ESPnet：端到端语音处理工具

ESPnet: End-to-End Speech Processing Toolkit

Shinji Watanabe1, Takaaki Hori2, Shigeki Karita3, Tomoki Hayashi4, Jiro Nishitoba5, Yuya Unno6,

Nelson Enrique Yalta Soplin7, Jahn Heymann8, Matthew Wiesner1, Nanxin Chen1, Adithya

Renduchintala1, Tsubasa Ochiai9,

1Johns Hopkins University, 2Mitsubishi Electric Research Laboratories, 3NTT Communication

Science Laboratories, 4Nagoya University, 5Retrieva, Inc., 6Preferred Networks, Inc., 7Waseda

University, 8Paderborn University, 9Doshisha University

shinjiw@jhu.edu

**摘要**

This paper introduces a new open source platform for end-to-end speech processing named ESPnet. ESPnet mainly focuses on end-to-end automatic speech recognition (ASR), and adopts widely-used dynamic neural network toolkits, Chainer and PyTorch, as a main deep learning engine. ESPnet also follows the Kaldi ASR toolkit style for data processing, feature extraction/format, and recipes to provide a complete setup for speech recognition and other speech processing experiments. This paper explains a major architecture of this software platform, several important functionalities, which differentiate ESPnet from other open source ASR toolkits, and experimental results with major ASR benchmarks.

Index Terms: speech recognition, open source software, end-to-end, dynamical neural network, Kaldi

本文介绍了一个名为ESPnet的新型开源语音处理平台。ESPnet主要用于解决端到端自动语音识别（ASR）任务。它使用目前广泛使用的动态神经网络工具——Chainer和PyTorch，作为深度学习引擎。ESPnet使用Kaldi（一个ASR工具箱）格式的数据处理、特征提取/格式和处理流程，提供完整的语音识别流程和其他语音处理实验流程。本文介绍了该软件平台的主要结构和几个让ESPnet区别于其他ASR工具的重要功能模块。此外，还给出与主流ASR基准对比的实验结果。

**关键词**：语音识别，开源软件，端到端，动态神经网络，Kaldi

2018.03.30 arxiv

**1 前沿**

在语音处理领域进行了大量的研究和开发之后，语音识别（ASR）已经是一项成熟的技术。特别的，它不仅被一些流行产品比如谷歌语音助手、亚马逊Alexa和苹果Siri所驱动，还被包括像Kaldi，HTK，Sphinx，Julius和RASR等开源项目所推动。这些开源工具包提供特征提取、基于隐马尔可夫模型（HMM）、高斯混合模型和深度神经网络（DNN）的声学建模以及解码功能，这使我们能够使用全套最先进的ASR研究和开发成果。

本文介绍了一个新的开源工具ESPnet（End-to-End speech processing toolkit），旨在为ASR和其他语音处理提供一个使用神经网络的端到端平台。与上述基于混合DNN/HMM架构的开源工具不同[7]，ESPnet提供了一个单一的神经网络架构，以端到端的方式进行语音识别。ESPnet采用了广泛使用的动态神经网络工具箱Chainer[8]和PyTorch[9]作为主要的深度学习引擎。ESPnet还遵循Kaldi ASR工具箱[1]的风格进行数据处理、特征提取/格式和流程示例，为语音识别和其他语音处理实验提供完整的教程。

ESPnet充分利用了基于连接主义时间分类（CTC）[10,11,12]和基于attention的编解码网络[13,14,15,16]的两种主要的端到端ASR方案的优点。基于attention的方法利用注意机制来实现声音帧和识别符号之间的对齐，而CTC则利用Markov假设，通过动态规划有效地解决序列问题。ESPnet是采用了CTC/attention混合结构的端到端ASR[17]，有效地利用了这两种结构在训练和解码方面的优势。在训练过程中，我们采用多目标学习框架来提高对不规则对齐的鲁棒性，实现快速收敛。在解码过程中，我们将attention和CTC的分数结合在一个一遍波束搜索算法中进行联合解码，以进一步消除不规则对齐。

除了上述基本架构之外，ESPnet还支持许多端到端的ASR技术，包括融合递归神经网络语言模型（RNNLM）[17]、使用warp CTC库进行快速CTC计算[12]和多种注意方法。通过这些先进的端到端ASR技术，ESPnet还为主流的ASR基准提供了许多方法，包括《华尔街日报》（WSJ）[18]、Librispeech[19]、TED-LIUM[20]、自发日语语料库（CSJ）[21]、AMI[22]、HKUST普通话CTS[23]、VoxForge[24]、CHiME-4/5[25、26]等，ESPnet提供了公开的最先进的端到端ASR设置，旨在加速这一新兴领域的发展。本文描述了它的基本架构、功能和基准测试结果。值得注意的是，在包括HKUST和CSJ在内的多个例子上，与最先进的基于无网格最大互信息训练的DNN/HMM混合结构系统相比，效果相当或更优[27]。

**2 相关研究**

本节主要关注ESPnet与其他公开可用的端到端ASR工具包的比较。我们可以根据CTC和attention架构将工具箱分为两类，如下所示：

•基于CTC：EESEN[11]，Stanford CTC[28]，百度Deepspeech[12]

•基于Attention：Attention-LVCSR[29]，OpenNMT speech to text[30]

注意到大多数端到端ASR工具包都基于CTC，而ESPnet是基于attention的编码器-解码器网络。与LVCSR和OpenNMT相比，ESPnet在语音识别应用时有更多特殊的功能，比如CTC/attention混合结构处理单调注意力，在解码过程中使用RNNLM，以及许多使用kaldi风格的示例，这些都使ESPnet区别于其他工具。

**3 功能**

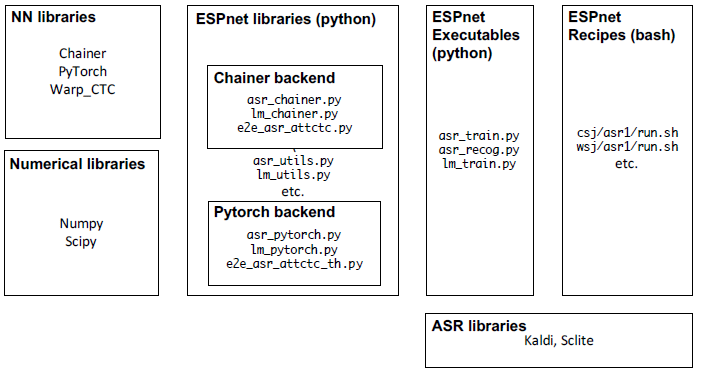


图 1 ESPnet架构

图1显示了ESPnet的体系结构。在ESPnet中，主要的神经网络训练和识别部分用python编写，通过切换后端选项调用Chainer和PyTorch。我们还提供了执行ASR实验的完整方法，这些方法是按照Kaldi方式以bash脚本编写的。以下各节介绍了ESPnet在现有其他工具包中的几个独特功能。

**3.1 Kaldi风格的数据预处理**

ESPnet将其数据预处理部分与Kaldi紧密集成，以便1）我们可以比较Kaldi混合系统与ESPnet端到端系统的性能；2）我们可以利用Kaldi recipe中开发的数据预处理。尽管多通道端到端ASR[31]在其网络中包括语音增强和特征提取，但ESPnet也对大多数配方使用Kaldi特征提取。

**3.2 基于attention的编解码器**

3.2.1 编码器

默认的编码器网络由双向长短时记忆（BLSTM）表示，通过降采样（称为金字塔BLSTM[15]）给定长度的语音特征序列o 1:*T*，得到高层特征序列h 1:*T*’。

h 1:*T’* = BLSTM（o 1:*T*） （1）

由于降采样，通常*T’*<*T*。Chainer后端支持VGG[32]结构的卷积网络层，受[33, 34]的启发，可以在两个VGG层后接BLSTM层，即

h 1:*T’*= BLSTM（VGG2（o 1:*T*）） （2）

在许多情况下，这比金字塔BLSTM效果好。

3.2.2 Attention

ESPnet使用位置感知注意机制（location-aware attention mechanism）[35]，作为默认attention。还支持dot-product attention[36]。虽然location-aware attention有更好的效果，但dot-product attention要快得多。除了上述注意机制外，PyTorch后端还支持11种以上的注意力机制，比如additive attention[37]、覆盖机制（coverage mechanism）[38]和多头注意（multi-head attention）[39]。

**3.3 混合CTC/attention**

ESPnet是一种采用了混合的CTC/Attention端到端ASR[17]，有效地利用了这两种结构在训练和解码方面的优势。

3.3.1 多目标训练

在训练过程中，我们采用多目标学习框架，结合CTC Loss和attention的交叉熵Loss，提高鲁棒性，实现快速收敛。具体如下：

（3）

这种训练方法中，CTC和attention共用一个编码器。我们有一个调整参数来线性插值两个目标函数，通常设置为a=0.5（等贡献）。

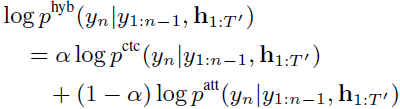
为了缓解过拟合问题，在训练过程中采用了标签平滑技术，通过将正确标签的概率密度与剩余标签的概率密度按一定比例进行分割，平滑目标分布。我们实现了unigram smoothing，其中剩余标签的分布被设置为与标签的unigram distribution成正比[40]。

3.3.2 Warp CTC

CTC是整个训练计算时间的主导部分之一。我们对Chainer和PyTorch后端使用了由[12]开发的warp-CTC库，与Chainer后端的内置CTC相比，它在总训练时间上提高了5-10%的速度。

3.3.3 联合解码

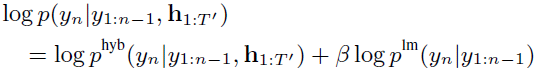
在解码过程中，我们结合attention和CTC的分数，采用一次波束搜索算法进行联合解码，进一步消除不规则对齐。假设在n位置的输出是yn，其历史输出为y1:n-1且编码器输出为h1:T’。在波束搜索过程中，使用attention *p*att和CTC *p*ctc对数概率进行以下分数组合

 （4）

这种混合的CTC/Attention结构（训练过程中的多目标学习和识别过程中的联合解码）是在[17]中提出的，与其他端到端ASR系统相比具有独特的功能。

**3.4 语言模型的使用**

基于Attention的端到端ASR最重要的功能之一是如何利用大量文本语料库训练的语言模型。ESPnet可以在解码过程中结合RNNLM的对数概率，如下所示

 （5）

β是一个附加的缩放参数。这种方法对应于译码器网络和RNNLM的浅层融合，最初在神经机器翻译[41]中提出，并应用于端到端语音识别[34]。

**3.5 不利环境下的ASR设置**

尽管ESPnet支持的大多数ASR示例都是标准的英语任务，但是当前ESPnet示例通过VoxForge支持其他语言，包括日语（CSJ）、汉语普通话（HKUST-CTS）和其他欧洲语言。有了这些不同的示例，ESPnet还可以通过遵循我们之前的研究[42]，实现多语言端到端ASR系统（例如，10种语言）。此外，ESPnet示例还包括噪声鲁棒性/远场语音识别任务，包括AMI[22]、CHiME-4[25]和CHiME-5任务[26]。尤其是ESPnet是CHiME-5挑战赛的官方端到端ASR基线系统。

**4 实施**

**4.1 标准处理流程**

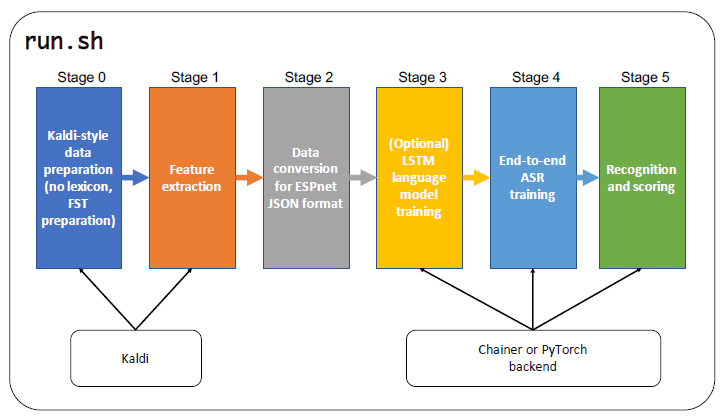


图 2标准ESPnet流程

图2显示了ESPnet中的标准处理流程。由于端到端ASR的优点，该流程大大简化了，例如，它不必包括词典准备、有限状态变换器（FST）编译、基于HMM和高斯混合建模的训练/对齐以及用于序列区分性训练的lattice生成。

标准示例run.sh中包括以下6个阶段：

第0步 数据准备：我们采用Kaldi数据目录格式，我们可以简单地使用Kaldi数据准备脚本（例如data\_prep.sh）。

第1步 特征提取：我们再次使用Kaldi特征提取。大多数示例使用80维对数Mel特征和pitch特征（总共83维）。

第2步 ESPnet数据准备：这个阶段将Kaldi数据目录中的所有信息（transcriptions、说话人和语言id、输入和输出长度）转换为一个JSON文件（json.data），除了输入特征。

第3步 语言模型训练：使用Chainer或PyTorch后端对基于字符的RNNLM进行训练。这是一个可选的阶段，有几个示例没有这个阶段。

第4步 端到端ASR训练：混合CTC/attention编解码器通过Chainer或PyTorch后端进行训练。

第5步 识别：分别使用第三阶段和第四阶段得到的RNNLM和端到端ASR模型进行语音识别。

**4.2 代码行**

除了减少实验步骤数，ESPnet还减少了代码量。表1比较了Kaldi、Julius和ESPnet的主要源代码。与Kaldi和Julius相比，ESPnet只需使用5K行python代码就可以实现语音识别，包括训练和识别功能。这得益于端到端识别的结构简单，Chainer或PyTorch作为神经网络后端，数据准备和特征提取使用Kaldi[3]。

表 1 代码行数比较

|  |  |  |
| --- | --- | --- |
| Toolkit | #lines | Language |
| Kaldi | 330K | C++ |
| Julius | 60K | c |
| ESPnet | 5.4K | python |

其中一个最简单的模块是模型表示部分，因为它不必显式地表示从语音特征、HMM状态、上下文相关音素、词汇到单词的复杂语音识别层次。这个层次结构由一个神经网络来表示，这个神经网络最多包含数千行python代码。这也简化了最多500行的识别模块，因为它是通过简单的输出同步波束搜索实现的。

**5 实验**

本节讨论了三个主要实验任务：WSJ，CSJ和HKUST。第一个实验通过几种实验配置证明了ESPnet在著名的WSJ任务上的有效性，并在端到端ASR框架下对这一任务的报告（其他论文结果）进行了比较。其他实验比较了ESPnet与最先进的ASR系统在CSJ和HKUST任务中的性能。选择这两种语言的主要原因是，这些表意文字语言的字母序列长度比字母语言的短，这大大降低了计算复杂度，并且使得在解码器网络中处理上下文信息变得容易。事实上，我们之前的调查显示，日语和汉语的端到端自动语调可以很容易地扩展，并且在不使用为大规模英语任务开发的各种技巧的情况下表现出合理的性能。

表 2 WSJ任务中与其他E2E系统的结果比较

|  |  |  |  |
| --- | --- | --- | --- |
| Method | Metric | dev93 | eval92 |
| ESPnet with VGG2-BLSTM | CER | 10.1 | 7.6 |
| + BLSTM layers (4->6) | CER | 8.5 | 5.9 |
| + char-LSTMLM | CER | 8.3 | 5.2 |
| + joint decoding | CER | 5.5 | 3.8 |
| + label smoothing | CER | 5.3 | 3.6 |
|  | WER | 12.4 | 8.9 |
| Seq2seq + CNN (no LM) [33] | WER | N/A | 10.5 |
| Seq2seq + FST word LM [35] | CER | N/A | 3.9 |
|  | WER | N/A | 9.3 |
| CTC + FST word LM [11] | WER | N/A | 7.3 |

|  |  |  |
| --- | --- | --- |
| Method | Wall Clock Time | #GPUs |
| ESPnet (Chainer) | 20 hours | 1 |
| ESPnet (PyTorch) | 5 hours | 1 |
| Seq2seq + CNN [33] | 120 hours | 10 |

表2展示了ESPnet中不同的技术在WSJ任务上进行了对比。在采用了更深层次的编码网络，集成了基于字符的LSTMLM，以及联合CTC/Attention解码后，性能稳步提高。

表2 也将ESPnet与其他报告做了对比。由于这些报告基于不同的条件（例如，[33]不使用任何语言模型，而[35]和[11]通过FST使用基于单词的语言模型），我们无法直接比较它们。但是，我们可以说，与先前的研究相比，ESPnet提供了合理的性能比较。表2还提供了主端到端ASR网络训练的计算时间，其中包含GPU的数量。ESPnet实现了非常快速的训练，特别是对于PyTorch后端，即使只有一个GPU（gtx 1080ti），对于相同的WSJ任务[33]。

然而，这些端到端ASR系统的一个问题是它们的性能没有达到最先进的混合HMM/DNN系统的性能。例如，用于WSJ任务的混合HMM/DNN系统的WER低于5%，这种退化可能是由于缺乏足够的训练数据。实际上，[12]和[16]报告说，在非常大的英语任务中，与最先进的混合HMM/DNN系统的性能相当或更高，尽管由于缺乏计算资源，许多研究团体通常无法完成这些结果。因此，通过保持较低的计算资源来扩展英语任务，或者通过缓解数据稀疏问题来提高英语任务的性能，是我们今后的重要研究方向之一。

表 3 CSJ任务比较

|  |  |  |  |
| --- | --- | --- | --- |
|  | Eval1 | Eval2 | Eval3 |
| ESPnet | 8.7 | 6.2 | 6.9 |
| ESPnet(5 GPUs) | 8.5 | 6.1 | 6.8 |
| HMM/DNN(kaldi nnet1) | 9.0 | 7.2 | 9.6 |
| CTC-syllable[43] | 9.4 | 7.3 | 7.5 |

表 4 HKUST普通话CTS任务比较（CER）

|  |  |
| --- | --- |
|  | eval |
| ESPnet | 28.3 |
| HMM/LSTM(Kaldi nnet3) | 33.5 |
| CTC with language model[11] | 34.8 |
| HMM/TDNN, LF MMI[27] | 28.2 |

与英语任务相比，在日语和汉语任务中，端到端ASR系统可以很容易地达到与现有的混合HMM/DNN系统相当的性能。请注意，ESPnet不使用词汇信息（发音词典和形态分析器），它们是HMM/DNN和CTC syllable系统中的基本组件。表格3和4，最佳结构的ESPnet（即VGG-BLSTM、字符RNNLM和联合解码）与混合HMM/DNN系统进行了对比。特别是在HKUST任务中，ESPnet几乎达到了采用lattice-free MMI训练[27]的HMM/DNN系统的最新最佳性能。

**6 结论**

介绍了一种新的端到端ASR工具包ESPnet。ESPnet充分利用动态神经网络工具箱Chainer和PyTorch作为主要的深度学习引擎，极大地简化了整个ASR流程的训练和识别。大量的实验和与其他报告的比较表明，ESPnet取得了较好的ASR性能，并且达到了与现有HMM/DNN系统相当的性能。ESPnet已经得到了积极的发展，多GPU功能、数据增强、multihead decoder、多通道端到端ASR和Babel多语种ASR实验正在准备中。特别是多GPU功能（5gpu），ESPnet仅用26小时就完成了581小时的CSJ任务训练。

**7 参考文献**

略