ANTONIS ANASTASOPOULOS
CS499 INTRODUCTION TO NLP

NEURAL LANGUAGE MODELS

https://cs.gmu.edu/~antonis/course/cs499-spring21/

STRUCTURE OF THIS LECTURE

1 Quick LM Recap

2 Recurrent Neural Nets

3 Evaluation

LANGUAGE MODELING

LANGUAGE MODELING

The goal is to obtain a model to compute:

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

What if we could have **infinite history**, instead of relying on finite n-grams?

AN ALTERNATIVE:
FEATURIZED LOG-LINEAR MODELS

AN ALTERNATIVE: FEATURIZED MODELS

Calculate features of the context

Based on the features, calculate probabilities

Optimize feature weights using gradient descent, etc.

EXAMPLE:

Previous words: “giving a"

Words we’re predicting	How likely are they?	How likely are they given prev. word is “a”?	How likely are they given 2nd prev. word is “giving”?	Total score
a	3.0	-6.0	-0.2	-3.2
the	2.5	-5.1	-0.3	-2.9
talk	-0.2	0.2	1.0	1.0
gift	0.1	0.1	2.0	2.2
hat	1.2	0.5	-1.2	0.6
…	…	…	…	…

$$b=\begin{pmatrix}3.0\\2.5\\-0.2\\0.1\\1.2\\\dots\end{pmatrix}\quad w_{1,a}=\begin{pmatrix}-6.0\\-5.1\\0.2\\0.1\\0.5\\\dots\end{pmatrix}\quad w_{2,giving}=\begin{pmatrix}-0.2\\-0.3\\1.0\\2.0\\-1.2\\\dots\end{pmatrix}\quad s=\begin{pmatrix}-3.2\\-2.9\\1.0\\2.2\\0.6\\\dots\end{pmatrix}$$

SOFTMAX

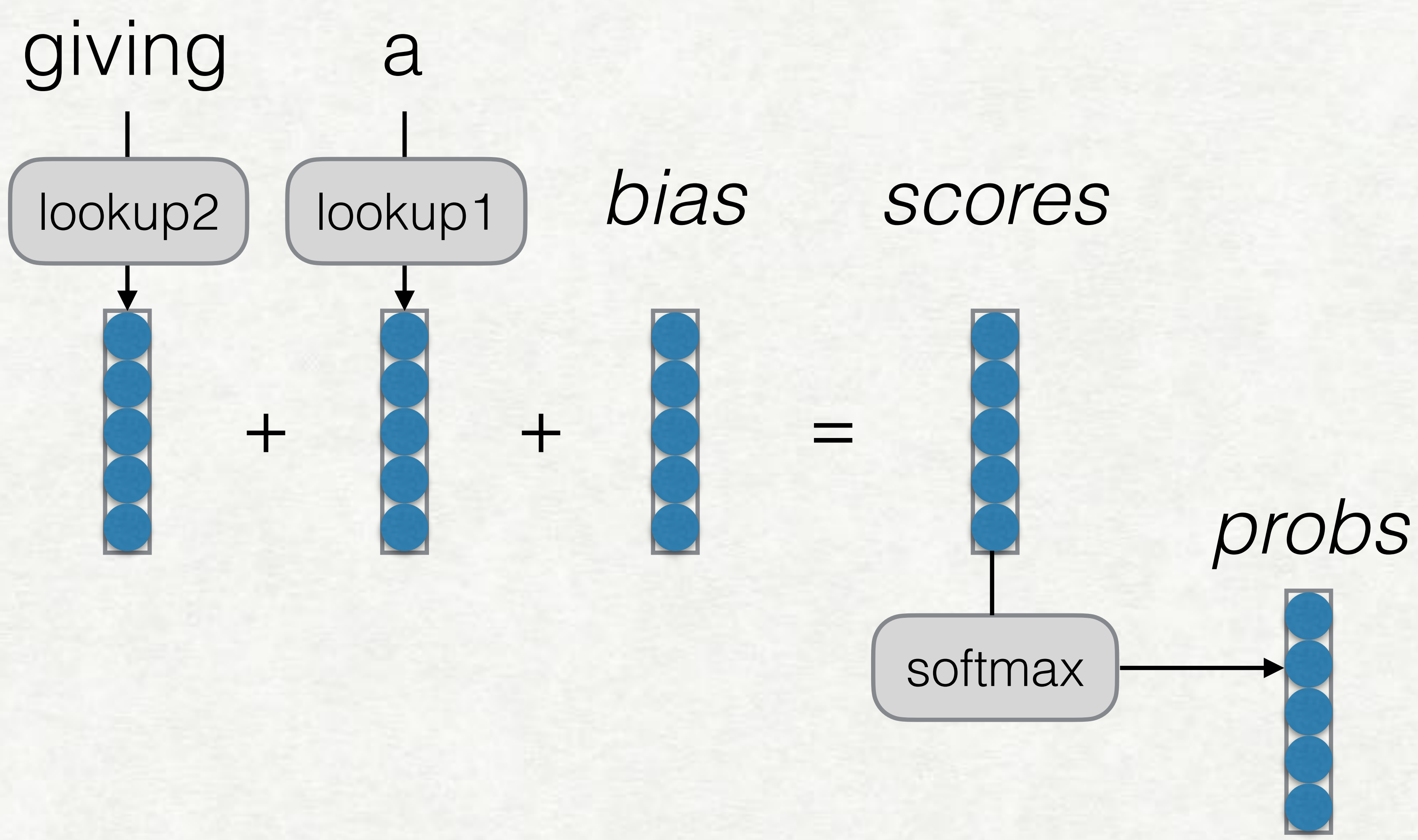
Convert scores into probabilities by taking the exponent and normalizing (softmax)

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

$$s = \begin{pmatrix} -3.2 \\ -2.9 \\ 1.0 \\ 2.2 \\ 0.6 \\ \dots \end{pmatrix} \longrightarrow p = \begin{pmatrix} 0.002 \\ 0.003 \\ 0.329 \\ 0.444 \\ 0.090 \\ \dots \end{pmatrix}$$

A COMPUTATION GRAPH VIEW

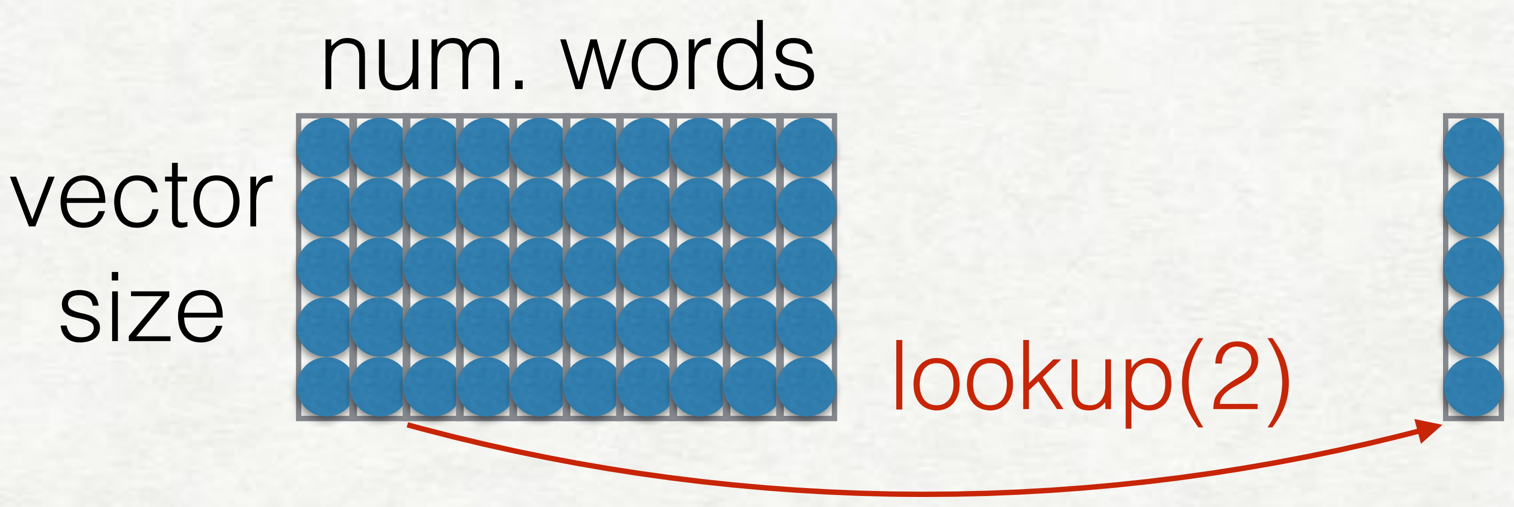
Each word has a vector of weights for each tag



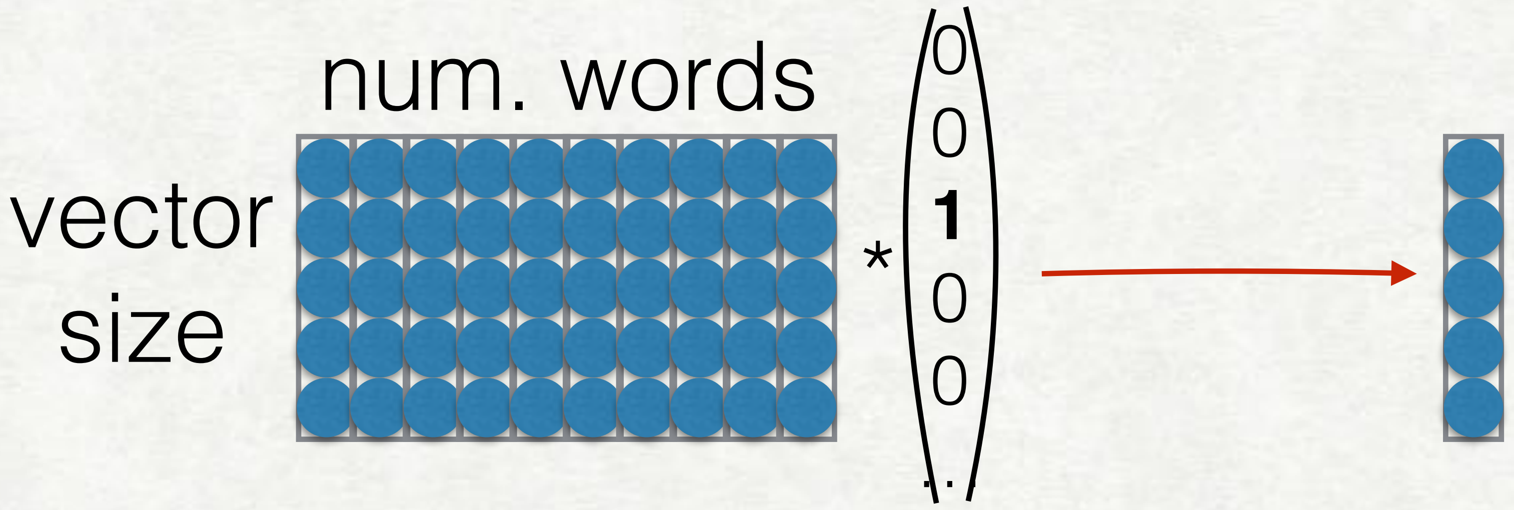
Each vector is size of output vocabulary

A NOTE: "LOOKUP"

Lookup can be viewed as "grabbing" a single vector from a big matrix of word embeddings



Similarly, can be viewed as multiplying with an "one-hot" vector



The former tends to be faster

TRAINING A NEURAL MODEL

To train, we calculate a "loss function" (a measure of how bad our predictions are), and move the parameters to reduce the loss

The most common loss function for probabilistic models is "negative log likelihood"

TRAINING A MODEL

Reminder: to train, we calculate a “loss function” (a measure of how bad our predictions are), and move the parameters to reduce the loss

The most common loss function for probabilistic models is “negative log likelihood”

If element 3 (or zero-indexed, 2) is the correct answer:

$$p = \begin{pmatrix} 0.002 \\ 0.003 \\ \boxed{0.329} \\ 0.444 \\ 0.090 \end{pmatrix} \ldots \xrightarrow{} -\log \xrightarrow{} 1.112$$

CHOOSING A VOCABULARY

UNKNOWN WORDS

Necessity for UNK words

- We won’t have all the words in the world in training data
- Larger vocabularies require more memory and computation time

Common ways:

- Frequency threshold (usually UNK <= 1)
- Rank threshold

UNKNOWN WORDS

A very large number of published documents contain text only. They often look boring, and they are often written in obscure language, using mile-long sentences and cryptic technical terms, using one font only, perhaps even without headings. Such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports, legal documents, or administrative papers. It is natural to think that such documents would benefit from a few illustrative images. (However, just adding illustration might be rather useless, if the text remains obscure and unstructured.)

UNKNOWN WORDS

a very large number of published documents contain text only . they often look boring , and they are often written in obscure language , using mile-long sentences and cryptic technical terms , using one font only , perhaps even without headings . such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports , legal documents , or administrative papers . it is natural to think that such documents would benefit from a few illustrative images . (however , just adding illustration might be rather useless , if the text remains obscure and unstructured .)

lowercase + tokenize

UNKNOWN WORDS

a very large number of published documents contain text only . they often look boring , and they are often written in obscure language , using mile-long sentences and cryptic technical terms , using one font only , perhaps even without headings . such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports , legal documents , or administrative papers . it is natural to think that such documents would benefit from a few illustrative images . (however , just adding illustration might be rather useless , if the text remains obscure and unstructured .)

Find rare words (e.g. with freq<2)

UNKNOWN WORDS

a very large number of published documents contain text only . they often look boring , and they are often written in obscure language , using UNK sentences and cryptic technical terms , using one font only , perhaps even without headings . such style, or lack of style, might be the one you are strongly expected to follow when writing eg scientific or technical reports , legal documents , or UNK papers . it is natural to think that such documents would benefit from a few illustrative images . (however , just adding UNK might be rather useless , if the text remains obscure and UNK .)

Substitute with UNK

EVALUATION AND VOCABULARY

Important: the vocabulary must be the same over models you compare

EVALUATION AND VOCABULARY

Important: the vocabulary must be the same over models you compare

Or more accurately, all models must be able to generate the test set (it's OK if they can generate *more* than the test set, but not less)

e.g. Comparing a character-based model to a word-based model is fair, but not vice-versa

RECURRENT NEURAL NETWORKS

LINEAR MODELS CAN'T LEARN FEATURE COMBINATIONS

farmers eat steak ➝ **high**
farmers eat hay ➝ **low**

cows eat steak ➝ **low**
cows eat hay ➝ **high**

These can't be expressed by linear features

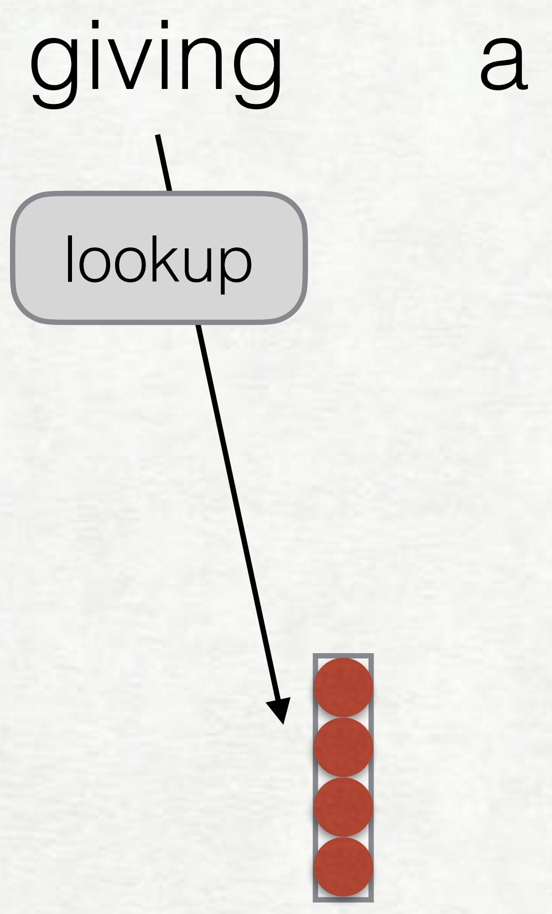
What can we do?

- Remember combinations as features (individual scores for "farmers eat", "cows eat")
  ➝ Feature space explosion!
- Neural nets

NEURAL LANGUAGE MODELS

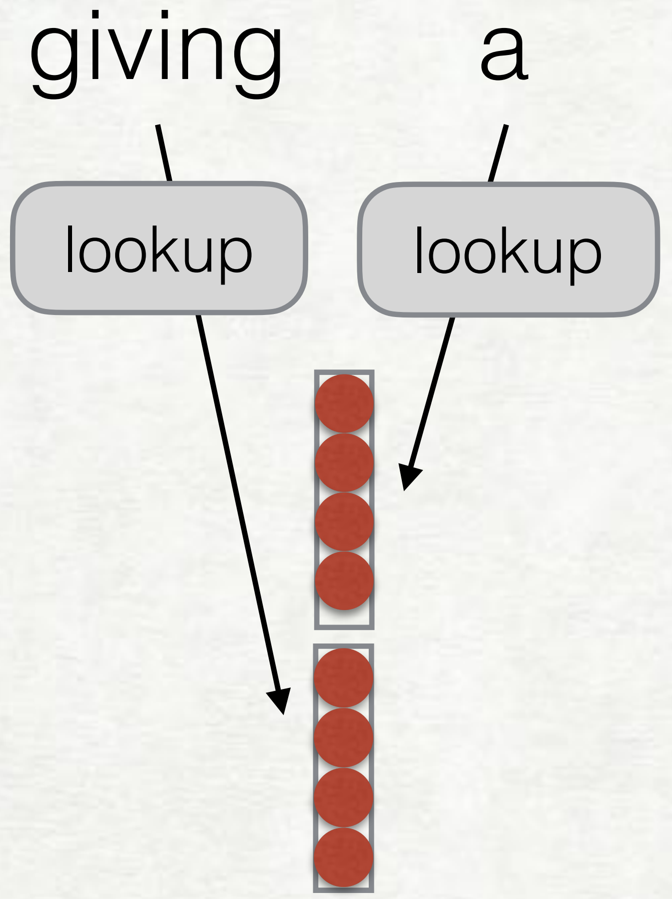
(See Bengio et al. 2004)

NEURAL LANGUAGE MODELS



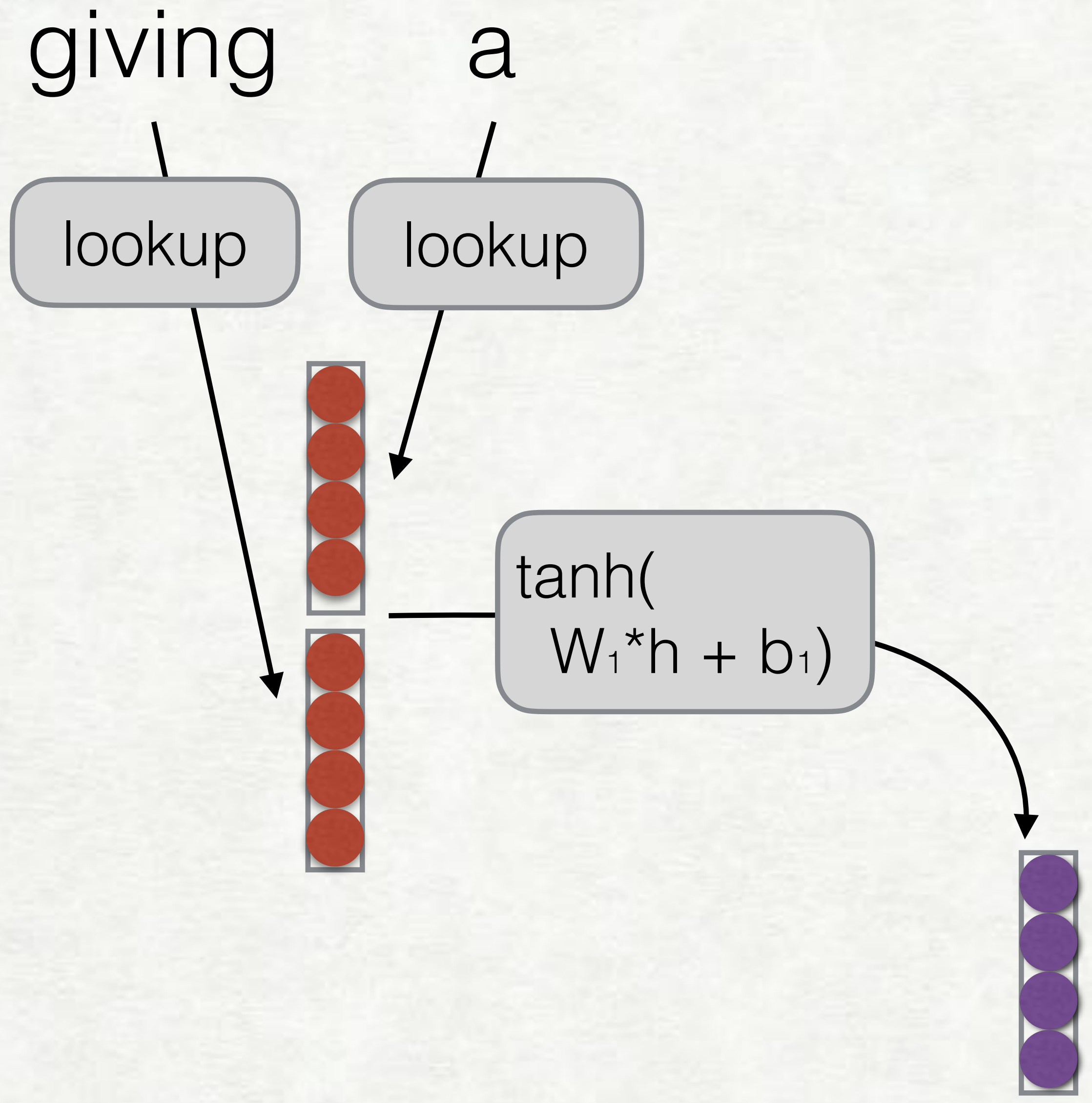
(See Bengio et al. 2004)

NEURAL LANGUAGE MODELS



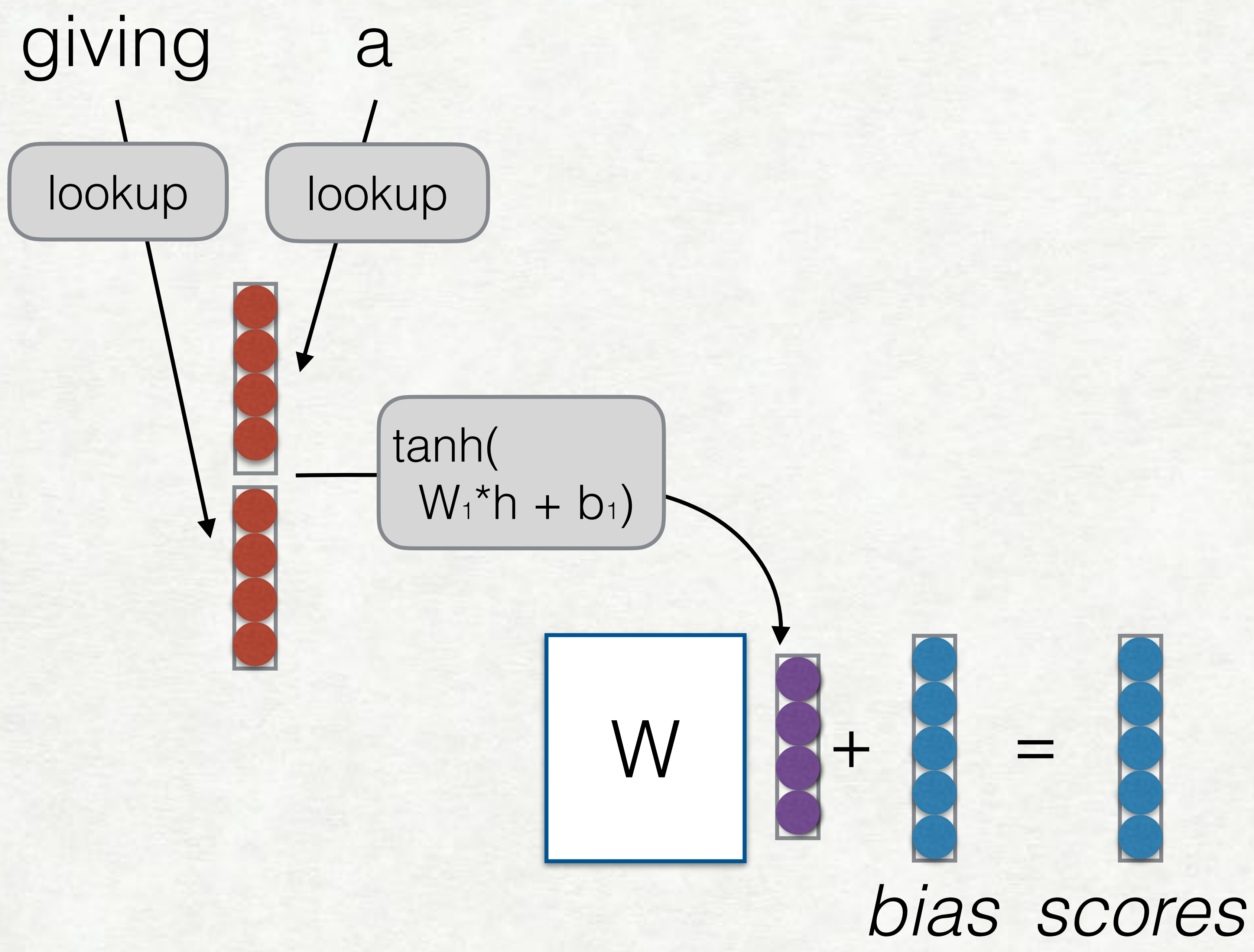
(See Bengio et al. 2004)

NEURAL LANGUAGE MODELS



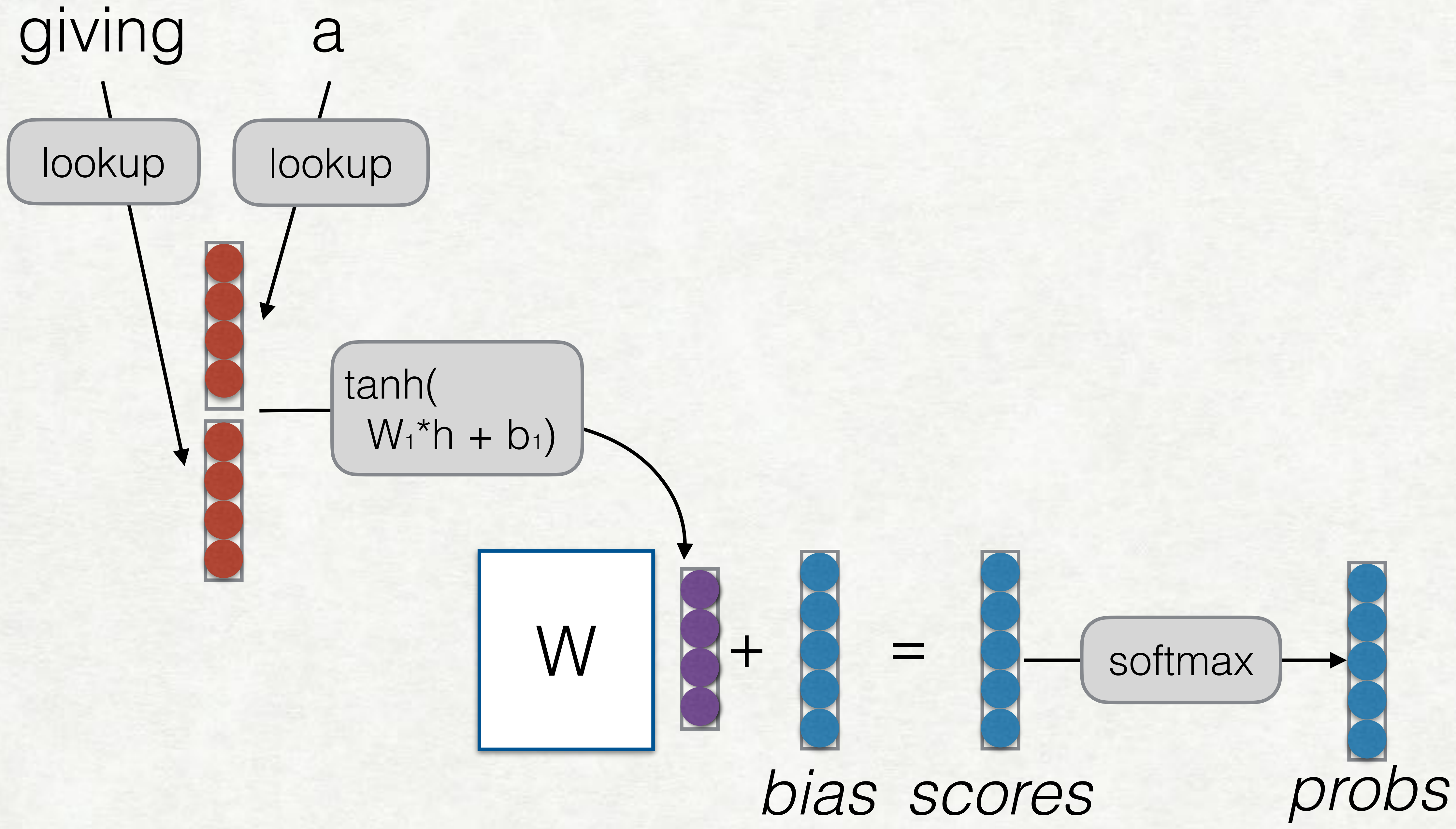
(See Bengio et al. 2004)

NEURAL LANGUAGE MODELS



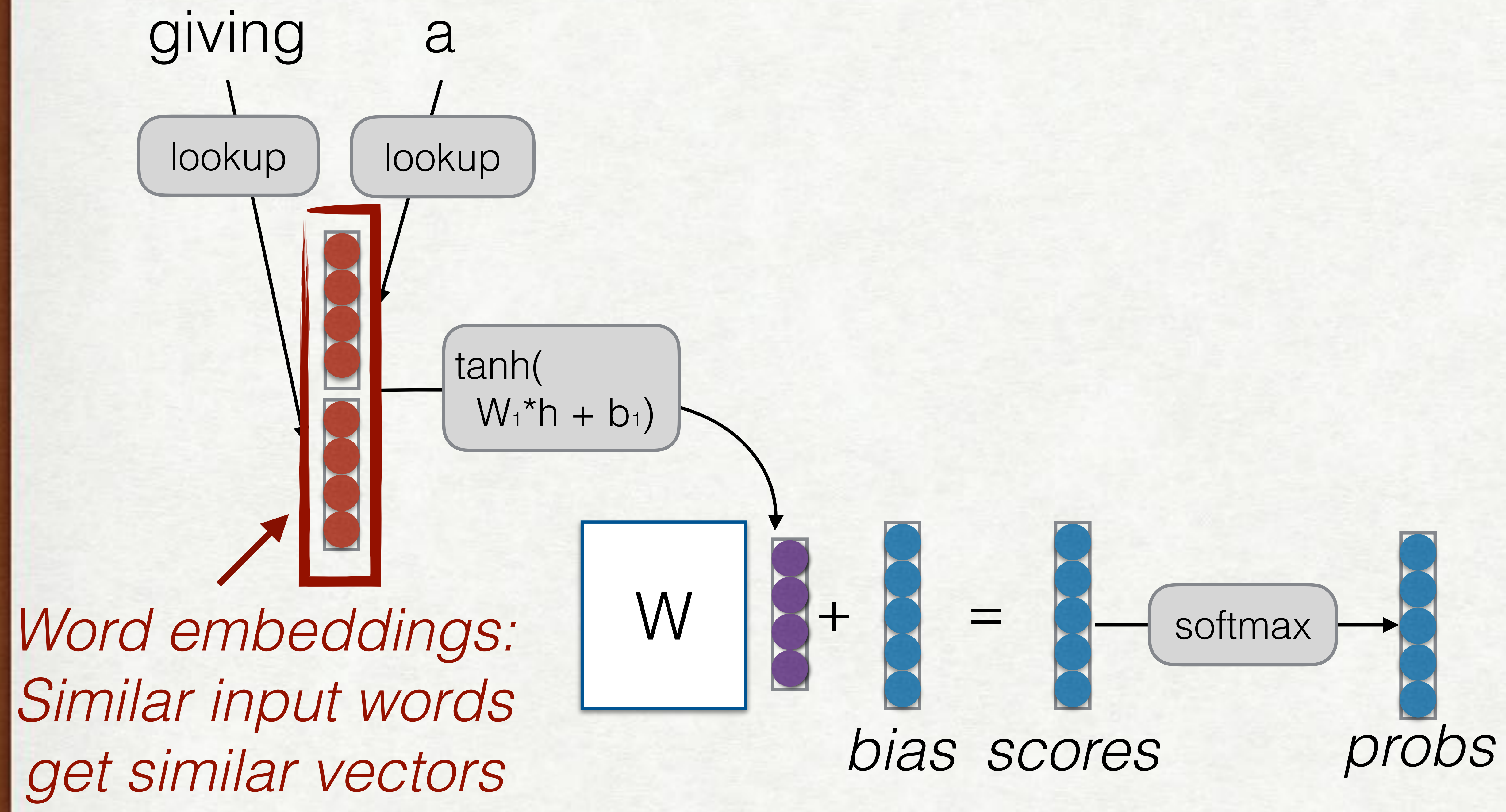
(See Bengio et al. 2004)

NEURAL LANGUAGE MODELS



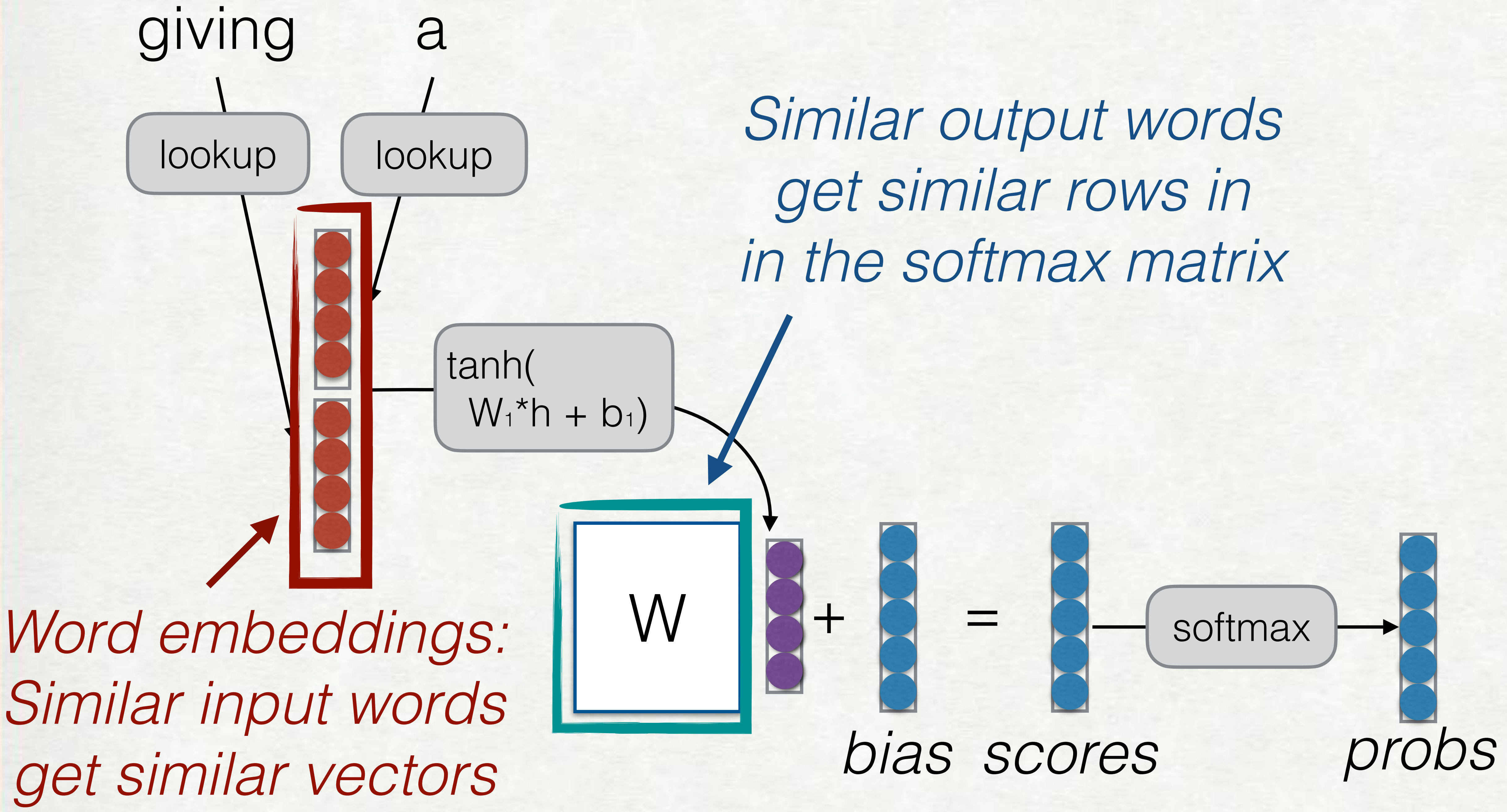
(See Bengio et al. 2004)

WHERE IS STRENGTH SHARED?



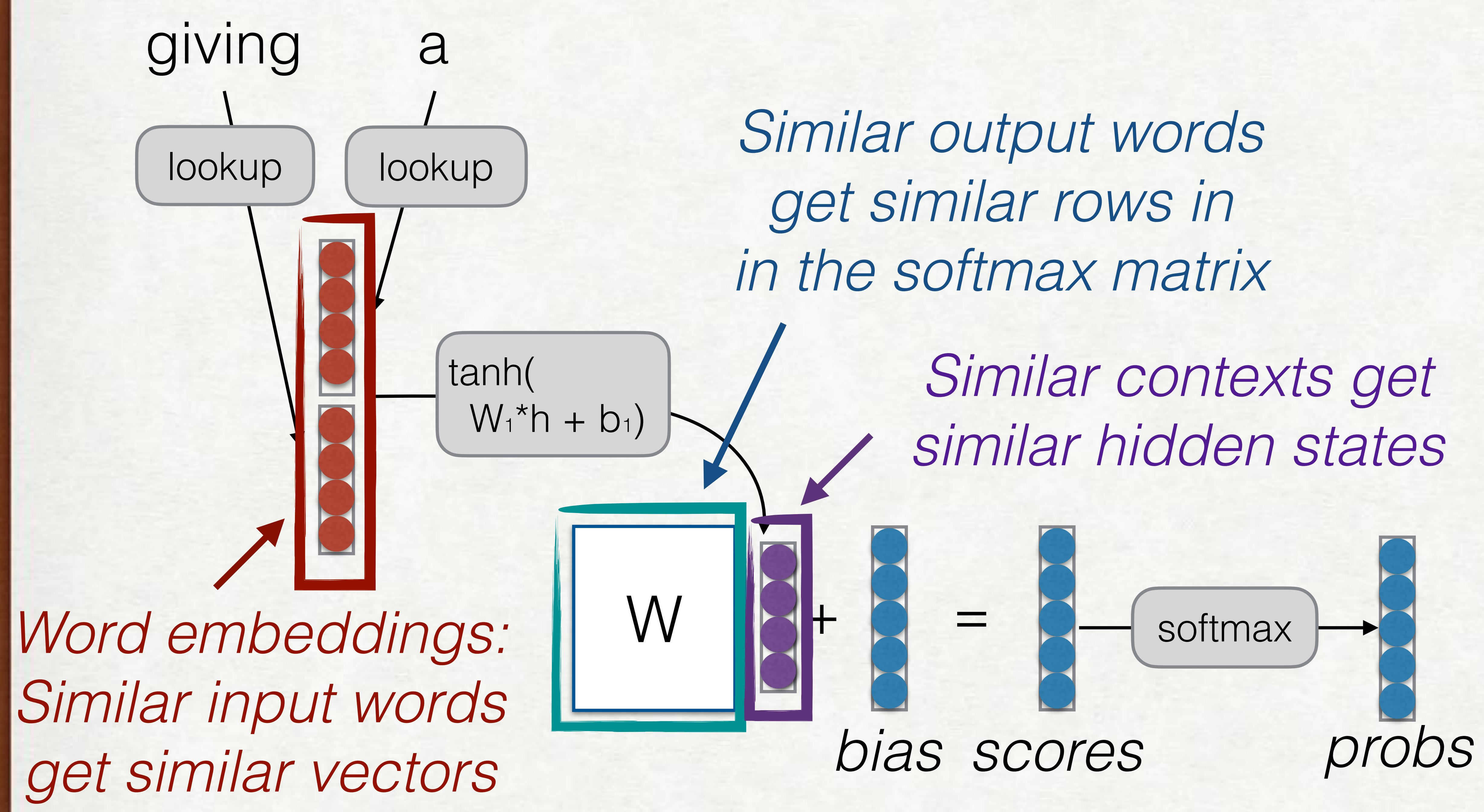
(See Bengio et al. 2004)

WHERE IS STRENGTH SHARED?



(See Bengio et al. 2004)

WHERE IS STRENGTH SHARED?



(See Bengio et al. 2004)

WHAT PROBLEMS ARE HANDLED?

Cannot share strength among **similar words**

she bought a car she purchased a car	she bought a bicycle she purchased a bicycle

→ solved, and similar contexts as well! 😀

Cannot condition on context with intervening words

Dr. Jane Smith	Dr. Gertrude Smith

→ solved! 😀

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

→ not solved yet 😞

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**.

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**.

CAN BE COMPLICATED!

What is the referent of "it"?

The trophy would not fit in the brown suitcase because it was too **big**.

Trophy

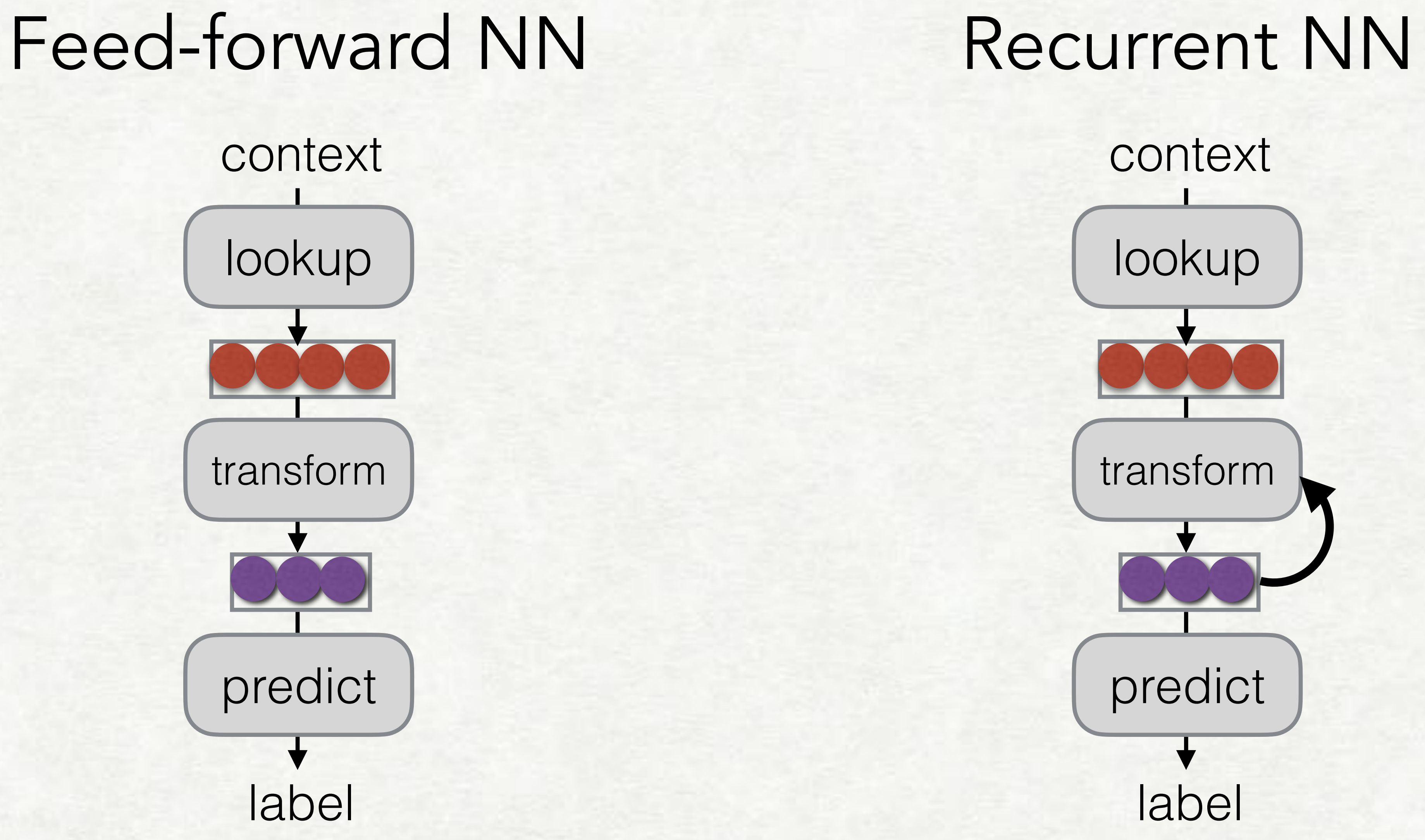
The trophy would not fit in the brown suitcase because it was too **small**.

Suitcase

(from Winograd Schema Challenge: http://commonsensereasoning.org/winograd.html)

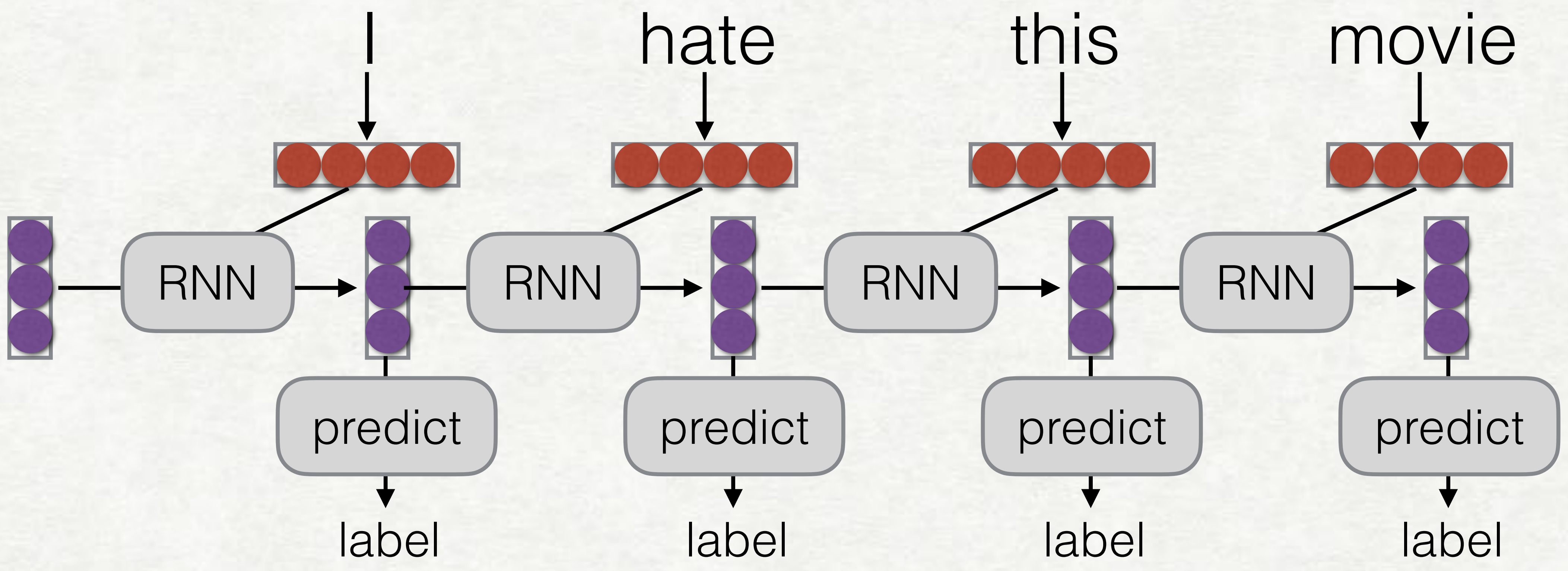
RECURRENT NEURAL NETWORKS (ELMAN 1990)

Tools to “remember” information



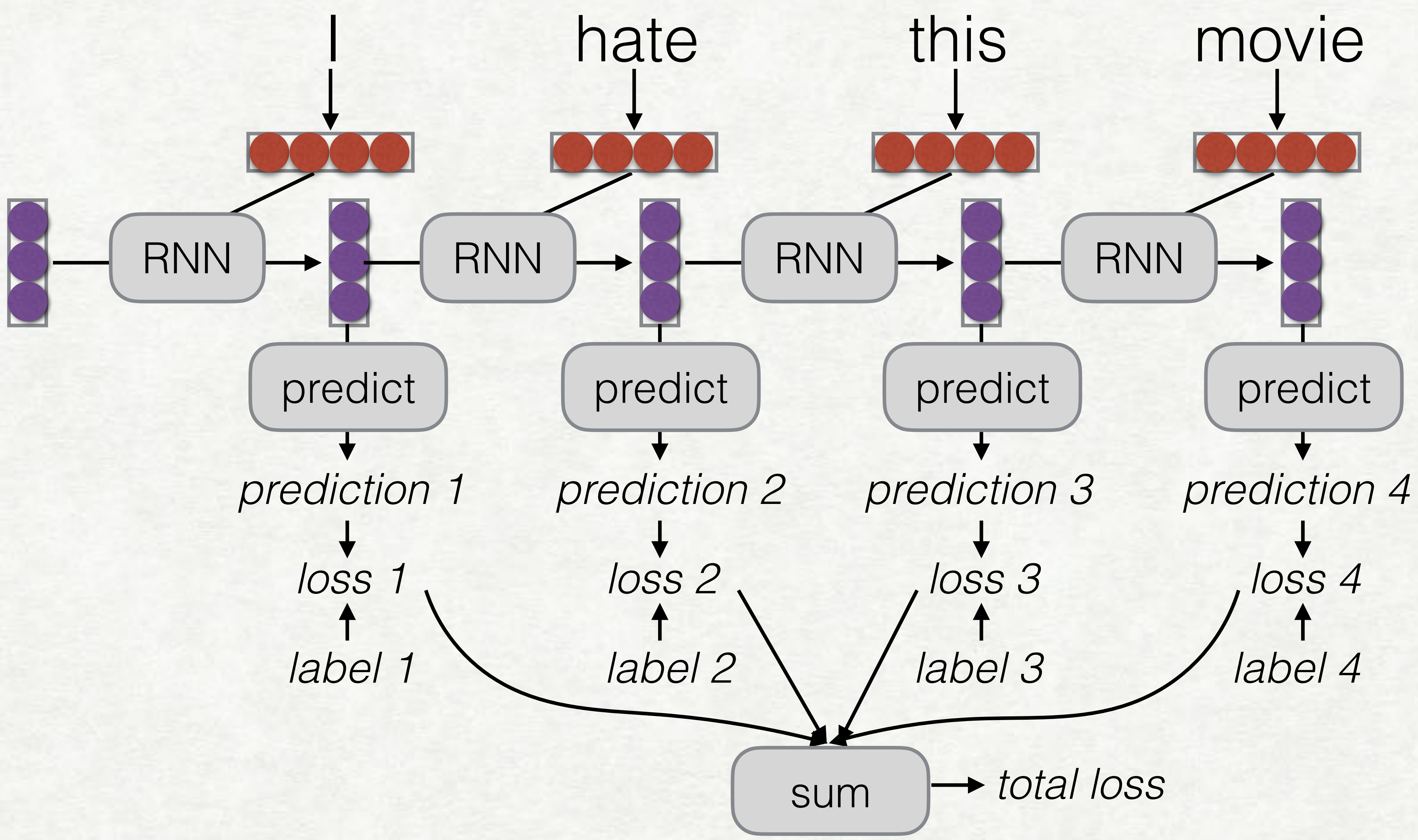
UNROLLING IN TIME

What does processing a sequence look like?



TRAINING RNNS

Calculating the loss

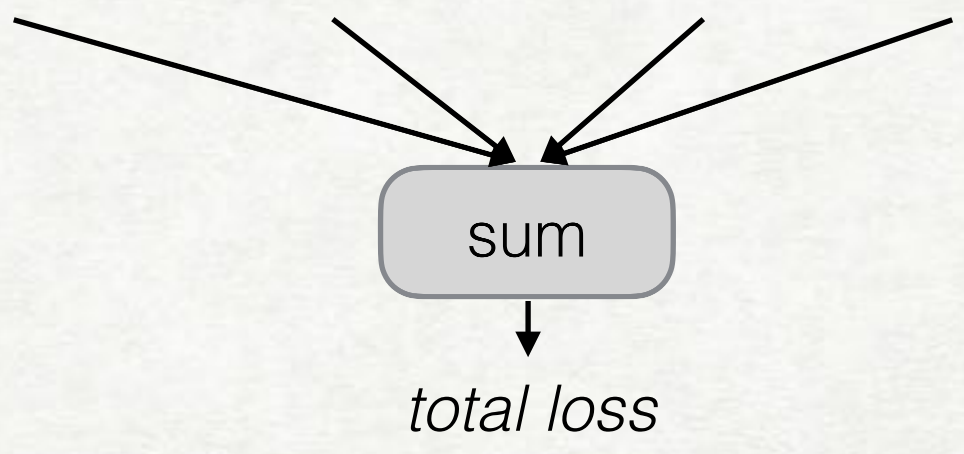


RNN TRAINING

The unrolled graph is a well-formed (DAG) computation graph—we can run backprop

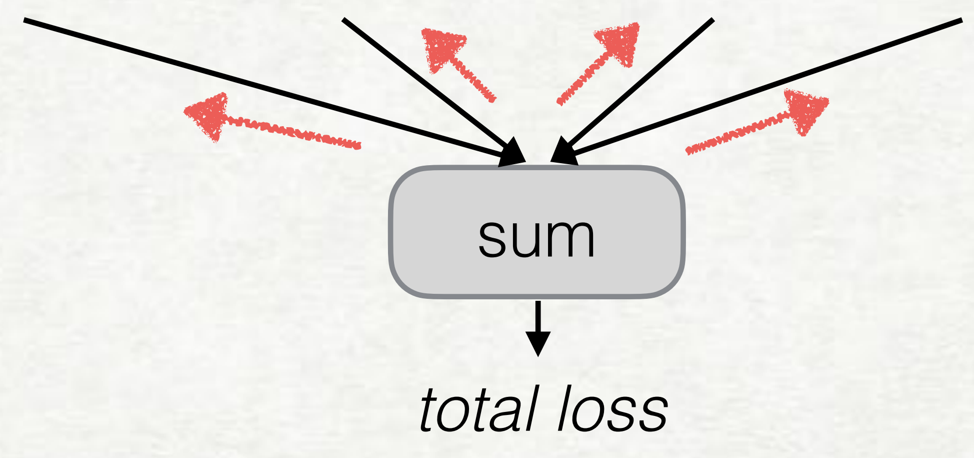
RNN TRAINING

The unrolled graph is a well-formed (DAG) computation graph—we can run backprop



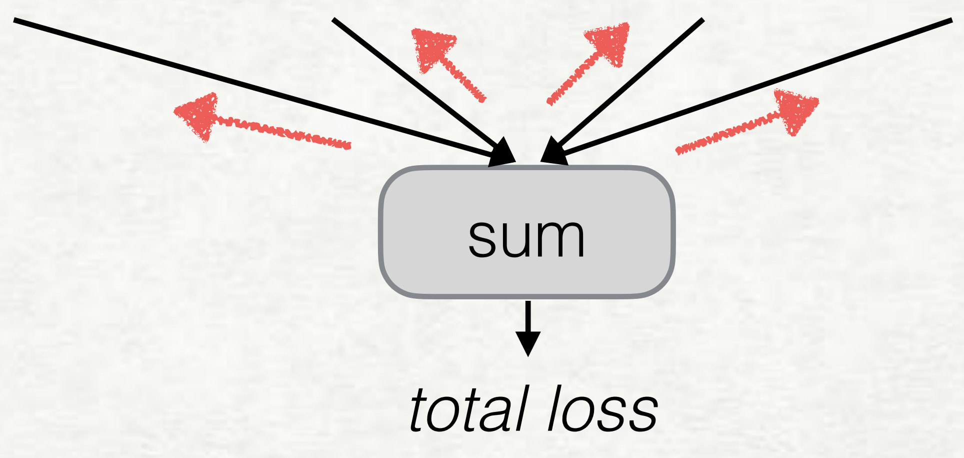
RNN TRAINING

The unrolled graph is a well-formed (DAG) computation graph—we can run backprop



RNN TRAINING

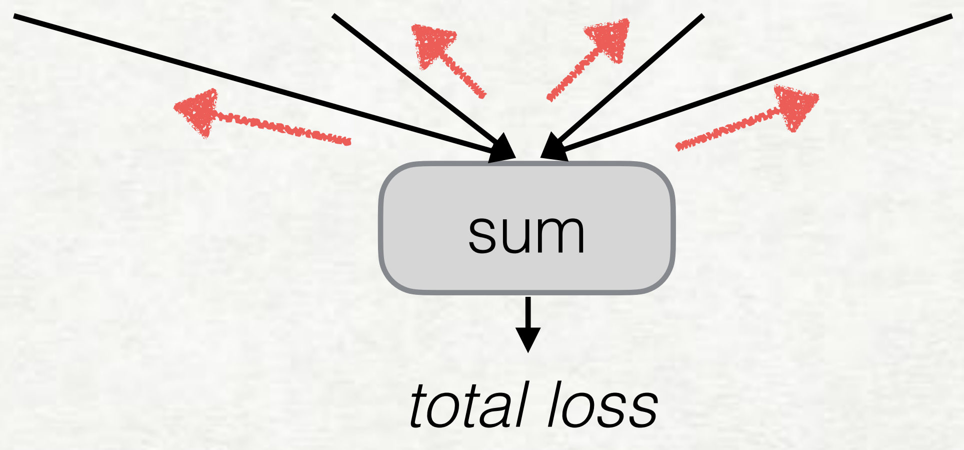
The unrolled graph is a well-formed (DAG) computation graph—we can run backprop



Parameters are tied across time, derivatives are aggregated across all time steps

RNN TRAINING

The unrolled graph is a well-formed (DAG) computation graph—we can run backprop

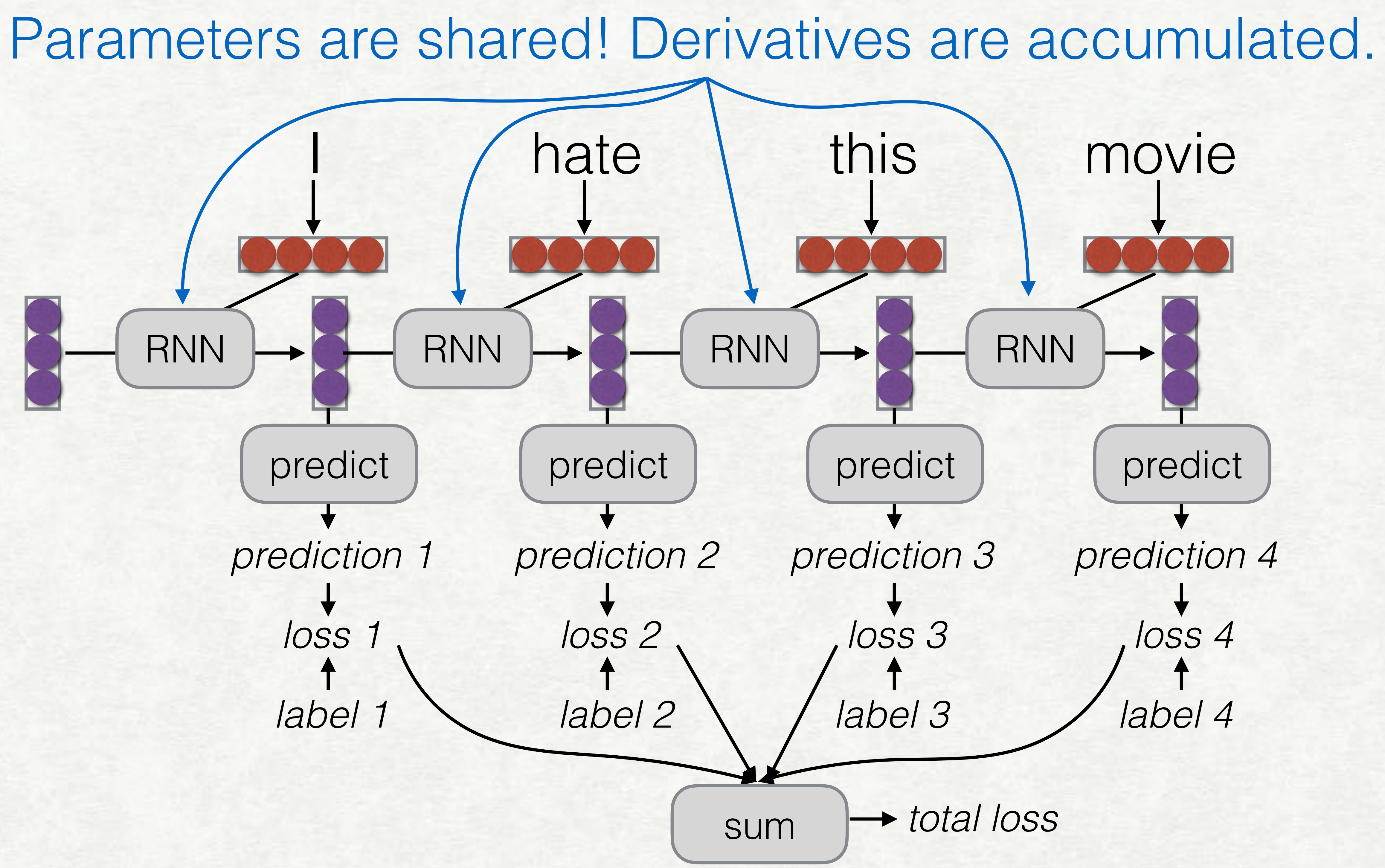


Parameters are tied across time, derivatives are aggregated across all time steps

This is historically called "backpropagation through time" (BPTT)

PARAMETER TYING

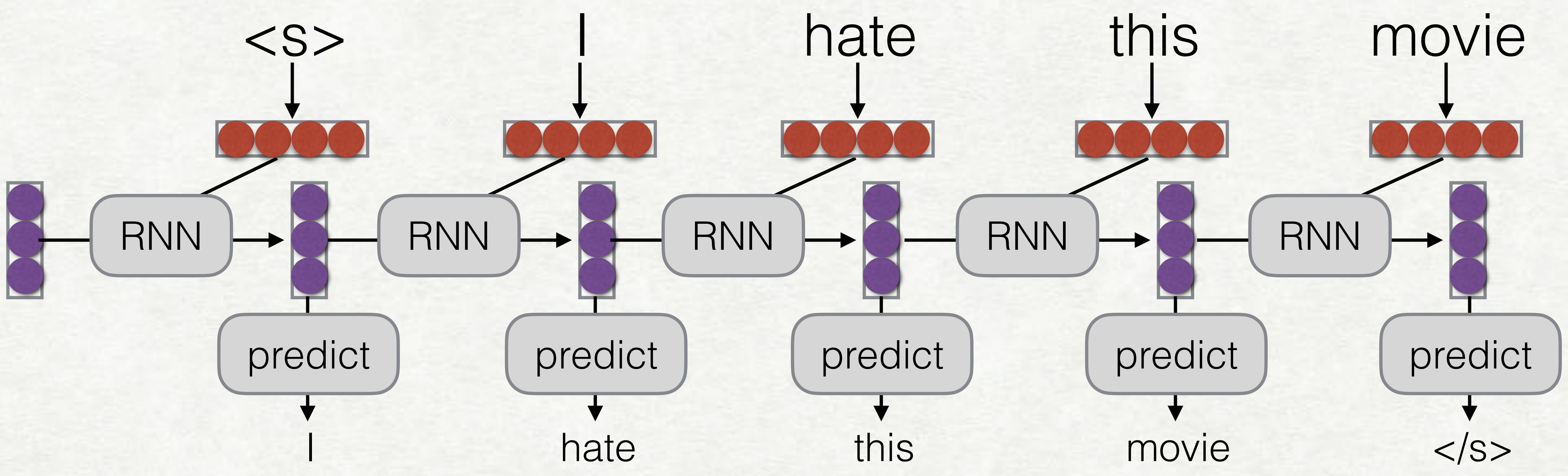
Calculating the loss



APPLICATIONS OF RNNS

E.G. LANGUAGE MODELING

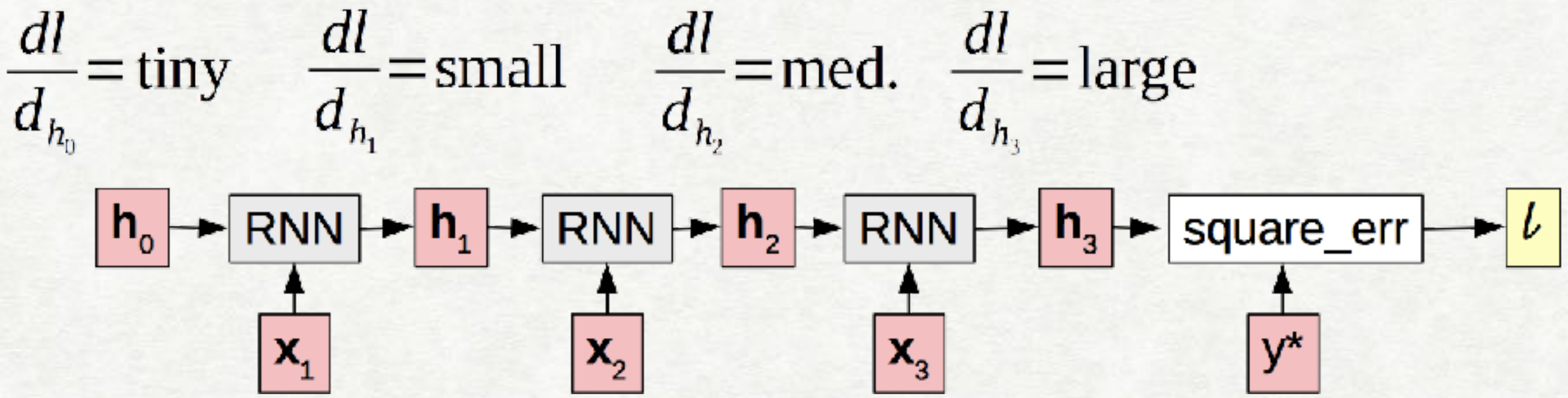
Language modeling is like a tagging task, where each tag is the next word!



VANISHING GRADIENTS

VANISHING GRADIENT

Gradients decrease as they get pushed back



Why? "Squashed" by non-linearities or small weights in matrices

A SOLUTION: LONG SHORT-TERM MEMORY
(HOCHREITER AND SCHMIDHUBER 1997)

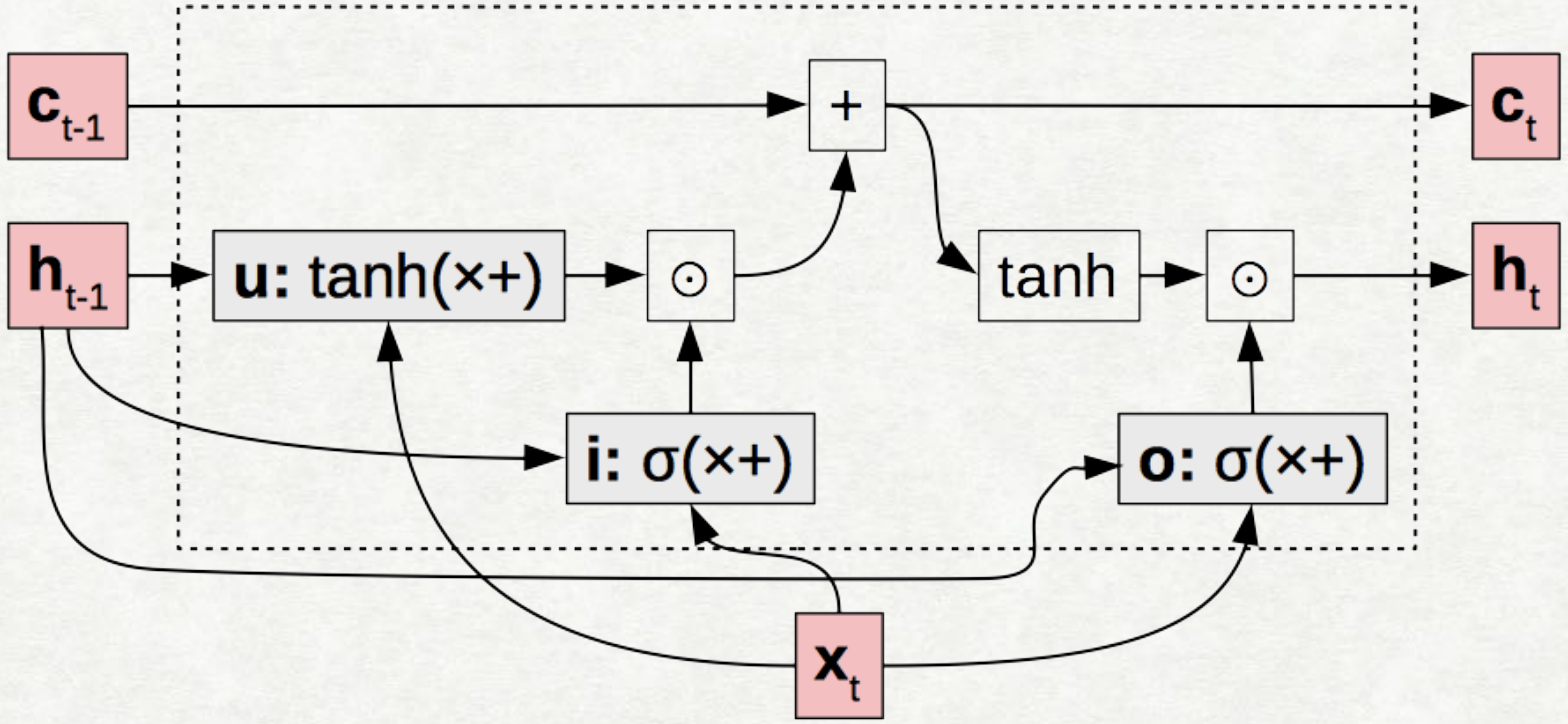
Basic idea: make additive connections between time steps

Addition does not modify the gradient, no vanishing

Gates to control the information flow

LSTM STRUCTURE

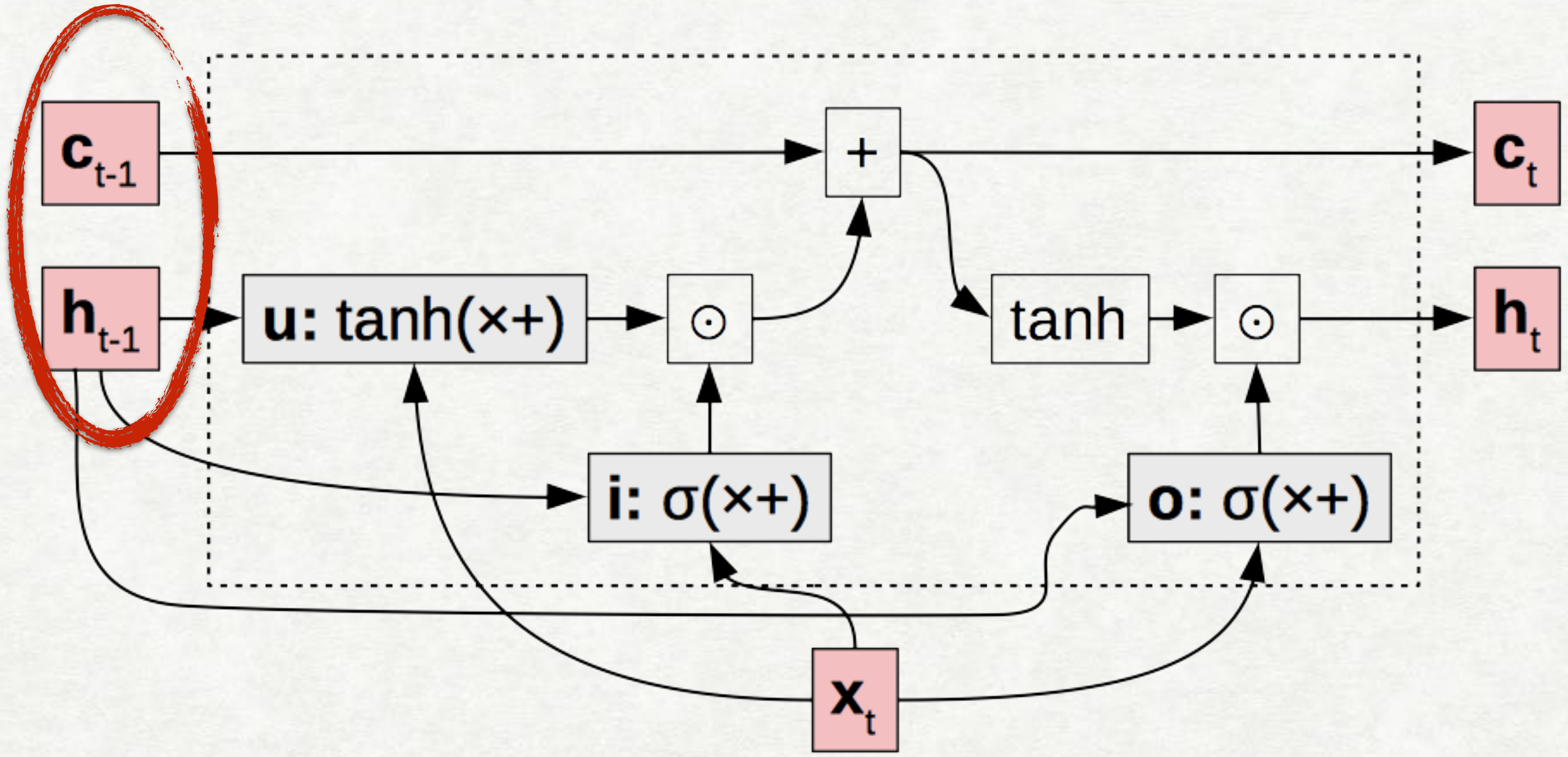
Most important idea: we want an additive connection between time steps.



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

LSTM STRUCTURE

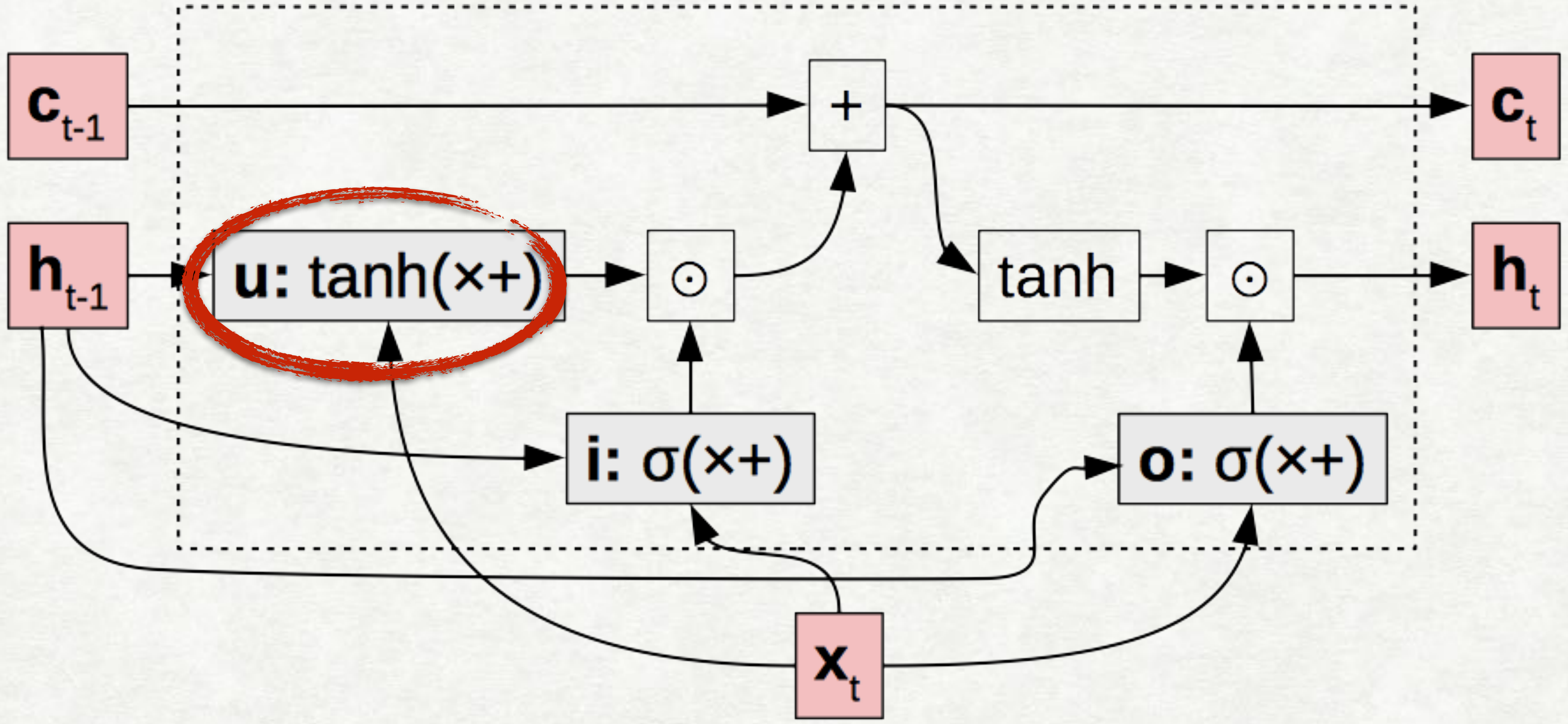
Most important idea: we want an additive connection between time steps.



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

LSTM STRUCTURE

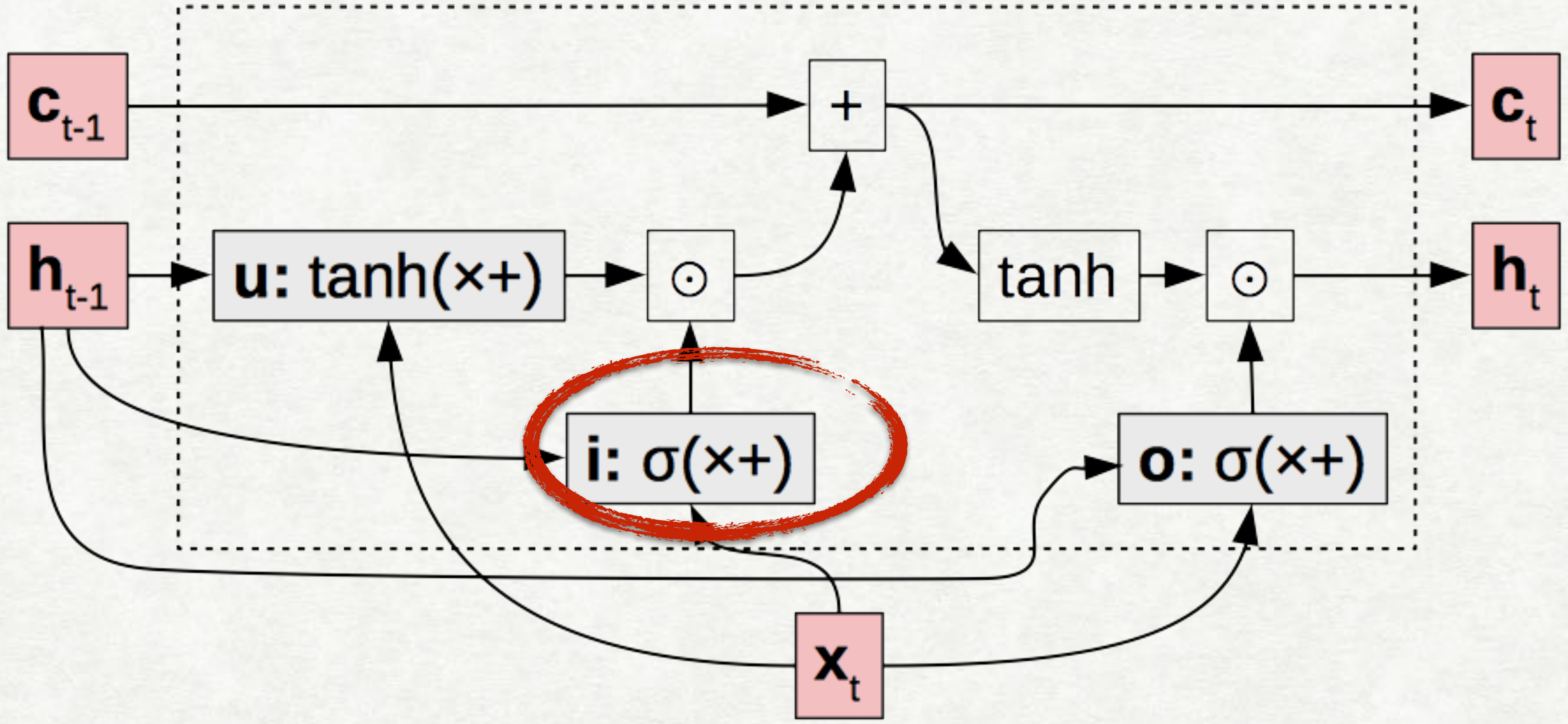
Most important idea: we want an additive connection between time steps



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

LSTM STRUCTURE

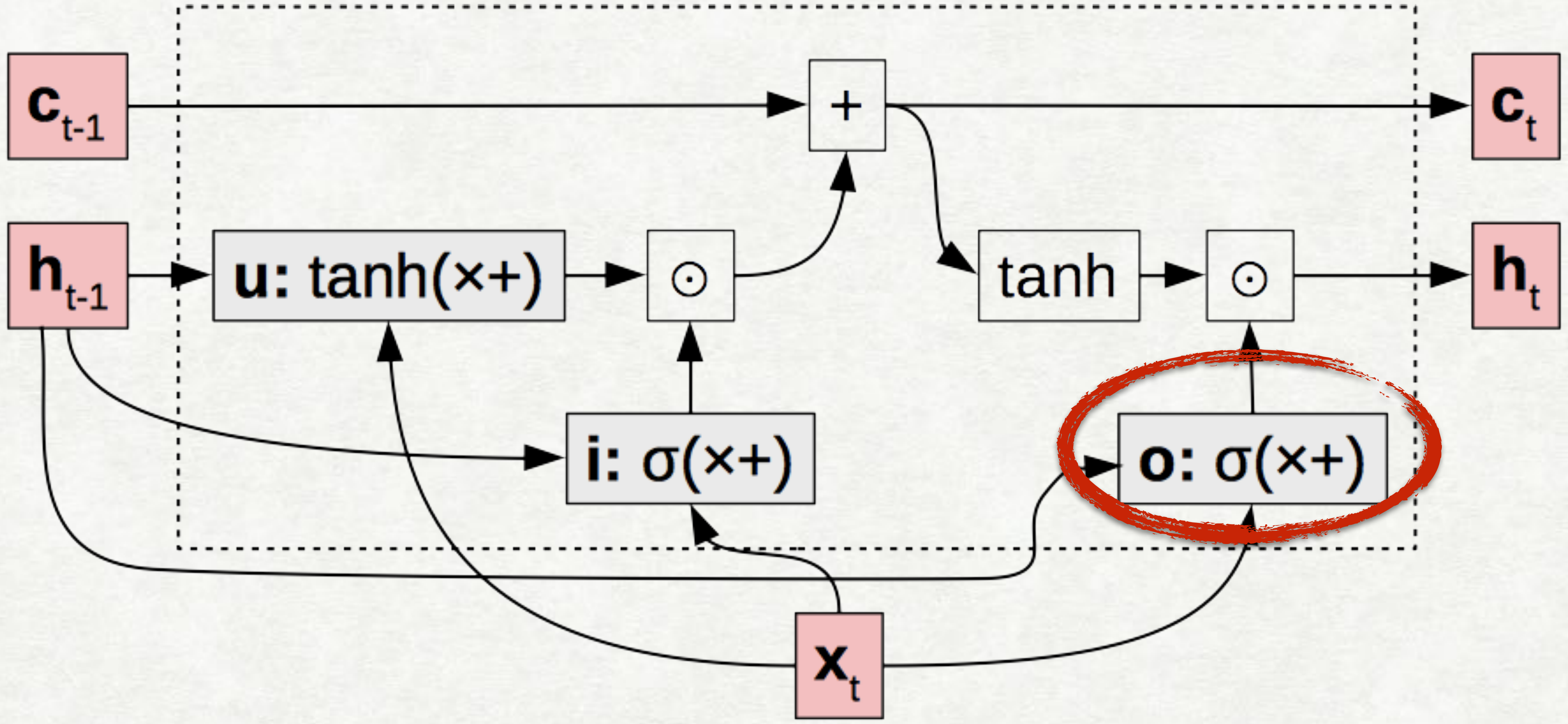
Most important idea: we want an additive connection between time steps.



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

LSTM STRUCTURE

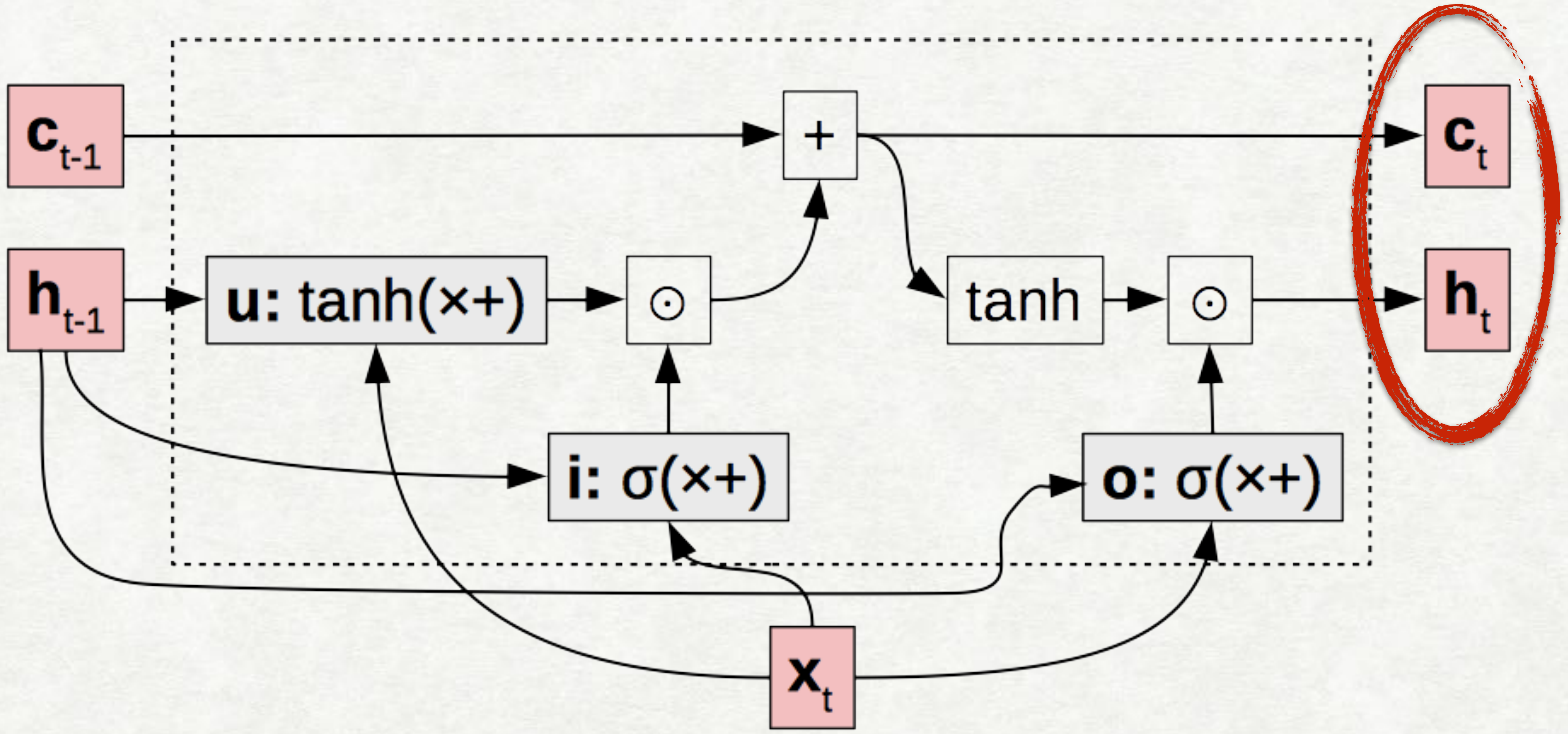
Most important idea: we want an additive connection between time steps.



update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

LSTM STRUCTURE

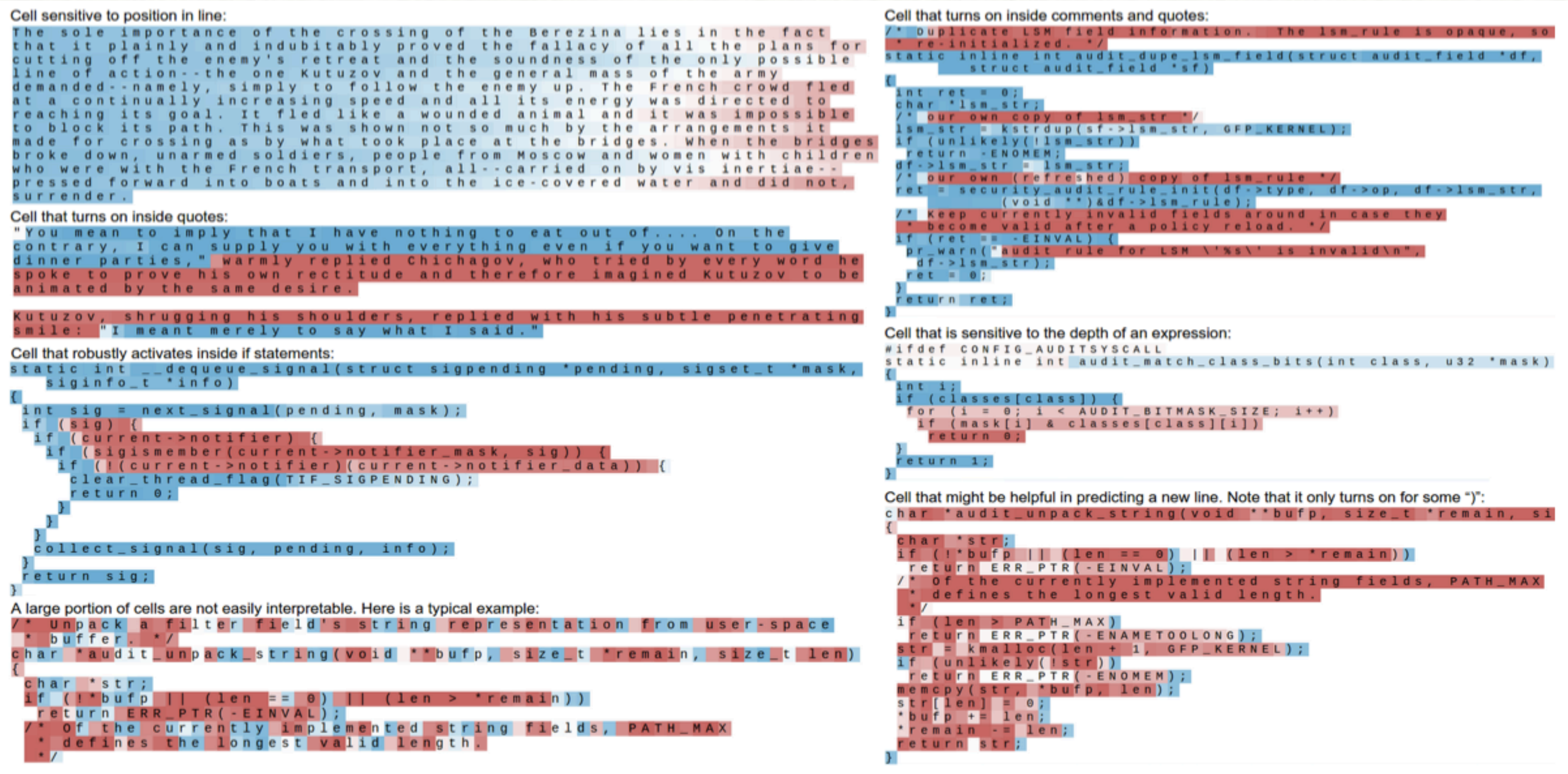
Most important idea: we want an additive connection between time steps.



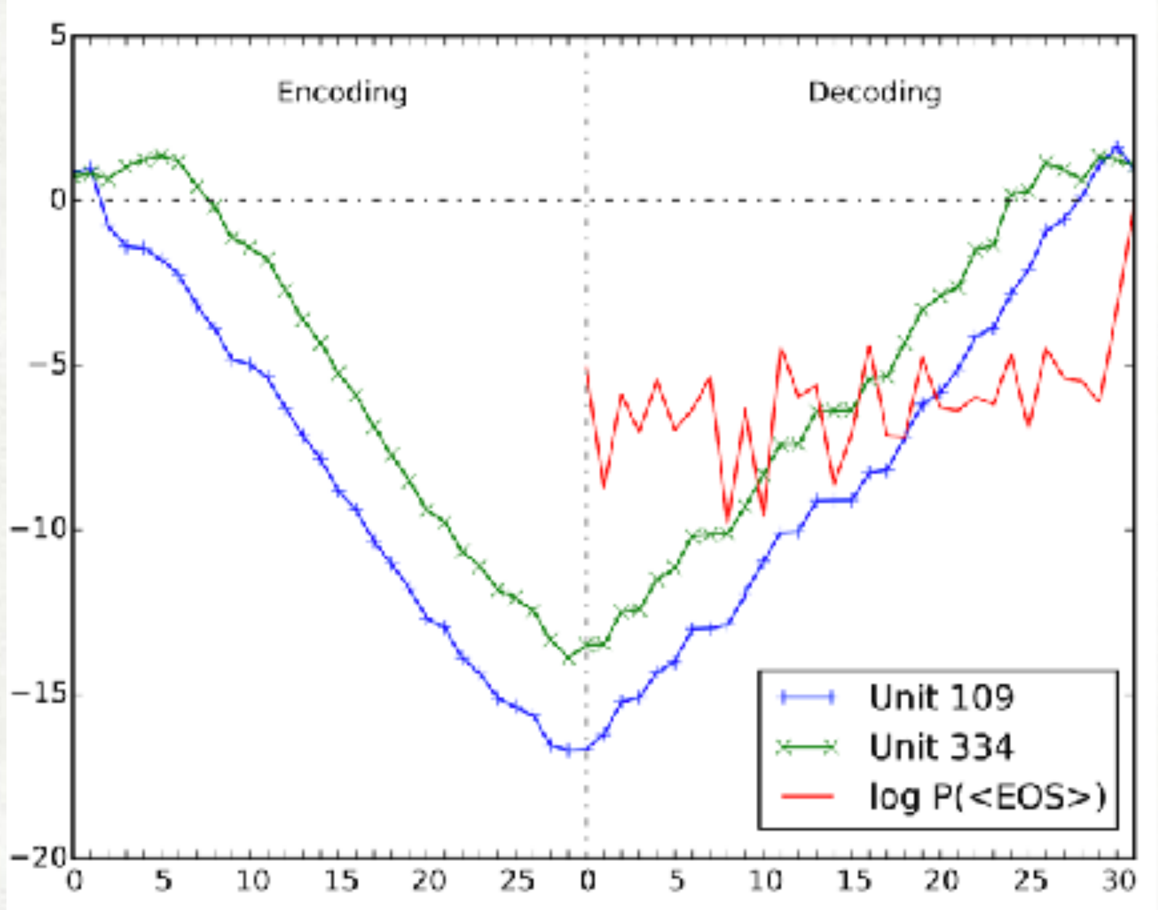
The c_t part captures long context and h_t captures the short term updates

update **u**: what value do we try to add to the memory cell?
input **i**: how much of the update do we allow to go through?
output **o**: how much of the cell do we reflect in the next state?

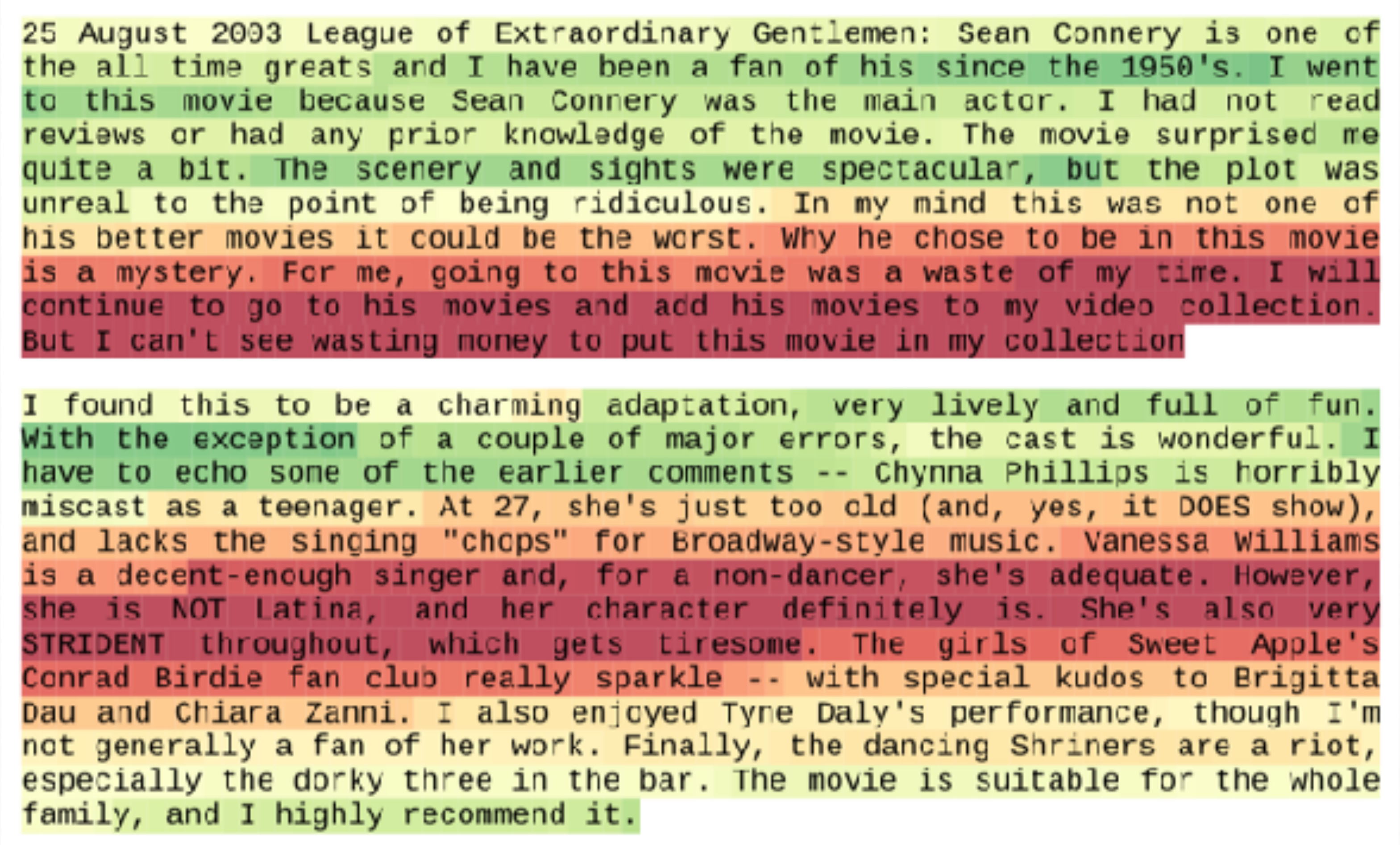
Additive connections make single nodes surprisingly interpretable



Count length of sentence



Sentiment(?)



NEXT CLASS PREVIEW

Language models produce good representations!

BERT and family