*(2023.01)
*G06N 3/084*(2023.01)

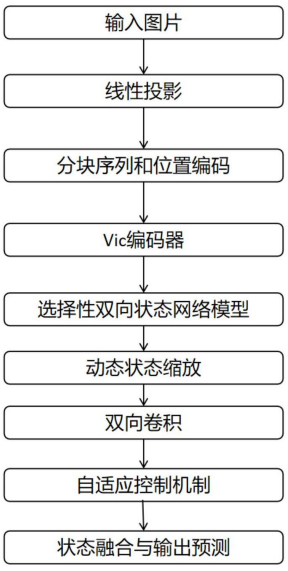
权利要求书3页 说明书9页 附图1页

(54)发明名称

# 一种基于自适应控制状态空间模型的图像分类方法

(57)摘要

本发明公布了一种基于自适应控制状态空间模型的图像分类方法,包括:图像分割模块和ViC编码器;设计的ViC编码器包括选择性双向状态模型模块、动态状态缩放模块、双向卷积模块、自适应控制状态空间模型、自适应控制机制模块。通过设计选择性双向状态模型(SBSM)、动态状态缩放(DSS)、双向卷积(Bi-Conv)和自适应控制机制(ACM),在处理大规模图像数据集时,更好地捕捉长序列依赖和复杂视觉模式,提高图像分类的准确性、效率和泛化能力。




CN 119625407 A

1.一种基于自适应控制状态空间模型的图像分类方法,其特征在于,包括如下步骤:

S1)将输入图像分割成图像小块,将图像小块转换为块标记,并处理形成初始标记序列;

S2)设计ViC编码器;利用ViC编码器对初始块标记序列进行双向处理,以捕获初始块标记序列中的长程依赖,得到图像块编码序列;

ViC编码器包括选择性双向状态模型模块、动态状态缩放模块、双向卷积模块、自适应控制状态空间模型、自适应控制机制模块;

ViC编码器中的选择性双向状态模型包括:前向选择性状态网络和后向选择性状态网络;前向选择性状态网络用于结合当前网络层的前向状态与前一网络层的后向输出,生成包含相关信息的选择性状态向量;后向选择性状态网络用于结合当前网络层的后向状态与后一网络层的前向输出,生成包含相关信息的选择性状态向量;通过所述选择性双向状态模型,首先对输入的标记序列进行前向传播,生成前向编码序列;再对前向编码序列进行后向传播,生成后向编码序列;将前向编码序列和后向编码序列进行融合,生成当前网络层的输出,即当前网络层的图像块编码序列;

通过动态状态缩放方法调整不同状态维度对输出的影响;包括:对选择性状态向量进行元素级缩放,生成动态状态缩放向量;再将动态状态缩放向量与选择性状态向量进行元素级乘法操作,生成调整后的状态向量;

双向卷积方法用于捕获编码序列中的局部模式;包括:使用可分离的1D卷积核对调整后的状态向量进行前向卷积,生成前向卷积输出;使用可分离的1D卷积核对调整后的状态向量进行后向卷积,生成后向卷积输出;

通过ViC编码器中的自适应控制状态空间模型学习状态转移矩阵和输入映射矩阵,以自适应地更新前状态和后状态;

通过自适应控制机制模块,动态调整模型的内部参数;所述自适应控制机制包括:计算当前层输出标记序列与期望输出标记序列之间的偏差;根据偏差更新控制增益矩阵;计算控制输入,以调整模型输出,使得所述偏差最小,由此得到最终的输出标记序列;

S3)将最终的输出标记序列输入到多层感知器的头部中进行预测,获得最终的图像分类。

2.如权利要求1所述基于自适应控制状态空间模型的图像分类方法,其特征在于,具体是采用线性投影技术将每个图像块映射到高维空间,得到图像块嵌入序列。

3.如权利要求1所述基于自适应控制状态空间模型的图像分类方法,其特征在于,步骤S2)中,Vic编码器为双向序列建模模块,用于对输入序列进行双向传播;Vic编码器的第1层的定义表示为:

$T_l = \mathrm{Vic}(T_{l-1}) + T_{l-1}$

其中,Vic表示Vic编码器;$\mathrm{Vic}(T_{l-1})$表示第l层的前向传播输出结果序列,$T_{l-1}$表示第l-1层输出的结果序列。

4.如权利要求3所述基于自适应控制状态空间模型的图像分类方法,其特征在于,具体是采用RMSNorm归一化技术,对Vic编码器输出的图像序列进行归一化处理,并输入到多层感知器的头部,由此获得最终的图像预测类别结果。

5.如权利要求3所述基于自适应控制状态空间模型的图像分类方法,其特征在于,步骤

S2)中,具体是使用Swish函数对选择性状态向量进行元素级缩放,生成动态状态缩放向量。

6.如权利要求3所述基于自适应控制状态空间模型的图像分类方法,其特征在于,选择性双向状态网络模型表示为:

$$\overrightarrow{\mathbf{s}}_l = \overrightarrow{\mathbf{S}}_\theta(\overrightarrow{\mathbf{x}}_l, \overleftarrow{\mathbf{T}}_{l-1})$$

$$\overleftarrow{\mathbf{s}}_l = \overleftarrow{\mathbf{S}}_\theta(\overleftarrow{\mathbf{x}}_l, \overrightarrow{\mathbf{T}}_{l+1})$$

其中,前向选择性状态网络$\overrightarrow{\mathbf{S}}_\theta$负责将当前层的前向状态$\overrightarrow{\mathbf{x}}_l$与来自上一层的反向输出$\overleftarrow{\mathbf{T}}_{l-1}$结合起来,生成一个包含相关信息的向量$\overrightarrow{\mathbf{s}}_l$;反向选择性状态网络$\overleftarrow{\mathbf{S}}_\theta$负责将当前层的反向状态$\overleftarrow{\mathbf{x}}_l$与来自下一层的前向输出$\overrightarrow{\mathbf{T}}_{l+1}$结合起来,生成另一个包含相关信息的向量$\overleftarrow{\mathbf{s}}_l$。

7.如权利要求6所述基于自适应控制状态空间模型的图像分类方法,其特征在于,具体是通过动态状态缩放向量调整不同状态维度对输出的影响,表示为:

$$\overrightarrow{\mathbf{x}}'_l = \overrightarrow{\mathbf{r}}_l \odot \overrightarrow{\mathbf{s}}_l$$

$$\overleftarrow{\mathbf{x}}'_l = \overleftarrow{\mathbf{r}}_l \odot \overleftarrow{\mathbf{s}}_l$$

动态状态缩放通过动态状态缩放向量$\overrightarrow{\mathbf{r}}_l$和$\overleftarrow{\mathbf{r}}_l$对选择性状态向量$\overrightarrow{\mathbf{s}}_l$和$\overleftarrow{\mathbf{s}}_l$进行逐元素缩放。

8.如权利要求7所述基于自适应控制状态空间模型的图像分类方法,其特征在于,使用可分离的1D卷积核捕获序列中的局部模式,表示为:

$$\overrightarrow{\mathbf{h}}_l = \overrightarrow{\mathbf{W}}_{\text{conv}} * \overrightarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$

$$\overleftarrow{\mathbf{h}}_l = \overleftarrow{\mathbf{W}}_{\text{conv}} * \overleftarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$

双向卷积更新通过使用可分离的1D卷积核来捕捉序列中的局部模式;

双向卷积更新分别计算前向卷积输出$\overrightarrow{\mathbf{h}}_l$和反向卷积输出$\overleftarrow{\mathbf{h}}_l$,表示为:

$$\overrightarrow{\mathbf{h}}_l = \overrightarrow{\mathbf{W}}_{\text{conv}} * \overrightarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$

$$\overleftarrow{\mathbf{h}}_l = \overleftarrow{\mathbf{W}}_{\text{conv}} * \overleftarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$

其中,$\overrightarrow{\mathbf{W}}_{\text{conv}}$和$\overleftarrow{\mathbf{W}}_{\text{conv}}$是可分离的1D卷积核,$\overrightarrow{\mathbf{x}}'_l$和$\overleftarrow{\mathbf{x}}'_l$是经过动态缩放的状态向量。

9.如权利要求8所述基于自适应控制状态空间模型的图像分类方法,其特征在于,自适应控制状态空间模型包括:

1)学习状态转移矩阵和输入映射矩阵,表示为:

$$\overrightarrow{\mathbf{A}}_\theta \leftarrow \overrightarrow{\mathbf{A}}_\theta(\overrightarrow{\mathbf{h}}_l), \quad \overrightarrow{\mathbf{B}}_\theta \leftarrow \overrightarrow{\mathbf{B}}_\theta(\overrightarrow{\mathbf{h}}_l)$$

$$\overleftarrow{\mathbf{A}}_\theta \leftarrow \overleftarrow{\mathbf{A}}_\theta(\overleftarrow{\mathbf{h}}_l), \quad \overleftarrow{\mathbf{B}}_\theta \leftarrow \overleftarrow{\mathbf{B}}_\theta(\overleftarrow{\mathbf{h}}_l)$$

其中,$\overrightarrow{\mathbf{A}}_\theta$、$\overleftarrow{\mathbf{A}}_\theta$、$\overrightarrow{\mathbf{B}}_\theta$和$\overleftarrow{\mathbf{B}}_\theta$是可学习的函数,用于将卷积输出$\overrightarrow{\mathbf{h}}_l$和$\overleftarrow{\mathbf{h}}_l$映射到对应的状态转移矩阵和输入映射矩阵;这些可学习的函数由多层神经网络组成,通过模型训练学习得到;

2)通过MLP计算得到动态调整项,表示为:

$$\Delta\overrightarrow{\mathbf{A}}_l = \mathrm{MLP}(\overrightarrow{\mathbf{h}}_l)$$

$$\Delta\overleftarrow{\mathbf{A}}_l = \mathrm{MLP}(\overleftarrow{\mathbf{h}}_l)$$

其中,MLP表示多层感知器;$\Delta\overrightarrow{\mathbf{A}}_l$和$\Delta\overleftarrow{\mathbf{A}}_l$是动态调节项,是通过可学习网络MLP,根据$\overrightarrow{\mathbf{h}}_l$和$\overleftarrow{\mathbf{h}}_l$计算得到;

3)更新前状态和后状态,表示为:

$$\overrightarrow{\mathbf{h}}_{l+1} = (\overrightarrow{\mathbf{A}}_l + \Delta\overrightarrow{\mathbf{A}}_l)\overrightarrow{\mathbf{h}}_l + (\overrightarrow{\mathbf{B}}_l)\mathbf{u}_l$$

进一步表示为:

$$\overrightarrow{\mathbf{x}}_{l+1} = (\overrightarrow{\mathbf{A}}_l + \Delta\overrightarrow{\mathbf{A}}_l)\overrightarrow{\mathbf{x}}'_l + (\overrightarrow{\mathbf{B}}_l)\mathbf{u}_l$$

$$\overleftarrow{\mathbf{x}}_{l-1} = (\overleftarrow{\mathbf{A}}_l + \Delta\overleftarrow{\mathbf{A}}_l)\overleftarrow{\mathbf{x}}'_l + (\overleftarrow{\mathbf{B}}_l)\mathbf{u}_l$$

其中,$\overrightarrow{\mathbf{h}}_l$是前状态,$\overrightarrow{\mathbf{h}}_{l+1}$是后状态;$\overrightarrow{\mathbf{x}}'_l$和$\overleftarrow{\mathbf{x}}'_l$是前一时刻的前向和反向状态,通过更新后的状态转移矩阵和输入映射矩阵,得到新的前向状态和反向状态$\overrightarrow{\mathbf{x}}_{l+1}$和$\overleftarrow{\mathbf{x}}_{l-1}$;$u_1$表示当前层的控制输入,是根据当前层的输出$T_1$和期望输出$d_1$之间的偏差$e_1$计算得到。

10.如权利要求8所述基于自适应控制状态空间模型的图像分类方法,其特征在于,具体是通过对前向和反向信息进行动态控制,以平衡两个方向的上下文信息;状态融合与输出预测包括如下步骤:

分别计算前向输出$\overrightarrow{\mathbf{y}}_l$和反向输出$\overleftarrow{\mathbf{y}}_l$,表示为:

$$\overrightarrow{\mathbf{y}}_l = \overrightarrow{\mathbf{W}}_y\overrightarrow{\mathbf{x}}_{l+1} + \overrightarrow{\mathbf{b}}_y$$

$$\overleftarrow{\mathbf{y}}_l = \overleftarrow{\mathbf{W}}_y\overleftarrow{\mathbf{x}}_{l-1} + \overleftarrow{\mathbf{b}}_y$$

其中,$\overrightarrow{\mathbf{W}}_y$和$\overleftarrow{\mathbf{W}}_y$是输出映射矩阵,$\overrightarrow{\mathbf{b}}_y$和$\overleftarrow{\mathbf{b}}_y$是偏置向量;

使用SwiGLU门控函数对前向和反向输出进行融合,生成当前层的输出$T_1$,表示为:

$$\mathbf{T}_l = \mathrm{SwiGLU}(\overrightarrow{\mathbf{W}}_g\overrightarrow{\mathbf{T}}_l + \overrightarrow{\mathbf{b}}_g)\odot\overrightarrow{\mathbf{y}}_l + \mathrm{SwiGLU}(\overleftarrow{\mathbf{W}}_g\overleftarrow{\mathbf{T}}_l + \overleftarrow{\mathbf{b}}_g)\odot\overleftarrow{\mathbf{y}}_l$$

其中,$\overrightarrow{\mathbf{T}}_l$和$\overleftarrow{\mathbf{T}}_l$分别是当前层的前向和反向输入,$\overrightarrow{\mathbf{W}}_g$和$\overrightarrow{\mathbf{b}}_g$是前向传播的门控参数,$\overleftarrow{\mathbf{W}}_g$和$\overleftarrow{\mathbf{b}}_g$是反向传播的门控参数。

# 一种基于自适应控制状态空间模型的图像分类方法

## 技术领域

[0001] 本发明涉及模式识别、机器学习、人工智能、图像处理技术领域,具体涉及一种基于自适应控制状态空间模型的图像分类方法。

## 背景技术

[0002] 图像分类作为计算机视觉领域的一个核心任务,其重要性随着技术进步而日益凸显。这一任务涉及到将图像数据按照其视觉内容自动划分到不同的类别中,广泛应用于安防监控、医疗诊断、自动驾驶、内容推荐等多个领域。然而,传统的图像分类方法依赖于手工特征提取,如边缘、角点、纹理等,这些特征提取方法不仅复杂,而且在处理大规模图像数据集时效率低下,难以适应现代应用中对实时性和准确性的高要求。

[0003] 随着深度学习技术的兴起,基于数据驱动的特征学习逐渐成为主流。深度学习模型,尤其是卷积神经网络(CNN),通过其强大的特征提取能力,在图像分类任务中取得了革命性的进展。CNN能够有效地捕捉图像的局部特征并逐层构建更为复杂和抽象的特征表示,从而在诸如ImageNet等大规模图像识别竞赛中取得了前所未有的成绩。

[0004] 然而,尽管CNN在图像分类领域取得了巨大成功,它们在处理长序列依赖和复杂视觉模式时仍存在局限性。例如,CNN通常采用局部感受野进行特征提取,这限制了其对长距离依赖关系的捕捉能力。此外,CNN模型在处理高维数据时,参数数量庞大,计算复杂度高,这在一定程度上限制了其在资源受限的设备上的应用。

[0005] 为了解决这些问题,研究者们开始探索更为先进的模型结构。视觉变换器(ViT)作为一种新兴的模型,通过将图像分割成小块(即tokens),并利用自注意力机制处理这些tokens,展现了处理长序列依赖的能力。ViT模型在多个视觉任务中展现出了与CNN相媲美甚至更优的性能,但其计算复杂度和对大规模数据集的处理能力仍有待优化。

[0006] 此外,现有的图像分类方法在泛化能力上也面临挑战。在面对新的、未见过的数据时,如何保持高准确率的分类结果,是当前研究中的一个重要课题。为了提高模型的泛化能力,研究者们尝试了多种方法,包括数据增强、正则化技术、元学习等,但这些方法在实际应用中仍存在局限性,在处理大规模图像数据集时,难以有效地捕捉长序列依赖和复杂视觉模式,在图像分类任务中的性能很难得以提升。

## 发明内容

[0007] 本发明提供一种基于自适应控制状态空间模型的图像分类方法,称为Vision Conba(ViC,视觉曼巴),该方法能够有效地处理图像数据,提高图像分类的准确性和效率。

[0008] 本发明提出了一种图像分类方法(Vision Conba(ViC)模型),本发明构建的ViC模型包括图像分割模块并设计创新的ViC编码器;ViC编码器包括选择性双向状态模型(SBSM)、动态状态缩放(DSS)、双向卷积(Bi-Conv)和自适应控制机制(ACM),能够提高图像分类的准确性、效率和泛化能力。本发明提供的ViC模型方法能够在处理大规模图像数据集时,更好地捕捉长序列依赖和复杂视觉模式,从而在图像分类任务中取得更优异的性能。

[0009] 为方便起见,本发明定义以下术语名称:

[0010] 图像分割:将输入图像分割成小块,并将这些块转换为块标记。

[0011] ViC编码器:利用ViC编码器对块标记序列进行双向处理,以捕获块标记序列中的长程依赖。

[0012] 选择性双向状态网络模型(Selective Bidirectional State Model,SBSM):通过选择性状态网络确定当前输入相关的信息,并根据需要传递或遗忘信息。

[0013] 动态状态缩放(Dynamic State Scaling,DSS)网络模型:通过动态状态缩放向量调整图像状态维度(一维维度)对输出的影响。

[0014] 双向卷积(Bidirectional Convolution,Bi-Conv)网络模型:使用可分离的一维(1D)卷积核捕获序列中的局部模式。

[0015] 自适应状态空间网络模型(A-SSM):学习状态转移矩阵和输入映射矩阵,以自适应地更新前后状态。

[0016] 自适应控制机制(Adaptive Control Mechanism,ACM)网络模型:通过闭环反馈机制和自适应控制律动态调整模型的内部参数。

[0017] 本发明采用以下技术方案:

[0018] 一种基于自适应控制状态空间模型的图像分类方法,包括如下步骤:

[0019] 1)将输入图像分割成小块,并将这些块转换为块标记;

[0020] 2)对每个块标记进行线性投影,以扩展其维度;

[0021] 3)将扩展维度后的块标记与位置嵌入(位置编码)相结合,形成初始标记序列;

[0022] 4)设计ViC编码器;将初始标记序列输入到ViC编码器中进行处理,输出最终标记序列;在ViC编码器中的处理包括步骤5)~9):

[0023] 5)通过ViC编码器的选择性双向状态模型(SBSM)确定当前输入的初始标记序列相关的信息,并根据需要传递或遗忘信息;

[0024] 6)对步骤5)输出的标记序列,利用动态状态缩放(DSS)调整不同状态维度对输出标记序列的影响;

[0025] 7)使用双向卷积(Bi-Conv)捕获步骤6)输出的标记序列中的局部模式;

[0026] 8)设计自适应状态空间模型A-SSM,学习状态转移矩阵和输入映射矩阵,以自适应地更新前后状态;

[0027] 9)利用自适应控制机制(ACM)动态调整ACM模型的内部参数;得到最终标记序列输出;

[0028]

[0029] 10)将步骤9)最终的输出标记序列输入到多层感知器(MLP)头中,以获得最终的分类预测。

[0030] 具体实施时,本发明实现了一种基于自适应控制状态空间模型的图像分类系统,包括:图像分割模块和ViC编码器;ViC编码器为包括选择性双向状态模型模块、动态状态缩放模块、双向卷积模块、自适应控制状态空间模型、自适应控制机制模块。

[0031] 进一步地,本发明设计的ViC编码器的工作过程包括:对输入的标记序列进行前向传播,依次经过选择性双向状态模型模块、动态状态缩放模块、双向卷积模块、自适应控制状态空间模型、自适应控制机制模块的各个网络层,生成前向编码序列;对前向编码序列进

行后向传播,逆向经过选择性双向状态模型模块、动态状态缩放模块、双向卷积模块、自适应控制状态空间模型、自适应控制机制模块的各个网络层,生成后向编码序列;将前向编码序列和后向编码序列进行融合,生成当前网络层的输出序列。

[0032] 进一步地,选择性双向状态模型(SBSM)包括前向选择性状态网络和后向选择性状态网络,分别为:

[0033] 前向选择性状态网络,用于结合当前层的图像的前向状态与前一层的后向输出,生成包含相关信息的向量,即选择性状态向量;

[0034] 后向选择性状态网络,用于结合当前层的后向状态与后一层的前向输出,生成包含相关信息的选择性状态向量。

[0035] 进一步地,设计动态状态缩放(DSS)网络模型,包括:

[0036] 使用文献(Ramachandran,P.,Zoph,B.,&Le,Q.V.(2017).Searching for activation functions.arXiv preprint arXiv:1710.05941)中的Swish函数对选择性双向状态模型输出的选择性状态向量进行元素级缩放,生成动态状态缩放向量;将动态状态缩放向量与选择性状态向量进行元素级乘法操作,生成调整后的图像状态向量。

[0037] 进一步地,双向卷积(Bi-Conv)网络模型包括:

[0038] 使用可分离的1D卷积核对调整后的状态向量进行前向卷积,生成前向卷积输出标记序列;使用可分离的1D卷积核对调整后的状态向量进行后向卷积,生成后向卷积输出标记序列。

[0039] 进一步地,自适应控制状态空间模型(A-SSM)包括:

[0040] 学习状态转移矩阵和输入映射矩阵,以自适应地更新前向和后向状态;

[0041] 计算动态调整项,以自适应地调整状态转移矩阵。

[0042] 进一步地,自适应控制机制(ACM)包括:

[0043] 计算当前层输出与期望输出之间的偏差;

[0044] 根据偏差更新控制增益矩阵;

[0045] 计算控制输入,以调整模型输出,使其接近期望目标。

[0046] 与现有技术相比,本发明的图像分类方法具有以下技术优点:

[0047] 本发明提供一种基于自适应控制状态空间模型的图像分类技术,通过双向处理、自适应控制机制、选择性状态模型和动态状态缩放,在处理大规模图像数据集时,更好地捕捉长序列依赖和复杂视觉模式,从而在图像分类任务中取得更优异的性能,能够提高图像分类的准确性、效率和泛化能力。本发明的技术优势包括如下几方面:

[0048] 高效的长程依赖捕获:通过双向处理,能够有效地捕获图像中的长程依赖关系。

[0049] 自适应性强:通过自适应控制机制,模型能够根据输出误差动态调整其内部参数,提高模型的适应性和泛化能力。

[0050] 计算效率高:通过选择性状态模型和动态状态缩放,减少了不必要的计算,提高了模型的计算效率。

[0051] 泛化能力佳:在多种图像分类任务中表现出色,具有较好的泛化能力。

## 附图说明

[0052] 图1为本发明提供的方法的流程框图。

## 具体实施方式

[0053] 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。

[0054] 如图1所示,本发明方法通过创新的ViC编码器,以及选择性双向状态模型(SBSM)、动态状态缩放(DSS)、双向卷积(Bi-Conv)和自适应控制机制(ACM),能够提高图像分类的准确性、效率和泛化能力。

[0055] 以下详细描述本发明的具体实施方式:

[0056] 1、通过图像分割得到图像的块标记序列;

[0057] 输入图像首先被分割成小块,每个块的大小为P×P。这些块被转换为块标记,并进行线性投影以扩展其维度,得到块标记序列。

[0058] 2、通过ViC编码器进行处理得到图像块编码序列;

[0059] 将块标记序列输入到ViC编码器中。ViC编码器通过以下步骤进行处理:

[0060] 前向传播:计算前向编码序列$\overrightarrow{\mathbf{T}}_l$

[0061] 后向传播:计算后向编码序列$\overleftarrow{\mathbf{T}}_l$

[0062] 状态融合:通过SwiGLU门控函数融合前向和后向输出,生成当前层的输出编码序列$T_1$。SwiGLU门控函数可参考文献(Shazeer,N.(2020).Glu variants improve transformer.arXiv preprint arXiv:2002.05202)。

[0063] 在本发明构建的Vision Conba图像分类模型中,输入图像首先被分割为一系列的图像patches$\mathbf{x}_p^j$,其中J表示图像块patch的数量。为了更好地利用这些patches的信息,我们采用线性投影技术将每个图像块patch映射到一个高维空间,得到patch嵌入序列$\{\mathbf{T}_p^j\}_{j=1}^J$:

[0064] $$\mathbf{T}_0 = \{\mathbf{T}_p^j\}_{j=1}^J = [\mathbf{t}_{\text{cls}}; \{\mathbf{x}_p^j \mathbf{W}_{\text{proj}}\}_{j=1}^J] + \{\mathbf{e}_{\text{pos}}^j\}_{j=1}^J$$

[0065] 其中,$W_{proj}$是一个可学习的投影矩阵,它将原始的图像patches投影到$R^D$空间中,从而使得模型能够更好地捕捉patches之间的关联性。同时,$\{\mathbf{e}_{\text{pos}}^j\}_{j=1}^J$是一个位置嵌入序列,它为每个patches添加了位置信息,使得模型能够区分不同位置的patches,从而更好地处理图像的空间结构。受ViT和BERT的启发,我们也使用类令牌来表示整个图像块序列,记为$t_{cls}$。

[0066] 在得到patch嵌入序列后,我们将其输入到Vic编码器的第l层,以获得输出$T_l$。Vic编码器是一个双向序列建模模块,它通过对输入序列进行双向传播,能够有效地捕捉序列中的长程依赖关系。具体来说,Vic编码器的第l层定义如下:

[0067] $T_l = Vic(T_{l-1}) + T_{l-1}$

[0068] 其中,Vic表示Vic编码器,$Vic(T_{l-1})$表示第l层的前向传播序列结果,$T_{l-1}$表示第l-1层的输出。通过这种方式,Vic编码器能够对输入序列进行逐层编码,从而获得更高层次的特征表示。最后,我们对Vic编码器输出的序列(类别令牌)$T_l$进行归一化处理,并输入到多层感知器(MLP)头部以获得最终的预测p。归一化层采用的是RMSNorm归一化技术,它通过对每个特征值进行平方根处理,能够有效地提高模型的稳定性和泛化能力。具体来说,RMSNorm的数学公式如下:

[0069] $$\mathbf{f} = \mathrm{RMSNorm}(\mathbf{T}_L) = \left( \frac{\mathbf{T}_L}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} |\mathbf{T}_{L,i}|^2}} \right)$$

[0070] 其中,L是网络层的数量,$T_L$表示最后输出的类别令牌,N表示令牌的数量。通过RMSNorm归一化处理,我们可以使得模型更加稳定,从而提高模型的性能和泛化能力。

[0071] 3、选择性双向状态网络模型(SBSM)

[0072] 通过选择性双向状态网络模型,确定当前输入相关的信息,并根据需要传递或遗忘信息。

[0073] 具体公式如下:

[0074] $$\overrightarrow{\mathbf{s}}_l = \overrightarrow{\mathbf{S}}_\theta(\overrightarrow{\mathbf{x}}_l, \overleftarrow{\mathbf{T}}_{l-1})$$
$$\overleftarrow{\mathbf{s}}_l = \overleftarrow{\mathbf{S}}_\theta(\overleftarrow{\mathbf{x}}_l, \overrightarrow{\mathbf{T}}_{l+1})$$

[0075] 选择性双向状态更新是一个关键的机制,它通过选择性状态网络$\overrightarrow{\mathbf{S}}_\theta$和$\overleftarrow{\mathbf{S}}_\theta$来确定哪些状态信息对于当前输入是相关的,哪些是冗余的。这种机制使得模型能够根据当前的数据自适应地传递或遗忘信息,从而降低计算复杂性并提高效率。具体来说,前向选择性状态网络$\overrightarrow{\mathbf{S}}_\theta$负责将当前层的前向状态$\overrightarrow{\mathbf{x}}_l$与来自上一层的反向输出$\overleftarrow{\mathbf{T}}_{l-1}$结合起来,生成一个包含相关信息的向量$\overrightarrow{\mathbf{s}}_l$。同样,反向选择性状态网络$\overleftarrow{\mathbf{S}}_\theta$负责将当前层的反向状态$\overleftarrow{\mathbf{x}}_l$与来自下一层的前向输出$\overrightarrow{\mathbf{T}}_{l+1}$结合起来,生成另一个包含相关信息的向量$\overleftarrow{\mathbf{s}}_l$。选择性状态网络$\overleftarrow{\mathbf{S}}_\theta$和$\overleftarrow{\mathbf{S}}_\theta$利用SwiGLU作为激活函数,可以学习将当前状态$\overrightarrow{\mathbf{x}}_l$和$\overleftarrow{\mathbf{x}}_l$与来自相邻层的输出$\overrightarrow{\mathbf{T}}_{l+1}$和$\overleftarrow{\mathbf{T}}_{l-1}$相结合,选择性地保留对当前输入相关的信息。

[0076] 通过将SwiGLU应用于当前状态和相邻层输出,$\overrightarrow{\mathbf{S}}_\theta$和$\overleftarrow{\mathbf{S}}_\theta$能够学习到哪些信息对当前输入是相关的,从而选择性地更新状态信息。这种机制使得Vic块能够灵活地适应不同的输入序列,提高了模型在处理长序列任务时的性能。为了更具体地描述$\overrightarrow{\mathbf{S}}_\theta$和$\overleftarrow{\mathbf{S}}_\theta$,我们可以考虑将它们表示为两个全连接神经网络,其中每个神经网络包含多个隐藏层。在每个隐藏层中,我们使用SwiGLU作为激活函数,并将当前状态和相邻层输出作为输入。例如,对于$\overrightarrow{\mathbf{S}}_\theta$,其前向状态$\overrightarrow{\mathbf{x}}_l$和上一层的反向输出$\overleftarrow{\mathbf{T}}_{l-1}$可以表示为:

[0077] $$\overrightarrow{\mathbf{x}}_l = [\overrightarrow{\mathbf{x}}_l^1, \overrightarrow{\mathbf{x}}_l^2, \ldots, \overrightarrow{\mathbf{x}}_l^H]$$
$$\overleftarrow{\mathbf{T}}_{l-1} = [\overleftarrow{\mathbf{T}}_{l-1}^1, \overleftarrow{\mathbf{T}}_{l-1}^2, \ldots, \overleftarrow{\mathbf{T}}_{l-1}^H]$$

[0078] 网络表示为:

[0079] $$\overrightarrow{\mathbf{s}}_l = \overrightarrow{\mathbf{S}}_\theta(\overrightarrow{\mathbf{x}}_l, \overleftarrow{\mathbf{T}}_{l-1}) = \overrightarrow{\mathbf{S}}_\theta([\overrightarrow{\mathbf{x}}_l^1, \overrightarrow{\mathbf{x}}_l^2, \ldots, \overrightarrow{\mathbf{x}}_l^H], [\overleftarrow{\mathbf{T}}_{l-1}^1, \overleftarrow{\mathbf{T}}_{l-1}^2, \ldots, \overleftarrow{\mathbf{T}}_{l-1}^H])$$

[0080] 其中,$\overrightarrow{\mathbf{S}}_\theta$是一个由多个全连接层组成的神经网络,每个层都使用SwiGLU作为激活函数。对于每个隐藏层,其输入是当前状态和相邻层输出,其输出是该层的激活值。具体地,假设$\overrightarrow{\mathbf{S}}_\theta$包含L个隐藏层,那么每个隐藏层的数学公式可以表示为:

[0081] $\overrightarrow{\mathbf{a}}^{l} = \mathrm{SwiGLU}(\overrightarrow{\mathbf{W}}^{l}\overrightarrow{\mathbf{x}}_{l} + \overleftarrow{\mathbf{W}}^{l}\overleftarrow{\mathbf{T}}_{l-1} + \overrightarrow{\mathbf{b}}^{l})$

[0082] 其中，$\overrightarrow{\mathbf{W}}^{l}$和$\overleftarrow{\mathbf{W}}^{l}$是该层的全连接权重矩阵，$\overrightarrow{\mathbf{b}}^{l}$和$\overrightarrow{\mathbf{b}}^{l}$是该层的偏置向量。

[0083] 对于反向选择性状态网络$\overleftarrow{\mathbf{S}}_{\theta}$，其数学公式可以表示为：

[0084] $\overleftarrow{\mathbf{s}}_{l} = \overleftarrow{\mathbf{S}}_{\theta}(\overleftarrow{\mathbf{x}}_{l}, \overrightarrow{\mathbf{T}}_{l+1}) = \overleftarrow{\mathbf{S}}_{\theta}([\overleftarrow{\mathbf{x}}_{l}^{1}, \overleftarrow{\mathbf{x}}l^{2}, \ldots, \overleftarrow{\mathbf{x}}_{l}^{H}], [\overrightarrow{\mathbf{T}}_{l+1}^{1}, \overrightarrow{\mathbf{T}}_{l+1}^{2}, \ldots, \overrightarrow{\mathbf{T}}_{l+1}^{H}])$

[0085] 其中，$\overleftarrow{\mathbf{S}}_{\theta}$也是一个由多个全连接层组成的神经网络，每个层都使用SwiGLU作为激活函数。对于每个隐藏层，其输入是当前状态和相邻层输出，其输出是该层的激活值。具体地，$\overleftarrow{\mathbf{S}}_{\theta}$包含L个隐藏层，那么每个隐藏层的数学公式可以表示为：

[0086] $\overleftarrow{\mathbf{a}}^{l} = \mathrm{SwiGLU}(\overleftarrow{\mathbf{W}}^{l}\overleftarrow{\mathbf{x}}_{l} + \overrightarrow{\mathbf{W}}^{l}\overrightarrow{\mathbf{T}}_{l+1} + \overleftarrow{\mathbf{b}}^{l})$

[0087] 其中，$\overleftarrow{\mathbf{W}}^{l}$和$\overrightarrow{\mathbf{W}}^{l}$是该层的全连接权重矩阵，$\overleftarrow{\mathbf{b}}^{l}$和$\overrightarrow{\mathbf{b}}^{l}$是该层的偏置向量。通过这种方式，$\overrightarrow{\mathbf{S}}_{\theta}$和$\overleftarrow{\mathbf{S}}_{\theta}$能够学习到哪些信息对当前输入是相关的，从而选择性地更新状态信息。这种机制使得Vic块能够灵活地适应不同的输入序列，提高了模型在处理长序列任务时的性能。

[0088] 4、动态状态缩放(DSS)模块

[0089] 通过动态状态缩放向量调整不同状态维度对输出的影响。具体公式如下：

[0090] $$\overrightarrow{\mathbf{x}}_{l}' = \overrightarrow{\mathbf{r}}_{l} \odot \overrightarrow{\mathbf{s}}_{l}$$
$$\overleftarrow{\mathbf{x}}_{l}' = \overleftarrow{\mathbf{r}}_{l} \odot \overleftarrow{\mathbf{s}}_{l}$$

[0091] 动态状态缩放通过动态状态缩放向量$\overrightarrow{\mathbf{r}}_{l}$和$\overleftarrow{\mathbf{r}}_{l}$对选择性状态向量$\overrightarrow{\mathbf{s}}_{l}$和$\overleftarrow{\mathbf{s}}_{l}$

[0092] 进行逐元素缩放，从而调整不同状态维度对输出的影响。这种机制使得模型能够更好地适应不同的输入分布和动态变化的视觉序列。具体来说，动态状态缩放向

[0093] $\overrightarrow{\mathbf{x}}_{l}' = \overrightarrow{\mathbf{r}}_{l} \odot \overrightarrow{\mathbf{s}}_{l}, \quad \overleftarrow{\mathbf{x}}_{l}' = \overleftarrow{\mathbf{r}}_{l} \odot \overleftarrow{\mathbf{s}}_{l}$

[0094] 其中，⊙表示逐元素乘积。动态状态缩放向量$\overrightarrow{\mathbf{r}}_{l}$通过Swish函数其数学公式如下：

[0095] $\overrightarrow{\mathbf{r}}_{l} = \mathrm{Swish}(\overrightarrow{\mathbf{Z}}_{l}\overrightarrow{\mathbf{s}}_{l} + \overrightarrow{\mathbf{U}}), \quad \overleftarrow{\mathbf{r}}_{l} = \mathrm{Swish}(\overleftarrow{\mathbf{Z}}_{l}\overleftarrow{\mathbf{s}}_{l} + \overleftarrow{\mathbf{U}})$

[0096] 滑且无界的，可以更好地捕捉输入信号中的重要特征，并通过参数β对信号进行自适应缩放。通过结合选择性状态网络和动态状态缩放，我们可以动态调整信息传递的通路，进一步增强了模型对复杂输入的适应能力。

[0097] 5、双向卷积(Bi-Conv)模块

[0098] 使用可分离的1D卷积核捕获序列中的局部模式。具体公式如下：

[0099] $$\overrightarrow{\mathbf{h}}_{l} = \overrightarrow{\mathbf{W}}_{\mathrm{conv}} * \overrightarrow{\mathbf{x}}_{l}' + \mathbf{b}_{\mathrm{conv}}$$
$$\overleftarrow{\mathbf{h}}_{l} = \overleftarrow{\mathbf{W}}_{\mathrm{conv}} * \overleftarrow{\mathbf{x}}_{l}' + \mathbf{b}_{\mathrm{conv}}$$

[0100] 双向卷积更新通过使用可分离的1D卷积核来捕捉序列中的局部模式。这种机制使得模型能够有效地学习序列数据的时空依赖关系，从而提高模型在处理视觉任务时的性能。具体来说，双向卷积更新分别计算前向卷积输出$\overrightarrow{\mathbf{h}}_{l}$和反向卷积输出$\overleftarrow{\mathbf{h}}_{l}$：

[0101] $$\overrightarrow{\mathbf{h}}_l = \overrightarrow{\mathbf{W}}_{\text{conv}} * \overrightarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$
$$\overleftarrow{\mathbf{h}}_l = \overleftarrow{\mathbf{W}}_{\text{conv}} * \overleftarrow{\mathbf{x}}'_l + \mathbf{b}_{\text{conv}}$$

[0102] 其中，$\overrightarrow{\mathbf{W}}_{\text{conv}}$和$\overleftarrow{\mathbf{W}}_{\text{conv}}$是可分离的1D卷积核，$\overrightarrow{\mathbf{x}}'_l$和$\overleftarrow{\mathbf{x}}'_l$是经过动态缩放的状态向量。可分离的1D卷积核$\overrightarrow{\mathbf{W}}_{\text{conv}}$和$\overleftarrow{\mathbf{W}}_{\text{conv}}$定义如下：

[0103] $$\overrightarrow{\mathbf{W}}_{\text{conv}} = \overrightarrow{\mathbf{W}}^{\text{depth}}_{\text{conv}} \circledast \overrightarrow{\mathbf{W}}^{\text{spatial}}_{\text{conv}}$$
$$\overleftarrow{\mathbf{W}}_{\text{conv}} = \overleftarrow{\mathbf{W}}^{\text{depth}}_{\text{conv}} \circledast \overleftarrow{\mathbf{W}}^{\text{spatial}}_{\text{conv}}$$

[0104] 其中，$\circledast$表示深度卷积和空间卷积的组合操作。具体来说，$\overrightarrow{\mathbf{W}}^{\text{depth}}_{\text{conv}}$和$\overleftarrow{\mathbf{W}}^{\text{depth}}_{\text{conv}}$是深度卷积核，沿着通道维度对输入进行卷积；而$\overrightarrow{\mathbf{W}}^{\text{spatial}}_{\text{conv}}$和$\overleftarrow{\mathbf{W}}^{\text{spatial}}_{\text{conv}}$是空间卷积核，沿着序列维度对输入进行卷积，其中K是卷积核大小。通过使用可分离卷积，我们可以显著降低参数量和计算复杂度，同时仍能有效捕捉局部模式。卷积输出$\overrightarrow{\mathbf{h}}_l$和$\overleftarrow{\mathbf{h}}_l$融合了不同尺度下的局部视觉特征，为下一步的自适应状态空间更新提供了丰富的上下文信息。在实际应用中，卷积核的大小、通道数和步长等参数可以通过实验调整，以获得最佳的模型性能。通过这种方式，模型能够有效地处理视觉任务中的时空依赖关系，从而提高模型的性能和泛化能力。

[0105] 6、自适应控制状态空间模型(A-SSM)

[0106] 学习状态转移矩阵和输入映射矩阵，以自适应地更新前后状态。具体步骤如下：

[0107] 1)通过学习得到状态转移矩阵和输入映射矩阵：

[0108] $$\overrightarrow{\mathbf{A}}_\theta \leftarrow \overrightarrow{\mathbf{A}}_\theta(\overrightarrow{\mathbf{h}}_l), \quad \overrightarrow{\mathbf{B}}_\theta \leftarrow \overrightarrow{\mathbf{B}}_\theta(\overrightarrow{\mathbf{h}}_l)$$
$$\overleftarrow{\mathbf{A}}_\theta \leftarrow \overleftarrow{\mathbf{A}}_\theta(\overleftarrow{\mathbf{h}}_l), \quad \overleftarrow{\mathbf{B}}_\theta \leftarrow \overleftarrow{\mathbf{B}}_\theta(\overleftarrow{\mathbf{h}}_l)$$

[0109] 其中，A、B均为状态空间的参数矩阵；A为学习状态转移矩阵，B为输入映射矩阵；$\overrightarrow{\mathbf{A}}_\theta$、$\overleftarrow{\mathbf{A}}_\theta$、$\overrightarrow{\mathbf{B}}_\theta$和$\overleftarrow{\mathbf{B}}_\theta$是可学习的函数，用于将卷积输出$\overrightarrow{\mathbf{h}}_l$和$\overleftarrow{\mathbf{h}}_l$映射到对应的状态转移矩阵和输入映射矩阵。这些可学习的函数通常由多层神经网络组成，通过模型训练学习得到。

[0110] 2)通过MLP计算得到动态调整项：

[0111] $$\Delta\overrightarrow{\mathbf{A}}_l = \text{MLP}(\overrightarrow{\mathbf{h}}_l)$$

[0112] 其中，MLP表示多层感知器；

[0113] 计算动态调节项进一步表示为：

[0114] $$\Delta\overrightarrow{\mathbf{A}}_l = \text{MLP}(\overrightarrow{\mathbf{h}}_l)$$
$$\Delta\overleftarrow{\mathbf{A}}_l = \text{MLP}(\overleftarrow{\mathbf{h}}_l)$$

[0115] 其中，$\Delta\overrightarrow{\mathbf{A}}_l$和$\Delta\overleftarrow{\mathbf{A}}_l$是动态调节项，根据当前输入自适应调节状态转移矩阵，赋予模型更强的适应性。动态调节项$\Delta\overrightarrow{\mathbf{A}}_l$和$\Delta\overleftarrow{\mathbf{A}}_l$是通过另一个可学习网络(MLP)，根据$\overrightarrow{\mathbf{h}}_l$和$\overleftarrow{\mathbf{h}}_l$计算得到的。

[0116] 3)更新前状态和后状态：

[0117] $$\overrightarrow{\mathbf{h}}_{l+1} = (\overrightarrow{\mathbf{A}}_l + \Delta\overrightarrow{\mathbf{A}}_l)\overrightarrow{\mathbf{h}}_l + (\overrightarrow{\mathbf{B}}_l)\mathbf{u}_l$$

[0118] $\overrightarrow{\mathbf{A}}_l$是当前层的状态转移矩阵，$\overrightarrow{\mathbf{B}}_l$是当前层的输入映射矩阵，$\Delta\overrightarrow{\mathbf{A}}_l$是当前层的动态

调整项，$u_1$是当前层的控制输入。

[0119] 其中，$\vec{\mathbf{h}}_l$是前状态，$\vec{\mathbf{h}}_{l+1}$是后状态。

[0120] 更新前向状态和反向状态进一步表示为：

[0121] $$\vec{\mathbf{h}}_{l+1} = (\vec{\mathbf{A}}_l + \Delta\vec{\mathbf{A}}_l)\vec{\mathbf{h}}'_l + \vec{\mathbf{B}}_l$$
$$\overleftarrow{\mathbf{h}}_{l-1} = (\overleftarrow{\mathbf{A}}_l + \Delta\overleftarrow{\mathbf{A}}_l)\overleftarrow{\mathbf{h}}'_l + \overleftarrow{\mathbf{B}}_l$$

[0122] 其中，$\vec{\mathbf{x}}'_l$和$\overleftarrow{\mathbf{x}}'_l$是前一时刻的前向和反向状态，通过更新后的状态转移矩阵和输入映射矩阵，得到新的前向状态和反向状态$\vec{\mathbf{x}}_{l+1}$和$\overleftarrow{\mathbf{x}}_{l-1}$，其中$u_1$表示当前层的控制输入。这个控制输入是根据当前层的输出$T_1$和期望输出$d_1$之间的偏差$e_1$计算得到的，其目的是引导模型输出逼近期望目标(即$e_1$最小)。

[0123] 自适应状态空间更新通过学习状态转移矩阵和输入映射矩阵，对前向状态和反向状态进行自适应更新。这种机制使得模型能够根据输入数据的动态变化，自适应地调整状态转移策略，从而提高模型在处理复杂视觉序列时的性能。

[0124] 通过上述这种自适应状态空间更新机制，模型可以学习到最优的状态转移模式，有效捕捉输入序列的内在动态规律。同时，动态调节项让模型能够灵活调整状态转移策略以适应不同输入，从而提高模型的鲁棒性和泛化能力。

[0125] 7、自适应控制机制(ACM)

[0126] 通过闭环反馈机制和自适应控制律动态调整模型的内部参数。具体步骤如下：

[0127] 1)计算偏差：

[0128] $e_1 = d_1 T_1$

[0129] 2)更新控制增益矩阵：

[0130] $$\mathbf{K}_{l+1} = \mathbf{K}_l - \alpha\frac{\partial \mathbf{e}_l}{\partial \mathbf{K}_l}$$

[0131] 3)计算控制输入：

[0132] $u_1 = -K_1 e_1 + u_d$、

[0133] 反馈及自适应控制通过引入闭环反馈机制和自适应控制律，实现了对模型性能的进一步提升和自适应调节。这种机制使得模型能够根据输出误差动态调整内部参数，提高适应能力，并能够有意识地优化输出，实现主动式学习。具体来说，反馈及自适应控制的步骤如下：

[0134] 计算当前层输出与期望输出之间的偏差$e_1$：

[0135] $e_1 = d_1 - T_1$

[0136] 其中，$d_1$是期望输出，$T_1$是当前层的输出。偏差$e_1$反映了实际输出与期望输出之间的差异。

[0137] 根据偏差更新控制增益矩阵$K_1$：

[0138] $$\mathbf{K}_{l+1} = \mathbf{K}_l - \alpha\frac{\partial \mathbf{e}_l}{\partial \mathbf{K}_l}$$

[0139] 其中，$\alpha$是学习率，$\frac{\partial \mathbf{e}_l}{\partial \mathbf{K}_l}$是误差$e_1$关于控制增益矩阵$K_1$的梯度。通过这个梯度信息，模型能够学习到如何调整控制增益矩阵，以减小输出误差。

[0140] 计算控制输入$u_1$,表示为:

[0141] $u_1 = -K_1 e_1 + u_d$

[0142] 其中,$u_d$是期望输入。控制输入$u_1$是通过更新后的控制增益矩阵$K_1$和偏差$e_1$计算得到的,它反映了模型应该如何调整输出以逼近期望目标。

[0143] 将控制输入$u_1$反馈到公式(22)和公式(23)中,实现闭环控制:通过将控制输入$u_1$反馈到模型中,模型能够根据输出误差动态调整内部参数,提高适应能力。同时,期望输入$u_d$也为模型提供了目标信息,使其有意识地优化输出,实现主动式学习。

[0144] 通过这种反馈及自适应控制机制,模型能够根据输出误差动态调整内部参数。提高适应能力。并能够有意识地优化输出,实现主动式学习。

[0145] 8、状态融合与输出预测

[0146] 状态融合与输出预测通过融合前向状态$\overrightarrow{x}_{l+1}$和反向状态$\overleftarrow{x}_{l-1}$来生成当前层的输出$T_1$。这一步骤不仅融合了两个方向的信息,还通过SwiGLU门控函数对前向和反向信息进行动态控制,以平衡两个方向的上下文信息。具体来说,状态融合与输出预测的步骤如下:分别计算前向输出$\overrightarrow{y}_l$和反向输出$\overleftarrow{y}_l$:

[0147]
$$\overrightarrow{y}_l = \overrightarrow{W}_y \overrightarrow{x}_{l+1} + \overrightarrow{b}_y$$
$$\overleftarrow{y}_l = \overleftarrow{W}_y \overleftarrow{x}_{l-1} + \overleftarrow{b}_y$$

[0148] 其中,$\overrightarrow{W}_y$和$\overleftarrow{W}_y$是输出映射矩阵,$\overrightarrow{b}_y$和$\overleftarrow{b}_y$是偏置向量。这些矩阵和向量是通过模型训练学习得到的。

[0149] 使用SwiGLU门控函数对前向和反向输出进行融合,生成当前层的输出$T_1$:

[0150] $$T_l = \mathrm{SwiGLU}(\overrightarrow{W}_g \overrightarrow{T}_l + \overrightarrow{b}_g) \odot \overrightarrow{y}_l + \mathrm{SwiGLU}(\overleftarrow{W}_g \overleftarrow{T}_l + \overleftarrow{b}_g) \odot \overleftarrow{y}_l$$

[0151] 其中,$\overrightarrow{T}_l$和$\overleftarrow{T}_l$分别是当前层的前向和反向输入,$\overrightarrow{W}_g$和$\overrightarrow{b}_g$是前向传播的门控参数,$\overleftarrow{W}_g$和$\overleftarrow{b}_g$是反向传播的门控参数。通过这种方式,SwiGLU门控函数可以自适应地控制前向和反向信息在输出中的贡献大小,从而平衡两个方向的上下文信息。

[0152] 通过这种状态融合与输出预测机制,Vision Conba模型能够有效地捕捉前向和反向传播中的时间依赖关系,并动态地调整两个方向信息的贡献,以生成更加准确和全面的输出。

[0153] 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。

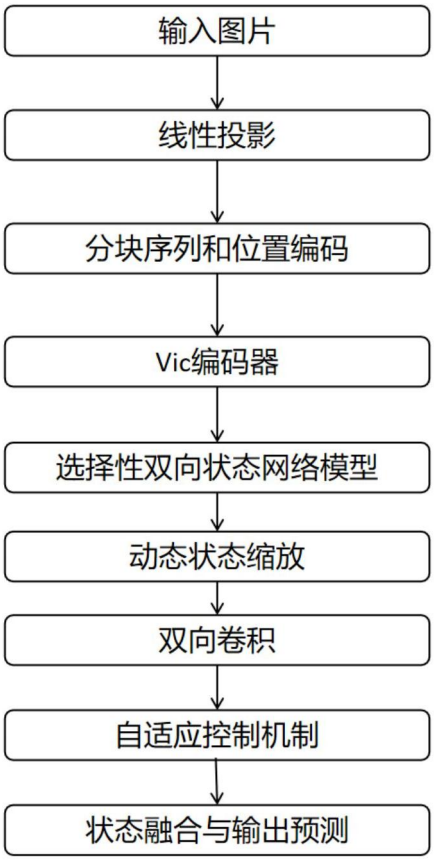

输入图片
线性投影
分块序列和位置编码
Vic编码器
选择性双向状态网络模型
动态状态缩放
双向卷积
自适应控制机制
状态融合与输出预测

图1