第七章

神经机器翻译的工作原理

Juan Antonio Pérez-Ortiz
西班牙阿利坎特大学

Mikel L. Forcada
西班牙阿利坎特大学

Felipe Sánchez-Martínez
西班牙阿利坎特大学

本章介绍神经机器翻译系统的主要工作原理。我们会一一讲解这些系统的关键概念，帮助读者全面了解其内部运作和可能性。这些概念包括神经网络、学习算法、词嵌入、注意力和编码器-解码器架构。

1 引言

了解神经机器翻译 (NMT)，首先应该知道它把翻译看作一项数学系统的数字处理任务。这些被称作“人工神经网络”的数学系统把句子转换成一系列数字。系统会加上一些数字（通常是几千或几百万个数字），然后乘以其他数字，再进行一些相对简单的数学运算，最终输出原句的译文。

你可能一直都从另一个角度来理解翻译，认为这是一项发生在人类大脑深层区域的智力任务，涉及的认知过程难以窥知。你的理解没错！不过，目前计算机模拟人工翻译的方法采取截然不同的路径：数百万次数学运算只需片刻便可生成译文，而译文的质量时好时坏。实践证明，过去几年里，生成恰当译文的概率有了显著提高。但从发展历程看来，人工神经网络是模拟自然神经网络（如人脑）运作方式设计的简化模型，来模拟人类大脑等自然神经网络的运作原理，其认知过程也是分布式神经计算过程的结果，这些过程与上述的数学运算并无太大区别。

Juan Antonio Pérez-Ortiz, Mikel L. Forcada & Felipe Sánchez-Martínez. 2025. 神经机器翻译的工作原理. In Dorothy Kenny (ed.), 机器翻译知识普及：为人工智能时代的用户赋能, 111–128. Berlin: Language Science Press. DOI: 10.5281/zenodo.14922297

本章会讲解神经机器翻译技术的关键要素。首先要介绍的是人类大脑处理翻译与神经机器翻译之间的联系，在此过程中我们会引入理解“机器学习”和“人工神经网络”所需的基本概念。“机器学习”和“人工神经网络”是神经机器翻译的两大基石。然后探讨词的计算机化表示方法——“语境无关的词嵌入”(non-contextual word embeddings) 的基本原则，这种方法如果伴随许多重要属性，若结合“注意力”机制，就可以得到实现神经机器翻译的关键因素——“语境单词嵌入”(contextual word embeddings)。以上所有内容能够帮助我们全面了解最常用的两种神经机器翻译模型——Transformer 模型和循环模型——的内部运作。本章最后还介绍了一系列次要主题，有助增进读者对系统内部运作的了解。

2 人工翻译与神经机器翻译的简单类比

为了简化讨论，我们姑且认为，翻译一篇文章就是翻译其中的每个句子。现在假设句子的翻译分两步：译者先确定整个原句的“阐释”或“意义”，然后一下子用目的语表达出差不多的意思。但译者每天都会遇到新的句子（如 The pencil slipped from my hand, stood up, and started talking to me），仍可以照常翻译。这是怎么做到的？语言学对此给出的解答是依靠“语义组合原则”(principle of semantic compositionality)，即把对每个单词的理解组合起来“构建”对句子的理解，单词的组合顺序由句法结构制约——单词组成短语、短语组成语段，直至得到完整的句子。译者分析源语句子的意义，然后进行上述过程的逆向操作，使用目标语生成对应的句子。当然，译者并非总把句子当成一个整体来处理，尤其是遇到长句时，他们可能采取简化方法来避免整个句子的复杂分析，不过我们暂且不考虑这些例外。

而神经机器翻译的工作原理与此类似。翻译句子时，系统会在“编码”阶段为原文的每个单词分配一个神经“表征”或“嵌入”。然后，将这些神经表示结合起来，为句子产生类似的表示。组合时，各个表示会根据语境进行调整，这可以视为理解或意义的语境化表现。到了“解码”阶段，句级表示逐步拆解，以逐个预测目标语句子的单词。完成“编码”和“解码”的是两个人工神经网络——“编码器”和“解码器”，它们相互连接组成单个复合神经网络。

与译者一样，在生成目标语的每个单词时，目前的神经网络架构也并非考量一整个句子，而是学习“注意”相关的源语单词和已经生成的目标语单词。

本章的余下部分会进一步解释神经表示的性质，描述使用选择性聚焦重点的“注意力”机制来构建和转变这些表示的人工神经网络（以下简称“神经网络”）的结构，以及通过翻译实例来介绍这些人工神经网络的训练方式。

3 人工神经网络

要想理解神经机器翻译，就得弄清实现这项技术的人工神经网络 (Goodfellow et al. 2016) 的构成、运作原理和训练方法。

“神经”一词很容易让人联想到神经元和动物的神经系统，尤其是人类大脑的运作方式。实际上，人工神经网络的确是由成千上万个人工神经元组成，这些神经元的“激活”（即“兴奋”或“抑制”的程度）取决于它们从其他神经元接收的信号以及携带这些信号的连接强度。

3.1 人工神经元

人工神经元是人工神经网络的主要组成部分。这些人工神经元（以下简称“神经元”）的状态更新或激活可分为两步。以图 1的简单情况为例，了解神经元 S_4 如何在接收来自神经元 S_1、S_2 和 S_3 的刺激后被激活。

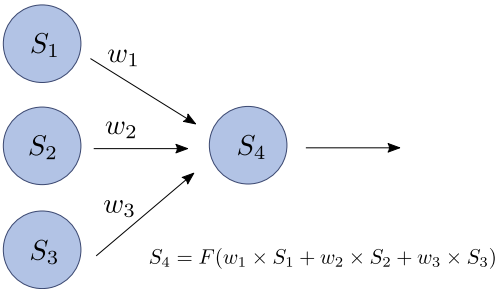

图 1: 神经元 S_4 接收来自神经元 S_1、S_2 和 S_3 的刺激后被激活。

如图所示，第一步，所有与神经元 S_4 相连接的神经元（S_1、S_2 和 S_3）都要乘以其对应的连接强度“权重”（w_1、w_2 和 w_3），得到的神经元 S_1、S_2 和 S_3 的激活值进行加法运算；这些权重决定了神经元 S_1、S_2 和 S_3 对神经元 S_4 的实际刺激值。权重可为正值或负值。例如，若权重 w_2 为正且 S_2 的激活值高，则神经元 S_2 会激发神经元 S_4（正刺激）；但若 w_2 为负，则会抑制神经元 S_4（负刺激）。一般来说，通过正权重连接的神经元会同时受到激发或抑制，而通过负权重连接的神经元则激活状态相反。说回神经元 S_4，若把来自每个神经元的刺激相加，可得到“净刺激”：

$$x = w_1 \times S_1 + w_2 \times S_2 + w_3 \times S_3. \tag{1}$$

净刺激 x 可为任意的正值或负值，但还不是神经元 S_4 的实际激活值。第二步，神经元 S_4“响应”这个刺激。在这个例子中，当刺激处于中间水平（即没有过度偏正或偏负），神经元 S_4 对它的响应便非常敏感。然而，当刺激变

大时（无论正负），刺激值的变化对结果的影响较小，因为神经元在很大程度上被抑制或激发。

在该示例中，神经元 S_4 的激活值范围在 -1 到 $+1$ 之间。图 2表示神经元 S_4 如何对等式 (1) 中的刺激值作出反应。该反应以“激活函数” $F(...)$ 来表示，用来计算刺激值；得到的结果便是 S_4 的激活值：

$$S_4 = F(x) = F(w_1 \times S_1 + w_2 \times S_2 + w_3 \times S_3). \tag{2}$$

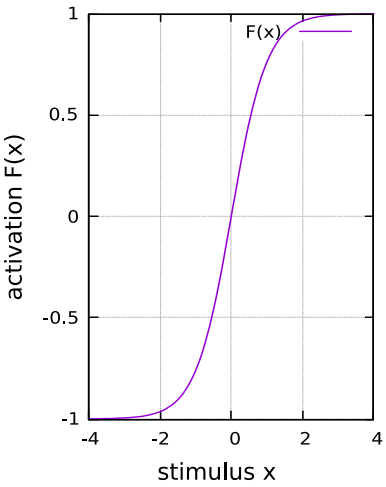

图 2: 神经元如何对接收到的总刺激值做出反应。

可以看出，横轴取值为 0 左右的反应与刺激成比例，但当刺激值过大或过小，神经元会处于过度激发或抑制状态，因此做出的反应要小得多。对于这种神经元，无论总刺激有多强，都不会达到 -1 和 $+1$ 的实际极值。如上所述，示例中的神经元 S_4 是激活值范围在 -1 到 $+1$ 之间的特殊神经元。还有取值范围不同的激活函数，但不在本章的讨论范围内。

3.2 从神经元到神经网络

上一节讨论过的神经元可以连接起来形成人工神经网络，完成特定的计算任务，以解决特定的问题。在神经网络中，有些被称为“输入神经元”，用于接收外部刺激作为神经网络的输入（正如眼睛连接到大脑并向其输入画面），表示一个需要解决的问题；有些被称为“隐藏神经元”，只接受来自其他神经元的刺激；最后，还有些被称为“输出神经元”，表示问题的解决方案（类似发送到手部肌肉的信号，让手做出某个动作）。图 3展示了含有五个神经元的

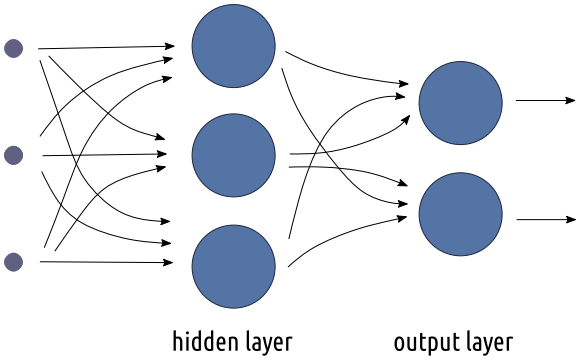

图 3: 含有三个隐藏神经元和两个输出神经元的人工神经网络。每个连接都对应一个权重，但图中并未显示。最左边的三个小圆圈表示输入神经元，直接传出外部输入值。与一般神经元不同，不需要通过激活函数来计算输入神经元接受的刺激值。

神经网络，其中三个输入向量将信号传递给三个隐藏神经元，然后又刺激两个输出神经元。

要建立神经网络来解决特定问题，首先得确定其“架构”，即神经元的数量和连接方式、哪些是输入神经元或输出神经元。不过，实际计算取决于神经网络所有连接的权重。第 3.5节会解释如何得出这些权重。这里只需指出，人工神经网络的优点是它们经“训练”后能执行示例中的任务，也就是说，可以通过观察一组已解决的示例，使其将权重设为特定值，每个示例都包括输入信号值（表示问题的特征）和期望的输出激活值（表示解决方案）。

3.3 神经元层

假设你是零基础学习者，想学一些风景油画的基本技巧。教材可能会告诉你有哪些步骤，如素描起稿（粗略构图）、色彩分布、细节微调和上色成图（完成最后的润色）。这里的重点不在于步骤数量或每步特点，而是整个过程层层递进，每一个步骤的输出即为下一个步骤的输入。每一步优化了上一步的结果——第二步（色彩分布）的结果比第一步（素描起稿）的结果更像一幅风景画。同样，从概念上讲，我们可以认为第四步（上色成图）的结果优于前面所有步骤产出的结果。

事实证明，神经网络计算也是类似的层层递进过程。早在 60 年代就有研究人员发现，使用多层神经元可以处理更复杂的任务。多层神经网络中的每一层都细化上一层的输出，并向最终解决方案迈出或大或小的一步。多层神经网络的最终架构类似图 3所示，但增加了一些隐藏层。图 3简单的神经网络清楚展示了这种分层结构——计算分两步，在两层神经元中进行。

由多层神经元组成的模型被称为“分层神经网络”。尽管理论结果表明，双层神经网络的算力足以执行几乎所有任务 (Hornik 1991)，但实际上，神经网络的算力似乎与神经网络的层数相关；含有多层神经元的模型通常被称为“深层神经网络”，相应的训练算法被称为“深度学习算法”。

深度神经网络内在结构极为复杂，例如，2020 年发布的自然语言生成领域最大的神经网络之一 GPT-3(Brown et al. 2020) 含有 96 层，每层的神经元达数万个，其训练算法需要学习大约 1750 亿个权重。GPT-3 系统的训练需要借助超级计算机，训练过程长达数周甚至数月，但据估计，如果使用性能绝佳的游戏台式电脑来学习这类模型的权重，估计需要 350 多年。[1]

3.4 神经机器翻译

如果能把一个源语句子表示为一组神经网络输入值，然后将神经网络输出解读成目标语句子，这就是“神经机器翻译”（neural machine translation, NMT）系统。神经机器翻译首先处理源语句子的单词。神经网络的编码器每读取一个源语单词，网络中一组特定的神经元的激活状态就会改变。处理完整个源语句子后，神经网络的解码器便开始运作。训练过的编码器在给定已输出的目标语单词的情况下，逐步得出译文中每个可能的目标语单词的概率。这一过程类似于现代智能手机的预测键盘的工作方式，但是，正如我们将看到的，神经机器翻译的单词预测也依赖于源语句子，因为毕竟预测的结果是源语句子的翻译。

神经机器翻译系统是深层神经网络，其架构将在后面的第6节中讨论。它们有数以千记的神经元和数以百万记的权重（甚至更多），神经元和权重的训练需要借助从包含数百万源句及其翻译的平行语料库中获取的示例，来训练这些神经元和权重。源语句子单词的数学表示作为输入传送到神经网络，而对应的目标语句子单词用于表示期望的输出。正如你所料，在合理时间内训练一个大型神经网络对算力的要求很高——需要能够处理大量复杂计算任务的高性能硬件，一遍遍地展示例子来训练神经网络。每次迭代都会给神经网络的权重带来细微调整，以改进其对目标语单词的预测。

3.5 神经网络的训练

神经网络的训练过程就是确定神经元之间的连接权重，以便在给定输入-输出示例的“训练集”的情况下，得到的实际输出尽可能接近相关示例。

训练初始权重或随机选取，或从解决类似任务的神经网络中提取。在训练期间，权重不断调整，以达到最小误差函数值（又称“损失函数”），该函数测量实际输出相对于期望输出的偏离程度。“训练算法”（又称“学习算法”）反

[1]“OpenAI’s GPT-3 language model: A technical overview” (2020). 检索自 https://lambdalabs.com/blog/demystifying-gpt-3。

复计算权重的微小调整（或更新），直到误差函数对于训练集的所有示例都是最小的或足够小，或者在不同“开发集”中观察到某种预期的性能，这也是预留开发集的目的所在（见第7.2节）。

训练算法的技术细节不在本章的讨论范围内，只需知道它通常基于计算当每个权重以固定但很小的程度（即误差函数的“梯度”）变化时，误差函数会出现多大变化，然后对每个权重进行调整以减小误差函数。[2] 这类训练被称为“梯度下降法”（gradient descent），它无法保证找到最佳权重，但很可能会找到不错的权重。权重变化的幅度由“学习率”（learning rate）这一参数来调节。训练算法执行之初，学习率通常较高，但是随着权重越来越接近最终值，学习率逐渐变小。注意，神经网络的训练过程很费力，因为需要大量实例而且多次呈现给系统学习才能成功。然而，这主要是由于训练算法的局限性，而非特定神经网络缺乏表示问题解决方案的能力。

权重一旦确定，训练随即停止（见第7.2节）。这时，可以向神经网络输入训练集以外的例子，得到相应输出。

3.6 神经网络中的泛化

“泛化”(generalisation) 是人类和动物的基本认知过程，使我们将已有知识运用到新的场景，这些新情况与当初的学习场景类似，但又不尽相同。例如，一个原本就会开车的人并不需要为了驾车驶入陌生道路或驾驶新车而重新学习开车。同样地，已经通过某种方式对某种刺激做出反应的生物体，能够以类似方式对类似刺激做出反应，这就是“泛化”。“泛化”也是语言学习的关键，幼儿很快就能学会说出从未听过的话。

理想情况下，神经网络可以在机器翻译的场景实现泛化，通过相似的输入得到相似的输出，无论这些输入是否存在于训练集中。神经网络的一大特点就是“平滑性”，这意味着如果输入值略有变化，计算的结果不会有显著变化。

在广义上，为了实现泛化，相似的句子应有相似的表征，并且由于句子表征来自单词表征，可以推断出，用相似的数字来表示相似的单词，是神经网络语言处理中泛化的先决条件。

下一节，我们将深入研究如何利用神经网络的平滑性，得到句子中单词的神经网络表征，使经过训练的系统能正确地泛化新句子。

4 词嵌入：词向量表征形式

在上一节中，我们讲到神经元通常分层排列，上一层神经元的输出是下一层神经元的输入。值得注意的是，某层神经元组的输出恰为当前一步正在处理的信息的表征。

[2] 看到这里，有些读者可能认出了“函数导数”这一数学概念。

在自然语言处理领域，如上所述，神经网络处理的信息由单词组成，而神经网络的单词表征常被称为“嵌入”(Mikolov et al. 2013)。使用词嵌入的重要特质在于，意义相近或在相同语境下共现的词嵌入也相近。为了更好地理解这一点，请在纸上画一个边长约 10 厘米的正方形。现在，根据意思的远近把以下单词填入正方形，意思越相近，单词距离越近。如果无法判断意思远近，可以根据单词在句子或段落中同时出现的频率来决定其位置。单词包括 restaurant（餐厅）、red（红色）、garden（花园）、fountain（喷泉）、flower（花）、tomato（番茄）、balloon（气球）、waiters（服务员）、knife（刀）、flowers（花）、menu（菜单）、cooked（煮熟）、chromosome（染色体）和 consistently（始终如一）。请先完成这一步，再继续阅读。

受词义远近程度的限制意味着不能把单词随意放在正方形内。你可能会把“餐厅”、“菜单”和“服务员”分成一组，而“花园”、“花”和“喷泉”为另一组。但有些词的位置难以决断，如“红色”明显与“番茄”相近，但同时与“花”也相近，可以折中把它放在两者之间。如果我们认为与“花”相比，“红色”这一属性与“番茄”的关系更为紧密，则“红色”更接近“番茄”，而不是“花”。

你可能已经注意到正方形里出现了几个词群，有的表示餐厅及相关事物的语义场，有的则围绕花园和果园的概念。还有一些离群词，尤其是“始终如一”一词，不得不把它放在远离其他单词的位置。难以归类的离群词还有“染色体”，但由于“花”和“服务员”都具有携带遗传信息的染色体，因此它可以放在这两个词之间，但同时不会太接近“红色”。如图4所示为以上单词的可能分布情况，不一定与您给出的答案相同。[3]

为了用数学方式表示这些单词，我们用坐标来表示每个单词在正方形中的位置。在这个二维空间里，每个单词需要一对坐标：第一个数字表示该词到正方形左侧垂直边的距离，第二个数字表示到正方形底部水平边的距离。例如，单词“餐厅”的坐标分别为 0.25 和 1.1，单词“菜单”的为 0.6 和 1.3，接近“餐厅”，如图4所示。这些坐标值可用“向量记法”来表示，只需用逗号将两个数字分开，放在中括号里。因此“餐厅”和“菜单”对应的向量分别为 [0.25⍰1.1] 和 [0.6⍰1.3]，有可能是代表这两个词的词嵌入之一。

虽然可能并不是非常直观，但使用两个数字而非一个数字来表示词嵌入，有助于解决单词远近位置的问题，因为我们有更多的自由来满足上述问题的限制条件。事实上，从二维增至更高维度，能更好地解决这个问题。一个词的五维表示可以是 [2.34⍰1.67⍰4.81⍰3.01⍰5.61]。神经机器翻译系统能处理包含数百个维度的词嵌入，而待翻译的句子便由这些规模庞大的词嵌入的集合来表示。

我们在第 refss:trainnn 节中介绍过神经网络权重的学习依靠算法实现，而同样的算法也可以用于词嵌入的学习。实际上，权重和词嵌入的学习同时进行。考虑到在神经网络机器翻译中，神经网络的输入层通常由输入句子的词

[3]我们特意将图4放在后几页，避免您做练习的时候看到。

嵌入组成，我们无需局限于固定的向量表示。相反，向量值在训练过程中不断更新，以便使误差函数的值最小化。。

4.1 泛化

如上所述，为使神经网络能适当“泛化”，即能学习翻译并能翻译新句子，则相似的句子应有相似的表征。由于句子表征来自词嵌入，我们可以推断出，用相似的数字表示相似的单词是神经自然语言处理中泛化的先决条件。例如，poured（倾注）、rained（下雨）、pouring（倾注）和 raining（下雨）的语义相似，所以它们的词嵌入应该相似；而 pouring 和 raining 的向量也应该更接近 driving 这类词，因为它们都是动名词，可能在相似语境中出现；poured 和 rained 应该处于邻近位置，因为它们都是过去式。高维向量表示的必要性由此可见：我们希望单词能同时根据不同判断标准彼此接近。

4.2 词嵌入的几何特性

词嵌入可以表示单词语义（或与语义相关的）特征。如前所述，词嵌入由实数组成（通常是成百上千个），每个数字似乎都表示词义的某个方面。例如，单词 Dublin 的词嵌入捕捉到的词义包括城市、爱尔兰首都和多家跨国公司的欧洲总部所在地等。

由于词嵌入不同维度具有专门含义，我们可以对词嵌入进行算术运算，并得到有意义的结果。两个词嵌入的对应向量值两两相加（或相减），例如 $[1.24\boxtimes 2.56\boxtimes 5.23] + [0.12\boxtimes 1.12\boxtimes 0.01] = [1.36\boxtimes 3.68\boxtimes 5.24]$。以下为神经机器翻译系统通常学习的两个词嵌入运算：

$$\begin{aligned} [king] - [man] + [woman] &\simeq [queen] \\ [Dublin] - [Ireland] + [France] &\simeq [Paris] \end{aligned}$$

其中，方括号指的是词嵌入，而 $\simeq$ 表示运算得到的词嵌入约等于符号右侧的词嵌入。这可以理解为 *king* 之于 *man*，正如 *queen* 之于 *woman*，即男君主或女君主；而 *Dublin* 之于 *Ireland*，正如 *Paris* 之于 *France*，即国家首都。

5 使用注意力机制的语境词嵌入

词语并非在所有句子中都具有相同含义。例如，letter 一词可以指“字母”，也可以指“信件”，因此应该使用不同的词嵌入。其实，对于神经机器翻译系统来说，更有趣的是，可以根据这个词是指“情书”，还是“投诉信”，用不同的

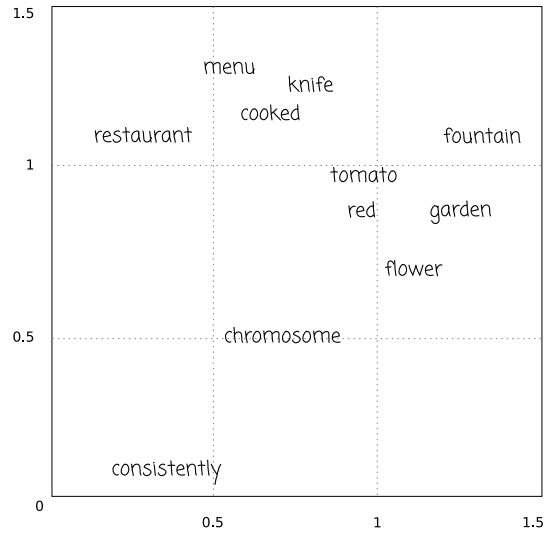

图 4: 在二维空间中, 意思相近的单词距离较近, 意思不近的单词距离较远。

词嵌入来表示。上文介绍的是“非语境”词嵌入：其计算主要根据词语是否经常同时出现在句子中, 并没有考虑一词多义的情况。

在神经机器翻译领域, “注意力”机制起着重要的作用, 使神经网络能计算“语境词嵌入”, 也就是说, 在不同句子中含义不同的词, 其表征也随之改变, 以此计算出词的向量表征。再次说明一下, “注意力”可以通过数学运算来实现, 而训练算法能轻松学习这些运算。在这个语境下, “注意力”类似我们在日常生活中对某人某事的关注。

如以下示例, 通过将“注意力”集中在某些单词上, season 一词在两个句子中的词嵌入向量有所不同：

1. The first episode will pick up right where the previous season left off.

2. Summer is the hottest season of the whole year.

原则上, 这听起来好像语境词嵌入是为了让单词的不同含义有不同的表征, 尽管通常可以这么理解, 但并不仅限于此。在句子“Winter is the coldest season of the year in polar and temperate zones”、“Summer is the hottest season of the whole year”, 甚至“Of the whole year, summer is the hottest season”中, season 一词的语境词嵌入都不同, 尽管彼此可能比在句子“The first episode will pick up right where the previous season left off”中的表征更接近。这些差

异源自句子中的单词或其排列顺序不同。值得注意的是，单词 the 在上述两个例句中的语境向量也不同，因为其所在的语境不同。

如何使用“注意力”机制对语境词嵌入进行数学计算？以上述第二个句子（Summer is the hottest season of the whole year.）为例，计算过程从得到（第 4 节中介绍过的）“语境无关的词嵌入”开始。该句含有 9 个单词，可得到 9 个向量。基于此，为了计算 season 一词的语境词嵌入，神经网络以数学方式得出“注意力”向量。该向量包含 9 个百分比，用来表示句中每个单词所需的“注意力”程度，以便得到 season 一词的表征。向量元素与句子单词的“注意力”位置一一对应。例如，“注意力”向量 [25%、8%、10%、15%、25%、8%、2%、0%、7%] 表示，为了计算例句中 season 一词的语境向量表征，单词 summer 和 season 的词嵌入的重要程度都极高（共占 50% 的“注意力”）。这也合理，因为从语义的角度来看，它们都与气象季节的概念相关。请注意，season 前面的限定词也获得一定比例的注意力（10%），这可能是因为它有助于将 season 标记为名词。谓语动词（8%）对语境词嵌入的贡献在于把 season 标记为单数。所有百分比之和一定为 100%。

至于如何结合“注意力”向量和原始的非语境词嵌入，以获得新的词嵌入，这超出了本章的内容范围。简单来说，这个过程涉及特定的数学运算，得到新的词嵌入位于多个原始词嵌入之间的某个位置。

回到上述的例句，计算该例句的语境词嵌入需要计算 9 个注意力向量（每个单词都有一个向量），然后应用于原始的语境无关词嵌入，从而获得 9 个对应句子单词的新词嵌入。这些新的嵌入可以被认为是语境嵌入，因为它们受到句子其他单词不同程度的影响。

5.1 注意力层，多多益善

我们在本章的第3.3节讨论过利用多层神经网络的模型来接连优化计算的好处。类似地，为了得到更精确的表征，新的词嵌入也可以结合新的注意力向量，继续获得新的词嵌入。事实上，另一个发布于 2020 年的最大语言模型之一——图灵自然语言生成（T-NLG）——含有 78 个注意力层，可以“接力式”优化高达 4256 个维度的词嵌入。[4] 回顾一下，这些运用多层网络习得的表征被称为“深层”表征。

5.2 注意力头，多多益善

在每一层神经网络中，每个单词都并不只有一个注意力向量。以句子“My grandpa baked bread in his oven daily”（我的爷爷每天都在他的烤箱里烤面

[4]“Turing-NLG: A 17-billion-parameter language model by Microsoft”, 2020. 检索于 https://www.microsoft.com/en-us/research/blog/turing-nlg-a-17-billion-parameter-language-model-by-microsoft/

包）为例，单词 oven（烤箱）的词嵌入很有意思，它不仅和 grandpa（爷爷）一词有关，说明这个烤箱属于一名老年人，还和 bread（面包）一词有关，表明面包是用烤箱烘焙的。只使用一个注意力向量就会导致表示不同注意力的信息混杂其中，可能不利于准确找出词嵌入所表征的那个单词的译文。因此，有些神经机器翻译系统会在每一层神经网络中，给每个单词分配不同的注意力向量，以此算出每个单词的不同词嵌入。可以说，每个词嵌入都是使用不同“注意力头”的计算所得。T-NLG 模型每层含有 28 个注意力头。因此，在该模型的最后一层神经网络，每个单词得到 28 个 4256 维的词嵌入。

5.3 自然语言处理中的语境词嵌入应用

词嵌入不仅是神经机器翻译的基石，也能助力许多其他自然语言处理的应用，如情感分析和自动摘要。举例来说，有些系统能够将包含产品评价的文本中的句子自动分为两类——正面情感和负面情感。首先，计算句中每个单词的深层语境词嵌入，然后将这些词嵌入输入结构更简单的神经网络，计算出表示句子情感倾向评分的结果，结果范围为 0 到 1 之间（例如，0.95 表示句子包含正面情感，0.2 表示句子包含负面情感，0.51 表示句子情感为中性）。这些系统通常使用人工标注的句子语料库来进行训练。目前，很多语言都可以免费使用基于数百万个句子的“预训练”模型，因此模型的词嵌入计算训练并非必须使用特定的语料库。

6 压轴登场：神经机器翻译

下面，我们会简单描述神经机器翻译的基本原理，希望前文的铺垫能帮助您更好地理解这一点。我们将主要介绍 Transformer 模型和循环神经网络这两种架构。

6.1 Transformer 模型：基于注意力机制的编码器-解码器

简言之，使用 Transformer 模型的神经机器翻译系统由两个模块组成：第一个模块计算源语句子中每个单词的语境词嵌入，而第二个模块接连预测目标语句子的每个单词。前者称为“编码器”，后者称为“解码器”。为了预测目标语的单词，解码器会关注源语句子所有单词的词嵌入，以及已经生成的目标语单词的词嵌入。这一整个架构称作 Transformer 模型 (Vaswani et al. 2017)。图5展示了有三层神经网络的编码器以及计算第二层和第三层的词嵌入使用的注意力程度。图6则在此基础上加入了解码器，展示了整个 Transformer 模型架构。

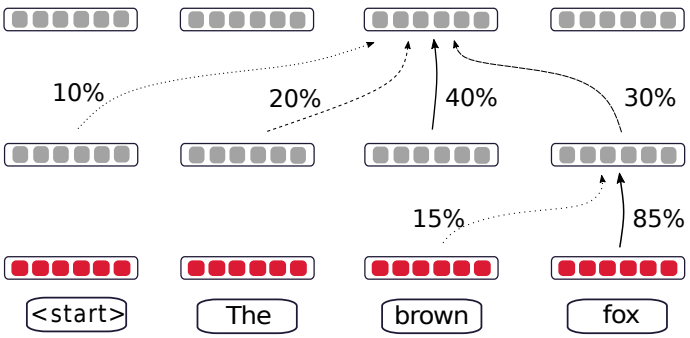

图 5: 基于 Transformer 模型的神经机器翻译系统的编码器。符号 *start* 通常为首标记句子的起点。从示意图中还可以看出，单词 brown 和 fox 的第一层词嵌入对生成 fox 的第二层词嵌入的贡献程度不同；同理，brown 在最后一层的词嵌入综合了第二层所有词嵌入的信息，每个词的注意力程度各不相同。

学习算法使用平行语料库来获得 Transformer 模型所需的权重、词嵌入和注意力向量，以便在一定程度上重现训练数据，使机器翻译系统能够对训练集以外的句子实现泛化。

以句子“My grandpa baked bread in his oven daily”为例，若要使用每层只有一个注意力头的 Transformer 模型将其译成西班牙语，首先编码器会生成 8 个词嵌入向量。然后，解码器算出一个八维的注意力向量，如 [60%, 10%, 0%, 0%, 0%, 30%, 0%, 0%]，并利用此得到源语句子的信息，从而获得目标语句子第一个单词的词嵌入。假设机器翻译系统正确生成了西班牙语单词 *mi*。然后，解码器算出一个九维的注意力向量，如 [50%, 10%, 0%, 0%, 0%, 20%, 0%, 0%, 20%]（最后一个百分比为分配给目标语句子第一个单词的注意力），并基于该向量得到目标语句子第二个单词的词嵌入。这个过程持续进行，直至解码器生成表示句末的特殊标记。

解码器的每一步输出，并不完全是对下一个词的词嵌入的估算。实际上，解码器的末端会额外增加一层神经网络，用以计算目标语词汇表中每个单词的概率或可能性向量。第7.3节将讨论如何使用这些概率来获得生成目标语句子的单词序列。

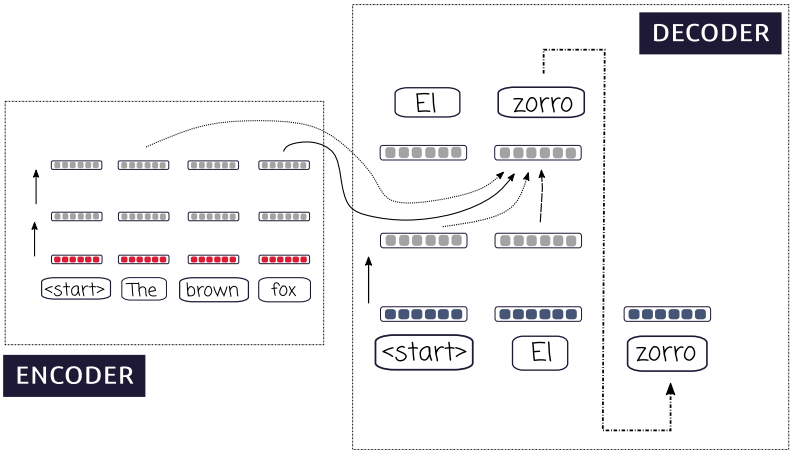

图 6: 基于 Transformer 模型的神经机器翻译系统的完整架构及其句子翻译过程。图5展示了放大的编码器。注意解码器是如何通过使用前一目标语单词的词嵌入，以及编码器最后一层的一些源语单词的词嵌入，来预测 zorro 一词。

6.2 循环神经网络

如上一节所述，Transformer 模型当前大多数用于商业神经机器翻译系统，不过也有可以取而代之的其他神经网络模型。另一个顶级模型是“循环”编码器-解码器模型 (Bahdanau et al. 2015)。它和 Transformer 模型一样，也有编码器和解码器，前者为源语句子单词生成词嵌入，后者通过整合源语单词和已生成的目标语单词的信息，使用注意力机制来计算每个目标语单词的词嵌入。但是，循环神经网络模型中的编码器和解码器以局部方式计算语境词嵌入，例如第五个编码单词的词嵌入不仅基于前四个单词的词嵌入，还基于下一个单词的词嵌入。这是通过从左到右、再从右到左遍历源语句子来实现的。图7所示模型只显示了从左到右的处理过程。

值得注意的是，对于需计算语境词嵌入的单词附近的单词，所使用的数学模型施加了相关性限制（如例子中第五个单词），因此该机制特别关注附近的单词，而倾向于忽略较远的单词的表征。与 Transformer 模型类似，解码器也会在最后一层网络算出一个向量，该向量给出每个目标语单词是输出句子对应位置单词的概率。Forcada (2017)更详细地描述了循环神经网络编码器-解码器模型，还讨论了神经机器翻译系统的输出类型。

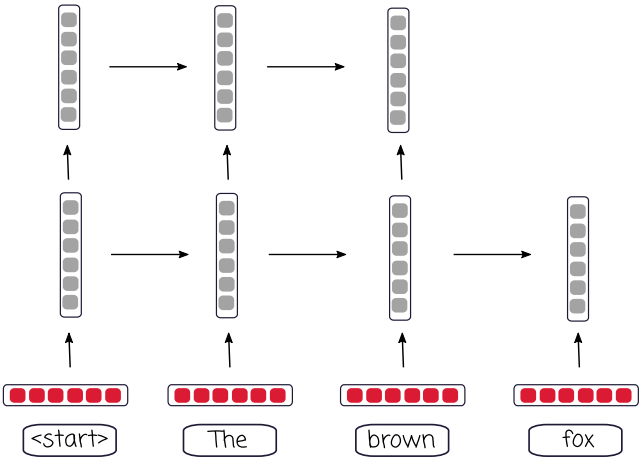

图 7: 处理“<start>The brown”之后，即将处理单词 fox 的循环神经机器翻译编码器的子模型系统（从左到右）。

7 其他设置

7.1 词与子词

根据本章以上内容，不管是使用 Transformer 模型还是循环模型，都可以在训练后得到每个单词的词嵌入。这是否意味着，我们最终可以得到该语言所有单词的词嵌入？不完全是。语言，特别是高度屈折语和黏着语，很可能有数不胜数的不同词形。为什么这对神经机器翻译系统来说是个难题？因为词嵌入的数量（即“词汇表”）决定了神经网络的权重数量，而且大型神经网络通常难以对未见过的数据实现泛化。只考虑出现在训练语料库的单词形式，可以缩小词汇表，但这通常仍意味着要考虑大量单词，且还会衍生出新问题：训练结束后，神经机器翻译系统翻译训练集以外的新句子时，这些未见过的单词会降低模型的性能和准确度，因为在这种情况下，每个未知单词都只有一个语境无关的词嵌入。

机器翻译算法工程师提出的解决方案是，将单词分成所谓的“子词单元”。理想情况下，这些字词单元应具有语言学意义并且能组合构成语义。例如，将单词 demystifying 拆分为“de-”+“-myst-”+“-ify-”+“-ing”，肯定比拆分为“dem-”+“-yif-”+“-yi-”+“-ng”更具有语言学意义（因此可能更有助于机器翻译）。但是，以符合语言学规则的方式拆分单词的前提是该语言已存在拆分规则和步骤，而很多语言可能都不满足这些条件。

常用的解决方法是通过历遍大量文本来自动学习拆分规则，如训练集的所有源语句子和目标语句子。有种常见的方法[5]叫做“字节对编码”(byte-pair

[5]还有更先进的方法，如 SentencePiece(Kudo & Richardson 2018)，它将整个文本视为一个字符序列，并一次性进行单词切分（“分词”）和子词切分。

encoding, BPE), 从字母大小的单元开始, 若相连的某两个或更多字母经常出现在语料库中, 则视为一个单元。[6] “字节对编码”的方法可能识别出许多频现的动词后缀-ing (如 marching 和 continuing), 然后将其删除, 即便是没有见过的形式 (如 bartsimpsoning)。就这样, 后缀-ing 会被转化为具有单元意义的语境词嵌入。

7.2 系统训练停止条件和译文质量评测指标

如第3.5节所述, 除了大型训练语料库, 通常还会预留一个不用于训练的小型“开发语料库”。该语料库的目的是监控神经机器翻译系统在训练过程中的翻译表现, 以决定何时停止训练。训练过程努力使误差函数最小化 (对于神经机器翻译, 实际上是使训练语料库的目标语句子的概率最大化)。还有可能出现的问题是, 在训练语料库上的过度训练, 降低了泛化能力, 因为这会导致神经网络最终“死记硬背”那些示例译文。这时, 开发语料库就有了用武之地: 在训练算法进行了一定次数的迭代或训练算法执行步骤后, 用神经网络来翻译开发语料库的源语句子, 然后使用简单的近似自动评测指标 (参见Rossi & Carré 2025 [本卷]), 自动对比机器译文与语料库的目标语句子, 其中最常见的评测指标是 BLEU(Papineni et al. 2002)。BLEU 会算出机器译文有多少个一词、双词、三词和四词序列出现在参考译文中, 计算结果范围为 0% (零匹配) 到 100% (完全匹配)。在训练期间, 如果开发集的 BLEU 分数显示系统性能下降, 则可停止训练, 或者先保存当前的所有权重, 然后继续训练一段时间, 看看 BLEU 分数是否回升。当然, 除了 BLEU, 还可以使用很多其他的质量自动评测指标。

7.3 集束搜索

如第6.1节和第6.2节所述, 神经机器翻译系统中的解码器按顺序生成译文句子, 每次生成一个目标语单词。在每个时间步长, 神经网络为目标语词汇表中的每个单词得出一个概率 (范围在 0% 到 100% 之间)。获得这一概率后, 可以选择输出概率最大的目标语单词, 忽略其他可能性。值得注意的是, 这么做的话, 我们就完全确定了神经机器翻译系统的后续步骤, 因为当前的预测结果会作为下一步解码器的输入 (例子可见图6所示的单词 *zorro*)。还有一种方法或许可以考虑更多可能性, 例如, 考虑概率最高的三个单词, 并将系统“克隆”成三个系统, 分别由三个选项来决定, 并观察其运行过程。但是该操作不能无限重复进行下去, 因为每运行一步, 都会出现三个系统翻译同一个句子, 则系统数量会呈指数增长。为了避免这种情况, 只允许一定数量的系统“存活”, 即在近似计算中, 完整译文句子的概率最高的系统。这通常被称

[6] 字节对编码最初是一种文本压缩算法: 频现的字母 (“字节”) 序列被存储一次并由短代码取代, 以减少所需的总存储空间。

为“集束搜索”(beam search)，这种近似方法也常见于人类语言处理（如语音识别）的其他概率模型中。

8 结语

训练一个神经机器翻译系统需要数百万个源语和目标语的句对。许多语言对、领域和文本体裁都不具备如此大量的资源，限制了许多特定应用，但是对于资源丰富的语言来说，通用神经机器翻译已经成为现实且运用非常广泛，用户不仅限于译者。此外，近年来，得益于多语言模型或无监督神经机器翻译等领域最新研究进展的出现，低资源语言领域的研究也取得了可喜的成果。[7]

本章介绍了神经机器翻译系统的关键内容和技术细节，并探讨其在“基于 Transformer 模型的神经网络”和“循环神经网络”这两种当前最流行的架构中的交互作用。在撰写本文时，这一领域的研究非常火热，几乎每个月都有新模型横空出世。如果有足够的平行语料库可用于训练，Transformer 模型就是现有的最佳范式，因为与循环神经网络相比，前者所需的训练时间更短，且能够细微优化质量。不过，情况可能随时发生巨变。

References

Bahdanau, Dzmitry, Kyunghyun Cho & Yoshua Bengio. 2015. Neural machine translation by jointly learning to align and translate. In Yoshua Bengio & Yann LeCun (eds.), *3rd International Conference on Learning Representations, ICLR 2015*. DOI: 10.48550/arXiv.1409.0473.

Brown, Tom B., Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Dario Amodei, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner,

[7] “多语言模型”是单一的神经网络，以完成多个语言对之间的翻译为训练目的，从而将资源丰富的语言的“知识”“转移”到资源匮乏的语言中。有趣的是，多语言模型使“零样本翻译”（zero-shot translation）(Ko et al. 2021) 成为可能。例如，即使没有西班牙语-上索布语的平行语料库可用，我们可以先用德语-上索布语和西班牙语-德语这两个语料库训练出一个多语言模型，然后机翻系统利用这个多语言模型模型来完成西班牙语-上索布语的翻译，得到质量还不错的译文。“无监督的神经机器翻译”则更进一步，只需学习单语语料库。

Sam McCandlish, Alec Radford, Ilya Sutskever & Dario Amodei. 2020. Language models are few-shot learners. *CoRR* abs/2005.14165. https://arxiv.org/abs/2005.14165.

Forcada, Mikel. 2017. Making sense of neural machine translation. *Translation Spaces* 6(2). 291–309.

Goodfellow, Ian, Yoshua Bengio & Aaron Courville. 2016. *Deep learning*. Cambridge, MA: MIT Press.

Hornik, Kurt. 1991. Approximation capabilities of multilayer feedforward networks. *Neural Networks* 4(2). 251–257.

Ko, Wei-Jen, Ahmed El-Kishky, Adithya Renduchintala, Vishrav Chaudhary, Naman Goyal, Francisco Guzmán, Pascale Fung, Philipp Koehn & Mona Diab. 2021. Adapting high-resource NMT models to translate low-resource related languages without parallel data. In *Proceedings of the 59th annual meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing*, 802–812.

Kudo, Taku & John Richardson. 2018. SentencePiece: A simple and language independent subword tokenizer and detokenizer for neural text processing. In *Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, 66–71. Brussels, Belgium: Association for Computational Linguistics.

Mikolov, Tomas, Ilya Sutskever, Kai Chen, Greg Corrado & Jeffrey Dean. 2013. Distributed representations of words and phrases and their compositionality. In *Advances in Neural Information Processing Systems 30*, 3111–3119.

Papineni, Kishore, Salim Roukos, Todd Ward & Wei-Jing Zhu. 2002. BLEU: A method for automatic evaluation of machine translation. In *Proceedings of the 40th Annual Meeting of the Association for Computational Linguistics*, 311–318. Philadelphia, Pennsylvania, USA: Association for Computational Linguistics. DOI: 10.3115/1073083.1073135.

Rossi, Caroline & Alice Carré. 2025. 如何选择合适的神经机器翻译解决方案：机器翻译质量评测. In Dorothy Kenny (ed.), 机器翻译知识普及：为人工智能时代的用户赋能, 39–63. Berlin: Language Science Press. DOI: 10.5281/zenodo.14922289.

Vaswani, Ashish, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser & Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30*, 5998–6008.