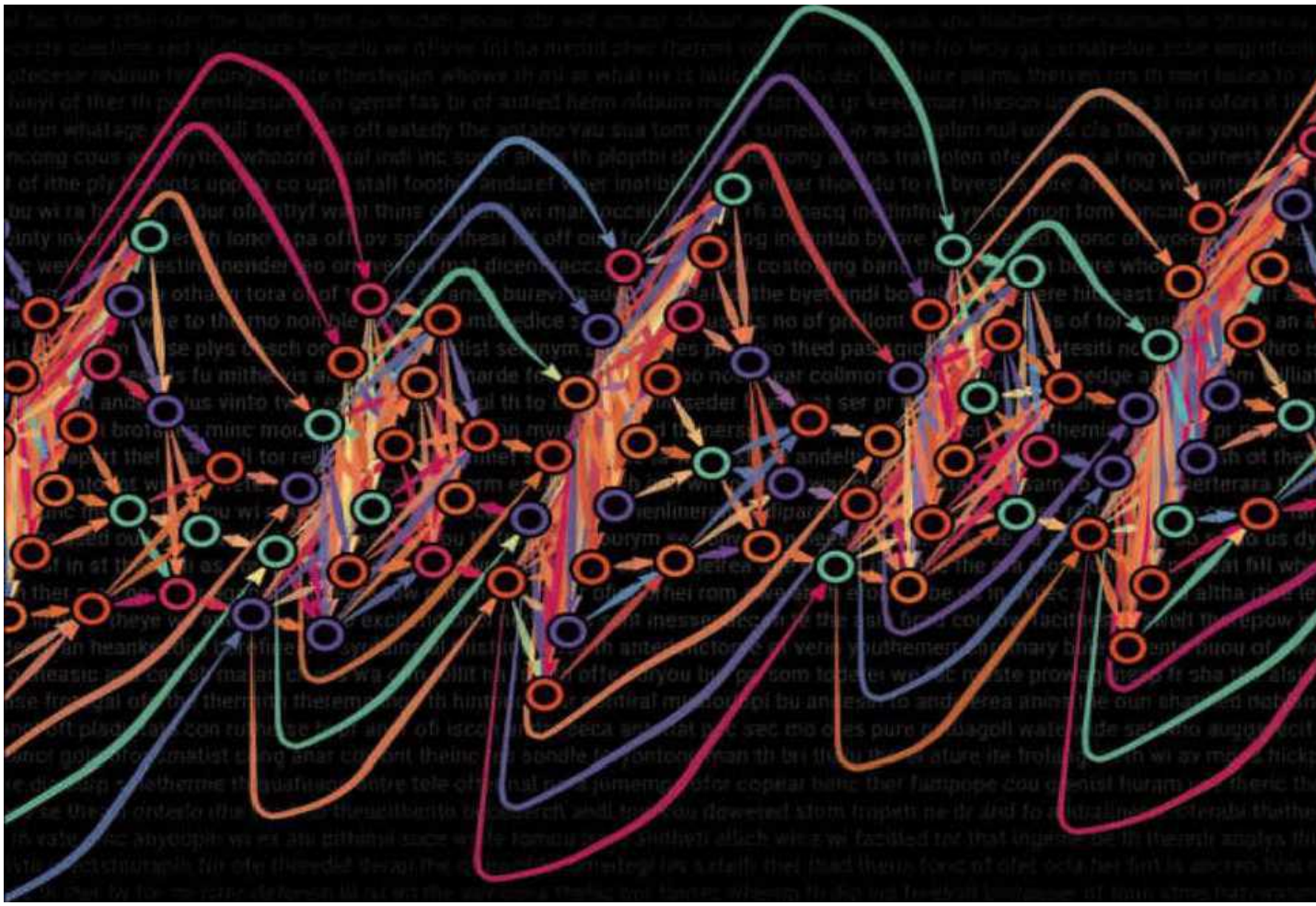
斯蒂芬沃尔夫兰

什么是

ChatGPT

做...

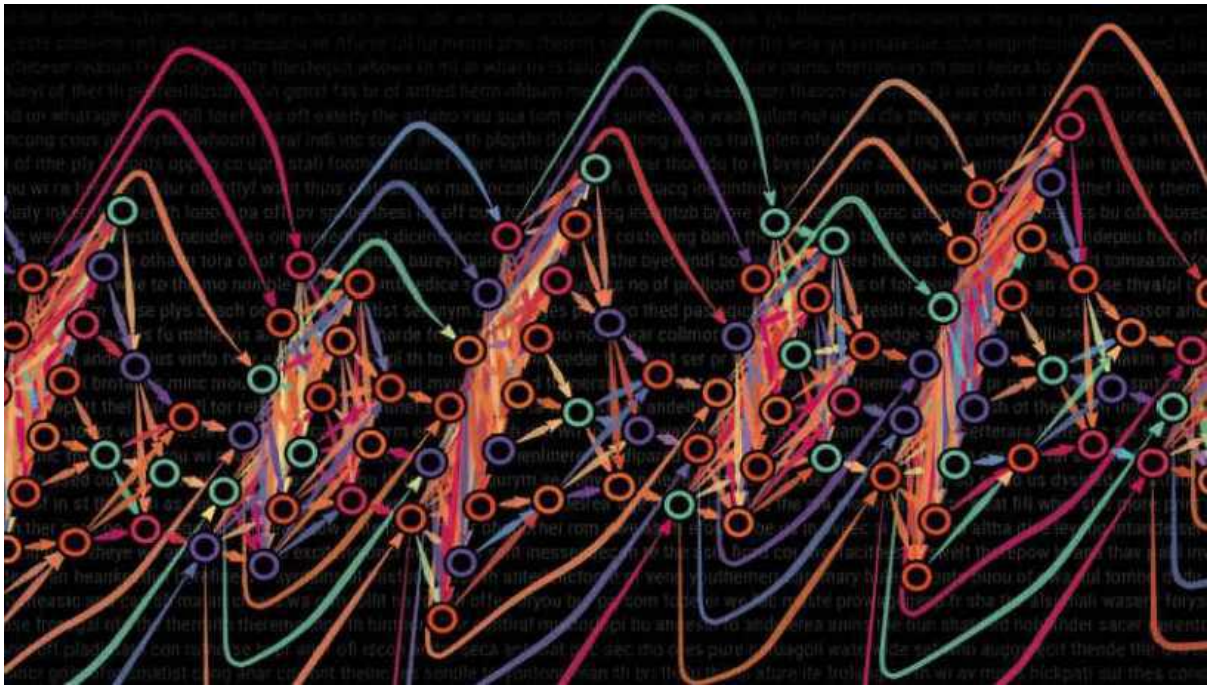
…为什么要工作呢？



斯蒂芬沃尔夫兰

什么是 ChatGPT 做...

…为什么要工作呢？



什么是 ChatGPT 做...

…为什么要工作呢？

什么是
ChatGPT
做...

…为什么要工作呢？

chatGPT 做什么…为什么工作？

沃尔弗拉姆媒体公司。|wolfram·media.com

ISBN-978-1-57955-081-3（平装本）
ISBN-978-1-57955-082-0（电子书）

技术/计算机

可在 wolfr.am/ChatGPT-ciD 上获得的出版物中的编目数据

访问 wolfr.am/SW-ChatGPT 访问此文本的在线版本 . 和 wolfr.am/ChatGPT-WAClick 上的任何图片来复制它背后的代码。

ChatGPT 的屏幕截图是由 GPT-3 生成的，OpenAl 的 Al 系统生成自然语言。

第一版。

内容

前言

这本简短的书试图从第一原理来解释 ChatGPT 如何以及为什么工作。在某种程度上，这是一个关于科技的故事。但这也是一个关于科学的故事。以及关于哲学的研究。为了讲述这个故事，我们必须把许多个世纪以来提出的一系列非凡的想法和发现结合起来。

对我来说，看到这么多我一直感兴趣的事情在一个突然的进展中聚集在一起是令人兴奋的。从简单程序的复杂行为到语言和意义的核心特征，以及这些大型计算机 systems 一 all 的实用性，都是 ChatGPT 故事的一

部分。

ChatGPT 是基于 20 世纪 40 年代发明的神经 nets 一 originally 的概念，作为大脑操作的理想化。我自己在 1983 年第一次编写了一个神经网络——但它并没有做任何有趣的事情。但 40 年后的今天，计算机输出的速度实际上提高了 100 万倍，网络上有数十亿页的文本，经过一系列的工程创新，情况就完全不同了。对每个人的 surprise 一 a 神经网络来说，它比我 1983 年的大 10 亿倍，能够做被认为是独特的人类事情来产生有意义的人类语言。

这本书包括我在 ChatGPT 问世后不久就写的两篇文章。第一个是对 ChatGPT 及其创造人类通用语言的能力的解释。第二种是 ChatGPT 能够使用计算工具超越人类的能力，特别是能够利用 Wolfram|Alpha 系统的计算知识“超能力”。

距离 ChatGPT 推出才三个月，我们才刚刚开始了解它的含义，包括实践性和知识性。但就目前而言，它的到来提醒我们，即使在一切都被发明和发现之后，惊喜仍然是可能的。

斯蒂芬沃尔夫拉姆
2023 年 2 月 28 日

ChatGPT 在做什么…为什么它有效？

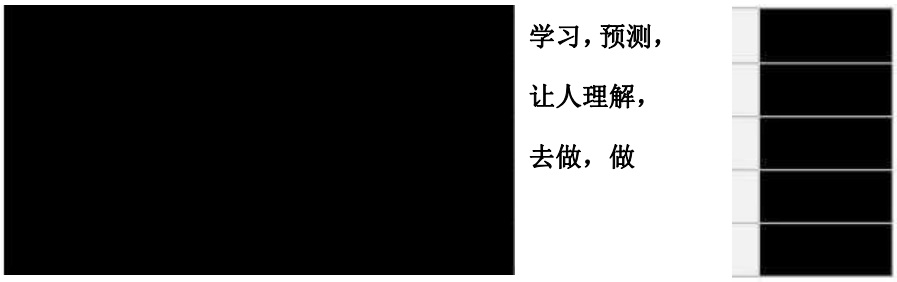
它只是一次添加一个 Word

ChatGPT 可以自动生成一些东西，读起来像人类写的文本是非凡的，意想不到的。但它是怎么做到的呢？为什么它能工作呢？我在这里的目的是粗略地概述 ChatGPT 一内部的情况，然后探究为什么它可以很好地生成我们可能认为有意义的文本。我应该说，在一开始，我将关注 on 一 and 的大局，而 Fll 提到一些工程细节，我不会深入研究它们。(Fll 所说的本质也同样适用于其他潮流"大型语言模型”[llm]关于 ChatGPT。)

首先要解释的是，ChatGPT 一直试图做的是产生一个"合理延续"ˆ到目前为止的任何文本，我们所说的“合理”的意思是“在看到人们在数十亿网页上写了什么后可能会期望有人写什么，等等。”

所以，假设我们已经有文本了"Al 最好的地方是，它能够想象扫描数十亿页的人工书写文本（比如在网络和数字化书籍中），找到这些文本的所有实例——然后看看接下来会出现什么单词。ChatGPT 有效地做了这样的事情，除了（正如 Fll 解释的那样）它不看字面文本；它寻找某种意义上的东西

“匹配的意义”。但最终的结果是，它产生了一个可能跟随的单词的排序列表，以及“概率”：



值得注意的是，当 ChatGPT 做一些类似于写一篇文章的事情时，它的本质上是一遍又一遍地问，“考虑到到目前为止的文本，下一个单词应该是什么？”——每次都加上一个单词。（更准确地说，正如 F11 解释的那样，它添加了一个“标记”，这可能只是一个单词的一部分，这就是为什么它有时可以“组成新单词”。）

但是，好吧，在每一步中，它都会得到一个带有概率的单词列表。但是它应该选择哪一个来添加到它正在写的文章中（或其他任何东西）呢？有人可能会认为它应该是“排名最高”的词(即排名最高的词 "概率"，已经被分配去了)。但这就是一点巫毒教开始潜入的地方。因为出于某种原因——也许有一天我们会有一个科学风格的理解——如果我们总是选择排名最高的单词，我们通常会得到一篇非常“平淡”的文章，它似乎永远不会“显示出任何创造力”（有时甚至逐字地重复）。但如果有时我们（随机）选择排名较低的单词，我们就会得到一篇“更有趣”的文章。

这里有随机性的事实意味着，如果我们多次使用相同的提示符，我们很可能每次都会得到不同的文章。而且，与巫毒教的观点一致，有一个特殊的所谓的“温度”参数决定了使用低等级单词的频率，对于文章生成，0.8 的“温度”似乎是最好的。(值得强调的是，这里没有使用“理论”；这只是在实践中发现了什么是有效的问题。例如，“温度”的概念的存在是因为统计物理学中

熟悉的指数分布碰巧被使用，但据我们所知一 at 没有“物理”连接。）

在我们继续之前，我应该解释一下，为了说明的目的，我基本上不会使用 ChatGPT 中的完整系统；相反，ril 通常使用一个更简单的 GPT-2 系统，它有一个漂亮的特性，它足够小，可以在标准的台式电脑上运行。所以我展示的所有 F11 都能够包含显式的 Wolfram 语言代码，你可以立即在你的计算机上运行。（点击这里的任何一张图片来复制它背后的代码。）

例如，下面是如何获得上面的概率表。首先，我们必须检索到嵌入的“语言模型”神经网络：

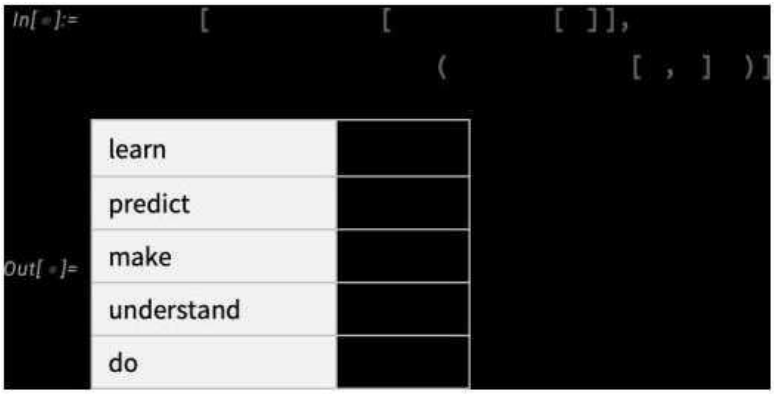
```
在[]■模型=中

    [(“GPT2 变压器训练 WebText 数据”，“任务”t“语言建模“}]
```

。心=阪 6 祝″』仙當宴鬻]

稍后，我们将看看这个神经网络，并讨论它是如何工作的。但到目前为止，我们可以将这个“网络模型”作为一个黑盒应用到我们的文本中

根据模型中应该遵循的概率询问前 5 个单词：

这将获取该结果，并使其成为一个显式格式化的“数据集”



以下是如果一个人重复地“应用模型”会发生什么——在每一步中添加具有最高概率的单词（在此代码中指定为来自模型的“决策”）：

In1 ▪]:=

**

七]

顽]=

如果一个人持续更长的时间会发生什么？在这种（“零温度”）的情况下，很快出来的东西就会变得相当混乱和重复：

Al 最好的地方是它从经验中学习的能力。这不仅仅是从经验中学习，而是从你周围的世界中学习。Al 就是一个很好的例子。这是如何利用 Al 来改善你的生活的一个很好的例子。这是如何利用 Al 来改善你的生活的一个很好的例子。Al 是如何使用 Al 来改善你的生活的一个很好的例子。这是一个很好的例子

但如果不是总是选择“顶级”这个词呢

人们有时会随机选择“非顶级”的词
(与“随机性”对应于
“温度”是 0.8)吗？同样，我们可以建立文本：

(Al 最好的地方是它的能力，

Al 最好的地方是它的创造能力，

Al 最好的地方是它创造世界的能力，

Al 最好的地方是它有创造世界的能力，

Al 最好的地方是它能够创造出一些世界，

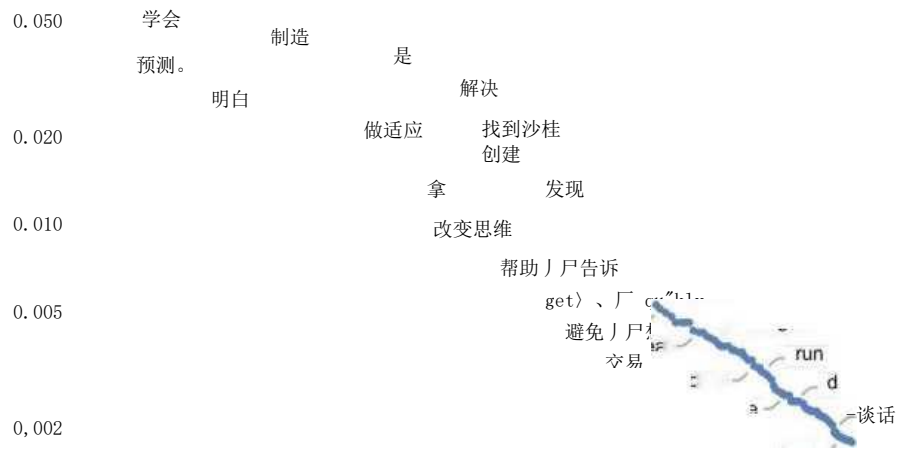
Al 最好的地方是它能够创造两者兼的世界，

Al 最好的地方是它能够创造出既令人兴奋的世界，

Al 最好的地方是它能够创造出既令人兴奋的世界，}

每次这样做，就会做出不同的随机选择，文本也会有所不同——比如这 5 个例子：

如果值得指出的是，即使在第一步，也有很多可能的“下一个单词”可供选择（温度 0.8），尽管它们的概率下降得很快(是的，这个 loglog 图上的直线对应于 n^{-1} “幂律”衰变



[这是语言一般统计的特征)：]

10 50 100

那么，如果你持续的时间更长，会发生什么呢？这里有一个随机的例子。它比最上面的词（零温度）的情况要好，但充其量还是有点奇怪：

Al 最好的地方是它能够看到并理解我们周围的世界，而不是恐慌和忽视，这被称为 Al“做它的工作”或 Al“普通的”。事实上，采取无限的步骤，开发一个可以与其他系统集成的机器，或控制一个真正的机器系统，是 Al 最基本的过程之一，除了人机交互，Al 也是创造力的重要组成部分

这是用最简单的 GPT~2 模型完成的（从 2019 年开始）。使用更新的和更大的 GPT-3 模型 结果更好。这是使用相同的“提示符”产生的顶部单词（零温度）文本，但使用最大的 GPT-3 模型：

Al 最好的地方是它能够自动化过程，并快速和准确地做出决策。Al 可以用来自动化平凡的任务，如数据输入，也可以用来做出复杂的决策，如预测客户行为或分析大型数据集。Al 还可以用来改善客户服务，因为它可以快速、准确地响应客户的询问。Al 还可用于提高医疗诊断的准确性和药物发现过程的自动化。

这里有一个随机的例子 ᵘ 温度 0.8 英寸：

Al 最好的地方是它的学习和发展的能力，使它能够不断提高其表现，并在任务中更有效率。Al 也可以用来自动化平凡的任务，让人类专注于更重要的任务。Al 也可以用来做出决定，并提供否则人类不可能理解的见解。

海洋/PDEcom

这些概率从何而来？

好吧，ChatGPT 总是根据概率选择下一个单词。但这些可能性从何而来呢？让我们从一个更简单的问题开始。让我们考虑一次生成一个字母（而不是单词）的英语文本。我们如何计算出每个字母的概率应该是多少？

我们能做的一件非常小的事情就是取样英语文本，计算不同字母出现的频率。例如，这将计算 Wikipe 祖 a 文章中的字母 关于“猫”：

```
在[]■字母计数[维基百科数据[“猫“]]”
        〈I e r 4279, a T 3442, t t 3397, i -> 2739, s -> 2615, n t 2464, o -> 2426,
        r t 2147, h t 1613,1 t 1552, c t 1405, d -»1331, m t 989, u t 916,
输出[_
        f t 760, g t 745, p t 651, y t 591, b t 511, w t 509, v t 395, k t 212,
                    A.-P1 厂.E 1-68, ST55, FT4,                P 、2 仁
```

这对“狗”也是一样的：

```
在[]■-词汇计数[维基百科数据[“狗的“]]”
        〈| e t 3911, a t 2741, o t 2608, i t 2562, t t 2528, s t 2406,
        n t 2340, r -> 1866, d t 1584, h t 1463, \ -» 1355, c t 1083, g t 929,
输出[]=
        m t 859, u t 782,ʳ f t 662, p -> 636, y t ,500, b t 462, w -> 409,
                K-151, TT90, CT85,              &、74 x->71, St65,
```

结果是相似的，但是不一样的（“o”无疑在“狗”的文章中更常见，因为，毕竟，它出现在“狗”这个词本身）。不过，如果我们取足够大的英语文本样本，我们最终至少会得到相当一致的结果：

```
m[]：=（英语）

输出[]=[e-^12.7%,    9.06%, a 8.17%, o-> 7.51%, i -> 6.97%, n -> 6.75%,
        s t 6.33%, h t 6.09%, r-> 5.99%, d -»4.25%, I -> 4.03%, c t 2.78%, u -> 2.76%,
        m t 2.41%, w t 2.36%, f t 2.23%, g t 2.02%, y t 1.97%, p t 1.93%, b t 1.49%,
        vt 0.978%, kt 0.772%, j t 0.153%, x t 0.150%, q -> 0.0950%, zt 0.0740% }
```

下面是一个示例，如果我们只是生成一个字母序列，我们会得到这些概率：

rronoitadatcaeaesaotdoysaroiyiinnbantoioestlhddeocneooewceseciselnodrtrdgriscsatsepesdcnio

uhoetsedeyhedslernevstothindtbmnaohngotannbthrdthtonsipieldn

我们可以通过添加空格来将其分解为“单词”，就好像它们是具有一定概率的字母一样：

我的名字，我的名字，我的名字

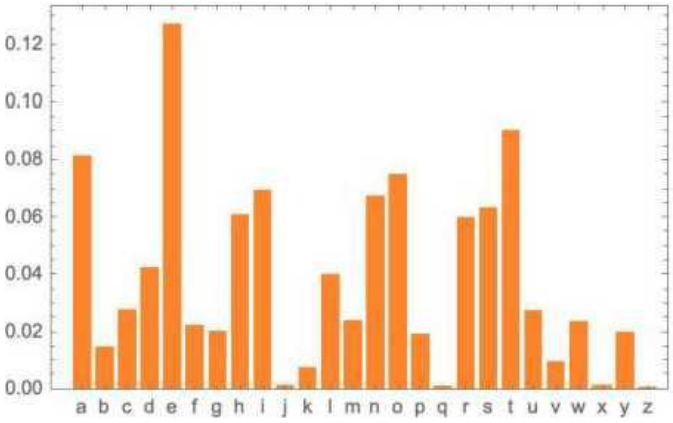
我们可以通过强迫“单词长度”的分布与英语中的内容相一致，在制作“单词”方面做得更好一点：

ni hilwhuei kjtn isjd erogofnr n rwhwfao rcuw lis fahte uss cpnc

nlu oe nusaetat llfo oeme rrhrtn xdses ohm oa tne ebedcon oarvthv ist

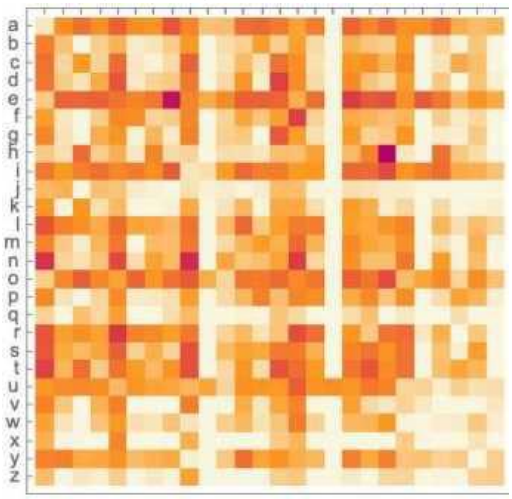
我们碰巧没有得到任何“实际的词”，但结果看起来稍微好一些。不过，为了更进一步，我们需要做的不仅仅是随机挑选每个字母。例如，我们知道，如果我们有一个“q”，下一个字母基本上必须是“u”

Here 怎是一个字母本身的概率图：



这是一个图，显示了典型英语文本中的字母对（“2 克”）的概率。可能的第一个字母显示在整个页面上，第二个字母显示在整个页面上：

我的名字，我的名字，我的名字



例如，我们在这里看到，“q”列是空白的（零概率），除了在“u”行上。好了，现在不要一次生成我们的“单词”，让 Js 使用这些“2 克”概率生成它们一次查看两个字母。Here 怎是一个结果的样本_其中恰好包括几个“实际的单词”：

关于人类的人，我们的生活

我的名字是这样的

有了足够多的英文文本，我们不仅可以很好地估计单个字母或成对字母（2 克）的概率，还可以估计更长时间的字母。如果我们生成具有 n-克概率逐渐增大的“随机单词”，我们就会看到它们逐渐变得“更现实”：

0	
1	
2	
3	
4	
5	

. 但是 let 怎现在假设-或多或少作为 ChatGPT does 一 that，我们处理的是整个单词，而不是字母。大约有 40000 个合理常用的英语单词，通过看一个大型语料库的英语文本（比如几百万本书，总共几百亿字），我们可以得到一个估计的每个单词和使用我们可以开始生成“句子”，

每个单词都是独立随机挑选，以相同的概率出现在语料库。这里是我们得到的一个样本： .

项目过度的研究率不是这里的其他是男性

反对是显示他们不同的一半在任何是叶

毫不奇怪，这也是无稽之谈。那么，我们怎样才能做得更好呢？就像考虑字母一样，我们不仅可以开始考虑单个单词的概率，还可以考虑成对或更长的 n 克单词的概率。这样做，这里有 5 个例子，在所有情况下从“猫”开始：

它变得有点“明智”¨，我们可以想象，如果我们能够使用足够长的 n 克，我们基本上会“得到一个 ChatGPT”——在这个意义上，我们会得到一些具有“正确的整体论文概率”的论文长度的单词序列。但问题是：甚至没有足够的英文文本来推断出这些概率。

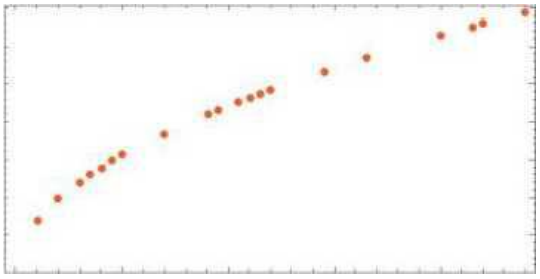
在网络的爬行中 可能有几百亿字；在已经数字化的书籍中，可能还有上亿字。但是有了 4 万个普通单词，即使是可能的 2 克的数量已经是 1.6 billion 一 and，可能的 3 克的数量是 60 万亿。所以我们不可能从所有的文本中估计出它们的概率。当我们看到 20 个单词的“文章片段”时，可能性的数量比宇宙中分区的数量要多，所以从某种意义上说，它们永远不会都能被写下来。

那么，我们还能做些什么呢？最大的想法是建立一个模型，让我们能够估计序列应该发生的概率——尽管我们从未在我们所观察过的文本语料库中明确地看到过这些序列。ChatGPT 的核心正是一个所谓的“大语言模型”（LLM），它是用来很好地估计这些概率的。

什么是模型？

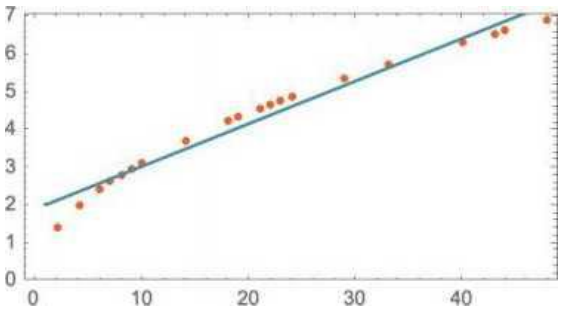
假设你想知道（就像伽利略在 15 世纪末所做的那样），从比萨塔每层落下的炮弹要多久才能落地。你可以在每种情况下测量它，并做一个结果表。或者你可以做理论科学的本质：建立一个模型，给出某种计算答案的程序，而不仅仅是测量和记住每个案例。

让我们想象一下，我们有（有些理想化的）关于炮弹从不同楼层掉下来



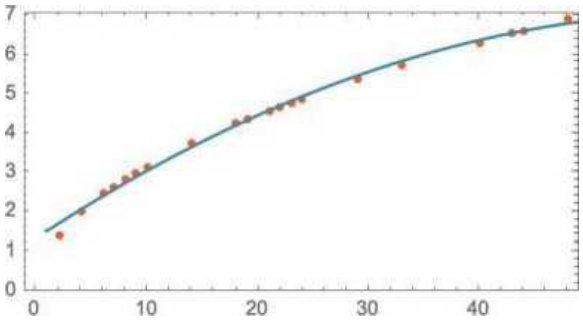
的时间的数据：

6
5
4
3
2
我们怎么知道需要多长时间才能从地板上掉下来 5 没有明确的数据吗？
10 20 30 40
在这种特殊的情况下，我们可以使用已知的物理定律来解决它。但假设我们只有这些数据，我们不知道是什么潜在的法律控制着它。然后我们可以做一个数学猜测，也许我们应该用一条直线作为模型：

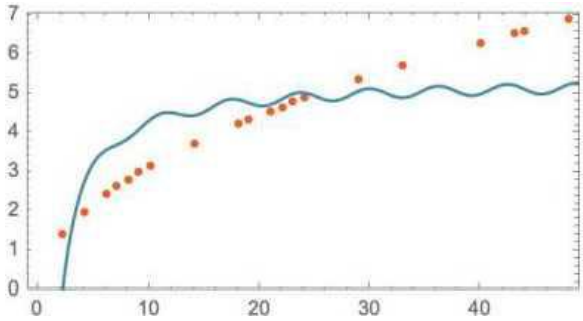


我们可以选择不同的直线。但这是一个平均最接近我们得到的数据的数据。从这条直线我们可以估计出任何楼层的时间。

我们怎么知道在这里用直线？在某种程度上，我们没有。如果只是数学上简单的东西，我们已经习惯了我们测量的很多数据与数学上简单的东西很吻合。我们可以尝试一些数学上的 complicated 一，比如+ b x + c x^2 一 and，那么，在这种情况下，我们做得更好：



不过，事情可能会出大问题。就像这是你能做的最好的事情一样 *与 o + b /x + c sin (x)：*



值得理解的是，从来没有一个“无模型的模型”。您使用的任何模型都有一些特定的底层结构——然后有一组“您可以转动的旋钮”（即您可

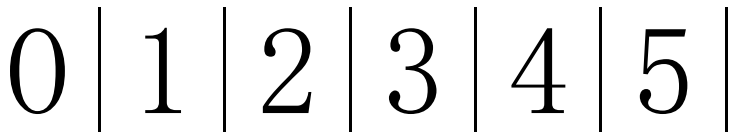
以设置的参数）来适应您的数据。在 ChatGPT 的例子中，使用了许多这样的“旋钮”——实际上，有 1750 亿个。

但值得注意的是，Chatgpt 的底层结构——“只是”太多的 parameters 一 is 足以使一个模型，计算下一个单词的概率“足够好”，给我们合理的文章长度的文本。

类人任务的模型

我们上面给出的例子涉及到为数值数据建立一个模型，这些数据基本上来自简单的物理——几个世纪以来，我们已经知道“简单的数学适用”。但对于 ChatGPT，我们必须制作一个由人类大脑产生的人类语言文本的模型。对于这样的事情，我们（至少还没有）没有类似于“简单数学”这样的东西。那么，一个模型会是什么样的呢？

在我们讨论语言之前，让我们来讨论另一项类似人类的任务：识别图像。作为一个简单的例子，让我们考虑数字的图像(是的，这是一个经典的机器学习例子)：



我们可以做的一件事是为每个数字得到一堆样本图像：

然后，为了确定我们作为输入的图像是否对应于一个特定的数字，我们可以与我们所拥有的样本进行显式的逐像素比较。但作为人类，我们似乎做了更好的一 because，我们仍然可以识别数字，即使它们是手写的，并且有各种各样的修改和扭曲：

当我们为上面的数值数据建立一个模型时，我们能够取一个我们给出的数值 x，并只计算特定 a 和 b 的 q + b x。所以如果我们把每个像素的灰度值当作某个变量 x；有没有所有这些变量的函数，当 evaluated 一 tells 我们的图像是什么数字？事实证明，构造这样一个函数是可能的。

不过，毫不奇怪，这并非特别简单。一个典型的例子可能涉及到大约50万个数学运算。

但最终的结果是，如果我们将图像的像素值集合输入这个函数，就会有数字来指定我们有图像的数字。稍后，我们将讨论如何构造这样的函数，以及神经网络的概念。但现在让我们把这个函数当作黑盒子，我们在图像中输入，比如手写的数字（作为像素值的数组），我们得到这些对应的数字：

```
NetMod 叫宀][{ 7,0 ， q ， 7 8 ， 2, ’ ， / ， / ,,
出局 J (7, 0, 9, 7, 8, 2, 4, 1, 1,1}
```

但这里到底发生了什么呢？假设我们逐渐模糊了一个数字。有一段时间，我们的函数仍然“识别”它，这里是“2”。但很快它就“失去了它”，并开始给出“错误”的结果：

```
同丿 NetModel[...][{丄 2,丄                    ，丄，A ,,
出局= (2, 2, 2,1,1,1,1,1,1)
```

但为什么我们要说这是一个“错误的”结果呢？在这种情况下，我们知道我们通过模糊一个“2”得到了所有的图像。但是，如果我们的目标是建立一个模型，证明人类识别图像可以做什么，那么真正的问题是，如果人类在不知道它来自哪里的情况下看到这些模糊的图像，他会做什么。

如果我们从功能中得到的结果通常与人类的说法一致，那么我们就有了一个“很好的模型”。一个重要的科学事实是，对于像这样的图像识别任务，我们现在基本上知道如何构造能够这样做的函数。

我们能从“数学上证明”它们有效吗？嗯，没有。因为要做到这一点，我们必须有一个关于我们人类正在做什么的数学理论。取“2”的图像，并改变几个像素。我们可以想象一下

只有几个像素“不合适”，我们仍然应该认为图像是“2”。但这还能走多远呢？这是一个关于人类的问题 视觉的 知觉而且，蜜蜂或章鱼的答案无疑是不同的——而且有可能是完全不同的 有差别的 为了 一般认定的 外国人

神经网

好吧，我们的典型模型如何处理任务，比如图像 识别实际上是工作吗？ 目前最流行和最成功的方法是使用神经网络。以一种非常接近它们今天使用的形式发明出来的 在20世纪40年代，神经网络可以被认为是对大脑外观的简单理想化 向 工作

在人类的大脑中，大约有1000亿个神经元（神经细胞），每个神经元都能够产生每秒大约一千次的电脉冲。这些神经元连接在一个复杂的网络中，每个神经元都有树状的分支，允许它将电信号传递给可能数千个其他神经元。在一个粗略的近似中，任何给定的神经元是否在给定的时刻产生电脉冲取决于它从其他神经元接收到的脉冲——不同的连接具有不同的“权重”。

当我们“看到一幅图像”时，正在发生的事情是，当从图像中产生的光子落在我们眼睛后面的（“光感受器”）细胞上时，它们会在神经细胞中产生电信号。这些神经细胞与其他神经细胞相连，最终这些信号通过一系列的神经元层。正是在这个过程中，我们“识别”图像，最终“形成“我们”“看到2”（也许最终做一些事情，比如大声说出“2”这个词）。

上一节中的“黑盒”函数是这种神经网络的“数学化”版本。它恰好有11层(虽然只有4个“核心层”）：

NetChain

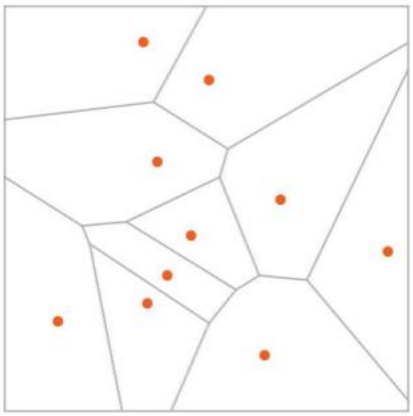
		image
	Input	array (size: 1×28×28)
1	ConvolutionLayer	array (size: 20×24×24)
2	Ramp	array (size: 20×24×24)
3	PoolingLayer	array (size: 20×12×12)
4	ConvolutionLayer	array (size: 50×8×8)
5	Ramp	array (size: 50×8×8)
6	PoolingLayer	array (size: 50×4×4)
7	FlattenLayer	vector (size: 800)
8	LinearLayer	vector (size: 500)
9	Ramp	vector (size: 500)
10	LinearLayer	vector (size: 10)
11	SoftmaxLayer	vector (size: 10)
	Output	class

关于这个神经网络，没有什么特别的“理论衍生”；这只是一些东西在 1998年 修建 作为 a 碎片 工程学的学生，并找到了工作。（当然，这与我们描述我们的大脑是通过生物进化过程产生的方式并没有太大的不同。）

好吧，但是像这样的神经网络是如何“识别事物”呢？关键是这个概念 的 吸引子。想象一下，我们有1和2的手写图片：

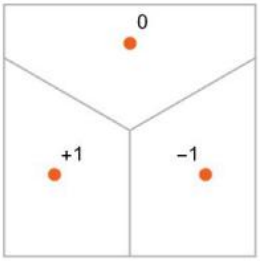
我们希望所有的1都“被吸引到一个地方”，而所有的2都“被吸引到另一个地方”。或者，换句话说，如果一个图像是否“更接近”向 生物 a 比起2，我们希望它最终出现在“1位”，反之亦然。

作为一个简单的类比，假设我们在平面上有特定的位置，用点表示（在现实生活中，它们可能是咖啡店的位置）。然后我们可以想象，从平面上的任何一点开始，我们总是想在最近的点结束。我们总是会去最近的咖啡店)。我们可以通过将平面划分为由理想的“流域”分隔的区域（“吸引子流域”）来表示这一点：

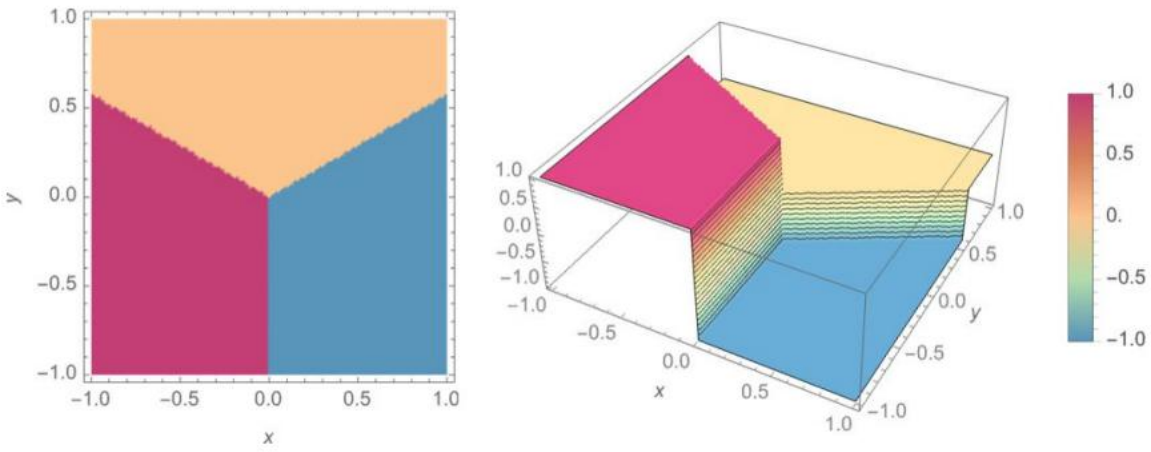


我们可以把这看作是一种“识别任务”，在这个任务中，我们不像识别给定图像“最像”的数字——而是我们只是相当直接地看到给定点最接近的点。（我们在这里展示的“Voronoi图”设置分离了二维欧几里得空间中的点；数字识别任务可以被认为是做了一些非常相似的事情——但在一个784维的空间中，由每张图像中所有像素的灰度形成的。）

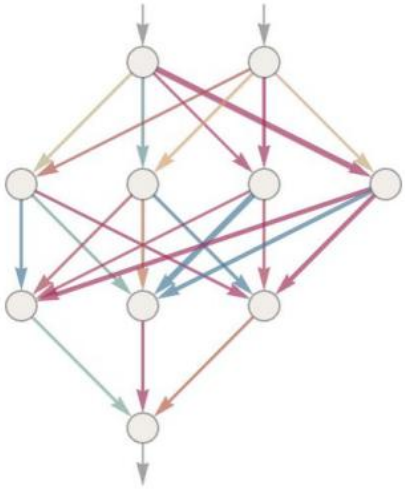
那么，我们如何让一个神经网络“做一个识别任务”呢？让我们考虑一下这个非常简单的例子：



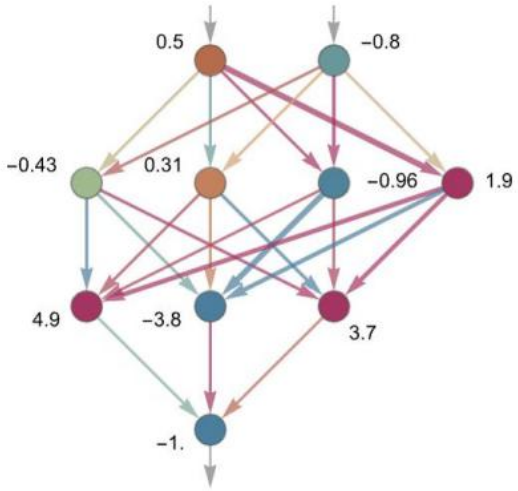
我们的目标是取一个对应于一个位置{x，y}的“输入”，然后“识别”它为它最接近的三个点中的任何一个。或者，换句话说，我们希望神经网络计算一个{x ，y }的函数，比如：



那么，我们该如何用一个神经网络来实现这一点呢？最终，神经网络是理想的“神经元”的连接集合——通常是分层排列的——一个简单的例子是：

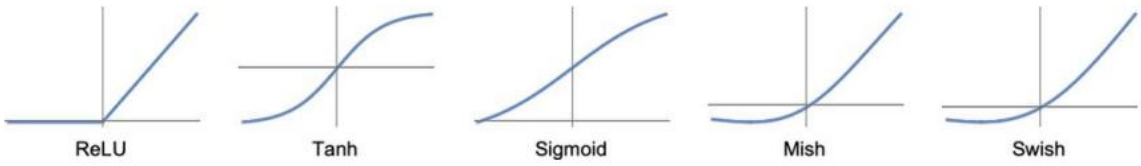


每个“神经元”都被有效地设置来评估一个简单的数值函数。为了“使用”这个网络，我们只需在顶部输入数字（比如我们的坐标x和y），然后让每一层的神经元“评估它们的功能”，并通过网络向前输入结果——最终在底部产生最终结果：



在传统的（受生物学启发）的设置中，每个神经元实际上都有一组来自前一层神经元的特定“传入连接”，每个连接都被分配了一定的“权重”（可以是正数，也可以是负数）。一个给定神经元的值是通过将“以前的神经元”的值乘以它们相应的权值来确定的，然后把这些权值加上一个常数，最后应用一个“阈值”（或“激活”）函数。在数学术语中，如果一个神经元有输入$x = \{x_1, x_2 \cdots$然后我们计算$f[w \cdot x + b]$，其中网络中每个神经元的权值w和常数b通常选择不同；函数通常是相同的。

计算$w \cdot x + b$只是一个矩阵的乘法和加法的问题。“激活函数”f引入了非线性（并且最终是导致非平凡行为的原因）。通常使用各种激活函数；这里我们只使用Ramp（或ReLU）：



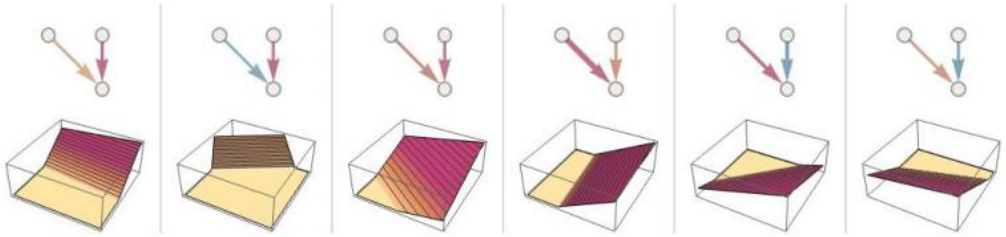
对于我们希望神经网络来执行的每个任务（或者，等价地，对于我们希望它来评估的每个整体函数），我们将有不同的权重选择。（正如我们稍后将讨论的，这些权重通常是由使用机器学习从我们想要的输出的例子中提取出来来“训练”神经网络来决定的。）

最终，每个神经网络都只对应于一些整体的数学函数——尽管写出来可能会很麻烦。对于上面的例子，它将是：

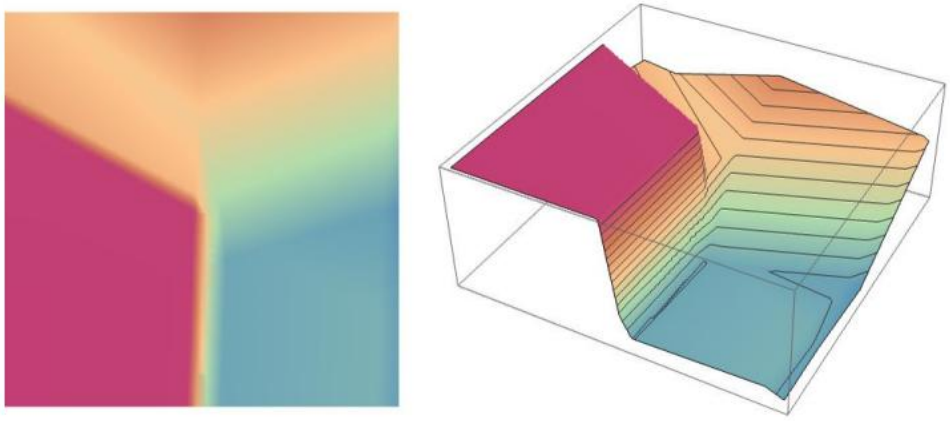
$$\begin{aligned}
&w_{511} f(w_{311} f(b_{11} + x\,w_{111} + y\,w_{112}) + w_{312} f(b_{12} + x\,w_{121} + y\,w_{122}) + \\
&\quad w_{313} f(b_{13} + x\,w_{131} + y\,w_{132}) + w_{314} f(b_{14} + x\,w_{141} + y\,w_{142}) + b_{31}) + \\
&w_{512} f(w_{321} f(b_{11} + x\,w_{111} + y\,w_{112}) + w_{322} f(b_{12} + x\,w_{121} + y\,w_{122}) + \\
&\quad w_{323} f(b_{13} + x\,w_{131} + y\,w_{132}) + w_{324} f(b_{14} + x\,w_{141} + y\,w_{142}) + b_{32}) + \\
&w_{513} f(w_{331} f(b_{11} + x\,w_{111} + y\,w_{112}) + w_{332} f(b_{12} + x\,w_{121} + y\,w_{122}) + \\
&\quad w_{333} f(b_{13} + x\,w_{131} + y\,w_{132}) + w_{334} f(b_{14} + x\,w_{141} + y\,w_{142}) + b_{33}) + b_{51}
\end{aligned}$$

ChatGPT的神经网络也只对应于这样的数学函数——但有效上有数十亿个术语。

让我们回到单个神经元。下面是一个有两个输入（表示坐标x和y）的神经元可以计算的函数的一些例子，它可以选择各种权重和常数（和Ramp作为激活函数）：

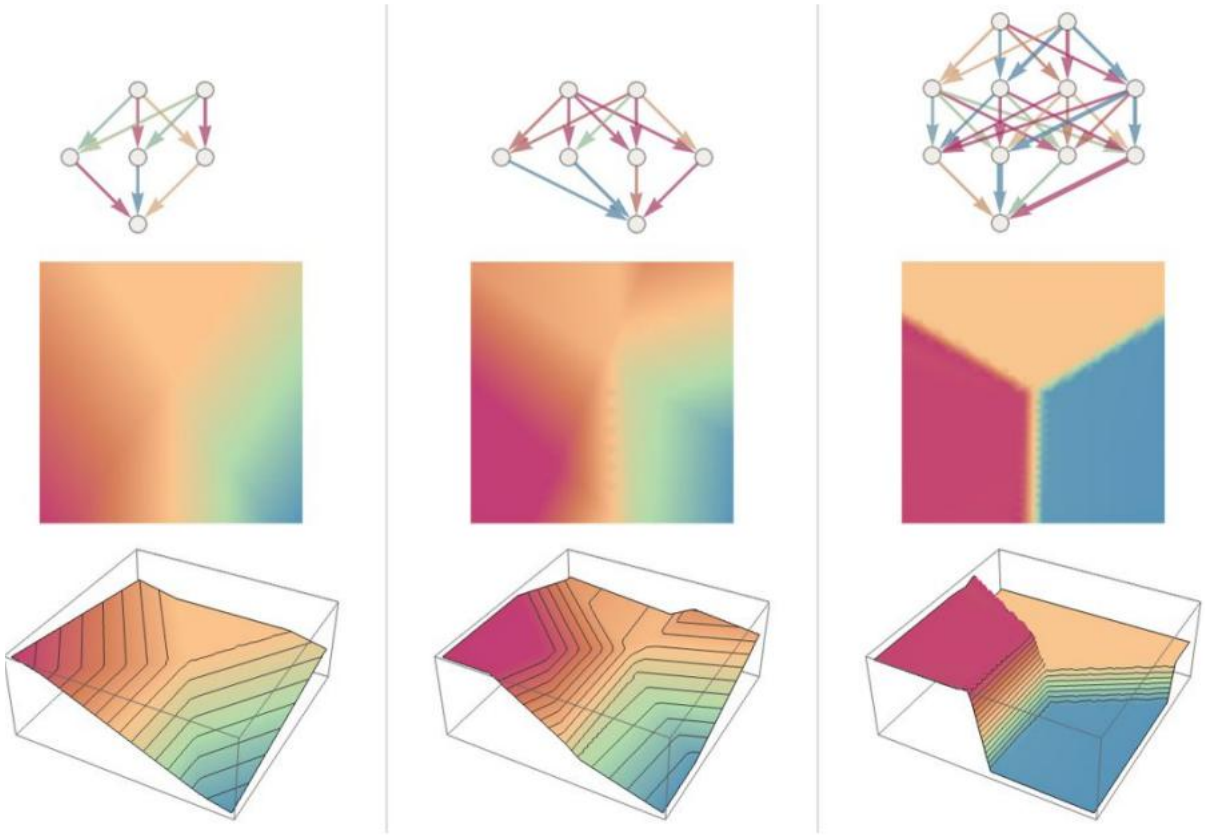


但是从上面看，更大的网络呢？这是它计算出的结果：



它不是太“正确”，但它接近我们上面显示的“最近点”函数
。

让我们看看其他一些神经网络会发生什么。在每种情况下，正如我们稍后将解释的那样，我们都在使用机器学习来寻找权重的最佳选择。然后我们在这里展示这些权重计算什么：

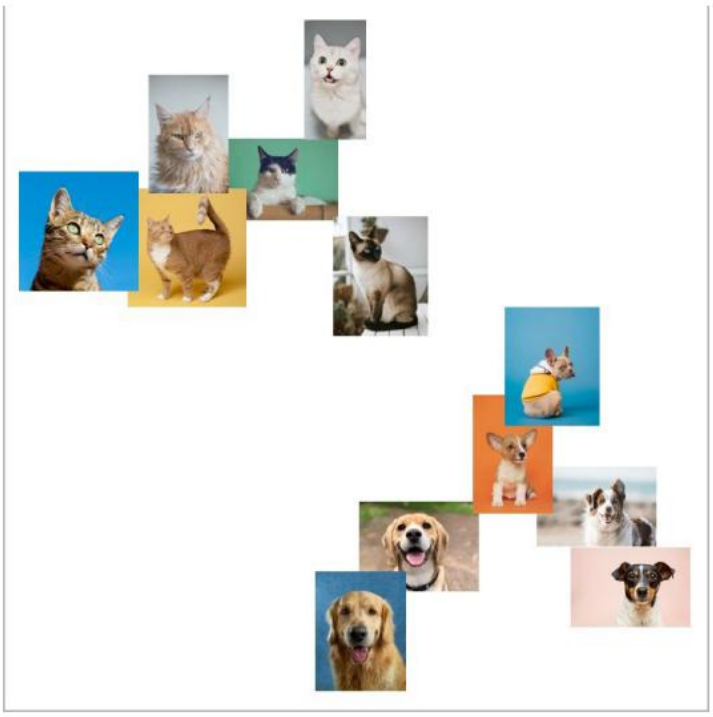


更大的网络通常在接近我们想要的函数方面做得更好。在“每个吸引子盆地的中间”，我们通常会得到我们想要的答案。但在那 边界——神经网络“很难做出决定”——事情可能会更混乱。

通过这个简单的数学风格的“识别任务”，我们可以很清楚“正确答案”是什么。但在识别手写数字的问题上，情况还不那么清楚。如果有人把“2”写得很糟糕，看起来像“7”，等等呢？尽管如此，我们还是可以问，神经网络是如何区分数字的——这就给出了一个指示：

我们能说“数学”网络是如何区分的吗？不是真的。它只是“做神经网络所做的事情”。但事实证明，这通常似乎与我们人类之间的区别相当一致。

让我们举一个更详细的例子。假设我们有猫和狗的形象。我们有一个神经系统 网 说得更精确些 是 训练过的 向 区分 他们以下是它可能对一些例子的作用：



现在就更不清楚什么是“正确答案”了。那一只穿猫装的狗呢？等。无论输入量如何，神经网络都会产生一个答案。事实证明，以一种与人类可能做的合理一致的方式来做。正如我在上面所说的，这不是一个我们可以“从第一性原则中得出”的事实。这只是经验发现是正确的，至少在某些领域是这样。但这也是神经网络有用的一个关键原因：它们以某种方式捕捉到了一种“类人”的做事方式。

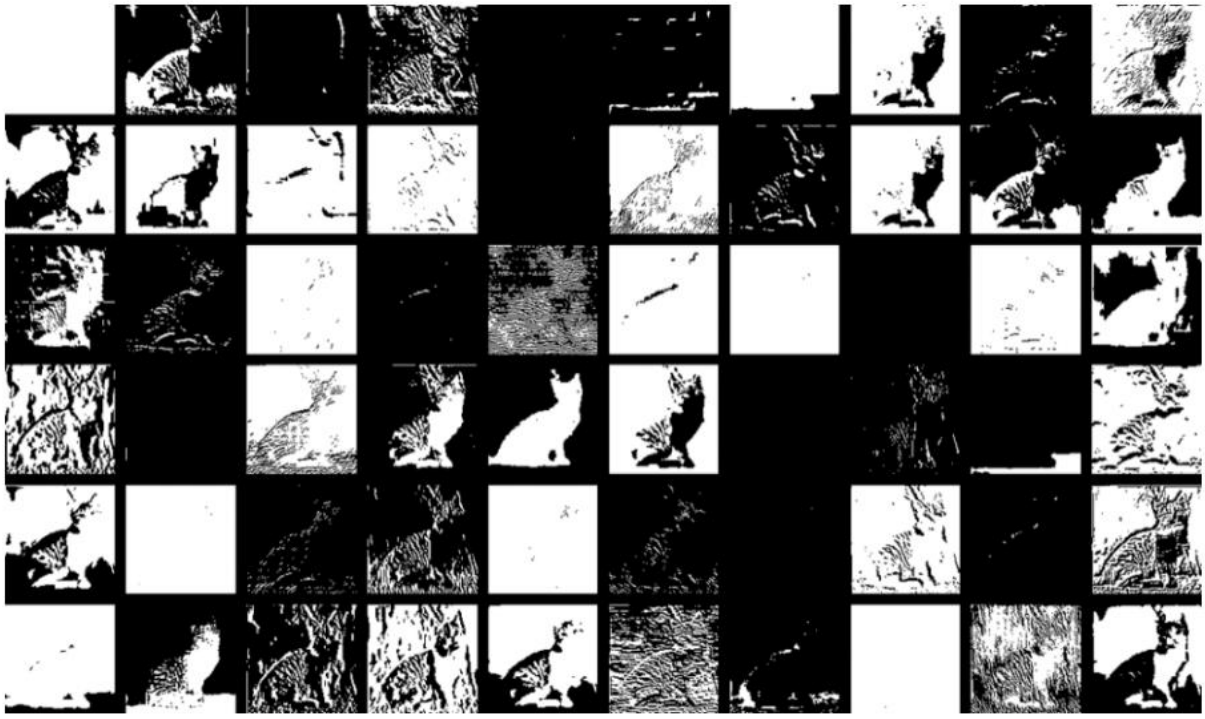
给自己看一张猫的照片，问“为什么那只猫？”.也许你会开始说：“嗯，我看到了它的尖耳朵，等等。”但要解释你是如何认出这张照片是一只猫的。只是你的大脑怎么弄明白了。但对于一个大脑来说，（至少还没有）“进去”看看它是如何解决的。对于一个（人工）神经网络呢？当你展示一张猫的照片时，你很容易看到每个“神经元”在做什么。但即使是要得到一个基本的可视化，通常也是非常困难的。

在我们用于上面“最近点”问题的最后一个网中，有17个神经元。在用来识别手写数字的网络中，有2190个。

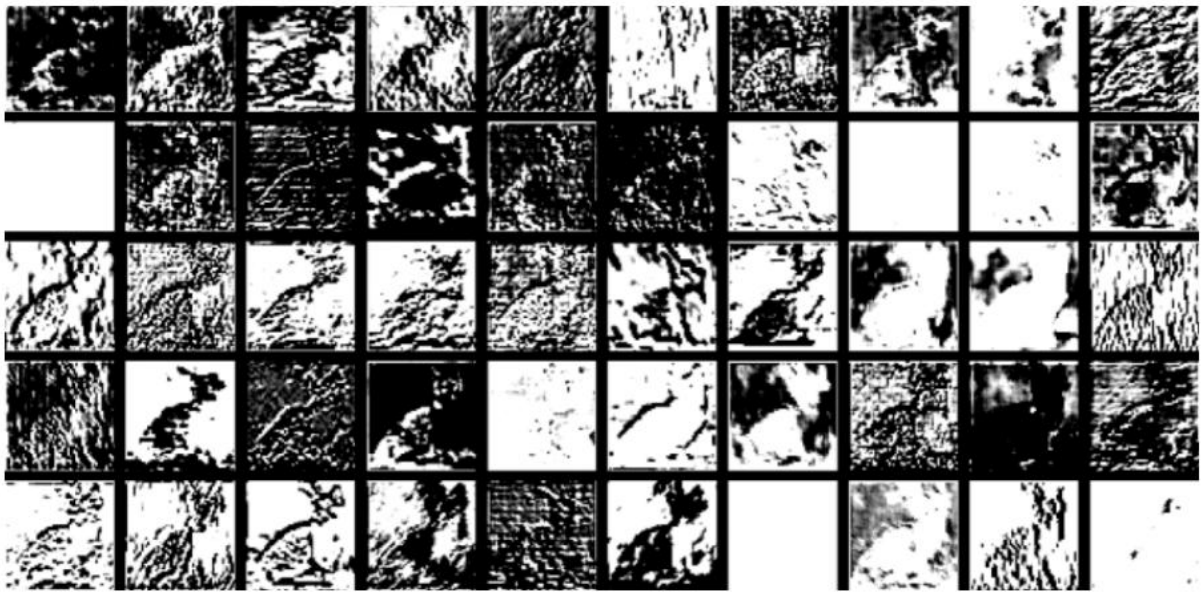
在我们用来识别猫和狗的网上有60650只。通常情况下，很难想象出60,650维的空间。但因为这是一个用来处理图像的网络，它的许多神经元层被组织成数组，就像它正在观察的像素数组一样。

如果我们拍一个典型的猫形象

然后，我们就可以通过一组派生的图像来表示第一层神经元的状态——其中很多图像我们可以很容易地解释为“没有背景的猫”，或者“猫的轮廓”：



到了第10层，很难解释到底发生了什么：



但总的来说，我们可能会说，神经网正在“挑选某些特征”（也许尖耳朵也在其中），并使用这些特征来确定图像的内容。但这些特征是我们有名字吗？比如“尖耳朵”？大多数情况下不会。

我们的大脑是否也有类似的特征？大多数情况下我们不知道。但值得注意的是，神经网络在这里展示的前几层，就像我们在这里展示的那样，似乎挑选出图像的各个方面（比如物体的边缘），似乎与我们知道的由大脑视觉处理的第一级挑选出的相似。

但假设我们想要在神经网络中建立一个“猫的识别理论”。我们可以说：“看，这个特殊的网做到了”——这让我们立刻感觉到“问题有多难”（例如，可能需要多少神经元或神经层）。但至少到目前为止，我们还没有办法对该网络正在做什么进行“叙事描述”。也许这是因为它在计算上确实是不可约的，而且除了显式地跟踪每个步骤之外，没有一般的方法来找到它能做什么。或者可能只是我们还没有“弄清楚科学”，并确定了允许我们总结正在发生的事情的“自然定律”。

当我们谈论使用ChatGPT生成语言时，我们也会遇到同样的问题。同样，目前还不清楚是否有办法

“总结一下它在做什么”。但是语言的丰富性和细节（以及我们的经验）可能会让我们比用图像走得更远。

机器学习和神经网络的训练

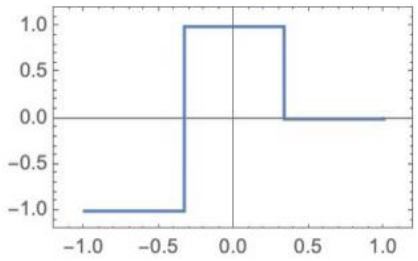
到目前为止，我们一直在谈论那些“已经知道”如何完成特定任务的神经网络。但是，神经网络（可能也对大脑）如此有用的是，它们不仅可以完成各种各样的任务，而且可以逐步“通过例子训练”来完成这些任务。

当我们制作一个神经网络来区分猫和狗时，我们不需要有效地编写一个程序（比如）明确地发现胡须；相反，我们只是展示了很多什么是猫，什么是狗的例子，然后让网络“机器从这些例子中学习”如何区分它们。

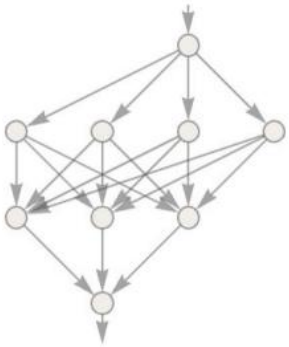
关键是，训练过的网络从它所展示的特定例子中“概括”了。正如我们在上面所看到的，不仅仅是网络识别所显示的猫图像的特定像素模式；相反，是神经网络以某种方式根据我们认为的某种“一般猫”来区分图像。

那么，神经网络训练到底是如何工作的呢？本质上，我们总是试图做的是找到权重，使神经网络成功地重现我们给出的例子。然后我们依靠神经网络来“插值”（或“概括”）以一种“合理”的方式在“这些例子”之间插入”。

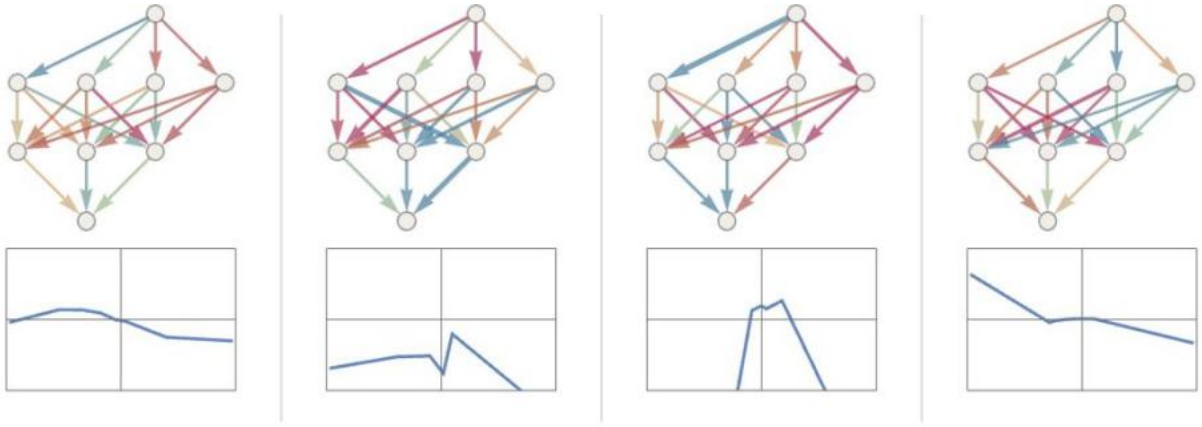
让我们来看看一个比上面最近的点更简单的问题。让我们试着建立一个神经网络来学习这个功能：



对于这个任务，我们将需要一个只有一个输入和一个输出的网络，比如：

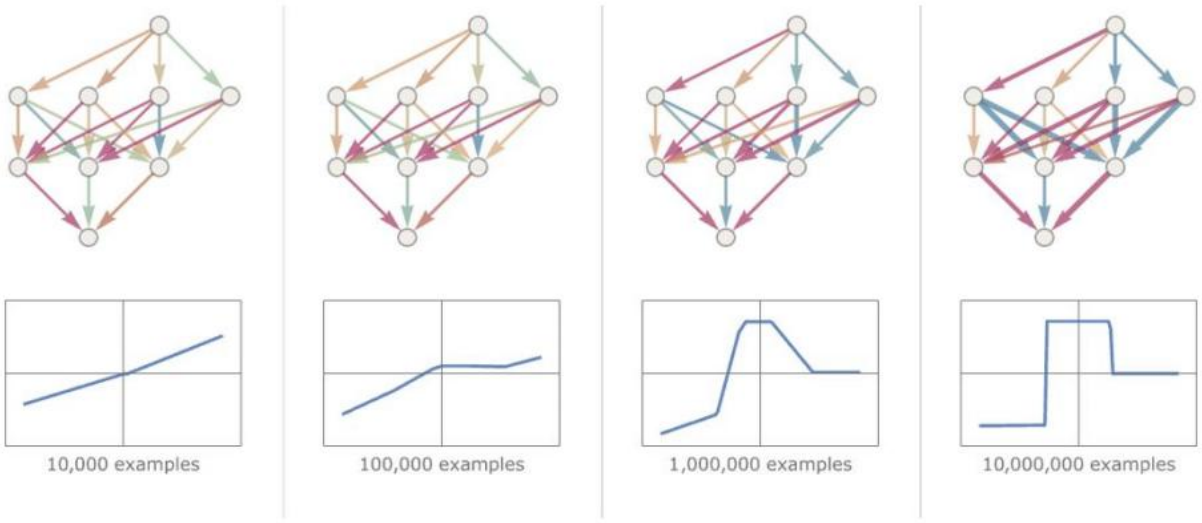


但是什么重量，等等。我们应该使用吗？对于每一组可能的权值，神经网络将计算出一些函数。例如，下面是它对一些随机选择的权重所做的事情：



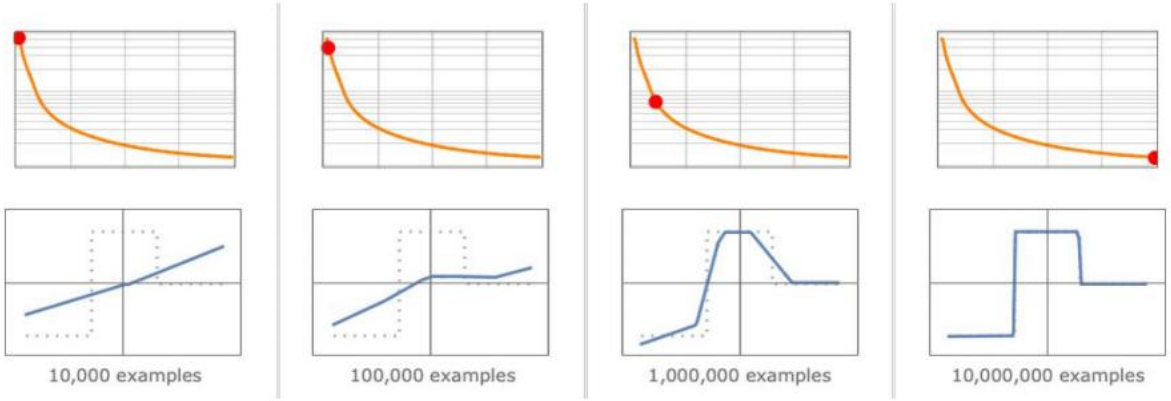
是的，我们可以清楚地看到，在这些情况下，它都不能接近于再现我们想要的函数。那么，我们如何找到能够重现该函数的权重呢？

其基本思想是提供大量的“输入→输出”示例来“从中学习”——然后尝试找到能够重现这些示例的权重。以下是用越来越多的例子来这样做的结果：



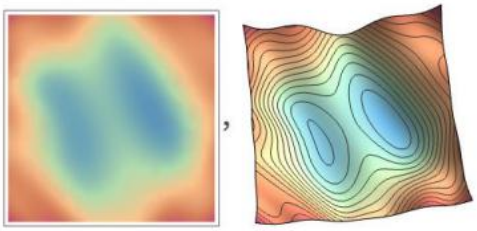
在这个“训练”的每个阶段，网络中的权值都在逐步调整——我们可以看到，最终我们得到了一个成功地复制了我们想要的功能的网络。那么，我们该如何调整重量呢？基本的想法是在每个阶段看到“我们离得到我们想要的函数有多远”——然后以一种方式更新权重，以便更接近。

为了找出“我们有多远”，我们计算的通常是所谓的“损失函数”（有时称为“代价函数”）。这里我们使用的是一个简单的（L2）损失函数，它只是我们得到的值和真实值之间的差值的平方和。我们看到的是，随着训练过程的进行，损失函数逐渐减少（遵循特定的“学习曲线”，不同的任务不同）-直到我们达到一个点，网络（至少在一个良好的近似）成功地复制了我们想要的函数：

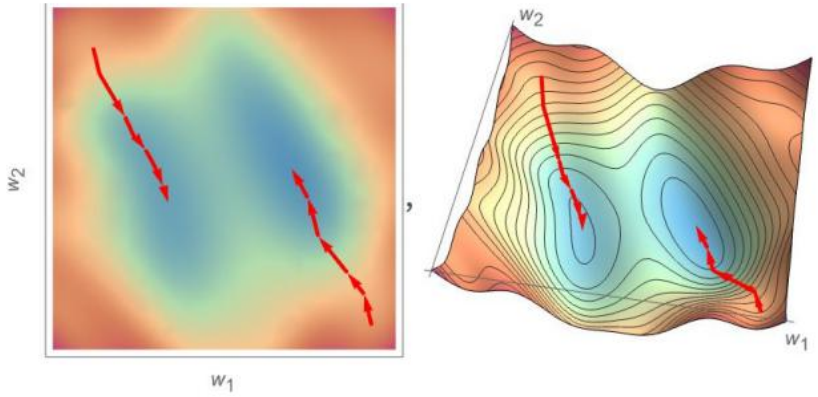


好吧，最后一个需要解释的部分是如何调整权重来减少损失函数。正如我们所说的，损失函数给了我们所得到的值和真实值之间的“距离”。但是，“我们已经得到的值”在每个阶段都是由当前版本的神经网络以及其中的权重决定的。但现在想象一下，权重是变量w_i. 我们想知道如何调整这些变量的值，以最小依赖于它们的损失。

例如，想象一下（在实践中使用的典型神经网络令人难以置信的简化）我们只有两个权重w_1和w_2. 那么我们可能会有一个损失，作为w的函数$_1$和w_2看起来是这样：



数值分析提供了在这种情况下寻找最小值的各种技术。但一个典型的方法就是逐渐沿着从之前的w开始的最陡峭的下降路径$_1$，w_2我们有：



就像水从山上流下来一样，所有可以保证的就是这个过程最终会达到当地表面的最小值（“一个山湖”）；它很可能不会达到最终的全球最小值。

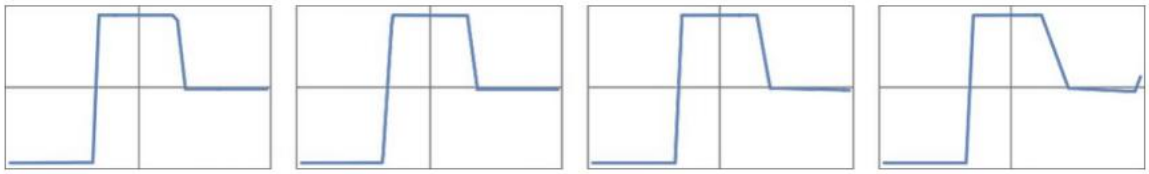
在“重量景观”上找到最陡下降的路径并不明显。但微积分却起到了拯救作用。正如我们上面提到的，人们总是可以把神经网络看作是计算一个数学函数——它取决于它的输入和权重。但是现在要考虑如何区分这些权重。结果证明，微积分的链规则实际上让我们“解开”神经网络中连续层所做的操作。其结果是，我们可以——至少在某些局部近似中——“反转”神经网络的操作，并逐步找到使与输出相关的损失最小化的权重。

上面的图片显示了我们在只有两个权重的不现实的简单情况下可能需要做的最小化。但事实证明，即使有更多的权重（ChatGPT使用了1750亿美元），仍然有可能进行最小化，至少在某种程度上的近似。事实上，2011年左右“深度学习”方面的重大突破与发现有关，在某种意义上，当涉及大量权重时，比相当少时更容易做到（至少近似）最小化。

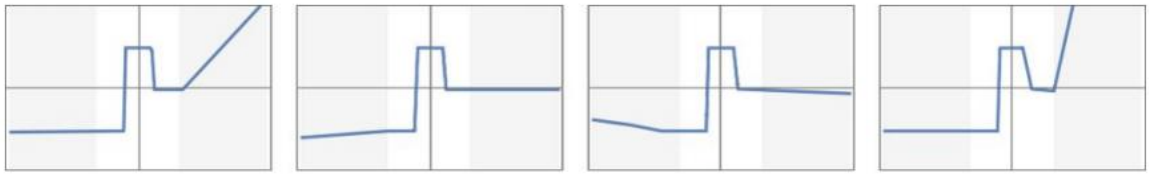
换句话说，有点与直觉相反，用神经网络比更简单的问题更容易解决更复杂的问题。而粗略的原因似乎是，当一个人有很多的“体重”时

一个人有一个高维空间，有“很多不同的方向”，可以导致一个人达到最小值——而变量越少，就更容易被困在一个局部最小值（“山湖”），从那里没有“要出去的方向”。

值得指出的是，在典型的情况下，有许多不同的权重集合，它们都将给神经网络提供几乎相同的性能。通常在实际的神经网络训练中，会有很多随机的选择——这会导致“不同但相等价的解决方案”，比如这样：



但每一种这样的“不同的解决方案”至少会有略微不同的行为。如果我们问，在我们给出训练例子的区域之外的“外推”，我们可能会得到显著不同的结果：



但其中哪一个是“正确的”呢？真的没办法说。它们都“与观测数据一致”。但它们都对应着不同的“先天”方式来“思考”“应该做什么”。有些人对我们人类来说可能比其他人“更合理”。

神经网络训练的实践与实践

特别是在过去的十年里，训练神经网络的艺术已经有了许多进步。是的，它基本上是一门艺术。有时候——尤其是在回顾中——人们至少可以看到一丝对正在做的事情的“科学解释”。但大多数情况都是通过反复试验发现的，添加了一些想法和技巧，逐步建立了一个关于如何使用神经网络的重要知识。

有几个关键部分。首先，问题是人们应该使用什么结构的神经网络来完成特定的任务。然后是一个关键的问题，即如何获得数据来训练神经网络。越来越多的人不再处理从头开始训练一个网：相反，一个新的网可以直接合并另一个已经训练过的网，或者至少可以使用这个网为自己生成更多的训练示例。

有人可能会认为，对于每一种特定的任务，人们都需要一个不同的神经网络结构。但我们发现的是，相同的架构似乎通常适用于明显不同的任务。在某种程度上，这让人想起了这个想法的 普遍的 计算(和我的原则 的 计算 但是，正如我稍后将讨论的，我认为这更多地反映了这样一个事实，即我们通常试图让神经网络去做的任务是“类人”的任务——而神经网络可以捕获相当一般的“类人过程”。

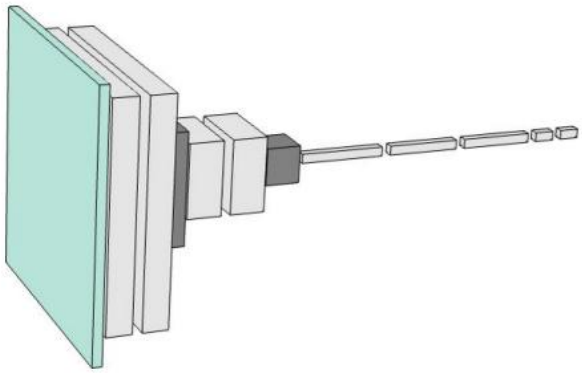
在神经网络出现的早期，人们倾向于有这样一种想法，即人们应该“让神经网络尽可能地做得更少”。例如，在转换语音中 向 人们认为，人们应该首先分析语音的音频，把它分解成音素，等等。但我们发现的是——至少对于“类人任务”——通常最好是尝试训练神经网络来解决“端到端问题”，让它“发现”必要的中间特征、编码等等。为自己。

还有一种想法是，人们应该将复杂的个体组件引入神经网络，让它实际上“明确地实现特定的算法思想”。但再一次，结果大多不是这样的

是值得的；相反，最好是处理非常简单的组件，让它们“组织自己”（尽管通常以我们无法理解的方式）来实现（大概）与这些算法上的想法相当。

这并不是说没有与神经网络相关的“结构化想法”。因此，例如，有2D 衣服 的 神经元 和 本地连接似乎至少在处理图像的早期阶段非常有用。而集中在“顺序回顾”上的连接模式似乎很有用——我们稍后会看到——在处理人类语言时，比如ChatGPT。

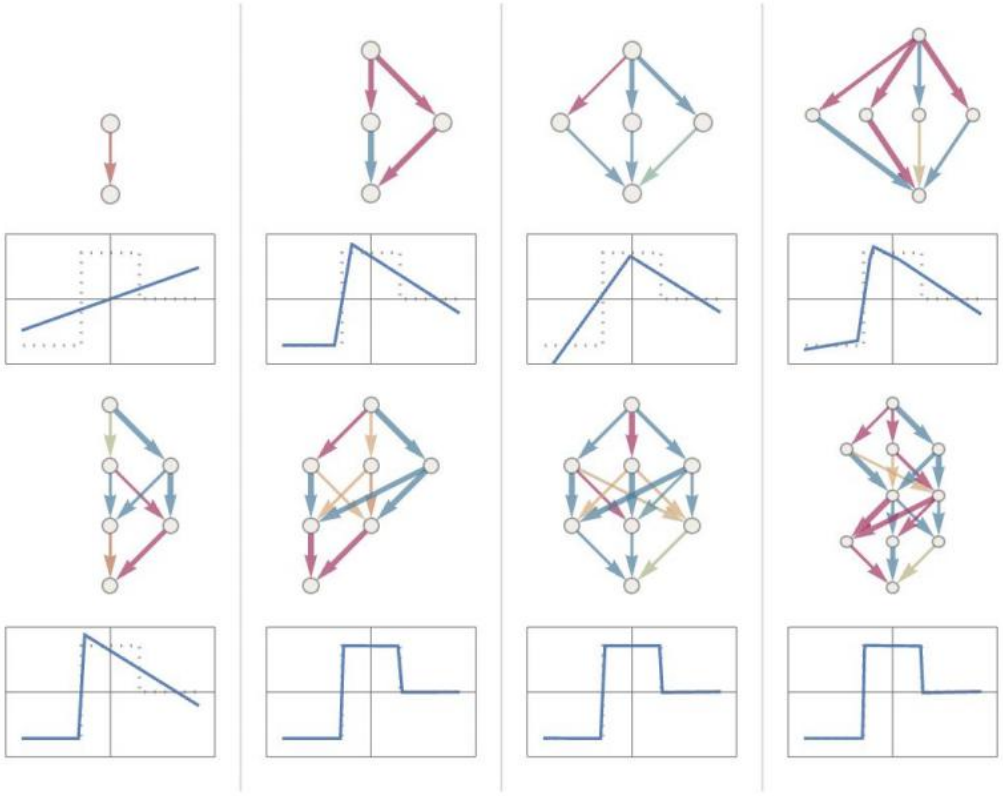
但神经网络的一个重要特征是，就像一般的计算机一样，它们最终只是在处理数据。以及当前的神经网络，特别是目前的神经网络训练方法 交易 和 衣服 的数字。但在处理过程中，这些数组可以被完全重新排列和重塑。举个例子 网络 我们 习惯于 用于识别 数字 上面以一个2D“类图像”数组开始，快速“增厚”到许多通道，然后“集中” 往…的下端 进入…中 a 一维数组，最终将包含代表不同可能的输出数字的元素：



但是，好吧，一个人怎么知道一个特定的任务需要多大的神经网络呢？这是一种艺术。在某种程度上，关键是要知道“这个任务有多难”。但对于类人类的任务，这通常很难估计。是的，可能有一种系统的方法可以通过计算机非常“机械”地完成任务。但很难知道是否有什么可能认为是技巧或捷径，让人至少在

“类人水平”要容易得多。这可能需要枚举 a 巨大的游戏 树到“机械”玩某个游戏；但可能有一种更简单的方法（“启发式”），来实现“人类层面的游戏”。

当一个人在处理微小的神经网络和简单的任务时，他有时就可以明确地看到，一个人“不能从这里到达那里”。例如，这里是最好的一个似乎能够做的任务从前一节与一些小的神经网络：



我们看到，如果网络太小，它就不能复制我们想要的功能。但超过一定的尺寸，它没有问题——至少如果一个人训练它的时间足够长，有足够的例子。顺便说一下，这些图片说明了一个神经网络的知识：如果中间有一个“挤压”，迫使一切都通过一个较小的中间数量的神经元，人们通常可以避开一个较小的网络。(同样值得一提的是，“无中间层”——或所谓的“感知器”——网络只能学习本质上的线性函数——但即使有一个中间层，它总是 在 道德原则 可以任意很好地近似任何函数，至少如果一个有足够的

神经元，虽然使其可行的训练，通常有某种正规化 或 规范化

好吧，假设我们已经确定了某种神经网络架构。现在的问题是如何获取数据来训练网络。围绕神经网络和一般机器学习的许多实际挑战都集中在获取或准备必要的训练数据上。在许多情况下（“监督学习”），人们想要得到输入和输出的明确示例。因此，例如，人们可能希望用图像中的内容或其他属性进行标记。也许一个人必须明确地通过——通常要付出很大的努力——并做标记。但通常情况下，人们可以利用一些已经做过的事情，或者用它作为某种代理。因此，例如，人们可以使用为网络上的图像提供的alt标签。或者，在另一个领域中，人们可以使用为视频创建的封闭字幕。或者，对于语言翻译培训，人们可以使用网页的并行版本或存在于不同语言中的其他文档。

你需要多少数据来显示一个神经网络来训练它来完成一个特定的任务？同样，我们很难从第一性原理来估计。当然，通过使用“迁移学习”来“迁移”，比如已经在其他网络中学习到的重要特征列表，可以显著降低这些需求。但一般来说，神经网络需要“看到很多例子”来训练好。至少对于某些任务来说，这是一个重要的神经网络知识，这些例子可以惊人地重复。事实上，这是一个标准的策略，反复展示一个神经网络的所有例子。在每一个“训练轮”（或“时代”）中，神经网络至少会处于略微不同的状态，并且以某种方式“提醒它”一个特定的例子有助于让它“记住那个例子”。（是的，也许这类似于人类记忆中重复的作用。）

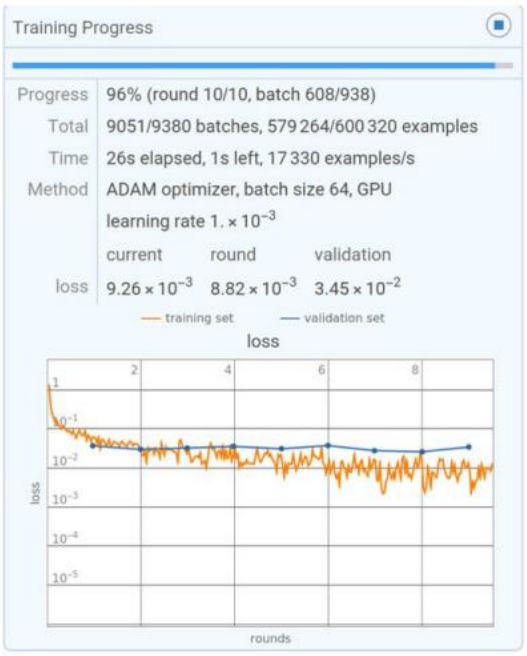
但经常只是一遍又一遍地重复同样的例子是不够的。也有必要显示这个例子的神经网络变化。这是神经网络知识的一个特征，而那些“数据增强”的变化却没有

必须要复杂才能有用。只要用基本的图像处理对图像进行轻微的修改，就可以使它们的神经网络训练基本上“和新的一样好”。同样地，当一个人耗尽了真正的视频，等等。为了训练自动驾驶汽车，你可以继续在一个模型视频游戏环境中运行模拟中获得数据，而不需要真实场景的所有细节。

像ChatGPT这样的东西怎么样？嗯，它有一个很好的特性，它可以做“无监督学习”，使它更容易得到它的例子来训练。回想一下，ChatGPT的基本任务是弄清楚如何继续它被给出的一段文本。所以，要得到它的“训练例子”，我们所要做的就是得到一段文本，并屏蔽它的末端，然后使用它作为“训练的输入”——“输出”是完整的、未屏蔽的文本。我们稍后将进一步讨论，但主要的一点是——不像，学习图像中的内容——不需要“明确的标记”；ChatGPT实际上可以直接从给出的任何文本例子中学习。

好吧，那么在神经网络中的实际学习过程如何呢？最后，这一切都是关于确定什么权重最能捕获已经给出的训练示例。还有各种各样的详细选择和“超参数设置”（之所以得名，是因为权重可以被认为是“参数”），可以用来调整如何这样做。有不同的选择的 损失 函数（平方和、绝对值和等）。有不同的方法来做损失最小化（在每一步的重量空间中移动多远，等等）。还有一些问题，比如一个“批”的例子可以得到一个试图最小化的损失的连续估计。是的，我们可以应用机器学习（就像我们在沃尔弗拉姆语言中所做的那样）来自动化机器学习，并自动设置诸如超参数之类的东西。

但最终，整个训练过程可以通过观察损失如何逐渐减少来描述(如在这个沃尔弗拉姆 语言 进度监视器 为了 a 小的 训练



人们通常会看到的是，损失会在一段时间内减少，但最终会达到某个恒定值。如果这个值足够小，那么培训就可以被认为是成功的；否则，这可能是一个人应该尝试改变网络架构的标志。

有人能知道“学习曲线”需要多长时间才能变平吗？就像许多其他事情一样，似乎有近似的幂律尺度[这种关系取决于神经网络的大小和人们所使用的数据量的关系。](#)但一般的结论是，训练一个神经网络是困难的，而且需要大量的计算努力。作为一个实际问题，绝大部分的工作都花在了数字数组上的操作上，这是gpu擅长的——这就是为什么神经网络训练通常受到gpu可用性的限制。

在未来，会有更好的方法来训练神经网络——或者通常做神经网络所做的事情？我想，几乎可以肯定。神经网络的基本思想是用大量简单的（本质上相同的）组件创建一个灵活的“计算结构”，并使这种“结构”可以逐步修改，从例子中学习。在当前的神经网络中，人们本质上是使用微积分应用于实数的思想来进行增量修改。

但越来越明显的是，拥有高精度的数字并不重要；即使使用现有的方法，8位或更少可能就足够了。

与计算 系统 像 细胞的 自动机基本上在并行操作的位从来不清楚如何向 做 这 种类 增量的 修改，但没有理由认为这是不可能的。事实上，很像“深度学习” 重大进展 的 在更复杂的情况下，这种增量修改可能会比在简单的情况下更容易。

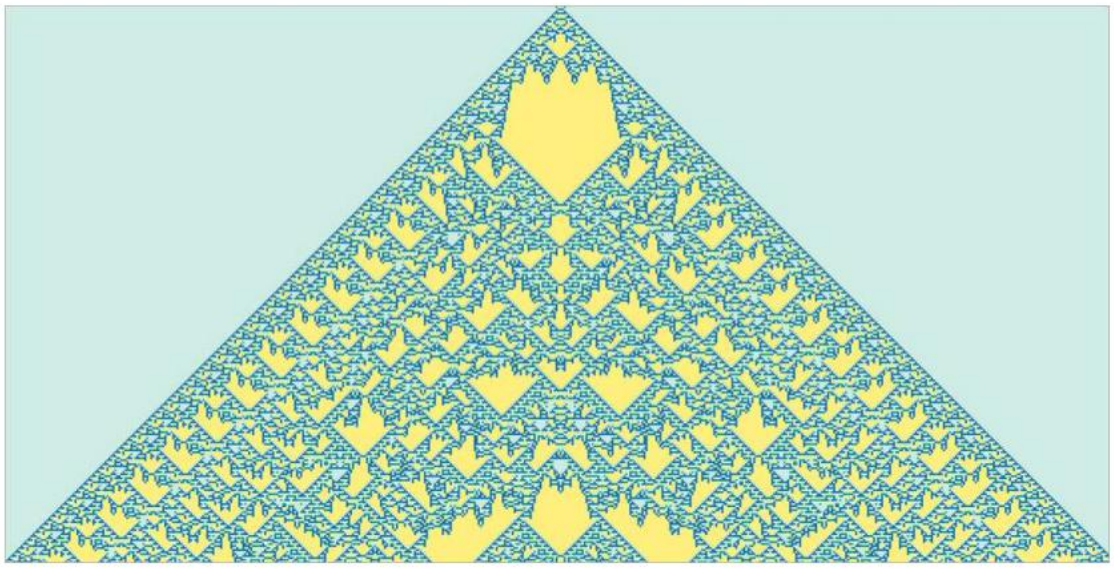
神经网络——也许有点像大脑——被设置成有一个本质上固定的神经元网络，被修改的是它们之间连接的强度（“重量”）。（也许至少在年轻的大脑中，大量的全新的联系也可以增长。）但是，虽然这可能对生物学来说是一个方便的设置，但目前还不清楚它是否接近于实现我们所需功能的最佳方式。还有一些相当于渐进式网络重写的东西(也许让人想起我们的物理学 项目)最终可能会更好。

但即使是在现有的神经网络框架内，目前也有一个关键的限制：现在进行的神经网络训练基本上是连续的，每批例子的效果被传播回以更新权重。事实上，对于目前的计算机硬件——甚至考虑到gpu——神经网络的大部分时间都是“空闲”的，一次只更新一部分。从某种意义上说，这是因为我们当前的计算机往往有与cpu（或gpu）分开的内存。但在大脑中，它可能是不同的——与每一个“记忆元素”有关。神经元)也是一个潜在的主动计算元素。如果我们能以这种方式建立我们未来的计算机硬件，我们就有可能更有效地进行培训。

“一个足够大的网络肯定能做任何事吧！”

像ChatGPT这样的东西的功能似乎令人印象深刻，以至于人们可能会想象，如果一个人能“继续前进”，训练越来越大的神经网络，那么他们最终就能够“做所有的事情”。如果一个人关心的是人类很容易直接思考的东西，那么很有可能就是这样。但过去几百年的科学经验的教训是，有些东西可以通过正式的过程来解决，但并不容易得到人类的直接思考。

非平凡数学就是一个很大的例子。但一般的情况是计算的。最终的问题是计算的不可约性的现象。有一些计算，人们可能会认为会采取很多步骤，但实际上可以“简化”为相当直接的事情。但是计算不可约性的发现意味着这并不总是有效的。相反，有一些过程——可能就像下面的过程——要解决不可避免地发生了什么，基本上需要追踪每个计算步骤：



我们通常用大脑做的事情大概是为了避免计算不可约性而专门选择的。在大脑里做数学需要特别的努力。在实践中，仅仅在大脑中“思考”任何重要程序的操作步骤基本上是不可能的。

当然，我们也有电脑。用计算机，我们可以很容易地做一些长时间的，计算上不可简化的事情。关键是，这些方法通常没有捷径。

是的，我们可以记住很多在某些特定的计算系统中发生的事情的具体例子。也许我们甚至可以看到一些（“计算可约的”）模式，让我们做一点概括。但关键是，计算的不可约性意味着我们永远不能保证意外不会发生——只有通过明确地进行计算，你才能知道在任何特定情况下实际发生了什么。

最后，在可学习性和计算不可约性之间只有一种基本的紧张关系。学习实际上涉及到压缩 数据由 促使…改变 规则性但是计算的不可约性意味着最终可能存在的规律是有限制的。

作为一个实际问题，人们可以想象将一些小的计算设备——如元胞自动机或图灵机——构建成可训练的系统，如神经网络。事实上，这样的设备可以作为神经网络的很好的“工具”——就像Wolfram|Alpha一样 能 是 a 好的 工具 为了 ChatGPT .但是计算的不可约性意味着，人们不能指望“进入”这些设备并让它们学习。

换句话说，在能力和可训练性之间有一个最终的权衡：你越希望一个系统“真正利用”它的计算能力，它就越能显示出计算的不可约性，它的可训练能力就越少。它从根本上可训练得越强，就越不能进行复杂的计算。

（对于目前的ChatGPT来说，情况实际上要极端得多，因为用于生成每个输出令牌的神经网络是一个纯粹的“前馈”网络，没有循环，因此没有能力用非平凡的“控制流”进行任何类型的计算。）

当然，人们可能会想，能够进行不可约的计算是否真的很重要。事实上，在人类历史的大部分时间里，这并不是特别重要。但是，我们的现代技术世界已经建立在工程学之上，这些工程学至少利用了数学计算——以及越来越多的更一般的计算。如果我们看看自然世界，它是充满了的 的 不可约的 计算——我们正在慢慢理解如何模拟和用于我们的技术目的。

是的，神经网络当然可以注意到自然世界中的规律，我们也很容易通过“独立的人类思维”注意到这些规律。但是，如果我们想计算出数学或计算科学范围内的东西，神经网络就无法做到——除非它有效地“作为工具”一个“普通的”计算系统。

但这一切可能会令人困惑。在过去，有很多任务——包括写论文——我们认为这些任务对计算机来说“从根本上说太难了”。现在我们看到像ChatGPT这样的人完成了它们，我们倾向于突然认为计算机一定变得更加强大——特别是超过了它们已经能够做的事情（比如逐步计算像元胞自动机这样的计算系统的行为）。

但这并不是一个正确的结论。计算上不可约的过程在计算上仍然是不可约的，而且对计算机来说仍然是根本困难的——即使计算机可以很容易地计算出它们的各个步骤。相反，我们应该得出结论的是，任务——比如写论文——我们人类可以做，但我们不认为计算机可以做，实际上在某种意义上，在计算上比我们想象的要容易。

换句话说，神经网络之所以能成功地写一篇文章，是因为写一篇文章是一个比我们想象的“计算上更浅”的问题。在某种意义上说，这让我们更接近于“拥有一个人”

关于我们人类如何设法写论文，或者一般处理语言。

如果你有一个足够大的神经网络，那么，是的，你可能可以做任何人类能轻易做的事情。但你不会捕捉到自然世界一般能做什么——也不会捕捉到我们从自然世界中创造出来的工具能做什么。正是这些工具的使用——包括实用的——概念性的——在近几个世纪里让我们超越了“纯粹的人类思维”的边界，并为人类的目的捕捉物理和计算宇宙中的东西。

嵌入的概念

神经网络——至少目前的设置——从根本上是基于数字的。所以如果我们要用它们来处理类似文本的东西，我们就需要一种方法来表示我们的文本和算术当然，我们可以开始为字典中的每个单词分配一个数字（本质上就像ChatGPT一样）。但有一个重要的想法——例如chatgpt的核心——它超出了这一点。这就是“嵌入”的概念。人们可以把嵌入看作是一种试图用一组数字来表示某物的“本质”的方法——“附近的东西”是由附近的数字表示。

例如，我们可以把一个词的嵌入看作是试图躺下言出必行在a种类的意思在空间中，“在附近”的单词出现在附近。实际使用的嵌入——比如在chatgpt中——往往涉及大量的数字列表。但是如果我们投射到二维，我们可以展示如何通过嵌入排列单词的例子：

是的，我们所看到的东西在捕捉典型的日常印象方面表现得非常好。但是我们如何才能构建这样的嵌入呢？大致来说，这个想法是查看大量的文本（这里是来自网络的50亿个单词），然后看看不同单词出现的“环境有多相似”。例如，“短吻鳄”和“鳄鱼”在其他相似的句子中几乎可以互换出现，这意味着它们将被放置在嵌入物的附近。但是“萝卜”和“鹰”不会出现在其他类似的句子中，所以它们在嵌入过程中会相距很远。

但是如何使用神经网络来实现这样的东西呢？让我们开始讨论嵌入的内容，而不是为了文字，而是为了图像。我们想找到某种方法来用数字列表来描述图像，这样“我们认为相似的图像”就会被分配给相似的数字列表。

我们如何判断我们是否应该“考虑图像相似”？好吧，如果我们的图像是手写的数字，我们可以“考虑两个相似的图像”，如果它们是相同的数字。在我们之前，我们讨论了一种经过训练来识别手写数字的神经网络。我们可以把这个神经网络看作是建立起来的，这样在它最终的输出中，它就会把图像放到10个不同的箱子里，每个数字一个。

但是如果我们在做出“这是一个“4”决定之前“拦截”神经网内发生的事情呢？我们可能会认为，在神经网络内部，有一些数字将图像描述为“大多是4-样的，但有点像2样的”或一些类似的图像。这个想法是选择这些数字作为嵌入中的元素。

这是一个概念。而不是直接试图描述“图像是其他图像”，我们相反考虑一个定义良好的任务（在这种情况下数字识别），我们可以得到明确的训练数据然后使用在做这个任务神经网络隐含地做出“接近决策”。所以，我们不再明确地谈论“图像的接近性”，而是谈论图像代表什么数字的具体问题，然后我们“把它留给神经网络”，含蓄地确定“图像的接近性”意味着什么。

那么，这是如何为数字识别网络更详细地工作的呢？我们可以把这个网络看作是由11个连续的层组成的，我们可以像这样总结一下（激活函数显示为单独的层）：

一开始，我们输入第一层实际图像，由像素值的二维数组表示。最后，从最后一层，我们得到一个10个值的数组，我们可以说网络“多确定”图像对应于每个数字0到9。

图像中的输入和最后一层神经元的值为：

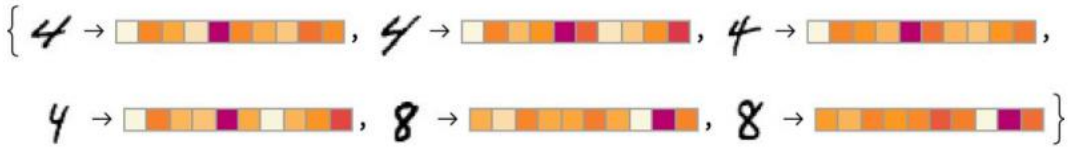
$$\{1.42071\times10^{-22}, 7.69857\times10^{-14}, 1.9653\times10^{-15}, 5.55229\times10^{-21}, 1.,$$
$$8.33841\times10^{-14}, 6.89742\times10^{-17}, 6.52282\times10^{-19}, 6.51465\times10^{-12}, 1.97509\times10^{-14}\}$$

换句话说，神经网络此时已经“令人难以置信地确定”了这幅图像是一个4——为了真正得到输出的“4”，我们只需要选出具有最大值的神经元的位置。

但如果我们提前看一步呢？网络中的最后一个操作是一个所谓的软性max，它试图“强制确定性”。但在此之前，神经元的值是：

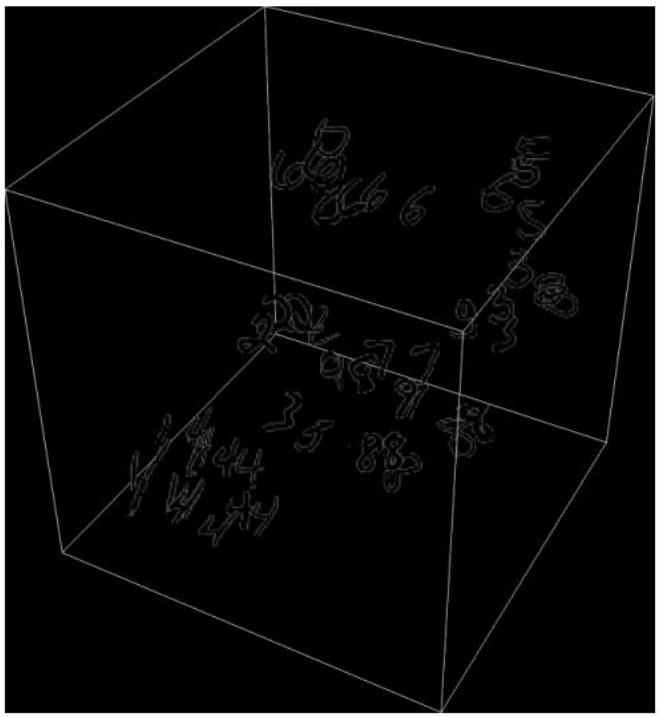
$$\{-26.134, -6.02347, -11.994, -22.4684,$$
$$24.1717, -5.94363, -13.0411, -17.7021, -1.58528, -7.38389\}$$

代表“4”的神经元仍然具有最高的数值。但其他神经元的值也有信息。我们可以期待，这个数字列表在某种意义上可以用来描述图像的“本质”，从而提供一些我们可以使用作为嵌入的东西。例如，这里的4都有一个稍微不同的“签名”（或“特征嵌入”）——都与8非常不同：



这里我们本质上是用10个数字来描述我们的图像。但通常最好使用更多。例如，在我们的数字识别网络中，我们可以通过进入前一层得到一个包含500个数字的数组。这可能是一个用作“图像嵌入”的合理数组。

如果我们想对手写数字的“图像空间”进行显式的可视化，我们需要“降维”，通过有效地投影500维的向量，比如说，三维空间：



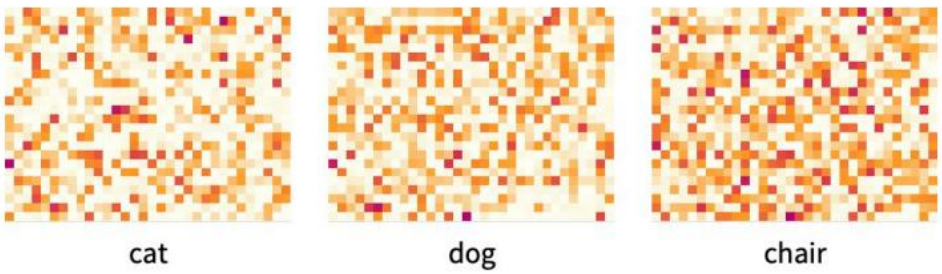
我们刚刚讨论了为图像创建一个表征（从而嵌入），通过确定（根据我们的训练集）它们是否对应于相同的手写数字来有效地识别图像的相似性。如果我们有一个训练集，可以识别5000种常见的物体（猫，狗，椅子，……）中的哪一种，我们就可以对图像做同样的事情。通过这种方式，我们可以通过对公共物体的识别来“锚定”一个图像嵌入，然后根据神经网络的行为来“概括”。关键是，只要这种行为与我们人类感知和解释图像的方式相一致，这将最终成为一种“在我们看来是正确的”的嵌入，并且在实践中对做“类人判断”的任务很有用。

好吧，那么我们如何遵循同样的方法来找到单词的嵌入物呢？关键是要从一项关于我们可以随时进行训练的单词的任务开始。而这样的标准任务是“单词预测”。想象一下，我们得到了“___猫”。基于大量的文本语料库（比如说，网络上的文本内容），不同的单词可能“填写空白”的概率是多少？或者，或者，给定“___black___”，不同的“侧翼单词”的概率是多少？

我们该如何为一个神经网络设置这个问题呢？最终，我们必须用数字来表述一切事物。其中一种方法就是给英语中大约5万个常见单词中的每个单词分配一个唯一的数字。例如，“the”可能是914，而“cat”（前面有一个空格）可能是3542。（这些是GPT-2实际使用的数字。）所以对于“___cat”问题，我们的输入可能是{914,3542}。输出应该是什么样子的？嗯，它应该是一个5万个左右数字的数字列表，有效地给出每个可能的“填充”单词的概率。再一次，为了找到一个嵌入，我们想在神经网络“得出结论”之前“拦截”它的“内部”——然后选择出现在那里的数字列表，我们可以认为这是“描述每个词”。

那么，这些特征是什么样子的呢？在过去的10年里，已经开发了一系列不同的系统（word2vec，GloVe，BERT，GPT，……），每一个都基于不同的神经网络方法。但最终，他们都用文字，用成百上千的数字来描述它们。

在它们的原始形式中，这些“嵌入向量”是相当没有信息的。例如，以下是GPT-2产生的作为三个特定单词的原始嵌入向量：



如果我们测量这些向量之间的距离，那么我们就可以找到单词“接近”的东西。稍后我们将更详细地讨论我们可能认为这种嵌入的“认知”意义。但现在的主要观点是，我们有一种方法，可以有效地将单词转化为“神经网络友好”的数字集合。

但实际上，我们可以更进一步，而不仅仅是用数字的集合来描述单词；我们也可以对单词序列，或者整个文本块这样做。在ChatGPT里面就是它处理事情的方式。它获取到目前为止的文本，并生成一个嵌入向量来表示它。然后，它的目标是找到接下来可能出现的不同单词的概率。它的答案是一个数字的列表，基本上给出了大约50,000个可能的单词中的每个单词的概率。

(严格地说，ChatGPT并不处理单词，而是处理单词和“标记”——方便的语言单位，可能是整个单词，也可能只是像“pre”、“ing”或“izen”这样的片段。使用代币可以使ChatGPT更容易处理稀有、复合和非英语单词，有时，无论好坏，也更容易发明新单词。)

ChatGPT内部

好了，我们终于准备好讨论ChatGPT内部的内容了。是的，最终，它是一个巨大的神经网络——目前是所谓的GPT-3网络的一个版本，有1750亿个权重。在很多方面，这是一个神经网络，与我们讨论过的其他神经网络非常相似。但这是一个专门用于处理语言的神经网络。它最显著的特点是一个叫做“变压器”的神经网络架构。

在我们上面讨论的第一个神经网络中，任何给定层上的每个神经元基本上都与之前这一层上的每个神经元连接（至少有一定的权重）。但是，如果一个人使用的是具有特定的、已知结构的数据，那么这种完全连接的网络（可能）是多余的。因此，例如，在处理图像的早期阶段，通常使用所谓的卷积 神经的 网络（“对流网络”），其中神经元被有效地放置在一个类似于图像中的像素的网格上，并且只连接到网格上附近的神经元。

变形金刚的想法是对组成一段文本的令牌序列做至少一些类似的事情。但是，变形金刚并没有仅仅定义序列中的固定区域，而是引入了“注意力”的概念——以及“关注”序列的某些部分而不是其他部分的概念。也许有一天，开始一个通用的神经网络，通过训练进行所有的定制是有意义的。但至少到目前为止，在实践中“模块化”事物似乎至关重要——就像变形金刚一样，也可能也像我们的大脑一样。

好吧，那么ChatGPT（或者，更确切地说，它所基于的GPT-3网络）到底在做什么呢？回想一下，它的总体目标是以一种“合理”的方式继续使用文本，基于它所接受的训练（包括从网络上查看数十亿页的文本等）。所以在任何给定的点上，它都有一定数量的文本——它的目标是为下一个标记的添加提供一个适当的选择。

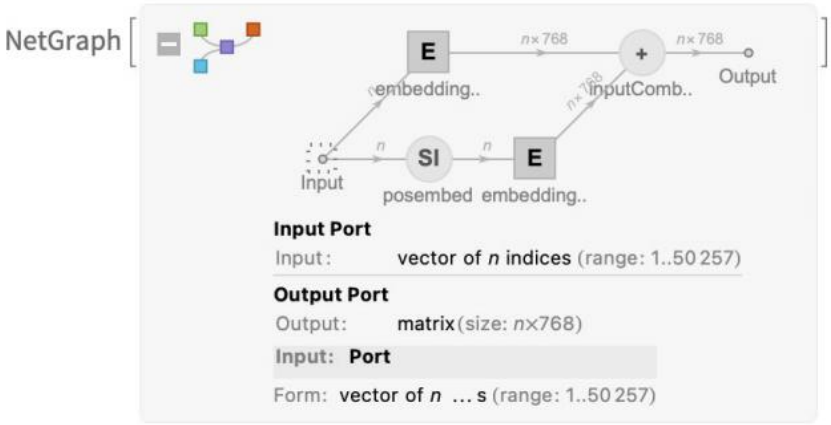
它在三个基本阶段中运作。首先，它取与迄今为止的文本对应的标记序列，并找到一个嵌入(即。表示这些数字的数字数组)。然后它在这个嵌入-在a中进行操作

“标准的神经网络方式”，值在网络中“涟漪”连续层，以产生一个新的嵌入(i。e.一个新的数字数组)。然后，它取这个数组的最后一部分，并从中生成一个大约50,000个值的数组，这些值转化为不同可能的下一个令牌的概率。（是的，碰巧的是，标记的数量与英语中常用单词相同，尽管只有大约3000个标记是整个单词，其余的是片段。）

一个关键的点是，这个管道的每个部分都是由一个神经网络实现的，其权值由网络的端到端训练决定。换句话说，实际上除了整体架构之外，没有什么都是“明确设计的”；一切都只是从训练数据中“学习”。

然而，关于该架构的设置方式，也有很多细节，反映了各种经验和神经网络知识。而且——尽管这肯定是进入杂草——我认为讨论其中一些细节是有用的，尤其是了解构建类似的东西需要什么ChatGPT。

首先是嵌入模块。下面是GPT-2的Wolfram语言表示示意图：

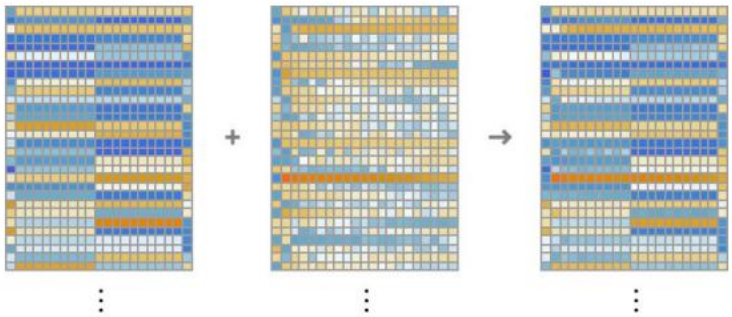


输入是一个向量 的 *n* 标记（如前一节中用1到大约50000的整数表示）。每个令牌都被单层转换 神经的 转换到一个嵌入向量（GPT-2的长度为768，ChatGPT的GPT-3的长度为12288）。与此同时，有一个“次要路径”来接受这个序列 的 整数 标记的位置，并从这些整数中创建另一个嵌入向量。 最后是嵌入向量

从令牌值和令牌位置中添加 一起-从嵌入模块中生成嵌入向量的最终序列。

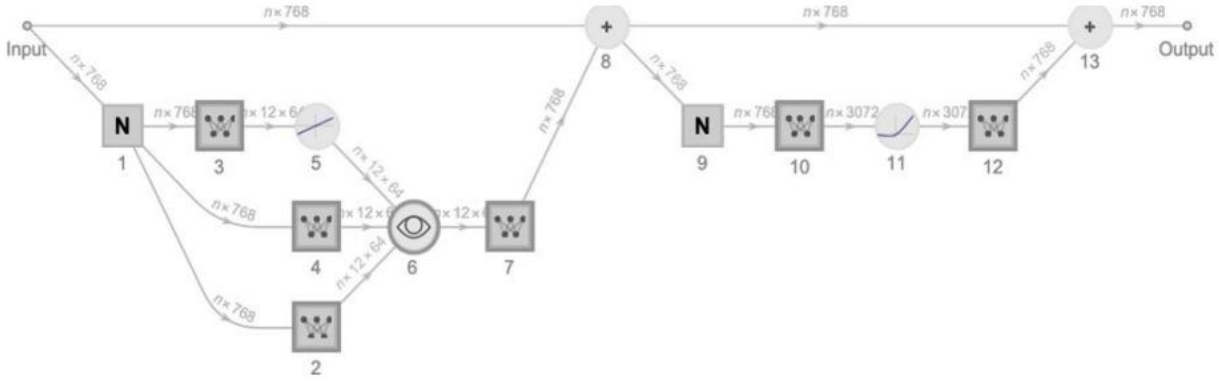
为什么要只把标记值和标记位置嵌入向量加在一起呢？我不认为这方面有什么特别的科学依据。只是人们尝试过各种不同的事情，而这似乎是有效的。这是神经网络的知识的一部分，在某种意义上，只要设置是“大致正确”通常可能在细节通过做足够的训练，而不需要“理解在工程水平”相当神经网络最终如何配置本身。

这是嵌入模块所做的，操作在字符串上你好你好你好你好你好你好你好你好你好你好你好你好你好你好你好你好你好你好再见：



每个标记的嵌入向量的元素显示在页面的下方，在整个页面中，我们首先看到了一连串的“hello”嵌入，然后是一连串的“再见”嵌入。上面的第二个数组是位置嵌入——它看起来有点随机的结构就是“碰巧被学到的东西”（在GPT-2中）。

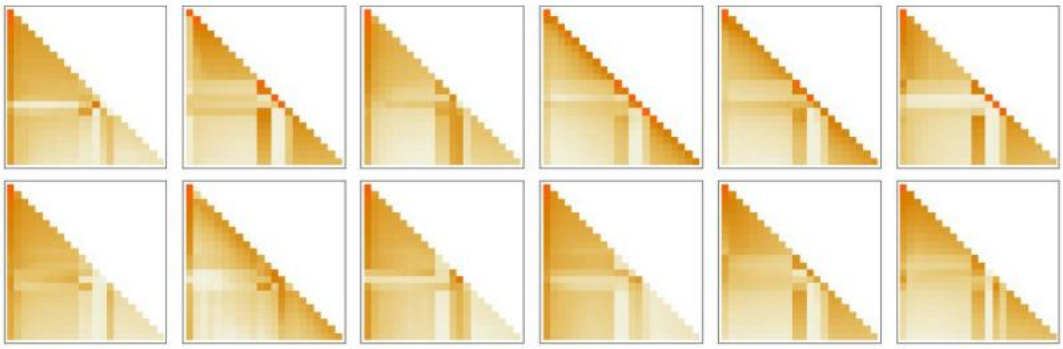
好的，所以在嵌入模块之后，出现了变压器的“主要事件”：一系列所谓的“注意块”（12为GPT-2,96为ChatGPT的GPT-3）。这一切都相当复杂——而且让人想起了典型的、难以理解的大型工程系统，或者，就此而言，是生物系统。但无论如何，这里是一个单一的“注意力块”（为GPT-2）的示意图：



在每个这样的注意块中都有一个“注意头”集合（GPT-212，ChatGPT的GPT-396）——每个注意头都在嵌入向量的不同值块上独立运行。（而且，是的，我们不知道有什么特别的原因，为什么分割嵌入向量是一个好主意，也不知道它的不同部分“意思”；这只是那些被“发现有效”的事情之一。）

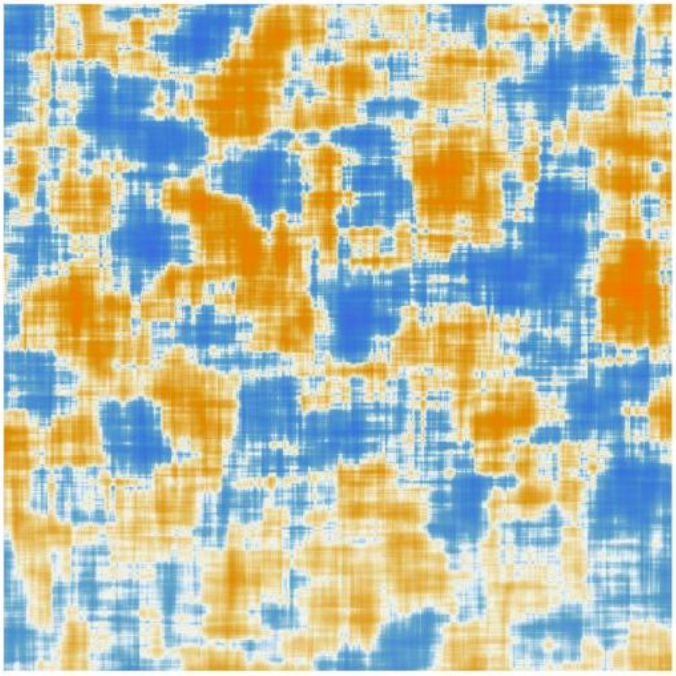
好吧，那么注意力头会做什么呢？基本上，它们是一种“回顾”标记序列的方式。e.)，并以一种对寻找下一个令牌很有用的形式“包装过去”。在 第一 部分 上面我们讨论了使用2克概率来根据它们的前身来选择单词。变形金刚中的“注意”机制是允许“注意”更早的单词——从而潜在地捕捉到，比如，动词可以指在一个句子中出现在许多单词面前的名词。

在更详细的层面上，一个注意力头所做的是重组与不同标记相关联的嵌入向量中的块，并具有一定的权重。因此，例如，第一个注意块（GPT-2）中的12个注意头具有以下（“hello，再见”字符串的“回顾到起点”）的“重组权重”模式：



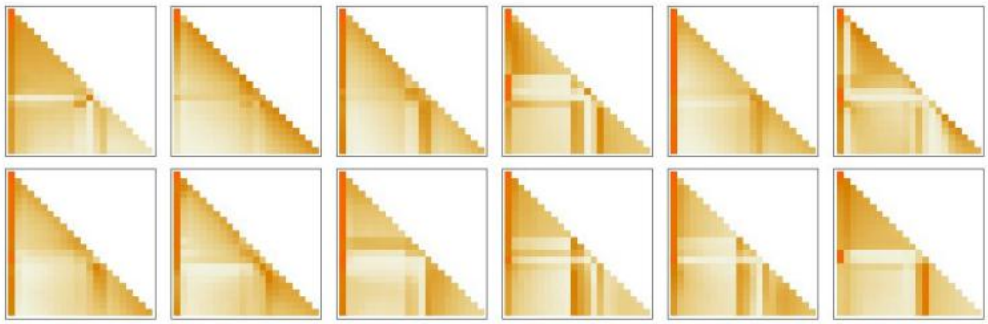
在被注意头处理后，得到的“重新加权嵌入向量”（GPT-2长度为768，ChatGPT的GPT-3长度为12288）完全通过标准“” 有关的 神经的 网 层很难掌握这一层在做什么。但这里是它所使用的768×768权重矩阵的曲线图（这里是GPT-2）：

取64×64移动平均线，一些（随机行走的）结构开始出现：

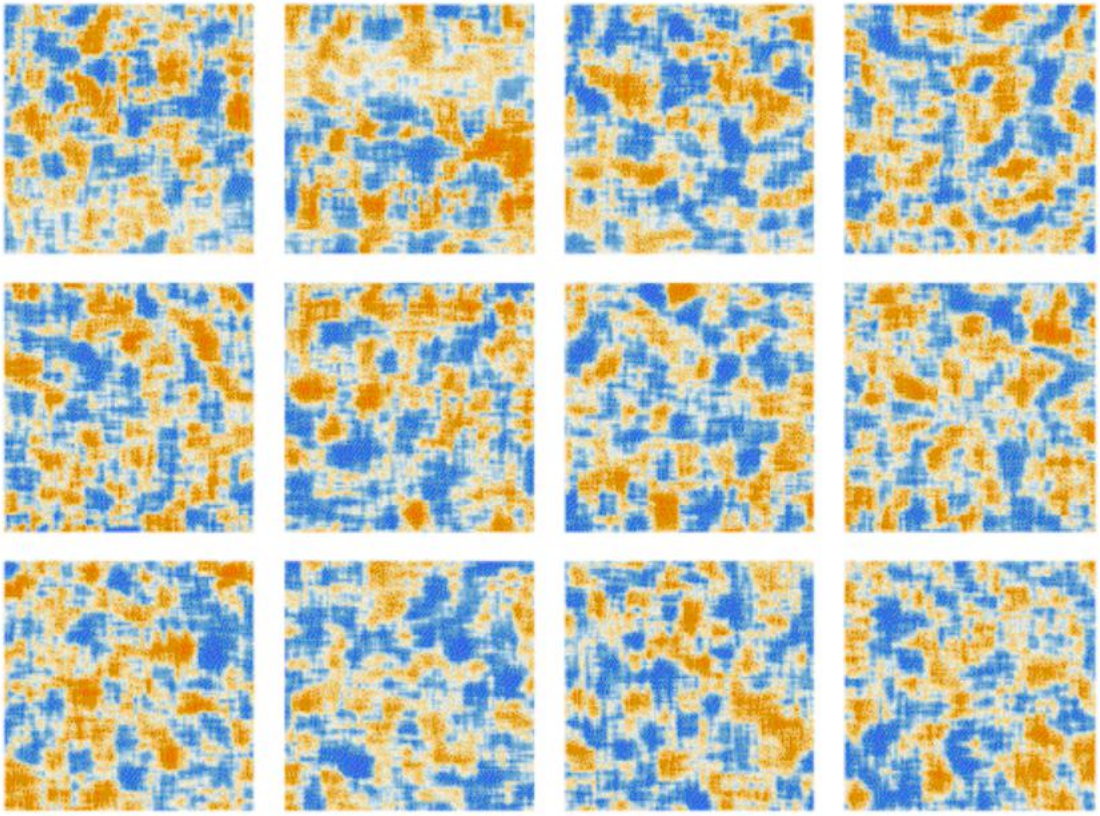


是什么决定了这个结构？最终，它可能是对人类语言特征的一些“神经网络编码”。但到目前为止，这些特性可能是什么还很不清楚。实际上，我们正在“打开ChatGPT的大脑”（或者至少是GPT-2），并发现，是的，它在那里是复杂的，而且我们不理解它——尽管它最终产生了可识别的人类语言。

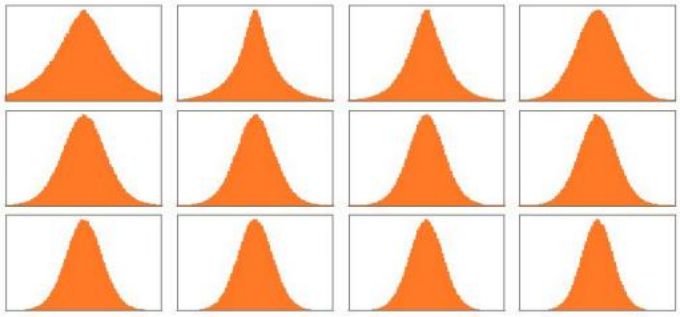
好的，在经历了一个注意块之后，我们得到了一个新的嵌入向量——然后它依次通过额外的注意块（GPT-2总共12个；GPT-3总共96个）。每个注意力块都有自己独特的“注意”和“完全连接”的权重模式。这里是GPT2的第一个注意头的“你好，再见”输入的注意权重顺序：



这里是（移动平均）完全连接层的“矩阵”：



奇怪的是，尽管这些在不同注意块中的“权值矩阵”看起来非常相似，但权值大小的分布可能有些不同（而且并不总是高斯分布）：



那么，在经历了所有这些注意力障碍之后，变压器的净效果是什么呢？本质上，它是将令牌序列的原始嵌入集合转换为最终集合。ChatGPT的特殊工作方式是在这个集合中获取最后一个嵌入，

并“解码”它，以产生一个关于接下来应该出现什么标记的概率列表。

这就是ChatGPT内部的大纲内容。它可能看起来很复杂（尤其是因为它有许多不可避免地有些任意的“工程选择”），但实际上所涉及的最终元素是非常简单的。因为最后我们要处理的只是一个由“人工神经元”组成的神经网络，每个神经元都做一个简单的操作，取一组数值输入，然后将它们与一定的权重相结合。

原始输入ChatGPT是一个数组数字（嵌入向量标记迄今为止），当ChatGPT“运行”产生一个新的令牌只是这些数字“涟漪”神经网络的层，每个神经元“做它的事情”，将结果传递给下一层神经元。没有循环或“返回”。一切都只是通过网络“前馈”。

这与一个典型的计算系统非常不同——比如图灵机——在图灵机中，结果被相同的计算元素重复地“重新处理”。在这里，至少在生成一个给定的输出标记时，每个计算元素(i。e.神经元)只使用一次。

但在某种意义上，仍然有一个“外部循环”，即使在ChatGPT中也会重用计算元素。因为当ChatGPT将生成一个新的令牌时，它总是“读取”(i。e.将出现在它之前的整个令牌序列作为输入)，包括ChatGPT本身之前“写”过的令牌。我们可以认为这个设置意味着ChatGPT——至少在其最外层——涉及一个“反馈循环”，尽管在这个循环中，每个迭代都是出现在它生成的文本中的标记。

但让我们回到ChatGPT的核心上来：一个被重复用于生成每个令牌的神经网络。在某种程度上，它非常简单：一组相同的人工神经元。而这个网络的一些部分只是由(“完全”组成“连接”的神经元层，其中给定一层上的每个神经元都（有一定的权重）连接到之前的那一层上的每个神经元。但特别是变压器结构，ChatGPT有更多的结构，其中只有不同层的特定神经元

有关的（当然，人们仍然可以说“所有的神经元都是相连的”——但有些神经的重量为零。）

此外，ChatGPT中的神经网络的一些方面并不是最自然地被认为是由“同质”层组成的。例如，正如上面的标志性摘要所指出的，在一个注意力块中，有一些地方对传入的数据进行“多个副本”，每个副本然后经过不同的“处理路径”，可能涉及不同数量的层，然后才进行重组。但是，虽然这可能是一个方便的表示正在发生的事情，但至少在原则上总是可以想到“密集填充”层，但只是有一些权重为零。

如果查看通过ChatGPT的最长路径，大约涉及400个（核心）层——在某些方面数量并不多。但现在有数百万个神经元——总共有1750亿个连接，因此有1750亿个权重。而需要意识到的一件事是，每次ChatGPT生成一个新的令牌时，它都必须做一个涉及到每个这些权重的计算。在实现上，这些计算可以在某种程度上“逐层”组织成高度并行的阵列操作，这可以方便地在gpu上完成。但是对于每个生成的令牌，仍然需要进行1750亿次计算（最终还会更多）——所以，是的，用ChatGPT需要一段时间来生成一个长文本也就不足为奇了。

但最终，值得注意的是，所有这些操作——尽管它们非常简单——可以以某种方式一起完成如此好的文本生成工作。必须再次强调的是（至少据我们所知）没有“终极理论理由”，为什么这样的事情应该奏效。事实上，正如我们将要讨论的，我认为我们必须把这视为一个潜在的令人惊讶的科学发现：在像ChatGPT这样的神经网络中，我们有可能捕捉到人类大脑在生成语言时所做的事情的本质。

ChatGPT的培训

好了，我们现在给出了ChatGPT建立后如何工作的大纲。但是它是怎么建立起来的呢？其神经网络中的所有1750亿个权重是如何确定的？基本上，它们是非常大规模的训练的结果，基于由人类编写的大量的网络、书籍等文本。正如我们所说的，即使考虑到所有的训练数据，神经网络是否能够成功地生成“类人”的文本也并不明显。而且，再一次，似乎需要一些详细的工程片段来实现这一点。但ChatGPT的最大的惊喜和发现是，它是完全有可能的。实际上，一个“只有”1750亿个权重的神经网络可以成为人类编写的文本的“合理模型”。

在现代，有很多由人类以数字形式编写的文本。这个公共网络至少有几十亿个人写的页面，总共可能有一万亿字的文本。如果一个包括非公共网页，这个数字可能至少要大100倍。到目前为止，已经有超过500万本数字化图书出版（在已出版的1亿本左右），又提供了1000亿个左右的文字。这甚至没有提到视频中的文本。(作为一个个人的比较，我的 总数 一生 产量 已出版的 材料在过去的30年里还不到300万字 年 已经 写了大约1500万个单词的电子邮件，总共输入了大约5000万个单词——就在过去的几年里，我已经直播了1000多万个单词。是的，我会从中训练一个机器人。)

但是，好吧，考虑到所有这些数据，我们该如何从中训练一个神经网络呢？基本过程正如我们在上面的简单例子中讨论的那样。您给出一批示例，然后调整网络中的权重，以最小化网络对这些示例所产生的错误（“损失”）。从错误中进行“反向传播”的主要代价是，每次你这样做时，网络中的每一个权重通常都会发生至少一点点的变化，而且只有很多权重需要处理

和（实际的“反向计算”通常只是一个很小的常数因素，比正向的更难。）

使用现代GPU硬件，可以直接并行地从数千个示例中计算结果。但是当涉及到实际更新神经网络中的权重时，当前的方法需要人们基本上是逐批来做的。（而且，是的，这可能是实际的大脑——它们结合了计算和记忆元素——目前至少具有建筑结构上的优势。）

即使在我们前面讨论过的学习数值函数的看似简单的情况下，我们也发现我们经常需要使用数百万个例子来成功地训练一个网络，至少从头开始。那么，这意味着我们需要多少个例子来训练一个“类人语言”模型呢？似乎没有任何基本的“理论”方法来知道。但在实践中，ChatGPT成功地接受了几百亿字的文本训练。

有些文本被输入了好几次，有些只有一次。但不知何故，它从它所看到的文本中“得到了它所需要的东西”。但是考虑到这卷需要学习的文本，一个网络需要多大才能“学习好”呢？同样，我们还没有一个基本的理论方法来说。最终——正如我们将在下面进一步讨论的——人类语言可能会有一种“总算法内容”，以及人类通常所说的话。但下一个问题是，神经网络在实现基于该算法内容的模型时的效率如何。但我们也不知道——尽管ChatGPT的成功表明它是相当有效的。

最后，我们可以注意到ChatGPT使用了几百亿个权重来做什么——在数量上与它给出的训练数据的单词（或标记）总数相当。在某些方面，这可能会令人惊讶（尽管在ChatGPT的较小类似物中也有经验观察到），似乎工作良好的“网络的规模”与“训练数据的规模”如此相似。毕竟，这肯定不是“在ChatGPT内部”，所有来自网络和书籍的文本等等

是“直接存储”。因为ChatGPT内部实际上是一堆数字——精度略低于10位数字——它们是对所有文本的聚合结构的某种分布式编码。

换句话说，我们可能会问，人类语言的“有效信息内容”是什么，以及它通常说的是什么。这里有一些语言例子的原始语料库。然后是ChatGPT的神经网络中的表示。这种表示很可能远远不是“算法最小”表示（我们将在下面讨论）。但它是一种很容易被神经网络使用的表示形式。在这种表示中，最终训练数据的“压缩”似乎很少；平均而言，基本上只需要不到一个神经净重来携带一个训练数据的“信息内容”。

当我们运行ChatGPT来生成文本时，我们基本上必须使用每个权重一次。所以如果有n个权重，我们就有n个计算步骤——尽管在实践中许多步骤通常可以在gpu中并行完成。但是，如果我们需要大约n个单词的训练数据来设置这些权重，那么从我们上面所说的内容中，我们可以得出结论，我们将需要大约n个单词^{2}进行网络训练的计算步骤——这就是为什么，用目前的方法，人们最终需要谈论数十亿美元的培训努力。

基础培训之外

培训ChatGPT的大部分努力都花在了“展示”大量来自网络、书籍等的现有文本上。但事实证明，还有另一个显然相当重要的部分。

一旦它完成了从原始文本语料库中进行的“原始训练”，ChatGPT内部的神经网络就准备好开始生成它自己的文本，继续从提示开始，等等。但是，尽管这样做的结果往往看起来是合理的，但它们往往——尤其是较长的文本——以不非人类的方式“走开”。这不是人们很容易发现的东西，比如，通过对文本进行传统的统计。但这是真正阅读文本的人很容易注意到的东西。

还有一把钥匙 想法 在 那 建造 的 ChatGPT在“被动阅读”像网络这样的东西之后又有了一步：让真正的人类主动与ChatGPT互动，看看它产生了什么，并实际上给它关于“如何成为一个好的聊天机器人”的反馈。但是神经网络是如何利用这种反馈呢？第一步只是让人类评估来自神经网络的结果。但随后又建立了另一个神经网络模型，试图预测这些评级。但现在这个预测模型可以在原始网络上运行——本质上就像一个损失函数——，实际上允许该网络被给出的人类反馈“调谐”。而在实践中的结果似乎对系统产生“类人”输出的成功有很大的影响。

总的来说，有趣的是，“戳”这个“最初训练过”的网络似乎几乎不需要让它有效地走向特定的方向。有人可能会认为，要让网络表现得好像它“学到了一些新的东西”，你就必须去运行一个训练算法，调整权重，等等。

但事实并非如此。相反，基本上一次告诉ChatGPT某件事似乎就足够了——作为你所提供的提示的一部分——然后它就可以成功地利用你在生成文本时告诉它的内容。再一次，我认为，这是一个重要的线索

理解ChatGPT“真正在做”什么，以及它如何与人类语言和思维的结构相关联。

当然有一些相当人性化的东西：至少一旦它经过了所有的预训练，你可以告诉它一次，它可以“记住它”——至少“足够长的时间”来使用它生成一段文本。那么在像这样的案子里发生了什么呢？它可能是“你可能告诉它的一切都已经在某个地方了”——而你只是在把它带到正确的地方。但这似乎不可信。相反，更有可能的是，是的，这些元素已经在那里了，但细节是由“这些元素之间的轨迹”定义的，这就是你告诉它一些事情时引入的。

事实上，就像人类一样，如果你告诉它一些奇怪的和意想不到的东西，但完全不符合它所知道的框架，它似乎无法成功地“整合”这些。只有当它基本上以一种相当简单的方式运行在它现有的框架之上时，它才能“集成”它。

值得再次指出的是，神经网络的“获取”不可避免地存在“算法限制”。告诉它“这就是那个”形式的“浅层”规则，等等，神经网络就很可能能够很好地表示和再现这些——事实上，它从语言中“已经知道”的东西会给它一个立即遵循的模式。但是试着给它一个实际的“深度”计算的规则，这涉及到许多潜在的计算不可约的步骤，它只是不会工作。（请记住，在每一步中，它总是在网络中“转发数据”，从不循环，除非生成新的令牌。）

当然，该网络可以学习特定的“不可约”计算的答案。但是，一旦有了可能性的组合数量，这种“表查找风格”的方法就无法奏效。所以，是的，就像人类一样，现在是时候让神经网络“接触”并使用实际的计算工具了。(是的，还有沃尔fam|Alpha和沃尔fram 语言是唯一的这很合适，因为它们是用来“谈论世界上的事情”的，就像语言模型的神经网络一样。）

到底是什么让ChatGPT工作？

人类的语言——以及产生它所涉及的思维过程——似乎总是代表着一种复杂性的顶峰。事实上，人类的大脑——其中“只有”大约1000亿个神经元网络（可能还有100万亿个连接），这似乎有些值得注意。也许，人们可能会想象到，大脑中还有一些比他们的神经元网络更多的东西——比如一些新的未被发现的物理学层。但现在有了ChatGPT，我们得到了一个重要的新信息：我们知道，一个纯粹的人工神经网络具有神经元的连接，能够在生成人类语言方面做得非常好。

是的，这仍然是一个庞大而复杂的系统——它的神经净权值大约和目前世界上现有的文本单词一样多。但在某种程度上，我们似乎仍然很难相信，所有丰富的语言和它所能谈论的东西都可以被封装在这样一个有限的系统中。正在发生的部分事情无疑是普遍现象的反映(这首先在这个例子中变得明显 的 规则 30)认为，计算过程实际上可以极大地放大系统的明显复杂性，即使它们的底层规则是简单的。但是，实际上，正如我们上面所讨论的，ChatGPT中使用的神经网络往往是专门构造来限制这种现象的影响——以及与之相关的计算不可约性——以便使它们的训练更容易获得。

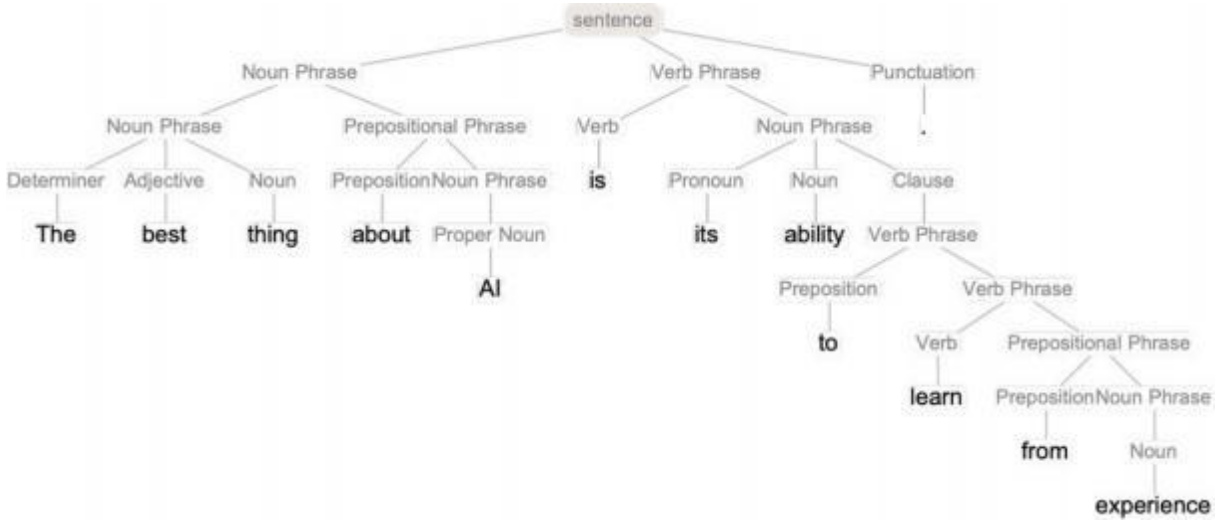
那么，像ChatGPT这样的东西是如何在语言中发挥作用呢？我认为，最基本的答案是，语言在一个最基本的层面上比它看起来更简单。这意味着chatgpt——即使它最终具有直接的神经网络结构——也能够成功地“捕捉到人类语言的本质”及其背后的思维。此外，在其训练中，ChatGPT以某种方式“含蓄地发现”了语言（和思维）中使这成为可能的任何规律。

我认为，ChatGPT的成功为我们提供了一个基本和重要的科学领域的证据：它表明我们可以期待它会出现

主要的新的“语言法则”——以及有效的“思想法则”——有待发现。在chatgpt这个神经网络中，这些法则充其量是隐含的。但如果我们能以某种方式使法律明确化，我们就有可能以ChatGPT更直接、更高效、更透明的方式来做类似的事情。

但是，好吧，这些法律是什么样子的？最终，他们必须给我们某种处方，让语言以及我们所说的东西如何组合在一起。稍后我们将讨论如何“查看ChatGPT内部”能够给我们一些方面的提示，以及我们从构建计算语言中所知道的如何提示前进的道路。但是首先让我们讨论两个广为人知的例子，说明什么是“语言定律”，以及它们如何与ChatGPT的运作有关。

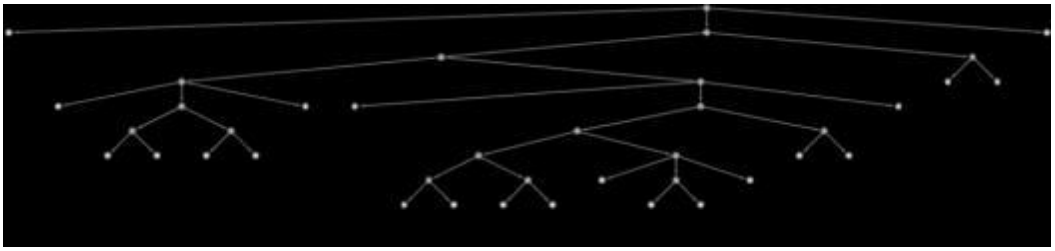
第一个是语言的语法。语言不仅仅是一堆随意混杂的单词。相反，有（相当地）明确的语法 关于不同种类的单词如何组合在一起的规则：例如，在英语中，名词的前面可以有形容词和动词，但通常两个名词不能完全相邻。这种语法结构可以（至少近似地）被一组规则所捕获，这些规则定义了如何进行“解析” 树 能 是 放置 一起



ChatGPT对这些规则没有任何明确的“知识”。但不知何故，在训练中，它含蓄地“发现”了它们——然后似乎很善于跟踪它们。那么这是如何工作的呢？在“大局”的层面上，它不是

易懂的但要深入了解，看一个更简单的例子可能会有指导意义。

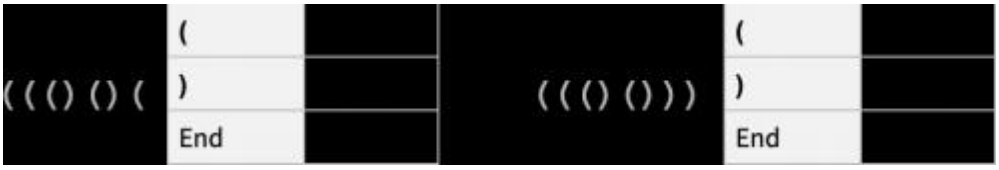
考虑一种由（s和）序列组成的“语言”，语法为 指定圆括号应该始终保持平衡，并由解析树表示，如：



我们能训练一个神经网络来产生“语法正确”的括号序列吗？在神经网络中有各种方法来处理序列，但是让我们使用变压器网络，就像ChatGPT所做的那样。给定一个简单的变压器网，我们可以开始给它提供语法上正确的括号序列作为训练示例。一个微妙之处（实际上也出现在ChatGPT的人类语言中）是，除了我们的“内容令牌”（这里是（和“））），我们还必须包含一个“End”令牌，生成的令牌表示输出不应该继续(i。e.对于ChatGPT来说，这个人已经到了“故事的结尾”)。

如果我们建立一个变压器网只有一个注意块8头和特征向量长度128（ChatGPT也使用特征向量长度128，但有96注意块，每个96头）那么似乎不可能让它学习括号语言。但是有了两个注意块，学习过程似乎收敛了——至少在给出了1000万左右的例子之后（而且，就像变压器网中常见的那样，显示更多的例子似乎只是降低了它的性能）。

因此，有了这个网络，我们就可以模拟ChatGPT所做的事情，并询问下一个标记的概率——用括号序列：



在第一种情况下，网络“非常确定”序列不能在这里结束——这很好，因为如果它结束了，圆括号就会保持不平衡。然而，在第二种情况下，它“正确地识别”序列可以在这里结束，尽管它也“指出”可以“重新开始”，放下一个“（”，大概是一个“）”来跟随。但是，哎呀，即使它有40万左右的重量，它说有15%的可能性有“)”作为下一个令牌——这是不对的，因为这必然会导致不平衡的括号。

以下是如果我们要求网络对越来越长的(‘s：

是的，在一定长度内，网络做得很好。但后来它就开始失败了。在这样一种“精确”的情况下，用神经网络（或一般的机器学习）看到这是一种非常典型的事情。对于人类“一眼就能解决”的情况，神经网络也能解决。但是需要做一些“更多算法”的案例。g.显式计算

括号，看看它们是否封闭)神经网络往往“计算太浅”，无法可靠地做到。（顺便说一下，即使是完整电流的ChatGPT也很难在长序列中正确匹配圆括号。）

那么，这对于像ChatGPT这样的东西和像英语这样的语言的语法意味着什么呢？括号语言是“严肃”——更像是一个“算法故事”。但在英语中，能够根据当地选择的单词和其他提示来“猜测”什么是适合的语法方式，要现实得多。是的，神经网络在这方面做得更好——即使它可能会错过一些“正式正确”的情况，嗯，人类可能也会错过。但主要的一点是，语言有一个整体的句法结构——包括所有的规律性——在某种意义上限制了神经网络必须学习“多少”。一个关键的“自然科学”观察是，像ChatGPT中的神经网络的转换结构似乎能够成功地学习那种嵌套树状语法结构，这种结构似乎存在于所有人类语言中（至少在某种近似中）。

语法对语言提供了一种约束。但显然还有更多。像“好奇的电子为鱼吃蓝色理论”这样的句子在语法上是正确的，但并不是人们通常期望说的，如果ChatGPT产生它，也不会被认为是成功的——因为，嗯，对于其中单词的正常含义，它基本上是没有意义的。

但是，有没有一种一般的方法来判断一个句子是否有意义呢？在这方面并没有传统的整体理论。但人们可以认为ChatGPT是在接受了数十亿个（可能有意义的）句子的训练后，含蓄地“发展了一种理论”。

这个理论会是什么样子的？有一个小角落已经知道两千年了，这就是逻辑。当然，在亚里士多德发现的三段论形式中，逻辑基本上是一种表达方式，即遵循某些模式的句子是合理的，而其他句子则不是。因此，例如，说“所有的X都是Y”是合理的。这不是Y，所以它不是X”(如“所有的鱼都是蓝色的”。这不是蓝色的，所以它不是一条鱼。”).就像人们可以有点异想天开地想象的那样，亚里士多德

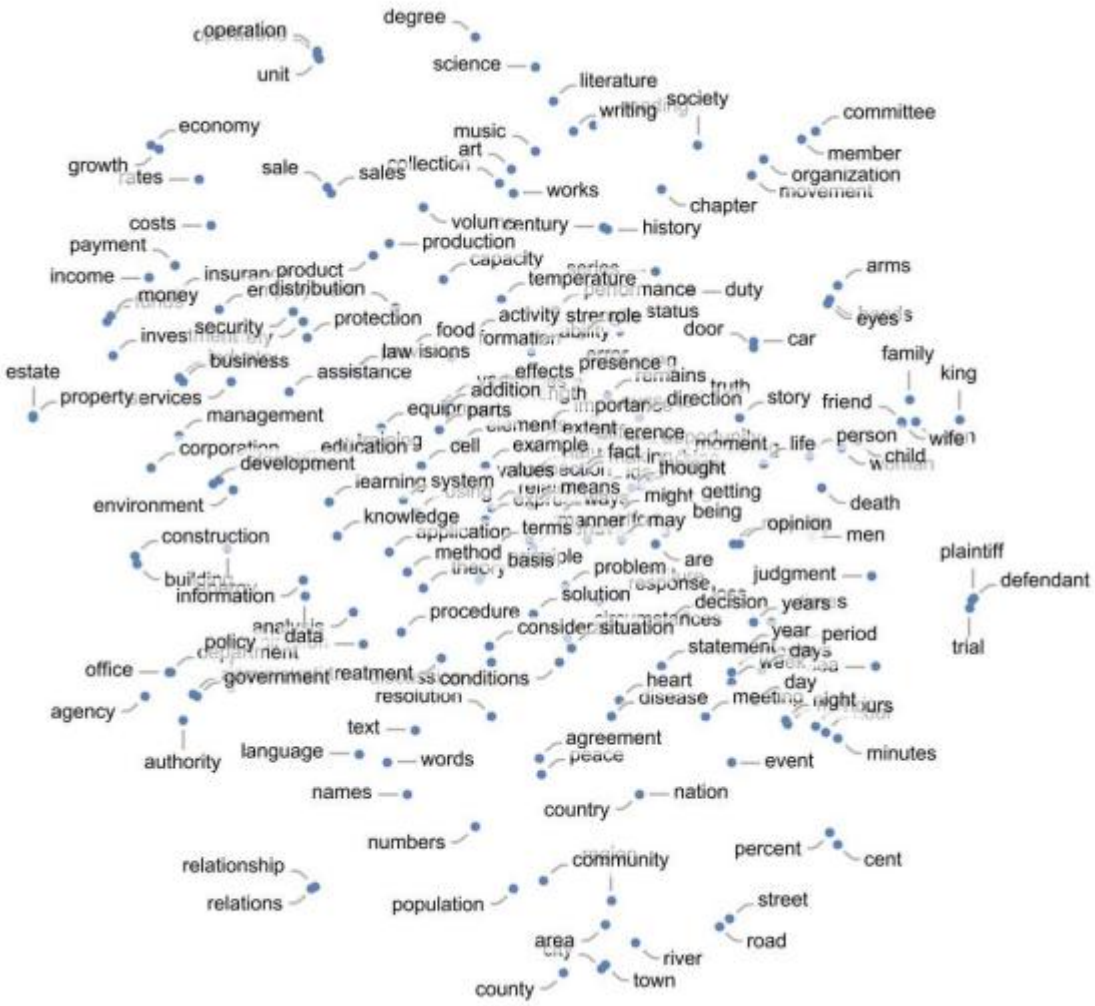
通过大量的修辞学例子（“机器学习风格”）发现了三段论逻辑，所以我们也可以想象，在ChatGPT的训练中，它将能够通过查看网络上的大量文本来“发现三段论逻辑”，等等。（是的，虽然可以期待ChatGPT产生文本包含“正确的推论”基于三段论逻辑，这是一个完全不同的故事时更复杂的正式逻辑，我认为可以期待它失败同样的原因失败在括号匹配。）

但是，除了狭隘的逻辑例子之外，关于如何系统地构建（或识别）甚至是看似有意义的文本，还有什么可以说的呢？是的，有些像疯狂 利布斯&reg 它使用了非常具体的“短语模板”。但不知何种方式，ChatGPT隐式地有一个更通用的方法。也许除了“当你有1750亿个神经净权值”之外，它还没有什么可说的。但我强烈怀疑，还有一个更简单、更有力的故事。

意义空间和运动的语义定律

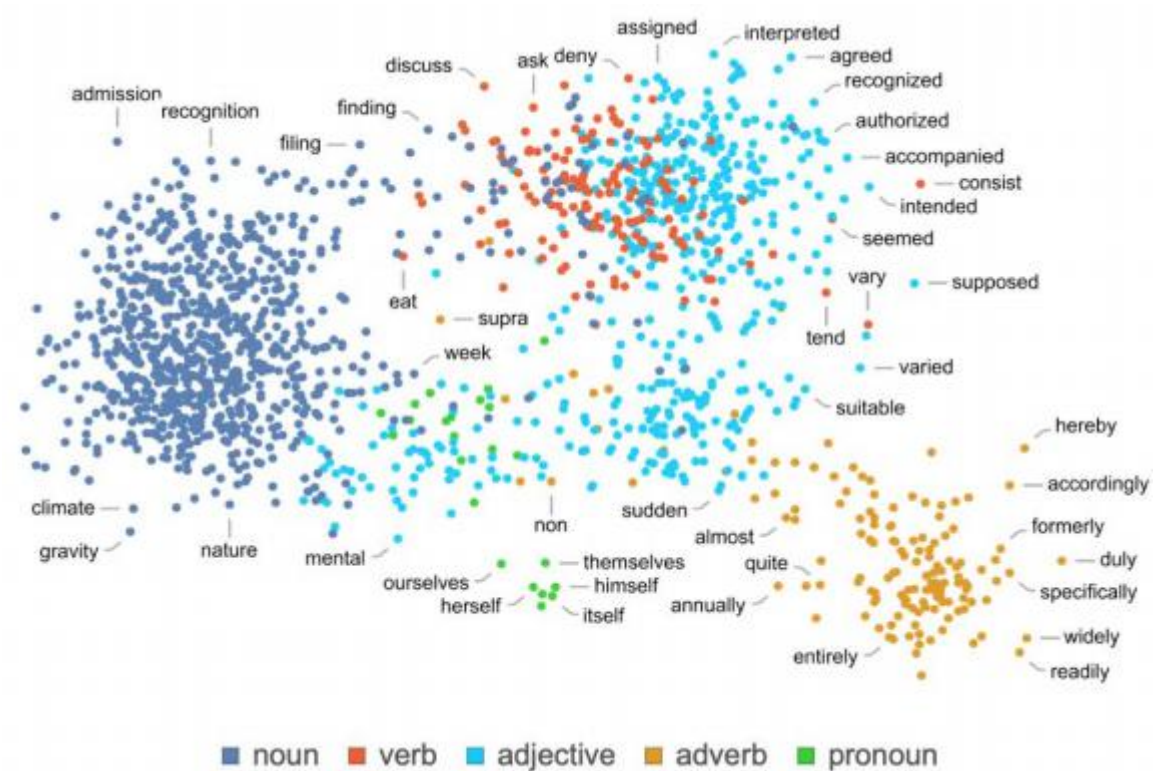
我们在上面讨论过，在ChatGPT内部，任何一段文本都可以有效地用一组数字来表示，我们可以将其看作是某种“语言特征空间”中的一个点的坐标。所以当ChatGPT继续一段文本时，这对应于在语言特征空间中追踪一个轨迹。但现在我们可以问，是什么使这个轨迹与我们认为有意义的文本相对应。也许会有某种“运动语义定律”来定义——或者至少限制——语言特征空间中的点如何在保持“意义”的同时移动？

那么这个语言特征空间是怎样的呢？这里有一个例子，说明如果我们将这样一个特征空间投影到二维，单个单词（这里是常见名词）是如何布局的：

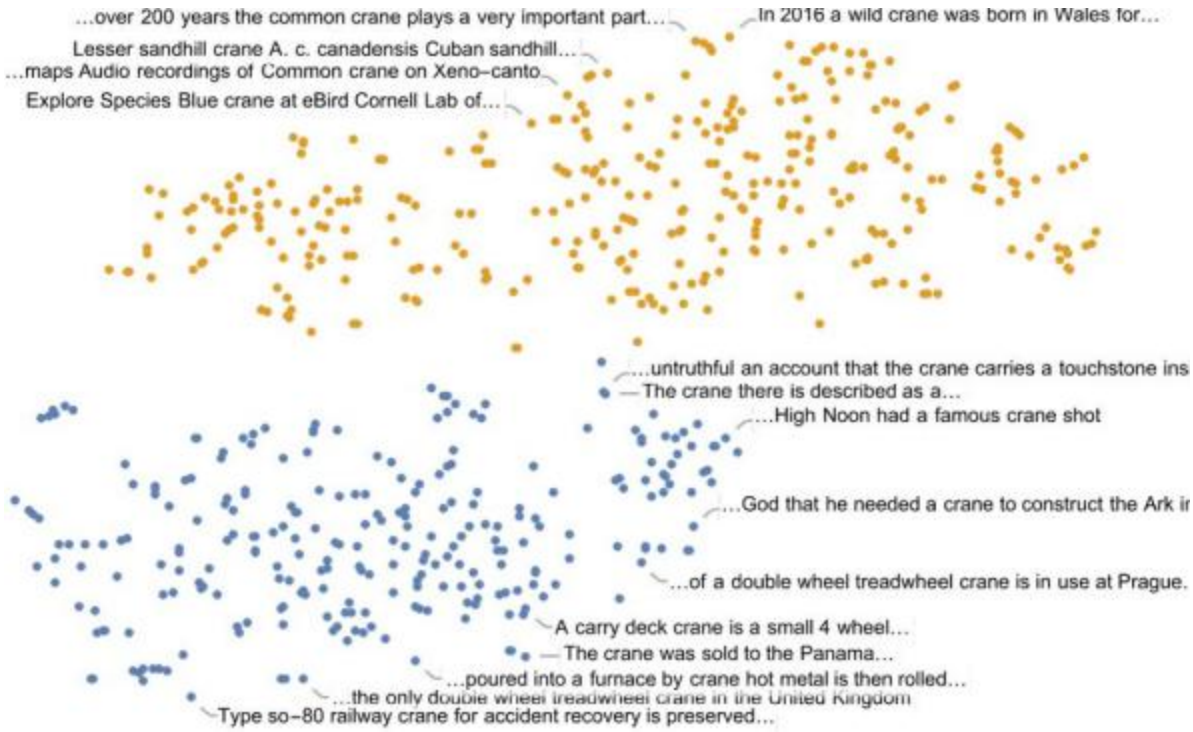


我们在上面看到了另一个基于代表植物和动物的单词的例子。但这两种情况的关键是，“语义上相似的单词”被放在附近。

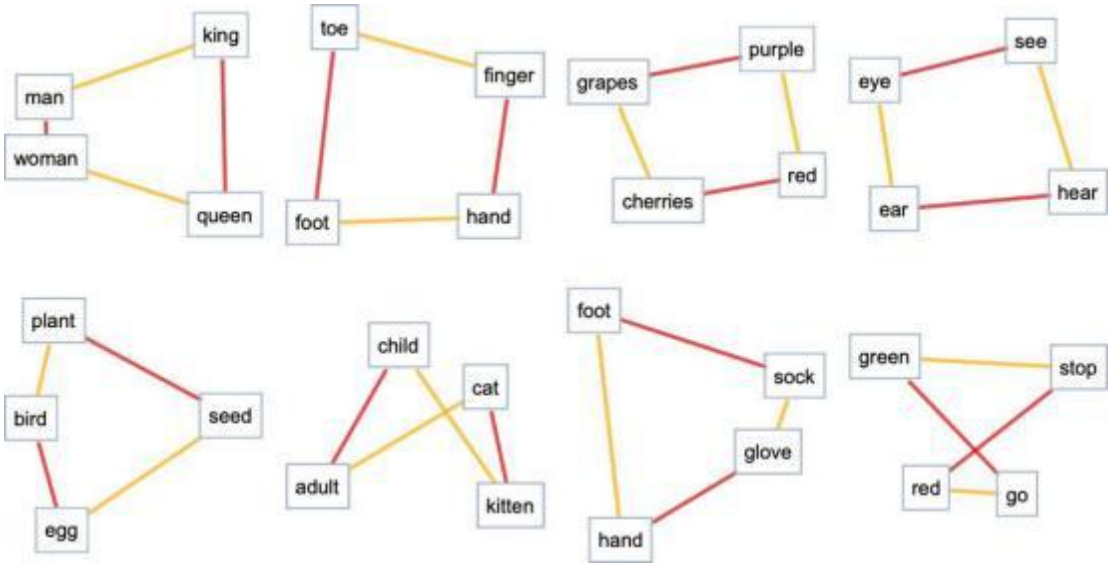
另一个例子是，下面是不同词性对应的单词：



当然，一个给定的单词一般并不只是有“一个意思”（或者一定只对应于说话的一部分）。通过观察包含一个单词的句子如何在特征空间中排列，人们通常可以“梳理”不同的含义——比如这里的例子“起重机”（鸟还是机器？）：

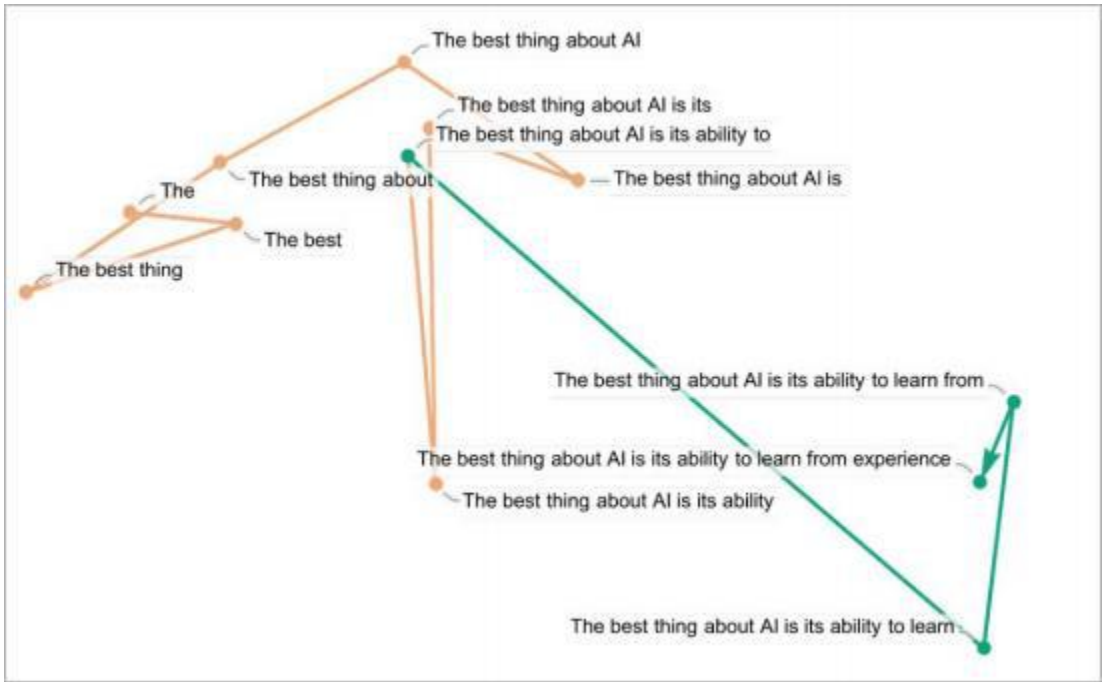


好吧，所以至少我们可以把这个特征空间当作“附近的单词”放在这个空间里。但是，在这个空间中，我们还能识别出什么样的额外结构呢？例如，有没有某种“平行运输”的概念可以反映空间中的“平坦度”？处理这个问题的一种方法是看看类比：



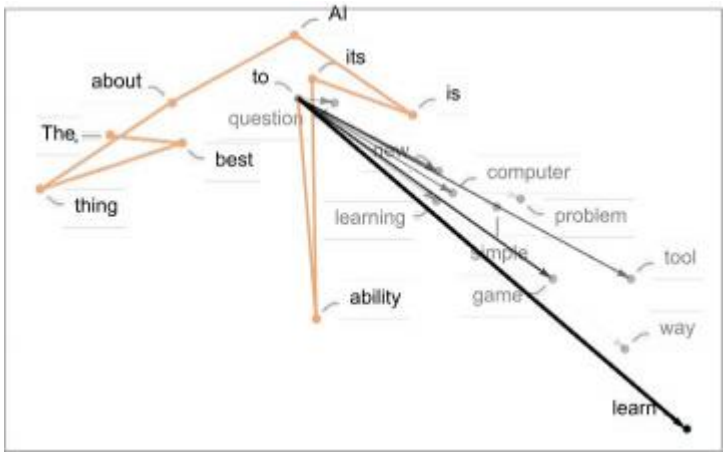
是的，即使当我们投射到2D时，也经常至少有一种“平坦的暗示”，尽管它肯定不是普遍可见的。

那么轨迹呢？我们可以查看ChatGPT提示在特征空间中所遵循的轨迹，然后我们就可以看到ChatGPT是如何继续下去的：

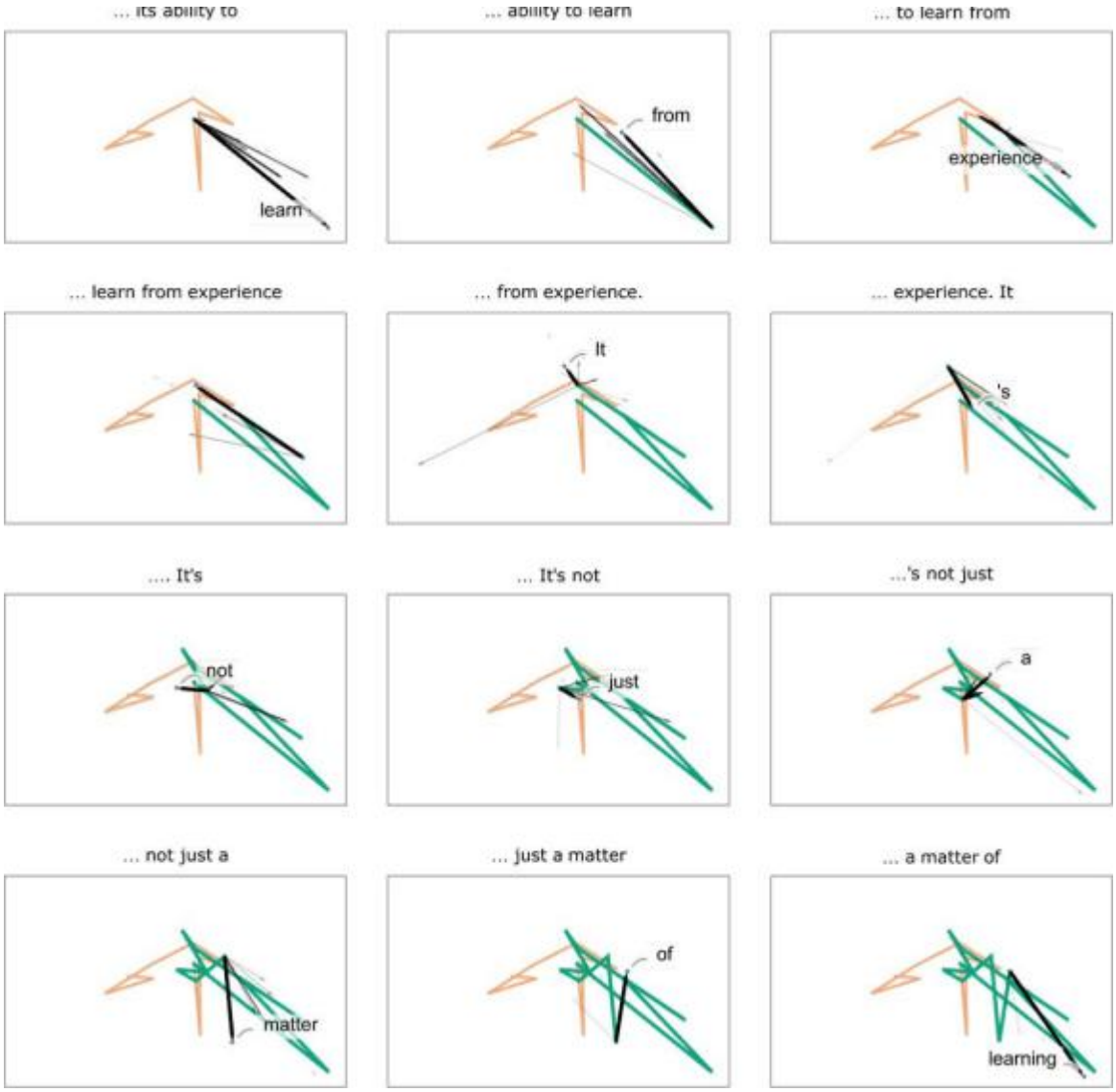


这里当然没有“几何上明显”的运动定律。[这一点也不奇怪；我们完全希望这将是一个相当可观的数字 更复杂 故事](#)例如，即使存在一个“语义运动定律”，也要找到什么样的嵌入（或者，实际上，是什么“变量”），这也很不明显。

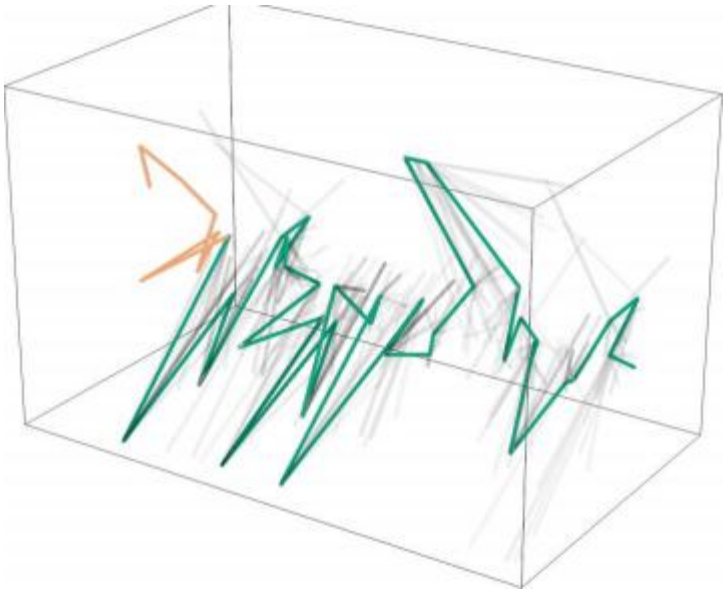
在上面的图片中，我们展示了“轨迹”中的几个步骤——在每一步中，我们选择ChatGPT认为最可能的单词（“零温度”的情况）。但我们也可以问，在一个给定的点上，什么词可以“下一步”：



我们在这种情况下看到的是，有一个高概率单词的“粉丝”，它们在特征空间中似乎或多或少地朝着一个明确的方向发展。如果我们走得更远会怎么样？以下是我们“沿着”轨迹前进时出现的连续“粉丝”：



这是一个3D表示法，总共需要40个步骤：



是的，这似乎是一团糟——并没有任何特别鼓励这样的想法，即通过经验研究“ChatGPT内部做什么”来识别“数学物理类”的“运动语义定律”。但也许我们只是在看“错误的变量”（或错误的坐标系），如果我们只看正确的变量，我们就会立即看到ChatGPT正在做一些“数学-物理简单”的事情，比如遵循测地线。但到目前为止，我们还没有准备好从它的“内在行为”中“经验解码”ChatGPT“发现”的人类语言如何“组合”。

语义语法和语言的力量
计算语言

如何才能产生“有意义的人类语言”？在过去，我们可能会认为它可能是人类的大脑。但现在我们知道，通过ChatGPT的神经网络，我们可以做到这一点。不过，也许这已经是我们所能做的事情了，这样就没有什么更简单的——或者更让人类可以理解的——是可行的了。但我强烈怀疑ChatGPT的成功含蓄地揭示了一个重要的“科学”事实：实际上有更多的结构和简单有意义的人类语言，最后可能会有相当简单的规则，描述这样的语言如何放在一起。

正如我们上面提到的，句法语法给出了如何在人类语言中将与不同词性对应的单词放在一起的规则。但要处理意义，我们做得更进一步。其中的一个版本不仅考虑语言的语法，还考虑语义语法。

为了句法，我们识别名词和动词。但出于语义学的目的，我们需要“更精细的层次化”。因此，例如，我们可以确定“移动”的概念，以及“保持其身份独立于位置”的“物体”的概念。这些“语义概念”都有无数的具体例子。但为了我们的语义语法的目的，我们只有一些一般的规则，基本上说“对象”可以“移动”。关于这一切是如何运作的，有很多话要说 的 哪一个 已经 上述的 在…之前但我将满足于我自己在这里的一些评论，表明一些潜在的前进道路。

值得一提的是，即使根据语义语法，一个句子是完全可以的，但这并不意味着它在实践中已经被实现（甚至可以被实现）。“前往月球的大象”无疑会“通过”我们的语义语法，但在我们的现实世界中（至少还没有）——尽管对于一个虚构的世界来说，这绝对是公平的游戏。

当我们开始谈论“语义语法”时，我们很快就会问：“在它下面是什么？”它假设的“世界模型”是什么？句法语法实际上是用单词构建语言。但是语义语法必然与某种“世界模型”有关——某种作为“骨架”，在骨架之上由实际单词构成的语言可以分层。

直到最近，我们可能还想象（人类）语言将是描述我们“世界模式”的唯一通用方式。早在几个世纪前，就开始有了特定种类的事物的形式化，特别是基于数学。但现在有了一种更普遍的形式化方法：计算性语言

是的，这是我四十多年来的大项目(正如现在体现在沃尔弗拉姆人身上的那样语言)：发展一种精确的符号表征，可以尽可能广泛地谈论世界上的事物，以及我们所关心的抽象事物。例如，我们有城市，分子，图像的符号表示，图像和神经网络，我们有关于如何计算这些东西的内置知识。

经过几十年的工作，我们已经以这种方式涵盖了很多领域。但在过去，我们并没有特别处理过“每天都要处理的问题”演讲在“我买了两磅苹果”中，我们可以轻松地做到表示（并做营养和其他计算）“两磅苹果”。但我们（现在还没有）有一个“我买了”的象征性表现。

这一切都与语义语法的概念有关——以及为概念提供一个通用的符号“构建工具包”的目标，这将给我们提供什么可以与什么相匹配的规则，从而为我们可能变成人类语言的“流动”提供规则。

但假设我们有这种“象征性的话语语言”。我们会怎么处理它呢？我们可以开始做像生成“本地有意义的文本”这样的事情。但最终我们可能想要更多“具有全球意义”的结果——这意味着“计算”更多地关于世界上可能存在或发生什么（或者在某个一致的虚构世界中）。

现在在沃尔弗拉姆语言中，我们有大量关于各种事物的内置计算知识。但是对于一种完整的符号语篇语言，我们必须构建关于世界上一般事物的额外“计算”：如果一个对象从a移动到B，从B移动到C，那么它就从a移动到C，等等。

给定一种象征性的话语语言，我们可以用它来做出“独立的陈述”。但我们也可以用它来问关于这个世界的问题，“Wolfram|Alpha风格”。或者我们可以用它来陈述我们“想要这样做”的事情，大概是通过一些外部驱动机制。或者我们可以用它来断言——也许是关于现实世界，或者是我们正在考虑的特定世界，虚构的或其他的。

人类语言从根本上是不精确的，尤其是因为它没有“绑定”到特定的计算实现，它的含义基本上是由用户之间的“社会契约”定义的。但是计算语言，就其本质而言，具有一定的基本精度——因为最终它所指定的内容总是可以“在计算机上明确地执行”。人类的语言通常会有一定的模糊性。（当我们说“行星”时，它是否包括系外行星，等等？）但在计算语言中，我们必须精确和清楚我们所做的所有区别。

利用普通的人类语言用计算语言编名字通常很方便。但它们在计算语言中的含义必然是精确的——可能也可能不涵盖典型人类语言使用中的某些特定含义。

人们应该如何找出适合一般符号话语语言的基本“本体论”？嗯，这并不容易。这也许就是为什么自从亚里士多德前亚里士多德创造的原始开端以来，这些研究所做的很少。但今天我们现在知道如何用计算来思考世界，这真的很有帮助(从我们的物理学中得到一个“基本形而上学”并没有坏处 项目和想法 的 那 ruliad).

但是在ChatGPT的背景下，这一切意味着什么呢？从它的训练中，ChatGPT有效地“拼凑”了一定数量（相当令人印象深刻）的语义语法。但它的成功给了我们一个理由，认为用计算语言的形式构建更完整的东西是可行的。而且，不像我们到目前为止所发现的ChatGPT的内部结构，我们可以期待设计出计算语言，使人类很容易理解它。

当我们讨论语义语法时，我们可以类比三段论逻辑。起初，三段论逻辑本质上是一个关于用人类语言表达的陈述的规则的集合。但是（是的，两千年后）当正式的 逻辑 是发展后，三段论逻辑的原始基本结构现在可以用来建造巨大的“正式塔”，例如，包括现代数字电路的操作。因此，我们可以预期，它将是更一般的语义语法。首先，它可能只能处理简单的模式，比如用文本表示。但一旦整个计算语言框架，我们可以期待它将能够被用来竖立高楼的“广义语义逻辑”，让我们在一个精确的和正式的方式与各种各样的东西从未访问我们之前，除了在“底层”通过人类语言，与所有的模糊。

我们可以把计算语言和语义语法的构造看作是一种表示事物的最终压缩。因为它允许我们谈论什么是可能的本质，而不是，例如，处理普通人类语言中存在的所有“短语的转折”。我们可以把ChatGPT的巨大力量看作是有点相似的：因为它在某种意义上“钻穿”，可以“以语义有意义的方式将语言放在一起”，而不考虑不同可能的短语变化。

那么，如果我们将ChatGPT应用到底层的计算语言中，会发生什么呢？计算语言可以描述可能的情况。但仍然可以添加的是一种“什么是受欢迎的”的感觉——例如，在阅读网络上所有的内容。但是，在下面，使用计算语言进行操作意味着像ChatGPT这样的东西已经有了

即时和基本的访问什么数量的最终工具，以利用潜在的不可约计算。这使得它成为一个不仅可以“生成合理的文本”的系统，而且可以期望计算出这些文本是否真的对世界做出了“正确的”的陈述——或者它应该谈论的任何东西。

...那么ChatGPT在做什么呢？为什么它会有效呢？

ChatGPT的基本概念在某种程度上是相当简单。从从网络、书籍等中创造的大量人工文本样本开始。然后训练一个神经网络来生成“像这样”的文本。特别是，让它能够从一个“提示符”开始，然后继续使用“就像它被训练过的东西一样”的文本。

正如我们所看到的，ChatGPT中实际的神经网络是由非常简单的元素组成的——尽管其中有数十亿个元素。神经网络的基本操作也非常简单，本质上是传递来自迄今为止生成的文本的元素（没有任何循环，等等）。对于它生成的每一个新单词（或单词的一部分）。

但值得注意的是，意想不到的是，这个过程可以成功地产生文本，就像”网络、书籍中的内容一样。它不仅是连贯的人类语言，而且“说”“遵循它的提示”，利用它“读”的内容。它并不总是说一些“在全局上是有意义的”的东西（或对应于正确的计算）——因为(没有，例如，访问“计算性的”超级大国 的Wolfram|Alpha)它只是根据训练材料中的东西“听起来像”来说一些“听起来正确”的东西。

ChatGPT的具体工程设计使它相当引人注目。但最终（至少在它能够使用外部工具之前）ChatGPT“只是”从它积累起来的“传统智慧的统计数据”中提取出一些“连贯的文本线索”。但令人惊讶的是，这些结果是多么的像人类。正如我所讨论的，这暗示了一些至少在科学上非常重要的东西：人类语言（以及其背后的思维模式）在结构上比我们想象的更简单，更像“法律”。ChatGPT已经含蓄地发现了它。但我们可以用语义语法、计算语言等来显式地公开它。

ChatGPT在生成文本方面所做的事情非常令人印象深刻——而且结果通常与我们人类将产生的结果非常相似。那么，这是否意味着ChatGPT就像大脑一样工作呢？它潜在的人工神经网络结构最终模拟了大脑的理想化。似乎很有可能当我们人类产生语言时所发生的许多方面都是非常相似的。

当涉及到训练（又名学习）时，大脑和当前计算机的不同“硬件”（也许，还有一些未开发的算法思想）迫使ChatGPT使用一种可能比大脑相当不同（在某种方面效率低得多）的策略。还有其他一些东西：与典型的算法计算不同，ChatGPT在内部没有“有循环”或“对数据重新计算”。这不可避免地限制了它的计算能力——即使是对当前的计算机，但肯定是对大脑。

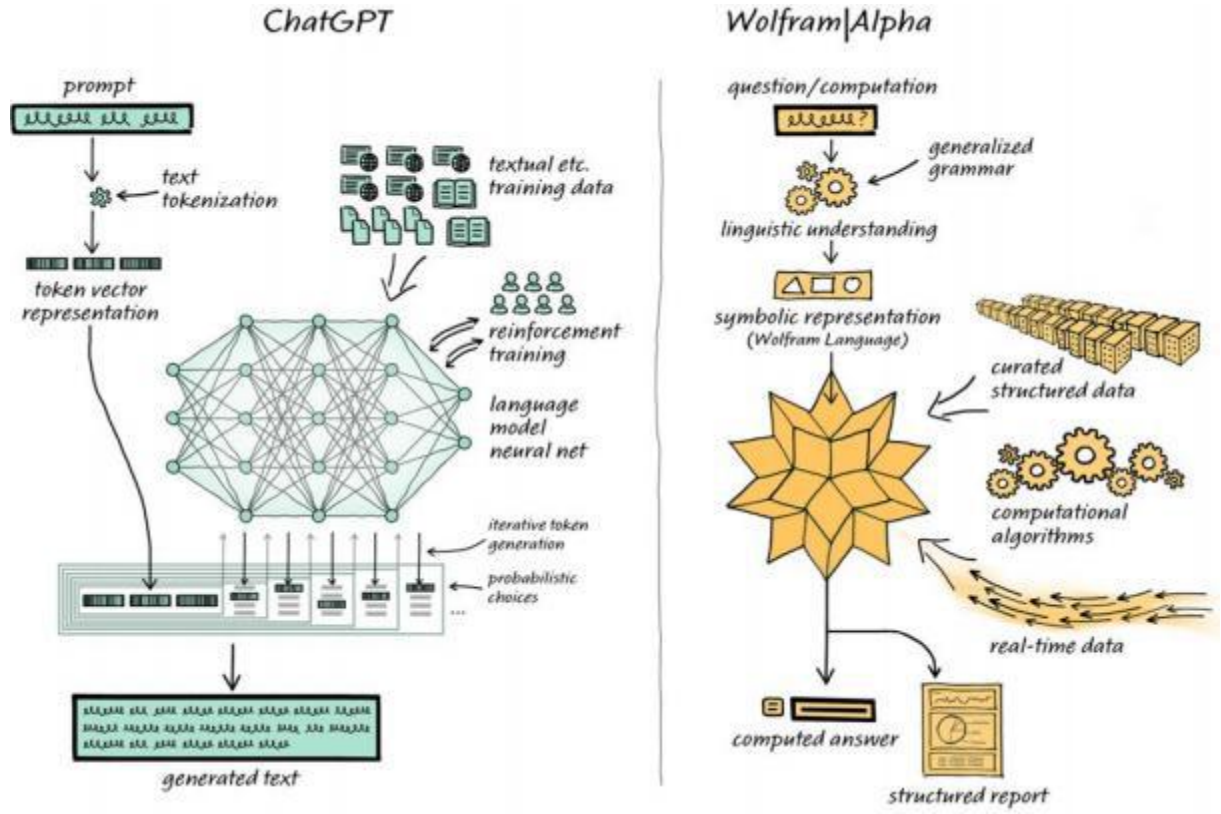
目前还不清楚如何“解决这个问题”，并仍然保持以合理的效率训练系统的能力。但这样做可能会让未来的ChatGPT做更多“类似大脑的事情”。当然，有很多事情大脑做得不太好——特别是涉及到不可约的计算。对于这些大脑和像ChatGPT这样的东西，它们都必须寻求“外部工具”——比如Wolfram 语言

但现在，看到ChatGPT已经能够做些什么是令人兴奋的。在某种程度上，这是基本科学事实的一个很好的例子，即大量简单的计算元素可以做一些非凡的和意想不到的事情。但它也提供了我们两千年来最好的动力来更好地理解人类语言的基本特征和原则的核心特征及其背后的思考过程。

谢谢

我研究神经网络的发展已经有43年了，在这段时间里，我和很多人交流过。其中一些来自很久以前，有些来自最近，有些来自多年，有：朱利奥·亚历山德里尼，达里奥·阿莫迪，艾蒂安·伯纳德，塔蒂埃森贝农，塞巴斯蒂安博登斯坦，格雷格·布罗克曼，杰克考恩，杰克考恩，佩德罗·多明戈斯，杰西加尔夫，罗杰·格蒙松，罗伯特赫克特-尼尔森，杰夫·辛顿，约翰霍普菲尔德，勒昆，杰里莱特文、洛拉杜尔、马文·明斯基、埃里克·姆乔斯内斯、凯登皮尔斯、托马索·帕尔瓦雷扎、特里塞诺夫斯基、奥利弗·塞尔弗里奇、戈登·肖、乔纳斯·斯约伯格，伊利亚·苏茨科弗，格里·特索罗和蒂莫西·维迪尔。对于这篇文章的帮助，我要特别感谢朱利奥·亚历山德里尼和布拉德·克利。

Wolfram|Alpha的方式
带来计算知识
超级大国ChatGPT



ChatGPT和Wolfram|阿尔法

当事情突然间“只是”
工作这件事发生在我们回来的Wolfram|Alpha身上
2009.这发生在我们的物理学中 项目在2020年。
现在OpenAI的ChatGPT正在发生。

我一直在跟踪神经 网 长期的技术
时间 关于 43 年 的确甚至有
看着过去几年的事态发展，我发现
ChatGPT的性能非常显著。
最后，突然之间，这里有一个系统可以
成功地生成了关于几乎任何东西的文本-

这和人类可能写的东西非常相似。
它令人印象深刻，也很有用。而且，正如我所讨论的
在其他地方，我认为它的成功可能告诉了我们
一些非常基本的东西
人类思维

但是，虽然ChatGPT是一个非凡的成就
自动化地做一些类似人类的主要事情，而不是
所有有用的事情都是“人类”的
像有些反而更正式和
有结构的这确实是最伟大的成就之一
我们在过去的几个世纪里的文明已经成功了
是为了建立数学的范式
精确的科学，最重要的是，现在
计算，并创建一个能力的塔
这与纯粹的类人思维完全不同
可以实现。

我自己也已经深入参与了
几十年来的计算范式，在美国
建立一个计算器的奇异追求 语言
在世界上代表尽可能多的事物
以正式的象征性方式。这是我的目标
已经建立了一个系统，可以“计算”
帮助”和增加我和其他人想做的事情。
我把事物看作是一个人。但我也可以
立即拜访沃尔夫拉姆 语言和
Wolfram|Alpha来利用一种独特的东西
“计算超能力”，让我做各种事情
超越人类的东西。

这是一种非常强大的工作方式。和
关键是，这不仅对我们人类很重要。
对于类人来说，它同样重要，如果不是更重要的话
AIs也会立即给他们我们能给的东西
作为计算知识的超能力，

它利用了非人类的结构化力量
计算和结构化的知识。

我们才刚刚开始探索这意味着什么
ChatGPT。但很明显，那些美妙的事情
是可能的。Wolfram|Alpha做了一些非常
与ChatGPT不同，以一种非常不同的方式。但是
它们有一个共同的界面：自然语言。和
这意味着ChatGPT可以“交谈”
Wolfram|Alpha，就像人类一样
Wolfram|Alpha将自然语言
从ChatGPT到精确的，符号化的计算
它可以应用其计算量的语言
知识的力量。

几十年来，在思维上一直存在着二分法
关于人工智能之间的“统计方法”的类型
ChatGPT使用的，和“符号方法”
影响Wolfram|Alpha的起点。但是现在
-多亏了ChatGPT的成功
我们在制作沃尔fram|Alpha时所做的工作
理解自然语言，最后就有了
有机会把它们结合起来做成什么
比两者都要强大得多
自己的

一个基本的例子

在其核心上，ChatGPT是一个生成语言输出的系统，它“遵循”网络、书籍和其他在训练中使用的材料的模式”。值得注意的是，作品很像人类，不仅是小规模，而是整个文章。它有连贯的东西，吸引它学到的概念，经常以有趣和意想不到的方式。它所产生的东西总是“统计上可信的”，至少在语言水平上是这样。但是，最终令人印象深刻的是，这当然并不意味着它自信地得出的所有事实和计算都一定是正确的。这是我刚刚注意到的一个例子（是的，ChatGPT有内在的内置随机性，所以如果你尝试这个，你可能不会得到相同的结果）：

这听起来很有说服力。但事实证明，这是错误的，因为Wolfram|Alpha可以告诉我们：

从芝加哥到东京有多远？

输入解释

芝加哥，伊利诺斯州距离
日本东京

结果

6313 英里

单位转换

10 160 公里（公里）

当然，公平地说，这正是 Wolfram|Alpha 所擅长的事情：可以变成精确的计算，可以基于它的结构化的知识。但简单的是，人们可以想到 Wolfram|Alpha 自动帮助 ChatGPT。你可以编程，可以问 Wolfram|Alpha 这个问题（你也可以使用 web API，等等）。：

在[“从芝加哥到东京的距离是多少”，“勺子结果”]

从伊利诺斯州的芝加哥市中心到东京市中心的距离约为 6313 英里

现在再问 ChatGPT 这个问题，并附上这个结果：

从芝加哥到东京有多远？从伊利诺斯州的芝加哥市中心到东京市中心的距离约为 6313 英里

谢谢你纠正我。你说得对，从伊利诺斯州的芝加哥中心到日本的东京中心的距离大约是 6313 英里（10160 公里）。这个距离可以根据每个城市内的具体位置略有不同。

从芝加哥到东京有多远？

从伊利诺斯州的芝加哥到日本的东京的距离约为 6313 英里（10160 公里）。这是一个非常长的距离，通过任何交通方式从一个地方旅行到另一个地方都需要大量的时间。从芝加哥到日本东京的实际飞行时间约为 16 个小时，但这并没有考虑到风和在机场的停留时间等因素。

ChatGPT 非常礼貌地接受纠正，如果你再次问这个问题，它会给出正确的答案。显然，可以有一种更精简的方法来处理 Wolfram|Alpha 的来回问题，但是很高兴看到即使是这种非常直接的纯自然语言方法基本上已经可以工作了。

但是为什么 ChatGPT 一开始就搞错了呢？如果它在训练中看到了在某个地方（例如来自网络）的芝加哥和东京之间的特定距离，它当然可以得到正确的结果。但在这种情况下，神经网络可以很容易地做到的泛化——比如从许多城市之间距离的例子中是不够的；需要一个实际的计算算法。而 Wolfram|Alpha 处理事情的方式则完全不同。它需要自然语言和 then 一 assuming,它是可能的——它将其转换为精确的计算语言(即 Wolfram 语言)，在这种情况下：

地理距离[芝加哥市，东京 c

输出[]=6296,06 英里

城市的坐标和计算它们之间距离的算法是沃尔弗拉姆语言内置计算知识的一部分。是的，沃尔弗拉姆语言有大量的内置合成知识一 the 结果几十年的工作，精心策划现在大量的不断更新数据，实现（通常发明）方法和模型和 algorithms 一和系统地建立一个整个连贯的计算语言。

还有几个例子

ChatGPT 和 Wolfram|Alpha 的工作方式非常不同，并且有非常不同的优势。但是为了了解 ChatGPT 可以利用 Wolfram|Alpha 的优势，让我们[?]在讨论一些情况下，它自己的 ChatGPT 不能做完全正确的事情。而像人类一样，chatgpt 最容易挣扎的一个领域是数学。

> 3 对 73 的功率是什么？
>
> 3 到功率 73 等于 3^73 岁，大约是 14091714236 岁。
>
> 要计算一个指数，您可以使用公式“基”人”指数“=”结果。例如，要计算 3^73，你可以插入 3 作为基数，73 作为指数得到 3^73 =大约 14091714236。
>
> 需要注意的是，这个结果只是一个估计值，因为实际值为 3^73 非常大，大多数计算器或计算机可能无法准确地表示。

如果这是一个有趣的，文章风格的回应。但实际的结果是错误的：

3 对 73 的功率是什么？

输入

373

结果

67 585198 634817 523 235 520443 624317 923

科学符号

$6.758519863481752323552044362431792 3 \times 10^{34}$

号码名称	全名
67 decillion ...	

67000 亿 100 亿美元。

但是，如果 ChatGPT“咨询”了 Wolfram|Alpha，它当然会能够得到正确的结果。让我们试试一些稍微复杂一点的东西：

半轴为 3 和 12 的椭圆的周长

一个椭圆的周长可以用以下公式来近似：

周长= it *人（a 八 2+b 八 2）/2)

其中，a 和 b 分别为椭圆的半长轴和半小轴的长度。半长轴是穿过椭圆中心并将其分为两半的两个轴中较长的一个，半小轴是两个轴中较短的一个。

在具有半长轴 3 和半小轴 12 的椭圆中，周长将为：

周长= it 人(3 人 2+12^2) / 2)
=approximately 27.3

这个椭圆的周长大约是 27.3。

需要注意的是，这个公式是一个近似值，并且使用更复杂的公式可以更准确地计算出椭圆的实际周长。

乍一看，这个结果看起来很棒，我也倾向于相信它。但事实证明，这是错误的，正如 Wolfram|Alpha 可以告诉我们的那样：

半轴为 3 和 12 的椭圆的周长 Q

输入解释

椭圆半轴长度 3　　12 周长

结果

48E(j| 卜 514705

E (m)是参数为 m = k 的第二类完全椭圆积分 ^2

是的，用 ChatGPT 来做数学作业（如果不能咨询 Wolfram| Alpha）可能是个坏主意。它可以给你一个非常合理的答案：

x 的积分是多少 ^2 cos (2x)?

x 的积分 ^2 cos（2x）是表示函数 y = x 曲线下面积的数学表达式 ^2 cos (2x).可用以下公式进行

计算：

∫2 cos (2x) dx = (1/4) x^3 sin（2x）-（1/8）x cos（2x）+ C

其中，C 是一个任意的常数，并且/表示积分符号。

一个函数的积分表示该函数在两点之间的曲线下的面积。它可以通过查找由曲线、x 轴和表示这两点的垂直线所限定的区域的面积来计算。

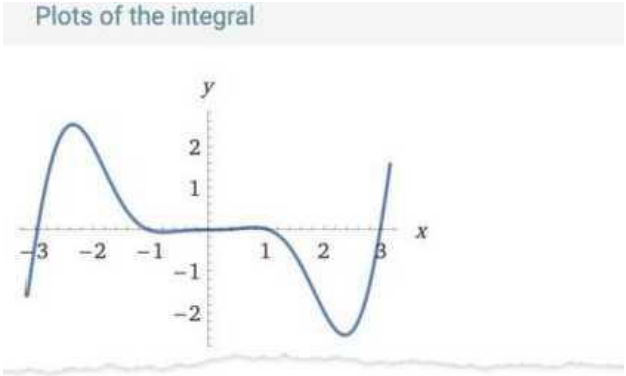
x 的积分的公式 ∫2 cos（2x）可以使用集成技术和规则来导出。需要注意的是，这个公式对于 x 的某些范围是有效的，并且对于 x 的其他值可能不能给出准确的结果。

但没有 u 真正理解数学。“ChatGPT 基本上不可能可靠地得到正确的答案。在这种情况下，答案还是错了：

x 的积分是多少 ∫2 cos (2x) Q

不定积分 [[g 逐步解决方案)

Jx²cos（2x）Jx=：（（2]一 1）sin（2x）+ 2xcos（2x））-常数



尽管如此，ChatGPT 甚至可以解释“它是如何得到答案的”（并不是说它是如何“做到”的）。而且，相当迷人（而且有趣），它给出的解释所犯的错误与一个不懂数学的人可能犯的错误非常相似：

展示一下你是如何派生出来的

推导出 x 的积分公式 ∫2 cos（2x），您可以使用以下步骤：

1. 首先写一个函数，你想找到的积分，在本例中是 y = x^2 cos (2x).

2. 使用集成技术和规则来找到函数的积分。例如，您可以使用以下规则：

，常数乘以函数的积分是常数乘以函数的积分。

，x 的积分 ^n 为（x^（n+1））/（n+1）+ C，其中 C 是一个常数，n 是一个正整数。

sin（x)的积分是-cos（x）+ C，而 cos（x)的积分是 sin（x）+ C，其中 C 是一个常数。

3. 将适当的值代入公式，并简化以得到最终结果。

例如，要找到 x“cos（2x）的积分，您可以使用以下步骤：

1. 从函数 y = x 开始 ^2 cos（2x).
2. 使用对 x 的积分的规则 ^n 来得到 x 的积分 ^2.这给了我们(x^3)/3 + C.
3. 使用 cos（x)的积分规则来找到 cos（2x）的积分。这给了我们（1/2）sin（2x）+ C。
4. 将第 2 步和第 3 步的结果相乘，得到 x 的积分 ^2 cos（2x).这给了我们
   （“3）/3+C)（（1/2）sin（2x）+C）=(x^3)/6 sin（2x）+ Cx^3 + C^2.
5. 简化结果

在各种情况下，“不真正理解事情的意思”会引起麻烦：

A 中美洲的第二大国家是什么国家

中美洲的第二大国家（按陆地面积计算）是危地马拉。危地马拉位于中美洲的北部，北部和西部与墨西哥接壤，东部与伯利兹和加勒比海接壤，南部与洪都拉斯和萨尔瓦多接壤。危地马拉的土地面积约为 108,889 平方公里（42,042 平方英里），略小于中美洲最大的国家尼加拉瓜。

这听起来令人信服。但这是不正确的：

中美洲最大的国家是哪些？

假设“最大”指的是总面积|使用 GDP 或人口代替假设中美洲|使用中美洲和墨西哥

输入解释

最大国家 按总面积

中美洲 更多

结果

1	尼加拉瓜	50 336 mi^2	
2	洪都拉斯	43 278 mi^2	1___ U
3	危地马拉	42042 mi^2	1’ II
4	巴拿马草帽	29 120 mi^2	1： T 1
5	哥斯达黎加	19730 mi^2	1_L__J

ChatGPT 似乎已经正确地学习了这个底层数据 somewhere 一，但它没有“理解它的意思”，能够正确地排列数字：

中美洲最大的国家是什么

按陆地面积排名，中美洲最大的国家有：

1. 尼加拉瓜-130370 平方公里（50193 平方英里）
2. 危地马拉-108889 平方公里（42042 平方英里）
3. 洪都拉斯-112492 平方公里（43278 平方英里）
4. 萨尔瓦多-21041 平方公里（8124 平方英里）
5. 哥斯达黎加-51100 平方公里（19656 平方英里）
6. 巴拿马-78200 平方公里（30193 平方英里）

值得注意的是，这些数字代表了每个国家的总土地总面积，包括属于该国一部分的任何岛屿。中美洲的一些国家，如洪都拉斯和萨尔瓦多，规模相对较小，而其他国家，如尼加拉瓜和危地马拉，规模较大。

是的，人们可以想象找到一种方法来“修复这个特定的错误”。但关键是，像 ChatGPT 这样的基于生成语言的 Al 系统的基本思想并不适合有结构化计算的事情要做的情况。换句话说，它需要“修复”几乎无限数量的“错误”来修补即使是几乎无穷小的 Wolfram|Alpha 能够以其结构化的方式实现的目标。

而且“计算链”变得越复杂，你就越有可能不得不使用 Wolfram|Alpha 来纠正它。在这里，ChatGPT 产生了一个相当令人困惑的答案：

水星有多大，还有什么行星卫星比它大吗？

水星是我们太阳系中最小的行星。它只比地球上的月球稍大一点，直径约为 3032 英里（4880 公里）。在我们的太阳系中，没有一个行星卫星比水星更大。太阳系中最大的卫星是木星的四个伽利利卫星：伽利利木卫三、银河系、木卫一和木卫二。甘尼米德是其中最大的一个，直径约为 3274 英里（5268 公里），这使得它比水星略大。然而，这些卫星围绕着木星运行，木星比水星要大得多。

而且，正如 Wolfram|Alpha 告诉我们的那样，它的结论是不正确的（因为它在某种意义上已经“知道”了）：

哪些行星卫星比水星还大？

假设“较大”是指平均半径|使用质量代替

输入解释

大于水星平均半径的行星卫星

结果

带去为众神司酒的美少年

巨子

轨道性质 显示更多度量

每当涉及到具体的（例如定量的）数据，甚至是相当原始的形式——事情往往必须是a“Wolfram|Alpha 的故事”。这里有一个例子，灵感来自于一个长期以来最喜欢的 Wolfram|Alpha 测试查询“土耳其有多少只火鸡？””:

告诉我关于土耳其的牲畜数量

土耳其是牲畜的主要生产国，包括家禽、牛、绵羊和山羊。根据联合国粮食及农业组织（粮农组织）的数据，截至 2020 年，土耳其的牲畜数量如下：

家禽：土耳其拥有一个庞大的家禽业，大约有 5.37 亿只鸡，3700 万只火鸡和 300 万只鸭子。
牛：土耳其的牛的数量相对较少，大约有 740 万头牛。
绵羊：土耳其的绵羊数量更多，大约有 2560 万只绵羊。山羊：土耳其还有大量的山羊数量，大约有 1070 万只山羊。

除了这些主要的牲畜物种，土耳其还有较少的其他动物，如马、驴和骆驼。

再一次，这似乎（一开始）完全可信，它甚至引用了一个相关的来源。但事实证明，这些数据基本上都是“编造出来的”：

土耳其牲畜数量

输入解释

土耳其牲畜数量

结果 更多

牛	1800 万(2020 年估计）
鸡	3.79 亿(2020 年估计）
山羊	1200 万(2020 年估计）
无源投资发电机	990 (2020 年估计）
羊	4210 万(2020 年估计）

不过，非常好的是，ChatGPT 可以很容易地“要求事实来核实”：

给我一个简单的关键字查询，它可以由您的响应中的每个数值来回答；只列出查询

“土耳其鸡数量”“土耳其火鸡数量”“土耳其鸭子数量”“土耳其牛数量”“土耳其绵羊数量”“土耳其山羊数量”

现在通过 Wolfram|Alpha API 提供这些信息：

在 WolframAlpha[#, "SpokenResult"] &/@
(“土耳其的鸡数量”、“土耳其的火鸡数量”、“土耳其的鸡鸭数量”、“土耳其的牛数量”、“土耳其的绵羊数量”、“土耳其的山羊数量”}

卜(2020 年，土耳其大约有 3.79 亿只鸡作为牲畜，
到 2020 年，土耳其有大约 480 万只火鸡作为牲畜，
到 2020 年，土耳其有 56 万只鸭子作为牲畜，
到 2020 年，土耳其有大约 1800 万头牛作为牲畜，
到 2020 年，土耳其有大约 4210 万只绵羊作为牲畜，
到 2020 年，土耳其有大约 1200 万只山羊作为牲畜}

现在我们可以要求 ChatGPT 修复它的原始响应，注入这些数据（甚至以粗体显示它做它的地方）：

将此修正后的数据插入原始响应，围绕每个修改的值
（“2020 年，土耳其大约有 3.79 亿只鸡作为牲畜”，“2020 年，土耳其大约有 480 万只火鸡作为牲畜”，“2020 年，土耳其有 56 万只鸭子作为牲畜[1]“2020 年，土耳其有 1800 万头牛作为牲畜”，“2020 年，土耳其有 4210 万只绵羊作为牲畜”，“2020 年，土耳其有 1200 万只山羊作为牲畜”）

下面是插入到我之前的响应中的修正数据，修改后的值被

土耳其是牲畜的主要生产国，包括家禽、牛、绵羊和山羊。根据联合国粮食及农业组织（粮农组织）的数据，截至 2020 年，土耳其的牲畜数量如下：

美国，家禽：2020 年，土耳其约有 3.79 亿只鸡作为牲畜，480 万只火鸡作为牲畜，以及约 56 万只鸭子。
例如，牛：在 2020 年，土耳其大约有 1800 万头牛。
在美国，绵羊：2020 年，土耳其大约有 4210 万只绵羊。
山羊：到 2020 年，土耳其大约有 1200 万只山羊。

除了这些主要的牲畜物种，土耳其还有较少的其他动物，如马、驴和骆驼。

当涉及到实时(或位置等的事情时，“注入事实”的能力特别好。依赖的)数据或计算。ChatGPT 不会立即回答这个问题：

今晚我能看到什么行星？

对不起，我没有任何关于今晚天空中行星的信息。为了知道今晚可以看到哪些行星，你需要使用一个天文馆的应用程序或网站。这些工具利用你的位置和当前时间提供在天空中可见的信息。一些流行的天文馆应用程序和网站包括天空指南、星光大道和石碑。或者，你也可以查看打印出来的天空地图，或者向天文学家询问从你的位置可以看到哪些行星的信息。

但这里有一些相关的 Wolfram| Alpha API 输出：

ini/。=值，“今晚我能看到什么行星”，
（（“输入”，叩 ropertyRanking：行星数据”），“明文“））]]
出[卜(行星|在地平线以上晚上 110 点，水星|水星在地平线以下金星 I 金星在地平线以下
火星一号火星在地平线之上

木星 I，木星在地平线之上

土星|土星在地平线以下

天王星|天王星在地平线之上

海王星|海王星在地平线之上}

如果我们把这个提供给 ChatGPT，它会产生一个很好的“文章风格”的结果：

> 修改你的答案，结合这些数据：（“行星|在地平线上晚上 110 点”，“水星|水星水星在地平线以下金星金星在地平线以下火星火星在地平线以上木星木星在地平线上土星土星在地平线以下

> 根据提供的信息，在晚上 10 点，在地平线上的行星是火星、木星、天王星和海王星。水星和金星在地平线以下，而土星也在地平线以下。我应该注意到，行星的能见度可能会根据你所处的位置和一年中的时间而有所不同。我建议使用天文馆应用程序或网站来获得最准确和最行星的最新信息，从你的位置。

有时，在计算量和人类的类似物之间会有一种有趣的相互作用。以下是一个关于沃尔 fram|Alpha 的相当异想天开的问题（它甚至会检查你是否想要“软服务”）：

一立方光年的冰淇淋中有多少卡路里？

假设冰淇淋|用软份冰淇淋代替

假设任何类型的冰淇淋|使用冰淇淋，香草或更多的▼代替

输入解释

冰淇淋量 1 升 ^{3}（立方光年）总热量

平均结果 显示详细信息

1.4x 10^{54}Cal（膳食卡路里）

单位转换

5.8 x 10^{57}J（焦耳）

5.8 x10^{64} 埃格斯
（单位正式弃用）

ChatGPT 一开始对体积的概念感到有点困惑：

一立方光年的冰淇淋中有多少卡路里？

对不起，不可能有一个立方光年，更不用说冰淇淋了。一光年是距离的单位，而不是体积，大约等于 5.88 万亿英里。它是衡量光在一年内传播的距离。因此，不可能有任何立方光年的任何物质，包括冰淇淋。

但后来它似乎“意识到”“那么多冰淇淋是相当愚蠢的：

一个立方光年是一个非常大的体积单位。事实上，它是如此之大，以至于不可能容纳任何已知的物质，包括冰淇淋，进入这个大小的体积。

作为参考，光年是距离的单位，等于光在一年内传播的距离。光在一年内传播的距离约为 5.88 万亿英里（9.46 万亿公里）。

考虑到一个光年的大小，很明显，一个立方光年是一个不可思议的大体积单位。因此，不可能计算一个立方光年冰淇淋的卡路里数，因为没有办法把那么多的 jce 奶油放进一个体积中。

前进的道路

机器学习是一种强大的方法，特别是在过去的十年里，它有一些显著的 successes 一 of，ChatGPT 是最新的。. 图像识别功能。语音到文本语言翻译，在每一种情况下，以及更多的情况下，一个阈值是 passed 一 usually 非常突然。有些任务来自于“基本上不可能完成”¨“基本可行”。

但结果基本上从来都不“完美”。也许有 95%的时间都很有效。但尽管有人可能会努力，但其他 5%仍然难以捉摸。对于某些目的，人们可能会认为这是一种失败。但关键是，通常有各种重要的用例，95%都“足够好”。也许这是因为输出中并没有一个真正的“正确答案”。也许是因为一个人只是试图展示一个 human 一 or 一个系统的算法——然后会从中挑选或改进。

一次生成文本标记的几百亿参数神经网络可以做 ChatGPT 可以做的事情，这是完全值得注意的。考虑到这个 dramatic 一 And unexpected 一 success，人们可能会认为，如果一个人可以继续下去，并“训练一个足够大的网络”，那么他就完全可以用它做任何事情。但这样就不行了。关于 computation 一 and 的基本事实，特别是计算不可约性——的概念，表明它最终不能。但更重要的是我们在机器学习的实际历史中所看到的东西。这将会有一个重大的突破（比如 ChatGPT）。而且，进步也不会停止。但更重要的是，会发现一些用例是成功的，而不能做的不会阻止。

是的，在那里[5]在很多情况下，“TawChatGPT”可以帮助人们写作、提出建议，或生成对各种文档或交互有用的文本。但当涉及到建立一些必须是完美的东西时，机器学习并不是这样做的方法——就像人类也不是一样。

而这正是我们在上面的例子中所看到的。ChatGPT 在“类人部分”方面做得很好，因为那里没有一个精确的“严格的答案”。但当它为了一些精确的东西而“准备到位”时，它经常会下降。但这里的重点是，有一个很好的方法来解决这个 problem 一，通过将 ChatGPT 与 Wolfram|Alpha 和它所有的计算知识“超能力”连接起来。

在 Wblfram|Alpha 中，所有的东西都被转换为计算语言，以及精确的 Wolfram

语言代码，在某种程度上必须“完美”才能可靠地有用。但关键的一点是，ChatGPT 并不需要生成这个。它可以产生它通常的自然语言，然后沃尔弗拉姆|Alpha 可以使用它的自然语言理解能力，将该自然语言翻译成精确的沃尔弗拉姆语言。

在很多方面，人们可能会说 ChatGPT 从来没有“真正理解”过一些事情；它只是"知道如何生产出有用的东西。”但关于 Wolfram|Alpha 的情况就不同了。因为一旦沃尔弗拉姆|Alpha 将某种东西转换为沃尔弗拉姆语言，它所得到的就是一个完整、精确、形式的表示，从中人们可以可靠地计算出东西。不用说，我们也有很多带有“人类利益”的事情[5]如果有正式的计算重构一though，我们仍然可以用自然语言来谈论它们，尽管它可能不精确。对于这些，ChatGPT 是独立的，具有非常令人印象深刻的能力。

但就像我们人类一样，有时 ChatGPT 需要一个更正式、更精确的“力量辅助”。但关键是，它不需要“正式和精确地”来表达它想要的东西。因为 Wolfram|Alpha 可以用相当于 ChatGPT 的母语 language 一自然语言与它交流。而 Wolfram| Alpha 在转换为其母语 Language 一 Wolfram 语言时，将负责“增加形式和精度”。这是一个很好的情况，我认为它有很大的实际潜力。

而且，这种潜力不仅仅体现在典型的聊天机器人或文本生成应用程序的层面上。它可以扩展到诸如做数据科学或其他形式的计算工作（或编程）等事情。从某种意义上说，这可以直接获得两个世界的最佳效果：ChatGPT 的类人世界，以及 Wolfram 语言的计算精确世界。

那么 ChatGPT 直接学习 Wolfram 语呢？是的，它可以做到，事实上它已经开始了。最后，我完全希望像 ChatGPT 这样的东西能够直接用 Wolfram 语言进行操作，并且在这方面非常强大。这是一种有趣而独特的情况，由于沃尔弗拉姆语言作为一种全面的计算语言的特性，它可以用计算术语广泛地谈论世界上和其他地方的事物。

沃尔弗拉姆语的整个概念是要接受我们人类所思考的东西，并能够通过计算来表示并与它们合作。普通的编程语言旨在提供告诉计算机具体要做什么的方法。Wolfram Language 一 in 作为一个比它更大的东西的全面计算 language 一 is。实际上，它是一种人类和计算机都可以“计算^”的语言

许多个世纪前，当数学符号被发明出来时，它第一次提供了一种“用数学方法思考”的流线型媒介，关于事情。它的发明很快导致了代数和微积分，并最终导致了所有的各种数学科学。Wolfram 语言的目标是为计算思维做一些类似的事情，尽管现在不仅仅是为 humans 一 and 启用所有可以被计算范式打开的“计算 X”域。

我自己也从沃尔弗拉姆语言作为一种“需要思考的语言”中获益良多，在过去的几十年里，看到人们通过沃尔弗拉姆语言“用计算术语思考”而取得了如此多的进步是很美妙的。那么 ChatGPT 呢？嗯，它也可以进入这个领域。我还不确定这一切将如何运作。但这并不是关于 ChatGPT 学习如何做 Wolfram 语言已经知道的计算工作。它是关于 ChatGPT 学习如何像人们一样使用沃尔弗拉姆语的。它是关于 ChatGPT 提出的模拟“创意”

但现在不是用自然语言写的，而是用计算语言写的。

我一直在讨论人类写的计算论文的概念，一 that 用自然语言和计算语言进行交流。现在的问题是，ChatGPT 是否能够编写 those 一 and，是否能够使用 Wolfram 语言作为一种传递“有意义的交流”的方式，不仅对人类，也对计算机。是的，有一个潜在的有趣的反馈循环，涉及到 Wolfram 语言代码的实际执行。但关键的一点是，沃尔弗拉姆语言代码 is 一 unlike 所代表的“思想”的丰富性和流动——这更接近 ChatGPT 在自然语言中“神奇”处理的那种东西。

或者，换句话说，Wolfram Language 一 like 自然语言——是一种足够表达力的东西，人们可以想象在其中为 ChatGPT 写一个有意义的“提示”。是的，沃尔弗拉姆语言可以直接在计算机上执行。但作为一个 ChatGPT 提示，它可以用来“表达一个想法”，其“故事”可以继续。它可能会描述一些计算结构，让 ChatGPT“重复”人们可能计算的结构，would 一 according 通过阅读 humans 一 be 写的东西学到的东西。

有各种各样的令人兴奋的可能性，突然打开了 ChatGPT 的意外成功。但现在马上就有机会通过 Wolfram|Alpha 赋予 ChatGPT 计算知识超能力。因此，它不能仅仅产生“看似合理的类人输出”，还可以利用整个用 Wolfram|Alpha 和|语言封装的计算和知识塔的输出。

其他资源

“ChatGPT 在做什么……为什么它有效？”
具有可运行代码 wolfr.am/SW-ChatGPT 的在线版本

“面向中学生的机器学习”，（斯蒂芬·沃尔夫拉姆）
简要介绍机器学习的基本概念
wolfr.am/ML-for-middle-schoolers

机器学习入门课程（艾蒂安·伯纳德著）
使用可运行代码的现代机器学习的书籍长度的指南
印刷：wolfr.am/IML 的书；在线 wolfr.am/IML

沃尔夫拉姆机器学习
沃尔弗拉姆语中的机器学习能力
wolfr.am/core-ML

沃尔弗拉姆大学的机器学习
交互式课程和机器学习的各种水平 wolfr.am/ML-courses

“我们应该如何和 ai 说话呢？”（斯蒂芬·沃尔夫拉姆著）
2015 年的一篇关于用自然和计算语言 wolfr.am/talk-AI 与人工智能进行交流的短文

沃尔弗拉姆语，wolfram.com/language

Wolfram|Alpha wolframalpha.com

在线链接到所有的资源：

wolfr.am/ChatGPT~resources