Very Deep self-attention networks for end2end speech recognition

Carlos Alberto Rios Rubiano
Heidelberg University

Dec, 2019

Very Deep self-attention networks for end2end speech recognition

Ngoc-Quan Pham[1], *Thai-Son Nguyen*[1], *Jan Niehues*[1], *Markus Müller*[1], *Sebastian Stüker*[1], *Alex Waibel*[1,2]

[1]Interactive Systems Lab, Karlsruhe Institute of Technology, Karlsruhe, Germany
[2]Carnegie Mellon University, Pittsburgh PA, USA

ngoc.pham@kit.edu, thai.nguyen@kit.edu

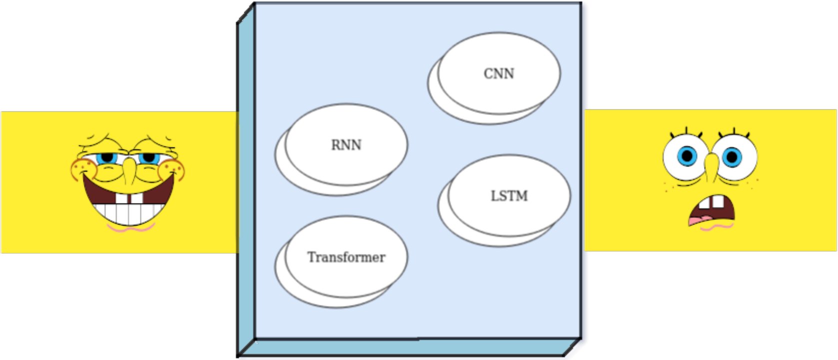
Very Deep self-attention networks for end2end speech recognition

Very Deep self-attention networks for end2end speech recognition

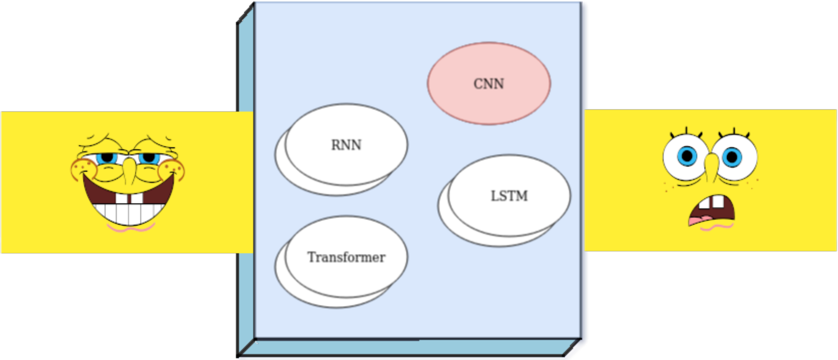
Very Deep self-attention networks for end2end speech recognition

I: Motivation and overview

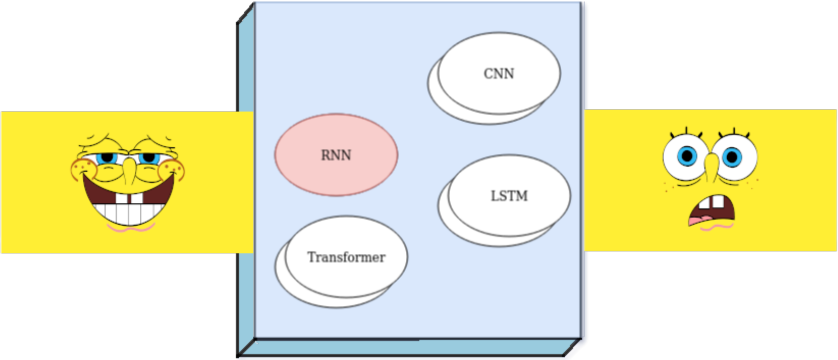
Previous research related to the paper:



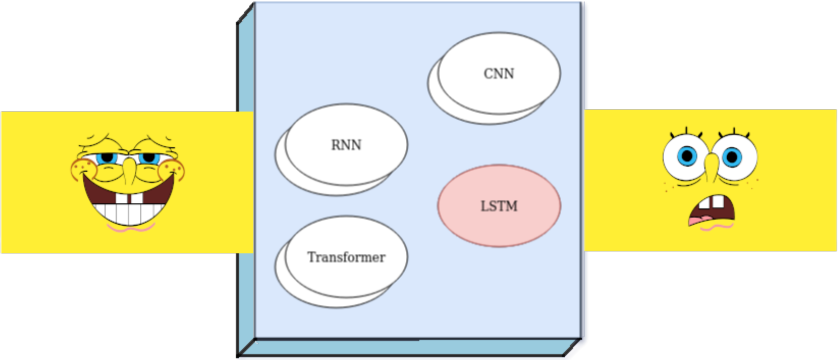
Previous research related to the paper:



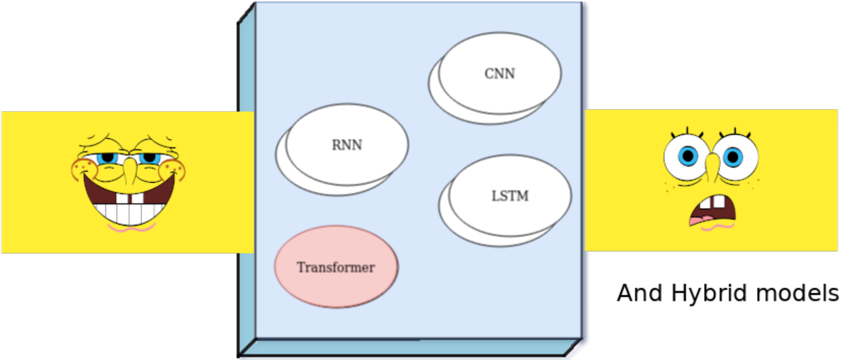
Previous research related to the paper:



Previous research related to the paper:

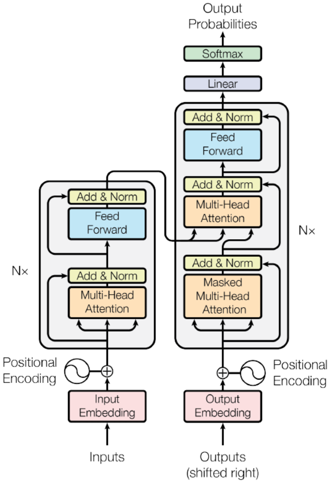


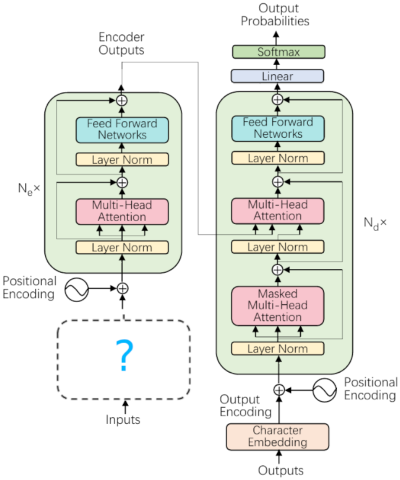
Previous research related to the paper:



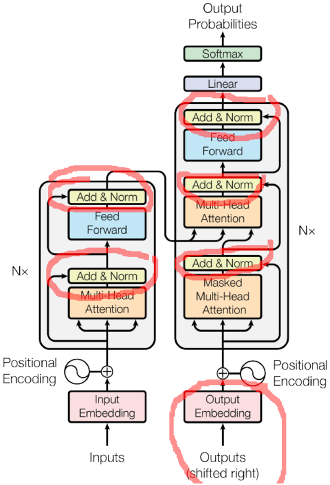
And Hybrid models

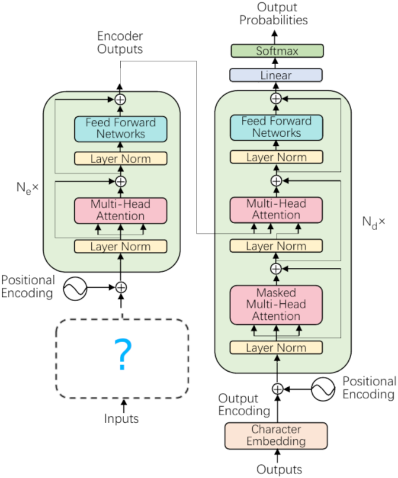
First differences:





First differences:

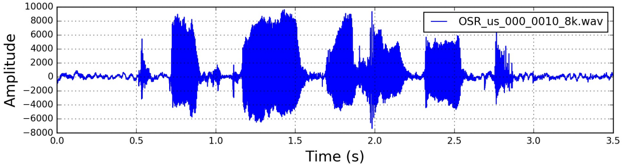




40 log Mel Filter bank (Intuition):

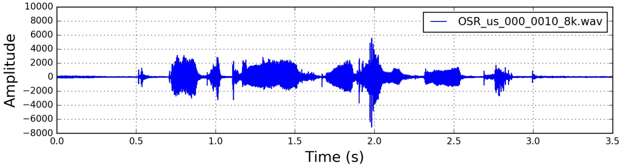
Aim: Mimic the non-linear human ear perception.

Raw audio:



Pre-Emphasis:
to amplify the high frequencies

$$y(t) = x(t) - \alpha x(t-1)$$

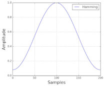


40 log Mel Filter bank (Intuition):

Aim: Mimic the non-linear human ear perception.

- **Framing:**
  Frame sizes typically from 20 ms to 40 ms with 50% overlaping.

- **Window:**

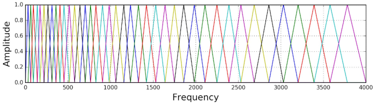


To counteract the assumption made by the FFT that the data is infinite

- **FFT and Power Spectrum:**

$$P = \frac{|FFT(x_i)|^2}{N}$$

- **Filter Banks:**

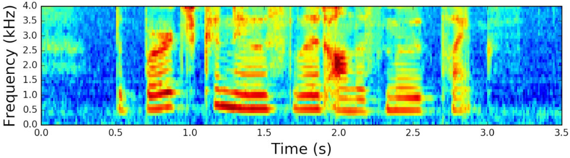


Mimic the non-linear human ear perception of sound, by being more discriminative at lower frequencies and less discriminative at higher frequenciest.

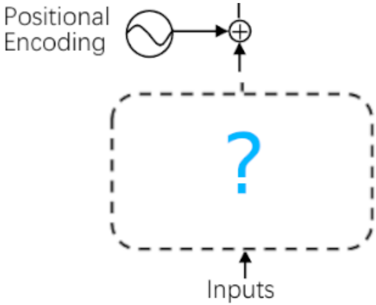
40 log Mel Filter bank (Intuition):

Aim: Mimic the non-linear human ear perception.

From 40 filters, to finally have something like:



Two challenges to face



1. Self attention memory grows quadratically in the sequence lenght.
2. How to incorporate positional information on the model?

Two challenges to face: First attempt

Self-Attentional Acoustic Models

Matthias Sperber[1], *Jan Niehues*[1], *Graham Neubig*[2], *Sebastian Stüker*[1], *Alex Waibel*[1,2]

[1]Karlsruhe Institute of Technology
[2]Carnegie Mellon University
{first}.{last}@kit.edu, gneubig@cs.cmu.edu

1. Self attention memory grows quadratically in the sequence lenght. (Downsampling.)
2. How to incorporate positional information on the model? (Hybrid model: LSTM and Transformer Blocks)

Two challenges to face: First attempt

WER results on position modeling.

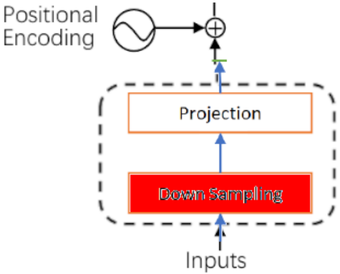
model	dev	test
add (trig.)	diverged	
concat (trig.)	30.27	38.60
concat (emb.)	29.81	31.74
stacked hybrid	16.38	17.48
interleaved hybrid	15.29	16.71

IV: Model

The proposed model:

1. Self attention memory grows quadratically in the sequence lenght. (Downsampling)
2. How to incorporate positional information on the model? (projecting the concatenated features to a higher dimension before adding the positional information)

The proposed model:

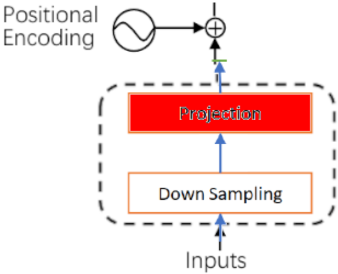


Downsampling (Reshaping operation by factor a)

$$\mathbf{X} \in \mathcal{R}^{l \times d} \rightarrow \hat{\mathbf{X}} \in \mathcal{R}^{l/a \times d * a}$$

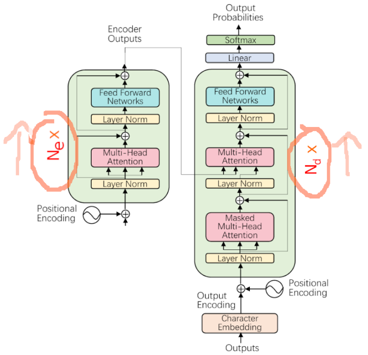
As usuall l stads on for the sequence lenght, and d is the hidden dimension.

The proposed model:



Projection:
Projecting the Downsampled features to a higher dimension.

The proposed model: How to go deeper on the Transformer model?

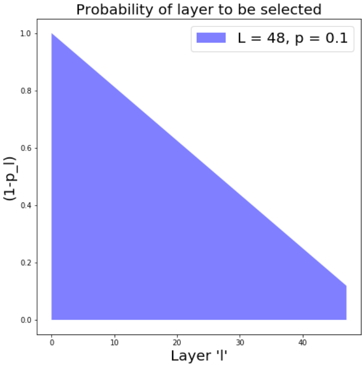


Stochastic Layers (During training): The residual connection of an input x and its corresponding neural layer F has the following form:

$$R(x) = LayerNorm(M * F(x) + x)$$

M takes 0 or 1 as values, generated from a Bernoulli distribution. Causing the effect of ensembling different sub-networks.

The proposed model: How to go deeper on the Transformer model?



p Is the global probability for dropping layers.

- Sub-layers inside each encoder or decoder layer share the same mask M.
- Each layer have a local probability $p_l = \frac{l}{L}(1-p)$.

The proposed model: How to go deeper on the Transformer model?

During the training

$$R(x) = \mathrm{LayerNorm}(M * F(x) * \frac{1}{1 - p_l} + x)$$

Scale the layer to their respective probability to be selected.

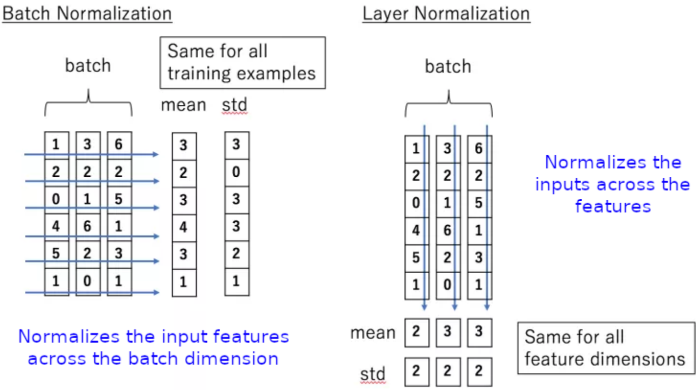
During inference

$$R(x) = \mathrm{LayerNorm}(F(x) + x)$$

p Is the global probability for dropping layers.

- Sub-layers inside each encoder or decoder layer share the same mask M.
- Each layer have a local probability $p_l = \frac{l}{L}(1 - p)$.

Layer Normalization vs Batch Normalization: Intuition



IV: Results

About the datasets:

For the experiments, two different data sets were used:

Switchboard-1 R2	Used to train: 300 hs conversation, 5 min/stuck average, Telephon speech USA,
HUB5'00	Used to test: Linguistic Data Consortium, Telephon Conversational Speech. which contains two types of data, Switchboard (SWBD) better matched to the training data and CallHome (CH)
TED-LIUM 3	Used to train(Second experiment): 452 hrs of TED Talks, 11 min/stuck average
TED-LIUM	Used to test(Second experiment): 118 hrs of TED Talks

Other useful information:

40 log Mel Filter bank: Mimic the non-linear human ear perception. (Discriminative at lower frequencies)

Speed perturbation: Data augmentation with speed factors of $0{,}9, 1{,}0$ and $1{,}1$

Abbreviation

TDNN	Time-delay neural networks
BLSTM	bidirectional LSTMs
LFMMI	Lattice-free maximum mutual information.
CTC + CharLM	Connectionist Temporal Classification + Character-level language model
LSTM w/att	Long Short Term memory with attention mechanism.
LSTM − LM	LSTM + 4-gram word language model
Seq2Seq	Attention based sequence-to-sequence model. With improvements.
CTC − A2W	CTC + Acoustic-to-Word (LSTM Model for Large Vocabulary Speech Recognition)

Performance

Layers	#Param	SWB	CH
04Enc-04Dec	21M	20.8	33.2
08Enc-08Dec	42M	14.8	25.5
12Enc-12Dec	63M	13.0	23.9
+*Stochastic Layers*		13.1	23.6
24Enc-24Dec	126M	12.1	23.0
+*Stochastic Layers*		11.7	21.5
+*Speed Perturbation*		10.6	20.4
48Enc-48Dec	252M	-	-
+*Stochastic Layers*		11.6	20.9
48Enc-48Dec (half-size)	63M	-	-
+*Stochastic Layers*		12.5	22.9
08Enc-08Dec (big)	168M	13.8	25.1

Performance

Layers	**#Param**	**SWB**	**CH**
24Enc-12Dec	113M	13.3	23.7
+*Stochastic Layers*		11.9	21.6
36Enc-8Dec	113M	12.4	22.6
+*Stochastic Layers*		11.5	20.6
36Enc-12Dec	113M	12.4	22.6
+*Speed Perturbation*		11.2	20.6
+*Stochastic Layers*		11.3	20.7
+*Both*		**10.4**	**18.6**
40Enc-8Dec	109M	–	–
+*Stochastic Layers*		11.9	21.4

Performance

Hybrid/End-to-End Models	Tgt Unit	SWB	CH
TDNN +LFMMI [23]	Phone	10.0	20.1
BLSTM +LFMMI [23]	Phone	**9.6**	19.3
CTC+CharLM [24]	Char	21.4	40.2
LSTM w/attention [1]	Char	15.8	36.0
Iterated-CTC +LSTM-LM [25]	Char	14.0	25.3
Seq2Seq +LSTM-LM [26]	BPE	11.8	25.7
Seq2Seq +Speed Perturbation [27]	Char	12.2	23.3
CTC-A2W +Speed Perturbation [28]	Word	11.4	20.8
36Enc-12Dec (Ours)	Char	10.4	18.6
48Enc-12Dec (Ours)	Char	10.7	19.4
60Enc-12Dec (Ours)	Char	10.6	19.0
Ensemble		9.9	**17.7**

Performance

TED-LIUM 3 training set.

Models	Test WER
CTC [19]	17.4
CTC/LM + speed perturbation [19]	13.7
12Enc-12Dec (Ours)	14.2
Stc. 12Enc-12Dec (Ours)	12.4
Stc. 24Enc-24Dec (Ours)	11.3
Stc. 36Enc-12Dec (Ours)	**10.6**

Conclusions

- The use of stochastic layers, allows to implement more layers in the transformers, obtaining good results, which is understood as an assembly of sub networks.

Conclusions

- The use of stochastic layers, allows to implement more layers in the transformers, obtaining good results, which is understood as an assembly of sub networks.
- Deeper networks with smaller size are more beneficial than a wider yet shallower configuration.

Conclusions

- The use of stochastic layers, allows to implement more layers in the transformers, obtaining good results, which is understood as an assembly of sub networks.
- Deeper networks with smaller size are more beneficial than a wider yet shallower configuration.
- This article showed that it is possible to solve the ASR task with a good performance based on the use of transformers, while retaining the appealing advantages that it offers.

Thanks!.

Outline I

Questions:

- What is Speed Perturbation?
- How do neural ASR models compare to the hybrid models in terms of computation requirements? Is there a difference regarding training vs. inference? How does the presented transformer compare to e2e systems like the Seq2Seq + LSTM-LM?
- The authors state that the encoder requires deeper networks than the decoder. Are there cases known where the opposite is the case?

Outline II

- What do the authors mean by sub-networks (mentioned multiple times in chapter 2.4)? They say in one section: "Studies about residual networks have shown that during training the network consists of multiple sub-networks taking different paths through shortcut connections [16], and thus there are redundant layers."
  It is not clear to me how these sub-networks and the shortcut paths come into being, and how they would cause the layers to be redundant?

Outline III

- The hyper-parameter search in this paper revolves around the base version of Transformer (chapter 3.2). However, the big Transformer generally achieves better results, and this paper too reports best results with the big Transformer. Thus, why would they not have used the big version instead of the base version for hyper-parameter tuning? Is there even a difference between using either model for tuning?
- I can understand the s to mask out some layers in training, however as we can see from the result in table 1: system can gain much better performance with more layers actually and the mask approach does not really beneficial for the system, they need to do more experiments on the mask probability with the setting of 24 layer to show a more convinced result or?

Outline IV

- How is the speech perturbation training set generated? The speed perturbation approach seems to be a nice approach in favor of speech recognition learning task.
- What does Word Error Rate (WER) measure?
- The increased dropout probability for higher layers seems to suggest that they are less important for achieving good results. How does this match with the finding that (great) deepness is highly beneficial?
- How did the authors actually managed to acquire linguistic features for the representations, let alone the speaker identities they discussed in the introduction?

Outline V

- As seen in WaveNet 1x1 Convolutions has been successful in processing audio data, yet however they critiqued CNNS and also mentioned in the paper that there is still a vanishing gradient problem with a LSTM over longer distance, is that true?
- Confused about Figure 1: The decoder has a layer named "source attention". What precisely is the difference between regular multihead-attention as used in the original Transformer and source attention, especially as it is simply called "self-attentionïn all other layers in the figure?
- Why does projecting the input features into a 512 dimension and then adding the positional encoding not harm the model, whereas doing the adding in the original dimension does harm the training process?

Outline VI

- I have no real questions towards the paper. Maybe we could talk about why they state: "While directly adding acoustic features to the positional encoding is harmful?". What do they mean with it?
- As a second investigation point: As dropout and residual connections are that helpful, I wonder if the addition of zone-out might have helped to make the network even deeper.
- What did they use as train, dev and test set? Their data split was confusing for me
- Why do we need the 1/(1-p) factor in equation 5.