Autumn 2024, Part 4

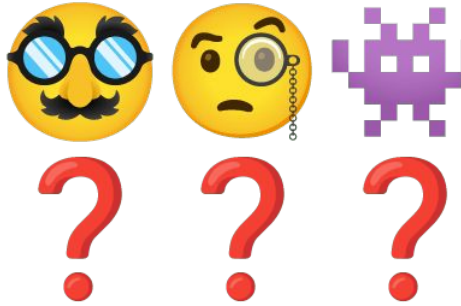
RNN, LSTM

Eliza Vialykh, @elf_lesnoy

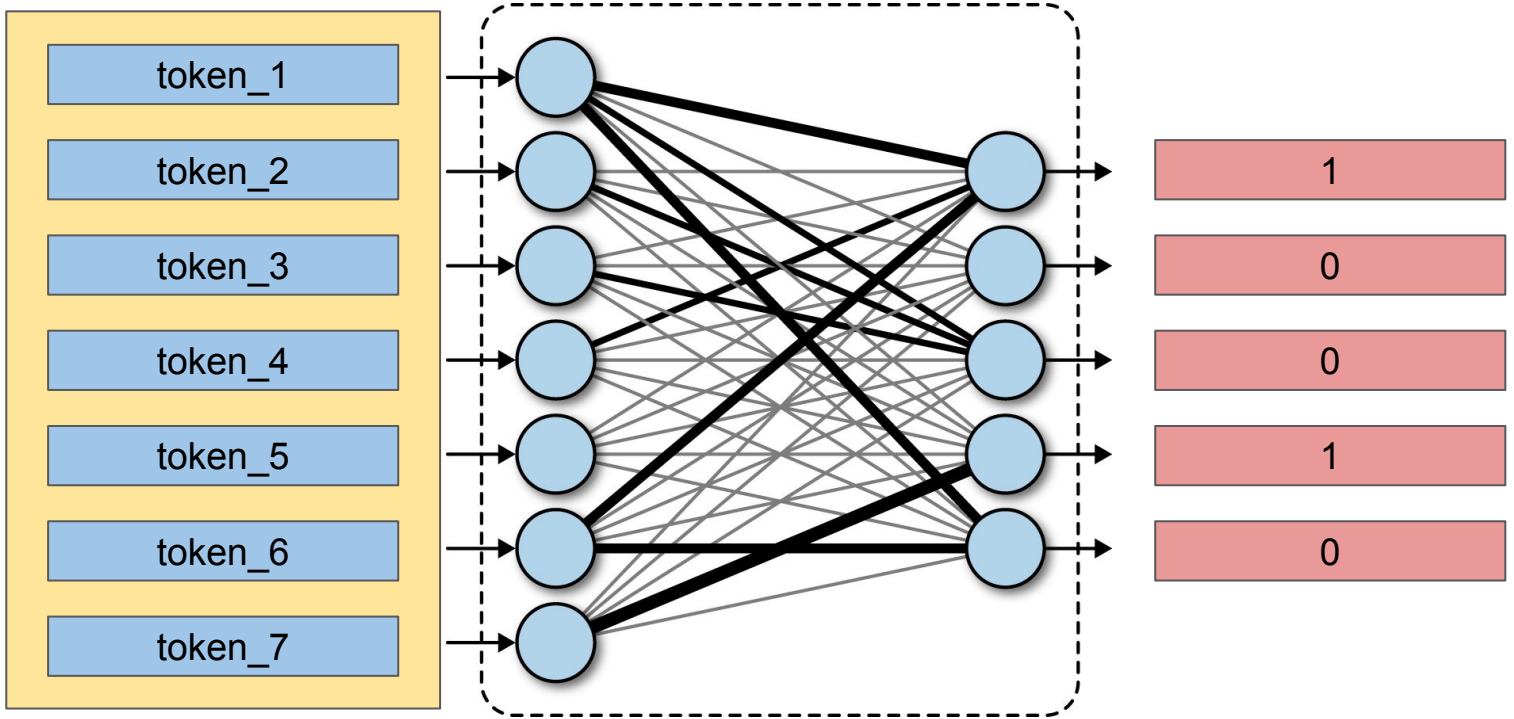
What is the fundamental difference between these tasks?

- Named Entity Recognition (NER)
- Machine Translation
- Speech Recognition
- Language Modeling

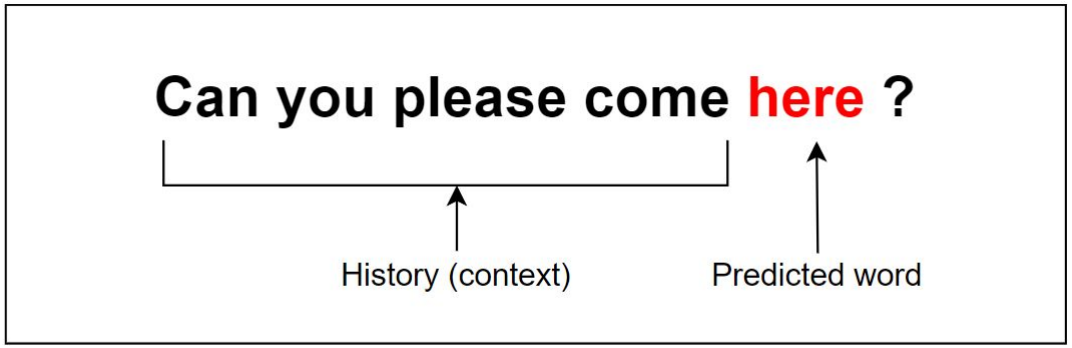
- Document Classification
- Topic Modeling
- Text Similarity/Document Similarity

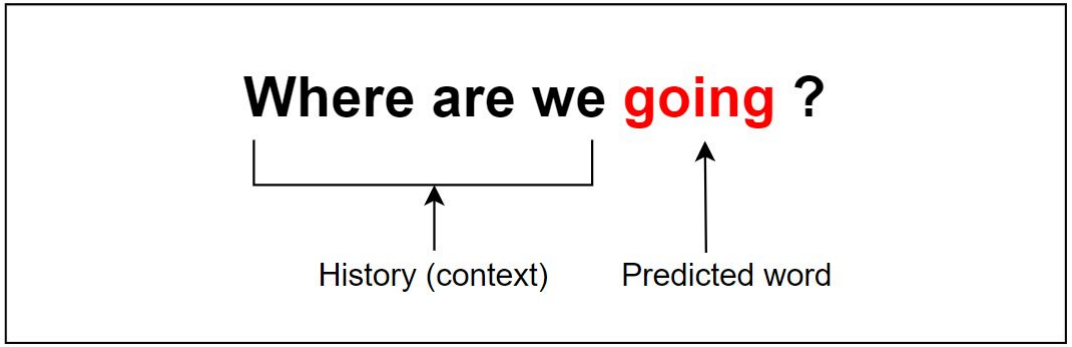


Why wasn't the classical neural network suitable for us?



Language modeling





Language modeling

Please give me my...

Language modeling

Please give me my...

Language modeling

Please give me my...

I want to eat. Please give me my...

Language modeling

Please give me my...

I want to eat. Please give me my...

I see my apple cake. I want to eat. Please give me my... 

Language modeling

Please give me my...

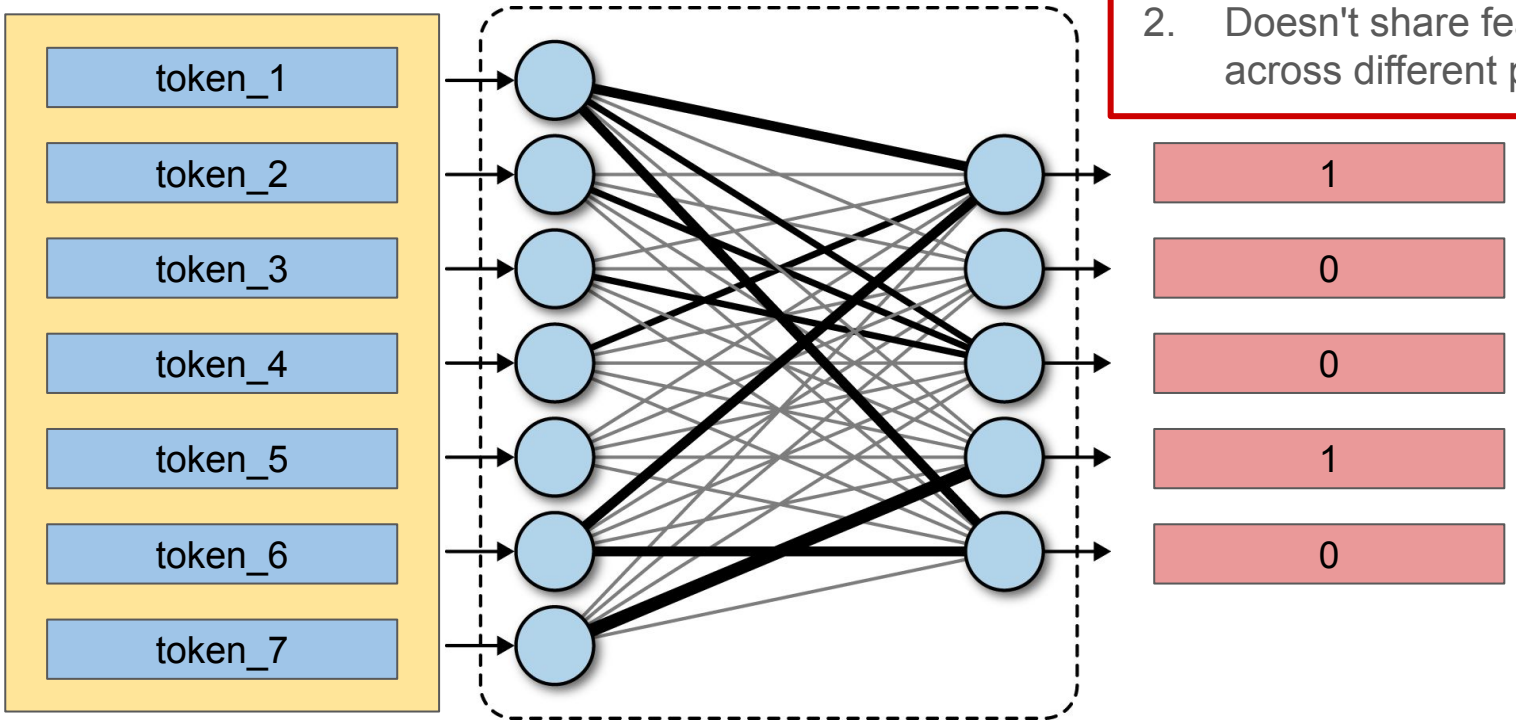
I want to eat. Please give me my...

I see my apple cake. I want to eat. Please give me my... 

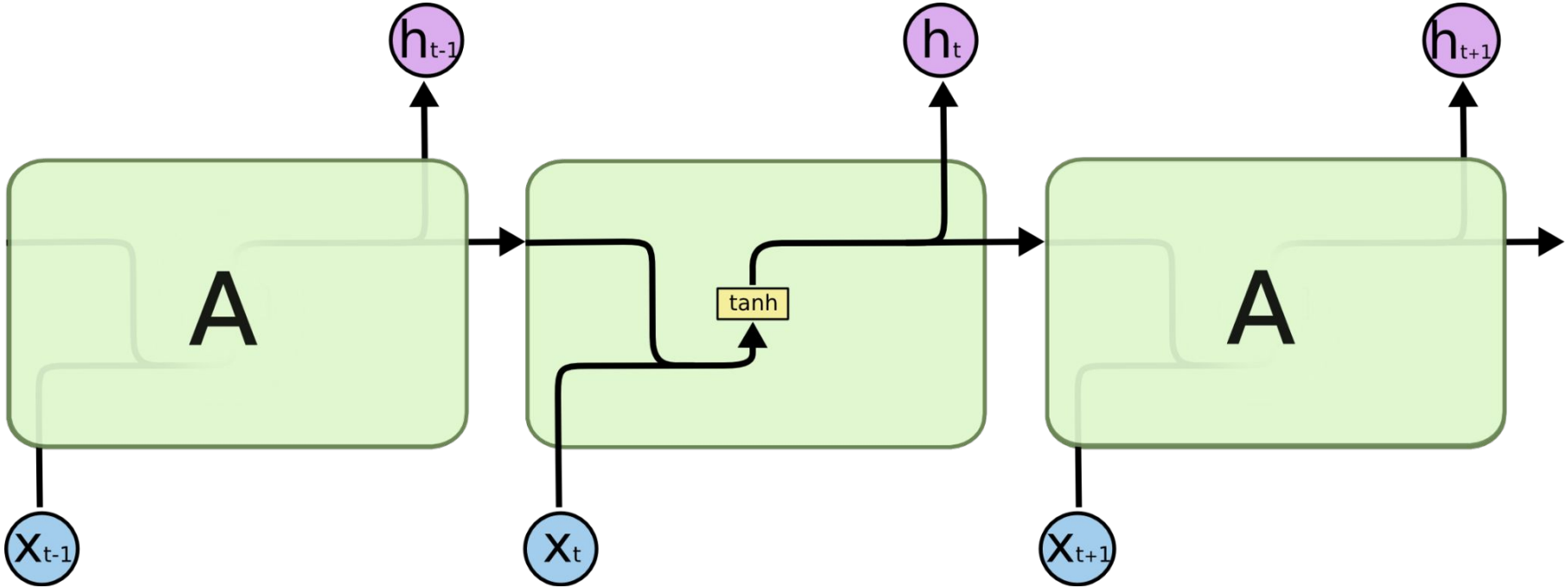
$P(w|h)$ probability of a word w given some history h.

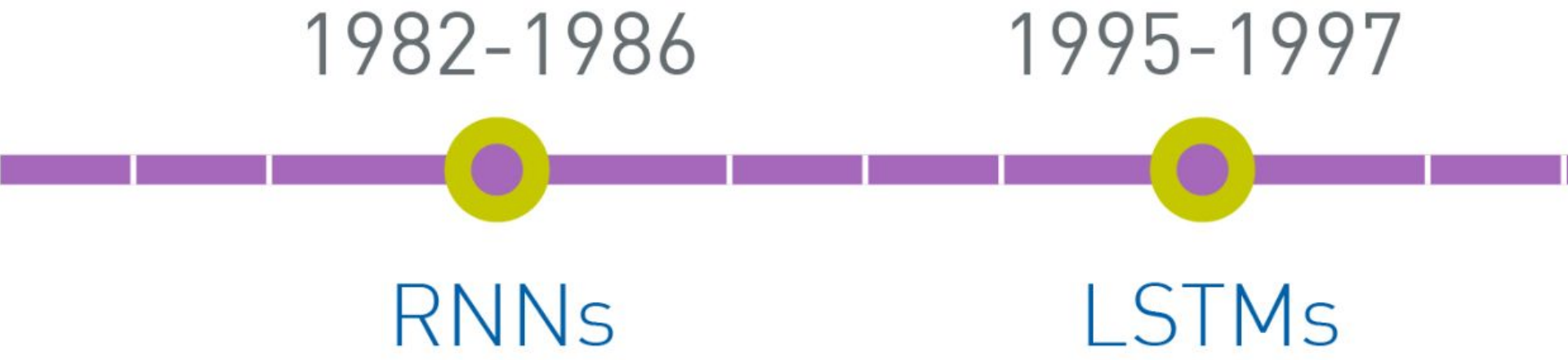
Why wasn't the classical neural network suitable for us?

1. Inputs and outputs can be different lengths
2. Doesn't share features learned across different positions of texts



RNN




1982-1986
1995-1997
RNNs
LSTMs

For $\bigoplus_{n=1,\ldots,m}$ where $\mathcal{L}_{m_\bullet} = 0$, hence we can find a closed subset $\mathcal{H}$ in $\mathcal{H}$ and any sets $\mathcal{F}$ on X, U is a closed immersion of S, then $U \to T$ is a separated algebraic space.

Proof. Proof of (1). It also start we get

$$S = \mathrm{Spec}(R) = U \times_X U \times_X U$$

and the compariciely in the fibre product covering we have to prove the lemma generated by $\coprod Z \times_U U \to V$. Consider the maps M along the set of points Sch_{fppf} and $U \to U$ is the fibre category of S in U in Section, ?? and the fact that any U affine, see Morphisms, Lemma ??. Hence we obtain a scheme S and any open subset $W \subset U$ in $Sh(G)$ such that $\mathrm{Spec}(R') \to S$ is smooth or an

$$U = \bigcup U_i \times_{S_i} U_i$$

which has a nonzero morphism we may assume that f_i is of finite presentation over S. We claim that $\mathcal{O}_{X,x}$ is a scheme where $x, x', s'' \in S'$ such that $\mathcal{O}_{X,x'} \to \mathcal{O}'_{X',x'}$ is separated. By Algebra, Lemma ?? we can define a map of complexes $\mathrm{GL}_{S'}(x'/S'')$ and we win. □

To prove study we see that $\mathcal{F}|_U$ is a covering of $\mathcal{X}'$, and $\mathcal{T}_i$ is an object of $\mathcal{F}_{X/S}$ for $i > 0$ and $\mathcal{F}_p$ exists and let $\mathcal{F}_i$ be a presheaf of $\mathcal{O}_X$-modules on $\mathcal{C}$ as a $\mathcal{F}$-module. In particular $\mathcal{F} = U/\mathcal{F}$ we have to show that

$$\widetilde{M^\bullet} = \mathcal{I}^\bullet \otimes_{\mathrm{Spec}(k)} \mathcal{O}_{S,s} - i_X^{-1}\mathcal{F})$$

is a unique morphism of algebraic stacks. Note that

$$\mathrm{Arrows} = (Sch/S)^{opp}_{fppf}, (Sch/S)_{fppf}$$

and

$$V = \Gamma(S, \mathcal{O}) \longmapsto (U, \mathrm{Spec}(A))$$

is an open subset of X. Thus U is affine. This is a continuous map of X is the inverse, the groupoid scheme S.

Proof. See discussion of sheaves of sets. □

The result for prove any open covering follows from the less of Example ??. It may replace S by $X_{spaces,étale}$ which gives an open subspace of X and T equal to S_{Zar}, see Descent, Lemma ??. Namely, by Lemma ?? we see that R is geometrically regular over S.

Lemma 0.1. *Assume (3) and (3) by the construction in the description.*

Suppose $X = \lim |X|$ *(by the formal open covering* X *and a single map* $\underline{Proj}_X(\mathcal{A}) = \mathrm{Spec}(B)$ *over* U *compatible with the complex*

$$Set(\mathcal{A}) = \Gamma(X, \mathcal{O}_{X,\mathcal{O}_X}).$$

When in this case of to show that $\mathcal{Q} \to \mathcal{C}_{Z/X}$ *is stable under the following result in the second conditions of (1), and (3). This finishes the proof. By Definition ?? (without element is when the closed subschemes are catenary. If* T *is surjective we may assume that* T *is connected with residue fields of* S*. Moreover there exists a closed subspace* $Z \subset X$ *of* X *where* U *in* X' *is proper (some defining as a closed subset of the uniqueness it suffices to check the fact that the following theorem*

(1) f *is locally of finite type. Since* $S = \mathrm{Spec}(R)$ *and* $Y = \mathrm{Spec}(R)$.

Proof. This is form all sheaves of sheaves on X. But given a scheme U and a surjective étale morphism $U \to X$. Let $U \cap U = \coprod_{i=1,\ldots,n} U_i$ be the scheme X over S at the schemes $X_i \to X$ and $U = \lim_i X_i$. □

The following lemma surjective restrocomposes of this implies that $\mathcal{F}_{x_0} = \mathcal{F}_{x_0} = \mathcal{F}_{\mathcal{X},\ldots,0}$.

Lemma 0.2. *Let* X *be a locally Noetherian scheme over* S, $E = \mathcal{F}_{X/S}$. *Set* $\mathcal{I} = \mathcal{J}_1 \subset \mathcal{I}'_n$. *Since* $\mathcal{I}^n \subset \mathcal{I}^n$ *are nonzero over* $i_0 \leq \mathfrak{p}$ *is a subset of* $\mathcal{J}_{n,0} \circ \overline{A}_2$ *works.*

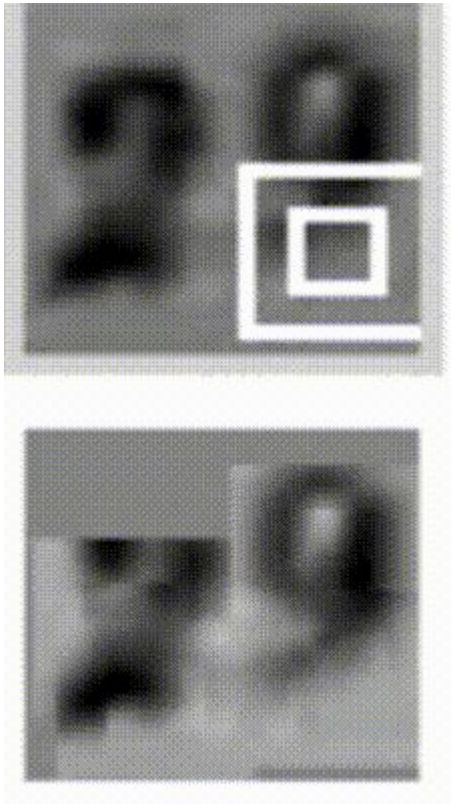
Lemma 0.3. *In Situation ??. Hence we may assume* $\mathfrak{q}' = 0$.

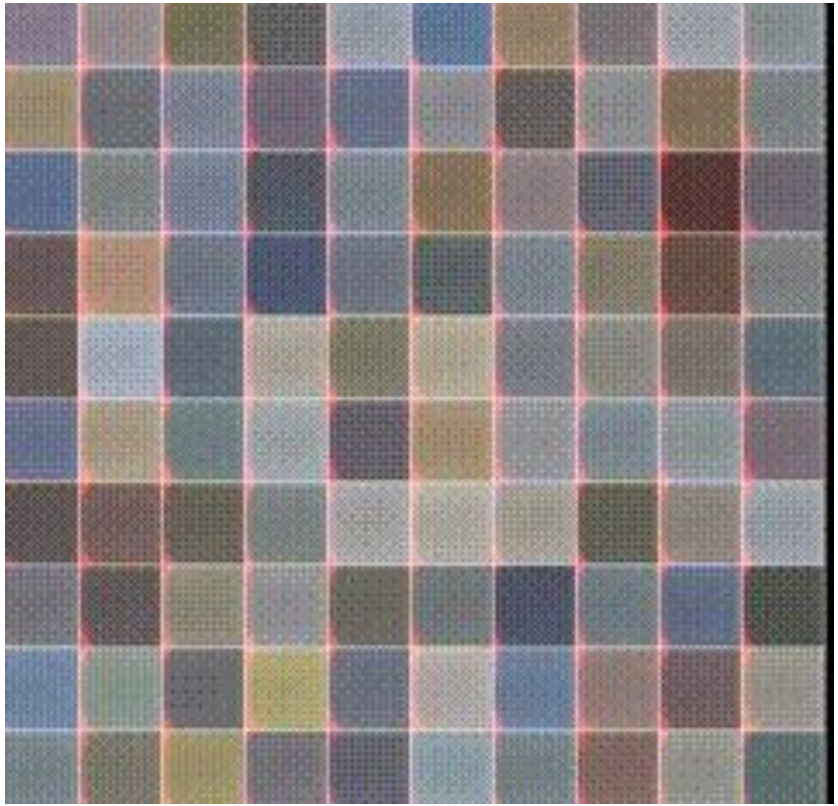
Proof. We will use the property we see that $\mathfrak{p}$ is the mext functor (??). On the other hand, by Lemma ?? we see that

$$D(\mathcal{O}_{X'}) = \mathcal{O}_X(D)$$

where K is an F-algebra where δ_{n+1} is a scheme over S. □

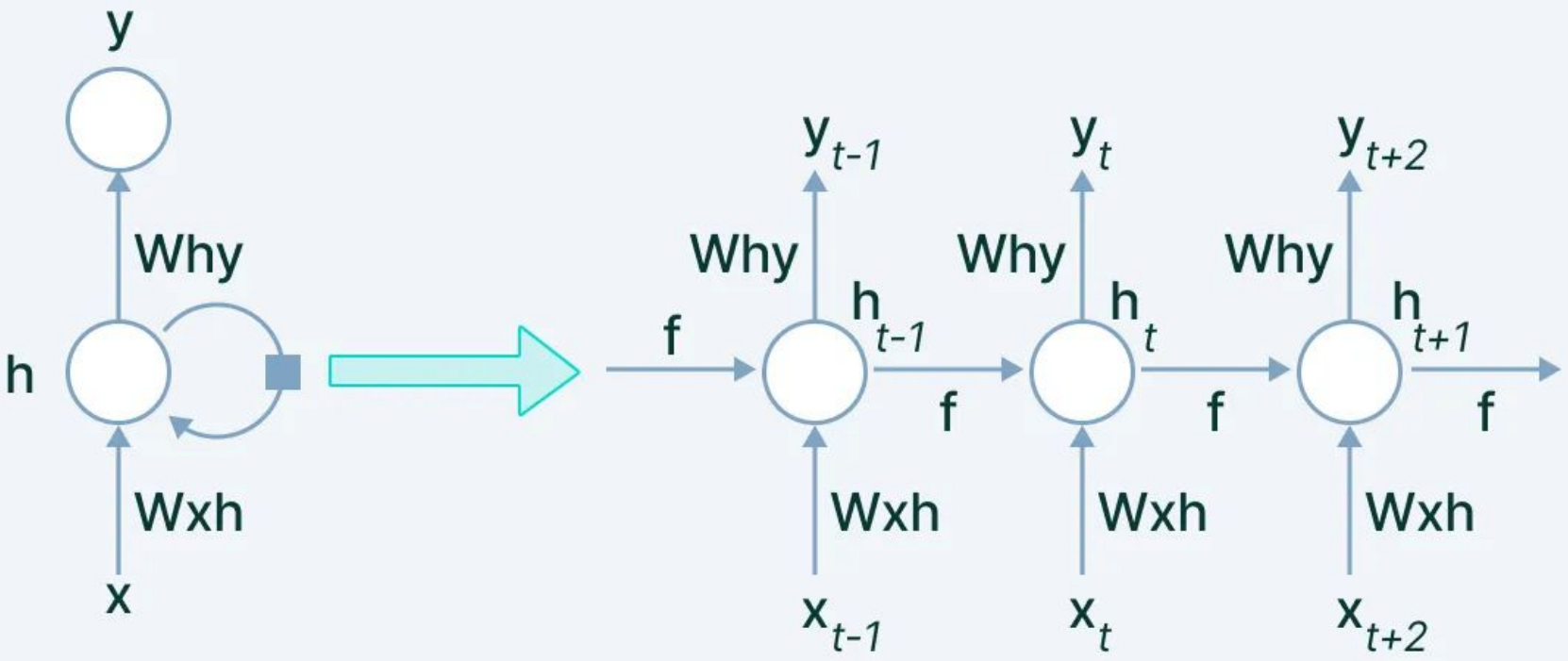
RNN





https://karpathy.github.io/2015/05/21/rnn-effectiveness/

RNN


y
Why
h
Wxh
x
f
y_{t-1}
Why
h_{t-1}
f
Wxh
x_{t-1}
y_t
Why
h_t
f
Wxh
x_t
y_{t+2}
Why
h_{t+1}
f
Wxh
x_{t+2}

RNN

“Teddy Roosevelt was a great president”

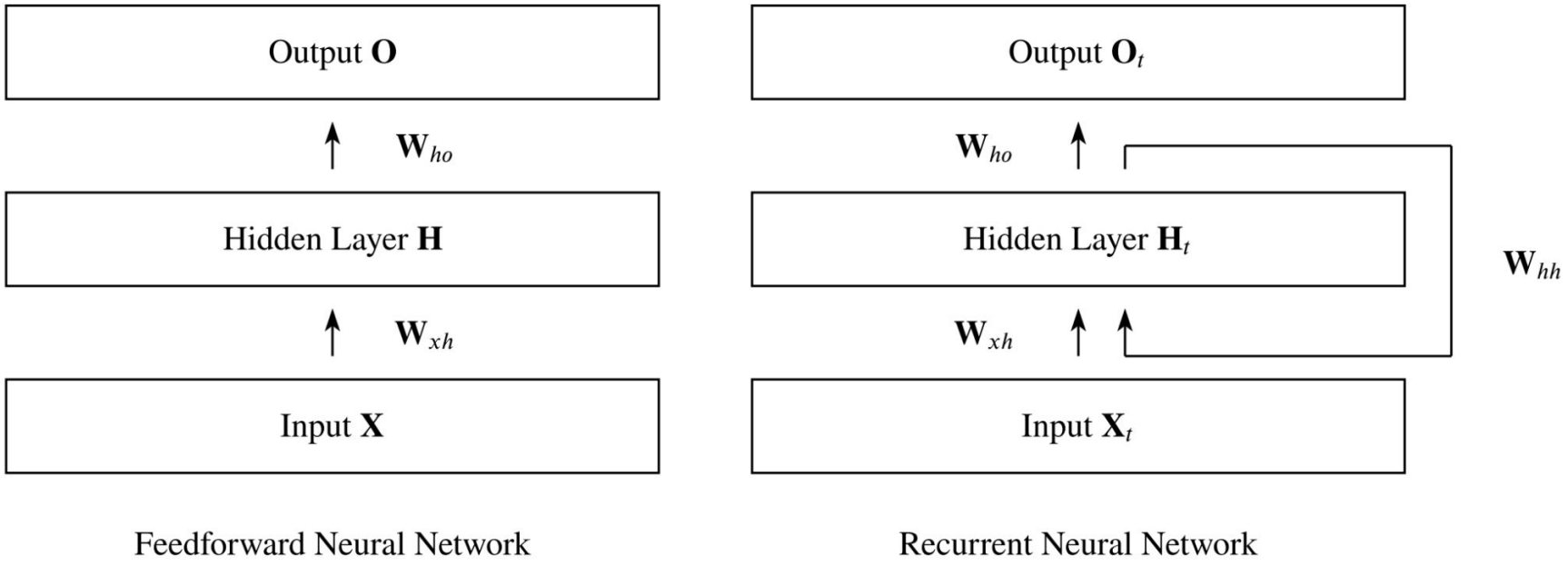
“Teddy bear are on sale!”

RNN

“Teddy Roosevelt was a great president”

“Teddy bear are on sale!”

RNN

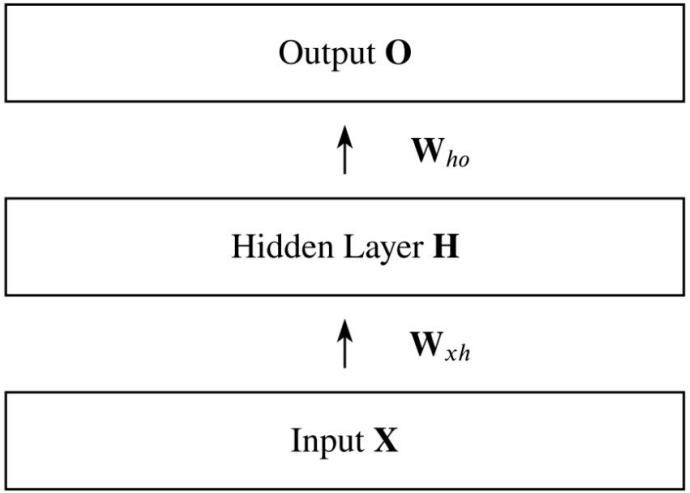


Feedforward Neural Network

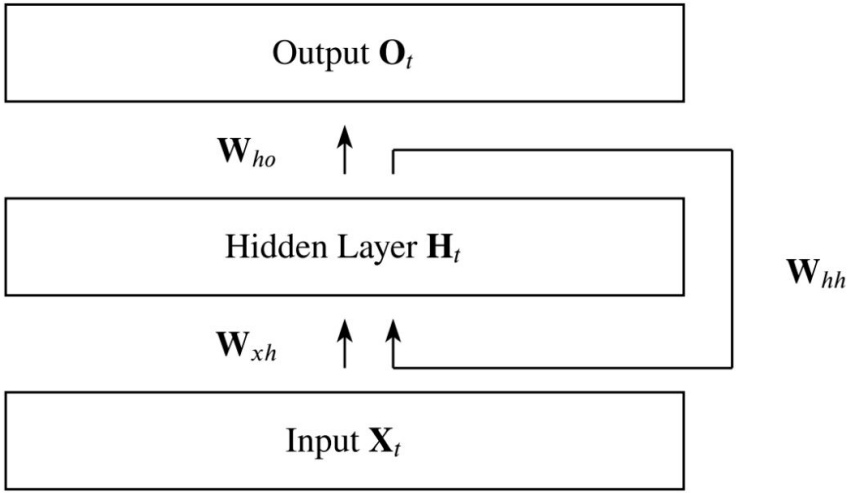
Recurrent Neural Network

Notice: RNN uses ***the same weights matrices for each time step*** instead of learning different matrices at different timesteps

RNN



Feedforward Neural Network



Recurrent Neural Network

$$\mathbf{H} = \phi_h \left(\mathbf{X}\mathbf{W}_{xh} + \mathbf{b}_h \right)$$

$$\mathbf{O} = \phi_o \left(\mathbf{H}\mathbf{W}_{ho} + \mathbf{b}_o \right)$$

$$\mathbf{H}_t = \phi_h \left(\mathbf{X}_t\mathbf{W}_{xh} + \mathbf{H}_{t-1}\mathbf{W}_{hh} + \mathbf{b}_h \right)$$

$$\mathbf{O}_t = \phi_o \left(\mathbf{H}_t\mathbf{W}_{ho} + \mathbf{b}_o \right)$$

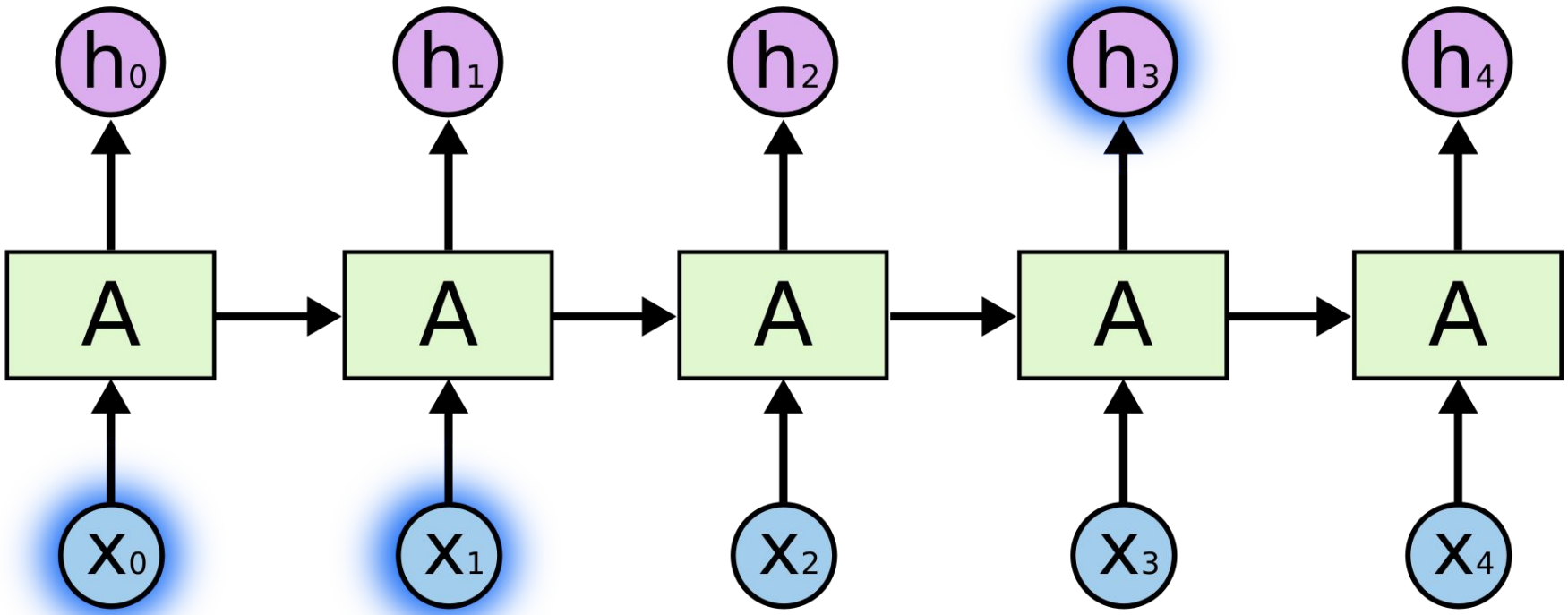
Vanishing/exploding gradient problem

$$L_t = \text{Loss}(y_t, \hat{y}_t)$$

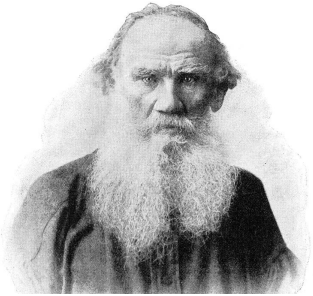
$$L = \sum_{t=1}^{T} L_t$$

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

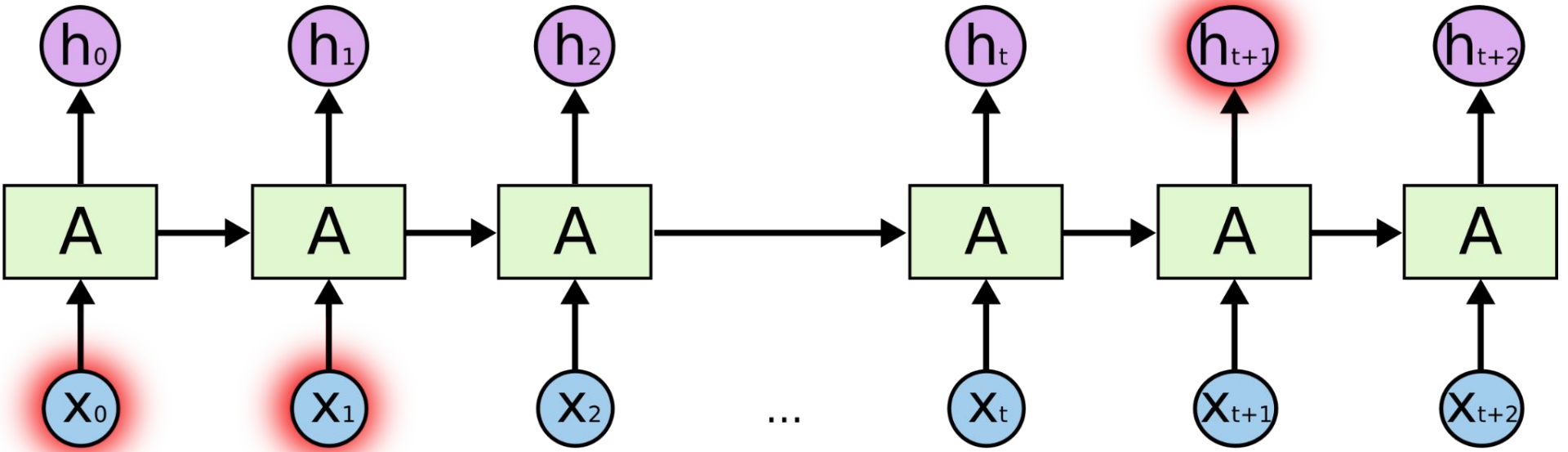
RNN “Звезды ярко сияли на темно-синем …”



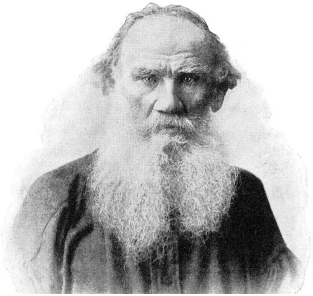
RNN



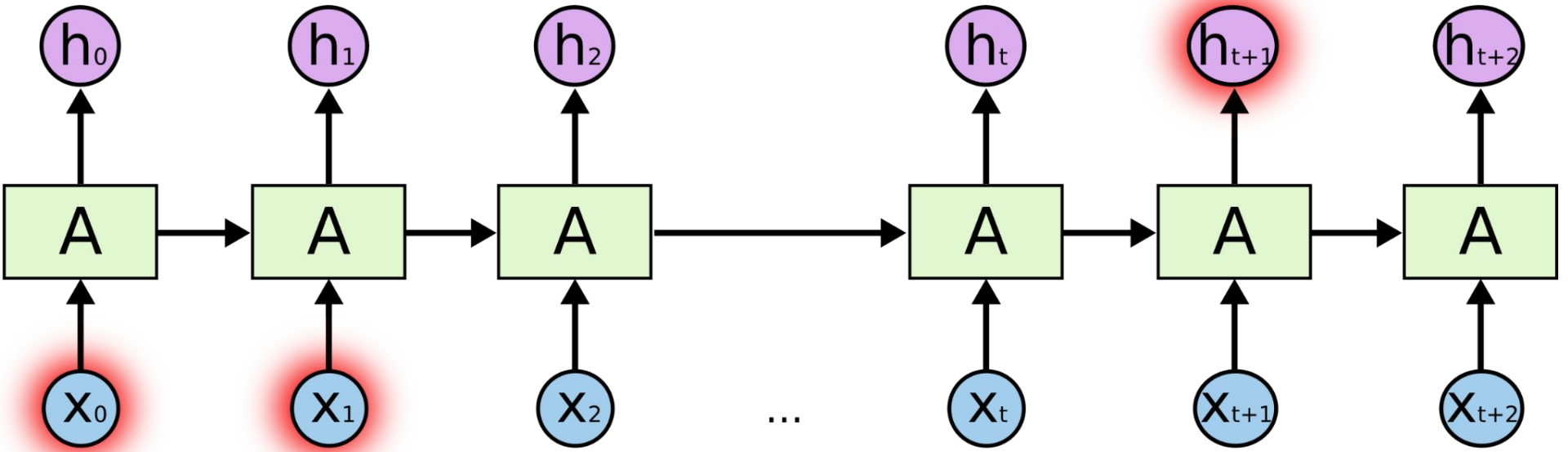
Французы, как доносил лазутчик, перейдя мост в Вене, усиленным маршем шли на Цнайм, лежавший по пути отступления Кутузова, впереди его более чем на сто верст. Достигнуть Цнайма прежде французов — значило получить большую надежду на спасение армии; дать французам предупредить себя в Цнайме — значило наверное подвергнуть всю армию позору, подобному ульмскому, или общей гибели. Но предупредить французов со всею армией было невозможно. Дорога французов от …



RNN

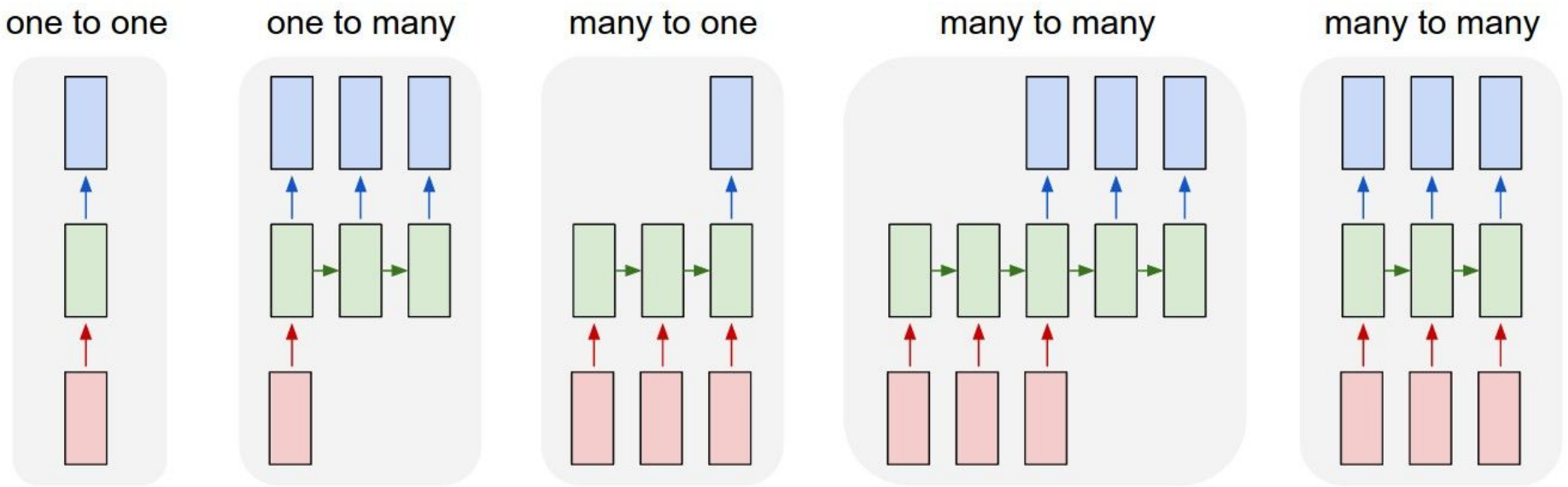


Французы, как доносил лазутчик, перейдя мост в **Вене**, усиленным маршем шли на Цнайм, лежавший по пути отступления Кутузова, впереди его более чем на сто верст. Достигнуть Цнайма прежде французов — значило получить большую надежду на спасение армии; дать французам предупредить себя в Цнайме — значило наверное подвергнуть всю армию позору, подобному ульмскому, или общей гибели. Но предупредить французов со всею армией было невозможно. Дорога французов от **Вены**

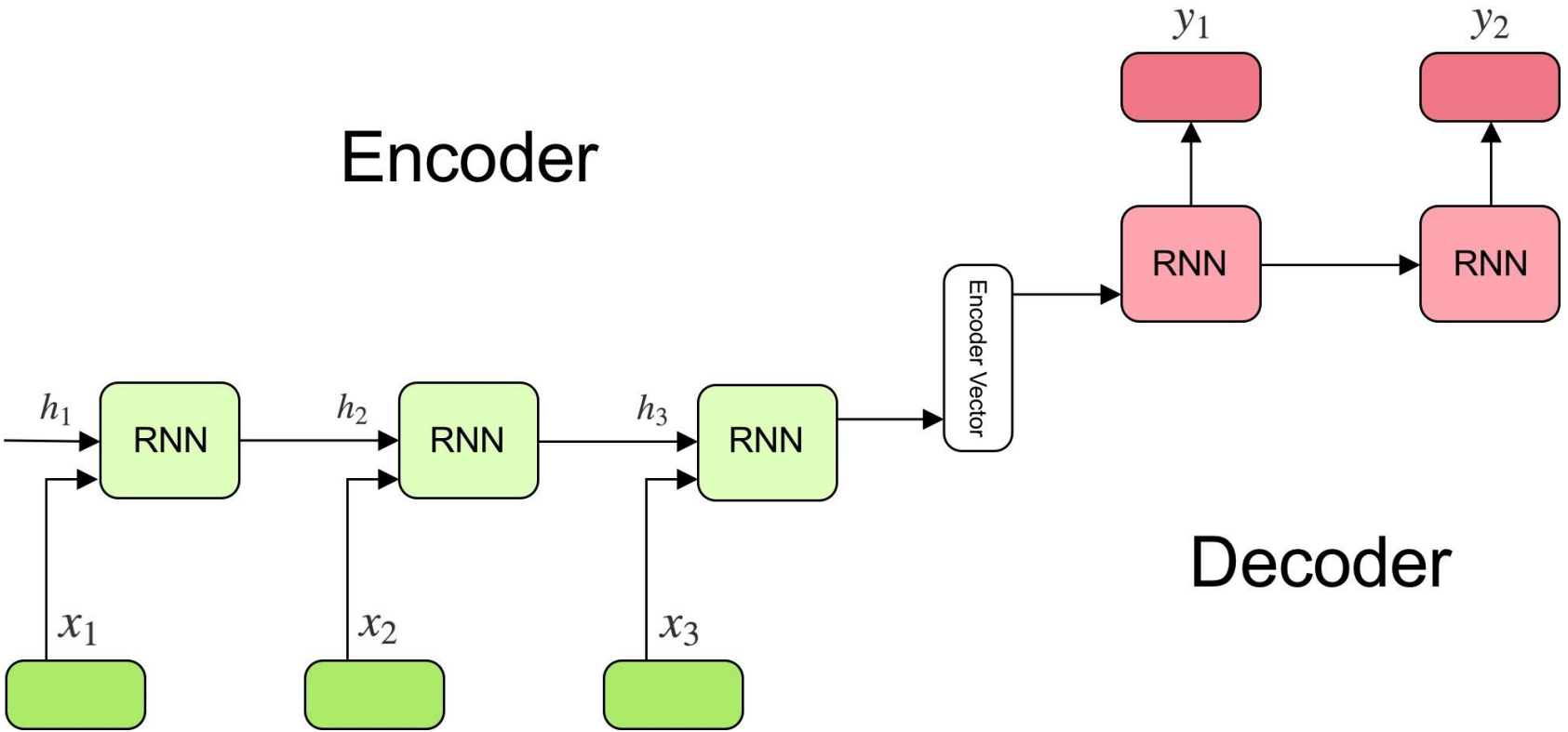


RNN

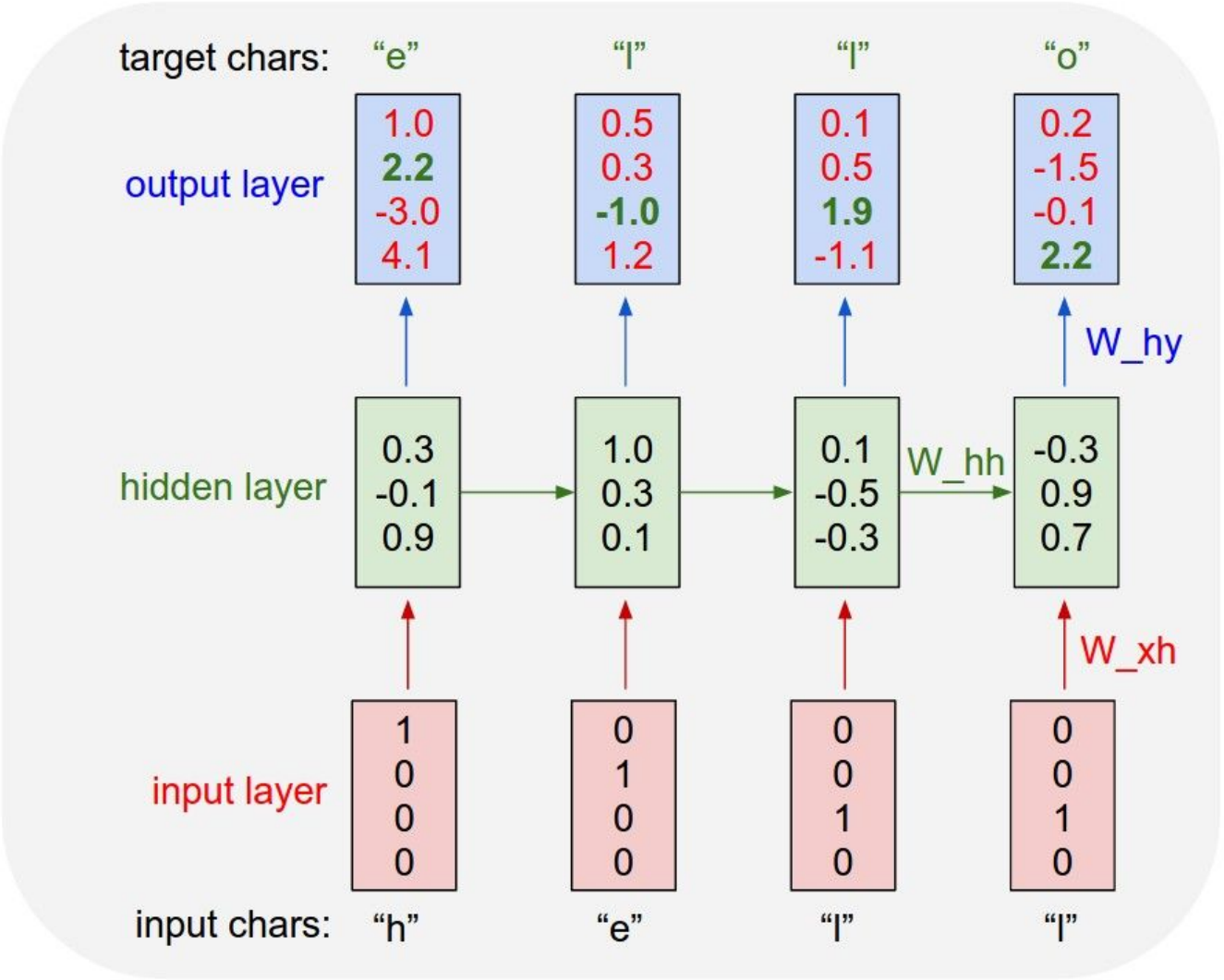
Each rectangle is a vector and arrows represent functions (e.g. matrix multiply). Input vectors are in red, output vectors are in blue and green vectors hold the RNN's state (more on this soon).



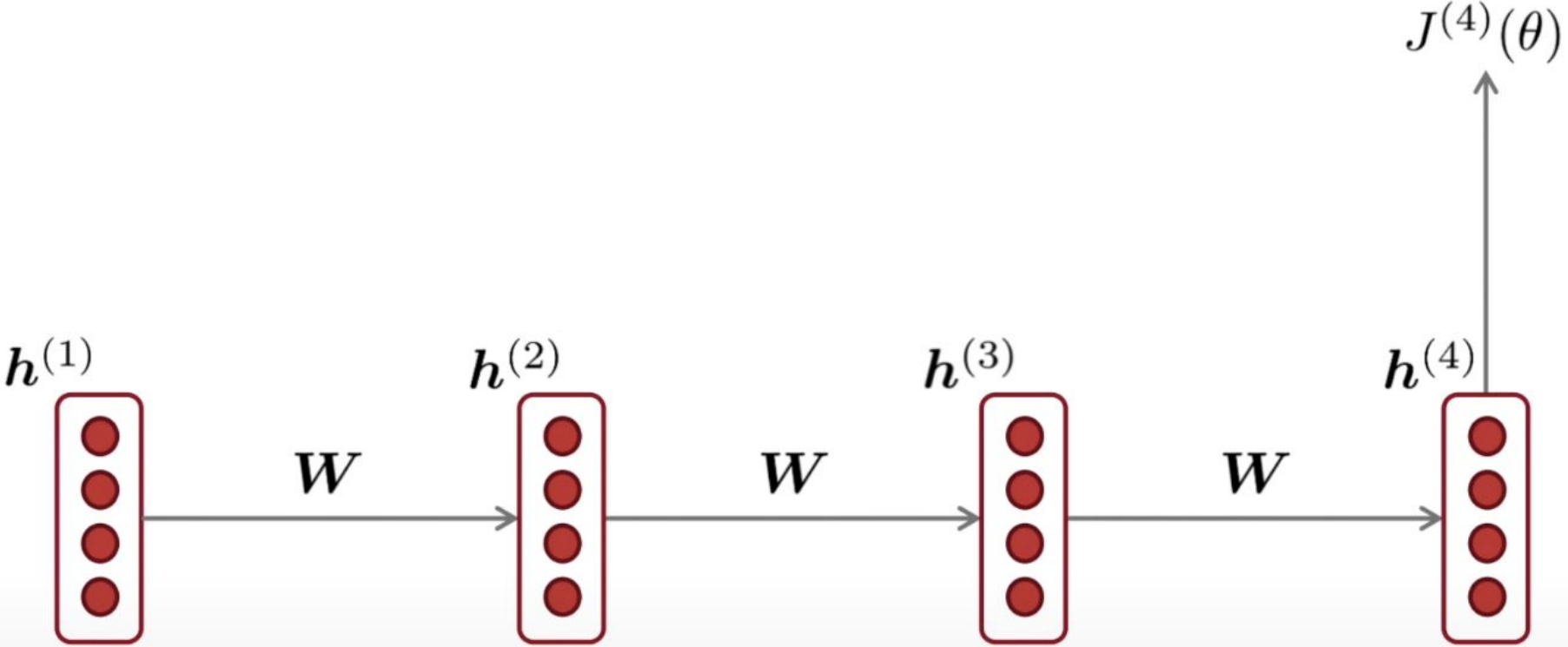
Encoder-decoder



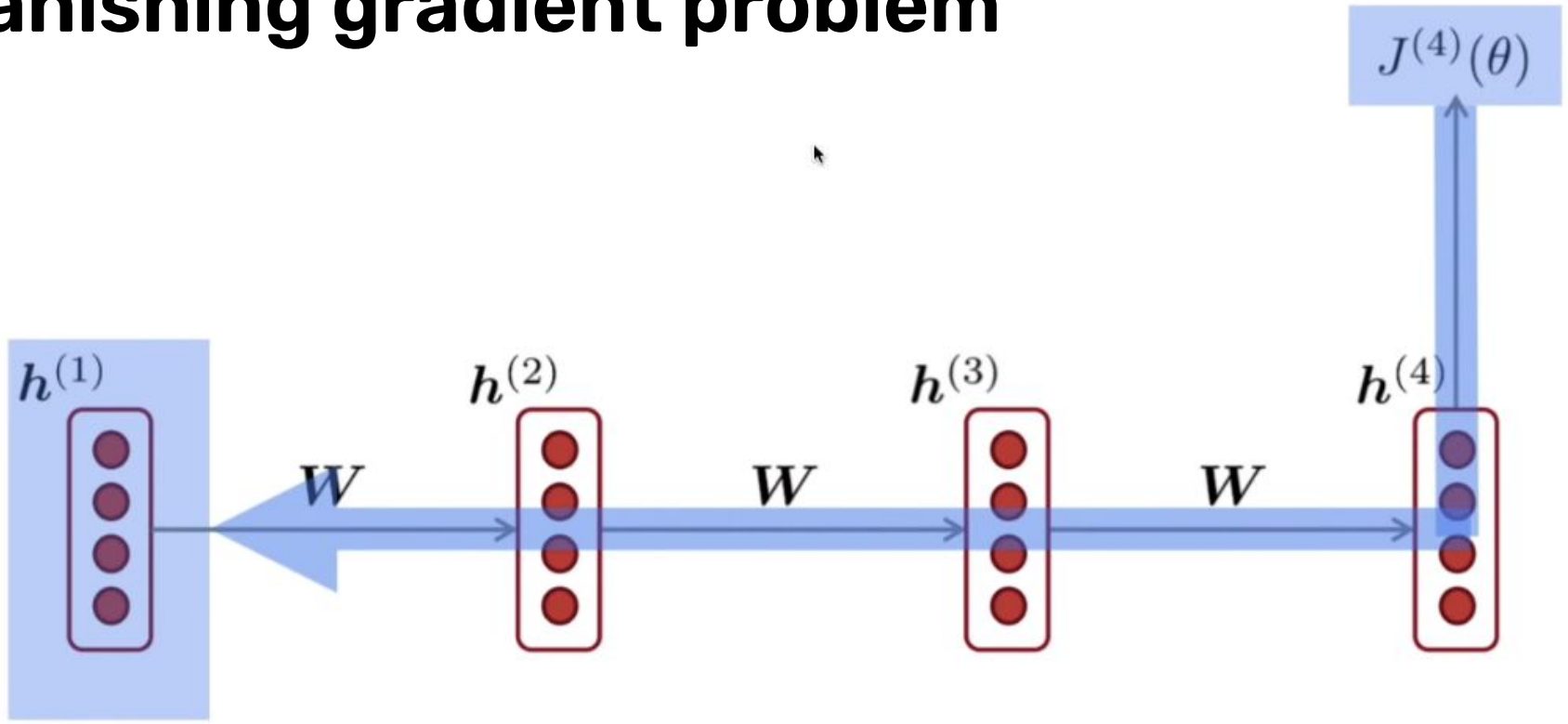
RNN


target chars:
"e"
"l"
"l"
"o"
output layer
1.0
2.2
-3.0
4.1
0.5
0.3
-1.0
1.2
0.1
0.5
1.9
-1.1
0.2
-1.5
-0.1
2.2
W_hy
hidden layer
0.3
-0.1
0.9
1.0
0.3
0.1
0.1
-0.5
-0.3
W_hh
-0.3
0.9
0.7
W_xh
input layer
1
0
0
0
0
1
0
0
0
0
1
0
0
0
1
0
input chars:
"h"
"e"
"l"
"l"

Vanishing gradient problem

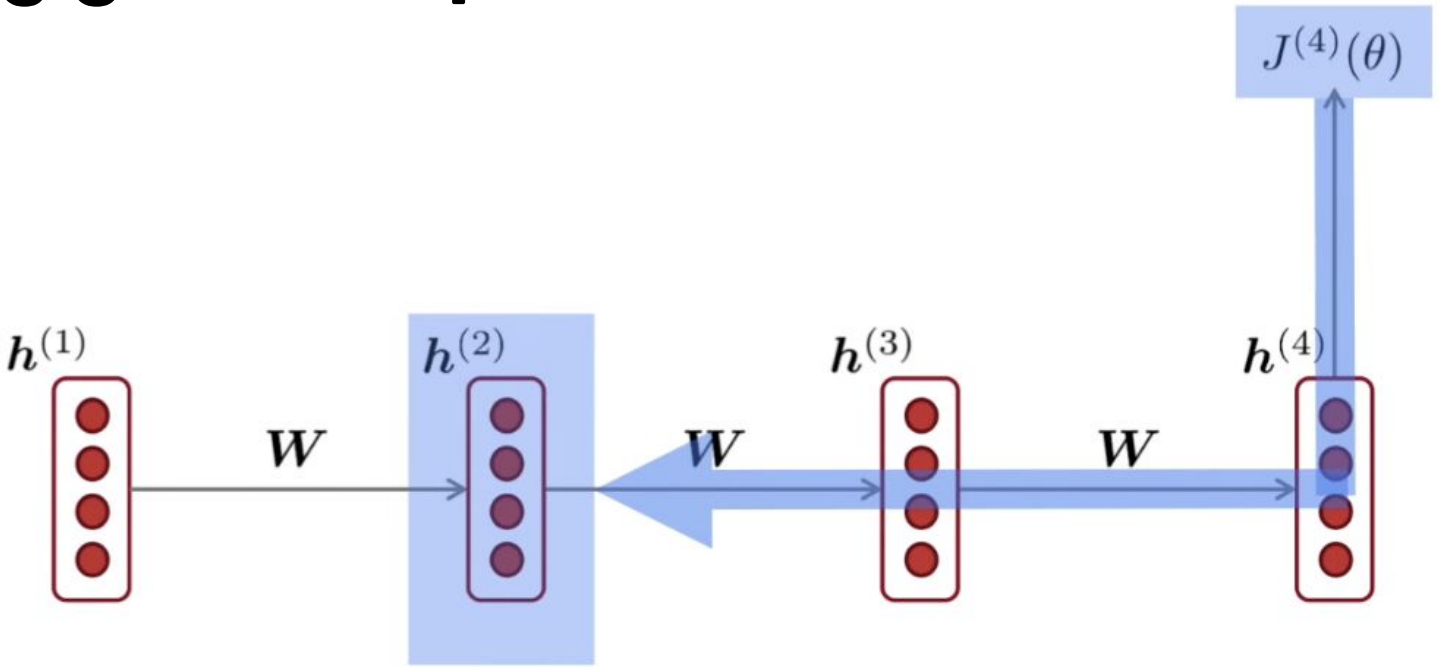


Vanishing gradient problem



$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = ?$$

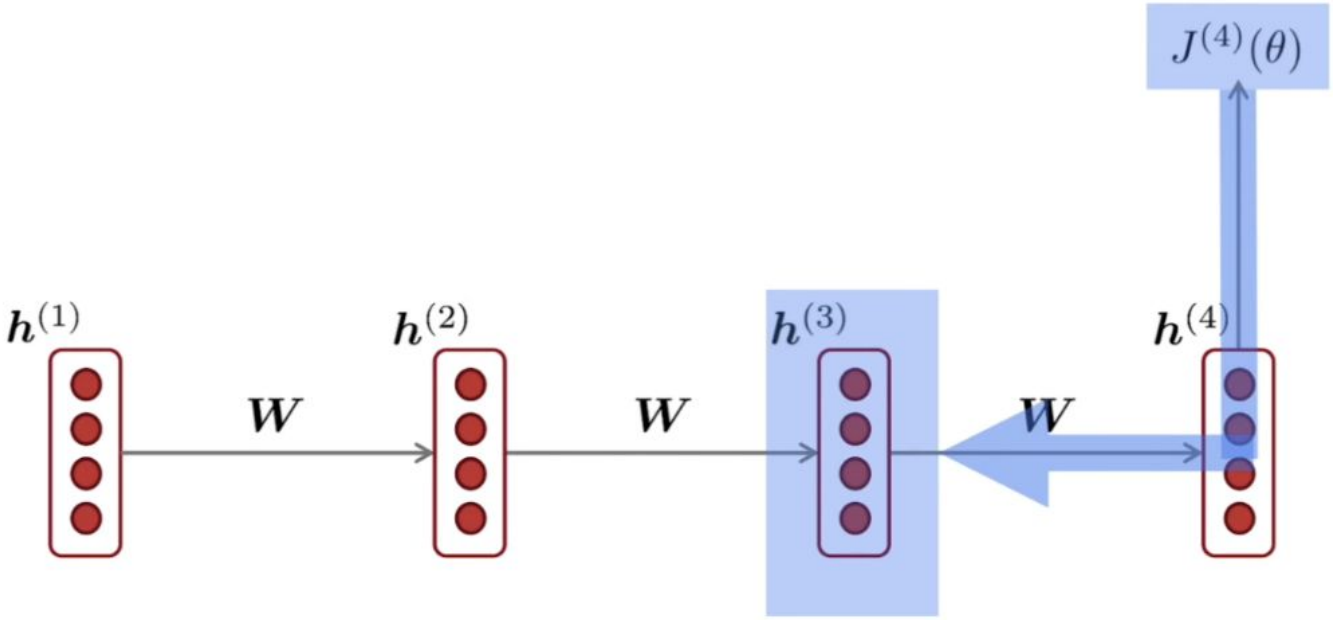
Vanishing gradient problem



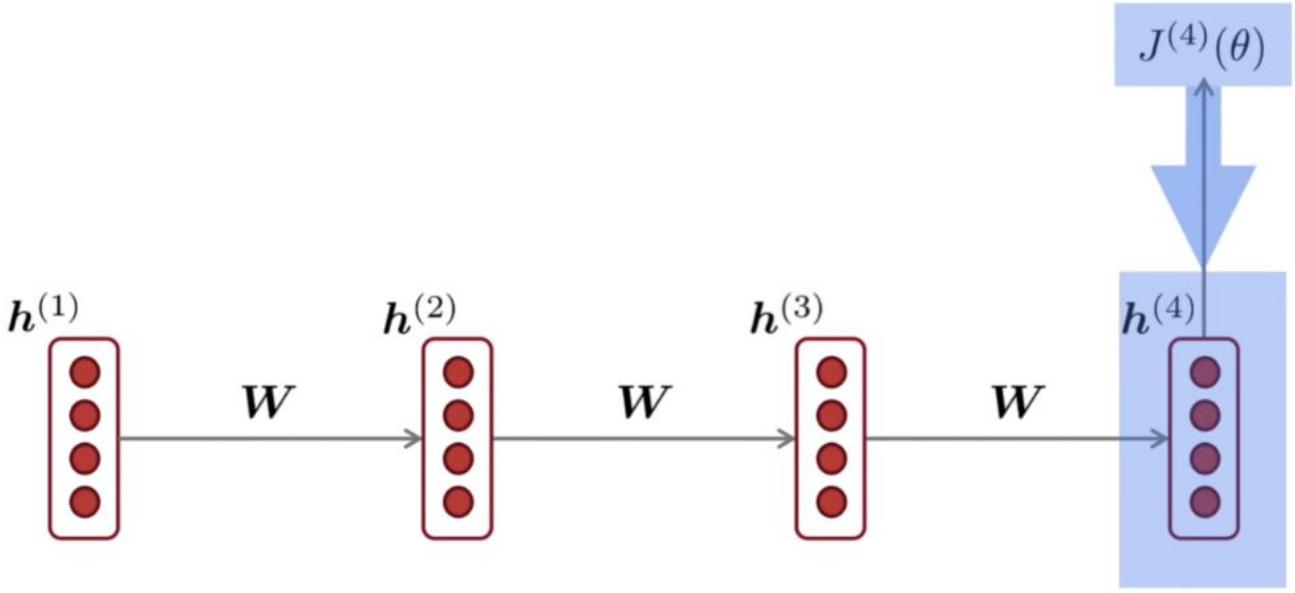
$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \frac{\partial h^{(2)}}{\partial h^{(1)}} \times \frac{\partial J^{(4)}}{\partial h^{(2)}}$$

chain rule!

Vanishing gradient problem



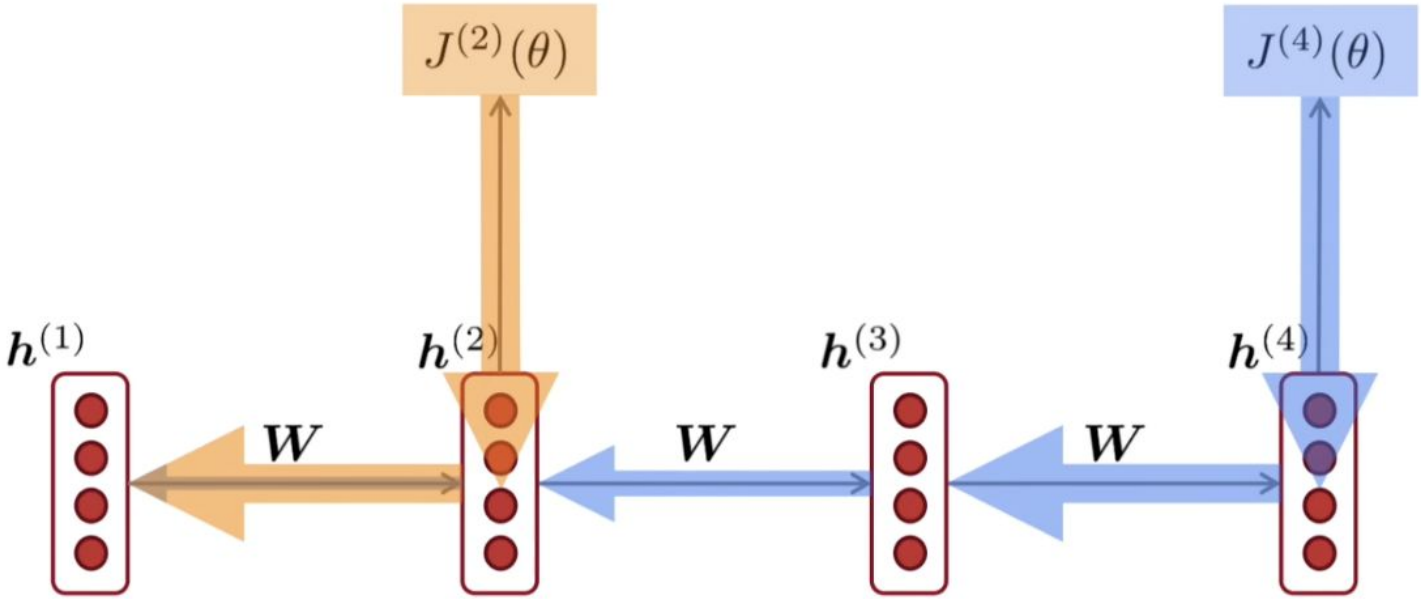
Vanishing gradient problem



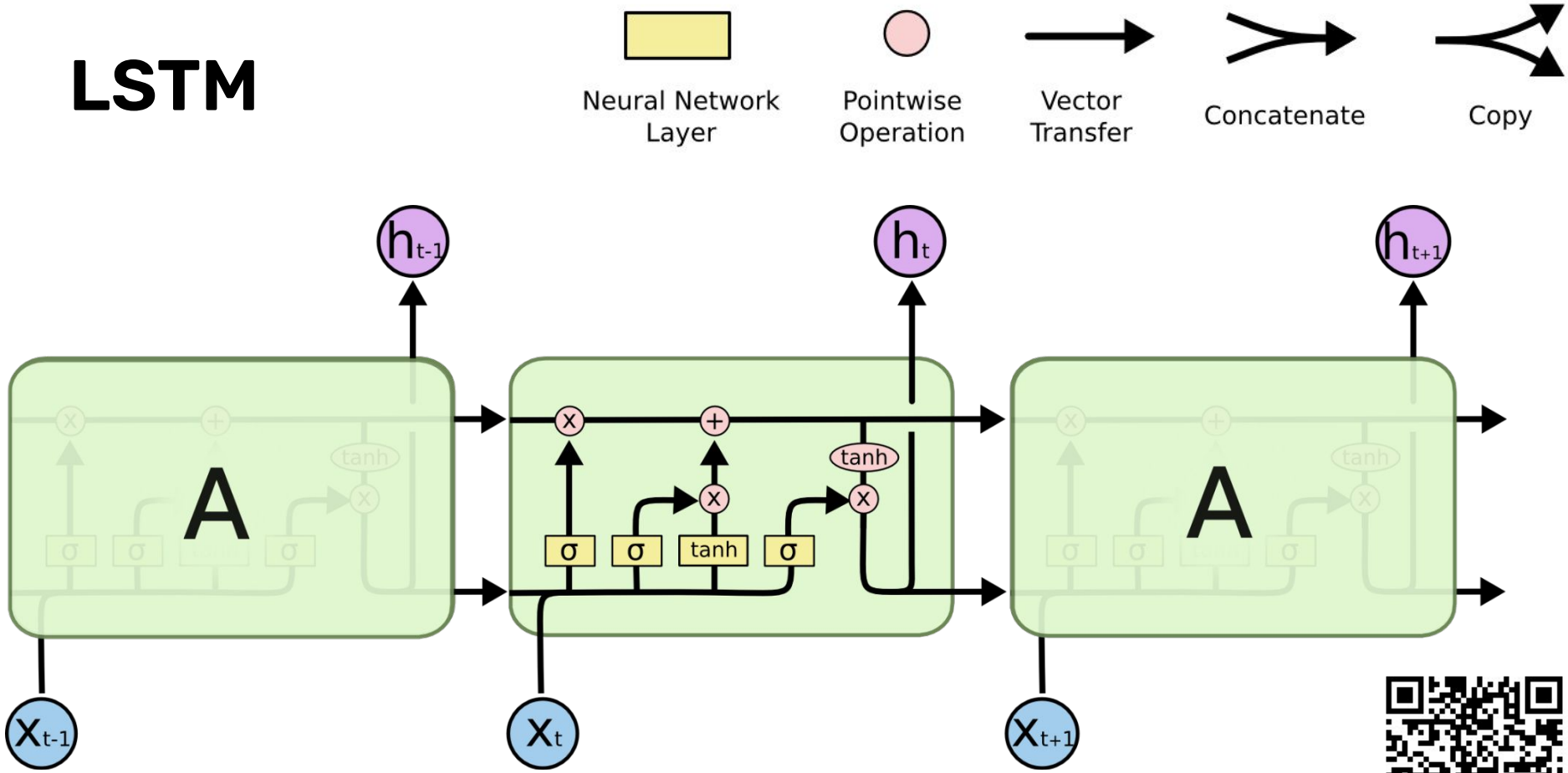
$$\frac{\partial J^{(4)}}{\partial h^{(1)}} = \frac{\partial h^{(2)}}{\partial h^{(1)}} \times \frac{\partial h^{(3)}}{\partial h^{(2)}} \times \frac{\partial h^{(4)}}{\partial h^{(3)}} \times \frac{\partial J^{(4)}}{\partial h^{(4)}}$$

chain rule!

Vanishing gradient problem

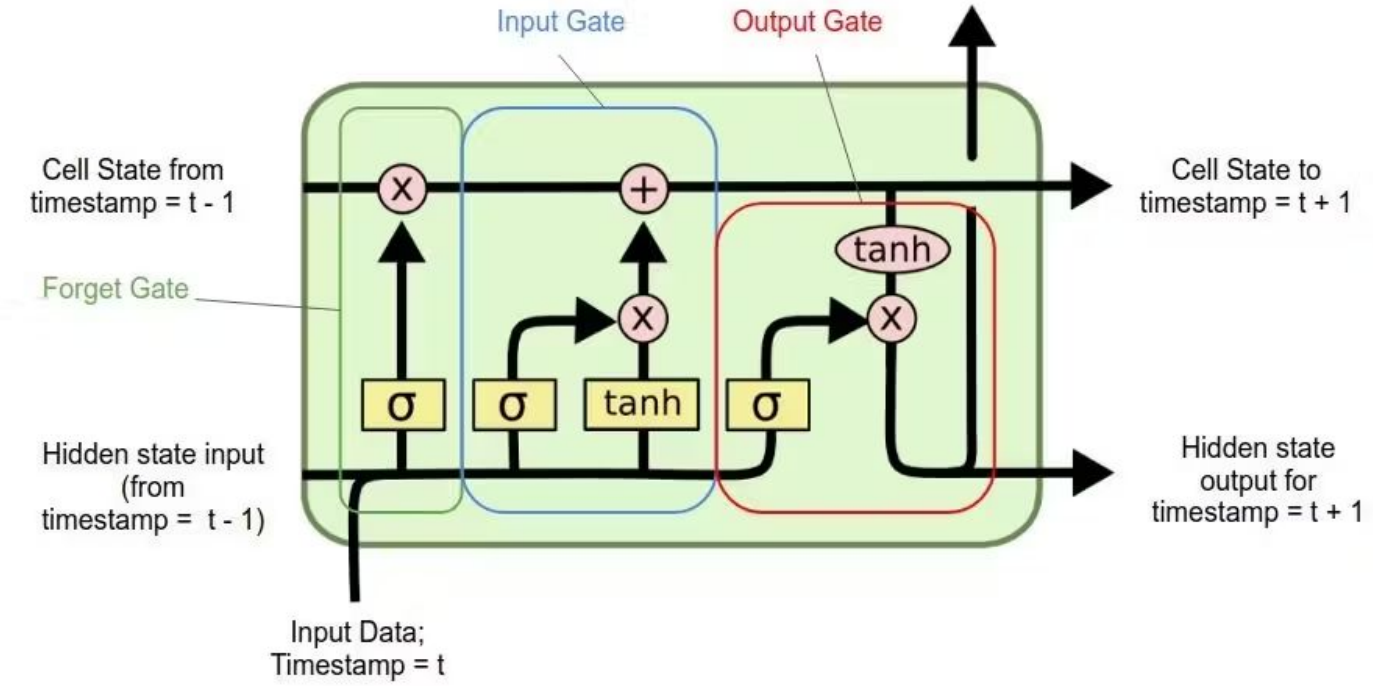


LSTM



https://www.bioinf.jku.at/publications/older/2604.pdf

LSTM


Input Gate
Output Gate
Cell State from timestamp = t - 1
Cell State to timestamp = t + 1
Forget Gate
tanh
σ
σ
tanh
σ
Hidden state input (from timestamp = t - 1)
Hidden state output for timestamp = t + 1
Input Data; Timestamp = t

LSTM: very short explanation

Рецензия на фильм «Хоббит: Нежданное путешествие»

“Очень классный фильм с необычной сюжетной линией, определенно понравится любителям жанра приключений. Несмотря на неоднозначные моменты, я бы мог рекомендовать его. Здорово подойдет для приятного вечера! ”

LSTM: very short explanation

Рецензия на фильм «Хоббит: Нежданное путешествие»

“Очень классный фильм с необычной сюжетной линией, определенно понравится любителям жанра приключений. Несмотря на неоднозначные моменты, я бы мог рекомендовать его. Здорово подойдет для приятного вечера! ”

“Я храню то, что нужно, а все остальное забываю.” **Ячейка©**

LSTM: very short explanation

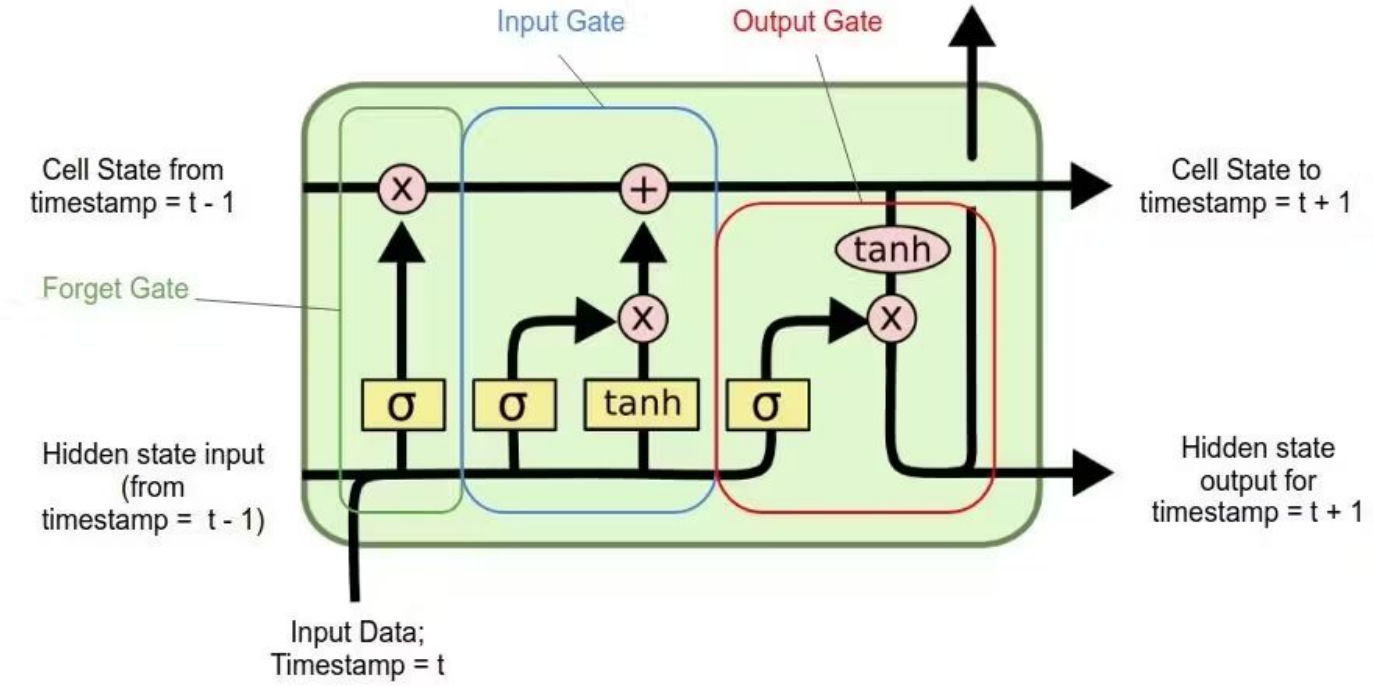
Рецензия на фильм «Хоббит: Нежданное путешествие»

“Очень классный фильм с необычной сюжетной линией, определенно понравится любителям жанра приключений. Несмотря на неоднозначные моменты, я бы мог рекомендовать его. Здорово подойдет для приятного вечера! ”

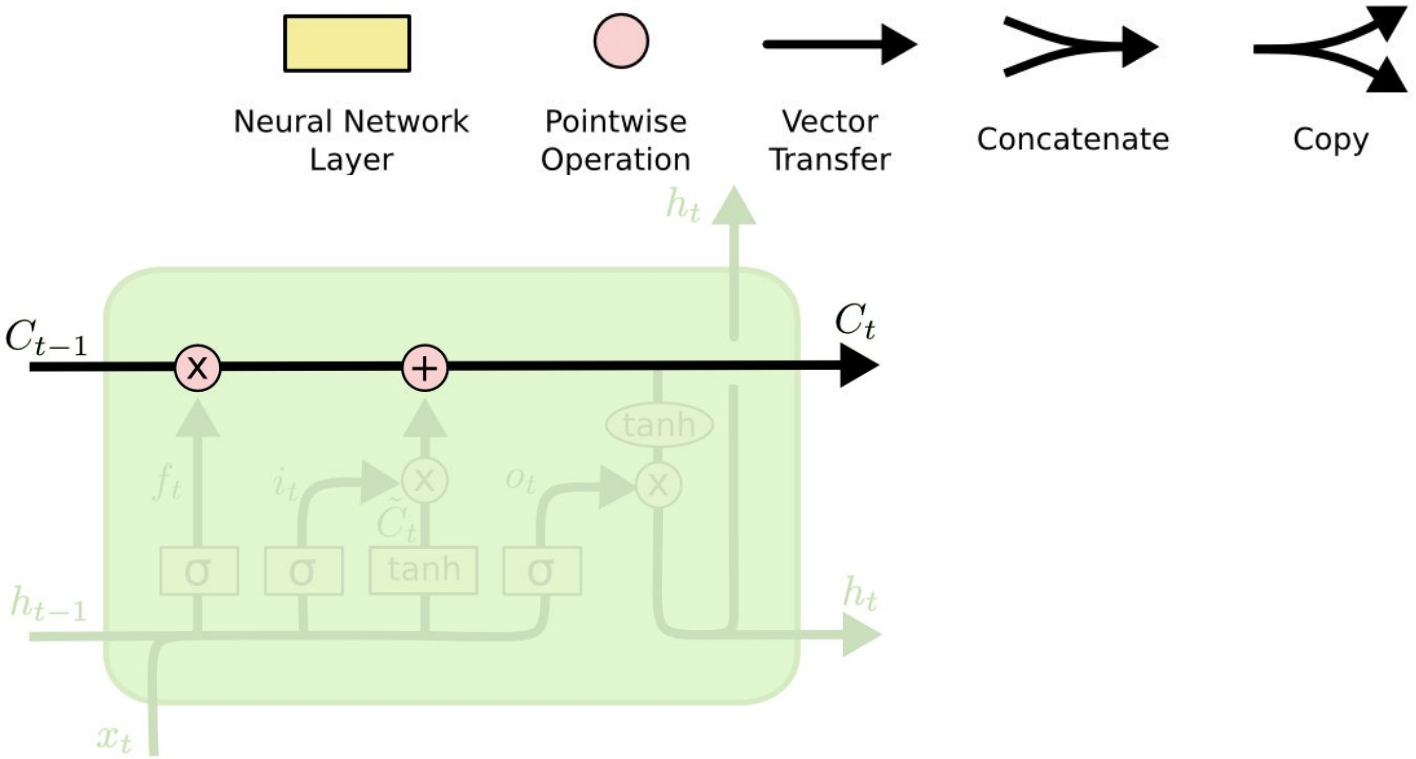
“Суть я уловил, но с чего все началось уже не помню”

~~**Студенты**~~ **Скрытое состояние©**

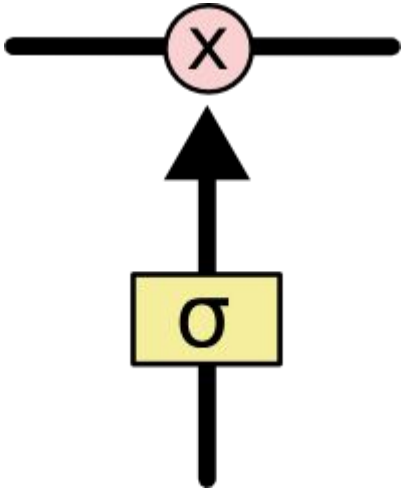
LSTM

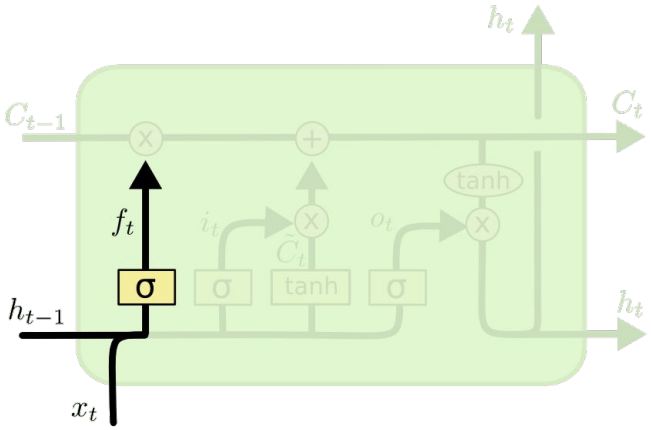

Input Gate
Output Gate
Cell State from timestamp = t - 1
Cell State to timestamp = t + 1
Forget Gate
tanh
σ
σ
tanh
σ
Hidden state input (from timestamp = t - 1)
Hidden state output for timestamp = t + 1
Input Data; Timestamp = t

LSTM


Neural Network Layer
Pointwise Operation
Vector Transfer
Concatenate
Copy
h_t
C_{t-1}
C_t
tanh
f_t
i_t
o_t
$\tilde{C}_t$
σ
σ
tanh
σ
h_{t-1}
h_t
x_t

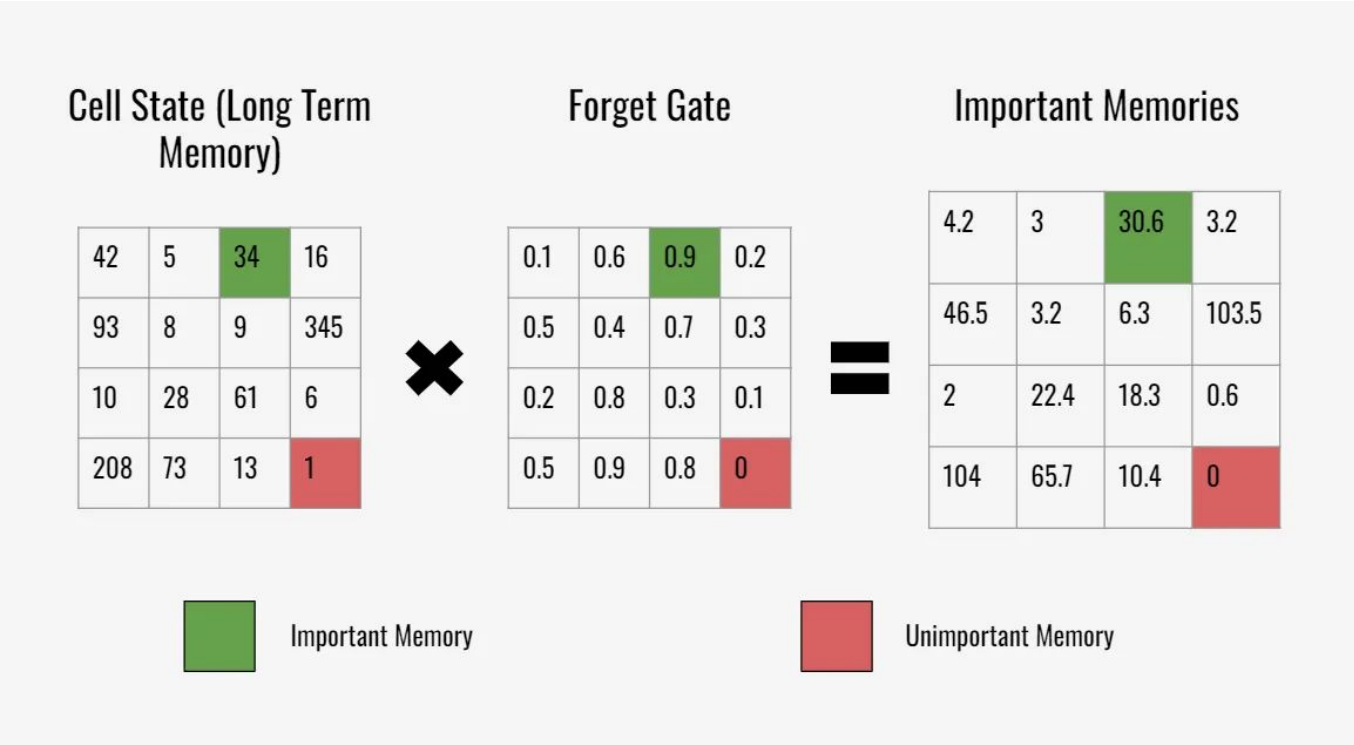
LSTM





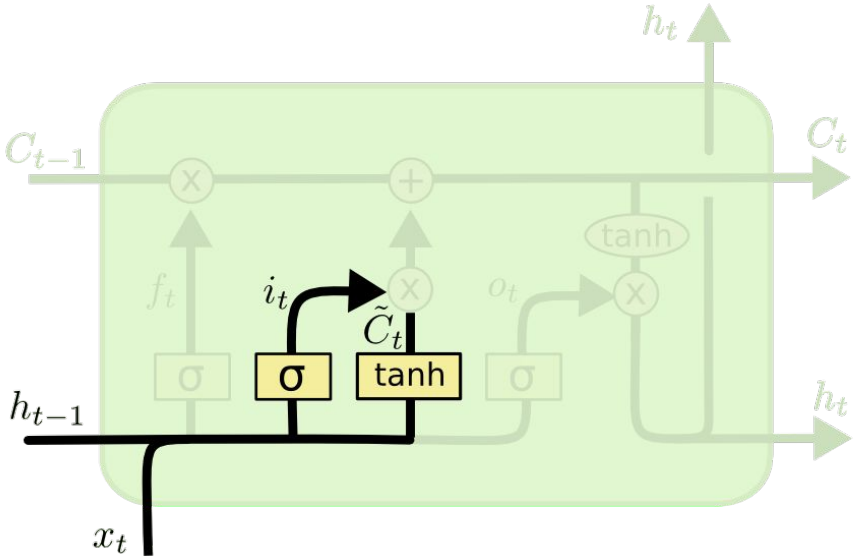
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

LSTM



https://medium.com/@CallMeTwitch/building-a-neural-network-zoo-from-scratch-the-long-short-term-memory-network-1cec5cf31b7

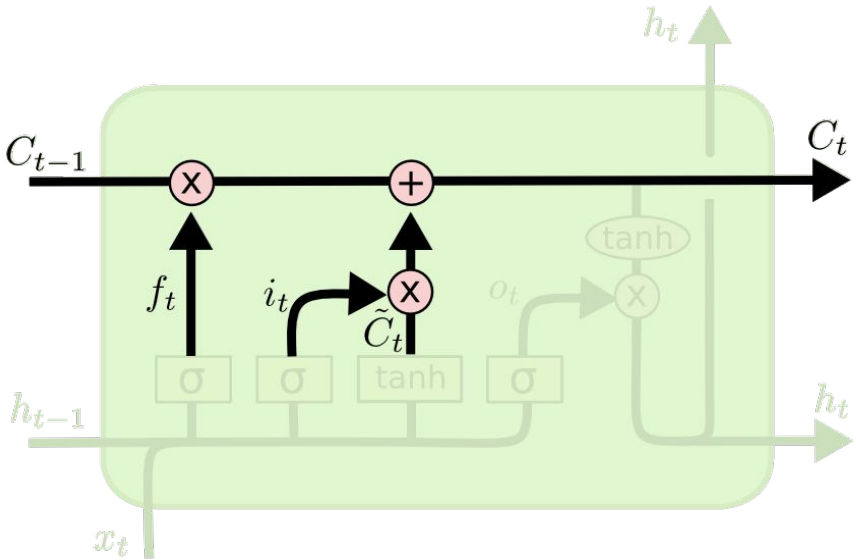
LSTM



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

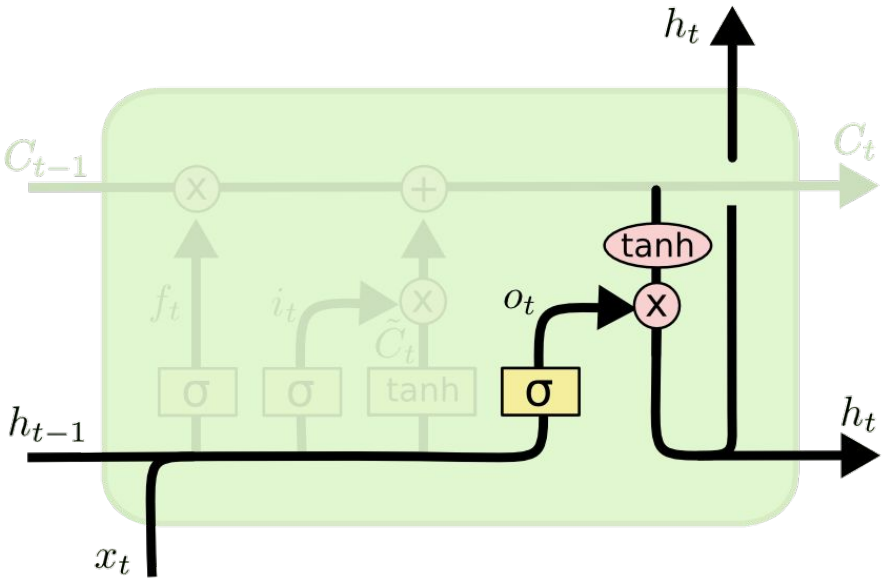
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

LSTM



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

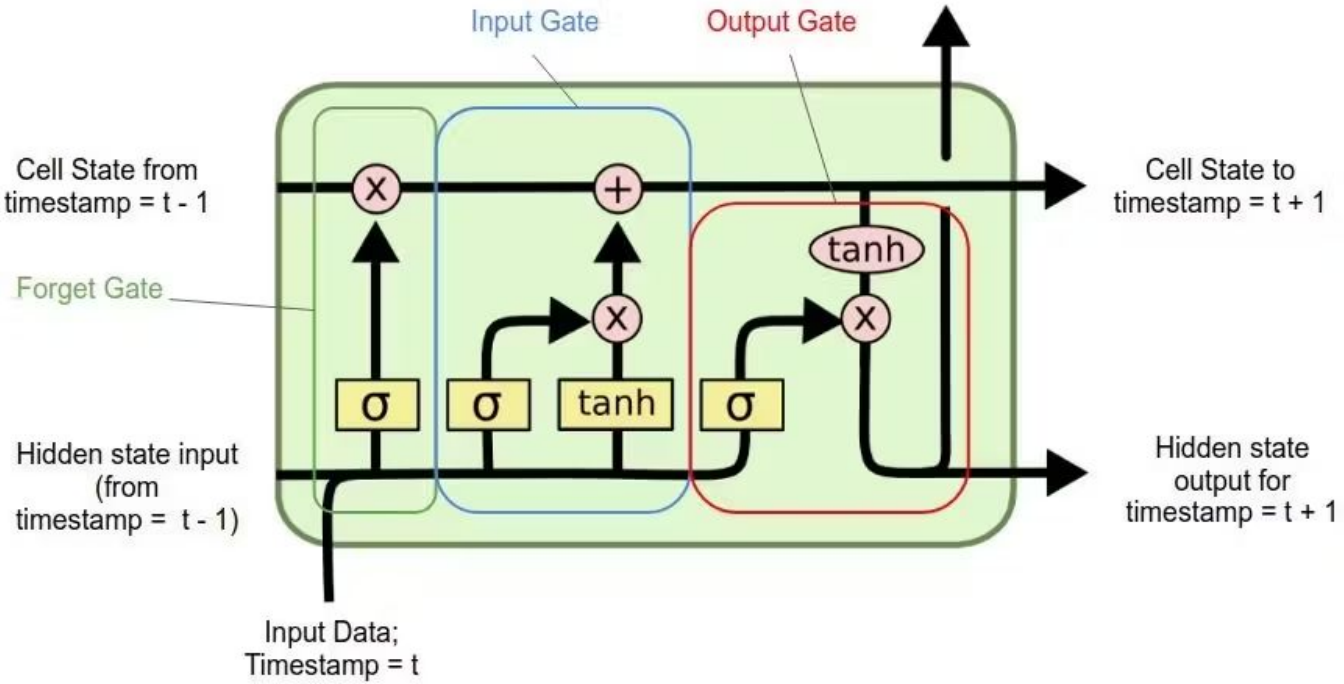
LSTM



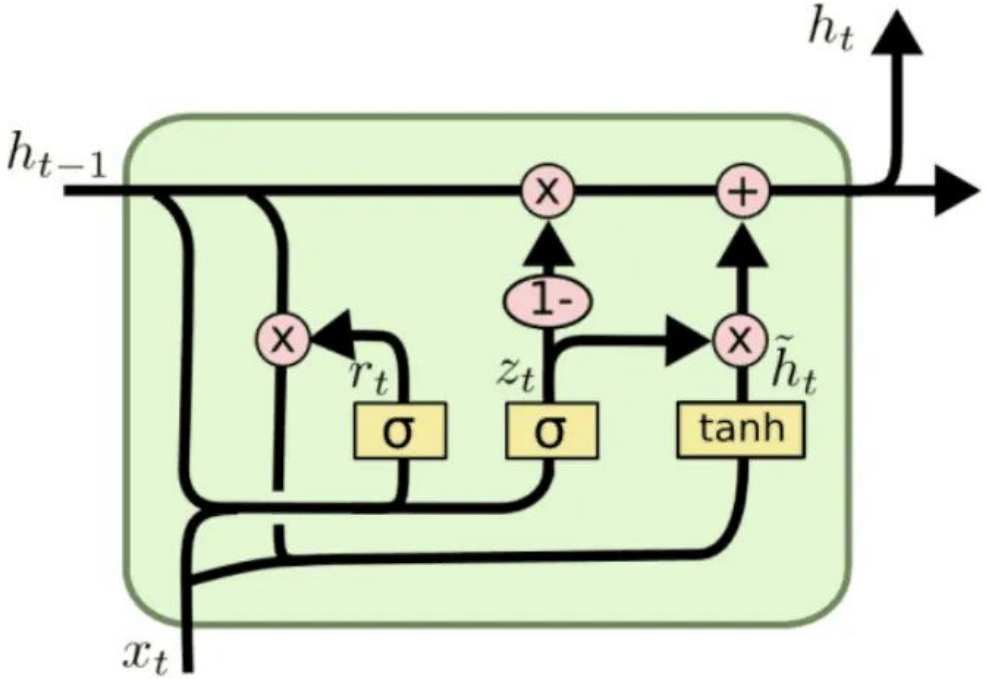
$$o_t = \sigma\left(W_o\left[h_{t-1}, x_t\right] + b_o\right)$$

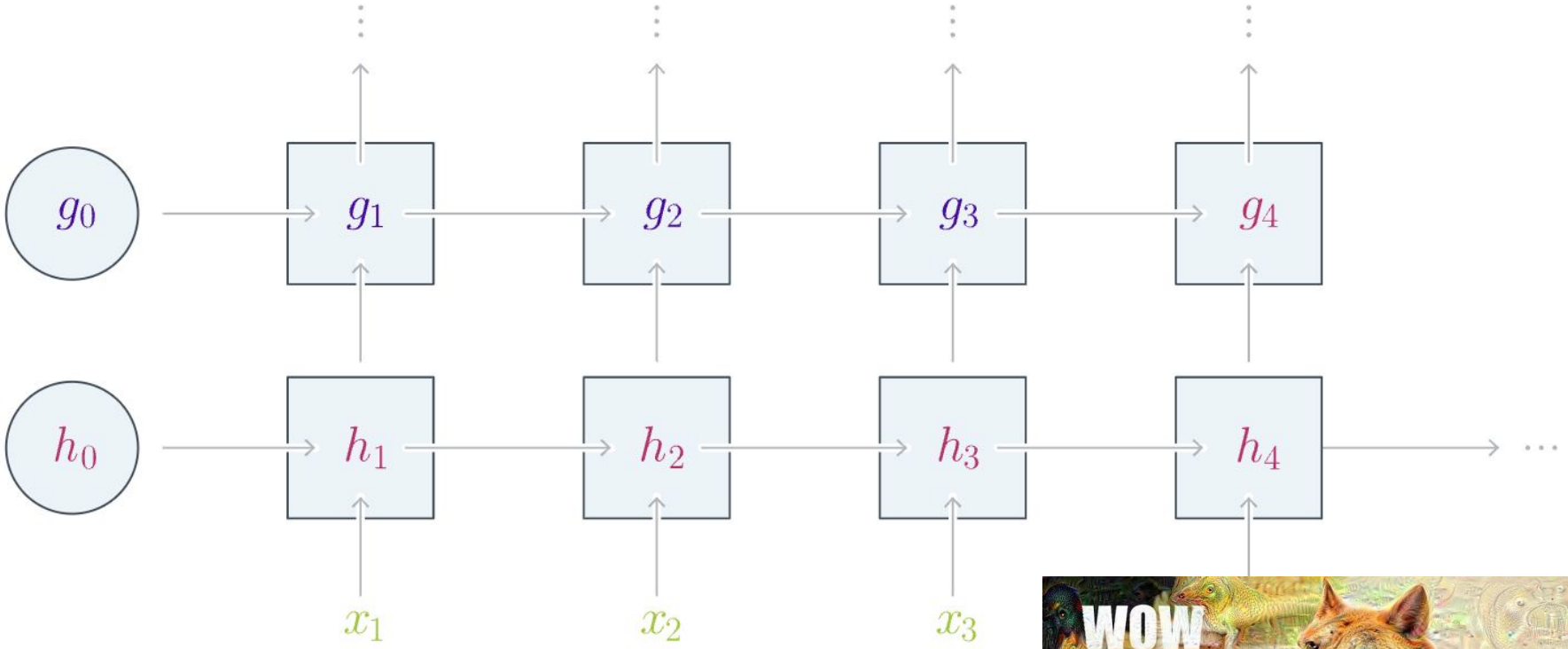
$$h_t = o_t * \tanh\left(C_t\right)$$

LSTM

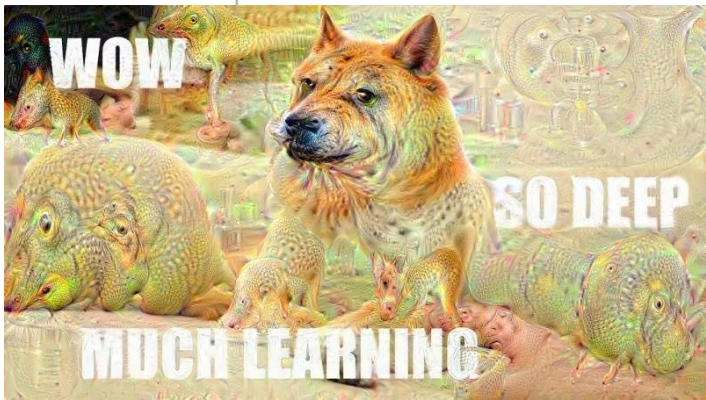

Input Gate
Output Gate
Cell State from
timestamp = t - 1
Cell State to
timestamp = t + 1
Forget Gate
tanh
σ
σ
tanh
σ
Hidden state input
(from
timestamp = t - 1)
Hidden state
output for
timestamp = t + 1
Input Data;
Timestamp = t

GRU

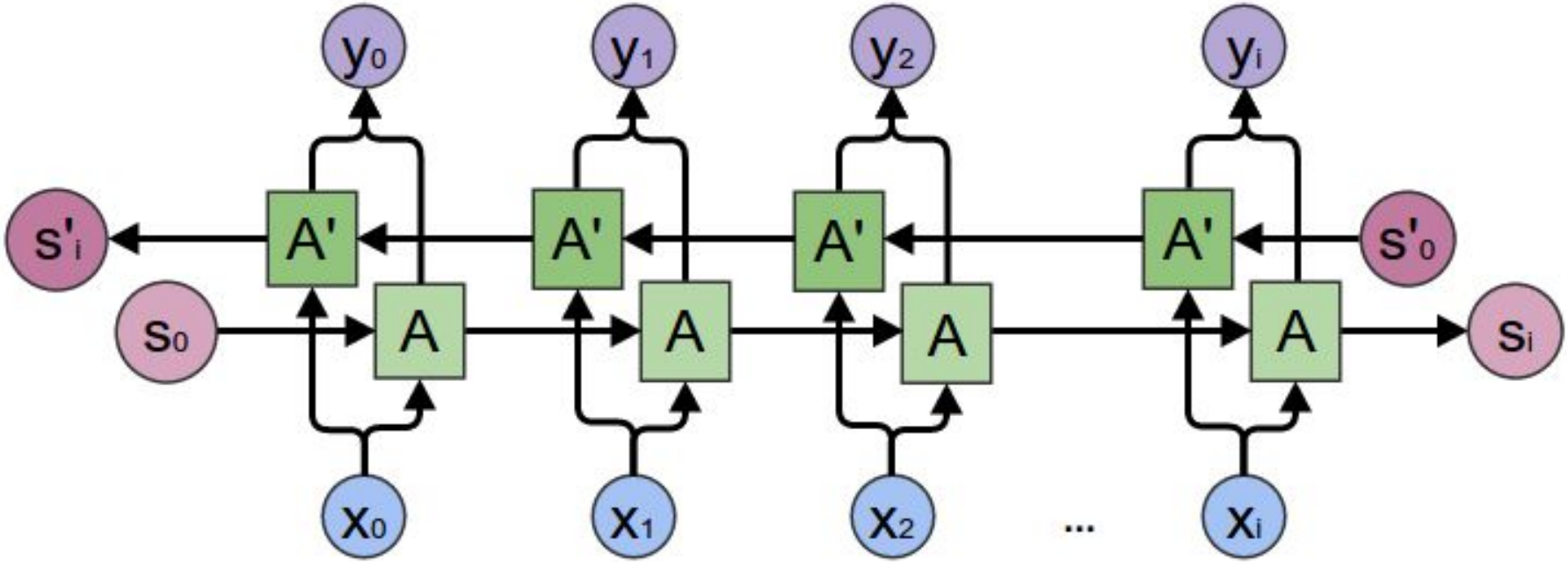




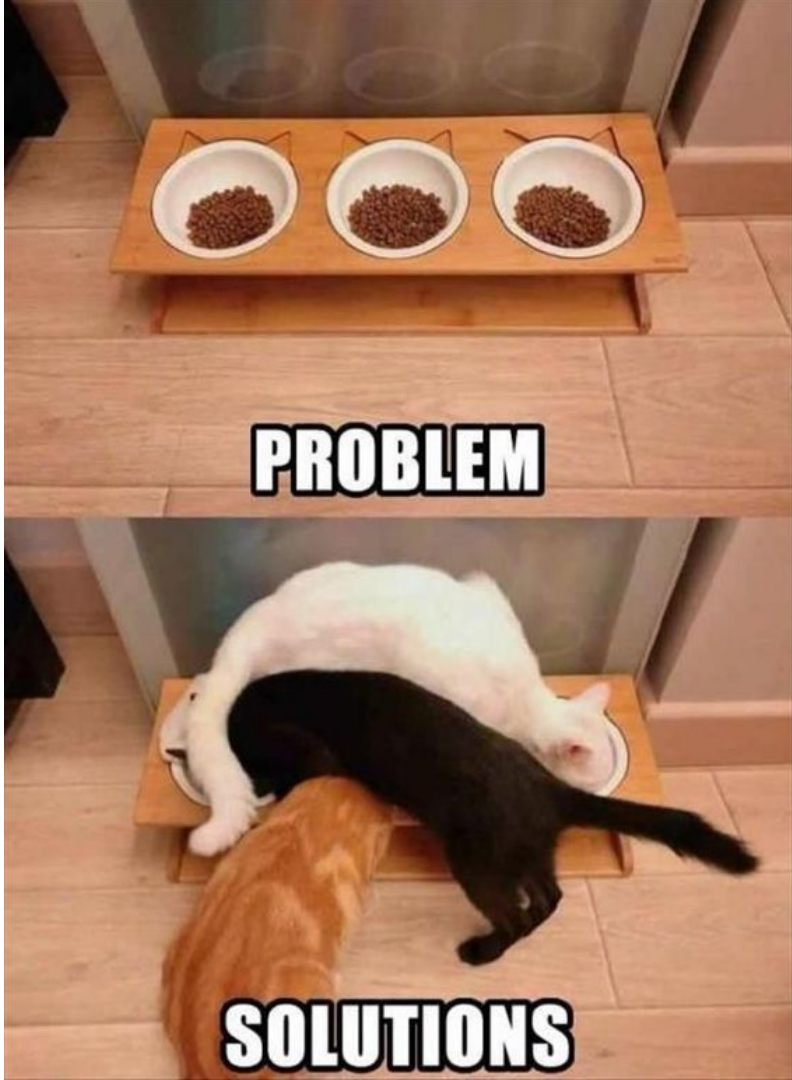
Deeper…



Bidirectional RNNs

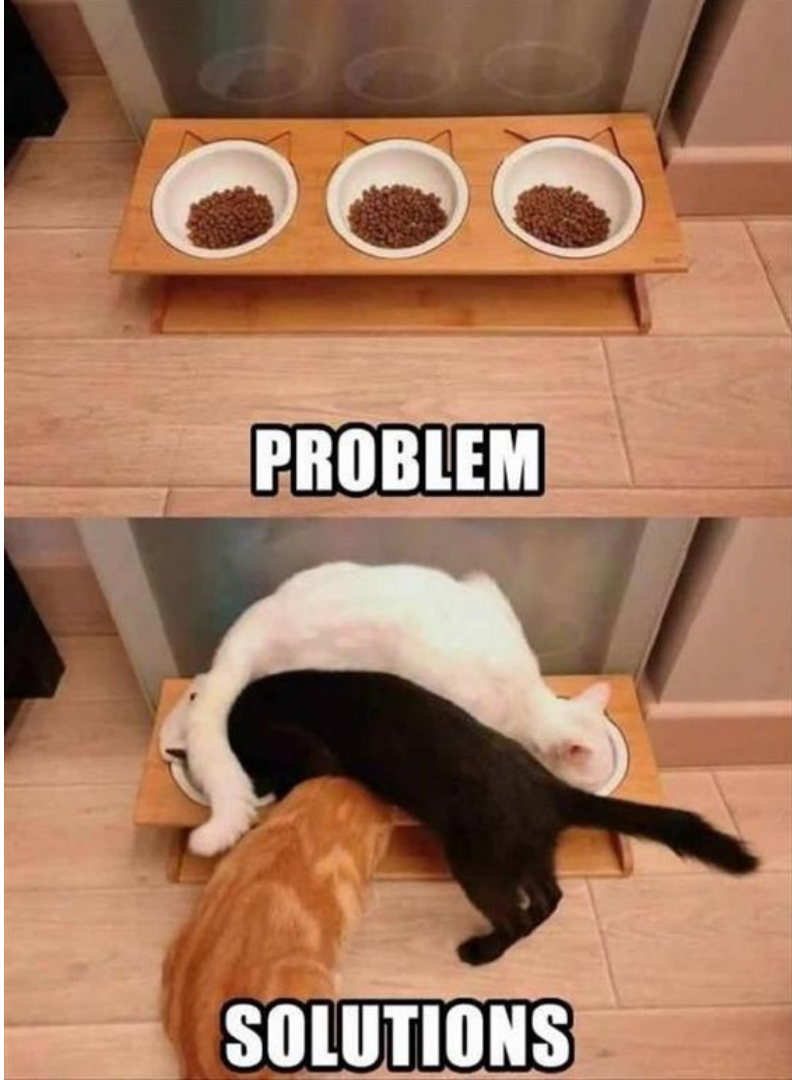


What is the problem with rnn and what solution have people come up with?



What is the problem with rnn and what solution have people come up with?

1. **Sequential processing:** sentences must be processed word by word.
2. **Past information retained through past hidden states:** sequence to sequence models follow the Markov property: each state is assumed to be dependent only on the previously seen state.

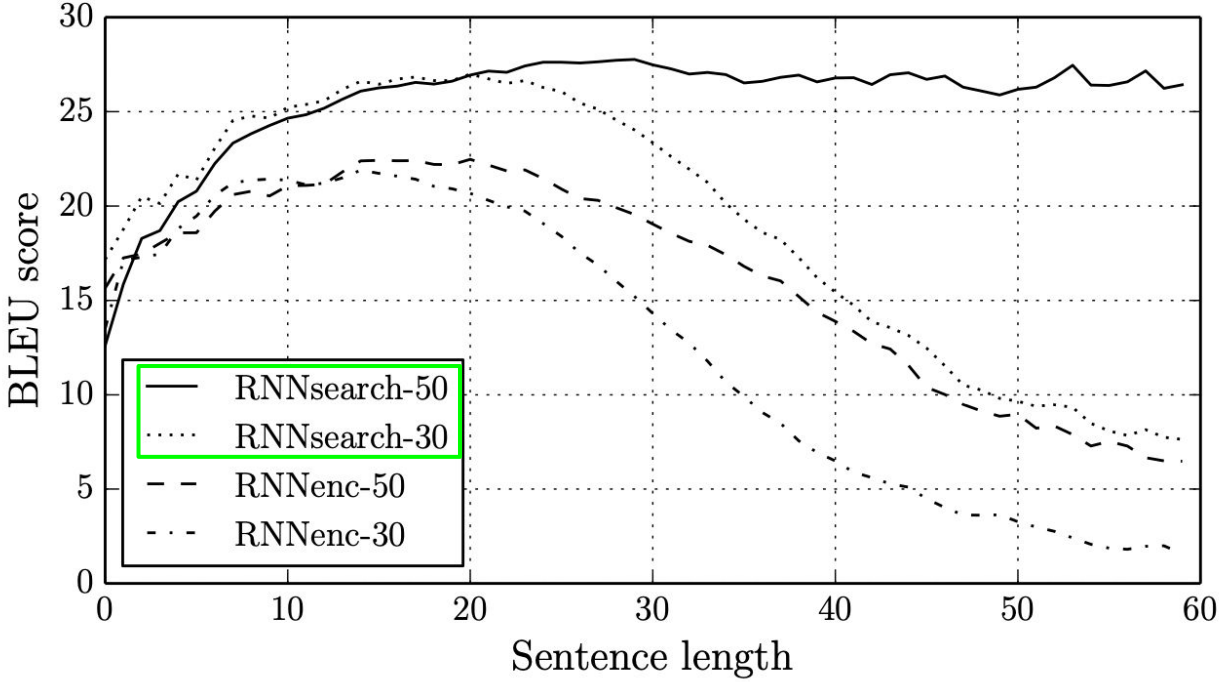


Transformers

- **Non sequential**: sentences are processed as a whole rather than word by word.
- **Self Attention**: this is the newly introduced 'unit' used to compute similarity scores between words
- **Positional embeddings**: another innovation introduced to replace recurrence. The idea is to use fixed or learned weights which encode information related to a specific position of a token in a sentence.

A little more about RNN

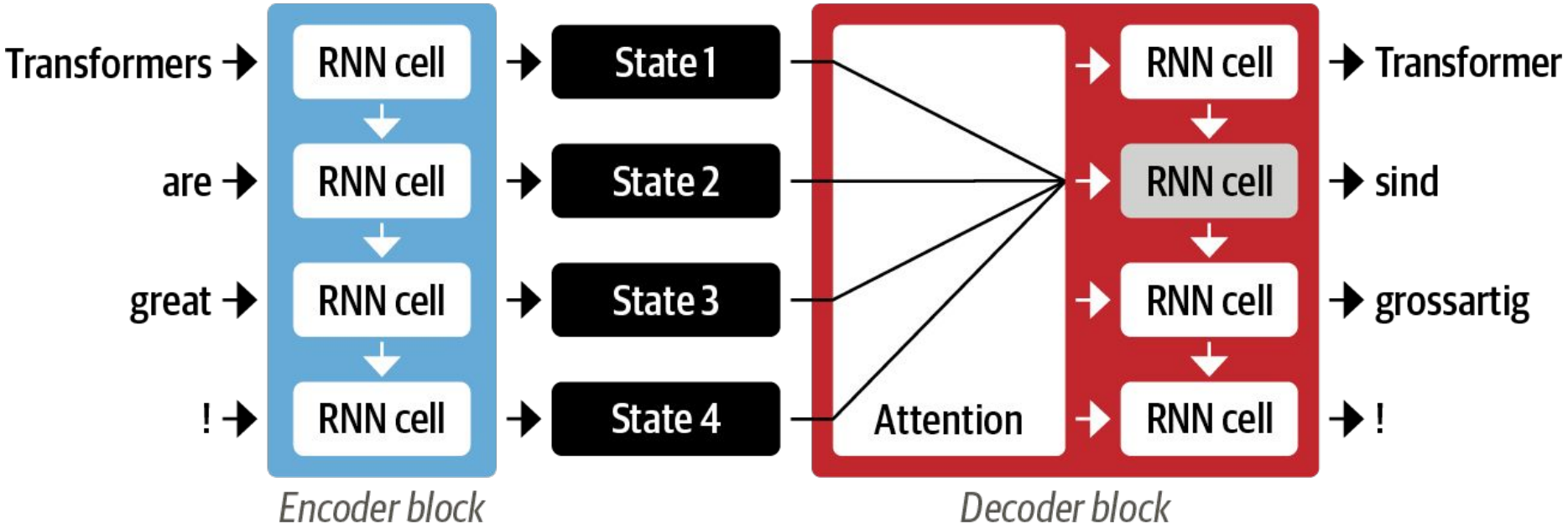
…..allowing a model **to automatically (soft-)search for parts of a source sentence** that are relevant to predicting a target word….



https://arxiv.org/pdf/1409.0473

Attention? Attention!

The main idea behind attention is **that instead of producing a single hidden state for the sequence, the encoder outputs a hidden state at each step that the decoder can access.**



Practical work_2🌚

Implement an RNN to generate poems💭

https://disk.yandex.ru/d/fi1RbCVc7y3wsg