E9: 309 Advanced Deep Learning
14-10-2020

Instructor: Sriram Ganapathy
sriramg@iisc.ac.in

Teaching Assistant : Akshara Soman, Prachi Singh, Jaswanth Reddy
aksharas@iisc.ac.in, prachis@iisc.ac.in, jaswanthk@iisc.ac.in

Schedule - MW - 430-6pm (Microsoft Teams)
http://leap.ee.iisc.ac.in/sriram/teaching/ADL2020/

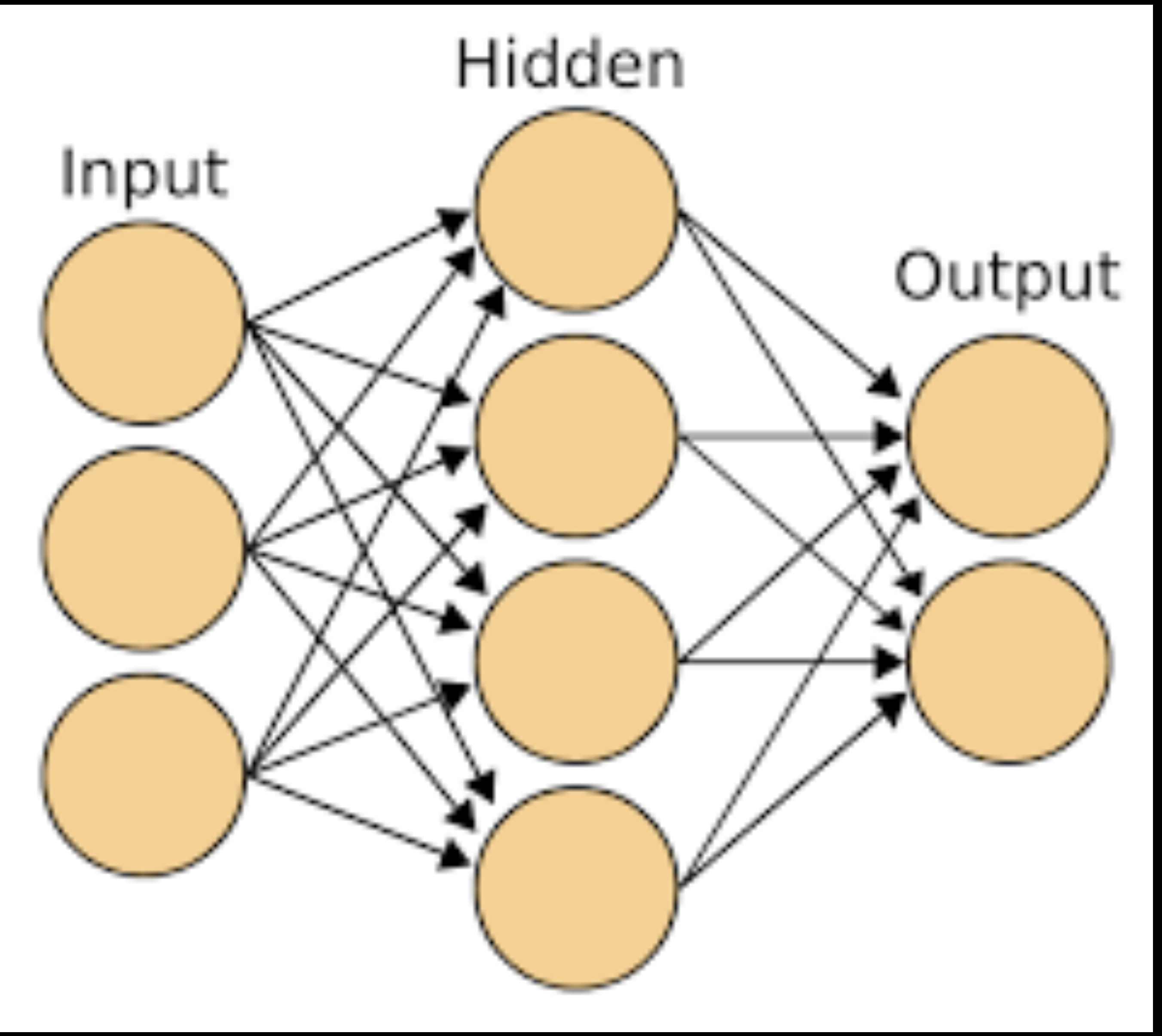
Housekeeping

- Filling the google form in the webpage
  - Contents will be made available to the folks in the creditors mailing lists.
    - Announcements regarding evaluations and projects will be shared only with creditors as well as video links.
      - Teams channel interaction and TA session for creditors only.
- Online registration portal from academics.iisc.ac.in
  - Your research/faculty advisor may need to approve also before the deadline (Oct. 20th?)

Recap of previous class

Some notations

* $\boldsymbol{x} \in \mathcal{R}^D$ - input data.
* $\boldsymbol{y} \in \mathcal{R}^C$ - neural network targets.
* $\hat{\boldsymbol{y}} \in \mathcal{B}^C$ - model outputs.
* $\boldsymbol{e}, \boldsymbol{h} \in \mathcal{R}^d$ - hidden model representations or embeddings.
* $\boldsymbol{\Theta}$ - collection of learnable parameters in the model.
* $E(\boldsymbol{y}, \hat{\boldsymbol{y}})$ - error function used in the model training.



Some notations

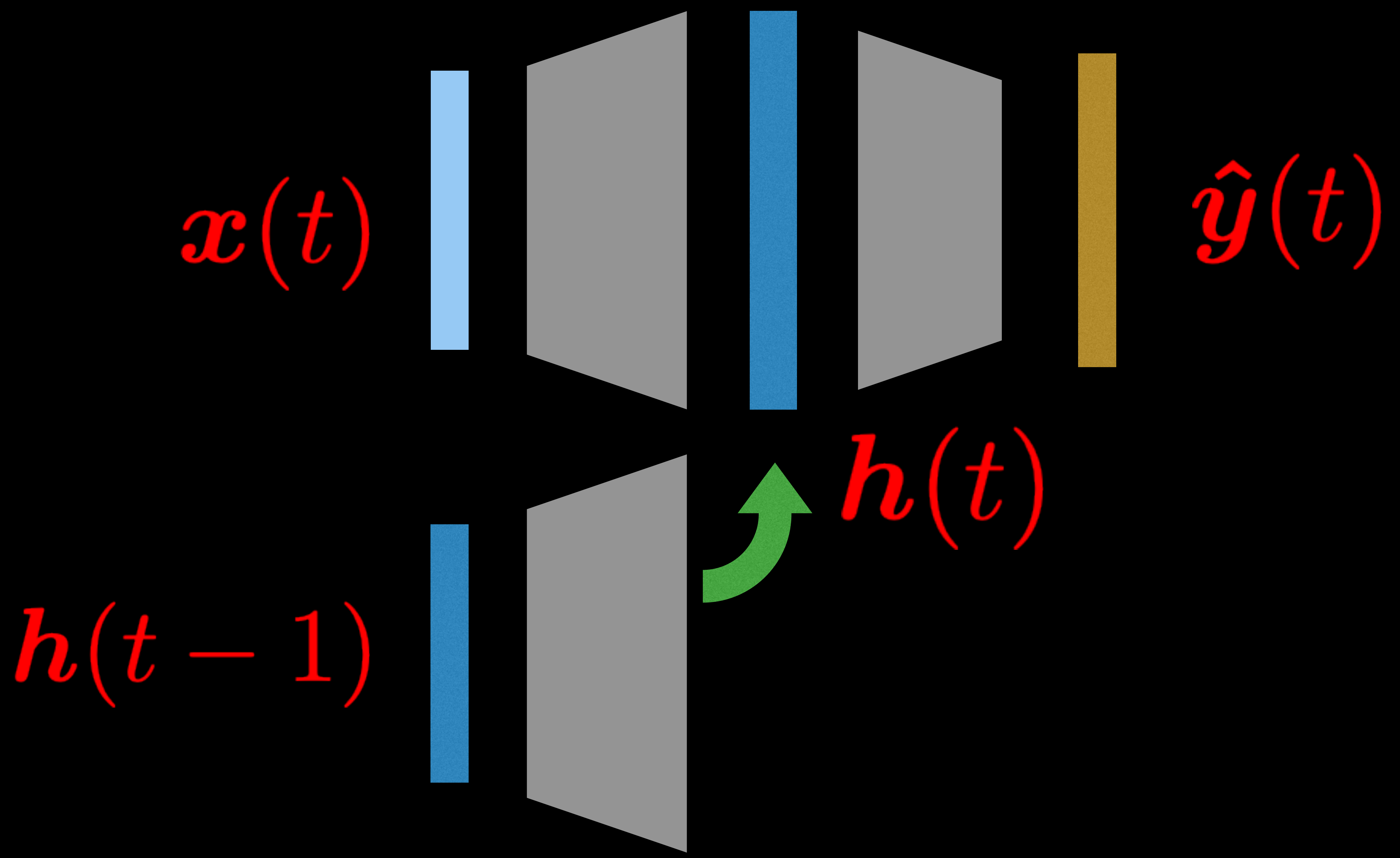
- $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N, \boldsymbol{y}_1, ..., \boldsymbol{y}_N\}$ - labeled training data
- $q = \{1...Q\}$ - iteration index.
- $t = \{1...T\}$ - discrete time index.
- $l = \{1...L\}$ - layer index
- η - learning rate (hyper-parameter)
- N_b - mini-batch size and B is the number of mini-batches.

Module - I Visual and Time Series Modeling

First order recurrence - hidden layer

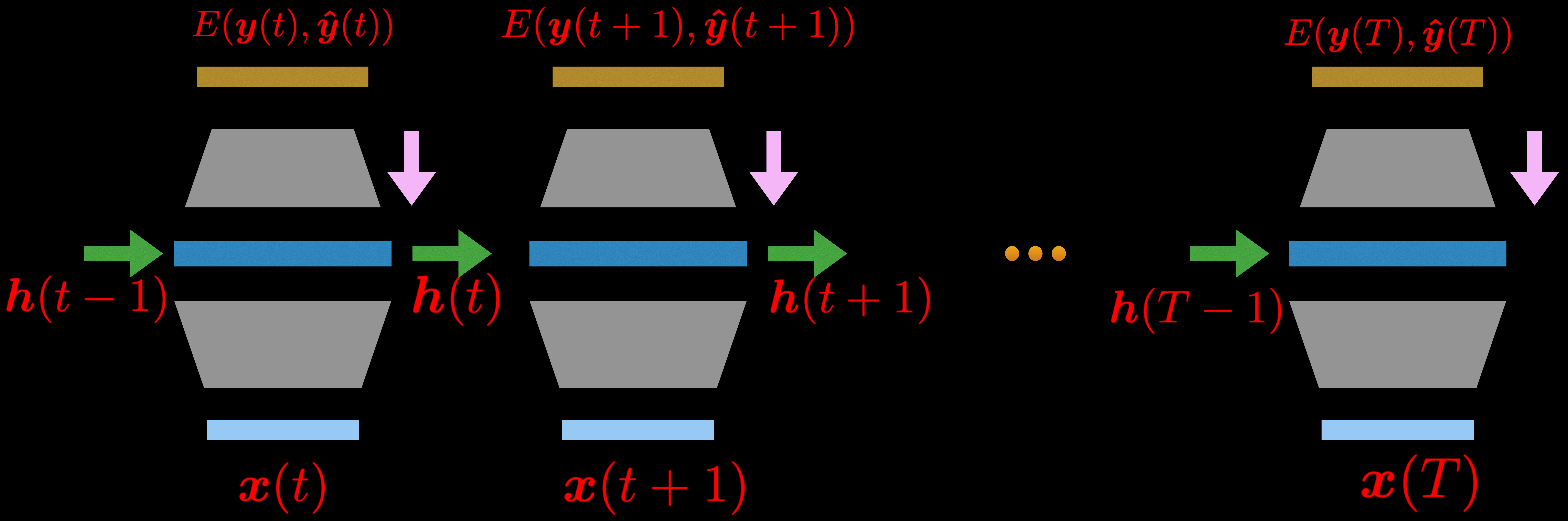
- Making the hidden layer a function of the previous outputs from the hidden layer along with the input

$$\boldsymbol{h}(t) = f\left(\boldsymbol{h}(t-1), \boldsymbol{x}(t)\right)$$



Error backpropagation

- Error functions are computed at every time-instant



- Total error $E = \sum_t E(\boldsymbol{y}(t), \hat{\boldsymbol{y}}(t))$

Error back propagation

$$\boldsymbol{a}^1(t) = \boldsymbol{W}^1\boldsymbol{x}(t) + \boldsymbol{U}^1\boldsymbol{h}^1(t-1) + b^1$$
$$\boldsymbol{h}^1(t) = tanh(\boldsymbol{a}^1(t))$$
$$\boldsymbol{a}^2(t) = \boldsymbol{W}^2\boldsymbol{h}^1(t) + b^2$$
$$\hat{\boldsymbol{y}}(t) = S(\boldsymbol{a}^2(t))$$

- ✓ Output activations

$$\frac{\partial E}{\partial \boldsymbol{a}^2(t)} = \hat{\boldsymbol{y}}(t) - \boldsymbol{y}(t) \quad for \quad t \; = \; 1 \, .. \, T$$

- ✓ Hidden activations at last instant T

$$\frac{\partial E}{\partial \boldsymbol{h}^1(T)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(T)}$$

- ✓ Hidden activations for previous instances t = T-1,… 1

$$\frac{\partial E}{\partial \boldsymbol{h}^1(t)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(t)} + \frac{\partial \boldsymbol{h}^1(t+1)}{\partial \boldsymbol{h}^1(t)} \frac{\partial E}{\partial \boldsymbol{h}^1(t+1)}$$

Error back propagation

$$\boldsymbol{a}^1(t) = \boldsymbol{W}^1\boldsymbol{x}(t) + \boldsymbol{U}^1\boldsymbol{h}^1(t-1) + b^1$$
$$\boldsymbol{h}^1(t) = tanh(\boldsymbol{a}^1(t))$$
$$\boldsymbol{a}^2(t) = \boldsymbol{W}^2\boldsymbol{h}^1(t) + b^2$$
$$\hat{\boldsymbol{y}}(t) = S(\boldsymbol{a}^2(t))$$

- ✓ Hidden activations for previous instances t = T-1,… 1

$$\frac{\partial E}{\partial \boldsymbol{h}^1(t)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(t)} + \frac{\partial \boldsymbol{h}^1(t+1)}{\partial \boldsymbol{h}^1(t)} \frac{\partial E}{\partial \boldsymbol{h}^1(t+1)}$$

$$\frac{\partial E}{\partial \boldsymbol{h}^1(t)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(t)} + (\boldsymbol{U}^1)^T diag(1 - (\boldsymbol{h}^1(t+1)).^2) \frac{\partial E}{\partial \boldsymbol{h}^1(t+1)}$$

- ✓ Here, the term $diag(1 - \boldsymbol{h}(t+1).^2)$ comes from the derivative of tanh
- ✓ and the notation $.^2$ denotes element wise operation of squaring.

Error back propagation

$$\boldsymbol{a}^1(t) = \boldsymbol{W}^1\boldsymbol{x}(t) + \boldsymbol{U}^1\boldsymbol{h}^1(t-1) + b^1$$
$$\boldsymbol{h}^1(t) = tanh(\boldsymbol{a}^1(t))$$
$$\boldsymbol{a}^2(t) = \boldsymbol{W}^2\boldsymbol{h}^1(t) + b^2$$
$$\hat{\boldsymbol{y}}(t) = S(\boldsymbol{a}^2(t))$$

- ✓ The derivatives of the output weights.

$$\frac{\partial E}{\partial \boldsymbol{W}^2} = \sum_t \frac{\partial E}{\partial \boldsymbol{a}^2(t)} (\boldsymbol{h}^1(t))^T$$

$$\frac{\partial E}{\partial \boldsymbol{b}^2} = \sum_t \frac{\partial E}{\partial \boldsymbol{a}^2(t)}$$

- ✓ Transferring derivatives to the first layer

$$\frac{\partial E}{\partial \boldsymbol{a}^1(t)} = diag(1 - \boldsymbol{h}(t).^2)\frac{\partial E}{\partial \boldsymbol{h}^1(t)}$$

Error back propagation

- ✓ The derivatives of the first layer weights.

$$\boldsymbol{a}^1(t) = \boldsymbol{W}^1\boldsymbol{x}(t) + \boldsymbol{U}^1\boldsymbol{h}^1(t-1) + b^1$$
$$\boldsymbol{h}^1(t) = tanh(\boldsymbol{a}^1(t))$$
$$\boldsymbol{a}^2(t) = \boldsymbol{W}^2\boldsymbol{h}^1(t) + b^2$$
$$\hat{\boldsymbol{y}}(t) = S(\boldsymbol{a}^2(t))$$

$$\frac{\partial E}{\partial \boldsymbol{W}^1} = \sum_t \frac{\partial E}{\partial \boldsymbol{a}^1(t)} (\boldsymbol{x}^1(t))^T$$

$$\frac{\partial E}{\partial \boldsymbol{U}^1} = \sum_t \frac{\partial E}{\partial \boldsymbol{a}^1(t)} (\boldsymbol{h}^1(t))^T$$

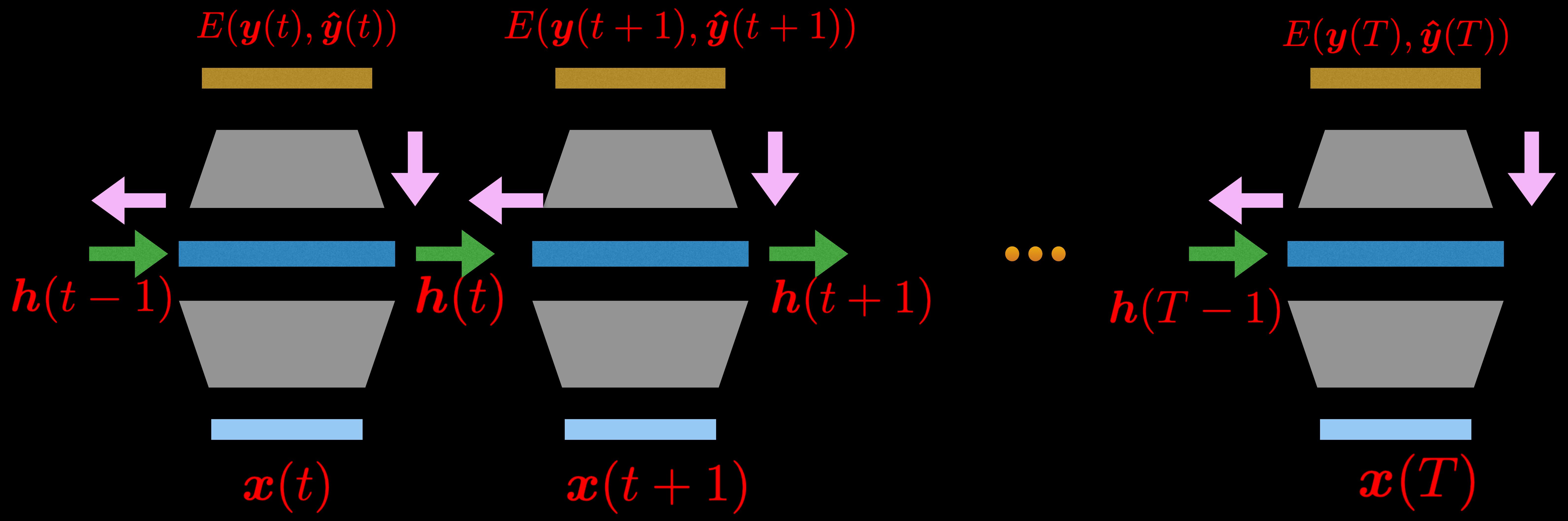
$$\frac{\partial E}{\partial \boldsymbol{b}^1} = \sum_t \frac{\partial E}{\partial \boldsymbol{a}^1(t)}$$

Error Backpropagation

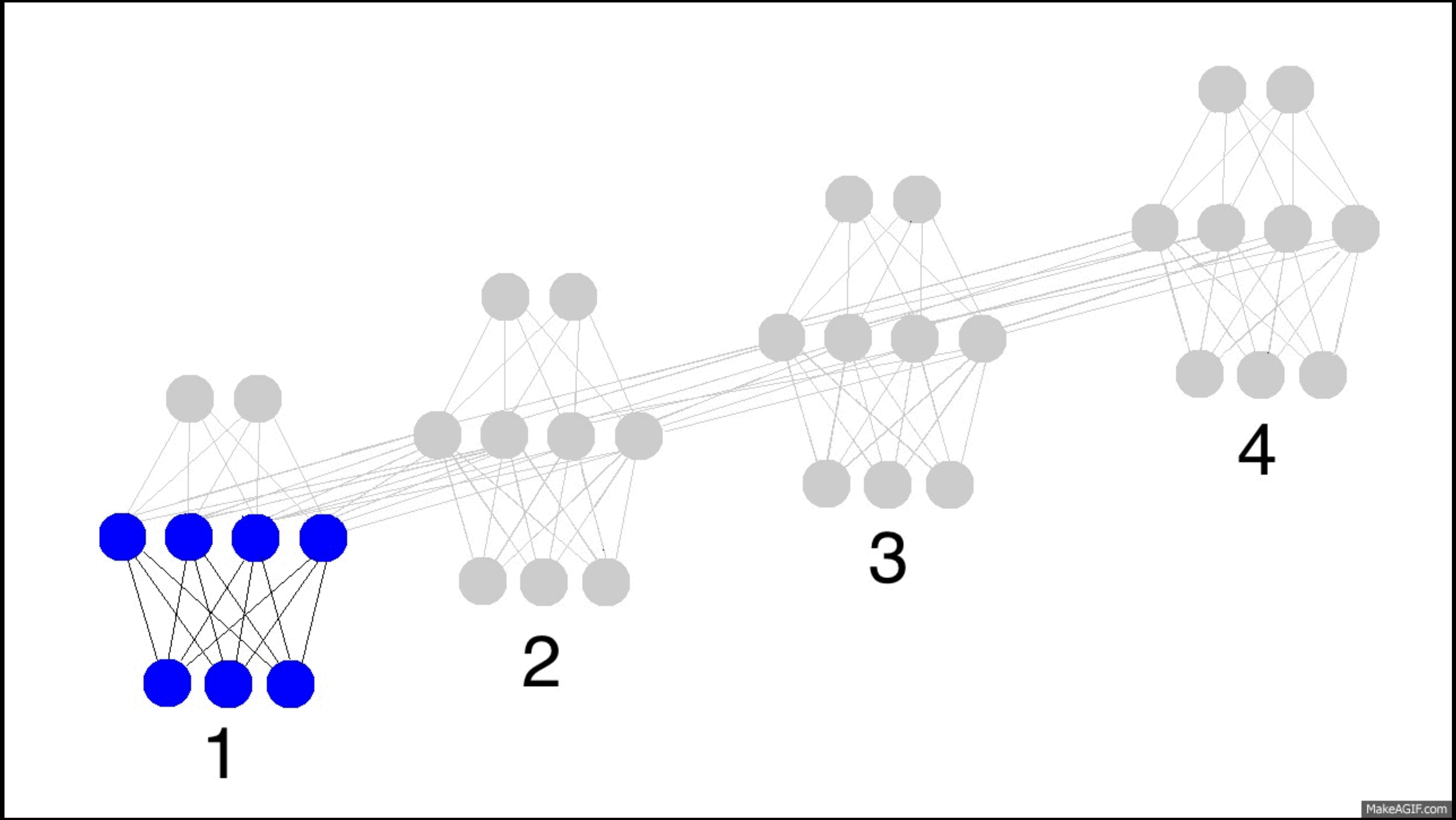
- Key equation in the backward direction

$$\frac{\partial E}{\partial \boldsymbol{h}^1(t)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(t)} + (\boldsymbol{U}^1)^T diag(1 - (\boldsymbol{h}^1(t+1)).^2) \frac{\partial E}{\partial \boldsymbol{h}^1(t+1)}$$

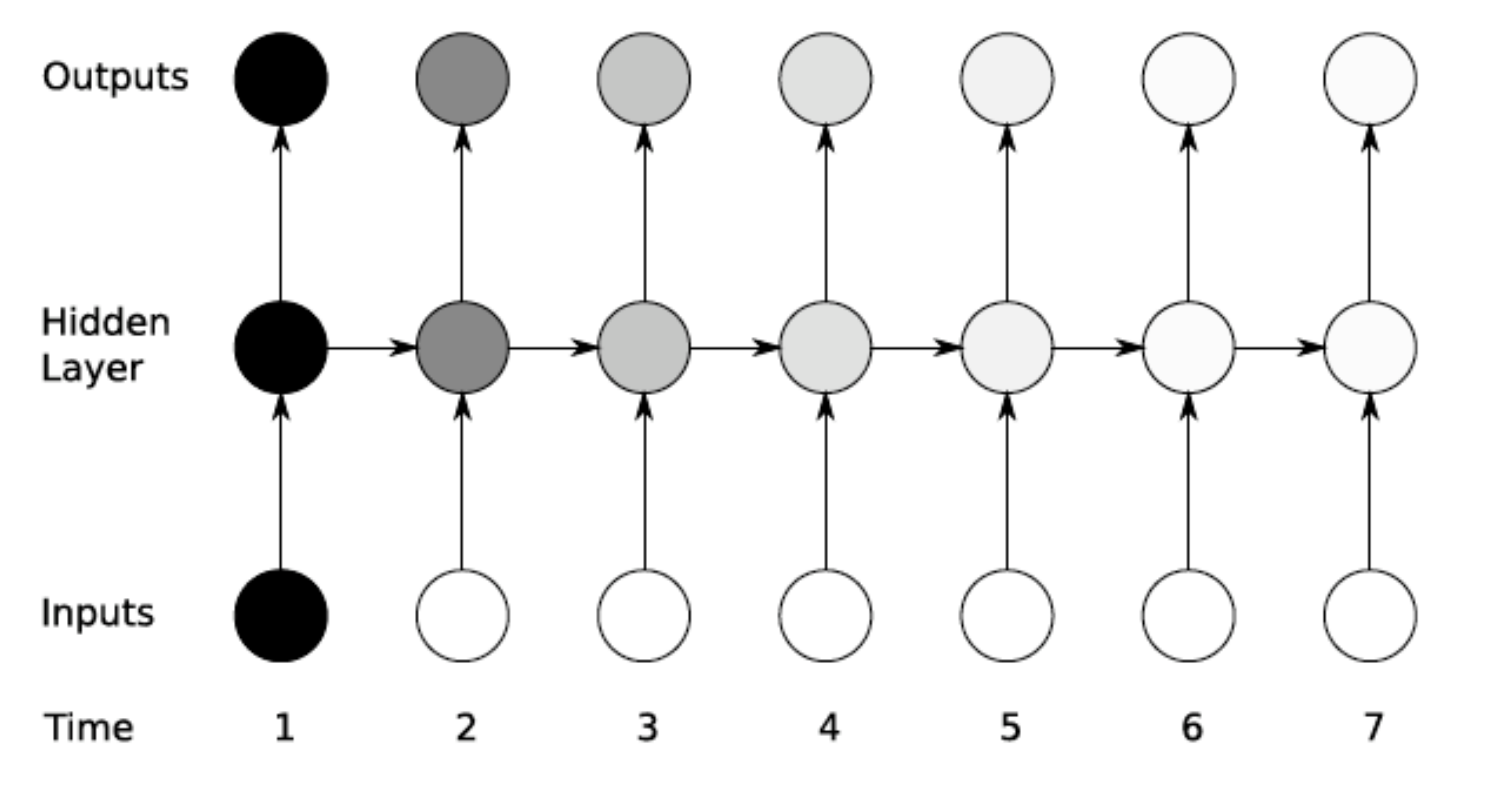
- When the model incorporates a recurrence in the forward direction.
    - ✓ Gradients incorporate a recurrence in the backward direction.



Back propagation through time

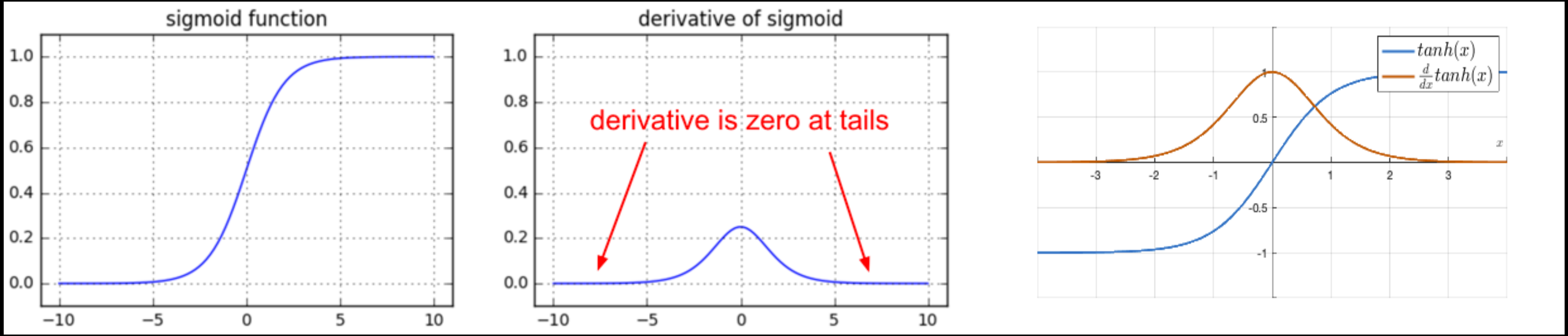


Long-term dependency issues



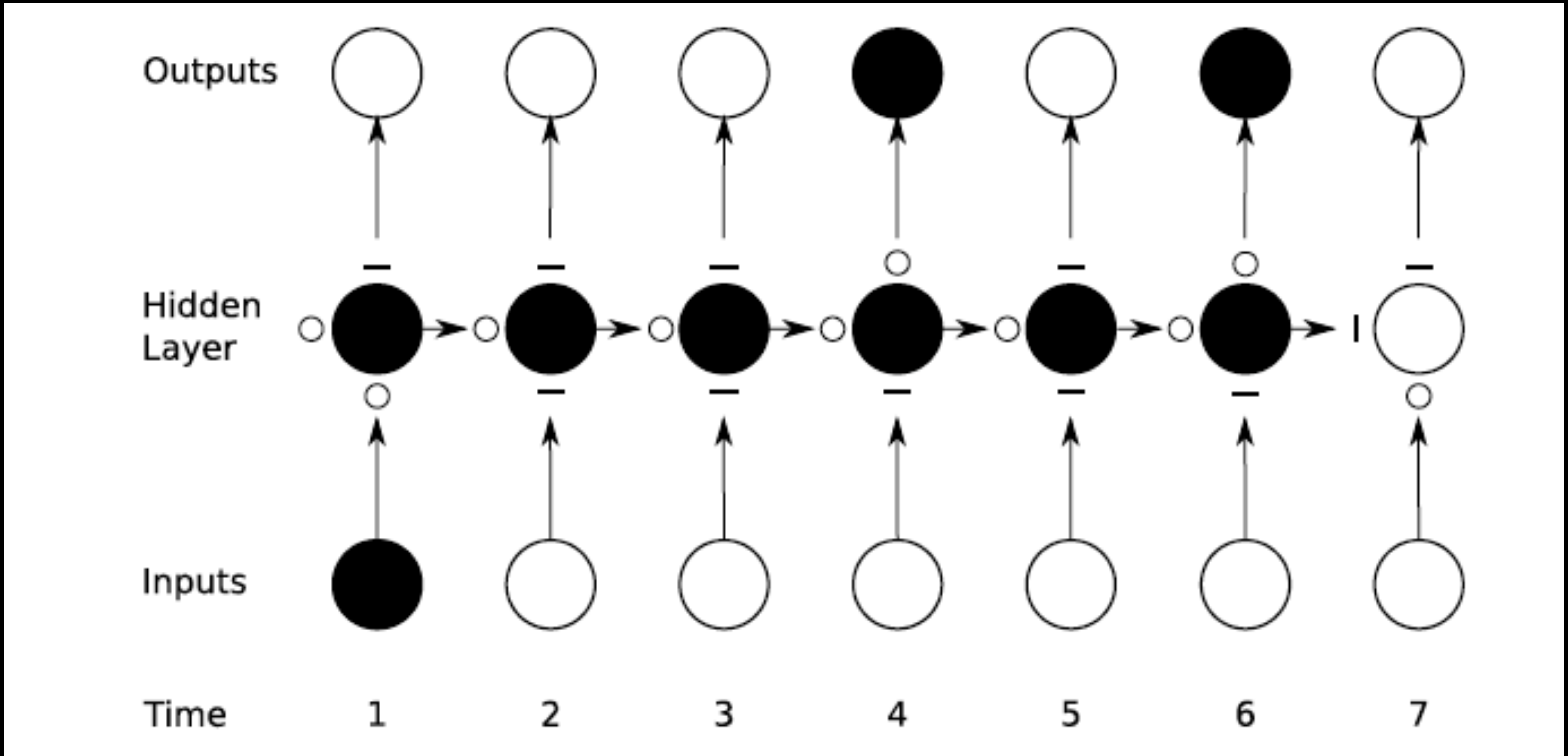
Long-term dependency issues

- ✓ Gradients tend to vanish or explode



- ✓ Intial frames may not have impact in the final predictions.

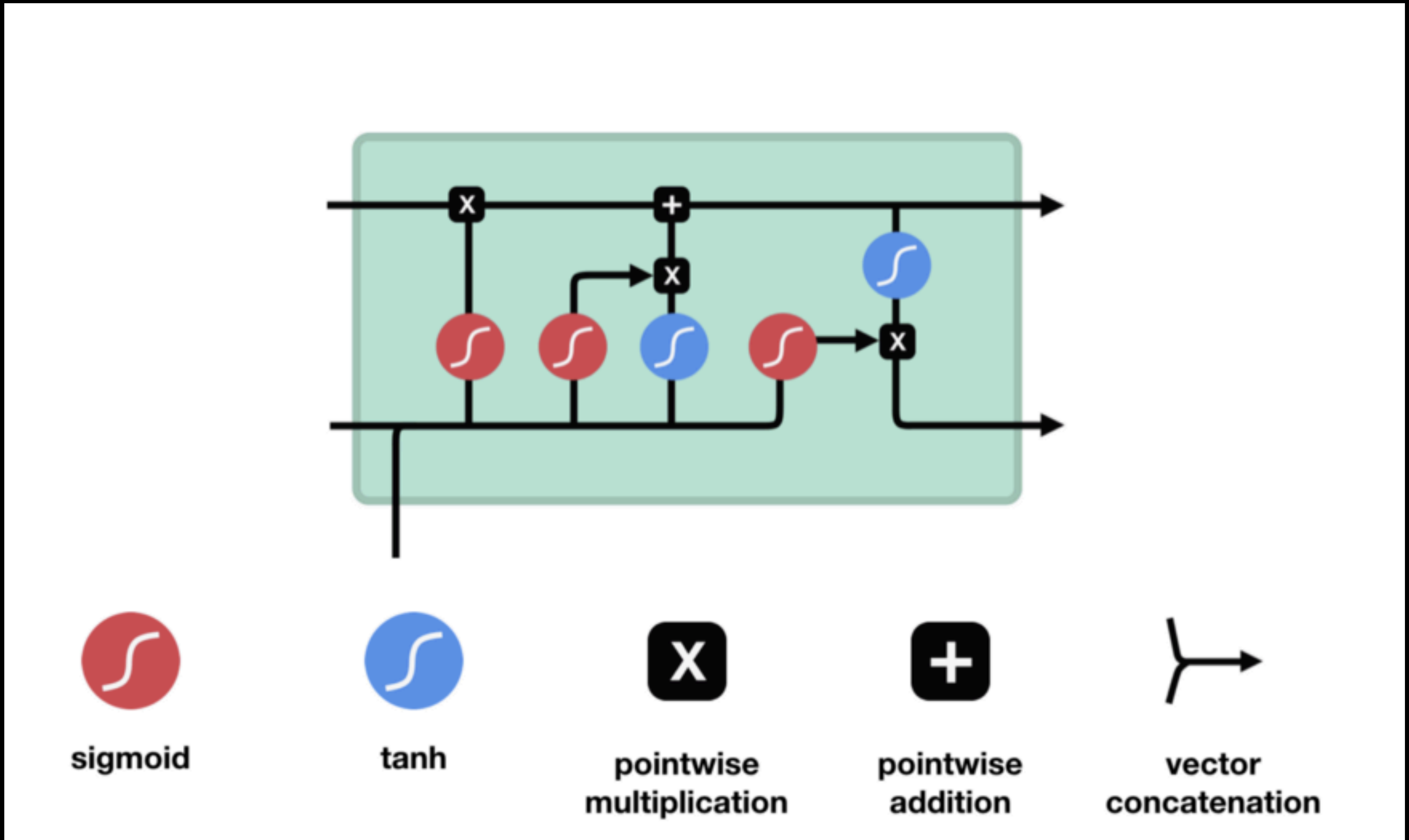
Long short term memory (LSTM) idea



Modeling questions

- ✳ How can we make adaptable gates with neural networks
  - ➡ How can we make gates dependent on the data itself.
    - ✓ Gates can be implemented as neural layers with sigmoidal outputs ?
      - ★ Sigmoids can approximate 0-1 functions
    - ◉ Modulate the gate output with inputs, hidden layer outputs or outputs

Long-short term memory - idea



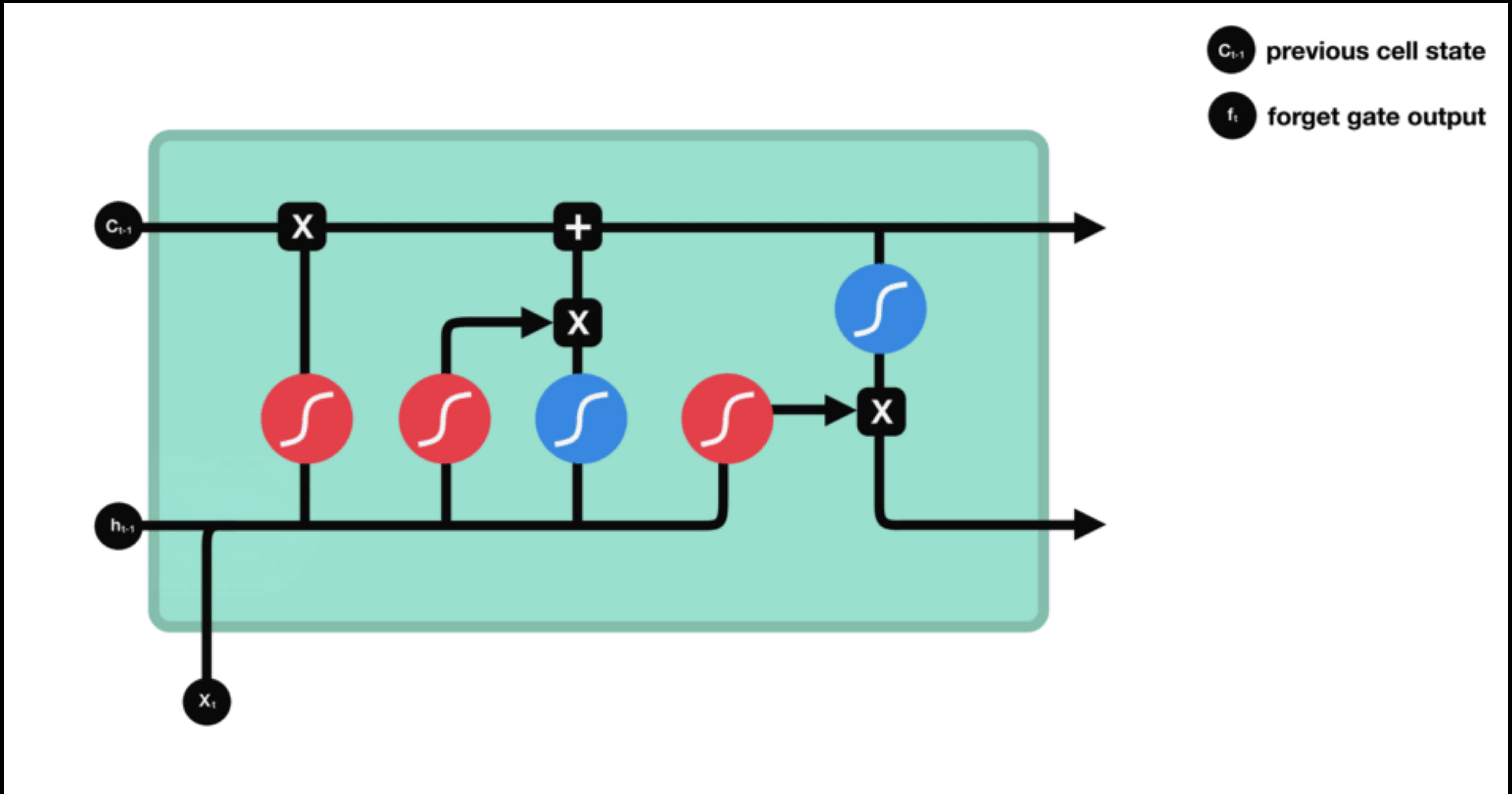
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Long short term memory - idea

✴ Forget gate

$$a_\phi^t = \sum_{i=1}^{I} w_{i\phi} x_i^t + \sum_{h=1}^{H} w_{h\phi} b_h^{t-1}$$

$$b_\phi^t = f(a_\phi^t)$$



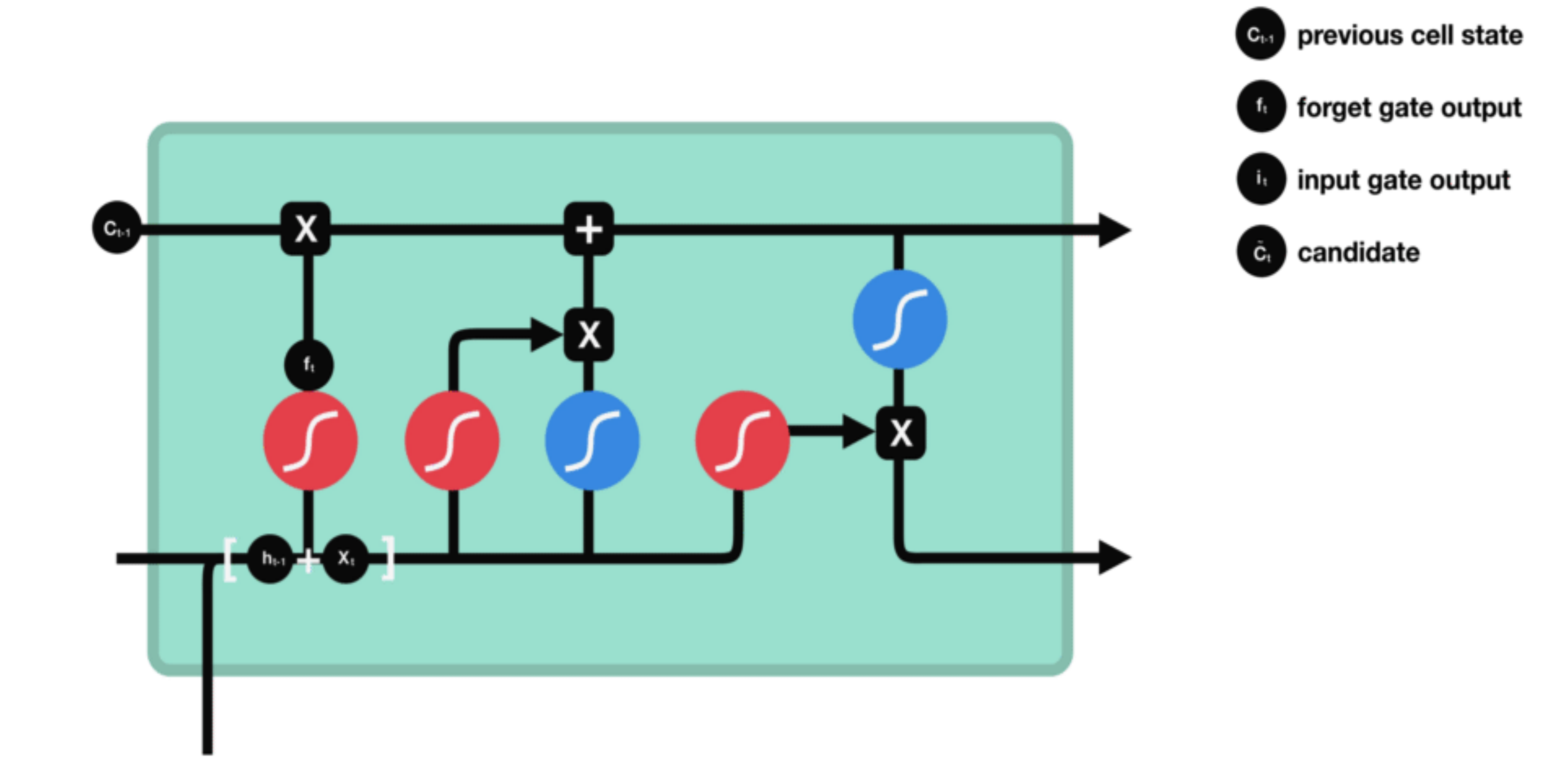
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Long short term memory - idea

- Input gate

$$a_\iota^t = \sum_{i=1}^{I} w_{i\iota} x_i^t + \sum_{h=1}^{H} w_{h\iota} b_h^{t-1}$$

$$b_\iota^t = f(a_\iota^t)$$



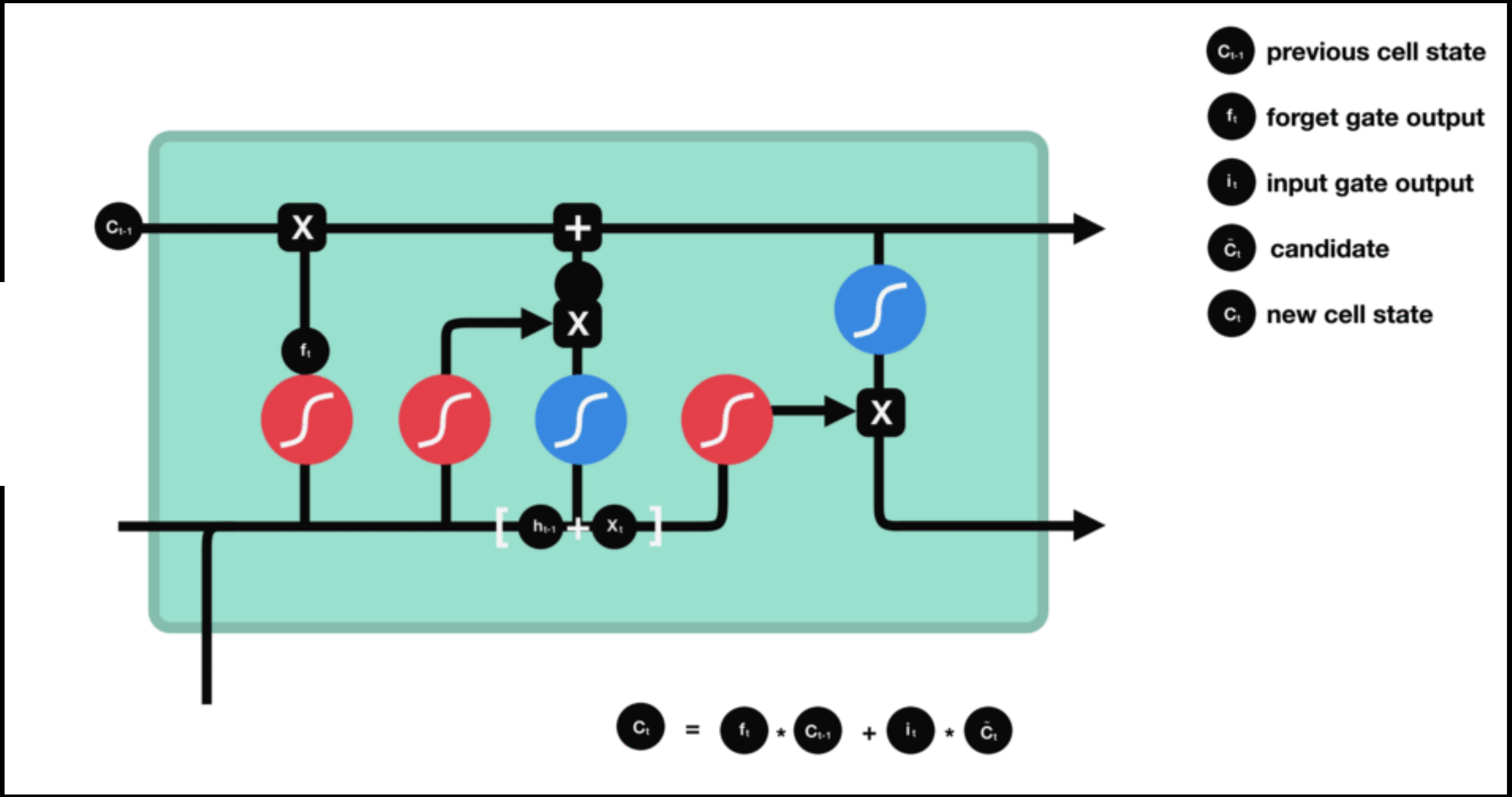
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Long-short term memory - idea

- Cell state

$$a_c^t = \sum_{i=1}^{I} w_{ic} x_i^t + \sum_{h=1}^{H} w_{hc} b_h^{t-1}$$

$$s_c^t = b_\phi^t s_c^{t-1} + b_\iota^t g(a_c^t)$$



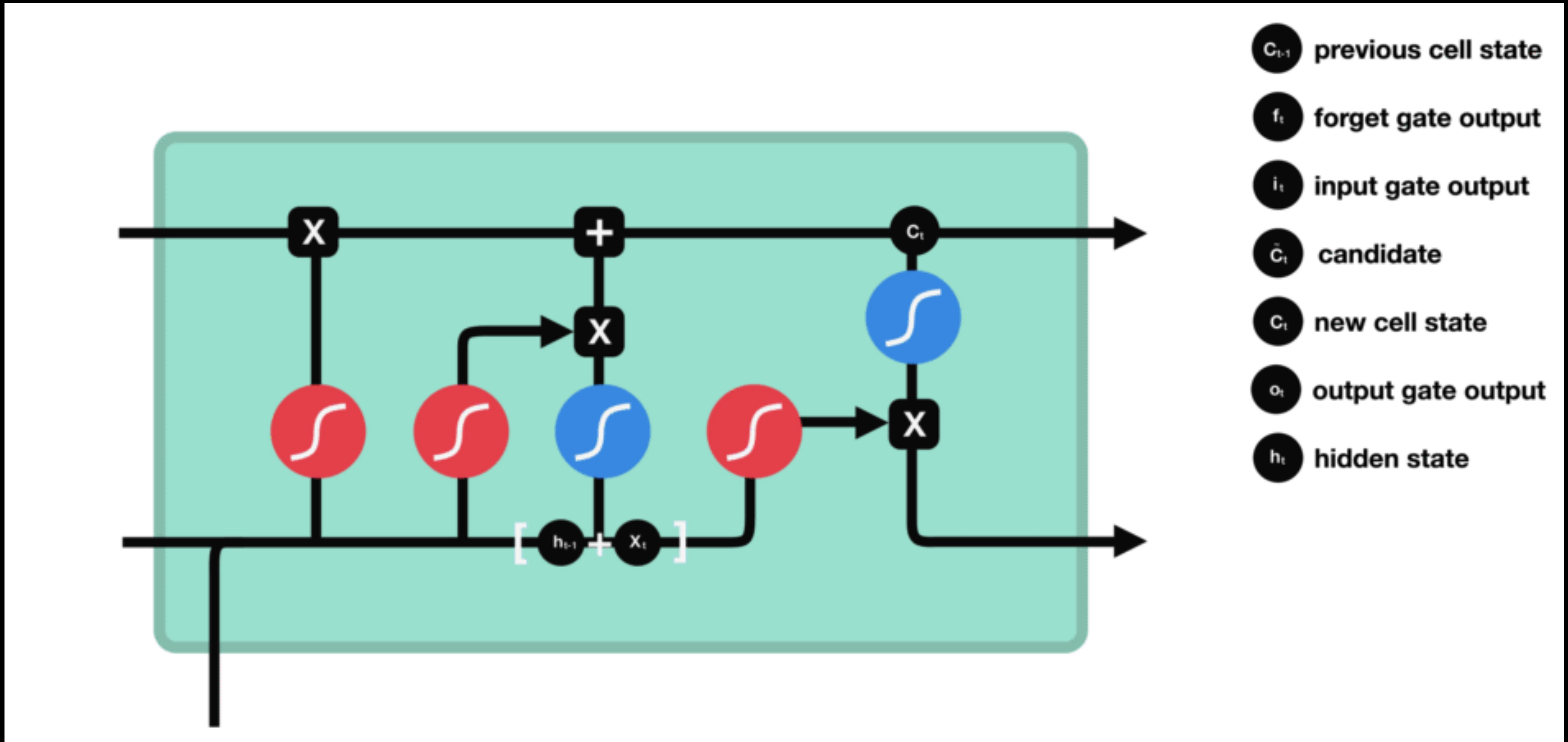
https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Long-short term memory - idea

- Output gate

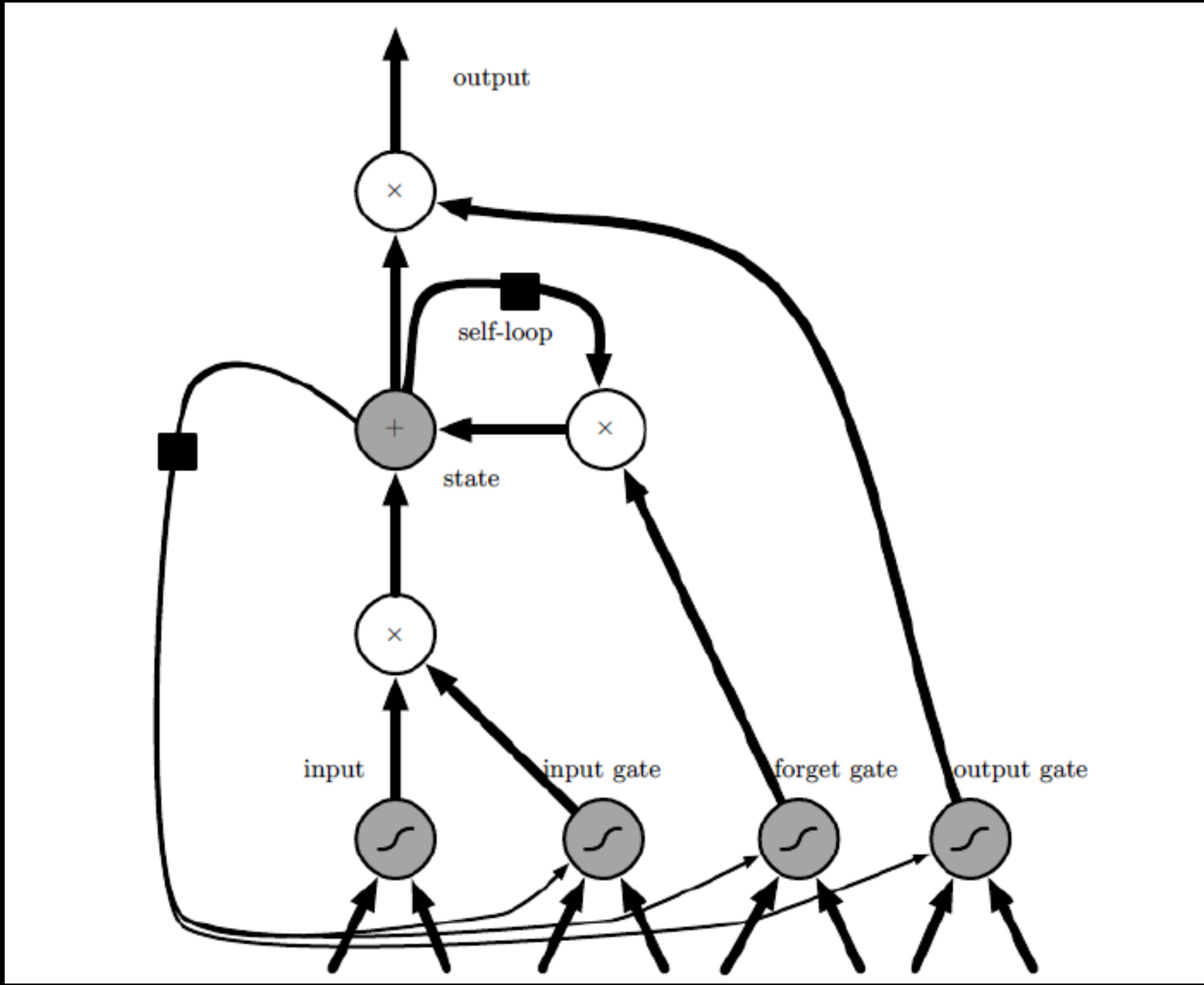
$$a_\omega^t = \sum_{i=1}^{I} w_{i\omega} x_i^t + \sum_{h=1}^{H} w_{h\omega} b_h^{t-1}$$

$$b_\omega^t = f(a_\omega^t)$$



https://towardsdatascience.com/illustrated-guide-to-lstms-and-gru-s-a-step-by-step-explanation-44e9eb85bf21

Long short-term memory - idea



Long-short term memory and GRUs