Language Independent End-to-End Architecture For Joint Language and Speech Recognition (2017)

Watanabe, S.; Hori, T.; Hershey, J.R.

Motivation / Goal

Recognize multiple languages at the same time

- Use a single model for 10 languages (EN, JP, CH, DE, ES, FR, IT, NL, PT, RU)
- Check if transfer learning between languages work
- two tasks: identify language AND recognize speech (simultaneously)

Problems

- How to input audio?

  $\rightarrow$ Spectral features of audio frames (e.g. in 10ms segments)

Problems

- How to input audio?

  $\rightarrow$ Spectral features of audio frames (e.g. in 10ms segments)

- How to output text?
  (a) word embeddings (word2vec)
  (b) characters (one-hot)
    - Different char sets for languages (abc, äàąå, 漢字, , ひらがな)
    - Just unify all character sets (5500 total)

Problems

- How to input audio?

  → Spectral features of audio frames (e.g. in 10ms segments)

- How to output text?
  (a) word embeddings (word2vec)
  (b) characters (one-hot)
    - Different char sets for languages (abc, äàąå, 漢字, , ひらがな)
    - Just unify all character sets (5500 total)

- how to output language id?
  (a) separate softmax output
  (b) [EN]Hello[CH] 你好

Related Work

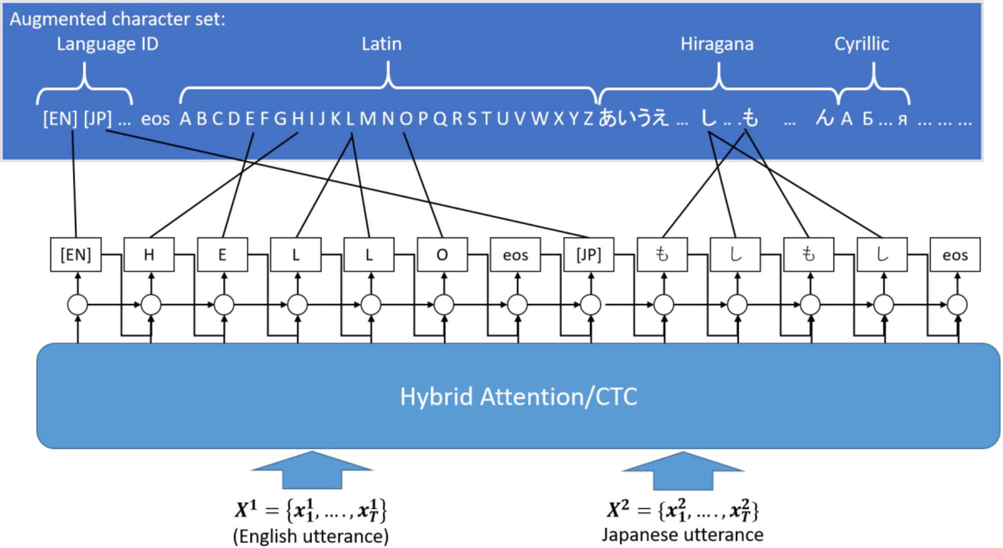
Related Work

(e.g. only attention)

- Multilingual Speech Recognition With A Single End-To-End Model (Shubham Toshniwal)
  - separate output for language id
  - only on 9 indian languages

Model overview

Model overview



Model overview

1. Input: for each audio frame one 2d input image, 3 channels (like RGB image processing)
   - spectral features
2. Encoder
   2.1 VGGNet Convolutional NN (first 6 layers)
   2.2 One bidirectional LSTM layer (320 cells x2)
3. Decoder (Attention + one directional LSTM)
   - Soft Attention for each input frame to each output character
   - LSTM (300 cells), input: previous hidden state and output, attention-weighted input features
   - fully connected layer (converts 300 outputs from LSTM -> N characters softmax)
4. Output
   - N characters from union of all languages (softmax)

Input

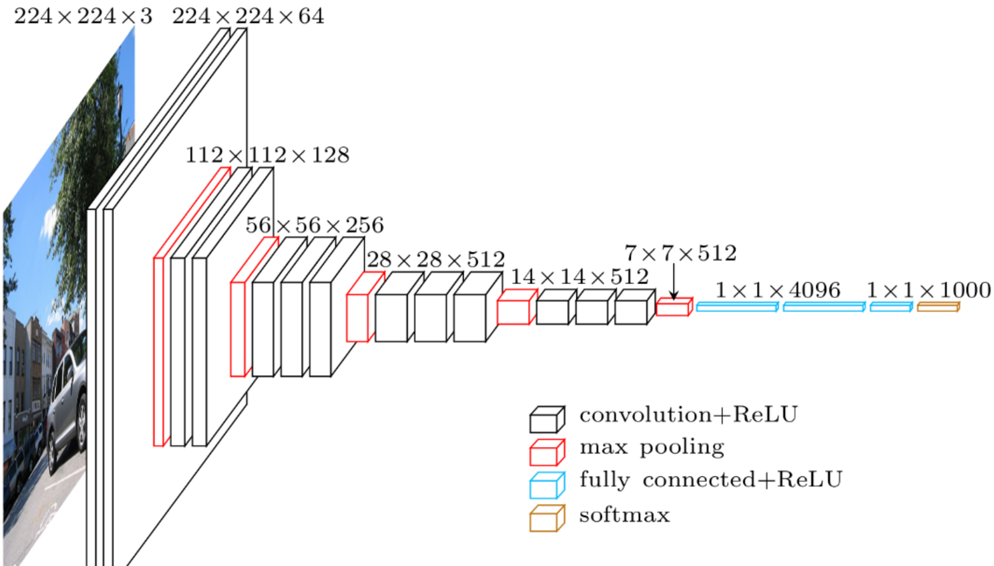
(Ab)use of image processing pipeline - input formatted like a RGB image

first channel: spectral features second channel: delta spectral features third channel: deltadelta spectral features

probably cepstral? fourier, fundamental frequency variation? etc

either just one feature map or they have some convolution issues

Encoder - VGG Net Architecture



Encoder - VGG Net Architecture - First six layers

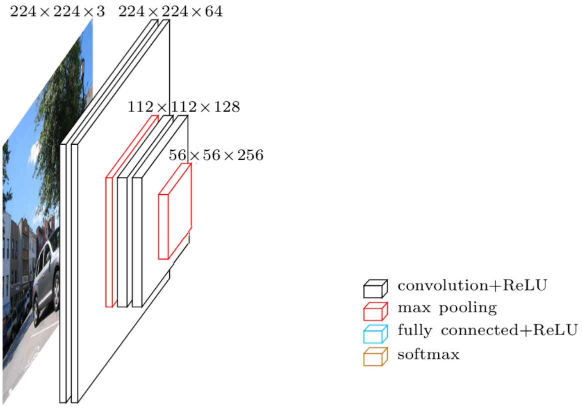


Figure 3: VGG Net - first 6 layers

(actual input dimensions are not mentioned)

Encoder - Bidirectional LSTM layer

320 cells for each direction $\rightarrow$ 640 outputs per time step

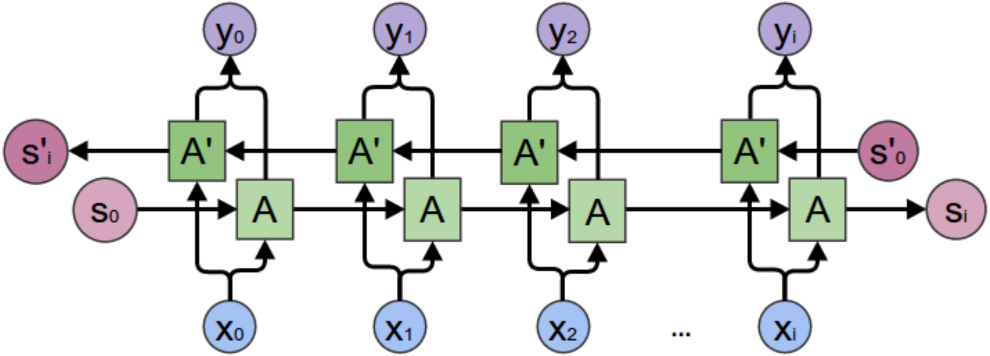


Figure 4: Bidirectional LSTM

http://colah.github.io/posts/2015-09-NN-Types-FP/

Decoder (Attention-based)

Additions to the base model

Second, Parallel Decoder (CTC)

1. Input (same as before)
2. Encoder (same as before)
3. Decoder (CTC) fully connected layer per time stemp (converts 640 outputs from BLSTM -> N characters, softmax)
4. One output character per input frame, normalized using CTC Loss

- First, add blank character to set. e.g. Hello -> {H, E, L, O, -}
- Inference: Remove duplicates: HHHH-EEEEEEEE-LL-LLL—-OOOOOO $\rightarrow$ H-E-L-L-O $\rightarrow$ HELLO
- Training: HELLO $\rightarrow$ H-E-L-L-O $\rightarrow$ all combinations of char duplications are ok

https://towardsdatascience.com/intuitively-understanding-connectionist-temporal-classification-3797e43a86c

RNN-LM

- Model distribution of characters in languages (ignores input speech)

Combine both decoders + RNN-LM

- AdaDelta optimization
- Training objective function: 0.5 * CTC loss + 0.5 * Attention loss + 0.1 RNN-LM loss
- Inference via beam search on attention output weighted by objective function

Conclusions

- adding a pure language model (RNN-LM) improves performance a bit

Language Confusion Matrix

		CH	EN	JP	DE	ES	FR	IT	NL	RU	PT
CH	train_dev	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	dev	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
EN	test_eval92	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	test_dev93	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
JP	eval1_jpn	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	eval2_jpn	0.0	0.0	100.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
	eval3_jpn	0.0	0.0	99.9	0.0	0.0	0.0	0.1	0.0	0.0	0.0
DE	et_de	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.3	0.0	0.0
	dt_de	0.0	0.0	0.0	99.7	0.0	0.0	0.0	0.3	0.0	0.0
ES	dt_es	0.0	0.0	0.0	0.0	67.9	0.0	31.9	0.0	0.0	0.2
	et_es	0.0	0.0	0.0	0.1	91.1	0.0	8.4	0.1	0.0	0.2
FR	dt_fr	0.0	0.0	0.0	0.1	0.0	99.4	0.0	0.2	0.0	0.3
	et_fr	0.0	0.0	0.0	0.1	0.0	99.5	0.0	0.1	0.0	0.3
IT	dt_it	0.0	0.0	0.0	0.0	0.3	0.4	99.1	0.0	0.0	0.3
	et_it	0.0	0.0	0.0	0.0	0.4	0.4	98.3	0.2	0.1	0.7
NL	dt_nl	0.0	0.0	0.0	1.3	0.0	0.1	0.1	97.2	0.0	1.3
	et_nl	0.0	0.0	0.0	1.0	0.0	0.2	0.2	97.6	0.0	0.9
RU	dt_ru	0.2	0.0	0.0	0.0	0.2	0.6	0.5	0.0	97.9	0.8
	et_ru	0.0	0.0	0.0	0.2	0.2	0.3	4.3	0.0	94.7	0.3
PT	dt_pt	0.0	0.0	0.0	0.3	0.3	2.6	1.7	3.4	0.6	91.2
	et_pt	0.0	0.3	0.0	0.3	0.0	0.0	3.9	3.6	0.3	91.5

Figure 5: Language identification (LID) accuracies/error rates (%). The diagonal elements correspond to the LID accuracies while the offdiagonal elements correspond to the LID error rates

Potential problems / future work?

- nothing ensures language does not switch mid sentence $\rightarrow$ Apparently not an issue
  - but maybe we want to allow this? (append utterances from different languages)
- uniform random parameter initialization with [-0.1, 0.1] sounds bad
- does not work in realtime (without complete input utterance)
  - Bidirectional LSTM in encoder
    - Could try one directional, but Language ID would completely break
    - aggregate limited number of future frames (e.g. 500ms latency)
  - CTC in realtime?
  - Attention does not work in realtime
- same latin characters are used for multiple languages, while others (RU, CN, JP) get their own character set

BC Utterance Selection

- Get a list of all backchannel phrases
- BC phrase list from the *Switchboard Dialog Act Corpus* (SwDA)

BC Utterance Selection

- Get a list of all backchannel phrases
- BC phrase list from the *Switchboard Dialog Act Corpus* (SwDA)

SwDA incomplete

→ Identify utterances only from their text

Something like “uh” can be a disfluency or a BC.

→ only include phrases with silence or BC before them.

→ Balanced data

Feature Selection

- Acoustic features like power, pitch
- Linguistic features (from the transcriptions)

Neural network design

Recurrent NNs

BCs are more probable after a longer period without BCs.

$\rightarrow$ Use RNN / LSTM

Recurrent NNs

BCs are more probable after a longer period without BCs.

$\rightarrow$ Use RNN / LSTM

Postprocessing

NN output is

- a value between 0 and 1
- quickly changing
- noisy

Postprocessing – Low-pass filter

Gauss filter looks into future

$\rightarrow$ Cut off filter and shift it

Thresholding / Triggering

- Use areas of output $>$ threshold t $(0 < t < 1)$
- Trigger at local maximum

Evaluation

Objective Evaluation

- Precision (portion of predictions that were correct)
- Recall (portion of correct BCs that were predicted)
- F1-Score (harmonic mean of Precision and Recall)

Lots of parameters to tune

- Context width
- Context stride
- Which features
- NN depth
- NN layer sizes
- LSTM vs Feed forward
- Trigger threshold
- Gaussian filter sigma
- Gaussian filter cutoff
- Prediction delay

Lots of parameters to tune

manually through trial and error:

- Context width
- Context stride
- Which features
- NN depth
- NN layer sizes
- LSTM vs Feed forward

automatically with Bayesian optimization:

- Trigger threshold
- Gaussian filter sigma
- Gaussian filter cutoff
- Prediction delay

Results

Context width

Context	Precision	Recall	F1-Score
500 ms	0.219	0.466	0.298
1000 ms	0.280	0.497	0.358
1500 ms	0.305	0.488	**0.375**
2000 ms	0.275	0.577	0.373

Table 1: Results with various context lengths. Performance peaks at 1500 ms.

LSTM vs FF

Layers	Parameter count	Precision	Recall	F1-Score
FF $(56:28)$	40k	0.230	0.549	0.325
FF $(70:35)$	50k	0.251	0.468	0.327
FF $(100:50)$	72k	0.242	0.490	0.324
LSTM $(70:35)$	38k	0.305	0.488	**0.375**

Table 2: LSTM outperforms feed forward architectures.

Layer sizes

Layer sizes	Precision	Recall	F1-Score
100	0.280	0.542	0.369
$50 : 20$	0.291	0.506	0.370
$70 : 35$	0.305	0.488	**0.375**
$100 : 50$	0.303	0.473	0.369
$70 : 50 : 35$	0.278	0.541	0.367

Table 3: Comparison of different network configurations. Two LSTM layers give the best results.

Features

Features	Precision	Recall	F1-Score
power	0.244	0.516	0.331
power, pitch	0.307	0.435	0.360
power, pitch, mfcc	0.278	0.514	0.360
power, ffv	0.259	0.513	0.344
power, ffv, mfcc	0.279	0.515	0.362
power, pitch, ffv	0.305	0.488	**0.375**
word2vec$_{dim=30}$	0.244	0.478	0.323
power, pitch, word2vec$_{dim=30}$	0.318	0.486	0.385
power, pitch, ffv, word2vec$_{dim=15}$	0.321	0.475	0.383
power, pitch, ffv, word2vec$_{dim=30}$	0.322	0.497	**0.390**
power, pitch, ffv, word2vec$_{dim=50}$	0.304	0.527	0.385

Table 4: Results with various input features, separated into only acoustic features and acoustic plus linguistic features.

Other research

Predictor	Precision	Recall	F1-Score
Baseline (random)	0.042	0.042	0.042
Müller et al. (offline)	–	–	0.109
Our results (offline, context of −750 ms to 750 ms)	0.114	0.300	**0.165**
Our results (online, context of −1500 ms to 0 ms)	0.100	0.318	0.153

Table 5: Comparison with previous research.

Varying margin of error

Margin of Error	Constraint	Precision	Recall	F1-Score
−200 ms to 200 ms		0.172	0.377	0.237
−100 ms to 500 ms		0.239	0.406	0.301
−500 ms to 500 ms		0.247	0.536	0.339
0 ms to 1000 ms	Baseline (random)	0.079	0.323	0.127
	Only acoustic features	0.294	0.488	0.367
	Acoustic and linguistic features	0.312	0.511	0.388

Table 6: Results with various margins of error used in other research. Performance improves with a wider margin width and a later margin center.

Survey

Randomly show participants 6 samples of the following categories

1. Random predictor
2. NN predictor
3. Ground truth

Survey

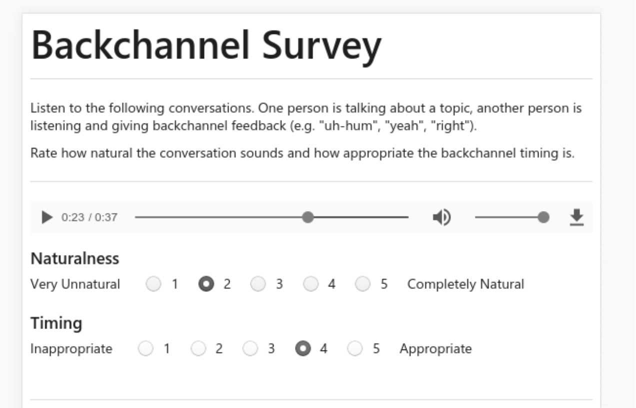


Figure 6: Screenshot of the survey interface.

Survey Results

Predictor	Sample	Timing	Naturalness	Sample Size
random	average	2.33 points	2.63 points	40
nn	average	3.48 points	3.08 points	40
truth	average	4.20 points	4.08 points	40

Table 7: Average user ratings of different BC predictors

Thank you for your attention

Addendum

Demo

http://localhost:3000/
#N4IgDgNghgngRIAxgawAoHsDOBLALt9AOxAC5CBXCCAGhADN0r0B3VaeJZUuqCTAU1
aeXH9dJA-WO5T-OQHoAgg-4APXOKS5+
AEwdQvoiaICI65LzYAF4ScgDqUGIM4gC2IcKEdNgA5nKE6KmRMQAy6Ii86aZmhLgASICE2f
cAurQAFvw5T7hyUOS46CHk17c5pdDDY7BYrEYTGZHC5PA1MIhxNgwLgHD4PCEwpgIhB
KFXEIMoVIRVfg1eqNZoUKhXbA3EgjB7PV7Zd6fb6-
Lk8voXaigf7c24DWiQ2xmcEgqHNEDONFeQgIpEohzMAq+
XTi24TWhYnF48RyYkYsmZHJ5AqrWn8UrICCVcTVOoNJqkDkQfkgF5vD4Er4-
EDA6zq1Xx5UarUAOVTAAJ0N8wN9MA4RDtVJ0shRCNgsn5OiNOtk+nB+
ItFHBOr5sOJcDASHxcClOg3MLgS6CVZ0wCwJNQwHhEE9qHQ6CJqEbxL5IGJEAB9NspOb
ehkgRBMIkB5qtdobD06iHMoiiKMMZx7MMyjjMsICrOsJC9KMMyjBMfSjKchzDBMwwzMcl
6Hswx6nuePSYCk6DoLgTyYIe7q2gSd78MkaSOhSLo0ni77eoyvrMv6bKkP+
HS9KoJx9Kocx7PMfSxcoEF9CsayAVMGHjAM0wRZhYFSkRAIkMoyj0SOKYuOx2a4LmuD5
EVtgVY1nWDZNpJ7adt2g5yeYQ4MaO47MJO04nupi7Lsa678FuO57p0B5HtIJn9WZFIWU8P
UKcmmBKSNKIjbO86TVpM1zXpC1LUZK1PGea3mZZ1k9HAwRHdi14ObeiTOQ+
bnPvdr7ut5Pp+
qyH3BaQ3Q9DMsWKH0cwzAMKiTGcSWIb0UEDBIAyA5K0G5cDBUXD95H-

SwDA categories

	name	act_tag	example	full_count
1	Statement-non-opinion	sd	Me, I'm in the legal department.	75145
2	**Acknowledge (Backchannel)**	b	Uh-huh.	38298
3	Statement-opinion	sv	I think it's great	26428
4	Agree/Accept	aa	That's exactly it.	11133
5	Abandoned or Turn-Exit	%	So, -	15550
6	Appreciation	ba	I can imagine.	4765
7	Yes-No-Question	qy	Do you have to have any special training?	4727

Figure 7: SwDA categories

Context stride

Stride	Precision	Recall	F1-Score
10ms	0.290	0.490	0.364
20ms	0.305	0.488	**0.375**
40ms	0.285	0.498	0.363

Table 8: Results with various context frame strides.