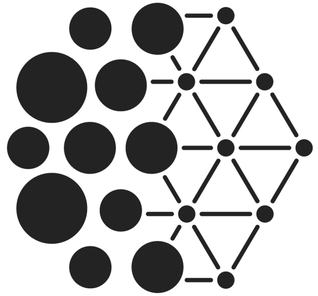
Language Modelling in Deep Learning

Arian Hosseini

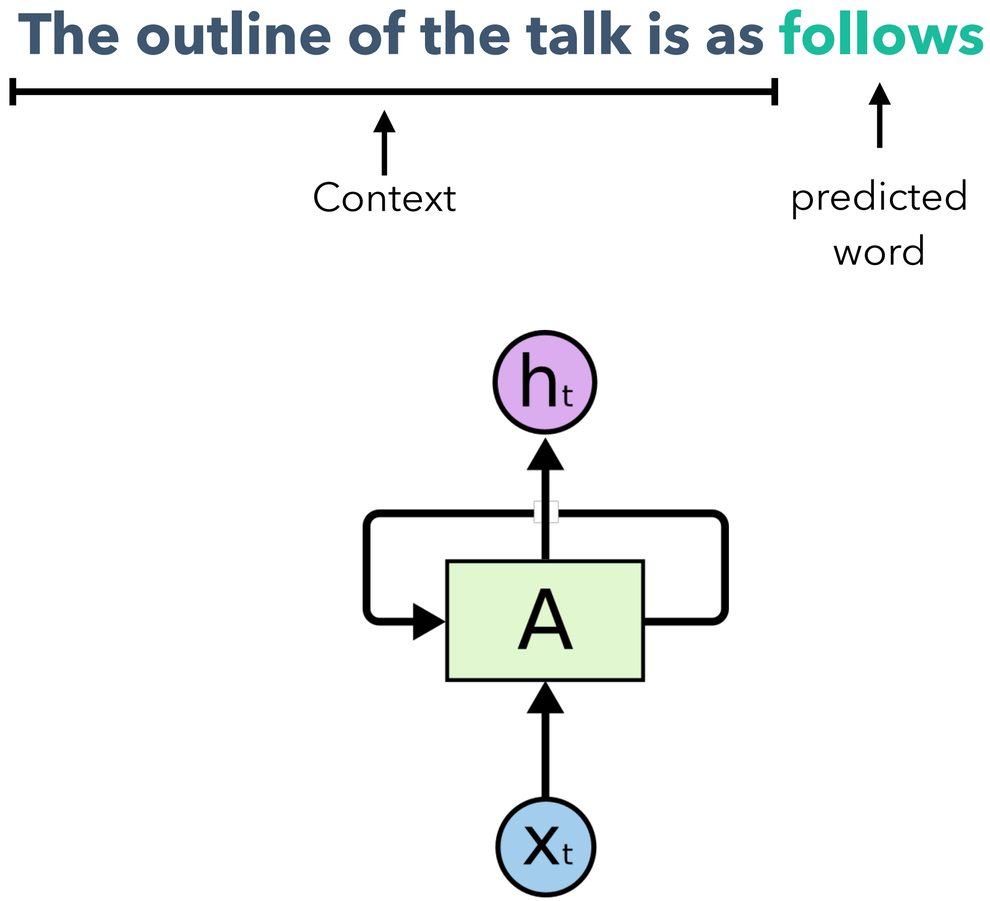
Fall - 2018



Mila

Outline

- Overview of RNNs
- How to use them for Language Modelling
- Discussing Long-term dependencies
- Overview of LSTMs
- Language Modelling in the literature
- Softmax bottleneck and multi-head loss
- The role of pre-training with a language modelling criterion in language understanding



Feedforward Neural Networks

All connections flow forward (no loops); each layer of hidden units is fully connected to the next.

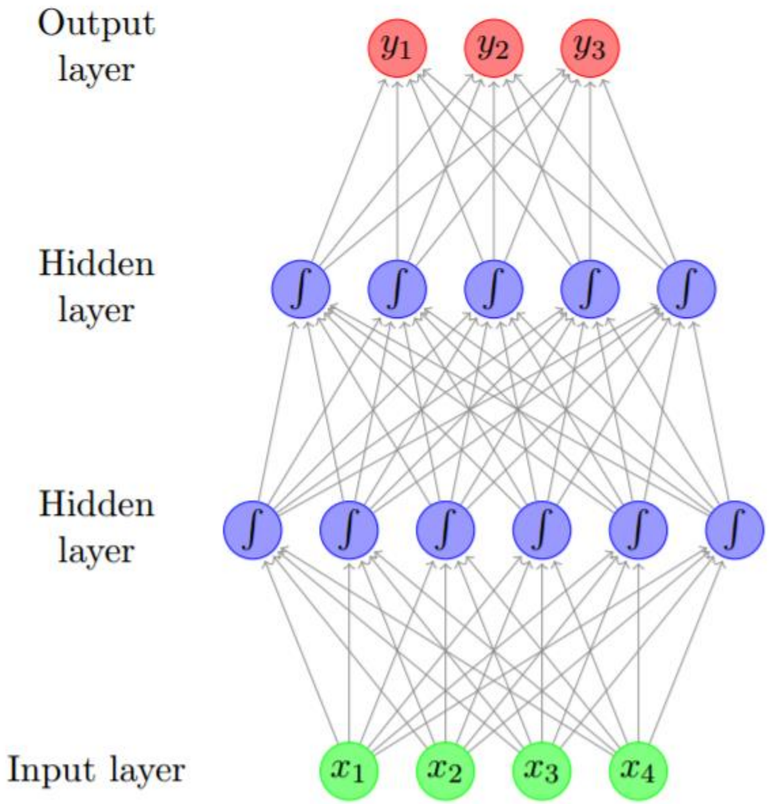


Figure 2: Feed-forward neural network with two hidden layers.

Figure from Goldberg (2015)

Inference in a FF Neural Network

Perform computations forwads through the graph:

$$\mathbf{h^1} = g^1(\mathbf{x}\mathbf{W^1} + \mathbf{b^1})$$
$$\mathbf{h^2} = g^2(\mathbf{h^1}\mathbf{W^2} + \mathbf{b^2})$$
$$\mathbf{y} = \mathbf{h^2}\mathbf{W^3}$$

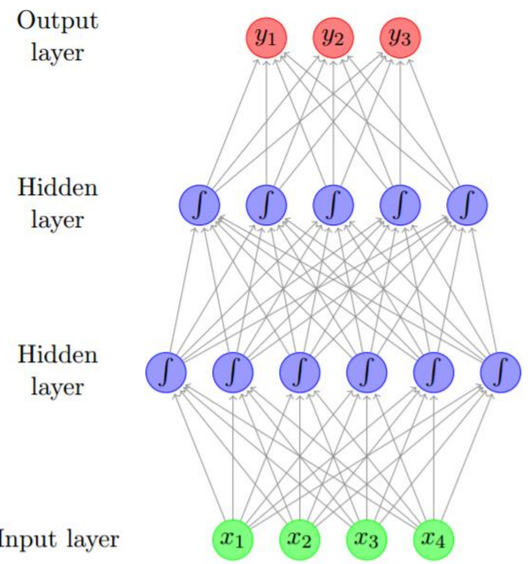
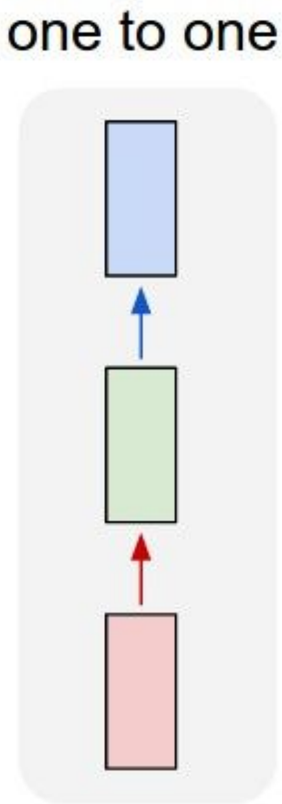


Figure 2: Feed-forward neural network with two hidden layers.

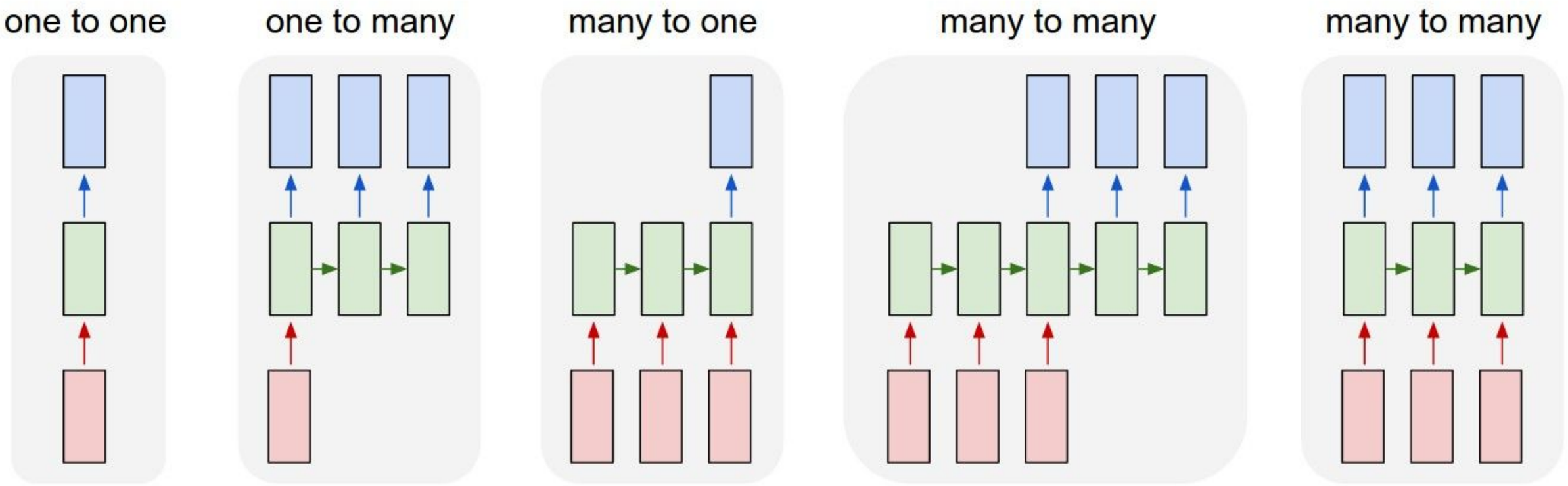
Note that we are now representing each layer as a vector; combining all of the weights in a layer across the units into a weight matrix

“Vanilla” Neural Network



Vanilla Neural Networks

Recurrent Neural Networks: Process Sequences



e.g. **Image Captioning**
image -> sequence of words

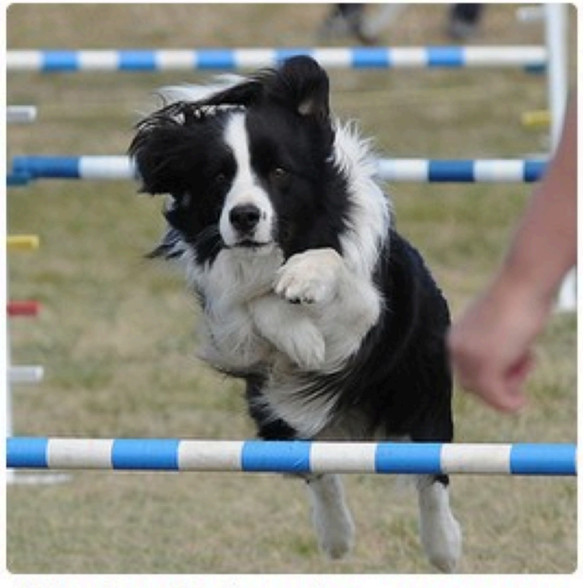
"man in black shirt is playing guitar."

"construction worker in orange safety vest is working on road."

"two young girls are playing with lego toy."

"boy is doing backflip on wakeboard."

"girl in pink dress is jumping in air."



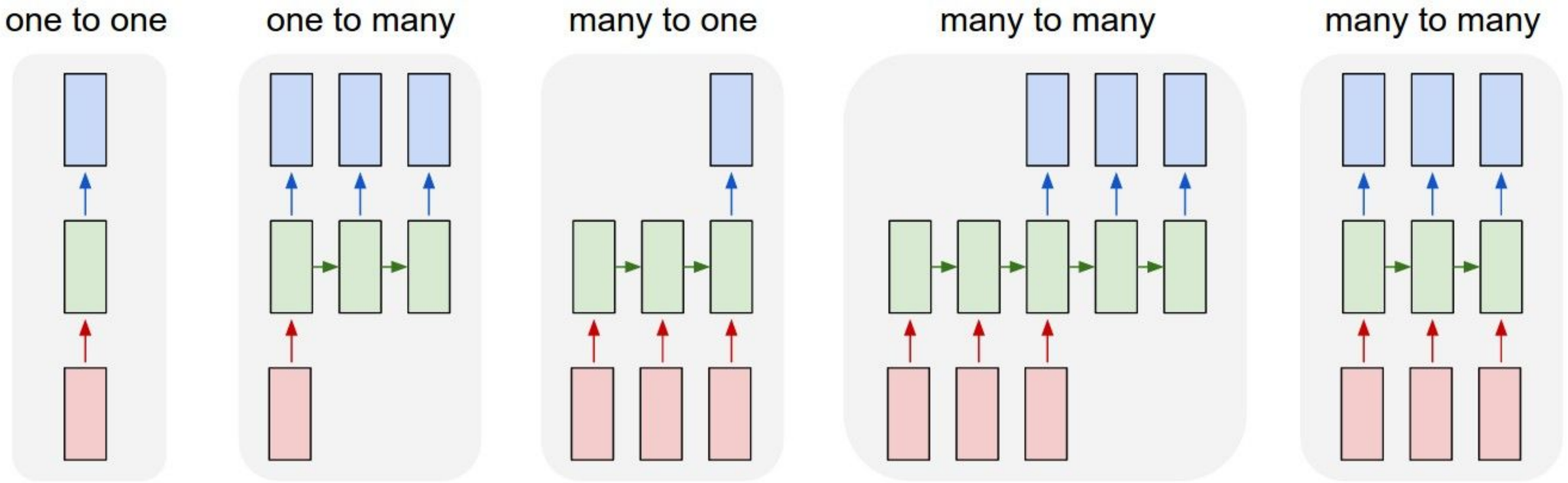
"black and white dog jumps over bar."

"young girl in pink shirt is swinging on swing."

"man in blue wetsuit is surfing on wave."

https://cs.stanford.edu/people/karpathy/sfmltalk.pdf

Recurrent Neural Networks: Process Sequences



e.g. **Sentiment Classification**
sequence of words -> sentiment

NEGATIVE

Totally dissatisfied with the service. Worst customer care ever.

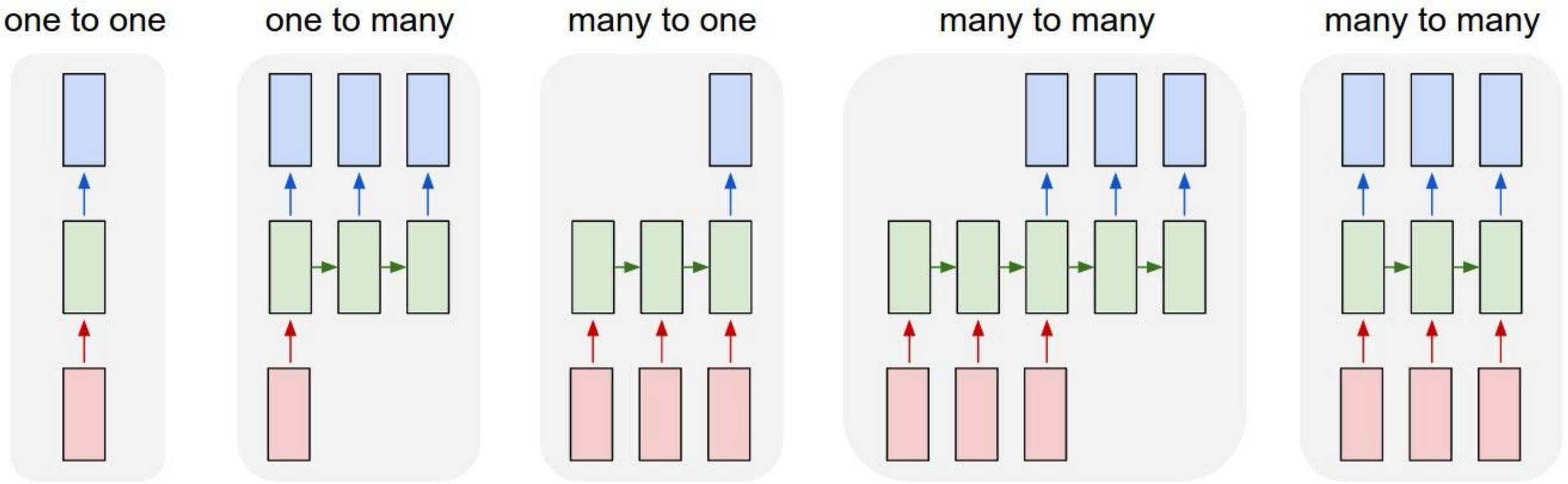
NEUTRAL

Good Job but I will expect a lot more in future.

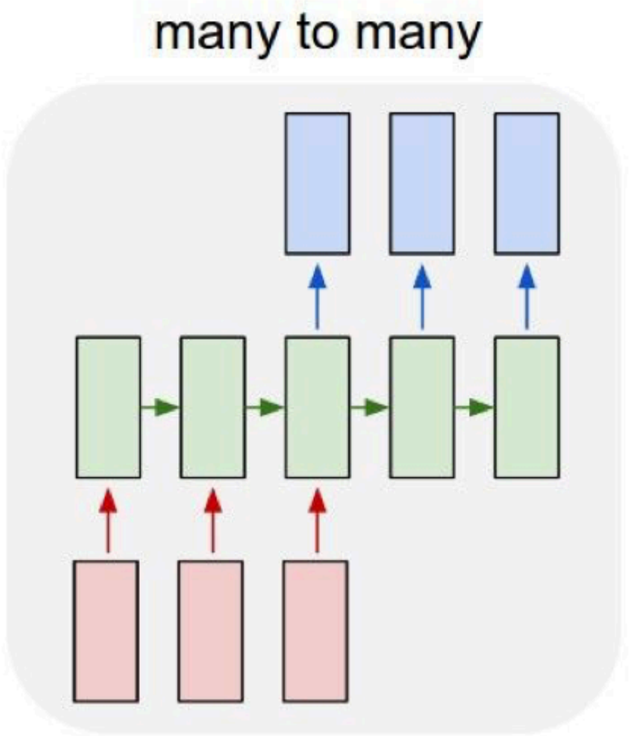
POSITIVE

Brilliant effort guys! Loved Your Work.

Recurrent Neural Networks: Process Sequences



e.g. **Machine Translation**
seq of words -> seq of words



Italian | Irish | English | Detect language

French | German | Italian | Translate

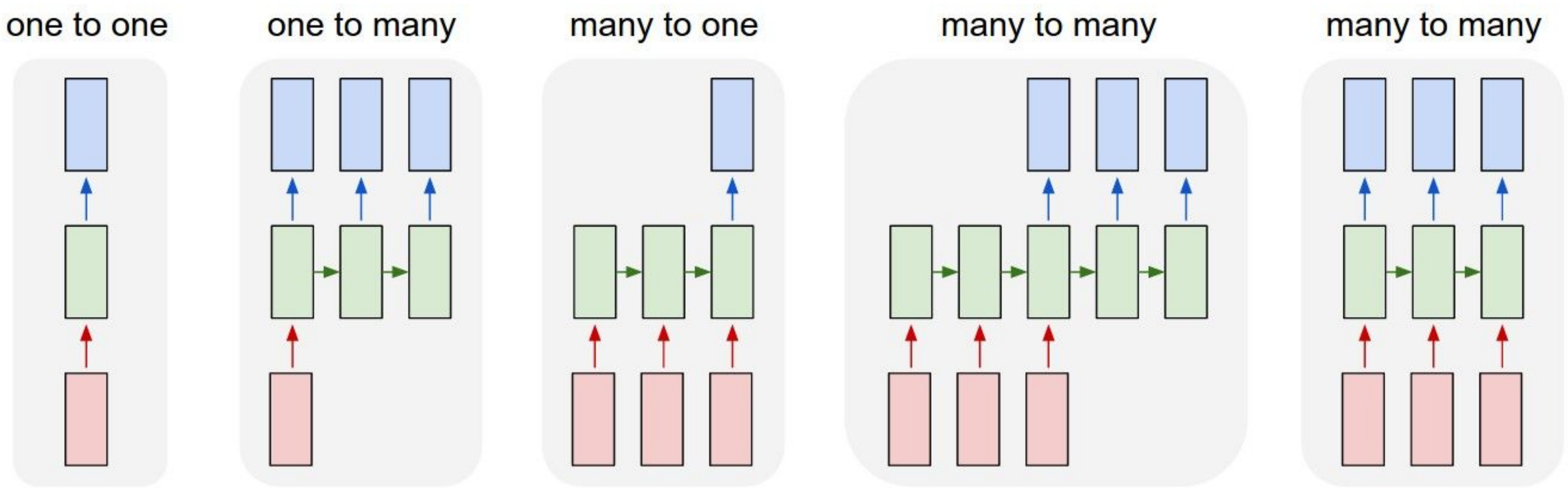
I'm definitely not using Google translator. My French is perfect.

65/5000

Je n'utilise certainement pas Google Translator. Mon français est parfait

Suggest an edit

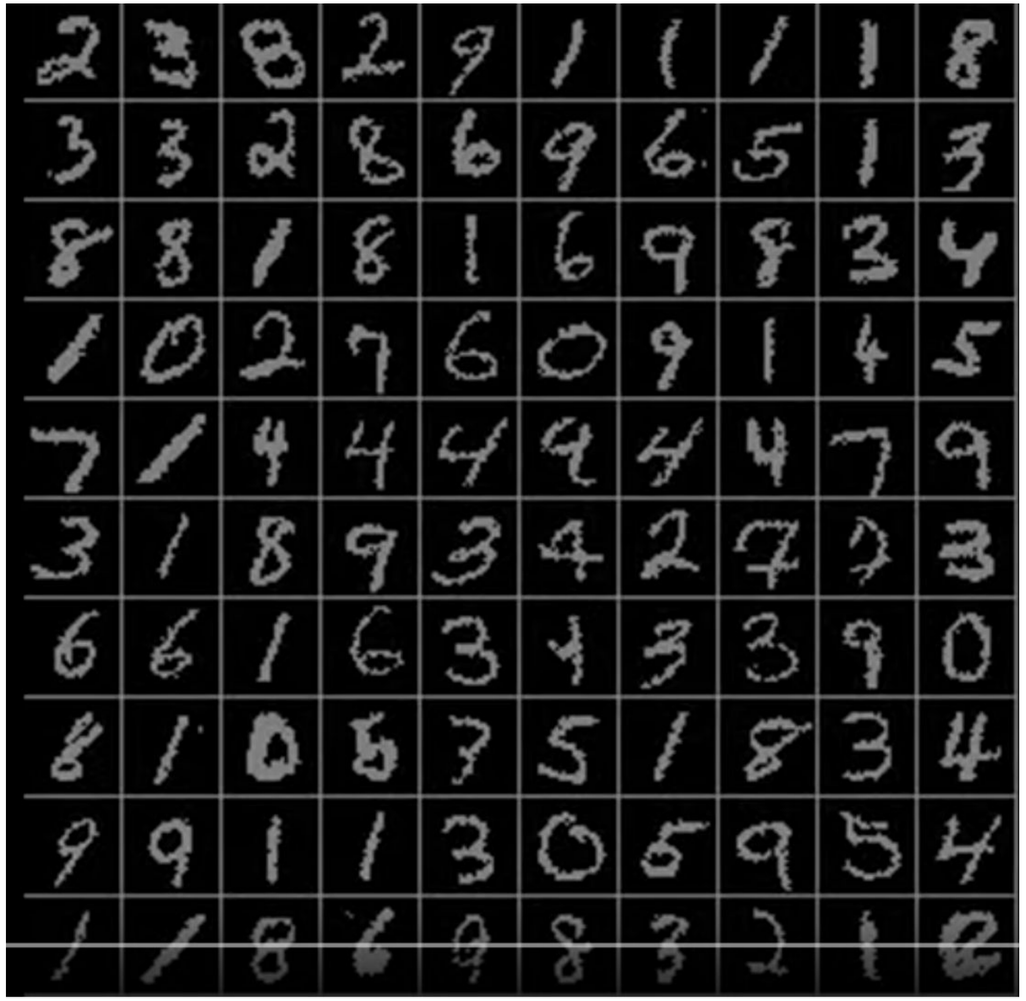
Recurrent Neural Networks: Process Sequences



e.g. **Video classification on frame level**

Sequential Processing of Non-Sequence Data

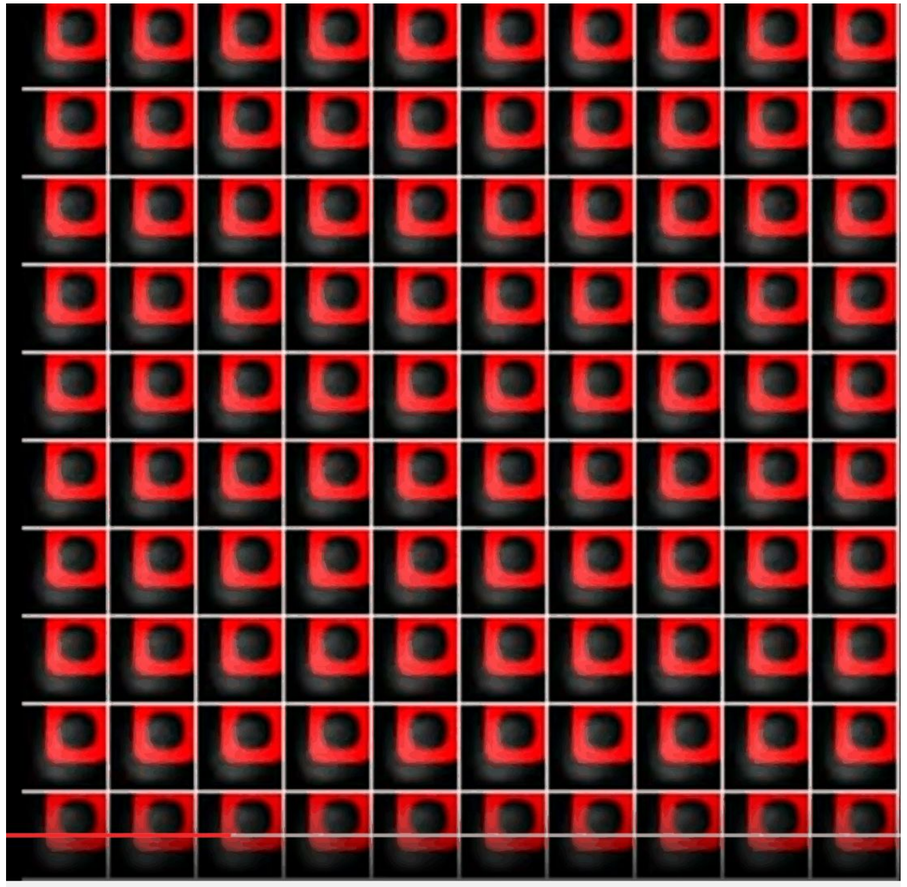
Classify images by taking a series of “glimpses”

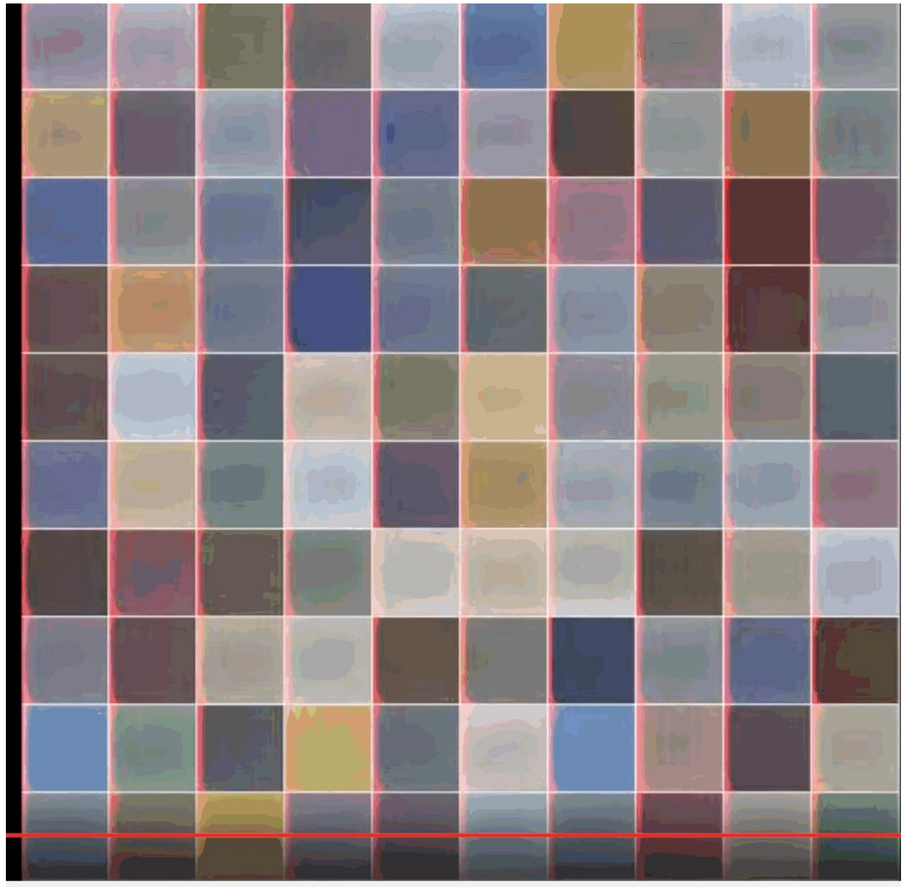


Ba, Mnih, and Kavukcuoglu, “Multiple Object Recognition with Visual Attention”, ICLR 2015.
Gregor et al, “DRAW: A Recurrent Neural Network For Image Generation”, ICML 2015
Figure copyright Karol Gregor, Ivo Danihelka, Alex Graves, Danilo Jimenez Rezende, and Daan Wierstra, 2015. Reproduced with permission.

Sequential Processing of Non-Sequence Data

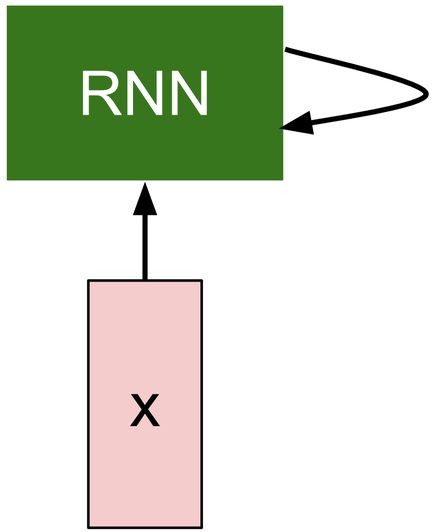
Generate images one piece at a time!



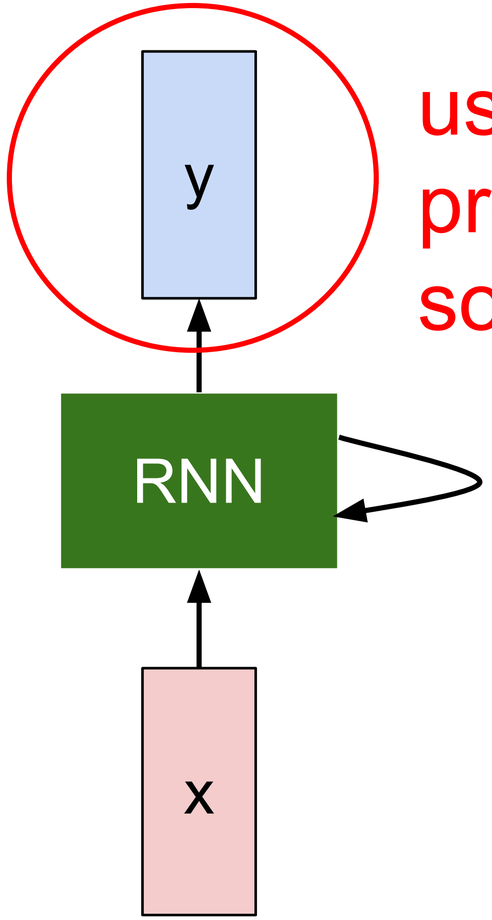


Gregor et al, "DRAW: A Recurrent Neural Network For Image Generation", ICML 2015

Recurrent Neural Network



Recurrent Neural Network



usually want to predict a vector at some time steps

Recurrent Neural Network

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

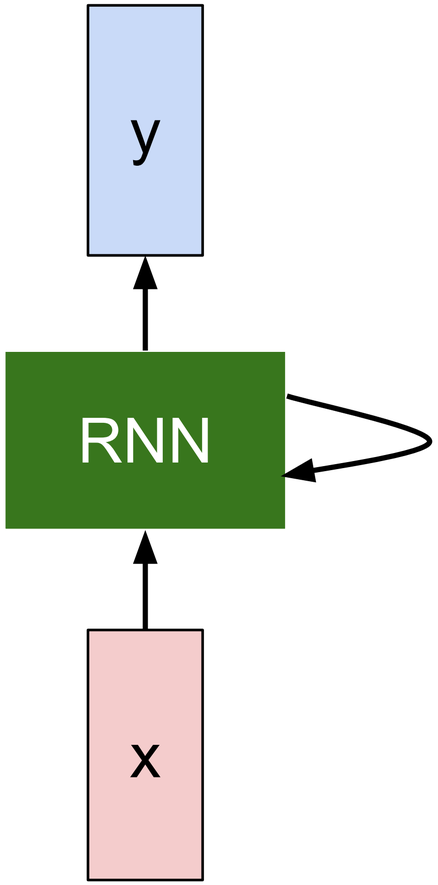
$$h_t = f_W(h_{t-1}, x_t)$$

new state

some function with parameters W

old state

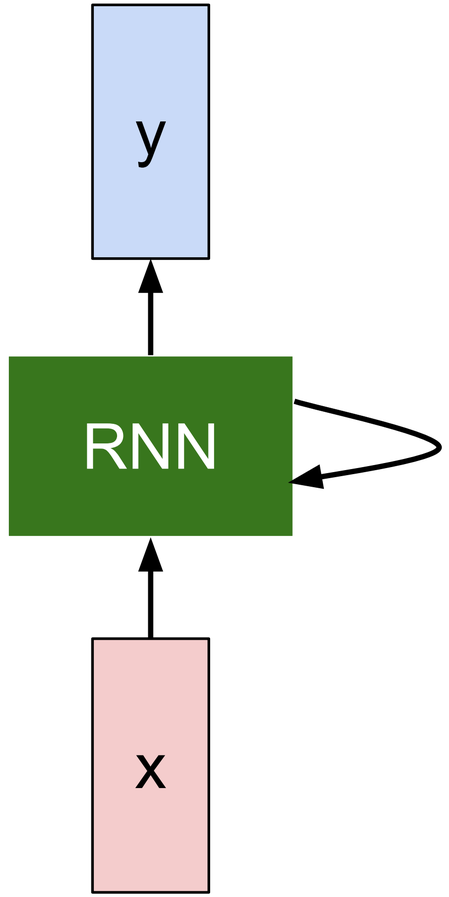
input vector at some time step



Recurrent Neural Network

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

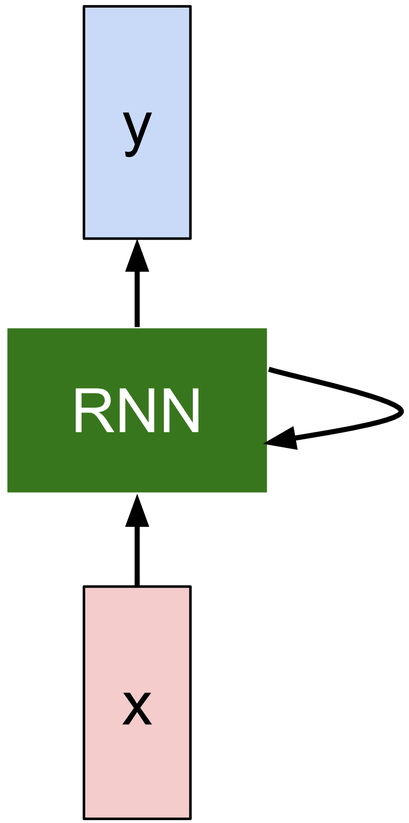
$$h_t = f_W(h_{t-1}, x_t)$$



Notice: the same function and the same set of parameters are used at every time step.

(Simple) Recurrent Neural Network

The state consists of a single *"hidden"* vector **h**:



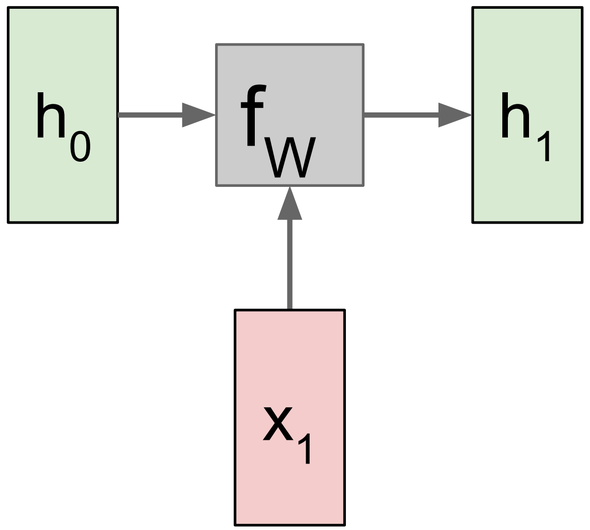
$$h_t = f_W(h_{t-1}, x_t)$$

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

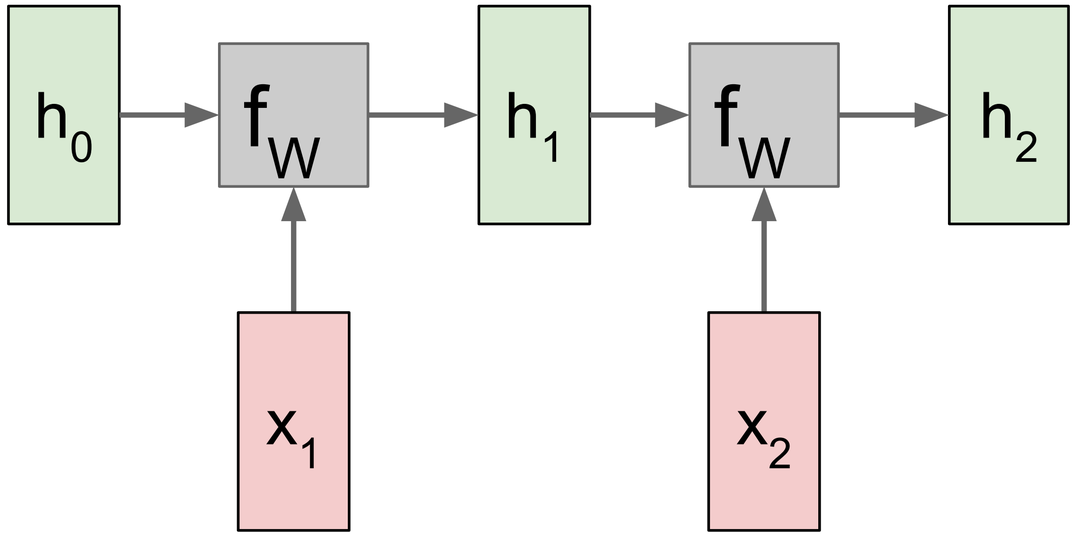
$$y_t = W_{hy}h_t$$

Sometimes called a "Vanilla RNN" or an "Elman RNN" after Prof. Jeffrey Elman

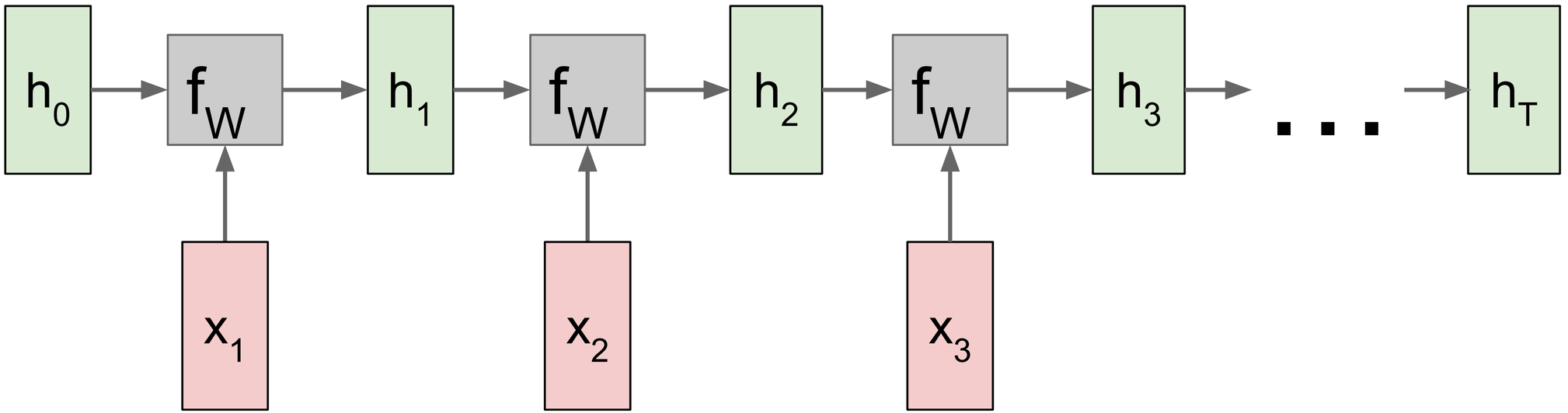
RNN: Computational Graph



RNN: Computational Graph

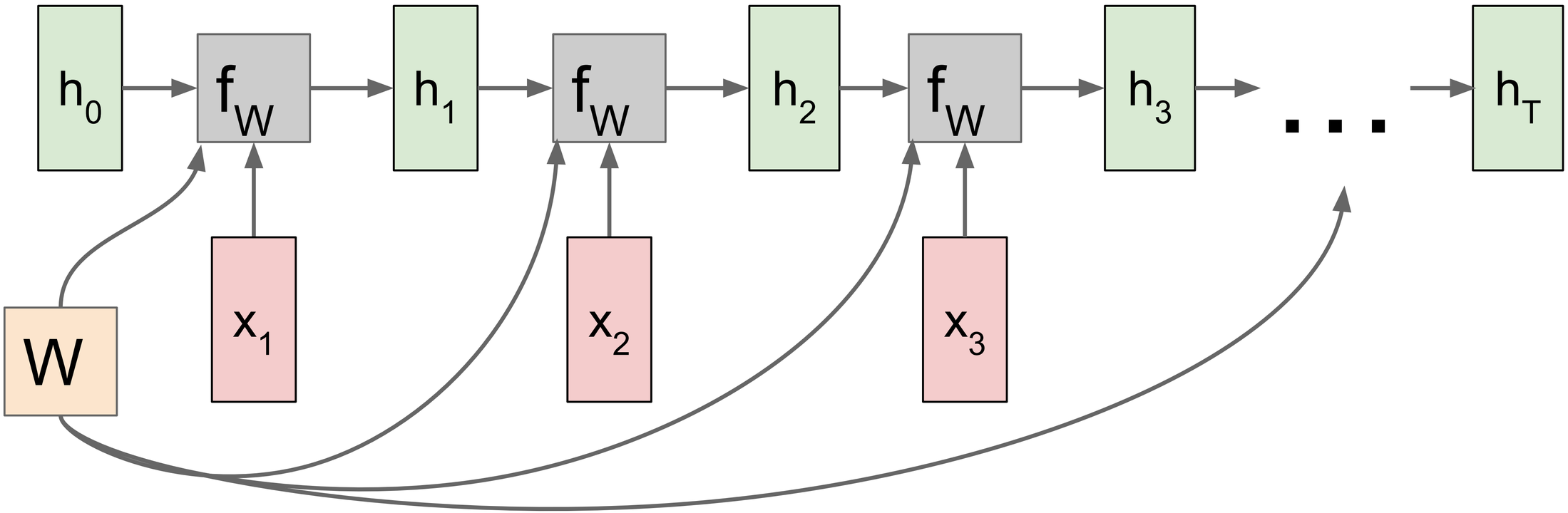


RNN: Computational Graph

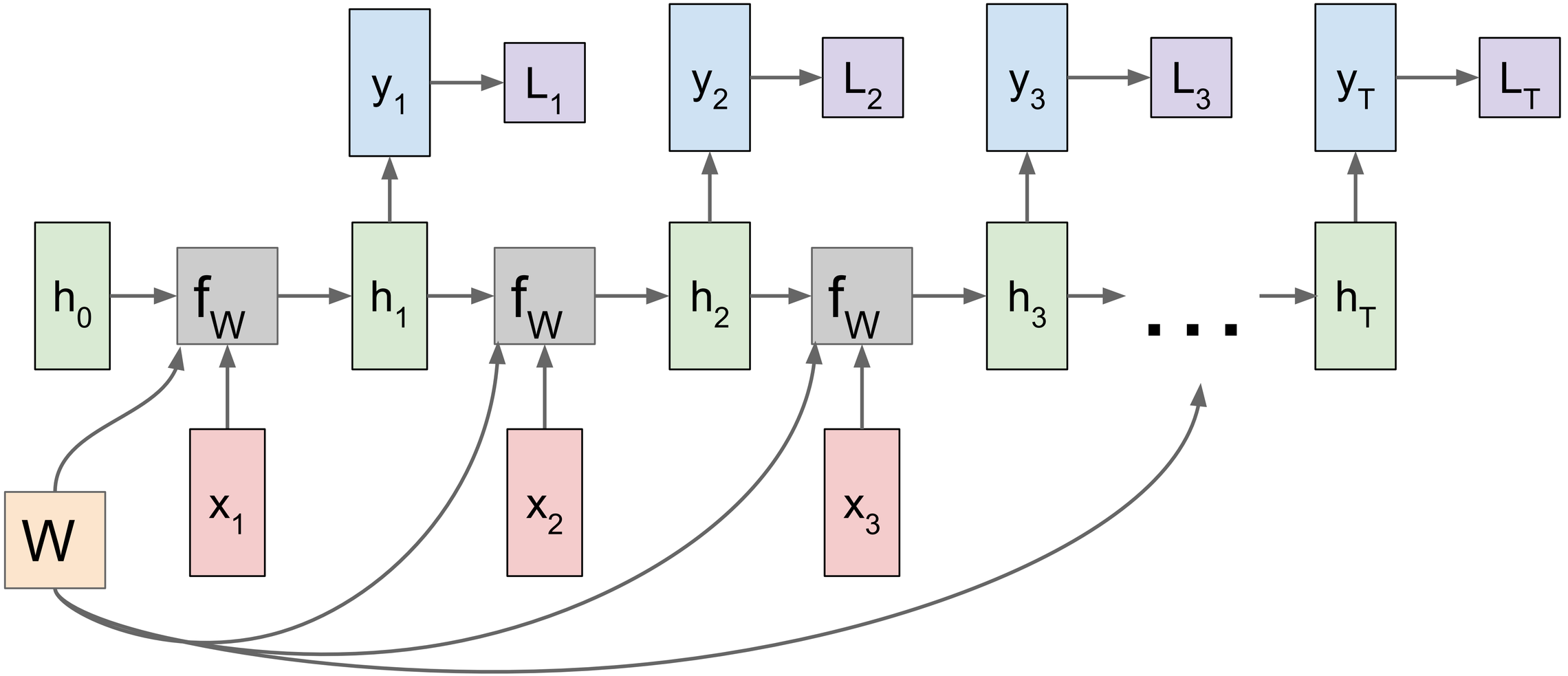


RNN: Computational Graph

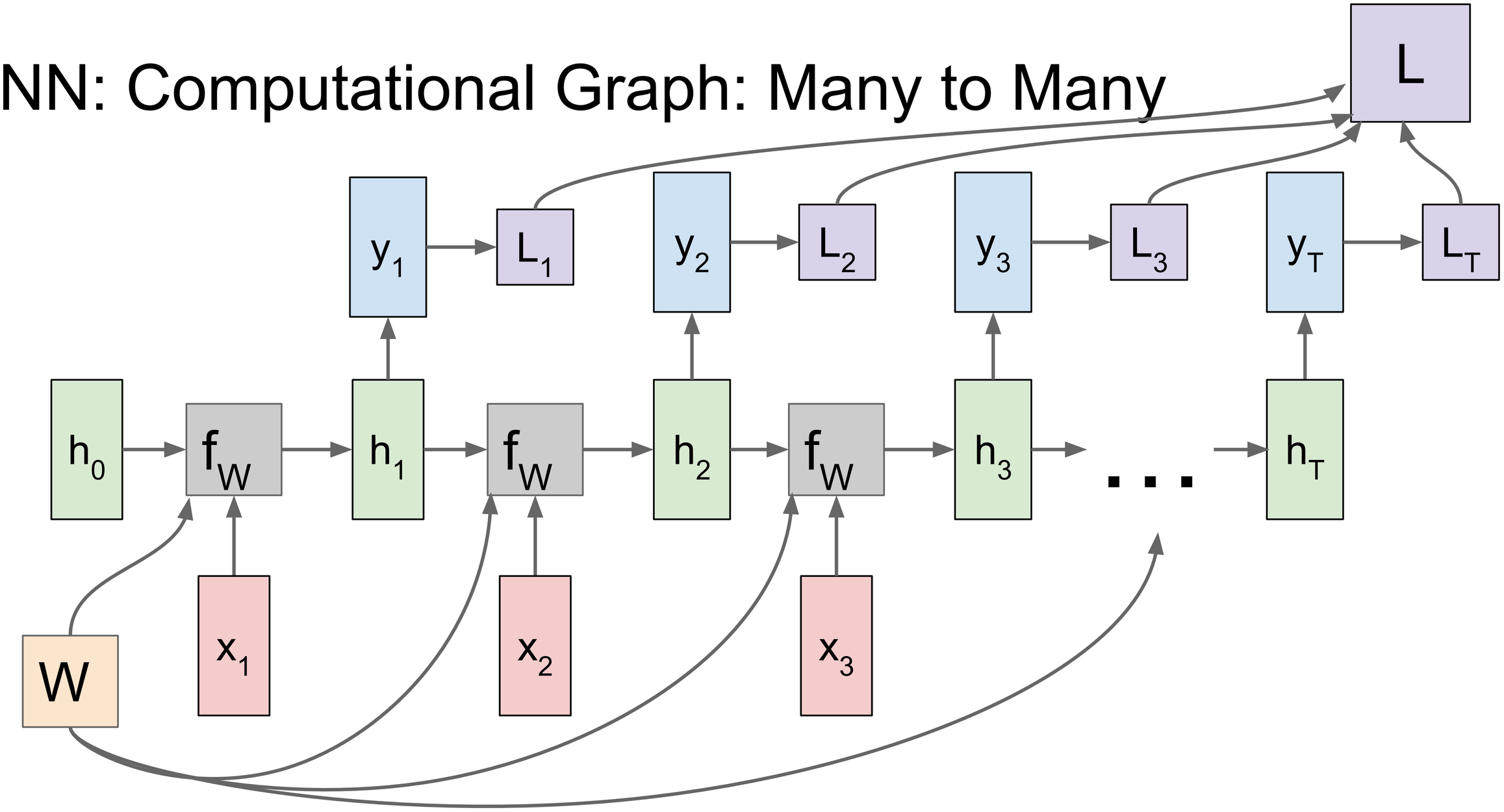
Re-use the same weight matrix at every time-step



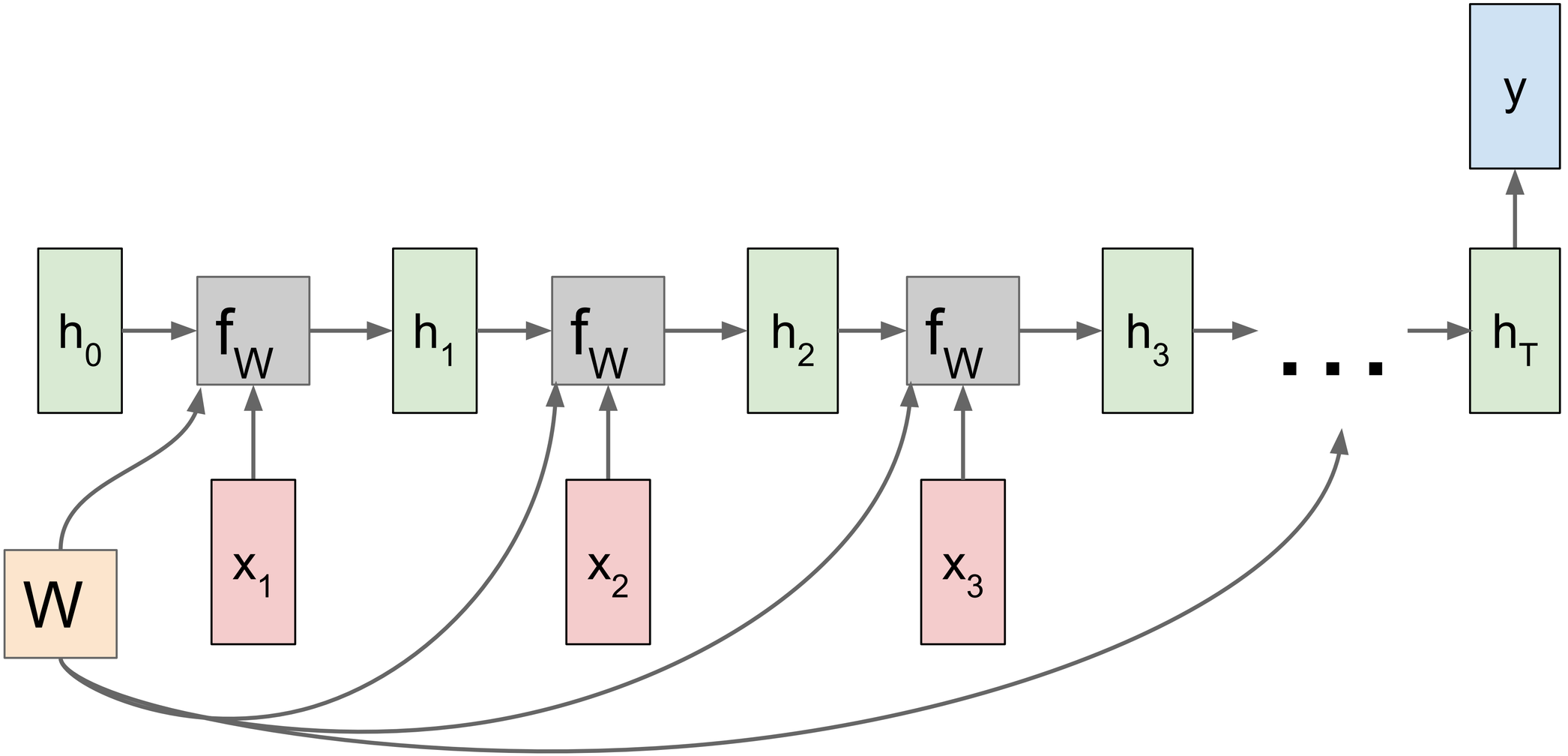
RNN: Computational Graph: Many to Many



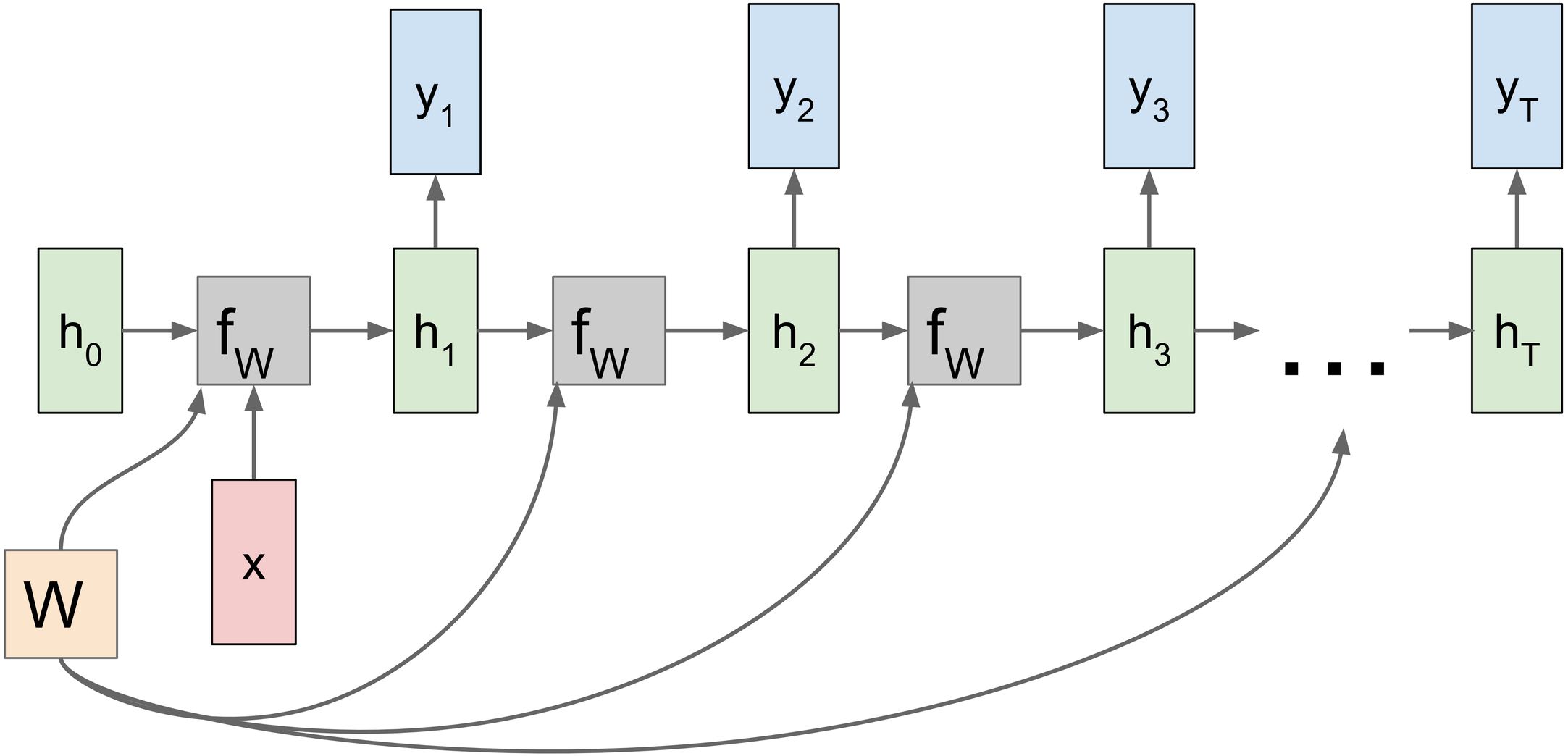
RNN: Computational Graph: Many to Many



RNN: Computational Graph: Many to One



RNN: Computational Graph: One to Many



How to use them for Language Modelling

- Language modelling mostly relies on a universal **auto-regressive** factorization of the joint probability
- It models each conditional factor using different approaches
- Given a corpus of tokens $X = (X_1, \ldots, X_T)$, the joint probability P(X) factorizes:

$$P(X) = \prod_t P(X_t \mid X_{\leq t}) = \prod_t P(X_t \mid C_t)$$

- where $C_t = X_{<t}$ is referred to as the **context** of the conditional probability.

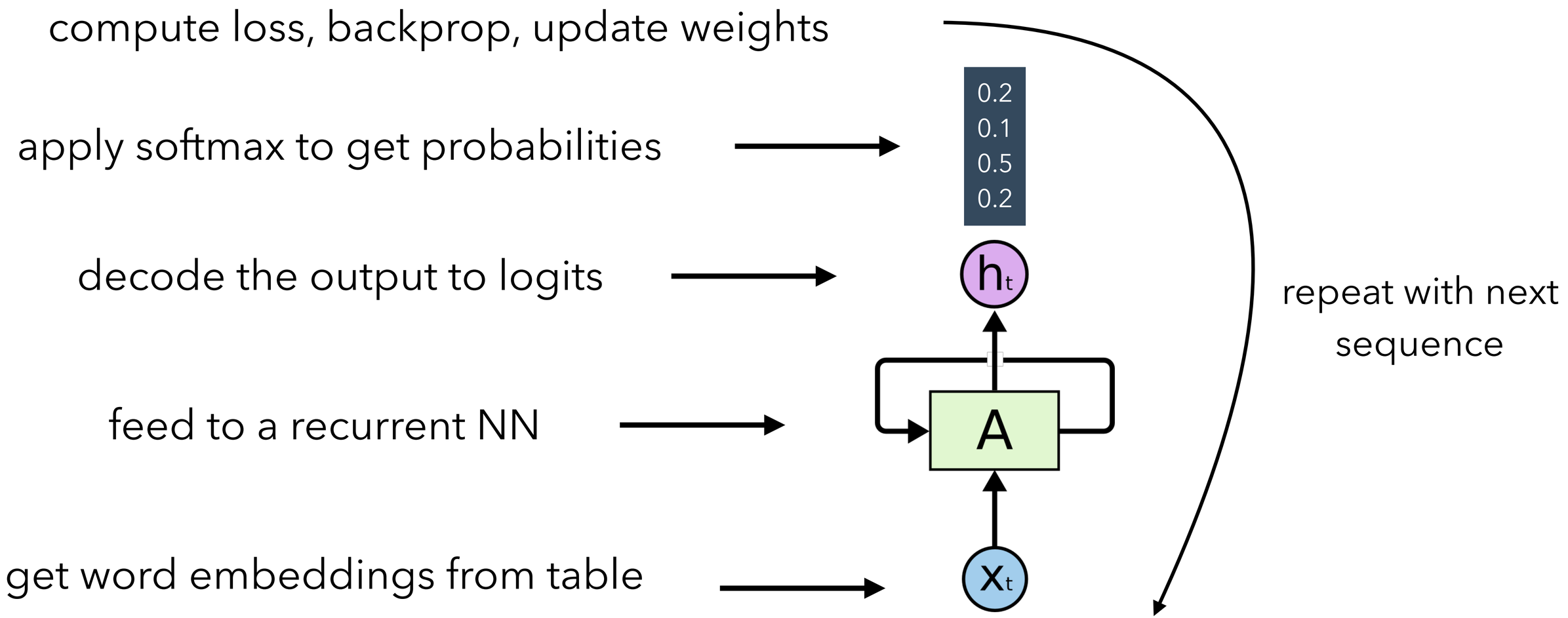
- One could use a **recurrent network** to encode the context into a **fixed size vector** (context representation)

- This vector could further be used in a decoder network (e.g. dot product by word embeddings) to obtain the **logits**

- The Logits can be fed into a **Softmax** function to get a categorical probability distribution over the **next token**.

softmax function $$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$ for j = 1, … , K

Let's see an example

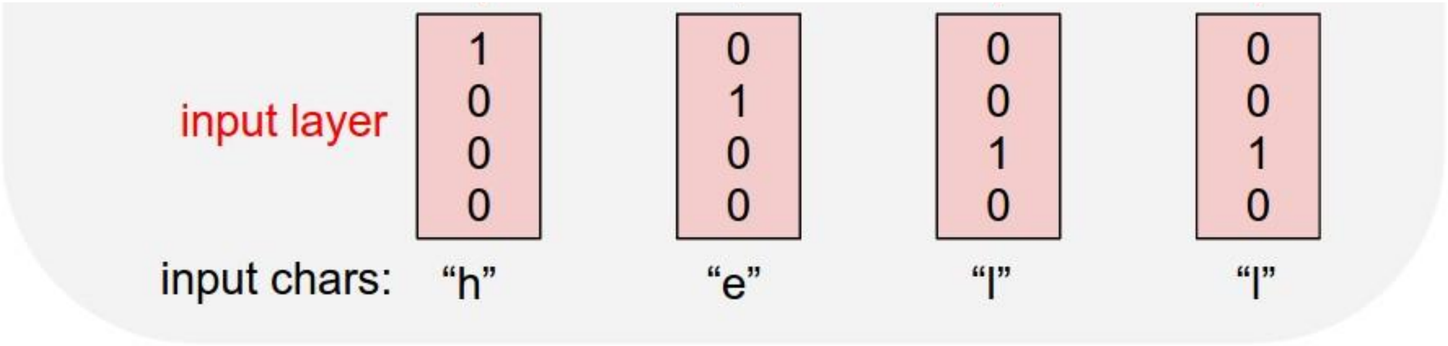


So the steps are

Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

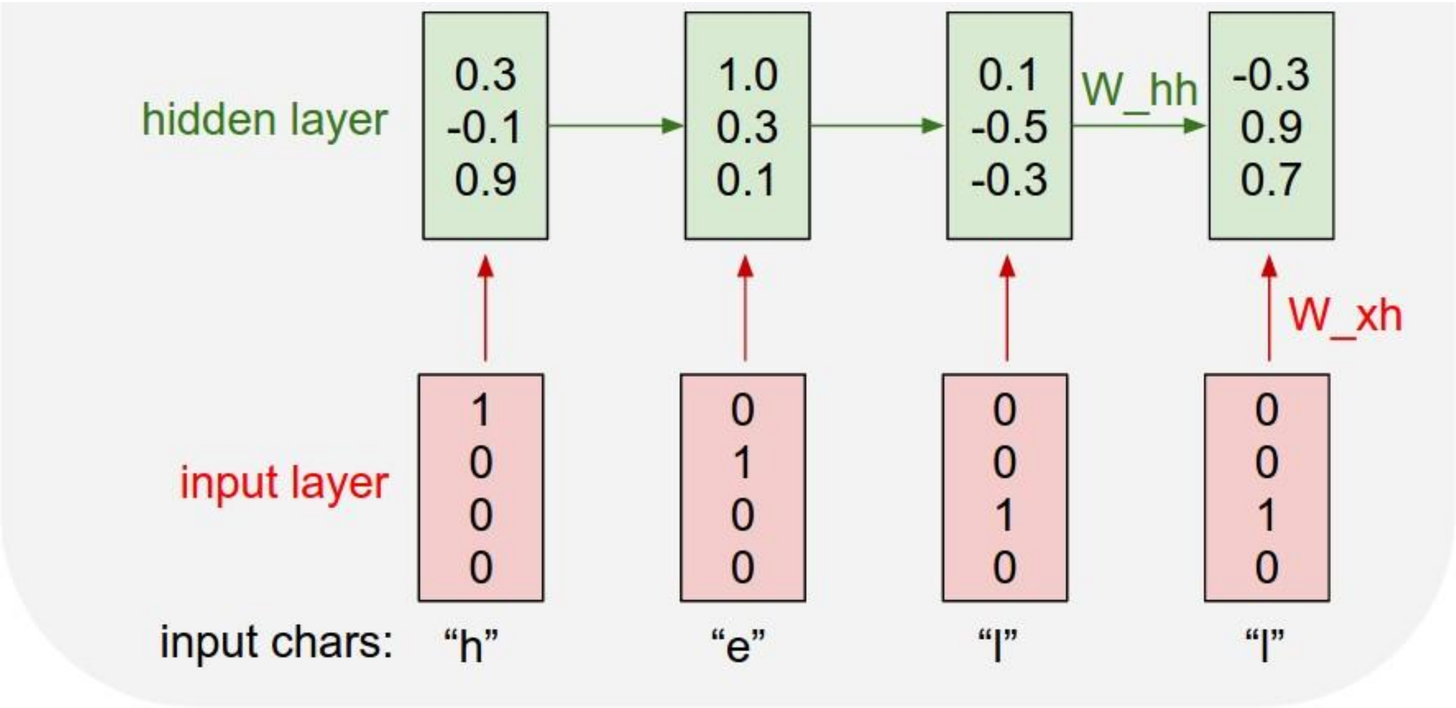


Example: Character-level Language Model

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

Vocabulary:
[h,e,l,o]

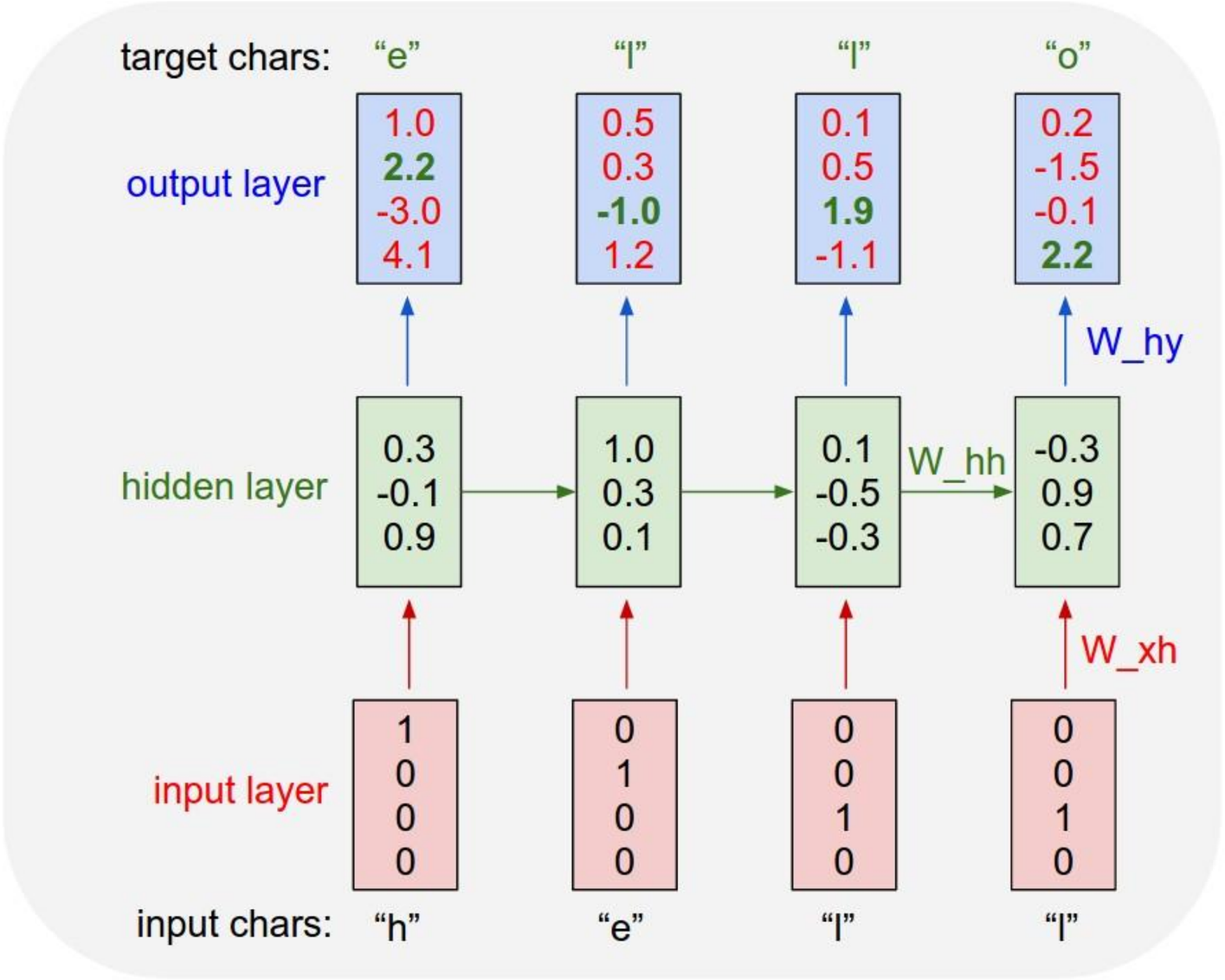
Example training sequence:
“hello”



Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

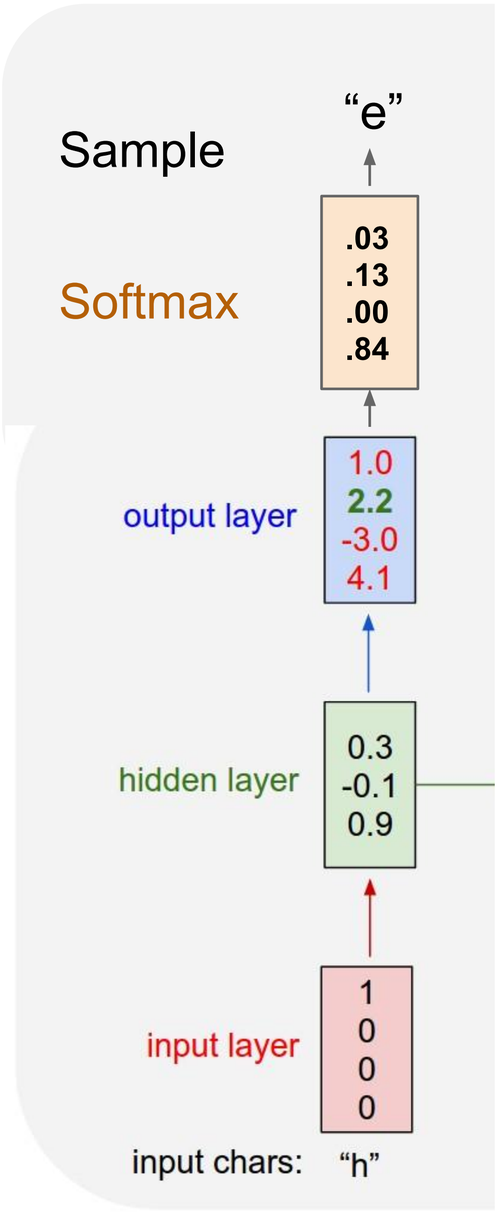
Example training sequence:
“hello”



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

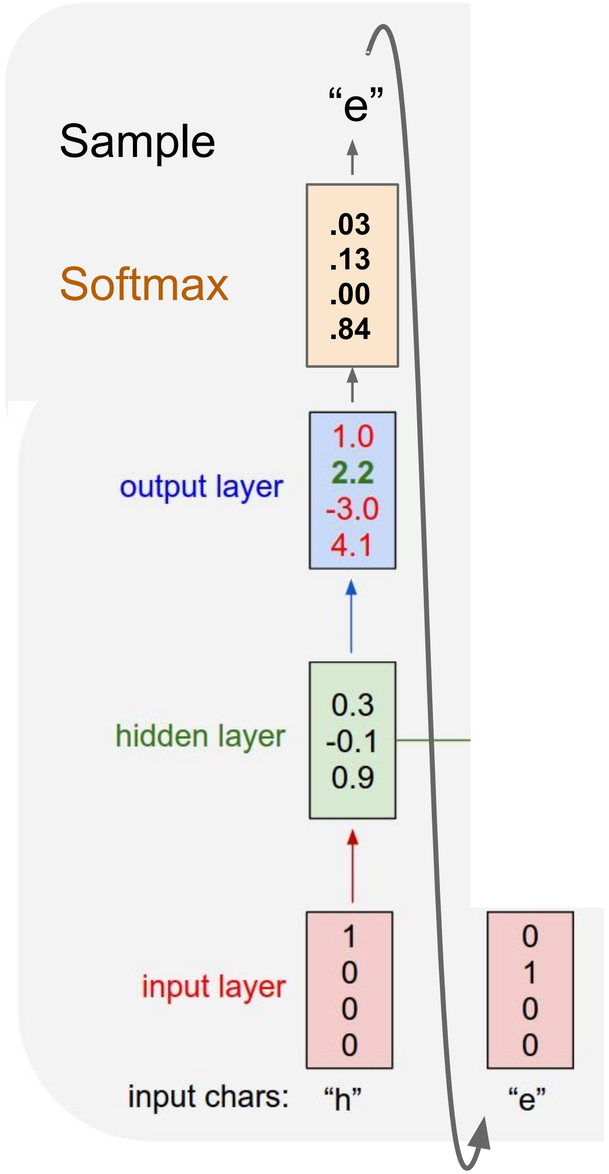
At test-time sample characters one at a time, feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

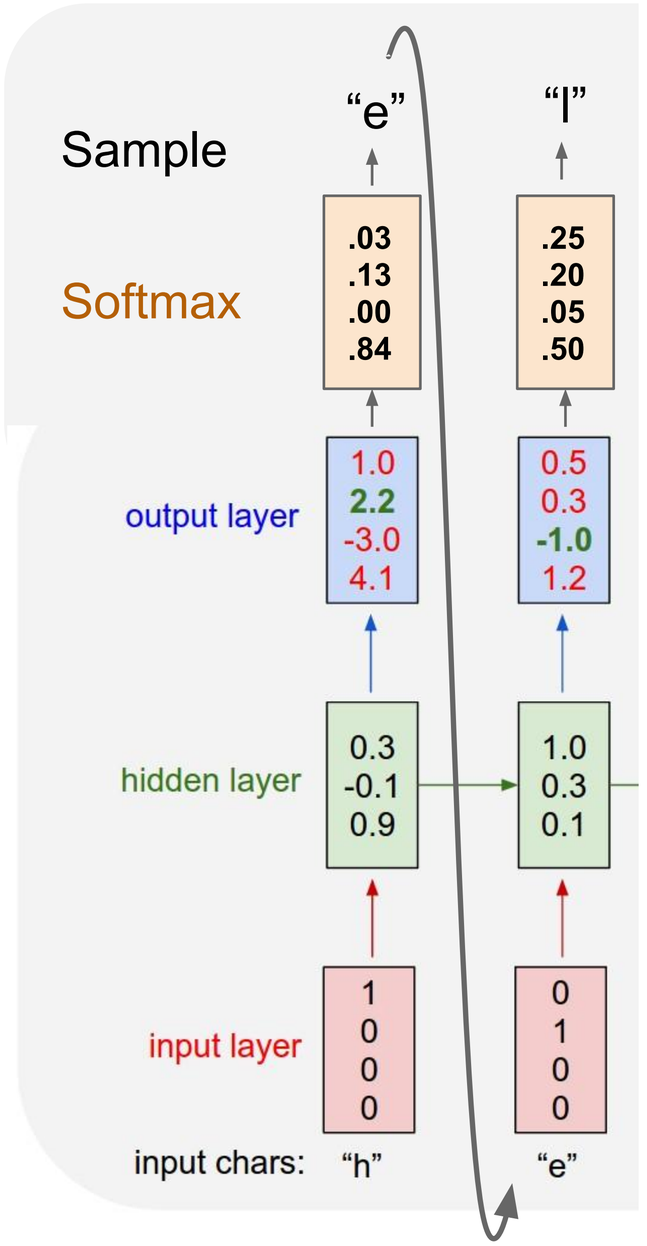
At test-time sample characters one at a time, feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

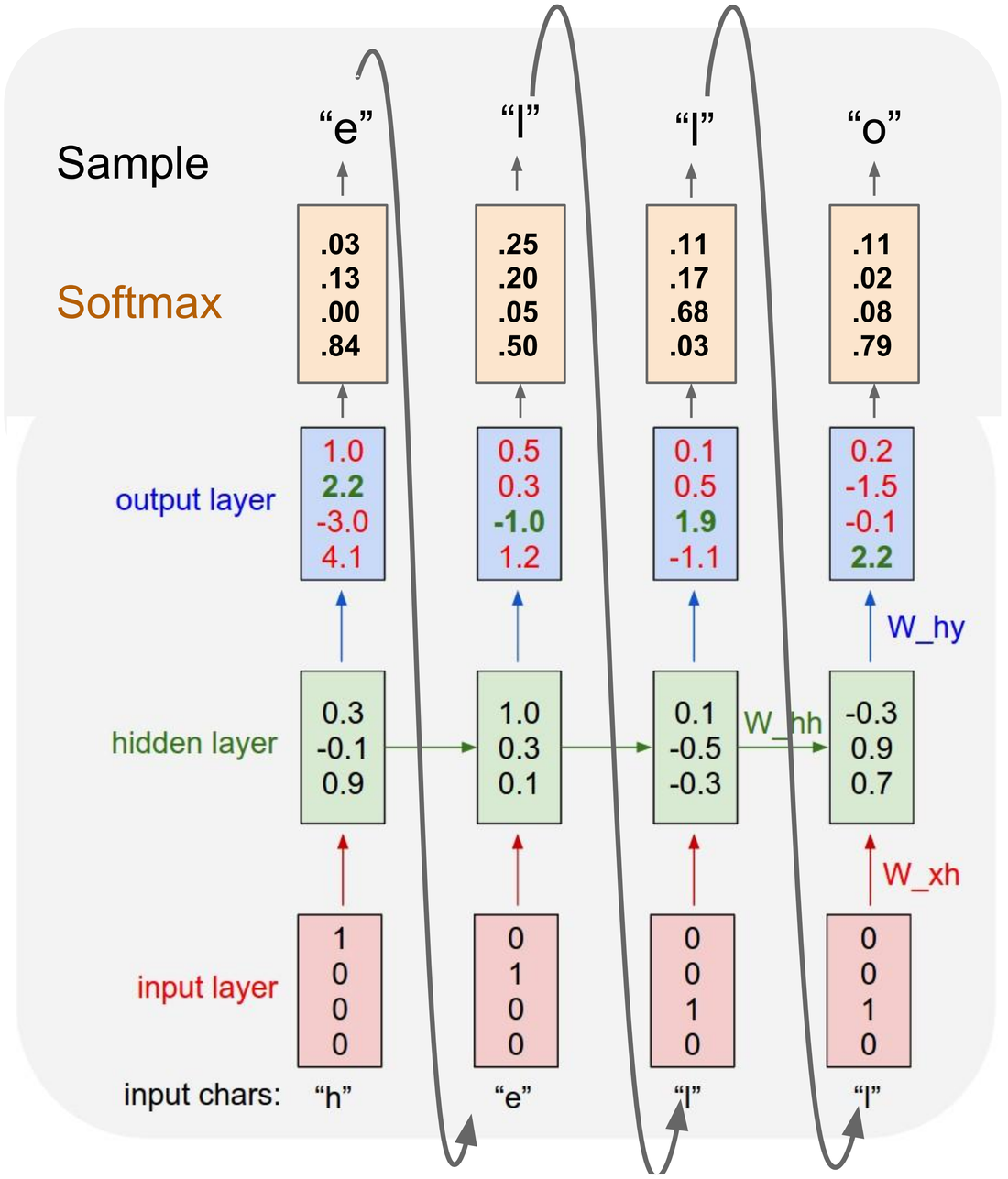
At test-time sample characters one at a time, feed back to model



Example: Character-level Language Model Sampling

Vocabulary:
[h,e,l,o]

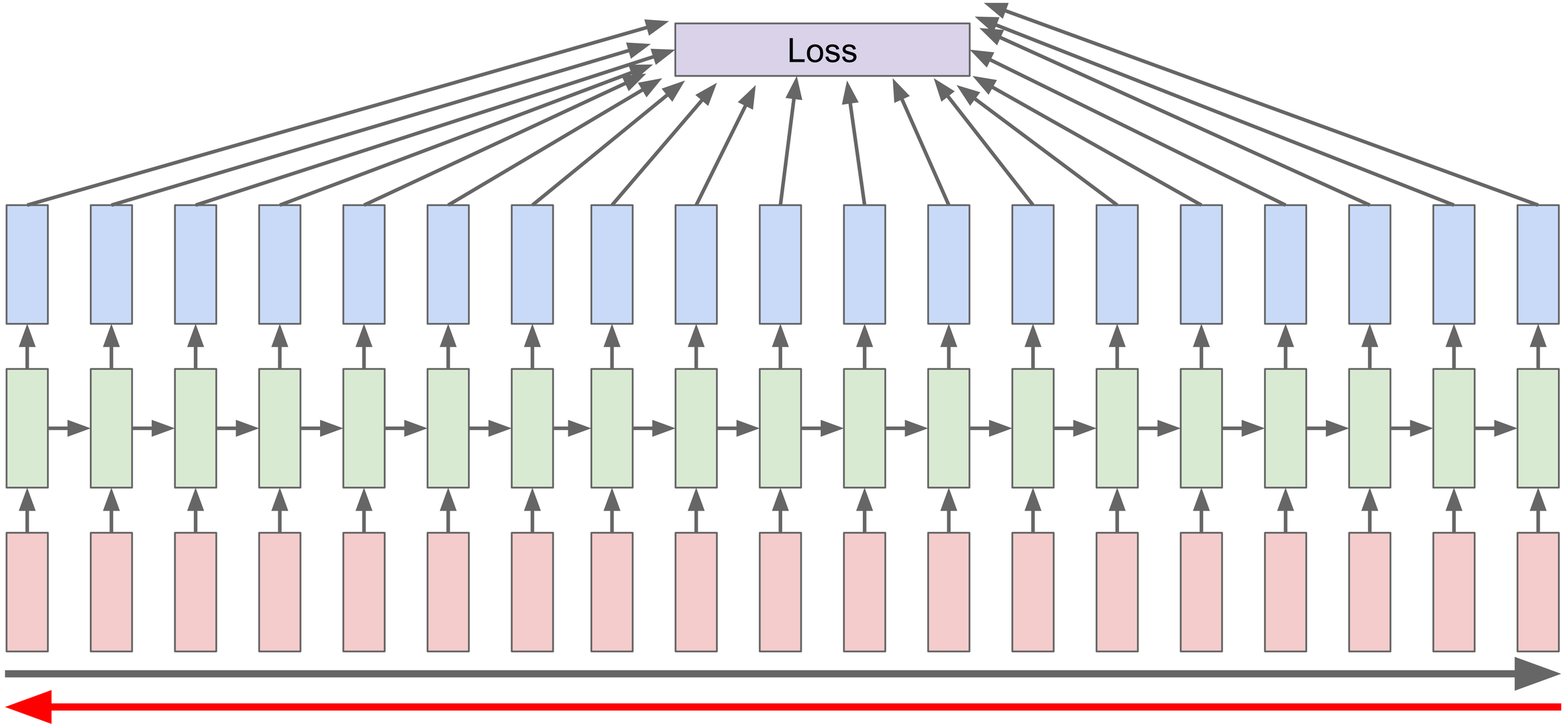
At test-time sample characters one at a time, feed back to model



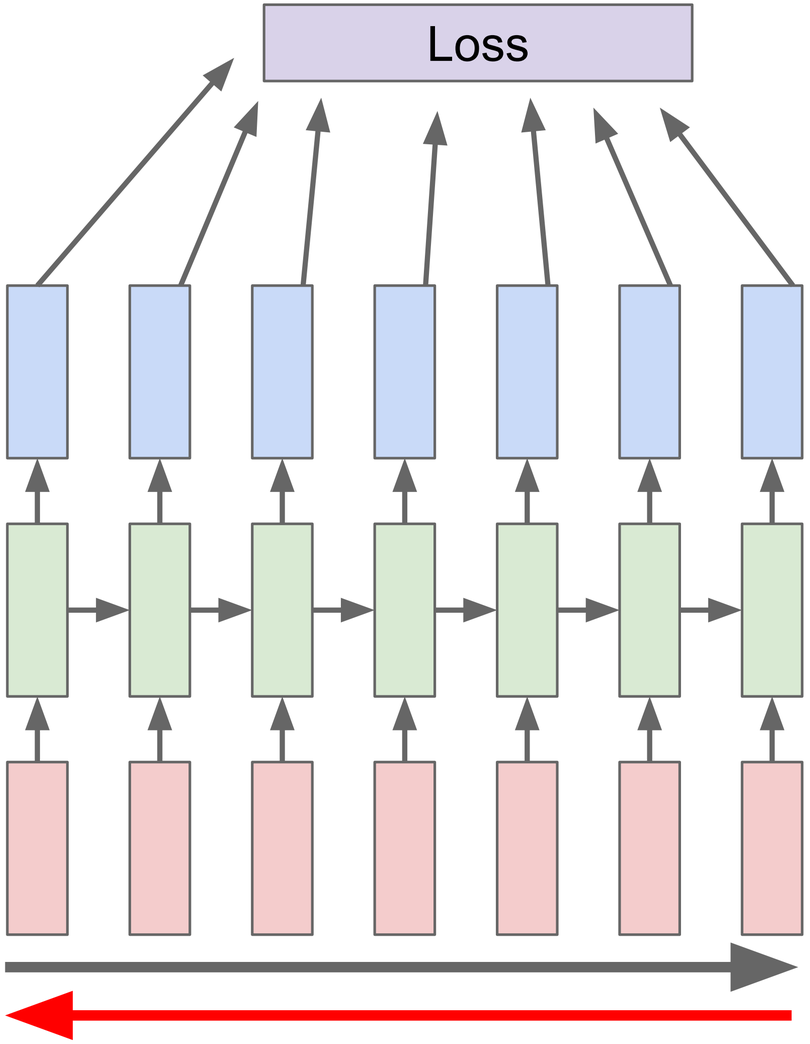
That’s all well and good, but how to learn those weights

Backpropagation through time

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

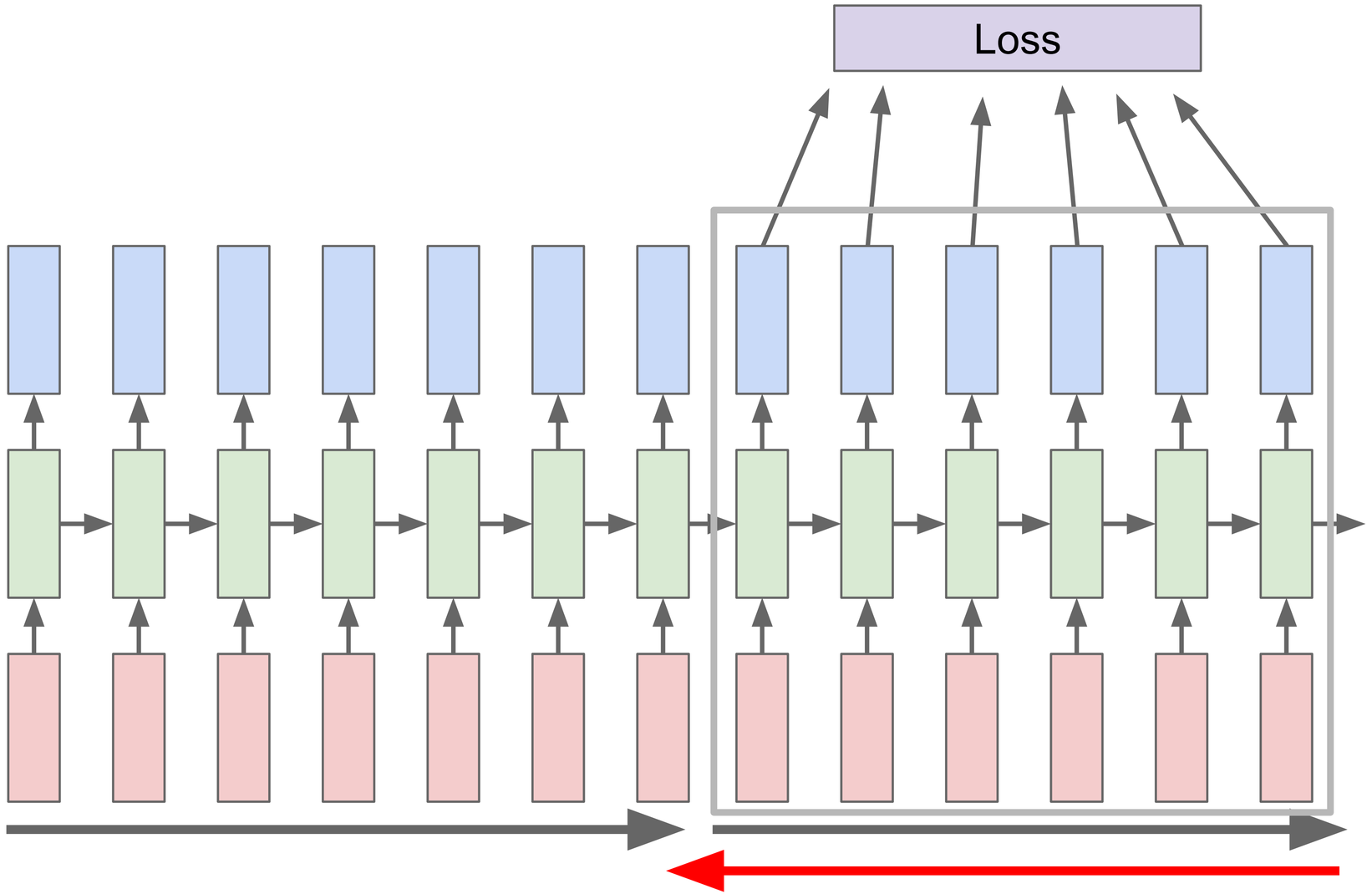


Truncated Backpropagation through time



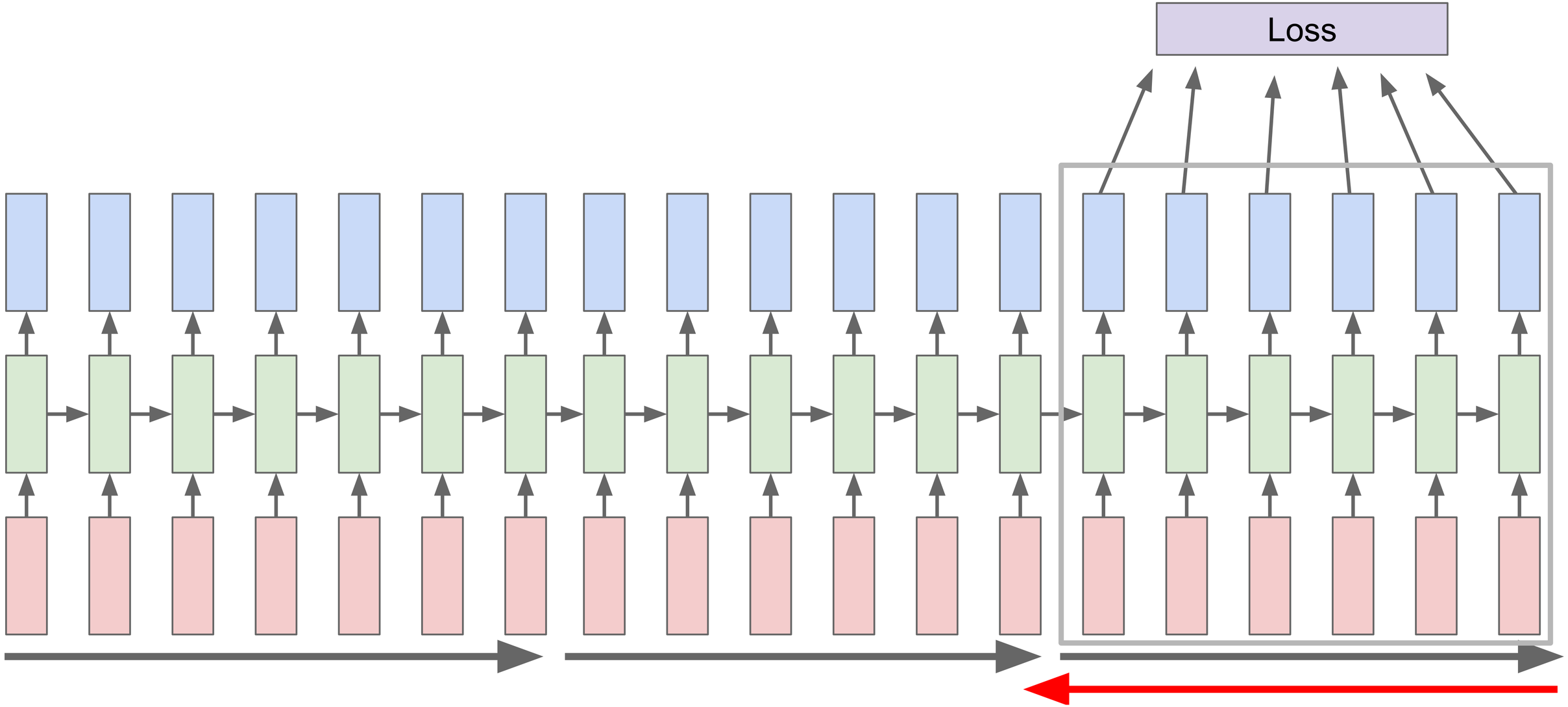
Run forward and backward through chunks of the sequence instead of whole sequence

Truncated Backpropagation through time



Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Truncated Backpropagation through time

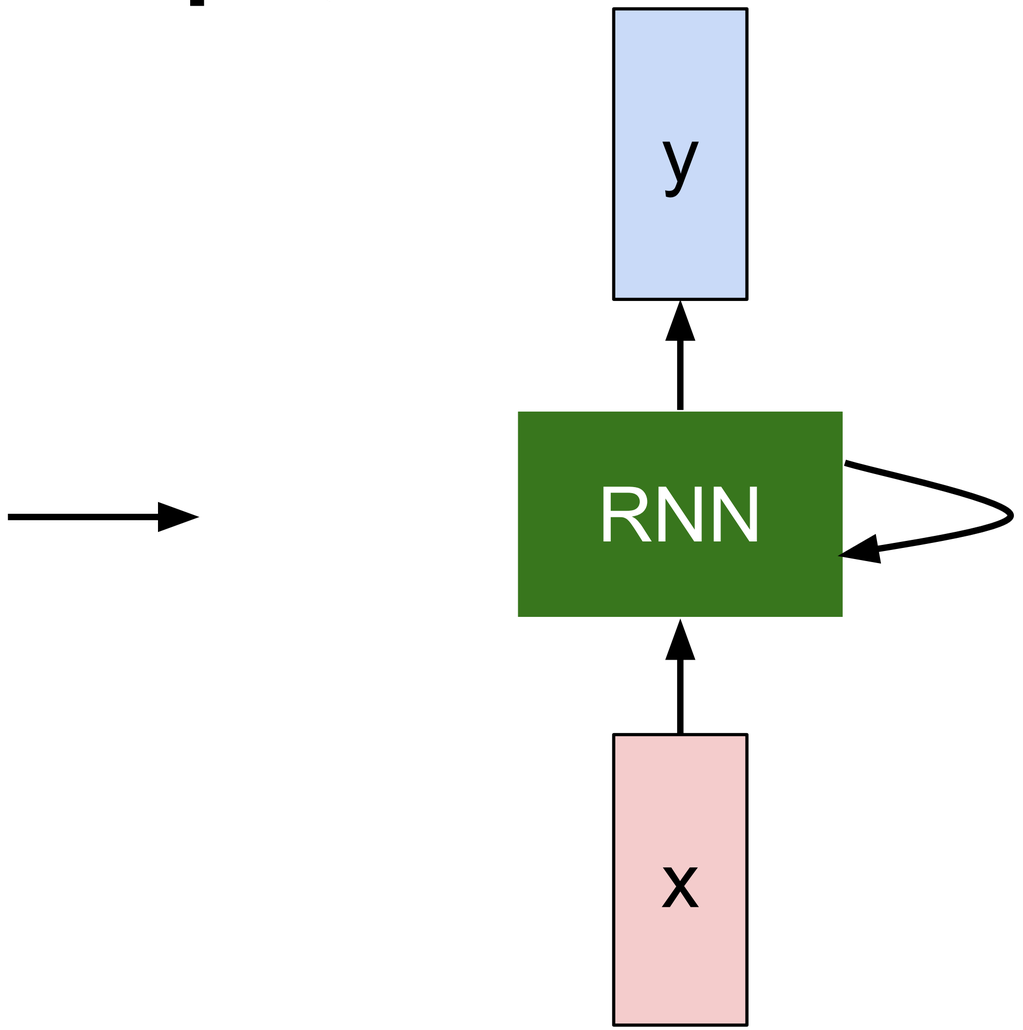


THE SONNETS, (fun-ish example)

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.



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.
```

```
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.
```

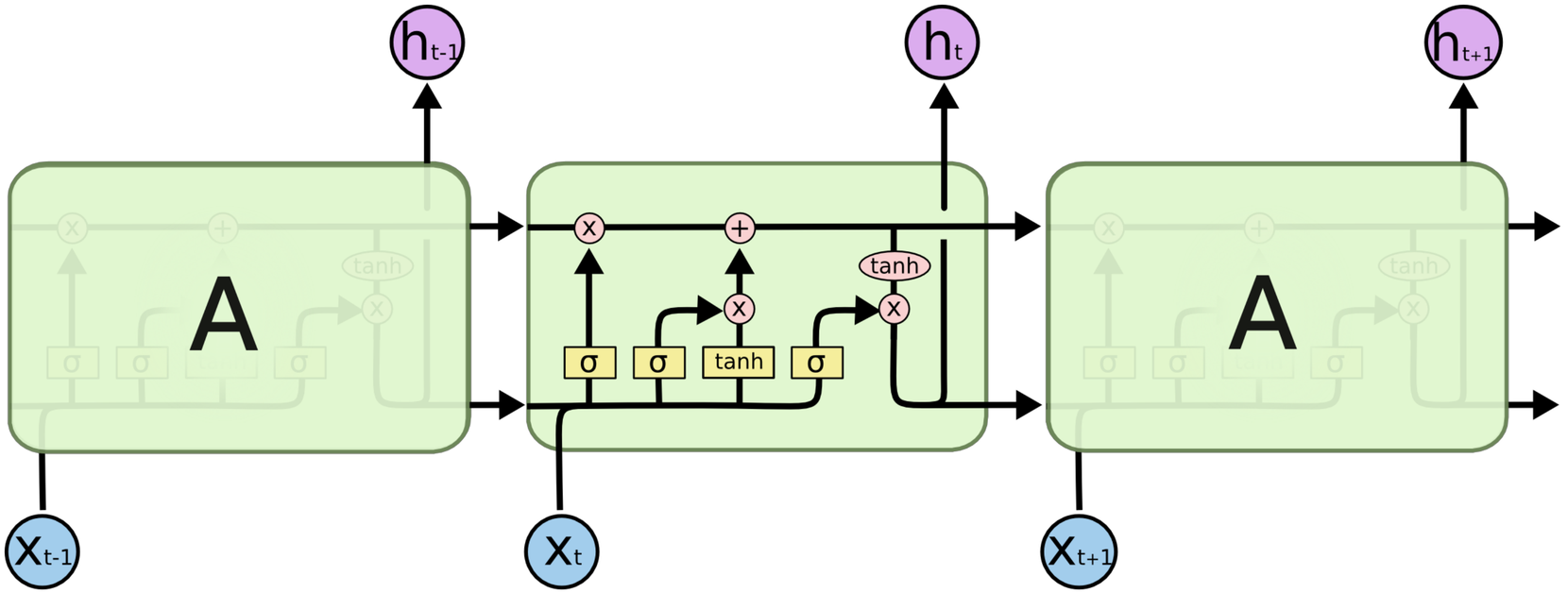
- How to make all of this better?
- How to learn better representations?
- Is it just the matter of estimating a better (long-term) context vector?

The Problem of Long-Term Dependencies

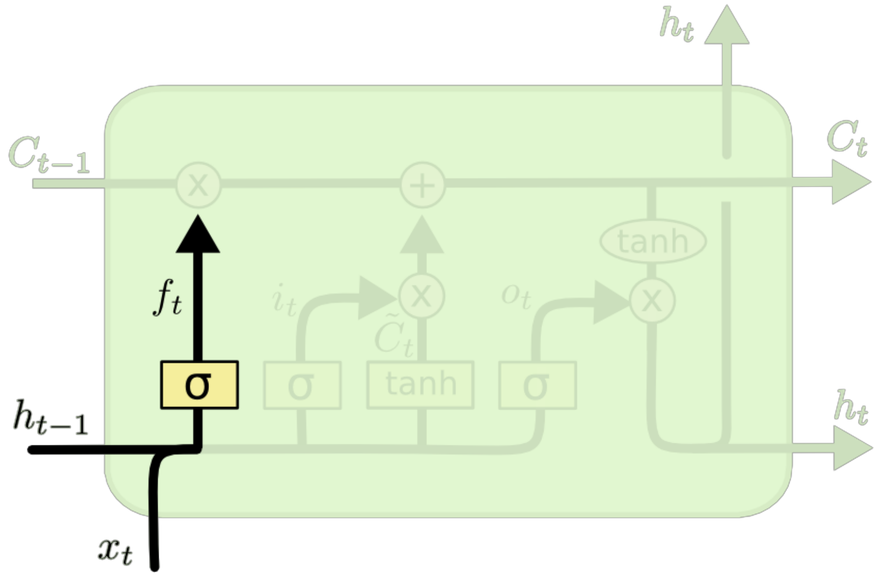
- Sometimes, we only need to take a look at the **very recent information** to predict the next token
- But there are also cases where we **need more context**
- *This **robot** {**is**, are}*
- *The **users** he mentioned {is, **are**}*
- *many **systems,** in addition to VBG a page of free text for each knowledge element, also {**permit**, permits}.* *[Yogatama, et.al. 2018]*

The Problem of Long-Term Dependencies

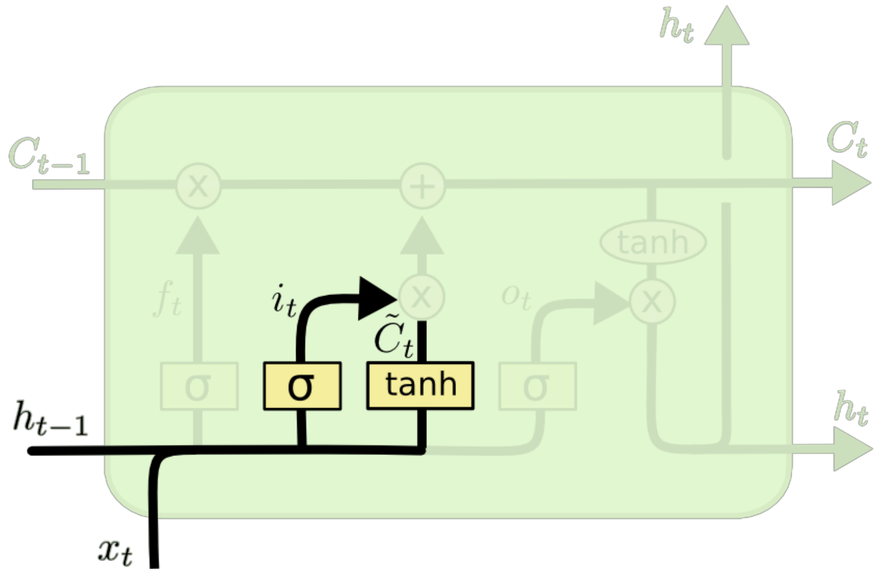
- As the **gap grows**, simple RNNs become unable to learn to **connect the information**
- Long Short Term Memory (LSTM) Networks don't have this problem [Hochreiter & Schmidhuber 1997]
- Just like RNNs, LSTMs also have a chain like structure, but with a different architecture.
- The key to LSTMs is the cell state



LSTM Walk Through



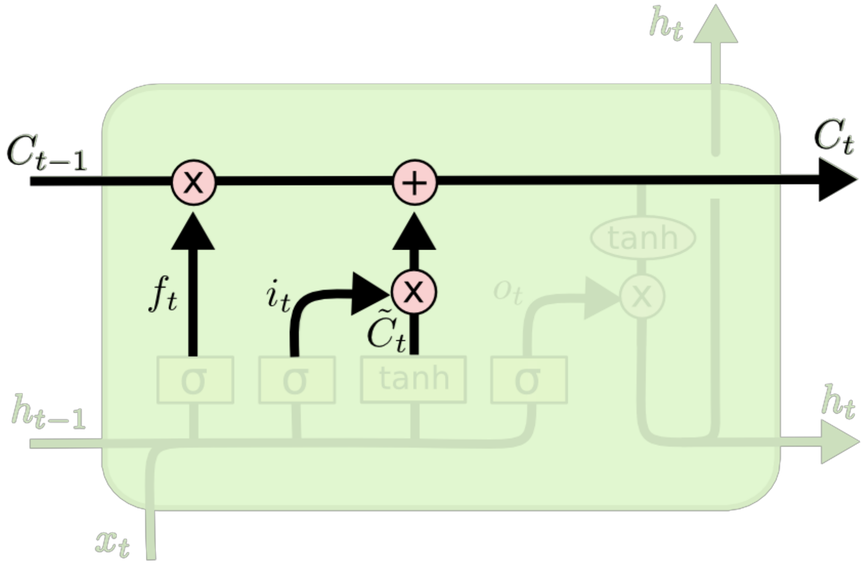
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$



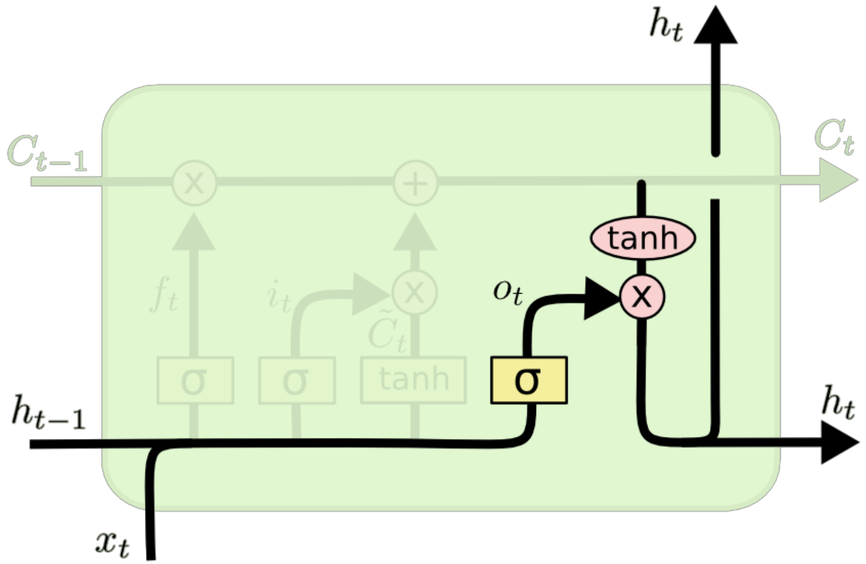
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$

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

LSTM Walk Through



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



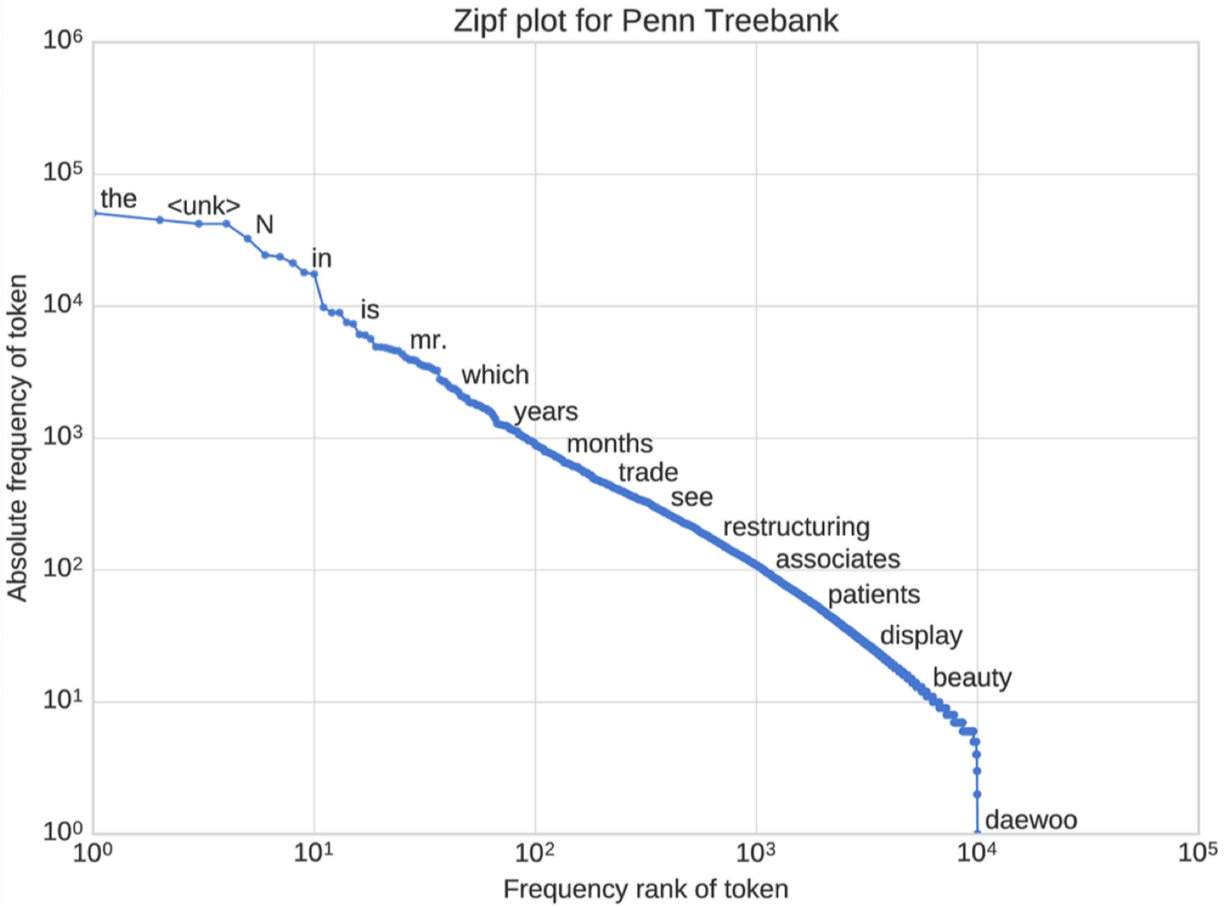
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

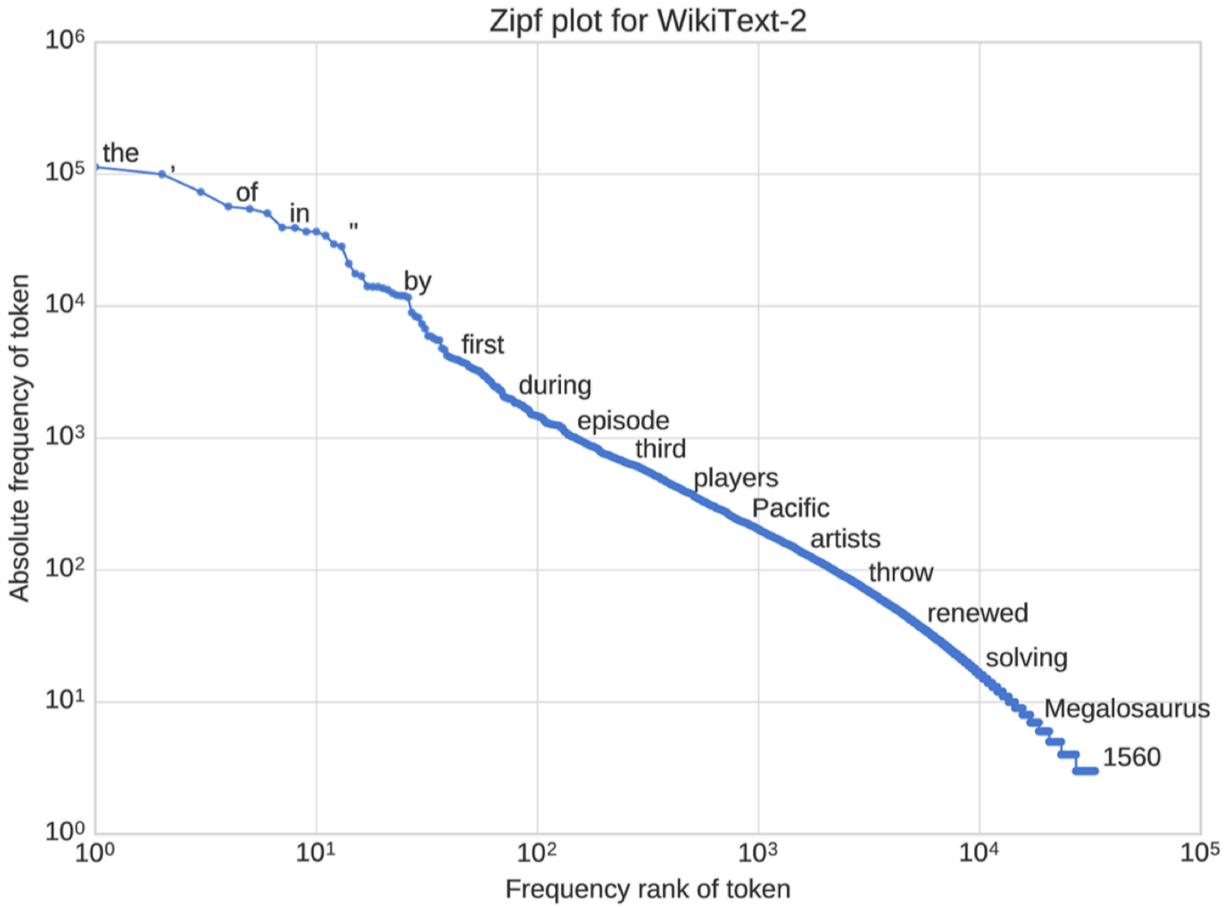
$$h_t = o_t * \tanh\left(C_t\right)$$

Language Modelling in Literature

- Datasets
  - Penn Treebank (PTB)
    - word level
    - pre-processed (lower-cased, numbers -> N, \n to *<eos>*, punctuations removed)
    - 929k training words, 73k valid words and 82k test words
    - 10k vocabulary (the rest are <unk>) - OoV rate: 4.8%
  - WikiText-103
    - 103 million words
    - One tenth of the One Billion Word Benchmark
    - features punctuation
    - original casing
    - larger vocabulary - OoV rate: 0.4%
    - numbers

Zipfian: PTB vs WikiText-2





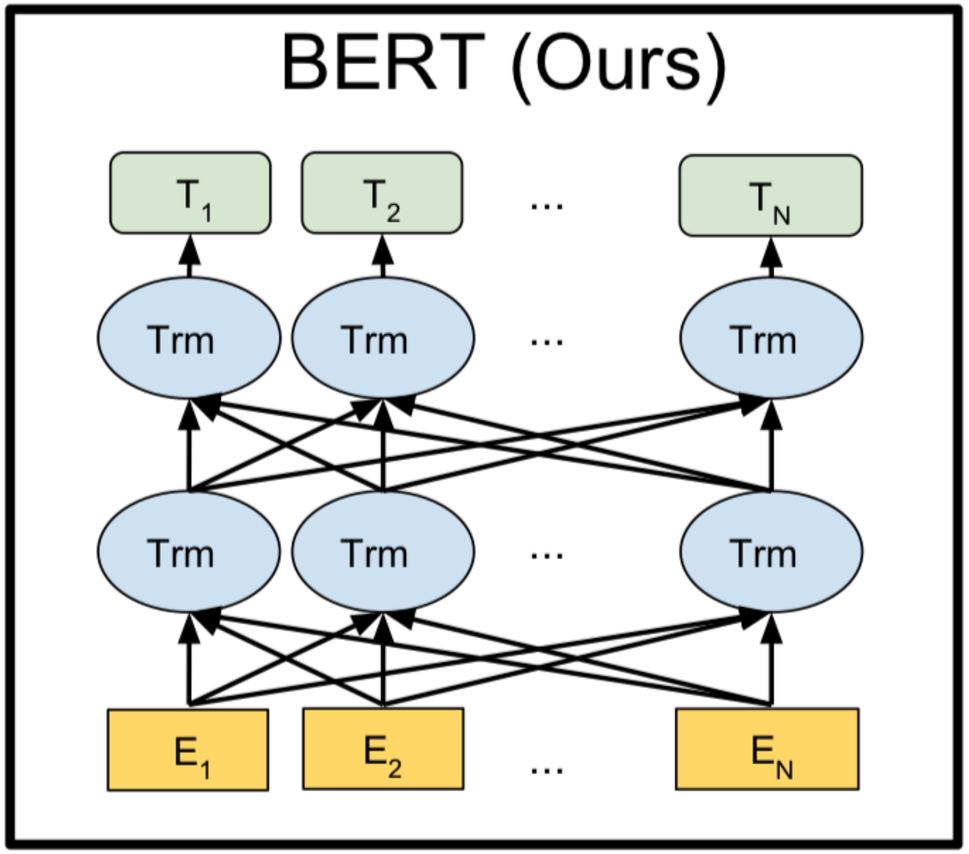
Zipf's law states that given some corpus of natural language utterances, **the frequency of any word** is inversely proportional to its **rank in the frequency table.**

Popular Regularization Techniques

- LSTM Dropout
- Embedding Dropout
- Weight Tying
- Variable Length Back-propagation Sequence
- ... [Merity et.al. 2017]

Pre-training with language modelling helps a lot

- Importance of transfer learning (knowledge from one domain to another)
- Rich, unsupervised pre-training is an integral part of many language understanding systems (e.g. QA, NMT …)
- Low resource tasks can benefit from deep architectures
- While the empirical results are strong, no investigation of the linguistic phenomena captured by their model (BERT)

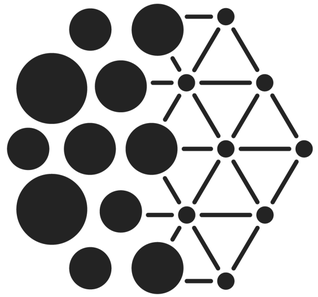


Your cup of tea

Yang, et.al. 2018

Thanks, Questions?

Fall - 2018



Mila

Language Modelling as Matrix Factorization

$$\mathbf{H}_\theta = \begin{bmatrix} \mathbf{h}_{c_1}^\top \\ \mathbf{h}_{c_2}^\top \\ \cdots \\ \mathbf{h}_{c_N}^\top \end{bmatrix}; \ \mathbf{W}_\theta = \begin{bmatrix} \mathbf{w}_{x_1}^\top \\ \mathbf{w}_{x_2}^\top \\ \cdots \\ \mathbf{w}_{x_M}^\top \end{bmatrix}; \ \mathbf{A} = \begin{bmatrix} \log P^*(x_1|c_1), & \log P^*(x_2|c_1) & \cdots & \log P^*(x_M|c_1) \\ \log P^*(x_1|c_2), & \log P^*(x_2|c_2) & \cdots & \log P^*(x_M|c_2) \\ \vdots & \vdots & \ddots & \vdots \\ \log P^*(x_1|c_N), & \log P^*(x_2|c_N) & \cdots & \log P^*(x_M|c_N) \end{bmatrix}$$

where $\mathbf{H}_\theta \in \mathbb{R}^{N \times d}$, $\mathbf{W}_\theta \in \mathbb{R}^{M \times d}$, $\mathbf{A} \in \mathbb{R}^{N \times M}$, and the rows of $\mathbf{H}_\theta$, $\mathbf{W}_\theta$, and $\mathbf{A}$ correspond to context vectors, word embeddings, and log probabilities of the true data distribution respectively.

what we model $\longrightarrow$ $H_\theta W_\theta^T = A'$

rank of left hand side is upper bounded by embedding size ***d***

They hypothesize that Natural Language Is High-Rank (log probablity matrix ***A***)

Yang, et.al. 2018

References

- http://cs231n.stanford.edu/slides/2018/cs231n_2018_lecture10.pdf

- http://colah.github.io/posts/2015-08-Understanding-LSTMs/