用深度学习进行语音识别简介

语音识别正在“入侵”我们的生活。它内置在我们的手机，游戏主机和智能手表里。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 - 一个能够让你订购比萨，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（2016年圣诞）这个假期太受欢迎了，以至于Amazon似乎都没货了！

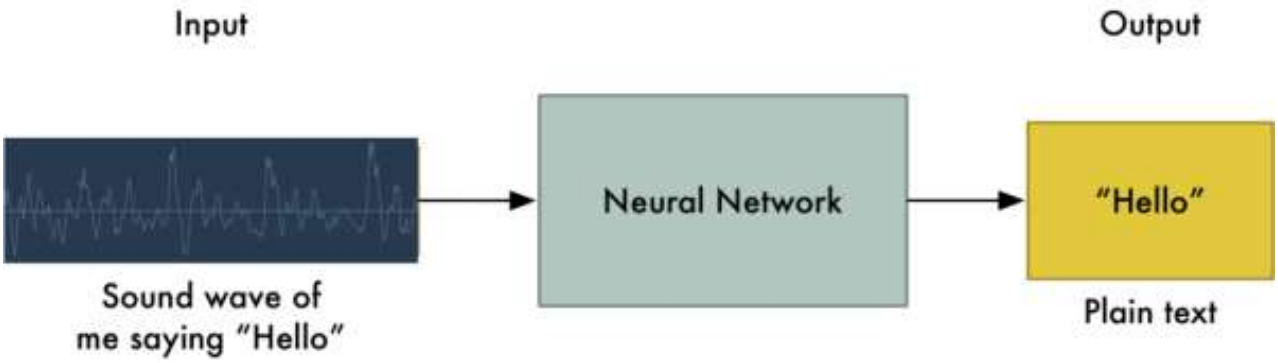
然而语音识别已经出现了几十年了，为何它才刚刚成为主流呢？原因是，深度学习，终于让语音识别，能够在非严格可控的环境下也能准确的识别。

吴恩达教授(百度首席科学家，人工智能和机器学习领域国际上最权威的学者之一，也是在线教育平台Coursera的联合创始人)长期以来预测，随着语音识别从95%精确度上升到99%，它将成为我们与计算机交互的主要方式。这个想法是基于，4%的精确度实际就是“太不靠谱”与“极度实用”之间的差别。感谢深度学习，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到神经网络中，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，但目前我们还没有完全做到（至少在我写这篇文章的时候没做到——我打赌，在未来的几年我们可以做到）。

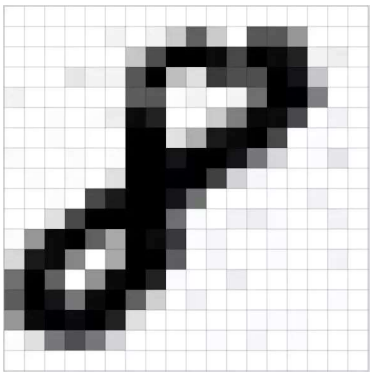
最大的问题是言速不同。一个人可能很快的说“hello!”而另一个人可能会非常缓慢说“heeeellllllllllllooooo!”。这产生了一个更长的声音文件和更多的数据。这两个声音文件都应该被识别为完全相同的文本“hello！”而事实证明，把各种长度的音频文件自动对齐到一个固定长度的文本是很难的一件事情。

为了解决这个问题，我们必须使用一些特殊的技巧和一些除了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“位（Bit）”

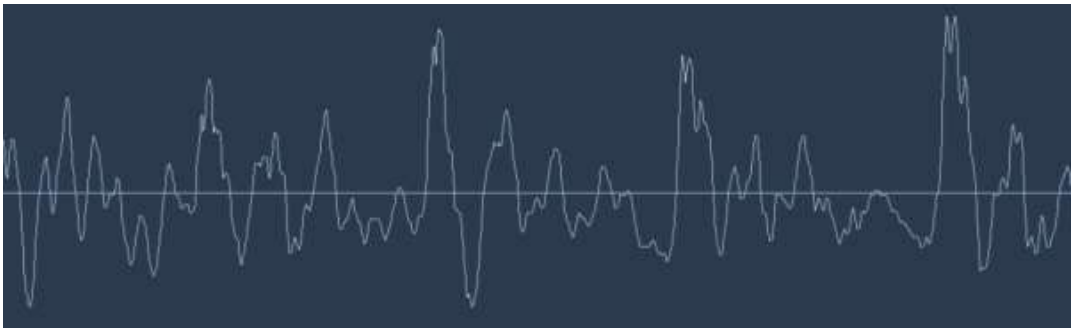
语音识别的第一步是很显而易见的——我们需要将声波输入到计算机当中。

在第3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：



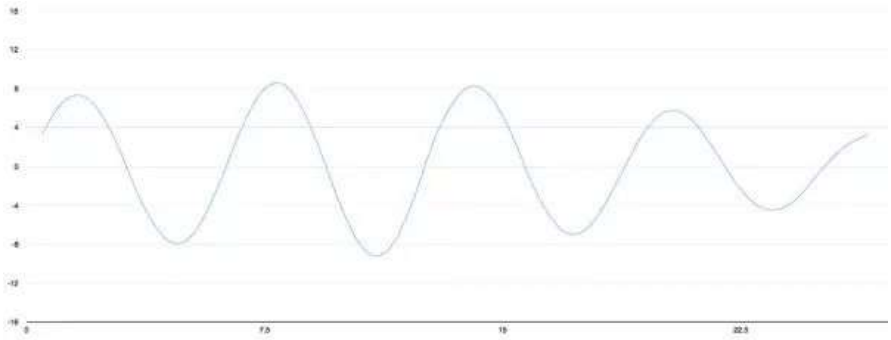
图像只是图片中每个像素深度的数字编码序列

但声音是作为***波(Waves)***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”这个声音片段我们例子：



我说“hello”的波形-声波是一维的。（译者注：其实是二维的，有时间，还有振幅）在每个时刻，基于波的高度，它们有一个值(译者注：叫做振幅)。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44,100个读数）进行采样的。但对于语音识别，16khz（每秒16,000个采样）的采样率足以覆盖人类语音的频率范围。

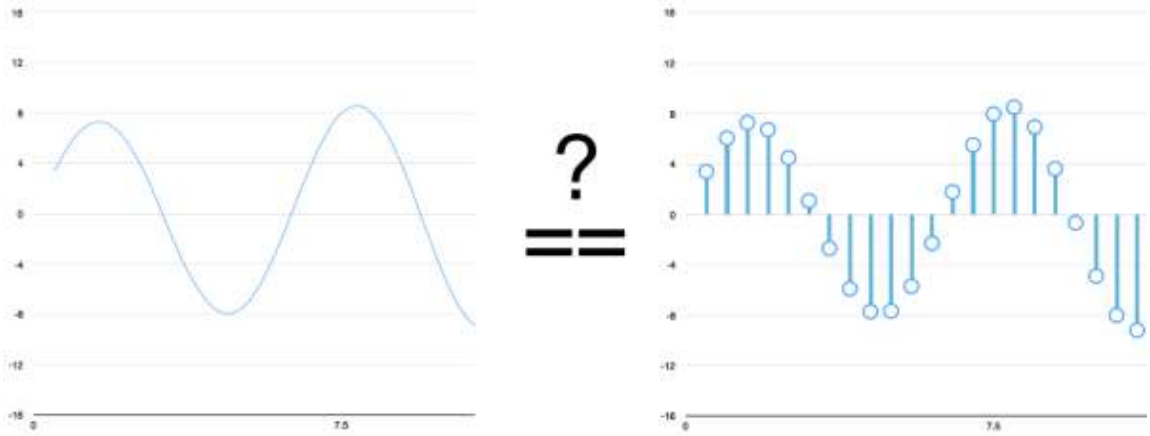
让我们把“Hello”的声波每秒采样16,000次。这是前100个采样：

```
[-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, -1486, -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]
```

每个数字表示在一秒钟的16000分之一处的声波的振幅

数字采样小助手

你可能认为采样只是对原始声波进行粗略近似估计，因为它只是间歇性的读取。我们的读数之间有间距，所以我们会丢失数据，对吗?



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的重建原始模拟声波——只要以我们希望得到的最高频率的两倍来采样就可以。我提到这一点，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能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

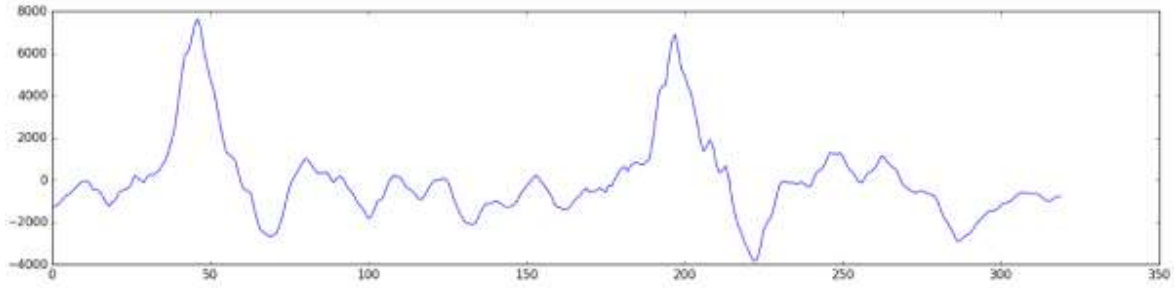
我们现在有一个数列，其中每个数字代表16000分之一秒的声波振幅。

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旧是困难的。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组为20毫秒长的块儿。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折线图，图中给出了20毫秒时间内原始声波的粗略估计：



虽然这段录音只有*50分之一*秒的长度，但即使这样短暂的时长也是由不同频率的声音复杂的组合在一起的。一些低音，中音，甚至高音混在一起。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件部分。我们将一步步分离低音部分，下一个最低音部分，以此类推。然后通过将（从低到高）每个频带中的能量相加，我们就为各个类别（音调）的音频片段创建了一个*指纹fingerprint*。

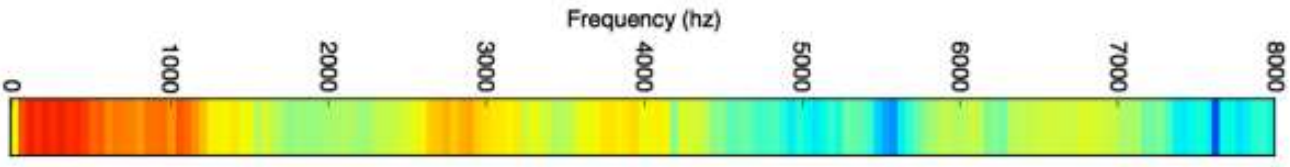
想象你有一段某人在钢琴上演奏C大调和弦的录音。这个声音是由三个音符组合而成的 - C，E和G – 他们都混合在一起组成一个复杂的声音。我们想把这个复杂的声音分解成单独的音符，以此来发现它们是C，E和G。这和我们（语音识别）的想法一样。

我们使用被称为*傅里叶变换Fourier Transform*的数学运算来做到这一点。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，我们将每一个包含的能量加在一起。

最终结果是每个频率范围的重要程度，从低音（即低音音符）到高音。下面的每个数字表示我们的20毫秒音频剪辑中每个50Hz频带中有多少能量：

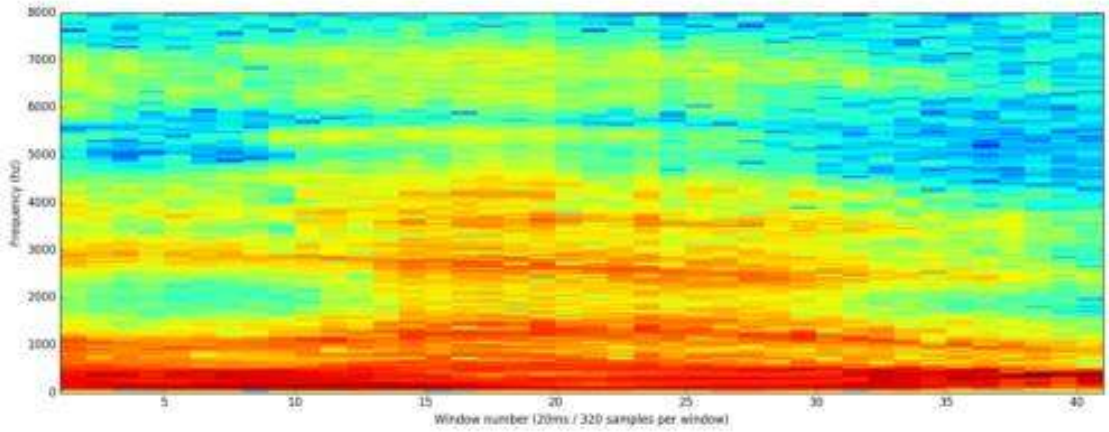
列表中的每个数字表示在50Hz频带中有多少能量

但是当你绘制一个图表时，你很容易看到这些能量：



你可以看到，我们的20毫秒声音片段中有很多低频率能量，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20ms的块）：

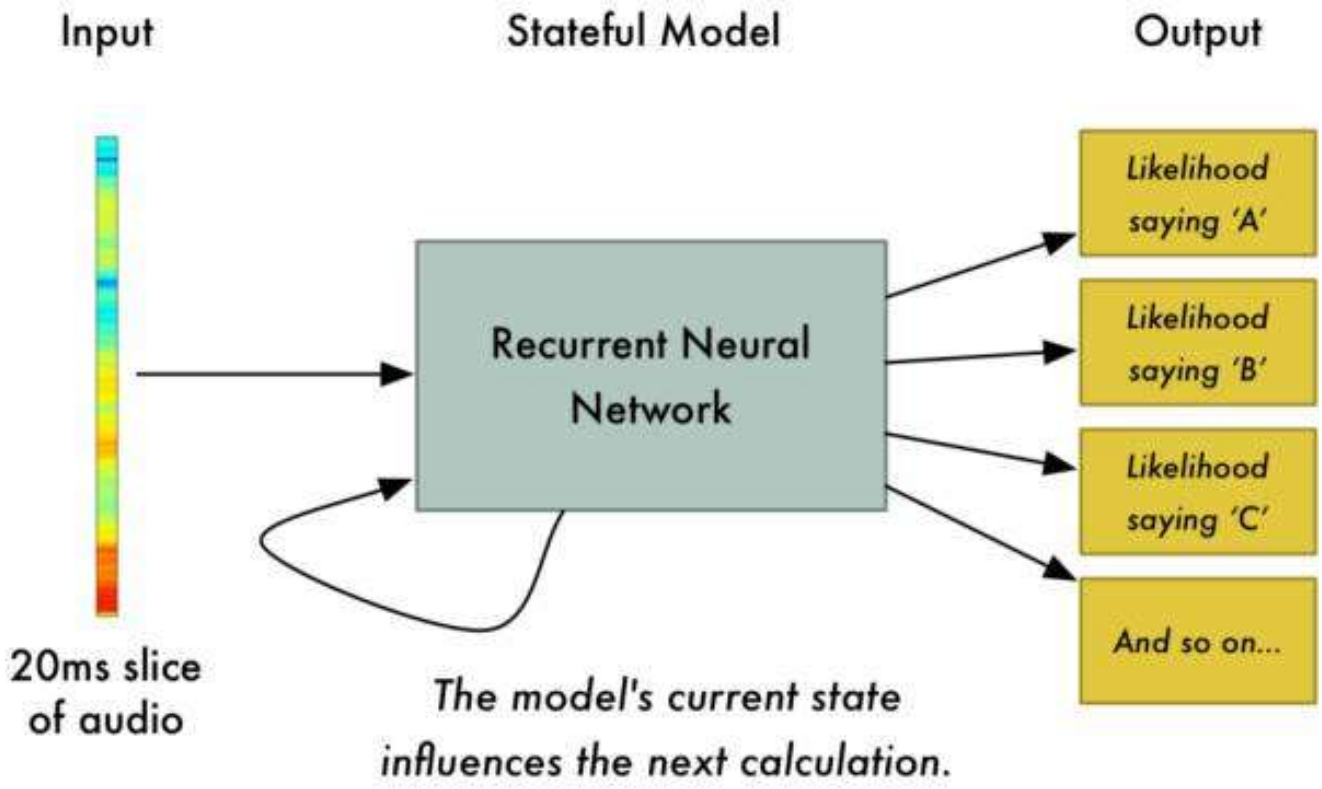


“hello”声音剪辑的完整谱图

频谱图很酷，因为你可以从音频数据中实际*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，它可以更加容易地从这种数据中找到规律。因此，这就是我们将实际输入到神经网络的数据表示方式。

从短声音识别字符

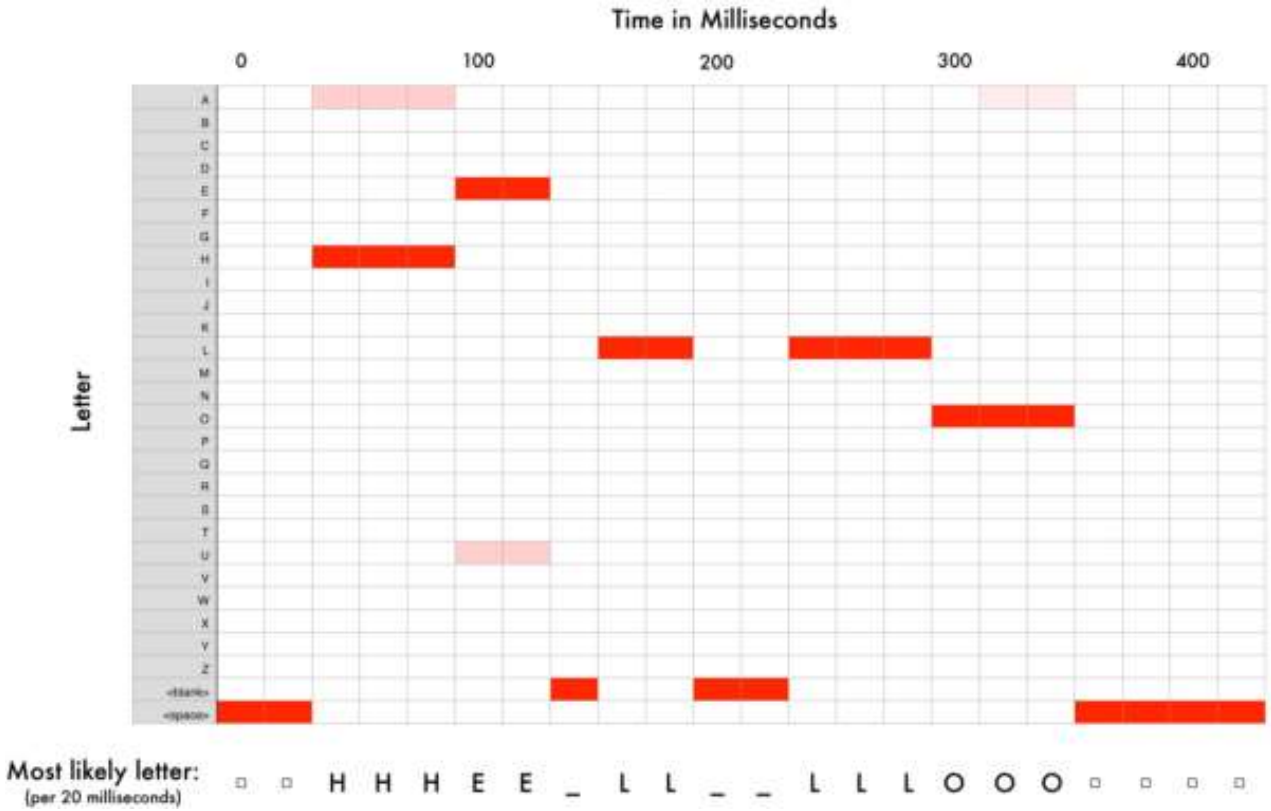
现在我们有了一个易于处理的格式的音频，我们将把它输入到深度神经网络中去。神经网络的输入将会是20毫秒的音频块。对于每个小的音频切片(Audio Slice)，它将试图找出当前正在说的声音对应的*字母（letter）*。



我们将使用一个循环神经网络 - 即一个拥有记忆来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下一个字母的预测可能性。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。

当我们通过神经网络运行我们的整个音频剪辑（一次一块）之后，我们将最终得到每个音频块和其最可能被说出的那个字母的一个映射（mapping）。这是一个看起来说”Hello”的映射：



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处：

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录 - “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”。因为它每次只预测一个字符，神经网络会得出一些试

探性的转录。例如，如果你说“He would not go”，它可能会给一个可能 “He wud net go” 的转录。

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）大数据库的可能性得分相结合。你抛弃掉最不可能的转录，而保留住最现实的转录。

在我们可能的转录“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），因此它可能是正确的。所以我们会选择“Hello” 而不是其他作为我们的最后的转录。完成!

等一下!

你可能会想“但是如果有人说Hullo”怎么办？这是一个有效的词。也许“Hello”是错误的转录!

“Hullo！Who dis?

当然可能有人实际上说“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”作为转录。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 ﾐ┻┻！它总是会理解为“Hello”。

不识别“Hullo”是一个合理的行为，但有时你会发现令人讨厌的情况: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被更多的数据训练来修复这些少数情况。

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，然后就能神奇的得到一个运行在你的游戏笔记本电脑的显卡上的世界级AI系统...*对吧?*

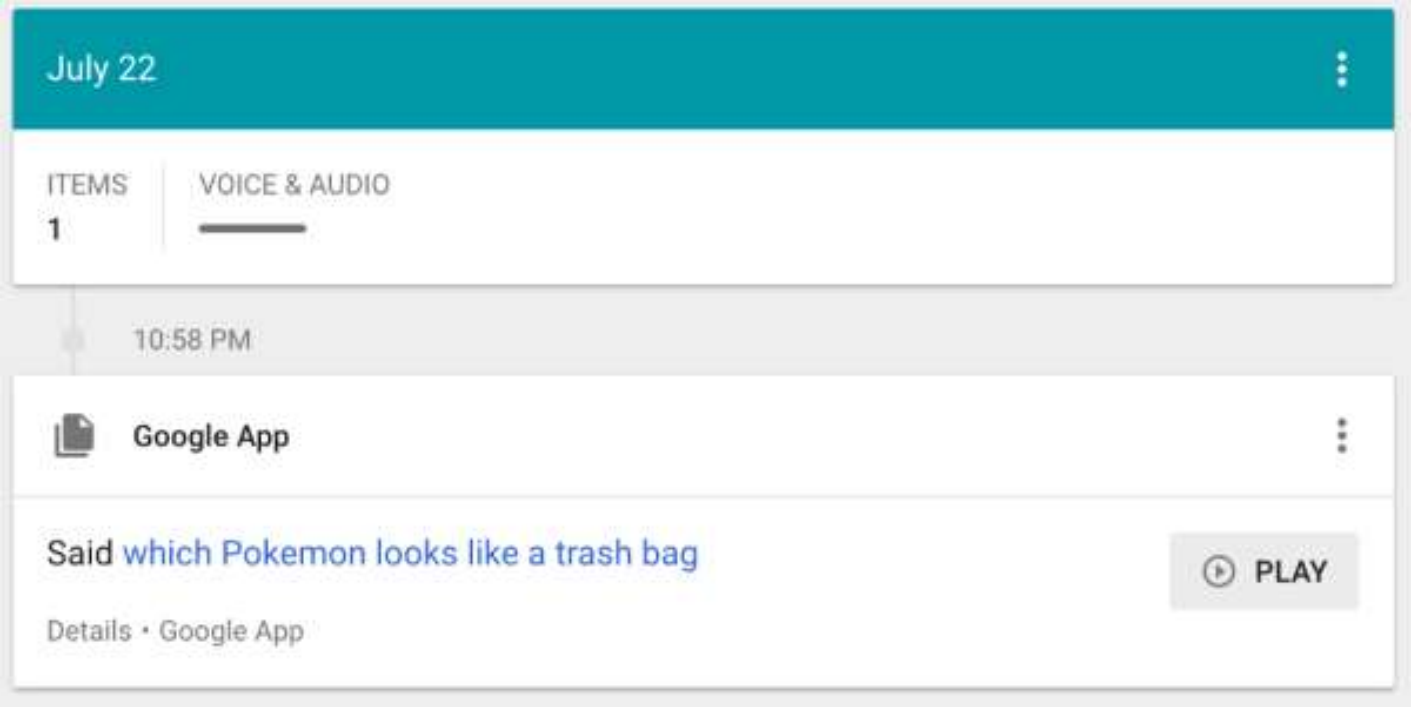
这在某些情况下是真实的，但对于语音识别并不成立。语音识别是一个困难的问题。你必须克服几乎无限的挑战：质量差的麦克风，背景噪音，混响和回声，口音变化，还有很多很多。所有这些问题都需要存在于你的训练数据中，以确保神经网络可以应对它们。

这里有另外一个例子：你知不知道，当你在一个充满噪音的房间里说话时，你不自觉地提高你的音调，以便能够盖过噪音。人类在什么情况下都可以理解你，但神经网络需要训练来处理这种特殊情况。所以你需要人们对着噪音大声说话的训练数据!

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 -如果你不雇佣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80%的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数十万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：***让你尽可能多的使用他们***。你对这些系统所说的每一句话都会***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，你应该找出一种能让人们把他们说几个小时话的录音给予你的方法。这种数据可以是你的产品。

作者：九五要当学霸
链接：https://zhuanlan.zhihu.com/p/24703268
来源：知乎

原文：Adam Geitgey

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

翻译：巡洋舰科技——赵95