Issue special-edition | 专题

机器学习原来这么有趣！（六）

亚当·盖特吉 12 个月前

语音识别正在「入侵」我们的生活。我们的手机、游戏主机和智能手表中都内置了语音识别的程序。它甚至在自动化我们的家园。只需 50 美元，你可以买到一个 Amazon Echo Dot，一个能够让你订比萨、获知天气预报，甚至购买垃圾袋的魔术盒——只要你大声说出你的需求：

Alexa，订一个大号的比萨！

Echo Dot 机器人在这个假期（2016 年圣诞）太受欢迎了，以至于 Amazon 似乎都断货了！

然而语音识别明明已经出现几十年了，为何直到现在才成为主流呢？那是因为深度学习终于将语音识别在非受控环境下的准确度提高到了一个足以投入实用的程度。

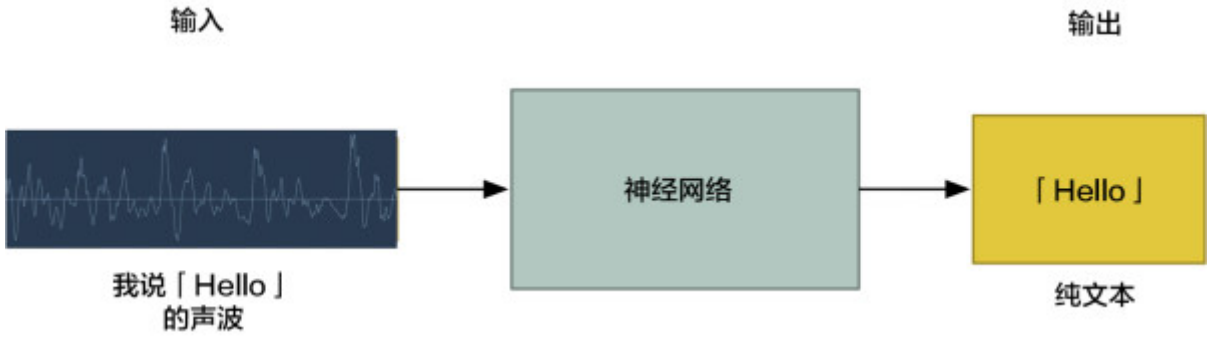
吴恩达教授[1]早有预言，当语音识别的准确度从 95% 上升到 99% 的时候，它将成为我们与计算机交互的主要方式。

这意味着，这 4% 的精度差实际就是「太不靠谱」与「实用极了」之间的差别。多亏了深度学习，我们终于达到了顶峰。

让我们了解一下如何用深度学习进行语音识别吧！

机器学习并不总是一个黑盒

如果你知道神经机器翻译是如何工作的，那么你可能会猜到，我们可以简单地将声音送入神经网络中，并训练使之生成文本：



这就是用深度学习进行语音识别的核心所在，但目前我们还没有完全掌握它（至少在我写这篇文章的时候还没有——我打赌，在未来的几年我们可以做到）。

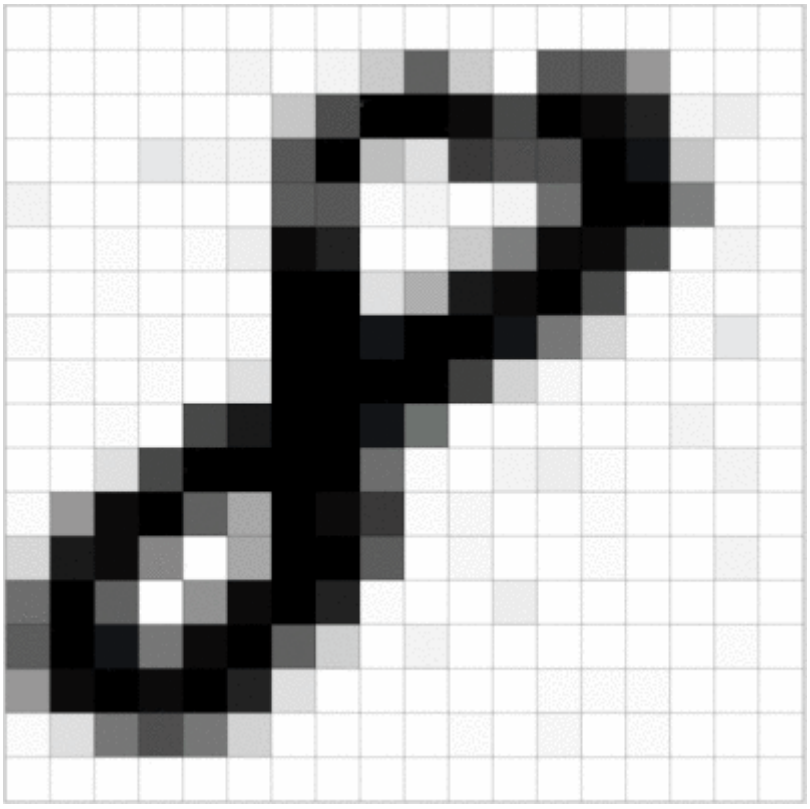
一个大问题是语速不同。一个人可能很快地说出「hello！」而另一个人可能会非常缓慢地说「heeeellllllllllllooooo！」。这产生了一个更长的声音文件，也产生了更多的数据。这两个声音文件都应该被识别为完全相同的文本「hello！」而事实证明，把各种长度的音频文件自动对齐到一个固定长度的文本是很难的一件事情。

为了解决这个问题，我们必须使用一些特殊的技巧，并进行一些深度神经网络以外的特殊处理。让我们看看它是如何工作的吧！

将声音转换成比特（Bit）

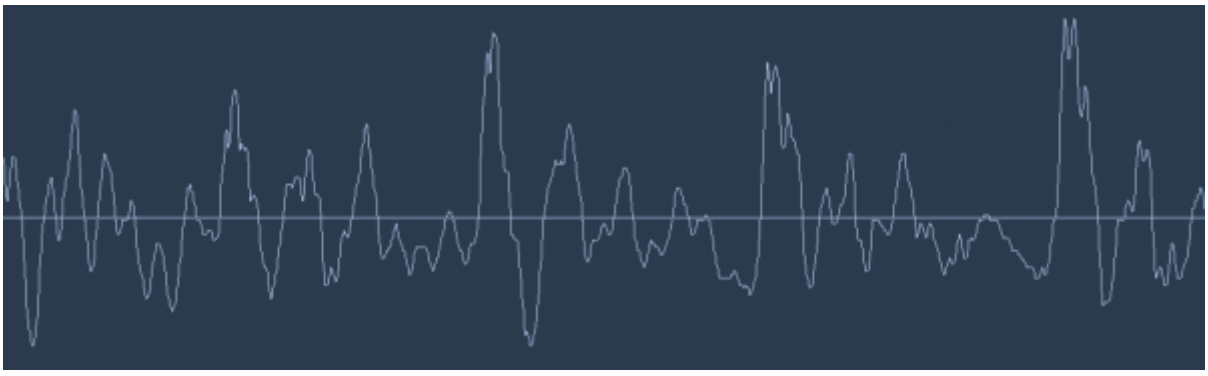
语音识别的第一步是很显而易见的——我们需要将声波输入到计算机当中。

在第三章中，我们学习了如何把图像视为一个数字序列，以便我们直接将其输入进神经网络进行图像识别：



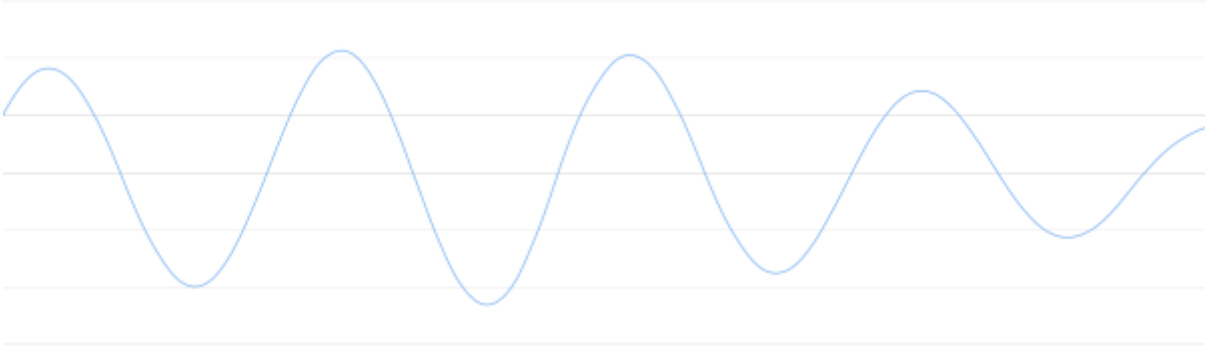
图像只是图片中每个像素深度的数字编码序列。

但声音是作为**波**（wave）的形式传播的。我们如何将声波转换成数字呢？让我们使用我说的「hello」这个声音片段举个例子：

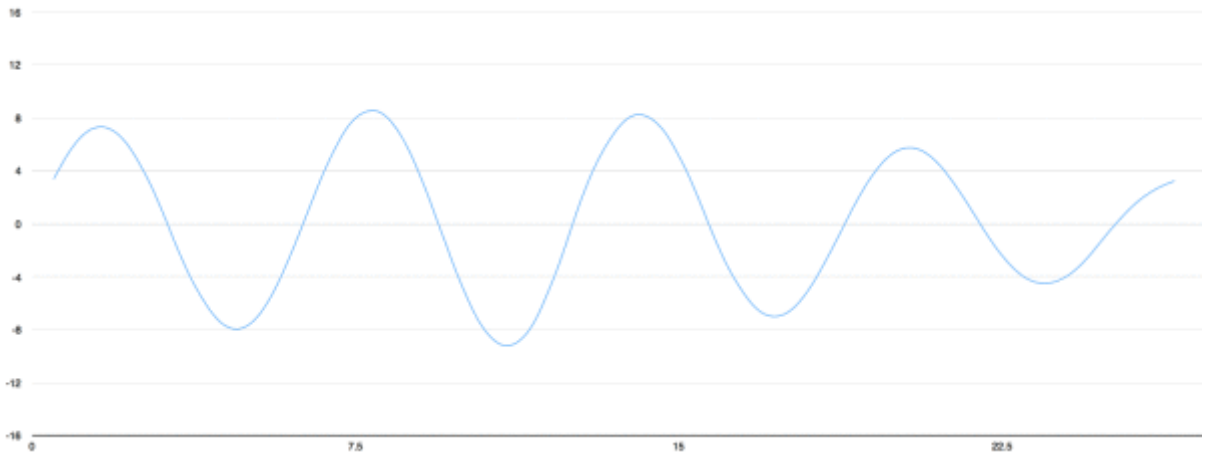


我说「hello」的波形。

声波是一维的，它在每个时刻都有一个基于其高度的值[2]。让我们把声波的一小部分放大看看：



为了将这个声波转换成数字，我们只记录声波在等距点的高度：



给声波采样。

这被称为**采样**（sampling）。我们每秒读取数千次，并把声波在该时间点的高度用一个数字记录下来。这基本上就是一个未压缩的 .wav 音频文件。

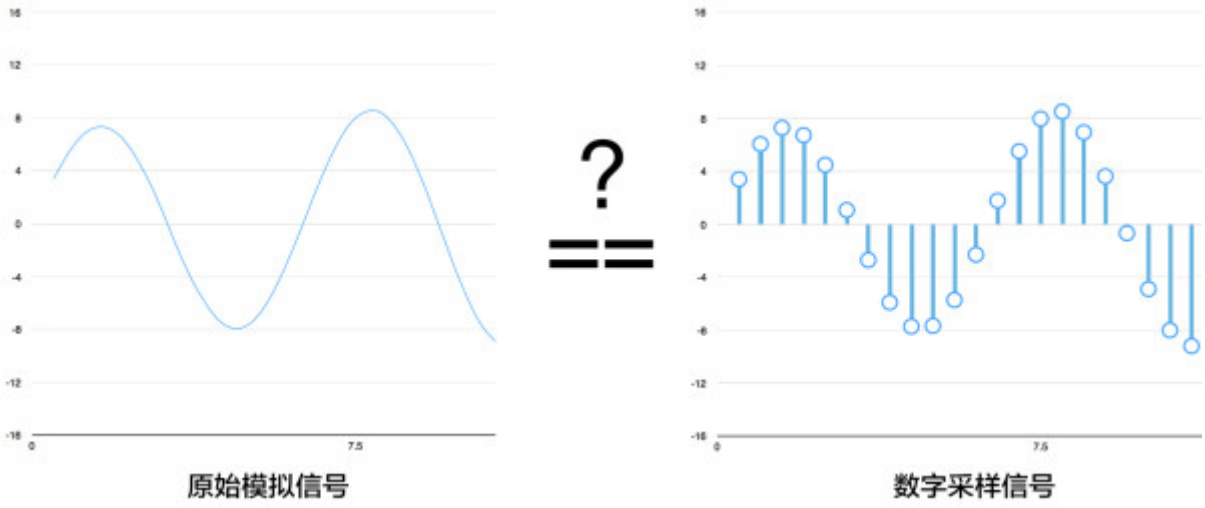
「CD 音质」的音频是以 44.1khz（每秒 44100 个读数）进行采样的。但对于语音识别，16khz（每秒 16000 个采样）的采样率就足以覆盖人类语音的频率范围了。

让我们把「Hello」的声波每秒采样 16000 次。这是前 100 个采样：

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448
, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -
1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -989, -1159, -1307, -1544]
```

每个数字表示声波在一秒钟的 16000分之一处的振幅。

数字采样小助手



数字采样能否完美重现原始声波？那些间距怎么办？

但是，由于采样定理（Nyquist theorem），我们知道我们可以利用数学，从间隔的采样中完美重建原始声波——只要我们的采样频率比期望得到的最高频率快至少两倍就行。

我提这一点，是因为几乎每个人都会犯这个错误，并误认为使用更高的采样率总是会获得更好的音频质量。其实并不是。

预处理我们的采样声音数据

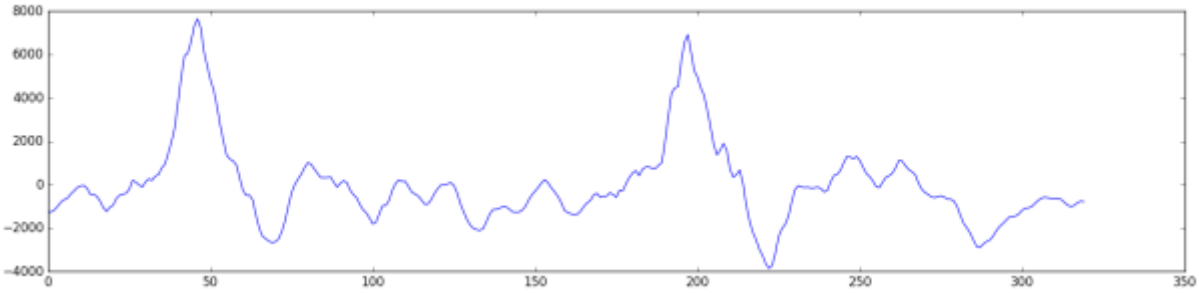
我们现在有一个数列，其中每个数字代表 1/16000 秒的声波振幅。

我们**可以**把这些数字输入到神经网络中，但是试图直接分析这些采样来进行语音识别仍然很困难。相反，我们可以通过对音频数据进行一些预处理来使问题变得更容易。

让我们开始吧，首先将我们的采样音频分成每份 20 毫秒长的音频块。这是我们第一个 20 毫秒的音频（即我们的前 320 个采样）：

```
[-1274, -1252, -1160, -986, -792, -692, -614, -429, -286, -134, -57, -41, -169, -456, -450, -541, -761, -1067, -1231, -1047, -952, -645, -489, -448
, -397, -212, 193, 114, -17, -110, 128, 261, 198, 390, 461, 772, 948, 1451, 1974, 2624, 3793, 4968, 5939, 6057, 6581, 7302, 7640, 7223, 6119, 5461,
 4820, 4353, 3611, 2740, 2004, 1349, 1178, 1085, 901, 301, -262, -499, -488, -707, -1406, -1997, -2377, -2494, -2605, -2675, -2627, -2500, -2148, -
1648, -970, -364, 13, 260, 494, 788, 1011, 938, 717, 507, 323, 324, 325, 350, 103, -113, 64, 176, 93, -249, -461, -606, -909, -1159, -1307, -1544,
-1815, -1725, -1341, -971, -959, -723, -261, 51, 210, 142, 152, -92, -345, -439, -529, -710, -907, -887, -693, -403, -180, -14, -12, 29, 89, -47, -
398, -896, -1262, -1610, -1862, -2021, -2077, -2105, -2023, -1697, -1360, -1150, -1148, -1091, -1013, -1018, -1126, -1255, -1270, -1266, -1174, -10
03, -707, -468, -300, -116, 92, 224, 72, -150, -336, -541, -820, -1178, -1289, -1345, -1385, -1365, -1223, -1004, -839, -734, -481, -396, -580, -52
7, -531, -376, -458, -581, -254, -277, 50, 331, 531, 641, 416, 697, 810, 812, 759, 739, 888, 1008, 1977, 3145, 4219, 4454, 4521, 5691, 6563, 6909,
6117, 5244, 4951, 4462, 4124, 3435, 2671, 1847, 1370, 1591, 1900, 1586, 713, 341, 462, 673, 60, -938, -1664, -2185, -2527, -2967, -3253, -3636, -38
59, -3723, -3134, -2388, -2032, -1831, -1457, -804, -241, -51, -113, -136, -122, -158, -147, -114, -181, -338, -266, 131, 418, 471, 651, 994, 1295,
 1267, 1197, 1291, 1110, 793, 514, 370, 174, -90, -139, 104, 334, 407, 524, 771, 1106, 1087, 878, 703, 591, 471, 91, -199, -357, -454, -561, -605,
-552, -512, -575, -669, -672, -763, -1022, -1435, -1791, -1999, -2242, -2563, -2853, -2893, -2740, -2625, -2556, -2385, -2138, -1936, -1803, -1649,
 -1495, -1460, -1446, -1345, -1177, -1088, -1072, -1003, -856, -719, -621, -585, -613, -634, -638, -636, -683, -819, -946, -1012, -964, -836, -762,
 -788]
```

将这些数字绘制为简单的折线图，我们就得到了这 20 毫秒内原始声波的大致形状：



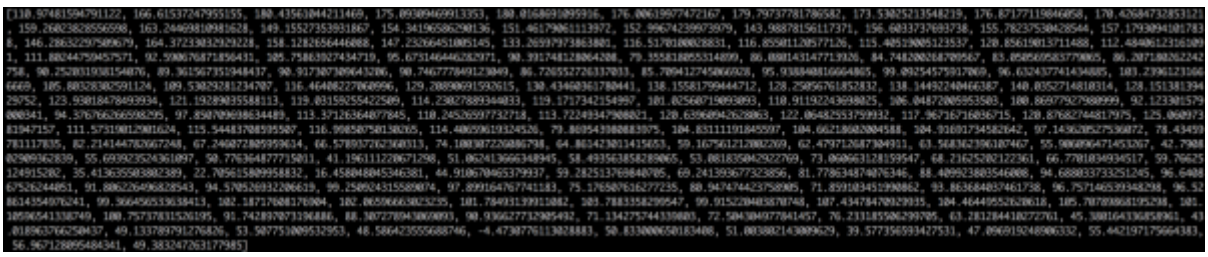
虽然这段录音**只有 1/50 秒的长度**，但即使是这样短暂的录音，也是由不同频率的声音复杂地组合在一起的。其中有一些低音，一些中音，甚至有几处高音。但总的来说，就是这些不同频率的声音混合在一起，才组成了人类的语音。

为了使这个数据更容易被神经网络处理，我们将把这个复杂的声波分解成一个个组成部分。我们将分离低音部分，再分离下一个最低音的部分，以此类推。然后将（从低到高）每个频段（frequency band）中的能量相加，我们就为各个类别的音频片段创建了一个**指纹**（fingerprint）。

想象你有一段某人在钢琴上演奏 C 大调和弦的录音。这个声音是由三个音符组合而成的：C、E 和 G。它们混合在一起组成了一个复杂的声音。我们想把这个复杂的声音分解成单独的音符，以此来分辨 C、E 和 G。这和语音识别是一样的道理。

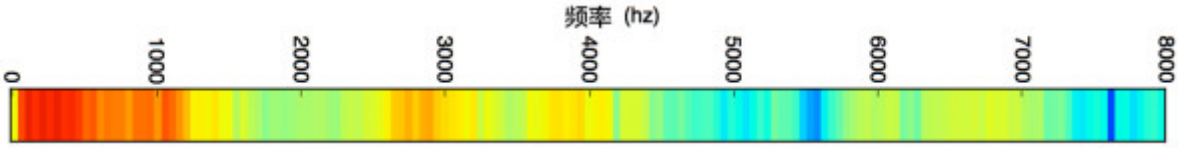
我们需要傅里叶变换（Fourier Transform）来做到这一点。它将复杂的声波分解为简单的声波。一旦我们有了这些单独的声波，我们就将每一份频段所包含的能量加在一起。

最终得到的结果便是从低音（即低音音符）到高音，每个频率范围的重要程度。以每 50hz 为一个频段的话，我们这 20 毫秒的音频所含有的能量从低频到高频就可以表示为下面的列表：



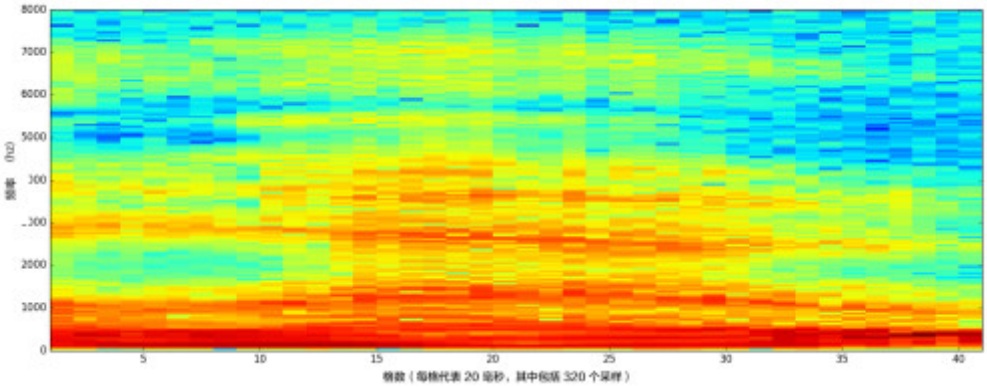
列表中的每个数字表示那份 50Hz 的频段所含的能量。

不过，把它们画成这样的图表会更加清晰：



你可以看到，在我们的 20 毫秒声音片段中有很多低频能量，然而在更高的频率中并没有太多的能量。这是典型「男性」的声音。

如果我们对每 20 毫秒的音频块重复这个过程，我们最终会得到一个频谱图（每一列从左到右都是一个 20 毫秒的块）：

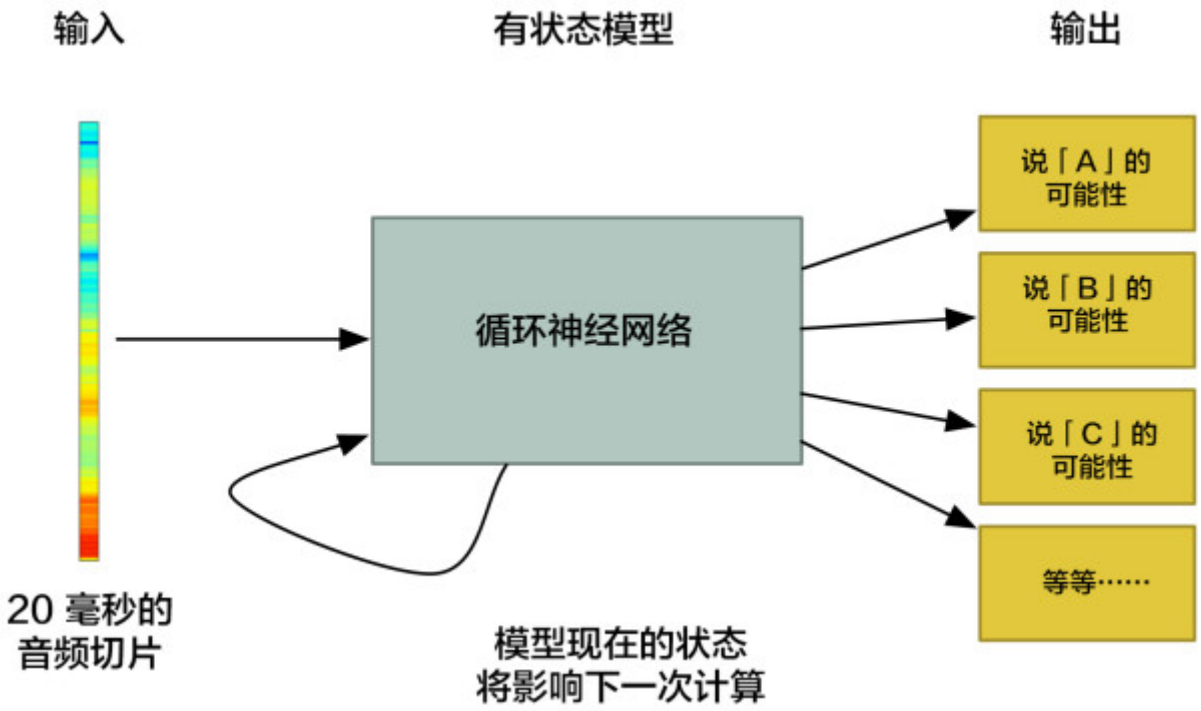


「hello」声音剪辑的完整声谱。

频谱图很酷，因为你可以在音频数据中实实在在地**看到**音符和其他音高模式。对于神经网络来说，相比于原始声波，从这种数据中寻找规律要容易得多。因此，这就是我们将要实际输入到神经网络中去的数据表示方式。

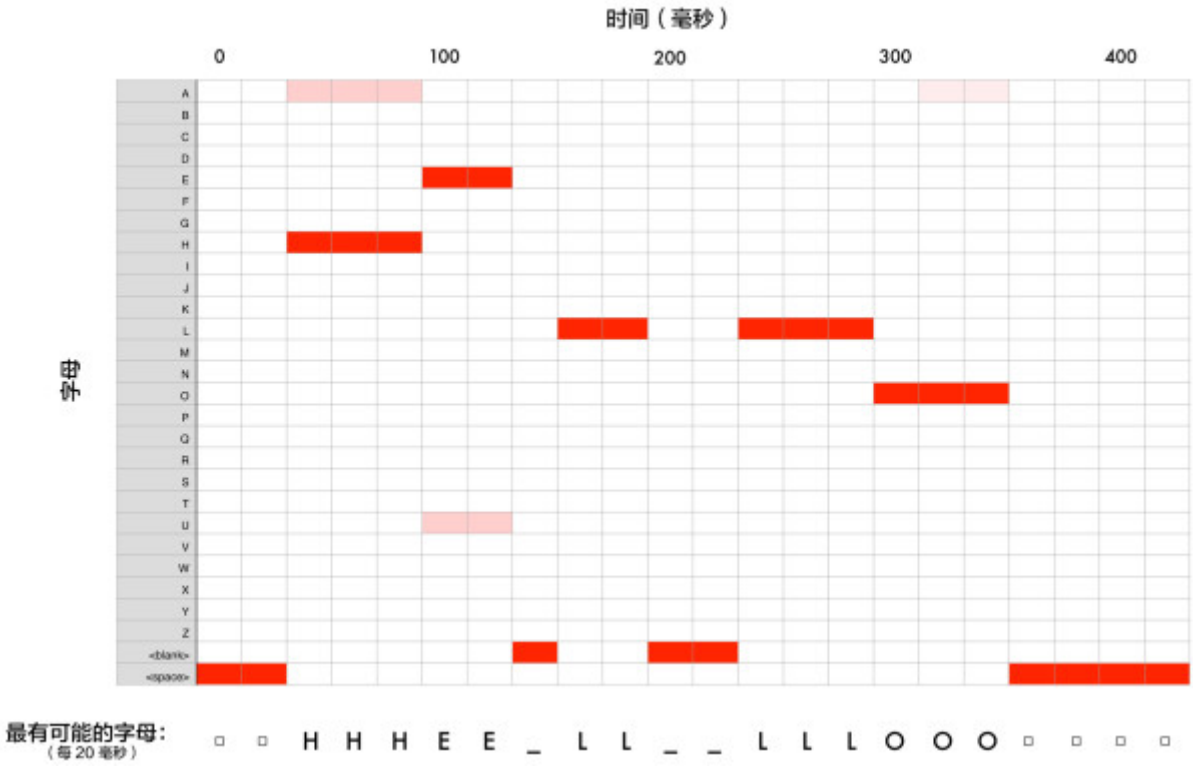
从短声音识别字符

现在我们有了格式易于处理的音频，我们将把它输入到深度神经网络中去。神经网络的输入将会是 20 毫秒的音频块。对于每个小的音频切片（audio slice），神经网络都将尝试找出当前正在说的声音所对应的**字母**。



我们将使用一个循环神经网络——即一个拥有记忆，能影响未来预测的神经网络。这是因为它预测的每个字母都应该能够影响它对下一个字母的预测。例如，如果我们到目前为止已经说了「HEL」，那么很有可能我们接下来会说「LO」来完成「Hello」。我们不太可能会说「XYZ」之类根本读不出来的东西。因此，具有先前预测的记忆有助于神经网络对未来进行更准确的预测。

当通过神经网络跑完我们的整个音频剪辑（一次一块）之后，我们将最终得到一份映射（mapping），其中标明了每个音频块和其最有可能对应的字母。这是我说那句「Hello」所对应的映射的大致图案：



我们的神经网络正在预测我说的那个词很有可能是「HHHEE_LL_LLLOOO」。但它同时认为我说的也可能是「HHHUU_LL_LLLOOO」，或者甚至是「AAAUU_LL_LLLOOO」。

我们可以遵循一些步骤来整理这个输出。首先，我们将用单个字符替换任何重复的字符：

- HHHEE_LL_LLLOOO 变为 HE_L_LO
- HHHUU_LL_LLLOOO 变为 HU_L_LO
- AAAUU_LL_LLLOOO 变为 AU_L_LO

然后，我们将删除所有空白：

- HE_L_LO 变为 HELLO
- HU_L_LO 变为 HULLO
- AU_L_LO 变为 AULLO

这让我们得到三种可能的转写——「Hello」、「Hullo」和「Aullo」。如果你大声说出这些词，所有这些声音都类似于「Hello」。因为神经网络每次只预测一个字符，所以它会得出一些**纯粹表示发音**的转写。例如，如果你说「He would not go」，它可能会给出一个「He wud net go」的转写。

解决问题的诀窍是将这些基于发音的预测与基于书面文本（书籍、新闻文章等）大数据库的

在我们可能的转写「Hello」、「Hullo」和「Aullo」中，显然「Hello」将更频繁地出现在文本数据库中（更不用说在我们原始的基于音频的训练数据中了），因此它可能就是正解。所以我们会选择「Hello」作为我们的最终结果，而不是其他的转写。搞定！

等一下！

你可能会想「但是如果有人说 Hullo 」怎么办？这个词的确存在。也许「Hello」是错误的转写！

「Hullo！Who dis？」

当然可能有人实际上说的是「Hullo」而不是「Hello」。但是这样的语音识别系统（基于美国英语训练）基本上不会产生「Hullo」这样的转写结果。用户说「Hullo」，它总是会认为你在说「Hello」，无论你发「U」的声音有多重。

试试看！如果你的手机被设置为美式英语，尝试让你的手机助手识别单词「Hullo」。这不行！它掀桌子不干了(ノ□)ノ ┻┻！它总是会理解为「Hello」。

不识别「Hullo」是一个合理的行为，但有时你会碰到令人讨厌的情况：你的手机就是不能理解你说的有效的语句。这就是为什么这些语音识别模型总是处于再训练状态的原因。它们

我能建立自己的语音识别系统吗？

机器学习最酷炫的事情之一就是它有时看起来十分简单。你得到一堆数据，把它输入到机器学习算法当中去，然后就能神奇地得到一个运行在你游戏本显卡上的世界级 AI 系统……**对吧**？

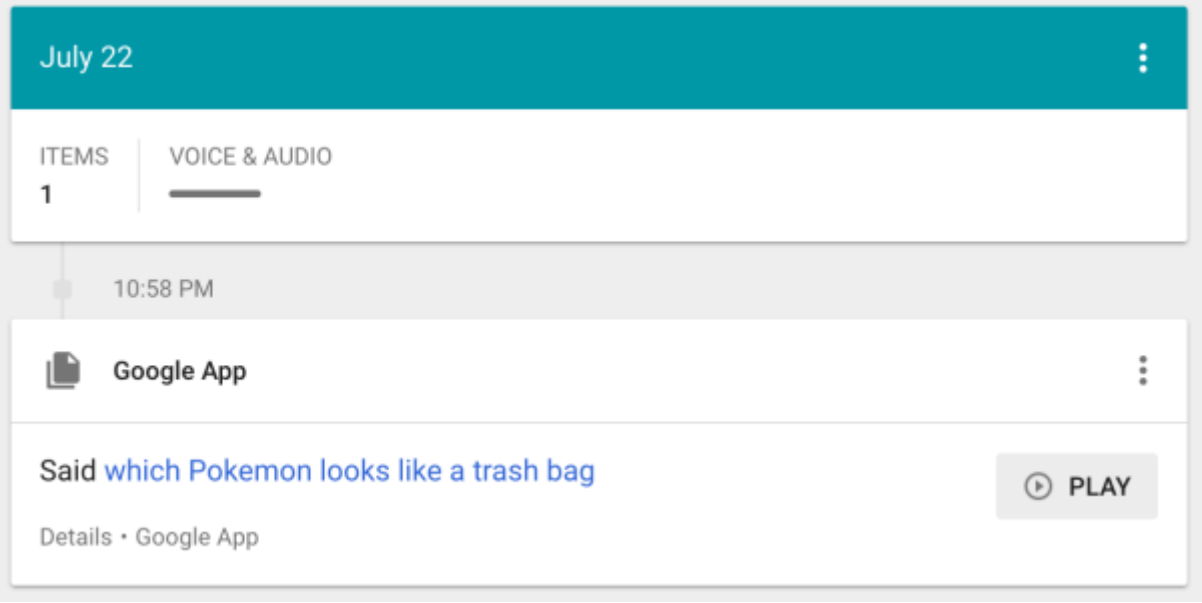
这在某些情况下是真实的，但对于语音识别并不成立。语音识别是一个困难的问题。你得克服几乎无穷无尽的挑战：劣质麦克风、背景噪音、混响和回声、口音差异等等。你的训练数据需要囊括这所有的一切，才能确保神经网络可以应对它们。

这里有另外一个例子：你知不知道，当你在一个嘈杂的房间里说话时，你会不自觉地提高你的音调，来盖过噪音。人类在什么情况下都可以理解你，但神经网络需要训练才能处理这种特殊情况。所以你需要人们在噪音中大声讲话的训练数据！

要构建一个能在 Siri、Google Now! 或 Alexa 等平台上运行的语音识别系统，你将需要**大量**的训练数据 。如果你不雇上数百人为你录制的话，它需要的训练数据比你自己能够获得的数据要多得多。由于用户对低质量语音识别系统的容忍度很低，因此你不能吝啬。没有人想要一个只有八成时间有效的语音识别系统。

对于像谷歌或亚马逊这样的公司，在现实生活中记录的成千上万小时的人声语音就是**黄金**。这就是将他们世界级语音识别系统与你自己的系统拉开差距的地方。让你免费使用 Google Now! *或 Siri，或是只要 50 美元购买 Alexa 而没有订阅费的意义就是：***让你尽可能多地使用它们**。你对这些系统所说的每一句话都会被**永远记录**下来，并用作未来版本语音识别算法的训练数据。这才是他们的真实目的！

不相信我？如果你有一部安装了 Google Now! 的 Android 手机，请点击这里收听你自己对它说过的每一句话：



你可以通过 Alexa 在 Amazon 上找到相同的东西。然而，不幸的是，苹果并不让你访问你的 Siri 语音数据。

因此，如果你正在寻找一个创业的想法，我不建议你尝试建立自己的语音识别系统来与 Google 竞争。相反，你应该想个办法，让人们把自己讲了几个小时的录音交给你。这种数据可以是你的产品。

路在远方……

这个用来处理不同长度音频的算法被称为连接时序分类（Connectionist Temporal Classification）或 CTC。你可以阅读这篇 2006 年文章。

百度的亚当·科茨（Adam Coates）在湾区深度学习学校做了关于「深度学习语音识别」的精彩演讲。你可以在 YouTube 上观看这段视频。强烈推荐。

作者：Adam Geitgey

原文：https://medium.com/@ageitgey/machine-learning-is-fun-part–6-how-to-do-speech-recognition-with-deep-learning–28293c162f7a#.42h1r63ev

译文：https://zhuanlan.zhihu.com/p/24703268

译者：巡洋舰科技——赵 95

亚当·盖特吉

软件工程师，Groupon 工程部总监，带领团队维护 Groupon 网页端。热爱计算机与机器学习。推特帐号 @ageitgey 。

评论

镜子君

10 个月前 下午9:13

请问这个系列能够一起下载mobi格式么？

回复

发表评论

电子邮件地址不会被公开。 必填项已用*标注

评论

姓名 *

电子邮件 *

站点

一个图灵小测试 *

订阅 往期 登录

× 二 =

发表评论

下一篇

卷首语