

Neural Language Models

Recap: Language modeling

Have a good ...

day

weather

terrible

time

4-gram language model:

$$p(\text{day} \mid \text{have a good}) =$$

c(have a good day)

Curse of dimensionality

Imagine you have seen the following many times:

- **Have a good day.**

However, you have not seen the following:

- **Have a great day.**

What happens than (even with smoothing)?

good

42

|V|

great

127



How to generalize better

- Learn **distributed representations** for words
- Express probabilities of sequences in terms of these distributed representations and learn parameters

good



great

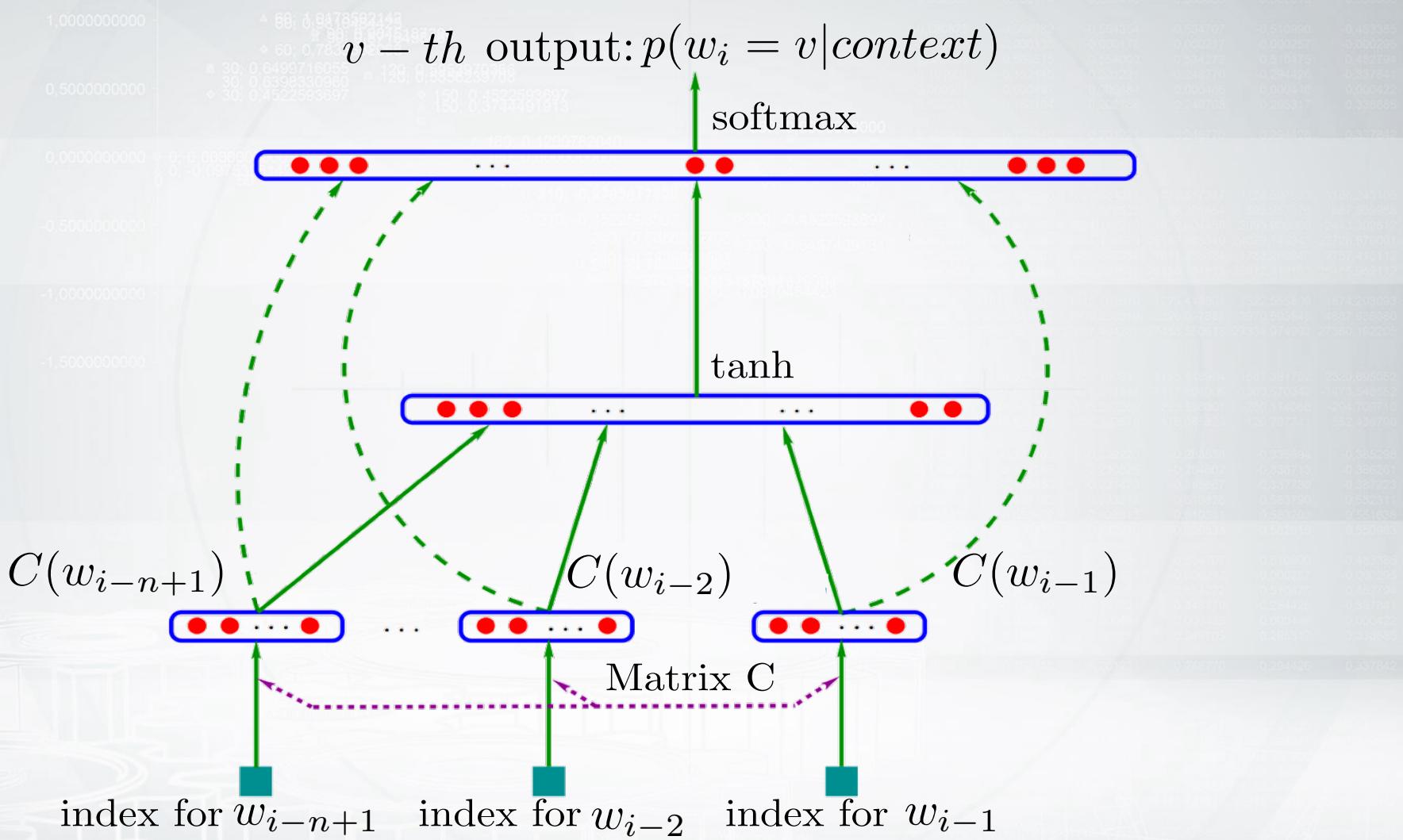


dog



$C^{|V| \times m}$ – matrix of distributed word representations.

Probabilistic Neural Language Model



Yoshua Bengio, Réjean Ducharme, Pascal Vincent, Christian Jauvin, A Neural Probabilistic Language Model, JMLR, 2003

Probabilistic Neural Language Model

$$p(w_i|w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

Probabilistic Neural Language Model

$$p(w_i|w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

Softmax over components of y

Probabilistic Neural Language Model

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

Softmax over components of y

Feed-forward NN with tons of parameters

Probabilistic Neural Language Model

$$p(w_i|w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(y_{w_i})}{\sum_{w \in V} \exp(y_w)}$$

$$y = b + Wx + U \tanh(d + Hx)$$

$$x = [C(w_{i-n+1}), \dots, C(w_{i-1})]^T$$

Softmax over components of y

Feed-forward NN with tons of parameters

Distributed representation of context words

It's over-complicated...

$$y = b + Wx + U \tanh(d + Hx)$$

$$m * (n - 1)$$

$$\begin{matrix} y \\ |V| \end{matrix} = \begin{matrix} b \\ |V| \end{matrix} + \begin{matrix} W \\ |V| \end{matrix} \times \begin{matrix} x \\ |V| \end{matrix} \times \begin{matrix} C(w_{i-n+1})^T \\ \vdots \\ C(w_{i-1})^T \end{matrix} + \dots$$

Log-Bilinear Language Model

- Has much less parameters and non-linear activations
- Measures similarity between the word and the context:

$$r_{w_i} = C(w_i)^T$$

Representation of word:

$$p(w_i | w_{i-n+1}, \dots, w_{i-1}) = \frac{\exp(\hat{r}^T r_{w_i} + b_{w_i})}{\sum_{w \in V} \exp(\hat{r}^T r_w + b_w)}$$

Representation of context:

$$\hat{r} = \sum_{k=1}^{n-1} W_k C(w_{i-k})^T$$

Andriy Mnih and Geoffrey Hinton. 2007. Three new graphical models for statistical language modelling. In *Proceedings of the 24th international conference on Machine learning (ICML '07)*