THE UNIVERSITY OF BRITISH COLUMBIA

# Topics in AI (CPSC 532L):
## Multimodal Learning with Vision, Language and Sound

Lecture 7: Word2Vec, Language Models and RNNs

# Course **Logistics**

- Assignment 1 grades (available on **Connect** ???)
  - Solutions will be posted over the  weekend
- Assignment 2 was **due Yesterday**
- Assignment 3 will be out **Friday, January 26th**
  - The due deadline will be extended

- Paper choices will be due **next week** (google form)
- **Projects** groups and short description (google form)

# Representing a **Word:** One Hot Encoding

**Vocabulary**

dog

cat

person

holding

tree

computer

using

*slide from V. Ordonex

# Representing a **Word:** One Hot Encoding

**Vocabulary**

| | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

*slide from V. Ordonex

# Representing a **Word:** One Hot Encoding

| **Vocabulary** | | **one-hot** encodings |
|---|---|---|
| dog | 1 | [ 1, 0, 0, 0, 0, 0, 0, 0, 0, 0 ] |
| cat | 2 | [ 0, 1, 0, 0, 0, 0, 0, 0, 0, 0 ] |
| person | 3 | [ 0, 0, 1, 0, 0, 0, 0, 0, 0, 0 ] |
| holding | 4 | [ 0, 0, 0, 1, 0, 0, 0, 0, 0, 0 ] |
| tree | 5 | [ 0, 0, 0, 0, 1, 0, 0, 0, 0, 0 ] |
| computer | 6 | [ 0, 0, 0, 0, 0, 1, 0, 0, 0, 0 ] |
| using | 7 | [ 0, 0, 0, 0, 0, 0, 1, 0, 0, 0 ] |

*slide from V. Ordonex

# Representing **Phrases**: Bag-of-Words

**Vocabulary**

| | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

**bag-of-words** representation

dog cat person holding tree computer using

*slide from V. Ordonex

# Representing **Phrases**: Bag-of-Words

| **Vocabulary** | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

**bag-of-words** representation

person holding dog {3, 4, 1} [ 1, 0, 1, 1, 0, 0, 0, 0, 0, 0 ]

dog cat person holding tree computer using

*slide from V. Ordonex

# Representing **Phrases**: Bag-of-Words

**Vocabulary**

| | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

**bag-of-words** representation

| | | |
|---|---|---|
| person holding dog | {3, 4, 1} | [ 1, 0, 1, 1, 0, 0, 0, 0, 0, 0 ] |
| person holding cat | {3, 4, 2} | [ 1, 1, 0, 1, 0, 0, 0, 0, 0, 0 ] |

dog cat person holding tree computer using

*slide from V. Ordonex

# Representing **Phrases**: Bag-of-Words

| | | **bag-of-words** representation |
|---|---|---|
| person holding dog | {3, 4, 1} | [ 1, 0, 1, 1, 0, 0, 0, 0, 0, 0 ] |
| person holding cat | {3, 4, 2} | [ 1, 1, 0, 1, 0, 0, 0, 0, 0, 0 ] |
| person using computer | {3, 7, 6} | [ 0, 0, 0, 1, 0, 1, 1, 0, 0, 0 ] |

dog cat person holding tree computer using

**Vocabulary**

| | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

*slide from V. Ordonex

# Representing **Phrases**: Bag-of-Words

| **Vocabulary** | |
|---|---|
| dog | 1 |
| cat | 2 |
| person | 3 |
| holding | 4 |
| tree | 5 |
| computer | 6 |
| using | 7 |

**bag-of-words** representation

| | | |
|---|---|---|
| person holding dog | {3, 4, 1} | [ 1, 0, 1, 1, 0, 0, 0, 0, 0, 0 ] |
| person holding cat | {3, 4, 2} | [ 1, 1, 0, 1, 0, 0, 0, 0, 0, 0 ] |
| person using computer | {3, 7, 6} | [ 0, 0, 0, 1, 0, 1, 1, 0, 0, 0 ] |

dog cat person holding tree computer using

| | | |
|---|---|---|
| person using computer<br>person holding cat | {3, 3, 7, 6, 2} | [ 0, 1, 2, 1, 0, 1, 1, 0, 0, 0 ] |

*slide from V. Ordonex

# **Distributional** Hypothesis [ Lenci, 2008 ]

— At least certain aspects of the meaning of lexical expressions depend on their distributional properties in the linguistic contexts

— The degree of semantic similarity between two linguistic expressions is a function of the similarity of the two linguistic contexts in which they can appear

# What is the meaning of “**bardiwac**”?

— He handed her glass of **bardiwac**.

— Beef dishes are made to complement the **bardiwacs**.

— Nigel staggered to his feet, face flushed from too much **bardiwac**.

— Malbec, one of the lesser-known **bardiwac** grapes, responds well to Australia’s sunshine.

— I dined off bread and cheese and this excellent **bardiwac**.

—The drinks were delicious: blood-red **bardiwac** as well as light, sweet Rhenish.

* Adopted from slides by Louis-Philippe Morency

# What is the meaning of “**bardiwac**”?

— He handed her glass of **bardiwac**.

— Beef dishes are made to complement the **bardiwacs**.

— Nigel staggered to his feet, face flushed from too much **bardiwac**.

— Malbec, one of the lesser-known **bardiwac** grapes, responds well to Australia’s sunshine.

— I dined off bread and cheese and this excellent **bardiwac**.

—The drinks were delicious: blood-red **bardiwac** as well as light, sweet Rhenish.

**bardic** is an alcoholic beverage made from grapes

* Adopted from slides by Louis-Philippe Morency

# **Geometric Interpretation**: Co-occurrence as feature

— Row vector describes usage of word in a corpus of text

— Can be seen as coordinates o the point in an n-dimensional Euclidian space

| | get | see | use | hear | eat | kill |
|---|---|---|---|---|---|---|
| knife | 51 | 20 | 84 | 0 | 3 | 0 |
| cat | 52 | 58 | 4 | 4 | 6 | 26 |
| dog | 115 | 83 | 10 | 42 | 33 | 17 |
| boat | 59 | 39 | 23 | 4 | 0 | 0 |
| cup | 98 | 14 | 6 | 2 | 1 | 0 |
| pig | 12 | 17 | 3 | 2 | 9 | 27 |
| banana | 11 | 2 | 2 | 0 | 18 | 0 |

**Co-occurrence** Matrix

* Slides from Louis-Philippe Morency

# **Geometric Interpretation**: Co-occurrence as feature

— Row vector describes usage of word in a corpus of text

— Can be seen as coordinates o the point in an n-dimensional Euclidian space

| | get | see | use | hear | eat | kill |
|---|---|---|---|---|---|---|
| knife | 51 | 20 | 84 | 0 | 3 | 0 |
| cat | 52 | 58 | 4 | 4 | 6 | 26 |
| **dog** | 115 | 83 | 10 | 42 | 33 | 17 |
| boat | 59 | 39 | 23 | 4 | 0 | 0 |
| cup | 98 | 14 | 6 | 2 | 1 | 0 |
| pig | 12 | 17 | 3 | 2 | 9 | 27 |
| banana | 11 | 2 | 2 | 0 | 18 | 0 |

**Co-occurrence** Matrix

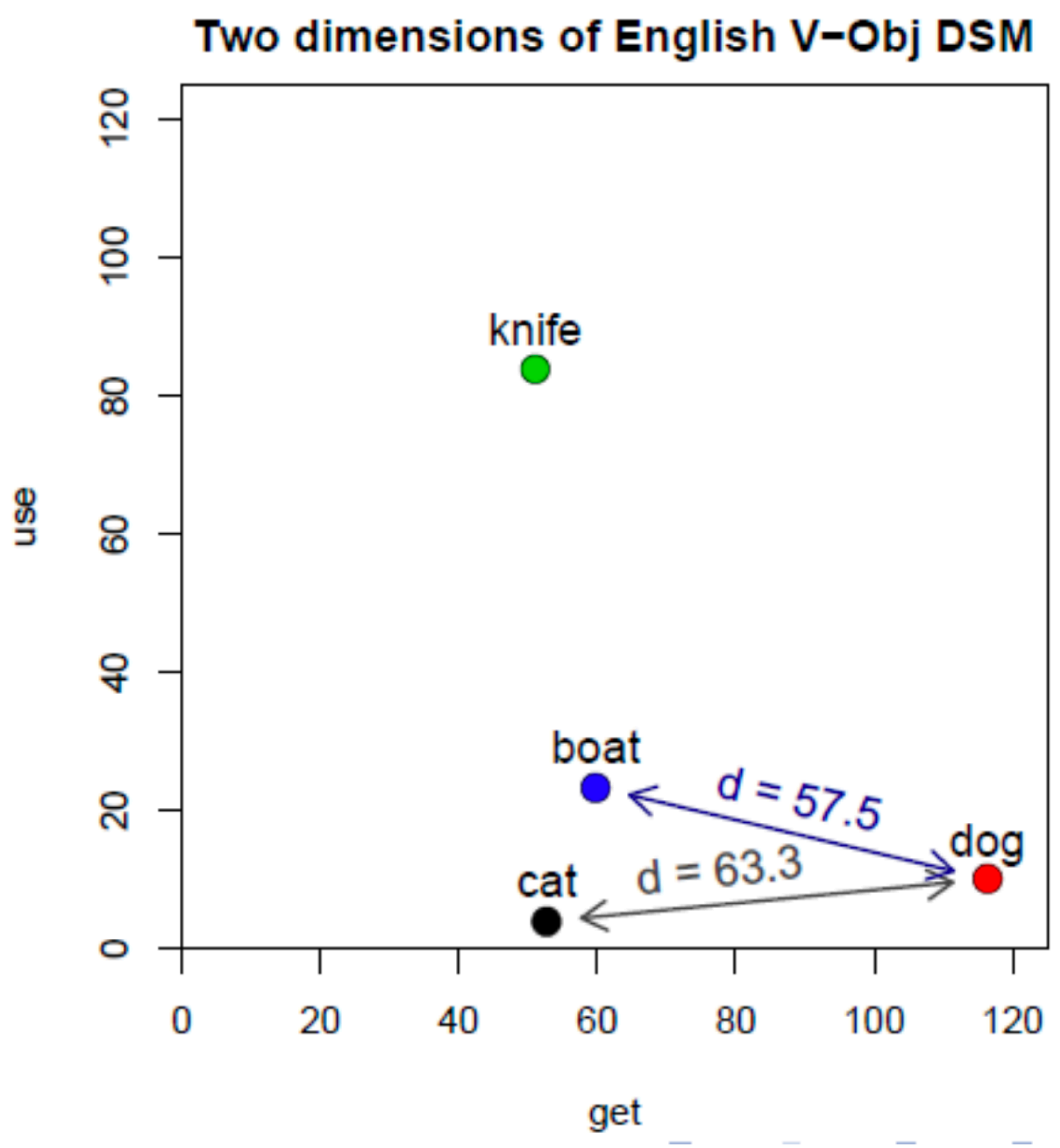
* Slides from Louis-Philippe Morency

# **Distance** and Similarity

— Illustrated in two dimensions

— Similarity = spatial proximity (Euclidian distance)

— Location depends on frequency of noun (dog is 27 times as frequent as ca)



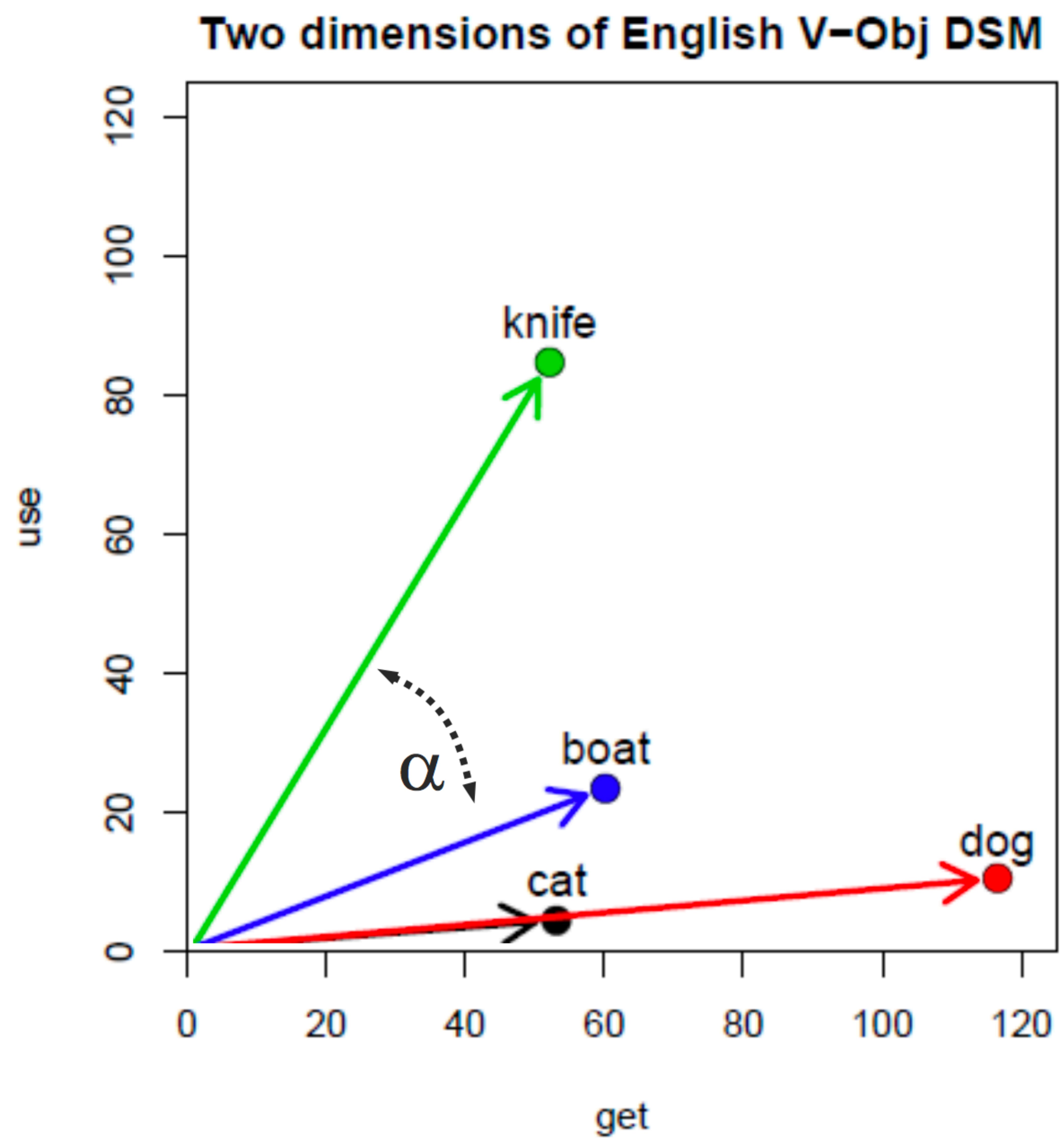

* Slides from Louis-Philippe Morency

# **Angle** and Similarity

— direction is more important than location

— normalize length of vectors

— or use angle as a distance measure




* Slides from Louis-Philippe Morency

# **Geometric Interpretation**: Co-occurrence as feature

— Row vector describes usage of word in a corpus of text

— Can be seen as coordinates of the point in an n-dimensional Euclidian space

| | get | see | use | hear | eat | kill |
|---|---|---|---|---|---|---|
| knife | 51 | 20 | 84 | 0 | 3 | 0 |
| cat | 52 | 58 | 4 | 4 | 6 | 26 |
| dog | 115 | 83 | 10 | 42 | 33 | 17 |
| boat | 59 | 39 | 23 | 4 | 0 | 0 |
| cup | 98 | 14 | 6 | 2 | 1 | 0 |
| pig | 12 | 17 | 3 | 2 | 9 | 27 |
| banana | 11 | 2 | 2 | 0 | 18 | 0 |

**Co-occurrence** Matrix

* Slides from Louis-Philippe Morency

# **Geometric Interpretation**: Co-occurrence as feature

— Row vector describes usage of word in a corpus of text

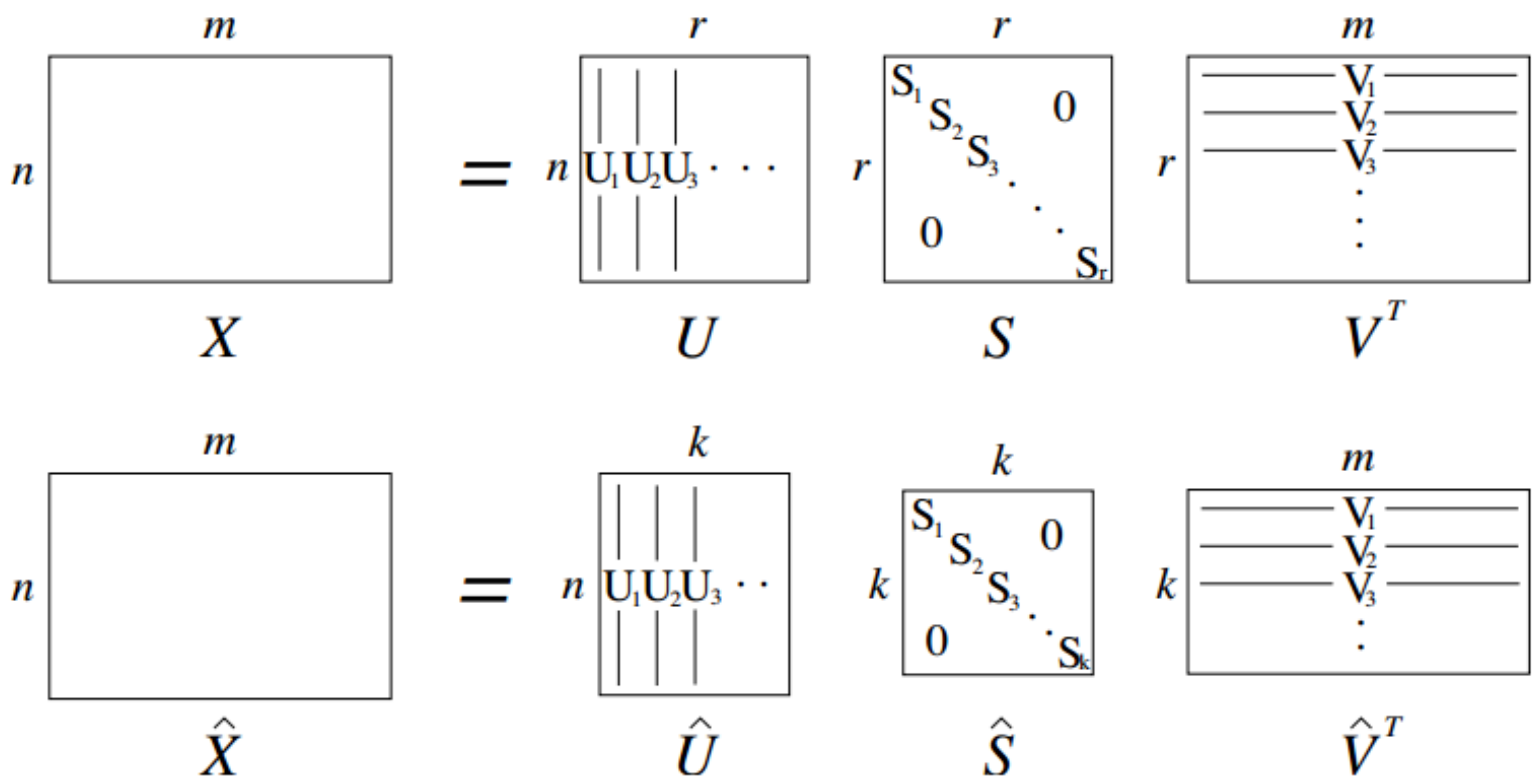
— Can be seen as coordinates of the point in an n-dimensional Euclidian space

**Way too high dimensional!**

| | get | see | use | hear | eat | kill |
|---|---|---|---|---|---|---|
| knife | 51 | 20 | 84 | 0 | 3 | 0 |
| cat | 52 | 58 | 4 | 4 | 6 | 26 |
| dog | 115 | 83 | 10 | 42 | 33 | 17 |
| boat | 59 | 39 | 23 | 4 | 0 | 0 |
| cup | 98 | 14 | 6 | 2 | 1 | 0 |
| pig | 12 | 17 | 3 | 2 | 9 | 27 |
| banana | 11 | 2 | 2 | 0 | 18 | 0 |

**Co-occurrence** Matrix

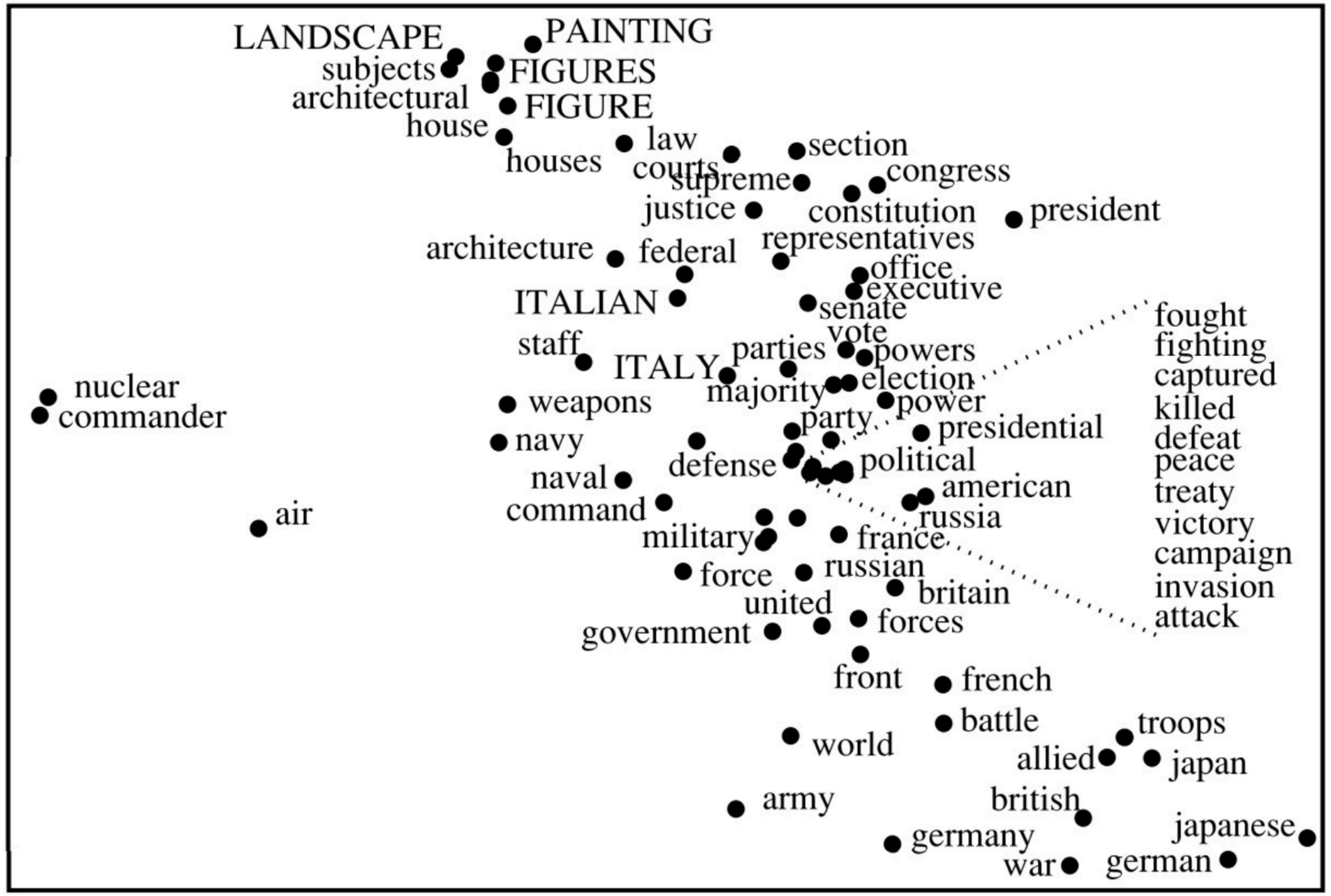
# **SVD** for Dimensionality Reduction




*slide from Vagelis Hristidis

# **Learned** Word Vector Visualization

We can also use other methods, like LLE here:




Nonlinear dimensionality reduction by locally linear embedding. Sam Roweis & Lawrence Saul. Science, v.290,2000

[ Roweis and Saul, 2000 ]

# Issues with **SVD**

**Computational** cost for a $d \times n$ matrix is $\mathcal{O}(dn^2)$, where $d < n$

— Makes it not possible for large number of word vocabularies or documents

It is hard to incorporate out of sample (**new**) words or documents

*slide from Vagelis Hristidis

# **word2vec**: Representing the Meaning of Words [ Mikolov et al., 2013 ]

**Key idea:** Predict surrounding words of every word

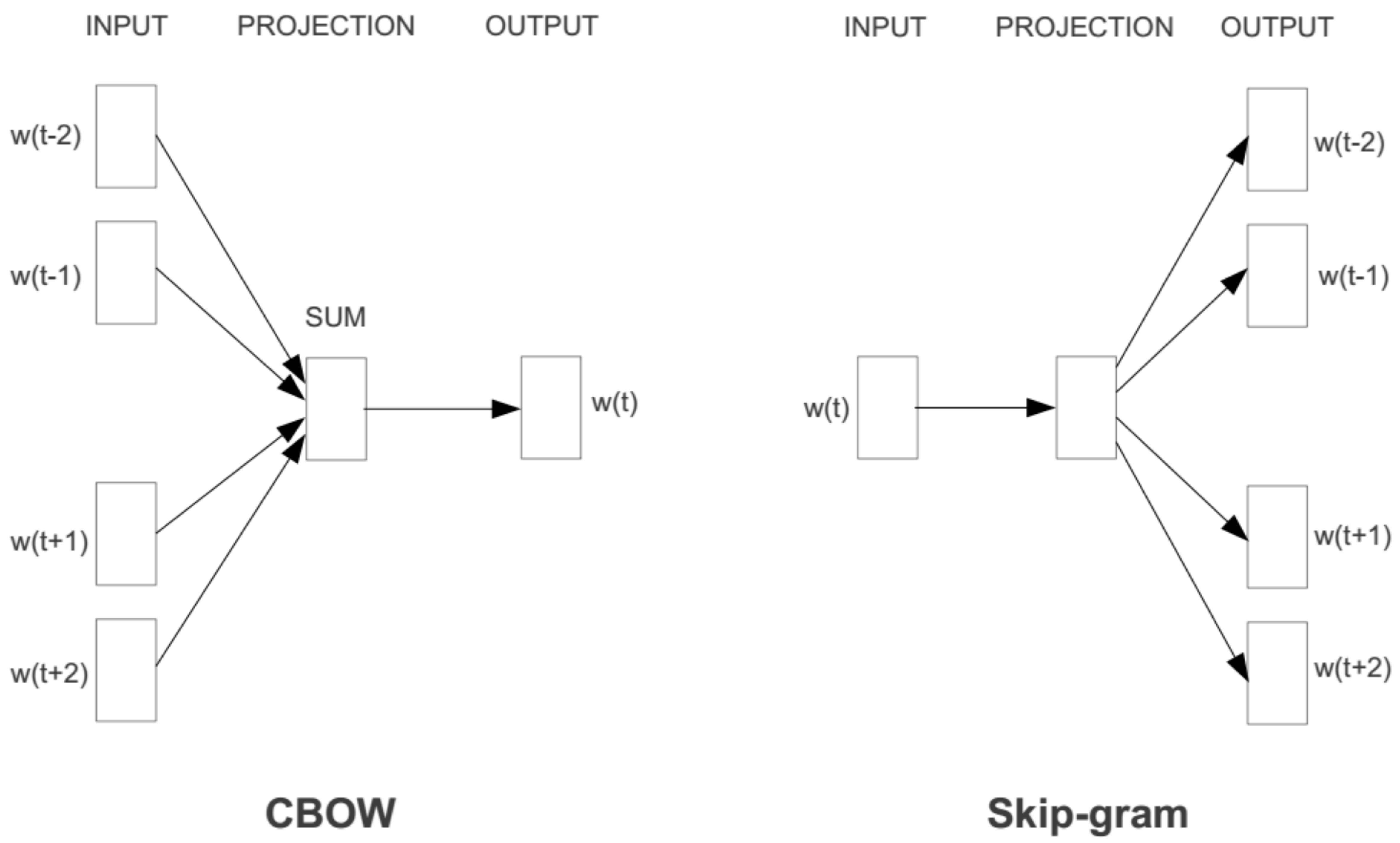
**Benefits:** Faster and easier to incorporate new document, words, etc.

*slide from Vagelis Hristidis

# **word2vec**: Representing the Meaning of Words [ Mikolov et al., 2013 ]

**Key idea:** Predict surrounding words of every word

**Benefits:** Faster and easier to incorporate new document, words, etc.




Continuous Bag of Words (**CBOW**): use context words in a window to predict middle word

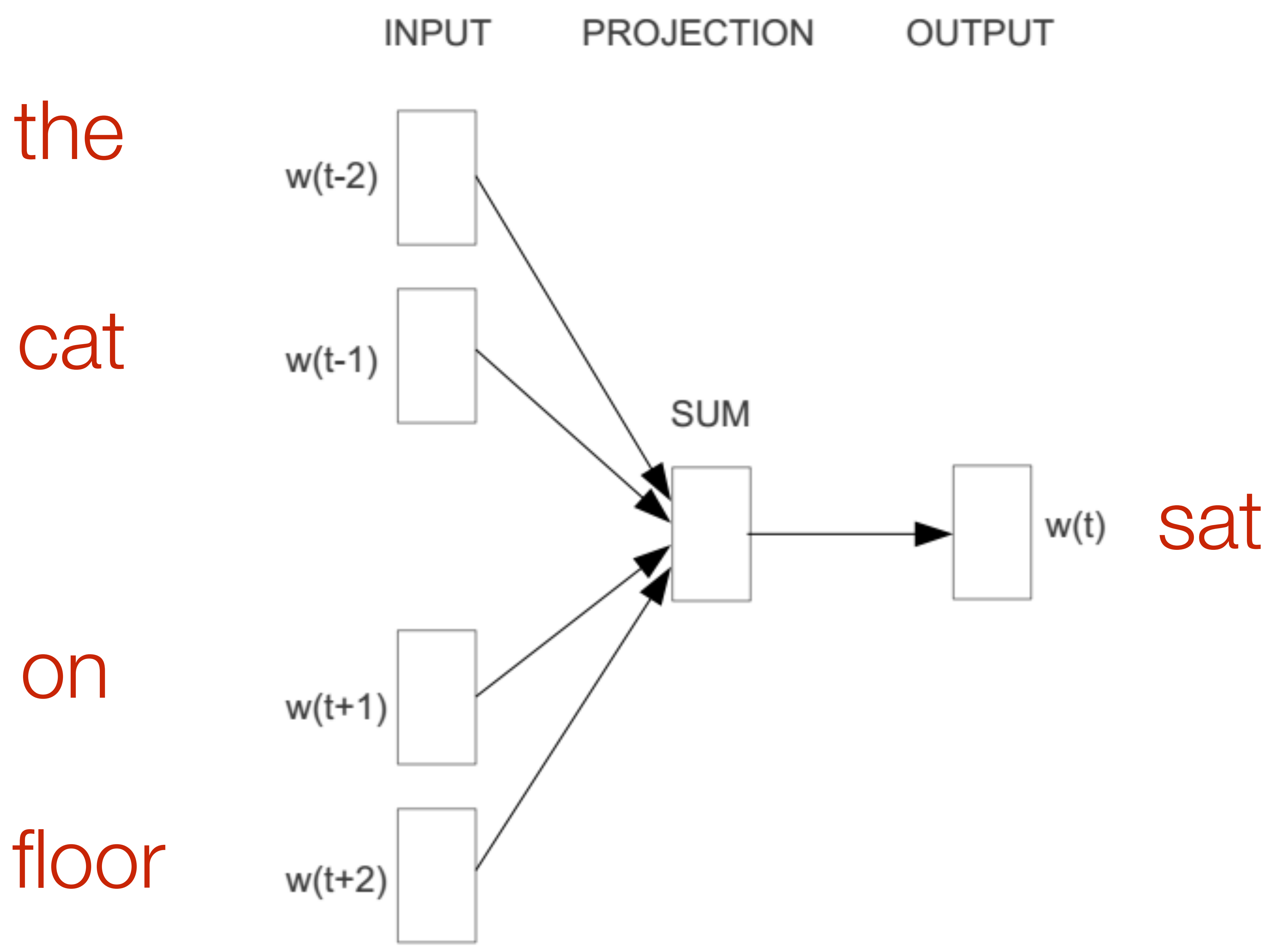
**Skip-gram:** use the middle word to predict surrounding ones in a window

*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

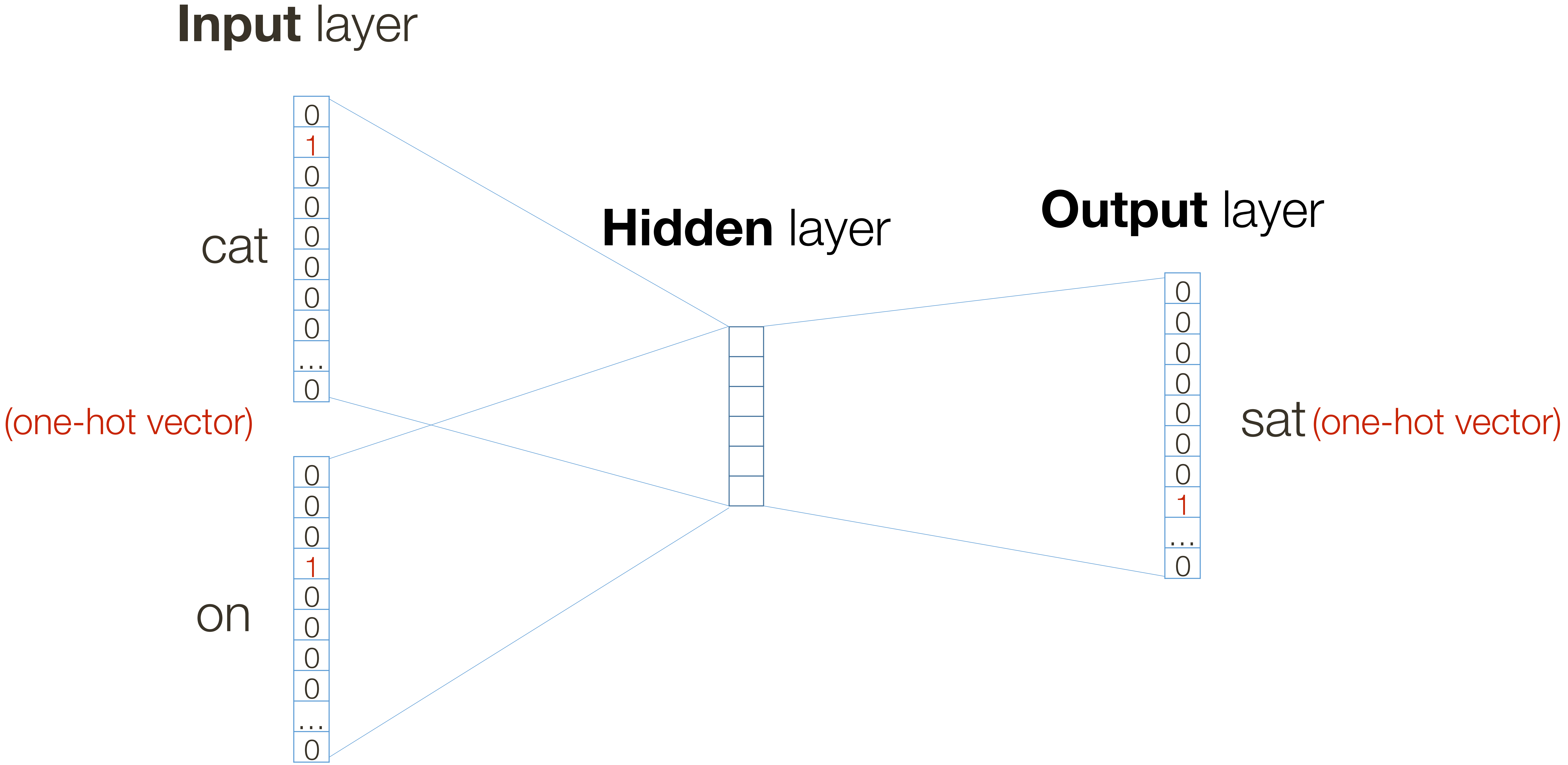
**Example:** "The cat sat on floor" (window size 2)




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

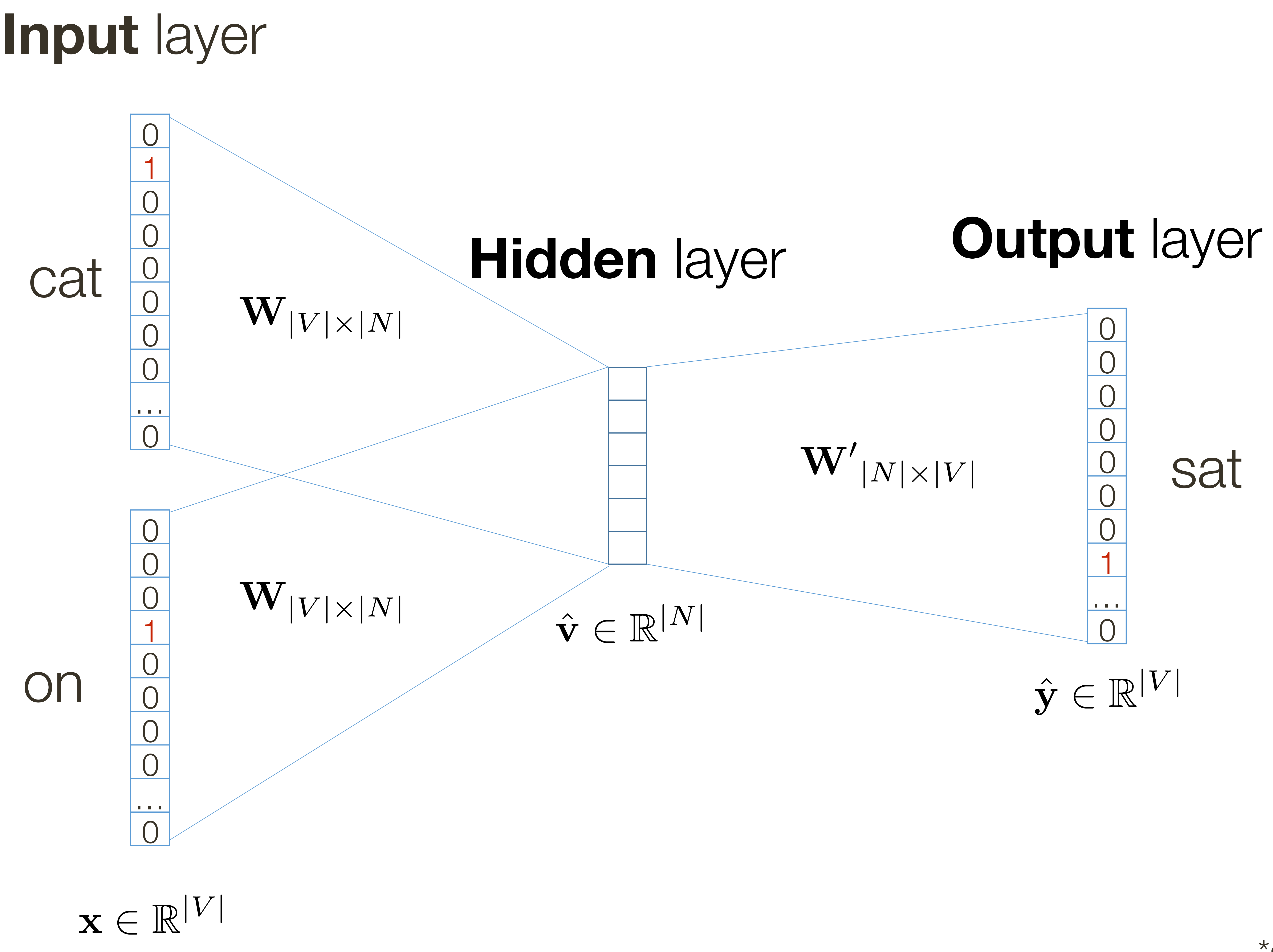
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

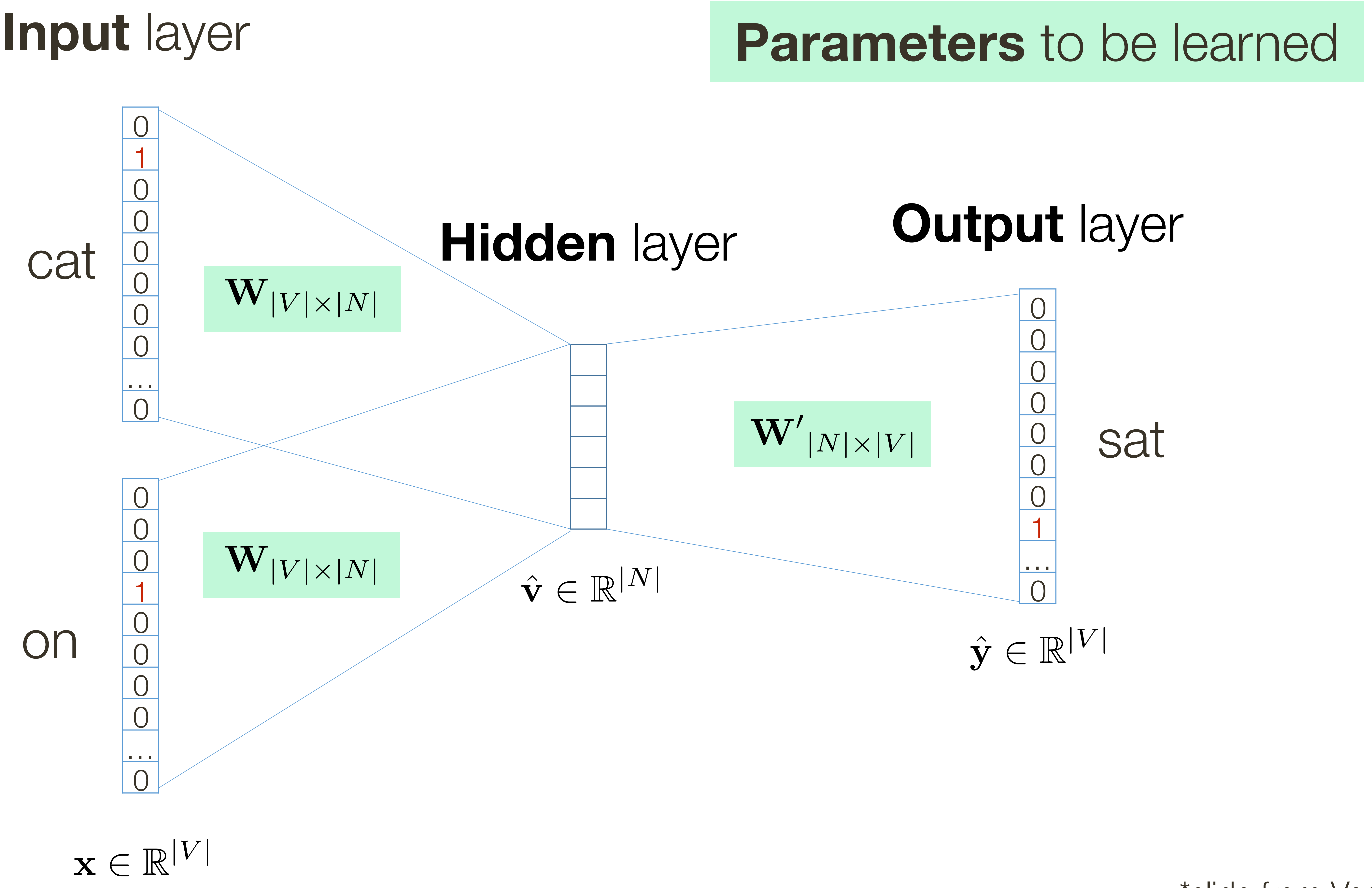
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

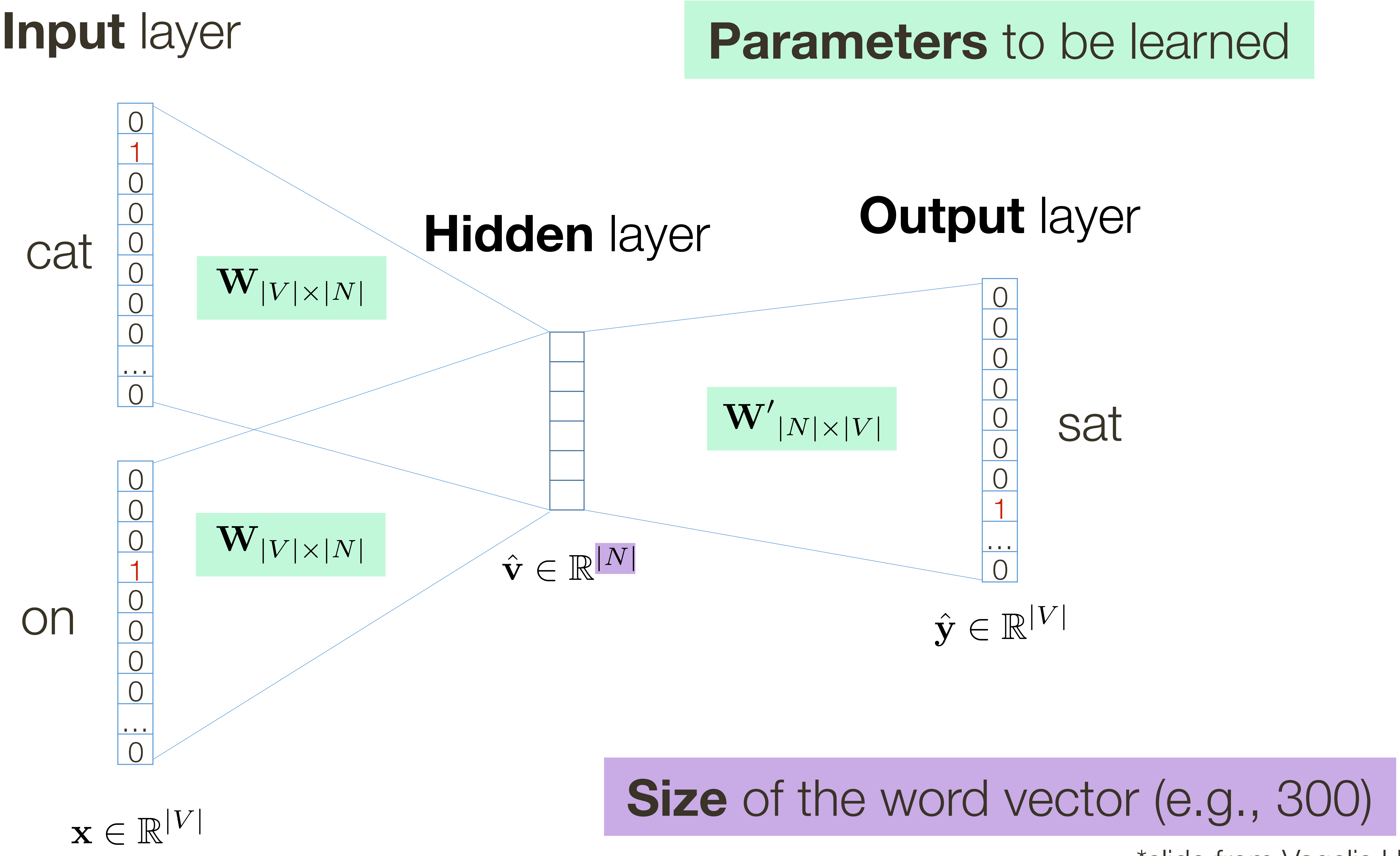
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

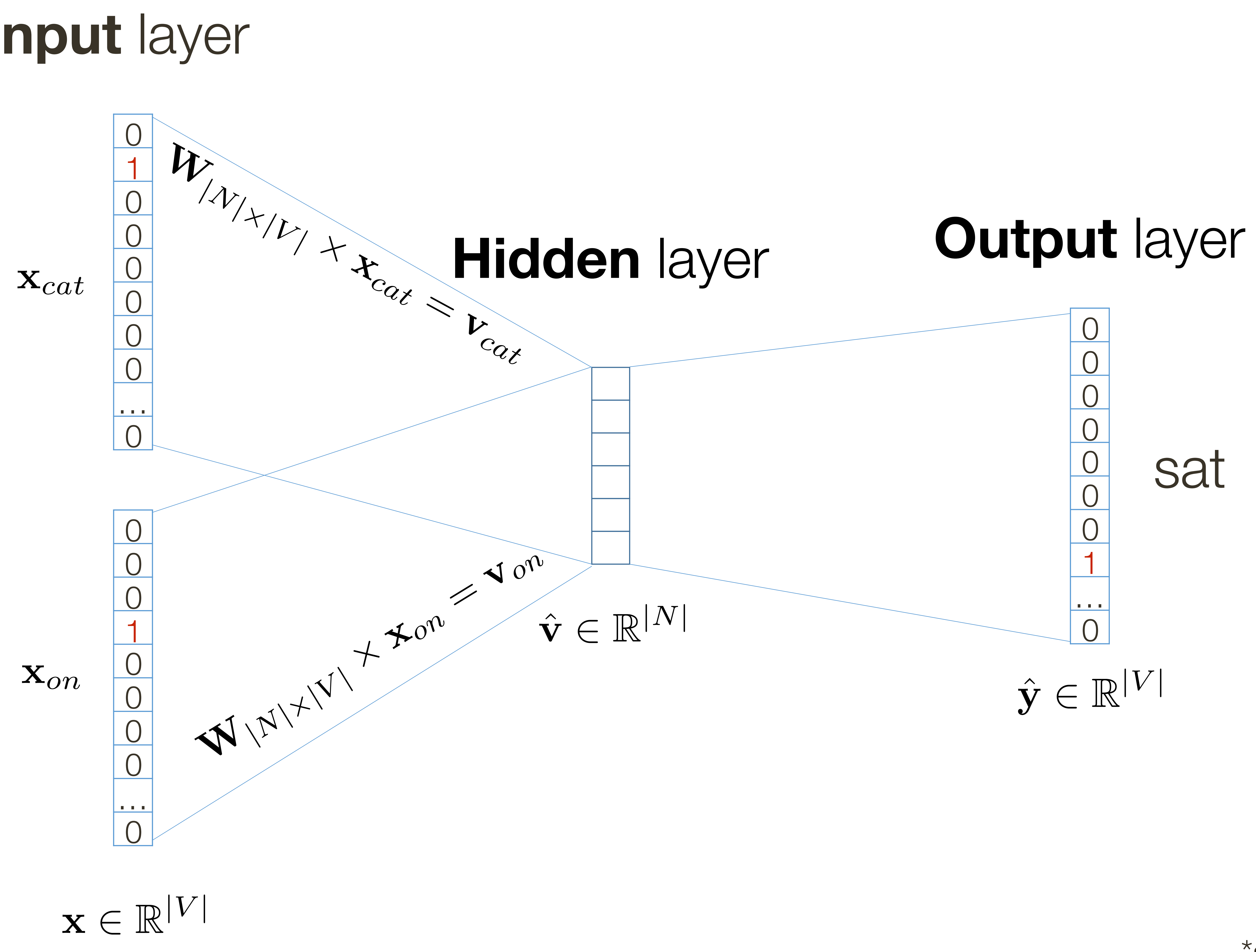
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

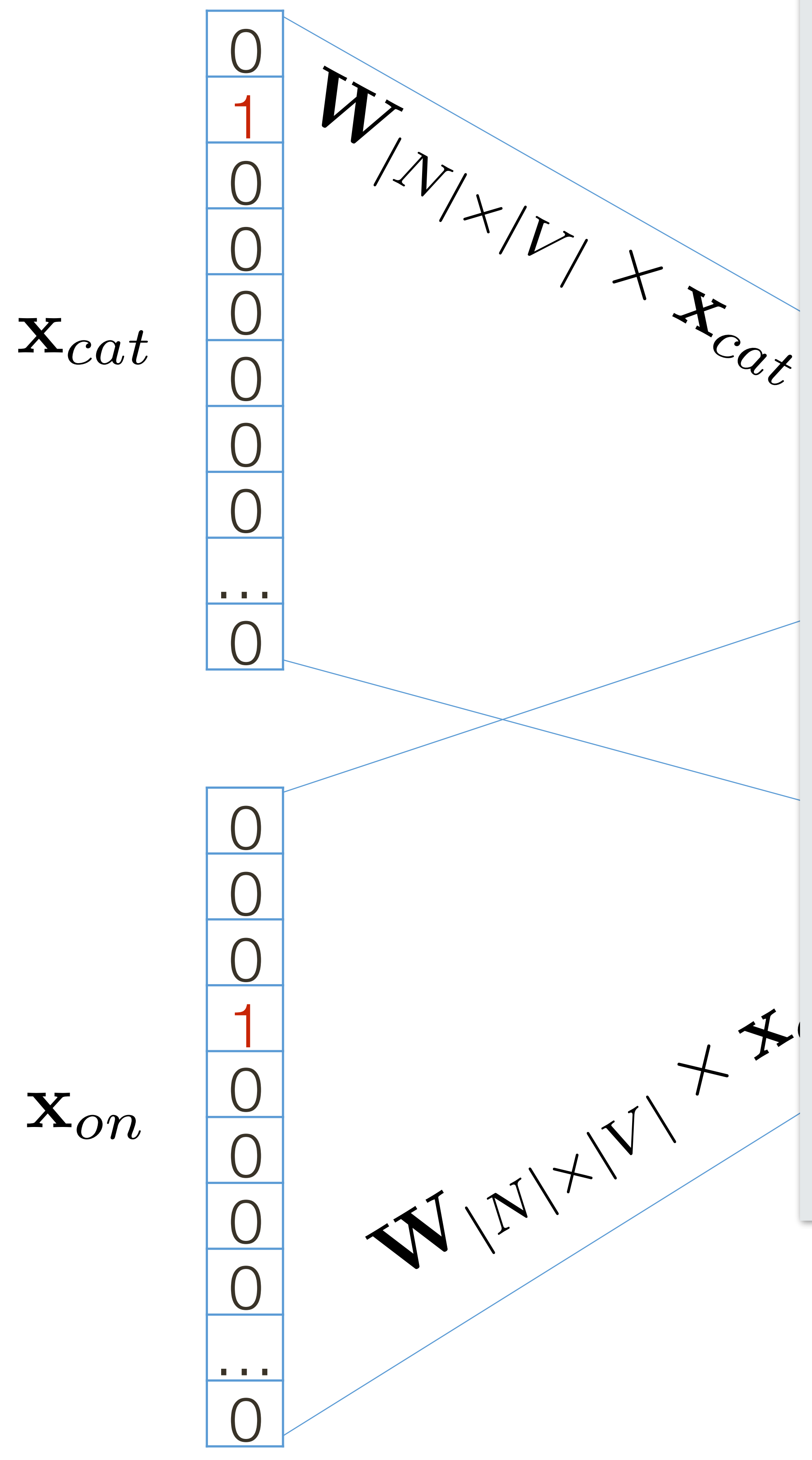



*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

**Input** layer




$\mathbf{x} \in \mathbb{R}^{|V|}$

$$\mathbf{W}^T_{|V|\times|N|} \times \mathbf{x}_{cat} = \mathbf{v}_{cat}$$

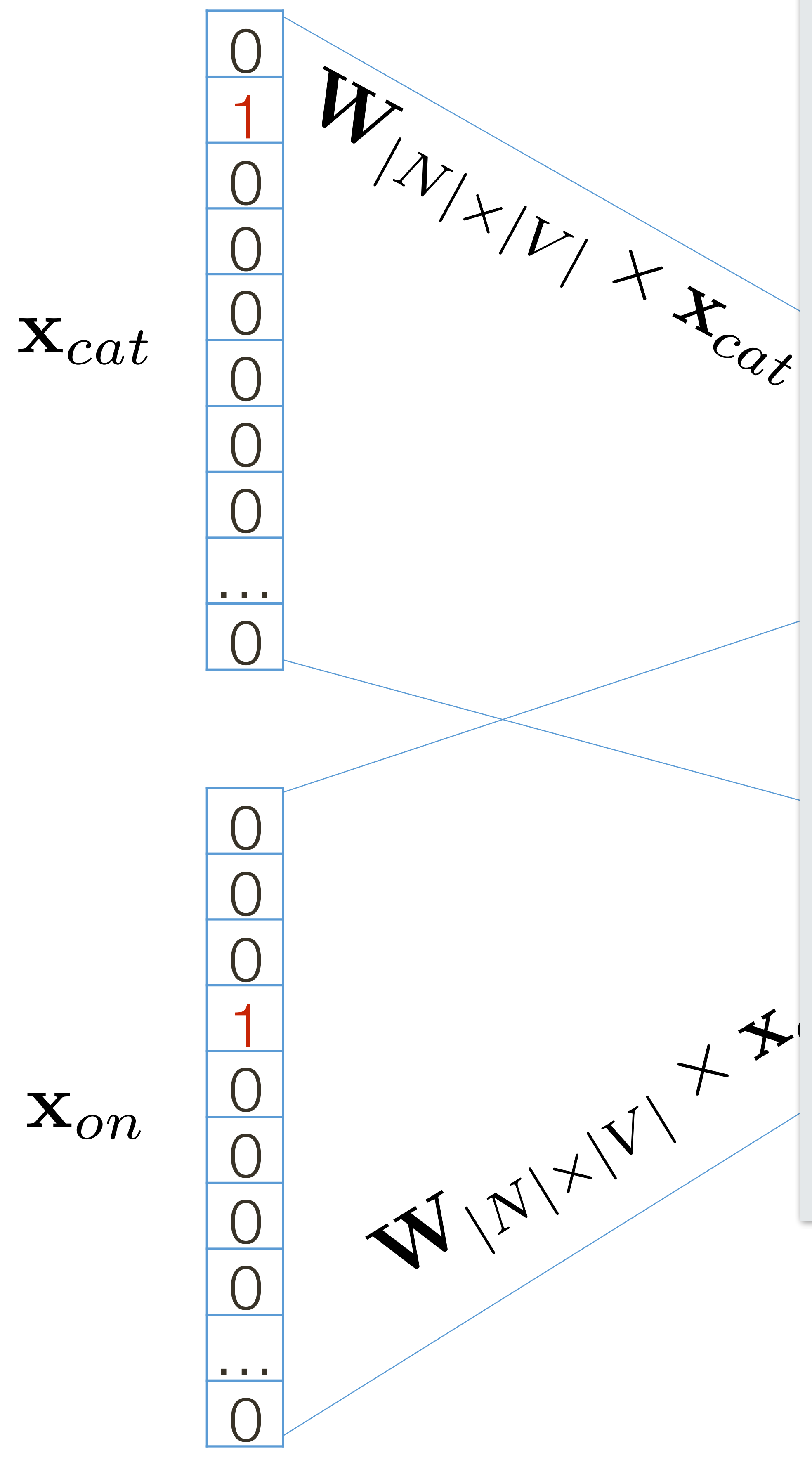
$$\begin{bmatrix} 0.1 & 2.4 & 1.6 & 1.8 & 0.5 & 0.9 & \dots & \dots & \dots & 3.2 \\ 0.5 & 2.6 & 1.4 & 2.9 & 1.5 & 3.6 & \dots & \dots & \dots & 6.1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.6 & 1.8 & 2.7 & 1.9 & 2.4 & 2.0 & \dots & \dots & \dots & 1.2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} 2.4 \\ 2.6 \\ \dots \\ \dots \\ 1.8 \end{bmatrix}$$

*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

## **Input** layer




$$\mathbf{x} \in \mathbb{R}^{|V|}$$

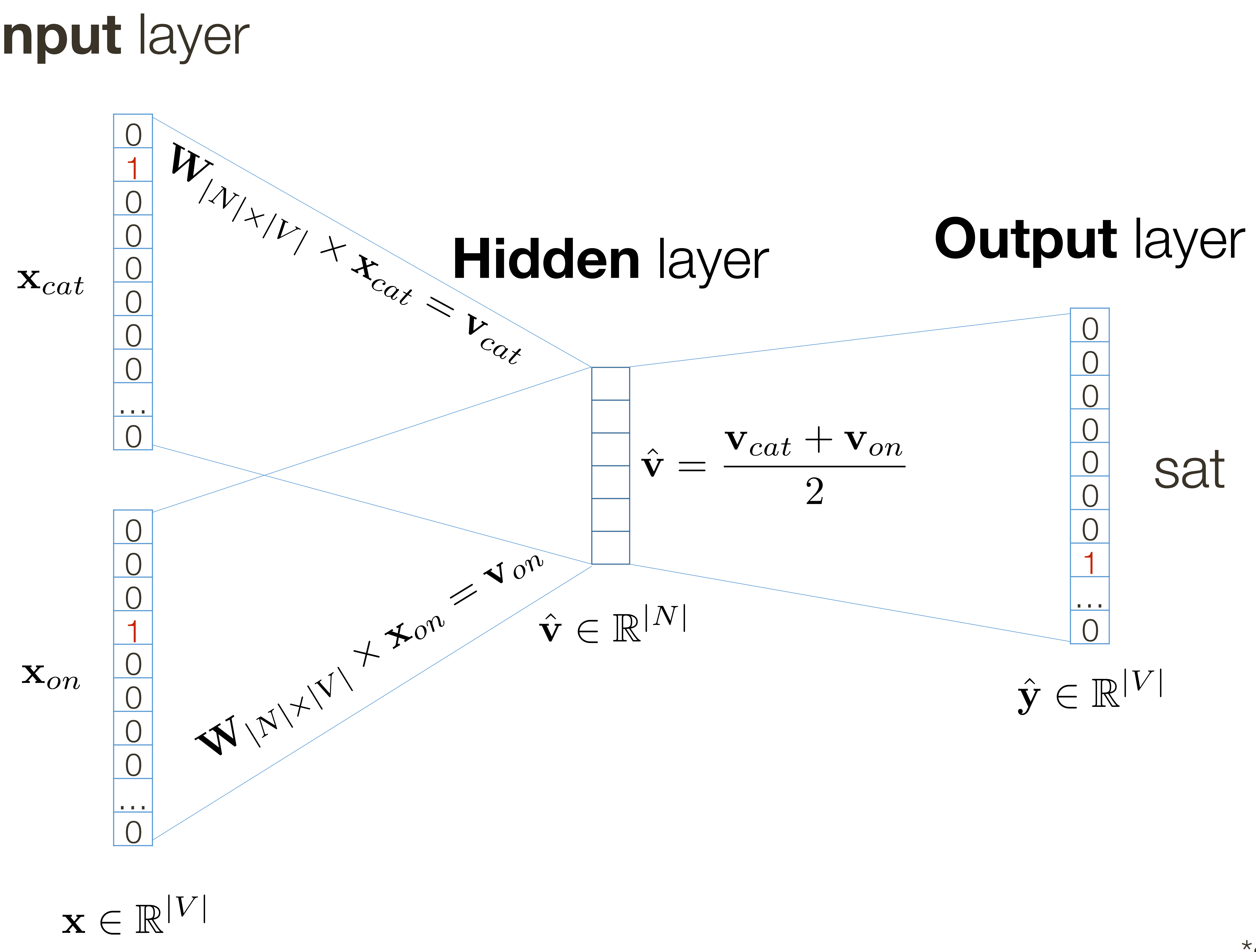
$$\mathbf{W}^{T}_{|V|\times|N|} \times \mathbf{x}_{on} = \mathbf{v}_{on}$$

$$\begin{bmatrix} 0.1 & 2.4 & 1.6 & 1.8 & 0.5 & 0.9 & \dots & \dots & \dots & 3.2 \\ 0.5 & 2.6 & 1.4 & 2.9 & 1.5 & 3.6 & \dots & \dots & \dots & 6.1 \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots & \dots \\ 0.6 & 1.8 & 2.7 & 1.9 & 2.4 & 2.0 & \dots & \dots & \dots & 1.2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0 \\ 0 \\ 1 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \\ 0 \end{bmatrix} = \begin{bmatrix} \mathbf{1.8} \\ \mathbf{2.9} \\ \dots \\ \dots \\ \mathbf{1.9} \end{bmatrix}$$

*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

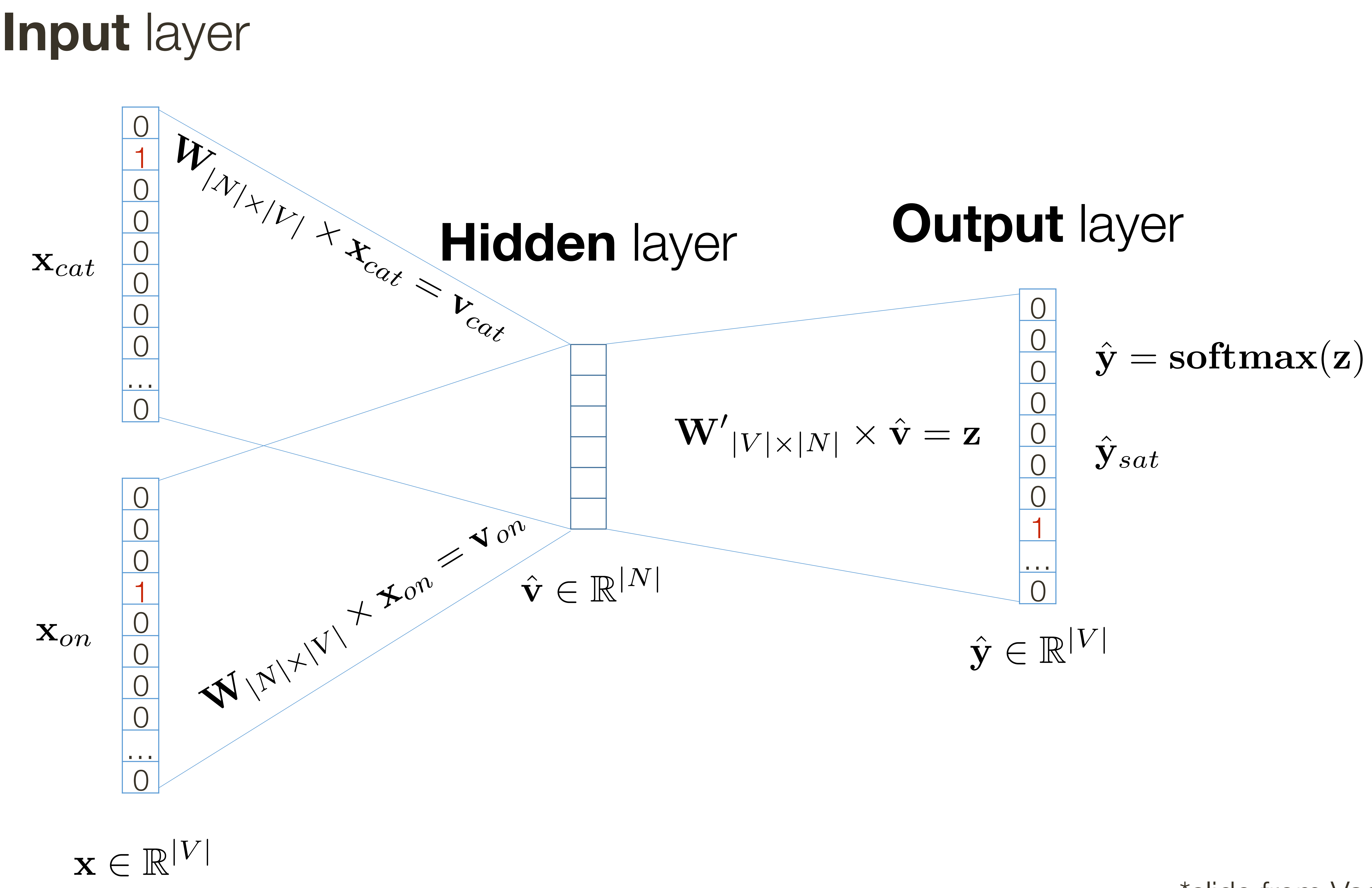
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

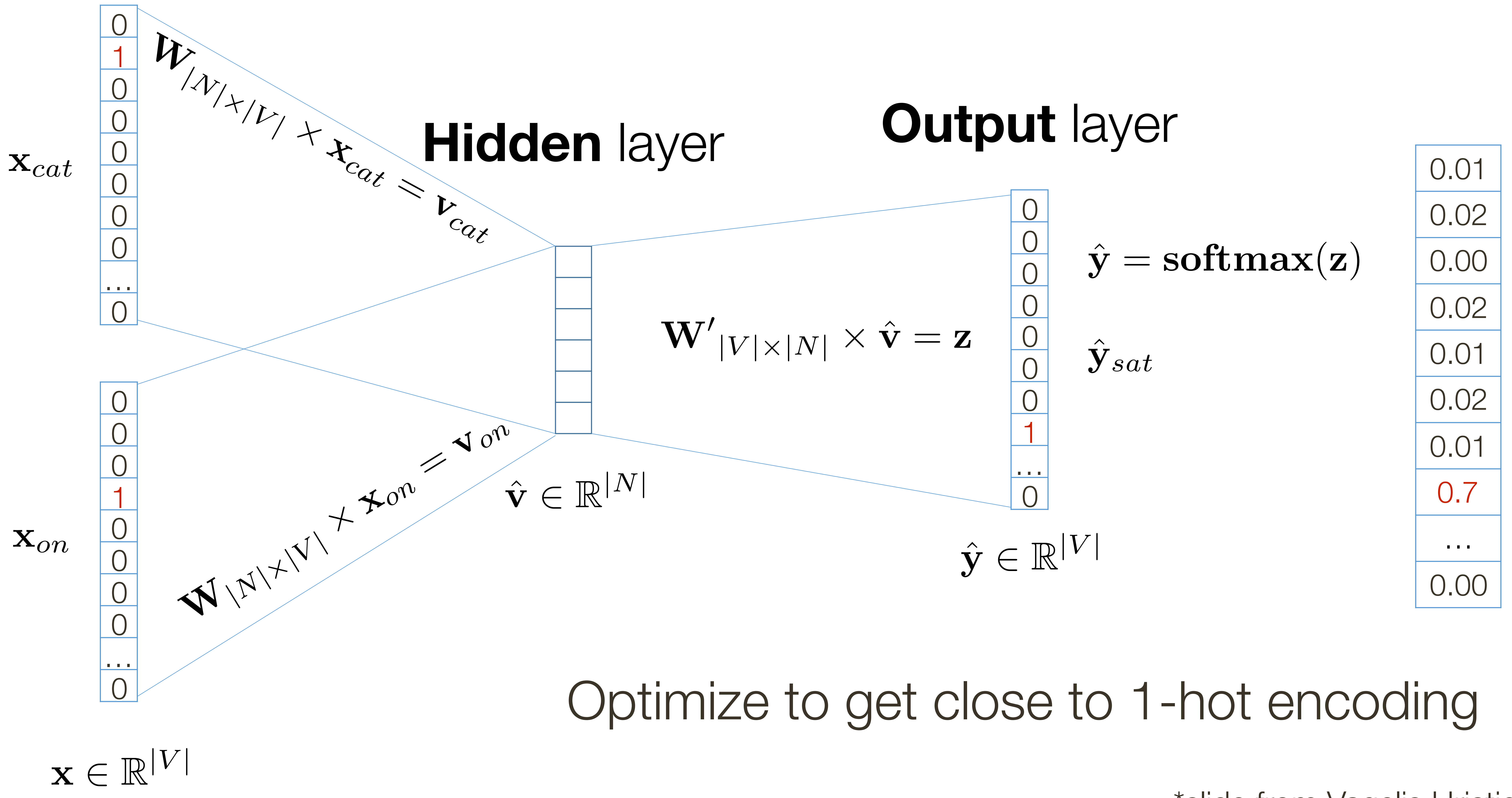
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]

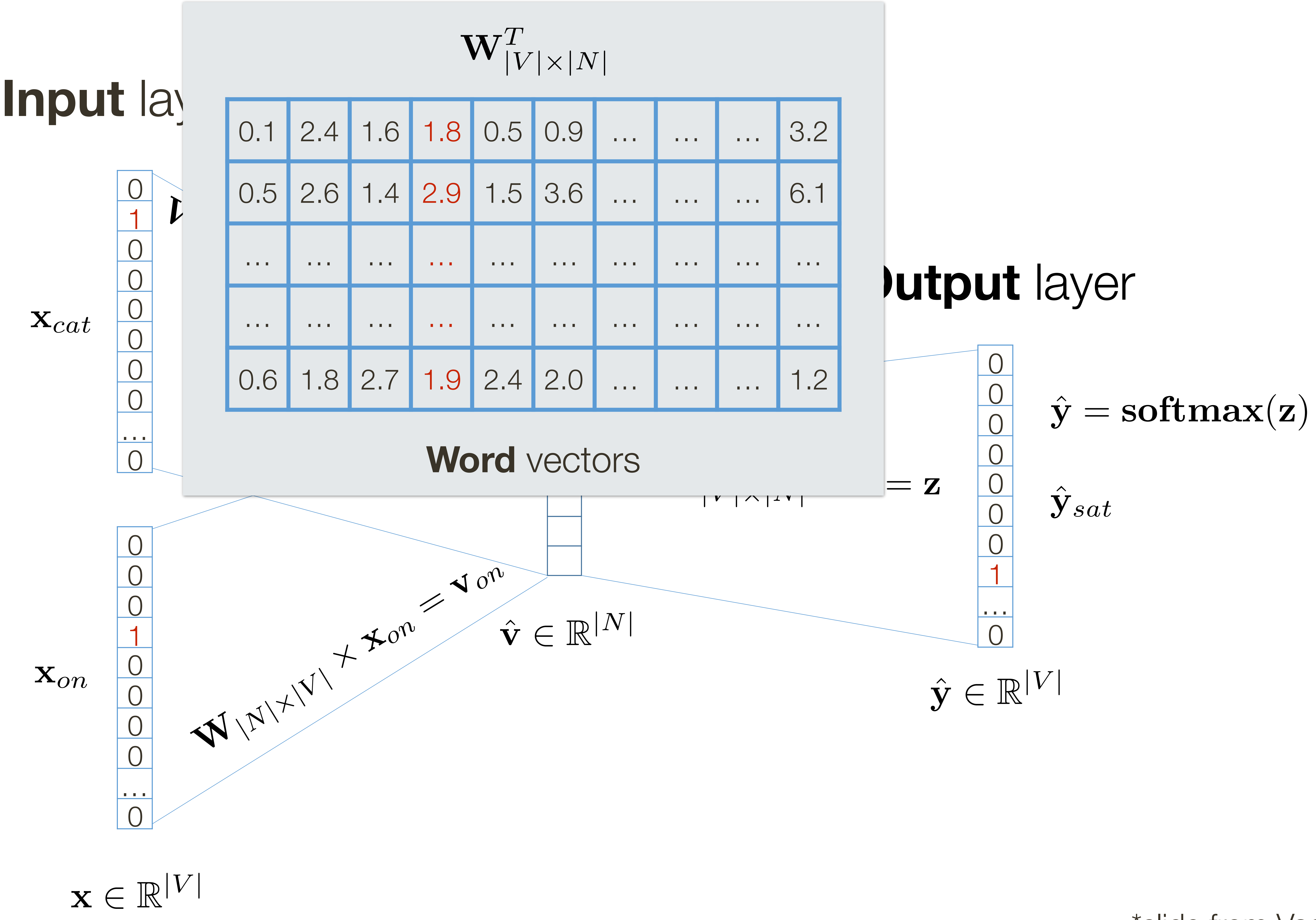



Optimize to get close to 1-hot encoding

*slide from Vagelis Hristidis

# **CBOW**: Continuous Bag of Words

[ Mikolov et al., 2013 ]




$\mathbf{W}^T_{|V|\times|N|}$

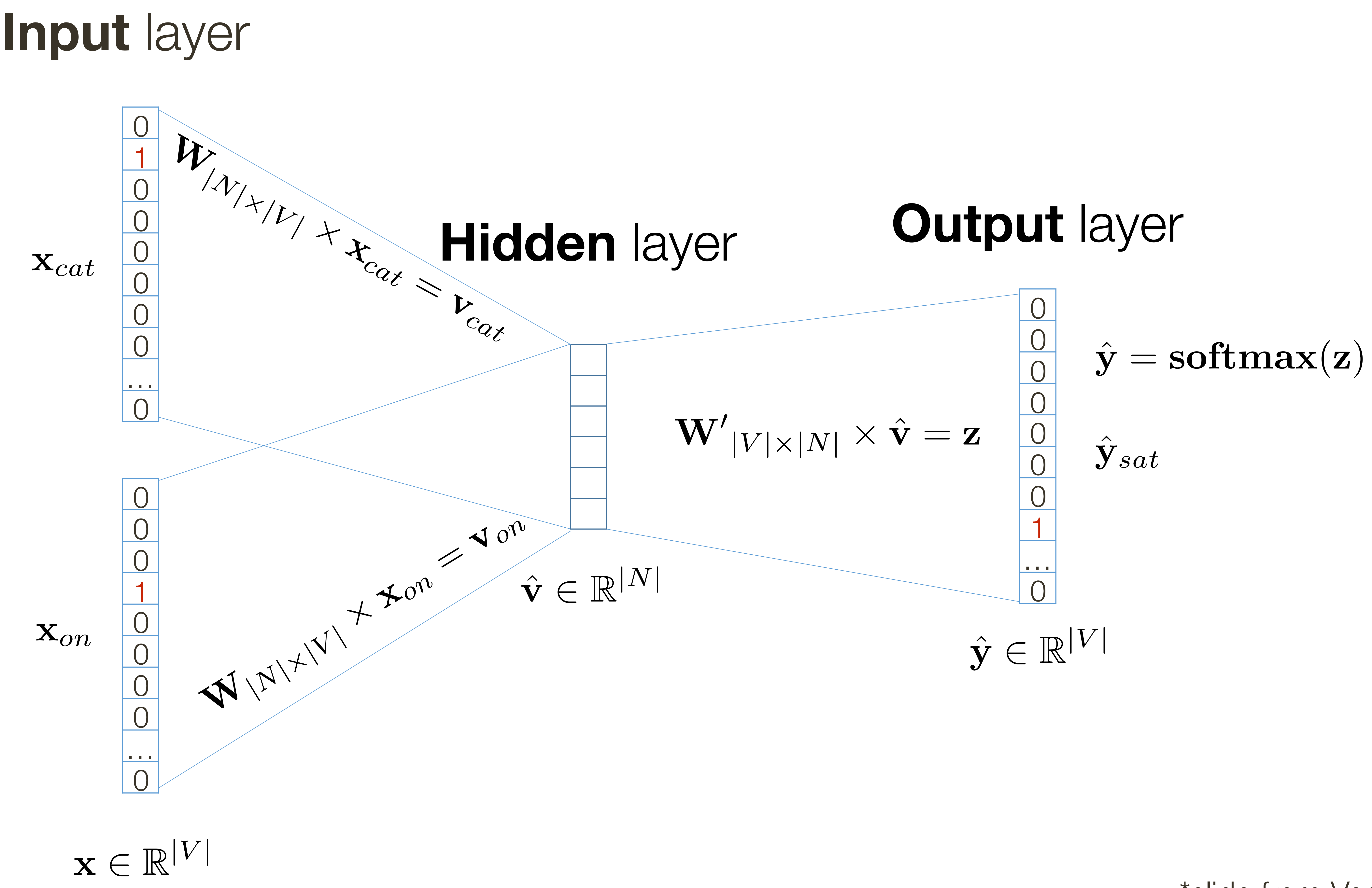
| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| 0.1 | 2.4 | 1.6 | 1.8 | 0.5 | 0.9 | … | … | … | 3.2 |
| 0.5 | 2.6 | 1.4 | 2.9 | 1.5 | 3.6 | … | … | … | 6.1 |
| … | … | … | … | … | … | … | … | … | … |
| … | … | … | … | … | … | … | … | … | … |
| 0.6 | 1.8 | 2.7 | 1.9 | 2.4 | 2.0 | … | … | … | 1.2 |

**Word** vectors

*slide from Vagelis Hristidis

# **CBOW**: Interesting Observation

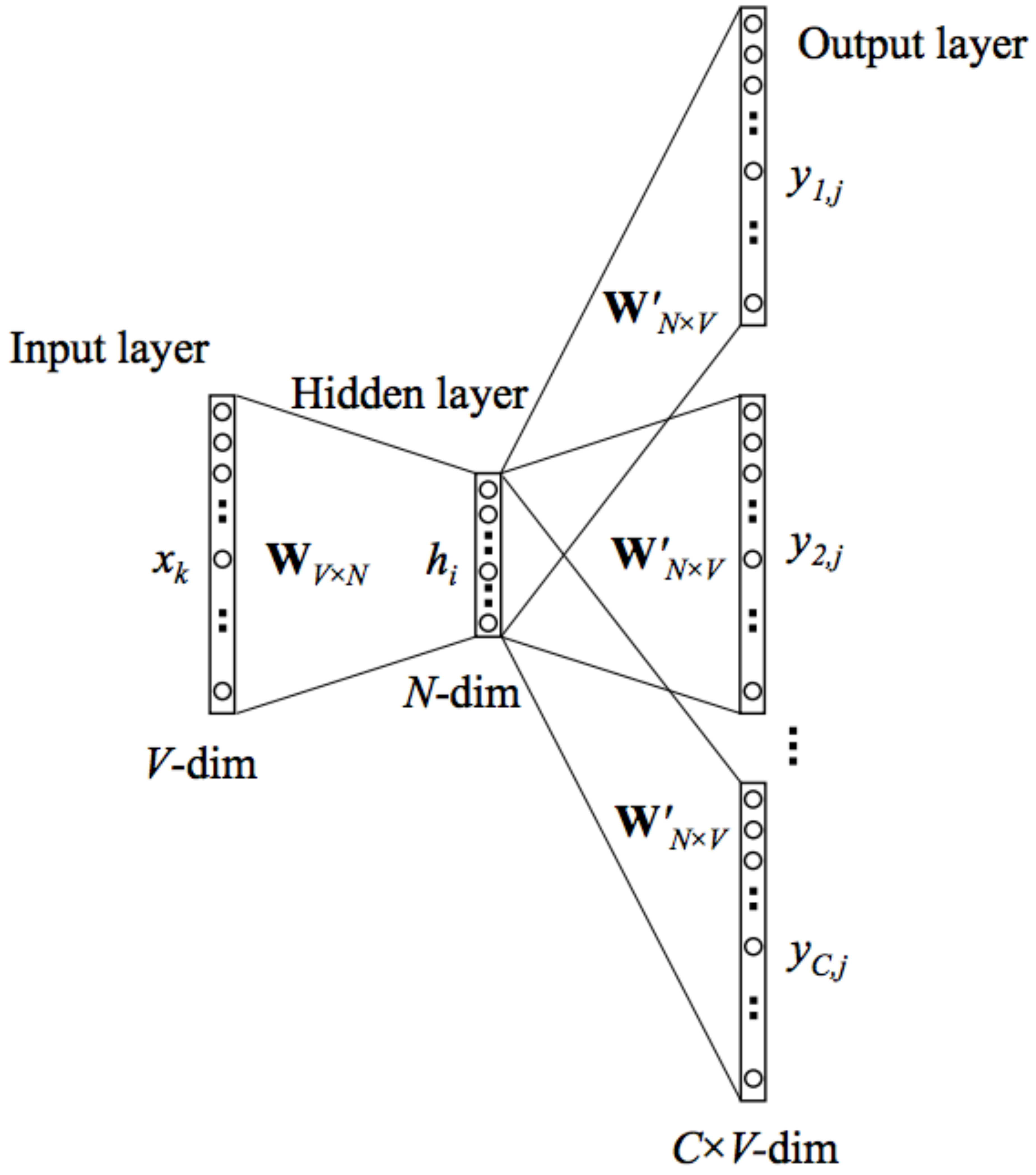
[ Mikolov et al., 2013 ]




*slide from Vagelis Hristidis

# **Skip-Gram** Model

[ Mikolov et al., 2013 ]



# Comparison

[ Mikolov et al., 2013 ]

– **CBOW** is not great for rare words and typically needs less data to train

– **Skip-gram** better for rate words and needs more data to train the model

| Model | Vector Dimensionality | Training words | Accuracy [%] | | |
|---|---|---|---|---|---|
| | | | Semantic | Syntactic | Total |
| Collobert-Weston NNLM | 50 | 660M | 9.3 | 12.3 | 11.0 |
| Turian NNLM | 50 | 37M | 1.4 | 2.6 | 2.1 |
| Turian NNLM | 200 | 37M | 1.4 | 2.2 | 1.8 |
| Mnih NNLM | 50 | 37M | 1.8 | 9.1 | 5.8 |
| Mnih NNLM | 100 | 37M | 3.3 | 13.2 | 8.8 |
| Mikolov RNNLM | 80 | 320M | 4.9 | 18.4 | 12.7 |
| Mikolov RNNLM | 640 | 320M | 8.6 | 36.5 | 24.6 |
| Huang NNLM | 50 | 990M | 13.3 | 11.6 | 12.3 |
| Our NNLM | 20 | 6B | 12.9 | 26.4 | 20.3 |
| Our NNLM | 50 | 6B | 27.9 | 55.8 | 43.2 |
| Our NNLM | 100 | 6B | 34.2 | **64.5** | 50.8 |
| CBOW | 300 | 783M | 15.5 | 53.1 | 36.1 |
| Skip-gram | 300 | 783M | **50.0** | 55.9 | **53.3** |

# Interesting Results: **Word Analogies**

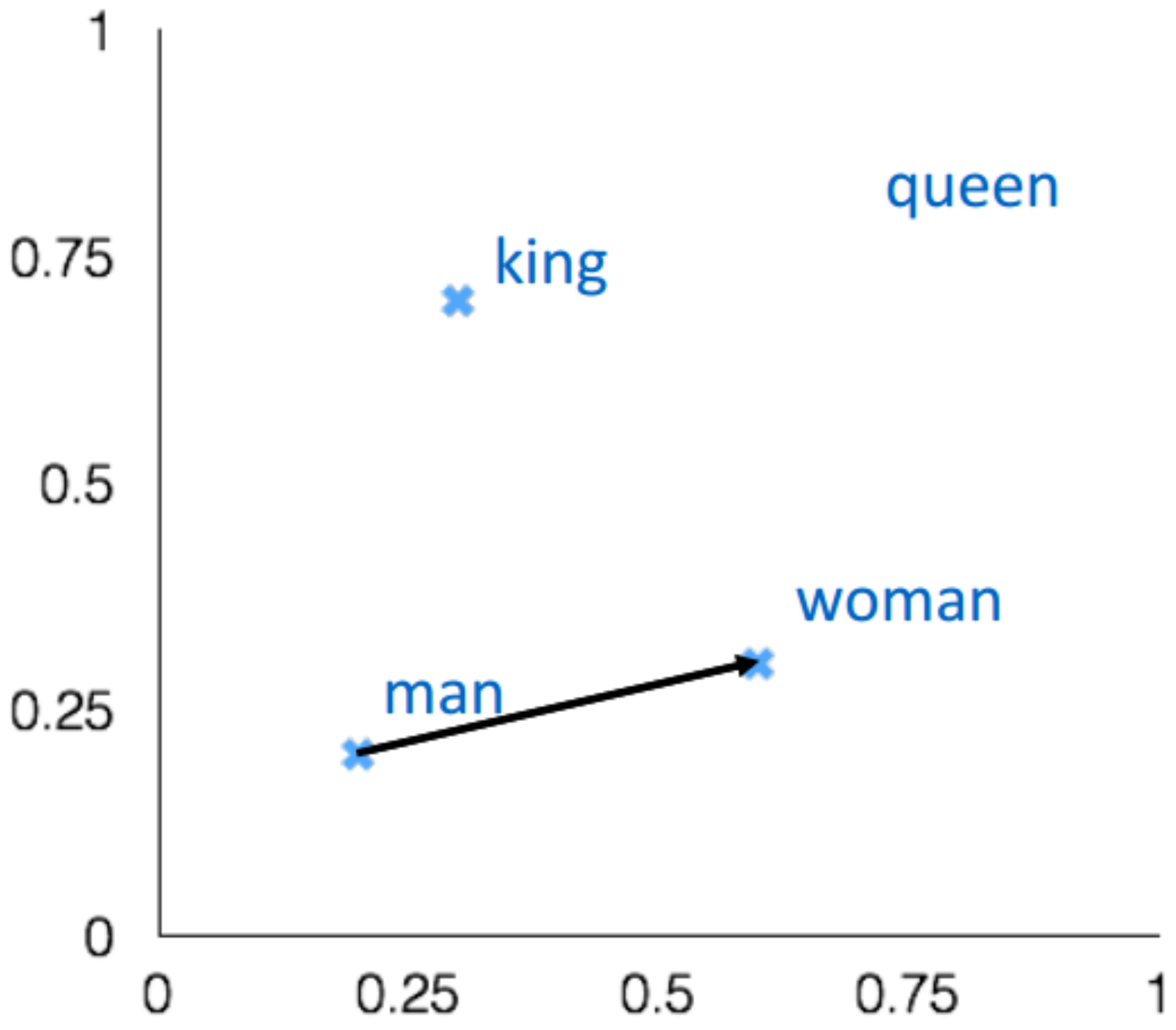
Test for linear relationships, examined by Mikolov et al. (2014)

a:b :: c:?

$$d = \arg\max_x \frac{(w_b - w_a + w_c)^T w_x}{||w_b - w_a + w_c||}$$

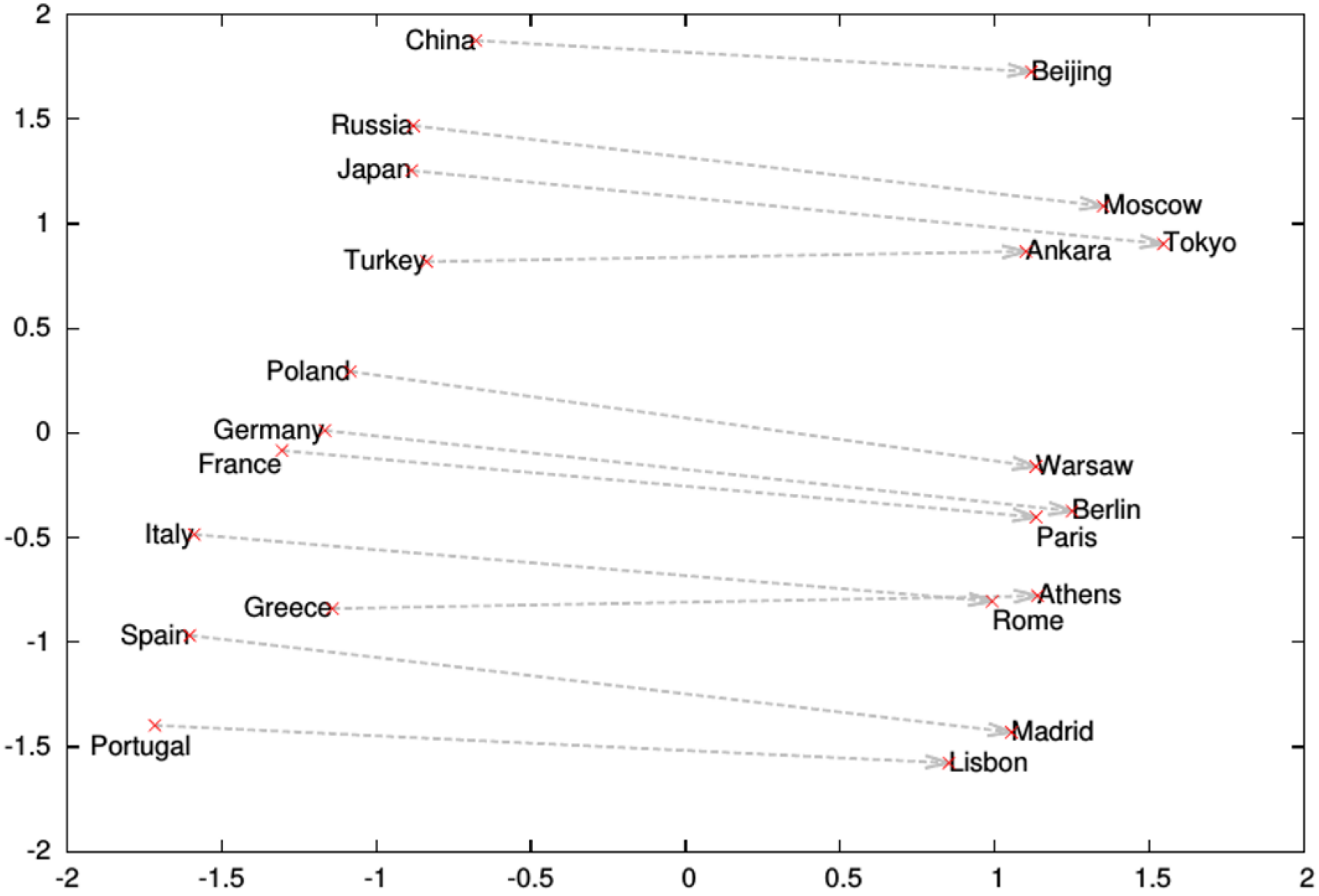
man:woman :: king:?

| | | |
|---|---|---|
| + | king | [ 0.30 0.70 ] |
| - | man | [ 0.20 0.20 ] |
| + | woman | [ 0.60 0.30 ] |
| | queen | [ 0.70 0.80 ] |



# Interesting Results: **Word Analogies**

[ Mikolov et al., 2013 ]



# **Language** Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

# **Language** Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

# **Language** Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

$$\arg\max_{wordsequence} P(wordsequence \mid acoustics) =$$

$$\arg\max_{wordsequence} \frac{P(acoustics \mid wordsequence) \times P(wordsequence)}{P(acoustics)}$$

$$\arg\max_{wordsequence} P(acoustics \mid wordsequence) \times P(wordsequence)$$

* Slides from Louis-Philippe Morency

# **Language** Models

Model the **probability of a sentence**; ideally be able to sample plausible sentences

Why is this useful?

$$\underset{wordsequence}{\arg\max}\, P(wordsequence \mid acoustics) =$$

$$\underset{wordsequence}{\arg\max}\, \frac{P(acoustics \mid wordsequence) \times \boxed{P(wordsequence)}}{P(acoustics)}$$

$$\underset{wordsequence}{\arg\max}\, P(acoustics \mid wordsequence) \times \boxed{P(wordsequence)}$$

# Simple **Language Models**: N-Grams

Given a word sequence: $w_{1:n} = [w_1, w_2, ..., w_n]$

We want to estimate $p(w_{1:n})$

# Simple **Language Models**: N-Grams

Given a word sequence: $w_{1:n} = [w_1, w_2, ..., w_n]$

We want to estimate $p(w_{1:n})$

Using **Chain Rule** of probabilities:

$$p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2) \cdots p(w_n|w_{1:n-1})$$

# Simple **Language Models**: N-Grams

Given a word sequence: $w_{1:n} = [w_1, w_2, ..., w_n]$

We want to estimate $p(w_{1:n})$

Using **Chain Rule** of probabilities:

$$p(w_{1:n}) = p(w_1)p(w_2|w_1)p(w_3|w_1, w_2)\cdots p(w_n|w_{1:n-1})$$

**Bi-gram** Approximation:

$$p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-1})$$

**N-gram** Approximation:

$$p(w_{1:n}) = \prod_{k=1}^{n} p(w_k|w_{k-N+1:k-1})$$

* Slides from Louis-Philippe Morency

# Estimating **Probabilities**

N-gram conditional probabilities can be estimated based on raw concurrence counts in the observed sequences

**Bi-gram**:

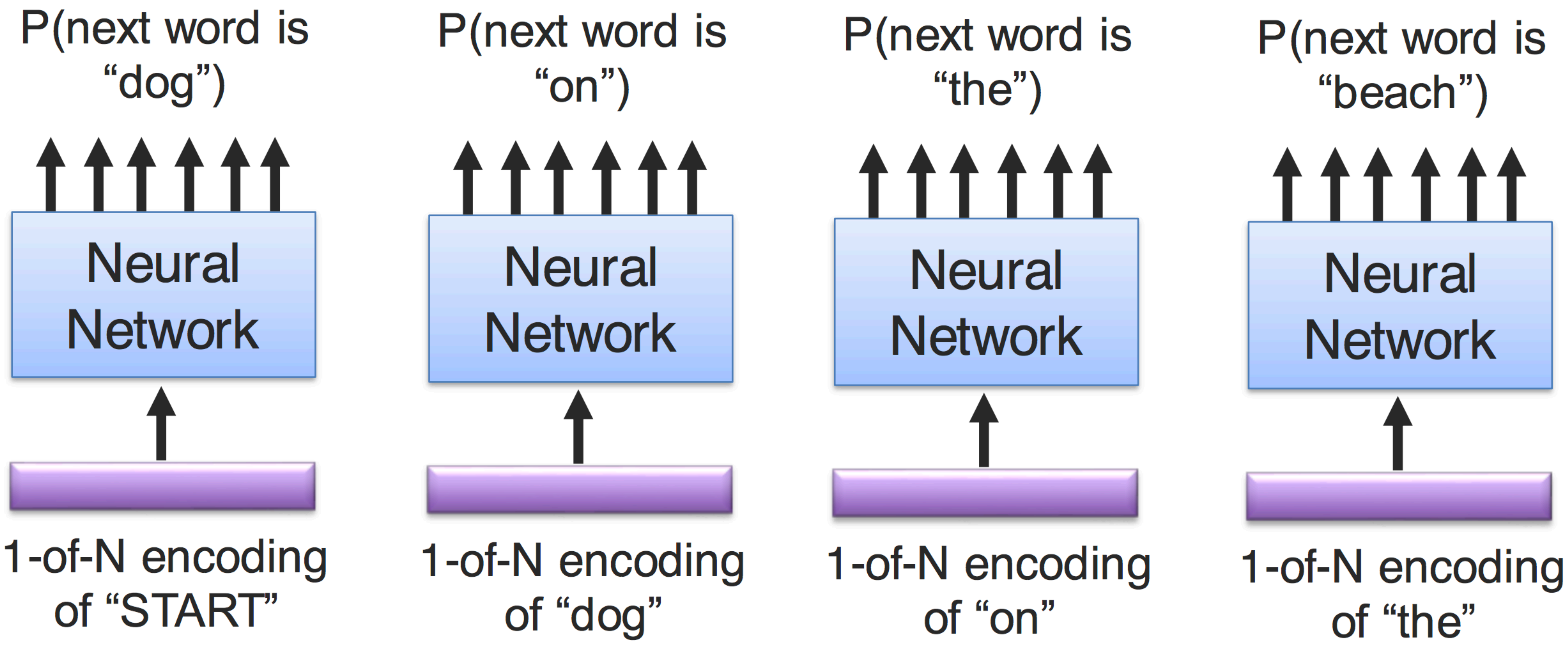
$$p(w_n|w_{n-1}) = \frac{C(w_{n-1}w_n)}{C(w_{n-1})}$$

**N-gram**:

$$p(w_n|w_{n-N-1:n-1}) = \frac{C(w_{n-N-1:n-1}w_n)}{C(w_{n-N-1:n-1})}$$

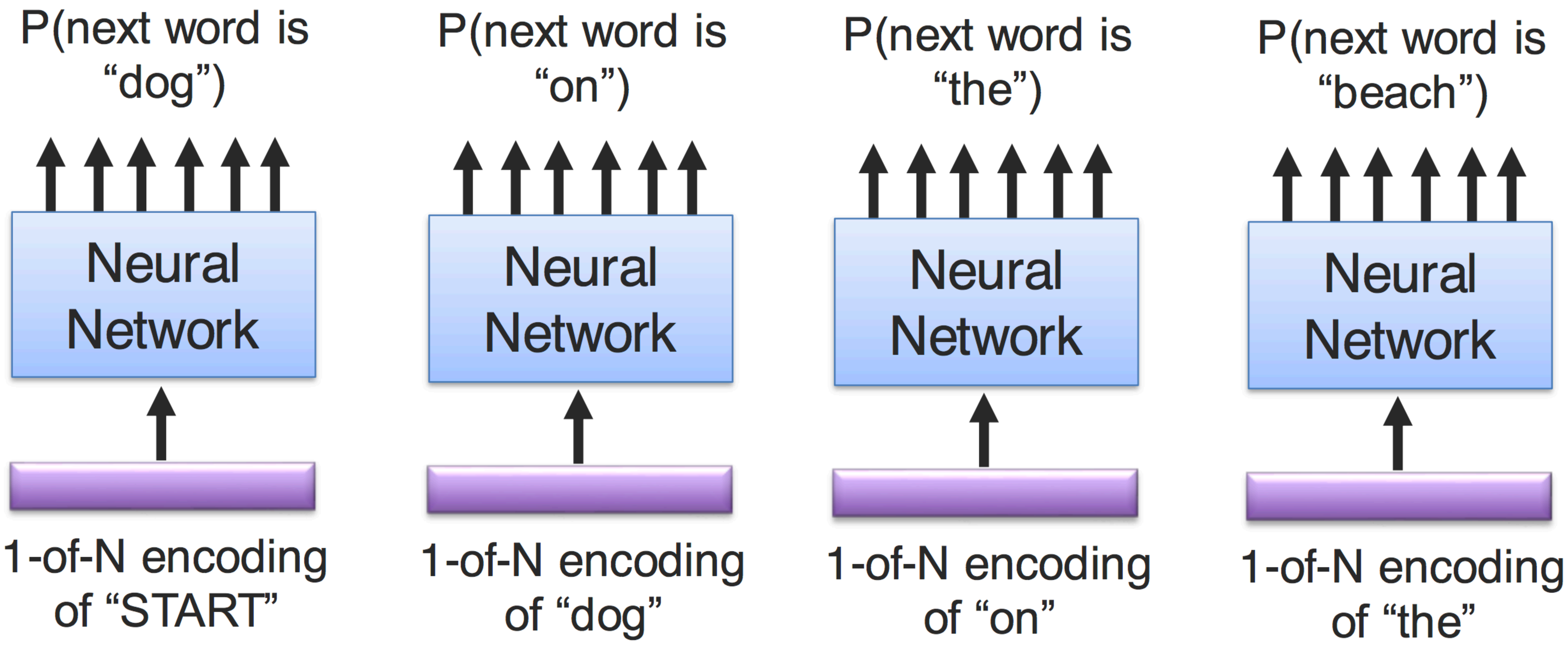
* Slides from Louis-Philippe Morency

# **Neural-based** Unigram Language Mode




* Slides from Louis-Philippe Morency

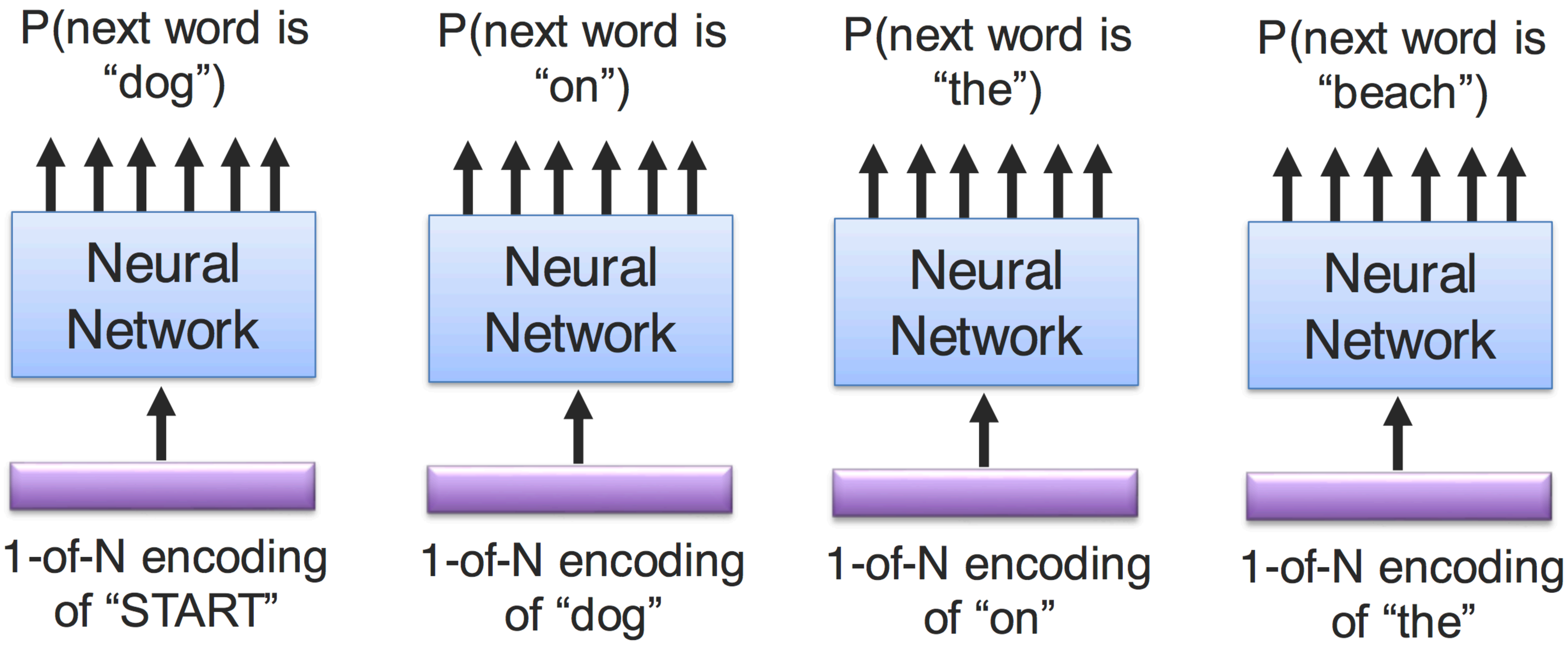
# **Neural-based** Unigram Language Mode




**Problem:** Does not model sequential information (too local)

* Slides from Louis-Philippe Morency

# **Neural-based** Unigram Language Mode




**Problem:** Does not model sequential information (too local)

**We need sequence modeling!**

* Slides from Louis-Philippe Morency

# Why Model **Sequences**?


→ FOREIGN MINISTER.

→ THE SOUND OF

$$\begin{array}{lllllllll} & & a_1{=}2 & a_2{=}0 & a_3{=}1 & a_4{=}3 & a_5{=}4 & a_6{=}2 & a_7{=}5 \\ \boldsymbol{x} & = & \text{bringen} & \text{sie} & \text{bitte} & \text{das} & \text{auto} & \text{zurück} & . \end{array}$$

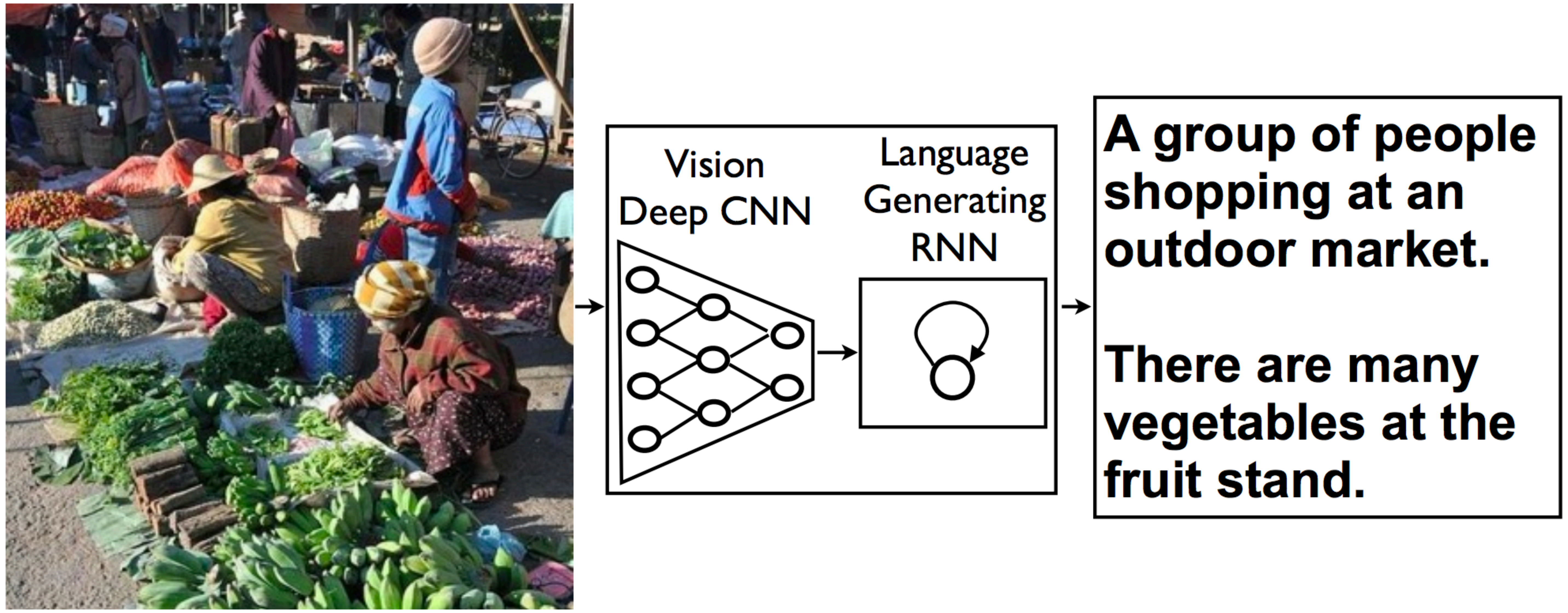
$$\boldsymbol{y} = \text{please} \quad \text{return} \quad \text{the} \quad \text{car} \quad .$$

Image Credit: Alex Graves and Kevin Gimpel

* slide from Dhruv Batra

# **Multi-modal** tasks

[ Vinyals *et al.*, 2015 ]



# **Sequences** where you don't expect them …

Classify images by taking a series of "glimpses"

[ Gregor et al., ICML 2015 ]

[ Mnih et al., ICLR 2015 ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Sequences** where you don't expect them …

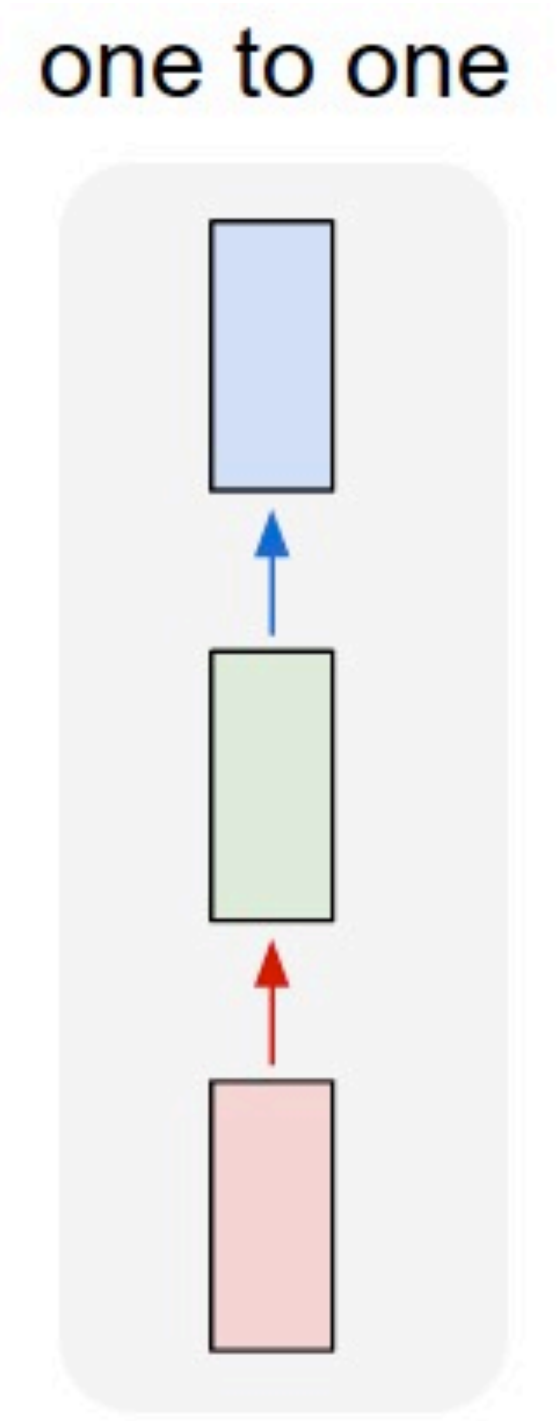
Classify images by taking a series of "glimpses"

[ Gregor et al., ICML 2015 ]

[ Mnih et al., ICLR 2015 ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Sequences** in Inputs or Outputs?

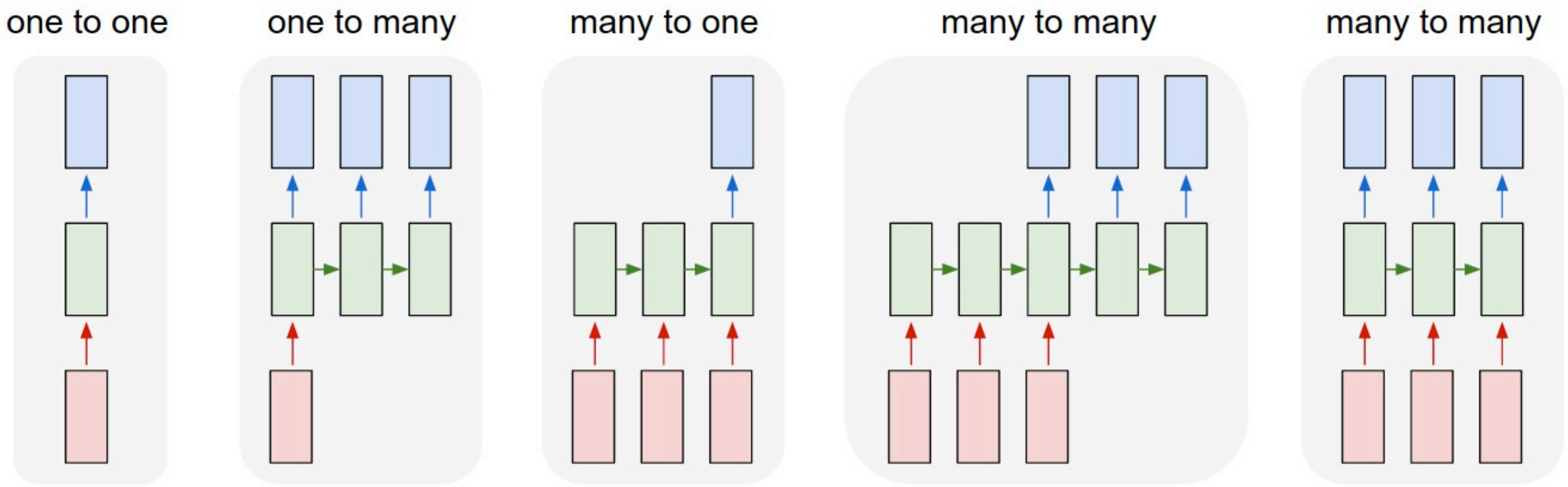



**Input:** No sequence

**Output:** No seq.

**Example:**
"standard"
classification /
regression problems

# **Sequences** in Inputs or Outputs?




**Input:** No sequence
**Output:** No seq.
**Example:** "standard" classification / regression problems

**Input:** No sequence
**Output:** Sequence
**Example:** Im2Caption

**Input:** Sequence
**Output:** No seq.
**Example:** sentence classification, multiple-choice question answering

**Input:** Sequence
**Output:** Sequence
**Example:** machine translation, video captioning, open-ended question answering, video question answering

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Key Conceptual** Ideas

**Parameter Sharing**

— in computational graphs = adding gradients

“**Unrolling**”

— in computational graphs with parameter sharing

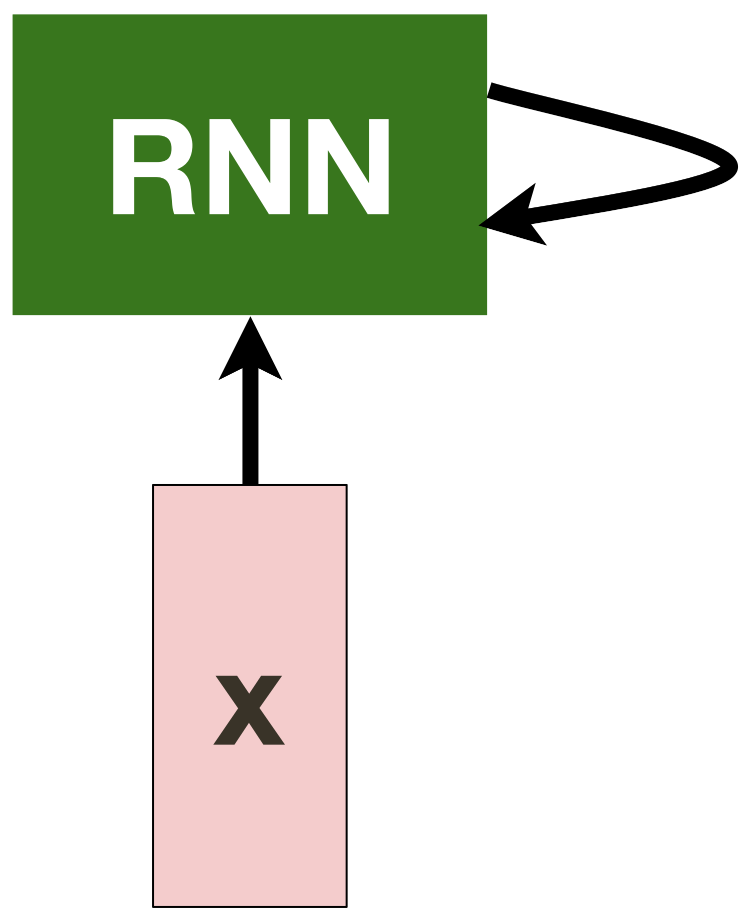
Parameter Sharing + “Unrolling”

— Allows modeling **arbitrary length sequences**!

— Keeps number of parameters in check

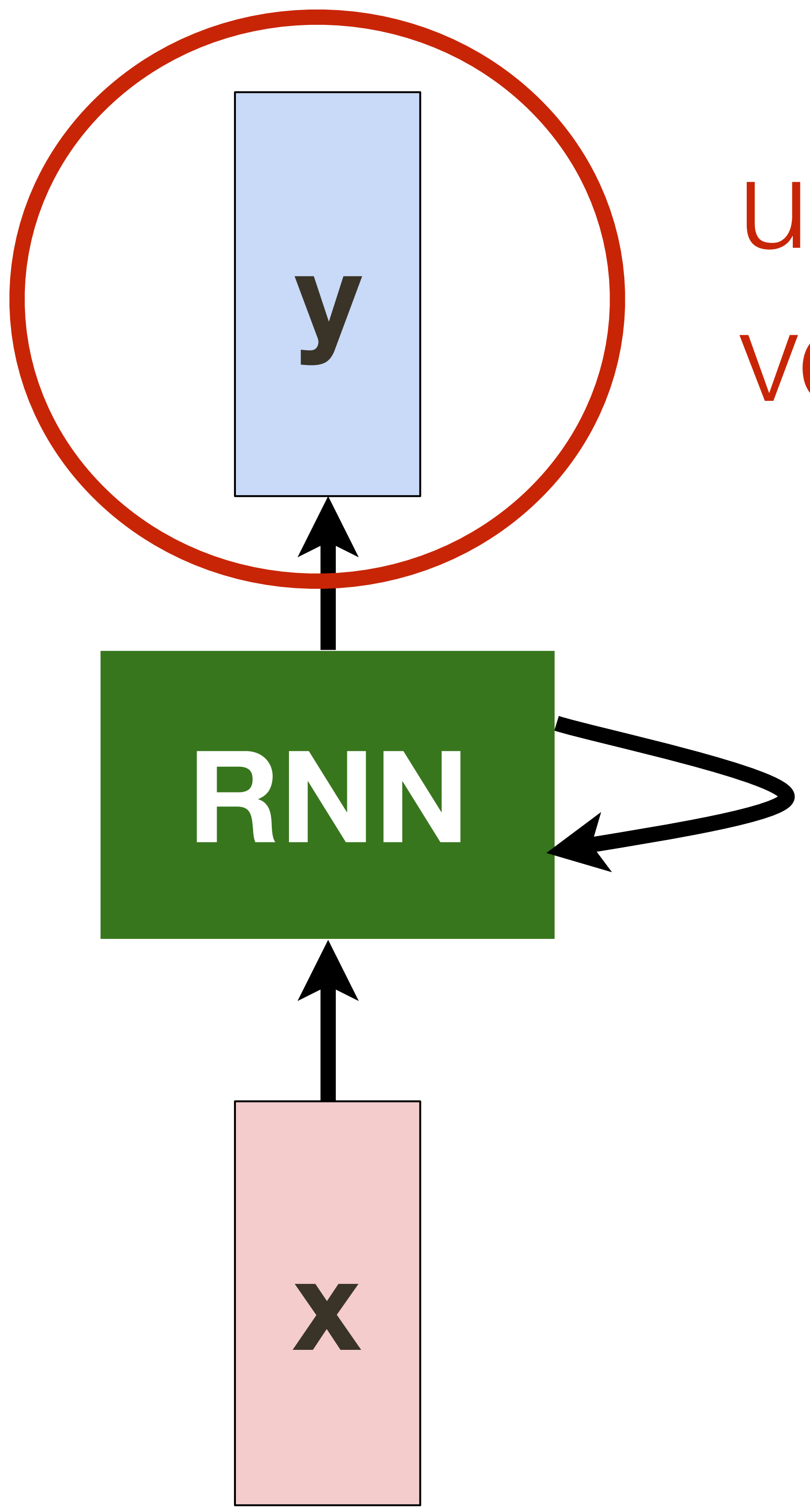
* slide from Dhruv Batra

# **Recurrent** Neural Network




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Recurrent** Neural Network




usually want to predict a vector at some time steps

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Recurrent** Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

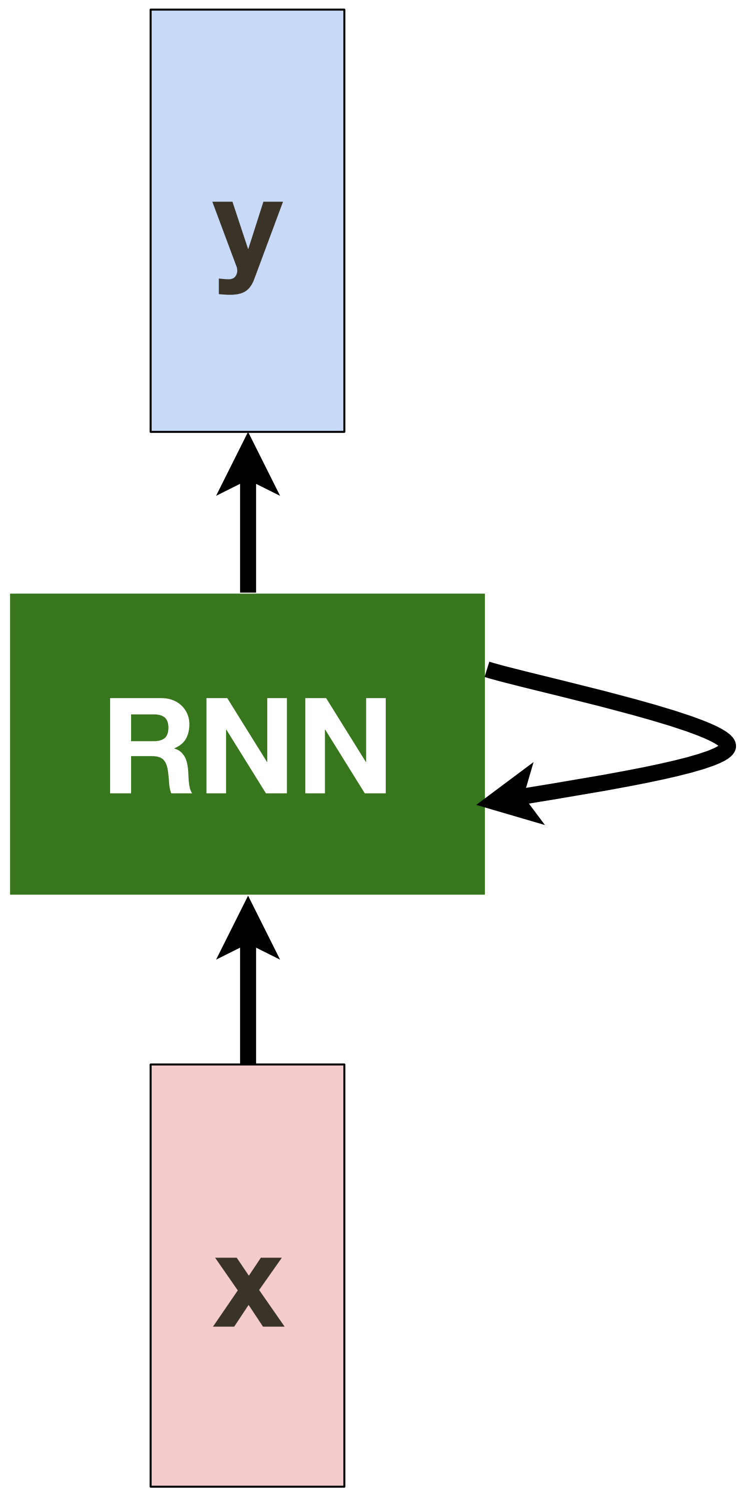
new **state**

old **state**

$$h_t = f_W(h_{t-1}, x_t)$$

**input** vector at some time step

some **function** with parameters W



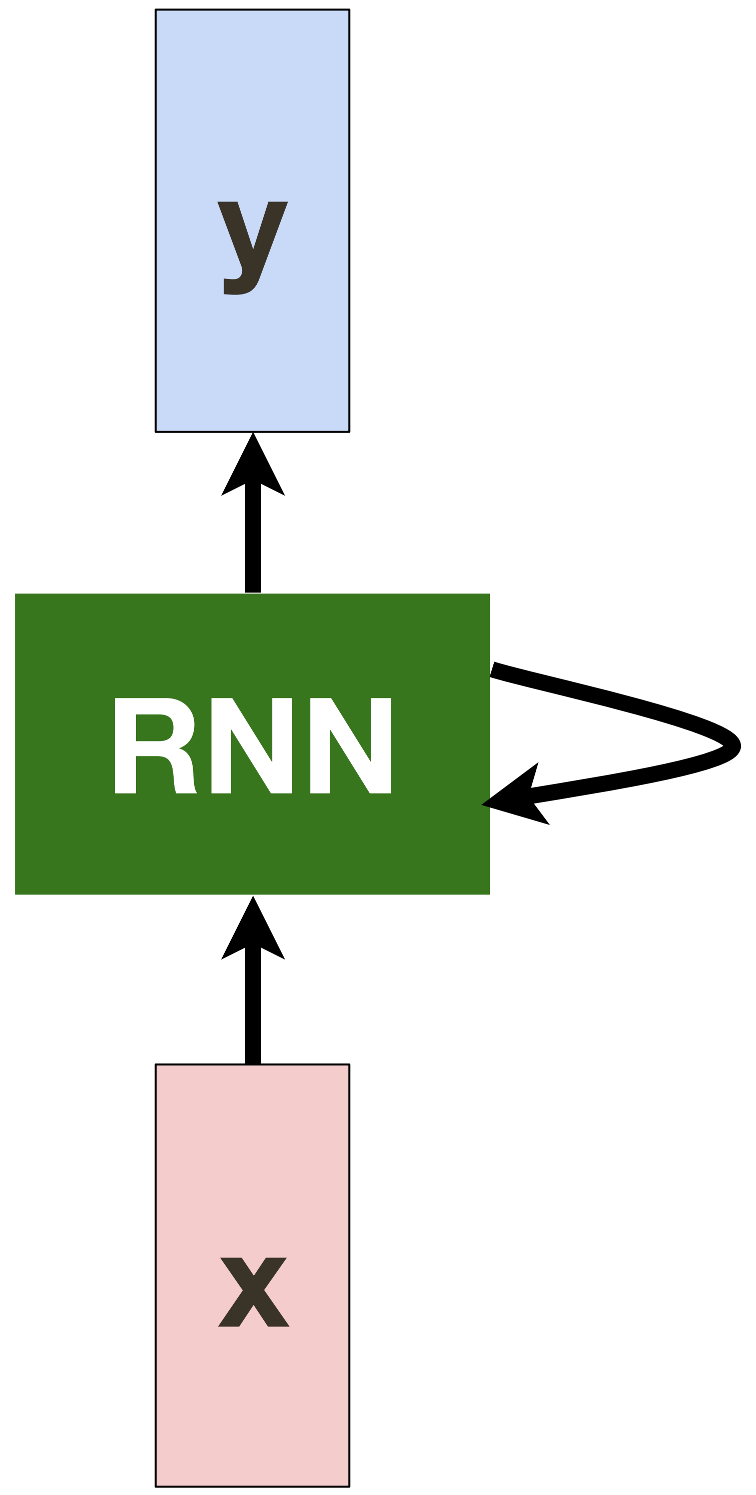

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Recurrent** Neural Network

We can process a sequence of vectors **x** by applying a **recurrence formula** at every time step:

$$h_t = f_W(h_{t-1}, x_t)$$

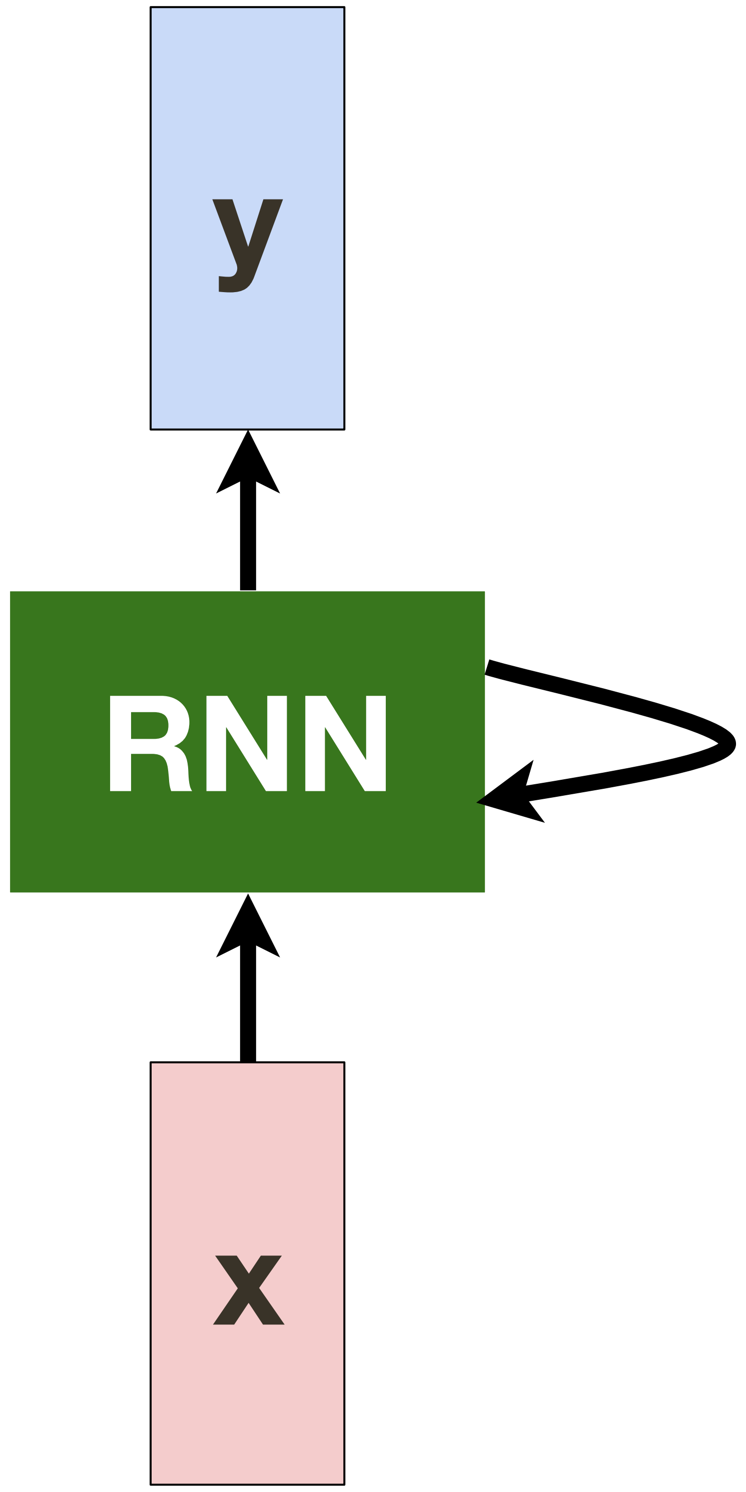
**Note:** the same function and the same set of parameters are used at every time step




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# (Vanilla) **Recurrent** Neural Network

$$h_t = f_W(h_{t-1}, x_t)$$



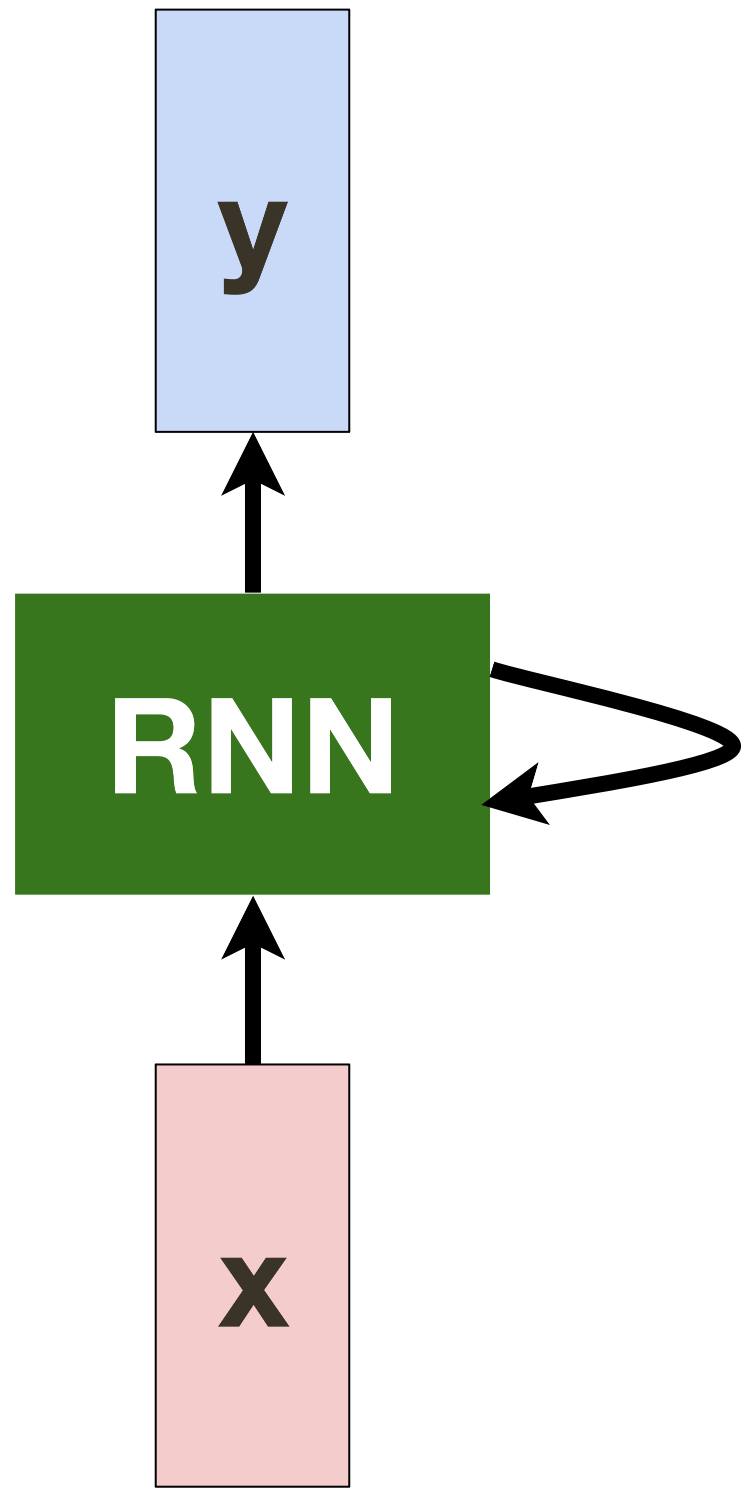

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# (Vanilla) **Recurrent** Neural Network

$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

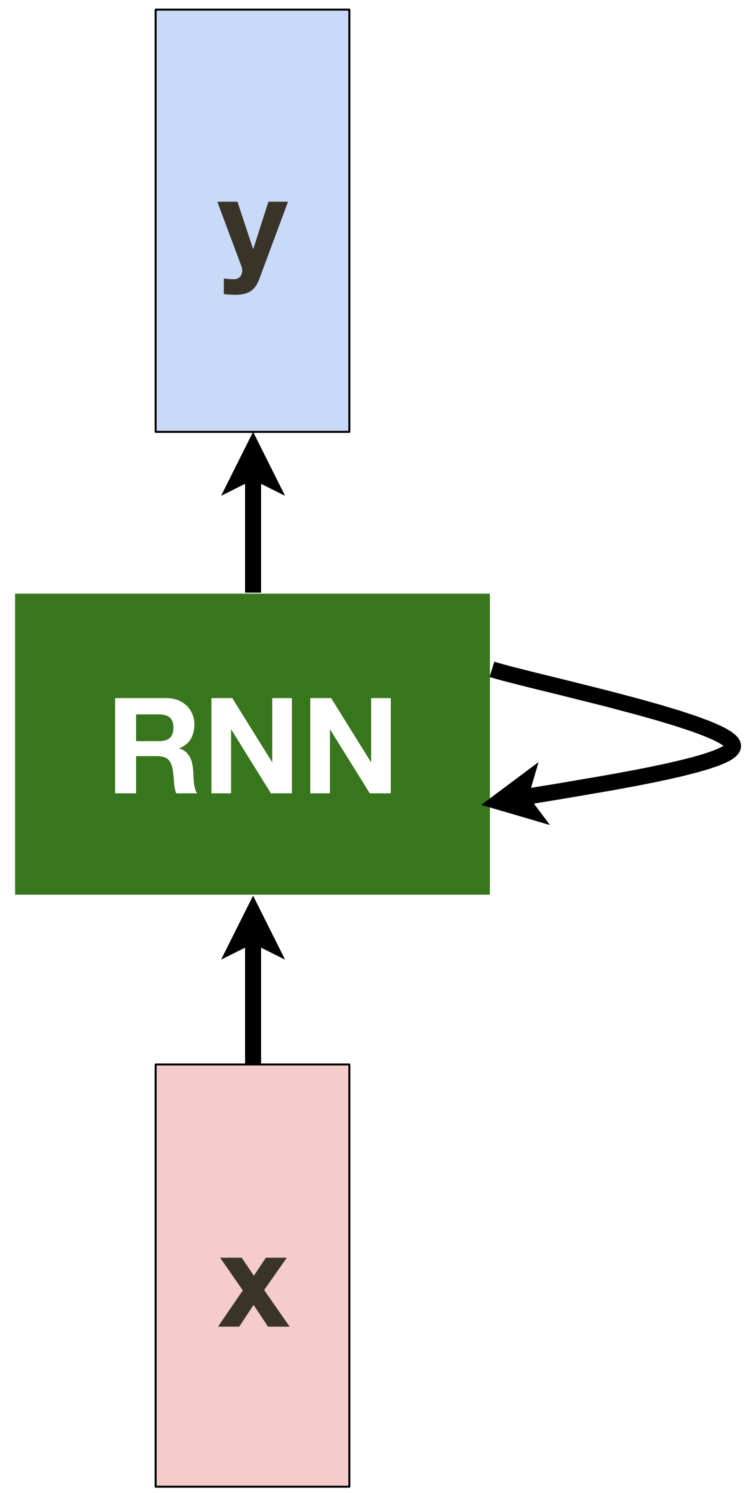
# (Vanilla) **Recurrent** Neural Network

$$y_t = W_{hy} h_t + b_y$$

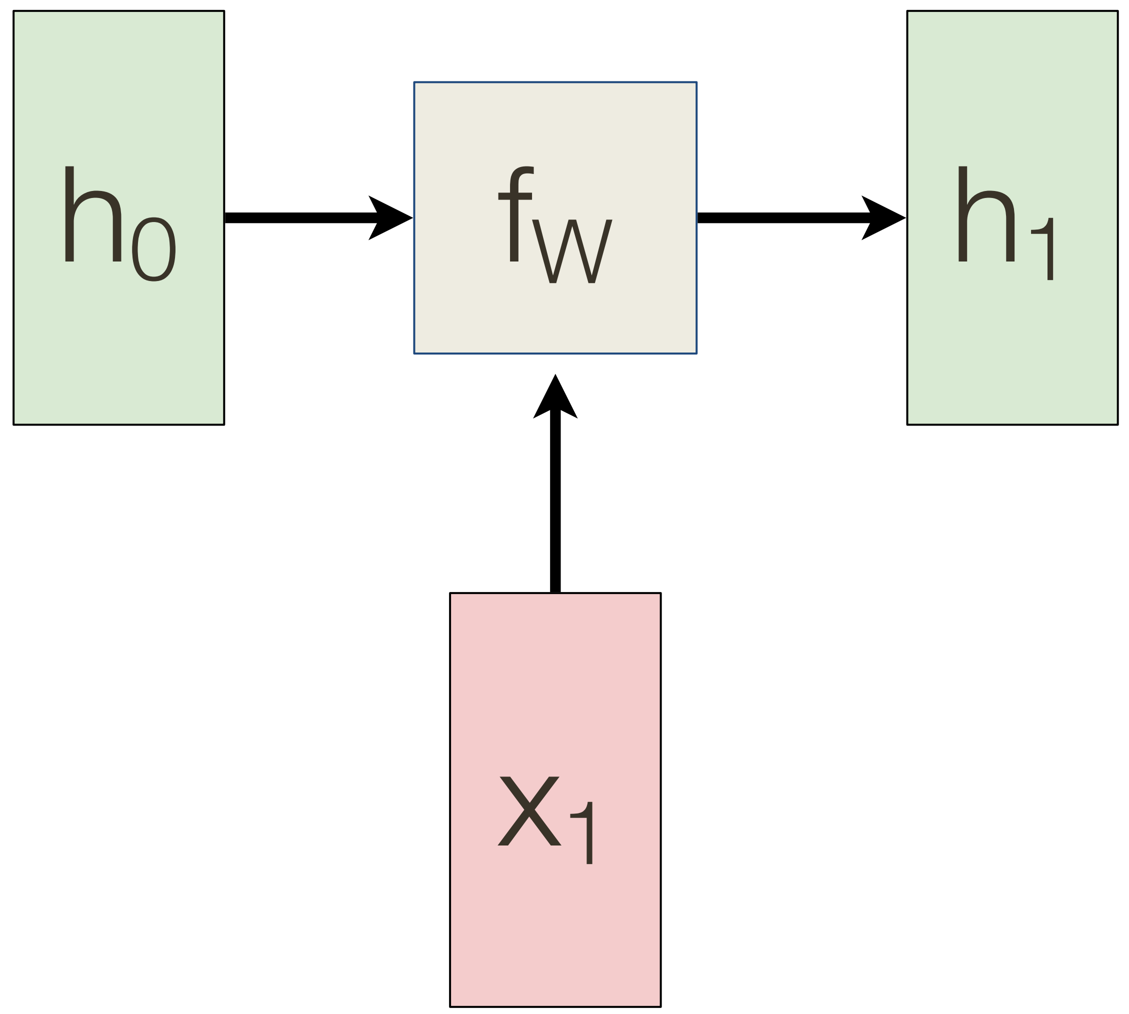
$$h_t = f_W(h_{t-1}, x_t)$$

$$\downarrow$$

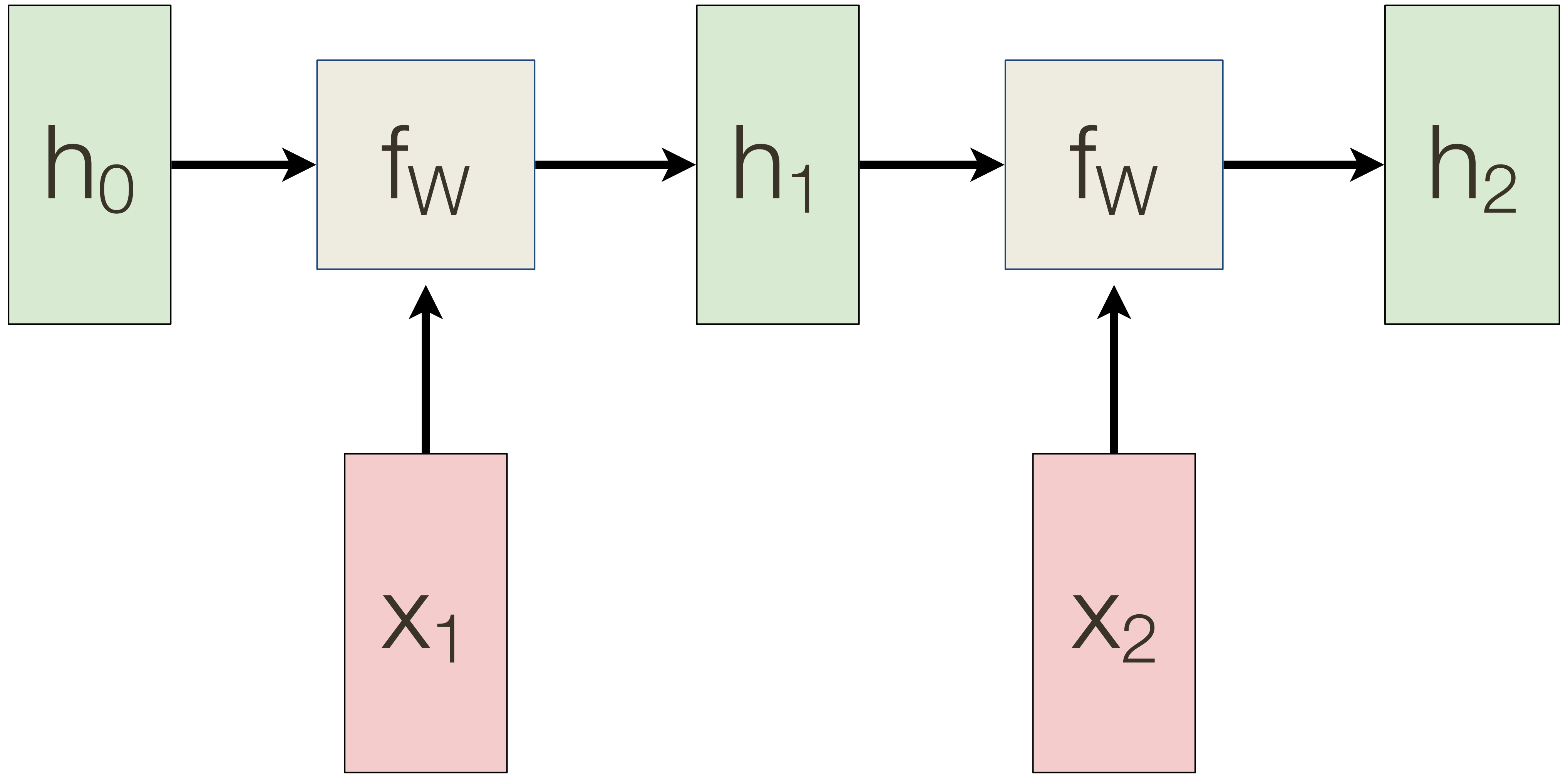
$$h_t = \tanh(W_{hh} h_{t-1} + W_{xh} x_t + b_h)$$



# RNN **Computational Graph**

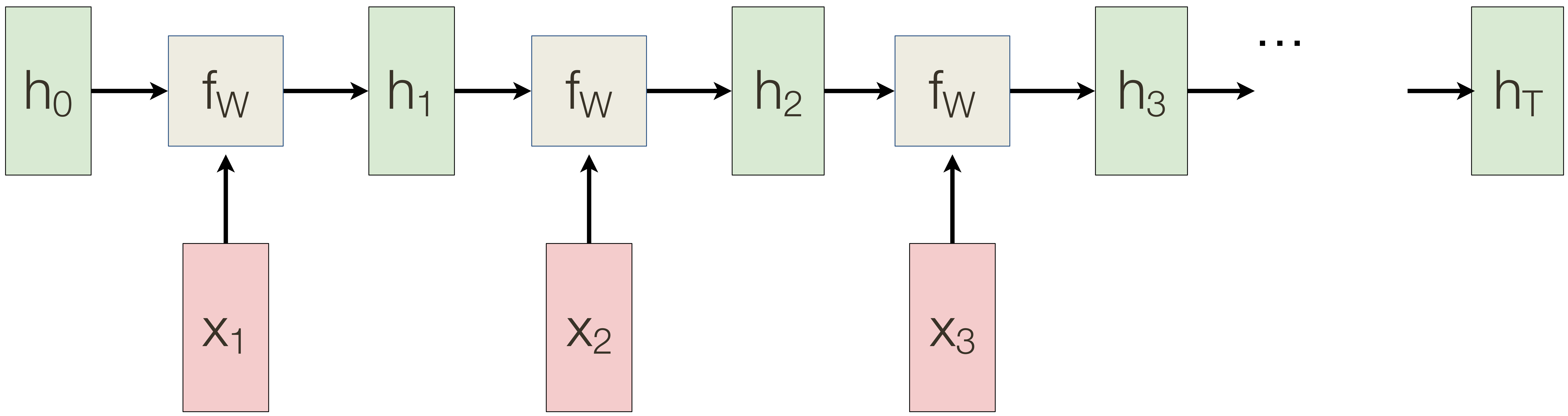


# RNN **Computational Graph**



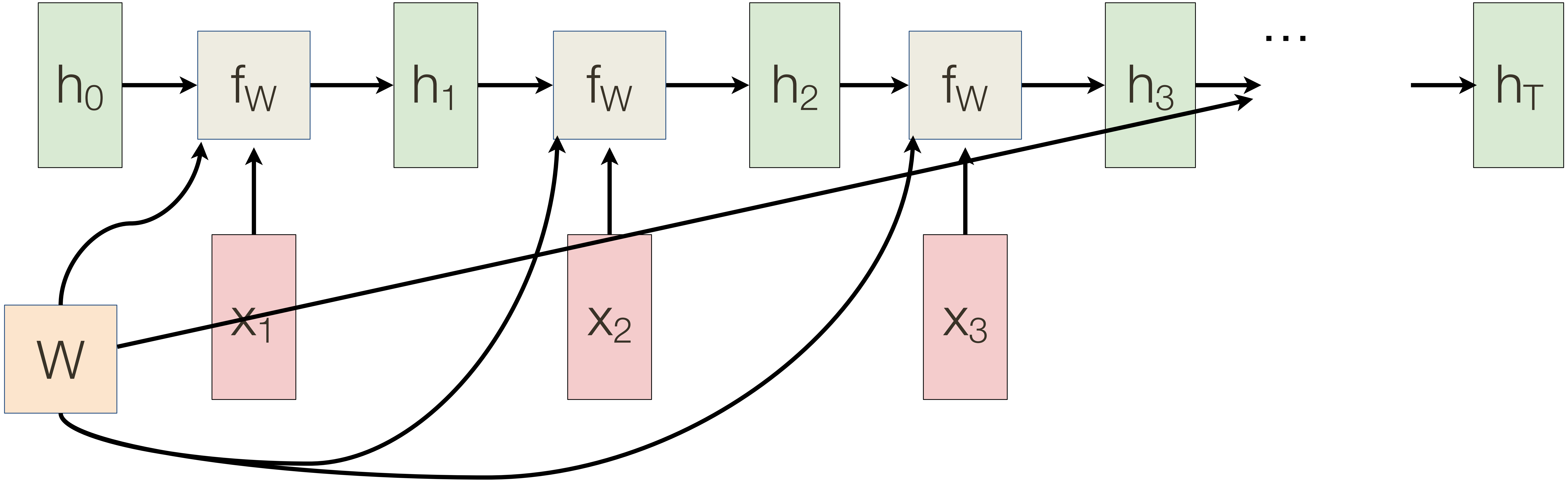

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# RNN **Computational Graph**



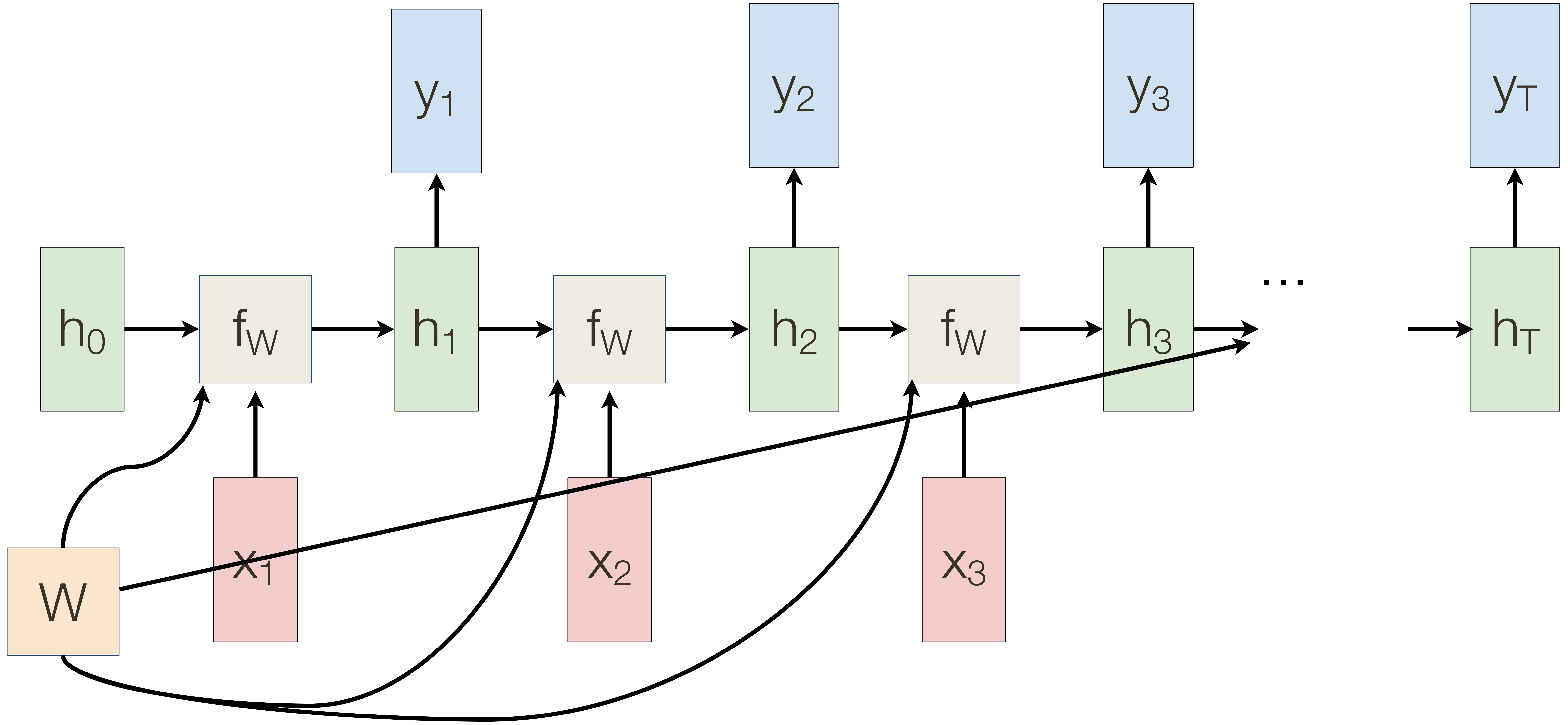
# RNN **Computational Graph**

Re-use the same weight matrix at every time-step

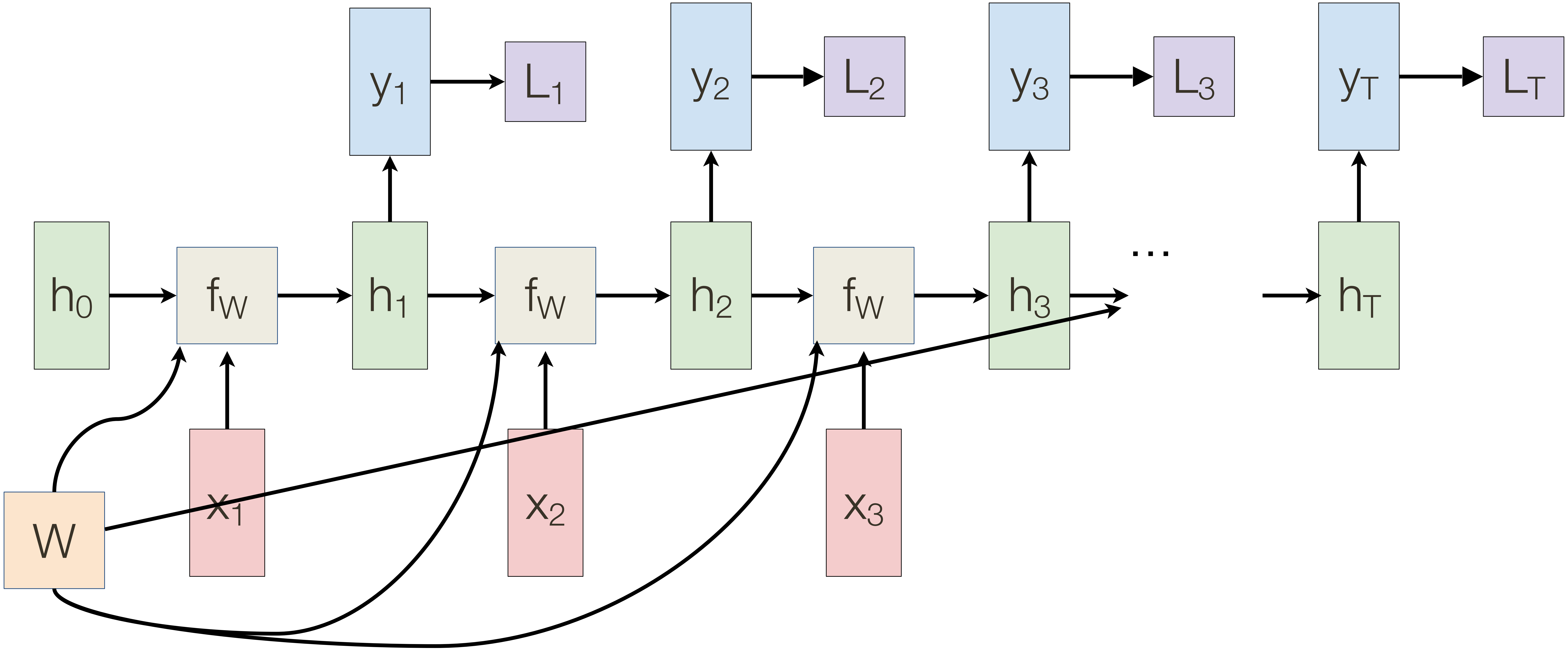



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

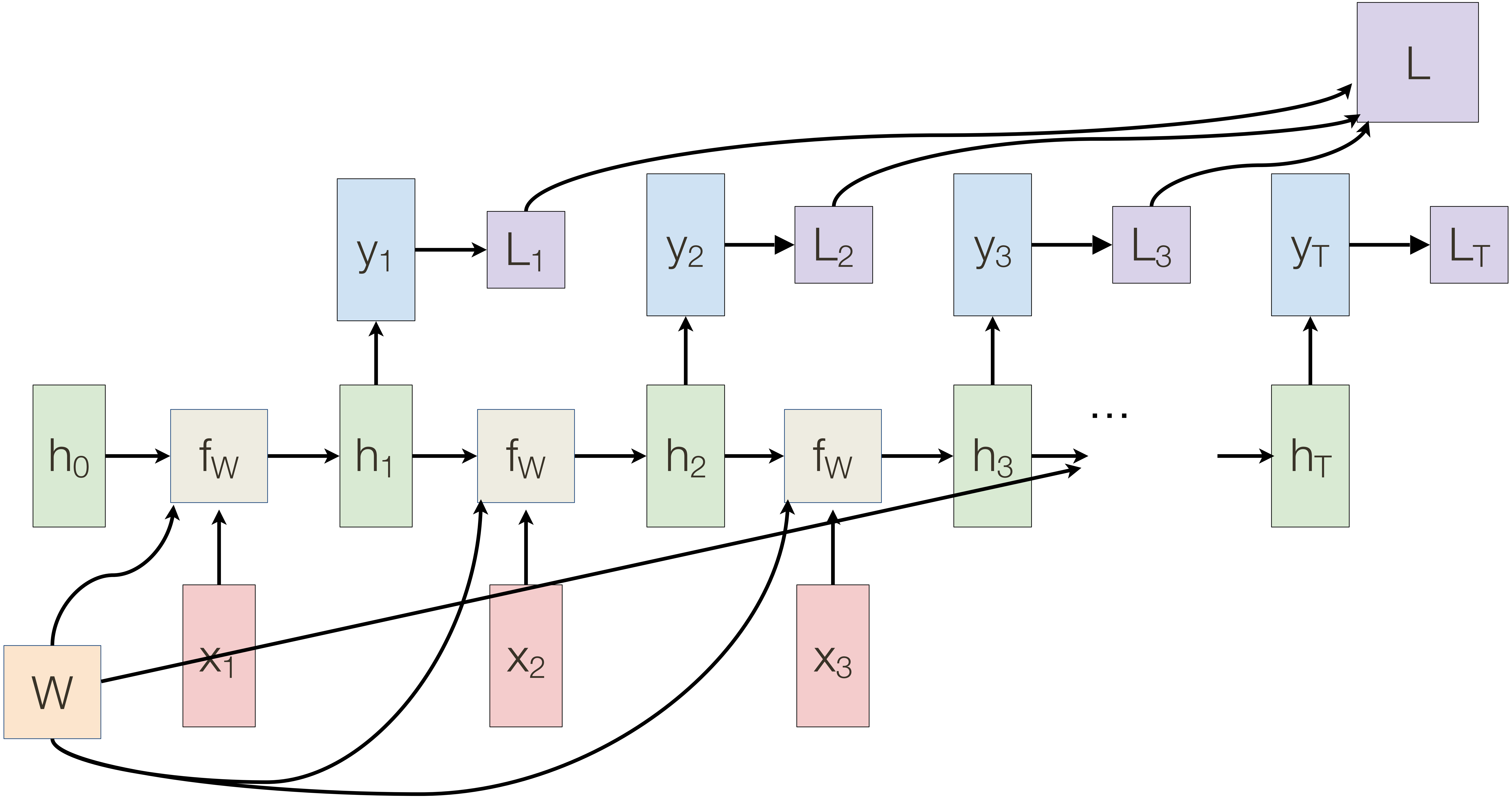
# RNN **Computational Graph**: Many to Many



# RNN **Computational Graph**: Many to Many

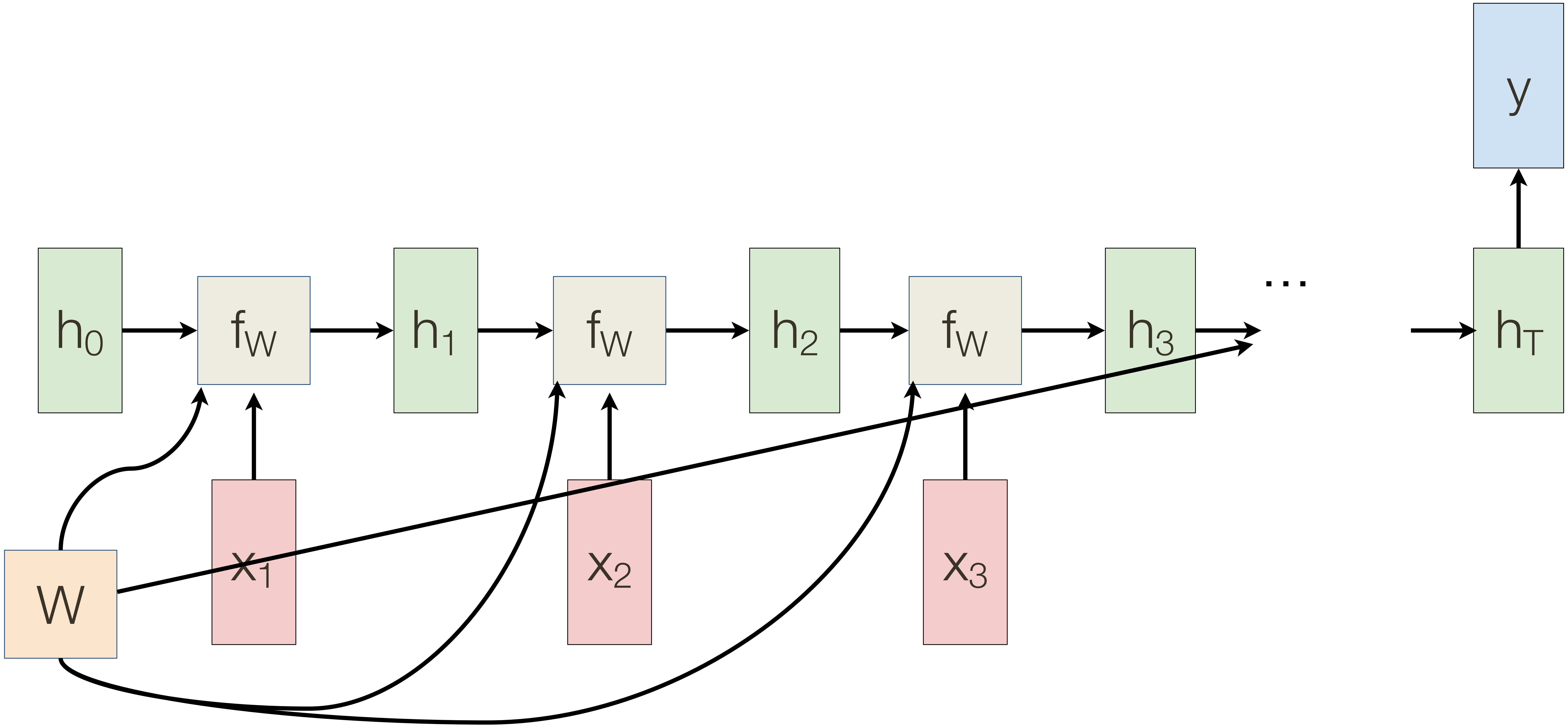


# RNN **Computational Graph**: Many to Many



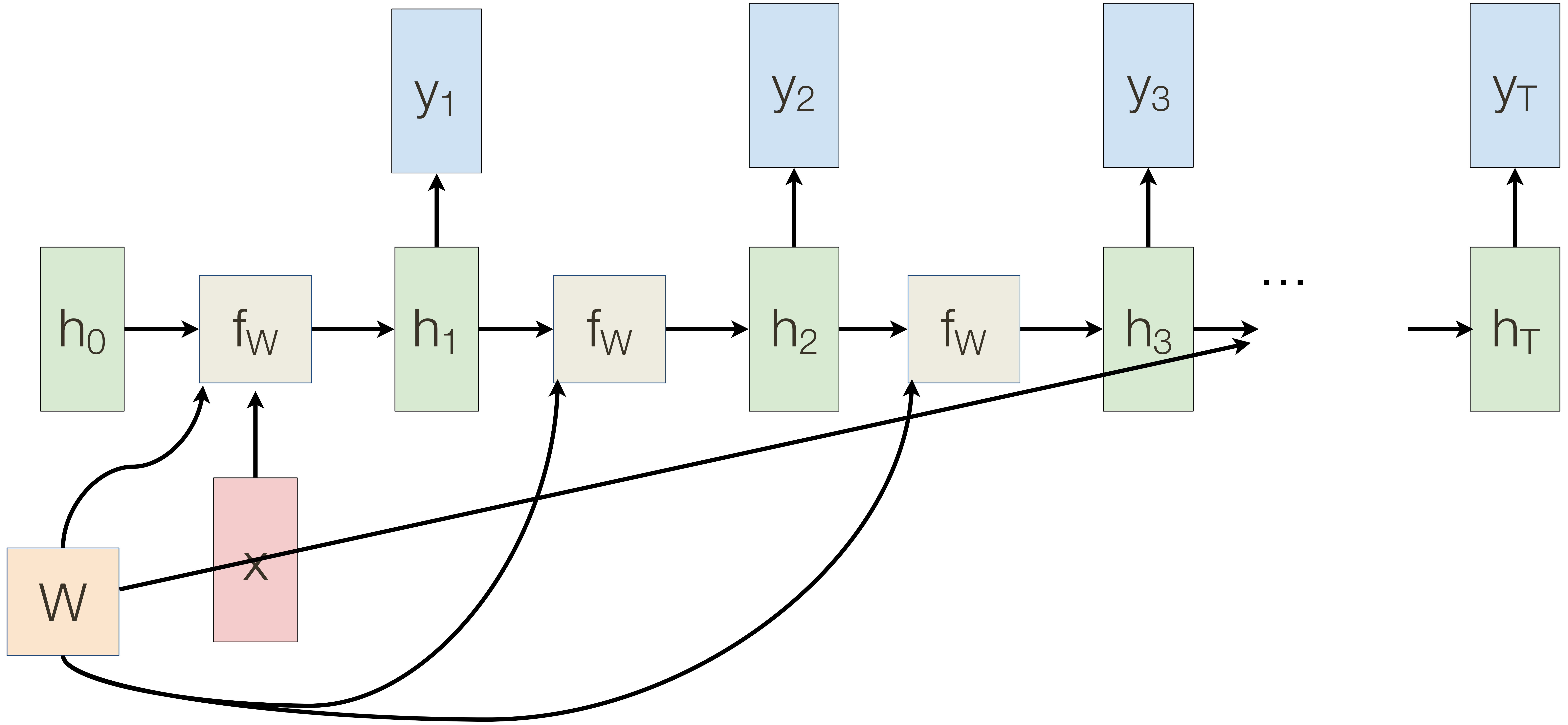

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# RNN **Computational Graph**: Many to One



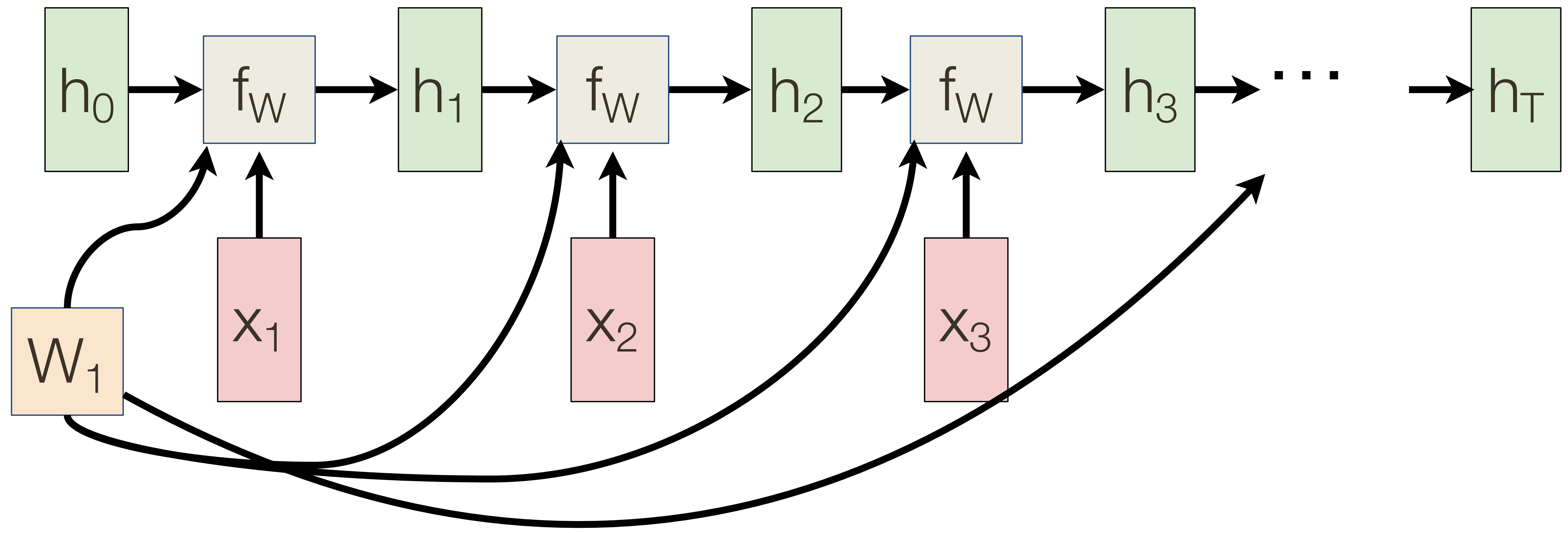

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# RNN **Computational Graph**: One to Many



# **Sequence to Sequence**: Many to One + One to Many

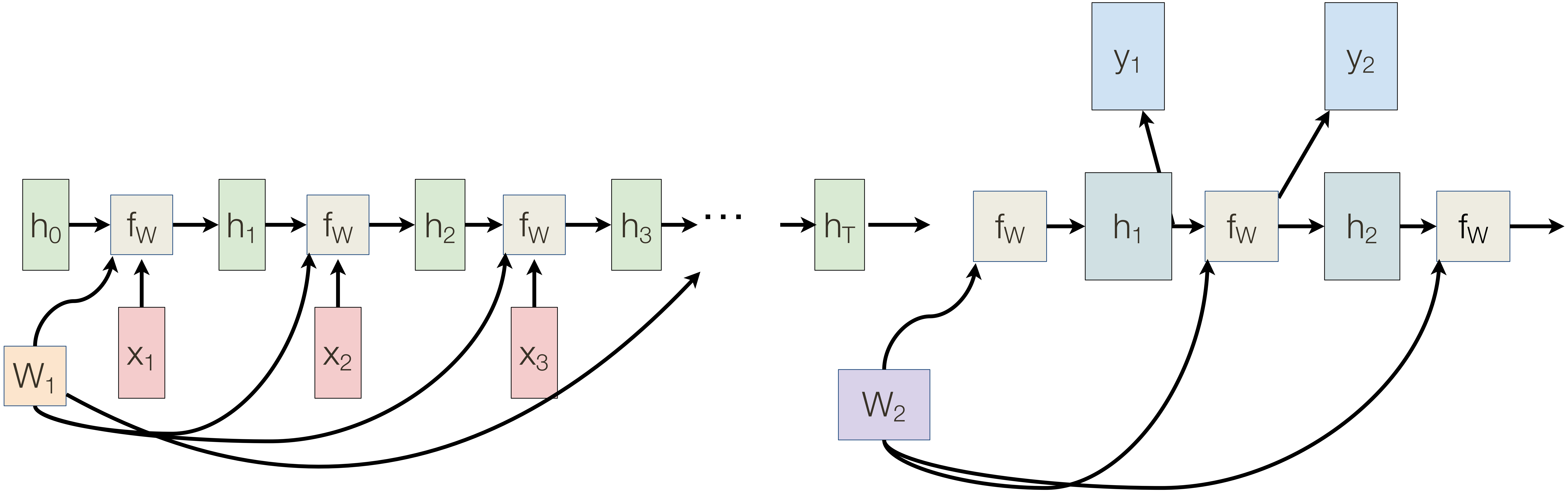
**Many to one:** Encode input sequence in a single vector



# **Sequence to Sequence**: Many to One + One to Many

**Many to one:** Encode input sequence in a single vector

**One to many:** Produce output sequence from single input vector



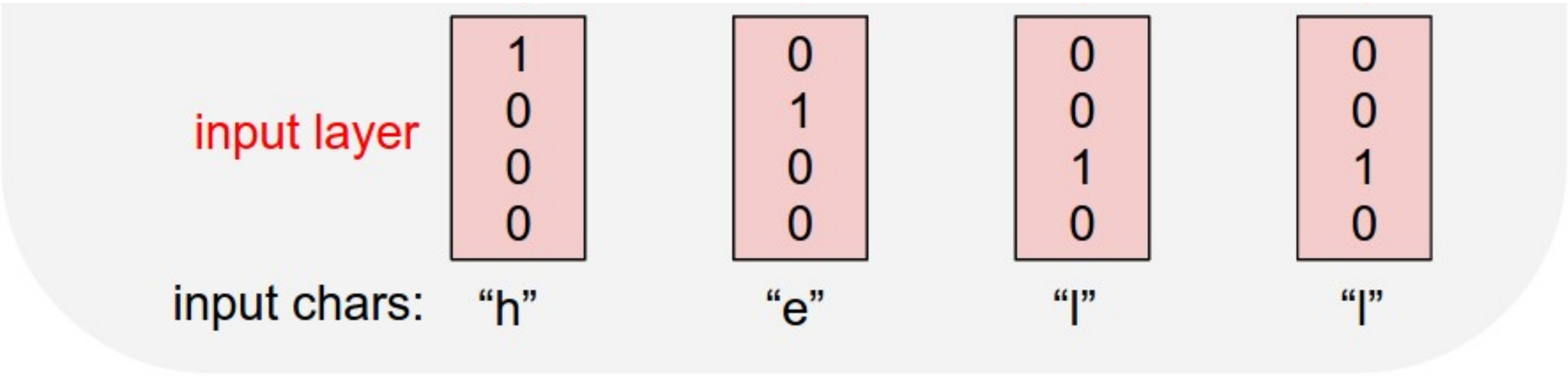
# **Example**: Character-level Language Model

**Vocabulary:**

['h', 'e', 'l', 'o']

Example training sequence:

"hello"



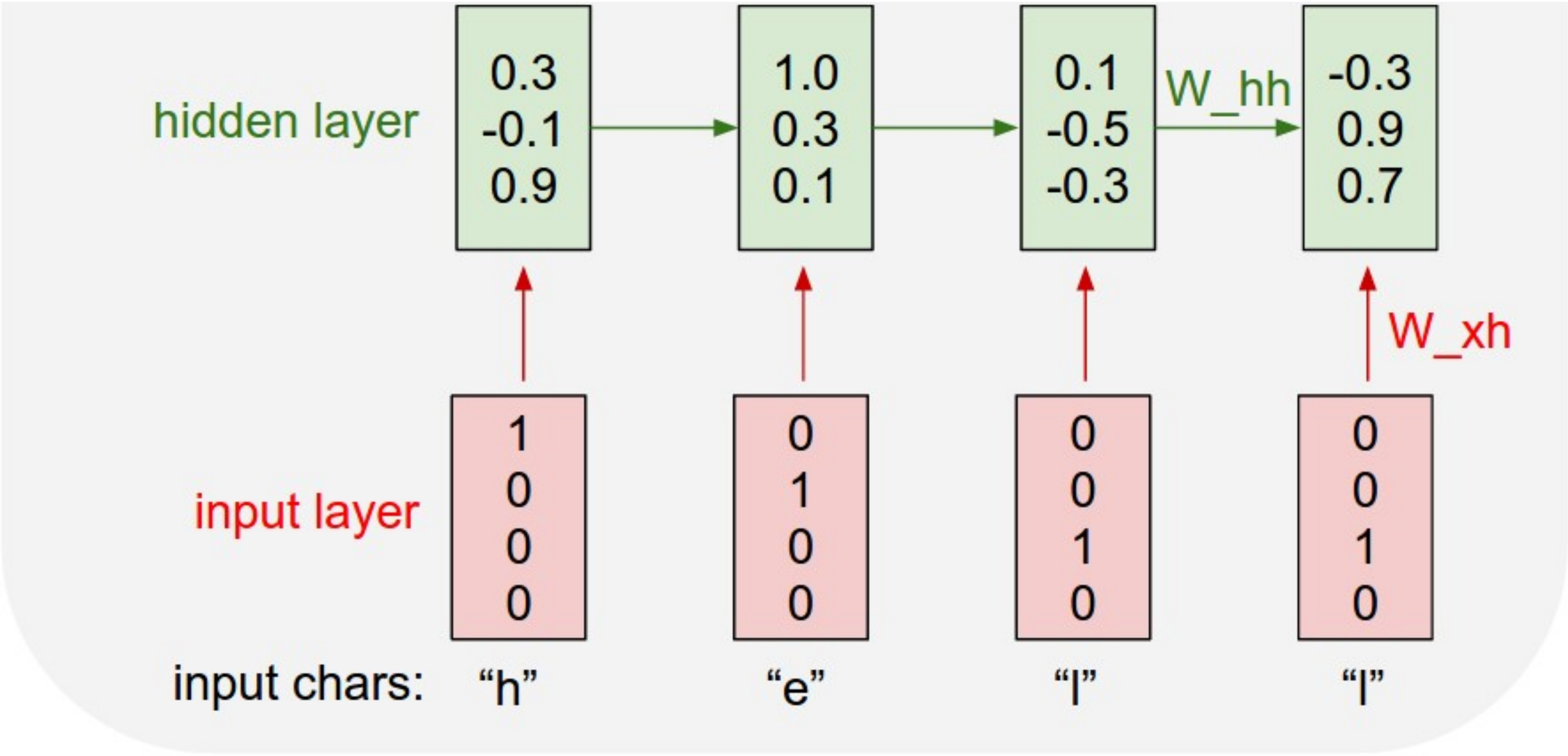

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Example**: Character-level Language Model

**Vocabulary:**
['h', 'e', 'l', 'o']

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t + b_h)$$

Example training sequence:
"hello"

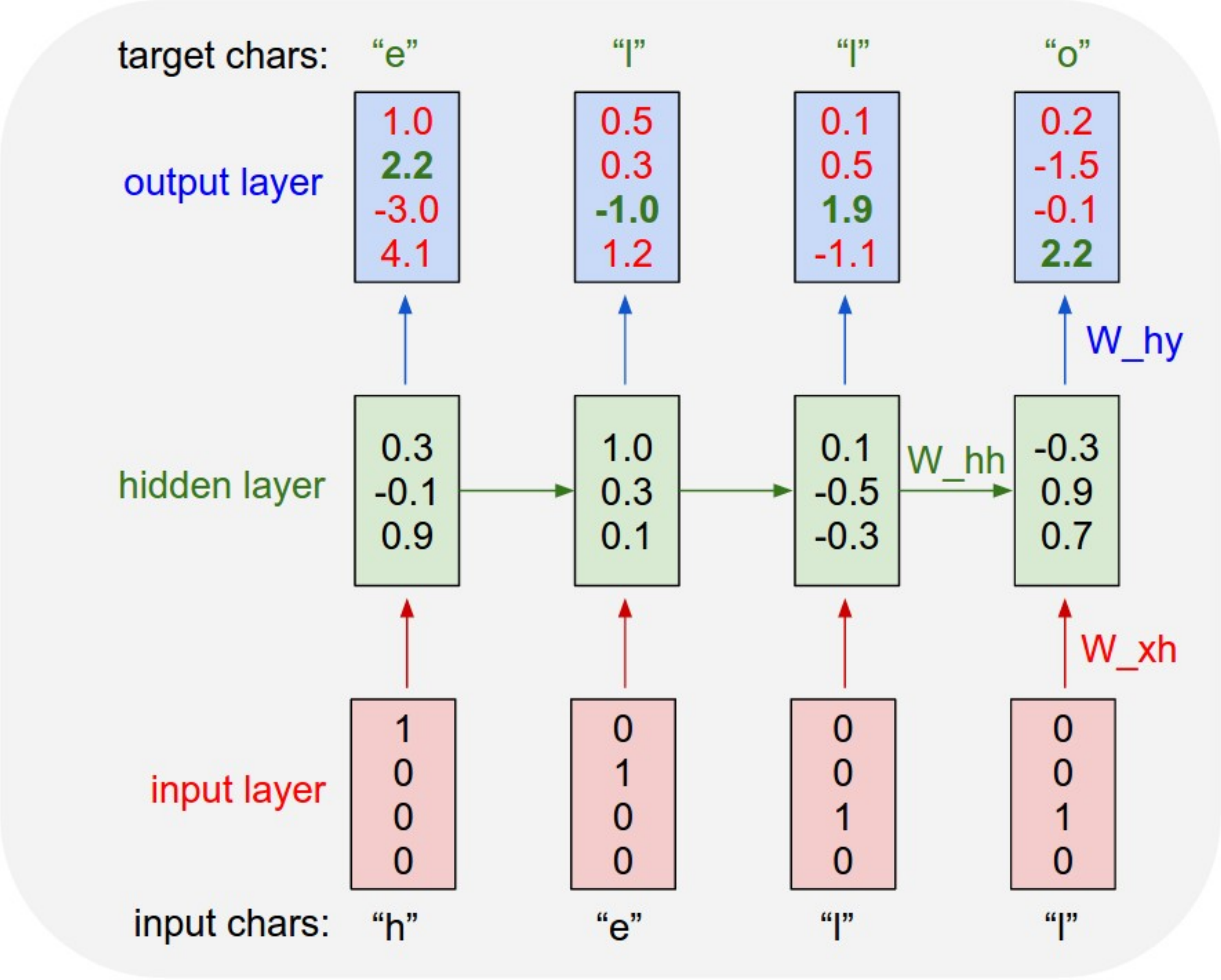



* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Example**: Character-level Language Model

**Vocabulary:**
['h', 'e', 'l', 'o']

Example training sequence:
"hello"



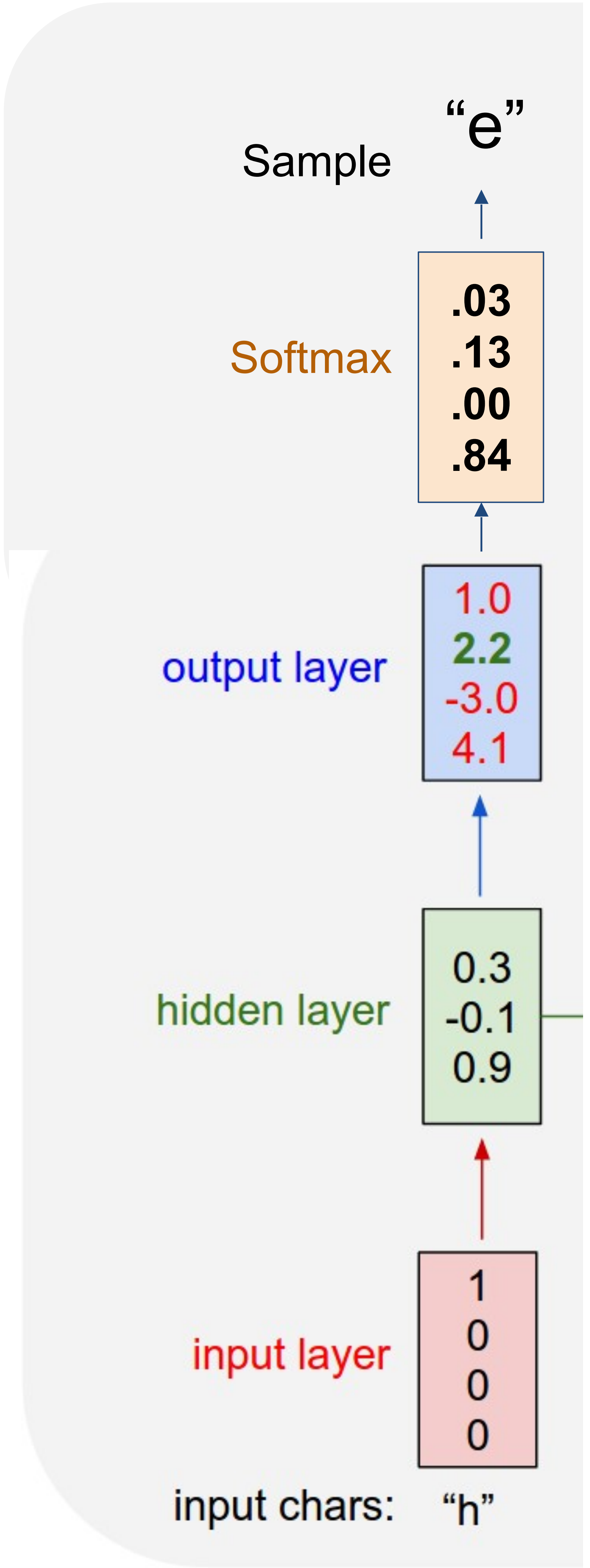

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Example**: Character-level Language Model (**Sampling**)

**Vocabulary:**

['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model



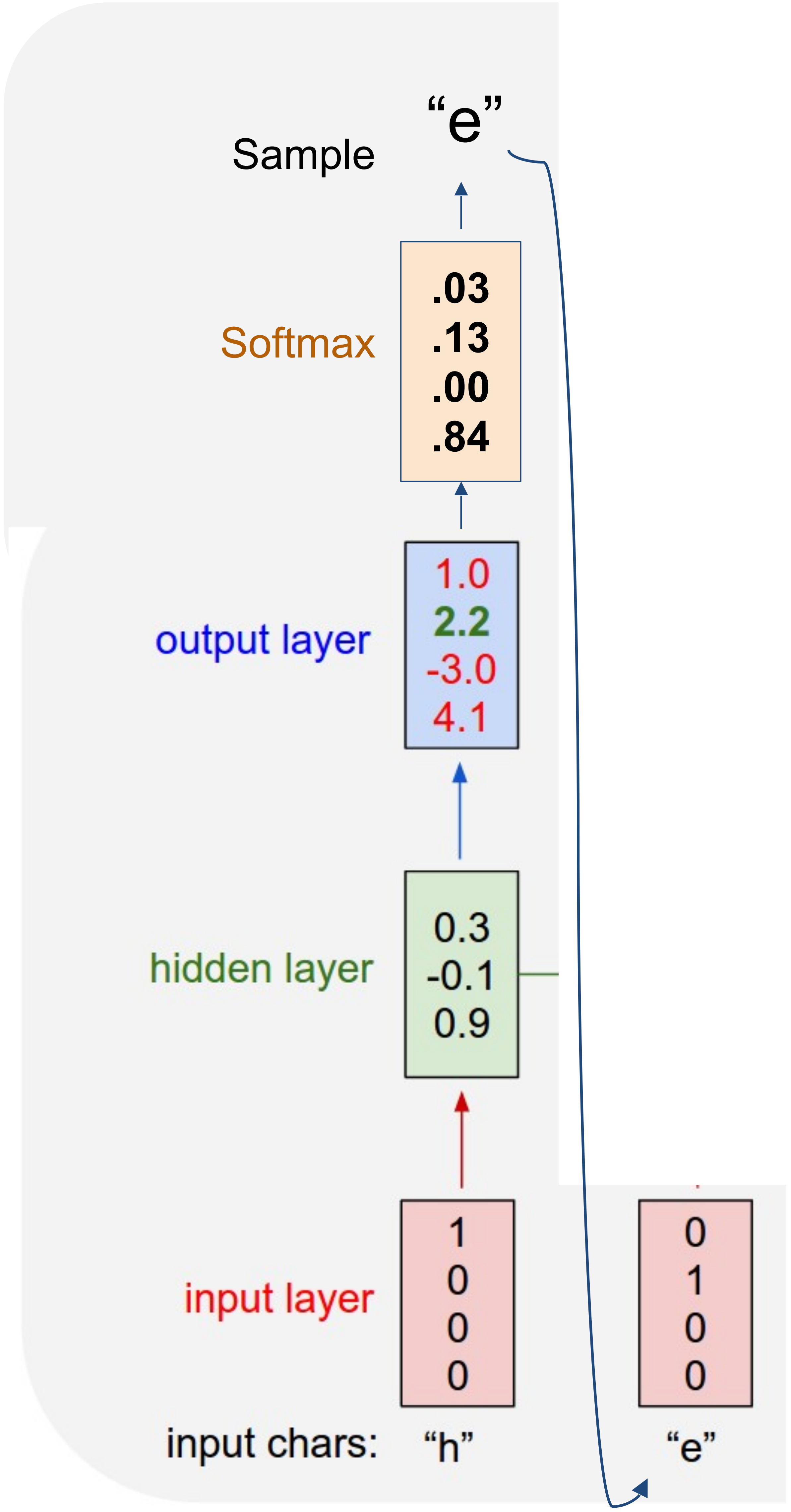

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Example**: Character-level Language Model (**Sampling**)

**Vocabulary:**

['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model



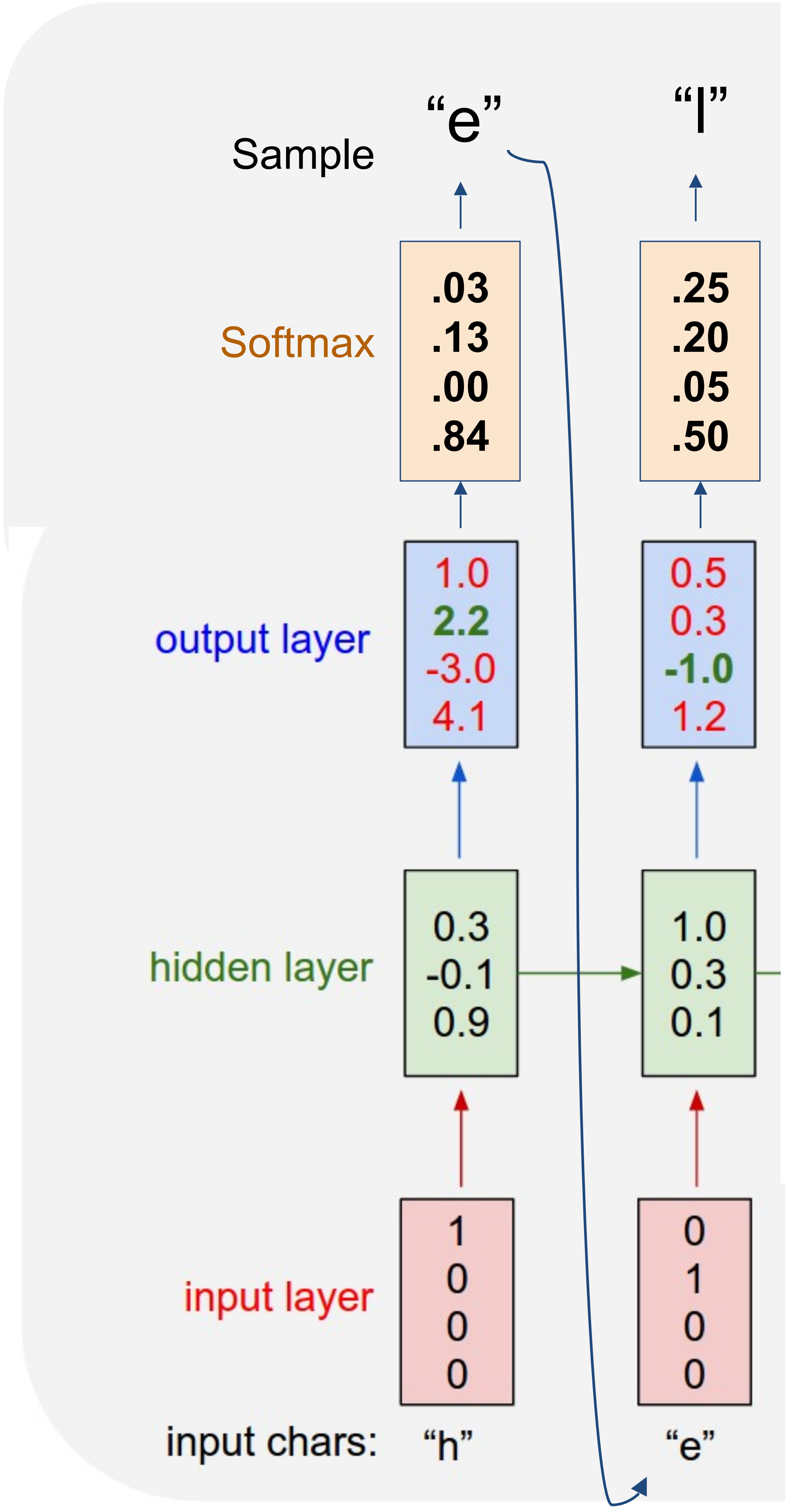

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Example**: Character-level Language Model (**Sampling**)

**Vocabulary:**

['h', 'e', 'l', 'o']

At test time sample one character at a time and feed back to the model

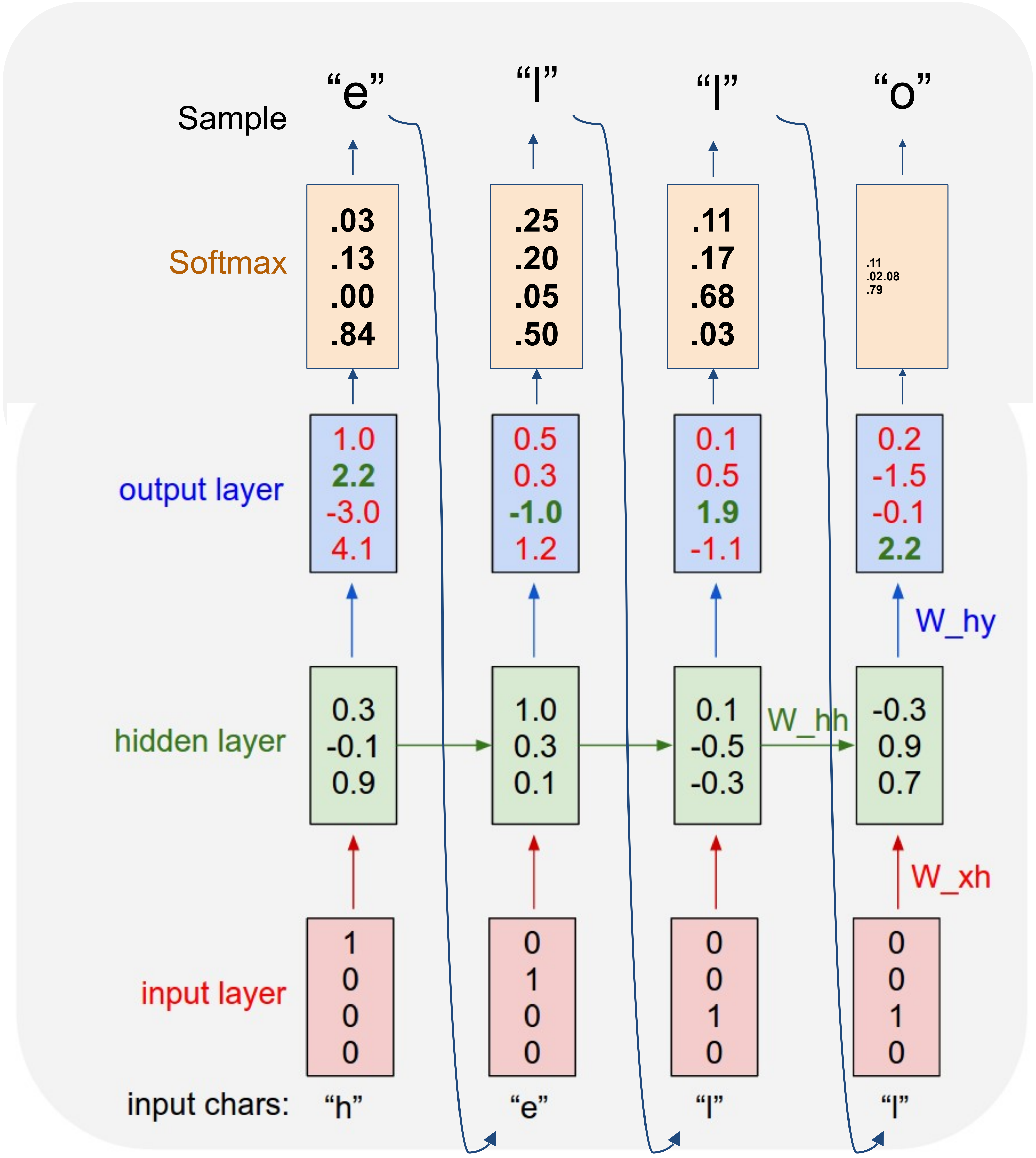


# **Example**: Character-level Language Model (**Sampling**)

**Vocabulary:**

[‘h’, ‘e’, ‘l’, ‘o’]

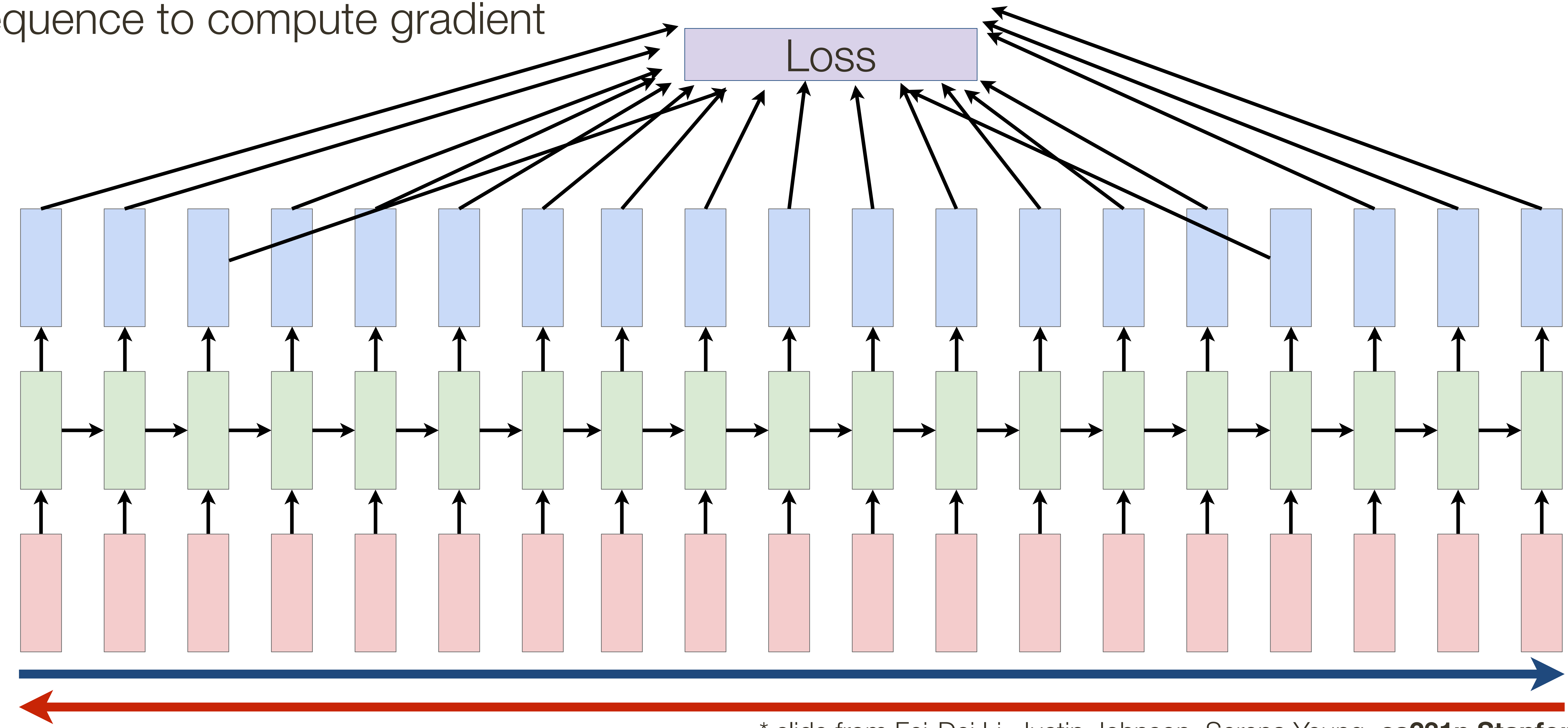
At test time sample one character at a time and feed back to the model




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **BackProp** Through Time

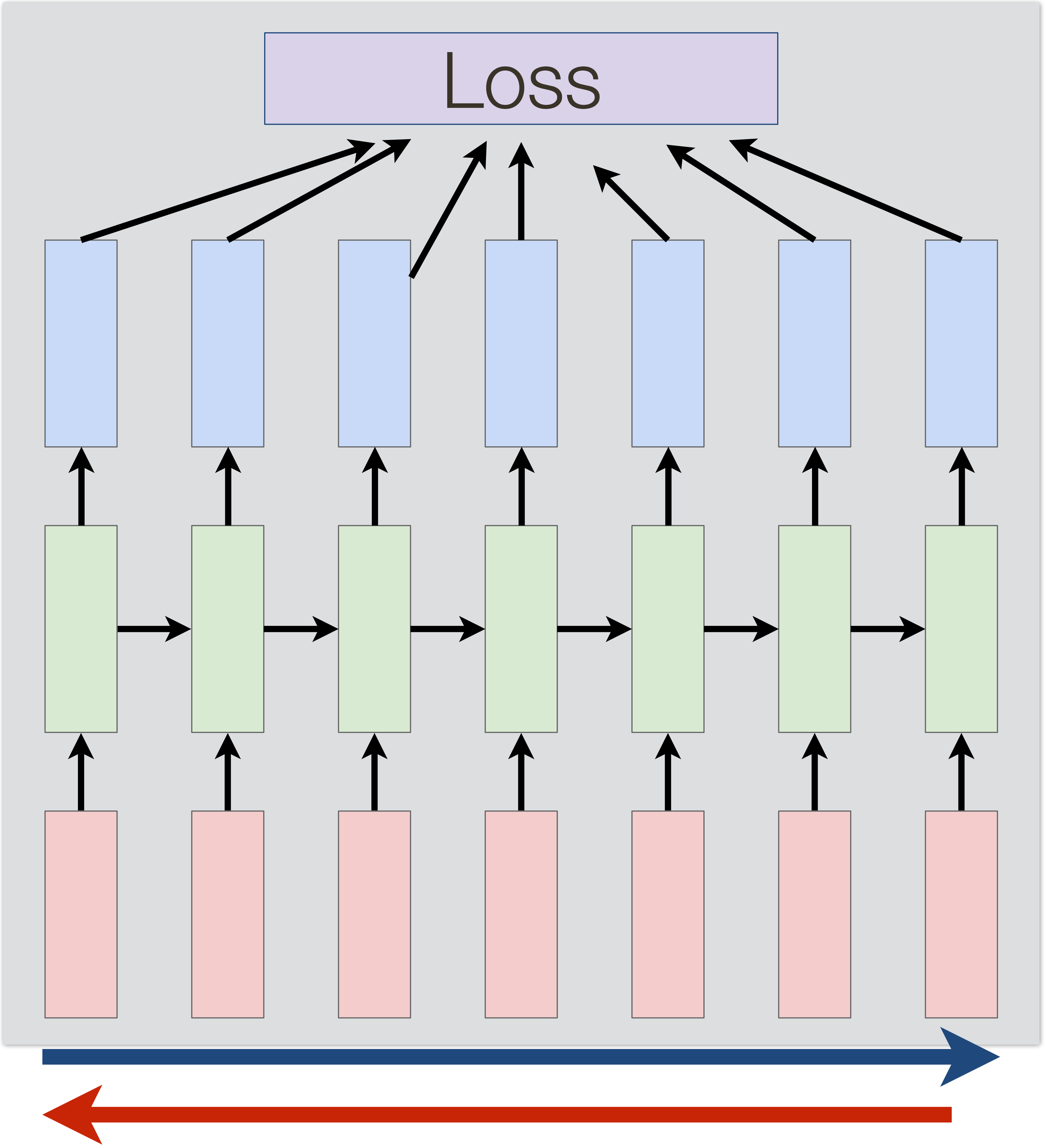
Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Truncated BackProp** Through Time

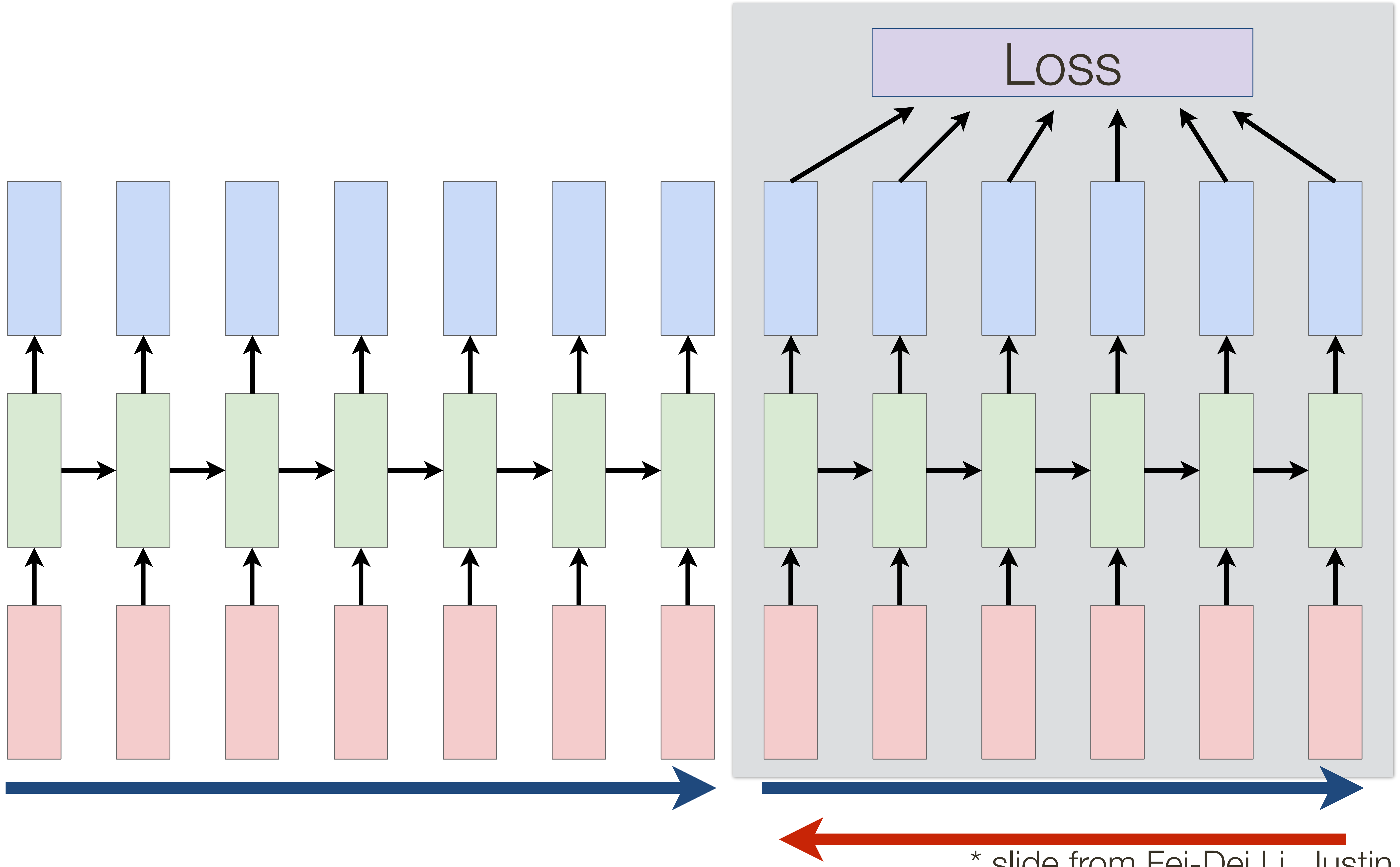
Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Truncated BackProp** Through Time

Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence

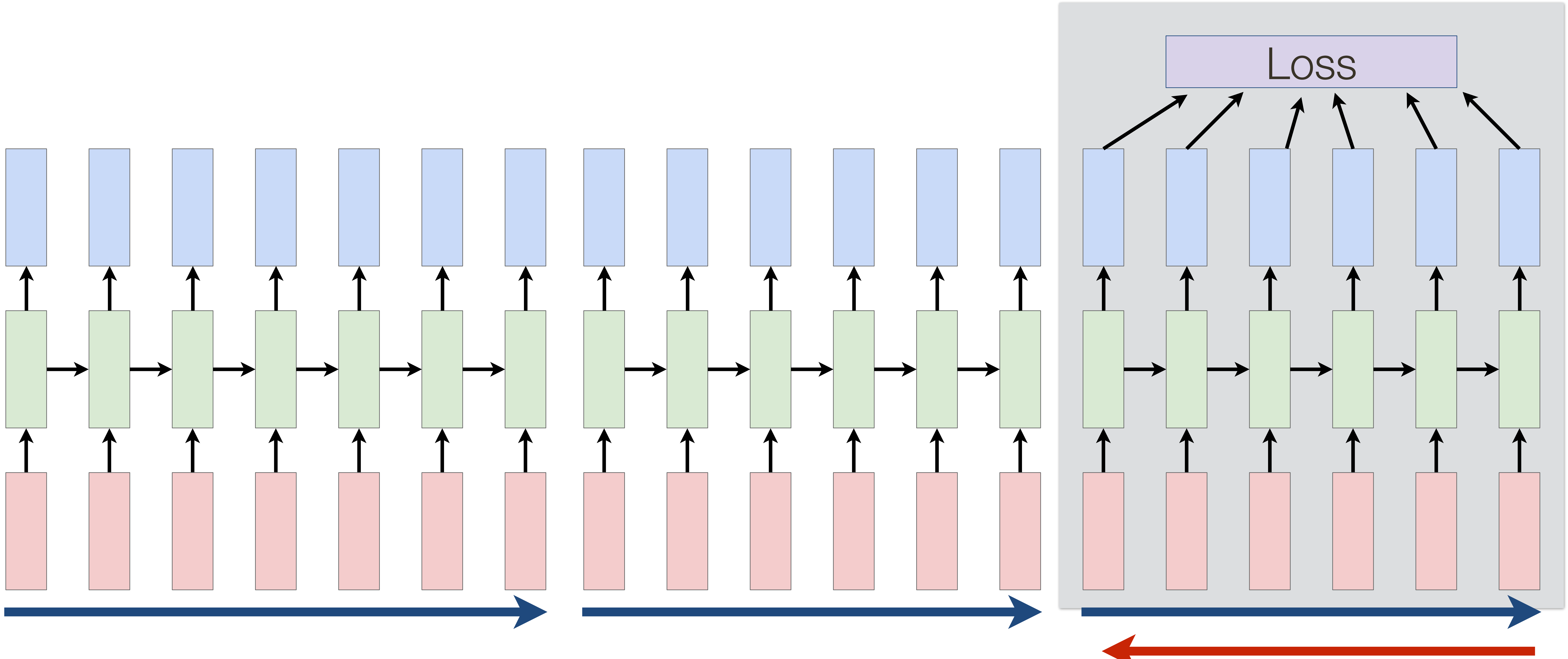



Carry hidden states forward, but only BackProp through some smaller number of steps

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Truncated BackProp** Through Time

Run backwards and forwards through (fixed length) **chunks of the sequence**, instead of the whole sequence




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **Implementation:** Relatively Easy

… you will have a chance to experience this in the **Assignment 3**

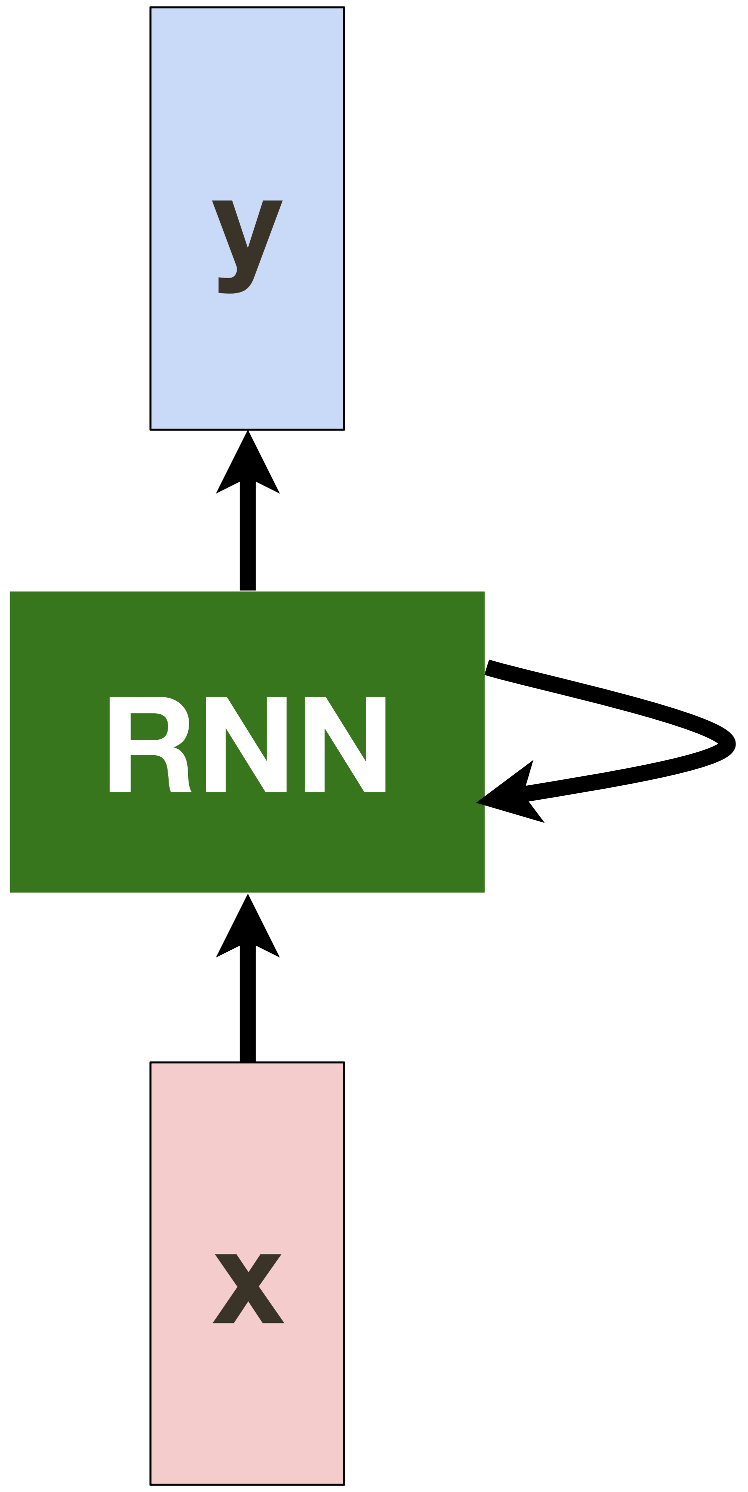
# **Learning to Write** Like Shakespeare

## THE SONNETS

### by William Shakespeare

From fairest creatures we desire increase,
That thereby beauty's rose might never die,
But as the riper should by time decease,
His tender heir might bear his memory:
But thou, contracted to thine own bright eyes,
Feed'st thy light's flame with self-substantial fuel,
Making a famine where abundance lies,
Thyself thy foe, to thy sweet self too cruel:
Thou that art now the world's fresh ornament,
And only herald to the gaudy spring,
Within thine own bud buriest thy content,
And tender churl mak'st waste in niggarding:
  Pity the world, or else this glutton be,
  To eat the world's due, by the grave and thee.

When forty winters shall besiege thy brow,
And dig deep trenches in thy beauty's field,
Thy youth's proud livery so gazed on now,
Will be a tatter'd weed of small worth held:
Then being asked, where all thy beauty lies,
Where all the treasure of thy lusty days;
To say, within thine own deep sunken eyes,
Were an all-eating shame, and thriftless praise.
How much more praise deserv'd thy beauty's use,
If thou couldst answer 'This fair child of mine
Shall sum my count, and make my old excuse,'
Proving his beauty by succession thine!
  This were to be new made when thou art old,
  And see thy blood warm when thou feel'st it cold.



# **Learning to Write** Like Shakespeare … after training a bit

**at first:**

```
tyntd-iafhatawiaoihrdemot  lytdws  e ,tfti, astai f ogoh eoase rrranbyne 'nhthnee e
plia tklrgd t o idoe ns,smtt   h ne etie h,hregtrs nigtike,aoaenns lng
```

↓ train more

```
"Tmont thithey" fomesscerliund
Keushey. Thom here
sheulke, anmerenith ol sivh I lalterthend Bleipile shuwy fil on aseterlome
coaniogennc Phe lism thond hon at. MeiDimorotion in ther thize."
```

↓ train more

```
Aftair fall unsuch that the hall for Prince Velzonski's that me of
her hearly, and behs to so arwage fiving were to it beloge, pavu say falling misfort
how, and Gogition is so overelical and ofter.
```

↓ train more

```
"Why do what that day," replied Natasha, and wishing to himself the fact the
princess, Princess Mary was easier, fed in had oftened him.
Pierre aking his soul came to the packs and drove up his father-in-law women.
```

# **Learning to Write** Like Shakespeare … after training

```
PANDARUS:
Alas, I think he shall be come approached and the day
When little srain would be attain'd into being never fed,
And who is but a chain and subjects of his death,
I should not sleep.

Second Senator:
They are away this miseries, produced upon my soul,
Breaking and strongly should be buried, when I perish
The earth and thoughts of many states.

DUKE VINCENTIO:
Well, your wit is in the care of side and that.

Second Lord:
They would be ruled after this chamber, and
my fair nues begun out of the fact, to be conveyed,
Whose noble souls I'll have the heart of the wars.

Clown:
Come, sir, I will make did behold your worship.

VIOLA:
I'll drink it.
```

```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

# **Learning** Code

Trained on entire source code of Linux kernel

```
static void do_command(struct seq_file *m, void *v)
{
  int column = 32 << (cmd[2] & 0x80);
  if (state)
    cmd = (int)(int_state ^ (in_8(&ch->ch_flags) & Cmd) ? 2 : 1);
  else
    seq = 1;
  for (i = 0; i < 16; i++) {
    if (k & (1 << 1))
      pipe = (in_use & UMXTHREAD_UNCCA) +
        ((count & 0x00000000fffffff8) & 0x000000f) << 8;
    if (count == 0)
      sub(pid, ppc_md.kexec_handle, 0x20000000);
    pipe_set_bytes(i, 0);
  }
  /* Free our user pages pointer to place camera if all dash */
  subsystem_info = &of_changes[PAGE_SIZE];
  rek_controls(offset, idx, &soffset);
  /* Now we want to deliberately put it to device */
  control_check_polarity(&context, val, 0);
  for (i = 0; i < COUNTER; i++)
    seq_puts(s, "policy ");
}
```

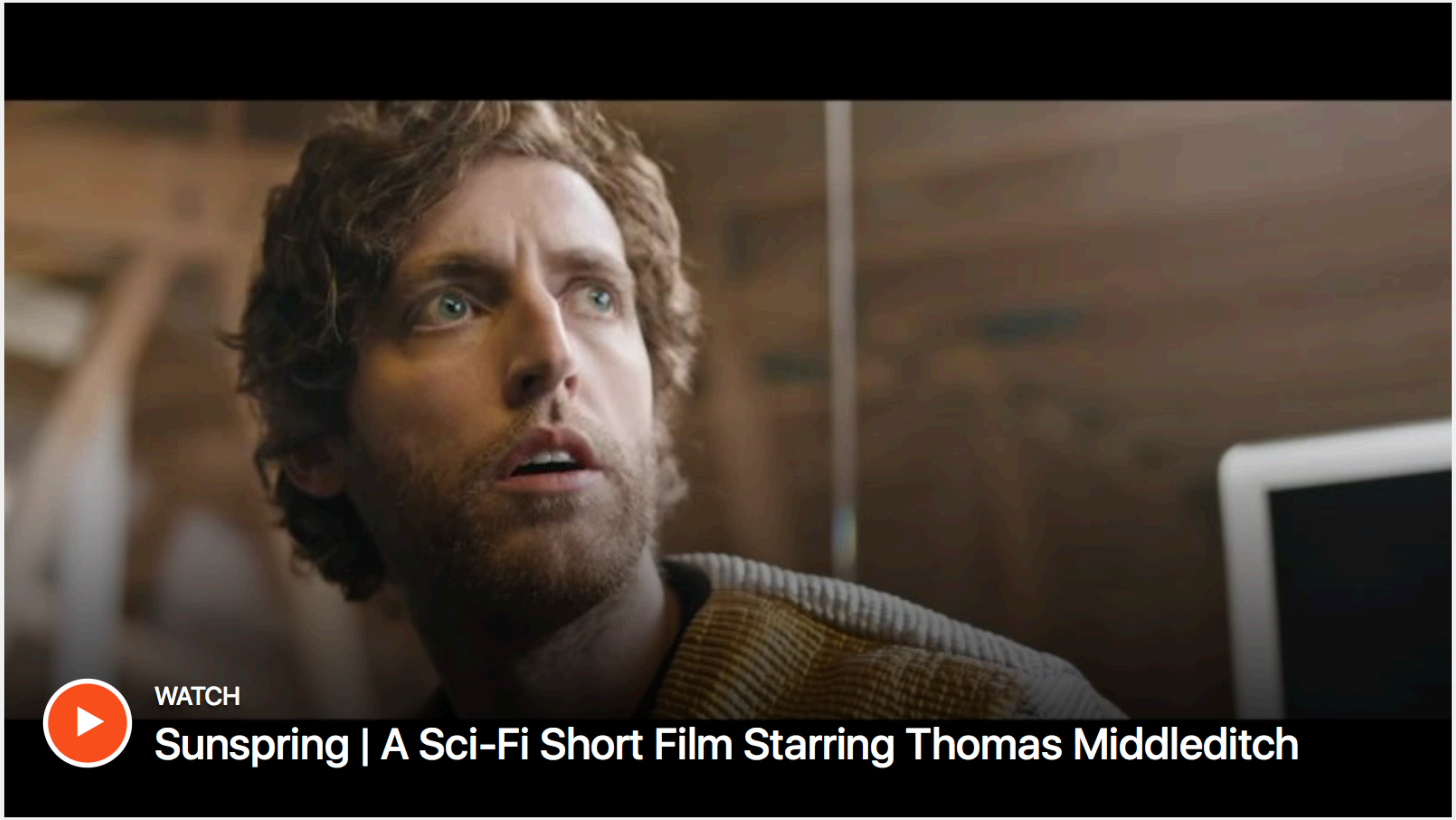
* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# **DopeLearning:** Computational Approach to Rap Lyrics

| | |
|---|---|
| Everybody got one | (2 Chainz - Extremely Blessed) |
| And all the pretty mommies want some | (Mos Def - Undeniable) |
| And what i told you all was | (Lil Wayne - Welcome Back) |
| But you need to stay such do not touch | (Common - Heidi Hoe) |
| They really do not want you to vote | (KRS One - The Mind) |
| what do you condone | (Cam'ron - Bubble Music) |
| Music make you lose control | (Missy Elliot - Lose Control) |
| What you need is right here ahh oh | (Wiz Khalifa - Right Here) |
| This is for you and me | (Missy Elliot - Hit Em Wit Da Hee) |
| I had to dedicate this song to you Mami | (Fat Joe - Bendicion Mami) |
| Now I see how you can be | (Lil Wayne - How To Hate) |
| I see u smiling i kno u hattig | (Wiz Khalifa - Damn Thing) |
| Best I Eva Had x4 | (Nicki Minaj - Best I Ever Had) |
| That I had to pay for | (Ice Cube - X Bitches) |
| Do I have the right to take yours | (Common - Retrospect For Life) |
| Trying to stay warm | (Everlast - 2 Pieces Of Drama) |

[ Malmi et al., KDD 2016 ]

# **Sunspring**: First movie generated by AI

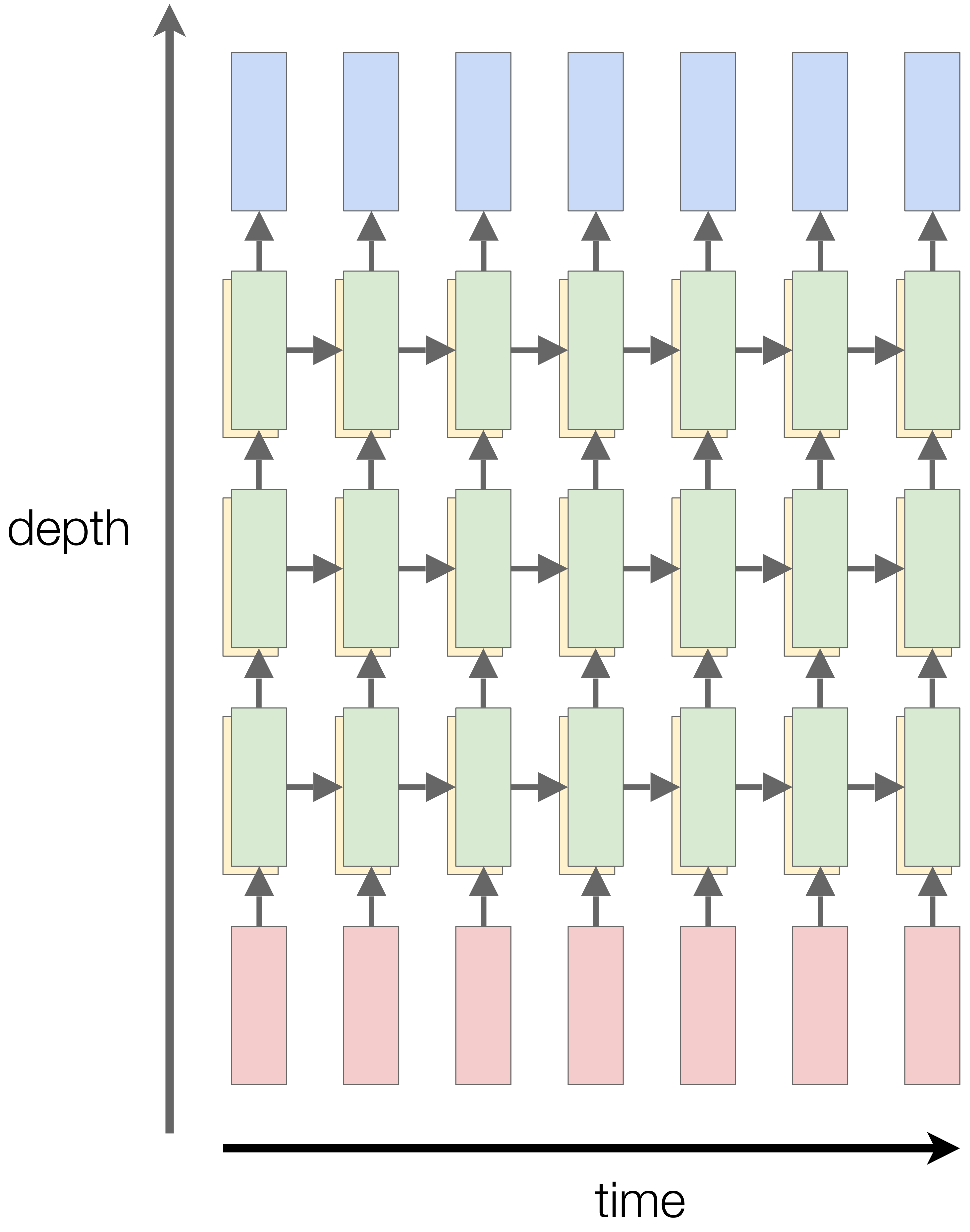



*Sunspring*, a short science fiction movie written entirely by AI, debuts exclusively on Ars today.

# **Multilayer** RNNs

$$h_t^l = \tanh W^l \begin{pmatrix} h_t^{l-1} \\ h_{t-1}^l \end{pmatrix}$$

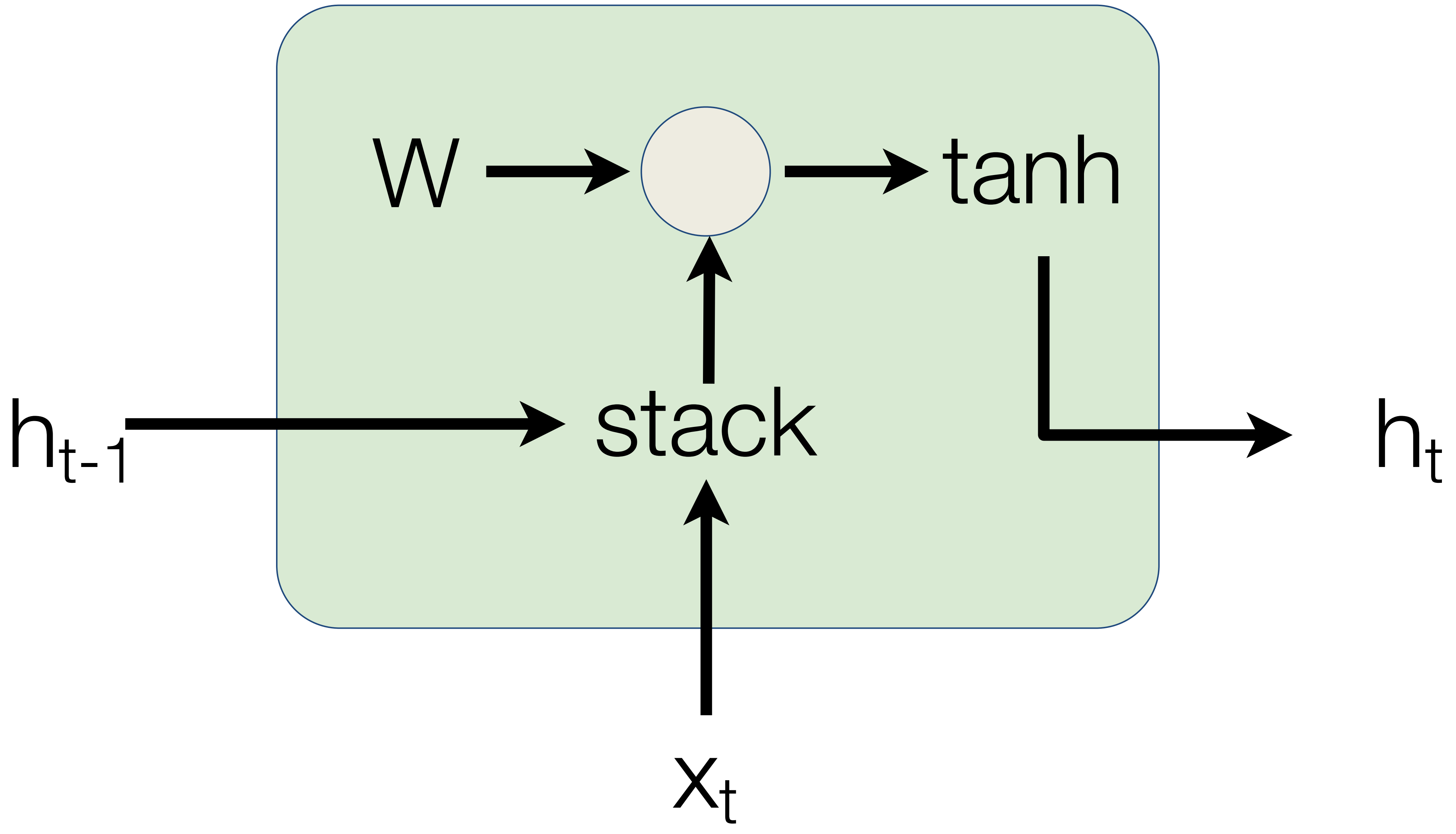
$h \in \mathbb{R}^n$ $\quad W^l \ [n \times 2n]$




* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]

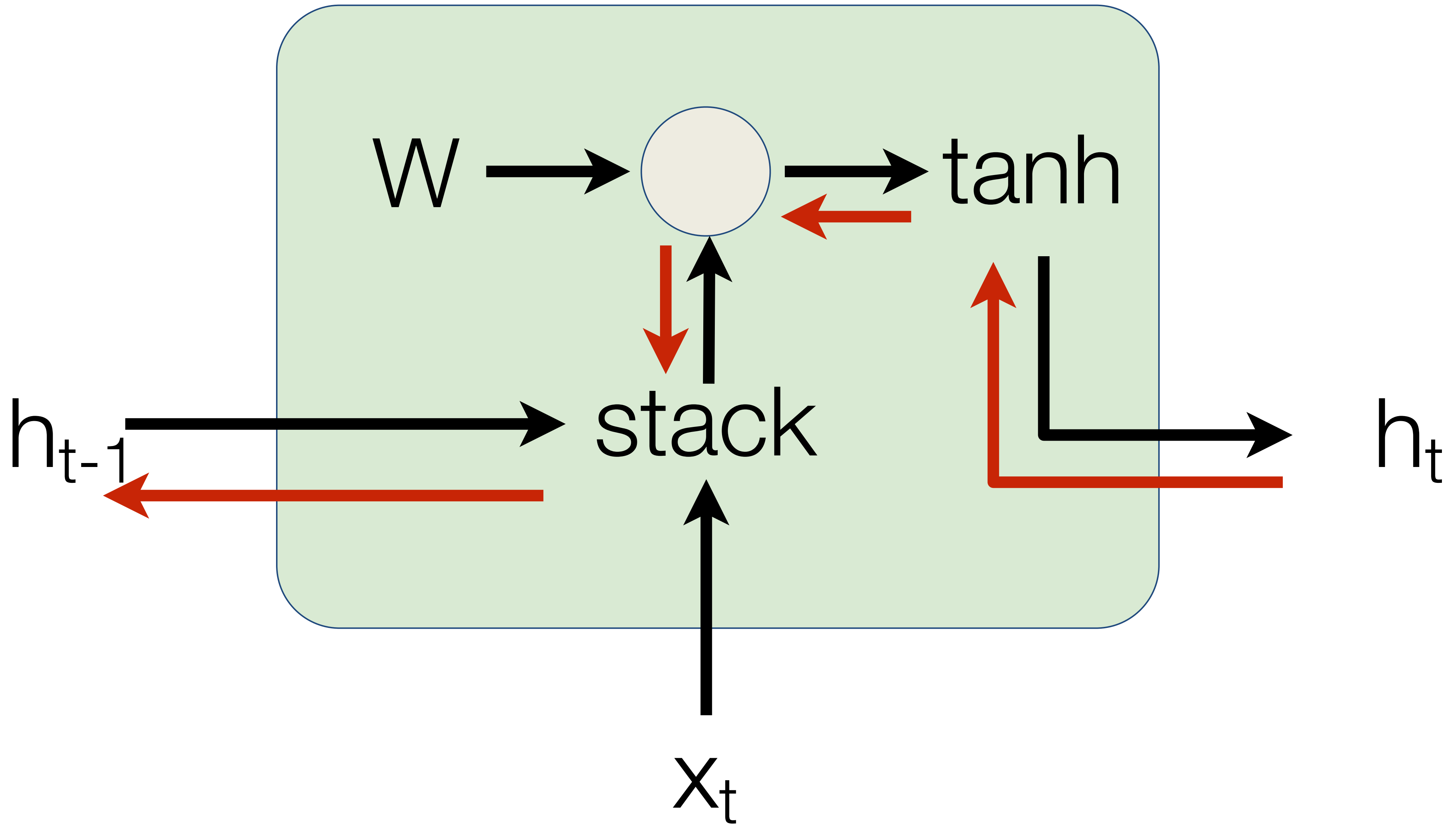



$$
\begin{aligned}
h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\
&= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\
&= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)
\end{aligned}
$$

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]

Backpropagation from $h_t$ to $h_{t-1}$ multiplies by W (actually $W_{hh}^T$)

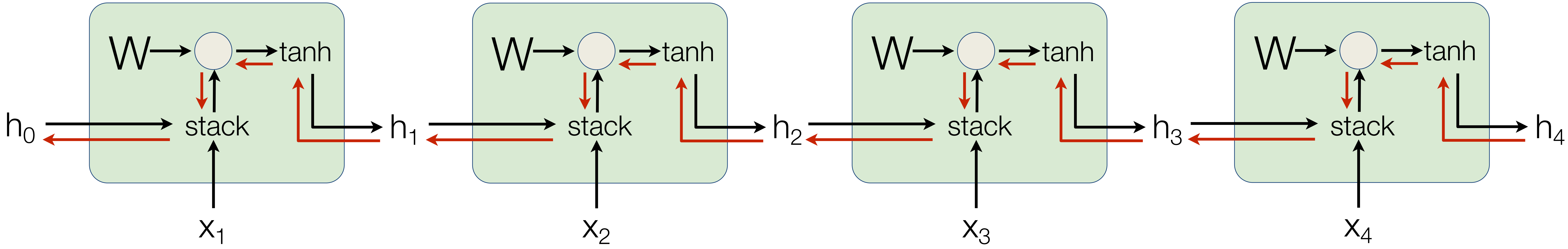



$$\begin{aligned} h_t &= \tanh(W_{hh}h_{t-1} + W_{xh}x_t) \\ &= \tanh\left(\begin{pmatrix} W_{hh} & W_{hx} \end{pmatrix} \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \\ &= \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right) \end{aligned}$$

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]

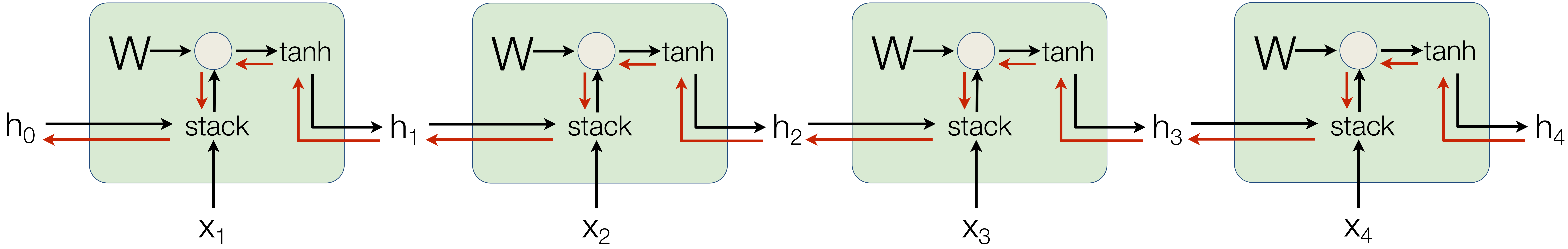
[ Pascanu et al., ICML 2013 ]




Computing gradient of $h_0$ involves many factors of W (and repeated tanh)

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]



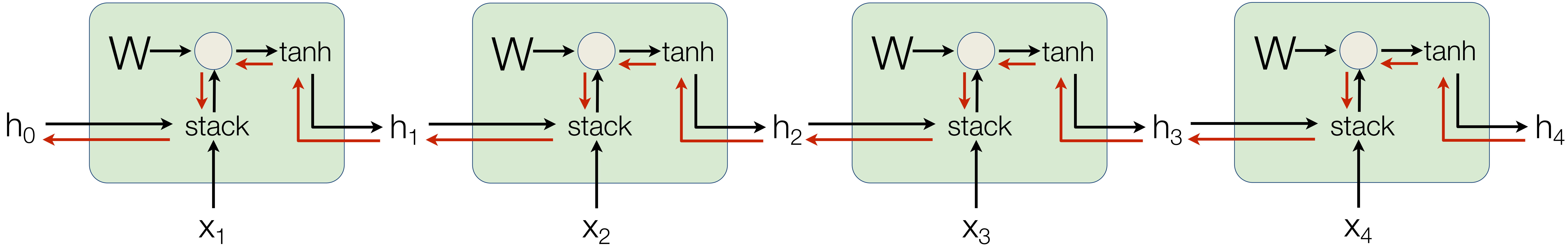

Computing gradient of $h_0$ involves many factors of W (and repeated tanh)

Largest singular value > 1:
**Exploding gradients**

Largest singular value < 1:
**Vanishing gradients**

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]




Computing gradient of $h_0$ involves many factors of W (and repeated tanh)

Largest singular value > 1:
**Exploding gradients**

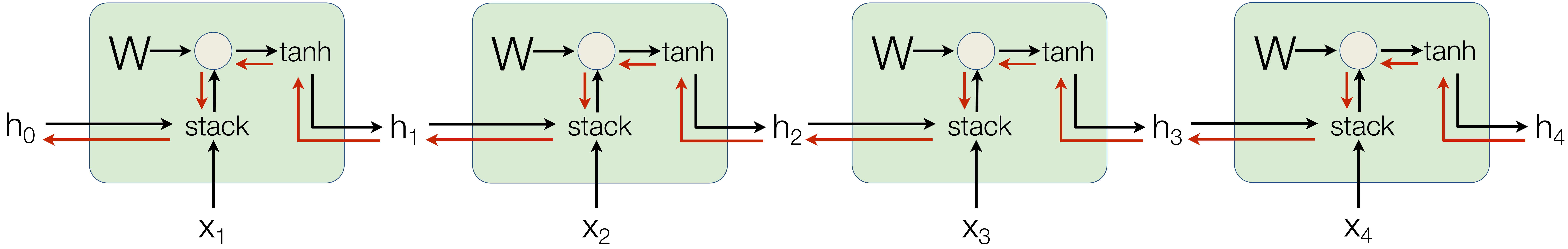
Largest singular value < 1:
**Vanishing gradients**

**Gradient clipping:** Scale gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
  grad *= (threshold / grad_norm)
```

# Vanilla RNN **Gradient Flow**

[ Bengio et al., 1994 ]
[ Pascanu et al., ICML 2013 ]




Computing gradient of $h_0$ involves many factors of W (and repeated tanh)

Largest singular value > 1:
**Exploding gradients**

Largest singular value < 1:
**Vanishing gradients**

Change RNN architecture

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# Long-Short Term Memory (**LSTM**)

**Vanilla** RNN

$$h_t = \tanh\left(W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}\right)$$

**LSTM**

$$\begin{pmatrix} i \\ f \\ o \\ g \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} W \begin{pmatrix} h_{t-1} \\ x_t \end{pmatrix}$$

$$c_t = f \odot c_{t-1} + i \odot g$$

$$h_t = o \odot \tanh(c_t)$$

[ Hochreiter and Schmidhuber, NC **1977** ]

* slide from Fei-Dei Li, Justin Johnson, Serena Yeung, **cs231n Stanford**

# Long-Short Term Memory (**LSTM**)

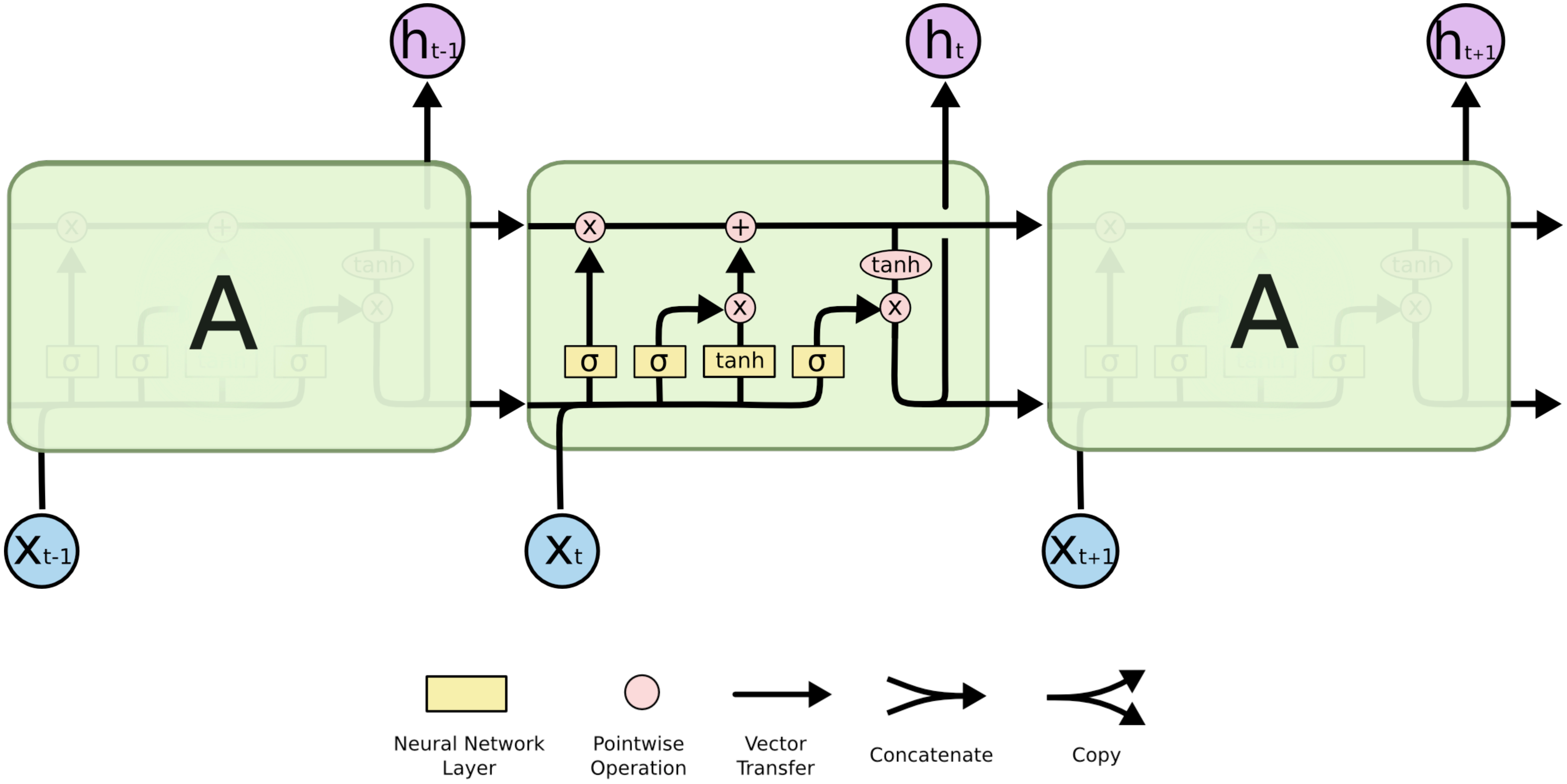



Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# Long-Short Term Memory (**LSTM**)

Cell state / **memory**

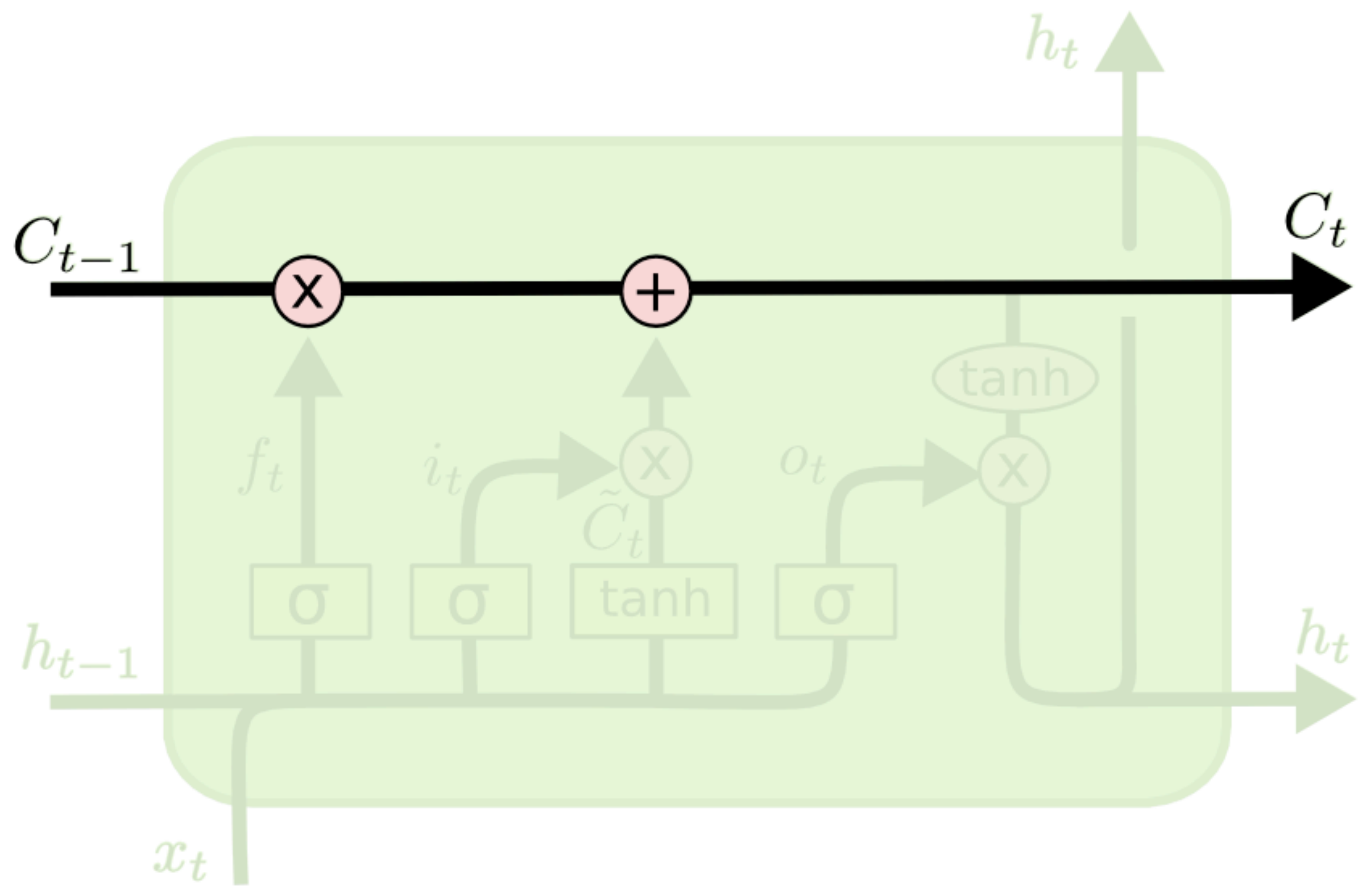
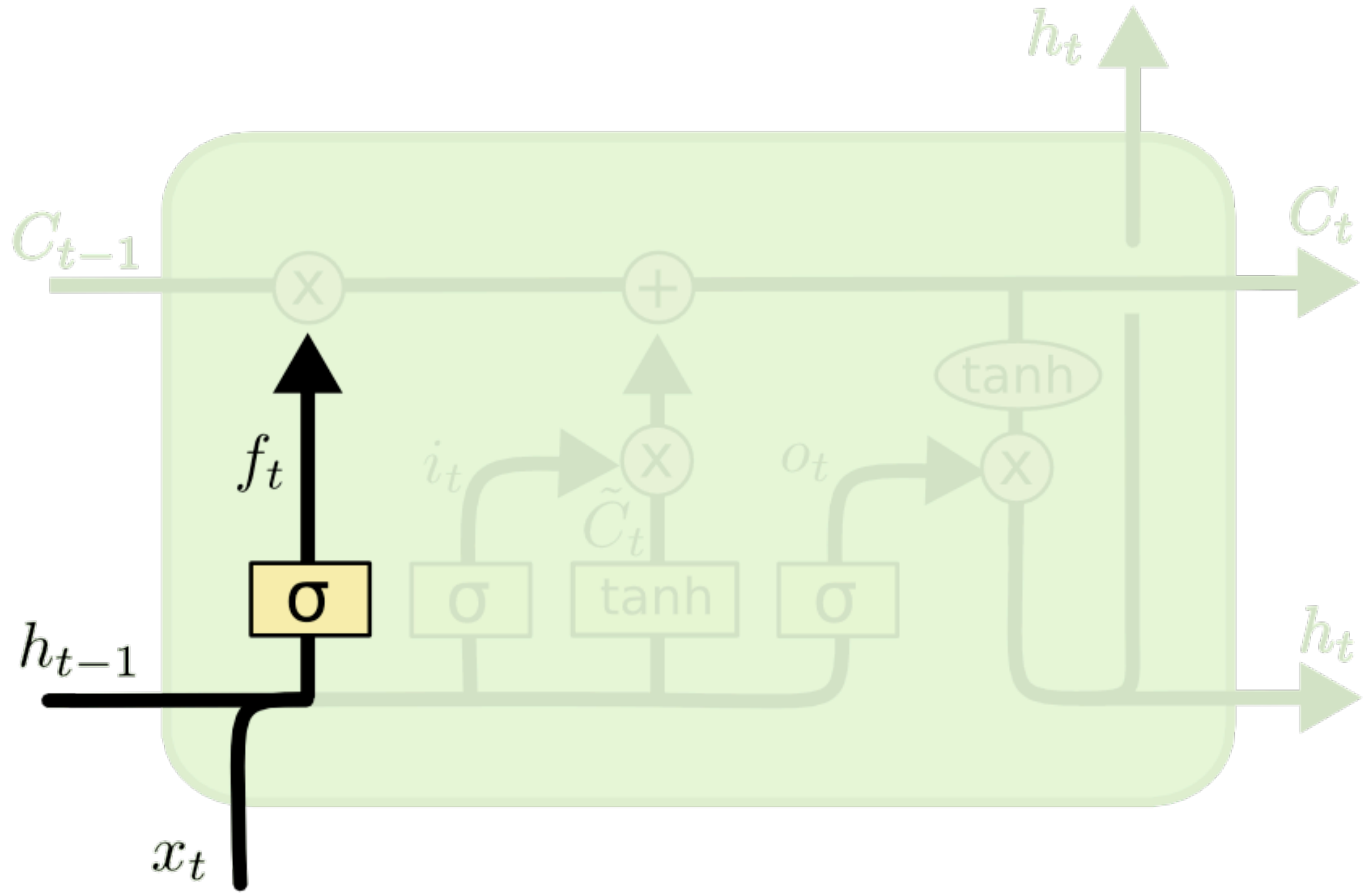



Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Forget Gate

Should we continue to **remember** this “bit” of information or not?



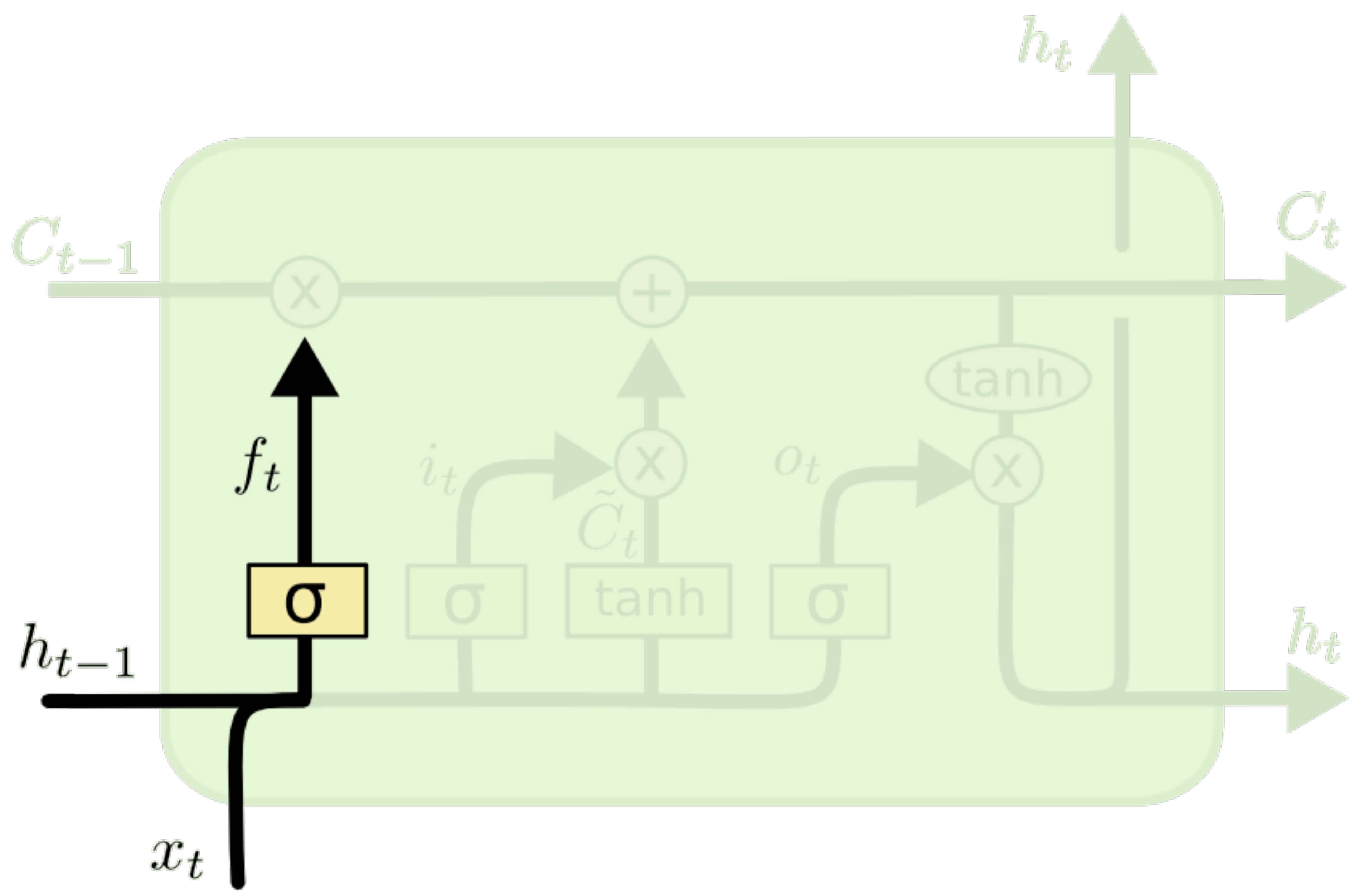

$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Forget Gate

Should we continue to **remember** this “bit” of information or not?




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \ + \ b_f\right)$$

**Intuition:** memory and forget gate output multiply, output of forget gate can be though of as binary (0 or 1)

anything x 1 = anything (remember)

anything x 0 = 0 (forget)

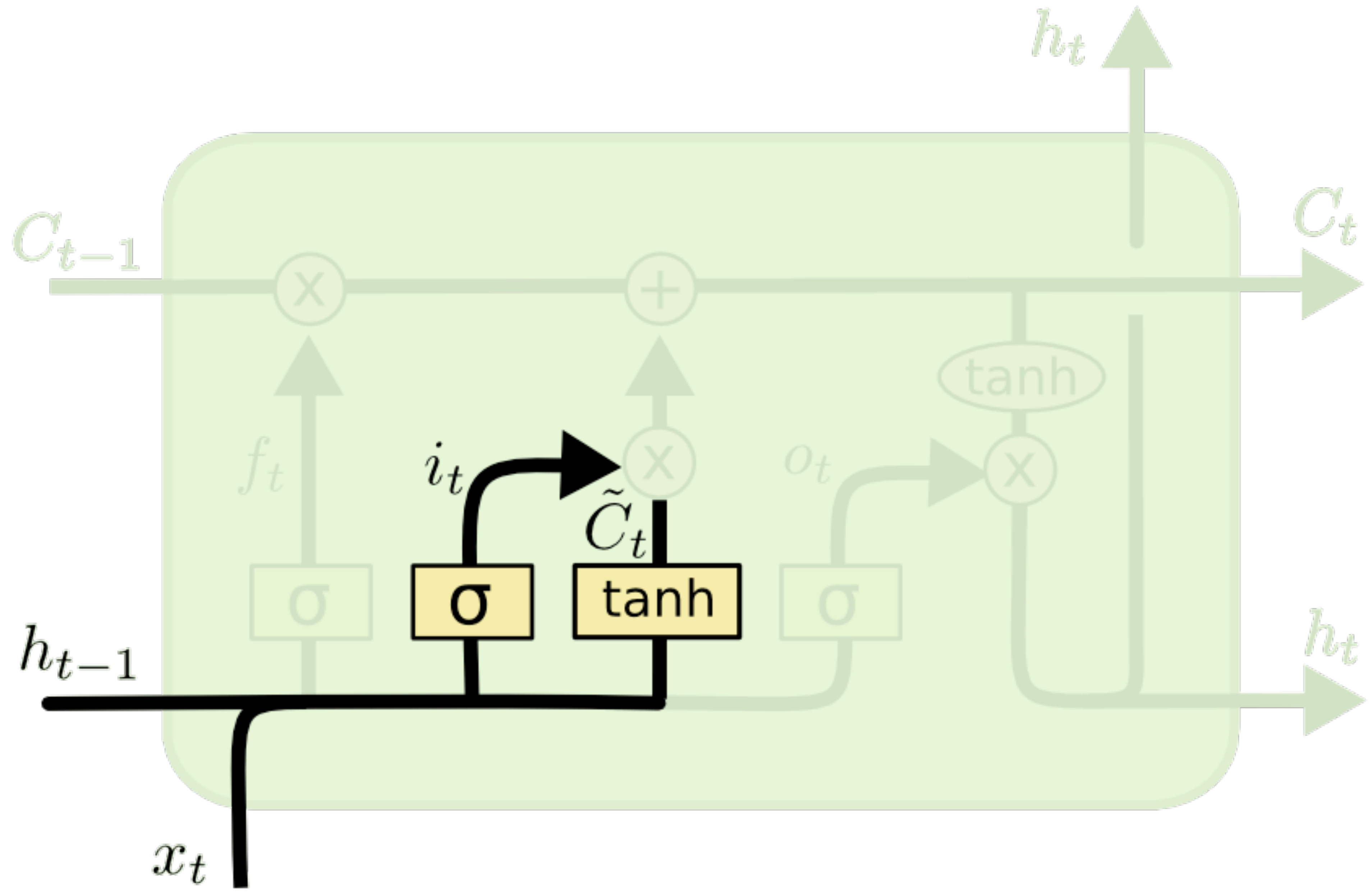
Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Input Gate

Should we **update** this "bit" of information or not?

If yes, then what should we **remember**?




$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \ + \ b_i\right)$$

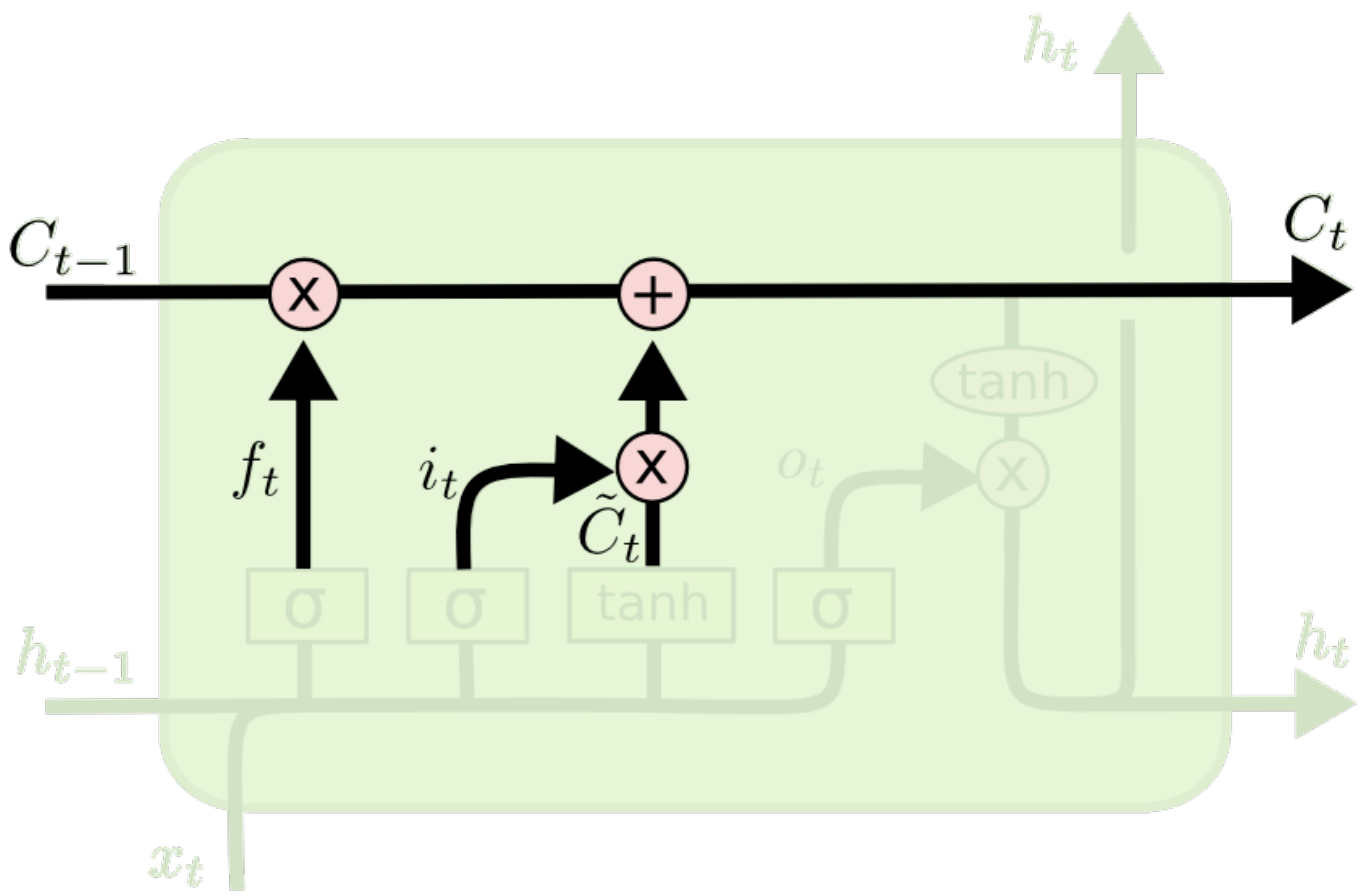
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \ + \ b_C)$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Memory Update

Forget what needs to be forgotten + memorize what needs to be remembered



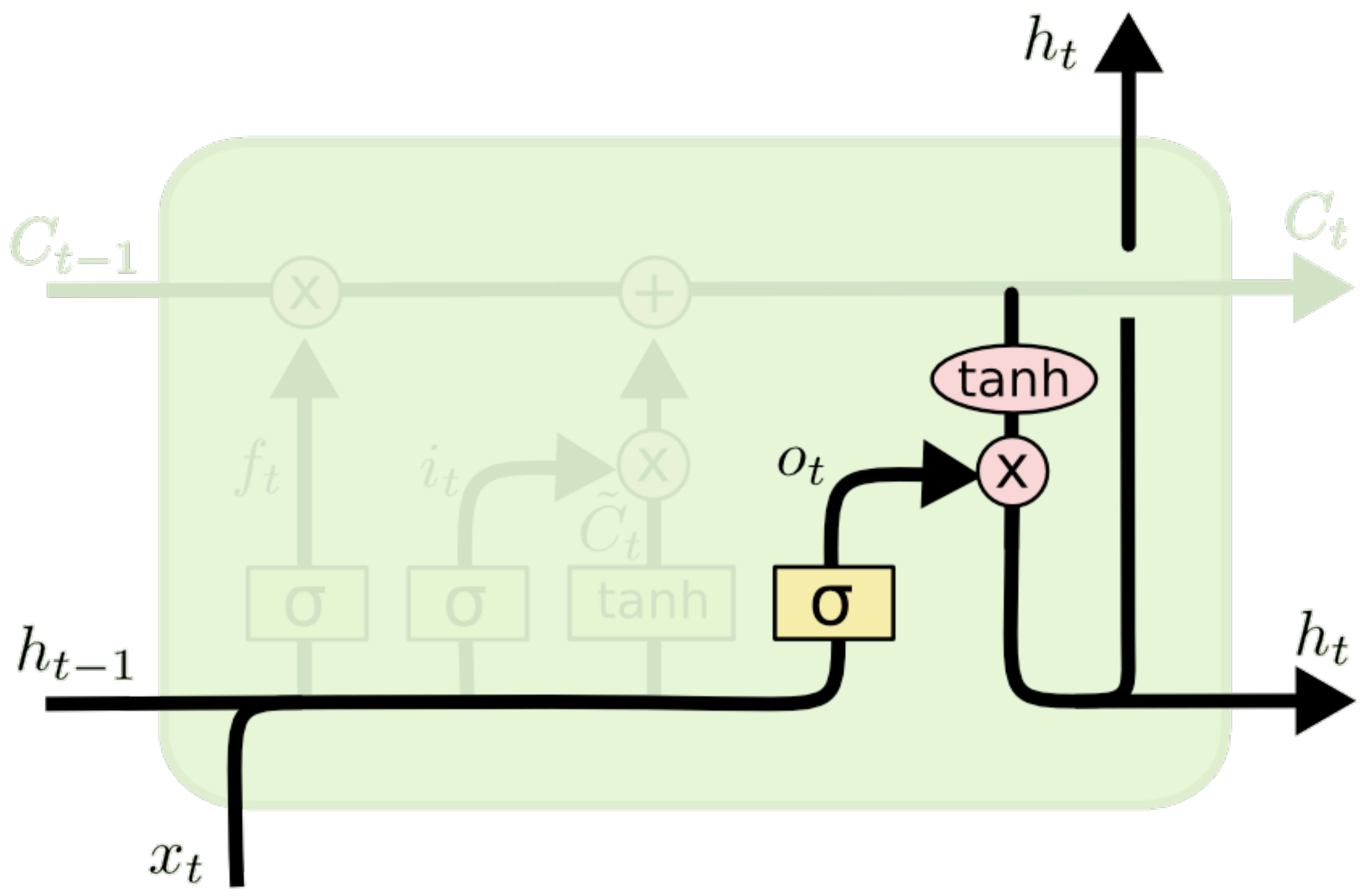

$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Output Gate

Should we output this bit of information (e.g., to "deeper" LSTM layers)?




$$o_t = \sigma\left(W_o\ [h_{t-1}, x_t]\ +\ b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Additive Updates

Backpropagation from $c_t$ to $c_{t-1}$ only elementwise multiplication by f, no matrix multiply by W

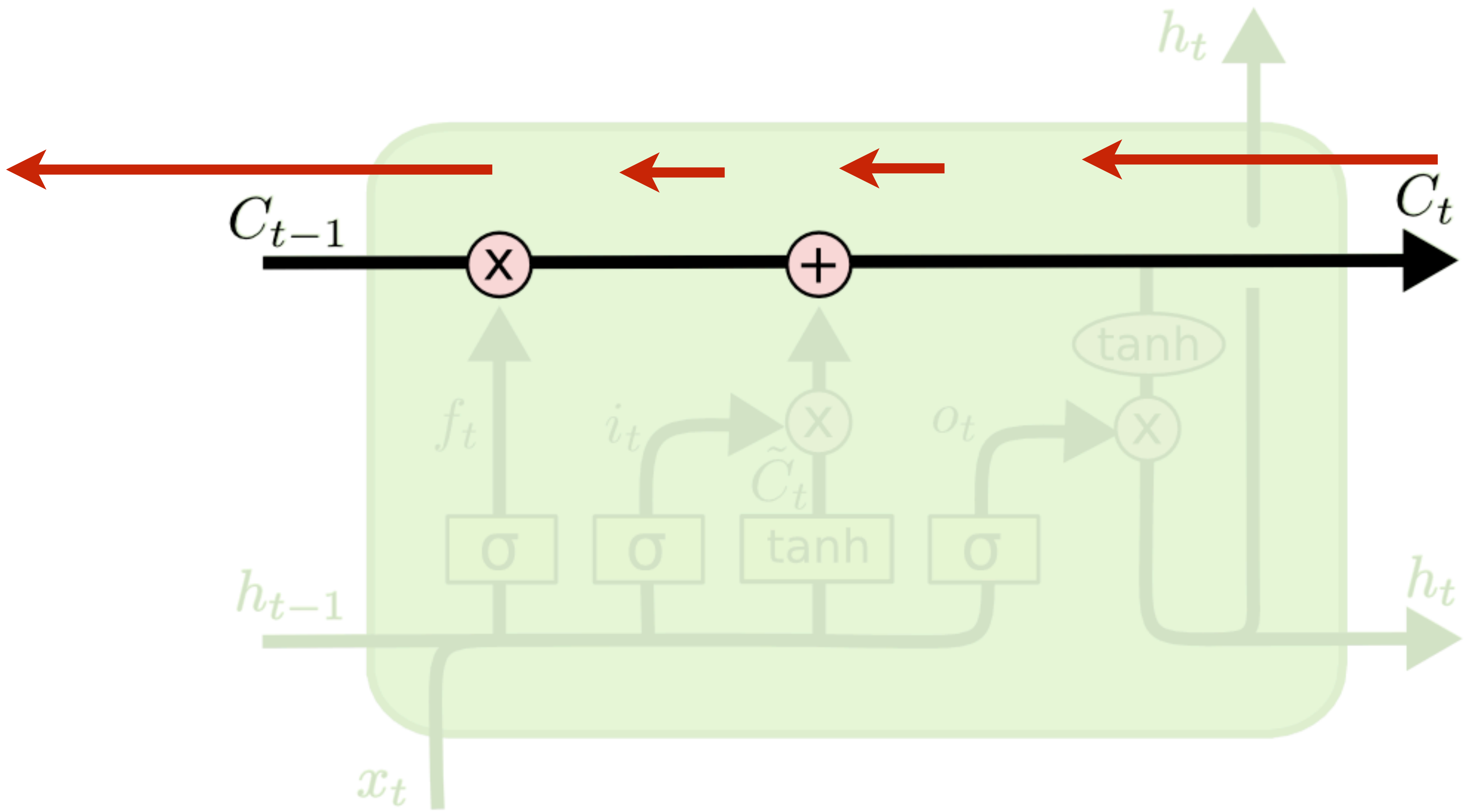
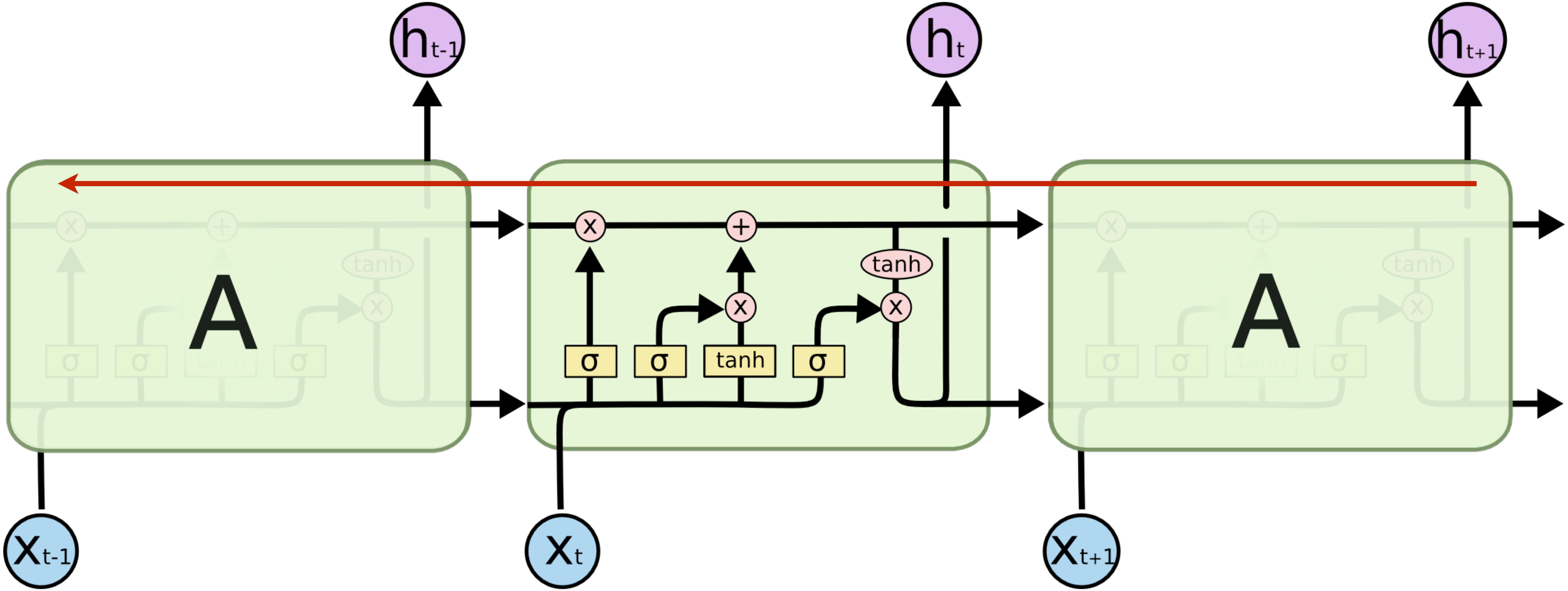



Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Additive Updates

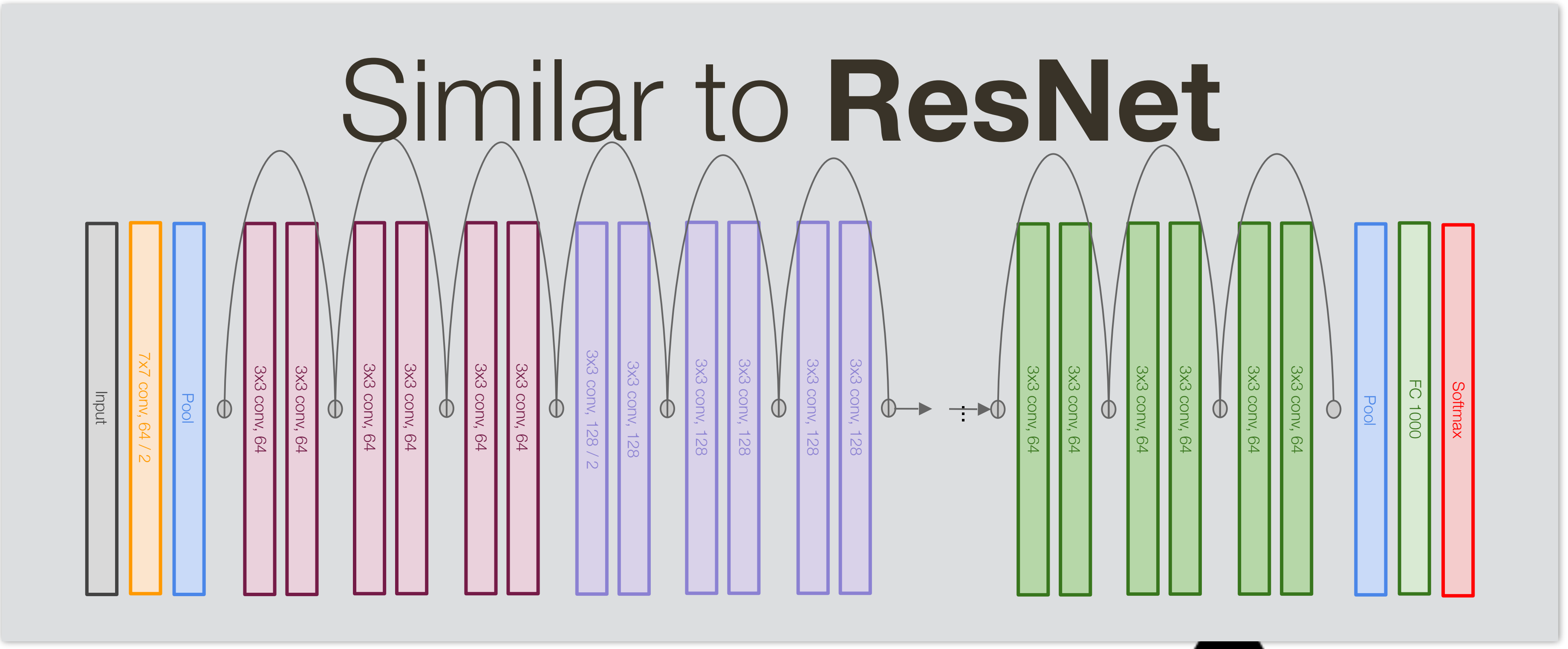


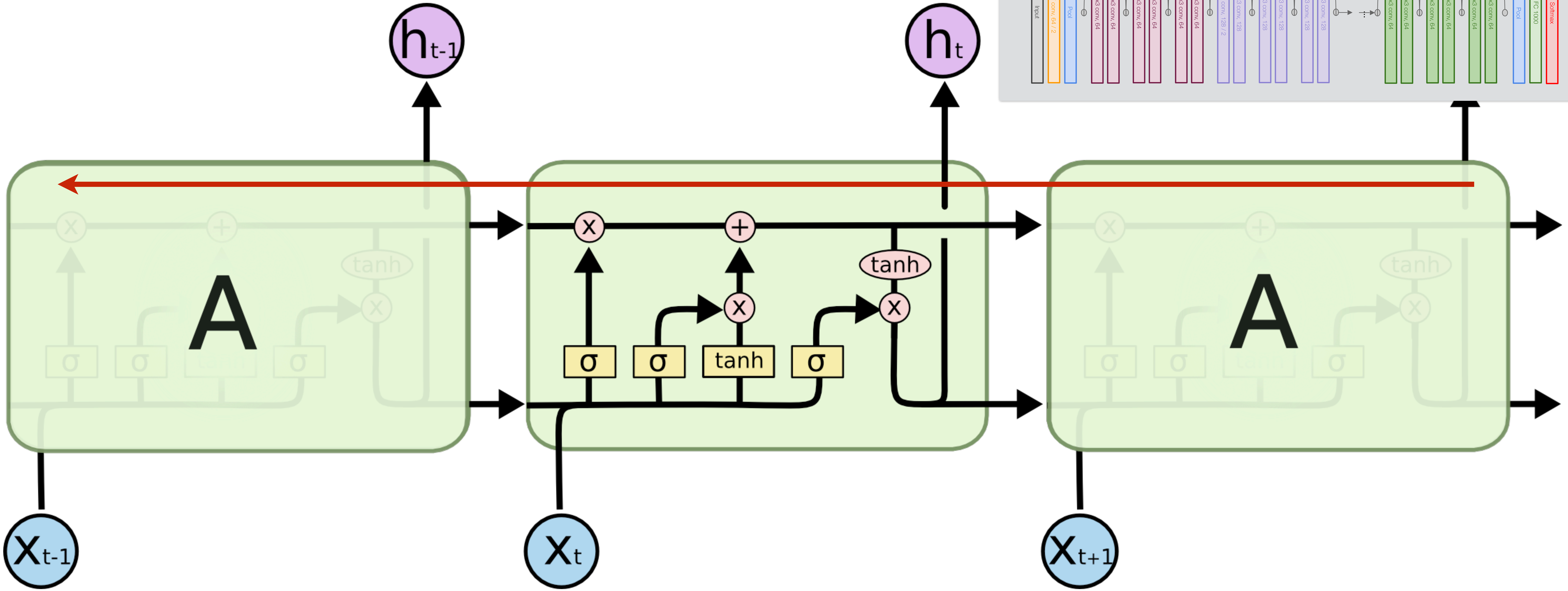

**Uninterrupted gradient flow!**

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Intuition**: Additive Updates



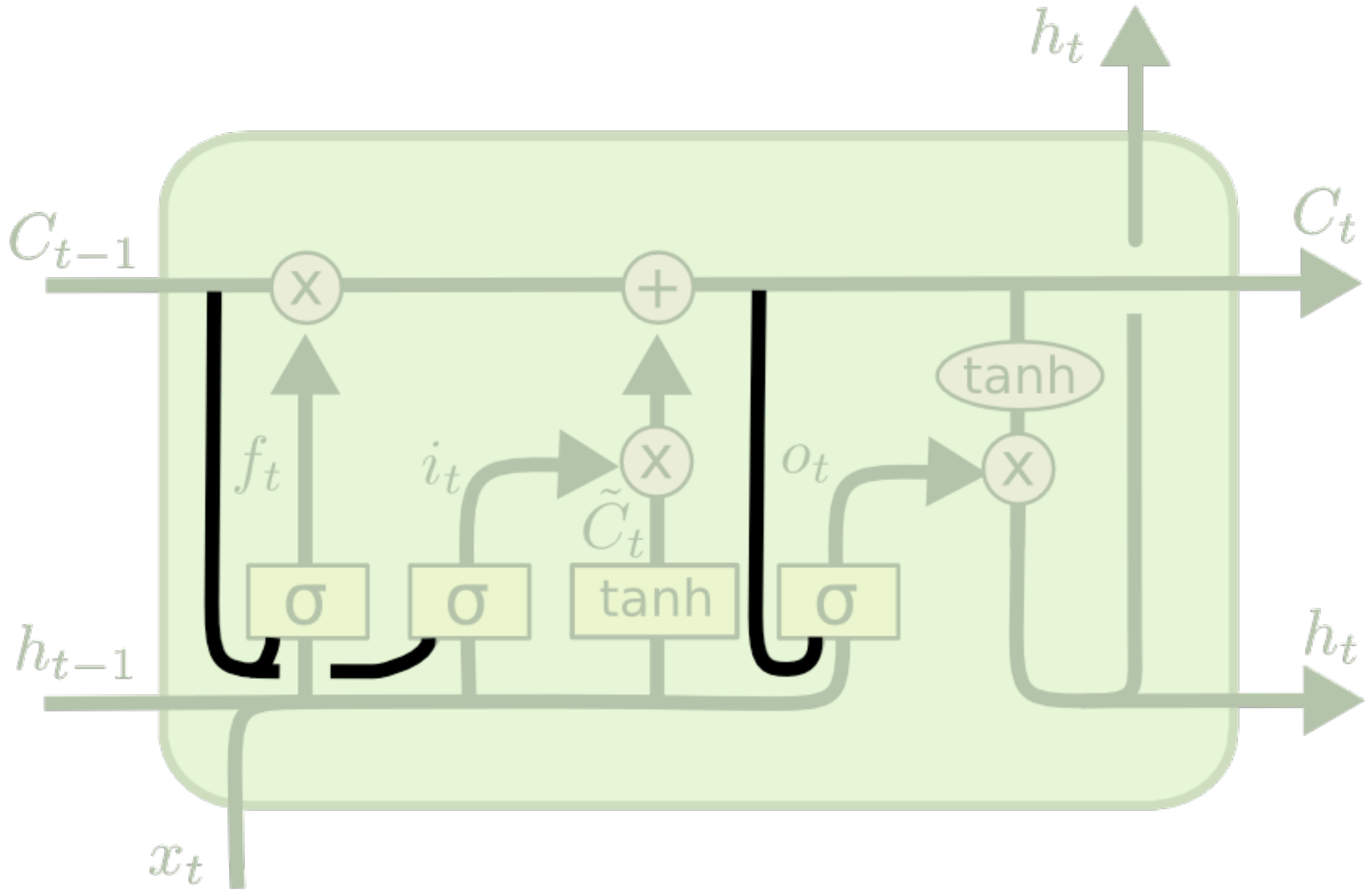

**Uninterrupted gradient flow!**

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Variants**: with Peephole Connections

Lets gates see the cell state / memory



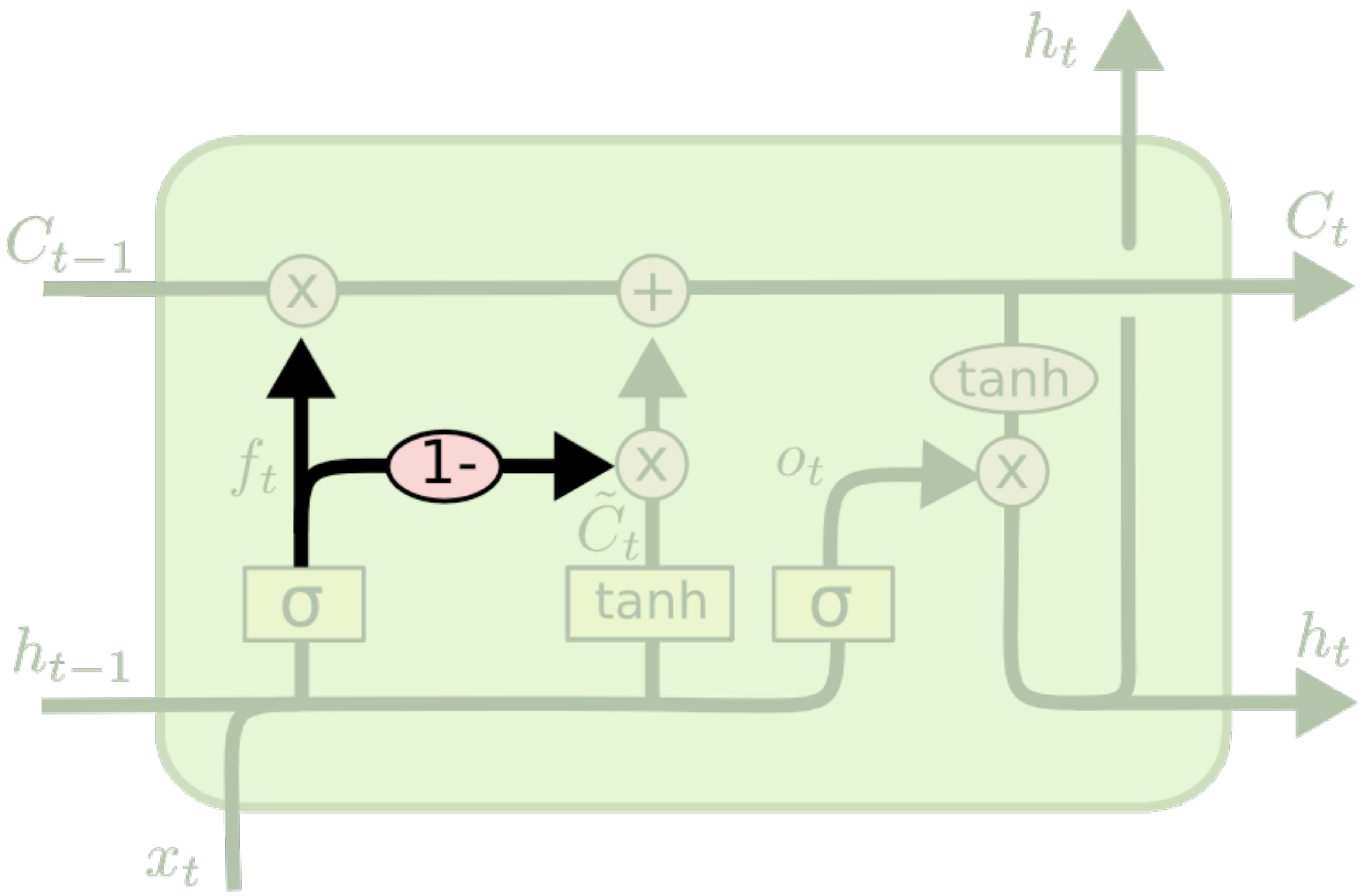

$$f_t = \sigma\left(W_f \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [\boldsymbol{C_{t-1}}, h_{t-1}, x_t] \ + \ b_i\right)$$
$$o_t = \sigma\left(W_o \cdot [\boldsymbol{C_t}, h_{t-1}, x_t] \ + \ b_o\right)$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **LSTM Variants**: with Coupled Gates

Only memorize new information when you're forgetting old



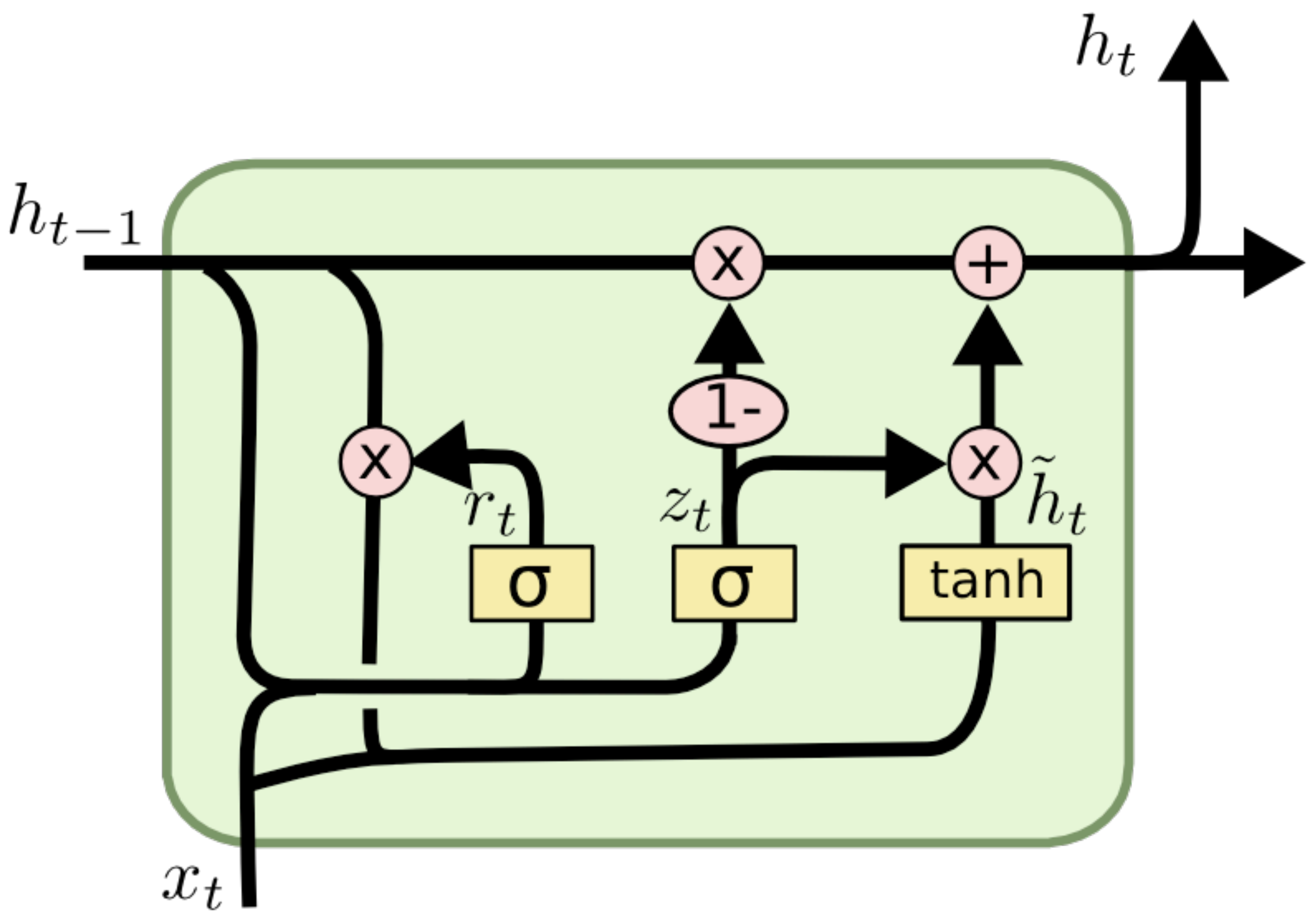

$$C_t = f_t * C_{t-1} + (1 - \boldsymbol{f_t}) * \tilde{C}_t$$

Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

* slide from Dhruv Batra

# **Gated** Recurrent Unit (**GRU**)

No explicit memory; memory = hidden output




$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

z = memorize new and forget old

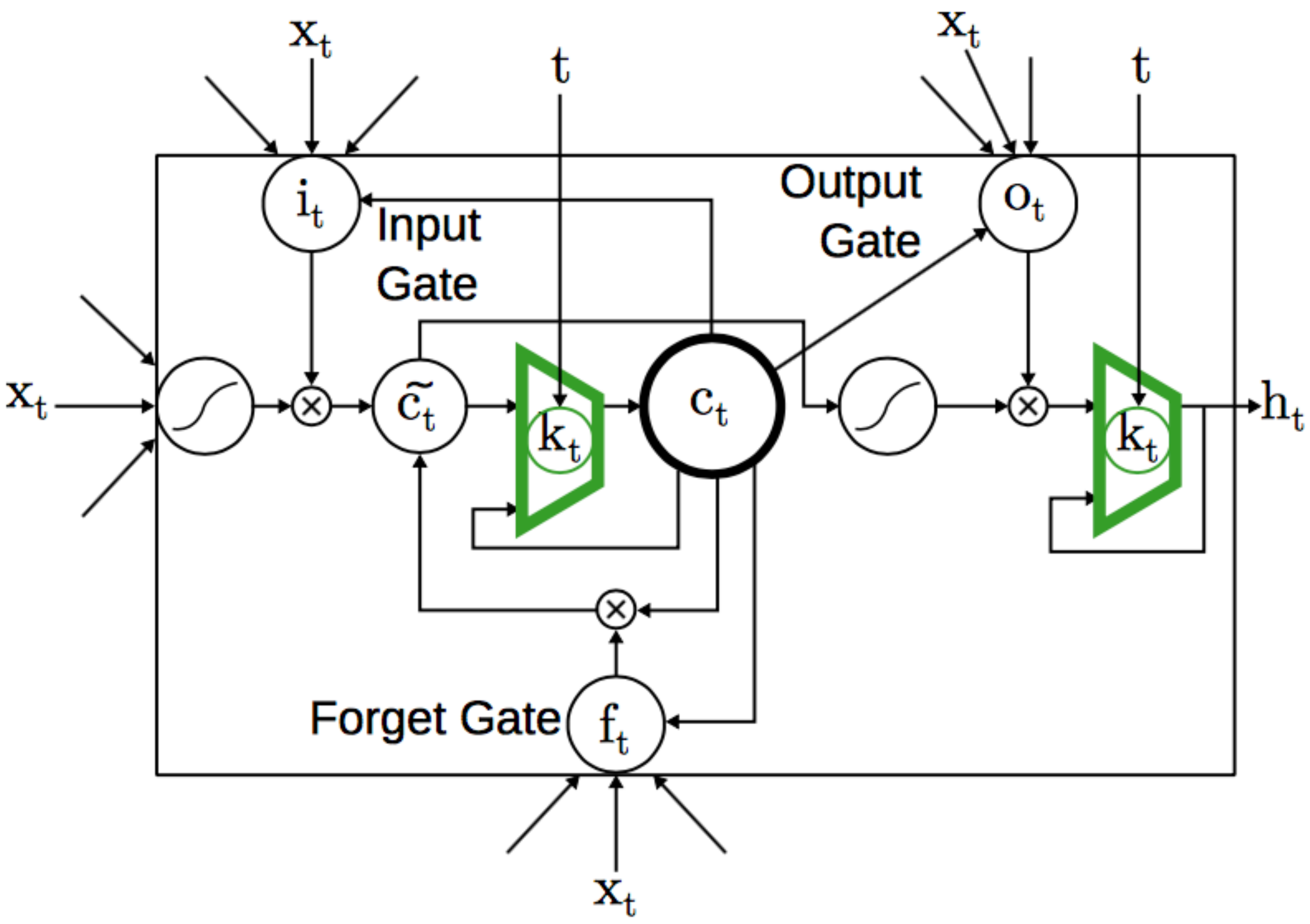
Image Credit: Christopher Olah (http://colah.github.io/posts/2015-08-Understanding-LSTMs/)

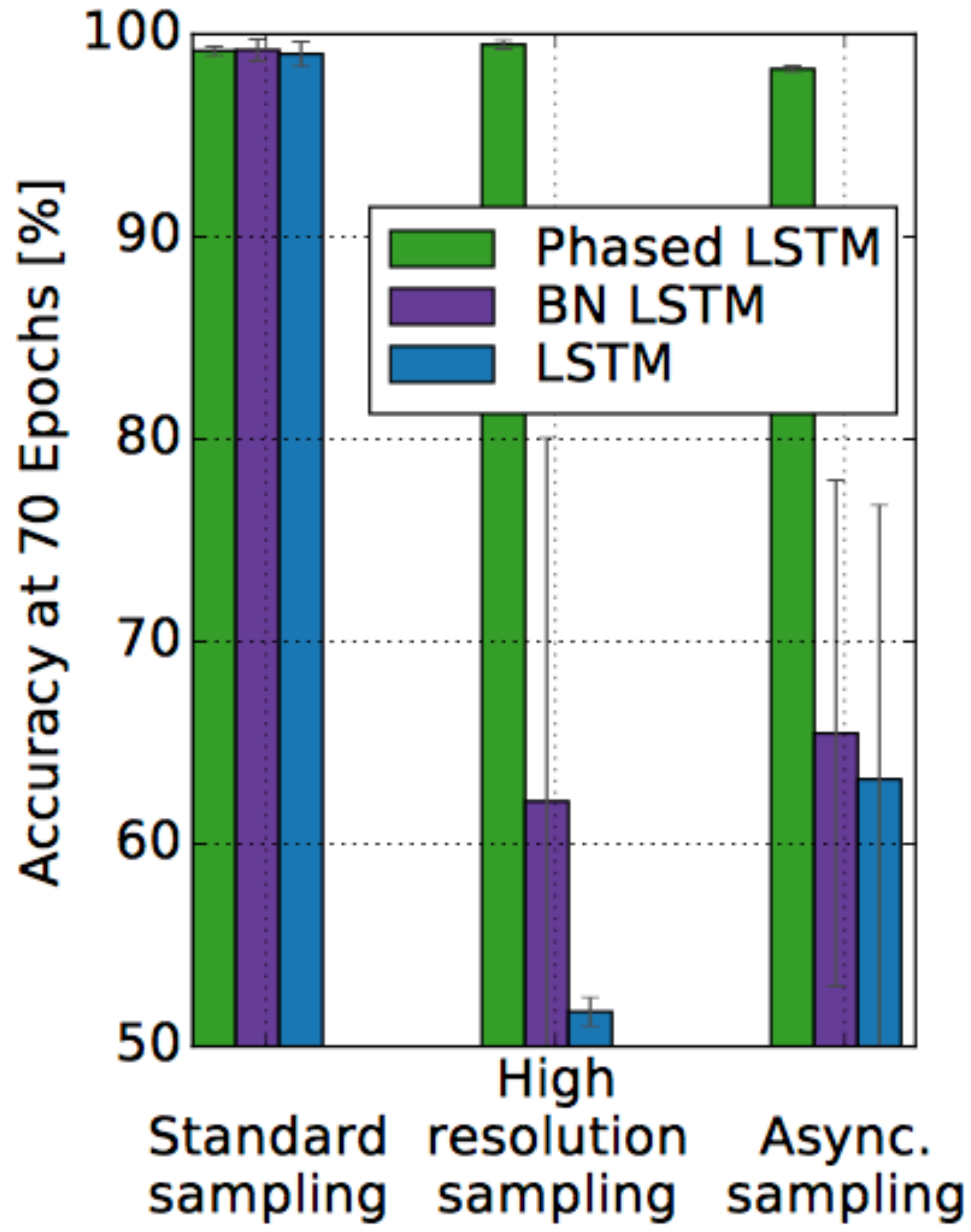
* slide from Dhruv Batra

# **Phased** LSTM

[ Neil et al., 2016 ]

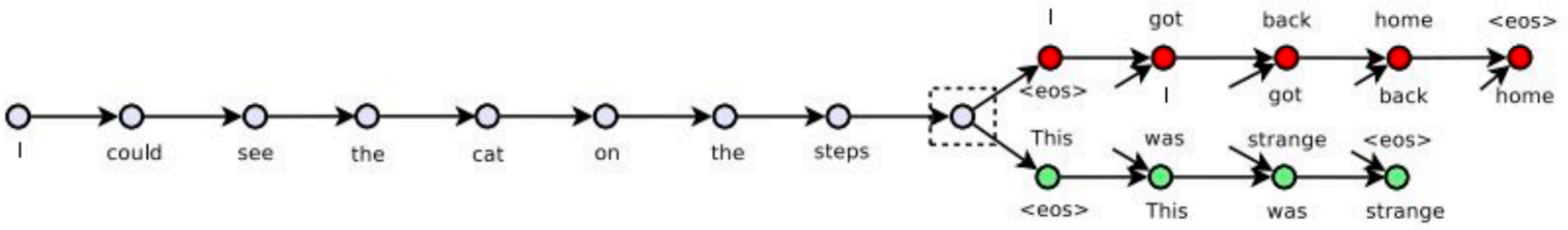
Gates are controlled by **phased** (periodic) **oscillations**



# **Skip-thought** Vectors

word2vec but for sentences, where each sentence is processed by an LSTM




[ Kiros et al., 2015 ]