权 利 要 求 书

1．一种用于大语言模型分布式推理的计算任务分配系统，其特征在于，包括：

环境部署模块，用于在多个安卓手机端搭建分布式机器学习环境；

模型分割模块，用于对 PC 端的大语言模型进行切割，得到多个子模型，并获取手机端的多个可执行子模型文件；

计算任务分配模块,用于对所述模型分割模块分割而成的各子模型的计算成本和通信成本进行优化，得到最佳部署方案；以及

子模型运行模块,用于根据所述最佳部署方案在所述多个安卓手机端部署子模型，分别启动客户端和服务端的加载代码，进行子模型的推理。

2．根据权利要求 1 所述的用于大语言模型分布式推理的计算任务分配系统，其特征在于，所述计算任务分配模块的优化参数包括：子模型的 Flops 和参数量、设备处理 Flops 速度、设备的最大内存、通信的延时、带宽、抖动和丢包率。

3．根据权利要求 1 所述的用于大语言模型分布式推理的计算任务分配系统，其特征在于，所述计算任务分配模块的优化采用以下公式：

$$minimize\left(T_{compute} + T_{data} + T_{complexity} + T_{QoS}\right)$$

其中，$T_{compute} = \sum\sum \epsilon_{i,j} \cdot x_{i,j}$；

$T_{data} = \sum\sum \left(L_{i,j} + t_{i,j}\right)$；

$T_{complexity} = w_c \cdot \sum\sum x_{i,j}$；

$T_{QoS} = \sum\sum w_{q1} \cdot Jitter_{i,j} + w_{q2} \cdot t_{i,j} \cdot PLR_{i,j} + w_{q3} \cdot PLR_{i,j}^2$；

$$\epsilon_{i,j} = \frac{NumFlop_{modj}}{FLOP/s_i}$$

$$t_{i,j} = \frac{O_{i,j}}{E_p \cdot B_{i,j}}$$

其中，$T_{complexity}$为系统复杂度惩罚项；

T_{QoS}为链路质量惩罚项；

$Jitter, t, PLR$分别为对应链路$\langle i,j\rangle$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_c、w_{q1}、w_{q2}、w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_c = 1, w_1 = 10$，$w_2 = 1$，$w_3 = 10000$；

E_p为通信协议的有效承载效率，典型值为 0.3；

$O_{i,j}$为在链路$\langle i,j\rangle$上的输出数据量；

$B_{i,j}$为链路$\langle i,j\rangle$的理论带宽；

$L_{i,j}$为链路$\langle i,j\rangle$上的延迟；

$NumFlop_{modj}$ 为子模型j所需要进行浮点运算次数；

$FLOP/s_i$为设备i每秒浮点运算次数；

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

4．根据权利要求 1 所述的用于大语言模型分布式推理的计算任务分配系统，其特征在于，所述计算任务分配模块还包括如下显式约束项：

$$\forall i \in \{0,\dots,m-1\} \sum_{j=0}^{n-1} M_{mod_j} \cdot x_{i,j} \leq \beta \cdot M_{device_i}$$

其中，m 和 n 分别代表设备数和子模型数，

M_{mod} 和 M_{device} 分别表示子模型的参数量和设备的最大内存，

β 是设备能够分配给模型推理的最大内存比例，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

5．根据权利要求 1 或 4 所述的用于大语言模型分布式推理的计算任务分配系统，其特征在于，所述计算任务分配模块还包括如下隐式约束项：

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} x_{i,j} = n$$

$$x[0,0] = 1$$

$$\forall j \in \{0,\dots,n-1\} \sum_{i=0}^{m-1} x_{i,j} = 1$$

其中，m 和 n 分别代表设备数和子模型数，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

6．一种用于大语言模型分布式推理的计算任务分配方法，其特征在于，包括以下步骤：

步骤 1，进行环境部署，在多个安卓手机端搭建分布式学习环境；

步骤 2，切割大语言模型，生成多个子模型，并获取手机端的多个可执行子模型文件；

步骤 3，对子模型的参数进行线性规划，得到子模型分配模型；

步骤 4，建立手机的客户端和服务端，运行子模型分配模型进行推理。

7．根据权利要求 6 所述的用于大语言模型分布式推理的计算任务分配方法，其特征在于，所述步骤 3 中线性规划的方法为：以子模型的部署

二进制矩阵为变量，根据子模型的Flops和参数量、设备处理Flops速度、设备的最大内存、通信的延时、带宽、抖动和丢包率建模最小推理时间表达式，根据设备的最大内存和模型的参数量约束变量，进行混合整数线性规划。

8．根据权利要求7所述的用于大语言模型分布式推理的计算任务分配方法，其特征在于，所述最小推理时间表达式为：

$$minimize\left(T_{compute}+T_{data}+T_{complexity}+T_{QoS}\right)$$

其中，$T_{compute}=\sum\sum\epsilon_{i,j}\cdot x_{i,j}$；

$T_{data}=\sum\sum\left(L_{i,j}+t_{i,j}\right)$；

$T_{complexity}=w_c\cdot\sum\sum x_{i,j}$；

$T_{QoS}=\sum\sum w_{q1}\cdot Jitter_{i,j}+w_{q2}\cdot t_{i,j}\cdot PLR_{i,j}+w_{q3}\cdot PLR_{i,j}^2$；

$$\epsilon_{i,j}=\frac{NumFlop_{modj}}{FLOP/s_i}$$

$$t_{i,j}=\frac{O_{i,j}}{E_p\cdot B_{i,j}}$$

其中，$T_{complexity}$为系统复杂度惩罚项；

T_{QoS}为链路质量惩罚项；

$Jitter, t, PLR$分别为对应链路$<i,j>$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_c、w_{q1}、w_{q2}、w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_c=1, w_1=10$，$w_2=1$，$w_3=10000$；

E_p为通信协议的有效承载效率，典型值为0.3；

$O_{i,j}$为在链路$\langle i,j\rangle$上的输出数据量；

$B_{i,j}$为链路$\langle i,j\rangle$的理论带宽；

$L_{i,j}$为链路$\langle i,j\rangle$上的延迟；

$NumFlop_{mod_j}$ 为子模型j所需要进行浮点运算次数；

$FLOP/s_i$为设备i每秒浮点运算次数；

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

9．根据权利要求 6 所述的用于大语言模型分布式推理的计算任务分配方法，其特征在于，所述步骤 3 中线性规划包括如下显式约束项：

$$\forall i \in \{0,\dots,m-1\} \sum_{j=0}^{n-1} M_{mod_j} \cdot x_{i,j} \leq \beta \cdot M_{device_i}$$

其中，m 和 n 分别代表设备数和子模型数，

M_{mod} 和 M_{device} 分别表示模型的参数量和设备的最大内存，

β 是设备能够分配给模型推理的最大内存比例，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

10.根据权利要求 6 或 9 所述的用于大语言模型分布式推理的计算任务分配方法，其特征在于，所述步骤 3 中线性规划包括如下隐式约束项：

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{i,j} = n$$

$$x[0,0] = 1$$

$$\forall j \in \{0,\dots,n-1\} \sum_{i=0}^{m-1} x_{i,j} = 1$$

其中，m 和 n 分别代表设备数和子模型数，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

说　　明　　书

用于大语言模型分布式推理的计算任务分配系统及方法

技术领域

本发明涉及一种计算任务的调度系统,具体涉及一种用于大语言模型分布式推理的计算任务分配系统。本发明还涉及一种用于大语言模型分布式推理的计算任务分配方法。

背景技术

大语言模型（Large Language Model，LLM）是一种基于深度学习的人工智能技术,旨在通过大规模的训练数据和深度神经网络来实现语言理解和生成的能力。这种模型可以用于多种自然语言处理任务,如机器翻译、文本生成、对话系统等；在智能助手和虚拟人物领域，大型语言模型可以模拟人类对话，与用户进行交互，并提供智能化的服务。在教育、医疗、金融等行业，大型语言模型也有广泛的应用，可以辅助教学、辅助医疗决策、智能客服等。

目前在基于分布式边缘设备推理的研究中,有些优化模型对于分布式推理时间的优化在于对计算时间和数据传输时间之和进行最小化。这种优化在理论上是成立的,但在实际部署中,未能充分考虑网络通信的复杂性,并且系统复杂性的增加会导致数据传输成本和风险的增加。

相对于计算，数据传输往往更花时间。传输时间不仅仅要考虑带宽和延迟，而且需要考虑网络通信质量的其他因素如抖动、丢包率等的影响。同时，在计算传输时间时，通信协议的实际有效载荷也是一大问题。

因此，目前大语言模型分布式边缘设备推理中，通信成本过高是其主要缺陷之一。

发明内容

本发明所要解决的技术问题是提供一种用于大语言模型分布式推理的计算任务分配系统，它可以降低通信成本。

为解决上述技术问题，本发明用于大语言模型分布式推理的计算任务分配系统的技术解决方案为：

包括环境部署模块、模型分割模块、计算任务分配模块和子模型运行模块，环境部署模块用于在多个安卓手机端搭建分布式机器学习环境；模型分割模块用于对 PC 端的大语言模型进行切割，得到多个子模型，并获取手机端的多个可执行子模型文件；计算任务分配模块用于对所述模型分割模块分割而成的各子模型的计算成本和通信成本进行优化，得到最佳部署方案；子模型运行模块用于根据所述最佳部署方案在所述多个安卓手机端部署子模型，分别启动客户端和服务端的加载代码，进行子模型的推理。

在另一实施例中，所述计算任务分配模块的优化参数包括：子模型的 Flops 和参数量、设备处理 Flops 速度、设备的最大内存、通信的延时、带宽、抖动和丢包率。

在另一实施例中，所述计算任务分配模块的优化采用以下公式：

$$minimize\left(T_{compute} + T_{data} + T_{complexity} + T_{QoS}\right)$$

其中，$T_{compute} = \sum\sum \epsilon_{i,j} \cdot x_{i,j}$；

$T_{data} = \sum\sum \left(L_{i,j} + t_{i,j}\right)$；

$$T_{complexity} = w_c \cdot \sum\sum x_{i,j};$$

$$T_{QoS} = \sum\sum w_{q1} \cdot Jitter_{i,j} + w_{q2} \cdot t_{i,j} \cdot PLR_{i,j} + w_{q3} \cdot PLR_{i,j}^2;$$

$$\epsilon_{i,j} = \frac{NumFlop_{modj}}{FLOP/s_i}$$

$$t_{i,j} = \frac{O_{i,j}}{E_p \cdot B_{i,j}}$$

其中，$T_{complexity}$为系统复杂度惩罚项；

T_{QoS}为链路质量惩罚项；

$Jitter, t, PLR$分别为对应链路$<i,j>$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_c、w_{q1}、w_{q2}、w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_c = 1, w_1 = 10$，$w_2 = 1$，$w_3 = 10000$；

E_p为通信协议的有效承载效率，典型值为 0.3；

$O_{i,j}$为在链路$<i,j>$上的输出数据量；

$B_{i,j}$为链路$<i,j>$的理论带宽；

$L_{i,j}$为链路$<i,j>$上的延迟；

$NumFlop_{modj}$ 为子模型j所需要进行浮点运算次数；

$FLOP/s_i$为设备i每秒浮点运算次数；

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

在另一实施例中，所述计算任务分配模块还包括如下显式约束项：

$$\forall i \in \{0, \ldots, m-1\} \sum_{j=0}^{n-1} M_{mod_j} \cdot x_{i,j} \leq \beta \cdot M_{device_i}$$

其中，m 和 n 分别代表设备数和子模型数，

M_{mod} 和 M_{device} 分别表示子模型的参数量和设备的最大内存，

β 是设备能够分配给模型推理的最大内存比例，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

在另一实施例中，所述计算任务分配模块还包括如下隐式约束项：

$$\sum_{i=0}^{m-1}\sum_{j=0}^{n-1} x_{i,j} = n$$

$$x[0,0] = 1$$

$$\forall j \in \{0,\dots,n-1\} \sum_{i=0}^{m-1} x_{i,j} = 1$$

其中，m 和 n 分别代表设备数和子模型数，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

本发明还提供一种用于大语言模型分布式推理的计算任务分配方法，其技术解决方案为，包括以下步骤：

步骤 1，进行环境部署，在多个安卓手机端搭建分布式学习环境；

步骤 2，切割大语言模型，生成多个子模型，并获取手机端的多个可执行子模型文件；

步骤 3，对子模型的参数进行线性规划，得到子模型分配模型；

步骤 4，建立手机的客户端和服务端，运行子模型分配模型进行推理。

在另一实施例中，所述步骤 3 中线性规划的方法为：以子模型的部署二进制矩阵为变量，根据子模型的Flops 和参数量、设备处理Flops 速度、

设备的最大内存、通信的延时、带宽、抖动和丢包率建模最小推理时间表达式，根据设备的最大内存和模型的参数量约束变量，进行混合整数线性规划。

在另一实施例中，所述最小推理时间表达式为：

$$minimize\left(T_{compute}+T_{data}+T_{complexity}+T_{QoS}\right)$$

其中，$T_{compute}=\sum\sum\epsilon_{i,j}\cdot x_{i,j}$；

$T_{data}=\sum\sum\left(L_{i,j}+t_{i,j}\right)$；

$T_{complexity}=w_c\cdot\sum\sum x_{i,j}$；

$T_{QoS}=\sum\sum w_{q1}\cdot Jitter_{i,j}+w_{q2}\cdot t_{i,j}\cdot PLR_{i,j}+w_{q3}\cdot PLR_{i,j}^2$；

$$\epsilon_{i,j}=\frac{NumFlop_{modj}}{FLOP/s_i}$$

$$t_{i,j}=\frac{O_{i,j}}{E_p\cdot B_{i,j}}$$

其中，$T_{complexity}$为系统复杂度惩罚项；

T_{QoS}为链路质量惩罚项；

$Jitter, t, PLR$分别为对应链路$<i,j>$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_c、w_{q1}、w_{q2}、w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_c=1, w_1=10$，$w_2=1$，$w_3=10000$；

E_p为通信协议的有效承载效率，典型值为0.3；

$O_{i,j}$为在链路$<i,j>$上的输出数据量；

$B_{i,j}$为链路$<i,j>$的理论带宽；

$L_{i,j}$为链路$\langle i,j\rangle$上的延迟；

$NumFlop_{mod_j}$ 为子模型j所需要进行浮点运算次数；

$FLOP/s_i$为设备i每秒浮点运算次数；

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

在另一实施例中，所述步骤 3 中线性规划包括如下显式约束项：

$$\forall i \in \{0,\dots,m-1\} \sum_{j=0}^{n-1} M_{mod_j} \cdot x_{i,j} \leq \beta \cdot M_{device_i}$$

其中，m 和 n 分别代表设备数和子模型数，

M_{mod} 和 M_{device} 分别表示模型的参数量和设备的最大内存，

β 是设备能够分配给模型推理的最大内存比例，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

在另一实施例中，所述步骤 3 中线性规划包括如下隐式约束项：

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{i,j} = n$$

$$x[0,0] = 1$$

$$\forall j \in \{0,\dots,n-1\} \sum_{i=0}^{m-1} x_{i,j} = 1$$

其中，m 和 n 分别代表设备数和子模型数，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

本发明可以达到的技术效果是：

本发明通过模型拆分、流水线式推理和线性规划，能够在手机端实现分布式的运行大语言模型进行推理等工作，充分调用边缘计算力，从而降

低通信成本。

本发明具备分布式特点，并且能够以最优解的方式部署到手机端等边缘设备进行大语言模型的推理，在当前以 GPU 推理和训练的大背景下，具有很高的开发潜力和应用场景。

附图说明

本领域的技术人员应理解，以下说明仅是示意性地说明本发明的原理，所述原理可按多种方式应用，以实现许多不同的可替代实施方式。这些说明仅用于示出本发明的教导内容的一般原理，不意味着限制在此所公开的发明构思。

结合在本说明书中并构成本说明书的一部分的附图示出了本发明的实施方式，并且与上文的总体说明和下列附图的详细说明一起用于解释本发明的原理。

下面结合附图和具体实施方式对本发明作进一步详细的说明：

图 1 是丢包率对 QoS 惩罚值的影响示意图；从图 1 中可以看出，随着丢包率的提高，即网络质量的劣化，该惩罚值会显著提高；

图 2 是抖动对 QoS 惩罚值的影响示意图；从图 2 中可以看出，抖动的增加会引起惩罚值的进一步提高。

具体实施方式

为使本发明实施例的目的、技术方案和优点更加清楚，下面将结合本发明实施例的附图，对本发明实施例的技术方案进行清楚、完整地描述。显然，所描述的实施例是本发明的一部分实施例，而不是全部的实施例。基于所描述的本发明的实施例，本领域普通技术人员在无需创造性劳动的

前提下所获得的所有其他实施例，都属于本发明保护的范围。除非另外定义，此处使用的技术术语或者科学术语应当为本发明所属领域内具有一般技能的人士所理解的通常意义。本文中使用的“第一”、“第二”以及类似的词语并不表示任何顺序、数量或者重要性，而只是用来区分不同的组成部分。“包括”等类似的词语意指出现该词前面的元件或者物件涵盖出现在该词后面列举的元件或者物件及其等同，而不排除其他元件或者物件。

本发明用于大语言模型分布式推理的计算任务分配系统，适用于Android（安卓）手机，包括：

环境部署模块，用于在手机端搭建分布式机器学习环境；

做为一实施例，可以在手机端安装完整的Linux ubuntu系统（乌班图计算机操作系统）和适用于arm架构（基于精简指令集的处理器架构）的conda python虚拟环境(一种开源的软件包管理系统和环境管理系统)；

模型分割模块，用于对PC端（电脑端）的大语言模型进行切割，得到多个子模型，并获取手机端的多个可执行子模型文件；

具体地，可以按模型定义结构的所有层进行切割，并全部转化为手机端的可执行ONNX（Open Neural Network Exchange ，开放式神经网络交换系统）文件格式；

由于手机的计算能力有限，大语言模型无法在单一手机上完成运算；本发明通过模型分割模块将大语言模型进行分割，以便将分割后的子模型部署到多个手机上来进行计算。

计算任务分配模块，用于对子模型的计算成本和通信成本进行优化，得到最佳的部署方案；

计算任务分配模块考虑的参数包括子模型的Flops（每秒浮点操作数，floating-point operations per second）和参数量、设备处理 Flops 的速度、设备的最大内存、通信的延时、带宽、抖动和丢包率，以推理时间的最小值为目标获取解决方案。

计算任务分配模块的优化对象为模型的 decoder 解码器层；

计算任务分配模块的优化方法为混合整数线性规划,考虑变量为二维二进制模型分布矩阵的情况。

优选地,计算任务分配模块可以采用以二项分布建模丢包率的数学惩罚项模型，增加非线性的不确定性，符合真实情况；

子模型运行模块,用于根据最佳的部署方案在多个手机端部署子模型,分别启动客户端和服务端的加载代码，进行子模型的推理；

子模型运行模块以大语言模型的 decoder 解码器层作为服务端，以 embedding 嵌入层和下游任务等作为客户端闭环执行推理。

具体地，子模型运行模块基于 ONNX 运行时编写子模型推理代码，以大语言模型的 decoder 解码器层作为服务端,以 embedding 嵌入层和下游任务等作为客户端闭环执行推理。

本发明用于大语言模型分布式推理的计算任务分配方法,包括以下步骤：

步骤 1，进行环境部署，在 Android 手机端搭建分布式学习环境；

在多个 Android 手机端部署手机端编程环境，搭建完整的操作系统，构建虚拟环境，下载基础的数据库。

步骤 2，进行模型分割，切割大语言模型，生成若干子模型；

对 Huggingface（开源模型库）上的模型源代码进行修改，在各层交接处添加模型转化代码,目的是将模型由电脑端格式转为手机端可执行的 ONNX 格式子模型，同时将模型的名称、输入和输出都进行显示转换；输入由于要适应不同长度的 embedding 嵌入层离散变量,需要设置输入的维度为动态；

在完成所有的子模型定义后，调用修改后的源码库进行模型推理，得到独立的子模型文件。

步骤 3，对子模型的参数进行线性规划，得到子模型分配模型；

线性规划过程采用一个变量、一个表达式项和若干约束项；对于模型切割出来的若干子模型 decoder 层，以子模型的部署二进制矩阵为变量，根据子模型的 Flops、设备处理 Flops 速度、通信的延时、带宽、抖动和丢包率这几个参数建模最小推理时间表达式,根据设备的最大内存和模型的参数量约束变量，进行混合整数线性规划。

在硬件条件有限的前提下，提高运行速度，即可降低通信成本。本发明通过寻找推理时间的最小值，实现大语言模型推理加速的目的。由于推理时间由计算时间和通信时间组成,因此最小推理时间表达式为计算成本和通信成本之和；

其中，计算成本$T_{compute}$为所有模型的 Flops 与所在设备处理 Flops 速度之比求和，即：

$$T_{compute} = \sum\sum \epsilon_{i,j} \cdot x_{i,j}$$

其中，$\epsilon_{i,j} = \frac{NumFlop_{modj}}{FLOP/s_i}$

$NumFlop_{modj}$ 为子模型j所需要进行浮点运算次数，

$FLOP/s_i$为设备i每秒浮点运算次数，

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

对于通信成本的计算，为了更好的模拟现实环境中通信的复杂程度，采用四个通信指标：延时、带宽、抖动和丢包率；其中，延时和带宽属于通信的常规量，由T_{data}表示；则T_{data}反应了理论上的通信时间，有：

$$T_{data} = \sum\sum (L_{i,j} + t_{i,j})$$

其中，$t_{i,j} = \frac{O_{i,j}}{E_p \cdot B_{i,j}}$

$O_{i,j}$为在链路$\langle i,j \rangle$上的输出数据量，

$B_{i,j}$为链路$\langle i,j \rangle$的理论带宽，

$L_{i,j}$为链路$\langle i,j \rangle$上的延迟，

E_p 为通信协议的有效承载效率，典型值为 0.3。

延时和带宽的计算可以在两台设备间构建简单的客户端和服务端通信，通过发送固定字节的数据（如 1MB）时间得到延时，字节数和延时之比得到带宽。模型的输出通过维度大小可以直接计算，如一个[1,1,4096]维度的输出总字节数为 1*1*4096*4=16384 字节。

抖动和丢包率属于通信的惩罚项，由T_{QoS}链路质量表示。其中，抖动为数据传输最大延时和最小延时之差;丢包率是无线网络通信中的一个随机故障，可以由二项分布评估。二项分布的数学期望可以看作传输时间段内发生的传输故障如丢包的平均次数,而方差可以用来评估预测的不确定性。则有：

$$T_{QoS} = \sum\sum w_{q1} \cdot Jitter_{i,j} + w_{q2} \cdot t_{i,j} \cdot PLR_{i,j} + w_{q3} \cdot PLR_{i,j}^2$$

其中，$Jitter, t, PLR$分别为对应链路$\langle i,j \rangle$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_{q1}, w_{q2}, w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_{q1} = 10$，$w_{q2} = 1$，$w_{q3} = 10000$；

因此，最小推理时间表达式为：

$$minimize\left(T_{compute} + T_{data} + T_{complexity} + T_{QoS}\right)$$

其中，$T_{compute} = \sum\sum \epsilon_{i,j} \cdot x_{i,j}$；

$T_{data} = \sum\sum \left(L_{i,j} + t_{i,j}\right)$；

$T_{complexity} = w_c \cdot \sum\sum x_{i,j}$；

$T_{QoS} = \sum\sum w_{q1} \cdot Jitter_{i,j} + w_{q2} \cdot t_{i,j} \cdot PLR_{i,j} + w_{q3} \cdot PLR_{i,j}^2$；

$$\epsilon_{i,j} = \frac{NumFlop_{modj}}{FLOP/s_i}$$

$$t_{i,j} = \frac{O_{i,j}}{E_p \cdot B_{i,j}}$$

其中，$T_{complexity}$为系统复杂度惩罚项；

T_{QoS}为链路质量惩罚项；

$Jitter, t, PLR$分别为对应链路$\langle i,j \rangle$的抖动、传输时间和丢包率；

i, j为对应通信节点的编号；

w_c、w_{q1}、w_{q2}、w_{q3}分别为对应项的权重，根据对系统复杂性和网络质量的要求来指定，典型值可以为$w_c = 1, w_1 = 10$，$w_2 = 1$，$w_3 = 10000$；

E_p为通信协议的有效承载效率，典型值为 0.3；

$O_{i,j}$为在链路$\langle i,j\rangle$上的输出数据量；

$B_{i,j}$为链路$\langle i,j\rangle$的理论带宽；

$L_{i,j}$为链路$\langle i,j\rangle$上的延迟；

$NumFlop_{mod_j}$ 为子模型j所需要进行浮点运算次数；

$FLOP/s_i$为设备i每秒浮点运算次数；

$x_{i,j} \in \{0,1\}$，为子模型 j 在设备 i 上的分配。

关于线性规划的约束项，根据设备的最大内存和模型的参数量，采用如下显式约束项：

$$\forall i \in \{0,\dots,m-1\} \sum_{j=0}^{n-1} M_{mod_j} \cdot x_{i,j} \leq \beta \cdot M_{device_i}$$

其中，m 和 n 分别代表设备数和子模型数，

M_{mod} 和 M_{device} 分别表示模型的参数量和设备的最大内存，

β 是设备能够分配给模型推理的最大内存比例，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

显式约束项的目的在于让一个设备的模型在全部加载时不会超过给定的设备模型的最大内存可使用量。

同时，考虑到变量在实际场景中的可部署性和运行推理的流水线模式，采用如下隐式约束项：

$$\sum_{i=0}^{m-1} \sum_{j=0}^{n-1} x_{i,j} = n$$

$$x[0,0] = 1$$

$$\forall j \in \{0, \ldots, n-1\} \sum_{i=0}^{m-1} x_{i,j} = 1$$

其中，m 和 n 分别代表设备数和子模型数，

$x_{i,j} \in \{0,1\}$为子模型 j 在设备 i 上的分配。

以上约束的目的是让所有模型完整的部署在不同设备上，且第一个 decoder 模型永远在第一个设备中，每一个模型不会被重复部署。

本发明步骤 3 的目的是将步骤 2 中切割的若干子模型进行分配，以分配到最合适的手机端上进行计算。

步骤 4，建立手机的客户端和服务端，运行子模型分配模型进行推理；

将手机分为客户端和服务端两类，客户端载入 embedding 嵌入层和下游任务层，控制子模型的输入和输出，服务端载入 decoder 解码器层，完成流水线式推理计算。

本发明通过对子模型的参数进行线性规划，得到子模型分配模型；该子模型分配模型能够使边缘算力参与到大语言模型的应用中来，有利于边缘算力的发展。

显然，本领域的技术人员可以对本发明进行各种改动和变形，而不脱离本发明的精神和范围。这样，倘若本发明的这些修改属于本发明权利要求及其同等技术的范围之内，则本发明也意图包含这些改动和变形在内。

说　　明　　书　　附　　图

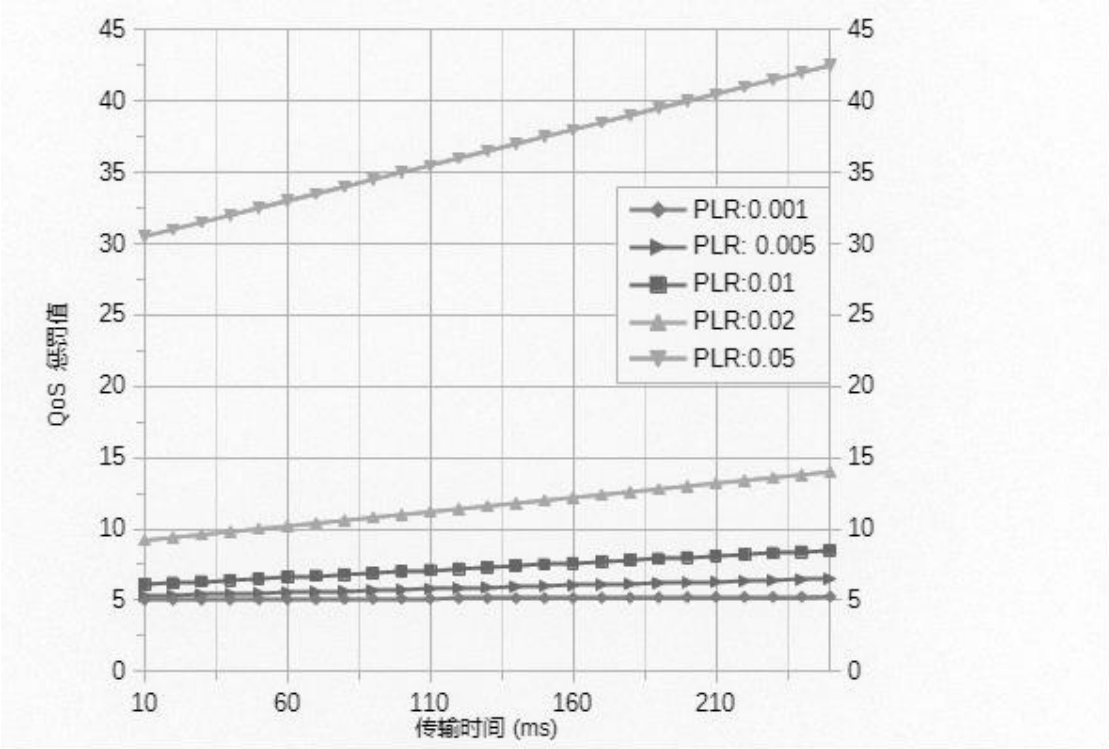

45
40
35
30
25
20
15
10
5
0
10
60
110
160
210
传输时间 (ms)
QoS 惩罚值
PLR:0.001
PLR: 0.005
PLR:0.01
PLR:0.02
PLR:0.05

图 1

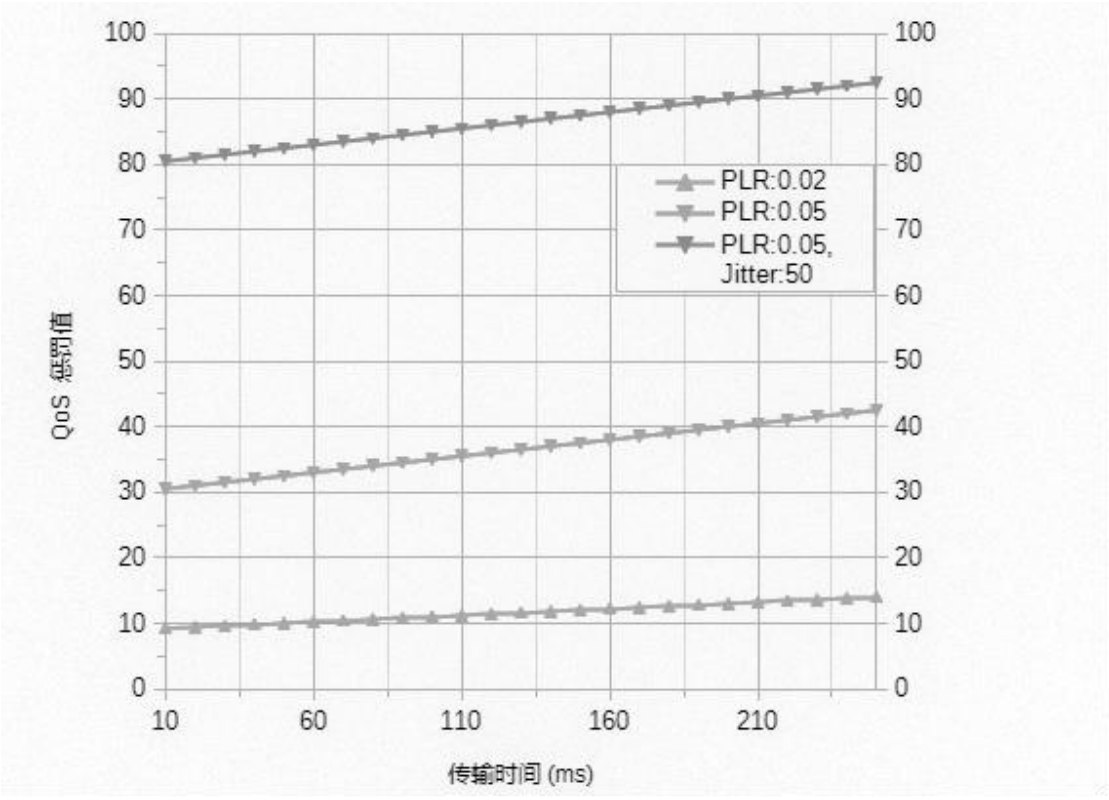
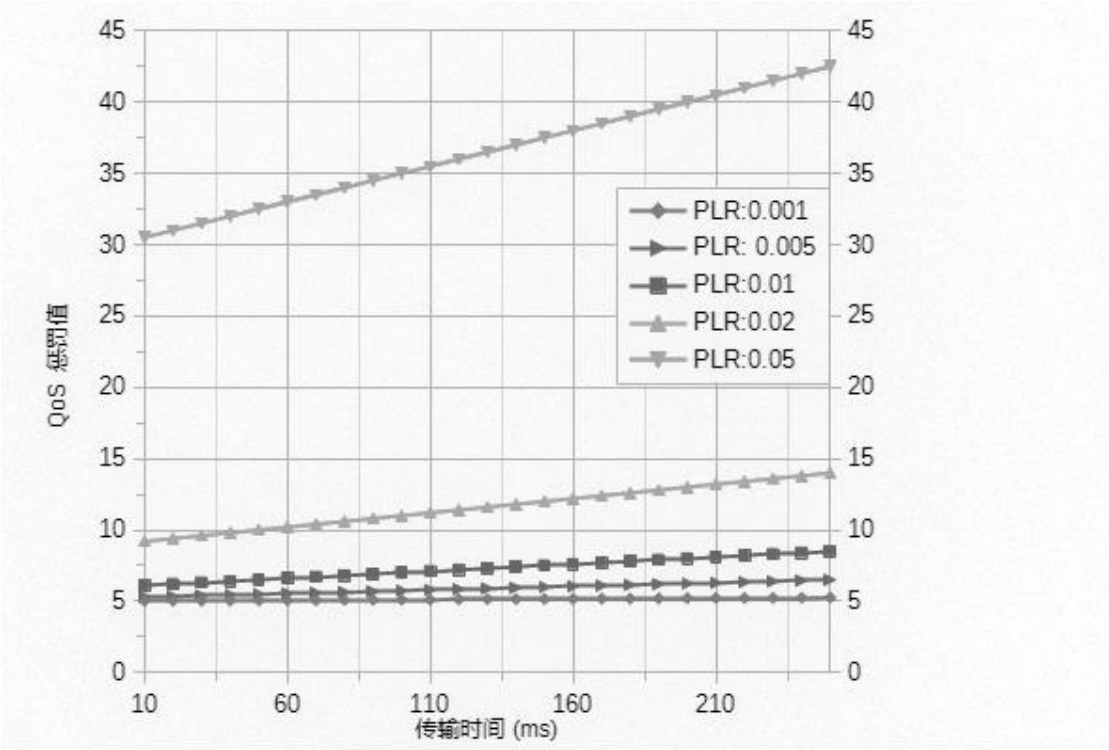

100
90
80
70
60
50
40
30
20
10
0
10
60
110
160
210
传输时间 (ms)
QoS 惩罚值
PLR:0.02
PLR:0.05
PLR:0.05,
Jitter:50

图 2

说　明　书　摘　要

本发明公开了一种用于大语言模型分布式推理的计算任务分配系统，包括环境部署模块、模型分割模块、计算任务分配模块和子模型运行模块。本发明通过模型拆分、流水线式推理和线性规划，能够在手机端实现分布式的运行大语言模型进行推理等工作，充分调用边缘计算力，从而降低通信成本。本发明还公开了一种用于大语言模型分布式推理的计算任务分配方法。

摘 要 附 图


45
40
35
30
25
20
15
10
5
0
10
60
110
160
210
传输时间 (ms)
QoS 惩罚值
PLR:0.001
PLR: 0.005
PLR:0.01
PLR:0.02
PLR:0.05