Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA, USA

Peeking into the AI Language Modeling Black Box

Juan Luis Gastaldi

ETH zürich

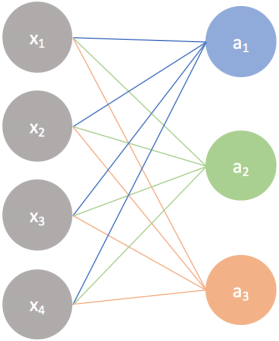
February 15th, 2023

This project has received funding from the
European Union's Horizon 2020 research and innovation programme
under grant agreement No 839730

Outline

Outline

Neural Networks

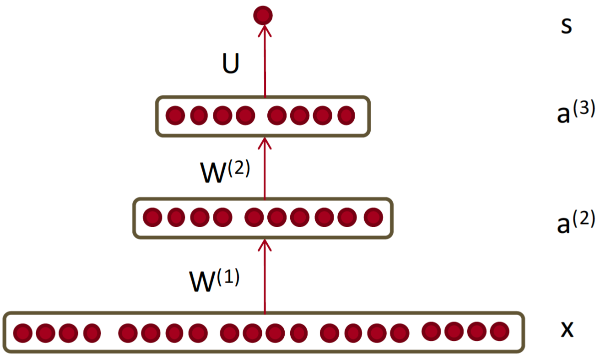


$$\begin{bmatrix} w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \\ w_1 & w_2 & w_3 & w_4 \end{bmatrix} \begin{bmatrix} x_1 \\ x_2 \\ x_3 \\ x_4 \end{bmatrix} + \begin{bmatrix} b \\ b \\ b \end{bmatrix} = \begin{bmatrix} w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \\ w_1x_1 + w_2x_2 + w_3x_3 + w_4x_4 + b \end{bmatrix} \xrightarrow[\text{activation}]{} \begin{bmatrix} a_1 \\ a_2 \\ a_3 \end{bmatrix}$$

Credit: Jeremy Jordan

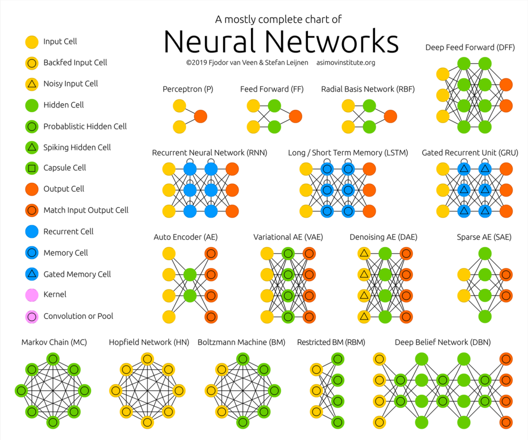
Deep Neural Nets (DNNs)

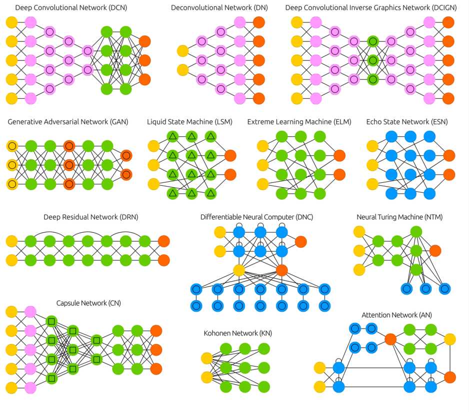
$$
\begin{aligned}
x &= z^{(1)} = a^{(1)} \\
z^{(2)} &= W^{(1)} x + b^{(1)} \\
a^{(2)} &= f\left(z^{(2)}\right) \\
z^{(3)} &= W^{(2)} a^{(2)} + b^{(2)} \\
a^{(3)} &= f\left(z^{(3)}\right) \\
s &= U^T a^{(3)}
\end{aligned}
$$



Credit: Manning & Socher, Stanford CS224n course, 2017

The Family of DNNs





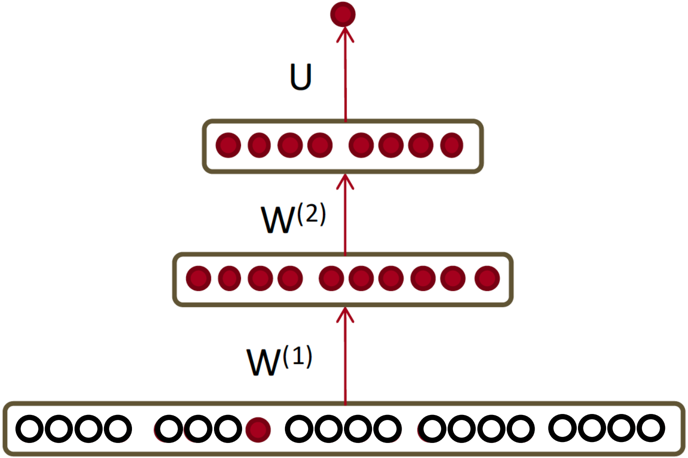
Credit: https://www.asimovinstitute.org/neural-network-zoo/

DNNs and Natural Language I

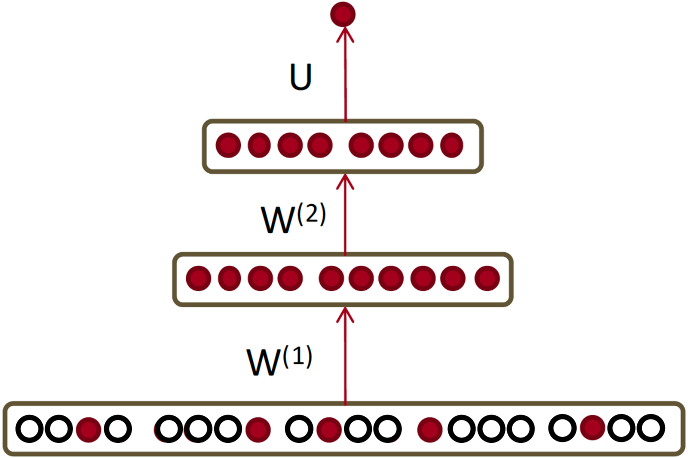
Index	Word
...	...
535	nearly
536	shares
537	member
538	campaign
539	media
540	needs
541	why
542	house
543	issues
544	costs
545	fire
...	...

$$v_{house} = (\underbrace{0, 0, 0, 0, 0, 0, 0, 0, \dots, 0, \overbrace{1}^{542^{\text{nd}}\ \text{position}}, 0, \dots, 0, 0, 0, 0, 0, 0, 0, 0}_{3\ \text{million dimensions}})$$

DNNs and Natural Language II



DNNs and Natural Language II



Outline

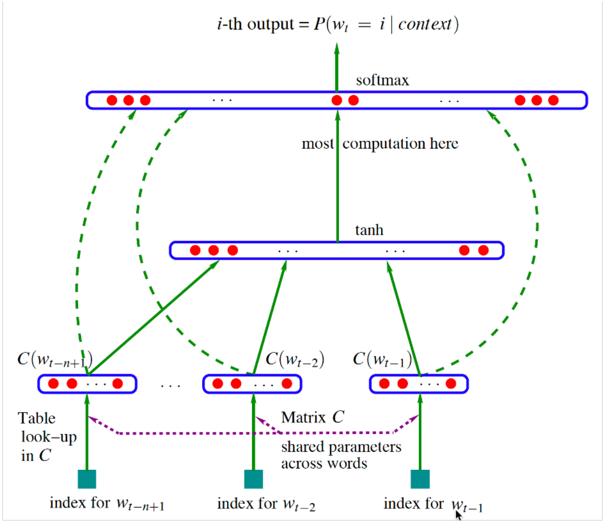
DNNs and Language

Word Embeddings

How Does It Work?

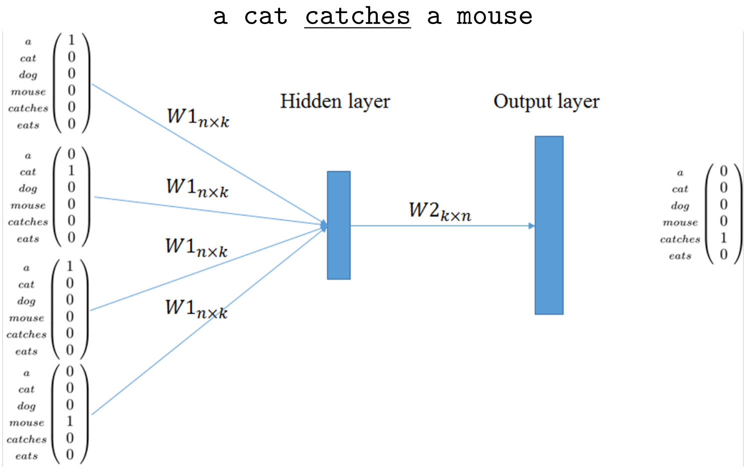
Discussion

Early Neural Language Models



(Bengio et al., 2003)

Word Embeddings: word2vec



Credit: Ferrone et al., 2017

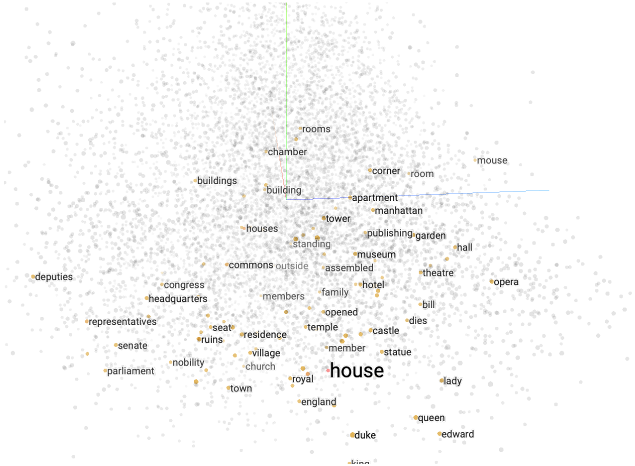
Dense Vector Representations

$$v_{house} = (\underbrace{0,0,0,0,0,0,0,0,\ldots,0,\overbrace{1}^{542^{\text{nd}}\text{ position}},0,\ldots,0,0,0,0,0,0,0,0}_{\text{million dimensions}})$$

$$v_{house} = (\underbrace{0.157227, -0.0708008, 0.0539551, \ldots, -0.041748, 0.00982666, -0.00494385, -0.032959}_{300\text{ dimensions}})$$

Word Embeddings: Similarity

house	cosine distance
houses	0.292761
bungalow	0.312144
apartment	0.3371
bedroom	0.350306
townhouse	0.361592
residence	0.380158
mansion	0.394181
farmhouse	0.414243
duplex	0.424206
homes	0.43802



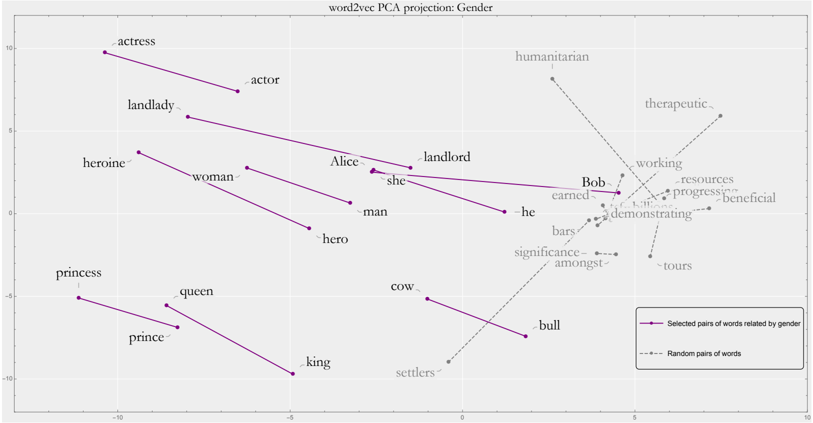
(https://projector.tensorflow.org)

Word Embeddings: Analogy

$$v_{house} - v_{city} + v_{countryside} \approx v_{farmhouse}$$

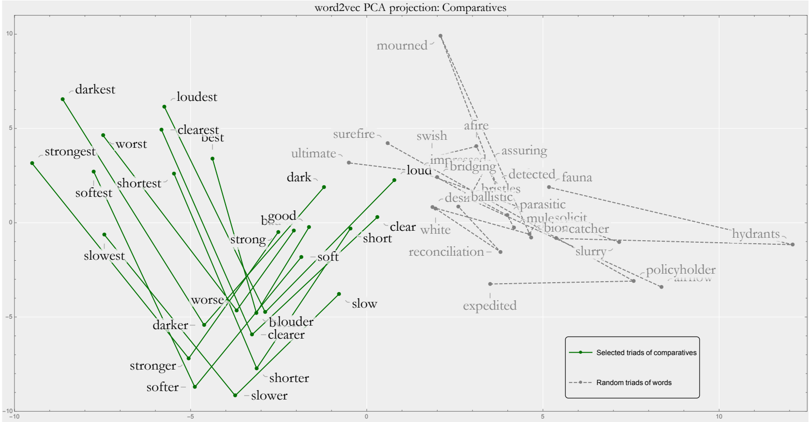
Word Embeddings: Analogy

$$v_{king} - v_{queen} \approx v_{hero} - v_{heroine}$$

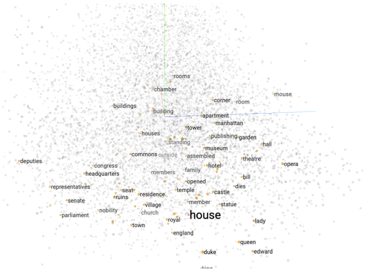


Word Embeddings: Analogy

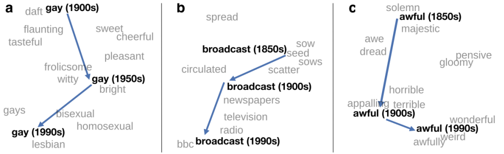
$$v_{good} - v_{better} \approx v_{soft} - v_{softer}$$



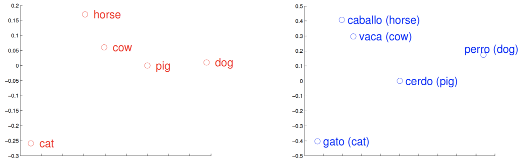
Embedding Space



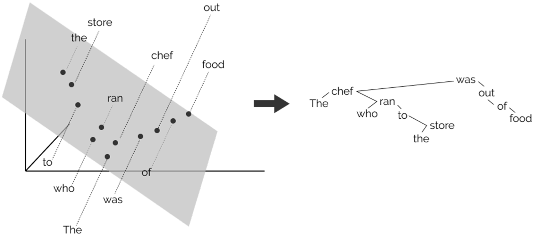
(https://projector.tensorflow.org)



(Hamilton et al., 2016)



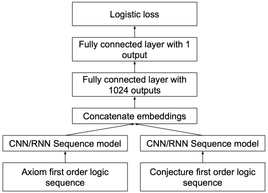
(Mikolov, Sutskever, et al., 2013)



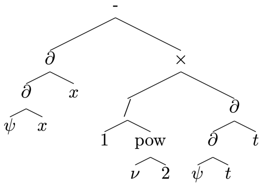
(https://nlp.stanford.edu/~johnhew/structural-probe.html)

Neural LMs Applications to Mathematics

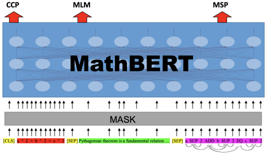
- Proof-Oriented
  - Bansal et al., 2019; Kaliszyk et al., 2017.
- Object-Oriented
  - Blechschmidt and Ernst, 2021; Charton, 2021; d'Ascoli et al., 2022; Lample and Charton, 2019; Li et al., 2021; Ryskina and Knight, 2021
- Skill-Oriented
  - Brown et al., 2020; Peng et al., 2021; Shen et al., 2021
- Heuristic-Oriented
  - Davies et al., 2021



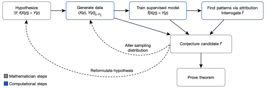
(Alemi et al., 2016)



(Lample and Charton, 2019)

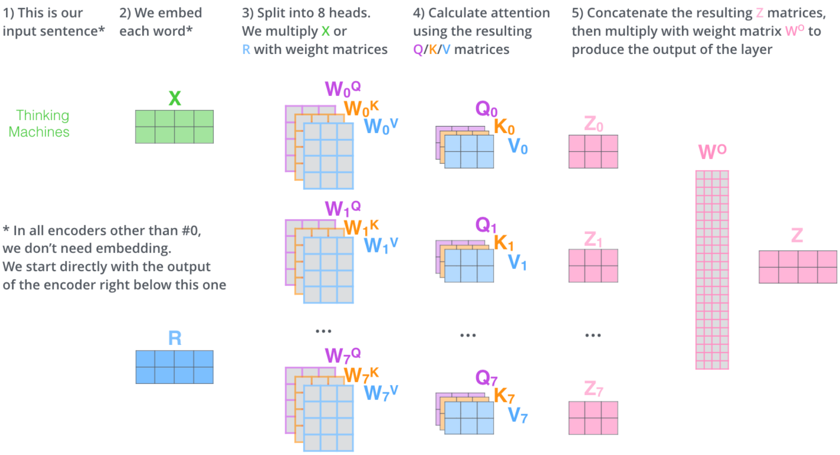


(Peng et al., 2021)



(Davies et al., 2021)

Attention Mechanism: Transformers



Credit: J. Alammar, *The Illustrated Transformer*

Outline

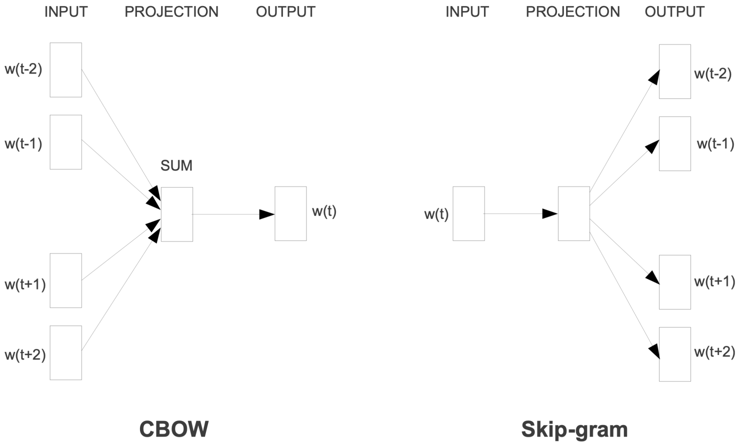
DNNs and Language

Word Embeddings

How Does It Work?

Discussion

word2vec Models



(Mikolov, Chen, et al., 2013)

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]$$

Where:

$\vec{w}$ = vector representation for word w

$\vec{c}$ = vector representation for context c

$\sigma(x)$ = $\frac{1}{1+e^{-x}}$

k = number of “negative” (arbitrary) samples

c_N = arbitrary context sampled from P_D

$P_D(c)$ = empirical unigram distribution of c in the data D, i.e. $\frac{\#(c)}{|D|}$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w, c)\left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]\right)$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w,c)\left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]\right)$$

$$\frac{\partial \ell}{\partial (\vec{w} \cdot \vec{c})} = 0$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w, c)\left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]\right)$$

$$\frac{\partial \ell}{\partial(\vec{w} \cdot \vec{c})} = 0 \quad \text{when} \quad \vec{w} \cdot \vec{c} = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w,c)\left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]\right)$$

$$\frac{\partial \ell}{\partial(\vec{w} \cdot \vec{c})} = 0 \quad \text{when} \quad \vec{w} \cdot \vec{c} = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

$$= PMI(w,c) - \log k$$

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w,c) \left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D} [\log \sigma(-\vec{w} \cdot \vec{c}_N)] \right)$$

$$\frac{\partial \ell}{\partial (\vec{w} \cdot \vec{c})} = 0 \quad \text{when} \quad \vec{w} \cdot \vec{c} = \log \left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)} \right) - \log k$$

$$= PMI(w,c) - \log k$$

Additional constraint: $\vec{w}$ and $\vec{c}$ should be low dimensional

word2vec as Implicit Matrix Factorization

(Levy and Goldberg, 2014)

$$\ell = \sum_{w \in V_w} \sum_{c \in V_c} \#(w,c)\left(\log \sigma(\vec{w} \cdot \vec{c}) + k \cdot \mathbb{E}_{c_N \sim P_D}[\log \sigma(-\vec{w} \cdot \vec{c}_N)]\right)$$

$$\frac{\partial \ell}{\partial(\vec{w} \cdot \vec{c})} = 0 \quad \text{when} \quad \vec{w} \cdot \vec{c} = \log\left(\frac{\#(w,c) \cdot |D|}{\#(w) \cdot \#(c)}\right) - \log k$$

$$= PMI(w,c) - \log k$$

Additional constraint: $\vec{w}$ and $\vec{c}$ should be low dimensional

There exists an exact solution …

Singular Value Decomposition (SVD)

$$M = U\Sigma V^*$$

Where:

$$\begin{aligned} M &= m \times n \text{ (real or complex) matrix} \\ U &= m \times m \text{ unitary matrix} \\ \Sigma &= m \times n \text{ non-negative real rectangular diagonal matrix} \\ V^* &= \text{conjugate transpose of } V\text{, a } n \times n \text{ unitary matrix} \end{aligned}$$

Singular Value Decomposition (SVD)

$$M = U\Sigma V^*$$

Where:

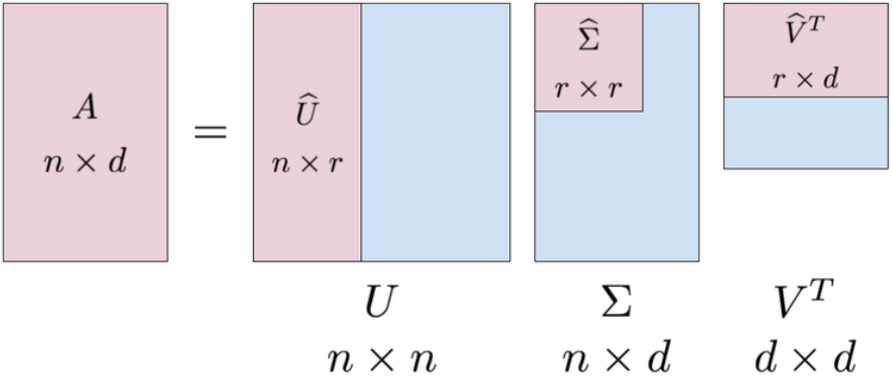
$$\begin{aligned} M &= m \times n \text{ (real or complex) matrix} \\ U &= m \times m \text{ unitary matrix} \\ \Sigma &= m \times n \text{ non-negative real rectangular diagonal matrix} \\ V^* &= \text{conjugate transpose of } V\text{, a } n \times n \text{ unitary matrix} \end{aligned}$$

In particular:

- The columns of U (left singular vectors) are eigenvectors of MM^*
- The rows of V^* (right singular values) are eigenvectors of M^*M
- The non-zero elements of Σ (non-zero singular values) are the square roots of the non-zero eigenvalues of MM^* or M^*M

Singular Value Decomposition (SVD)

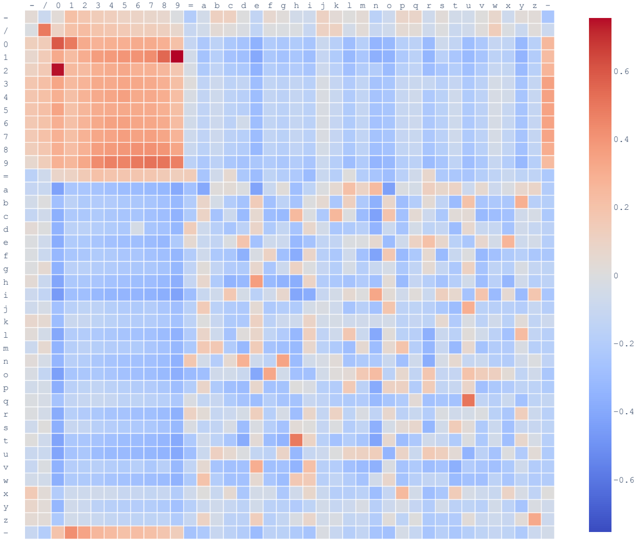
$$M = U\Sigma V^*$$



Credit: Angela Ju

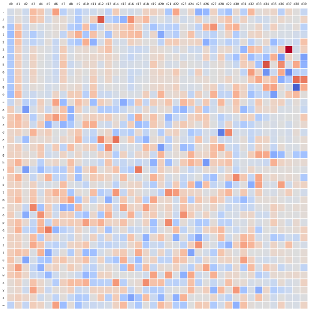
Example: Characters in Wikipedia

$$PMI(w,c) = \log \frac{p(w,c)}{p(w)p(c)}$$

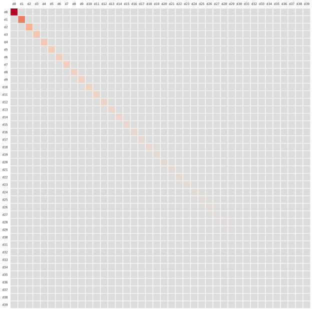


Example: Characters in Wikipedia

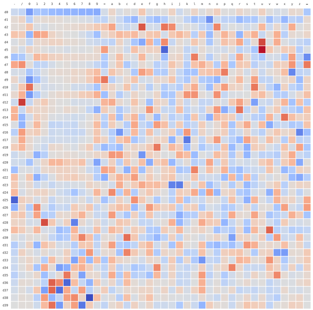
U



Σ

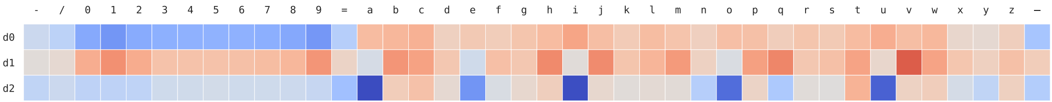


V^*

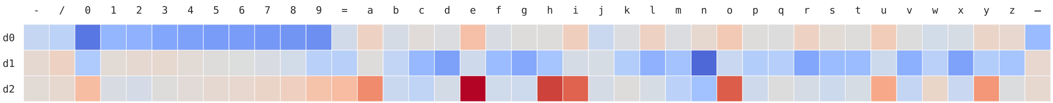


Example: Characters in Wikipedia

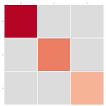
Left Singular Vectors:



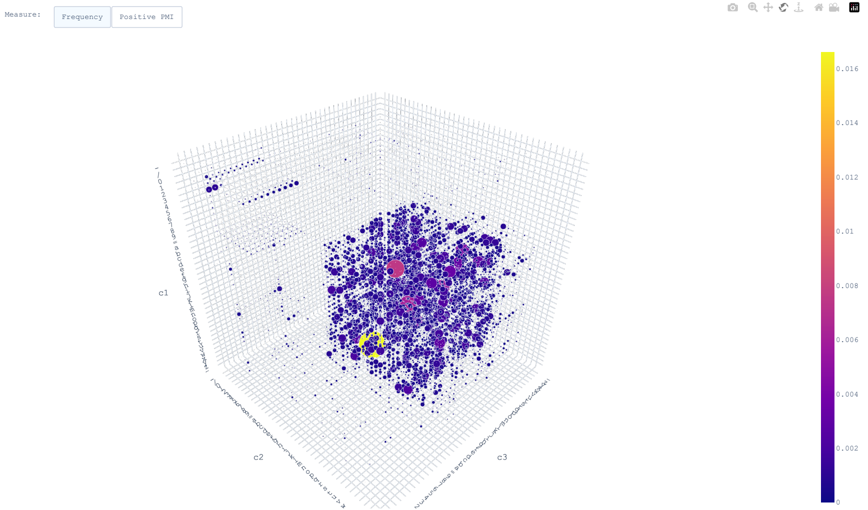
Right Singular Vectors:



Singular Values:



Generalization



Outline

Discussion

- Elementary properties of neural NLP do not depend on the neural nature of models
- There seem to be algebraic structures underlying NL data
- Explicit and symbolic representations could be built upon such structures
- Such representations could be used to explore the capabilities and limits of current neural models
- They could also provide new interpretability principles and techniques
- We need new tools at the interface between algebra and statistics

Reference Paper

- J. L. Gastaldi. Why Can Computers Understand Natural Language? In: *Philosophy & Technology* 34.1 (2021), pp. 149–214.

Related Papers

- J. L. Gastaldi and L. Pellissier. The calculus of language: explicit representation of emergent linguistic structure through type-theoretical paradigms In: *Interdisciplinary Science Reviews* 46.4 (2021), pp. 569–590.
- J. L. Gastaldi, Content from Expressions: The Place of Textuality in Deep Learning Approaches to Mathematics Under review at *Synthese. SI: Linguistically Informed Philosophy of Mathematics.* Fisseni, B., Kant, D., Sarikaya, D. and Schröder, B. (Eds.).

References I

Alemi, A. A., Chollet, F., Een, N., Irving, G., Szegedy, C., & Urban, J. (2016). Deepmath - deep sequence models for premise selection. *Proceedings of the 30th International Conference on Neural Information Processing Systems*, 2243–2251.

Bansal, K., Loos, S. M., Rabe, M. N., Szegedy, C., & Wilcox, S. (2019). Holist: An environment for machine learning of higher-order theorem proving (extended version). *CoRR*, *abs/1904.03241*. http://arxiv.org/abs/1904.03241

Bengio, Y., Ducharme, R., Vincent, P., & Janvin, C. (2003). A neural probabilistic language model. *J. Mach. Learn. Res.*, *3*, 1137–1155.

Blechschmidt, J., & Ernst, O. G. (2021). Three ways to solve partial differential equations with neural networks — a review. *GAMM-Mitteilungen*, *44*(2), e202100006. https://doi.org/https://doi.org/10.1002/gamm.202100006

Brown, T. B., Mann, B., Ryder, N., Subbiah, M., Kaplan, J., Dhariwal, P., Neelakantan, A., Shyam, P., Sastry, G., Askell, A., Agarwal, S., Herbert-Voss, A., Krueger, G., Henighan, T., Child, R., Ramesh, A., Ziegler, D. M., Wu, J., Winter, C., … Amodei, D. (2020). Language models are few-shot learners.

Charton, F. (2021). Linear algebra with transformers. *CoRR*, *abs/2112.01898*. https://arxiv.org/abs/2112.01898

d'Ascoli, S., Kamienny, P., Lample, G., & Charton, F. (2022). Deep symbolic regression for recurrent sequences. *CoRR*, *abs/2201.04600*.

Davies, A., Veličković, P., Buesing, L., Blackwell, S., Zheng, D., Tomašev, N., Tanburn, R., Battaglia, P., Blundell, C., Juhász, A., Lackenby, M., Williamson, G., Hassabis, D., & Kohli, P. (2021). Advancing mathematics by guiding human intuition with AI. *Nature*, *600*(7887), 70–74. https://doi.org/10.1038/s41586-021-04086-x

Hamilton, W. L., Leskovec, J., & Jurafsky, D. (2016). Diachronic word embeddings reveal statistical laws of semantic change. *CoRR*, *abs/1605.09096*.

References II

Kaliszyk, C., Chollet, F., & Szegedy, C. (2017). Holstep: A machine learning dataset for higher-order logic theorem proving. *CoRR*, *abs/1703.00426*. http://arxiv.org/abs/1703.00426

Lample, G., & Charton, F. (2019). Deep learning for symbolic mathematics.

Levy, O., & Goldberg, Y. (2014). Neural word embedding as implicit matrix factorization. *Proceedings of the 27th International Conference on Neural Information Processing Systems - Volume 2*, 2177–2185.

Li, Z., Kovachki, N. B., Azizzadenesheli, K., liu, B., Bhattacharya, K., Stuart, A., & Anandkumar, A. (2021). Fourier neural operator for parametric partial differential equations. *International Conference on Learning Representations*. https://openreview.net/forum?id=c8P9NQVtmnO

Mikolov, T., Chen, K., Corrado, G., & Dean, J. (2013). Efficient estimation of word representations in vector space. *CoRR*, *abs/1301.3781*.

Mikolov, T., Sutskever, I., Chen, K., Corrado, G., Dean, J., Le, Q., & Strohmann, T. (2013). *Learning representations of text using neural networks. NIPS deep learning workshop 2013 slides*.

Peng, S., Yuan, K., Gao, L., & Tang, Z. (2021). Mathbert: A pre-trained model for mathematical formula understanding. *CoRR*, *abs/2105.00377*. https://arxiv.org/abs/2105.00377

Ryskina, M., & Knight, K. (2021). Learning mathematical properties of integers. *Proceedings of the Fourth BlackboxNLP Workshop on Analyzing and Interpreting Neural Networks for NLP*, 389–395. https://doi.org/10.18653/v1/2021.blackboxnlp-1.30

Shen, J. T., Yamashita, M., Prihar, E., Heffernan, N. T., Wu, X., & Lee, D. (2021). Mathbert: A pre-trained language model for general NLP tasks in mathematics education. *CoRR*, *abs/2106.07340*.

Department of Philosophy
Carnegie Mellon University
Pittsburgh, PA, USA

Peeking into the AI Language Modeling Black Box

Juan Luis Gastaldi

ETH zürich

February 15th, 2023

This project has received funding from the
European Union's Horizon 2020 research and innovation programme
under grant agreement No 839730