The Recursive Neural Network

Deep Learning Lecture

Sara Veldhoen MSc

March 3, 2016

Outline

The Model
- Variations
- Unsupervised

My Research
- Logic
- Arithmetic
- Analysis

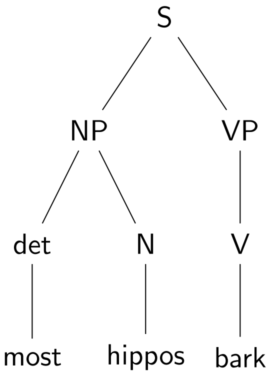
- Goal: model semantics of linguistic utterances
- Lexical distributional semantics: successful
- What about composition?
- How to deal with variable (sentence) length?
- Sequence Models: recurrent connections as memory. All the work is done by a single cell.

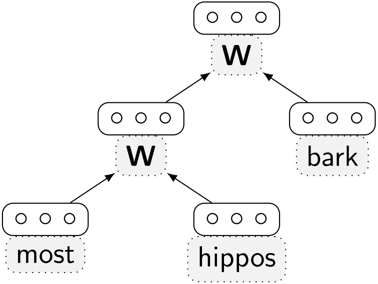
- Observation: language is structured
- Compositionality: meaning of a complex expression is a function of its parts and the way they are (syntactically) combined
- Symbolic implementation: Montague Grammar
- Distributional implementation: Recursive Neural Network (RNN) Socher *et al.* (2012)

Outline

The Model

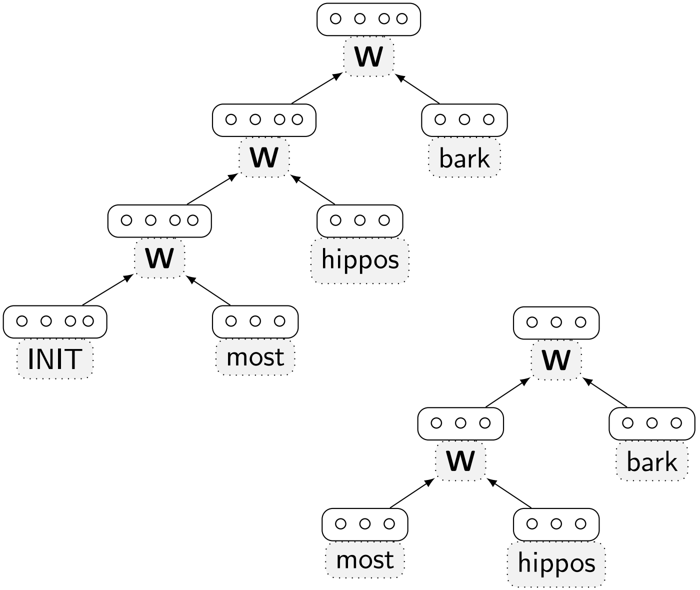
Recursive Neural Network:





$$\mathbf{p} = f(\mathbf{W} \cdot [\mathbf{c}_0; \mathbf{c}_1] + \mathbf{b})$$

The Model



The Model - Variations

Matrix-Vector (Socher *et al.* , 2012)

$$\mathbf{p}_{MV} = f(\mathbf{W}_v[\mathbf{C}_1\mathbf{c}_2; \mathbf{C}_2\mathbf{c}_1]) \tag{1}$$

$$\mathbf{P}_{MV} = \mathbf{W}_M[\mathbf{C}_1; \mathbf{C}_2] \tag{2}$$

The Model - Variations

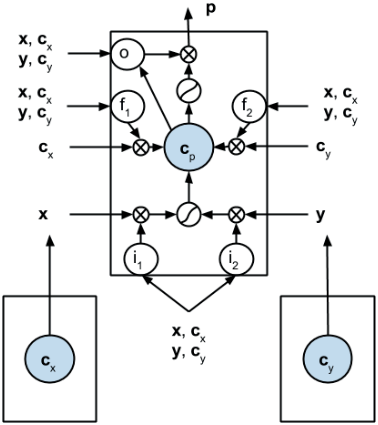
Tensor (Bowman & Potts, 2015)

$$\mathbf{p} = f(\mathbf{W} \cdot [\mathbf{c}_0; \mathbf{c}_1] + \mathbf{b}) \tag{3}$$

$$\mathbf{p}_{MV} = \mathbf{p} + f(\mathbf{c}_1^T \mathbf{T}^{[1 \ldots n]} \mathbf{c}_2) \tag{4}$$

The Model - Variations

Tree-LSTM (Le & Zuidema, 2015)

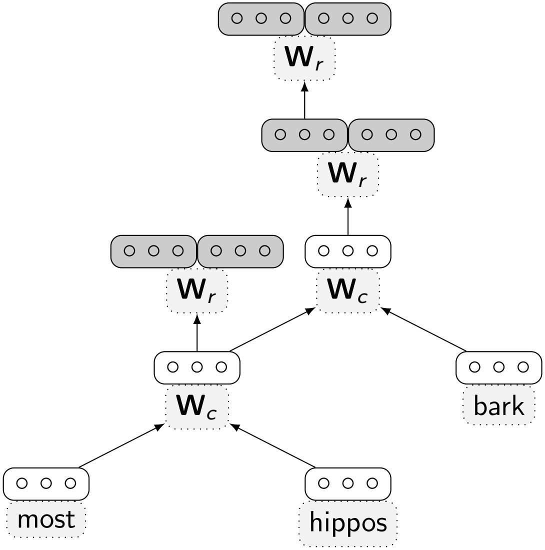


The Model - Variations

Deep composition (Socher *et al.* , 2010)
Add more layers between children and parent representation

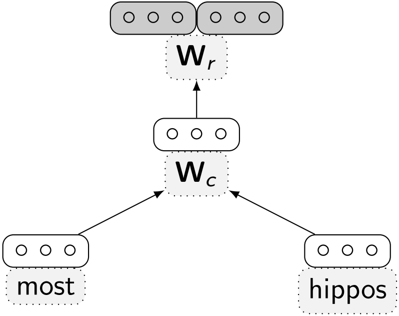
The Model - Unsupervised

(Unfolding) Recursive Auto-Encoder (Socher *et al.* , 2011)



The Model - Unsupervised

(Unfolding) Recursive Auto-Encoder (Socher *et al.* , 2011)

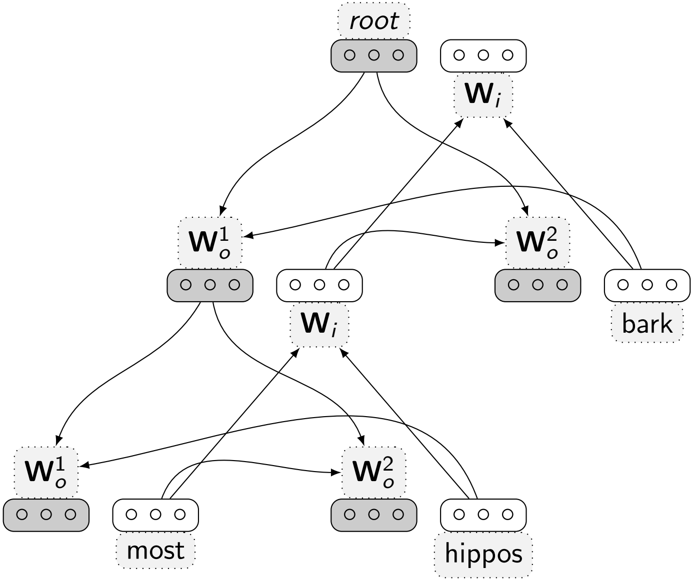


$$\mathbf{p} = f(\mathbf{W_c} \cdot [\mathbf{c}_0; \mathbf{c}_1] + \mathbf{b_c}) \tag{5}$$

$$[\mathbf{c}_0^r; \mathbf{c}_1^r] = f(\mathbf{W_r} \cdot \mathbf{p} + \mathbf{b_r}) \tag{6}$$

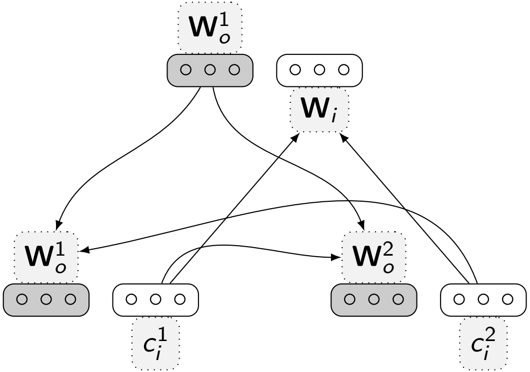
The Model - Unsupervised

Inside-Outside RNN (Le & Zuidema, 2014)



The Model - Unsupervised

Inside-Outside RNN (Le & Zuidema, 2014)



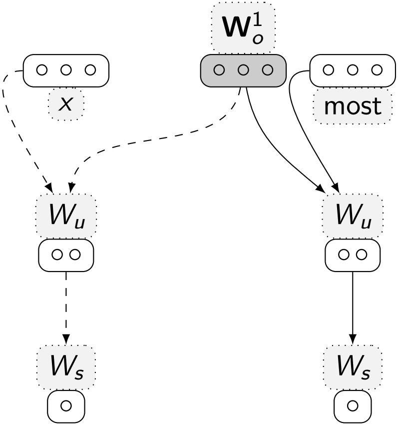
$$\mathbf{p_i} = f(\mathbf{W_i} \cdot [\mathbf{c}_i^1; \mathbf{c}_i^2] + \mathbf{b_i}) \tag{7}$$

$$\mathbf{c_o^1} = f(\mathbf{W_o^1} \cdot [\mathbf{p}_o; \mathbf{c}_i^2] + \mathbf{b_o^1}) \tag{8}$$

$$\mathbf{c_o^2} = f(\mathbf{W_o^1} \cdot [\mathbf{p}_o; \mathbf{c}_i^1] + \mathbf{b_o^2}) \tag{9}$$

The Model - Unsupervised

Inside-Outside RNN (Le & Zuidema, 2014) training:



$$c(w, x) = \max\{0, 1 - s(w, \mathbf{o_w}) + s(x, \mathbf{o_w})\} \tag{10}$$

$$s(x, \mathbf{o_w}) = \mathbf{W_s} f(\mathbf{W_u}[\mathbf{o_w}; \mathbf{i_x}] + \mathbf{b_u}) \tag{11}$$

Outline

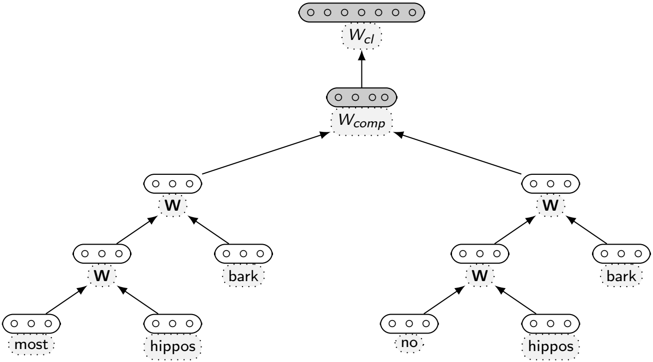
My Research - Logic

Bowman & Potts (2015) train RNN directly for NLI

- Train RNN through classifier
- Data: pairs of sentences with inference relation

(most hippo) bark	$\mid$	(no hippo) bark
(most hippo) bark	$\sqsupset$	(no hippo) (not bark)
(two hippo) bark	$\#$	(some (not hippo)) bark
(three hippo) Parisian	$\sqsubset$	(three hippo) French
(all hippo) Parisian	$\sqsubset$	(all hippo) French

My Research - Logic



My Research - Logic

Replication of Bowman & Potts (2015) and extension

	2014 Data Composition		2015 Data Composition	
	Fixed	Trained	Fixed	Trained
Fixed Embs	45.9	99.7	29.7	97.3
Trained Embs	84.8	99.6	58.2	92.9

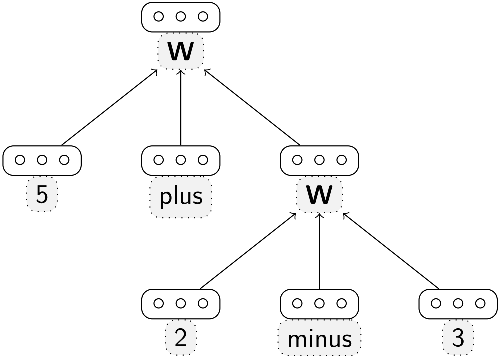
Table: The effect of fixing the word embeddings or the composition function. Accuracy (%) on the test data.

Does the model actually capture logical semantics?

My Research - Arithmetic

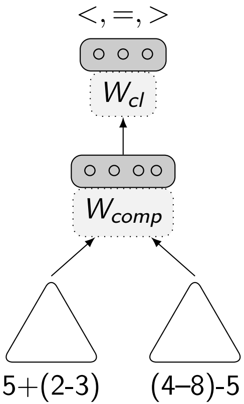
Simple task: arithmetic expression trees

Principled solution captures sense of numbers: *numerosity*



My Research - Arithmetic

Set-up: comparison layer and soft-max classifier on top of two trees



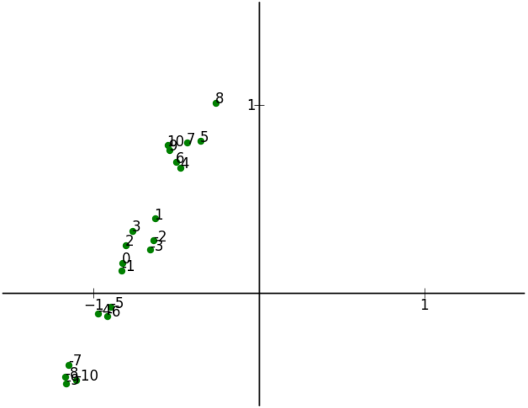
My Research - Arithmetic

Setting		Dimensionality 1	2	10
trained embs	trained comp	49.8	88.2*	97.0
	fixed comp	50.3	51.6	63.2
fixed embs	trained comp	49.0	50.7	68.5
	fixed comp	47.2	49.6	50.5

Table: Accuracy (%) on held-out data. *: the variance over different runs was less than one percentage point in all cases but one: the 2 dimensional setting had one run performing considerably worse than the others; 63.1% accuracy vs. an average of 96.5% for the rest.

My Research - Arithmetic

- The RNN can learn to do addition and subtraction, and decide which of two expressions is greater.
- What has the model learned? Is it a principled solution?
- 2-D case: can be plotted.



My Research - Analysis

Step-by-step analysis: Project-Sum-Squash

- Break up the composition function:

$$\mathbf{p} = f(\mathbf{W} \cdot [\mathbf{c}_0; \mathbf{c}_1; \mathbf{c}_2] + \mathbf{b})$$

$$\mathbf{p} = f(\mathbf{W}_0\mathbf{c}_0 + \mathbf{W}_1\mathbf{c}_1 + \mathbf{W}_2\mathbf{c}_2 + \mathbf{b})$$

- The intermediate results can be plotted

My Research - Analysis

Breaking up the composition function

$$\begin{pmatrix} a & b & c & d & e & f \\ g & h & i & j & k & l \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

$$= \begin{pmatrix} a1 + b2 + c3 + d4 + e5 + f6 \\ g1 + h2 + i3 + j4 + k5 + l6 \end{pmatrix}$$

$$= \begin{pmatrix} a1 + b2 \\ g1 + h2 \end{pmatrix} + \begin{pmatrix} c3 + d4 \\ i3 + j4 \end{pmatrix} + \begin{pmatrix} e3 + f4 \\ k3 + l4 \end{pmatrix}$$

$$= \begin{pmatrix} a & b \\ g & h \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} c & d \\ i & j \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} + \begin{pmatrix} e & f \\ k & l \end{pmatrix} \times \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

My Research - Analysis

Breaking up the composition function

$$\begin{pmatrix} a & b & c & d & e & f \\ g & h & i & j & k & l \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

$$= \begin{pmatrix} a1 + b2 + c3 + d4 + e5 + f6 \\ g1 + h2 + i3 + j4 + k5 + l6 \end{pmatrix}$$

$$= \begin{pmatrix} a1 + b2 \\ g1 + h2 \end{pmatrix} + \begin{pmatrix} c3 + d4 \\ i3 + j4 \end{pmatrix} + \begin{pmatrix} e3 + f4 \\ k3 + l4 \end{pmatrix}$$

$$= \begin{pmatrix} a & b \\ g & h \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} c & d \\ i & j \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} + \begin{pmatrix} e & f \\ k & l \end{pmatrix} \times \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

My Research - Analysis

Breaking up the composition function

$$\begin{pmatrix} a & b & c & d & e & f \\ g & h & i & j & k & l \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \\ 3 \\ 4 \\ 5 \\ 6 \end{pmatrix}$$

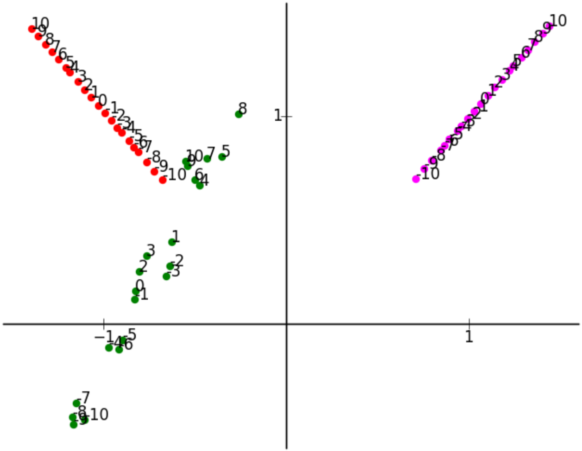
$$= \begin{pmatrix} a1 + b2 + c3 + d4 + e5 + f6 \\ g1 + h2 + i3 + j4 + k5 + l6 \end{pmatrix}$$

$$= \begin{pmatrix} a1 + b2 \\ g1 + h2 \end{pmatrix} + \begin{pmatrix} c3 + d4 \\ i3 + j4 \end{pmatrix} + \begin{pmatrix} e3 + f4 \\ k3 + l4 \end{pmatrix}$$

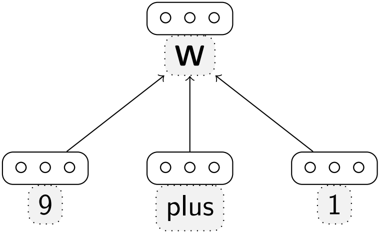
$$= \begin{pmatrix} a & b \\ g & h \end{pmatrix} \times \begin{pmatrix} 1 \\ 2 \end{pmatrix} + \begin{pmatrix} c & d \\ i & j \end{pmatrix} \times \begin{pmatrix} 3 \\ 4 \end{pmatrix} + \begin{pmatrix} e & f \\ k & l \end{pmatrix} \times \begin{pmatrix} 5 \\ 6 \end{pmatrix}$$

My Research - Analysis

Lexical embeddings and projections

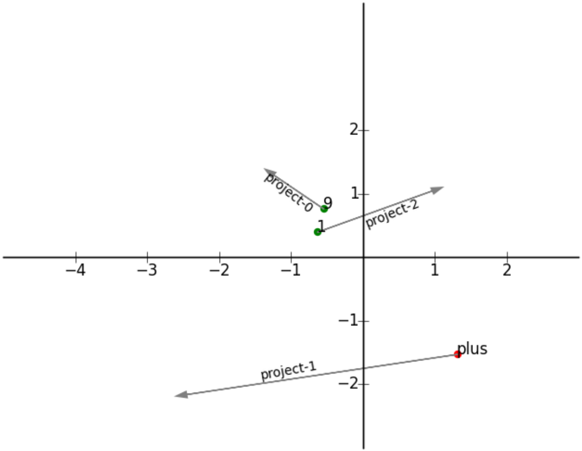


My Research - Example: 9+1



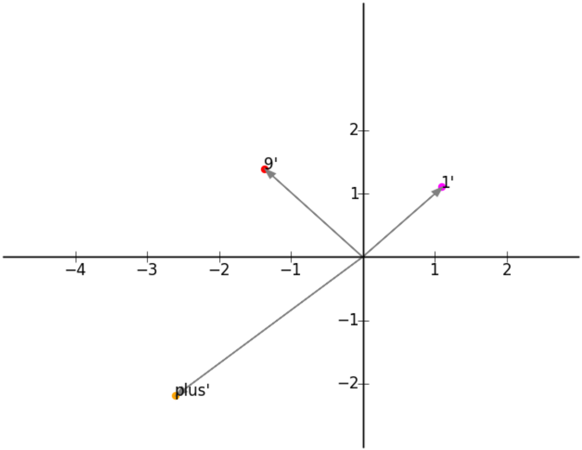
My Research - Example: 9+1

Project



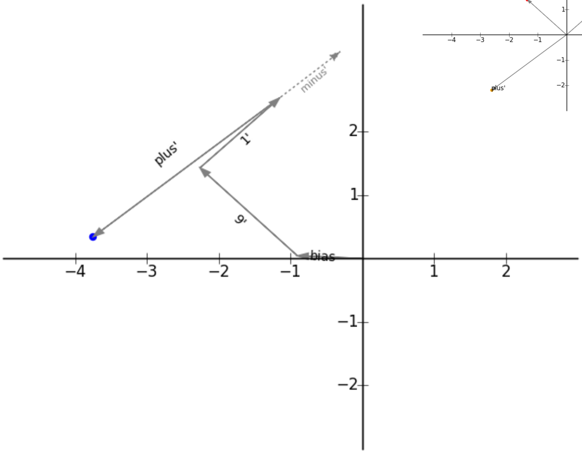
My Research - Example: 9+1

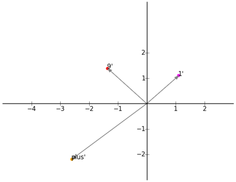
Project



My Research - Example: 9+1

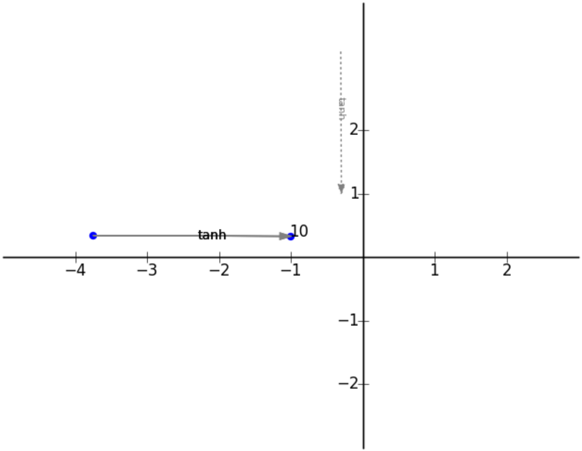
Sum





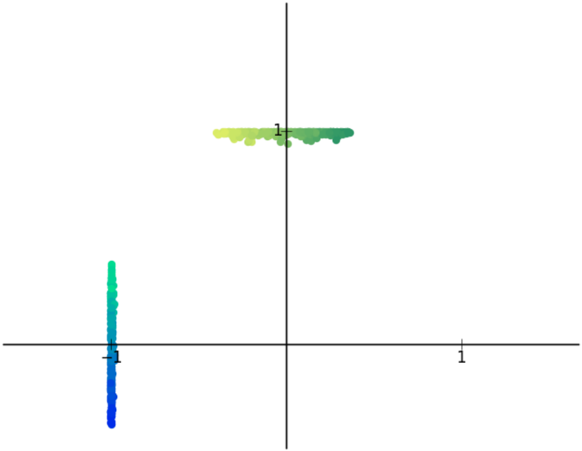
My Research - Example: 9+1

Squash



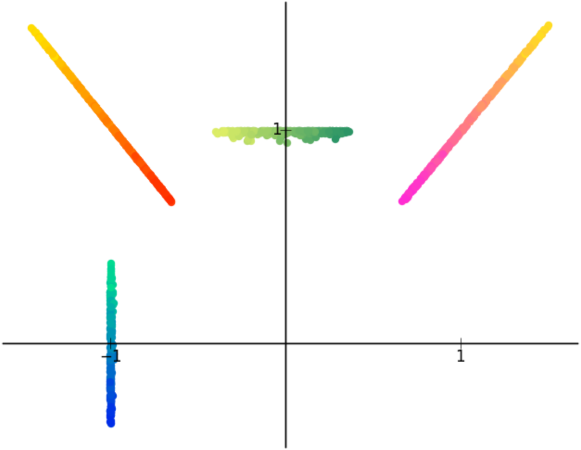
My Research -

Parent representations



My Research -

Parent representations and projections



My Research -

- Project-Sum-Squash provides information on how the model fulfills a task
  Interpretation: is it a principled solution?
- The same technique can be applied to the higher dimensional case
  One needs dimensionality reduction, e.g. PCA

My Research - Future Work

- Can the RNN really learn logical reasoning?
- Compare different composition functions
- Unsupervised training
- Language generation from sentence representations
- Reduce reliance on syntactic parse

References

Bowman, Samuel R, & Potts, Christopher. 2015. Recursive neural networks can learn logical semantics. *ACL-IJCNLP 2015*, 12.

Le, Phong, & Zuidema, Willem. 2014. Inside-Outside Semantics: A Framework for Neural Models of Semantic Composition.

Le, Phong, & Zuidema, Willem. 2015. Compositional distributional semantics with long short term memory. *arXiv preprint arXiv:1503.02510*.

Socher, Richard, Manning, Christopher D, & Ng, Andrew Y. 2010. Learning continuous phrase representations and syntactic parsing with recursive neural networks. *Pages 1–9 of: Proceedings of the NIPS-2010 Deep Learning and Unsupervised Feature Learning Workshop*.

Socher, Richard, Huang, Eric H, Pennin, Jeffrey, Manning, Christopher D, & Ng, Andrew Y. 2011. Dynamic pooling and unfolding recursive autoencoders for paraphrase detection. *Pages 801–809 of: Advances in Neural Information Processing Systems*.

Socher, Richard, Huval, Brody, Manning, Christopher D, & Ng, Andrew Y. 2012. Semantic compositionality through recursive matrix-vector spaces. *Pages 1201–1211 of: Proceedings of the 2012 Joint Conference on Empirical Methods in Natural Language Processing and Computational Natural Language Learning*. Association for Computational Linguistics.