The project Kaldi

Open source speech recognition

Karel Vesely

Speech@FIT, BUT

ZRE, Brno, 3.5.2017

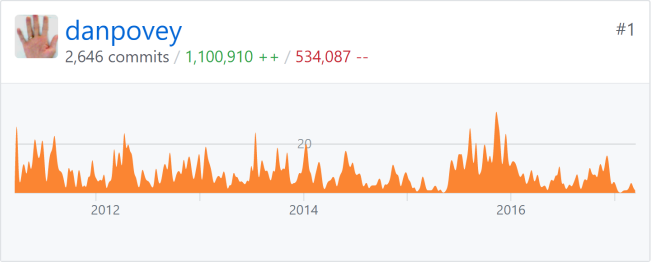
What is Kaldi?

- Wiki: A legendary Ethiopian goatherd who tried the coffee seeds after seeing the ‘energetic jumping goats’ eating it.
- Github: Open-source toolkit for building **speech recognition** systems.

A bit of history...

- 2009: Summer workshop at Johns Hopkins University (Baltimore, USA)
  - ASR team worked on *Sub-space Gaussian Mixture Models* (part of model parameters is shared across languages)
  - A toolkit was needed to integrate the new model!
- 2010: Dan Povey started coding Kaldi at Microsoft
- **2010, 2011, 2012, 2013: Kaldi development workshops**
  - **several weeks of summer coding in 'zámeček' at FIT**
  - **international team of self-funded volunteers**
    (USA, Canada, China, India, Germany, Czech Republic, ...)
- 2011: Kaldi toolkit presented at conferences ICASSP (Prague), ASRU (Hawaii)
- 2012: Dan Povey joins JHU in Baltimore (leaving Microsoft)
- 2015: Kaldi moved from SourceForge to **GitHub**

Who is this ’Dan Povey’?



- The ’#1’, i.e. the main architect of Kaldi.
- He is believed to write C++ code at the speed of light!

What is Kaldi? II.

Kaldi = GitHub project[1], it consists of:

- Set of command-line **programs for training and representing speech recognition models** (C++).
- **example recipes** = set of "**standard experiments**" on cluster computer (BASH, perl, awk, SGE cluster)
- **Documentation**[2]**:** Doxygen with tutorial, topic-based pages and C++ code reference
- **Support:** discussion forum (response usually in < 1 day)

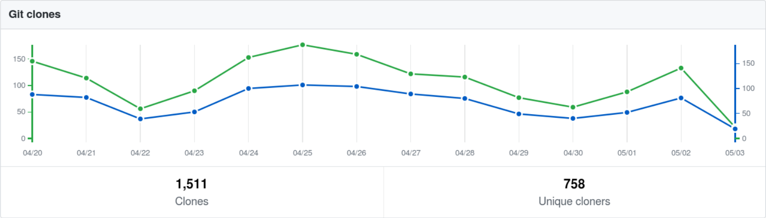
---

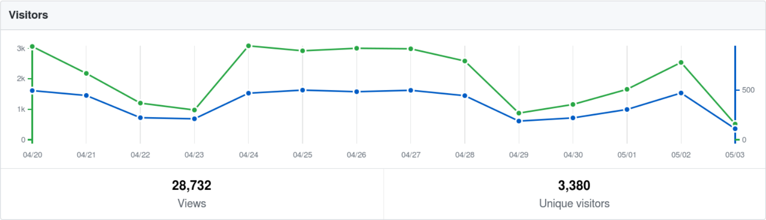
[1] `https://github.com/kaldi-asr/kaldi`

[2] `http://kaldi-asr.org/doc/`

What is Kaldi? III.

Github traffic stats from last 14-days
(the blue curves are unique 'cloners' and 'visitors'),





The example recipes = main strength of Kaldi

The recipes are main strength of Kaldi compared to other toolkits! (HTK, Sphinx, Julius, ...)

- Toy examples: yes/no, tidigits,
- Free-databases: AMI meetings (80h), TED-LIUM talks (120h), librispeech, voxforge, vystadial_cz
- The standard tasks (from easy to difficult, +/- paid data):
  - **Read speech:** Resource Management (3h, WER=1.5%), TIMIT (3h), Wall Street Journal (80h, **WER[3]=4%**),
  - **Conversational telephone speech:** Switchboard (300h, **WER=10%**), Fisher (2000h)
  - **Spontaneous 'distant microphone-array' speech:** AMI meetings (80h captured by 8 mic-array **WER=36%**, with 'close-talk mic' we get **WER=23%**)

[3] WER = word error rate

Why is Kaldi good for research?

- **Experiments are very easy to reproduce:** (all researchers can work with same baseline systems)
- No need to implement everything from scratch
- The toolkit is easy to extend or modify
- It is a **community project**, anybody can:
  - propose a change
  - send bugfix
  - fork and create derived project
- **License:** Apache v2.0, a very liberal legal framework: allows modifications and commercial use.

Speech recognition research ecosystem

Researchers:
- are using the toolkit
- some are contributors

KALDI

Big companies:
- some use Kaldi
- all have access to the code

Start-ups:
- getting free ASR technology
- creating new ASR applications

Big companies doing speech research: Nuance, IBM, Google, Microsoft, Apple, Amazon, Baidu, Telefonica, Samsung.
Many have open work positions...

Implemented techniques

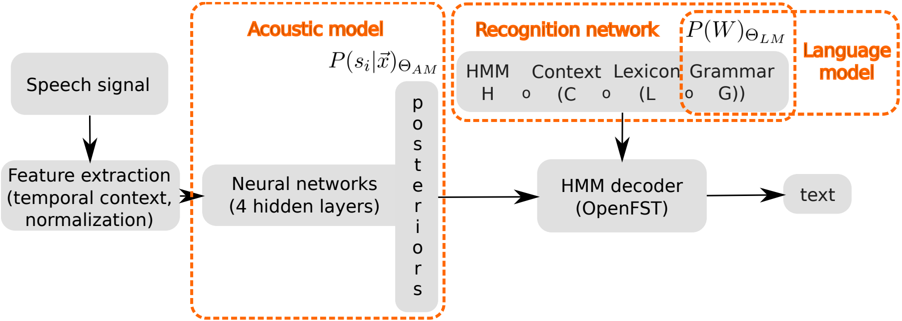
Speech recognition:

- HMM decoder using WFST transducers
- keyword search based on WFSTs

- Acoustic models: GMM, SGMM, DNN (nnet1,2,3)
  (DNN types: feed-forward, Convolutional, LSTM, BLSTM)
- Language models: N-GRAM, RNNLM

- speaker adaptation techniques
  (CMVN, VTLN, fMLLR, iVector based)
- sequence-discriminative training bMMI, sMBR
  (global optimization instead of ‘per-frame’ training)

Speech recognition: A hybrid approach



The decoding formula:

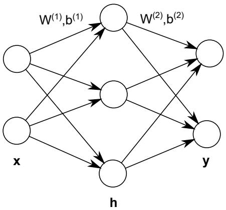
$$\tilde{W} = \underset{W}{\operatorname{argmax}}\, P(W|X)_{\Theta} \propto \underset{W}{\operatorname{argmax}}\, P(X|\vec{s}_W)_{\Theta_{AM}} P(W)_{\Theta_{LM}}$$

We use Bayes rule to convert NN posteriors into likelihoods:

$$P(\vec{x}|s_i)_{\Theta_{AM}} = P(s_i|\vec{x})_{\Theta_{AM}} / P(s_i)$$

Acoustic model: Neural network

Example: feed-forward neural network with one hidden layer,



$\mathbf{x}$ input vector
$\mathbf{h}$ hidden-layer vector
$\mathbf{y}$ output vector

$\mathbf{W^{(1)}}, \mathbf{W^{(2)}}$ matrices of trainable weights
$\mathbf{b^{(1)}}, \mathbf{b^{(2)}}$ vectors of trainable biases

Sigmoid,

$$h_i^{(1)} = \sigma(a_i^{(1)}) = \frac{1}{1 + \exp(-a_i^{(1)})}$$

Softmax,

$$y_i = \frac{\exp(a_i^{(2)})}{\sum_j \exp(a_j^{(2)})}, \quad \sum_i y_i = 1$$

Forward pass,

$$\mathbf{y} = \mathrm{softmax}\left(\mathbf{W}^{(2)}\,\sigma\left(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\right) + \mathbf{b}^{(2)}\right)$$

Acoustic model: Neural network

How does it look practically?
(layer = linear transform + non-linearity)

```
number-of-parameters 9.73591 millions
component 1 : <AffineTransform>, input-dim 567, output-dim 1024
component 2 : <Sigmoid>, input-dim 1024, output-dim 1024
component 3 : <AffineTransform>, input-dim 1024, output-dim 1024
component 4 : <Sigmoid>, input-dim 1024, output-dim 1024
component 5 : <AffineTransform>, input-dim 1024, output-dim 1024
component 6 : <Sigmoid>, input-dim 1024, output-dim 1024
component 7 : <AffineTransform>, input-dim 1024, output-dim 1024
component 8 : <Sigmoid>, input-dim 1024, output-dim 1024
component 9 : <AffineTransform>, input-dim 1024, output-dim 5859
component 10 : <Softmax>, input-dim 5859, output-dim 5859
```

- 4 hidden layers, each composed of 1024 neurons,
- 5859 classes on the output
  (triphone states = acoustic units)

Acoustic model: Training the Neural Network

- supervised training of a classifier
  (input features classified into triphone tied-states),
- training labels are generated by 'aligning' the transcriptions to the speech signal with an existing model,
- training algorithm: mini-batch Stochastic Gradient Descent:

$$\vec{w}_{t+1} = \vec{w}_t - \eta \nabla E(\vec{w}_t)$$

- avoiding over-training by reducing the learning rate,
  we observe accuracy on held-out set

Where we use kaldi

- in research, for publishing results in conference articles,
- for cooperation with international colleagues,
- in various funded research projects,

What can you do with Kaldi

- Play with the toy examples:
  `yesno`, `voxforge`, `vystadial_cz`
- Think of a creative 'speech-based' application (pre-built models are available `http://kaldi-asr.org/downloads/all/`).

Useful links

GitHub project:

- **https://github.com/kaldi-asr/kaldi**

Documentation:

- **http://kaldi-asr.org/doc/**

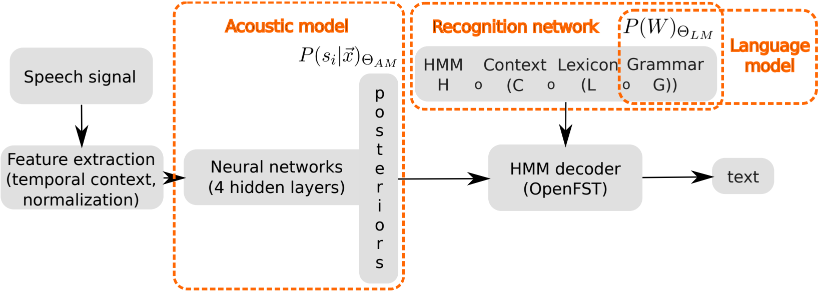
Support forum:

- https://groups.google.com/forum/#!forum/kaldi-help

Other resources:

- http://www.danielpovey.com/kaldi-lectures.html
- http://www.danielpovey.com/publications.html
- http://www.danielpovey.com/
- http://kaldi-asr.org

The DEMO, I.



Feature extraction:

- `compute-fbank-feats`
- `compute-pitch-feats`
- `paste-feats`, `apply-cmvn`

Acoustic model evaluation: `nnet-forward`
HMM decoder: `decode-faster-mapped`
Showing the output: `utils\int2sym.pl`

The DEMO, II.

Show the script...

- Lexicon with 579k ‘words’,
- HCLG network has 1.4GBs (after LM pruning),
- Acoustic model has 9.7 million trainable parameters (feed-forward neural network with 4 hidden layers and 5862 outputs),
- On-line cepstral mean normalization,
- Acoustic-model + HMM-decoder are background processes (communicating via ‘named pipes’),

The DEMO, III., FBANK features

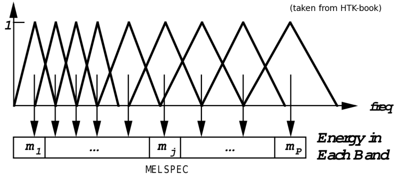
FBANK features = a smooth spectrogram,

- 10ms time-steps, non-uniform steps in frequency

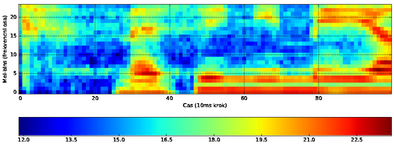
  (but uniform on Mel-scale, according to which we hear),
- $\log$ of the 'power' at particular frequency as integrated with the triangular Mel-filters,
- we splice 21 FBANK frames to form the DNN input

  (i.e. we take a window over 21 time-steps),

(taken from HTK-book)



Bank of Mel-filters

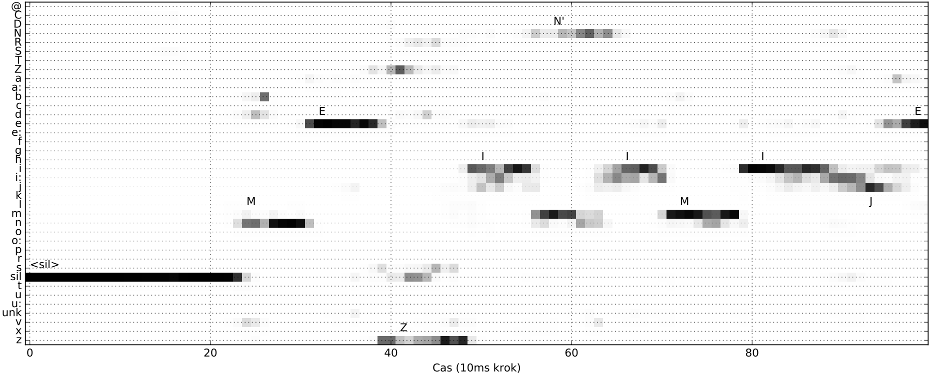


FBANK features

The DEMO, IV., NN posteriors

How does the Neural Network output look like?
(posterior probabilities)

For illustration we summed the 5859 outputs into 36+2 phonemes:



The DEMO, V.

Let's try it out!

Hints:

- Unusual words? (science, slang, ...)
- Casual Czech? (hovorová čeština)
- Poetry? (Polednice, Máj, ...)
- Classical books? (Babička, ...)

For optimal performance, please put the mike right at your lips.

The end

Thank you!