CMPUT 651 (Fall 2019)

Word Embeddings
&
Language Modeling

Lili Mou
lmou@ualberta.ca
lili-mou.github.io

UNIVERSITY OF ALBERTA

Last Lecture

- Logistic regression/Softmax: Linear classification

- Non-linear classification
  - Non-linear feature engineering
  - Non-linear kernel
  - Non-linear function composition

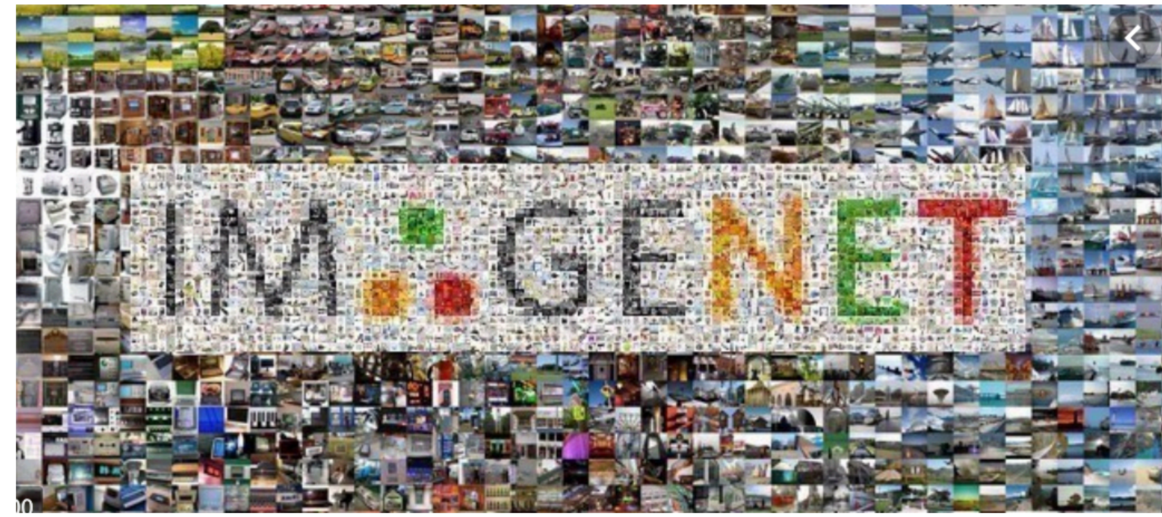
- Neural networks
  - Forward propagation: Compute activation
  - Backward propagation: Compute derivative

    (greedy dynamic programming)

UNIVERSITY OF ALBERTA

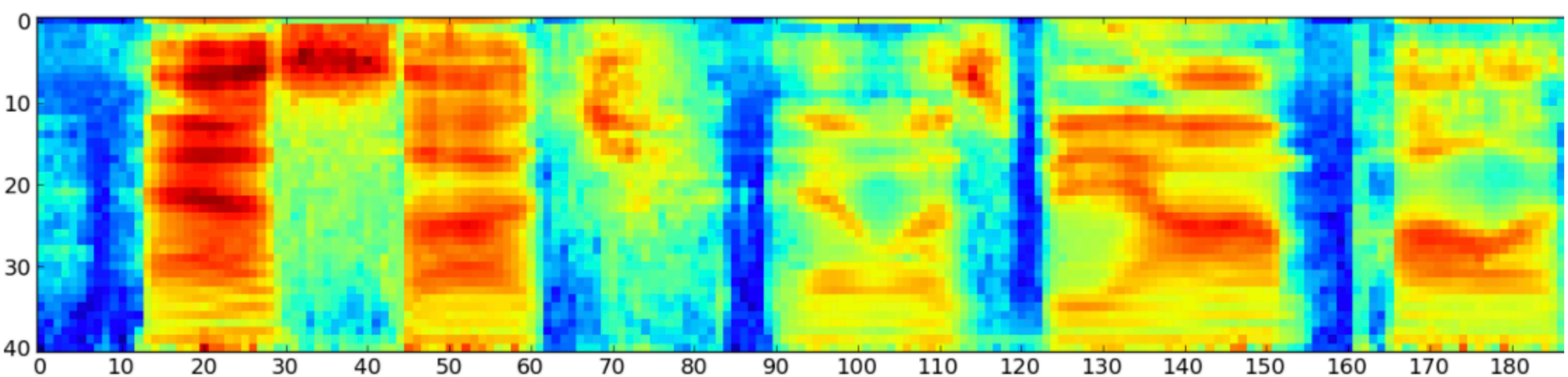
Advantages of DL

- Work with raw data
  - Images processing: pixels



ImageNet

  - Speech processing: frequency



[Graves+, ICASSP'13]

UNIVERSITY OF ALBERTA

How about Language?

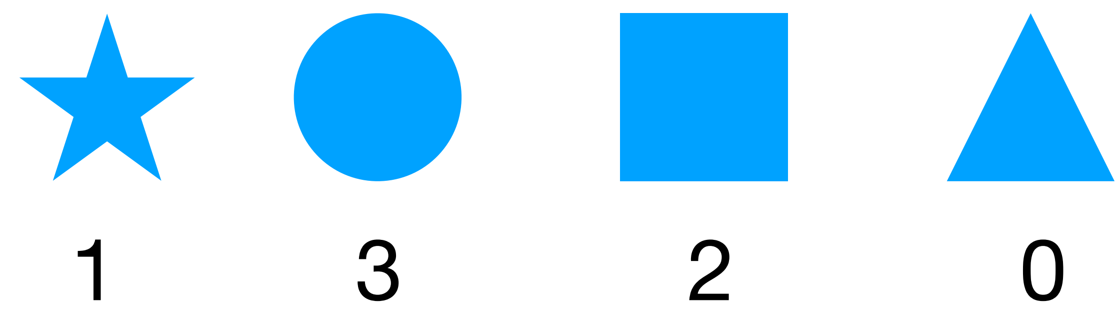
- The raw input of language

  I like the course

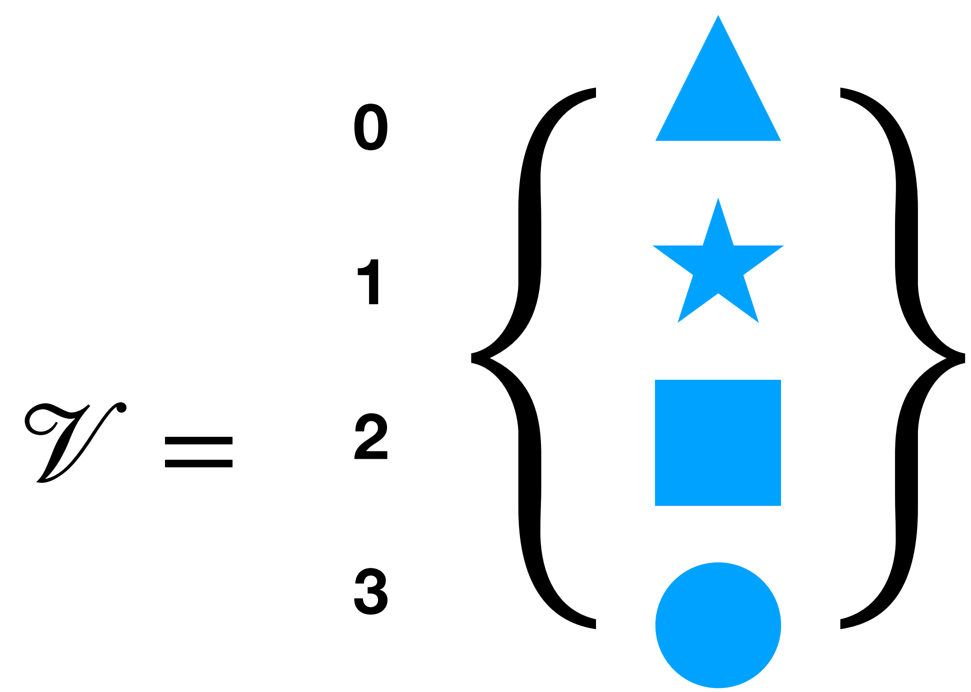
- Problem: **Words are discrete tokens**!

Representing Words

- **Attempt#1:**
  - By index in the vocabulary



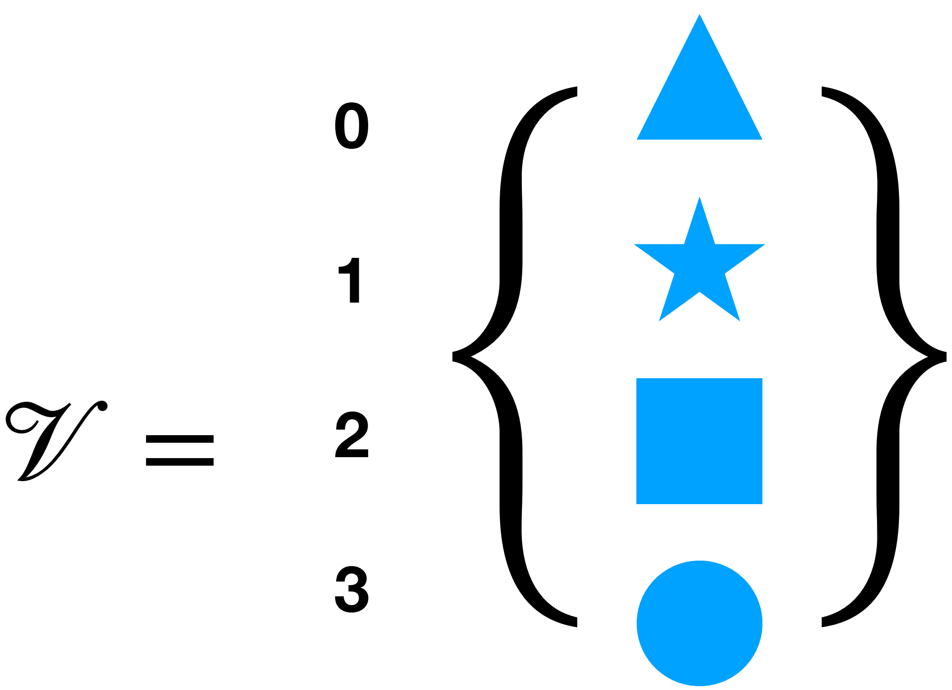
- Problem
  - Introducing artefacts
    - Order, metric, inner-product
    - Extreme non-linearity



Representing Words

- **Attempt#2:** One-hot representation
  - X Separability doesn't generalize
  - X Metric is trivial

$$\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \quad \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

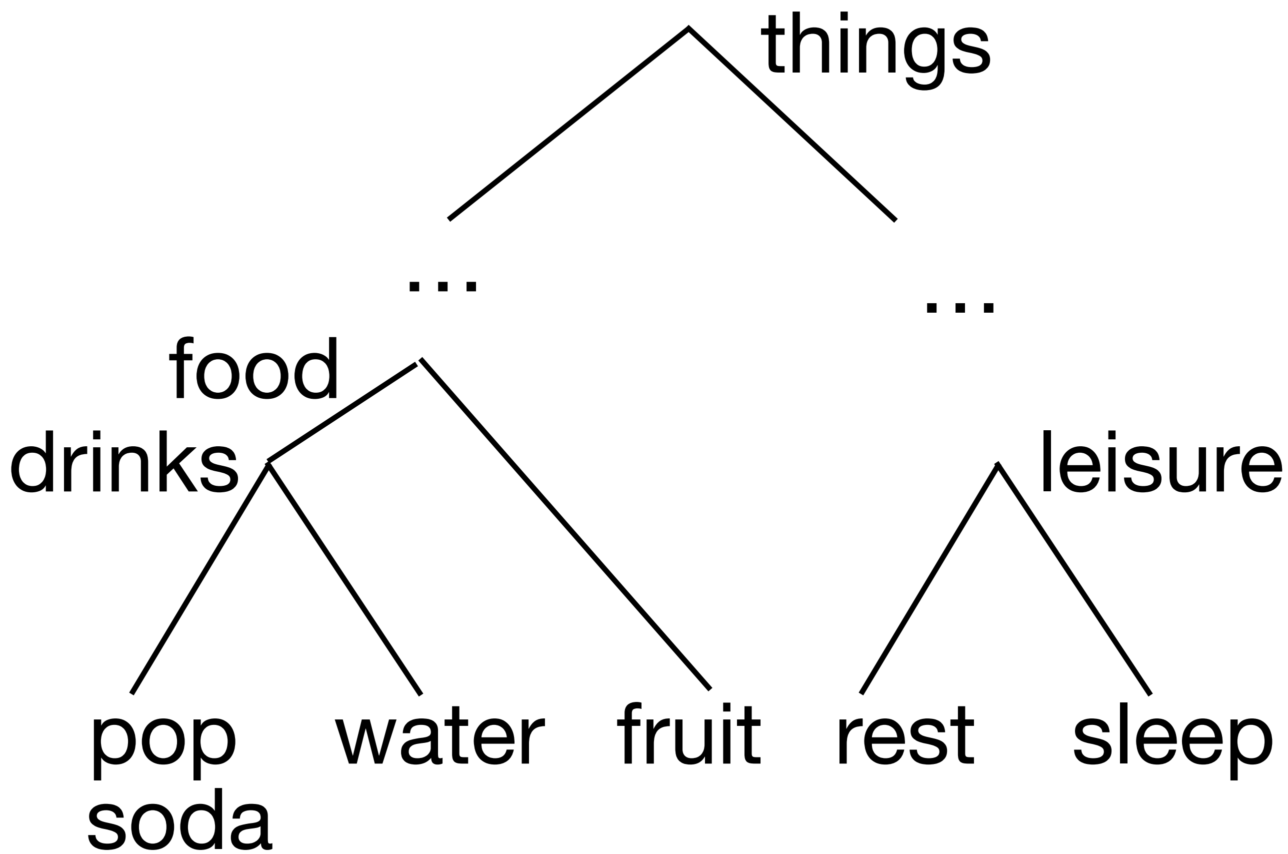


Metric in the Word Space

- Design a **metric** $d(\cdot, \cdot)$ to evaluate the "distance" of two words in terms of some aspect
  - E.g., semantic similarity

I'd like to have some pop/soda/water/fruit/rest

- Traditional method: WordNet distance (if it's a metric).



If not, doesn't matter.

$$\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \quad \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

Metric in the Word Space

- Design a **metric** $d(\cdot\,,\cdot)$ to evaluate the “distance” of two words in terms of some aspect
  - E.g., semantic similarity

I’d like to have some pop/soda/water/fruit/rest

- A straightforward metric on one-hot vector:
  - Discrete metric

$d(\boldsymbol{x}_i, \boldsymbol{x}_j) = 1$ if $\boldsymbol{x}_i = \boldsymbol{x}_j$, 0 otherwise

$$\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \quad \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \quad \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \quad \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$$

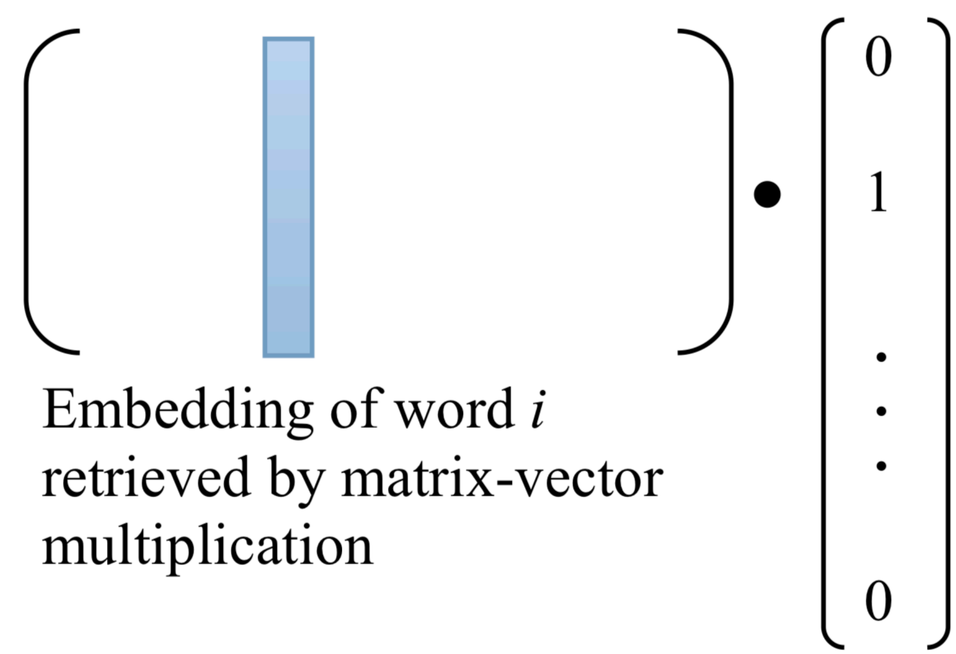
Non-informative

ID and One-Hot

		ID representation	One-hot representation
		1 3 2 0	$\begin{pmatrix}0\\1\\0\\0\end{pmatrix} \begin{pmatrix}0\\0\\0\\1\end{pmatrix} \begin{pmatrix}0\\0\\1\\0\end{pmatrix} \begin{pmatrix}1\\0\\0\\0\end{pmatrix}$
Dimension		One-dimensional	$\lvert \mathscr{V} \rvert$-dimensional
Metric	**Euclidean**	Artefact	Non-informative
	Discrete	Non-informative	Non-informative
	Learnable	Difficult	Possible but may not generalize Need to explore more

Something in Between

- Map a word to a low-dimensional space
  - Not as low as one-dimensional ID representation
  - Not as high as $|\mathscr{V}|$-dimensional one-hot representation
- **Attempt#3:** Word vector representation (a.k.a., word embeddings)
  - Mapping a word to a vector
  - Equivalent to linear tranformation of one-hot vector

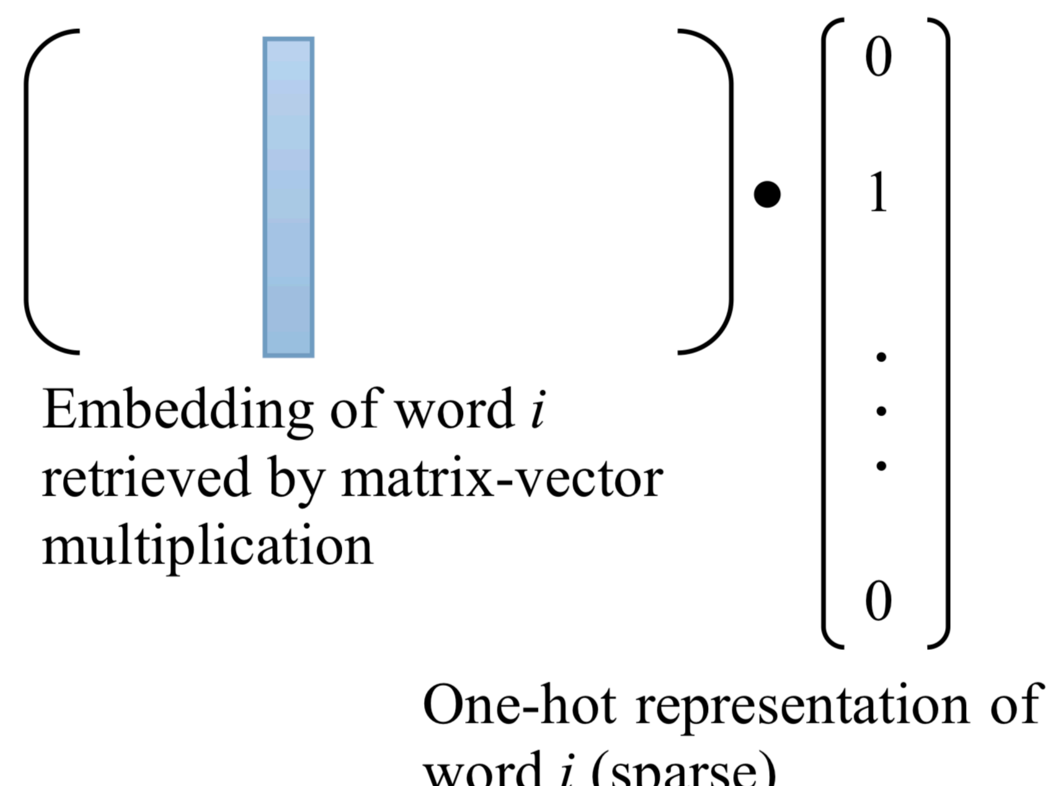


Embedding of word i retrieved by matrix-vector multiplication

One-hot representation of word i (sparse)

Obtaining the Embedding Matrix

- **Attemp#1:** Treat as neural weights as usual
  - Random initialization & gradient descent
- Properties of the embedding matrix
  - Huge, $|\mathscr{V}| \times d_{NN}$ parameters (cf. weight for layerwise MLP)
  - Sparsely updated
- Nature of language
  - Power law distribution
- Good if corpus is large



Embedding Learning

- **Attempt #2:**
  - Manually specifying the distance metric/inner-product, etc.
  - Humans are not rational
- **Attempt #3:**
  - Pre-training on a massive corpus with a different (pre-training) objective
  - Then, we can fine-tune those pre-trained embeddings in almost any specific task.

UNIVERSITY OF ALBERTA

Pretraining Criterion

- Language Modeling
  - Given a corpus $\mathbf{x} = x_1 x_2 \cdots x_t$
  - Goal: Maximize $p(\mathbf{x})$
- Is it meaningful to view language sentences as a random variable?
  - Frequentist: Sentences are repetitions of i.i.d. experiments
  - Bayesian: Everything unknown is a random variable

Factorization

- $p(\mathbf{x}) = p(x_1, \cdots, x_t)$ cannot be parametrized
- Factorizing a giant probability

$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1, \cdots, x_t) \\
&= p(x_1) p(x_2 \mid x_1) \cdots p(x_t \mid x_1, \cdots, x_{t-1})
\end{aligned}
$$

- Still unable to parametrize, especially $p(x_n \mid x_1, \cdots, x_{n-1})$
- **Questions:**

- Can we decompose any probabilistic distribution defined on $\mathbf{x}$ into this form? Yes.
- Is it necessary to decompose the distribution a probabilistic distribution in this form? No.

Markov Assumptions

$$p(\mathbf{x}) = p(x_1, \cdots, x_t)$$
$$= p(x_1)p(x_2 \,|\, x_1)\cdots p(x_t \,|\, x_1, \cdots, x_{t-1})$$

- Independency
  - Given the current "state," independent with previous ones
  - State at step t: $(x_{t-n+1}, x_{t-n+2}, \cdots, x_{t-1})$
  - $x_t \perp x_{\le t-n} \,|\, x_{t-n+1}, x_{t-n+2}, \cdots, x_{t-1}$
- Stationary property
  - $p(x_t \,|\, x_{t-1}, \cdots, x_{t-n+1}) = p(x_s \,|\, x_{s-n+1}, \cdots, x_{s-1})$ for all t, s

Parametrizing $p(\mathbf{w})$

$$\begin{aligned} p(\mathbf{x}) &= p(x_1, \cdots, x_t) \\ &= p(x_1)p(x_2 \,|\, x_1) \cdots p(x_t \,|\, x_1, \cdots, x_{n-1}) \\ &\approx p(x_1)p(x_2 \,|\, x_1) \cdots p(x_n \,|\, x_1, \cdots, x_{t-n+1}) \end{aligned}$$

Direct parametrization:

Each multinomial distribution is directly parametrized

$p(w_n \,|\, w_1, \cdots, w_{n-1})$ (notation abuse)

UNIVERSITY OF ALBERTA

N-gram Model

$$
\begin{aligned}
p(\mathbf{x}) &= p(x_1, \cdots, x_n) \\
&= p(x_1)p(x_2 | x_1) \cdots p(x_n | x_1, \cdots, x_{n-1}) \\
&\approx p(x_1)p(x_2 | x_1) \cdots p(x_n | x_1, \cdots, x_{t-n+1})
\end{aligned}
$$

$$
\hat{p}(w_n | w_1, \cdots, w_{n-1}) = \frac{\# w_1 \cdots w_n}{\# w_1 \cdots w_{n-1}}
$$

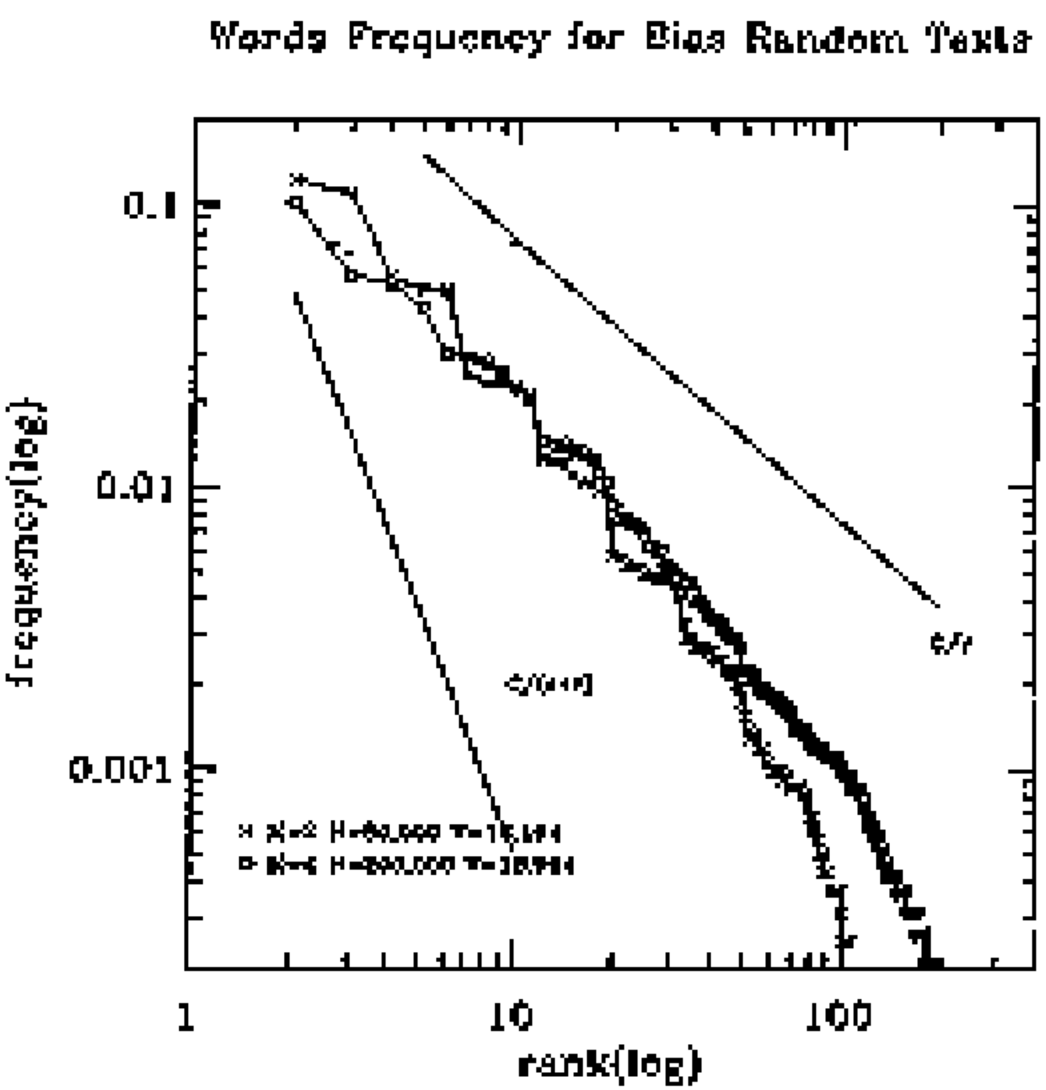
Questions:

- How many multinomial distributions?
- How many parameters in total?

UNIVERSITY OF ALBERTA

Problems of n-gram models

- #para $\propto$ exp(n)
- Power-law distribution
  - Severe data sparsity even if n is small



- Normal distribution

$$p(x) \propto \exp(-\tau x^2)$$

- Power-law distribution

$$p(x) \propto x^{-k}$$

Smoothing Techniques

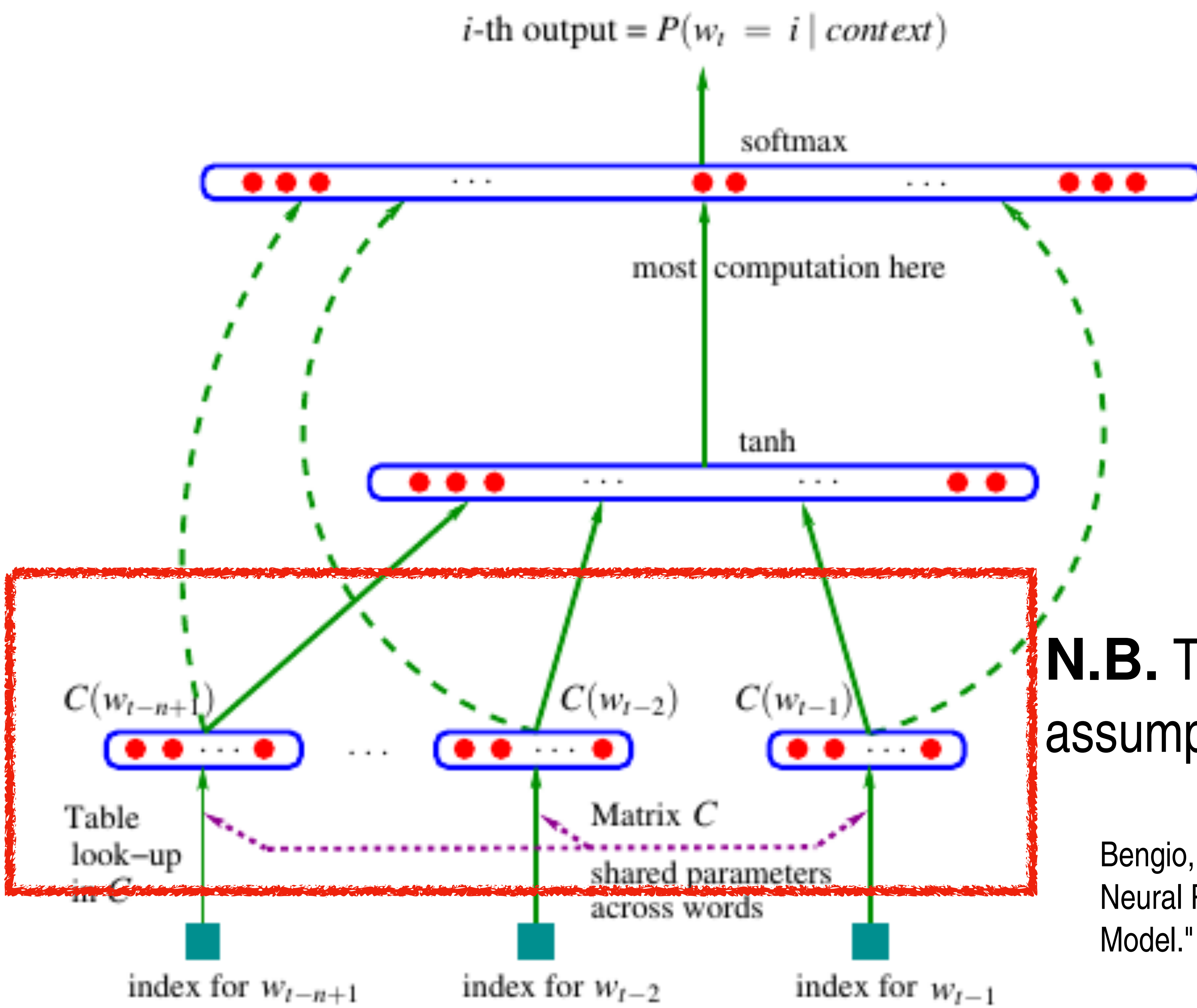
- Add-one smoothing
- Interpolation smoothing
- Backoff smoothing

Useful link: https://nlp.stanford.edu/~wcmac/papers/20050421-smoothing-tutorial.pdf

Parametrizing LM by NN

- Is it possible to parametrize LM by NN?
- Yes
  - $p(w_n \mid w_1, \cdots, w_{n-1})$ is a classification problem
  - NNs are good at (esp. non-linear) classification

Feed-Forward Language Model



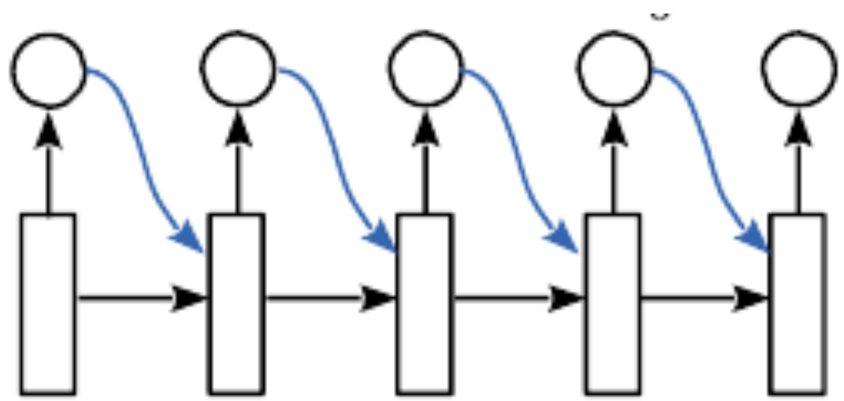
N.B. The Markov assumption also holds.

Bengio, Yoshua, et al. "A Neural Probabilistic Language Model." JMLR. 2003.

By product: Embeddings are pre-trained in a meaningful way

Recurrent Neural Language Model

- RNN keeps one or a few hidden states
- The hidden states change at each time step according to the input



$$\boldsymbol{h}_t = \mathrm{RNN}(\boldsymbol{x}_t, \boldsymbol{h}_{t-1})$$
$$= f\left(W_{\mathrm{in}}\boldsymbol{x}_t + W_{\mathrm{hid}}\boldsymbol{h}_{t-1}\right)$$
$$p(w_t|\boldsymbol{w}_0^{t-1}) \approx \mathrm{softmax}\left(W_{\mathrm{out}}\boldsymbol{h}_t\right)$$

- RNN directly parametrizes $p(\boldsymbol{w}) = \prod_{t=1}^{m} p(w_t|\boldsymbol{w}_1^{t-1})$
  rather than $p(\boldsymbol{w}) \approx \prod_{t=1}^{m} p\left(w_t|\boldsymbol{w}_{t-n+1}^{t-1}\right)$

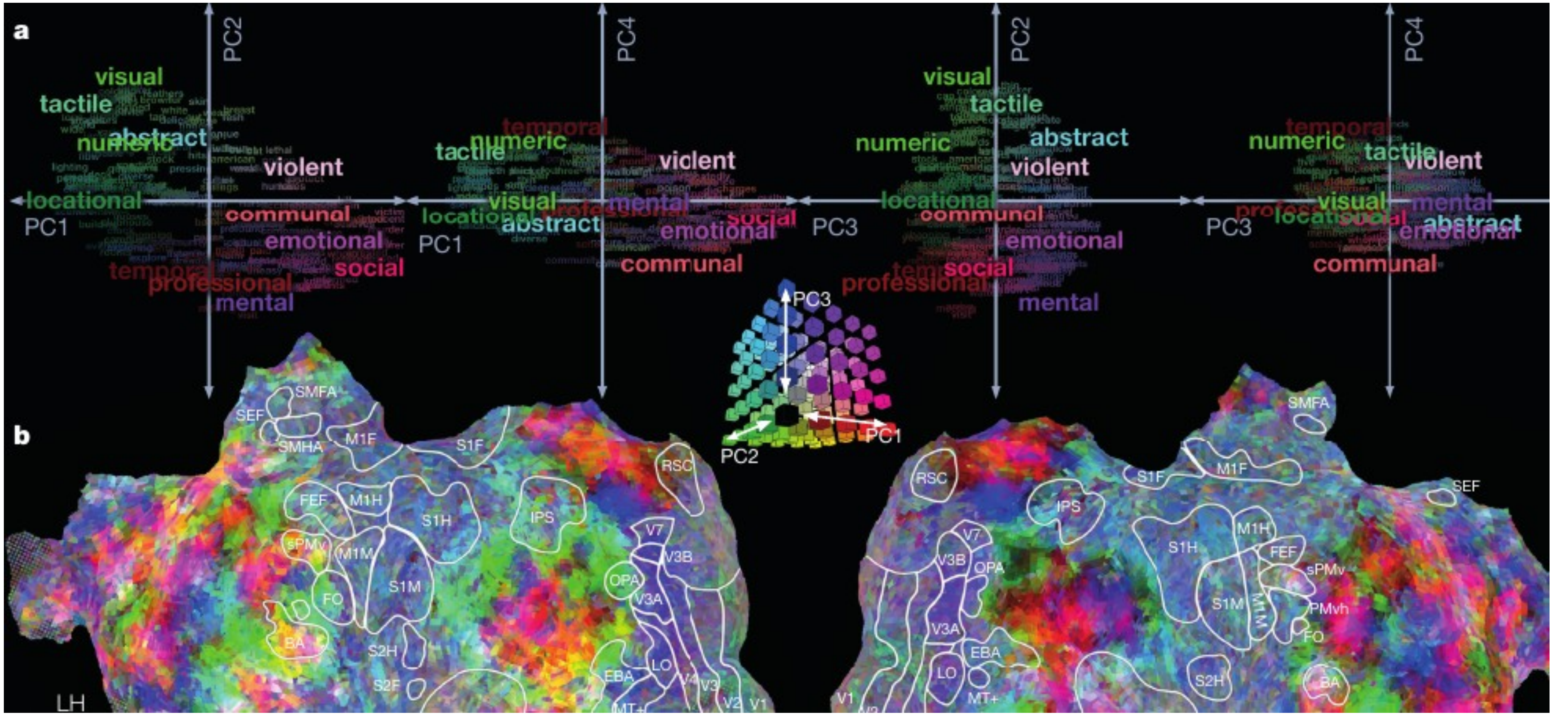
Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In INTERSPEECH, 2010.

How can we use word embeddings?

- Embeddings demonstrate the internal structures of words
  - Relation represented by vector offset

    "man" – "woman" = "king" – "queen" [Mikolov+NAACL13]
  - Word similarity
- Embeddings can serve as the initialization of almost every supervised task
  - A way of pretraining
  - **N.B.:** may not be useful when the training set is large enough

Word Embeddings in our Brain



Huth, Alexander G., et al. "Natural speech reveals the semantic maps that tile human cerebral cortex." Nature 532.7600 (2016): 453-458.

“Somatotopic Embeddings” in our Brain

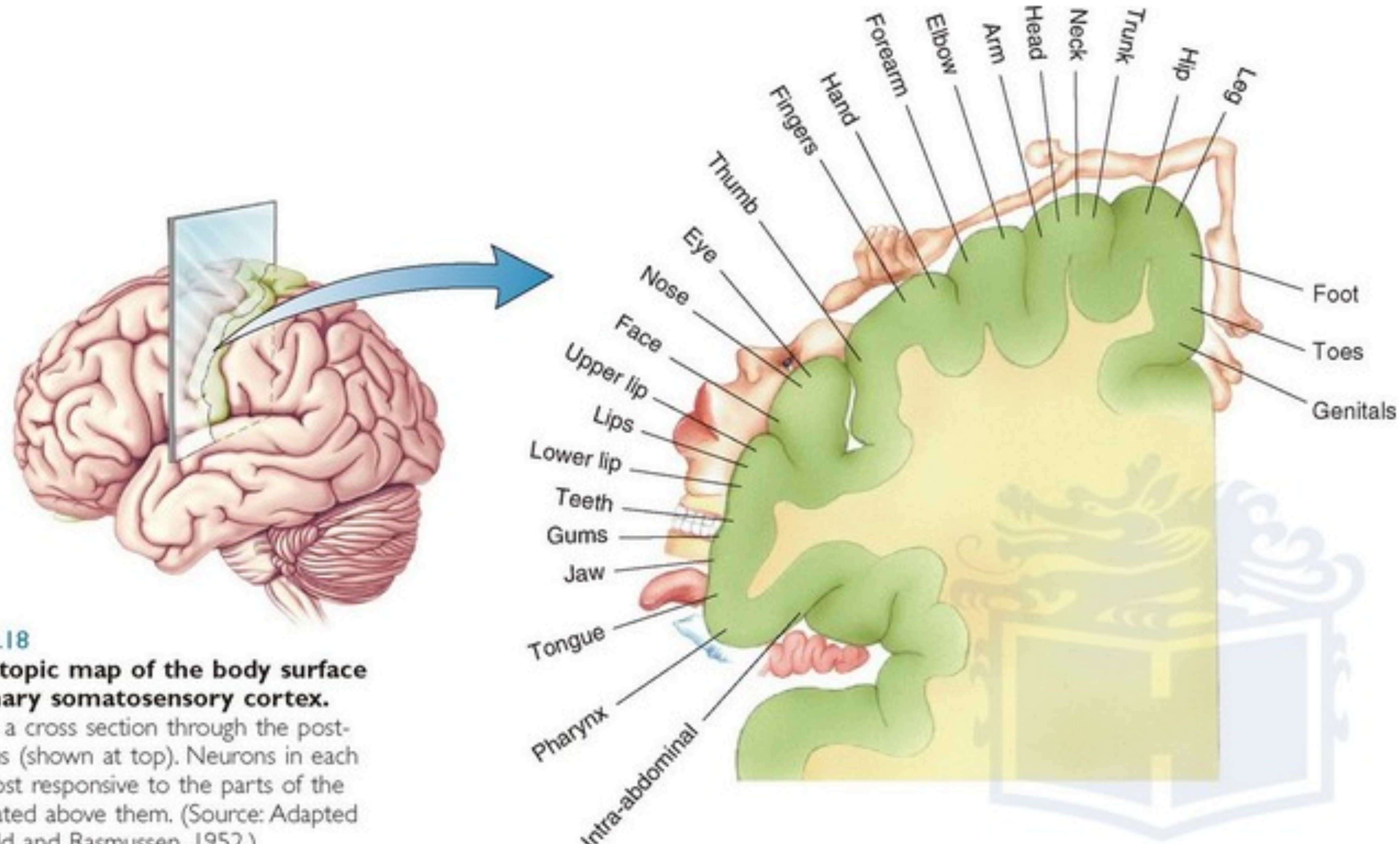


FIGURE 12.18
A somatotopic map of the body surface onto primary somatosensory cortex. This map is a cross section through the post-central gyrus (shown at top). Neurons in each area are most responsive to the parts of the body illustrated above them. (Source: Adapted from Penfield and Rasmussen, 1952.)

Bear MF, Connors BW, Michael A. Paradiso. Neuroscience: Exploring the Brain. 2007

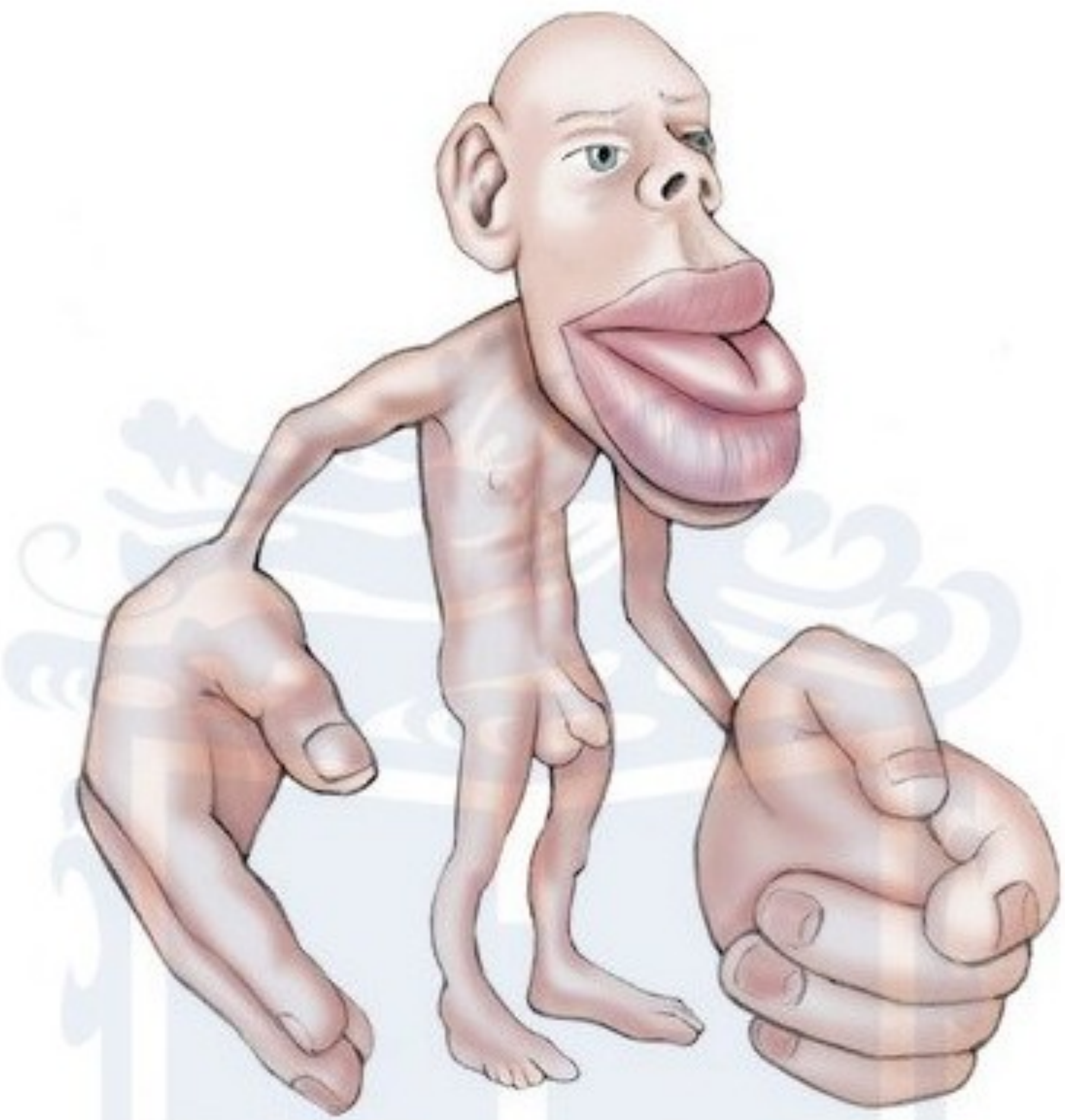


FIGURE 12.19
The homunculus.

Bear MF, Connors BW, Michael A. Paradiso. Neuroscience: Exploring the Brain. 2007

Complexity Concerns

- Time complexity
  - Hierarchical softmax [1]
  - Negative sampling: Hinge loss [2], Noisy contrastive estimation [3]
- Memory complexity
  - Compressing LM [4]
- Model complexity
  - Shallow neural networks are still too “deep.”
  - CBOW, SkipGram [3]

[1] Mnih A, Hinton GE. A scalable hierarchical distributed language model. NIPS, 2009.

[2] Collobert R, Weston J, Bottou L, Karlen M, Kavukcuoglu K, Kuksa P. Natural language processing (almost) from scratch. JMLR, 2011.

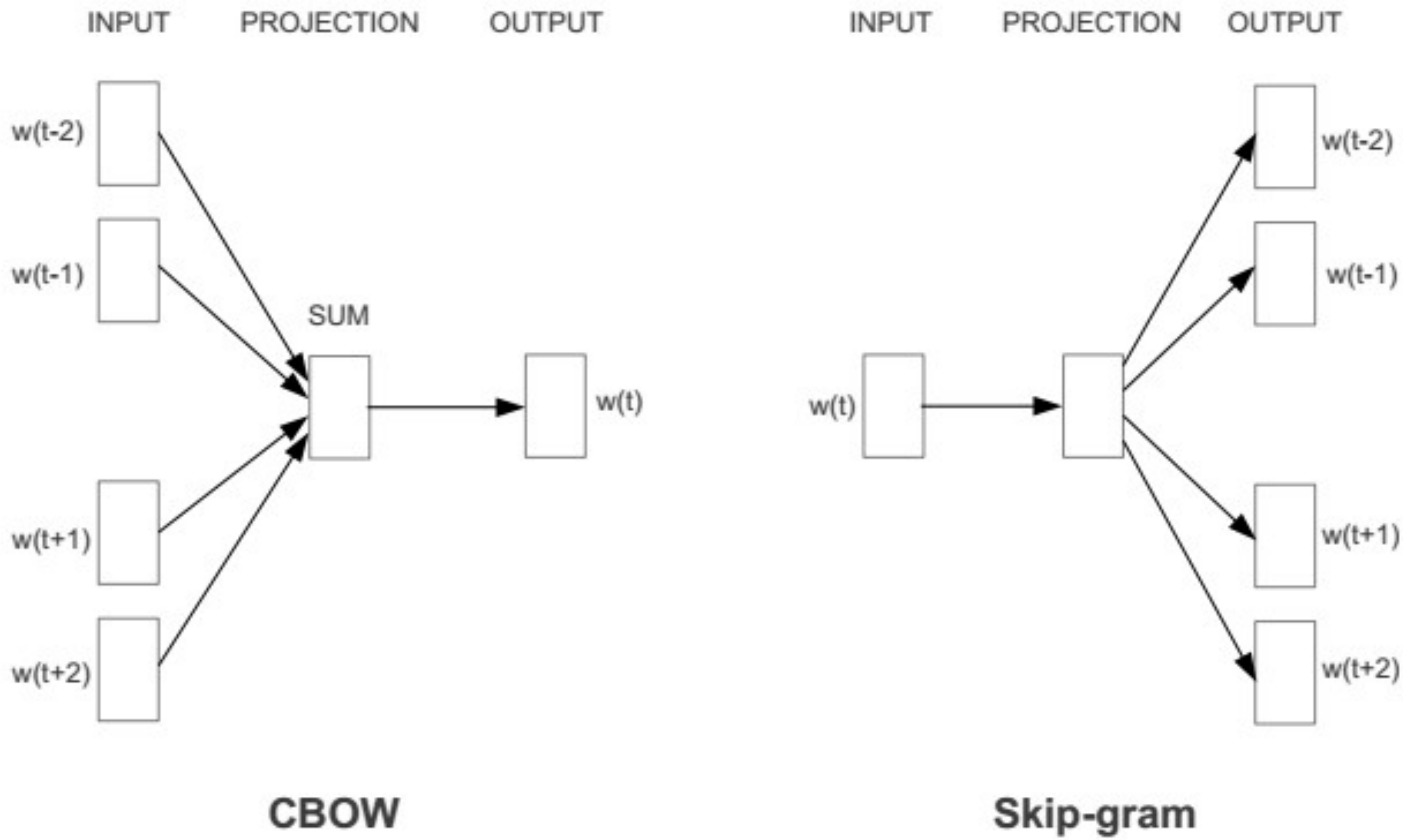
[3] Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013

[4] Yunchuan Chen, Lili Mou, Yan Xu, Ge Li, Zhi Jin. "Compressing neural language models by sparse word representations." In ACL, 2016.

Deep neural networks:
To be, or not to be? That is the question.

UNIVERSITY OF ALBERTA

CBOW, SkipGram (word2vec)



Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013

Hierarchical Softmax and Negative Contrastive Estimation

- HS

$$p(w|w_I) = \prod_{j=1}^{L(w)-1} \sigma\left([\![n(w, j+1) = \mathrm{ch}(n(w,j))]\!] \cdot {v'_{n(w,j)}}^\top v_{w_I} \right)$$

- NCE

$$\log \sigma({v'_{w_O}}^\top v_{w_I}) + \sum_{i=1}^{k} \mathbb{E}_{w_i \sim P_n(w)} \left[\log \sigma(-{v'_{w_i}}^\top v_{w_I}) \right]$$

Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013

Tricks in Training Word Embeddings

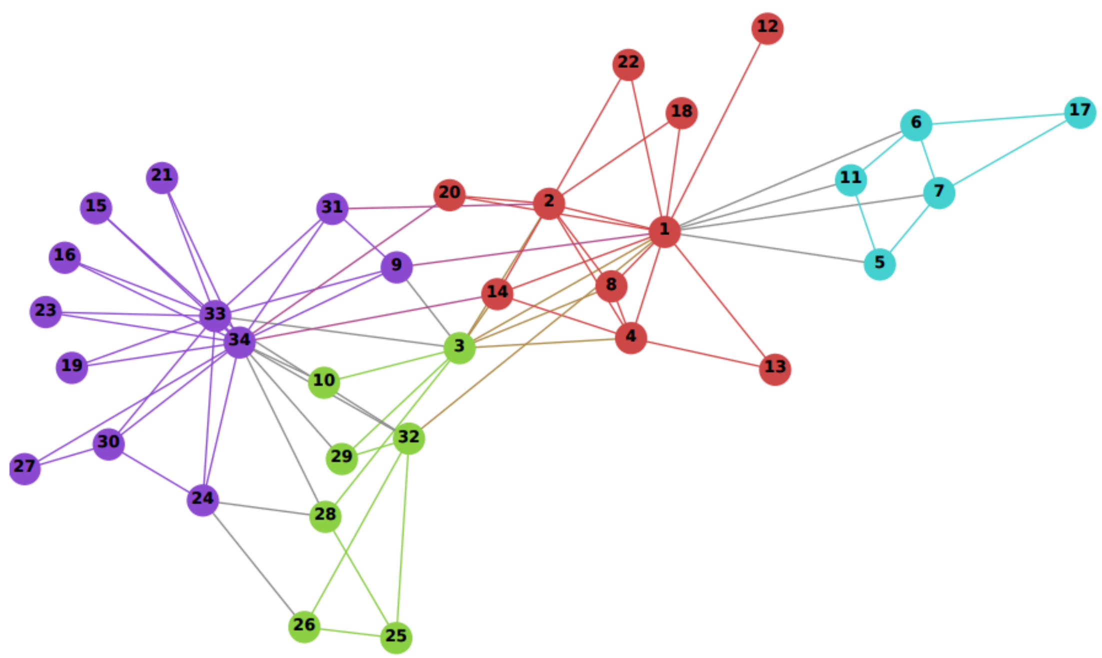
- The # of negative samples?
  - The more, the better.
- The distribution from which negative samples are generated? Should negative samples be close to positive samples?
  - The closer, the better.
- Full softmax vs. NCE vs. HS vs. hinge loss?

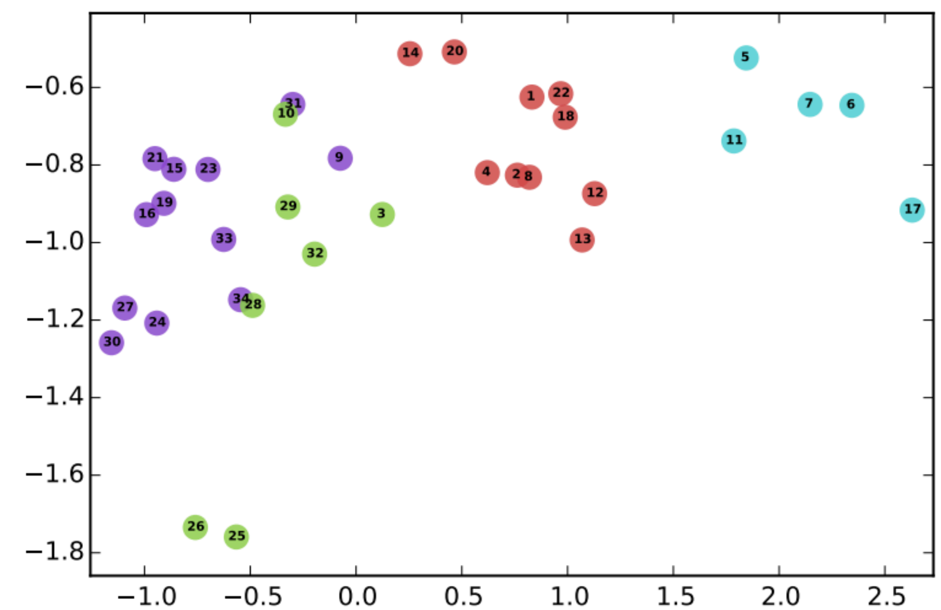
Recent Advances in Pretraining

- Pretraining the embedding mapping for words is not enough
  - E: Vocabulary $\to \mathbb{R}^n$
- Context info?
  - Why not pre-train follow-up layers as well?
  - E.g., ELMo, BERT
  - Represent a word in a context, with LM-like pretraining
  - Factorization of $p(\mathbf{w}) = p(w_1)p(w_2 \mid w_1)\cdots p(w_n \mid w_1 \cdots w_{n-1})$ is unnecessary

Learning Embeddings of Other Stuff

- Node embeddings of a network





[DeepWalk, KDD 2014]

- General criteria of embedding learning
  - Atomic token represented by an embedding
  - Training embeddings by predicting “context”

Mindmap

Representing Words

Index

One-hot

[E] *

Real-valued embedding

Embeddings in general
- Discrete token -> vector
- Learned by predicting context

One pretraining method

Language modeling
- Max Pr(corpus)

N-gram
- Markov assumption
- MLE = counting %
- Sparsity
  - Para ∝ exp(n)
  - Power law dist.

NN-LM
- Predict the next word
- Embeddings pretrained
- Recent advance: Pretrain LM

Suggested Reading

- **Neural LM:** Bengio, Yoshua, et al. "A Neural Probabilistic Language Model." JMLR. 2003.
- **word2vec:** Mikolov T, Chen K, Corrado G, Dean J. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781. 2013
- **ELMo:** Peters, M.E., Neumann, M., Iyyer, M., Gardner, M., Clark, C., Lee, K. and Zettlemoyer, L., 2018. Deep contextualized word representations. In *NAACL*, 2018.
- **BERT:** Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
- **DeepWalk:** Perozzi, B., Al-Rfou, R. and Skiena, S. DeepWalk: Online learning of social representations. In KDD, 2014.

More References

- Graves, A., Abdel-rahman M., and Geoffrey H. Speech recognition with deep recurrent neural networks. *In ICASSP*, 2013.
- Mikolov, T., Sutskever, I., Chen, K., Corrado, G. S., & Dean, J. (2013). Distributed representations of words and phrases and their compositionality. In *NIPS*, 2013.
- Li, W. Random texts exhibit Zipf's-law-like word frequency distribution. *IEEE Transactions on Information Theory*, 38(6), 1842-1845, 1992.
- Bengio, Yoshua, et al. A Neural Probabilistic Language Model. *JMLR*. 2003.
- Mikolov T, Karafiát M, Burget L, Cernocký J, Khudanpur S. Recurrent neural network based language model. In *INTERSPEECH*, 2010.
- Devlin, J., Chang, M.W., Lee, K. and Toutanova, K., 2018. Bert: Pre-training of deep bidirectional transformers for language understanding. In *NAACL*, 2019.
- Mikolov, T., Chen, K., Corrado, G. and Dean, J., 2013. Efficient estimation of word representations in vector space. arXiv preprint arXiv:1301.3781.
- Mikolov, T., Yih, W.T. and Zweig, G., June. Linguistic regularities in continuous space word representations. In *NAACL*, 2013.