语音学与深度学习

张志毅
南方科技大学

2023 年 12 月 27 日

摘要

在语音识别技术的发展中，深度学习尤其是卷积神经网络（CNN）的应用，为提高识别精度和效率提供了新的可能性。本研究专注于利用卷积神经网络进行特定单词集的识别训练和测试，旨在探索 CNN 在精确识别单词方面的能力。该系统首先接受大量包含 32 个不同单词的语音样本进行学习和训练。系统通过这些样本学习识别每个单词的特征。完成训练后，系统被测试其对新语音样本的识别能力，判断这些新样本中包含的单词。

研究结果显示，CNN 模型能够有效地从语音样本中学习并识别特定单词。尤其在处理含有不同口音和语速变化的语音样本时，模型表现出较高的准确性和适应性。此外，实验也展示了 CNN 在处理语音信号时的高效性和稳定性。

本研究的成果不仅展示了卷积神经网络在特定单词识别任务中的应用潜力，也为进一步开发高效且准确的语音识别系统提供了有价值的参考。随着深度学习技术的不断进步，这一方法预计将在语音交互系统中扮演更加重要的角色。

关键词： 深度学习，语音识别，卷积神经网络，语音处理

1 引言

语音识别技术作为人机交互的关键环节，在过去几十年中经历了显著的进步。特别是深度学习技术，尤其是卷积神经网络（CNN），为提高语音识别的准确性和效率带来了新的突破。本研究聚焦于使用 CNN 对特定单词集进行识别的能力，探索其在语音识别中的应用潜力。

当前的挑战包括如何有效处理含有不同口音、语速以及背景噪声的语音。本文提出一种基于 CNN 的方法，通过学习 32 个特定单词的语音样本，来测试模型在新语音样本中识别这些单词的能力。

本研究的目的是验证 CNN 在单词识别任务中的性能，尤其是在处理不同语音特征时的准确性和鲁棒性。这对于进一步发展高效的语音识别系统以及优化人机交互具有重要意义。

2 语音识别综述–概念部分

首先我们来看一张图片。

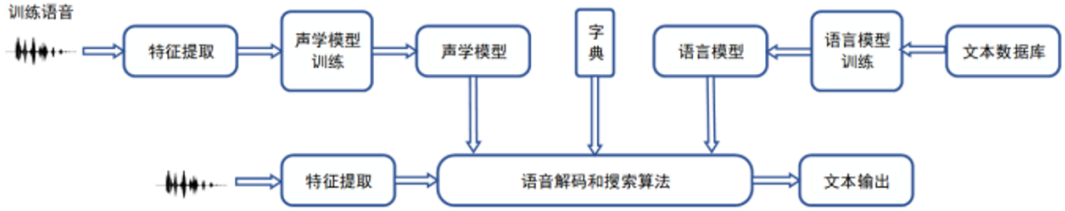

图 1: 自动语音识别流程

这张图展示了语音识别系统的处理流程。首先，系统接收到原始的声音信号，然后经过信号处理阶段，其中包括噪声降低和特征提取，以便从声音中提取有用信息。提取的特征随后被输入到声学模型，该模型通常基于深度学习技术来识别语音中的模式。与此同时，语言模型负责理解单词间的关系和语法规则。最后，解码算法结合声学模型和语言模型的输出，生成最终的语音识别结果，将语音转换为文本形式。整个流程体现了从声音信号到文本输出的转换机制，是现代语音识别系统的核心。

3 卷积神经网络

卷积神经网络（CNN）是一种深度学习架构，特别适用于处理具有网格结构的数据，如图像（二维网格）和音频信号（一维网格）。CNN 通过模拟生物视觉系统的处理方式，有效地进行特征提取和模式识别。

卷积层是 CNN 中的核心组件，它的数学表达为离散卷积操作：

$$(f * g)[n] = \sum_{m=-M}^{M} f[m] \cdot g[n-m] \tag{1}$$

其中，f 表示输入数据，g 表示卷积核，n 和 m 是空间或时间的离散索引，M 是卷积核的大小。

激活函数如 ReLU（Rectified Linear Unit）引入了非线性，其定义为：

$$ReLU(x) = \max(0, x) \tag{2}$$

这使得 CNN 可以捕捉输入数据中的复杂模式。

池化层通常跟随卷积层，用于降低特征的空间维度，提高计算效率。最大池化操作可表示为：

$$h_{ij} = \max_{u,v \in W_{ij}} f(u,v) \tag{3}$$

其中，h_{ij} 是池化层的输出，W_{ij} 是覆盖窗口，$f(u,v)$ 是该窗口内的输入特征。

经过若干卷积和池化层的处理，特征被送入全连接层进行最终的分类。在训练过程中，采用反向传播算法更新网络权重。损失函数关于权重的梯度计算如下：

$$\frac{\partial L}{\partial w_i} = \frac{\partial L}{\partial y} \cdot \frac{\partial y}{\partial w_i} \tag{4}$$

其中，L 是损失函数，y 是网络的输出，w_i 是网络权重。使用梯度下降法，CNN 可以学习到识别数据中的复杂模式。

4 神经网络在语音识别中的应用

接下来使用代码来展示这一部分。

4.1 数据集介绍以及数据读取

Speech Commands 数据集包含超过 105,000 个音频文件，每个文件记录了人们说出 8 个不同单词中的一个，如 “down”，“go”，“left”，“no”，“right”，“stop”，“up” 和 “yes”。这些音频片段都很短，大多在一秒或更短，并且按照每个语音命令存储在八个不同的文件夹中。

```
DATASET_PATH = 'data/mini_speech_commands'

data_dir = pathlib.Path(DATASET_PATH)
if not data_dir.exists():
  tf.keras.utils.get_file(
      'mini_speech_commands.zip',
      origin="http://storage.googleapis.com/download.tensorflow.org/data/mini_speech_commands.zip",
      extract=True,
      cache_dir='.', cache_subdir='data')
```

图 2: 加载数据集

4.2 数据预处理

音频剪辑为 16HZ，将所有音频长度设置为 16000，即 1s，较长与正常人说一个单词的时长，便于批量处理。

```
train_ds, val_ds = tf.keras.utils.audio_dataset_from_directory(
    directory=data_dir,
    batch_size=64,
    validation_split=0.2,
    seed=0,
    output_sequence_length=16000,
    subset='both')

label_names = np.array(train_ds.class_names)
print()
print("label names:", label_names)
```

图 3: 音频剪切

4.3 波形图转频谱图

语音识别中一个关键步骤是将音频信号从时域的波形图转换为频域的频谱图。这一转换通常通过快速傅里叶变换（FFT）实现。FFT 是一种高效的算法，能将复杂的波形信号分解成不同频率的成分。

频谱图在语音识别中的应用带来了多方面的好处，包括：

- 揭示隐藏特征：频谱图能够展示音频信号的频率成分，揭示时域波形图中难以辨识的细节，如音高和节奏。
- 改善模式识别：频谱图使得神经网络更容易识别和学习语音信号中的模式，因为它将复杂的时域信号转换为更易于处理的频域表示。
- 降低噪声影响：频谱图中可以更有效地识别和过滤掉背景噪声，从而提高语音识别的准确性。
- 适应性强：频谱图对不同说话者的声音特征和不同环境下的录音具有较好的适应性，有助于提升系统的泛化能力。

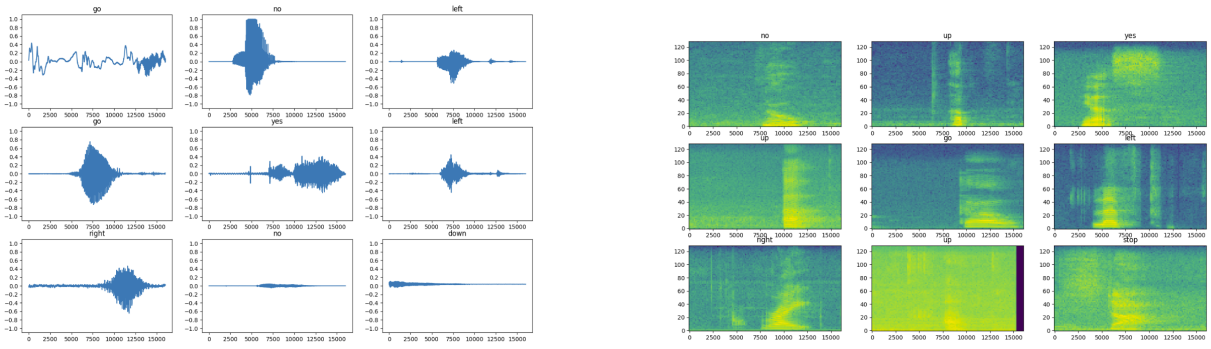

图 4: 左图为波形图，右图为频谱图

4.4 卷积神经网络对输入频谱图进行处理

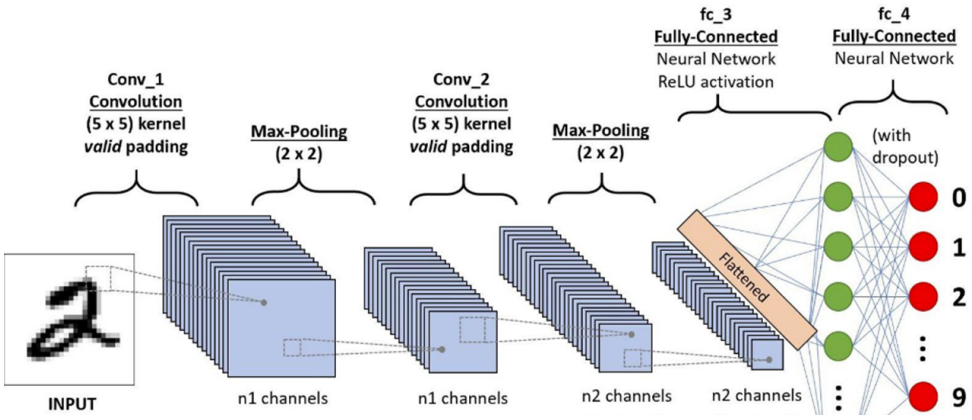

图 5: 卷积神经网络

总体上可以分为向前传播和反向传播两个过程。

- 向前传播：可以想象成解谜游戏。网络接收输入（如图像），然后逐层使 用不同的过滤器（卷积核）来“解读”这个图像。。
- 反向传播：可以比作是一个“学习从错误中改进”的过程。这类似于你在解谜时回溯，了解哪一部分错了，然后修正你的策略。

```
metrics = history.history
plt.figure(figsize=(16,6))
plt.subplot(1,2,1)
plt.plot(history.epoch, metrics['loss'], metrics['val_loss'])
plt.legend(['loss', 'val_loss'])
plt.ylim([0, max(plt.ylim())])
plt.xlabel('Epoch')
plt.ylabel('Loss [CrossEntropy]')

plt.subplot(1,2,2)
plt.plot(history.epoch, 100*np.array(metrics['accuracy']), 100*np.array(metrics['val_accuracy']))
plt.legend(['accuracy', 'val_accuracy'])
plt.ylim([0, 100])
plt.xlabel('Epoch')
plt.ylabel('Accuracy [%]')
```

图 6: 卷积神经网络代码

4.5 损失函数和准确率

以上步骤完成之后，就可以对新的单词音频进行预测，我们看一下 loss 函数和 accuracy。

以下便是图像

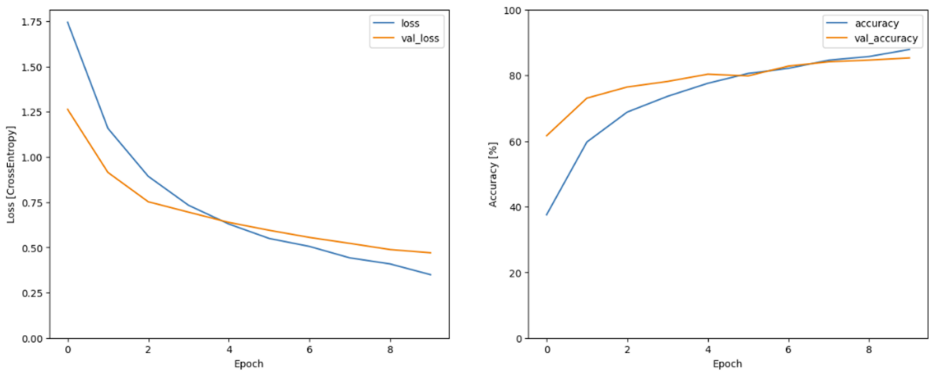

图 7: loss 和 accuracy

4.6 对音频文件运行推理

```
x = data_dir/'no/01bb6a2a_nohash_0.wav'
x = tf.io.read_file(str(x))
x, sample_rate = tf.audio.decode_wav(x, desired_channels=1, desired_samples=16000,)
x = tf.squeeze(x, axis=-1)
waveform = x
x = get_spectrogram(x)
x = x[tf.newaxis,...]

prediction = model(x)
x_labels = ['no', 'yes', 'down', 'go', 'left', 'up', 'right', 'stop']
plt.bar(x_labels, tf.nn.softmax(prediction[0]))
plt.title('No')
plt.show()

display.display(display.Audio(waveform, rate=16000))
```

图 8: 音频推理的代码

这一部分中，我们将探讨如何对音频文件进行有效的数据推理。这一过程涉及将音频文件输入到训练好的神经网络模型中，并解释模型的输出以获取有关音频内容。

下图是一个分类的结果，我们可以看到并不是只有一个推断，，但是我们可以看到 go 这个单词占了绝大多数概率，所以这便是预测的结果。

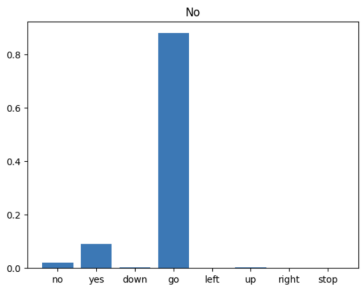

图 9: 音频推理结果

5 结论

在这篇报告中，我们探讨了深度学习在语音识别方面的应用，特别关注了卷积神经网络和循环神经网络的角色。我们还讨论了将波形信号转换为频谱图的过程及其优势，并通过形象的比喻解释了神经网络中的前向和反向传播。此外，还涉及了对音频文件执行推理的步骤和面临的挑战。总体来说，这项研究强调了深度学习在处理和理解语音数据方面的重要性，对于相关领域的研究和实际应用提供了有价值的见解。