Learning to Represent Semantics

Yoshua Bengio

Words2Actions Workshop,
NAACL HLT 2012, Montreal
From AI to Deep Learning

• AI requires operational knowledge
• Handcrafting it all is daunting, brittle, incomplete, failed: learn it
• Most common now: hand-crafted features + simple (linear) ML
• Without the right (task-specific) features: curse of dimensionality
• Need for learning the features: representation-learning
• Theoretical and empirical evidence in favor of multiple levels of representation (Deep Learning)
Deep Learning: General Motivation

- Learning features
  - Learn features as part of a machine learning system
  - Not all features can be explicitly described by experts

- Biologically inspired learning
  - Brain has a deep architecture
  - Cortex seems to have a generic learning algorithm
  - Humans first learn simpler concepts and then compose them to more complex ones
Deep Learning: General Motivation

- It works well already for vision, NLP, collaborative filtering,…
- Wins two transfer learning competitions in 2011
- State of the art performance for POS, NER, Chunking

<table>
<thead>
<tr>
<th>Task</th>
<th>Benchmark</th>
<th>SENNA</th>
</tr>
</thead>
<tbody>
<tr>
<td>Part of Speech (POS) (Accuracy)</td>
<td>97.24 %</td>
<td>97.29 %</td>
</tr>
<tr>
<td>Chunking (CHUNK) (F1)</td>
<td>94.29 %</td>
<td>94.32 %</td>
</tr>
<tr>
<td>Named Entity Recognition (NER) (F1)</td>
<td>89.31 %</td>
<td>89.59 %</td>
</tr>
</tbody>
</table>

(Colloboert et al., 2011)

- Sentiment analysis on opinions, experiences, movies
- Paraphrase detection (Socher et al. 2011)
- Relation classification
- Language Modeling (Schwenk et al, Mikolov et al)
Deep Learning Motivation for Semantics

- Language Models: model joint probability of word sequences

- Training sentence
  
  *The cat is walking in the bedroom*

- Test sentence:
  
  *A dog was running in a room*

- Sparsity / curse of dim. problem for longer n-grams
- Possible Solutions: back-off, word classes (too coarse)
- Better: similar representations for semantically similar phrases
1st step: represent words

- Deep learning can learn a distributed continuous-valued vector for each word from raw text:

<p>| | | | | |</p>
<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>France</td>
<td>Jesus</td>
<td>XBOX</td>
<td>Reddish</td>
<td>Scratched</td>
</tr>
<tr>
<td>Spain</td>
<td>Christ</td>
<td>Playstation</td>
<td>Yellowish</td>
<td>Smashed</td>
</tr>
<tr>
<td>Italy</td>
<td>God</td>
<td>Dreamcast</td>
<td>Greenish</td>
<td>Ripped</td>
</tr>
<tr>
<td>Russia</td>
<td>Resurrection</td>
<td>PS###</td>
<td>Brownish</td>
<td>Brushed</td>
</tr>
<tr>
<td>Poland</td>
<td>Prayer</td>
<td>SNES</td>
<td>Bluish</td>
<td>Hurled</td>
</tr>
<tr>
<td>England</td>
<td>Yahweh</td>
<td>WH</td>
<td>Creamy</td>
<td>Grabbed</td>
</tr>
<tr>
<td>Denmark</td>
<td>Josephus</td>
<td>NES</td>
<td>Whitish</td>
<td>Tossed</td>
</tr>
<tr>
<td>Germany</td>
<td>Moses</td>
<td>Nintendo</td>
<td>Blackish</td>
<td>Squeezed</td>
</tr>
<tr>
<td>Portugal</td>
<td>Sin</td>
<td>Gamecube</td>
<td>Silvery</td>
<td>Blasted</td>
</tr>
<tr>
<td>Sweden</td>
<td>Heaven</td>
<td>PSP</td>
<td>Greyish</td>
<td>Tangled</td>
</tr>
<tr>
<td>Austria</td>
<td>Salvation</td>
<td>Amiga</td>
<td>Paler</td>
<td>Slashed</td>
</tr>
</tbody>
</table>

Collobert & Weston, ICML’2008
In contrast to the “atomic” or “localist” representations employed in traditional cognitive science, a distributed representation is one in which “each entity is represented by a pattern of activity distributed over many computing elements, and each computing element is involved in representing many different entities.”

Local vs Distributed Latent Variables/Attributes

Clustering

Multi-clustering
2nd step: learn to compose words into phrases and semantic relations
“Deep” computer program
subroutine1 includes subsub1 code and subsub2 code and subsubsub1 code

subroutine2 includes subsub2 code and subsub3 code and subsubsub3 code and ...

main

“Shallow” computer program
“Deep” circuit
“Shallow” circuit

Falsely reassuring theorems: one can approximate any reasonable (smooth, boolean, etc.) function with a 2-layer architecture
Deep Architectures are More Expressive

Theoretical arguments:

2 layers of

- Logic gates
- Formal neurons
- RBF units

= universal approximator

RBMss & auto-encoders = universal approximator

Theorems on advantage of depth:
(Hastad et al 86 & 91, Bengio et al 2007, Bengio & Delalleau 2011, Braverman 2011)

Functions compactly represented with k layers may require exponential size with 2 layers
Sharing Components in a Deep Architecture

Polynomial expressed with shared components: advantage of depth may grow exponentially

\[(x_1 x_2)(x_2 x_3) + (x_1 x_2)(x_3 x_4) + (x_2 x_3)^2 + (x_2 x_3)(x_3 x_4)\]
Deep Architectures and Sharing Statistical Strength, Multi-Task / Transfer Learning

- Generalizing better to new tasks & domains is crucial to approach AI
- Deep architectures can learn good intermediate representations shared across tasks
- Good representations are often those making sense for many tasks because they capture underlying factors = semantics
Unsupervised and Transfer Learning Challenge + Transfer Learning Challenge: Deep Learning 1\textsuperscript{st} Place

ICML’2011 workshop on Unsup. & Transfer Learning

NIPS’2011 Transfer Learning Challenge
Paper: ICML’2012
Invariance and Disentangling

• Invariant features

• Which invariances?

• Alternative: learning to disentangle factors

• Good disentangling → avoid the curse of dimensionality
Advantages of Sparse Representations

- Just add a penalty on learned representation
- Information disentangling (compare to dense compression)
- More likely to be linearly separable (high-dimensional space)
- Locally low-dimensional representation = local chart
- Hi-dim. sparse = efficient variable size representation = data structure
Deep & Distributed NLP

- See “Neural Net Language Models” Scholarpedia entry
- NIPS’2000 and JMLR 2003 “A Neural Probabilistic Language Model”
  - Each word represented by a distributed continuous-valued code
  - Generalizes to sequences of words that are semantically similar to training sequences
Deep Learning: Motivations for NLP

- Allows to generalize to sequences of words that are semantically similar to training sequences

- Training sentence
  
  The cat is walking in the bedroom

- Can generalize to
  
  A dog was running in a room

- Because of the similarity between distributed representations for (a, the), (cat, dog), (is, was), etc.
Neural Networks for Learning Word Vectors

• Idea: A word and its context is a positive training sample, a random word in that same context is a negative training sample:

  - *cat chills on a mat*  *cat chills Jeju a mat*

• Similar: Implicit negative evidence in Contrastive Estimation, Smith and Eisner (2005)
A neural network for learning word vectors

• Idea: A word and its context is a positive training sample, a random word in that same context is a negative training sample.
• score(\textit{cat chills on a mat}) > score(\textit{cat chills Jeju a mat})
• How to compute the score?
  • With a neural network
  • Each word is associated with an n-dimensional vector
Word embedding matrix

- Initialize all word vectors randomly to form a word embedding matrix

\[ L \in \mathbb{R}^{n \times |V|} \]

\[ L = \begin{bmatrix} \vdots & \vdots & \vdots & \vdots & \vdots & \cdots & \vdots \\ \text{the} & \text{cat} & \text{mat} & \text{...} \end{bmatrix} \]

- These are the word features we want to learn
- Also called look-up table
t-SNE of Embeddings: zoom 2
t-SNE of Embeddings: zoom 3
Joint Image-Query Embedding Space

S. Bengio, J. Weston et al @ Google


Learn $\Phi_1(\cdot)$ and $\Phi_w(\cdot)$ to optimize precision@k.
Some results with deep distributed representations for NLP

- *(Bengio et al 2001, 2003)*: beating n-grams on small datasets (Brown & APNews), but much slower
- *(Collobert & Weston 2008)*: reaching state-of-the-art in multiple NLP tasks (SRL, POS, NER, chunking) thanks to unsupervised pre-training and multi-task learning
- *(Bai et al 2009)*: ranking & semantic indexing (info retrieval).
- *(Collobert 2010)*: Deep Learning for Efficient Discriminative Parsing
- *(S. Bengio, J. Weston et al @ Google, 2009, 2010, 2011)*: joint embedding space for images and keywords, Google image search
- *(Sutskever & Martens 2011)*: beating SOA in text compression.
- *(Socher et al 2011)*: parsing with recursive nets, ICML 2011 distinguished application paper award
- *(Mikolov et al 2011)*: beating the SOA in perplexity with recurrence
Domain Adaptation (ICML 2011)

Small (4-domain) Amazon benchmark: we beat the state-of-the-art handsomely

Sparse rectifiers
Stacked Denoising Autoencoders find more features that tend to be useful either for predicting domain or sentiment, not both = disentangling?
Sentiment Analysis: Transfer Learning

- 25 Amazon.com domains: toys, software, video, books, music, beauty, ...
- Unsupervised pre-training of input space on all domains
- Supervised SVM on 1 domain, generalize out-of-domain
- Baseline: bag-of-words + SVM
Representing Sparse High-Dimensional Stuff

$$f(x) = \max(0, x)$$


Speedup from *Sampled Reconstruction*
Modeling Semantics

Learning Structured Embeddings of Knowledge Bases, Bordes, Weston, Collobert & Bengio, AAAI 2011

Joint Learning of Words and Meaning Representations for Open-Text Semantic Parsing, Bordes, Glorot, Weston & Bengio, AISTATS 2012
Modeling Relations with Matrices

Model \((\text{lhs}, \text{relation}, \text{rhs})\)
Each concept = 1 embedding vector
Each relation = 2 matrices
Ranking criterion
Energy = low for training examples, high o/w

Diagram:
- \(\text{lhs}\) and \(\text{rhs}\) provide input vectors.
- \(\text{relation}\) accepts two matrices.
- Energy is calculated as the norm of the product of matrices and vectors.

Choose matrices and vectors to optimize the energy function.
Verb = relation. Too many to have a matrix each.
Each concept = 1 embedding vector
Each relation = 1 embedding vector
Can handle relations on relations on relations
Use SENNA (Collobert 2010) = embedding-based NLP tagger for Semantic Role Labeling, breaks sentence into (subject part, verb part, object part)
Use max-pooling to aggregate embeddings of words inside each part
Combining Multiple Sources of Evidence with Shared Embeddings

- The undirected graphical model version of relational learning
- With embeddings (shared representations) to help propagate information among data sources: here WordNet, XWN, Wikipedia, FreeBase,…
- Different energy functions can be used for different types of relations, or a generic representation and generic relation symbols used for everything
Open-Text Semantic Parsing (AISTATS 2012)

- Semantic Parsing: map a sentence into a Meaning Representation. Meaning Representation (MR): formal representation of the meaning. It can be in PROLOG, MySQL, ... or any structured language.
- Examples:
  - “What are the high points of states surrounding Mississippi ?”
    answer(A,(high point(B,A),state(B),next to(B,C),const(C,stateid(mississippi)))))
  - “Show me flights from Boston to New York.”
    SELECT flight id FROM flight WHERE from airport = ‘boston’ AND to airport = ‘new york’
  - Open-text: ability to handle any sentence regardless of its vocabulary (opposite to closed-domain).
Processing Pipeline

• 3 steps:

```
```

• last formula defines the Meaning Representation (MR).
Training Criterion

- Intuition: if an entity of a triplet was missing, we would like our model to predict it correctly i.e. to give it the lowest energy. For example, this would allow us to answer questions like “what is part of a car?”
- Hence, for any training triplet \( x_i = (\text{lhs}_i, \text{rel}_i, \text{rhs}_i) \) we would like:
  
  1. \( E(\text{lhs}_i, \text{rel}_i, \text{rhs}_i) < E(\text{lhs}_j, \text{rel}_i, \text{rhs}_i) \),
  2. \( E(\text{lhs}_i, \text{rel}_i, \text{rhs}_i) < E(\text{lhs}_i, \text{rel}_j, \text{rhs}_i) \),
  3. \( E(\text{lhs}_i, \text{rel}_i, \text{rhs}_i) < E(\text{lhs}_i, \text{rel}_i, \text{rhs}_j) \),

That is, the energy function \( E \) is trained to rank training samples below all other triplets.
Training Algorithm:
pseudo-likelihood + uniform sampling of negative variants

Train by stochastic gradient descent:

1. Randomly select a positive training triplet $x_i = (lhs_i, rel_i, rhs_i)$.
2. Randomly select constraint (1), (2) or (3) and an entity $\tilde{e}$:
   - If constraint (1), construct negative triplet $\tilde{x} = (\tilde{e}, rel_i, rhs_i)$.
   - Else if constraint (2), construct $\tilde{x} = (lhs_i, \tilde{e}, rhs_i)$.
   - Else, construct $\tilde{x} = (lhs_i, rel_i, \tilde{e})$.
3. If $E(x_i) > E(\tilde{x}) - 1$ make a gradient step to minimize:
   \[ \max(0, 1 - E(\tilde{x}) + E(x_i)) \]
4. Constraint embedding vectors to norm 1
<table>
<thead>
<tr>
<th>lhs</th>
<th>rel</th>
<th>rhs</th>
<th>TextRunner</th>
</tr>
</thead>
<tbody>
<tr>
<td>_army_NN_1</td>
<td>_attack_VB_1</td>
<td><strong>Israel</strong>&lt;br&gt;the village&lt;br&gt;another army&lt;br&gt;the city&lt;br&gt;the fort</td>
<td><strong>army</strong>&lt;br&gt;attacked&lt;br&gt;Israel&lt;br&gt;the village&lt;br&gt;another army&lt;br&gt;the city&lt;br&gt;the fort</td>
</tr>
<tr>
<td>troop_NN_4</td>
<td>_armed_service_NN_1 &lt;br&gt;_ship_NN_1 &lt;br&gt;_territory_NN_1 &lt;br&gt;_military_unit_NN_1</td>
<td>People&lt;br&gt;Players&lt;br&gt;one&lt;br&gt;Students&lt;br&gt;business</td>
<td></td>
</tr>
<tr>
<td>_business_firm_NN_1 &lt;br&gt;_person_NN_1 &lt;br&gt;_family_NN_1 &lt;br&gt;_payoff_NN_3 &lt;br&gt;_card_game_NN_1</td>
<td>_earn_VB_1</td>
<td>earn</td>
<td></td>
</tr>
<tr>
<td>money_NN_1</td>
<td></td>
<td>money</td>
<td></td>
</tr>
</tbody>
</table>

MRs inferred from text define triplets between WordNet synsets.

Model captures knowledge about relations between nouns and verbs.

→ Implicit addition of new relations to WordNet!

→ Generalize Freebase!
Question Answering: Ranking Score

% of answers in Top-10

- Random
- Bigram Counts
- WN only
- All

Scores:
- Random: 0
- Bigram Counts: 10
- WN only: 60
- All: 35
Embedding Near Neighbors of Words & Senses

<table>
<thead>
<tr>
<th>mark NN</th>
<th>mark NN_1</th>
<th>mark NN_2</th>
</tr>
</thead>
<tbody>
<tr>
<td>indication NN</td>
<td>score NN_1</td>
<td>marking NN_1</td>
</tr>
<tr>
<td>print NN_3</td>
<td>number NN_2</td>
<td>symbolizing NN_1</td>
</tr>
<tr>
<td>print NN</td>
<td>gradation NN</td>
<td>naming NN_1</td>
</tr>
<tr>
<td>roll NN</td>
<td>evaluation NN_1</td>
<td>marking NN</td>
</tr>
<tr>
<td>pointer NN</td>
<td>tier NN_1</td>
<td>punctuation NN_3</td>
</tr>
<tr>
<td>take VB</td>
<td>canary NN</td>
<td>different JJ_1</td>
</tr>
<tr>
<td>bring VB</td>
<td>sea mew NN_1</td>
<td>eccentric NN</td>
</tr>
<tr>
<td>put VB</td>
<td>yellowbird NN_2</td>
<td>dissimilar JJ</td>
</tr>
<tr>
<td>ask VB</td>
<td>canary bird NN_1</td>
<td>same JJ_2</td>
</tr>
<tr>
<td>hold VB</td>
<td>larus marinus NN_1</td>
<td>similarity NN_1</td>
</tr>
<tr>
<td>provide VB</td>
<td>mew NN</td>
<td>common JJ_1</td>
</tr>
</tbody>
</table>
Word Sense Disambiguation

- Senseval-3 results
  (only sentences with Subject-Verb-Object structure)
  
  MFS = most frequent sense
  All = training from all sources
  Gamble = Decadt et al 2004 (Senseval-3 SOA)

- XWN results
  XWN = eXtended WN
Recursive Application of Relational Operators

Relations on Multiple Data Types

• Add energy terms associated to relations from different data sources, shared embeddings

\[
\begin{align*}
\text{energy} & (\text{object image, is-a, object label}) + \\
\text{energy} & (\text{part image, is-a, part label}) + \\
\text{energy} & (\text{part image, image-part-of, object image}) + \\
& \text{energy} (\text{part label, label-part-of, object label})
\end{align*}
\]

Table 1: Summary of Test Set Results on ImageNet-WordNet. Precision at 1 and 10, and Mean Average Precision (MAP) are given. (IW) resp. (I) refers to the (Image,Word) setup resp. (Image).
Recurrent and Recursive Nets

- Replicate a parametrized function over different time steps or nodes of a DAG
- Output state at one time-step / node is used as input for another time-step / node
- Very deep once unfolded!
Conclusion

- AI → learning → representation-learning
- Deep learning to disentangle factors of variation and discover representations for higher-level abstractions
- No immediate generalization from discrete spaces → learn a distributed semantic representation for discrete objects
- Word embeddings generalize across semantically similar words
- Combine word embeddings into representations and energy functions for phrases and relations
- Applications to language modeling (speech recognition, language translation), sentiment analysis, parsing, paraphrasing, word sense disambiguation, question answering…
LISA team: Merci! Questions?