Neural Language Models

Recap: Language modeling

Have a good …

day

weather

terrible

time

4-gram language model:

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

Curse of dimensionality

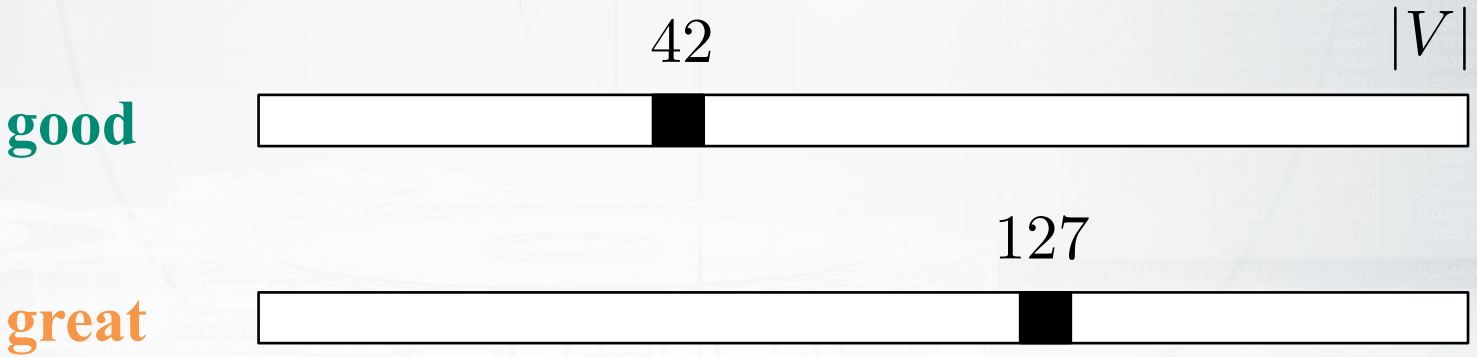
Imagine you have seen the following many times:

- **Have a good day.**

However, you have not seen the following:

- **Have a great day.**

What happens than (even with smoothing)?

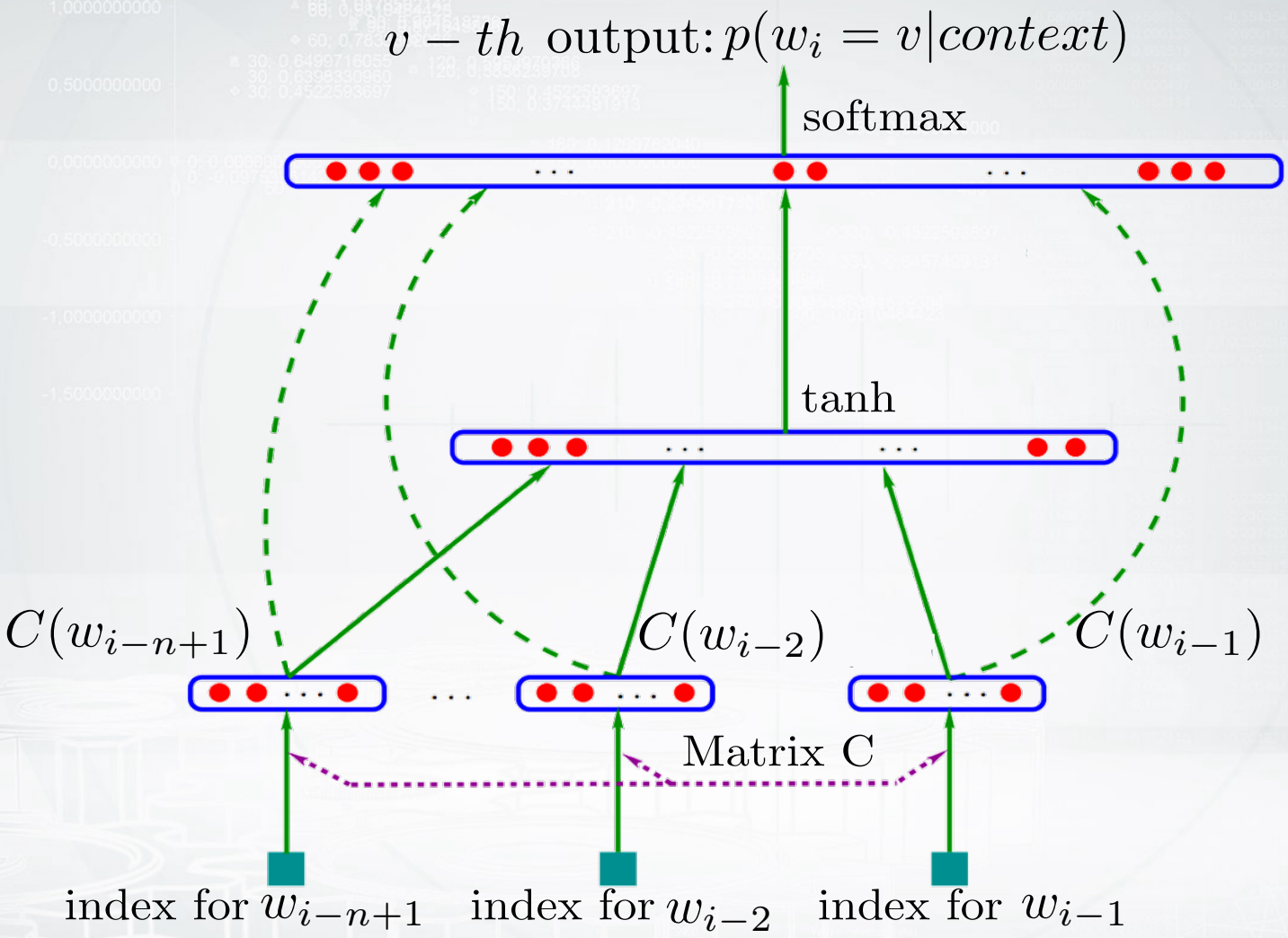


How to generalize better

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

$C^{|V|\times m}$ – 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

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

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

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

Probabilistic Neural Language Model

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

Softmax* over components of *y

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

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

Probabilistic Neural Language Model

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

Softmax* over components of *y

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

***Feed-forward NN* with tons of parameters**

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

Probabilistic Neural Language Model

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

Softmax* over components of *y

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

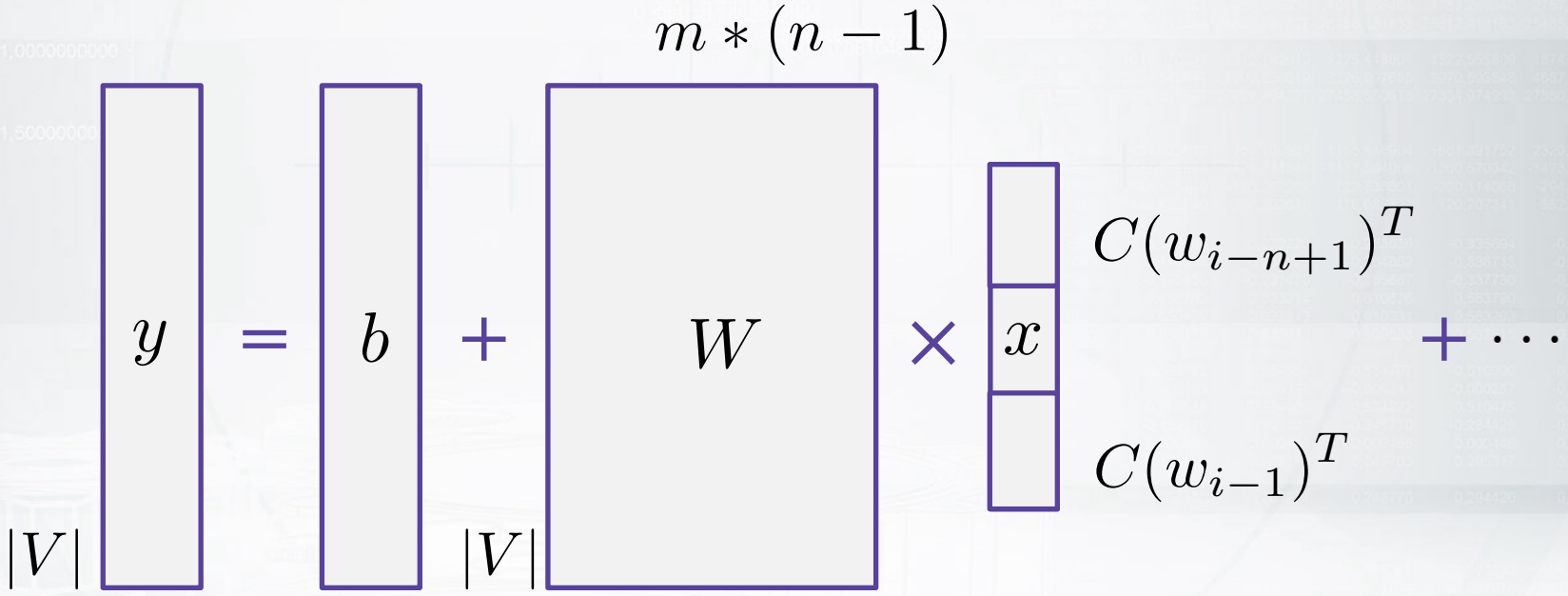
***Feed-forward NN* with tons of parameters**

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

***Distributed representation* of context words**

It's over-complicated…

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



Log-Bilinear Language Model

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

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

Representation of word:

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

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)