Neural NLP Tutorial

Language Modeling: Models of P(text)

Are These Sentences OK?

- Jane went to the store.
- store to Jane went the.
- Jane went store.
- Jane goed to the store.
- The store went to Jane.

Engineering Solutions

- Jane went to the store.
- store to Jane went the.
- Jane went store.
- Jane goed to the store.
- The store went to Jane.

} Create a grammar of the language

} morphology and exceptions

Consider Semantic preference categories,

What Can we Do w/ LMs?

- Score sentences:

Jane went to the store . →
high store to Jane went the .
→ low

(same as calculating loss for training)

- Generate sentences:

while didn't choose end-of-sentence symbol:
calculate probability
sample a new word from the probability distribution

Calculating the Probability of a Sentence

$$P(X) = \prod_{i=1}^{I} P(x_i \mid x_1, \ldots, x_{i-1})$$

Next Word

Context

This is a classification problem over the next word!

$$P(x_i \mid x_1, \ldots, x_{i-1})$$

How do we do this?!?!

Count-based Language Models

- Count up the frequency and divide:

$$P_{ML}(x_i \mid x_{i-n+1}, \ldots, x_{i-1}) := \frac{c(x_{i-n+1}, \ldots, x_i)}{c(x_{i-n+1}, \ldots, x_{i-1})}$$

https://books.google.com/ngrams

Any problem?

Problems w/ Count-based Models

- Cannot share strength among **similar words**

she bought a car

she purchased a car

she bought a bicycle

she purchased a bicycle

- Cannot handle **long-distance dependencies**

for tennis class he wanted to buy his own racquet

for programming class he wanted to buy his own computer

A Slight Simplification: Sentence Classification Models of P(label | text)

An Example Prediction Problem: Sentence Classification

I hate this movie

very good
good
neutral
bad
very bad

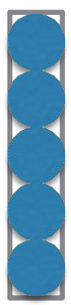
I love this movie

very good
good
neutral
bad
very bad

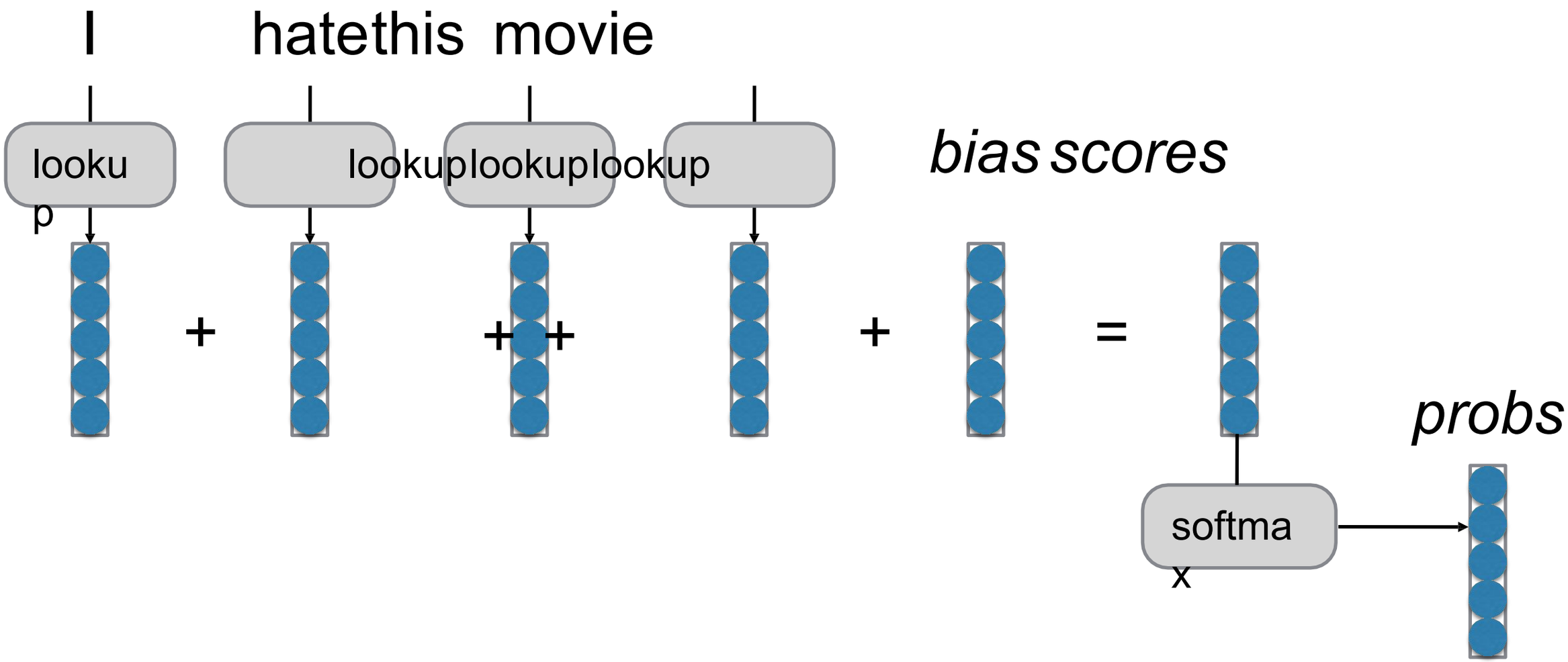
Word Representation (task-specific)

- Each word has its own 5 elements corresponding to [very good, good, neutral, bad, very bad]

- “hate” will have a high value for “very bad”, etc.



A First Try:
Bag of Words (BOW)

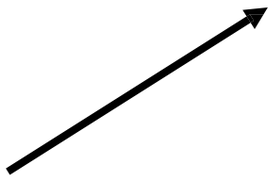


Adversarial Examples

I don’t love this movie

very good
good
neutral
bad
very bad

There’s nothing I don’t love about this movie



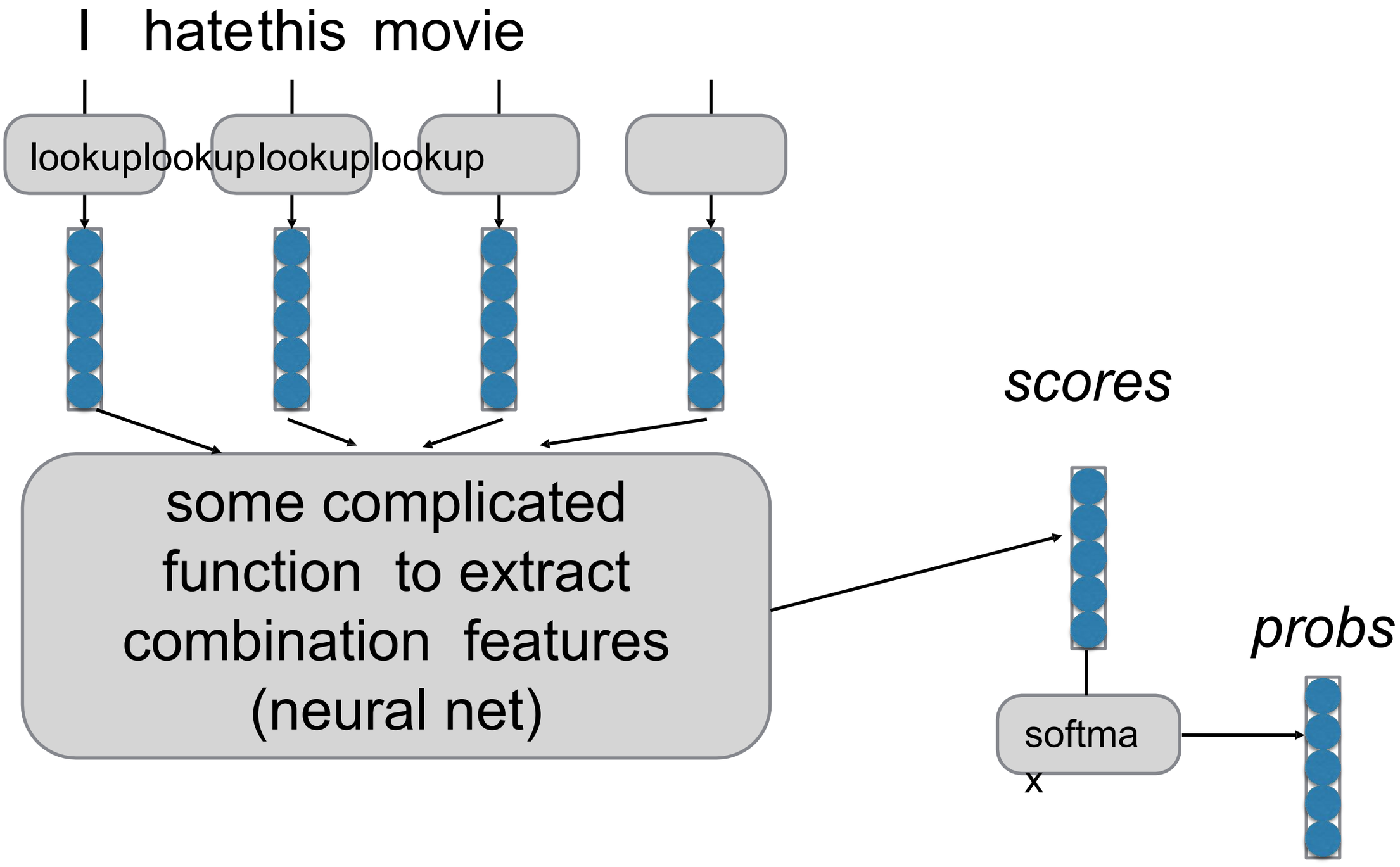
very good
good
neutral
bad
very bad

Combination Features

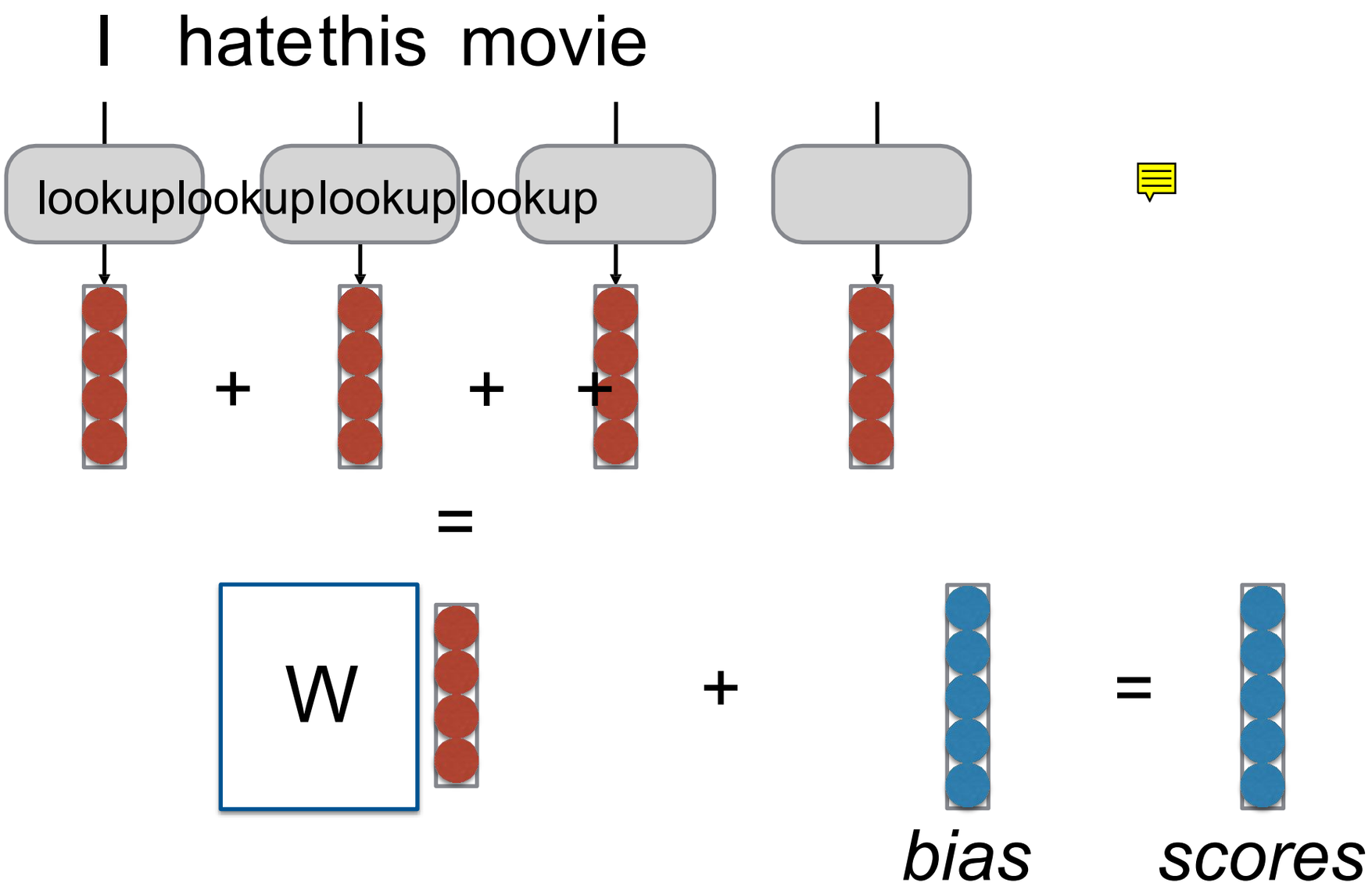
- Does it contain “don’t” and “love”?

- Does it contain “don’t”, “i”, “love”, and “nothing”?

Idea 1: Neural Networks



Idea 2: Continuous Bag of Words (CBOW)



What do Our Vectors Represent?

- Each vector has “features” (e.g. is this an animate object? is this a positive word, etc.)
- We sum these features, then use these to make predictions
- Still no combination features: only the expressive power of a linear model, but dimension reduced

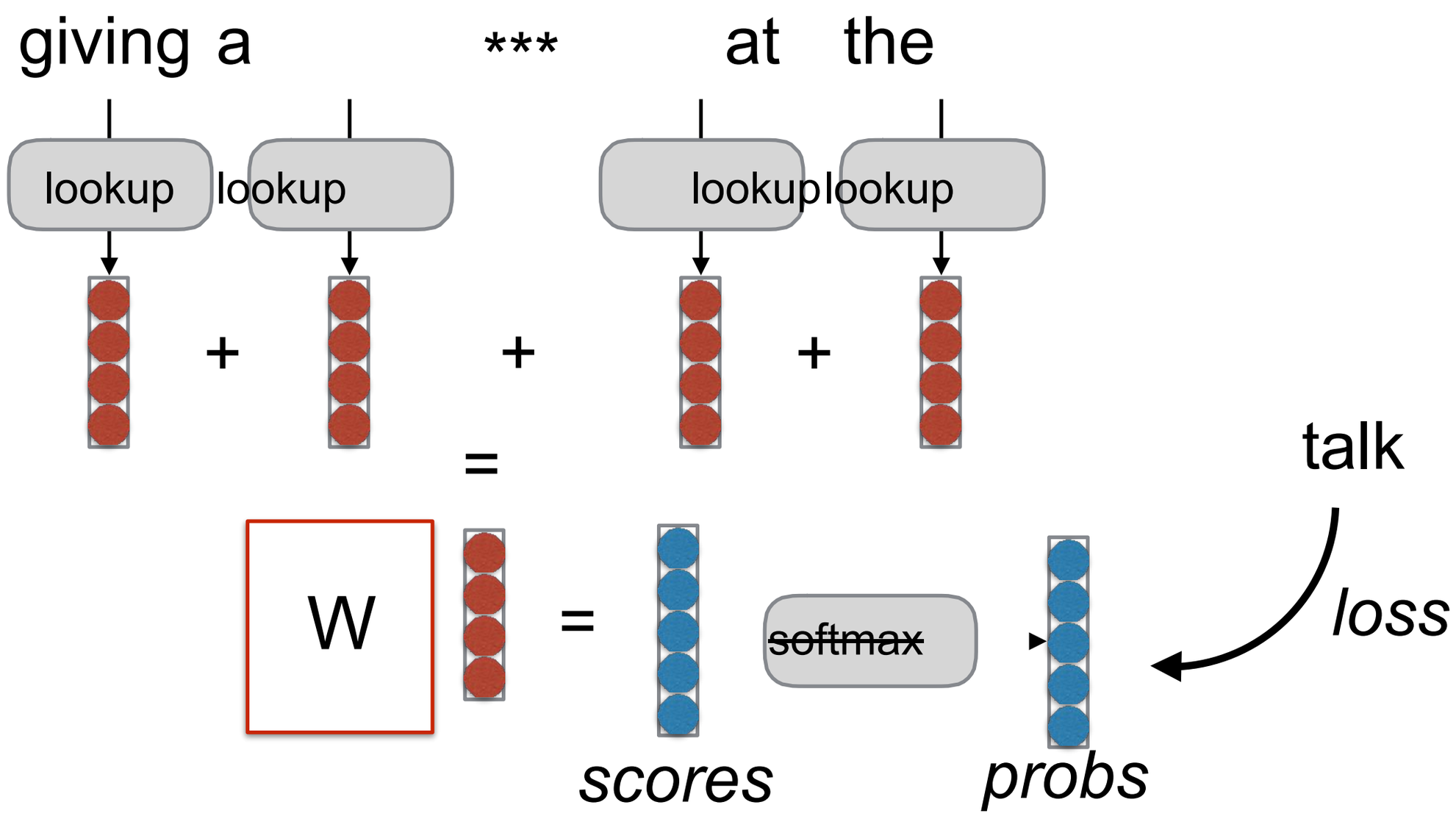
Revisiting LM

LM as a classification problem into words

CBO

(Mikolov et al. 2013)

- Predict word based on sum of surrounding embeddings



Softma

X

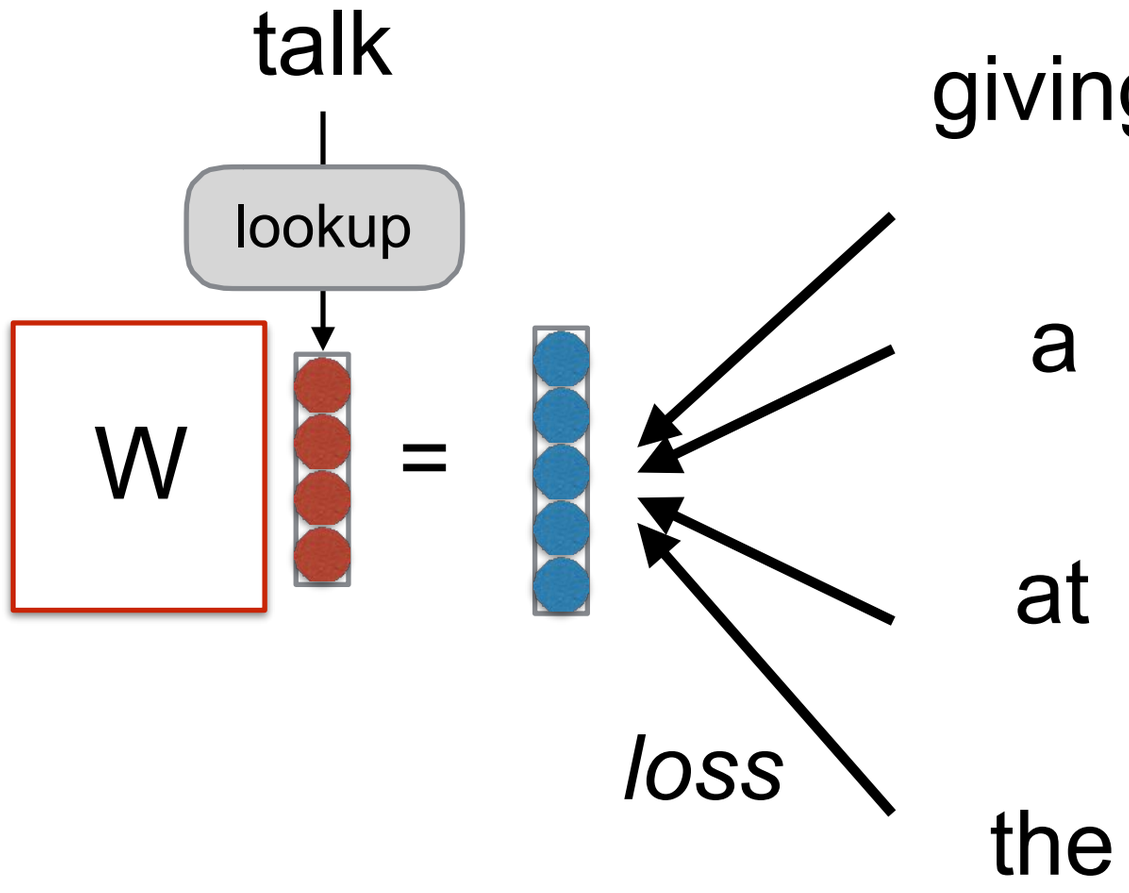
- Convert scores into probabilities by taking the exponent and normalizing (softmax)

$$P(x_i \mid x_{i-n+1}^{i-1}) = \frac{e^{s(x_i|x_{i-n+1}^{i-1})}}{\sum_{\tilde{x}_i} e^{s(\tilde{x}_i|x_{i-n+1}^{i-1})}}$$

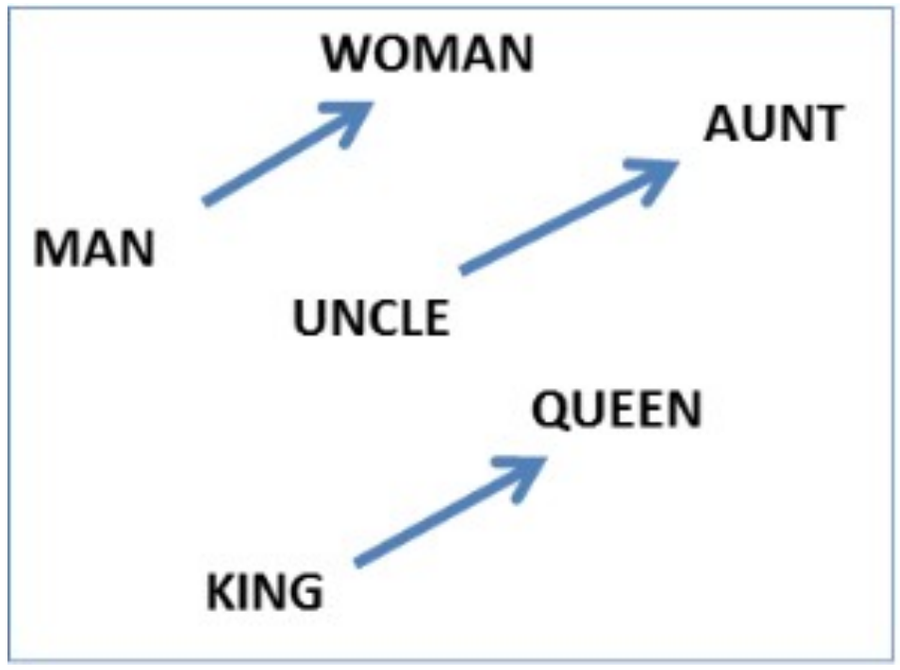
s = (-3.2, -2.9, 1.0, 2.2, 0.6, …) → p = (0.002, 0.003, 0.329, 0.444, 0.090, …)

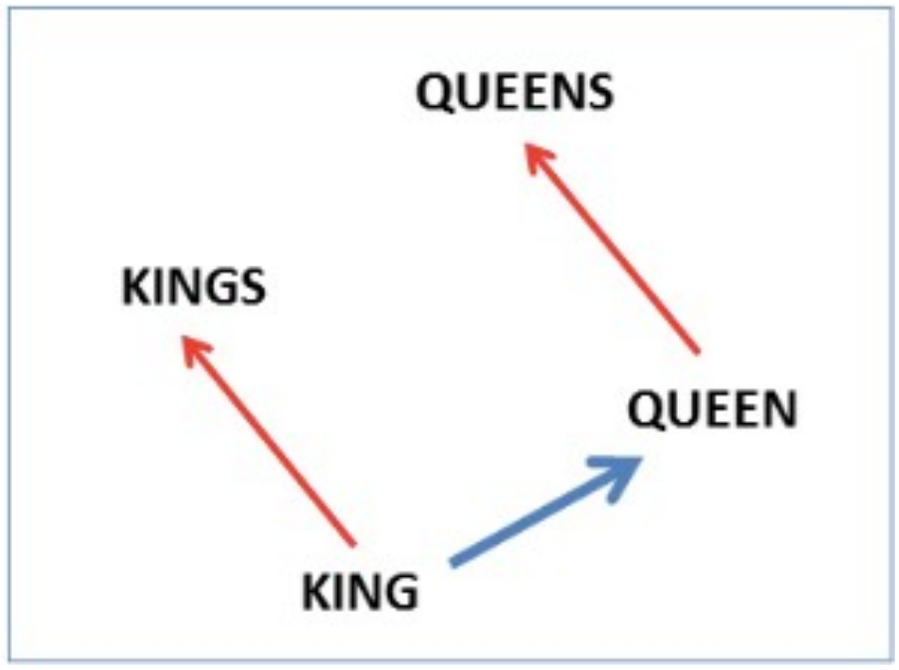
Skip-gram
(Mikolov et al. 2013)

- Predict each word in the context given the word



Word Embeddings

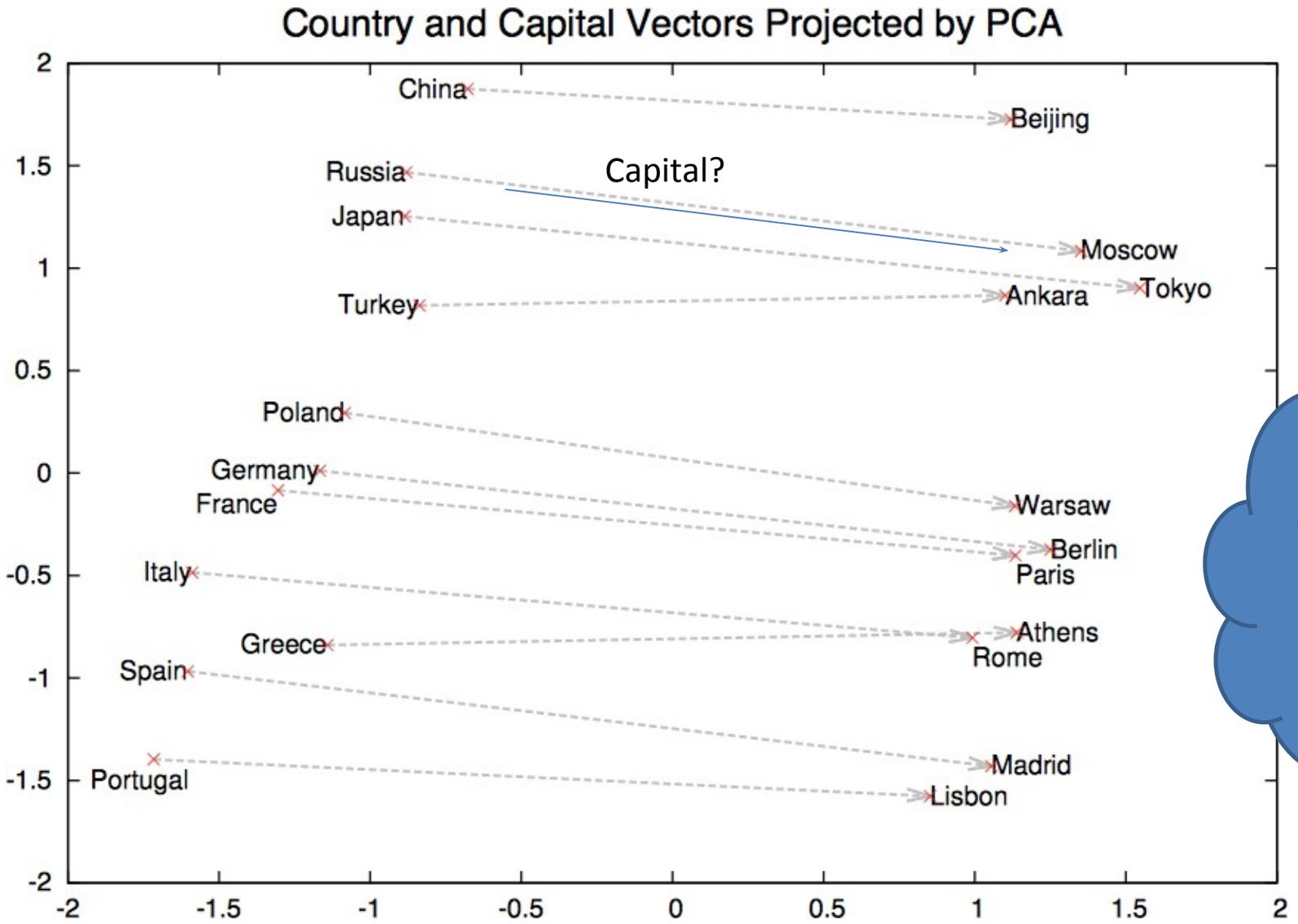




- “What is the female equivalent of king?” is not easily accessible in many traditional resources

Visualization of Embeddings

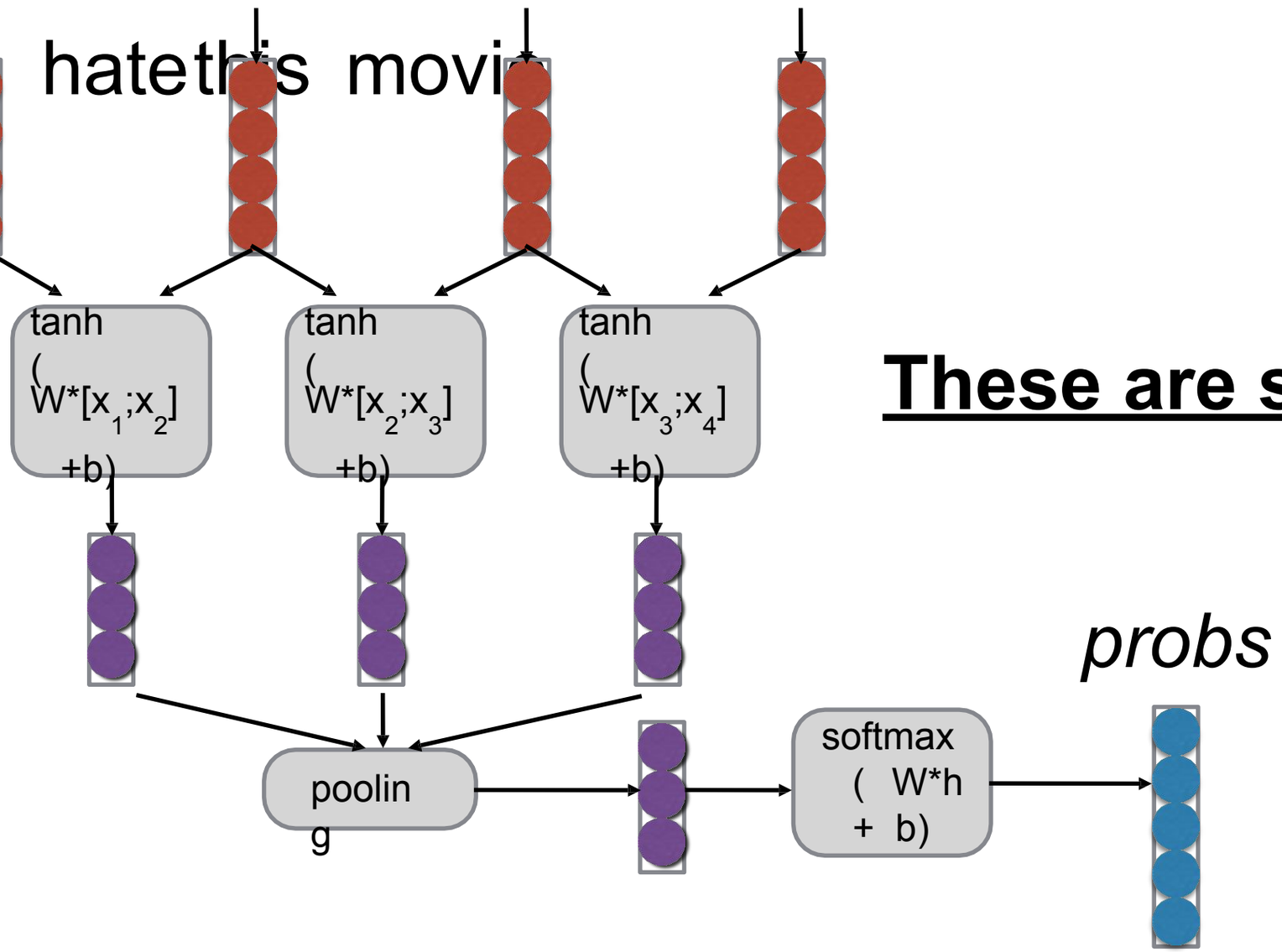
- Reduce high-dimensional embeddings into 2/3D for visualization (e.g. Mikolov et al. 2013)



(Korea,?, Seoul) =
True
If h+?=t

Time Delay Neural Networks

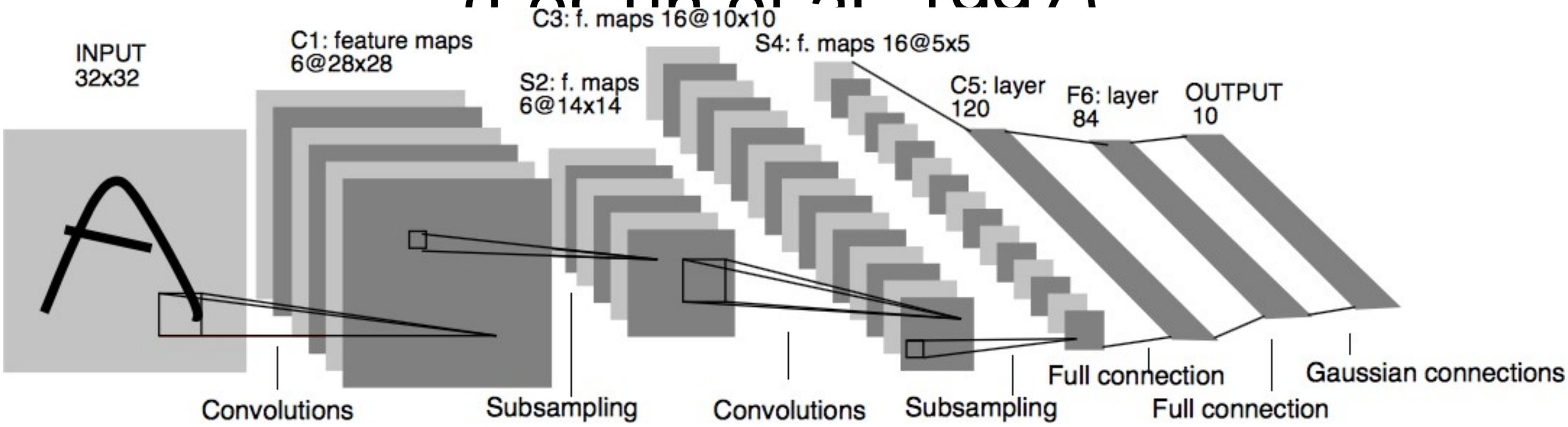
(Waibel et al. 1989)



These are soft 2-grams!

Convolutional Networks

(LeCun et al. 1997)



Parameter extraction performs a 2D sweep, not 1D

Poolin g

- Calculate some reduction function feature-wise
- **Max pooling:** “Did you see this feature anywhere in the range?” (most common)
- **Average pooling:** “How prevalent is this feature over the entire range”
- **k-Max, Dynamic, ...**

Weaknesses of CNNs

- CNNs are great for short-distance feature extractors
- But don't have holistic view of the sentence to capture long-distance dependencies

Long-distance Dependencies in Language

- Agreement in number, gender, etc.

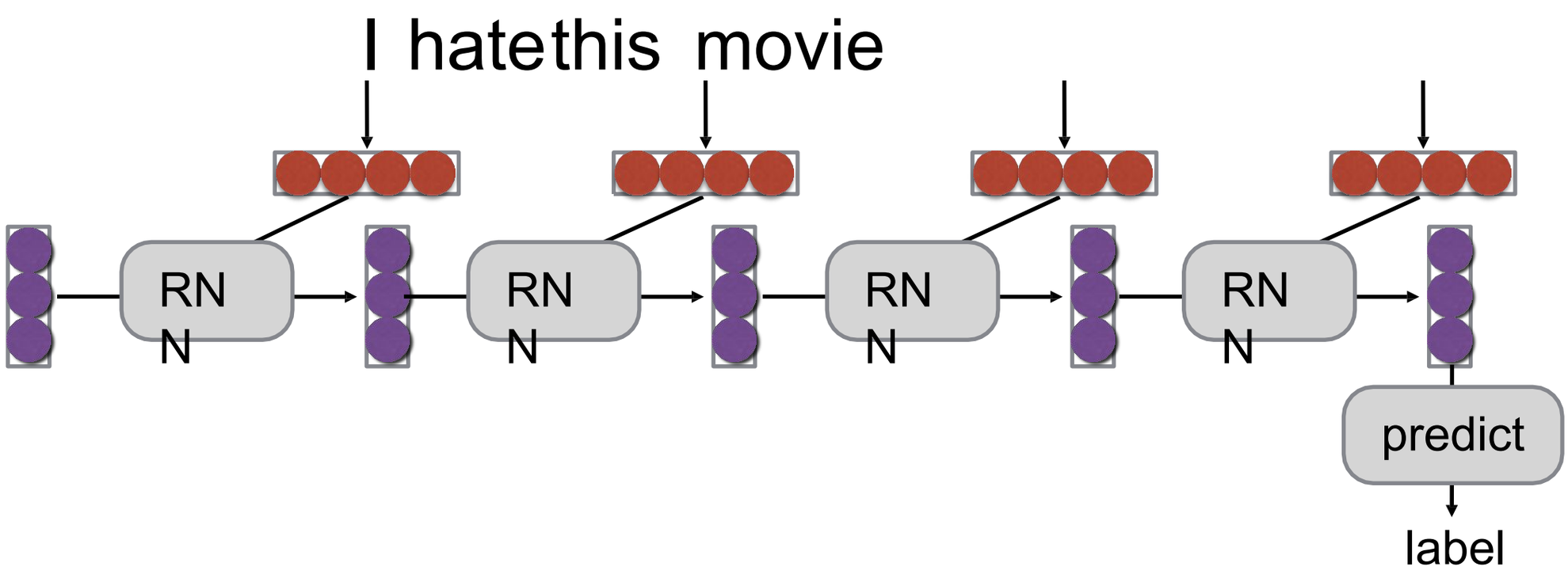
He does not have very much confidence in **himself**. **She** does not have very much confidence in **herself**.

- Selectional preference

The **reign** has lasted as long as the life of the **queen**. The **rain** has lasted as long as the life of the **clouds**.

Remember History w/ RNNs

- What does processing a sequence look like?



Weaknesses of RNNs

- Indirect passing of information, credit assignment more difficult
  - Made better by LSTMs/GRUs/etc. but not perfect
- Can be slow, due to incremental processing