Language Models

Dan Klein
UC Berkeley

Neural LMs: Preview

A GPT2 Continuation

- **The computer I had put into the machine room on the fifth floor just just** outside our landing was taken by a lot of people. It was going to be recovered from outside the machine room as soon as we could, but after the machine room was shut down, we had to open all of the windows and doors to save ourselves, and the computer would have been lost. Luckily we are prepared for this. We had five steel doors on each side of the landing to hold any rain and wind that might have hit, so we weren't affected by the storm.

  The situation was really grim for days. A few days before, the Apache at Qush Tefah had been shot. This was the most damaged machine in the Armies Corps. The command was only meant to repair Qush

A GPT2 Continuation

- **I shall go seek the traitor Gloucester**, when he hath declared his vengeance. Cunegund Shall I back her to kill my cousin in my stead? Witbane Far first shall the brother take his Queen's soul and that in the flames. Clotilda For to slay the King of Scotland with, the blood of my cousin, lie true; And she shall have the right to my feather." Sefton leapt into the rushes, and stole along in the sunlight: the small figure flashed like lightning in the west. In the likeness of a snare he had laid to catch the rushes and made of them a snares, a road to flee from his pursuers; but he now came to an oak where the branches were wreathed in an oak-

Neural LMs: Three Key Ideas

- Word embeddings
  - Different words are not entirely unrelated events
  - Words can be more and less similar, in complex ways
- Partially factored representations
  - Multiple semi-independent processes happen in parallel in language
  - It's too expensive to track language in an unfactored way, and too inaccurate to assume everything of interest is independent
- Long distance dependencies
  - Information can be relevant without being local
  - Different notions of locality are important at different times

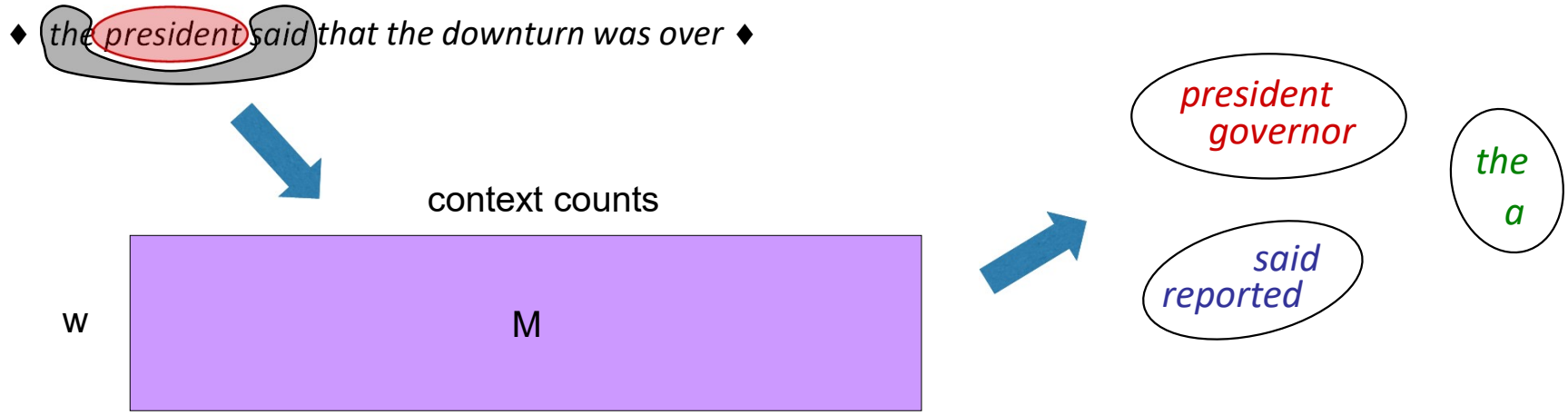
Words: Clusterings and Embeddings

Stuffing Words into Vector Spaces?

Cartoon: Greg Durrett

Distributional Similarity

- Key idea in clustering and embedding methods: characterize a word by the words it occurs with (cf Harris' distributional hypothesis, 1954)
  - "You can tell a word by the company it keeps." [Firth, 1957]
  - Harris / Chomsky divide in linguistic methodology



Clusterings

Clusterings

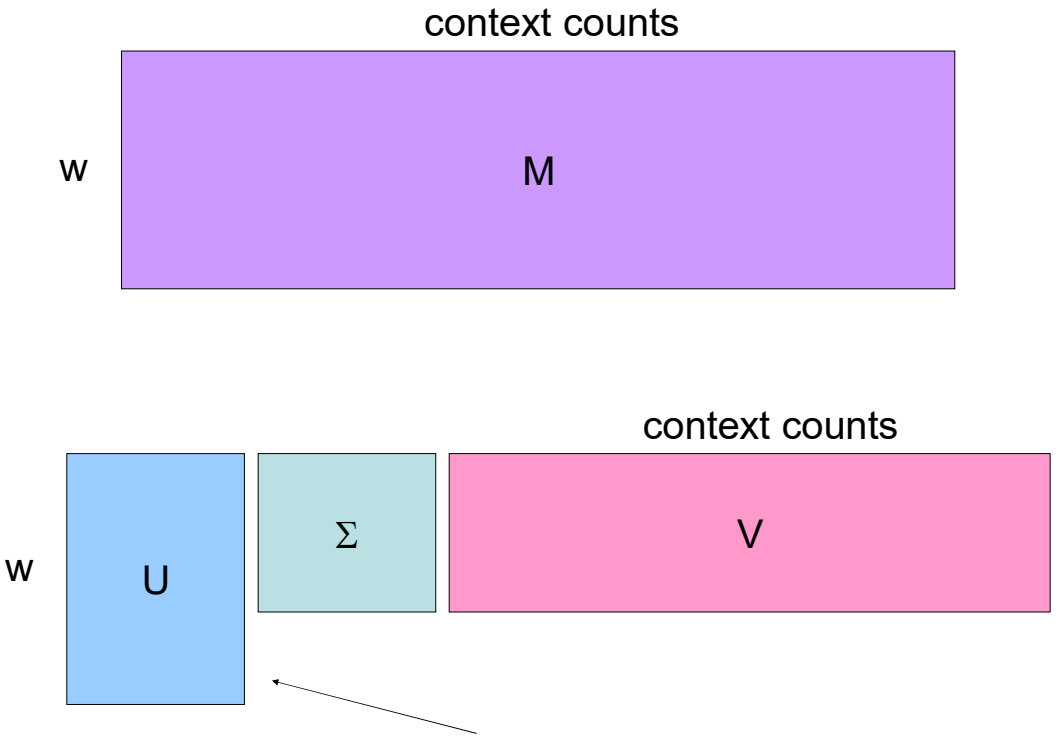
- Automatic (Finch and Chater 92, Shuetze 93, many others)

word	nearest neighbors
accompanied	submitted banned financed developed authorized headed canceled awarded barred
almost	virtually merely formally fully quite officially just nearly only less
causing	reflecting forcing providing creating producing becoming carrying particularly
classes	elections courses payments losses computers performances violations levels pictures
directors	professionals investigations materials competitors agreements papers transactions
goal	mood roof eye image tool song pool scene gap voice
japanese	chinese iraqi american western arab foreign european federal soviet indian
represent	reveal attend deliver reflect choose contain impose manage establish retain
think	believe wish know realize wonder assume feel say mean bet
york	angeles francisco sox rouge kong diego zone vegas inning layer
on	through in at over into with from for by across
must	might would could cannot will should can may does helps
they	we you i he she nobody who it everybody there

- Manual (e.g. thesauri, WordNet)

Vector Space Methods

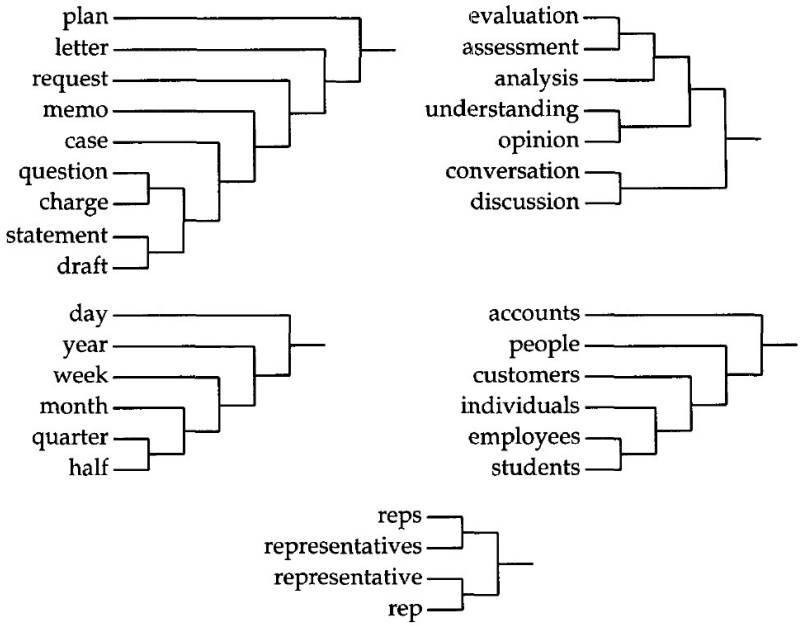
- Treat words as points in R^n (eg Shuetze, 93)
  - Form matrix of co-occurrence counts
  - SVD or similar to reduce rank (cf LSA)
  - Cluster projections
  - People worried about things like: log of counts, U vs UΣ
- Today we'd call this an embedding method (it's basically GLoVe), but we didn't want embeddings in 1993



Models: Brown Clustering

- Classic model-based clustering (Brown et al, 92)
  - Each word starts in its own cluster
  - Each cluster has co-occurrence stats
  - Greedily merge clusters based on a mutual information criterion
  - Equivalent to optimizing a class-based bigram LM.

$$P(w_i|w_{i-1}) = P(c_i|c_{i-1})P(w_i|c_i)$$



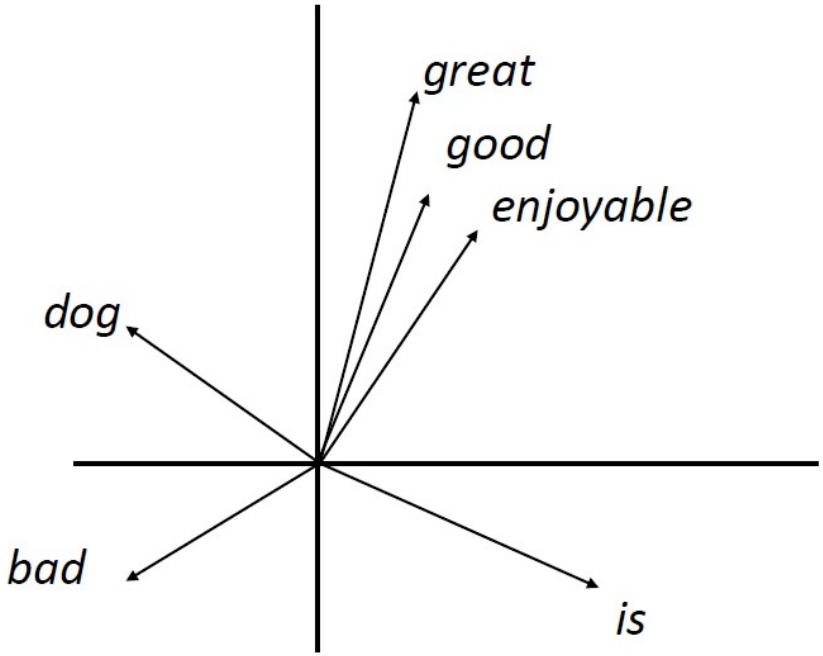
  - Produces a dendrogram (hierarchy) of clusters

Embeddings

Most slides from Greg Durrett

Embeddings

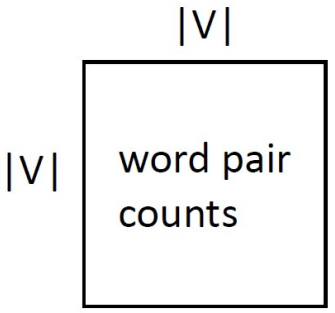
- Embeddings map discrete words (eg |V| = 50k) to continuous vectors (eg d = 100)
- Why do we care about embeddings?
  - Neural methods want them
  - Nuanced similarity possible; generalize across words
- We hope embeddings will have structure that exposes word correlations (and thereby meanings)



Embedding Models

- Idea: compute a representation of each word from co-occurring words

the dog bit the man



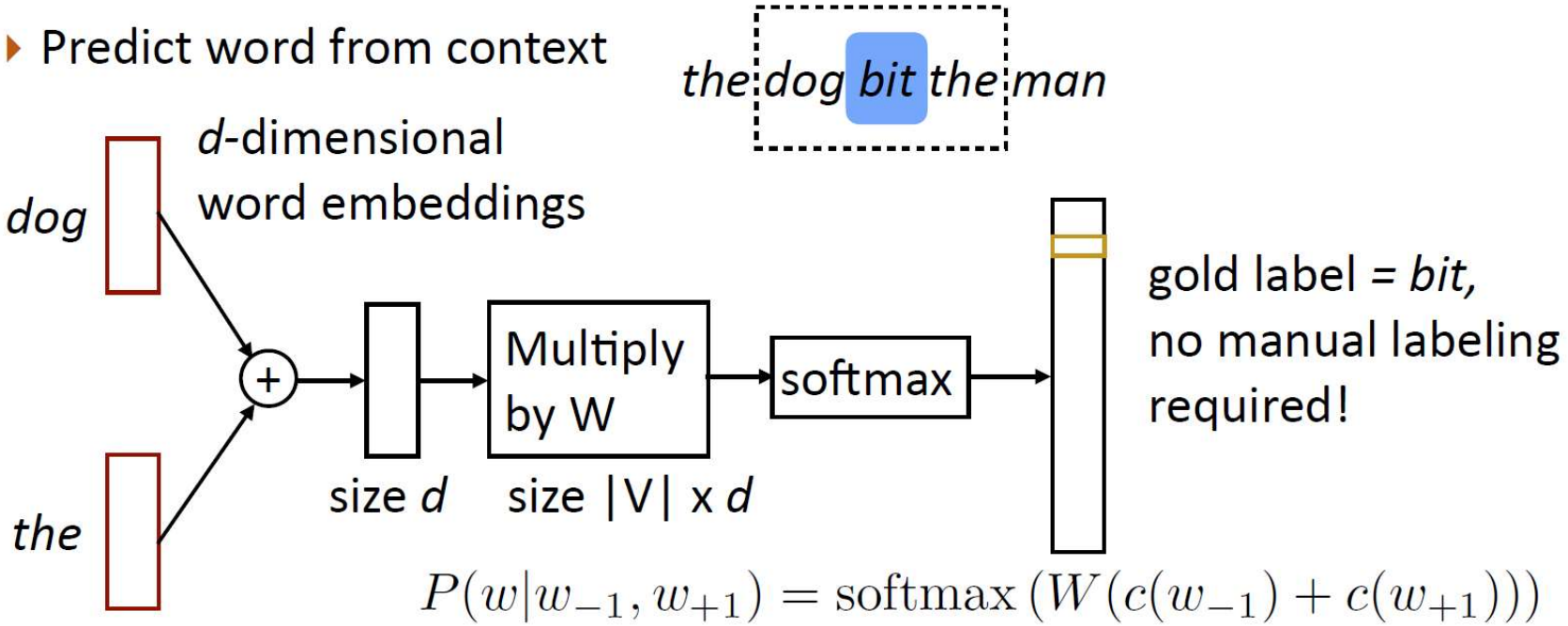
Token-Level

Type-Level

- We'll build up several ideas that can be mixed-and-matched and which frequently get used in other contexts

word2vec: Continuous Bag-of-Words

- Predict word from context



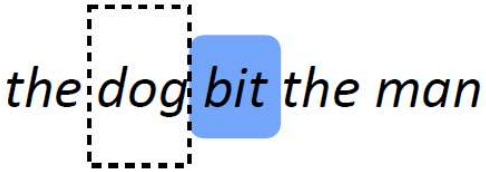
$$P(w|w_{-1}, w_{+1}) = \text{softmax}\left(W(c(w_{-1}) + c(w_{+1}))\right)$$

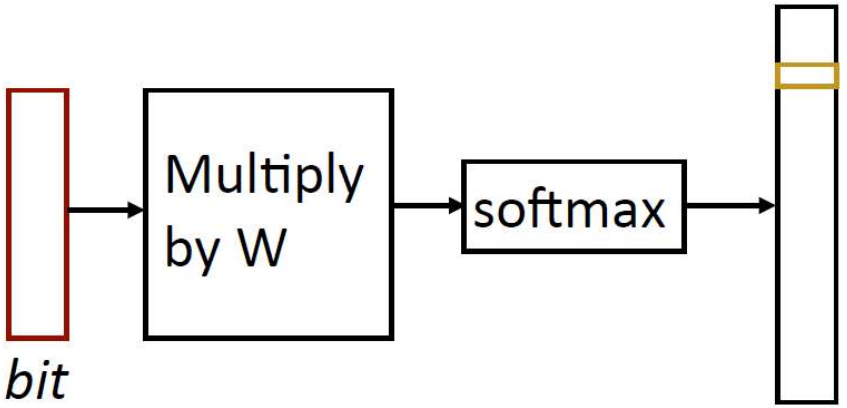
- Parameters: *d* x |V| (one *d*-length context vector per voc word),
  |V| x *d* output parameters (W)

Mikolov et al. (2013)

word2vec: Skip-Grams

- Predict one word of context from word





$$P(w'|w) = \text{softmax}(We(w))$$

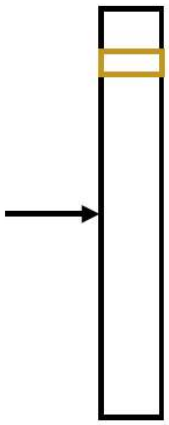
- Another training example: *bit* -> *the*
- Parameters: *d* x |V| vectors, |V| x *d* output parameters (W) (also usable as vectors!)

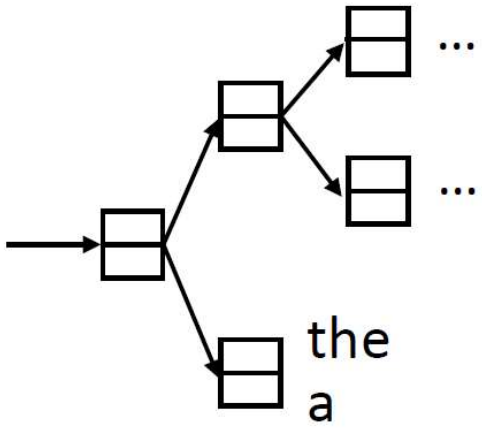
Mikolov et al. (2013)

word2vec: Hierarchical Softmax

$$P(w|w_{-1}, w_{+1}) = \text{softmax}\left(W(c(w_{-1}) + c(w_{+1}))\right) \qquad P(w'|w) = \text{softmax}(We(w))$$

- Matmul + softmax over |V| is very slow to compute for CBOW and SG





- Huffman encode vocabulary, use binary classifiers to decide which branch to take
- log(|V|) binary decisions

- Standard softmax: [|V| x *d*] x *d*
- Hierarchical softmax: log(|V|) dot products of size *d*, |V| x *d* parameters

Mikolov et al. (2013)

word2vec: Negative Sampling

- Take (word, context) pairs and classify them as "real" or not. Create random negative examples by sampling from unigram distribution

  (*bit, the*) => +1

  (*bit, cat*) => -1

  (*bit, a*) => -1

  (*bit, fish*) => -1

$$P(y=1|w,c) = \frac{e^{w \cdot c}}{e^{w \cdot c} + 1}$$

words in similar contexts select for similar *c* vectors

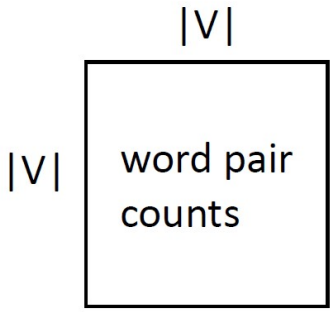
- *d* x |V| vectors, *d* x |V| context vectors (same # of params as before)
- Objective = $\log P(y=1|w,c) + \frac{1}{k}\sum_{i=1}^{n} \log P(y=0|w_i,c)$

sampled

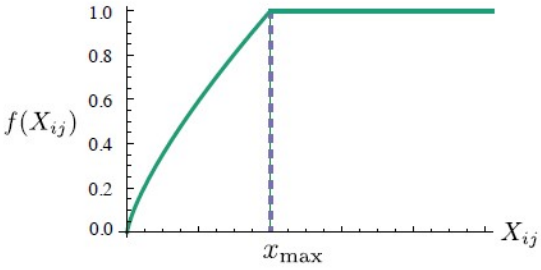
Mikolov et al. (2013)

GloVe

- Idea: Fit co-occurrence matrix directly (weighted least squares)



$$J = \sum_{i,j=1}^{V} f\left(X_{ij}\right)\left(w_i^T \tilde{w}_j + b_i + \tilde{b}_j - \log X_{ij}\right)^2$$



- Type-level computations (so constant in data size)
- Currently the most common word embedding method

Pennington et al, 2014

Bottleneck vs Co-occurrence

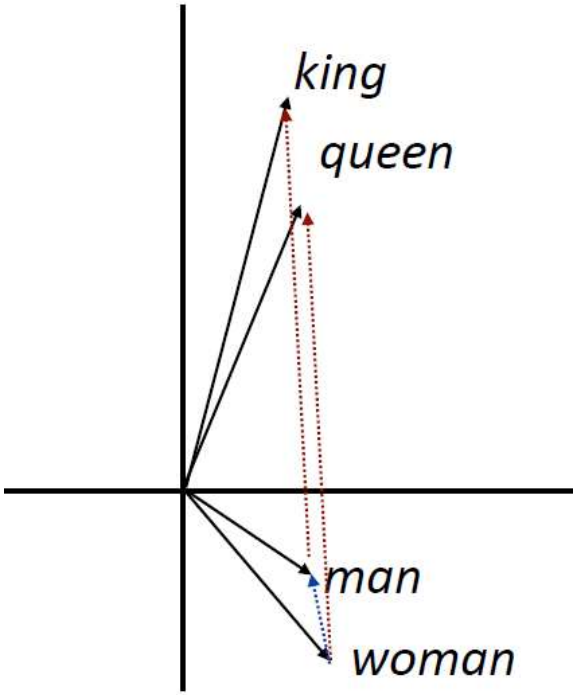
- Two main views of inducing word structure
  - Co-occurrence: model which words occur in similar contexts
  - Bottleneck: model latent structure that mediates between words and their behaviors

- These turn out to be closely related!

Language Models

Structure of Embedding Spaces

- How can you fit 50K words into a 64-dimensional hypercube?
- Orthogonality: Can each axis have a global “meaning” (number, gender, animacy, etc)?
- Global structure: Can embeddings have algebraic structure (eg king – man + woman = queen)?



Bias in Embeddings

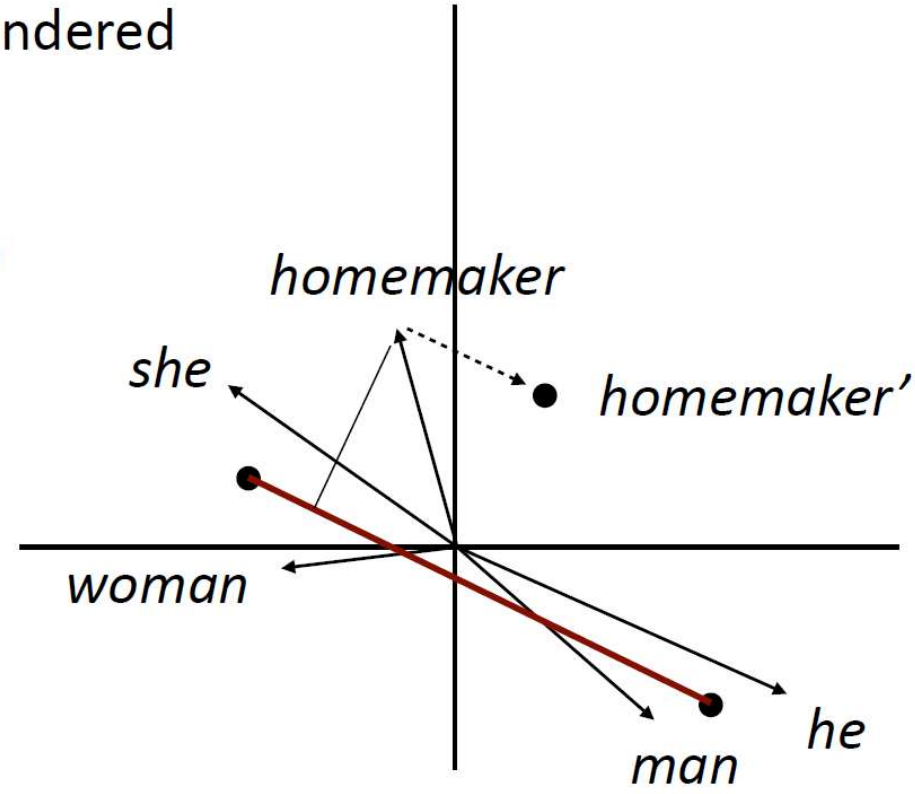
- Embeddings can capture biases in the data! (Bolukbasi et al 16)

$$\overrightarrow{\text{man}} - \overrightarrow{\text{woman}} \approx \overrightarrow{\text{king}} - \overrightarrow{\text{queen}}$$

- Debiasing methods (as in Bolukbasi et al 16) are an active area of research

Debiasing?

- Identify gender subspace with gendered words
- Project words onto this subspace
- Subtract those projections from the original word

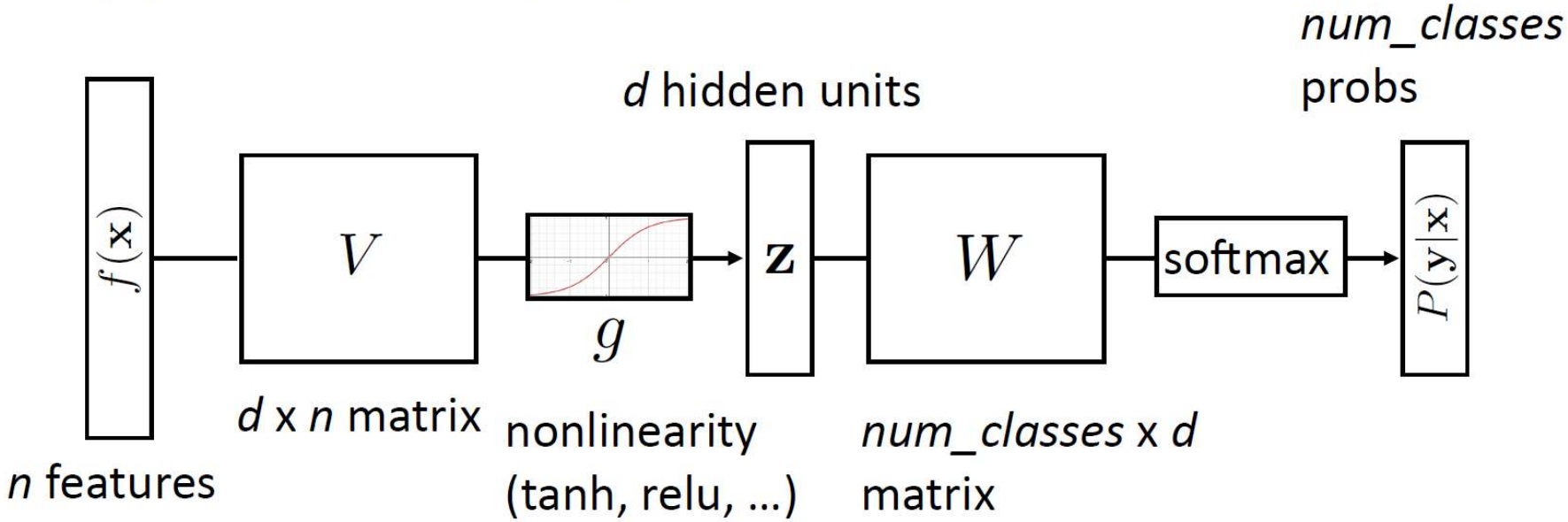


Bolukbasi et al. (2016)

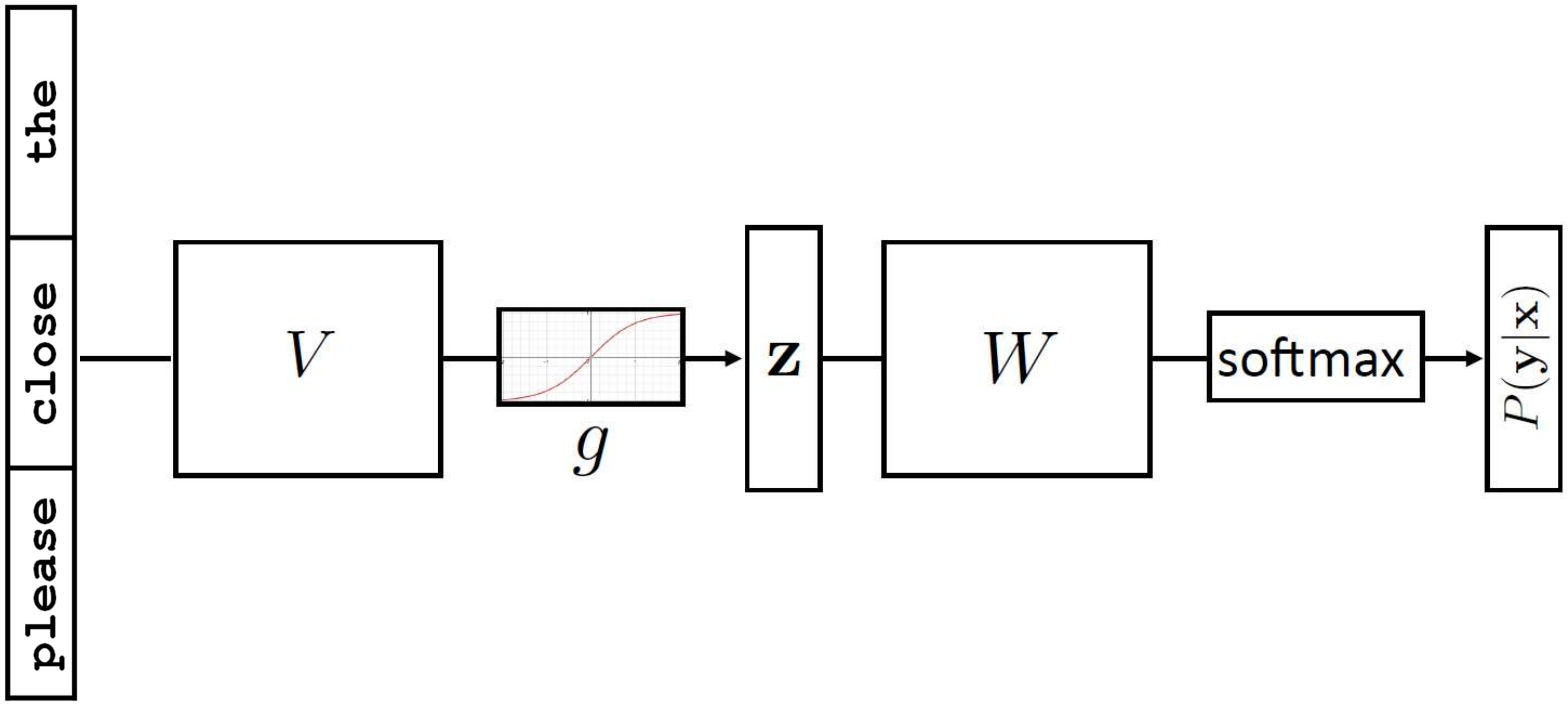
Neural Language Models

Reminder: Feedforward Neural Nets

$$P(\mathbf{y}|\mathbf{x}) = \text{softmax}(W\,g(V\,f(\mathbf{x})))$$

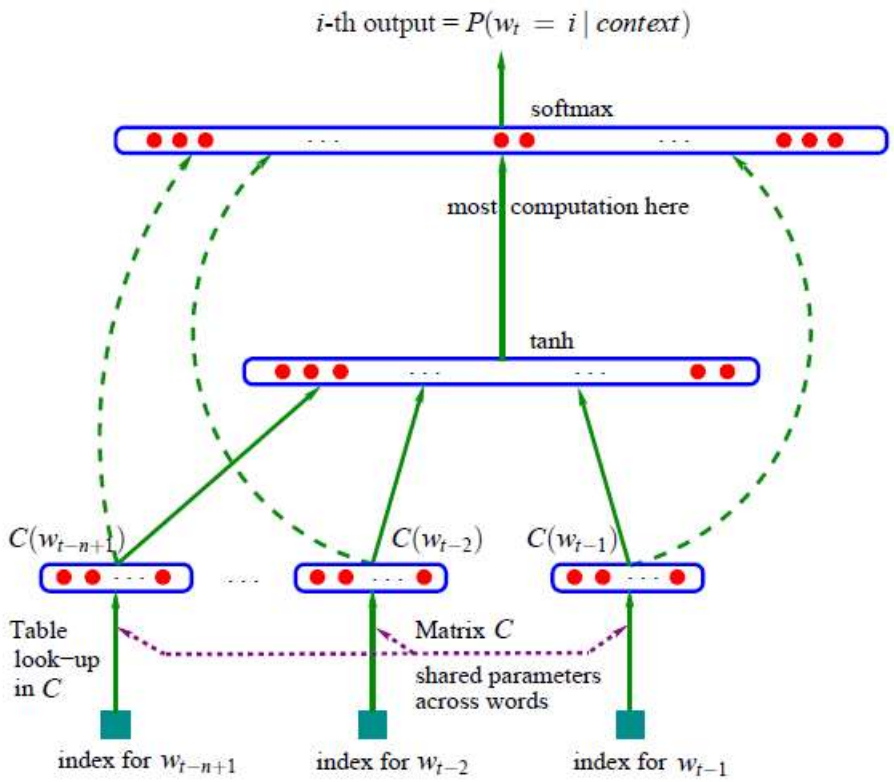


A Feedforward N-Gram Model?



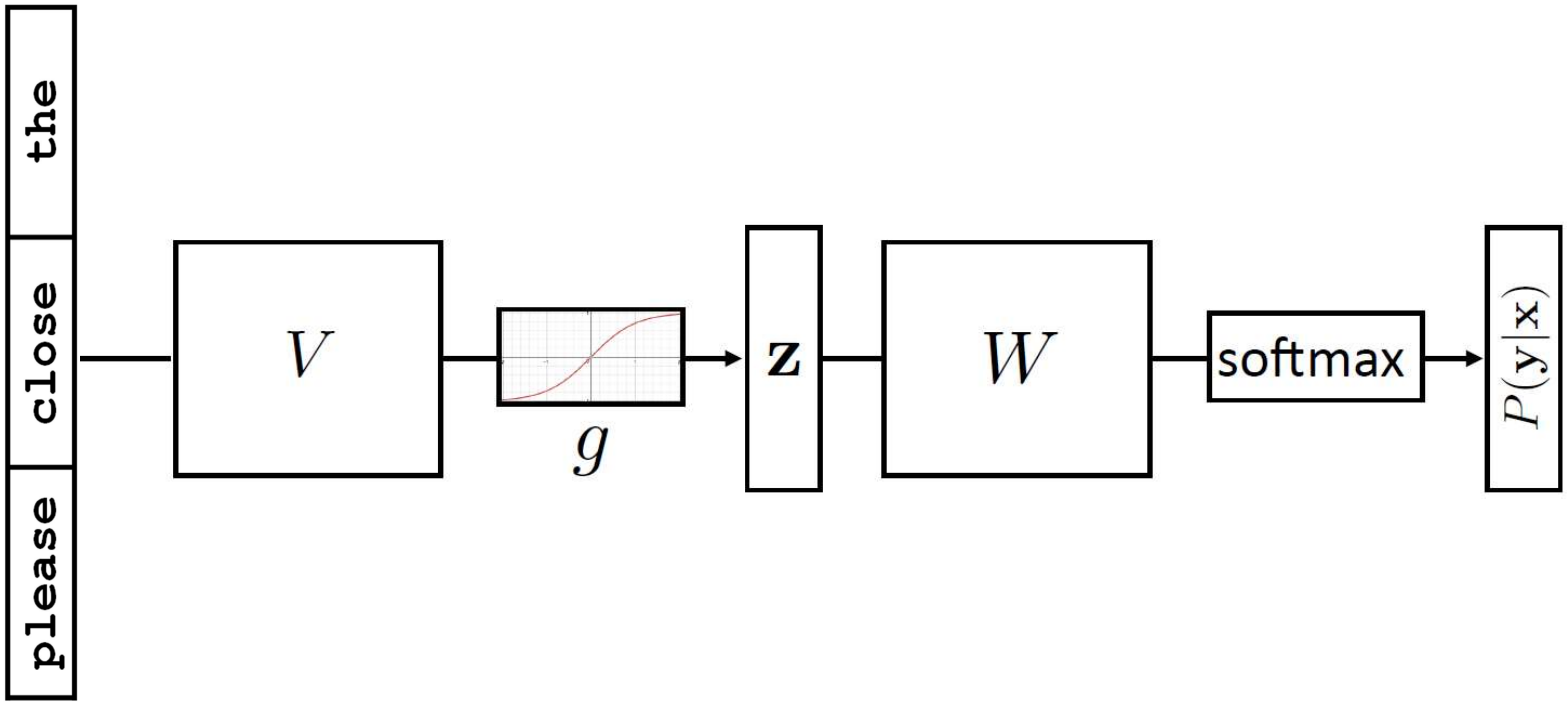
Early Neural Language Models

- Fixed-order feed-forward neural LMs
  - Eg Bengio et al 03
  - Allow generalization across contexts in more nuanced ways than prefixing
  - Allow different kinds of pooling in different contexts
  - Much more expensive to train



Bengio et al 03

Using Word Embeddings?



Using Word Embeddings

- Approach 1: learn embeddings as parameters from your data
  - Often works pretty well
- Approach 2: initialize using GloVe, keep fixed
  - Faster because no need to update these parameters
- Approach 3: initialize using GloVe, fine-tune
  - Works best for some tasks

Limitations of Fixed-Window NN LMs?

- What have we gained over N-Gram LMs?
- What have we lost?
- What have we not changed?