Multi-Task Learning for Acoustic Modeling

Sharing Data for ASR in Low Resource Languages

John Morgan

Applications Team (ATeam) Multilingual Computing and Analytics Branch

February 28, 2018

Abstract

Is ASR a solved problem?

The Problem:

- Commercial: $>$ 10000 hours of training data
- US Army: $<$ 10 hours of training data

Possible Solution:

Apply MTL to ASR to share representations

- Source – Channel Model
- Source
- Target
- Weighted Finite State Transducer Framework
- AI ASR State Of The Art
- 2 Modeling Problems
- Invariants
- Model
- Criteria
- End To End

What is ASR?

The Big Picture

Given acoustic evidence, recover words

The Acoustic Evidence
Waveform

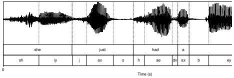

Figure: waveformA waveform of the sentence “She just had a baby”.

Acoustic Evidence

The Spectrogram

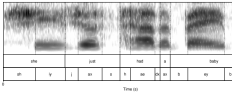

Figure: SpectrogramA spectrogram of “she just had a baby”.

Acoustic Evidence
MFCC

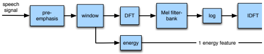

Figure: Mel Frequency Cepstral CoefficientsMFCC PROCESS

How Would YOU Do ASR?

- Linguists
- Electrical Engineers
- Computer Scientists
- Physicists
- Mathematicians
- Psychologists

AI and ML

Computer Scientist

Convert ASR into Classification Problem

- What are the classes?
- Sentences?
- Words?
- Morphemes?
- Syllables?
- Phonemes?
- Graphemes?
- Articulatory Features?

Goals of Classification
Generalization

Important Goal:

Classification of Previously Unseen Events

Unimportant Goal:

Classification as Memorization

Communication Channel

$$\text{Transmitter} \xrightarrow{\textit{NoisyChannel}} \text{Receiver}$$

Transmitter
2 parts

Mind
speech Producer

Mind

- Source of communication
- Generates concepts
- Specifies words
- Grammar combines words into phrases and sentences

Modulator
Speech Producer

Vocal organs:
Encode concepts into sound

Move air :
Lungs, Vocal Folds

Modify air stream :
Throat and Mouth

Produce fine grain meaning bearing differences :
Pharynx, Uvula, Tongue, Lips, Teeth, Palate

Receiver
2 parts

- Model transmitter
- Warning: AI is hungry!

Acoustic Processor
Linguistic Processor

Acoustic Processor

- Capture Sound
- Convert air movement into signals
- Digitize signal
- Quantize signals
- Extract energy at different frequencies
- Output vectors with characteristic information

linguistic Decoder
Traditional Model

- acoustic models
- Phonetic Unit Models
- Dynamic Phonetic Models
- lexical Models
- Syntax models
- hypothesis search algorithm

WARNING: AI is eating ASR software!

WFST

What are they?

- State Pairs
- Transitions
- Labels
- Weights
- Semi Ring
- Solid mathematical foundation

A WFST for each Component

acoustic Phonetic Unit:
Neural Network or GMM

Dynamic Phonetic:
Hidden Markov Model

Phonetic Context Dependency:
Decision Tree Clustering

Lexicon:]
Pronunciation or Phonological Model

Syntax:
Statistical N-gram Model

Traditional Pipeline

- acoustic vectors $\xrightarrow{acousticmodel}$ PDF over CD phones
- PDF over CD phones $\xrightarrow{HMM}$ CD phones
- CD phones $\xrightarrow{\text{Dicision tree}}$ CI phones
- CI phones $\xrightarrow{lexicon}$ words
- words $\xrightarrow{grammar}$ sentences

Decision Trees

- Allophones are phones expressed in context
- Triphones: context of previous and following phones
- How many triphones are there?
- At most n^3 where n is the number of phones
- $n = 43$ for English
- 43^3 too large to model accurately
- Decision trees are used to cluster triphones

Decoding

- run-time decoder task: combine and optimize transducers
- finds pronunciations in lexicon
- substitutes them into grammar
- Phonetic tree representations reduce path redundancy
- improve search efficiency
- identifies CD models for each CD phone
- substitutes them to create HMM transducer

Human Knowledge Versus Artificial Intelligence

How much ASR can AI eat?

- Recall traditional ASR components:
  - Acoustic Model
  - phonetic Model
  - Dynamic Phonetic Model
  - Context Dependency model
  - Lexical Model
  - Syntax Model
- AI can eat all of these components

End to End Systems

AI Eats The Whole ASR Pipeline!

- Grapheme-based ASR
- Input:

  Acoustic Waveform

  Output:

  Sentences
- 1 Neural Network Component

State of the Art ASR

Hybrid HMM DNN

- Components converted to FSTs
- Deep Neural Network Acoustic Models
- AI eats Acoustic Model component
- But all other componentsstill rely on human design

ARL Experiments

Training Data Source:
Tunisian Modern Standard Arabic

Collected by:
Dr. Steve LaRocca

Transcribed by:
MCAB A-Team

System	WER
CD GMM HMM DICT ngram	8.70
HYBRID DNN HMM DICT ngram	7.30
EESEN	27.90
mtl	4.61

ASR as Pattern Recognition

What are the patterns?

- Formants characterize phones
- Context modifies formant
- Formants themselves change over time
- Beginning, middle and end
- Caveat: Not all phones display formants

Problem 1
Phone Variation

- Speech is Dynamic
- Occurrs in a time series
- Feature representations sampled over short period
- Relevant Acoustic events occur at longer periods
- Important: movement of formant in time
- cue to identity of phone
- irrelevant: whether same events occurs sooner or later

Problem 2

Allophones

2 competing modeling goals

1. Model acoustic vector change
2. Model phone characteristic invariants

Goals

- 2 competing goals
  1. represent temporal relationships between acoustic events
  2. provide invariance under time translation
- Goal 1 solved for short intervals
- Goal 2 requires more work
- How does Machine Learning deal with this?

Solutions

2 Approaches

1. Precise time alignments and parameter tieing (HMM)
2. Convolution (nn)

What is a Neural Network?

Basics

- Layers of computing units
- Linear Transformation at each unit
- Weighted sum
- Nonlinearity at each unit

Representations

- Lower layers extract features
- Upper layers perform classification

Recurrent Neural Network

RNN

- Input: $\boldsymbol{x} = (x_1, \ldots, x_T)$
- computes hidden vectors: $\boldsymbol{h} = (h_1, \ldots, h_T)$
- outputs vectors: $\boldsymbol{y} = (y_1, \ldots, y_T)$
- iterates from $t = 1$ to T:

$$h_t = \mathcal{H}\left(W_{xh}x_t + W_{hh}h_{t-1} + b_h\right) \quad (1)$$

$$y_t = W_{hy}h_t + b_y \quad (2)$$

- weight matrices: W
- input-hidden weight matrix: W_{xh}
- bias vectors: b
- hidden bias vector: b_h
- hidden layer function: $\mathcal{H}$

RNN

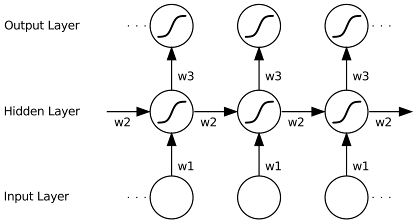

Figure: Recurrent Neural Network

Long-term Short Term Memory RNN

LSTM

$$i_t = \sigma\left(W_{xi}x_t + W_{hi}h_{t-1} + W_{ci}c_{t-1} + b_i\right) \quad (3)$$

$$f_t = \sigma\left(W_{xf}x_t + W_{hf}h_{t-1} + W_{cf}c_{t-1} + b_f\right) \quad (4)$$

$$c_t = f_t c_{t-1} + i_t \tanh\left(W_{xc}x_t + W_{hc}h_{t-1} + b_c\right) \quad (5)$$

$$o_t = \sigma\left(W_{xo}x_t + W_{ho}h_{t-1} + W_{co}c_t + b_o\right) \quad (6)$$

$$h_t = o_t \tanh(c_t) \quad (7)$$

- i: input gate
- f: forget gate
- o: output gate
- c: cell activation vectors

LSTM

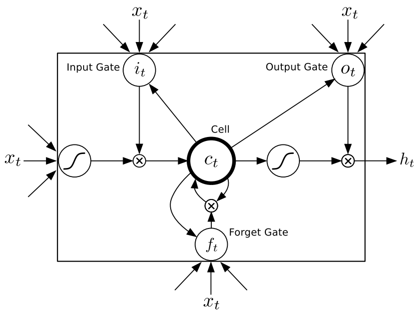

Figure: Long Short-term Memory Cell

Convolutional Neural Network
CNN

- In CNN layer
- set of filters convolved with input
- results in multiple output-maps
- one per filter
- followed by element-wise activation function $\sigma(\cdot)$
- layer performs operation on two axes
- spectrogram: time $\times$ frequency

CNN Equations

$$h_{i,j,k} = \sigma\left(\sum_{l=0}^{L-T}\sum_{m=0}^{M-F} x_{i+l,j+m} \cdot w_{l,m,k}\right), k = 1 \dots K \qquad (8)$$

- L dimensionality of time-axis
- M dimensionality of frequency-axis
- $T \times F$ is the size of filters
- k index of filter
- K number of filters
- $w_{l,m,k}$

More on CNNs

- CNN learned parameters followed by pooling
  summarises patches in each output map
  by computing average or maximum
  allows for invariance to shifts in location of feature
  full weight sharing (FWS)
  same filter applied across entire input space
  assumes feature occurs across entire input space
  valid assumption for temporal axis
  done in TDNN architecture

Time delayed neural network
TDNN

- Feed Forward
-
- initial transforms learnt on narrow contexts
- initial layers learn to detect features within narrow temporal contexts
- later layers operate on larger temporal context
- shift invariance: critically important property
- deeper layers process hidden activations from wider temporal context
- higher layers learn wider temporal relationships
- Each layer in TDNN operates at different temporal resolution

More ON TDNN and Invariants

- back-propagation:
- lower layers updated
- by gradient accumulated
- over all time steps of input temporal context
- lower layers forced to learn translation invariant feature transforms
- fully connected
- input: stack of frames
- replicated across different time-steps
- following layer takes as input stack of different time-steps of preceding layer
- replicated across different time-steps
- 1-d convolution

TDNN

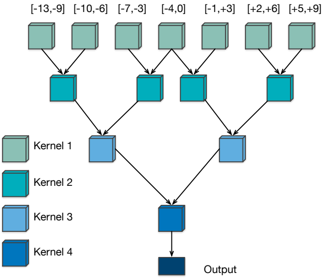

Figure: Time Delayed Neural NetworkBaseline TDNN Structure

CTC
Connectionist Temporal Classification

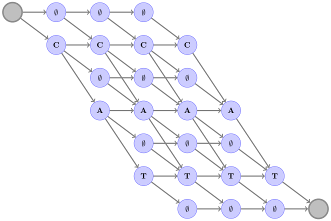

Figure: Connectionist Temporal ClassificationThe CTC graph which represents all the acceptable sequences of letters for the transcription “cat” over 6 frames.

EM

Expectation Maximization

MMI

Maximum Mutual Information

Back Propagation

- Supervised Method
- Input example
- Forward pass
- Compute Errors with output example
- Update weights in backward pass

Neural Network

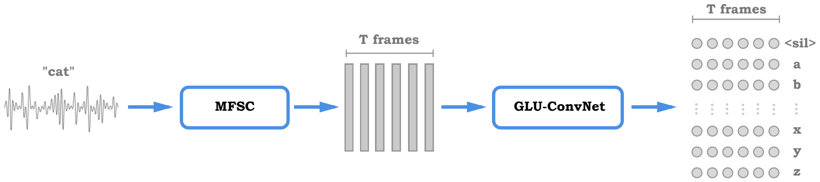

Figure: Overview of the acoustic model, which computes log-mel filterbanks which are fed to a TDNN. The TDNN outputs one score for each letter in the dictionary, and for each input feature frame. At inference time, these scores are fed to a decoder to form the most likely sequence of words. At training time, the scores are fed to the CTC criterion, which promotes sequences of letters leading to the transcription sequence (here “c a t”).

Criteria

Objective Functions

- CTC
- MMI

Original Definition

Multitask Learning is an approach to inductive transfer that improves generalization by using the domain information contained in the training signals of related tasks as an inductive bias.

It does this by learning tasks in parallel while using a shared representation; what is learned for each task can help other tasks be learned better.

RICH CARUANA 1997

What do We Share?

Pancake Stack

- ARL experimenting with 9 layers
- Share first 7 layers
- Layer 7 is Bottleneck
- Last 2 layers are language specific
- Layer 8 is affine
- Layer 9 is softmax

Summary

Traditional Statistical Method:

CD GMM HMM NGram LM

Hybrid HMM DNN NGram LM: State of the Art

AI is Eating ASR!: End to End

Outlook

- Unsupervised Learning
- Reinforcement Learning