(19)中华人民共和国国家知识产权局

(12)发明专利申请

(10)申请公布号 CN 113330457 A
(43)申请公布日 2021.08.31

(21)申请号 202080010238.6
(22)申请日 2020.01.23
(30)优先权数据
62/796,038 2019.01.23 US
62/815,908 2019.03.08 US
(85)PCT国际申请进入国家阶段日
2021.07.21
(86)PCT国际申请的申请数据
PCT/US2020/014842 2020.01.23
(87)PCT国际申请的公布数据
WO2020/154538 EN 2020.07.30
(71)申请人 谷歌有限责任公司
地址 美国加利福尼亚州
(72)发明人 雅各布·D·乌斯克雷特
米切尔·托马斯·斯特恩
杰米·瑞安·基罗斯 威廉·常
(74)专利代理机构 中原信达知识产权代理有限责任公司 11219
代理人 邓聪惠 周亚荣
(51)Int.Cl.
G06N 3/04(2006.01)
G06N 3/08(2006.01)

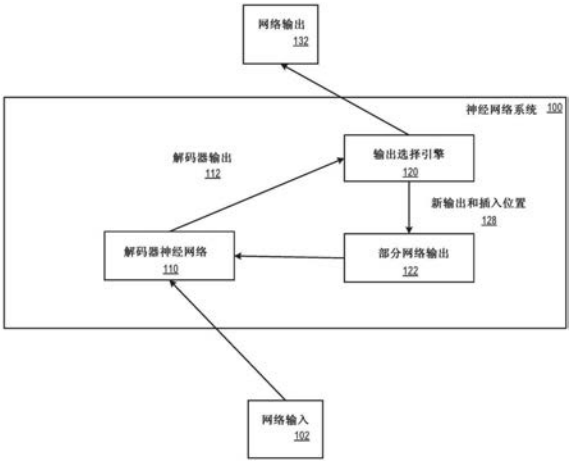
权利要求书3页 说明书11页 附图4页

(54)发明名称

使用插入操作生成神经网络输出

(57)摘要

用于使用插入操作生成网络输出的方法、系统和装置,包括在计算机存储介质上编码的计算机程序。



CN 113330457 A

1.一种由一个或多个计算机执行的方法,所述方法包括:

接收网络输入;以及

从所述网络输入生成网络输出,其中,所述网络输出包括来自根据输出顺序排列的输出的词汇表的多个输出,所述生成包括,在多个生成时间步中的每一个:

识别截至所述生成时间步已经被生成的当前部分网络输出,所述当前部分网络输出包括来自根据部分输出顺序排列的所述输出的词汇表的零个或更多个输出;

使用解码器神经网络以(i)所述网络输入的至少一部分和(ii)所述当前部分网络输出中的任何输出为条件,生成解码器输出,所述解码器输出为多个插入位置中的每一个定义输出的词汇表上的相应得分分布,其中,每个插入位置是所述部分输出顺序中的不同的新位置,在所述新位置处,没有所述当前部分网络输出中的输出;

使用所述解码器输出选择插入位置中的一个或多个,并且对于每个选择的插入位置,从所述词汇表中选择插入的输出;以及

生成新的部分网络输出,所述新的部分网络输出包括(i)所述当前部分网络输出中的零个或更多个输出以及(ii)对于每个选择的插入位置,在所述部分输出顺序中的对应新位置处插入的来自所述词汇表的插入的输出。

2.根据权利要求1所述的方法,其中,所述解码器神经网络是基于注意的神经网络,所述基于注意的神经网络被配置为通过将注意机制应用在所述网络输入的编码的表示上并且将自注意机制应用在所述当前部分网络输出中的输出上来生成所述解码器输出。

3.根据权利要求2所述的方法,其中,使用所述解码器神经网络生成所述解码器输出包括:

生成解码器输入,所述解码器输入包括所述网络输入的编码的表示和根据所述部分输出顺序排列的所述当前部分网络输出中的输出。

4.根据权利要求3所述的方法,

其中,生成解码器输入进一步包括将两个标记输出添加到所述当前部分网络输出,

其中,所述解码器神经网络被配置为在两个标记输出已经被添加之后,生成所述部分输出顺序中的每个位置的相应的表示向量,以及

其中,生成所述解码器输出包括:

通过级联所述部分输出顺序中的每个相邻位置对的表示向量,生成每个插入位置的相应插槽表示;以及

至少从以下来生成每个插入位置的得分分布:该插入位置的插槽表示。

5.根据任一前述权利要求所述的方法,其中,所述词汇表包括序列结束令牌。

6.根据权利要求5所述的方法,其中,使用所述解码器输出选择插入位置中的一个或多个,并且对于每个选择的插入位置,从所述词汇表中选择插入的输出包括:

确定具有所有插入位置-输出组合中的最高得分的插入位置-输出组合不包括所述序列结束令牌;以及

作为响应,仅选择具有所有插入位置-输出组合中的最高得分的插入位置-输出组合。

7.根据权利要求5所述的方法,其中,使用所述解码器输出选择插入位置中的一个或多个,并且对于每个选择的插入位置,从所述词汇表中选择插入的输出包括:

确定存在具有最高得分的输出不是所述序列结束令牌的至少一个插入位置;以及

作为响应,仅选择具有包括具有最高得分的输出不是序列结束令牌的插入位置的所有插入位置-输出组合中的最高得分的插入位置-输出组合。

8.根据权利要求5所述的方法,其中,使用所述解码器输出选择插入位置中的一个或多个,并且对于每个选择的插入位置,从所述词汇表中选择插入的输出包括:

从所述解码器输出并且对于每个插入位置,识别具有所述插入位置的最高得分的输出;

确定存在具有最高得分的输出不是序列结束令牌的至少一个插入位置;以及

作为响应,选择具有最高得分的输出不是序列结束令牌的每个插入位置,并且选择具有所述插入位置的最高得分的对应输出。

9.根据任一前述权利要求所述的方法,其中,所述解码器神经网络被配置为生成每个插入位置的相应的插槽表示。

10.根据权利要求9所述的方法,其中,生成所述解码器输出包括:

使用投影矩阵投影由所述插槽表示生成的解码器隐藏状态矩阵,以生成内容-位置对数矩阵;

将所述内容-位置对数矩阵展平为内容-位置对数向量;以及

将softmax应用在所述内容-位置对数向量上以生成所有插入位置-输出组合上的概率分布。

11.根据权利要求9所述的方法,其中,生成所述解码器输出包括:

通过将softmax应用于从所述插槽表示生成的解码器隐藏状态矩阵与所学习的查询向量的乘积,生成每个位置的相应的概率;

对于每个位置:

使用投影矩阵,将所述位置的插槽表示投影到得分向量中,所述得分向量包括所述词汇表中的每个输出的相应得分;

将softmax应用在所述得分向量上以生成所述词汇表中的每个输出的初始概率;以及

将每个初始概率乘以所述位置的概率,以生成所述词汇表中的每个输出的最终概率。

12.根据权利要求9-11中的任一项所述的方法,其中,生成所述解码器输出包括:

通过将最大池化应用在所述插槽表示上来生成上下文向量;

从所述上下文向量生成偏差向量,所述偏差向量包括所述词汇表中的每个输出的相应偏差值;以及

从所述偏差向量和所述插槽表示生成所述解码器输出。

13.根据任一前述权利要求所述的方法,其中:

所述方法包括语音识别方法,其中,所述网络输入包括表示所说的话语的音频数据的序列,并且所述网络输出包括所说的话语的转录;或者

所述方法包括医学诊断方法,其中,所述网络输入包括来自电子医疗记录的数据的序列,并且所述网络输出包括一个或多个预测的治疗的序列;或者

所述方法包括图像处理方法,其中,所述网络输入包括图像的像素值的序列,并且所述网络输出包括描述所述图像的文本的序列;或者

所述方法包括神经机器翻译方法,其中,所述网络输入包括第一语言的单词的序列,并且所述网络输出包括表示所述第一语言的单词的序列的第二语言的单词的序列,所述第一

语言与所述第二语言彼此不同。

14.一个或多个存储指令的计算机可读存储介质,所述指令当由一个或多个计算机执行时,使所述一个或多个计算机执行任一前述权利要求的方法中的任何一个的相应操作。

15.一种包括一个或多个计算机以及一个或多个存储设备的系统,所述一个或多个存储设备存储指令,所述指令当由一个或多个计算机执行时,使所述一个或多个计算机执行权利要求1-13中的任一项所述的方法中的任一个的相应操作。

使用插入操作生成神经网络输出

[0001] 相关申请的交叉引用

[0002] 本申请要求2019年1月23日提交的美国临时专利申请No.62/796,038和2019年3月8日提交的美国临时专利申请No.62/815,908的优先权。上述申请的每一个的全部内容均通过引用并入本文。

技术领域

[0003] 本说明书涉及使用神经网络生成输出。

背景技术

[0004] 神经网络是采用一个或多个非线性单元层以预测接收到的输入的输出的机器学习模型。除了输出层之外,一些神经网络还包括一个或多个隐藏层。每个隐藏层的输出被用作网络中的下一层,即,下一隐藏层或输出层,的输入。网络的每个层根据相应参数集的当前值从接收到的输入生成输出。

发明内容

[0005] 本说明书描述了一种被实现为一个或多个位置中的一个或多个计算机上的计算机程序的生成网络输出,该网络输出包括来自在输出顺序中的多个位置中的每一个的输出词汇表的相应输出。在一些情况下,输出是一维序列,例如文本序列,而在其他情况下,输出是高维数组,例如图像。

[0006] 能够实现本说明书中描述的主题的具体实施例以实现以下优点中的一个或多个。

[0007] 已经示出自回归模型在各种输出生成任务上实现高质量的性能,例如语音识别、机器翻译、图像生成等。然而,自回归模型需要在多个时间步的每一个将新输出添加到当前输入序列的末尾。另一方面,所描述的技术允许新输出被添加在当前输入序列内的任意位置,并且在一些情况下,允许多个输出在单个时间步被添加在多个不同位置。

[0008] 当在每个时间步仅添加单个输出时,向神经网络提供的以选择单个输出被添加的位置的额外的灵活性提高了传统的自回归模型的性能,而不增加所需的生成时间步的数量。

[0009] 当能够在每个时间步添加多个输出时,因为减少了需要执行处理的生成时间步的数量,所描述的技术允许比由自回归模型生成输出更快地生成输出(同时使用更少的计算资源),而没有显著降低(并且在某些情况下,有增加)输出生成质量。换句话说,通过在同一时间步并行生成多个不同的输出,系统能够在比常规系统更少的生成时间步上生成输出并且使用更少的计算资源,同时仍然生成在质量上与这些传统系统可比较的网络输出。

[0010] 在附图和以下描述中阐述了本说明书中描述的主题的一个或多个实施例的细节。主题的其他特征、方面和优点根据说明书、附图和权利要求将变得显而易见。

附图说明

[0011] 图1示出了示例神经网络系统。

[0012] 图2图示了使用神经网络系统生成示例网络输出。

[0013] 图3是用于生成网络输出的示例过程的流程图。

[0014] 图4是用于生成得分分布的示例过程的流程图。

[0015] 各个附图中相同的附图标记和名称表示相同的元件。

具体实施方式

[0016] 本说明书描述了一种被实现为一个或多个位置中的一个或多个计算机上的计算机程序的生成网络输出的系统，该网络输出包括来自按输出顺序在多个位置中的每一个的输出词汇表的相应输出。在一些情况下，输出是一维序列，例如文本序列，而在其他情况下，输出是高维数组，例如图像。输出的词汇表能够包括当执行机器学习任务以生成网络输出时能够被选择的每个可能的输出。

[0017] 例如，系统可以是神经机器翻译系统。即，如果网络输入是原始语言的单词的序列，诸如句子或短语，网络输出可以是网络输入到目标语言的翻译，即表示原始语言中的单词序列的目标语言中的单词序列。

[0018] 作为另一示例，系统可以是语音识别系统。即，如果网络输入是表示所说话语的音频数据的序列，网络输出可以是表示话语的字素、字符或单词的序列，即是网络输入的转录。

[0019] 作为另一示例，系统可以是自然语言处理系统。例如，如果网络输入是原始语言中的单词的序列，例如，句子或短语，网络输出可以是原始语言中的网络输入的摘要，即具有少于网络输入的单词但保留了网络输入的基本含义的序列。作为另一示例，如果网络输入是形成问题的单词的序列，网络输出能够是形成问题的答案的单词的序列。

[0020] 作为另一示例，系统可以是计算机辅助的医学诊断系统的一部分。例如，网络输入能够是来自电子病历的数据序列，并且网络输出能够是预测的治疗的序列。

[0021] 作为另一示例，系统可以是图像处理系统的一部分。例如，网络输入能够是图像，即来自图像的颜色值的序列，并且输出能够是描述图像的文本的序列。作为另一示例，网络输入能够描述图像的上下文，例如，是文本的序列，并且网络输出能够是描述上下文的图像。

[0022] 图1示出了示例性神经网络系统100。神经网络系统100是被实现为一个或多个位置中的一个或多个计算机上的计算机程序的系统的示例，其中，以下系统、组件和技术被实现。

[0023] 系统100接收网络输入102、处理网络输入102以生成网络输入102的网络输出132。

[0024] 具体地，系统100在多个生成时间步上生成网络输出132。

[0025] 在每个生成时间步，系统100以网络输入102和截至生成时间步已经被生成的当前部分网络输出122为条件从要被添加到网络输出的词汇表中选择一个或多个新输出128。

[0026] 截至任何给定的生成时间步，当前部分网络输出122具有来自根据部分输出顺序排列的输出的词汇表的零个或多个输出。换句话说，在第一生成时间步，当前部分网络输出122是空的，即具有零个输出，并且，在所有其他生成时间步，当前部分网络输出122具有在

先前生成时间步被先前添加的输出。

[0027] 更详细地，在每个生成时间步，系统100识别截至生成时间步时已经被生成的当前部分网络输出122。

[0028] 然后，系统100执行一个或多个插入操作以将一个或多个新输出128添加到当前部分网络输出122。

[0029] 具体地，系统100使用以(i)网络输入102的至少一部分和(ii)当前部分网络输出122中的输出为条件的解码器神经网络110，生成解码器输出112。

[0030] 解码器输出112为多个插入位置中的每一个定义输出词汇表上的相应得分分布。

[0031] 每个插入位置是部分输出顺序中没有当前部分网络输出中的输出的不同的新位置，即(i)在当前部分网络输出中的所有输出之前、(ii)在当前部分网络输出中的两个输出之间，或(iii)当前部分网络输出中的所有输出之后能够被添加到部分输出顺序的新位置。

[0032] 对于第一生成时间步，可能仅存在将第一输出添加到当前部分网络输出122的单个插入位置，即，因为对于第一生成时间步，当前部分网络输出122是空的并且没有已经在当前部分网络输出122中的输出。

[0033] 然后，系统100内的输出选择引擎120使用解码器输出112选择一个或多个插入位置，并且对于每个选择的插入位置，从词汇表中选择插入的输出，即，选择一个或多个新的输出128和一个或多个插入位置128。

[0034] 然后，系统100通过生成新的部分网络输出来更新当前部分网络输出122，该新的部分网络输出包括(i)当前部分网络输出122中的任何输出以及(ii)对于每个选择的插入位置，插入在部分输出顺序中的相应新位置处的来自词汇表的插入的输出。

[0035] 在一些实施方式中(被称为“贪婪解码”)，引擎120在每个生成时间步仅选择单个插入位置，即，将单个输出添加到当前输出。在这些实施方式中，因为引擎120选择单个输出将被添加的插入位置，而不是如传统系统所做的那样将单个输出自动地添加到当前输出122的末尾，所以最终输出的质量相对于传统系统能够被提高，即，因为当将输出添加到当前输出122的末尾是不适的当或次优的时，网络输出能够根据更灵活的排序被生成。

[0036] 在一些其他实施方式中(被称为“并行解码”)，引擎120可以在任何给定生成时间步选择多个插入位置，即，将多个输出添加到当前部分输出122。在这些实施方式中，因为与传统系统不同，多个输出能够在单个生成时间步被添加，所以生成网络输出132所需的时间步的数量能够被大大地减少，因此生成网络输出132所需的时间和计算资源量能够被大大地减少。

[0037] 下面参考图3更详细的描述使用贪婪解码和并行解码两者来选择插入位置和输出。

[0038] 系统100能够在生成时间步持续添加输出，直到已经达到终止标准，即，直到满足包含词汇表中的序列结束令牌的某些标准，如下面参考图3更详细地描述。

[0039] 也就是说，在每个生成时间步，系统100使用解码器输出112检查，以查看是否满足标准，并且如果不满足标准，仅将新输出128添加到当前部分输出122。

[0040] 如果满足标准，系统100确定当前部分输出122是要被生成的最终输出，不再向当前部分输出122添加任何输出，并且输出当前部分输出122作为最终网络输出132。

[0041] 在一些情况下，解码器神经网络110是基于注意的解码器神经网络，其将注意应用

在网络输入的编码的表示上，即，由编码器神经网络(其能够是系统100的一部分或在网络输入被提供到系统100之前对网络输入102进行编码的外部系统)生成的，并且将自注意力应用在当前部分输出122中的输出上。

[0042] 这种神经网络在2018年5月23日提交的PCT申请No.PCT/US2018/034224中被描述，其全部内容通过引用整体并入本文。一些基于注意的解码器将因果掩码自注意应用在当前部分输出中的输出上，以防止任何给定位置的输出受到在未来位置的输出的影响。因为系统100能够在任何插入位置插入输出，与这些其他基于注意的解码器不同，解码器神经网络110不应用来自解码器的因果自注意掩码，使得所有位置能够顾及所有其他位置，而不是仅当前位置的左侧的那些位置。这允许每个决策以在任何给定的生成时间步的当前部分输出的完整上下文为条件。下面参考图4描述当生成插入操作时可以提高神经网络的性能的对这些神经网络的其他修改。

[0043] 然而，在其他情况下，解码器神经网络110是将网络输入映射到网络输出的不同类型的神经网络。

[0044] 例如，解码器110能够是递归神经网络，该递归神经网络通过递归状态以当前部分输出122为条件，并且将注意应用在由编码器神经网络生成的网络输入的编码的表示上。

[0045] 作为另一示例，解码器110可以是卷积神经网络，该卷积神经网络接收当前部分输出作为输入并且具有以网络输入的表示为条件的一个或多个卷积神经网络层。

[0046] 图2示出了使用神经网络系统100生成示例性网络输出。在图2的示例中，正在被生成的网络输出是[三个、朋友、吃、午餐、一起]([three,friends,ate,lunch,together])，其中，输出中的每个单词都是从单词词汇表中选择的。

[0047] 图2的区段210图示了在列“t”中所示的六个时间步0-5上使用“贪婪”解码的网络输出的生成。在标记为“画布”(“Canvas”)的列中示出截至六个时间步中的任何一个的当前部分输出，而在列“插入”(“Insertion”)中示出在时间步执行的插入操作。因此，如从区段210能够看出，在贪婪解码中，系统在每个时间步将单个输出添加到当前部分输出。然而，与传统系统不同，例如，在时间步2，单词“朋友”(“friends”)被添加在位置0，即，被添加到部分输出[ate,together]的前面。

[0048] 图2的区段220图示了在四个时间步0-4上使用“并行解码”的网络输出的生成。从区段220能够看出，在并行解码中，系统能够在给定时间步将多个输出添加到当前部分输出。例如，在时间步1，“朋友”(“friends”)被添加在插入位置0，被添加到当前部分输出的前面，而“一起”(“together”)被添加到插入位置1。作为另一示例，在时间步2，“三个”(“three”)被添加在插入位置0，而“午餐”(“lunch”)被添加在插入位置2，即，在在单词“吃”(“ate”)与“一起”(“together”)之间。

[0049] 图3是用于从网络输入生成网络输出的示例过程300的流程图。为方便起见，过程300将被描述为由位于一个或多个位置的一个或多个计算机的系统执行。例如，适当编程的神经网络系统，例如图1的神经网络系统100能够执行过程300。

[0050] 系统能够在多个生成时间步中的每一个执行过程300以从网络输入生成网络输出。具体地，系统持续执行过程300，直到在步骤306满足终止标准。

[0051] 系统识别截至生成时间步已经被生成的当前部分网络输出(步骤302)。

[0052] 当前部分网络输出具有来自根据部分输出顺序排列的输出词汇表的零个或多个

输出。换句话说，在第一生成时间步，部分输出是空的，即，具有0个输出，并且在所有其他生成时间步，部分输出具有在先前的生成时间步被先前添加的输出。

[0053] 系统使用解码器神经网络并以(i)网络输入的至少一部分和(ii)当前部分网络输出中的任何输出为条件，生成解码器输出(步骤304)。

[0054] 解码器输出为多个插入位置中的每一个定义输出的词汇表上的相应得分分布。每个插入位置是当前部分网络中没有输出的部分输出顺序中的不同的新位置。例如，在第一时间步，部分输出顺序中只有单个新位置，即，第一输出能够被添加到网络输出的位置。在其他生成时间步，新位置包括(i)当前部分网络输出中的任何输出之前的位置、(ii)当前部分网络输出中的所有输出之后的位置，以及(iii)如果当前部分网络输出中有多于一个网络输出，当前部分网络输出中每个连续网络输出对之间的相应位置。作为具体的示例，如果当前部分网络输出是[朋友、吃、一起]([friends,ate,together])，示出的带有新输出可以被插入的新位置(由“_”表示)的部分网络输出将是[_，朋友，_，吃，_，一起，_]([_，friends,_,ate,_,together,_])。

[0055] 下面参考图4更详细地描述生成解码器输出。

[0056] 然后，系统基于解码器输出确定网络输出是否应当被终止(步骤306)，如果不，使用解码器输出选择一个或多个插入位置，并且对于每个选择的插入位置从词汇表选择插入的输出(步骤308)。

[0057] 也就是说，如果系统确定网络输出应当被终止，系统使用当前部分输出作为最终网络输出。如果系统确定网络输出不应当被终止，系统选择要被添加到当前部分输出的一个或多个输出。

[0058] 系统确定网络输出是否应当被终止的方式，以及如果不终止，系统选择插入位置的方式取决于系统被配置为执行贪婪解码还是并行解码。

[0059] 如上所述，在贪婪解码中，系统在每个生成时间步仅选择单个插入位置，即将单个输出添加到当前输出。

[0060] 在系统执行贪婪解码的一些实施方式中，当具有包括作为组合中的输出的序列结束令牌的所有插入位置-输出组合中的最高得分的插入位置-输出组合时，系统能够确定终止网络输出。通常，序列结束令牌是被添加到词汇表但将永远不被产生为网络输出的一部分并且仅由系统使用以确定何时终止网络输出的预定令牌。给定插入位置-给定输出组合的得分是指如由解码器输出定义的给定插入位置的得分分布中的给定输出的得分。

[0061] 在这些实施方式中，当系统确定具有所有插入位置-输出组合中的最高得分的插入位置—输出组合不包括序列结束令牌时，系统仅选择具有所有插入位置-输出组合中的最高得分的插入位置—输出组合。

[0062] 换句话说，系统选择具有所有插入位置-输出组合中的最高得分的插入位置-输出组合作为插入位置，并且从具有所有插入位置-输出组合中的最高得分的插入位置-输出组合选择输出作为选择的插入位置的插入的输出。

[0063] 在系统执行贪婪解码的其他实施方式中，仅当具有最高得分的输出是所有插入位置的序列结束令牌插入时，系统能够确定终止网络输出。

[0064] 换句话说，无论何时存在具有最高得分的输出不是序列结束令牌的至少一个插入位置，系统确定不终止网络输出。

[0065] 在这些实施方式中，响应于确定不终止，系统仅选择具有所有插入位置-输出组合中的最高得分的插入位置-输出组合。

[0066] 如上所述，在并行解码中，系统能够在任何给定的生成时间步选择多个插入位置，即将多个输出添加到当前部分输出。

[0067] 当执行并行解码时，系统从解码器输出和每个插入位置识别具有插入位置的得分分布中的最高得分的输出。然后，仅当序列结束令牌是具有所有插入位置的得分分布中的最高得分的输出时，系统确定终止网络输出。

[0068] 换句话说，当存在具有最高得分的输出不是序列结束令牌的至少一个插入位置时，系统确定不终止。

[0069] 响应于确定不终止，系统选择具有最高得分的输出不是序列结束令牌的每个插入位置。对于每个选择的位置，系统然后选择具有选择的插入位置的最高得分的相应输出。因此，当存在具有最高得分的输出是除序列结束令牌以外的输出的多个插入位置时，系统在生成时间步选择多个插入位置。

[0070] 然后，系统生成新的部分网络输出，其包括(i)当前部分网络输出中的任何输出以及(ii)对于每个选择的插入位置，插入在部分输出顺序中的相应新位置的来自词汇表的插入的输出(步骤310)。

[0071] 图4是用于生成解码器输出的示例过程400的流程图。为方便起见，过程400将被描述为由位于一个或多个位置的一个或多个计算机的系统执行。例如，适当编程的神经网络系统，例如图1的神经网络系统100，能够执行过程400。

[0072] 系统生成生成时间步的解码器输入(步骤402)。如上所述，解码器输入使神经网络以截至时间步的当前部分网络输出和网络输入为条件。

[0073] 特别地，如上所述，在一些实施方式中，解码器神经网络是自注意解码器神经网络或自回归卷积解码器神经网络，其接收包括根据部分输出顺序排列的当前部分网络输出中的输出和网络输入的编码的表示的解码器输入。

[0074] 在这些实施方式中，系统能够通过将两个标记输出添加到部分输出顺序中的预定位置中的当前部分网络输出，例如，在部分输出顺序中的第一输出之前以及在部分输出顺序中的最后一个输出之后，增大当前部分网络输出，作为生成解码器输入的一部分。标记输出是永远不会作为网络输出的一部分被发出并且仅被用于增大部分网络输出的预先确定的输出。

[0075] 系统使用神经网络处理解码器输入，以生成每个插入位置的相应插槽表示(步骤404)。

[0076] 特别地，在每个时间步，自注意解码器和自回归卷积解码器两者被配置为截至时间步的当前部分输出中的每个输出生成相应的表示向量。

[0077] 因为当前部分输出已经利用标记输出被增大，当当前部分输出包括N个输出时，解码器神经网络生成N+2个表示：一个用于N个输出中的每一个，并且一个用于标记输出中的每一个。

[0078] 此外，当有N个输出时，有N+1个可能的插入位置：(i)当前部分网络输出中的任何输出之前的一个位置，(ii)当前部分网络输出中的所有输出之后的一个位置，以及(iii)如果当前部分网络输出中有多于一个网络输出，当前部分网络输出中的每个连续网络输出对

之间的N-1个相应位置。

[0079] 为了生成N+1个插入位置的插槽表示，对于每个插入位置，系统通过级联部分输出顺序中的每个相邻位置对的表示向量来生成插入位置的相应插槽表示。例如，为了生成部分输出顺序中的第二与第三输出之间的插入位置的相应的插槽表示，系统将级联第二和第三输出的表示向量。

[0080] 系统至少从插槽表示生成得分分布(步骤406)。

[0081] 系统能够以多种方式中的任何一种从插槽表示生成得分分布。

[0082] 作为一个示例，系统能够使用投影矩阵将从插槽表示生成的解码器隐藏状态矩阵，即具有作为矩阵的行或列的插槽表示的矩阵，进行投影以生成内容-位置对数矩阵。该投影矩阵能够在解码器神经网络的训练期间被学习。

[0083] 然后，系统能够将内容-位置对数矩阵展平为内容-位置对数向量，并且在内容-位置对数向量上应用softmax以生成所有插入位置-输出组合上的概率分布。

[0084] 作为另一示例，通过将softmax应用于解码器隐藏状态矩阵与所学习的查询向量的乘积，即，当应用于解码器隐藏状态矩阵时，将隐藏状态矩阵映射到包括每个插入位置的相应值的向量的学习查询向量，系统能够生成每个插入位置的相应概率。

[0085] 对于每个插入位置，系统然后能够使用投影矩阵(其也在解码器训练期间被学习)将位置的插槽表示投影到包括词汇表中的每个输出的相应得分的得分向量中，并且将softmax应用在得分向量上以生成词汇表中的每个输出的初始概率。

[0086] 为生成每个插入位置的得分分布，系统然后将每个初始概率乘以位置的概率，以生成词汇表中的每个输出的最终概率。

[0087] 在这些示例中的任何一个中，系统还能够可选地合并偏差向量以增加跨插入位置共享的信息。具体地，系统能够通过将最大池化应用在插槽表示上来生成上下文向量，然后从上下文向量生成偏差向量，其中，偏差向量是包括词汇表中的每个输出的相应偏差值的向量。

[0088] 然后，系统能够从偏差向量和插槽表示生成解码器输出。例如，在对任何对数集计算softmax之前，系统能够将偏差向量添加到对数。通常，合并该偏差向量可能有助于向解码器神经网络提供覆盖信息，或有助于传播关于应当出现在网络输出中的多个位置的公共输出的计数信息。

[0089] 为了使系统被有效地用于生成网络输出，即，为生成系统能够使用以生成高质量网络输出的高质量解码器输出，系统基于训练数据训练神经网络以优化目标函数。

[0090] 能够被用于训练神经网络的目标函数的一个示例是，使用软顺序奖励框架以便训练解码器神经网络在生成网络输出时遵循“预言”(“oracle”)策略。具体地，对于将输出w插入到输出顺序中的位置i与位置j之间的插入位置s的在给定生成时间步的任何给定插入操作a，系统能够计算奖励值，该奖励值等于将动作映射到实数的阶函数的负数，越低的值对应于越好的动作。

[0091] 然后，系统能够使取决于(i)基于奖励值的生成时间步的oracle策略与(ii)在给定生成时间步由解码器神经网络生成的得分分布之间的KL散度的损失最小化。

[0092] 具体地，系统能够使用传统的基于梯度的神经网络训练技术来训练神经网络以使以下损失最小化：

[0093] $$R(a) = \begin{cases} -O(a) & \forall a \in A^* \\ -\infty & \forall a \notin A^* \end{cases}$$

[0094] $$q_{\text{oracle}}(a) = \frac{\exp(R(a)/\tau)}{\sum_{a' \in A^*} \exp(R(a')/\tau)}$$

[0095] $$\mathcal{L} = \text{KL}(q_{\text{oracle}} || p)$$

[0096] 其中，A*是在给定生成时间步的有效动作集，p是神经网络生成的得分分布，τ是恒温参数，0(a)是阶函数。如果导致来自实际网络输出的输出被插入到在生成时间步可用的插入位置中的一个的部分输出中，动作是在给定生成时间步的有效动作。通过使用不同的阶函数，系统能够训练神经网络根据各种不同的排序生成网络输出。在下面的1中示出了能够使用的阶函数的一些示例。

[0097]

Order Function $O(a)$
0
$\lvert s - (i+j)/2 \rvert$
$\text{rank}(\text{hash}(w))$
$\pm s$
$\pm\text{rank}(\text{frequency}(w))$
$\pm\text{rank}(\text{length}(w))$
$\pm\text{rank}(w)$
$\pm \log p(a)$

[0098] (Order Function阶函数)

[0099] 表1

[0100] 作为另一示例，系统能够训练神经网络以使特别鼓励神经网络以从左到右的方式产生其输出的损失函数最小化。在该示例中，损失函数能够是以下形式：

[0101] $$\text{loss}(x, \hat{y}) = -\log p(y_{k+1}, k \mid x, \hat{y}).$$

[0102] 其中，k是网络输入x在实际网络输出y中随机采样的位置，$\hat{y}$是包括实际网络输出的前k个输出的前缀，并且p是由神经网络分配给给定输出、插入位置对的得分。

[0103] 作为又一示例，系统能够训练神经网络以获得最大并行度，以鼓励生成网络输出中的平衡的二叉树排序来。在该示例中，损失函数能够是以下形式：

[0104] $$\text{slot-loss}(x, \hat{y}, l) = \sum_{i=i_l}^{j_l} -\log p(y_i, l \mid x, \hat{y}) \cdot w_l(i).$$

[0105] $$\text{loss}(x, \hat{y}) = \frac{1}{k+1} \sum_{l=0}^{k} \text{slot-loss}(x, \hat{y}, l).$$

[0106] 其中，$w_l(i)$是网络输出中的位置l的插入位置i的权重，该权重基于插入位置i与

来自尚未在位置l产生的目标输出的输出跨度的中心的距离。

[0107] 作为又一示例，系统能够训练神经网络以向每个正确的动作分配相等的概率质量，而没有对于网络输出中的哪些位置被首先生成的偏好。在该示例中，损失函数能够是以下形式：

[0108] $$\text{slot-loss}(x,\hat{y},l)=\frac{1}{j_l-i_l+1}\sum_{i=i_l}^{j_l}-\log p(y_i,l\mid x,\hat{y}).$$

[0109] $$\text{loss}(x,\hat{y})=\frac{1}{k+1}\sum_{l=0}^{k}\text{slot-loss}(x,\hat{y},l).$$

[0110] 其中，i_1是尚未在位置l产生的第一输出，并且j_1是尚未在位置l产生的最后一个输出。

[0111] 本说明书连同系统和计算机程序组件一起使用术语“被配置”。对于要被配置为执行特定操作或动作的一个或多个计算机的系统意味着系统已经在其上安装了在操作中使系统执行这些操作或动作的软件、固件、硬件或软件、固件、硬件的组合。对于要被配置为执行特定操作或动作的一个或多个计算机程序意味着一个或多个程序包括当由数据处理装置执行时使装置执行操作或动作的指令。

[0112] 本说明书中描述的主题和功能操作的实施例能够以数字电子电路、有形地体现的计算机软件或固件、包括本说明书中公开的结构及其结构等价物的计算机硬件或者它们中的一个或多个的组合而被实现。本说明书中描述的主题的实施例能够被实现为一个或多个计算机程序，即，编码在有形非暂时性存储介质上以用于由数据处理装置执行或者控制数据处理装置的操作的计算机程序指令的一个或多个模块。计算机存储介质能够是机器可读的存储设备、机器可读的存储基板、随机或串行访问存储设备或它们中的一个或多个的组合。可替代地或者另外，程序指令能够被编码在人工生成的传播信号上，例如，机器生成的电、光或电磁信号，传播信号被生成以对信息进行编码以用于传输到适合的接收器装置以用于由数据处理装置执行。

[0113] 术语“数据处理装置”表示数据处理硬件并且涵盖用于处理数据的所有种类的装置、设备和机器，例如包括可编程处理器、计算机或多个处理器或计算机。装置还能够是或者进一步包括专用逻辑电路，例如，FPGA(现场可编程门阵列)或ASIC(专用集成电路)。装置除了包括硬件之外还能够可选地包括创建计算机程序的执行环境的代码，例如，构成处理器固件、协议栈、数据库管理系统、操作系统或它们中的一个或多个的组合的代码。

[0114] 也可以被称为或者被描述为程序、软件、软件应用、app、模块、软件模块、脚本或代码的计算机程序能够以包括编译或解释语言或声明或过程语言的任何形式的编程语言被编写；并且它能够被以包括作为独立程序或者作为模块、组件、子例行程序或适合于在计算环境中使用的其它单元的任何形式被部署。程序可以但是不必须对应于文件系统中的文件。程序能够被存储在保持其它程序或数据的文件的一部分中，例如，存储在标记语言文档中的一个或多个脚本，在专用于所涉及的程序的单个文件中或者在多个协调文件中，例如，存储代码的一个或多个模块、子程序或部分的文件。计算机程序能够被部署成在一个计算机上或者在位于一个站点处或者分布在多个站点上并通过数据通信网络互连的多个计算机上被执行。

[0115] 在本说明书中，术语“数据库”被广泛地用于表示任何数据的集合：数据不需要被以任何特定方式被构造，或者根本不构造，并且它能够被存储在一个或多个位置中的存储设备上。因此，例如，索引数据库能够包括多个数据的集合，其每一个可以被不同地组织和访问。

[0116] 类似地，在本说明书中术语“引擎”被广泛地用于表示被编程为执行一个或多个特定功能的基于软件的系统、子系统或过程。通常，引擎将被实现为安装在一个或多个位置中的一个或多个计算机上的一个或多个软件模块或组件。在一些情况下，一个或多个计算机将被专用于特定引擎；在其它情况下，多个引擎能够在同一计算机或多个计算机上被安装并运行。

[0117] 本说明书中描述的过程和逻辑流程能够由执行一个或多个计算机程序的一个或多个可编程计算机执行以通过对输入数据进行操作并生成输出来执行功能。过程和逻辑流程还能够由例如FPGA或ASIC的专用逻辑电路执行，或者通过专用逻辑电路和一个或多个编程的计算机的组合来执行。

[0118] 适合执行计算机程序的计算机能够基于通用微处理器或专用微处理器或两者，或任何其它种类的中央处理器。通常，中央处理单元将从只读存储器或随机存取存储器或两者接收指令和数据。计算机的元件是用于执行或者实行指令的中央处理单元和用于存储指令和数据的一个或多个存储设备。中央处理单元和存储器能够由专用逻辑电路补充或者被并入专用逻辑电路中。通常，计算机还将包括或者可操作地被耦合，以从用于存储数据的一个或多个大容量存储设备，例如磁盘、磁光盘或光盘，接收数据或者将数据传送到一个或多个大容量存储设备，或者两者。然而，计算机不必须具有这种设备。此外，计算机能够被嵌入在另一设备中，例如，移动电话、个人数字助理(PDA)、移动音频或视频播放器、游戏控制器、全球定位系统(GPS)接收器或便携式存储设备，例如通用串行总线(USB)闪存驱动器，仅举几例。

[0119] 适合存储计算机程序指令和数据的计算机可读介质包括所有形式的非易失性存储器、介质和存储设备，例如包括半导体存储设备，例如，EPROM、EEPROM和闪速存储器设备；磁盘，例如，内部硬盘或可移动盘；磁光盘；以及CD ROM和DVD-ROM盘。

[0120] 为了提供与用户的交互，本说明书中描述的主题的实施例能够在计算机上被实现，计算机具有用于向用户显示信息的显示设备，例如，CRT(阴极射线管)或LCD(液晶显示器)监视器，以及用户能够通过其向计算机提供输入的键盘和定点设备，显示设备例如，鼠标或轨迹球。其它种类的设备也能够被用于提供与用户的交互；例如，提供给用户的反馈能够是任何形式的感官的反馈，例如视觉反馈、听觉反馈或触觉反馈；并且来自用户的输入能够以包括声、语音或触觉输入的任何形式被接收，。另外，计算机能够通过向由用户使用的设备发送文档并从由用户使用的设备接收文档来与用户交互；例如，通过响应于从网络浏览器接收请求向用户的设备上的网络浏览器发送网页。另外，计算机能够通过向例如正在运行消息传送应用的智能电话的个人设备发送文本消息或其它形式的消息并且反过来从用户接收响应消息来与用户交互。

[0121] 用于实现机器学习模型的数据处理装置还能够包括例如用于处理机器学习训练或生产，即推理、工作负载的公共和计算密集部分的专用硬件加速器单元。

[0122] 机器学习模型能够使用机器学习框架被实现和部署，例如，TensorFlow框架、

Microsoft Cognitive Toolkit框架、Apache Singa框架或Apache MXNet框架。

[0123] 本说明书中描述的主题的实施例能够被实现在计算系统中，计算系统包括后端组件，例如作为数据服务器；或者包括中间件组件，例如应用服务器；或者包括前端组件，例如具有用户能够通过其与本说明书中描述的主题的实现方式交互的图形用户界面、网络浏览器或app的客户端计算机；或者包括一个或多个这种后端、中间件或前端组件的任何组合。系统的组件能够通过例如通信网络的数字数据通信的任何形式或介质的来互连。通信网络的示例包括局域网(LAN)和广域网(WAN)，例如互联网。

[0124] 计算系统能够包括客户端和服务器。客户端和服务器一般地彼此远离并通常通过通信网络来交互。客户端与服务器的关系借助于在相应的计算机上运行并且具有彼此之间的客户端-服务器关系的计算机程序而产生。在一些实施例中，服务器向用户设备传输例如HTML页面的数据，例如，用于向与作为客户端的设备交互的用户显示数据并从该用户接收用户输入的目的。在用户设备处生成的数据，例如，用户交互的结果，能够在服务器处从设备被接收。

[0125] 虽然本说明书包含许多具体实施方式细节，但是这些不应当被解释为对任何发明的或可能要求保护的范围的限制，而是相反地被解释为对可能特定于特定发明的特定实施例的特征的描述。在本说明书中被描述的单独的实施例的上下文中的某些特征也能够在单个实施例中组合地被实现。相反地，在单个实施例的上下文中被描述的各种特征也能够单独地或者以任何适合的子组合在多个实施例中被实现。此外，尽管特征可能在上面被描述为以某些组合起作用并且甚至最初被如此要求保护，但是来自要求保护的组合的一个或多个特征在一些情况下能够从组合中被去除，并且所要求保护的组合可以针对子组合或子组合的变化。

[0126] 类似地，虽然操作在附图中被描绘并以特定次序在权利要求书中被记载，但是这不应当被理解为要求以所示的特定次序或者以顺序的次序执行这种操作，或者要求所有图示的操作被执行以实现期望的结果。在某些情况下，多任务处理和并行处理可以是有利的。此外，上述实施例中的各种系统模块和组件的分离不应当被理解为在所有实施例中要求这种分离，并且应当理解的是，所描述的程序组件和系统一般地能够被一起集成在单个软件产品中或者被封装到多个软件产品中。

[0127] 已经描述了主题的具体实施例。其它实施例在所附权利要求的范围内。例如，权利要求中记载的动作能够以不同的次序被执行并仍然实现期望的结果。作为一个示例，附图中描绘的过程不一定要求所示的特定次序或顺序的次序以实现所预期的结果。在一些情况下，多任务处理和并行处理可以是有利的。

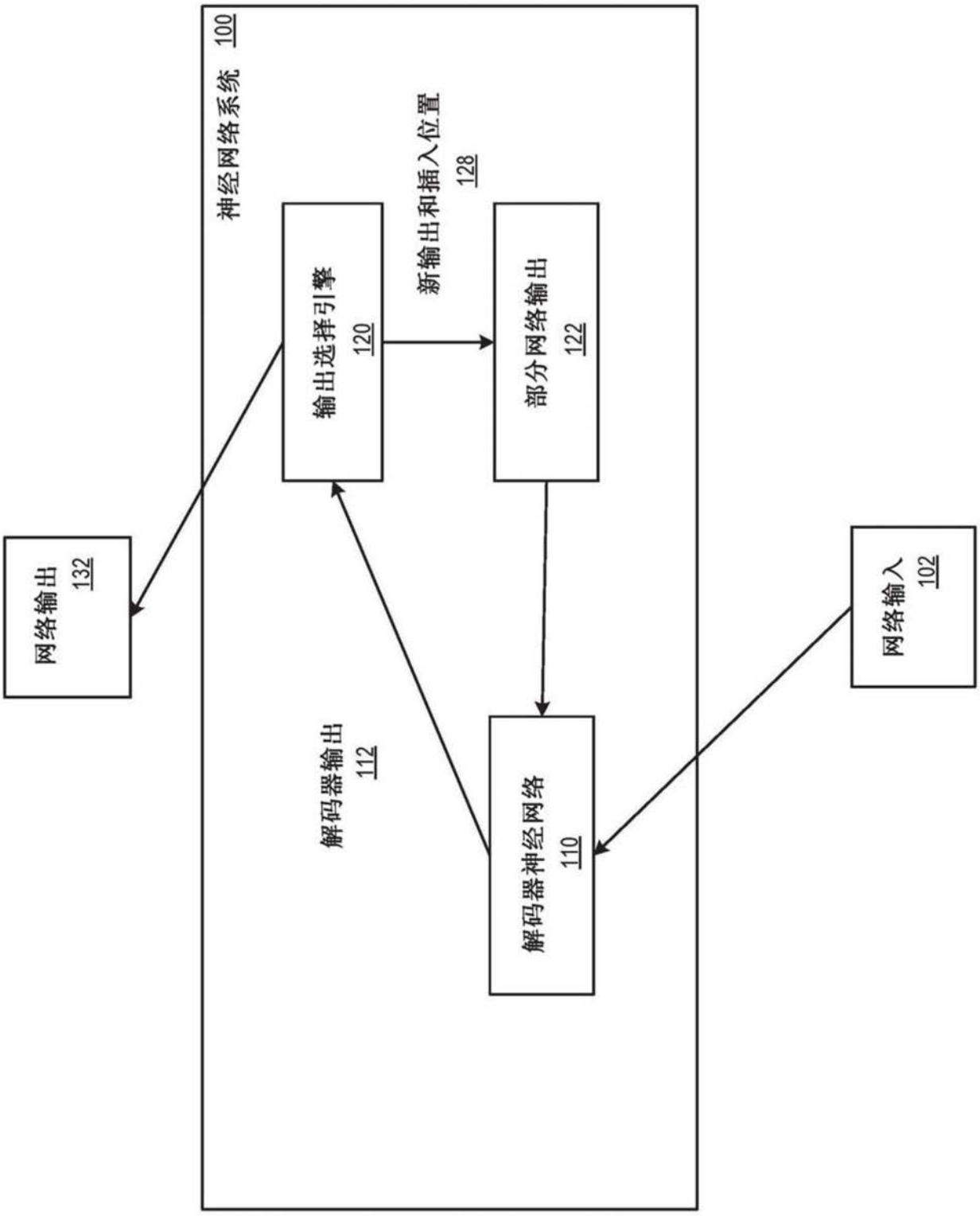

神经网络系统 100
输出选择引擎 120
新输出和插入位置 128
部分网络输出 122
网络输出 132
解码器输出 112
解码器神经网络 110
网络输入 102

图1

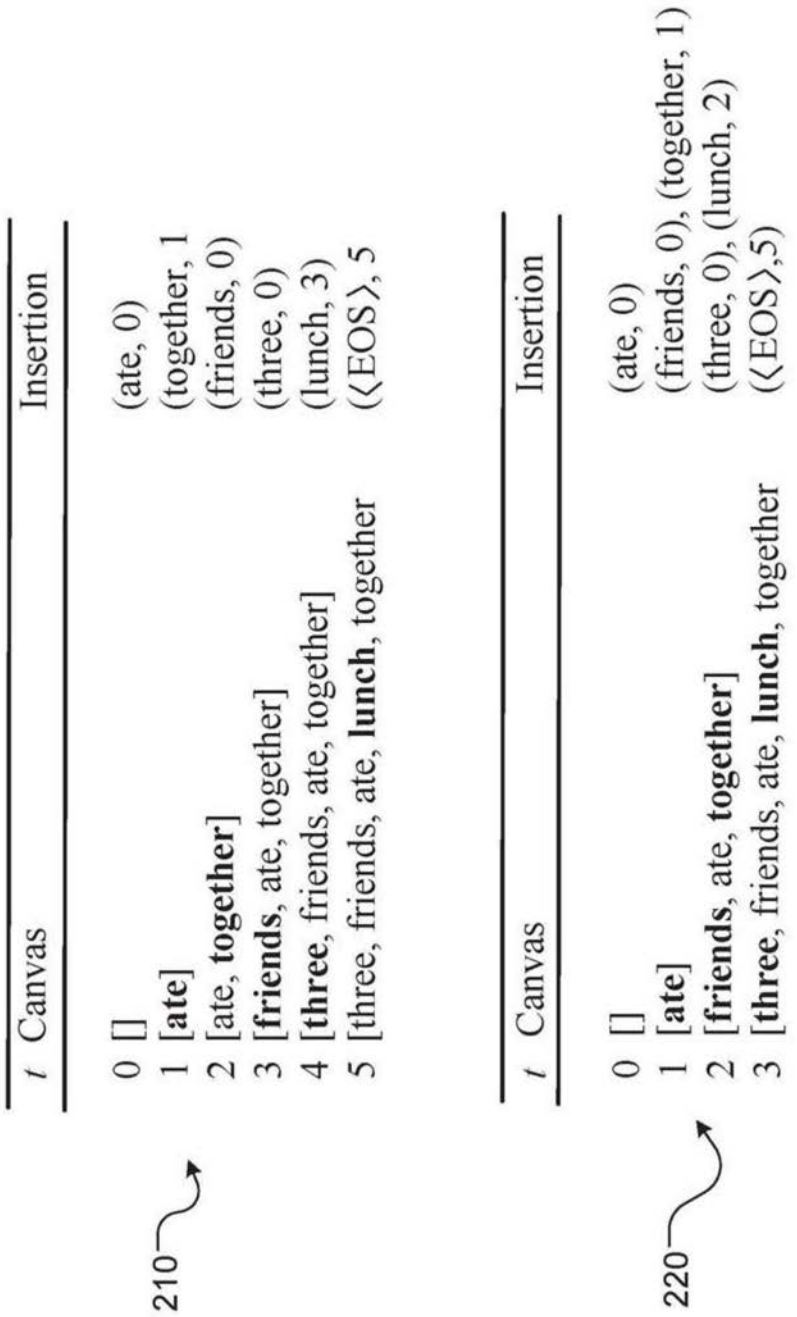

t Canvas Insertion
0 [] (ate, 0)
1 [ate] (together, 1
2 [ate, together] (friends, 0)
3 [friends, ate, together] (three, 0)
4 [three, friends, ate, together] (lunch, 3)
5 [three, friends, ate, lunch, together ⟨EOS⟩, 5
210
t Canvas Insertion
0 [] (ate, 0)
1 [ate] (friends, 0), (together, 1)
2 [friends, ate, together] (three, 0), (lunch, 2)
3 [three, friends, ate, lunch, together ⟨EOS⟩,5)
220

图2

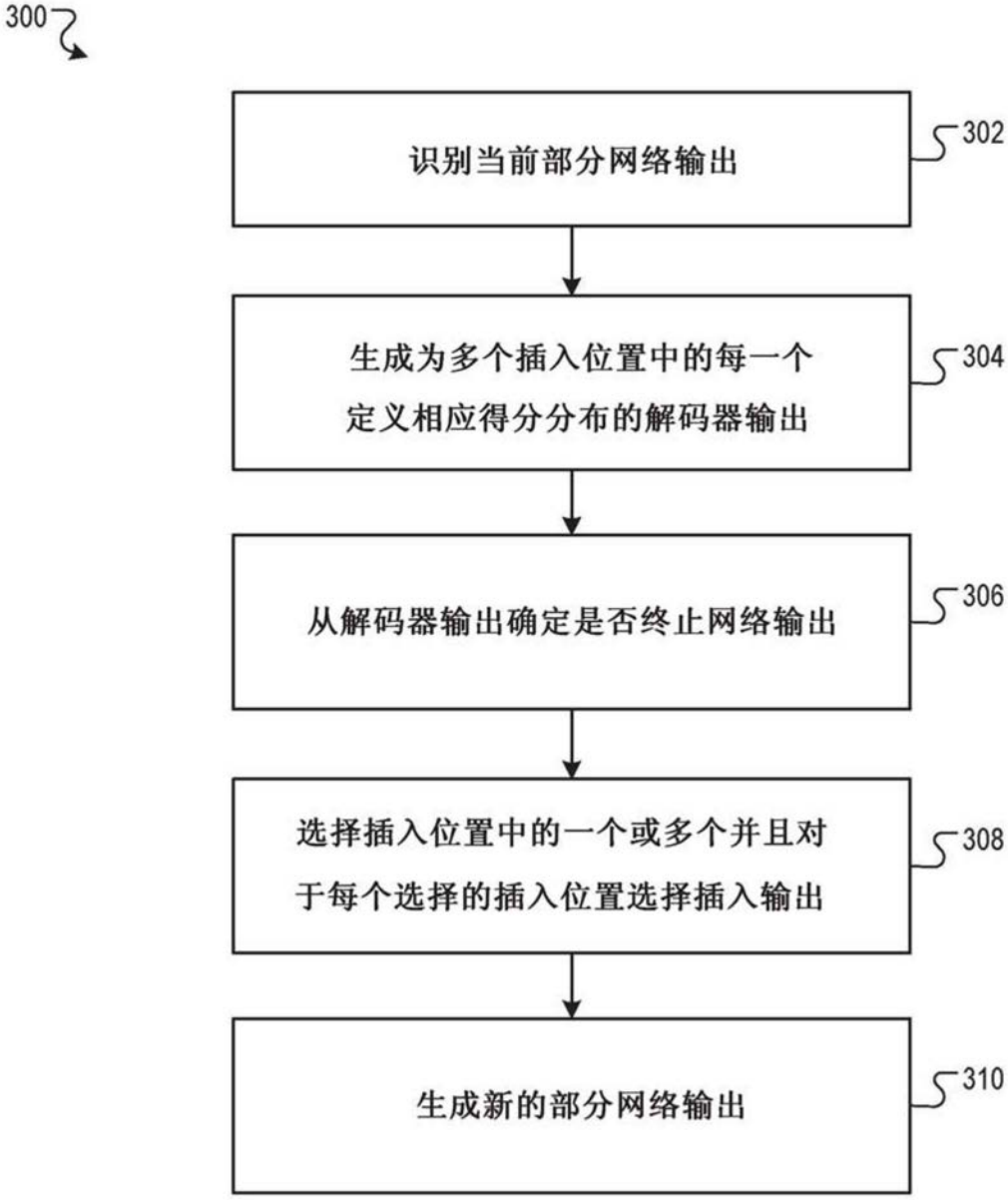

300
识别当前部分网络输出
302
生成为多个插入位置中的每一个
定义相应得分分布的解码器输出
304
从解码器输出确定是否终止网络输出
306
选择插入位置中的一个或多个并且对
于每个选择的插入位置选择插入输出
308
生成新的部分网络输出
310

图3

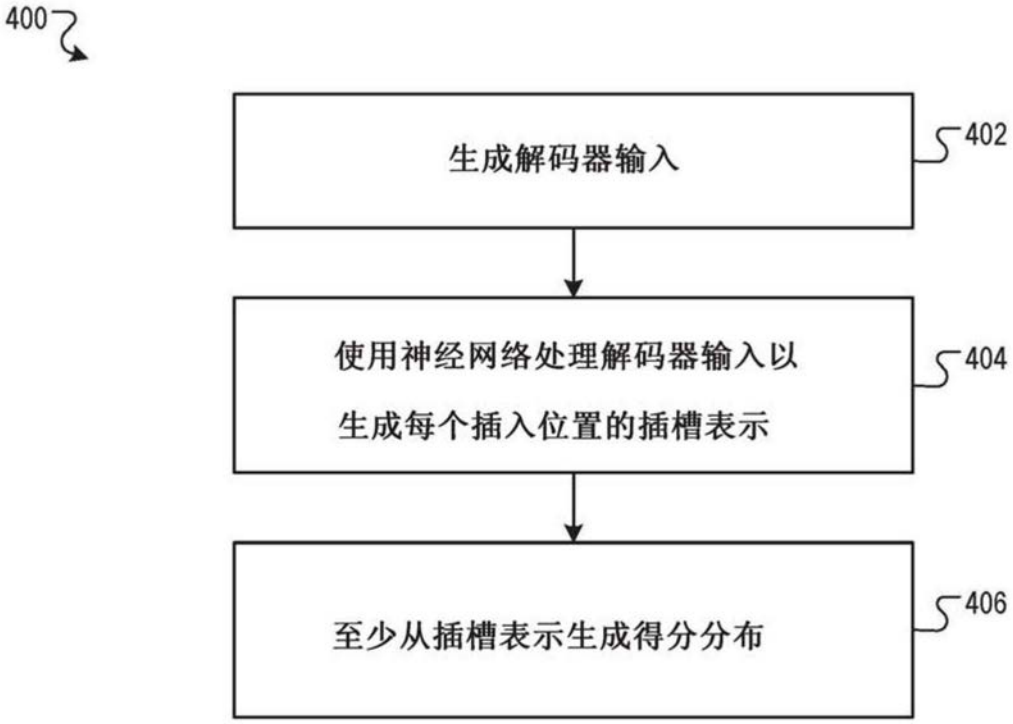

400
生成解码器输入
402
使用神经网络处理解码器输入以
生成每个插入位置的插槽表示
404
至少从插槽表示生成得分分布
406

图4