Connectionist Temporal Classification
for
End-to-End Speech Recognition

Yajie Miao, Mohammad Gowayyed, and
Florian Metze

July 14, 2016

Fundamental Equation of Speech Recognition

• **Given:** an observation (ADC, FFT)
  \[ X = x_1, x_2, \ldots, x_T \]

• **Wanted:** the corresponding word sequence
  \[ W' = w_1, w_2, \ldots, w_m \]

• **Search:** the most likely word sequence \( W' \)

\[
W' = \arg \max_w P(W' | X) = \arg \max_w \frac{p(X | W) P(W)}{p(X)} = \arg \max_w p(X | W) P(W)
\]

\[ \uparrow \] (Bayes)

• \( p(X | W) \) = The *Acoustic Model (AM)*
  (how likely is it to observe \( X \) when \( W \) is spoken)

• \( P(W) \) = The *Language Model (LM)*
  (how likely is it that \( W \) is spoken a-priori)
Recognition Conceptually: AM and LM

• Let’s be pragmatic and keep AM and LM separate
• Simply count to get $P(W)$
• How to get an estimate for $p(X|W)$?
  • Take “spectrograms” and compare the recordings of two utterances using DTW
  • Accumulate cost along best path, using Hidden Markov Model (instead of 2nd utterance)

Hidden Markov Models

A “Hidden Markov Model” is a 5-tuple consisting of:

• $S$ The set of states $S=\{s_1,s_2,\ldots,s_n\}$, $n$ is the number of states
• $\pi$ The initial probability distribution, $\pi(s_i) = P(q_1 = s_i)$ probability of $s_i$ being the first state of a sequence
• $A$ The matrix of state transition probabilities: $1 \leq i, j \leq n$
  $A=(a_{ij})$ with $a_{ij} = P(q_{t+1} = s_j|q_t = s_i)$ going from state $s_i$ to $s_j$
• $B$ The set of emission probability distributions/ densities,
  $B=\{b_1,b_2,\ldots,b_n\}$ where $b_j(x)=P(o_t = x|q_t = s_j)$ is the probability of observing $x$ when the system is in state $s_j$
• $V$ Set of symbols, $v$ is the number of distinct symbols. The observable feature space can be discrete: $V=\{x_1,x_2,\ldots,x_v\}$, or continuous $V=\mathbb{R}^d$
Context-Dependent States

• Not complicated enough?
  • No – co-articulation influences the pronunciation
  • We have a generative model and want to be able to increase the model size
• Cluster ~50 phones into ~5000 context-dependent states:
  A(F,L)-b, etc.

• It’s ugly!

Duration Modeling

• Phonemes have a certain minimal duration in practice
• We could fit curves through them (Gamma, Poisson)
• But we use an exponential decay for states (which approximates phones with the “convoluted” curve)
• It’s ugly!

State-of-the-Art ASR

• Use a CD-HMM structure for the acoustic model
• Compile it into a Weighted FST together with the language model (typically a n-gram)
• Learn AM and LM using very different criteria
  • Decision trees, discriminative training at frame-level (or sequence criteria)
• Decode, evaluate with yet another criterion

• It’s ugly:

Let’s Take a Step Back

• We’re looking at a sequence transduction problem
• All the complexity is really man-made
  • Most of the time, the system is not in a discrete state, but in some transition
  • The inflation of states was created for Gaussians, not DNNs (but it works well for them, too)
• Maybe we don’t need all this
  • No need to explicitly segment or partition the training data
  • As long as the target sequence can be read off somewhere
Connectionist Temporal Classification

• Alex Graves (2006) described the “CTC” loss function
  • Sum over all possible frame alignments permitted for output sequence using Forward-Backward
  • Plays well with RNN or LSTM neural network models
• CTC introduces a new symbol: blank (-)
  • “Cannot decide with confidence given the current information”
  • “No output”, but do not confuse with silence
• Most of the time, the network will output (-)
  • Class imbalance not a problem in a connectionist architecture
  • As long as the target symbols appear from time to time

Observations

• Sparse representations (spikes) appear
• Any modeling unit can be used: phone, syllable, word
Problem with Best Path Decoding

Fig. 7.5 Problem with best path decoding. The single most probable path contains no labels, and best path decoding therefore outputs the labelling ‘blank’. However the combined probabilities of the paths corresponding to the labelling ‘A’ is greater.

[Graves 2012]

Enter WFST Decoding

• Turns out WFSTs can decode CTC-AMs well
  • FSTs can map $aaaa$, $aa$, $a$, … to “A”
• Resulting FSTs will typically be much smaller
  • $S = T \circ \min(\det(L \circ G))$
• Traditional HCLG
  • $S = \min(\det(H \circ \min(\det(C \circ \min(\det(L \circ G)))))$)
  • Don’t need HMM and Context FST any more, can replace by extremely simple Token FST
• Need to do some work on normalization of posteriors
  • Our experiments show it is most reliable to simply count the phones – which is also the simplest solution
Results on Read Speech

- WSJ: Baseline is unadapted Kaldi lMEL DNN
  - 20k vocabulary, quite optimized
  - Similar numbers for 5k vocab
  - No gains by going to CD models in our experiments (although Google reports gains)

<table>
<thead>
<tr>
<th>Task (WER)</th>
<th>Traditional</th>
<th>CTC</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>WSJ</td>
<td>7.1%</td>
<td>7.3%</td>
<td>CI Phones</td>
</tr>
<tr>
<td></td>
<td>8.9%</td>
<td></td>
<td>Characters</td>
</tr>
<tr>
<td></td>
<td>9.3%</td>
<td></td>
<td>Syllables</td>
</tr>
</tbody>
</table>

Detox of (Kaldi WSJ) Training Scripts

- Prepare Data
- Build LM
- Train GMM models
- Create CD Search Graph
- Decode Test Data
- Adapt
- Re-decode Test Data
- Lattice Rescoring (LM)
- Train DNN model

DNN ⟷ CTC
Results on Conversational Speech

- Switchboard – conversational telephony speech
  - One of the hardest benchmarks out there
  - Very sloppy speech in addition to hard channels

<table>
<thead>
<tr>
<th>Task</th>
<th>Trad.</th>
<th>CTC</th>
<th>Remark</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWB 300h</td>
<td>16.8%</td>
<td>13.5%</td>
<td>Unadapted IMEL features</td>
</tr>
<tr>
<td></td>
<td>15.1%</td>
<td></td>
<td>Adapted fMLLR DNN</td>
</tr>
</tbody>
</table>

- CTC relatively better on larger data sets (LSTM effect?)
- CTC training: twice that of feed-forward DNNs
- Decoding: 0.2x RT, using 30ms frame step, 25% memory

CTC Conclusions

- Drastic reduction in amount & complexity of code & fudge factors for ~ accuracy
  - Requires little Human supervision (but a bit more computation)
  - Good for the non-expert! Or Low resource languages?
  - ~50 states rather than 5000 ⟷ go back to dynamic decoding?
- Less explicit model assumptions; no number of states, context decision tree, initial alignment, etc. to decide
- Almost everything is a “deep learning” hyper-parameter
  - A very elegant end-to-end framework
  - Quite a bit more flexible than encoder-decoder models
Thank You!

Questions? ➔ fmetze@cs.cmu.edu


http://speechkitchen.org/
References