(19)国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 114841342 A
(43)申请公布日 2022.08.02

(21)申请号 202210556441.4

(22)申请日 2022.05.19

(71)申请人 湖北楚天高速数字科技有限公司
地址 430000 湖北省武汉市汉阳区四新大道26号湖北国展中心广场B4地块东塔栋23层(1)办号2303室

(72)发明人 朱晨露 刘德彬 阮一恒 张立杰 邓贤君 杨天若

(74)专利代理机构 深圳市科冠知识产权代理有限公司 44355
专利代理师 王丽坤

(51)Int.Cl.
G06N 3/08(2006.01)
G06K 9/62(2022.01)

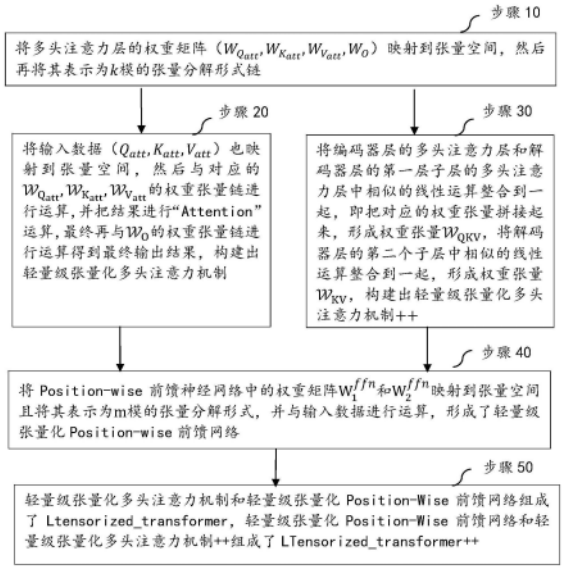
权利要求书4页 说明书9页 附图1页

(54)发明名称

一种基于张量的高效Transformer的架构方法

(57)摘要

本发明适用人工智能领域,提供了一种基于张量的高效Transformer的架构方法,本发明通过权重矩阵拼接和张量多模态分解方法设计了三个即插即用的轻量级Transformer组件,通过利用三个轻量级的Transformer组件,构建了LTensorized_transformer和LTensorized_transformer++,从而使得轻量级的Transformer模型能够部署到资源受限的工业嵌入式设备上。



CN 114841342 A

1.一种基于张量的高效Transformer的架构方法，其特征在于，所述方法包括：

步骤10、将多头注意力层的权重矩阵$(W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}, W_O)$映射到张量空间，然后再将其表示为k模的张量分解形式链；

步骤20、将输入数据（Q_{att}，K_{att}，V_{att}）映射到张量空间，然后与对应的$\mathcal{W}_{Q_{att}}, \mathcal{W}_{K_{att}}, \mathcal{W}_{V_{att}}$的权重张量链进行运算，并把结果进行“Attention”运算，再与$\mathcal{W}_O$的权重张量链进行运算得到最终输出结果，构建出轻量级张量化多头注意力机制；

步骤30、将编码器层的多头注意力层和解码器层的第一层子层的多头注意力层中相似的线性运算整合到一起，即把对应的权重张量拼接起来，形成权重张量$\mathcal{W}_{QKV}$，将解码器层的第二个子层中相似的线性运算整合到一起，形成权重张量$\mathcal{W}_{KV}$，构建出轻量级张量化多头注意力机制++；

步骤40、将Position-wise前馈神经网络中的权重矩阵W_1^{ffn}和W_2^{ffn}映射到张量空间且将其表示为m模的张量分解形式，并与输入数据进行运算，构建出轻量级张量化Position-wise前馈网络；

步骤50、将轻量级张量化多头注意力机制和轻量级张量化Position-Wise前馈网络组成Ltensorized_transformer，将轻量级张量化Position-Wise前馈网络和轻量级张量化多头注意力机制++组成LTensorized_transformer++，构建出轻量级Transformer架构。

2.根据权利要求1所述的基于张量的高效Transformer的架构方法，其特征在于，所述步骤10包括以下具体步骤：

将queries，keys和values分别打包成矩阵Q_{att}，K_{att}和V_{att}，并对矩阵Q_{att}，K_{att}和V_{att}进行了h次线性投影，其中涉及到的权重矩阵可以统一表示为$W_{Q_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$，$W_{K_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$，$W_{V_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$和$W_O \in \mathbb{R}^{d_{model} \times d_{model}}$，将$d_{model}$表示成多个正数因子的乘积，$d_{model} = \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}$；

将权重矩阵$W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}$和$W_0$映射到张量空间就得到权重张量$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$（$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$，$\mathcal{W}_O \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}}$）；

根据公式(1)将权重张量$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$表示为k模的张量分解形式，公式(1)的定义如下：

$$Tensorized\left(\mathcal{W}'\right) = (\mathcal{W}_1', \mathcal{W}_2', \ldots, \mathcal{W}_{2m}')$$

$$\rightarrow \mathcal{W}_1' \ *_{r_1^{(1)},\ldots,r_k^{(1)}} \mathcal{W}_2' \ *_{r_1^{(2)},\ldots,r_k^{(2)}} \cdots *_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}} \mathcal{W}_{2m}' \qquad (1)$$

3.根据权利要求2所述的基于张量的高效Transformer的架构方法，其特征在于，所述步骤20包括以下具体步骤：

步骤21、通过把输入数据映射到张量空间$(\mathcal{Q}_{att},\mathcal{K}_{att},\mathcal{V}_{att})$，并与对应的小型权重张量核进行运算，运算的过程如公式(2)所示，所述公式(2)的定义如下：

$$Lightweight\_Connect(\mathcal{X},Tensorized(\mathcal{W}))$$
$$=\mathcal{X}*_{d_1,\ldots,d_k}\mathcal{W}_1*_{r_1^{(1)},\ldots,r_k^{(1)},d_{k+1},\ldots,d_{2k}}\mathcal{W}_2*\ldots*_{r_1^{(m-1)},\ldots,r_k^{(m-1)},d_{(m-1)k+1},\ldots,d_{mk}}$$
$$\mathcal{W}_m*_{r_1^{(m)},\ldots,r_k^{(m)}}\mathcal{W}_{m+1}*_{r_1^{(m+1)},\ldots,r_k^{(m+1)}}\mathcal{W}_{m+2}*\ldots*_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}}\mathcal{W}_{2m} \quad (2)$$

步骤22、将运算结果进行reshape操作，并将其拆分成h等分，用列表L存储拆分的结果，整个计算过程如下：

$D'=\mathrm{Reshape}(D,[-1,d_{model}])$ (3)

$T=\mathrm{Spht}(D')=(D'_1,\ldots,D'_h)$ (4)

步骤23、将列表L中的存储数据取出来，$\mathcal{Q}',\mathcal{K}',\mathcal{V}'\leftarrow L$，$\mathcal{Q}'$、$\mathcal{K}'$和$\mathcal{V}'$中各有由h个矩阵组成，利用公式(5)来获取对应下标的输入矩阵(Q_{att}^i，K_{att}^i和V_{att}^i)，并进行注意力计算，从而获得对应的注意力输出，所述公式(5)的定义如下：

$R_i=\mathrm{Get}(R,i)$ (5)

步骤24、利用公式(6)计算每个注意力的输出结果($head_i$)，并将每个注意力的结果拼接在一起，与小型的权重张量核($\mathcal{W}_O$的k模张量分解形式的结果)利用公式(2)进行多步特征计算得到最终的多头注意力层的输出结果，所述公式(6)的定义如下：

$$head_i=Attention\left(Q_{att}^i,K_{att}^i,V_{att}^i\right)=softmax\frac{(Q_{att}^i{K_{att}^i}^T)}{\sqrt{d}}V_{att}^i \quad (6)$$

4.根据权利要求3所述的基于张量的高效Transformer的架构方法，其特征在于，所述步骤30包括以下具体步骤：

步骤31、编码器层的多头注意力层和解码器层的第一层子层多头注意力层的结构性质相同，查询矩阵Q_{att}、键矩阵K_{att}和值矩阵V_{att}做了相似的线性映射操作，将权重矩阵$W_{Q_{att}}$，$W_{K_{att}}$和$W_{V_{att}}$连接成一个巨大的权重矩阵$W_{QKV}\in\mathbb{R}^{d_{model}\times 3d_{model}}$，如公式(7)所示，然后再利用公式(1)将权重矩阵W_{QKV}映射到张量空间并将得到的权重张量$\mathcal{W}_{QKV}\in\mathbb{R}^{\{d_1\times\ldots\times d_k\}\times\{d_{k+1}\times\ldots\times d_{2k}\}\times\ldots\times\{d_{(m-1)k+1}\times\ldots\times d_{mk}\}\times\{d_1\times\ldots\times d_k\}\times\{d_{k+1}\times\ldots\times d_{2k}\}\times\ldots\times\{d_{(m-1)k+1}\times\ldots\times 3d_{mk}\}}$表示为k模的张量分解形式，所述公式(7)的定义如下：

$$W_{QKV}=Concat\left(W_{Q_{att}},W_{K_{att}},W_{V_{att}}\right) \quad (7)$$

步骤32、对输入数据M进行reshape操作，然后利用公式(2)计算M与小型权重张量链($\mathcal{W}_{QKV}$)的结果，结果记为$\mathcal{A}$，对$\mathcal{A}$进行reshape操作，然后进行切片，获得对应的$\mathcal{Q}_{\mathrm{in}}$，$\mathcal{K}_{in}$和$\mathcal{V}_{in}$，具体过程如下：

$$\mathcal{M}=reshape\left(M,\left[-1,d_1,\ldots,d_k,d_{k+1},\ldots,d_{2k},\ldots,d_{(m-1)k+1},\ldots,d_{mk}\right]\right) \quad (8)$$

$$\mathcal{A}=ightweight\_Connect\left(\mathcal{M},Tensorized\left(\mathcal{W}_{QKV}\right)\right) \quad (9)$$

$$\mathcal{A} = reshape(\mathcal{A}, [-1, d_{model} \times 3]) \quad (10)$$

$$\mathcal{Q}_{in} = \mathcal{A}[:, 0: d_{model}] \quad (11)$$

$$\mathcal{K}_{in} = \mathcal{A}[:, d_{model}: 2 \times d_{model}] \quad (12)$$

$$\mathcal{V}_{in} = \mathcal{A}[:, 2 \times d_{model}:] \quad (13)$$

对$\mathcal{Q}_{in}$，$\mathcal{K}_{in}$和$\mathcal{V}_{\text{in}}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出；

步骤33、解码器层的第二个子层多头注意力层中的键矩阵和值矩阵的线性投影过程相似，因此将权重矩阵$W_{K_{att}}$和$W_{V_{att}}$连接起来形成权重矩阵$W_{KV} \in \mathbb{R}^{d_{model} \times 2d_{model}}$，如公式（14）所示，同样也将权重矩阵$W_{KV}$映射到张量空间并将得到的权重张量$\mathcal{W}_{KV} \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times 2d_{mk}\}}$表示为k模的张量分解形式，所述公式(14)的定义如下：

$$W_{KV} = Concat(W_{K_{att}}, W_{V_{att}}) \quad (14)$$

步骤34、对输入数据N进行reshape操作，然后利用公式(2)计算N与小型权重张量链($\mathcal{W}_{KV}$)的结果，结果记为$\mathcal{B}$，对$\mathcal{B}$进行reshape操作，然后进行切片，获得对应的$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$，具体过程如下：

$$\mathcal{N} = reshape\left(N, \left[-1, d_1, \ldots, d_k, d_{k+1}, \ldots, d_{2k}, \ldots, d_{(m-1)k+1}, \ldots, d_{mk}\right]\right) \quad (15)$$

$$\mathcal{B} = Lightweight\_Connect\left(\mathcal{N}, Tensorized(\mathcal{W}_{KV})\right) \quad (16)$$

$$\mathcal{B} = reshape(\mathcal{B}, [-1, d_{model} \times 2]) \quad (17)$$

$$\mathcal{K}_{in}^{'} = \mathcal{B}[:, 0: d_{model}] \quad (18)$$

$$\mathcal{V}_{in}^{'} = \mathcal{B}[:, d_{model}:] \quad (19)$$

其中，$\mathcal{Q}_{\text{in}}^{'}$的计算流程与步骤21中$\mathcal{Q}$的计算流程一致，同样要对$\mathcal{Q}_{in}^{'}$，$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出。

5.根据权利要求4所述的基于张量的高效Transformer的架构方法，其特征在于，所述步骤40包括以下具体步骤：

步骤41、将d_{model}和d_{ff}转换为数值较小的正整数因子乘积，$d_{model}=\{O_1 \times \ldots \times O_m\} \times \{O_{m+1} \times \ldots \times O_{2m}\} \times \ldots \times \{O_{(n-1)m+1} \times \ldots \times O_{nm}\}$和$d_{ff}=\{P_1 \times \ldots \times P_m\} \times \{P_{m+1} \times \ldots \times P_{2m}\} \times \ldots \times \{P_{(n-1)m+1} \times \ldots \times P_{nm}\}$，权重矩阵$W_1^{ffn} \in \mathbb{R}^{d_{model} \times d_{ff}}$和$W_2^{ffn} \in \mathbb{R}^{d_{ff} \times d_{model}}$变为权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$，其中

$\mathcal{W}_1^{ffn} \in \mathbb{R}^{\{O_1 \times \ldots \times O_m\} \times \{O_{m+1} \times \ldots \times O_{2m}\} \times \ldots \times \{O_{(n-1)m+1} \times \ldots \times O_{nm}\} \times \{P_1 \times \ldots \times P_m\} \times \{P_{m+1} \times \ldots \times P_{2m}\} \times \ldots \times \{P_{(n-1)m+1} \times \ldots \times P_{nm}\}}$和

$\mathcal{W}_2^{\text{ffn}} \in \mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}\times\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$，偏移向量$b_1^{ffn}$和$b_2^{ffn}$变成偏移张量$\mathcal{B}_1^{ffn} \in \mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}}$和$\mathcal{B}_2^{ffn} \in \mathbb{R}^{\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$；

步骤42、利用公式(1)将权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$表示为m模的张量分解形式，以此降低网络模型中的训练参数数量和计算复杂度；

步骤43、将Position-wise前馈网络的输入数据映射到张量空间，并与小型的权重张量链进行多步计算，其计算流程如公式(20)和(21)所示：

$$First = max\,(0, Lightweight\_Connect(\mathcal{X}, Tensorized(\mathcal{W}_1^{ffn}))) + \mathcal{B}_1^{ffn} \quad (20)$$

$$Lightweight\_Feed\_Forward\_Network = Lightweight\_Connect(First, Tensorized(\mathcal{W}_2^{ffn})) + \mathcal{B}_2^{ffn} \quad (21)$$

6.根据权利要求1所述的基于张量的高效Transformer的架构方法，其特征在于，所述步骤3和步骤4的顺序不分先后。

一种基于张量的高效Transformer的架构方法

技术领域

[0001] 本发明属于人工智能领域，尤其涉及一种基于张量的高效Transformer的架构方法。

背景技术

[0002] 随着人工智能技术、通信技术和电子芯片技术的快速发展，工业物联网将绿色基础设施、智慧城市、智慧医疗、智能电网和智能交通系统中的智能移动设备、可穿戴设备、传感器等数百亿终端设备紧密的连接了起来，提供了更为智能的服务，使得人们的生产生活变得更加便捷。然而，工业物联网会产生体量庞大的多源异构数据，其次，由于深度神经网络模型表现出的优异性能，因此，深度神经网络模型被广泛的应用于特征提取和智能决策。然而，由于任务的复杂性和庞大的数据量，通常需要采用大型的深度神经网络模型对数据进行训练，从而获得优异的任务性能。

[0003] Transformer作为目前较为受欢迎的深度神经网络模型，已被广泛的应用于工业物联网。Transformer完全抛弃了卷积模块和循环模块的设计理念，该网络模型仅由注意力层和全连接层构成。Transformer以及衍生出来的其他模型，例如BERT、VIT、GPT和Universal Transformer等在自然语言处理、计算机视觉、推荐系统、智能交通等方面取得了优异的任务性能。然而，这些网络模型的规模都十分庞大，通常包含了数亿个训练参数，因此需要高性能的芯片花费几周甚至几个月来训练这些网络模型。所以，在网络模型的训练过程中会消耗大量的计算资源和能源资源。此外，随着联邦学习范式的出现和实时性应用需求的增加，智能移动设备和其他嵌入式设备需要参与到智能任务的训练和决策。然而，智能移动设备和其他嵌入式设备的芯片性能通常较差且高性能芯片体型庞大，无法将其安装在边缘设备上。因此，现有的规模庞大的深度神经网络模型无法在智能移动设备和其他嵌入式设备上进行训练和部署。此外，在联邦学习范式中，各个客户端需要上传网络模型数据到云端，然后在云端完成模型融合过程。然而，由于网络模型规模庞大且在上传过程中为了保护网络模型的数据安全，需要利用加密方法对模型数据进行加密。对网络模型进行加密可以有效防止隐私泄露，但是进一步增加了通信数据量。为了减少联邦学习过程中的通信数据量，降低带宽占用和提高通信效率，如何降低网络模型中的训练参数数量则变得尤为重要。此外，在交通标识识别和故障检测等一些实时性应用中，将数据上传到云端过程中容易遭受到网络攻击，从而导致隐私泄露和安全事故。而且，将数据上传到云端、云端处理数据和云端下发结果到终端需要消耗一定的时间，因而，会造成时延增大，从而不满足实时性任务的需求。综上所述，将计算任务卸载到云端不是最好的选择。此外，规模庞大的网络模型在训练过程中会消耗大量的能源资源，增大了碳排放量，加剧了全球环境的恶化。因此，如何在保证网络模型性能不变的前提下或精度损失在可容忍范围内，大规模降低深度神经网络模型中的训练参数数量和浮点运算操作，加快网络模型的训练过程，降低训练过程中的能源消耗，以便部署到资源受限的边缘设备上是一个亟需解决的问题。幸运的是，规模庞大的网络模型中通常存在大量的冗余学习参数和浮点操作，因此可以将网络模型中的

冗余部分剔除，从而降低模型的学习参数数量和计算复杂度，加快模型的训练过程，减少电力资源的消耗，推动绿色经济的发展。

[0004] 由于Transformer在各个领域中都表现出了优异的性能，但是其网络模型的复杂性限制了其在智能移动设备和其他嵌入式设备上的训练和部署。因此，越来越多的研究人员开始研究如何降低模型的复杂度。直接根据Transformer模型结构特性进行优化，减少Transformer中Block的数量，虽然可以降低模型的大小和复杂性，但是模型的性能会大幅度降低。因此，如何在保证模型性能不变的前提下降低网络模型的参数数量和计算复杂度是一个研究热点。目前用于模型压缩的方法主要有模型剪枝、低阶近似、模型量化和知识蒸馏。Liu等人对训练后的vision Transformer模型进行量化，降低了模型的内存存储和计算成本。Chung等人提出了一种混合精度量化策略，通过较少的比特数来表示Transformer的权值，以此来降低模型的内存占用，提高模型的推理速度。Zhu等人通过对Vision Transformer进行剪枝来降低模型的参数数量。Mao等人通过分析Transformer组件的属性对其进行剪枝，从而降低模型的参数数量，降低模型的推理时间。Jiao等人提出了一种新的Transformer蒸馏方法，通过利用这种新的知识蒸馏方法，将复杂的教师模型的知识转移到小的学生模型中。张量分解方法作为一种新兴高效的网络模型压缩方法，已经在其他网络模型压缩中取得了优异的性能。Hrinchunk等人利用张量链分解方法对嵌入层的参数进行压缩，从而降低模型的复杂度。Ma等人基于张量分解和参数共享的思想，提出了一种基于BT分解的自注意模型，降低了Transformer模型的参数数量。上述方法虽然可以高效降低模型的学习参数数量和计算复杂度，但是仍然存在缺陷。模型量化方法虽然可以大幅度降低模型的内存占用，但是模型的性能通常会有较大的损失。模型剪枝和知识蒸馏方法虽然可以降低模型的复杂度，但是过程通常过于繁琐，且对于新的模型需要重新进行压缩方案的设计。因此，这两种方法的复用性较差。而Hrinchunk仅对嵌入层进行了压缩，对于其他不含嵌入层的模型则没有压缩效果。Ma等人虽然降低了Transformer的学习参数数量，但是破坏了注意力机制的特性。

发明内容

[0005] 本发明的目的在于提供一种基于张量的高效Transformer的架构方法，旨在解决背景技术中提到的针对当前Transformer模型无法部署到智能移动设备和其他嵌入式设备进行训练和推理的问题问题。

[0006] 本发明提供了一种基于张量的高效Transformer的架构方法，所述方法包括以下步骤：

[0007] 步骤10、将多头注意力层的权重矩阵$(W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}, W_O)$映射到张量空间，然后再将其表示为k模的张量分解形式链；

[0008] 步骤20、将输入数据(Q_{att}，K_{att}，V_{att})映射到张量空间，然后与对应的$\mathcal{W}_{Q_{att}}, \mathcal{W}_{K_{att}}, \mathcal{W}_{V_{att}}$的权重张量链进行运算，并把结果进行“Attention”运算，再与$\mathcal{W}_O$的权重张量链进行运算得到最终输出结果，构建出轻量级张量化多头注意力机制；

[0009] 步骤30、将编码器层的多头注意力层和解码器层的第一层子层的多头注意力层中相似的线性运算整合到一起，即把对应的权重张量拼接起来，形成权重张量$\mathcal{W}_{QKV}$，将解码

器层的第二个子层中相似的线性运算整合到一起，形成权重张量$\mathcal{W}_{KV}$，构建出轻量级张量化多头注意力机制++；

[0010] 步骤40、将Position-wise前馈神经网络中的权重矩阵W_1^{ffn}和W_2^{ffn}映射到张量空间且将其表示为m模的张量分解形式，并与输入数据进行运算，构建出轻量级张量化Position-wise前馈网络；

[0011] 步骤50、将轻量级张量化多头注意力机制和轻量级张量化Position-Wise前馈网络组成Ltensorized_transformer，将轻量级张量化Position-Wise前馈网络和轻量级张量化多头注意力机制++组成LTensorized_transformer++，构建出轻量级Transrormer架构。

[0012] 进一步地，所述步骤10包括以下具体步骤：

[0013] 将queries，keys和values分别打包成矩阵Q_{att}，K_{att}和V_{att}，并对矩阵Q_{att}，K_{att}和V_{att}进行了h次线性投影，其中涉及到的权重矩阵可以统一表示为$W_{Q_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$，$W_{K_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$，$W_{V_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$和$W_O \in \mathbb{R}^{d_{model} \times d_{model}}$，将$d_{model}$表示成多个正数因子的乘积，$d_{model}=\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}$；

[0014] 将权重矩阵$W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}$和$W_0$映射到张量空间就得到权重张量$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$（$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$，$\mathcal{W}_O \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}}$）；

[0015] 根据公式(1)将权重张量$\mathcal{W}_{Q_{att}}$，$\mathcal{W}_{K_{att}}$，$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$表示为k模的张量分解形式，公式(1)的定义如下：

[0016]
$$Tensorized\left(\mathcal{W}'\right) = (\mathcal{W}_1', \mathcal{W}_2', \ldots, \mathcal{W}_{2m}')$$
$$\rightarrow \mathcal{W}_1' *_{r_1^{(1)},\ldots,r_k^{(1)}} \mathcal{W}_2' *_{r_1^{(2)},\ldots,r_k^{(2)}} \cdots *_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}} \mathcal{W}_{2m}' \quad (1)$$

[0017] 进一步地，所述步骤20包括以下具体步骤：

[0018] 步骤21、通过把输入数据映射到张量空间$(\mathcal{Q}_{att}, \mathcal{K}_{att}, \mathcal{V}_{att})$，并与对应的小型权重张量核进行运算，运算的过程如公式(2)所示，所述公式(2)的定义如下：

[0019]
$$Lightweight\_Connect(\mathcal{X}, Tensorized(\mathcal{W}))$$
$$= \mathcal{X} *_{d_1,\ldots,d_k} \mathcal{W}_1 *_{r_1^{(1)},\ldots,r_k^{(1)},d_{k+1},\ldots,d_{2k}} \mathcal{W}_2 * \cdots *_{r_1^{(m-1)},\ldots,r_k^{(m-1)},d_{(m-1)k+1},\ldots,d_{mk}}$$
$$\mathcal{W}_m *_{r_1^{(m)},\ldots,r_k^{(m)}} \mathcal{W}_{m+1} *_{r_1^{(m+1)},\ldots,r_k^{(m+1)}} \mathcal{W}_{m+2} * \cdots *_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}} \mathcal{W}_{2m} \quad (2)$$

[0020] 步骤22、将运算结果进行reshape操作，并将其拆分成h等分，用列表L存储拆分的结果，整个计算过程如下：

[0021] $D' = \text{Reshape}(D, [-1, d_{model}])$ (3)

[0022] $T = \text{Split}(D') = (D'_1, \ldots, D'_h)$ (4)

[0023] 步骤23、将列表L中的存储数据取出来，$\mathcal{Q}', \mathcal{K}', \mathcal{V}' \leftarrow L$，$\mathcal{Q}'$、$\mathcal{K}'$和$\mathcal{V}'$中各有由h个矩阵组成，利用公式(5)来获取对应下标的输入矩阵(Q_{att}^i, K_{att}^i和V_{att}^i)，并进行注

意力计算，从而获得对应的注意力输出，所述公式(5)的定义如下：

[0024] $R_i = Get(R, i)$ (5)

[0025] 步骤24、利用公式(6)计算每个注意力的输出结果($head_i$)，并将每个注意力的结果拼接在一起，与小型的权重张量核($\mathcal{W}_O$的k模张量分解形式的结果)利用公式(2)进行多步特征计算得到最终的多头注意力层的输出结果，所述公式(6)的定义如下：

[0026] $$head_i = Attention\left(Q_{att}^i, K_{att}^i, V_{att}^i\right) = softmax\frac{\left(Q_{att}^i {K_{att}^i}^T\right)}{\sqrt{d}} V_{att}^i \quad (6)$$

[0027] 进一步地，所述步骤30包括以下具体步骤：

[0028] 步骤31、编码器层的多头注意力层和解码器层的第一层子层多头注意力层的结构性质相同，查询矩阵Q_{att}、键矩阵K_{att}和值矩阵V_{att}做了相似的线性映射操作，将权重矩阵$W_{Q_{att}}$，$W_{K_{att}}$和$W_{V_{att}}$连接成一个巨大的权重矩阵$W_{QKV} \in_{\mathbb{R}}^{d_{model} \times 3d_{model}}$，如公式(7)所示，然后再利用公式(1)将权重矩阵W_{QKV}映射到张量空间并将得到的权重张量$\mathcal{W}_{QKV} \in$

$$\mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times 3d_{mk}\}}$$

[0029] 表示为k模的张量分解形式，所述公式(7)的定义如下：

[0030] $$W_{QKV} = Concat\left(W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}\right) \quad (7)$$

[0031] 步骤32、对输入数据M进行reshape操作，然后利用公式(2)计算M与小型权重张量链($\mathcal{W}_{QKV}$)的结果，结果记为$\mathcal{A}$，对$\mathcal{A}$进行reshape操作，然后进行切片，获得对应的$\mathcal{Q}_{\text{in}}$，$\mathcal{K}_{in}$和$\mathcal{V}_{in}$，具体过程如下：

[0032] $$\mathcal{M} = reshape\left(M, \left[-1, d_1, \ldots, d_k, d_{k+1}, \ldots, d_{2k}, \ldots, d_{(m-1)k+1}, \ldots, d_{mk}\right]\right) \quad (8)$$

[0033] $$\mathcal{A} = ightweight\_Connect\left(\mathcal{M}, Tensorized\left(\mathcal{W}_{QKV}\right)\right) \quad (9)$$

[0034] $$\mathcal{A} = reshape\left(\mathcal{A}, \left[-1, d_{model} \times 3\right]\right) \quad (10)$$

[0035] $$\mathcal{Q}_{in} = \mathcal{A}[:, 0: d_{model}] \quad (11)$$

[0036] $$\mathcal{K}_{in} = \mathcal{A}[:, d_{model}: 2 \times d_{model}] \quad (12)$$

[0037] $$\mathcal{V}_{in} = \mathcal{A}[:, 2 \times d_{model}:] \quad (13)$$

[0038] 对$\mathcal{Q}_{in}$，$\mathcal{K}_{in}$和$\mathcal{V}_{\text{in}}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出；

[0039] 步骤33、解码器层的第二个子层多头注意力层中的键矩阵和值矩阵的线性投影过程相似，因此将权重矩阵$W_{K_{att}}$和$W_{V_{att}}$连接起来形成权重矩阵$W_{KV} \in \mathbb{R}^{d_{model} \times 2d_{model}}$，如公式(14)所示，同样也将权重矩阵$W_{KV}$映射到张量空间并将得到的权重张量$\mathcal{W}_{KV} \in$

$$\mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times 2d_{mk}\}}$$

[0040] 表示为k模的张量分解形式，所述公式(14)的定义如下：

[0041] $$W_{KV} = Concat\left(W_{K_{att}}, W_{V_{att}}\right) \quad (14)$$

[0042] 步骤34、对输入数据N进行reshape操作，然后利用公式(2)计算N与小型权重张量链($\mathcal{W}_{KV}$)的结果，结果记为$\mathcal{B}$,对$\mathcal{B}$进行reshape操作，然后进行切片，获得对应的$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$,具体过程如下：

[0043] $$\mathcal{N} = reshape\left(N, \left[-1, d_1, \ldots, d_k, d_{k+1}, \ldots, d_{2k}, \ldots, d_{(m-1)k+1}, \ldots, d_{mk}\right]\right) \quad (15)$$

[0044] $$\mathcal{B} = Lightweight\_Connect\left(\mathcal{N}, Tensorized(\mathcal{W}_{KV})\right) \quad (16)$$

[0045] $$\mathcal{B} = reshape\left(\mathcal{B}, \left[-1, d_{model} \times 2\right]\right) \quad (17)$$

[0046] $$\mathcal{K}_{in}^{'} = \mathcal{B}[:, 0: d_{model}] \quad (18)$$

[0047] $$\mathcal{V}_{in}^{'} = \mathcal{B}[:, d_{model}:] \quad (19)$$

[0048] 其中，$\mathcal{Q}_{\text{in}}^{'}$的计算流程与步骤21中$\mathcal{Q}$的计算流程一致，同样要对$\mathcal{Q}_{in}^{'}$，$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出。

[0049] 进一步地，所述步骤40包括以下具体步骤：

[0050] 步骤41、将d_{model}和d_{ff}转换为数值较小的正整数因子乘积，$d_{model}=\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}$和$d_{ff}=\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}$，权重矩阵$W_1^{ffn}\in\mathbb{R}^{d_{model}\times d_{ff}}$和$W_2^{ffn}\in\mathbb{R}^{d_{ff}\times d_{model}}$变为权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$,其中

$\mathcal{W}_1^{ffn}\in\mathbb{R}^{\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}\times\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}}$和

$\mathcal{W}_2^{\text{ffn}}\in\mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}\times\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$，

偏移向量b_1^{ffn}和b_2^{ffn}变成偏移张量

$\mathcal{B}_1^{ffn}\in\mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}}$和

$\mathcal{B}_2^{ffn}\in\mathbb{R}^{\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$；

[0051] 步骤42、利用公式(1)将权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$表示为m模的张量分解形式，以此降低网络模型中的训练参数数量和计算复杂度；

[0052] 步骤43、将Position-wise前馈网络的输入数据映射到张量空间，并与小型的权重张量链进行多步计算，其计算流程如公式(20)和(21)所示：

$$First = max\left(0, Lightweight\_Connect(\mathcal{X}, Tensorized(\mathcal{W}_1^{ffn}))\right) + \mathcal{B}_1^{ffn} \quad (20)$$

[0053] $$Lightweight\_Feed\_Forward\_Network = Lightweight\_Connect(First, Tensorized(\mathcal{W}_2^{ffn})) + \mathcal{B}_2^{ffn} \quad (21)$$

[0054] 进一步地，所述步骤3和步骤4的顺序不分先后。

[0055] 本发明的有益效果：

[0056] (1)设计了即插即用的轻量级张量化多头注意力机制和轻量级张量化position-wise前馈网络，降低了Transformer的学习参数数量和浮点操作运算，加快了模型的训练过程，降低了训练过程中的能源消耗。

[0057] (2)针对Transformer模型的编码过程和解码过程的特性，对多头注意力层的权重矩阵进行不同方式的拼接，并将拼接的权重矩阵表示为低秩的多模态张量分解形式，形成了轻量级张量化多头注意力机制++，以此进一步降低LTensorized_transformer学习参数数量和浮点操作运算，使得该网络模型能够部署到资源受限的边缘设备上。

[0058] (3)高效轻量级的张量耦合Transformer可以通过多个小型权重张量核对输入数据进行分步特征提取，保留Transformer模型性能的同时兼顾了Transformer的设计理念，三个轻量级模块可以灵活的嵌入到各种网络模型中，降低对应网络的训练参数数量和计算复杂度。

附图说明

[0059] 图1是本发明实施例提供的基于张量的高效Transformer的架构方法的实现流程图。

具体实施方式

[0060] 为了使本发明的目的、技术方案及优点更加清楚明白，以下结合附图及实施例，对本发明进行进一步详细说明。应当理解，此处所描述的具体实施例仅仅用以解释本发明，并不用于限定本发明。

[0061] 以下结合具体实施例对本发明的具体实现进行详细描述：

[0062] 实施例：

[0063] 如图1示出了本发明实施例提供的基于张量的高效Transformer的架构方法的实现流程，为了便于说明，仅示出了与本发明实施例相关的部分。详述如下：

[0064] 步骤10、将多头注意力层的权重矩阵$(W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}, W_O)$映射到张量空间，然后再将其表示为k模的张量分解形式链；

[0065] 步骤20、将输入数据$(Q_{att}, K_{att}, V_{att})$映射到张量空间，然后与对应的$\mathcal{W}_{Q_{att}}, \mathcal{W}_{K_{att}}, \mathcal{W}_{V_{att}}$的权重张量链进行运算，并把结果进行“Attention”运算，再与$\mathcal{W}_O$的权重张量链进行运算得到最终输出结果，构建出轻量级张量化多头注意力机制；

[0066] 步骤30、将编码器层的多头注意力层和解码器层的第一层子层的多头注意力层中相似的线性运算整合到一起，即把对应的权重张量拼接起来，形成权重张量$\mathcal{W}_{QKV}$，将解码器层的第二个子层中相似的线性运算整合到一起，形成权重张量$\mathcal{W}_{KV}$，构建出轻量级张量化多头注意力机制++；

[0067] 步骤40、将Position-wise前馈神经网络中的权重矩阵W_1^{ffn}和W_2^{ffn}映射到张量空间且将其表示为m模的张量分解形式，并与输入数据进行运算，构建出轻量级张量化Position-wise前馈网络；

[0068] 步骤50、将轻量级张量化多头注意力机制和轻量级张量化Position-Wise前馈网

络组成Ltensorized_transformer,将轻量级张量化Position-Wise前馈网络和轻量级张量化多头注意力机制++组成LTensorized_transformer++,构建出轻量级Transformer架构。

[0069] 进一步地,步骤10包括以下具体步骤:

[0070] 将queries,keys和values分别打包成矩阵Q_{att},K_{att}和V_{att},并对矩阵Q_{att},K_{att}和V_{att}进行了h次线性投影,其中涉及到的权重矩阵可以统一表示为$W_{Q_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$,$W_{K_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$,$W_{V_{att}} \in \mathbb{R}^{d_{model} \times d_{model}}$和$W_O \in \mathbb{R}^{d_{model} \times d_{model}}$,将$d_{model}$表示成多个正数因子的乘积,$d_{model}=\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}$;

[0071] 将权重矩阵$W_{Q_{att}}$,$W_{K_{att}}$,$W_{V_{att}}$和W_0映射到张量空间就得到权重张量$\mathcal{W}_{Q_{att}}$,$\mathcal{W}_{K_{att}}$,$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$($\mathcal{W}_{Q_{att}}$,$\mathcal{W}_{K_{att}}$,$\mathcal{W}_{V_{att}}$,$\mathcal{W}_O \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\}}$);

[0072] 根据公式(1)将权重张量$\mathcal{W}_{Q_{att}}$,$\mathcal{W}_{K_{att}}$,$\mathcal{W}_{V_{att}}$和$\mathcal{W}_O$表示为k模的张量分解形式,公式(1)的定义如下:

[0073]
$$Tensorized\left(\mathcal{W}'\right) = (\mathcal{W}_1', \mathcal{W}_2', \ldots, \mathcal{W}_{2m}') \rightarrow \mathcal{W}_1' *_{r_1^{(1)},\ldots,r_k^{(1)}} \mathcal{W}_2' *_{r_1^{(2)},\ldots,r_k^{(2)}} \cdots *_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}} \mathcal{W}_{2m}' \quad (1)$$

[0074] 进一步地,步骤20包括以下具体步骤:

[0075] 步骤21、通过把输入数据映射到张量空间$(\mathcal{Q}_{att}, \mathcal{K}_{att}, \mathcal{V}_{att})$,并与对应的小型权重张量核进行运算,运算的过程如公式(2)所示,公式(2)的定义如下:

[0076]
$$Lightweight\_Connect(\mathcal{X}, Tensorized(\mathcal{W})) = \mathcal{X} *_{d_1,\ldots,d_k} \mathcal{W}_1 *_{r_1^{(1)},\ldots,r_k^{(1)},d_{k+1},\ldots,d_{2k}} \mathcal{W}_2 * \cdots *_{r_1^{(m-1)},\ldots,r_k^{(m-1)},d_{(m-1)k+1},\ldots,d_{mk}} \mathcal{W}_m *_{r_1^{(m)},\ldots,r_k^{(m)}} \mathcal{W}_{m+1} *_{r_1^{(m+1)},\ldots,r_k^{(m+1)}} \mathcal{W}_{m+2} * \cdots *_{r_1^{(2m-1)},\ldots,r_k^{(2m-1)}} \mathcal{W}_{2m} \quad (2)$$

[0077] 步骤22、将运算结果进行reshape操作,并将其拆分成h等分,用列表L存储拆分的结果,整个计算过程如下:

[0078] $D' = \text{Reshape}(D, [-1, d_{model}])$ (3)

[0079] $T = \text{Split}(D') = (D'_1, \ldots, D'_h)$ (4)

[0080] 步骤23、将列表L中的存储数据取出来,$\mathcal{Q}', \mathcal{K}', \mathcal{V}' \leftarrow L$,$\mathcal{Q}'$、$\mathcal{K}'$知$\mathcal{V}'$中各有由h个矩阵组成,利用公式(5)来获取对应下标的输入矩阵(Q_{att}^i,K_{att}^i和V_{att}^i),并进行注意力计算,从而获得对应的注意力输出,公式(5)的定义如下:

[0081] $R_i = \text{Get}(R, i)$ (5)

[0082] 步骤24、利用公式(6)计算每个注意力的输出结果($head_i$),并将每个注意力的结果拼接在一起,与小型的权重张量核($\mathcal{W}_O$的k模张量分解形式的结果)利用公式(2)进行多步特征计算得到最终的多头注意力层的输出结果,公式(6)的定义如下:

[0083] $$head_i = Attention\left(Q_{att}^i, K_{att}^i, V_{att}^i\right) = softmax\frac{\left(Q_{att}^i {K_{att}^i}^T\right)}{\sqrt{d}} V_{att}^i \quad (6)$$

[0084] 进一步地，步骤30包括以下具体步骤：

[0085] 步骤31、编码器层的多头注意力层和解码器层的第一层子层多头注意力层的结构性质相同，查询矩阵Q_{att}、键矩阵K_{att}和值矩阵V_{att}做了相似的线性映射操作，将权重矩阵$W_{Q_{att}}$，$W_{K_{att}}$和$W_{V_{att}}$连接成一个巨大的权重矩阵$W_{QKV} \in \mathbb{R}^{d_{model} \times 3d_{model}}$，如公式(7)所示，然后再利用公式(1)将权重矩阵W_{QKV}映射到张量空间并将得到的权重张量$\mathcal{W}_{QKV} \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times 3d_{mk}\}}$

[0086] 表示为k模的张量分解形式，公式(7)的定义如下：

[0087] $$W_{QKV} = Concat\left(W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}\right) \quad (7)$$

[0088] 步骤32、对输入数据M进行reshape操作，然后利用公式(2)计算M与小型权重张量链$(\mathcal{W}_{QKV})$的结果，结果记为$\mathcal{A}$，对$\mathcal{A}$进行reshape操作，然后进行切片，获得对应的$\mathcal{Q}_{\text{in}}$，$\mathcal{K}_{in}$和$\mathcal{V}_{in}$，具体过程如下：

[0089] $$\mathcal{M} = reshape\left(M, \left[-1, d_1, \ldots, d_k, d_{k+1}, \ldots, d_{2k}, \ldots, d_{(m-1)k+1}, \ldots, d_{mk}\right]\right) \quad (8)$$

[0090] $$\mathcal{A} = ightweight\_Connect\left(\mathcal{M}, Tensorized\left(\mathcal{W}_{QKV}\right)\right) \quad (9)$$

[0091] $$\mathcal{A} = reshape\left(\mathcal{A}, \left[-1, d_{model} \times 3\right]\right) \quad (10)$$

[0092] $$\mathcal{Q}_{in} = \mathcal{A}[:, 0: d_{model}] \quad (11)$$

[0093] $$\mathcal{K}_{in} = \mathcal{A}[:, d_{model}: 2 \times d_{model}] \quad (12)$$

[0094] $$\mathcal{V}_{in} = \mathcal{A}[:, 2 \times d_{model}:] \quad (13)$$

[0095] 对$\mathcal{Q}_{in}$，$\mathcal{K}_{in}$和$\mathcal{V}_{\text{in}}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出；

[0096] 步骤33、解码器层的第二个子层多头注意力层中的键矩阵和值矩阵的线性投影过程相似，因此将权重矩阵$W_{K_{att}}$和$W_{V_{att}}$连接起来形成权重矩阵$W_{KV} \in \mathbb{R}^{d_{model} \times 2d_{model}}$，如公式(14)所示，同样也将权重矩阵$W_{KV}$映射到张量空间并将得到的权重张量$\mathcal{W}_{KV} \in \mathbb{R}^{\{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times d_{mk}\} \times \{d_1 \times \ldots \times d_k\} \times \{d_{k+1} \times \ldots \times d_{2k}\} \times \ldots \times \{d_{(m-1)k+1} \times \ldots \times 2d_{mk}\}}$

[0097] 表示为k模的张量分解形式，公式(14)的定义如下：

[0098] $$W_{KV} = Concat\left(W_{K_{att}}, W_{V_{att}}\right) \quad (14)$$

[0099] 步骤34、对输入数据N进行reshape操作，然后利用公式(2)计算N与小型权重张量链$(\mathcal{W}_{KV})$的结果，结果记为$\mathcal{B}$，对$\mathcal{B}$进行reshape操作，然后进行切片，获得对应的$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$，具体过程如下：

[0100] $$\mathcal{N} = reshape\left(N, \left[-1, d_1, \ldots, d_k, d_{k+1}, \ldots, d_{2k}, \ldots, d_{(m-1)k+1}, \ldots, d_{mk}\right]\right) \quad (15)$$

[0101] $$\mathcal{B} = Lightweight\_Connect\left(\mathcal{N}, Tensorized(\mathcal{W}_{KV})\right) \quad (16)$$

[0102] $$\mathcal{B} = reshape(\mathcal{B}, [-1, d_{model} \times 2]) \quad (17)$$

[0103] $$\mathcal{K}_{in}^{'} = \mathcal{B}[:, 0: d_{model}] \quad (18)$$

[0104] $$\mathcal{V}_{in}^{'} = \mathcal{B}[:, d_{model}:] \quad (19)$$

[0105] 其中，$\mathcal{Q}_{\text{in}}^{'}$的计算流程与步骤21中$\mathcal{Q}$的计算流程一致，同样要对$\mathcal{Q}_{in}^{'}$，$\mathcal{K}_{in}^{'}$和$\mathcal{V}_{in}^{'}$进行拆分操作，并将拆分结果用列表存储起来，并对其进行步骤23和步骤24的操作，从而得到最终的输出。

[0106] 进一步地，步骤40包括以下具体步骤：

[0107] 步骤41、将d_{model}和d_{ff}转换为数值较小的正整数因子乘积，$d_{model}=\{0_1\times\ldots\times0_m\}\times\{0_{m+1}\times\ldots\times0_{2m}\}\times\ldots\times\{0_{(n-1)m+1}\times\ldots\times0_{nm}\}$和$d_{ff}=\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}$，权重矩阵$W_1^{ffn}\in\mathbb{R}^{d_{model}\times d_{ff}}$和$W_2^{ffn}\in\mathbb{R}^{d_{ff}\times d_{model}}$变为权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$，其中

$\mathcal{W}_1^{ffn}\in\mathbb{R}^{\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}\times\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}}$和

$\mathcal{W}_2^{ffn}\in\mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}\times\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$，偏移向量$b_1^{ffn}$和$b_2^{ffn}$变成偏移张量$\mathcal{B}_1^{ffn}\in\mathbb{R}^{\{P_1\times\ldots\times P_m\}\times\{P_{m+1}\times\ldots\times P_{2m}\}\times\ldots\times\{P_{(n-1)m+1}\times\ldots\times P_{nm}\}}$和$\mathcal{B}_2^{ffn}\in\mathbb{R}^{\{O_1\times\ldots\times O_m\}\times\{O_{m+1}\times\ldots\times O_{2m}\}\times\ldots\times\{O_{(n-1)m+1}\times\ldots\times O_{nm}\}}$；

[0108] 步骤42、利用公式(1)将权重张量$\mathcal{W}_1^{ffn}$和$\mathcal{W}_2^{ffn}$表示为m模的张量分解形式，以此降低网络模型中的训练参数数量和计算复杂度；

[0109] 步骤43、将Position-wise前馈网络的输入数据映射到张量空间，并与小型的权重张量链进行多步计算，其计算流程如公式(20)和(21)所示：

$$First = max\,(0, Lightweight\_Connect(\mathcal{X}, Tensorized(\mathcal{W}_1^{ffn}))) + \mathcal{B}_1^{ffn} \quad (20)$$

[0110] $$Lightweight\_Feed\_Forward\_Network = Lightweight\_Connect(First, Tensorized(\mathcal{W}_2^{ffn})) + \mathcal{B}_2^{ffn} \quad (21)$$

[0111] 进一步地，步骤3和步骤4的顺序不分先后。

[0112] 以上仅为本发明的较佳实施例而已，并不用以限制本发明，凡在本发明的精神和原则之内所作的任何修改、等同替换和改进等，均应包含在本发明的保护范围之内。

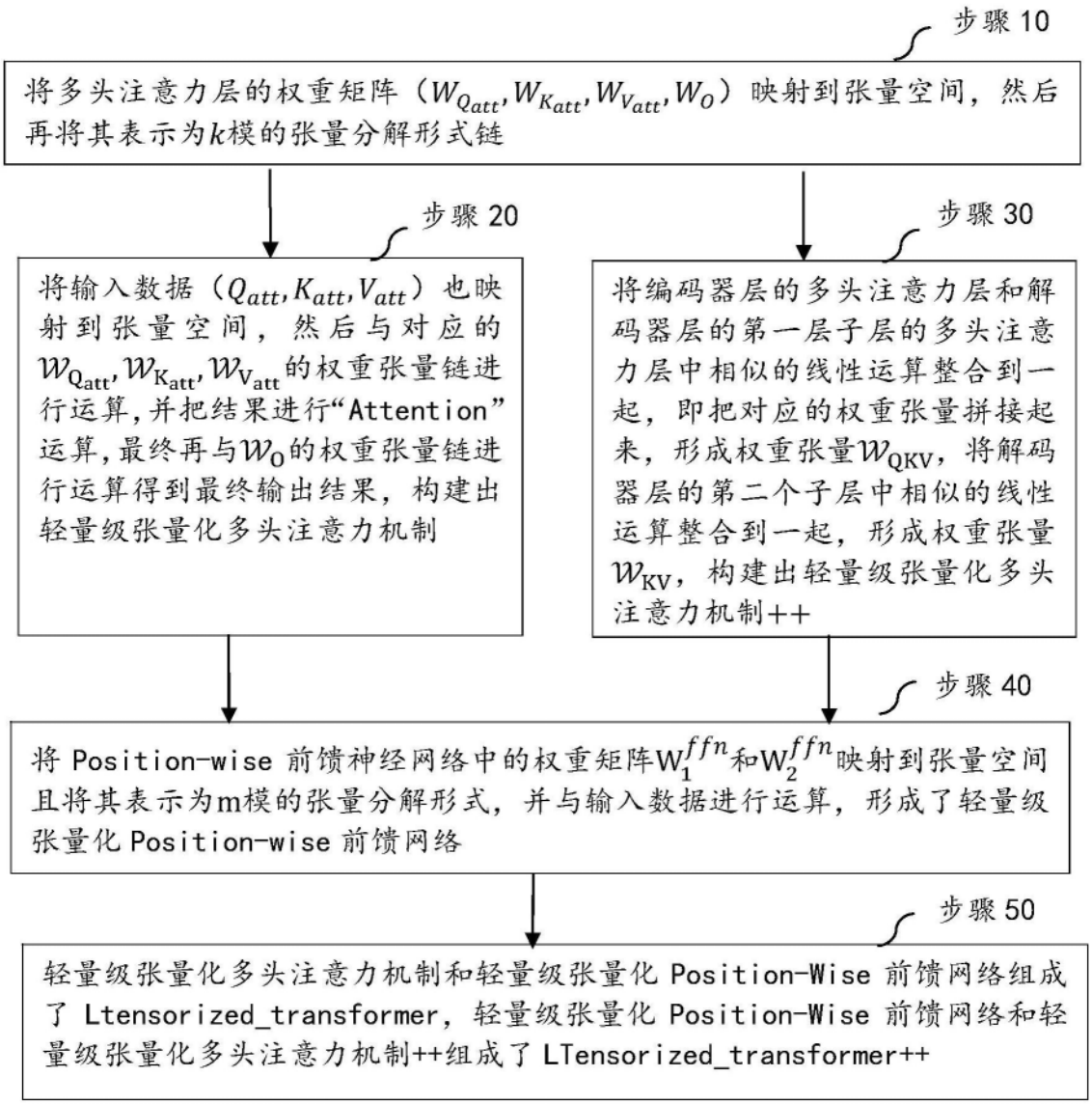

步骤 10
将多头注意力层的权重矩阵（$W_{Q_{att}}, W_{K_{att}}, W_{V_{att}}, W_O$）映射到张量空间，然后再将其表示为$k$模的张量分解形式链
步骤 20
将输入数据（$Q_{att}, K_{att}, V_{att}$）也映射到张量空间，然后与对应的$\mathcal{W}_{\mathrm{Q_{att}}}, \mathcal{W}_{\mathrm{K_{att}}}, \mathcal{W}_{\mathrm{V_{att}}}$的权重张量链进行运算，并把结果进行“Attention”运算，最终再与$\mathcal{W}_{\mathrm{O}}$的权重张量链进行运算得到最终输出结果，构建出轻量级张量化多头注意力机制
步骤 30
将编码器层的多头注意力层和解码器层的第一层子层的多头注意力层中相似的线性运算整合到一起，即把对应的权重张量拼接起来，形成权重张量$\mathcal{W}_{\mathrm{QKV}}$，将解码器层的第二个子层中相似的线性运算整合到一起，形成权重张量$\mathcal{W}_{\mathrm{KV}}$，构建出轻量级张量化多头注意力机制++
步骤 40
将 Position-wise 前馈神经网络中的权重矩阵W_1^{ffn}和W_2^{ffn}映射到张量空间且将其表示为m模的张量分解形式，并与输入数据进行运算，形成了轻量级张量化 Position-wise 前馈网络
步骤 50
轻量级张量化多头注意力机制和轻量级张量化 Position-Wise 前馈网络组成了 Ltensorized_transformer，轻量级张量化 Position-Wise 前馈网络和轻量级张量化多头注意力机制++组成了 LTensorized_transformer++

图1