Jasper: An End-to-End Convolutional Neural Acoustic Model

Jason Li[1], *Vitaly Lavrukhin*[1], *Boris Ginsburg*[1], *Ryan Leary*[1], *Oleksii Kuchaiev*[1], *Jonathan M. Cohen*[1], *Huyen Nguyen*[1], *Ravi Teja Gadde*[2]

[1]NVIDIA, Santa Clara, USA
[2]New York University, New York, USA

概要:

speech recognition. Inspired by wav2letter's convolu- tional approach, we build a deep and scalable model, which requires a well-designed residual topology, effective regulariza- tion, and a strong optimizer. As our architecture studies demonstrated, a combination of standard components leads to SOTA results on LibriSpeech and competitive results on other bench- marks. Our Jasper architecture is highly efficient for training and inference, and serves as a good baseline approach on top of which to explore more sophisticated regularization, data aug- mentation, loss functions, language models, and optimization strategies. We are interested to see if our approach can continue to scale to deeper models and larger datasets.

我们为端到端语音识别提出了一种新的神经体系结构。受wav2letter的卷积方法启发，我们构建了一个深度和可扩展的模型，这需要一个精心设计的残余拓扑，有效的正则化，和一个强大的优化器。正如我们的体系结构研究表明的那样，标准组件的组合导致了LibriSpeech上的SOTA结果和其他基准测试上的竞争性结果。我们的Jasper架构对于训练和推断是非常高效的，并且可以作为一个很好的基线方法，在此基础上探索更复杂的正则化、数据配置、损失函数、语言模型和优化策略。我们很想知道我们的方法是否可以继续扩展到更深入的模型和更大的数据集。

1.模型结构

1.Jasper uses mel-filterbank features calculated from 20ms windows with a 10ms overlap, and outputs a probability dis- tribution over characters per frame2.

Jasper使用mel-filterbank特征，从20ms的窗口计算10ms重叠，输出每帧字符的概率分布。

2.Jasper has a block archi- tecture: a Jasper BxR model has B blocks, each with R sub- blocks. Each sub-block applies the following operations: a 1D- convolution, batch norm, ReLU, and dropout. All sub-blocks in a block have the same number of output channels.

Jasper有一个块结构:一个Jasper BxR模型有B个块，每个块有R个子块。每个子块应用以下操作:一维卷积、批范数、ReLU和dropout。一个块中的所有子块具有相同数量的输出通道。

3.Each block input is connected directly into the last sub- block via a residual connection. The residual connection is first projected through a 1x1 convolution to account for differ- ent numbers of input and output channels, then through a batch norm layer. The output of this batch norm layer is added to the output of the batch norm layer in the last sub-block. The re- sult of this sum is passed through the activation function and dropout to produce the output of the sub-block.

每个块输入通过一个残差连接直接连接到最后一个子块。这个残差连接首先通过1x1卷积投影，以考虑不同数量的输入和输出通道，然后通过批处理归一化层。此批归一化层的输出被添加到最后子块中的批处理归一化层的输出中。这个和的结果通过激活函数和dropout产生子块的输出。

4.The sub-block architecture of Jasper was designed to facil- itate fast GPU inference. Each sub-block can be fused into a single GPU kernel: dropout is not used at inference-time and is eliminated, batch norm can be fused with the preceding convo- lution, ReLU clamps the result, and residual summation can be treated as a modified bias term in this fused operation.

Jasper的子块架构是为了方便快速的GPU推断而设计的。每个子块可以融合成一个GPU内核:在推理时不使用dropout并消除，将批范数与前面的卷积进行融合，ReLU对结果进行夹取，并将残差求和作为一个修改的偏置项进行融合。

5.All Jasper models have four additional convolutional blocks: one pre-processing and three post-processing. See Fig- ure 1 and Table 1 for details.

所有Jasper模型都有四个额外的卷积块:一个预处理和三个后处理。详见图1和表1。

6.We also build a variant of Jasper, Jasper Dense Residual (DR). Jasper DR follows DenseNet [16] and DenseRNet [17], but instead of having dense connections within a block, the out- put of a convolution block is added to the inputs of all the fol- lowing blocks. While DenseNet and DenseRNet concatenates the outputs of different layers, Jasper DR adds them in the same way that residuals are added in ResNet. As explained below, we find addition to be as effective as concatenation.

我们还构建了Jasper的变体，Jasper密残(DR)。Jasper DR遵循DenseNet[16]和DenseRNet[17]，但在一个块内没有密集的连接，卷积块的输出被添加到所有下面块的输入中。虽然DenseNet和DenseRNet连接不同层的输出，但Jasper DR以与ResNet中添加残差相同的方式添加它们。正如下面所解释的，我们发现加法和串联一样

2.标准化和激活

In our study, we evaluate performance of models with:

• 3 types of normalization: batch norm [11], weight norm [10], and layer norm [18]

• 3 types of rectified linear units: ReLU, clipped ReLU (cReLU), and leaky ReLU (lReLU)

• 2 types of gated units: gated linear units (GLU) [9], and gated activation units (GAU) [19]

All experiment results are shown in Table 2. We first experi- mented with a smaller Jasper5x33 model to pick the top 3 set- tings before training on larger Jasper models. We found that layer norm with GAU performed the best on the smaller model. Layer norm with ReLU and batch norm with ReLU came sec- ond and third in our tests. Using these 3, we conducted further experiments on a larger Jasper10x4. For larger models, we no- ticed that batch norm with ReLU outperformed other choices. Thus, leading us to decide on batch normalization and ReLU for our architecture.

在我们的研究中，我们用以下方法评估模型的性能:

•3种归一化类型:批标准[11]，权重标准[10]，层标准[18]

•3种整流线性单元:ReLU，夹式ReLU (cReLU)，漏式ReLU (lReLU)

•2种门控单元:门控线性单元(GLU)[9]和门控激活单元(GAU) [19]

所有实验结果如表2所示。我们首先用一个较小的Jasper5x33模型进行实验，在较大的Jasper模型上进行训练之前，挑选出前3个集合。我们发现带GAU的层范数在较小的模型上表现最好。在我们的测试中，带ReLU的层范数和带ReLU的批范数分别位居第二和第三。使用这3个，我们在更大的Jasper10x4上进行了进一步的实验。对于较大的模型，我们没有发现使用ReLU的批处理规范优于其他选择。因此，我们决定将批处理规范化和ReLU用于我们的体系结构。

During batching, all sequences are padded to match the longest sequence. These padded values caused issues when us- ing layer norm. We applied a sequence mask to exclude padding values from the mean and variance calculation. Further, we computed mean and variance over both the time dimension and channels similar to the sequence-wise normalization proposed by Laurent et al. [20]. In addition to masking layer norm, we additionally applied masking prior to the convolution operation, and masking the mean and variance calculation in batch norm. These results are shown in Table 3. Interestingly, we found that while masking before convolution gives a lower WER, using masks for both convolutions and batch norm results in worse performance.

在批处理期间，所有序列都被填充以匹配最长的序列。这些填充值在我们层规范时引起了问题。我们应用序列掩码从平均值和方差计算中排除填充值。此外，我们计算了时间维度和通道上的平均值和方差，类似于Laurent等人提出的顺序归一化。[20]。除了遮蔽层范数外，我们在卷积运算前附加了遮蔽，并在批范数中遮蔽了均值和方差计算。这些结果如表3所示。有趣的是，我们发现，尽管在卷积之前使用掩码可以获得较低的WER，但在卷积和批范数中同时使用掩码会导致较差的性能。

3.残差连接

For models deeper than Jasper 5x3, we observe consistently that residual connections are necessary for training to converge. In addition to the simple residual and dense residual model described above, we investigated DenseNet [16] and DenseR- Net [17] variants of Jasper. Both connect the outputs of each sub-block to the inputs of following sub-blocks within a block. DenseRNet, similar to Dense Residual, connects the output of each output of each block to the input of all following blocks. DenseNet and DenseRNet combine residual connections using concatenation whereas Residual and Dense Residual use addi- tion. We found that Dense Residual and DenseRNet perform similarly with each performing better on specific subsets of Lib- riSpeech. We decided to use Dense Residual for subsequent ex- periments. The main reason is that due to concatenation, the growth factor for DenseNet and DenseRNet requires tuning for deeper models whereas Dense Residual simply just repeats a sub-blocks.

对于比Jasper 5x3更深的模型，我们一致观察到残余连接对于训练收敛是必要的。除了上面描述的简单残差和密集残差模型外，我们还研究了Jasper的DenseNet[16]和DenseR- Net[17]变体。两者都将每个子块的输出连接到一个块中的以下子块的输入。DenseRNet与致密残差(density Residual)类似，它将每个块的每个输出的输出连接到后面所有块的输入。DenseNet和DenseRNet使用连接连接的方法组合剩余连接，而residual和密residual则使用添加的方法。我们发现稠密残差和DenseRNet在Lib- riSpeech的特定子集上表现相似，且各自表现更好。我们决定在后续的实验中使用密集残差法。主要原因是由于连接，DenseNet和DenseRNet的生长因子需要调整更深入的模型，而致密残差只是重复子块。

3语言模型

A language model (LM) is a probability distribution over ar- bitrary symbol sequences P(w1,...,wn) such that more likely sequences are assigned high probabilities. LMs are frequently used to condition beam search. During decoding, candidates are evaluated using both acoustic scores and LM scores.Traditional N-gram LMS have been augmented with neural LMs in recent years

语言模型(LM)是任意的符号序列P(w1，...，wn)上的概率分布，这样更可能的序列被赋予高概率。LMs经常被用来调节光束搜索。在解码过程中，使用声学评分和LM评分对候选人进行评估。近年来，神经LMS对传统的N-gram LMS进行了扩充.

We experiment with statistical N-gram language models [24] and neural Transformer-XL [12] models. Our best results use acoustic and word-level N-gram language models to gen- erate a candidate list using beam search with a width of 2048. Next, an external Transformer-XL LM rescores the final list. All LMs were trained on datasets independently from acoustic models. We show results with the neural LM in our Results section. We observed a strong correlation between the quality of the neural LM (measured by perplexity) and WER as shown in Figure 3.
我们使用统计N-gram语言模型[24]和神经Transformer-XL[12]模型进行实验。我们的最佳结果是使用声学和词级N-gram语言模型，使用宽度为2048的束搜索生成候选列表。接下来，一个外部Transformer-XL LM恢复最终列表。所有LMs在独立于声学模型的数据集上进行训练。我们在结果部分展示了使用神经LM的结果。我们观察到神经LM的质量(通过困惑度测量)和WER之间有很强的相关性，如图3所示。

优化器:

For training, we use either Stochastic Gradient Descent (SGD) with momentum or our own *NovoGrad*, an optimizer similar to Adam [15], except that its second moments are computed per layer instead of per weight. Compared to Adam, it reduces memory consumption and we find it to be more numerically stable.

对于训练，我们要么使用带有动量的随机梯度下降(SGD)，要么使用我们自己的NovoGrad，这是一个类似于Adam[15]的优化器，只是它的第二个力矩是按层计算而不是按权重计算的。与Adam相比，它减少了内存消耗，而且我们发现它在数值上更稳定。

在每一个时间步t,NovoGrad计算随机梯度g_t^l.让后计算每个层的二阶矩v_t^l.

再计算一阶矩m_t^l之前,需要使用二阶矩v_t^l对梯度g_t^l进行缩放.

如果用到L2正则化,就将权重衰减$d * w_{t-1}$加在后面

最后,利用学习率a_t计算新的权重.

结果:

We evaluate Jasper across a number of datasets in various do- mains. In all experiments, we use dropout and weight decay as regularization. At training time, we use speed perturba- tion with fixed +/-10% [30] for LibriSpeech. For WSJ and Hub5'00, we use a random speed perturbation factor between [-10%, 10%] as each utterance is fed into the model. All models have been trained on NVIDIA DGX-1 in mixed precision [31] using OpenSeq2Seq [32]. Source code, training configurations, and pretrained models are available.

我们评估Jasper在不同的数据集在不同的主网。在所有的实验中，我们使用dropout和权值衰减作为正则化。在训练时，我们使用固定+/-10%[30]的LibriSpeech速度摄动。对于WSJ和Hub5'00，当每个话语被输入模型时，我们使用[-10%，10%]之间的随机速度扰动因子。所有模型都已使用OpenSeq2Seq[32]在NVIDIA DGX-1上进行混合精度[31]训练。可以获得源代码、训练配置和预训练模型。

我们评估了Jasper在两个读语音数据集上的性能:LibriSpeech和华尔街日报(WSJ)。对于Lib- riSpeech，我们使用我们的NovoGrad op- timizer对Jasper DR进行了10x5的训练，训练时间为400 epoch。我们在test-clean子集上实现了SOTA性能，在test-other上实现了端到端语音识别模型之间的SOTA性能。

我们用SGD和动量优化器训练了一个较小的Jasper 10x3模型，在合并的WSJ数据集(80小时)上训练了400个epoch: LDC93S6A (WSJ0)和LDC94S13A (WSJ1)。结果如表6所示。

We also evaluate the Jasper model's performance on a conversa- tional English corpus. The Hub5 Year 2000 (Hub5'00) evalua- tion (LDC2002S09, LDC2005S13) is widely used in academia. It is divided into two subsets: Switchboard (SWB) and Call- home (CHM). The training data for both the acoustic and language models consisted of the 2000hr Fisher+Switchboard training data (LDC2004S13, LDC2005S13, LDC97S62). Jasper DR 10x5 was trained using SGD with momentum for 50 epochs. We compare to other models trained using the same data and report Hub5'00 results in Table 7.

表现进行了评价。Hub5 2000年(Hub5'00)评价(LDC2002S09, LDC2005S13)在学术界得到广泛应用。它分为两个子集:总机(SWB)和呼回(CHM)。声学和语言模型的训练数据均由2000小时Fisher+Switchboard训练数据(LDC2004S13, LDC2005S13, LDC97S62)组成。Jasper DR 10x5使用SGD进行了50课时的动量训练。我们将其与使用相同数据训练的其他模型进行比较，并在表7中报告Hub5 ' 00结果。