machine learning conference

OpenSpeech:

a deep learning platform for speech recognition experiments and benchmarks

Julian Salazar (julsal@)
Machine Learning Scientist | AWS AI

Collaborators

Zhiheng Huang
(zhiheng@)

Julian Salazar
(julsal@)

Karishma Malkan
(mkarishm@)

Ajay Mishra
(misaja@)

Sheng Zha
(zhasheng@)

Alex Smola
(smola@)

Jerry Zhang
(zhongyue@)

Hassan Sawaf
(hassan@)

Davis Liang
(liadavis@)

Background

- Automatic Speech Recognition (ASR):

  audio → text

  (+ speaker IDs, punctuation, code-switching, etc.)

- Amazon:
  - As a component: **Echo, Alexa, Lex**
  - As a service: **Transcribe**

Background

- Traditional ASR consists of a number of components: feature extractor, lexicon, acoustic model, language model, decoder, etc.
- Deep, “end-to-end”, robust ASR is still a topic of research

ASR system

Feature extraction

Decoder

"Hello AMLC!"

Acoustic modeling

Language modeling

Lexicon

Speech

Text

Days or weeks to train

ASR system

¡DEEP LEARNING!

“Hello AMLC!”

Background

- Many ASR platforms:
  - Open source: Kaldi, CMU Sphinx, HTK, EESEN, Mozilla DeepSpeech
  - Closed source: Alexa, Nuance Dragon, Siri, Google Cloud Speech, Bing Speech
- Current open source platforms do not provide all of:
  - A complete pipeline, with all components, at scale (tens of thousands of hours, multi-GPU)
  - Principled comparison of different architectures, esp. at the neural network level (HMM-NN vs. CTC vs. Seq2Seq)
  - Simple recipes with easy declaration and substitution of ASR components

OpenSpeech

- A deep learning, ASR research platform, built on Apache MXNet/Gluon for large-scale, multi-GPU training
  - CTC-based acoustic models (convolutional and recurrent)
  - Basic sequence-to-sequence with attention models
  - Support for phoneme and grapheme lexicons
  - Experiment management (weight/hyperparameter logging, TensorBoard views, MXNet profiling)

OpenSpeech

- Feature creation and processing recipes (via Kaldi/EESEN)
- Language model construction (via IRSTLM) and weighted finite state transducer (WFST) decoding (via OpenFST)
- OpenSpeech provides close to state-of-the-art ASR accuracy

OpenSpeech (example recipe)

40-dim filterbank

NN+WFST

"Hello AMLC!"

BLSTM+ CTC loss AM

4-gram, 250k word LM

CMU lexicon

Speech

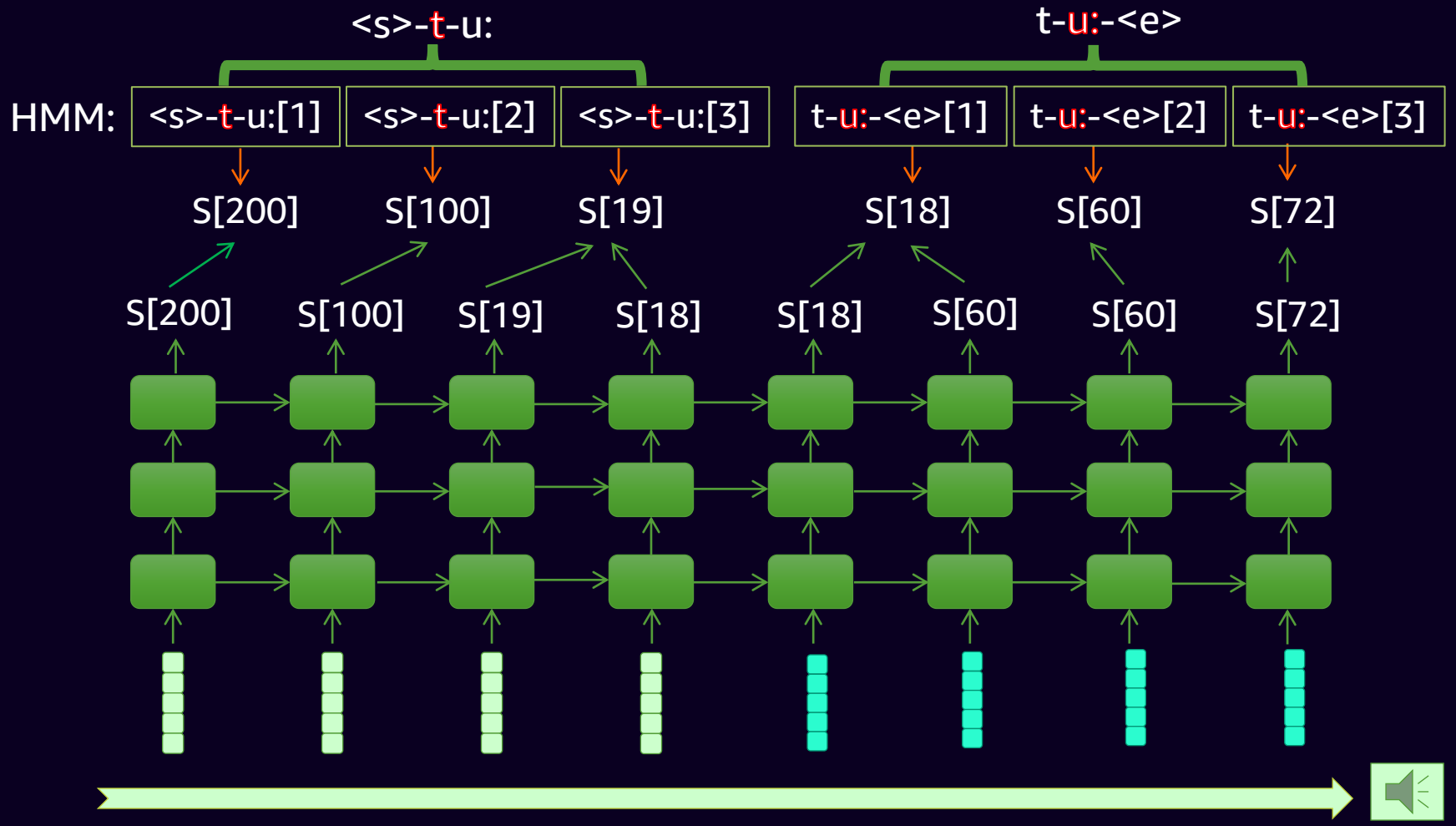
Text

Hours or days to train

ASR approaches

- HMM-NN framework
  - G. E. Dahl et al., 2012, *"Context-Dependent Pre-trained Deep Neural Networks for Large Vocabulary Speech Recognition"*
- CTC framework
  - A. Graves et al., 2006, *"Connectionist Temporal Classification: Labelling Unsegmented Sequence Data with Recurrent Neural Networks"*
- Sequence-to-sequence (seq2seq) framework
  - J. Chorowski et al., 2015, *"Attention-Based Models for Speech Recognition"*

HMM-NN framework



HMM-NN framework

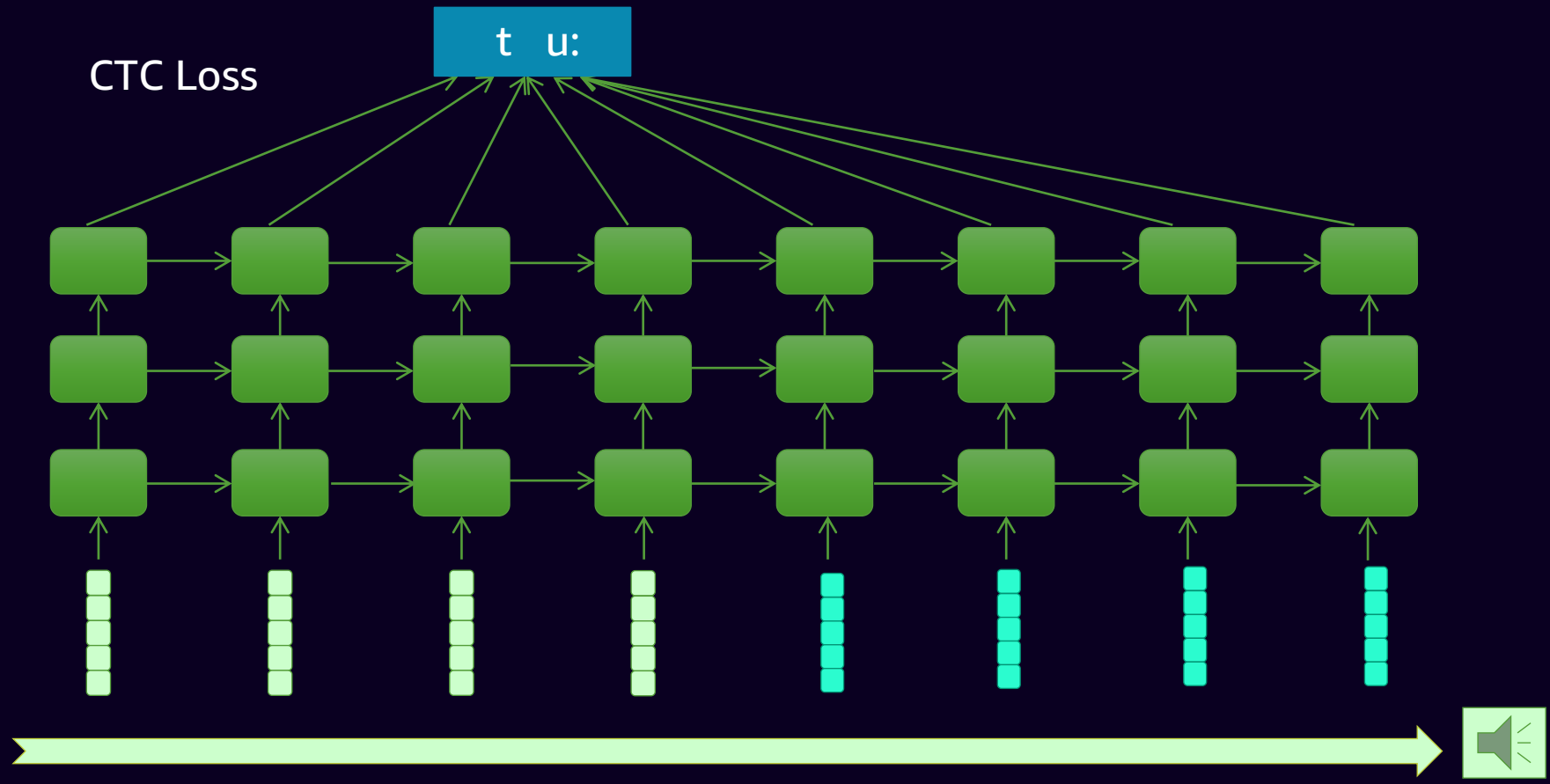
HMM-NN frameworks require forced alignment of training data

Actual training data:

audio file → “to be or not to be”

Need to guess at training time!

CTC framework



Connectionist Temporal Classification (CTC)

- CTC introduces a specific <blank> “_” label to represent no prediction at a frame
- Many CTC paths map to a label sequence “t u:” by removing repetitions and <blank>
- The likelihood of a label sequence is the (differentiable) sum of the probabilities of its CTC paths, computed via dynamic programming

seq2seq framework

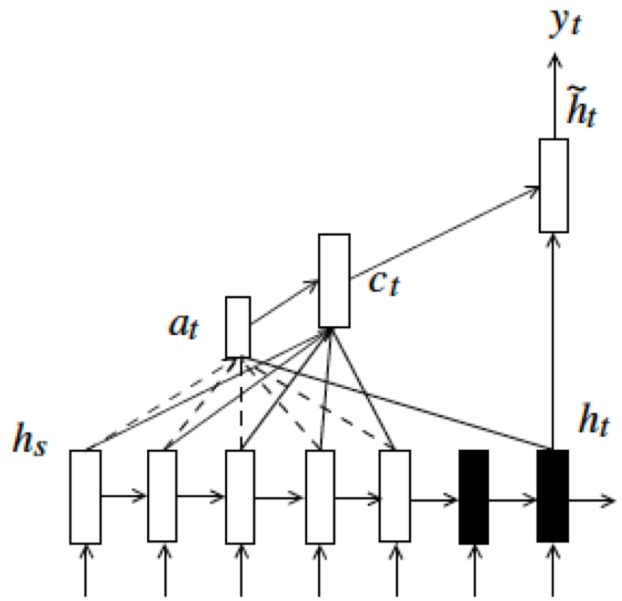


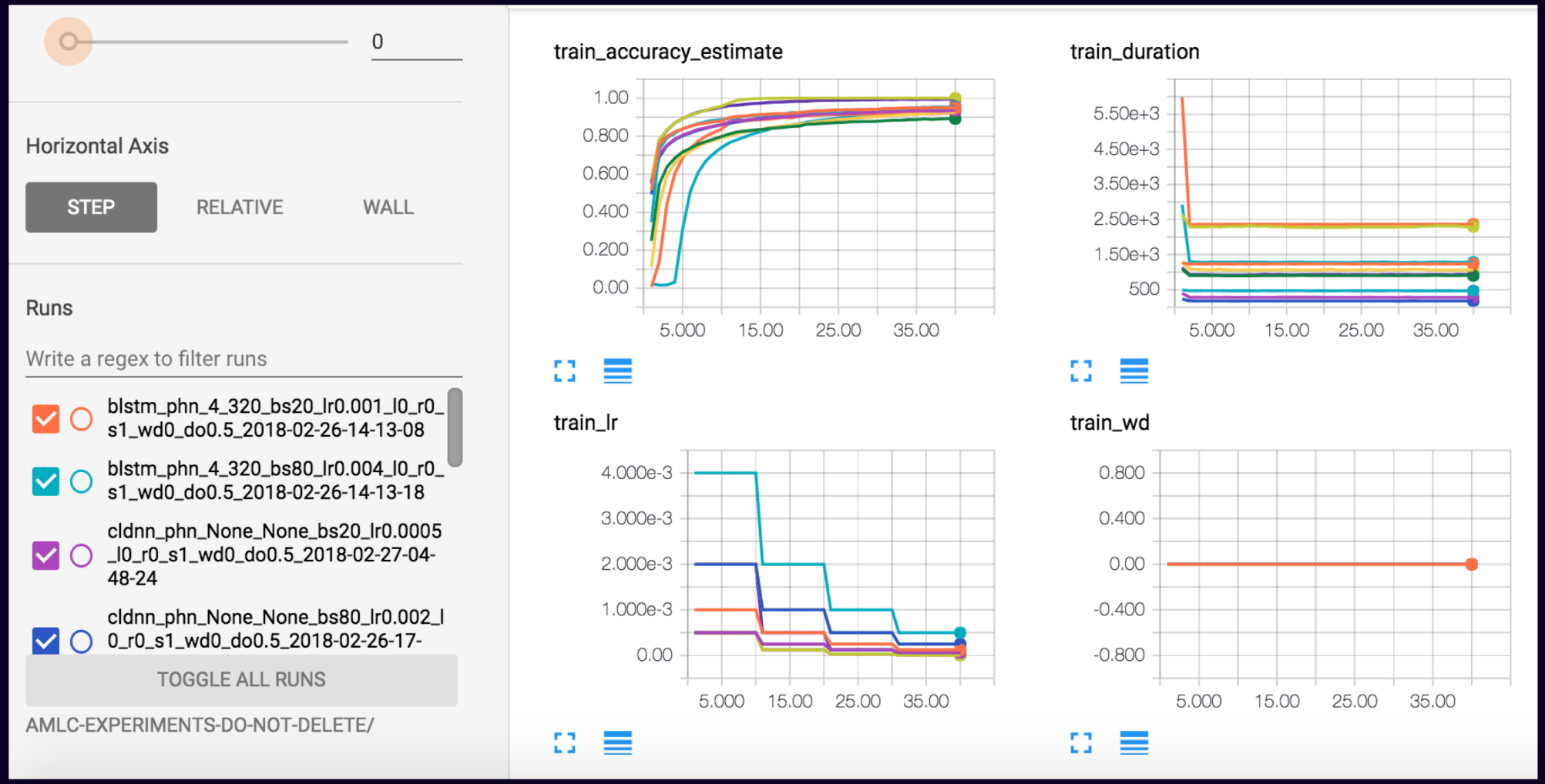
Figure 2: Sequence-to-sequence model architecture: at each time step t, the model infers a variable-length alignment weight vector $a_t(s)$ based on the current target state h_t and all source states $h_{s'}$. A global context vector c_t is then computed as the weighted average of all source states with weights specified in $a_t(s)$.

WSJ dataset (80 hours)

Table 1: Parameter size, batch size (BS), learning rate (LS), training speed, phoneme error rate (PER), and word error rates (WER) of our BLSTM, CLDNN, DS2, DiCNN and GCNN models on the WSJ dataset. They are grouped first by the number of GPUs, then by the type of architecture (recurrent-only, convolutional-recurrent, then convolutional only).

	Model	# params	BS	LR	hours/epoch	PER (dev93)	WER (dev93/eval92)
1 GPU	BLSTM	9M	20	1×10^{-3}	0.346	**10.21**	**8.33 / 5.48**
	CLDNN	11M	20	5×10^{-4}	0.081	15.44	11.50 / 7.50
	DS2	32M	10	5×10^{-4}	0.641	11.99	10.91 / 6.57
	DiCNN	21M	20	5×10^{-4}	0.252	13.85	9.10 / 5.79
	GCNN	186M	20	5×10^{-4}	0.661	11.25	9.39 / 6.52
4 GPU	BLSTM	9M	80	4×10^{-3}	0.134	**10.48**	**9.55** / 5.97
	CLDNN	11M	80	2×10^{-3}	0.052	15.58	12.35 / 7.96
	DS2	32M	40	2×10^{-3}	0.263	12.05	12.41 / 7.74
	DiCNN	21M	80	2×10^{-3}	0.299	12.77	**9.55 / 5.58**
	GCNN	186M	80	5×10^{-4}	0.358	11.59	10.34 / 6.79

WSJ dataset (80 hours)



WSJ dataset (80 hours)

Model Architecture	Systems	Training data size (hours)	WER (dev93 / eval92)
BLSTM	EESEN (Miao et al., 2015)	80	10.98 / 6.70
	OpenSpeech	80	**8.33 / 5.48**
DS2	Baidu Deep Speech 2 (Amodei et al., 2015)	~11,000	**4.98 / 3.60**
	OpenSpeech	80	10.54 / 6.57

LibriSpeech (960 hours)

Model Architecture	Systems	Training data size (hours)	WER (dev93 / eval92)
BLSTM	EESEN (Miao et al., 2015)	960	7.44 / 8.15
	OpenSpeech	960	**5.51 / 5.82**
DS2	Baidu Deep Speech 2 (Amodei et al., 2015)	~11,000	**N/A / 5.33**
	Mozilla Deep Speech	~5,000	N/A / 6.50
	OpenSpeech	960	6.74 / 6.99
GCNN	Facebook Wav2Letter (Liptchinsky et al., 2017)	960	N/A / **4.80**
	OpenSpeech	960	N/A / N/A

Other experiments

10,000 hours: BLSTM+CTC is competitive with HMM+RNN models on production data

Grapheme-based systems: 1 GPU (Tesla K80) on WSJ dataset

Model	#params	#epochs	Hours/epoch	CER (dev93)	WER (dev93/eval92)
BLSTM	9M	20	0.71	11.69	11.25 / 6.86

seq2seq model: 4 GPU (Tesla K80), PER over top 380 utterances in dev93

Model	#epochs	Hours/epoch	PER(dev93)
4BLSTM-LSTM-dot	23	0.40	17.97
4BLSTM-LSTM-mlp	23	0.40	15.83

Roadmap for 2018

- Traditional frame-wise cross-entropy training
- Sequence-to-sequence modeling improvements
- Multi-GPU improvements / multi-machine support
- Neural language model training and (re)scoring
- Language and telephony expansion
- Runtime code and streaming decoding support

Thank you!

Code.Amazon (mirror):
code.amazon.com/packages/Openspeech/trees/master

GitHub (requires permission):
github.com/awslabs/openspeech

Inquiries:
zhiheng@, julsal@