Deep Learning for NLP
(without Magic)

Richard Socher and Christopher Manning
Stanford University
NAACL 2013, Atlanta
http://nlp.stanford.edu/courses/NAACL2013/

*with a big thank you to Yoshua Bengio, with whom we participated in the previous ACL 2012 version of this tutorial*
Discussion: Simple RNN

• Good results with single matrix RNN (more later)

• Single weight matrix RNN could capture some phenomena but not adequate for more complex, higher order composition and parsing long sentences

• The composition function is the same for all syntactic categories, punctuation, etc
Solution: Syntactically-Untied RNN

- Idea: Condition the composition function on the syntactic categories, “untie the weights”
- Allows for different composition functions for pairs of syntactic categories, e.g. Adv + AdjP, VP + NP
- Combines discrete syntactic categories with continuous semantic information
Solution: CVG = PCFG + Syntactically-Untied RNN

- Problem: Speed. Every candidate score in beam search needs a matrix-vector product.

- Solution: Compute score using a linear combination of the log-likelihood from a simple PCFG + RNN
  - Prunes very unlikely candidates for speed
  - Provides coarse syntactic categories of the children for each beam candidate

- Compositional Vector Grammars: CVG = PCFG + RNN
Details: Compositional Vector Grammar

- Scores at each node computed by combination of PCFG and SU-RNN:

\[ s\left( p^{(1)} \right) = (v^{(B,C)})^T p^{(1)} + \log P(P_1 \rightarrow B \ C) \]

- Interpretation: Factoring discrete and continuous parsing in one model:

\[ P((P_1, p_1) \rightarrow (B, b)(C, c)) = P(p_1 \rightarrow b \ c|P_1 \rightarrow B \ C) P(P_1 \rightarrow B \ C) \]

- Socher et al (2013): More details at ACL
Related Work

• Resulting CVG Parser is related to previous work that extends PCFG parsers
• Klein and Manning (2003a) : manual feature engineering
• Petrov et al. (2006) : learning algorithm that splits and merges syntactic categories
• Lexicalized parsers (Collins, 2003; Charniak, 2000): describe each category with a lexical item
• Hall and Klein (2012) combine several such annotation schemes in a factored parser.
• CVGs extend these ideas from discrete representations to richer continuous ones
• Hermann & Blunsom (2013): Combine Combinatory Categorial Grammars with RNNs and also untie weights, see upcoming ACL 2013
Experiments

- Standard WSJ split, labeled F1
- Based on simple PCFG with fewer states
- Fast pruning of search space, few matrix-vector products
- 3.8% higher F1, 20% faster than Stanford parser

<table>
<thead>
<tr>
<th>Parser</th>
<th>Test, All Sentences</th>
</tr>
</thead>
<tbody>
<tr>
<td>Stanford PCFG, (Klein and Manning, 2003a)</td>
<td>85.5</td>
</tr>
<tr>
<td>Stanford Factored (Klein and Manning, 2003b)</td>
<td>86.6</td>
</tr>
<tr>
<td>Factored PCFGs (Hall and Klein, 2012)</td>
<td>89.4</td>
</tr>
<tr>
<td>Collins (Collins, 1997)</td>
<td>87.7</td>
</tr>
<tr>
<td>SSN (Henderson, 2004)</td>
<td>89.4</td>
</tr>
<tr>
<td>Berkeley Parser (Petrov and Klein, 2007)</td>
<td>90.1</td>
</tr>
<tr>
<td>CVG (RNN) (Socher et al., ACL 2013)</td>
<td>85.0</td>
</tr>
<tr>
<td>CVG (SU-RNN) (Socher et al., ACL 2013)</td>
<td>90.4</td>
</tr>
<tr>
<td>Charniak - Self Trained (McClosky et al. 2006)</td>
<td>91.0</td>
</tr>
<tr>
<td>Charniak - Self Trained-ReRanked (McClosky et al. 2006)</td>
<td>92.1</td>
</tr>
</tbody>
</table>
SU-RNN Analysis

- Learns notion of soft head words

DT-NP

VP-NP
Analysis of resulting vector representations

All the figures are adjusted for seasonal variations
1. All the numbers are adjusted for seasonal fluctuations
2. All the figures are adjusted to remove usual seasonal patterns

Knight-Ridder wouldn’t comment on the offer
1. Harsco declined to say what country placed the order
2. Coastal wouldn’t disclose the terms

Sales grew almost 7% to $UNK m. from $UNK m.
1. Sales rose more than 7% to $94.9 m. from $88.3 m.
2. Sales surged 40% to UNK b. yen from UNK b.
SU-RNN Analysis

• Can transfer semantic information from single related example

• Train sentences:
  • He eats spaghetti with a fork.
  • She eats spaghetti with pork.

• Test sentences
  • He eats spaghetti with a spoon.
  • He eats spaghetti with meat.
SU-RNN Analysis

(a) Stanford factored parser

(b) Compositional Vector Grammar
Labeling in Recursive Neural Networks

- We can use each node’s representation as features for a softmax classifier:

\[ p(c|p) = \text{softmax}(Sp) \]

- Training similar to model in part 1 with standard cross-entropy error + scores
Scene Parsing

Similar principle of compositionality.

- The meaning of a scene image is also a function of smaller regions,
- how they combine as parts to form larger objects,
- and how the objects interact.
Algorithm for Parsing Images

Same Recursive Neural Network as for natural language parsing!
(Socher et al. ICML 2011)
# Multi-class segmentation

<table>
<thead>
<tr>
<th>Method</th>
<th>Accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pixel CRF (Gould et al., ICCV 2009)</td>
<td>74.3</td>
</tr>
<tr>
<td>Classifier on superpixel features</td>
<td>75.9</td>
</tr>
<tr>
<td>Region-based energy (Gould et al., ICCV 2009)</td>
<td>76.4</td>
</tr>
<tr>
<td>Local labelling (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>76.9</td>
</tr>
<tr>
<td>Superpixel MRF (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>77.5</td>
</tr>
<tr>
<td>Simultaneous MRF (Tighe &amp; Lazebnik, ECCV 2010)</td>
<td>77.5</td>
</tr>
<tr>
<td>Recursive Neural Network</td>
<td><strong>78.1</strong></td>
</tr>
</tbody>
</table>

Stanford Background Dataset (Gould et al. 2009)
Recursive Deep Learning

1. Motivation
2. Recursive Neural Networks for Parsing
3. Theory: Backpropagation Through Structure
4. Compositional Vector Grammars: Parsing
5. Recursive Autoencoders: Paraphrase Detection
6. Matrix-Vector RNNs: Relation classification
7. Recursive Neural Tensor Networks: Sentiment Analysis
Semi-supervised Recursive Autoencoder

- To capture sentiment and solve antonym problem, add a softmax classifier
- Error is a weighted combination of reconstruction error and cross-entropy
- Socher et al. (EMNLP 2011)
Paraphrase Detection

• Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses.
• Basically, the plaintiffs did not show that omissions in Merrill’s research caused the claimed losses.

• The initial report was made to Modesto Police December 28.
• It stems from a Modesto police report.
How to compare the meaning of two sentences?
Unsupervised Recursive Autoencoders

• Similar to Recursive Neural Net but instead of a supervised score we compute a reconstruction error at each node. Socher et al. (EMNLP 2011)

\[
E_{rec}([c_1; c_2]) = \frac{1}{2} \left\| [c_1; c_2] - [c'_1; c'_2] \right\|^2
\]

\[
y_2 = f(W[x_1; y_1] + b)
\]

\[
y_1 = f(W[x_2; x_3] + b)
\]
Unsupervised unfolding RAE

- Attempt to encode entire tree structure at each node
Recursive Autoencoders for Full Sentence Paraphrase Detection

- Unsupervised Unfolding RAE and a pair-wise sentence comparison of nodes in parsed trees
- Socher et al. (NIPS 2011)
Recursive Autoencoders for Full Sentence Paraphrase Detection

- Experiments on Microsoft Research Paraphrase Corpus
- (Dolan et al. 2004)

<table>
<thead>
<tr>
<th>Method</th>
<th>Acc.</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>Rus et al. (2008)</td>
<td>70.6</td>
<td>80.5</td>
</tr>
<tr>
<td>Mihalcea et al. (2006)</td>
<td>70.3</td>
<td>81.3</td>
</tr>
<tr>
<td>Islam et al. (2007)</td>
<td>72.6</td>
<td>81.3</td>
</tr>
<tr>
<td>Qiu et al. (2006)</td>
<td>72.0</td>
<td>81.6</td>
</tr>
<tr>
<td>Fernando et al. (2008)</td>
<td>74.1</td>
<td>82.4</td>
</tr>
<tr>
<td>Wan et al. (2006)</td>
<td>75.6</td>
<td>83.0</td>
</tr>
<tr>
<td>Das and Smith (2009)</td>
<td>73.9</td>
<td>82.3</td>
</tr>
<tr>
<td>Das and Smith (2009) + 18 Surface Features</td>
<td>76.1</td>
<td>82.7</td>
</tr>
<tr>
<td>F. Bu et al. (ACL 2012): String Re-writing Kernel</td>
<td>76.3</td>
<td>--</td>
</tr>
<tr>
<td>Unfolding Recursive Autoencoder (NIPS 2011)</td>
<td>76.8</td>
<td>83.6</td>
</tr>
</tbody>
</table>
# Recursive Autoencoders for Full Sentence Paraphrase Detection

<table>
<thead>
<tr>
<th>L</th>
<th>Pr</th>
<th>Sentences</th>
<th>Sim.Mat.</th>
</tr>
</thead>
</table>
| P | 0.95 | (1) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion - Australian football - as the world champion relaxed before his Wimbledon title defence  
(2) LLEYTON Hewitt yesterday traded his tennis racquet for his first sporting passion-Australian rules football-as the world champion relaxed ahead of his Wimbledon defence |         |
| P | 0.82 | (1) The lies and deceptions from Saddam have been well documented over 12 years  
(2) It has been well documented over 12 years of lies and deception from Saddam |         |
| P | 0.67 | (1) Pollack said the plaintiffs failed to show that Merrill and Blodget directly caused their losses  
(2) Basically, the plaintiffs did not show that omissions in Merrill’s research caused the claimed losses |         |
| N | 0.49 | (1) Prof Sally Baldwin, 63, from York, fell into a cavity which opened up when the structure collapsed at Tiburtina station. Italian railway officials said  
(2) Sally Baldwin, from York, was killed instantly when a walkway collapsed and she fell into the machinery at Tiburtina station |         |
| N | 0.44 | (1) Bremer, 61, is a onetime assistant to former Secretaries of State William P. Rogers and Henry Kissinger and was ambassador-at-large for counterterrorism from 1986 to 1989  
(2) Bremer, 61, is a former assistant to former Secretaries of State William P. Rogers and Henry Kissinger |         |
| N | 0.11 | (1) The initial report was made to Modesto Police December 28  
(2) It stems from a Modesto police report |         |
Recursive Deep Learning

1. Motivation
2. Recursive Neural Networks for Parsing
3. Theory: Backpropagation Through Structure
4. Compositional Vector Grammars: Parsing
5. Recursive Autoencoders: Paraphrase Detection
6. Matrix-Vector RNNs: Relation classification
7. Recursive Neural Tensor Networks: Sentiment Analysis
Compositionality Through Recursive Matrix-Vector Spaces

One way to make the composition function more powerful was by untying the weights $W$.

But what if words act mostly as an operator, e.g. “very” in “very good”?

Proposal: A new composition function

$$p = \tanh(W \begin{pmatrix} c_1+ \\ c_2 \end{pmatrix} b)$$
Compositionality Through Recursive Matrix-Vector Recursive Neural Networks

\[ p = \tanh(W \begin{pmatrix} c_1 \\ c_2 \end{pmatrix} + b) \]

Recursive Matrix-Vector Model

\[ f(Ba, Ab) = \]

\[ Ba = \]

\[ Ab = \]

\[ \ldots \]

\[ (a, A) \]

\[ (b, B) \]

\[ (c, C) \]

- vector

- matrix
Predicting Sentiment Distributions

- Good example for non-linearity in language
MV-RNN for Relationship Classification

<table>
<thead>
<tr>
<th>Relationship</th>
<th>Sentence with labeled nouns for which to predict relationships</th>
</tr>
</thead>
<tbody>
<tr>
<td>Cause-Effect(e2,e1)</td>
<td>Avian [influenza]_e1 is an infectious disease caused by type a strains of the influenza [virus]_e2.</td>
</tr>
<tr>
<td>Entity-Origin(e1,e2)</td>
<td>The [mother]_e1 left her native [land]_e2 about the same time and they were married in that city.</td>
</tr>
<tr>
<td>Message-Topic(e2,e1)</td>
<td>Roadside [attractions]_e1 are frequently advertised with [billboards]_e2 to attract tourists.</td>
</tr>
</tbody>
</table>

---

<table>
<thead>
<tr>
<th>Classifier</th>
<th>Feature Sets</th>
<th>F1</th>
</tr>
</thead>
<tbody>
<tr>
<td>SVM</td>
<td>POS, stemming, syntactic patterns</td>
<td>60.1</td>
</tr>
<tr>
<td>SVM</td>
<td>word pair, words in between</td>
<td>72.5</td>
</tr>
<tr>
<td>SVM</td>
<td>POS, WordNet, stemming, syntactic patterns</td>
<td>74.8</td>
</tr>
<tr>
<td>SVM</td>
<td>POS, WordNet, morphological features, thesauri, Google n-grams</td>
<td>77.6</td>
</tr>
<tr>
<td>MaxEnt</td>
<td>POS, WordNet, morphological features, noun compound system, thesauri, Google n-grams</td>
<td>77.6</td>
</tr>
<tr>
<td>SVM</td>
<td>POS, WordNet, prefixes and other morphological features, POS, dependency parse features, Levin classes, PropBank, FrameNet, NomLex-Plus, Google n-grams, paraphrases, TextRunner</td>
<td>82.2</td>
</tr>
<tr>
<td>RNN</td>
<td>-</td>
<td>74.8</td>
</tr>
<tr>
<td>Lin.MVR</td>
<td>-</td>
<td>73.0</td>
</tr>
<tr>
<td>MV-RNN</td>
<td>-</td>
<td>79.1</td>
</tr>
<tr>
<td>RNN</td>
<td>POS,WordNet,NER</td>
<td>77.6</td>
</tr>
<tr>
<td>Lin.MVR</td>
<td>POS,WordNet,NER</td>
<td>78.7</td>
</tr>
<tr>
<td>MV-RNN</td>
<td>POS,WordNet,NER</td>
<td>82.4</td>
</tr>
</tbody>
</table>
Sentiment Detection

• Sentiment detection is crucial to business intelligence, stock trading, ...
Sentiment Detection and Bag-of-Words Models

• Most methods start with a bag of words + linguistic features/processing/lexica

• But such methods (including tf-idf) can’t distinguish:
  - white blood cells destroying an infection
  - an infection destroying white blood cells
Sentiment Detection and Bag-of-Words Models

- Sentiment is that sentiment is “easy”
- Detection accuracy for longer documents ~90%
- Lots of easy cases (… horrible… or … awesome …)

- For dataset of single sentence movie reviews (Pang and Lee, 2005) accuracy never reached above 80% for >7 years

- Harder cases require actual understanding of negation and its scope and other semantic effects
Stealing Harvard doesn't care about cleverness, wit or any other kind of intelligent humor.

There are slow and repetitive parts but it has just enough spice to keep it interesting.
Two missing pieces for improving sentiment

1. Compositional Training Data

2. Better Compositional model
1. New Sentiment Treebank
1. New Sentiment Treebank

- Parse trees of 11,855 sentences
- 215,154 phrases with labels
- Allows training and evaluating with compositional information
2. New Compositional Model

- Recursive Neural Tensor Network
- More expressive than any other RNN so far
- Idea: Allow more interactions of vectors

\[
\begin{pmatrix}
  b \\
  c
\end{pmatrix}^T V
\begin{pmatrix}
  b \\
  c
\end{pmatrix}
\]
2. New Compositional Model

- Recursive Neural Tensor Network
2. New Compositional Model

- Recursive Neural Tensor Network
Recursive Neural Tensor Network

\[ p = f \left( \begin{bmatrix} b \\ c \end{bmatrix} V^{[1:2]} \begin{bmatrix} b \\ c \end{bmatrix} \right) + W \begin{bmatrix} b \\ c \end{bmatrix} \]
Experimental Result on Treebank

<table>
<thead>
<tr>
<th>Model</th>
<th>Fine-grained</th>
<th>Positive/Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>All</td>
<td>Root</td>
</tr>
<tr>
<td>NB</td>
<td>67.2</td>
<td>41.0</td>
</tr>
<tr>
<td>SVM</td>
<td>64.3</td>
<td>40.7</td>
</tr>
<tr>
<td>BiNB</td>
<td>71.0</td>
<td>41.9</td>
</tr>
<tr>
<td>VecAvg</td>
<td>73.3</td>
<td>32.7</td>
</tr>
<tr>
<td>RNN</td>
<td>79.0</td>
<td>43.2</td>
</tr>
<tr>
<td>MV-RNN</td>
<td>78.7</td>
<td>44.4</td>
</tr>
<tr>
<td>RNTN</td>
<td><strong>80.7</strong></td>
<td><strong>45.6</strong></td>
</tr>
</tbody>
</table>
Experimental Result on Treebank

- RNTN can capture X but Y
- RNTN accuracy of 72%, compared to MV-RNN (65), biNB (58) and RNN (54)
Negation Results

Negation of "Roger Dodger is one of the most compelling variations on this theme".

Negation of "Roger Dodger is one of the least compelling variations on this theme".

Negation of "One of the year is the most significant moviegoing pleasures".

Negation of "One of the year is the most significant moviegoing pleasures".
Negation Results

- Most methods capture that negation often makes things more negative (See Potts, 2010)
- Analysis on negation dataset
Negation Results

- But how about negating negatives?
- Positive activation should increase!

<table>
<thead>
<tr>
<th>Model</th>
<th>Negated Positive</th>
<th>Negated Negative</th>
</tr>
</thead>
<tbody>
<tr>
<td>biNB</td>
<td>19.0</td>
<td>27.3</td>
</tr>
<tr>
<td>RNN</td>
<td>33.3</td>
<td>45.5</td>
</tr>
<tr>
<td>MV-RNN</td>
<td>52.4</td>
<td>54.6</td>
</tr>
<tr>
<td>RNTN</td>
<td>71.4</td>
<td>90.9</td>
</tr>
</tbody>
</table>

**Negated Positive Sentences:** Change in Activation

- biNB: 0.01
- RNN: -0.01
- MV-RNN: -0.34
- RNTN: -0.5

**Negated Negative Sentences:** Change in Activation

- biNB: 0.01
- RNN: 0.01
- MV-RNN: -0.01
- RNTN: +0.25
Overview of RNN Model Variations

• Objective Functions
  • **Supervised Scores for Structure Prediction**
  • **Classifier for Sentiment, Relations, Visual Objects, Logic**
  • **Unsupervised autoencoding immediate children** or entire tree structure

• Composition Functions
  • **Syntactically-Untied Weights**
  • **Matrix Vector RNN**
  • **Tensor-Based Models**

• Tree Structures
  • **Constituency Parse Trees**
  • Combinatory Categorical Grammar Trees
  • Dependency Parse Trees
  • Fixed Tree Structures (Connections to CNNs)
Summary: Recursive Deep Learning

- Recursive Deep Learning can predict hierarchical structure and classify the structured output using compositional vectors
- State-of-the-art performance (all with code on www.socher.org)
  - Parsing on the WSJ (Java code soon)
  - Sentiment Analysis on multiple corpora
  - Paraphrase detection with unsupervised RNNs
  - Relation Classification on SemEval 2011, Task8
  - Object detection on Stanford background and MSRC datasets
Part 3.2

Deep Learning
General Strategy and Tricks
Non-linearities: What’s used

logistic ("sigmoid")
\[ f(z) = \frac{1}{1 + \exp(-z)}. \]
\[ f'(z) = f(z)(1 - f(z)) \]

\[ \text{tanh} \]
\[ f(z) = \tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}, \]
\[ f'(z) = 1 - f(z)^2 \]

tanh is just a rescaled and shifted sigmoid
\[ \tanh(z) = 2 \text{logistic}(2z) - 1 \]
tanh is what is most used and often performs best for deep nets
Non-linearities: There are various other choices

- hard tanh
  \[
  \text{HardTanh}(x) = \begin{cases} 
  -1 & \text{if } x < -1 \\
  x & \text{if } -1 \leq x \leq 1 \\
  1 & \text{if } x > 1 
  \end{cases}
  \]

- soft sign
  \[
  \text{softsign}(z) = \frac{a}{1 + |a|}
  \]

- rectifier
  \[
  \text{rect}(z) = \max(z, 0)
  \]

- hard tanh similar but computationally cheaper than tanh and saturates hard.
- [Glorot and Bengio AISTATS 2010, 2011] discuss softsign and rectifier
MaxOut Network

- A very recent type of nonlinearity/network
- Goodfellow et al. (2013)

\[ f_i(z) = \max_{j \in [1, k]} z_{ij} \]

- Where \[ z_{ij} = x^T W_{..ij} + b_{ij} \]

- This function too is a universal approximator
- State of the art on several image datasets
Gradient Checks are Awesome!

- Allows you to know that there are no bugs in your neural network implementation!
- Steps:
  1. Implement your gradient
  2. Implement a finite difference computation by looping through the parameters of your network, adding and subtracting a small epsilon ($\sim 10^{-4}$) and estimate derivatives

$$g_i(\theta) \approx \frac{J(\theta^{(i+)}) - J(\theta^{(i-)})}{2 \times \text{EPSILON}}.$$  
$$\theta^{(i+)} = \theta + \text{EPSILON} \times \vec{e}_i$$

3. Compare the two and make sure they are the same
Parameter Initialization

- Initialize hidden layer biases to 0 and output (or reconstruction) biases to optimal value if weights were 0 (e.g. mean target or inverse sigmoid of mean target).
- Initialize weights $\sim$ Uniform($-r, r$), $r$ inversely proportional to fan-in (previous layer size) and fan-out (next layer size):
  \[
  \sqrt{6/(\text{fan-in} + \text{fan-out})}
  \]
  for tanh units, and 4x bigger for sigmoid units [Glorot AISTATS 2010]
- Pre-training with Restricted Boltzmann machines
Stochastic Gradient Descent (SGD)

- Gradient descent uses total gradient over all examples per update, SGD updates after only 1 or few examples:

\[ \theta(t) \leftarrow \theta(t-1) - \epsilon_t \frac{\partial L(z_t, \theta)}{\partial \theta} \]

- \( L \) = loss function, \( z_t \) = current example, \( \theta \) = parameter vector, and \( \epsilon_t \) = learning rate.

- Ordinary gradient descent as a batch method, very slow, should never be used. Use 2\(^{nd}\) order batch method such as LBFGS. On large datasets, SGD usually wins over all batch methods. On smaller datasets LBFGS or Conjugate Gradients win. Large-batch LBFGS extends the reach of LBFGS [Le et al ICML’2011].
Learning Rates

- Simplest recipe: keep it fixed and use the same for all parameters.
- Collobert scales them by the inverse of square root of the fan-in of each neuron.
- Better results can generally be obtained by allowing learning rates to decrease, typically in $O(1/t)$ because of theoretical convergence guarantees, e.g., with hyper-parameters $\epsilon_0$ and $\tau$
  \[ \epsilon_t = \frac{\epsilon_0 \tau}{\max(t, \tau)} \]
- Better yet: No learning rates by using L-BFGS or AdaGrad (Duchi et al. 2011)
Long-Term Dependencies and Clipping Trick

• In very deep networks such as recurrent networks (or possibly recursive ones), the gradient is a product of Jacobian matrices, each associated with a step in the forward computation. This can become very small or very large quickly [Bengio et al 1994], and the locality assumption of gradient descent breaks down.

• The solution first introduced by Mikolov is to clip gradients to a maximum value. Makes a big difference in RNNs.
Prevent Overfitting: Model Size and Regularization

- Simple first step: Reduce model size by lower number of units and layers and other parameters
- Standard L1 or L2 regularization on weights
- Early Stopping: Use parameters that gave best validation error
- Sparsity constraints on hidden activations, e.g. add to cost:
  \[ KL \left( \frac{1}{N} \sum_{n=1}^{N} d_i^{(n)} \| 0.0001 \right) \]
- Dropout (Hinton et al. 2012):
  - Randomly set 50% of the inputs at each layer to 0
  - At test time half the outgoing weights (now twice as many)
  - Prevents Co-adaptation
Deep Learning Tricks of the Trade

  - Unsupervised pre-training
  - Stochastic gradient descent and setting learning rates
  - Main hyper-parameters
    - Learning rate schedule & early stopping
    - Minibatches
    - Parameter initialization
    - Number of hidden units
    - L1 or L2 weight decay
    - Sparsity regularization
  - Debugging → Finite difference gradient check (Yay)
  - How to efficiently search for hyper-parameter configurations