Perspectives for Low-Resource Tasks of Connectionist Natural Language Processing

Overview and Discussion

David Kaumanns
Center for Information and Language Processing
University of Munich

david@heidenblog.de

Machine Learning Workshop Trondheim - September 22-23, 2016

Content

- Connectionism and Neural Networks
- The Data Problem of DL-NLP
- The Language Problem of DL-NLP
- Perspectives for Low-Resource-NLP
  - Toy Tasks
  - Character-Based Neural Network Models
  - Deep Transfer Learning
    - Example: Word Embeddings

Connectionism

- Paradigm of **Symbol Processors**: Deterministic rules manipulate symbols encoding complex information.
- **Connectionism**: High-level functions are performed by a *large network* of *simple computational units*.
  - "What fires together, wires together." (Donald Hebb, 1940s)
- **Artificial Neural Networks** (1974)
  - ... fueled by **natural language sequences** (1990/91)
  - ... unveil patterns of **morphosyntax and semantics**
  - ... while degrading gracefully in face of noisy input.

Neural Networks

Basic Building Block: Nonlinear Transformation

$$h = f(W_{xh}x + b_h) \tag{1}$$

- f : smooth nonlinear function, e.g. logistic sigmoid or hyperbolic tangent (tanh)
- $x \in \mathbb{R}^M$: upstream layer as vector of size M
- $W_{xh} \in \mathbb{R}^{N \times M}$: weight matrix of size $N \times M$ for connection from upstream layer to hidden layer
- $b_h \in \mathbb{R}^N$: bias of size N for affine transformation of hidden layer
- $h \in \mathbb{R}^N$: hidden layer

Deep Learning

Stacked hidden layers: “Deep Learning”

Family of Deep Learning Architectures

- Standard Feedforward Networks ("Multi-Layer Perceptrons" (MLP))
- Convolutional Neural Networks (CNN)
- Recurrent Neural Networks (RNN)
- Recursive Neural Networks

Extensions and Modifications

- Long Short-Term Memory cells (LSTM)
- Memory
- Attention
- Reinforcement Learning ("delayed gratification training")

How Deep Neural Networks Learn

- Deep Neural Networks are basically
  - pattern recognizers
  - that learn sophisticated rules
  - in order to produce soft decisions.
- **Supervised learning** by design
  - One input, one target, one error, one update
  - ("instant gratification training")
- Task: "How do I analyse the input with respect to a target?"
  - Many-to-one classifiers and many-to-many transducers (e.g. for Machine Translation) just change the definition of input and target.

The Data Problem of DL-NLP

Recently at ACL conferences, there has been an over-focus on numbers, on beating the state of the art. Call it playing the Kaggle game. More of the field's effort should go into problems, approaches, and architectures.

I would encourage everyone to think about problems, architectures, cognitive science, and the details of human language, how it is learned, processed, and how it changes, rather than just chasing state-of-the-art numbers on a benchmark task.

Christoph Manning, 2016

- 1990s: Shift from analytical research to empirical research
  - Big data + generic architectures + high-level performance evaluation
- Some *high-level tasks* have benefited a lot.
  - E.g. Speech Recognition and Machine Translation
- *"Less routine" tasks* not as much.
  - E.g. POS Tagging, Named-Entity Recognition, Document Classification, robust semantic/syntactic/morphological parsing

- How do we get the food for our hungry NNs?
- **Unsupervised learning** would help, but is unpractical for high-level NLP tasks.
    - Maybe we can leverage our tasks via the inherent structure of unannotated data?
    - Example: language model

Semi-supervised Deep Learning

- No manual annotations.
  - Features are implicit and have to be learned by the system.
- DL-NLP models become fancy symbol correlation models, tuned to a specific task.
  - (Works great for English.)

Trend:

- Completely discard linguistics and annotated features.
  - Rely solely on correlations hidden in tons of data.
  - “End-to-End systems”
- Use generic NN architectures for everything.

  Natural Language Processing (almost) from Scratch“, Collobert & Weston, 2011

- Seems to work: Low-resource NN language models (5K tokens) still perform better than n-gram models.

> *I get pitched regularly by startups doing "generic machine learning" which is, in all honesty, a pretty ridiculous idea. Machine learning is not undifferentiated heavy lifting [...] and closer to design than coding.*

Joseph Reisinger (`http://thedatamines.com/post/13177389506/why-generic-machine-learning-fails`)

Although current deep learning research tends to claim to encompass NLP, I'm (1) much less convinced about the strength of the results, compared to the results in, say, vision; (2) much less convinced in the case of NLP than, say, vision, the way to go is to couple huge amounts of data with black-box learning architectures.

Michael Jordan (`https://www.reddit.com/r/MachineLearning/comments/2fxi6v/ama_michael_i_jordan`)

The Language Problem of DL-NLP

Basic Assumptions of End-to-End DL-NLP

- Language is a sequence of distinct symbols.
- Their order yields sufficient information for syntax.
- Correlations yield sufficient information for symbol meaning.
  - (Or at least make up for lack of features.)
- Sentence meaning is a nonlinear transformation of symbol meanings.

Morphology Becomes A Challenge

- English is highly **analytic**: low morpheme-per-word ratio
- Russian is highly **synthetic**
- Turkish is highly **agglutinating**

Syntax Becomes A Challenge

(e.g. for sentence branching)

- English is mostly **right-branching**: main subject is followed by modifiers and additional information
- Chinese is mostly **left-branching**.

(e.g. for Japanese)
“Mary was made by John to buy a book.”

```
Mary-ga John-ni hon-o kaw-sase-rare-ta.
Mary-ga hon-o John-ni kaw-sase-rare-ta.
John-ni Mary-ga hon-o kaw-sase-rare-ta.
John-ni hon-o Mary-ga kaw-sase-rare-ta.
Hon-o Mary-ga John-ni kaw-sase-rare-ta.
Hon-o John-ni Mary-ga kaw-sase-rare-ta.
```

Summary of Problems

- **Data Problem**: Neural Networks require huge resources (both data and power).
- **Language Problem**: Generic Deep Learning is not a natural fit for NLP.
  - NLP works off symbols for complex information.
  - The more features, the harder the acquisition of training data.

Perspectives of Low-Resource Tasks of DL-NLP

Challenges

- Large NNs require tons of training data, i.e. annotated samples.
- NLP tasks furthermore require rich features per symbol, not just correlations.

Character-Based Neural Network Models

Idea

- Instead of learning on word-level, we learn on character-level.
- Word representations are learned automatically on deeper layers.

The unreasonable effectiveness of recurrent neural networks, Karpathy et al., 2015

Benefits

- Significantly reduces the representation space for 1-of-k encodings.
  - E.g. instead of a word vocabulary of 100,000, we have a character vocabulary of 50.
  - (Though in practice, this does not give much benefit in speed, only model complexity.)
- Allows flexibility in terms of morphology.
  - Words are allowed to differ more and less.
- Good solution to the morphology problem, if combined with appropriate network architectures
  - ... and possibly something better than orthographic characters.

Character-Aware Neural Language Models, Kim et al., 2015

Downsides

- Benefits in terms of parameter count are a bit offset by more updates and deeper architectures.
  - Possibly longer trainings
  - Increased data requirements for acceptable convergence

Toy Tasks

Idea

- Synthetic data sets of tightly controlled complexity help develop better techniques/designs in order to “escape the local minima in algorithm space”.

  Towards AI-Complete Question Answering: A Set of Prerequisite Toy Tasks, Weston et al., 2015

- Used to concept-proof Memory Networks
- Obvious problems: limited domain, non-scalable, unrealistic, cognitively implausible

Alternatives to Toy Tasks

- Crowd-sourcing of annotated data
- Real-life documents -> templates -> multiplied against a fixed vocabulary -> more training data

... but they are just delaying the underlying problem: we are bound to hit a wall of feasibility if we depend on algorithms that need boatloads of data.
Transfer Learning/ Model Adaptation/ Multitask Learning

Idea

Knowledge learned about one data/task leverages/kickstarts knowledge acquisition about another data/task.

Deep Transfer Learning

- The point of **Deep** Learning is to learn higher abstractions over the data.
- Idea: How about re-using the parameters of these deeper layers?

  Transfer Learning for Speech and Language Processing, Wang & Zheng, 2015

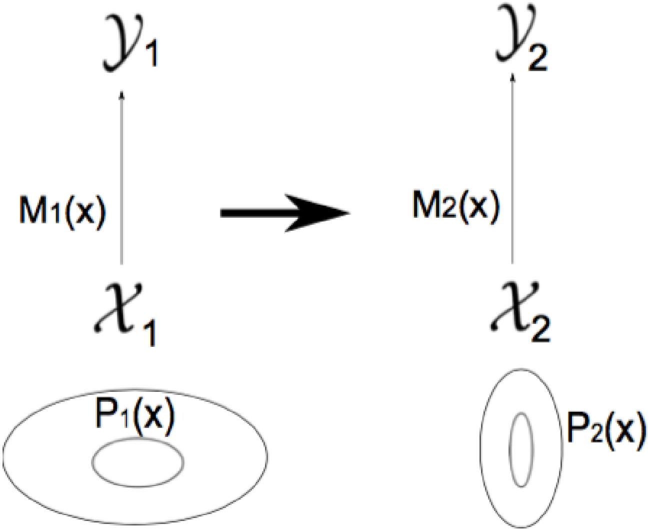

Figure 1: Relation of conditional factors in transfer learning paradigm

		$\mathcal{Y}+$		$\mathcal{Y}-$
		M(x)+	M(x) -	
$\mathcal{X}+$	P(X)+	Conventional ML	Model transfer[10]	Multitask learning[11]
	P(X)-	Model Adaptation[12], [13], incremental learning[14]		
$\mathcal{X}-$			Co-training[15] Heterogeneous transfer learning[16], [17]	Analogy learning [18]

Figure 2: Categories of Deep Transfer Learning

Transfer of learned parameters is possible between different:

- Tasks (Multitask Learning)
- Languages (e.g. speaker adaptation & multilingual speech recognition)
- Neural models (even in different depths)
- Modalities
- ML Algorithms (?)
- . . .

Idea

- Pre-train on large corpus, refine on small corpus.
  - Based on assumption of equivalence/similarity between both data representations.
  - Example: cross-lingual domain adaption for dependency parsing
    - … aided by parallel data for constraint transfer (e.g. a bilingual dictionary)
    - Two NN parsers share parameters at higher levels of abstraction.

Combining labeled and unlabeled data with co-training, Blum et al., 1998

Cross-language parser adaptation between related languages, Zeman et al., 2008

Multi-Source Transfer of Delexicalized Dependency Parsers, McDonald et al., 2011

A Neural Network Model for Low-Resource Universal Dependency Parsing, Duong et al., 2015

Example for Model Adaptation: Word Embeddings

- Sparse 1-of-k symbol representations are translated to dense low-dimensional vectors (“word embeddings”)
  - ... by tuning a randomly initialized lookup table on a specific task.
- Basically best-practice today: initialize your symbol lookup table with language model word embeddings
  - ... efficiently trained on large corpora
  - ... and possibly updated during further training on the new task.

Conc lusion

- Deep Learning is powerful, but Neural Networks need data.
- The Deep Learning community (even for NLP) has widely neglected:
  - Low-resource languages
  - Linguistics in general (e.g. morphology and grounding)
- Promising next steps:
  - Data generation techniques (e.g. from templates or generative DCGs)
  - Transfer Learning (including Word Embeddings)
    - Related languages kickstart each other's models.
    - Shared deep layers provide the bridge between (small) datasets.