Deep Speech Recognition

New-Generation Models & Methodology for Advancing Speech Technology and Information Processing

Li Deng

Microsoft Research, Redmond, USA

IEEE ChinaSIP Summer School, July 6, 2013

(including joint work with colleagues at MSR, U of Toronto, etc.)
Outline

PART I: Basics of Deep Learning (DL)
--- including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition

PART II: Deeper Substance of DL
--- including connections to other ML paradigms, examples of incorporating speech knowledge in DL architecture, and recent experiments in speech recognition
Deep Learning (DL) Basics

1. **Deep Learning** (aka Deep Structured Learning, Hierarchical Learning): a class of machine learning techniques, where many layers of information processing stages in hierarchical architectures are exploited for unsupervised feature learning and for pattern analysis/classification.

2. **Deep belief nets (DBN)**: probabilistic generative models composed of multiple layers of stochastic, hidden variables. The top two layers have undirected, symmetric connections between them. The lower layers receive top-down, directed connections from the layer above. (key: stacked RBMs; Hinton: *Science, 2006*).

3. **Boltzmann machine (BM)**: a network of symmetrically connected, neuron-like units that make stochastic decisions about whether to be on or off.

4. **Restricted Boltzmann machine (RBM)**: a special BM consisting of a layer of visible units and a layer of hidden units with no visible-visible or hidden-hidden connections. (Key: contrastive divergence learning).

5. **Deep neural nets (DNN, or “DBN” before Nov 2012)**: multilayer perceptrons with many hidden layers, whose weights are often initialized (pre-trained) using stacked RBMs or DBN (DBN-DNN) or discriminative pre-training.

6. **Deep auto-encoder**: a DNN whose output is the data input itself, often pre-trained with DBN (Deng/Hinton, interspeech 2010; Hinton, Science 2006).

7. **Distributed representation**: a representation of the observed data in such a way that they are modeled as being generated by the interactions of many hidden factors. A particular factor learned from configurations of other factors can often generalize well. Distributed representations form the basis of deep learning.
Distributed Representation

• A representation of the observed data in such a way that they are modeled as being generated by the interactions of many hidden factors. A particular factor learned from configurations of other factors can often generalize well. Distributed representations form the basis of deep learning.

• In contrast to the “atomic” or “localist” representations employed in traditional cognitive science (and in GMM-HMM speech recognition systems), a distributed representation is one in which “each entity is represented by a pattern of activity distributed over many computing element, and each computing element is involved in representing many different entities”. (Hinton, 1984)

• In GMM-HMM, each sound is associated with its own set of parameters. Not so for DNN-HMM.
More on “Deep Learning”

- **Definition 1**: A class of machine learning techniques that exploit many layers of non-linear information processing for supervised or unsupervised feature extraction and transformation, and for pattern analysis and classification.

- **Definition 2**: “A sub-field within machine learning that is based on algorithms for learning multiple levels of representation in order to model complex relationships among data. Higher-level features and concepts are thus defined in terms of lower-level ones, and such a hierarchy of features is called a deep architecture. Most of these models are based on unsupervised learning of representations.” (Wikipedia on “Deep Learning” around March 2012.)

- **Definition 3**: “A sub-field of machine learning that is based on learning several levels of representations, corresponding to a hierarchy of features or factors or concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts. Deep learning is part of a broader family of machine learning methods based on learning representations. An observation (e.g., an image) can be represented in many ways (e.g., a vector of pixels), but some representations make it easier to learn tasks of interest (e.g., is this the image of a human face?) from examples, and research in this area attempts to define what makes better representations and how to learn them.” see Wikipedia on “Deep Learning” as of this writing in February 2013; see [http://en.wikipedia.org/wiki/Deep_learning](http://en.wikipedia.org/wiki/Deep_learning).

- **Definition 4**: “Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence. Deep Learning is about learning multiple levels of representation and abstraction that help to make sense of data such as images, sound, and text.” See [https://github.com/lisa-lab/DeepLearningTutorials](https://github.com/lisa-lab/DeepLearningTutorials)
Deep Learning is a new term that is starting to appear in the data science/machine learning news. 

- Communications of the ACM just published a story on the topic, Deep Learning Comes of Age. 
- Deep Learning was named as one of the Top 10 Breakthrough Technologies of 2013 by MIT Technology Review. 
- Jeremy Howard, Chief Scientist at Kaggle declared Deep Learning – The Biggest Data Science Breakthrough of the Decade. 
- The New York Times published Scientists See Promise in Deep-Learning Programs

What is Deep Learning?

According to DeepLearning.net, the definition goes like this:

“Deep Learning is a new area of Machine Learning research, which has been introduced with the objective of moving Machine Learning closer to one of its original goals: Artificial Intelligence.”

Wikipedia provides the following definition:

“Deep learning is set of algorithms in machine learning that attempt to learn layered models of inputs, commonly neural networks. The layers in such models correspond to distinct levels of concepts, where higher-level concepts are defined from lower-level ones, and the same lower-level concepts can help to define many higher-level concepts.”

Deep Learning is sometimes referred to as deep neural networks since much of deep learning focuses on artificial neural networks. Artificial neural networks are a technique in computer science modelled after the connections (synapses) of neurons in the brain. Artificial neural networks, sometimes just called neural nets, have been around for about 50 years, but advances in computer processing power and storage are finally allowing neural nets to improve solutions for complex problems such as speech recognition, computer vision, and Natural Language Processing (NLP).
Useful Sites on Deep Learning

- http://www.cs.toronto.edu/~hinton/
- http://ufldl.stanford.edu/wiki/index.php/UFLDL_Tutorial (Andrew Ng’s group)
- http://deeplearning.net/reading-list/ (Bengio’s group)
- http://deeplearning.net/tutorial/
- http://deeplearning.net/deep-learning-research-groups-and-labs/
- Google+ Deep Learning community
Deep Learning Research Groups

Some labs and research groups that are actively working on deep learning:
University of Toronto - **Machine Learning Group** (Geoff Hinton, Rich Zemel, Ruslan Salakhutdinov, Brendan Frey, Radford Neal)
Université de Montréal - **Lisa Lab** (Yoshua Bengio, Pascal Vincent, Aaron Courville, Roland Memisevic)
New York University – **Yann LeCun**'s and **Rob Fergus**' group
Stanford University – **Andrew Ng**'s group
UBC – **Nando de Freitas**'s group
**Google Research** – Jeff Dean, Samy Bengio, Jason Weston, Marc’Aurelio Ranzato, Dumitru Erhan, Quoc Le et al
Microsoft Research – **Li Deng** et al
SUPSI – **IDSIA** (Schmidhuber’s group)
UC Berkeley – **Bruno Olshausen**'s group
University of Washington – **Pedro Domingos**' group
IDIAP Research Institute - **Ronan Collobert**'s group
University of California Merced – **Miguel A. Carreira-Perpinan**'s group
University of Helsinki - **Aapo Hyvärinen**'s Neuroinformatics group
Université de Sherbrooke – **Hugo Larochelle**'s group
University of Guelph – **Graham Taylor**'s group
University of Michigan – **Honglak Lee**'s group
Technical University of Berlin – **Klaus-Robert Muller**'s group
Baidu – **Kai Yu**'s group
Aalto University – **Juha Karhunen**'s group
U. Amsterdam – **Max Welling**'s group
U. California Irvine – **Pierre Baldi**'s group
Ghent University – **Benjamin Shrauwen**'s group
University of Tennessee – **Itamar Arel**'s group
IBM Research – **Brian Kingsbury** et al
University of Bonn – **Sven Behnke's** group
**Gatsby Unit** @ University College London – Maneesh Sahani, Yee-Whye Teh, Peter Dayan

Last modified on April 10, 2013, at 1:27 pm by Caglar Gulcehre
Deep Learning

With massive amounts of computational power, machines can now recognize objects and translate speech in real time. Artificial intelligence is finally getting smart.

Temporary Social Media

Messages that quickly self-destruct could enhance the privacy of online communications and make people freer to be spontaneous.

Prenatal DNA Sequencing

Reading the DNA of fetuses will be the next frontier of the genomic revolution. But do you really want to know about the genetic problems or musical aptitude of your unborn child?

Additive Manufacturing

Skeptical about 3-D printing? GE, the world’s largest manufacturer, is on the verge of using the technology to make jet parts.

Baxter: The Blue-Collar Robot

Rodney Brooks’s newest creation is easy to interact with, but the complex innovations behind the robot show just how hard it is to get along with people.

Memory Implants

A maverick neuroscientist believes he has deciphered the code by which the brain forms long-term memories. Next: testing a prosthetic implant for people suffering from long-term memory loss.

Smart Watches

The designers of the Pebble watch realized that a mobile phone is more useful if you don’t have to take it out of your pocket.

Ultra-Efficient Solar Power

Doubling the efficiency of a solar cell would completely change the economics of renewable energy. Nanotechnology just might make it possible.

Big Data from Cheap Phones

Collecting and analyzing information from simple cell phones can provide surprising insights into how people move about and behave — and even help us understand the spread of diseases.

Supergrids

A new high-power circuit breaker could finally make highly efficient DC power grids practical.
Plenary Keynote (9:50-10:40am, May 28)

Recent Developments in Deep Neural Networks

Geoffrey E. Hinton

Host: Li Deng
Scientists See Promise in Deep-Learning Programs

John Markoff
November 23, 2012

Rich Rashid in Tianjin, October, 25, 2012
Learning Curve: No Longer Just a Human Trait

By JOHN MARKOFF

Using an artificial intelligence technique inspired by theories about how the brain recognizes patterns, technology companies are reporting startling gains in fields as diverse as computer vision, speech recognition, and the identification of promising new molecules for designing drugs.

The advances have led to widespread enthusiasm among researchers who design software to perform human activities like seeing, listening and thinking. They offer the promise of machines that converse with humans and perform tasks like driving cars and working in factories, raising the specter of automated robots that could replace human workers.

The technology, called deep learning, has already been put to use in services like Apple’s Siri virtual personal assistant, which is based on Nuance Communications’ speech recognition service, and in Google’s Street View, which uses machine vision to identify specific addresses.

But what is new in recent months is the growing speed and accuracy of deep-learning programs, often called artificial neural networks or just “neural nets” for their resemblance to the neural connections in the brain.

“Tremendously numerous are the new results with deep-learning methods,” said Yann LeCun, a computer scientist at New York University who did pioneering work in the area.

Scientists See Promise in Deep-Learning Programs

A voice recognition program translated a speech given by Richard F. Rashid, Microsoft’s top scientist, into Mandarin Chinese.

A student team led by the computer scientist Geoffrey E. Hinton used deep-learning technology to design software.

Modern artificial neural networks are composed of an array of software components, divided into inputs, hidden layers and output. The arrays, or “activations,” can be “trained” by repeated exposure to patterns like images or sounds.

The techniques, added to the growing speed and power of modern computers, have led to rapid improvements in speech recognition, drug discovery and even computer vision.

Deep-learning systems have recently outperformed humans in certain limited recognition tasks.

Last year, for example, a program created by scientists at the Swiss AI Lab at the University of Lugano won a pattern recognition contest, beating out human participants for the first time.

These days, however, the technology has moved into new applications. A recent paper by scientists at Google revealed that a deep-learning system had achieved state-of-the-art performance in translating speech to text.

Such advances have raised concerns about the impact of artificial intelligence on jobs and society, with some experts predicting that the technology could replace human workers in many fields.

But others argue that the potential benefits of deep learning outweigh the risks.

In the end, the technology may offer a path to a future in which machines are able to perform a wide range of tasks, from recognizing speech to diagnosing diseases.
Acoustic Modeling and Deep Learning

June 19th, 2013
Vincent Vanhoucke

Thanks to Vincent for the permission of using his slides & discussions/corrections of information in some slides
Neural Networks for Speech in the 90's

- **Time-Delay Neural Networks**

- **Recurrent Neural Networks**

- **Hybrid Systems**

- **Bidirectional Recurrent Neural Networks**

- **Hierarchical Neural Networks**

- **TANDEM**
Speech Recognition

- DSP
- Feature Extraction
- Acoustic Model
- Language Model
Speech Recognition + Deep Neural Networks?
Speech Recognition + Deep Neural Networks!

3 months - 10%

word error rate relative reduction
Voice Search

Application Of Pretrained Deep Neural Networks To Large Vocabulary Speech Recognition, Navdeep Jaitly, Patrick Nguyen, Andrew Senior, Vincent Vanhoucke, Interspeech 2012.
Similar Stories across the Industry

Microsoft
Li Deng
Frank Seide
Dong Yu

IBM
Tara Sainath
Brian Kingsbury

Google
Andrew Senior
Georg Heigold
Marc’Aurelio Ranzato

University of Toronto
Geoff Hinton
George Dahl
Abdel-rahman Mohamed

And many others...

Some of Microsoft’s Stories..., Since 2009...
DL Took off in Speech Recognition from MSR

• Speech recognition: the first big (and real-world) success of deep learning
• From MSR (initial collaboration with Hinton et al., 2009-2010) and then to the entire speech industry
• Got out of “local optimum” of GMM-HMM stayed for many years
• Now used by Microsoft, Google, Apple/Nuance/IBM, Baidu, I FlyTech, etc. doing voice search in the cloud for smart phones (plus many other applications.)
Renaissance of Neural Network
--- “Deep Learning,” 2006

- Geoff Hinton invented Deep Belief Networks (DBN) to make neural net learning fast and effective; *Science, 2006*
  - Pre-train each layer from bottom up
  - Each pair of layers is an Restricted Boltzmann Machine (RBM)
  - Jointly fine-tune all layers using back-propagation
**Industry Scale Deep Learning**

**Started at MSR, 2009**

- 2008 NIPS: Geoff Hinton & Li Deng reconnected

- Earlier 2009: Initial exploration of DBN/DNN at MSR (image and speech)

- Later 2009: Proof of concept by Mohamed et al.; MSR & Hinton collaborated on applying DBN-DNN to speech feature coding (on spectrogram) and speech recognition

- Dec 2009: NIPS workshop (organizers: Deng, Yu, & Hinton)
Li Deng, Dong Yu, Geoffrey Hinton

Microsoft Research; Microsoft Research; University of Toronto

Deep Learning for Speech Recognition and Related Applications

7:30am - 6:30pm Saturday, December 12, 2009

Location: Hilton: Cheakamus

Abstract: Over the past 25 years or so, speech recognition technology has been dominated by a “shallow” architecture --- hidden Markov models (HMMs). Significant technological success has been achieved using complex and carefully engineered versions of HMMs. The next generation of the technology requires solutions to remain effective in the challenges under diversified deployment environments. These challenges, not previously addressed in the past, arise from the many types of variability present in the speech generation process. Overcoming these challenges is likely to require “deep” architectures with efficient learning algorithms. For speech recognition and related sequential data recognition applications, some attempts have been made in the past to develop computational architectures that are “deeper” than conventional HMMs, such as deep neural networks (DNNs).
Anecdote: Speechless summary presentation of the NIPS 2009 Workshop on Speech

Deep Learning for Speech Recognition and Related Applications

Li Deng, Dong Yu (Microsoft Research)
Geoffrey Hinton (University of Toronto)
They met in year 2009...
I was told you are smart.
Because I am deeper.
Can you understand speech as I do?
You bet! I can recognize phonemes.
That’s a nice first step!
What else are you looking for?
Recognizing noisy sentences spoken by unknown people.
Maybe we can work together.
Deep speech recognizer is born.

- Multi-objective
- Competitive Learning
- Hierarchical
- Deep Belief Net
- Conditional
- Recurrent
- Scalable
Continued at MSR, 2010, 2011...

-2010: slowly more people in MSR-speech joined DBN-DNN research

-August 2010: success of DNN in large-vocabulary speech recognition (voice search); paper in ICASSP-2011 (Dahl/Yu/Deng)

-Oct 2010: MSR/MSRA collaboration started on Switchboard task

The History of Automatic Speech Recognition Evaluations at NIST

NIST STT Benchmark Test History – May. ’09

Range of Human Error In Transcription
The History of Automatic Speech Recognition Evaluations at NIST

NIST STT Benchmark Test History – May. '09

WER(%) Range of Human Error In Transcription

1989 2011
Deep Learning has been the hottest topic in speech recognition in the last 2 years

- A few long-standing performance records were broken with deep learning methods
- Microsoft and Google have both deployed DL-based speech recognition system in their products
- Microsoft, Google, IBM, Nuance, AT&T, and all the major academic and industrial players in speech recognition have projects on deep learning

Deep Learning is the hottest topic in Computer Vision

- Feature engineering is the bread-and-butter of a large portion of the CV community, which creates some resistance to feature learning
- But the record holders on ImageNet and Semantic Segmentation are convolutional nets

Deep Learning is becoming hot in Natural Language Processing

Deep Learning/Feature Learning in Applied Mathematics
In Many Fields, Feature Learning Has Caused a Revolution (methods used in commercially deployed systems)

- **Speech Recognition I (late 1980s)**
  - Trained mid-level features with Gaussian mixtures (2-layer classifier)

- **Handwriting Recognition and OCR (late 1980s to mid 1990s)**
  - Supervised convolutional nets operating on pixels

- **Face & People Detection (early 1990s to mid 2000s)**
  - Haar features generation/selection (Viola-Jones 2001)

- **Object Recognition I (mid-to-late 2000s: Ponce, Schmid, Yu, YLC....)**
  - Trainable mid-level features (K-means or sparse coding)

- **Low-Res Object Recognition: road signs, house numbers (early 2010's)**
  - Supervised convolutional net operating on pixels

- **Speech Recognition II (circa 2011)**
  - Deep neural nets for acoustic modeling

- **Object Recognition III, Semantic Labeling (2012, Hinton, YLC,...)**
  - Supervised convolutional nets operating on pixels
Outline

PART I: Basics of Deep Learning (DL)
(including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition)

PART II: Deeper Substance of DL
(including connections to other ML paradigms, example of incorporating speech knowledge in DL architecture, and recent experiments in speech recognition)
Machine Learning Paradigms for Speech Recognition: An Overview

Li Deng, Fellow, IEEE, and Xiao Li, Member, IEEE

Abstract—Automatic Speech Recognition (ASR) has historically been a driving force behind many machine learning (ML) techniques, including the ubiquitously used hidden Markov model, discriminative learning, structured sequence learning, Bayesian learning, and adaptive learning. Moreover, ML can and occasionally does use ASR as a large-scale, realistic application to rigorously test the effectiveness of a given technique, and to inspire new problems arising from the inherently sequential and dynamic nature of speech. On the other hand, even though ASR is available commercially for some applications, it is largely an unsolved problem—for almost all applications, the performance of ASR is not on par with human performance. New insight from modern ML methodology shows great promise to advance the state-of-the-art in ASR technology. This overview article provides readers with an overview of modern ML techniques as utilized in the current and as relevant to future ASR research and systems. The intent is to foster further cross-pollination between the ML community to make assumptions about a problem, develop precise mathematical theories and algorithms to tackle the problem given those assumptions, but then evaluate on data sets that are relatively small and sometimes synthetic. ASR research, on the other hand, has been driven largely by rigorous empirical evaluations conducted on very large, standard corpora from real world. ASR researchers often found formal theoretical results and mathematical guarantees from ML of less use in preliminary work. Hence they tend to pay less attention to these results than perhaps they should, possibly missing insight and guidance provided by the ML theories and formal frameworks even if the complex ASR tasks are often beyond the current state-of-the-art in ML.

This overview article is intended to provide readers of IEEE TRANSACTIONS ON AUDIO, SPEECH, AND LANGUAGE
PART I: Basics of Deep Learning (DL) (including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition)

PART II: Deeper Substance of DL

--- Technical introduction: RBM, DBN, DNN, CNN, RNN
--- Advanced: 2 examples of incorporating domain knowledge (speech) into DL architectures
--- Novel DL architectures and recent experiments
Deep Neural Networks for Acoustic Modeling in Speech Recognition

The shared views of four research groups
First generation neural networks

- Perceptrons (~1960) used a layer of hand-coded features and tried to recognize objects by learning how to weight these features.
  - There was a neat learning algorithm for adjusting the weights.
  - But perceptrons are fundamentally limited in what they can learn to do.

(Slide from Hinton)
Support Vector Machine is a perceptron

- Vapnik and his co-workers developed a very clever type of perceptron called a Support Vector Machine.
  - Instead of hand-coding the layer of non-adaptive features, each training example is used to create a new feature using a fixed recipe.
    - The feature computes how similar a test example is to that training example.
  - Then a clever optimization technique is used to select the best subset of the features and to decide how to weight each feature when classifying a test case.
    - But it's just a perceptron and has all the same limitations.
- In the 1990’s, many researchers abandoned neural networks with multiple adaptive hidden layers because Support Vector Machines worked better.

(Slide modified from Hinton)
Second generation neural networks (~1985)

Back-propagate error signal to get derivatives for learning

Compare outputs with correct answer to get error signal

(Slide from Hinton)
What is wrong with back-propagation?
(a plausible story, but false; Hinton ICASSP-2013)

• It requires labeled training data.
  – Almost all data is unlabeled.
• The learning time does not scale well
  – It is very slow in networks with multiple hidden layers.
• It can get stuck in poor local optima.
  – These are often quite good, but for deep nets they are far from optimal
• Deep learning (partially) overcomes these difficulties by using undirected graphical model
What was actually wrong with back-propagation?

- We didn’t collect enough labeled data.
- We didn’t have fast enough computers.
- We didn’t initialize the weights correctly

- If we fix these three problems, it works really well.

(Hinton: ICASSP-2013)
What has happened since 1985

• Labeled datasets got much bigger.

• Computers got much faster.

• We found better ways to initialize the weights of a deep net using unlabeled data.

• As a result, deep neural networks are now state-of-the-art for tasks like object recognition or acoustic modeling for speech recognition

(Hinton: ICASSP-2013)
Initializing the weights in a deep neural net using unlabeled data

• This was historically important in overcoming the belief that deep neural networks could not be trained effectively. (Hinton: ICASSP-2013)
  – This was a very strong belief.
  – It prevented papers being published in good conferences and journals.

• For the tasks with small amounts of labeled training data, such initialization is still very useful
Restricted Boltzmann Machines (RBM)

- We restrict the connectivity to make learning easier.
  - Only one layer of hidden units.
  - No connections between hidden units.
- In an RBM, the hidden units are conditionally independent given the visible states.
- So we can quickly get an unbiased sample from the posterior distribution when given a data-vector.
RBM: Weights $\rightarrow$ Energies $\rightarrow$ Probabilities

- Joint distribution $p(v, h; \theta)$ is defined in terms of an energy function $E(v, h; \theta)$

$$p(v, h; \theta) = \frac{\exp(-E(v, h; \theta))}{Z}$$

- For a Bernoulli-Bernoulli RBM

$$E(v, h; \theta) = - \sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j - \sum_{i=1}^{V} b_i v_i - \sum_{j=1}^{H} a_j h_j$$

- For a Gaussian-Bernoulli RBM

$$E(v, h; \theta) = - \sum_{i=1}^{V} \sum_{j=1}^{H} w_{ij} v_i h_j + \frac{1}{2} \sum_{i=1}^{V} (v_i - b_i)^2 - \sum_{j=1}^{H} a_j h_j$$

- $p(v, h; \theta)$ $\rightarrow$ generative model!
Restricted Boltzmann Machine (RBM)

- Conditional probabilities are very easy to calculate
- For a Bernoulli-Bernoulli RBM
  \[
  p(h_j = 1|v; \theta) = \sigma \left( \sum_{i=1}^{V} w_{ij} v_i + a_j \right)
  \]

- For a Gaussian-Bernoulli RBM
  \[
  p(h_j = 1|v; \theta) = \sigma \left( \sum_{i=1}^{V} w_{ij} v_i + a_j \right)
  \]
  \[
  p(v_i|h; \theta) = N \left( \sum_{j=1}^{H} w_{ij} h_j + b_i, 1 \right)
  \]

- Proof next page. (This is a “neural net” with stochastic units rather than the deterministic MLP that you may be more familiar with)
\[ P(h|v) = \frac{e^{-E(v,h)}}{\sum_{\tilde{h}} e^{-E(v,\tilde{h})}} = \frac{e^{b^Tv + c^T\tilde{h} + v^TWh}}{\sum_{\tilde{h}} e^{b^Tv + c^T\tilde{h} + v^TWh}} = \frac{e^{c^T\tilde{h} + v^TWh}}{\sum_{\tilde{h}} e^{c^T\tilde{h} + v^TWh}} = \frac{\prod_i e^{c_i h_i + v^T W_{*,i} h_i}}{\sum_{\tilde{h}_1} \cdots \sum_{\tilde{h}_N} \prod_i e^{c_i \tilde{h}_i + v^T W_{*,i} \tilde{h}_i}} = \frac{\prod_i e^{-\gamma_i(v, h_i)}}{\sum_{\tilde{h}_1} \cdots \sum_{\tilde{h}_N} \prod_i e^{-\gamma_i(v, \tilde{h}_i)}} = \frac{\prod_i e^{-\gamma_i(v, h_i)}}{\prod_i e^{-\gamma_i(v, \tilde{h}_i)}} = \prod_i \frac{e^{-\gamma_i(v, h_i)}}{\sum_{\tilde{h}_i} e^{-\gamma_i(v, \tilde{h}_i)}} = \prod_i P(h_i | v). \tag{5} \]

Since the \( h_i \in \{0, 1\} \), the sum in the denominator of equation (5) has only two terms and thus

\[ P(h_i = 1|v) = \frac{e^{-\gamma_i(v, 1)}}{e^{-\gamma_i(v, 1)} + e^{-\gamma_i(v, 0)}} = \sigma(c_i + v^T W_{*,i}), \]

yielding

\[ P(h = 1|v) = \sigma(c + v^T W), \tag{7} \]
Maximum likelihood learning for RBM

Start with a training vector on the visible units.

Then alternate between updating all the hidden units in parallel and updating all the visible units in parallel.

\[
\frac{\partial \log p(v)}{\partial w_{ij}} = <v_i h_j>_0 - <v_i h_j>^\infty
\]
Training RBMs

• \( \Delta w_{ij} = \langle v_i h_j \rangle_{\text{data}} - \langle v_i h_j \rangle_{\text{model}} \)

• Approximate \( \langle v_i h_j \rangle_{\text{model}} \)
  i. Initialize \( v_0 \) at data
  ii. Sample \( h_0 \sim p(h|v_0) \)
  iii. Sample \( v_1 \sim p(v|h_0) \)
  iv. Sample \( h_1 \sim p(h|v_1) \)
  v. Call \((v_1, h_1)\) a sample from the model.

• \((v_\infty, h_\infty)\) is a true sample from the model.
  \((v_1, h_1)\) is a very rough estimate but worked

• Contrastive divergence algorithm (CD)
RBM versus GMM

• Gaussian Mixture Model
  – Local representation
  – (In practice,) data vector explained by only a single Gaussian
  – Tend to over-fit

• Bernoulli-Gaussian RBM
  – Distributed representation, very powerful
  – Product of Gaussians
  – Tend to under-fit
Building a Deep Network

• This is the main reason why RBM’s are interesting (as a building block)

• First train a layer of hidden units that receive input directly from the data (image, speech, coded text, etc).

• Then treat the activations of hidden units (the trained “features”) as if they were “data” and learn features of features in a second hidden layer.

• It can be proved that each time we add another layer of features we improve a variational lower bound on the log probability of the training data.
  – The proof is complicated (Hinton et al, 2006)
  – Based on an equivalence between an RBM and a deep directed model
Deep Belief Net (DBN) & Deep Neural Net (DNN)

- DBN: Undirected at top two layers which is an RBM; directed Bayes net (top-down) at lower layers (good for *synthesis and recognition*)
- DNN: Multi-layer perceptron (bottom up) + unsupervised pre-training w. RBM weights (good for *recognition only*)

![Diagram showing a deep belief net and a deep neural net with labeled layers and visible layers.](image-url)
First train a stack of three models each of which has one hidden layer. Each model in the stack treats the hidden variables of the previous model as data.

Then compose them into a single Deep Belief Network.

Then add outputs and train the DNN with backprop.

Hinton, Deng, Yu, Mohamed, Dahl... etc. IEEE Sig. Proc. Mag. (Nov 2012)
Quiz Questions

1. DBN & DNN: which one is generative? Which one is discriminative?
2. How can a generative model be used for recognition? (Bayes rule as for HMM speech recognition)
3. How does DBN do synthesis?
4. How does DBN do recognition?
5. How does DNN do recognition?
6. For recognition, is RBN or DNN better?
7. What is the difference between DBN and Dynamic Bayes Net (a.k.a. “DBN”)?
The Answer to Quiz Question 3:

- To generate data:
  1. Get an equilibrium sample from the top-level RBM by performing alternating Gibbs sampling for a long time.
  2. Perform a top-down pass to get states for all the other layers.

So the lower level bottom-up connections are **not** part of the generative model. They are just used for inference.
Samples generated by letting the associative memory run with one label clamped. There are 1000 iterations of alternating Gibbs sampling between samples (example from Hinton).
The top two layers form an associative memory whose energy landscape models the low dimensional manifolds of the digits.

The energy valleys have names.

The model learns to generate combinations of labels and images.

**To perform recognition** we start with a neutral state of the label units and do an up-pass from the image followed by a few iterations of the top-level associative memory ---> probability of that digit label; then repeat for all digit labels; then compare.

(slide modified from Hinton)
DBN & DNN: Fine-tuning for discrimination

• First learn one layer at a time greedily.
• Then treat this as “pre-training” that finds a good initial set of weights which can be fine-tuned by a local search procedure.
• For DBN: Contrastive wake-sleep (see Hinton’s)
• For DNN: Back-propagation
  – This overcomes many of the limitations of standard backpropagation (if you do not have large labeled training data).
Fine Tuning DNN after pre-training: Optimization view

• Stacking RBMs one layer at a time scales well to really big networks

• Do not start back-propagation until sensible feature detectors are found by RBM pre-training that should already be very helpful for the discrimination task.

• Back-propagation only needs to perform a local search from a sensible starting point.
Fine Tuning DNN after pre-training:
Regularization view

- Information in the pre-trained weights comes from modeling the distribution of input vectors in an “unsupervised” manner.
- The input vectors generally contain a lot more information than the labels.
- The precious information in the labels is only used for the final fine-tuning.
- The fine-tuning only modifies the features slightly to get the category boundaries right. No need to discover “features”.
- Hence less prone to overfit (unlike the old neural nets with typically random weight initialization)
- This type of backpropagation works well even if most of the training data is unlabeled.
- The unlabeled data is still very useful for discovering good features.
DNN with class posteriors (not DBN)

- As stacked RBMs
- Pre-train each layer from bottom up by considering each pair of layers as an RBM.
- Transform the output of the last hidden layer into a multinomial distribution using the softmax operation

\[
p(l = k | h; \theta) = \frac{\exp \left( \sum_{i=1}^{H} \lambda_{ik} h_i + a_k \right)}{Z(h)}
\]

- Why? Needed for (ASR) sequence recognition (not needed for static or frame-level recognition)
- For ASR: Use GMM-HMM forced alignment to get the label for the final layer when using frame-level training.
- Jointly fine-tune all layers using back-propagation algorithm.
Theoretical Insights of DBN

1. Restricted Boltzmann Machine (RBM) as the building block of DBN
2. RBM can be viewed as infinitely deep directed Bayesian/Belief network with tied weights over layers
3. Complementary prior (Hinton et. al. 2006)
4. Regularization vs. optimization
5. Generative vs. discriminative
6. Theory is still weak
The current wisdom on unsupervised pre-training

- Pre-training achieves two things:
  - It makes optimization easier.
  - It reduces overfitting.
- We now know more about how to initialize weights sensibly by hand.
  - So unsupervised pre-training is not required to make the optimization work.
- Unsupervised pre-training is still very effective at preventing over-fitting when labeled data is scarce.
  - It is not needed when labeled data is abundant.

Hinton: ICASSP-2013
DNN-HMM
(replacing GMM only; longer MFCC/filter-back windows w. no transformation)

Model tied triphone states directly

Many layers of nonlinear feature transformation + SoftMax
CD-DNN-HMM: Architecture
(Shallow) GMM-HMM

• Model frames of acoustic data with two stochastic processes:
  – A hidden Markov process to model state transition
  – A Gaussian mixture model to generate observations

• Train with maximum likelihood criterion using EM followed by discriminative training (e.g. MPE)
Voice Search with DNN-HMM

- First attempt in using deep models for large vocabulary speech recognition (summer 2010)
- Published in ICASSP-2011 & 2012 Special issue of T-ASLP:

Context-Dependent Pre-Trained Deep Neural Networks for Large-Vocabulary Speech Recognition

George E. Dahl, Dong Yu, Senior Member, IEEE, Li Deng, Fellow, IEEE, and Alex Acero, Fellow, IEEE

Abstract—We propose a novel context-dependent (CD) model for large-vocabulary speech recognition (LVSP) that leverages recent advances in deep neural network (DNN) models. The CD-DNN model simultaneously learns context-dependent models for all prosodic states, and combines them together with a large-vocabulary acoustic model. This model has been shown to produce better speech recognition performance than conventional LVSP models on both the 2010 and 2011 NIST SRE data sets.
MSR Key Innovations (2009-2013)

• Scale the success to large industrial speech tasks
  – Grew output neurons from context-independent phones (100-200) to context-dependent ones (9k-32k)
  – Motivated initially by saving huge MSFT investment in huge speech decoder software infrastructure (e.g. Entropic acquisition)
  – Extremely fast decoder
  – Developed novel deep learning architectures & techniques: DCN/DSN, tensor-DSN, kernel-DCN, tensor-DNN, etc.

• Engineering for large systems:
  – Expertise in DNN and speech recognition
  – Close collaboration among MSRR, MSRA, & speech product teams (Deng, Yu, Seide, Gang Li, Jinyu Li, Jui-Ting Huang, Yifan Gong, etc.)
Some Recent News by Reporters

• **DNN Research Improves Bing Voice Search** (very fast decoder)

• **How technology can bridge language gaps: Speech-to-speech translation promises to help connect our world**

• **Scientists See Promise in Deep-Learning Programs** (NYT: speech to speech)

• **Microsoft Research shows a promising new breakthrough in speech translation technology**

• **Bing Makes Voice Recognition on Windows Phone More Accurate and Twice as Fast**

• **Microsoft revs speedier, smarter speech recognition for phones**
PART II: Deeper Substance of DL

--- Technical introduction: RBM, DBN, DNN, DNN-HMM, CNN, RNN

--- Examples of incorporating domain knowledge (about speech) into DL architectures
1. Hidden/articulatory Speech dynamics into RNN
2. Speech invariance/class-discrim.into deep-CNN

--- A few new, promising DL architectures
Outline

PART I: Basics of Deep Learning (DL)
(including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition)

PART II: Deeper Substance of DL
---Example 1: incorporating domain knowledge: Hidden/Deep Dynamics in Human Speech
Deep/Dynamic Models are Natural for Speech

• Hierarchical structure in human speech generation
  – Global concept/semantics formation
  – Word sequence formation / prosodic planning
  – Phonological encoding (phones, distinctive features)
  – Phonetic encoding (motor commands, articulatory targets)
  – Articulatory dynamics
  – Acoustic dynamics (clean speech)
  – Distorted speech
  – Interactions between speakers and listener/machine

• Hierarchical structure in human speech perception
  – Cochlear nonlinear spectral analysis
  – Attribute/phonological-feature detection at higher level(s)
  – Phonemic and syllabic detection at still higher level(s)
  – Word and sequence detection
  – Syntactic analysis and semantic understanding at deeper auditory cortex
Production & Perception: Closed-Loop Chain

Speech Acoustics in closed-loop chain
message

motor/articulators

distortion factors & feedback to articulation

distortion-free acoustics

distorted acoustics

targets

articulation

(Deep) Dynamic Bayesian Net
A MULTIMODAL VARIATIONAL APPROACH TO LEARNING AND INFECTION IN SWITCHING STATE SPACE MODELS

Leo J. Lee\textsuperscript{1,2}, Hagai Attias\textsuperscript{2}, Li Deng\textsuperscript{2} and Paul Fieguth\textsuperscript{3}

University of Waterloo
\textsuperscript{1}Electrical & Computer Engineering
\textsuperscript{3}Systems Design Engineering
Waterloo, ON, N2L 3G1
Canada

\textsuperscript{2}Microsoft Corporation
Microsoft Research
One Microsoft Way
Redmond, WA 98052-6339
USA

ABSTRACT
An important general model for discrete-time signal processing is the switching state space (SSS) model, which generalizes the hidden Markov model and the Gaussian state space model. Inference and parameter estimation in this model are known to be computationally intractable. This paper presents a powerful new approximation to the SSS model. The approximation is based on a variational technique that preserves the multimodal nature of the continuous state posterior distribution. Furthermore, by incorporating a windowing technique, the resulting EM algorithm has complexity that is just linear in the length of the time series. An alternative Viterbi decoding with frame-based likelihood is also presented which is crucial for the speech application that originally motivates this work. Our experiments focus on demonstrating the effectiveness of the algorithm by extensive simulations. A typical example in speech processing is also included to show the potential of this approach for practical applications.

Fig. 1. The model (a) and the variational posterior (b) represented as Bayesian networks.
Generative Modeling

Fig. 5. Tracking VTRs for a speech sentence.
Structured Speech Modeling

Li Deng, Fellow, IEEE, Dong Yu, Member, IEEE, and Alex Acero, Fellow, IEEE

Abstract—Modeling dynamic structure of speech is a novel paradigm in speech recognition research within the generative modeling framework, and it offers a potential to overcome make it indistinguishable with human–human verbal interaction, at present, when users interact with any existing speech recog-

<table>
<thead>
<tr>
<th>Method</th>
<th>PER</th>
</tr>
</thead>
<tbody>
<tr>
<td>CD-HMM [26]</td>
<td>27.3%</td>
</tr>
<tr>
<td>Augmented conditional Random Fields [26]</td>
<td>26.6%</td>
</tr>
<tr>
<td>Randomly initialized recurrent Neural Nets [27]</td>
<td>26.1%</td>
</tr>
<tr>
<td>Bayesian Triphone GMM-HMM [28]</td>
<td>25.6%</td>
</tr>
<tr>
<td><strong>Monophone HTMs [29]</strong></td>
<td>24.8%</td>
</tr>
<tr>
<td>Heterogeneous Classifiers [30]</td>
<td>24.4%</td>
</tr>
<tr>
<td>Monophone randomly initialized DNNs (6 layers)[13]</td>
<td>23.4%</td>
</tr>
<tr>
<td>Monophone DBN-DNNs (6 layers) [13]</td>
<td>22.4%</td>
</tr>
<tr>
<td>Monophone DBN-DNNs with MMI training [31]</td>
<td>22.1%</td>
</tr>
<tr>
<td>Triphone GMM-HMMs discriminatively trained w/ BMMI [32]</td>
<td>21.7%</td>
</tr>
<tr>
<td>Monophone DBN-DNNs on fbank (8 layers) [13]</td>
<td>20.7%</td>
</tr>
<tr>
<td>Monophone mcRBM-DBN-DNNs on fbank (5 layers) [33]</td>
<td>20.5%</td>
</tr>
<tr>
<td>Monophone convolutional DNNs on fbank (3 layers) [34]</td>
<td>20.0%</td>
</tr>
</tbody>
</table>

Fig. 1. Illustration of time-varying coarticulatory vectors $a_k$'s for a TIMIT utterance. See text for detailed explanations.
(Hidden) Dynamic Models

- Many types of dynamic models since 90’s
- Good survey article on earlier work (Ostendorf et al. 1996)
- Hidden Dynamic Models (HDM/HTM) since late 90’s
- This is “deep” generative model with >2 layers
- Pros and cons of different models
- All intended to create more realistic speech models “deeper” than HMM for speech recognition
- But with different assumptions on speech dynamics
- How to embed such dynamic properties into the DNN framework?
DBN (Deep) vs. DBN* (Dynamic)

• DBN-DNN (2009-2012) vs. HDM/HTM (1990’s-2006)
• Distributed vs. local representations
• Massive vs. parsimonious parameters
• Product of experts vs. mixture of experts
• Generative-discriminative hybrid vs. generative models
• Longer windows vs. shorter windows

• A neat way of “pre-training” RNN by HDM and then “fine-tuning” RNN by backprop (non-trivial gradient derivation and computation)
Building Dynamics into Deep Recurrent Models

• (Deep) recurrent neural networks for ASR: both acoustic and language modeling
  – generic temporal dependency
  – lack of constraints provided by hidden speech dynamics
  – Information redundancy & inconsistency: long windows for each “frame” introducing undesirable “noise”
    – Need to go beyond unconstrained temporal dependence and ESN (while easier to learn)

• An active and exciting research area to work on
PART I: Basics of Deep Learning (DL)
(including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition)

PART II: Deeper Substance of DL

---Example 2: incorporating domain knowledge:
Speech invariance/variability vs.
Phonetic discrimination in Conv. NN
A Deep Convolutional Neural Net Using Heterogeneous Pooling to Tradeoff Acoustic Invariance w. Phonetic Distinction

Li Deng, Ossama Abdel-Hamid, and Dong Yu
Microsoft Research, Redmond
York University, Toronto

ICASSP, May 28, 2013
**Background: Convolutional Nets (CNN)**

- **Convolution layer** (w. tying weights): a.k.s. “time/spatial”-invariant FIR filter
- Gives maps of replicated features; neural activities “equivariant” to translation
- **Pooling layer** (max of neighboring units in conv layer): Data reduction & some degree of invariance.
- 2D deep-CNN: State of the art in object recognition (Krizhevsky et al., 2012; LeCun et al.; Ciresan et al.)
Background: Convolutional Nets (CNN)

- **Difficulties of CNN:**
  - 2D Images: Information lost about the precise positions of parts \(\rightarrow\) object confusion
  - 2D Speech spectrogram: spectral-temporal information lost about phonetic distinction
  - E.g. 1-D CNN along freq axis (Abdel-Hamid et al., 2012): (TDNN & TF-trajectory CNN)
    - local weight sharing + max pooling over a range \(\rightarrow\) invariance to freq shift
      (VTL normalization)
    - But if freq range too small \(\rightarrow\) not enough VTL normalization (acoustic invariance)
      too large \(\rightarrow\) formant patterns of a sound shift \(\rightarrow\) phone confusion

- **Solutions for image recognition:** (tried some for speech, no clear success)
  - Transforming autoencoder (Hinton et al., 2011)
  - Tiled CNN (Le et al., 2012)
  - Deconvolutional nets (Zeiler et al., 2011)

- **A good solution for speech recognition** is surprisingly simple
Main Ideas of This Paper

• Bring “confusion” into designing CNN intended for “invariance”
• Exploit the knowledge of how increasing the degree of invariance (to shift along frequency-axis) may reduce phonetic discrimination
• (Kai Yu this morning: Spatial Pyramid Matching for vision)
• Examine/predict how the pooling size (i.e. range of freq-shift invariance) affects phonetic classification errors
  – Theoretic guidance possible; e.g.
  – Phonetic reduction (in casual, conversation speech) shrinks formant space → tradeoff towards “distinction” from “invariance” → smaller pooling size
• Use of many feature maps (afforded by CNN weight tying)
• Different pooling sizes (heterogeneous pooling) for different feature maps
  – Design and use a distribution of pooling sizes and randomly sample it.
  – Special case: use a fixed pooling size, optimized by validation or predicted by acoustic-phonetic “theory” (consistent for TIMIT; not as good as HP)
CNN with a Fixed Pooling Size
(a special case of HP-CNN w. P=3)

(limited weight sharing)
$HP$-CNN $[P_1, N_1 | P_2, N_2 | ... | P_m, N_m]$

$P_1 = 2; P_2 = 3$
Regularizing HP-CNN with “Dropout”

• A variant of the Dropout method for DNN (Hinton et al., 2012)

• Dropout in both conv and pooling layers of CNN is helpful, in addition to fully-connected DNN layers

• Dropout in the input layer (filterbanks) is not helpful

• In TIMIT, for CNN w. N=100 feature maps, and DNN hid=2000, the best dropout rate=0.2

• With dropout rate=0.5 & DNN hid=5000, error rate increases
Standard TIMIT Task: Core Testset Results

<table>
<thead>
<tr>
<th>Systems</th>
<th>Phone Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN (fully-connected 5 layers)</td>
<td>22.3%</td>
</tr>
<tr>
<td>CNN-DNN; P=1 (2 CNN &amp; 3 DNN layers)</td>
<td>21.8%</td>
</tr>
<tr>
<td>CNN-DNN; P=12</td>
<td>20.8%</td>
</tr>
<tr>
<td>CNN-DNN; P=6 (fixed P, optimal)</td>
<td>20.4%</td>
</tr>
<tr>
<td>CNN-DNN; P=6 (add dropout)</td>
<td>19.9%</td>
</tr>
<tr>
<td>CNN-DNN; P=1:m (HP, m=12)</td>
<td>19.3%</td>
</tr>
<tr>
<td>CNN-DNN; above (add dropout)</td>
<td>18.7%</td>
</tr>
</tbody>
</table>

- CNN-DNN; P=1 → equivariance: 21.8% > 20.4% (invariance w. fixed, optimal pooling size=6)
- CNN-DNN; P=1:12 → Heterogeneous pooling: 19.3% < 20.4%
- Dropout is always helpful (thanks Geoff!): 18.7% < 19.3%; 19.9% < 20.4%

18.7% WAS the record low error rate on this standard task (until this morning by LSTM-RNN)
Training Criterion: Cross-Entropy

Correlation of error rate & X-entropy

\[ y = 0.2399x + 0.0942 \]

\[ R^2 = 0.9893 \]
Effects of Training Epochs (Time)

- Each training epoch (1.12M frames in TIMIT) → 2 hrs of CPU time on my PC (no GPU)

% Accuracy for dev and core test sets

% Frame error rate for training set
## Confusion Matrix (Hresults)

| a | b | c | d | e | f | g | h | i | j | k | l | m | n | o | p | q | r | s | t | u | v | w | x | y | z |
| ae | 103 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| ne | 8 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| en | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| en | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| sh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| nh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |
| dh | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 | 0 |

---

**Overall Results**

**Confusion Matrix**
Recognition Error Breakdown

- Percentage Phone Errors for Each of 39 Classes
- Comparing five different phone recognizers: Effects of HP and dropout
HP-CNN for Large Vocabulary Speech Recognition

- On a voice search task
- Dozens of hours of labeled training data
- Not yet optimized the single fixed pooling size
- No change (yet) from the TIMIT system: m=12, N=104, N₁, N₂, N₃, N₄, N₅, etc.

<table>
<thead>
<tr>
<th>Systems</th>
<th>Word Error Rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN baseline (fully-connected 5 layers)</td>
<td>32.4%</td>
</tr>
<tr>
<td>CNN-DNN; P=1 (2 CNN &amp; 3 DNN layers)</td>
<td>32.0%</td>
</tr>
<tr>
<td>CNN-DNN; P=1:m (HP, m=12) same distribution as in TIMIT experiment</td>
<td>30.1%</td>
</tr>
</tbody>
</table>
Conclusions (of ICASSP-2013 paper)

- Effectiveness of convolution/pooling in image recognition can be ported to speech recognition
- Esp. when speech-specific properties incorporated
- Bring “confusion” into designing CNN intended for “invariance”
- Tradeoffs can be made by adjusting pooling size in CNN
- Optimizing (a single) pooling size provides desirable tradeoffs
- A much better way is to use varying pooling sizes for different feature maps (hence HP) → record-low TIMIT error rate
- HP-CNN of this paper is limited to convolution along freq-axis
- Can be extended to spectro-temporal patches in spectrograms
- Analogy: Object parts (image) ↔ formant trajectories (speech)
- This is exciting time to integrate speech knowledge into deep learning models
PART II: Deeper Substance of DL

--- Technical introduction: RBM, DBN, DNN, DNN-HMM, CNN, RNN

--- Examples of incorporating domain knowledge (about speech) into DL architectures
1. Hidden/articulatory Speech dynamics into RNN
2. Speech invariance/class-discrim into deep-CNN

--- A few new, promising DL architectures
Deep Convex Networks

• A simple approach to build a deep model using only convex optimization techniques.
• Successfully ‘convexifying’ the problem is an interesting line of research.
• Very competitive and fast to train.
• So far, best performance still obtained with non-convex fine tuning and many more layers than DNNs.

Deep Tensor Networks

- One example of several attempts at incorporating multiplicative nodes into deep networks.
- Very promising area of research attempting to factor out ‘style’ (speaker, environment) from ‘content’ (phonetic label) using multiplicative gating interactions.
Tensor Deep Stacking Networks

Brian Hutchinson, Student Member, IEEE, Li Deng, Fellow, IEEE, and Dong Yu, Senior Member, IEEE

Abstract—A novel deep architecture, the Tensor Deep Stacking Network (T-DSN), is presented. The T-DSN consists of multiple stacked blocks, where each block contains a bilinear mapping from two hidden layers to the output layer, using a weight tensor to incorporate higher-order statistics of the hidden binary ([0, 1]) features. A learning algorithm for the T-DSN’s weight matrices and tensors is developed and described, in which the main parameter estimation burden is shifted to a convex sub-problem with a closed-form solution. Using an efficient and scalable parallel implementation for CPU clusters, we train sets of T-DSNs in three popular tasks in an increasing order of the data size: handwritten digit recognition using MNIST (60k), isolated state/phone classification and continuous phone recognition using TIMIT (1.1m), and isolated phone classification using WSJ0 (5.2m). Experimental results in all three tasks demonstrate the effectiveness of the T-DSN and the associated learning methods in a consistent manner. In particular, a sufficient depth of the T-DSN, a symmetry in the two hidden layers structure in each T-DSN block, our model parameter learning algorithm, and a softmax layer on top of T-DSN are shown to have all contributed to the low error rates observed in the experiments for all three tasks.

Index Terms—Deep learning, stacking networks, tensor, bilinear models, handwriting image classification, phone classification and recognition, MNIST, TIMIT, WSJ

INTRODUCTION

Recently, a deep classification architecture built the T-DSN retains the same linear-nonlinear interlaced structure as DSN in building up the deep architecture.
Fig. 1. An example T-DSN architecture with three stacking blocks, where each block consists of three layers, and an asymmetric symbol to indicate the block uses a dense weight matrix.

Fig. 2. Equivalent architecture to the bottom block of Fig. 1, where the tensor is unfolded into a large matrix.
Deep Stacking Networks for Information Retrieval

Li Deng, Xiaodong He, and Jianfeng Gao
Microsoft Research, Redmond

ICASSP, May 30, 2013
Outline

• Motivation: deep learning for Information Retrieval (IR)
  – Learning to rank
  – Semantic feature extraction for ranking

• Deep Stacking Net (DSN)
  – Basic modular architectures
  – Novel discriminative learning algorithm

• Applying DSN for IR --- learning to rank
  – Formulating IR as a classification problem
  – Special role of regularization

• Experiments
  – IR task, data sets, and features
  – Relationship between NDCG score & classification error rate
  – NDCG results on an IR task (Ads selection)
• Goal of IR: ranking text documents (D) for a query (Q)

• Common methods:
  – Lexical matching: suffers from text discrepancy btw Q and D (e.g. vocabulary, word usage, expression style, etc.)
    • E.g., TF-IDF weighted vector space model
  – Semantic matching: to bridge lexical gaps btw Q and D
    • E.g., Latent Semantic Analysis (LSA), PLSA, LDA, etc.
  – Learning Q-D matching using clickthrough data
    • E.g., translation models, bilingual topic models etc.
  – These linear models suffer from restricted expressive power
Deep Learning for IR

• Multilayers of nonlinearities
  – Greater expressive power
  – Better able to capture semantic contents in Q and D
  • E.g., semantic hashing (Hinton et al, 2007)
  – More effective use of supervised clickthrough data

• Use of (labeled) clickthrough data for IR ranking
  – Shallow linear models: Gao et al., 2010;2011
  – Shallow nonlinear models: Burges et al., 2005;2006
Deep Stacking Net (DSN)

- Deep Stacking Nets (Deng & Yu, Interspeech’10; Deng, Yu, Platt, ICASSP’12)
- Interleave linear/nonlinear layers
- Exploit closed-form constraints among network’s weights
- Much easier to learn than DNN
- Naturally amenable to parallel training
- (Largely) convex optimization
- Extended to tensor version (Hutchinson et al, ICASSP’12, TPAMI-2013)
- Extended to kernel version (Deng et al, SLT’12)
- Works very well for MNIST, TIMIT, WSJ, SLU
- This paper: a more recent application to IR ranking
Learning DSN Weights --- Main Ideas

- Learn weight matrices $U$ and $W$ in individual modules separately.
- Given $W$ and linear output layer, $U$ can be expressed as explicit nonlinear function of $W$.
- This nonlinear function is used as the constraint in solving nonlinear least square for learning $W$.
- Initializing $W$ with RBM (bottom layer)
- For higher layers, part of $W$ is initialized with the optimized $W$ from the immediately lower layer and part of it with random numbers.
Learning DSN Weights --- Single Module

\[ E = \frac{1}{2} \sum_n ||y_n - t_n||^2, \quad \text{where } y_n = U^T h_n = U^T \sigma(W^T x_n) = G_n(U, W) \]

\[ \frac{\partial E}{\partial U} = 2H(U^T H - T)^T \rightarrow U = (HHT)^{-1} HT^T = F(W), \quad \text{where } h_n = \sigma(W^T x_n) \]

\[ E = \frac{1}{2} \sum_n ||G_n(U, W) - t_n||^2, \quad \text{subject to } U = F(W), \]

Use of Lagrange multiplier method:

\[ E = \frac{1}{2} \sum_n ||G_n(U, W) - t_n||^2 + \lambda ||U - F(W)|| \]

to learn \( W \) and then \( U \rightarrow \) no longer backpropagation

• Advantages found:
  --- less noise in gradient than using chain rule ignoring explicit constraint \( U = F(W) \)
  --- batch learning is effective, aiding parallel training
Experimental Evaluation

• IR task
  – Sponsored Search: retrieve and rank relevant ads given a query

• Data sets
  – Training: 189K query–ads pairs
  – Testing: 58K query–ads pairs

• Features to DSN
  – A total of 160 features in two categories
    • Text features: TF-IDF, word overlap, length, etc.
    • User click features: clickthrough, clicked queries, etc.

• State-of-the-art baseline system (Burges et al. 2006)
  – LambdaRank, a single-hidden-layer neural network
  – Trained to maximize (a smoothed approximation of) NDCG via heuristic lambda-function
Evaluation Metric

- Metric: Normalized Discounted Cumulative Gain (NDCG)
- DCG at rank $p = \text{relevance}_1 + \sum_{i=2}^{p} \frac{\text{relevance}_i}{\log_2 i}$; $\text{relevance}_i$: human label of $\text{doc}_i$, scale 0-4
- IDCG: Ideal DCG, DCG score when assuming docs are ranked by human label
- NDCG = DCG/IDCG
- 1 NDCG pt (0.01) in our setting is statistically significant
NDCG Results

IR Quality measures (NDCG) for the DSN System vs. Baseline

<table>
<thead>
<tr>
<th>IR Systems</th>
<th>NDCG@1</th>
<th>NDCG@3</th>
<th>NDCG@10</th>
</tr>
</thead>
<tbody>
<tr>
<td>LambdaRank</td>
<td>0.331</td>
<td>0.347</td>
<td>0.382</td>
</tr>
<tr>
<td>DSN system</td>
<td>0.359</td>
<td>0.366</td>
<td>0.402</td>
</tr>
</tbody>
</table>
Analysis

Relationship between classification error rates and NDCG@1 measure

Observations:
• Correlation is clearly evidenced for NDCG1 < 0.35
• Weaker correlation in the high IR-quality region, i.e., NDCG1 > 0.35

Implication:
• Due to the inconsistency between the training objective and the IR-quality measure
• It is desirable to train the model to optimize the end-to-end IR quality directly
Learning Curves
Conclusions (of this ICASSP-2013 paper)

• First study on the use of deep learning techniques for learning-to-rank in IR problems
• Significantly better than shallow neural network
• Model trained by MSE
  – Generally correlated well with the NDCG as the IR quality measure
  – But weaker correlation in the region of high IR quality
• Deep learning using end-to-end IR-relevant metric is a key future direction
PART II: Deeper Substance of DL

---Technical introduction: RBM, DBN, DNN, DNN-HMM, CNN, RNN

---Examples of incorporating domain knowledge (about speech) into DL architectures
1. Hidden/articulatory Speech dynamics into RNN
2. Speech invariance/class-discrim.into deep-CNN

-A few new, promising DL architectures (CONTINUED)
New Types of Deep Neural Network & Learning for Speech Recognition+
An Overview

Li Deng, Geoffrey Hinton, Brian Kingsbury
MSR, U. Toronto/Google, IBM

ICASSP Special Session, May 28, 2013
Special Session Motivations

• Huge impact of deep neural nets (DNN) in speech (and vision, language, etc.)
Special Session Motivations

- Review article (2011-2012)
- Key factors:
  - Deeper network
  - Faster hardware
  - Larger network output layer (& hidden, input layers)
  - Better network initialization (not essential with big data)
- Rather standard MLP architecture
- Also standard backprop learning (1980’s)
Take-Away from This Special Session

- New models and new learning methods
- Key capabilities of DNNs in knowledge transfer, learning representations, etc.
- Advances in DNNs since the SPM overview paper
Recent History of “Deep” Models in Speech

- U Toronto’s DBN-DNN (2006-2009)

(a simple “recipe”)
**Hinton’s 2009 “Recipe”**

[Diagram of the neural network structure.]

---

FIG1] The sequence of operations used to create a DBN with three hidden layers and to convert it to a pretrained DBN-DNN. First, a GRBM is trained to model a window of frames of real-valued acoustic coefficients. Then the states of the binary hidden units of the GRBM are used as data for training an RBM. This is repeated to create as many hidden layers as desired. Then the stack of RBMs is converted to a single generative model, a DBN, by replacing the undirected connections of the lower level RBMs by top-down, directed connections. Finally, a pretrained DBN-DNN is created by adding a “softmax” output layer that contains one unit for each possible state of each HMM. The DBN-DNN is then discriminatively trained to predict the HMM state corresponding to the central frame of the input window in a forced alignment.
Li Deng, Dong Yu, Geoffrey Hinton

Microsoft Research; Microsoft Research; University of Toronto

Deep Learning for Speech Recognition and Related Applications

7:30am - 6:30pm Saturday, December 12, 2009

Location: Hilton: Cheakamus

Abstract: Over the past 25 years or so, speech recognition technology has been dominated by a “shallow” architecture — hidden Markov models (HMMs). Significant technological success has been achieved using complex and carefully engineered variants of HMMs. The next generation of the technology requires solutions to remaining technical challenges under diversified deployment environments. These challenges, not adequately addressed in the past, arise from the many types of variability present in the speech generation process. Overcoming these challenges is likely to require “deep” architectures with efficient learning algorithms. For speech recognition and related sequential pattern recognition applications, some attempts have been made in the past to develop computational architectures that are “deeper” than conventional HMMs, such as hierarchical HMMs, hierarchical point-process models, hidden dynamic models, and multi-level detection-based architectures, etc. While positive recognition results have been reported, there has been a conspicuous lack of systematic learning techniques and theoretical guidance to facilitate the development of these deep architectures. Further, there has been virtually no effective communication between machine learning researchers and speech recognition researchers who are both advocating the use of deep architecture and learning. One goal of the proposed workshop is to bring together these two groups of researchers to review the progress in both fields and to identify promising and synergistic research directions for potential future cross-fertilization and collaboration.

Deep Learning for Phone Recognition
(a stunning discovery at MSR, 2009)

<table>
<thead>
<tr>
<th>METHOD</th>
<th>Error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Shallow) GMM-HMM (1987-2010)</td>
<td>27.3%</td>
</tr>
<tr>
<td>AUGMENTED CONDITIONAL RANDOM FIELDS [26]</td>
<td>26.6%</td>
</tr>
<tr>
<td>RANDOMLY INITIALIZED RECURRENT NEURAL NETS [27]</td>
<td>26.1%</td>
</tr>
<tr>
<td>BAYESIAN TRIPHONE GMM-HMM [28]</td>
<td>25.6%</td>
</tr>
<tr>
<td>Deep/hidden trajectory model (MSR, 2006)</td>
<td>24.8%</td>
</tr>
<tr>
<td>MONOPHONE RANDOMLY INITIALIZED DNNs (SIX LAYERS) [13]</td>
<td>23.4%</td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs (SIX LAYERS) [13]</td>
<td>22.4%</td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs WITH MMI TRAINING (MSR, 2010)</td>
<td>22.1%</td>
</tr>
<tr>
<td>TRIPHONE GMM-HMMs DT W/ BMMI (IBM, 2010)</td>
<td>21.7%</td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs ON FBANK (EIGHT LAYERS) [13]</td>
<td>20.7%</td>
</tr>
<tr>
<td>MONOPHONE MCRBM-DBN-DNNs ON FBANK (FIVE LAYERS) [33]</td>
<td>20.5%</td>
</tr>
<tr>
<td>Deep convolutional nets w. DropOut &amp; Heter. Pooling (MSR, 2012)</td>
<td>18.7%</td>
</tr>
</tbody>
</table>

Error pattern A
Error pattern B
Deep Learning for Large-Vocabulary Speech Recognition

<table>
<thead>
<tr>
<th>TASK</th>
<th>HOURS OF TRAINING DATA</th>
<th>DNN-HMM</th>
<th>GMM-HMM WITH SAME DATA</th>
<th>GMM-HMM WITH MORE DATA</th>
</tr>
</thead>
<tbody>
<tr>
<td>SWITCHBOARD</td>
<td>309</td>
<td>16.1</td>
<td>23.6</td>
<td>17.1 (2,000 H)</td>
</tr>
<tr>
<td>ENGLISH BROADCAST NEWS</td>
<td>50</td>
<td>17.5</td>
<td>18.8</td>
<td></td>
</tr>
<tr>
<td>BING VOICE SEARCH (SENTENCE ERROR RATES)</td>
<td>24</td>
<td>30.4</td>
<td>36.2</td>
<td></td>
</tr>
<tr>
<td>GOOGLE VOICE INPUT</td>
<td>5,870</td>
<td>12.3</td>
<td></td>
<td>16.0 (&gt;&gt; 5,870 H)</td>
</tr>
<tr>
<td>YOUTUBE</td>
<td>1,400</td>
<td>47.6</td>
<td></td>
<td>52.3</td>
</tr>
</tbody>
</table>
New Discoveries about the DNN “Recipe” since 2009

• Pre-training not needed when a lot of labeled data are available (2010)
• The recipe works well for LVCSR when DNN output units correspond to CD HMM states (2010)
• Decoding alg. & infrastructure largely unchanged, enabling industry-scale speech recognition (2010-2013)
• Filterbank features (closer to waveform) better than MFCCs for DNNs (opposite to GMM systems) (2011-2013)
• DNN works surprisingly well for noisy speech (2012)
• Fully-connected DNN can be modified to include “convolutional” layers to handle speech variability (2012-2013)
• DNN highly effective for multi-task/transfer learning (e.g. multilingual ASR, 2012-2013)
• DNN effective for applications beyond ASR.
Five Technical Papers in Our Special Session

- RECENT ADVANCES IN DEEP LEARNING FOR SPEECH RESEARCH AT MICROSOFT
- IMPROVING DEEP NEURAL NETWORKS FOR LVCSR USING RECTIFIED LINEAR UNITS AND DROPOUT
- DEEP CONVOLUTIONAL NEURAL NETWORKS FOR LVCSR
- MULTILINGUAL ACOUSTIC MODELS USING DISTRIBUTED DEEP NEURAL NETWORKS
- ADVANCES IN OPTIMIZING RECURRENT NETWORKS
Themes in the Session
Themes: Better Inputs

MFCCs $\rightarrow$ log Mel spectra
Themes: Nonlinearities
Themes: Architectures

Recurrent NNs

Convolutional NNs
Themes: Optimization

- Distributed asynchronous SGD
- Distributed Hessian-free optimization
- Nesterov's accelerated gradient method
- Gradient clipping
Themes: **Regularization**

Sparsity in hidden representations

Dropout
Themes: **Hyperparameters**

Grid search → Sampling

Bayesian optimization
Themes: **Multi-task Learning**

Multi-lingual acoustic modeling

Mixed-bandwidth acoustic modeling
Recent Advances in Deep Learning for Speech Research at Microsoft

Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Mike Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, Alex Acero
Outline

• Advances in deep learning for **features/** representations
• Advances in deep learning for **models/** architectures
• Systems and applications in acoustic modeling, language modeling, dialogue, (and information retrieval/search)
Learning Features/Representations

• Advances in deep learning for features/representations
• Advances in deep learning for models/architectures
• Systems and applications in acoustic modeling, language modeling, dialogue, (and information retrieval/search)
Scientists See Promise in Deep-Learning Programs

John Markoff
November 23, 2012
Speech Recognition Progress:
--- gleaned from NIST evaluations

After no improvement for 10+ years by the research community...

...MSR used deep learning to reduce error rate from \(~23\%\) to \(~13\%\) on SWBD (and under 7\% for Rick Rashid’s demo)!
Back to Primitive Spectral Features

• Philosophy of deep learning:
  – Learning representations automatically instead of manually engineering/design them (e.g., MFCC, PLP)
• DNN capability in representing correlated feature dimensions
• → eliminate cosine transform in MFCC in favor of filterbanks in spectral domain
Back to Primitive Spectral Features

- Philosophy of deep learning:
  - Learning representations automatically instead of manually engineering/design them (e.g., MFCC, PLP)
- DNN capability in representing correlated feature dimensions
- \( \rightarrow \) eliminate cosine transform in MFCC in favor of filterbanks and spectrograms in the spectral domain
In early 2010, we discovered:

For deep autoencoding of speech features:

- Both spectrogram/filterbank features are better than MFCCs
- Better to use spectrogram features than filterbanks
- “Better” in terms of coding efficiency (i.e., errors/energy)
# LVCSR Using Spectral Features

<table>
<thead>
<tr>
<th>LVCSR Systems</th>
<th>Word error rate</th>
</tr>
</thead>
<tbody>
<tr>
<td>Best GMM-HMM (MFCCs; fMPE+BMMI)</td>
<td>34.7%</td>
</tr>
<tr>
<td>DNN (MFCCs)</td>
<td>31.6%</td>
</tr>
<tr>
<td>DNN (Spectrogram --- 256 log FFT bins)</td>
<td>32.3%</td>
</tr>
<tr>
<td>DNN (29 log filter-banks)</td>
<td>30.1%</td>
</tr>
<tr>
<td>DNN (40 log filter-banks)</td>
<td>29.9%</td>
</tr>
</tbody>
</table>

- Filterbanks > MFCC > Spectrograms
- Not quite consistent with deep autoencoder results
- Further research: regularization, online feature normalization at sentence level, etc.
Learning Multi-Task Features

Mixed-Band DNN architecture:

Multilingual DNN architecture:
Shared Hidden Layers with Language-Specific Output Layers
Learning Multi-Task Features

English or French or Chinese words with fewer errors

8KHz or 16KHz English or French or Chinese speech input

8KHz waveforms
16KHz waveforms
English training data
French training data
Chinese training data

Training

Runtime
Multi-Band ASR Summary Results

Practical Goal: exploit narrowband labeled data from earlier telephone-based applications

<table>
<thead>
<tr>
<th>Training Data</th>
<th>Test WER (Wideband)</th>
<th>Test WER (Narrowband)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Wideband only</td>
<td>30.0%</td>
<td>71.2%</td>
</tr>
<tr>
<td>Narrowband only</td>
<td>-</td>
<td>29.0%</td>
</tr>
<tr>
<td>Wideband+Narrowband</td>
<td>28.3%</td>
<td>29.3%</td>
</tr>
</tbody>
</table>
## Multilingual ASR Summary Results

<table>
<thead>
<tr>
<th>Speech Recognizers</th>
<th>WER on ENU</th>
</tr>
</thead>
<tbody>
<tr>
<td>DNN trained with only ENU data</td>
<td>30.9%</td>
</tr>
<tr>
<td>+FRA, retrain all layers with ENU</td>
<td>30.6%</td>
</tr>
<tr>
<td>or +FRA, retrain the top layer with ENU</td>
<td>27.3%</td>
</tr>
<tr>
<td>or +FRA+ DEU+ ESP+ITA, retrain top layer</td>
<td>25.3%</td>
</tr>
</tbody>
</table>
Deep Convolutional Net w. Spectral Features

• “Spatial” (freq-domain) invariance of speech due to vocal-tract-length differences across speakers
• Convolution/pooling makes sense for
  – spectral features, not MFCC
  – “spatial” dimension, not (pure) temporal dimension
• Excellent results on TIMIT:

<table>
<thead>
<tr>
<th>DNN</th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>MONOPHONE RANDOMLY INITIALIZED DNNs (SIX LAYERS) [13]</td>
<td>23.4%</td>
<td></td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs (SIX LAYERS) [13]</td>
<td>22.4%</td>
<td></td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs WITH MMI TRAINING (MSR, 2010)</td>
<td>22.1%</td>
<td></td>
</tr>
<tr>
<td>TRIPHONE GMM-HMMs DT W/ BMMI (IBM, 2010)</td>
<td>21.7%</td>
<td></td>
</tr>
<tr>
<td>MONOPHONE DBN-DNNs ON FBANK (EIGHT LAYERS) [13]</td>
<td>20.7%</td>
<td></td>
</tr>
<tr>
<td>MONOPHONE MCRBM-DBN-DNNs ON FBANK (FIVE LAYERS) [33]</td>
<td>20.5%</td>
<td></td>
</tr>
<tr>
<td>Deep convolutional nets w. DropOut &amp; Heter. Pooling (MSR, 2012)</td>
<td><strong>18.7%</strong></td>
<td></td>
</tr>
</tbody>
</table>

Disastrous when MFCC is used for convolutional net over “spatial” dimension
Noise Robust DNN Features

- Beating state-of-the-art WER results on Aurora4 task (medium vocabulary task based on WSJ0).
- DNN: not yet exploited explicit noise compensation algorithm.
- DNN: no multi-pass decoding allowing for adaptation.

**ASR Word Error Rate % for Aurora4:**

<table>
<thead>
<tr>
<th></th>
<th>A</th>
<th>B</th>
<th>C</th>
<th>D</th>
<th>AVG</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-HMM (Baseline)</td>
<td>12.5</td>
<td>18.3</td>
<td>20.5</td>
<td>31.9</td>
<td>23.9</td>
</tr>
<tr>
<td>GMM (MPE+VAT)</td>
<td>7.2</td>
<td>12.8</td>
<td>11.5</td>
<td>19.7</td>
<td>15.3</td>
</tr>
<tr>
<td>GMM + Deriv. Kernels</td>
<td>7.4</td>
<td>12.6</td>
<td>10.7</td>
<td>19.0</td>
<td>14.8</td>
</tr>
<tr>
<td>DNN (7x2000)</td>
<td>5.6</td>
<td>8.8</td>
<td>8.9</td>
<td>20.0</td>
<td>13.4</td>
</tr>
</tbody>
</table>
DNN “Model” Adaptation

<table>
<thead>
<tr>
<th>Speech Recognition Systems</th>
<th>WER</th>
</tr>
</thead>
<tbody>
<tr>
<td>GMM-HMM</td>
<td>43.6%</td>
</tr>
<tr>
<td>DNN</td>
<td>34.1%</td>
</tr>
<tr>
<td>DNN + AdaptSoftMax (SGD)</td>
<td>29.4%</td>
</tr>
<tr>
<td>DNN + fDLR (SGD)</td>
<td>28.5%</td>
</tr>
</tbody>
</table>
New “Model” Architecture: Recurrent Net

- For language modeling application
- Exploit context dependency (side channel information) in RNN
- **Side channel** consists of slowly varying LSA vectors of preceding text
- Evaluation on Penn Treebank data
- Baseline (KN 5-gram LM w. cache) - perplexity= 126
- RNN w. side channel - perplexity= 110 (lowest single-model perplexity for this data)
Deep Learning for Dialogue State Tracking

- Fertile area with preliminary exploration
- Interleave linear/nonlinear layers
- Exploit closed-form constraints among network’s weights
- Much easier to learn than DNN
- Naturally amenable to parallel training
- (Largely) convex optimization
- Extended to tensor and kernel versions
- Works very well for MNIST, TIMIT, WSJ, SLU, and IR ranking (Deng, He, Gao: ICASSP 2013)
- Here we show a more recent application to state tracking task in spoken dialogue systems.
Dialogue State Tracking Example (Jason Williams)

• \[ \text{Prob} \left[ \text{CorrectUserGoals}_t \mid \text{DialogueHistory}_{1,2,\ldots,t-1}, \text{UserInfo}_{1,2,\ldots,t-1} \right] \]

<table>
<thead>
<tr>
<th>System output</th>
<th>User speech</th>
<th>SLU output + confidence</th>
<th>Per-hypothesis features</th>
<th>General features</th>
<th>Dialog state hyps</th>
<th>Distribution over dialog state hyps</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hello, which bus route?</td>
<td>sixty one c</td>
<td>61B 0.4, 61D 0.3, 61C 0.1</td>
<td>61B, 61D, 61C</td>
<td>QuestionType: Ask, Times-asked: 1, Times-confirmed: 0, Hyp-count: 3</td>
<td>61B, 61D, 61C, Rest</td>
<td><img src="image1" alt="Distribution" /></td>
</tr>
<tr>
<td>Sorry, which bus route?</td>
<td>sixty one c</td>
<td>63 0.5, 53 0.4, 61C 0.2</td>
<td>63, 61B, 61D, 61C</td>
<td>QuestionType: Ask, Times-asked: 2, Times-confirmed: 0, Hyp-count: 5</td>
<td>61B, 61D, 61C, 63, Rest</td>
<td><img src="image2" alt="Distribution" /></td>
</tr>
<tr>
<td>Sixty one c, is that right?</td>
<td>yes</td>
<td>YES 0.9</td>
<td>63, 53, 61B, 61D, 61C</td>
<td>QuestionType: Confirm, Times-asked: 2, Times-confirmed: 1, Hyp-count: 5</td>
<td>61B, 61D, 61C, 63, 53, Rest</td>
<td><img src="image3" alt="Distribution" /></td>
</tr>
</tbody>
</table>

Each hypothesis is described by \( M \) features in each turn. In this example, \( M=3 \).

Each turn also has \( K \) features that describe general dialog context. In this example, \( K=4 \).

At a given turn, there are \( G \) dialog state hypotheses to score. At this turn, \( G=5 \).
Task: Dialog state tracking (defined in Spoken Dialogue Challenge 2010)
Strong interactions among features \(\rightarrow\) strength of deep networks
Can be framed as a multiple binary classification problem
Baseline: carefully tuned, highly optimized Max Entropy classifier (J. Williams)
Deep Stacking Nets (slightly tuned) achieve similar accuracy\% for all 5 slots:

<table>
<thead>
<tr>
<th></th>
<th>State of the Art baseline</th>
<th>Deep Stacking Networks</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Bus route</strong></td>
<td>58.0%</td>
<td>58.1%</td>
</tr>
<tr>
<td><strong>Origin location</strong></td>
<td>56.4%</td>
<td>57.1%</td>
</tr>
<tr>
<td><strong>Destination location</strong></td>
<td>66.5%</td>
<td>65.4%</td>
</tr>
<tr>
<td><strong>Date</strong></td>
<td>83.9%</td>
<td>84.6%</td>
</tr>
<tr>
<td><strong>Time</strong></td>
<td>63.1%</td>
<td>62.5%</td>
</tr>
</tbody>
</table>
Conclusions

• Deep learning is a powerful technology
  – Automatic learning of representations
  – Multi-task learning
  – Factorizing/disentangling multiple causes of variations

• Future directions
  – More effective deep architectures and learning algms
  – Scale deep mode training with bigger data
  – Extend applications of deep learning: acoustic models, language models, dialogue, end-2-end language understanding & translation, IR/search, synthesis, music processing, etc.
Summary

PART I: Basics of Deep Learning (DL)
--- including impact and recent history of DL (Deep Neural Net, DNN) in speech recognition

PART II: Deeper Substance of DL
--- including connections to other ML paradigms
--- two examples of incorporating speech knowledge in DL architectures,
--- recent experiments in speech recognition with new DL architectures beyond DNN


Selected References


Grégoire Mesnil, Xiaodong He, Li Deng, and Yoshua Bengio, Investigation of Recurrent-Neural-Network Architectures and Learning Methods for Spoken Language Understanding, in Interspeech 2013, August 2013

Ossama Abdel-Hamid, Li Deng, and Dong Yu, Exploring convolutional neural network structures and optimization techniques for speech recognition, in Proc. Interspeech, Lyon, France, August 2013


Li Deng, Jinyu Li, Jui-Ting Huang, Kaisheng Yao, Dong Yu, Frank Seide, Michael Seltzer, Geoff Zweig, Xiaodong He, Jason Williams, Yifan Gong, and Alex Acero, Recent Advances in Deep Learning for Speech Research at Microsoft, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2013


Jui-Ting Huang, Jinyu Li, Dong Yu, Li Deng, and Yifan Gong, CROSS-LANGUAGE KNOWLEDGE TRANSFER USING MULTILINGUAL DEEP NEURAL NETWORK WITH SHARED HIDDEN LAYERS, in IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2013

Li Deng, Xiaodong He, and Jianfeng Gao, Deep Stacking Networks for Information Retrieval, IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), May 2013


Po-Sen Huang, Kshitiz Kumar, Chaojun Liu, Yifan Gong, and Li Deng, PREDICTING SPEECH RECOGNITION CONFIDENCE USING DEEP LEARNING WITH WORD IDENTITY AND SCORE FEATURES, in Proc. ICASSP, May 2013


“DBN vs DBN” (for fun)

<table>
<thead>
<tr>
<th>Acronym</th>
<th>Definition</th>
</tr>
</thead>
<tbody>
<tr>
<td>DBN</td>
<td>1, 5-Diazabicyclo(4.3.0)Non-5-Ene (chemical compound)</td>
</tr>
<tr>
<td>DBN</td>
<td>Doing Business - Not</td>
</tr>
<tr>
<td>DBN</td>
<td>Dialog Broadband Networks (Dialog Telekom PLC; Sri Lanka)</td>
</tr>
<tr>
<td>DBN</td>
<td>De Bonis Non (Legal: appointment of a personal representative to a vacancy)</td>
</tr>
<tr>
<td>DBN</td>
<td>Divisible by None (band)</td>
</tr>
<tr>
<td>DBN</td>
<td><strong>Deep Belief Network (machine learning)</strong></td>
</tr>
<tr>
<td>DBN</td>
<td>Dynamic Bayes Network</td>
</tr>
<tr>
<td>DBN</td>
<td>Data Bus Network</td>
</tr>
<tr>
<td>DBN</td>
<td>Dial-Back Number</td>
</tr>
<tr>
<td>DBN</td>
<td>Day Beacon</td>
</tr>
<tr>
<td>DBN</td>
<td>Domain-Border Node</td>
</tr>
<tr>
<td>DBN</td>
<td>Digital Billboard Network (Australia)</td>
</tr>
<tr>
<td>DBN</td>
<td>Drunk Before Noon</td>
</tr>
<tr>
<td>DBN</td>
<td>District Borough Number (New York City Department of Education school identifier)</td>
</tr>
<tr>
<td>DBN</td>
<td>Database Notification</td>
</tr>
<tr>
<td>DBN</td>
<td>Directed Bipartite Network</td>
</tr>
</tbody>
</table>