Deep Learning for Healthcare

Recurrent Neural Networks (RNN)

Jimeng Sun

ILLINOIS

Recurrent Neural Networks

ILLINOIS

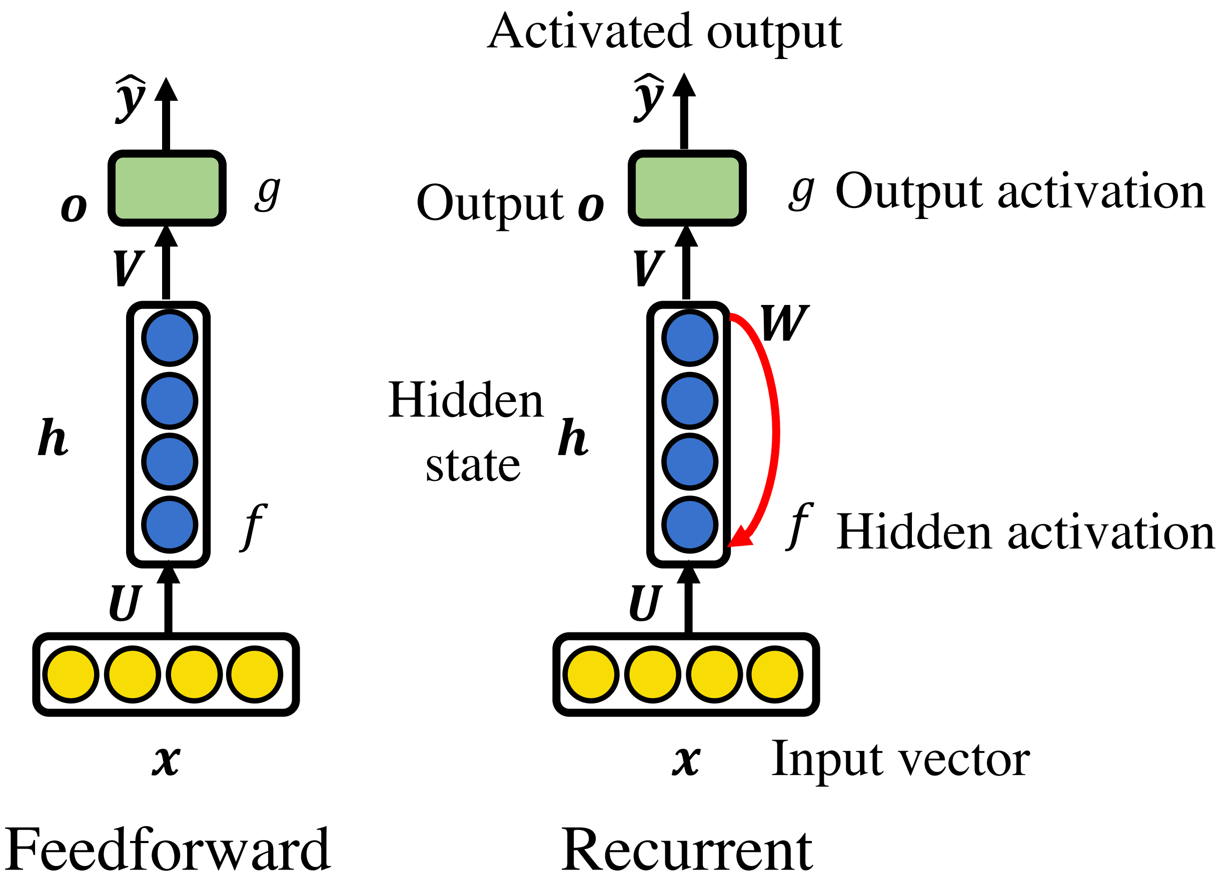
Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

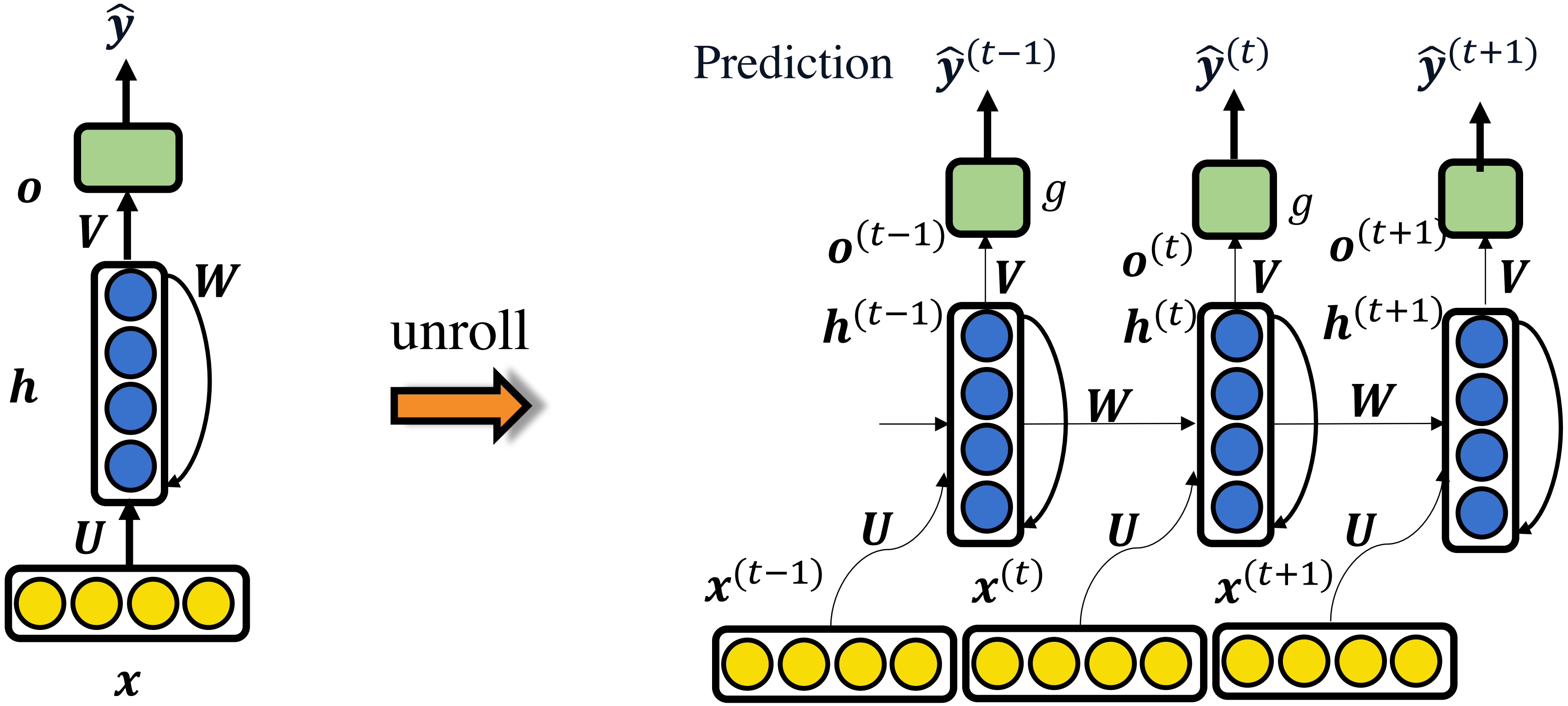
Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

Basic Concepts of RNN

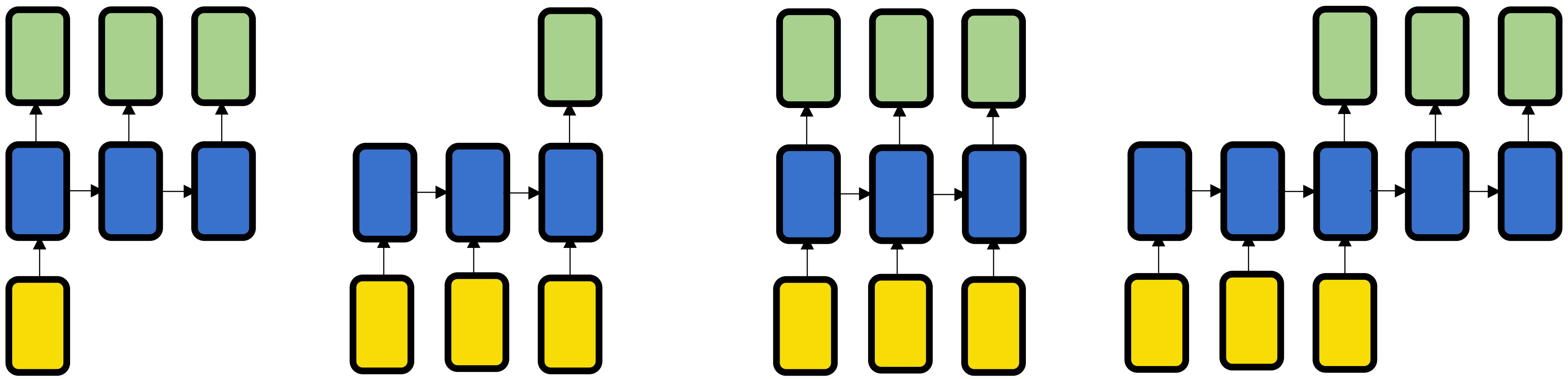


Basic RNN Structure



$$\boldsymbol{h}^{(t)} = f(\boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_1)$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{V}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2$$
$$\hat{\boldsymbol{y}}^{(t)} = g(\boldsymbol{o}^{(t)})$$

Basic RNN Structure



One-to-Many
(e.g., image
to text)

Many-to-One
(e.g., sequence
classification)

Many-to-Many
(e.g., sequential
prediction)

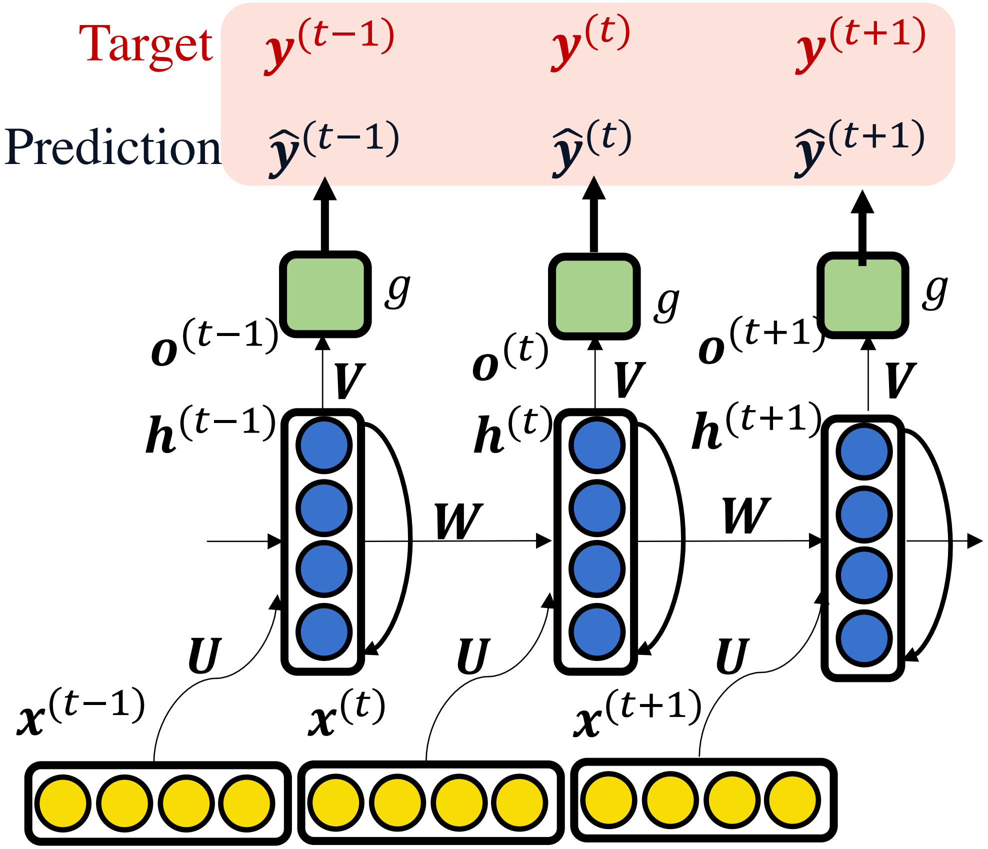
Many-to-Many
(e.g., seq2seq)

The figure is inspired by page 12 on http://cs231n.stanford.edu/slides/2017/cs231n_2017_lecture10.pdf

Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

Forward Computation



$$\boldsymbol{z}^{(t)} = \boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_1$$
$$\boldsymbol{h}^{(t)} = f(\boldsymbol{z}^{(t)})$$
$$\boldsymbol{o}^{(t)} = \boldsymbol{V}\boldsymbol{h}^{(t)} + \boldsymbol{b}_2$$
$$\hat{\boldsymbol{y}}^{(t)} = g(\boldsymbol{o}^{(t)})$$
$$L = \sum_t L^{(t)} = -\sum_t \log p(\boldsymbol{y}^{(t)} | \{\boldsymbol{x}^{(t)}, \cdots, \boldsymbol{x}^{(T)}\})$$

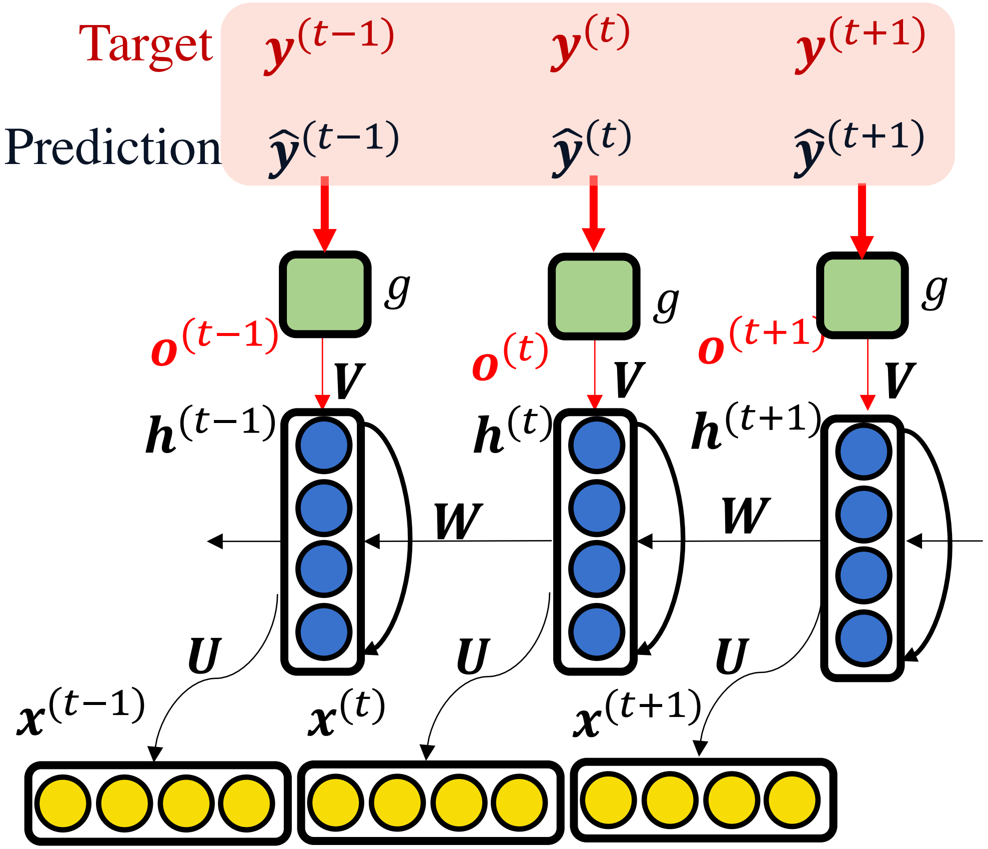
e.g.

- f is tanh
- g is softmax, which produces a normalized probability over output classes
- $L^{(t)}$ is negative log-likelihood loss

e.g., in binary classification

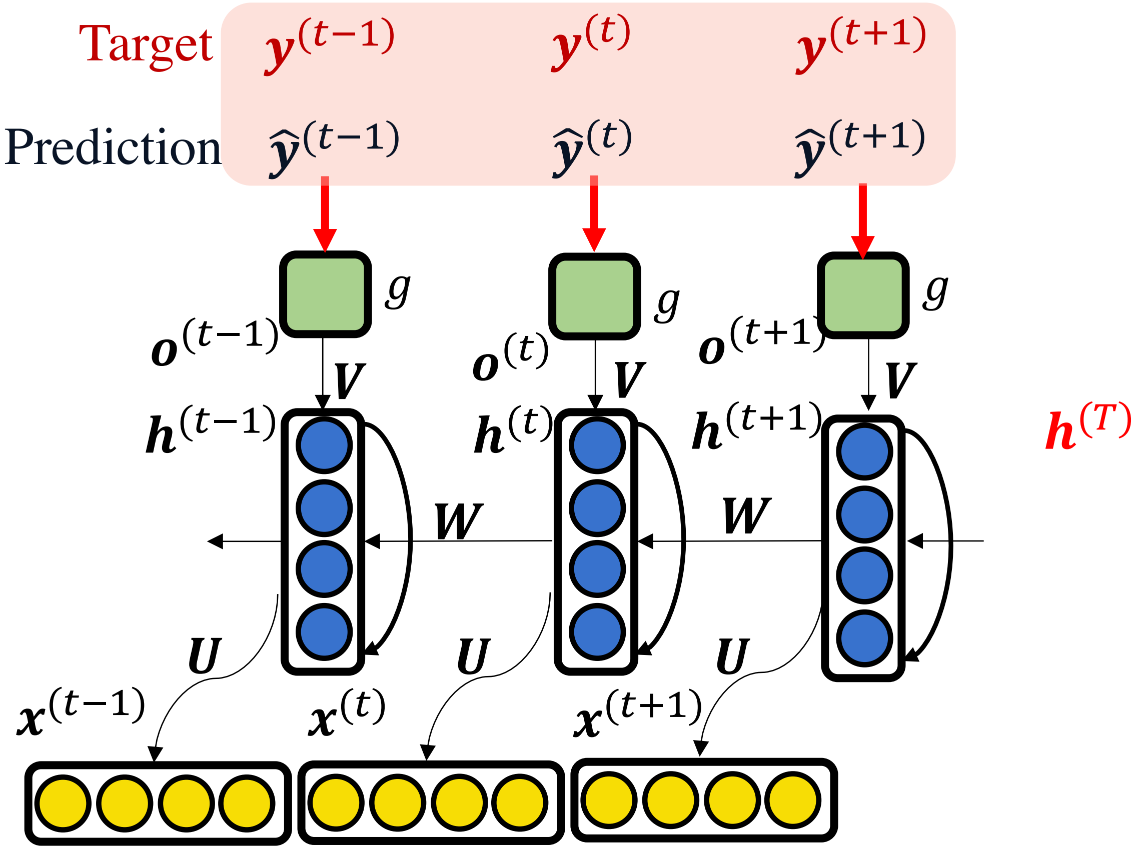
$$L = -\sum_t y^{(t)} \log \hat{y}^{(t)} + (1 - y^{(t)}) \log(1 - \hat{y}^{(t)})$$

Backpropagation through time (BPTT): $\nabla_{\boldsymbol{o}^{(t)}} L$



$$\left(\nabla_{\boldsymbol{o}^{(t)}} L\right)_i = \frac{\partial L}{\partial \boldsymbol{o}_i^{(t)}} = \frac{\partial L}{\partial L^{(t)}} \frac{\partial L^{(t)}}{\partial \boldsymbol{o}_i^{(t)}}$$

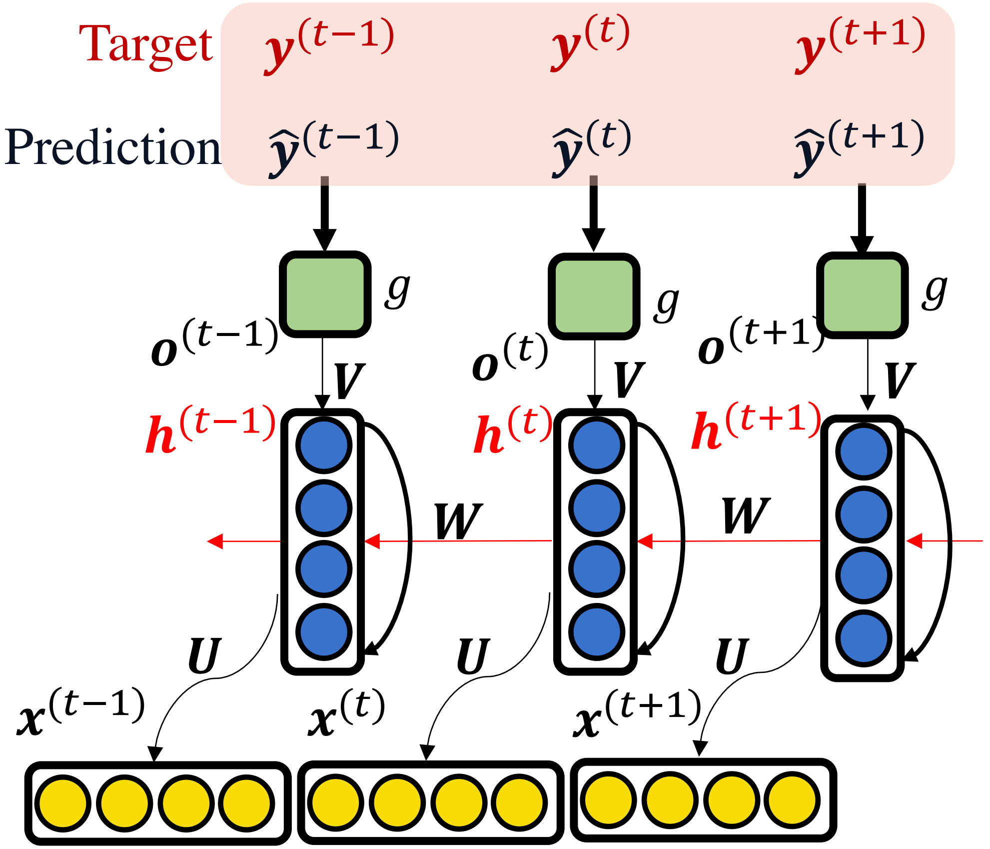
Backpropagation through time (BPTT): $\nabla_{\boldsymbol{h}^{(T)}} L$



Last time stamp

$$\nabla_{\boldsymbol{h}^{(T)}} L = V^T \nabla_{\boldsymbol{o}^{(T)}} L$$

Backpropagation through time (BPTT): $\nabla_{\boldsymbol{h}^{(t)}} L$



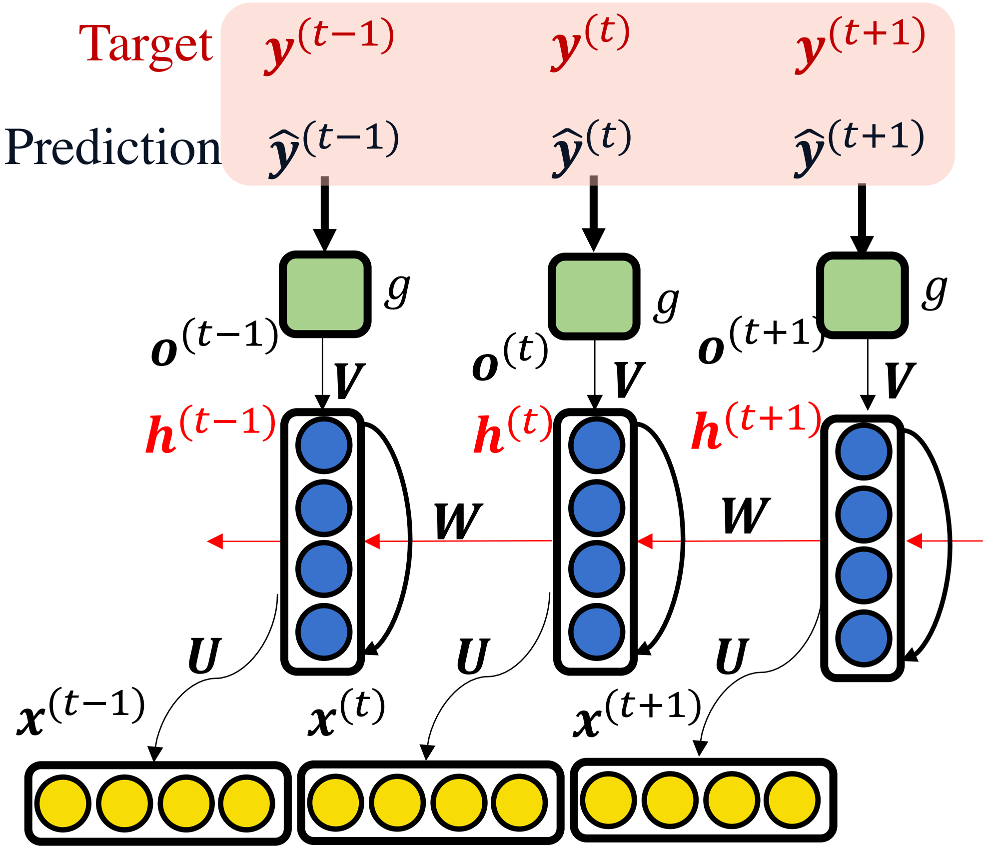
Last time stamp

$$\nabla_{\boldsymbol{h}^{(T)}} L = V^T \nabla_{\boldsymbol{o}^{(T)}} L$$

Other time stamp

$$\nabla_{\boldsymbol{h}^{(t)}} L = \left(\frac{\partial \boldsymbol{h}^{(t+1)}}{\partial \boldsymbol{h}^{(t)}}\right)^{\boldsymbol{T}} \left(\nabla_{\boldsymbol{h}^{(t+1)}} L\right) + \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{h}^{(t)}}\right)^{\boldsymbol{T}} \left(\nabla_{\boldsymbol{o}^{(t)}} L\right)$$

Backpropagation through time (BPTT)



$$\nabla_{\boldsymbol{V}}\mathcal{L} = \sum_t \sum_k \left(\frac{\partial \mathcal{L}}{\partial o_k^{(t)}}\right) \nabla_{\boldsymbol{V}} o_k^{(t)} = \sum_t (\nabla_{\boldsymbol{o}^{(t)}}\mathcal{L})\boldsymbol{h}^{(t)^T}$$

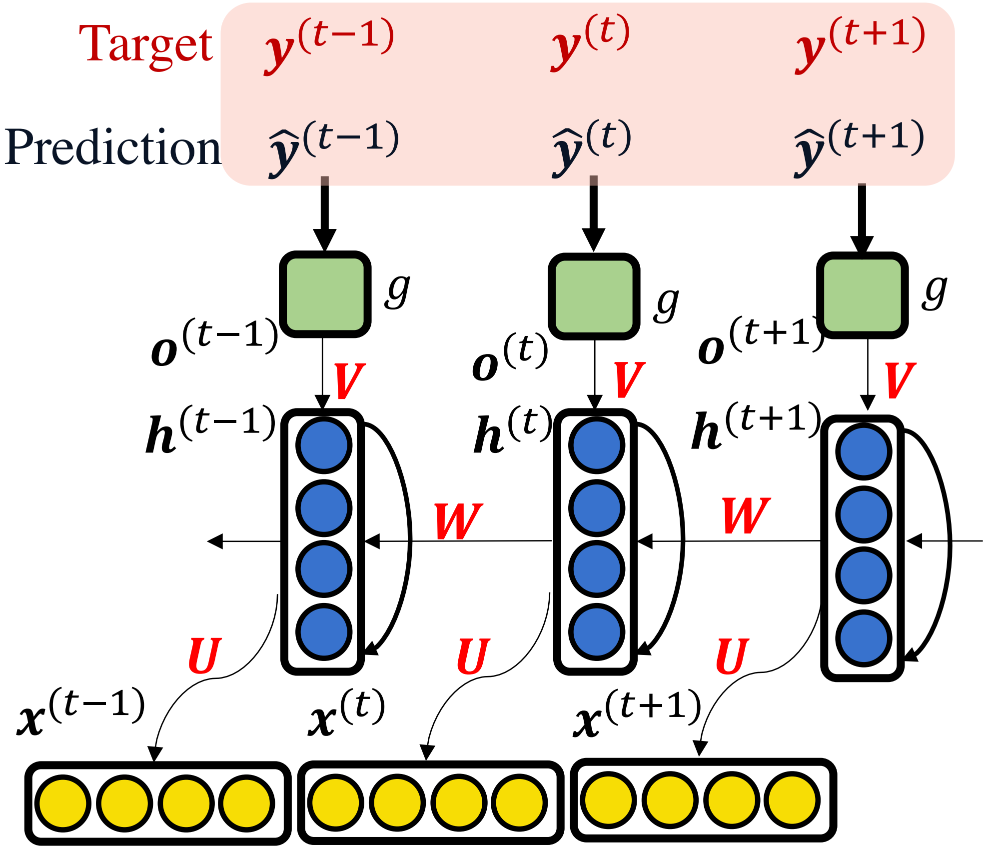
$$\nabla_{\boldsymbol{W}}\mathcal{L} = \sum_t \sum_j \left(\frac{\partial \mathcal{L}}{\partial h_j^{(t)}}\right) \nabla_{\boldsymbol{W}} h_j^{(t)} = \sum_t diag(1-(\boldsymbol{h}^{(t)})^2)(\nabla_{\boldsymbol{h}^{(t)}}\mathcal{L})\boldsymbol{h}^{(t-1)^T}$$

$$\nabla_{\boldsymbol{U}}\mathcal{L} = \sum_t \sum_j \left(\frac{\partial \mathcal{L}}{\partial h_j^{(t)}}\right) \nabla_{\boldsymbol{U}} h_j^{(t)} = \sum_t diag(1-(\boldsymbol{h}^{(t)})^2)(\nabla_{\boldsymbol{h}^{(t)}}\mathcal{L})\boldsymbol{x}^{(t)^T}$$

$$\nabla_{\boldsymbol{b}_1}\mathcal{L} = \sum_t \left(\frac{\partial \boldsymbol{h}^{(t)}}{\partial \boldsymbol{b}^{(t)}}\right)^T \nabla_{\boldsymbol{h}^{(t)}}\mathcal{L} = \sum_t diag(1-(\boldsymbol{h}^{(t)})^2)\nabla_{\boldsymbol{h}^{(t)}}\mathcal{L}$$

$$\nabla_{\boldsymbol{b}_2}\mathcal{L} = \sum_t \left(\frac{\partial \boldsymbol{o}^{(t)}}{\partial \boldsymbol{b}_2}\right)^T \nabla_{\boldsymbol{o}^{(t)}}\mathcal{L} = \sum_t \nabla_{\boldsymbol{o}^{(t)}}\mathcal{L}$$

BPTT: Vanishing gradient problem



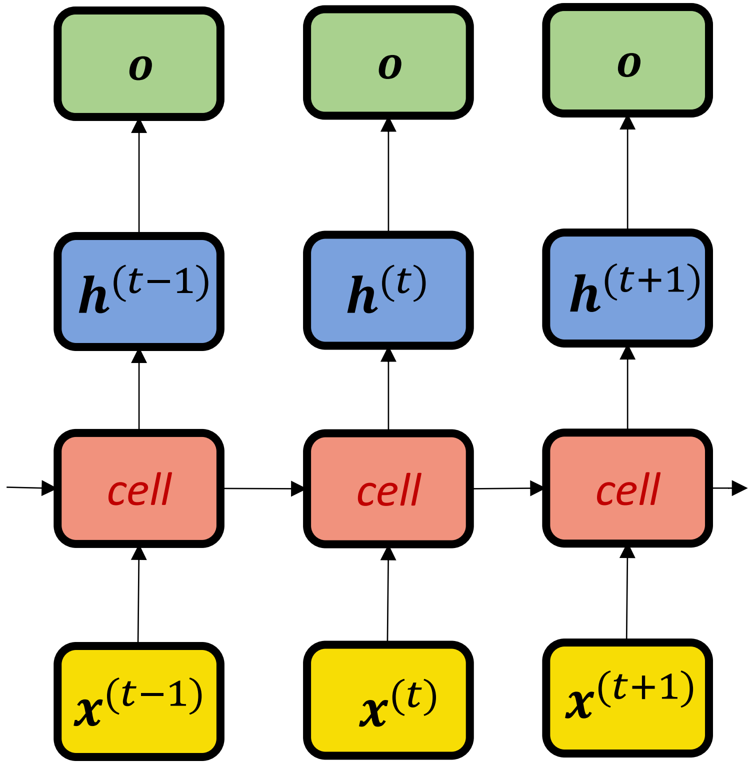
- Gradient can become very small over a long sequence
- Standard RNN will have difficulty to "remember" state from early part of the input sequence

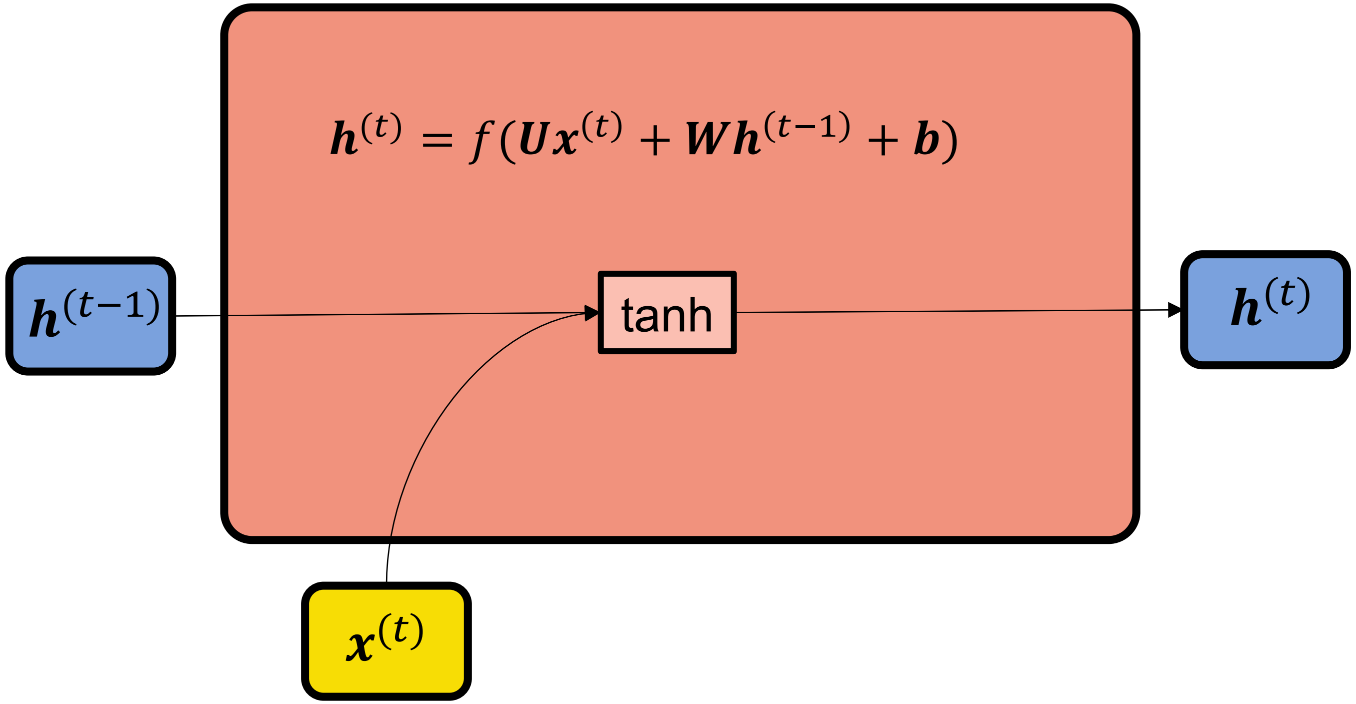
Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

ILLINOIS

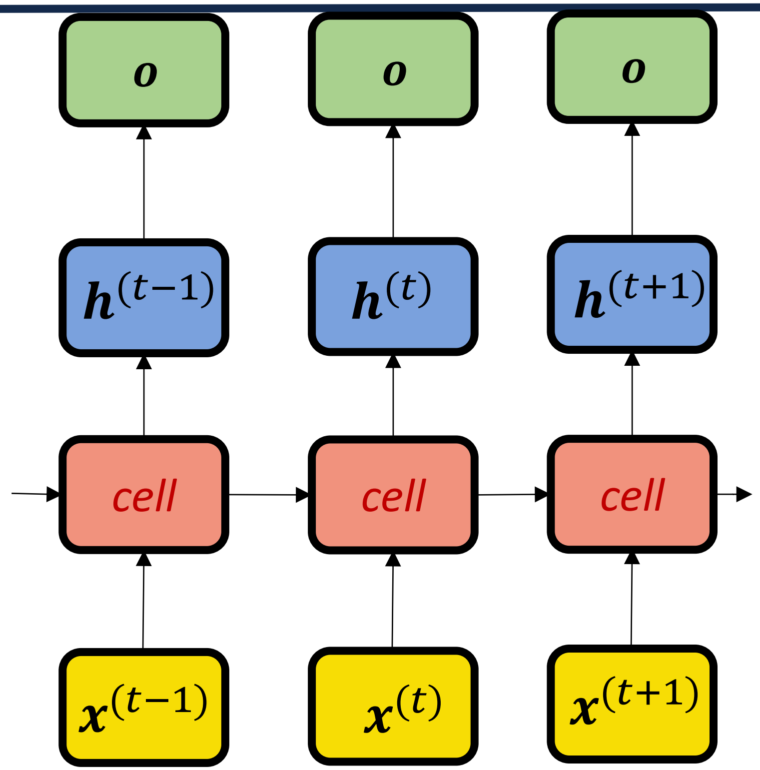
Standard RNN



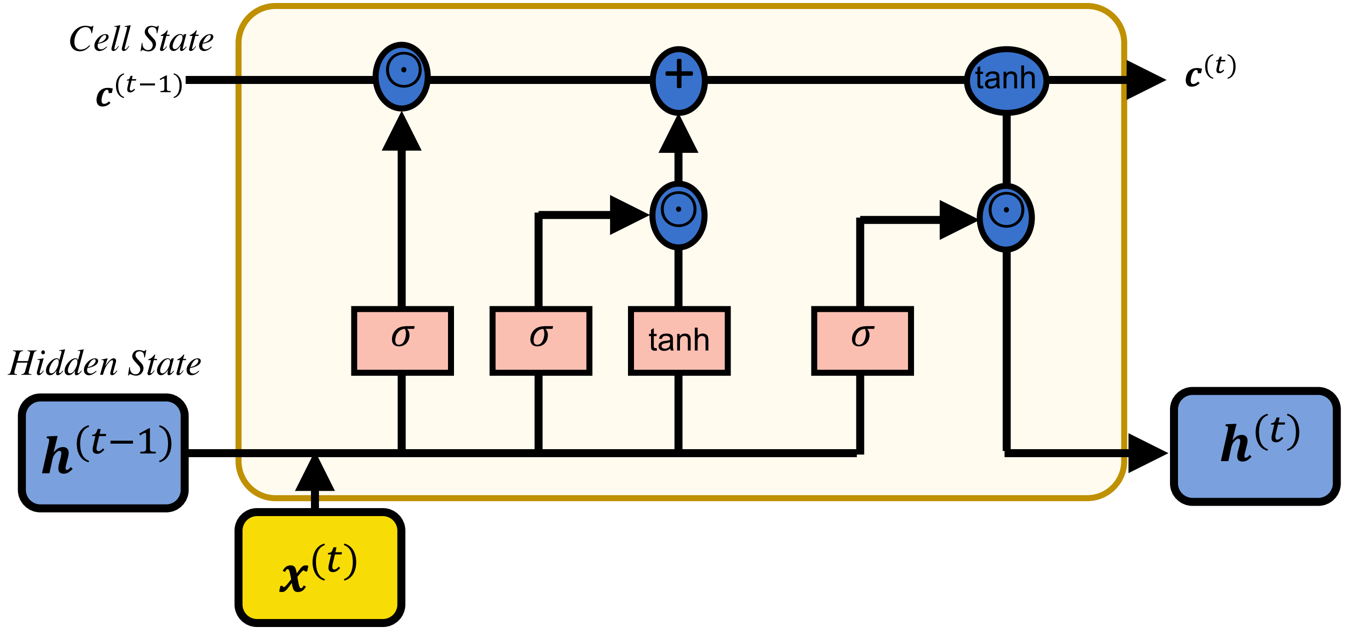


- Standard RNN has a simple computation *cell* from input $\boldsymbol{x}$ to latent state $\boldsymbol{h}$

Long short term memory networks (LSTM)

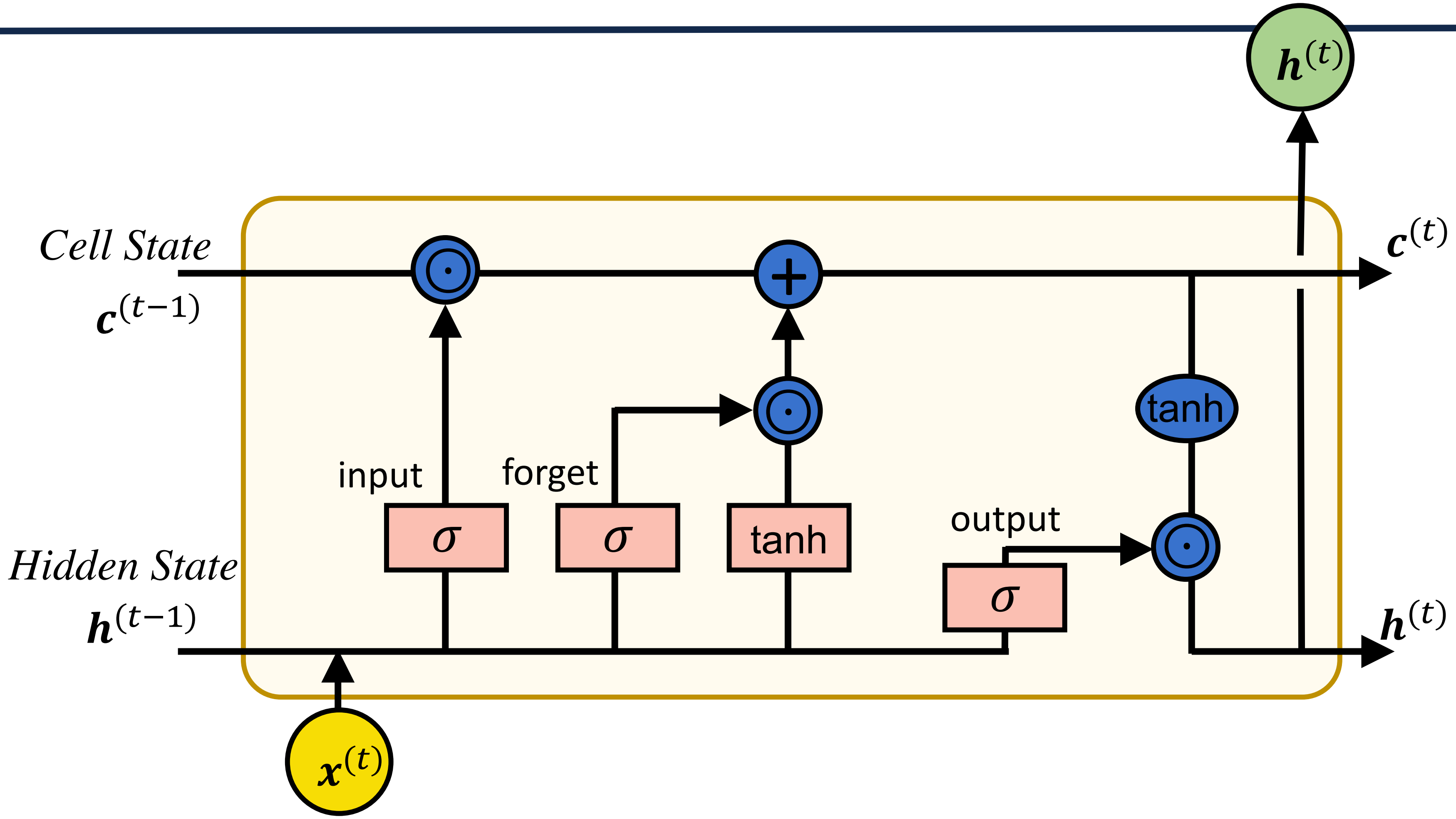


$$h^{(t)} = f(Ux^{(t)} + Wh^{(t-1)} + b)$$

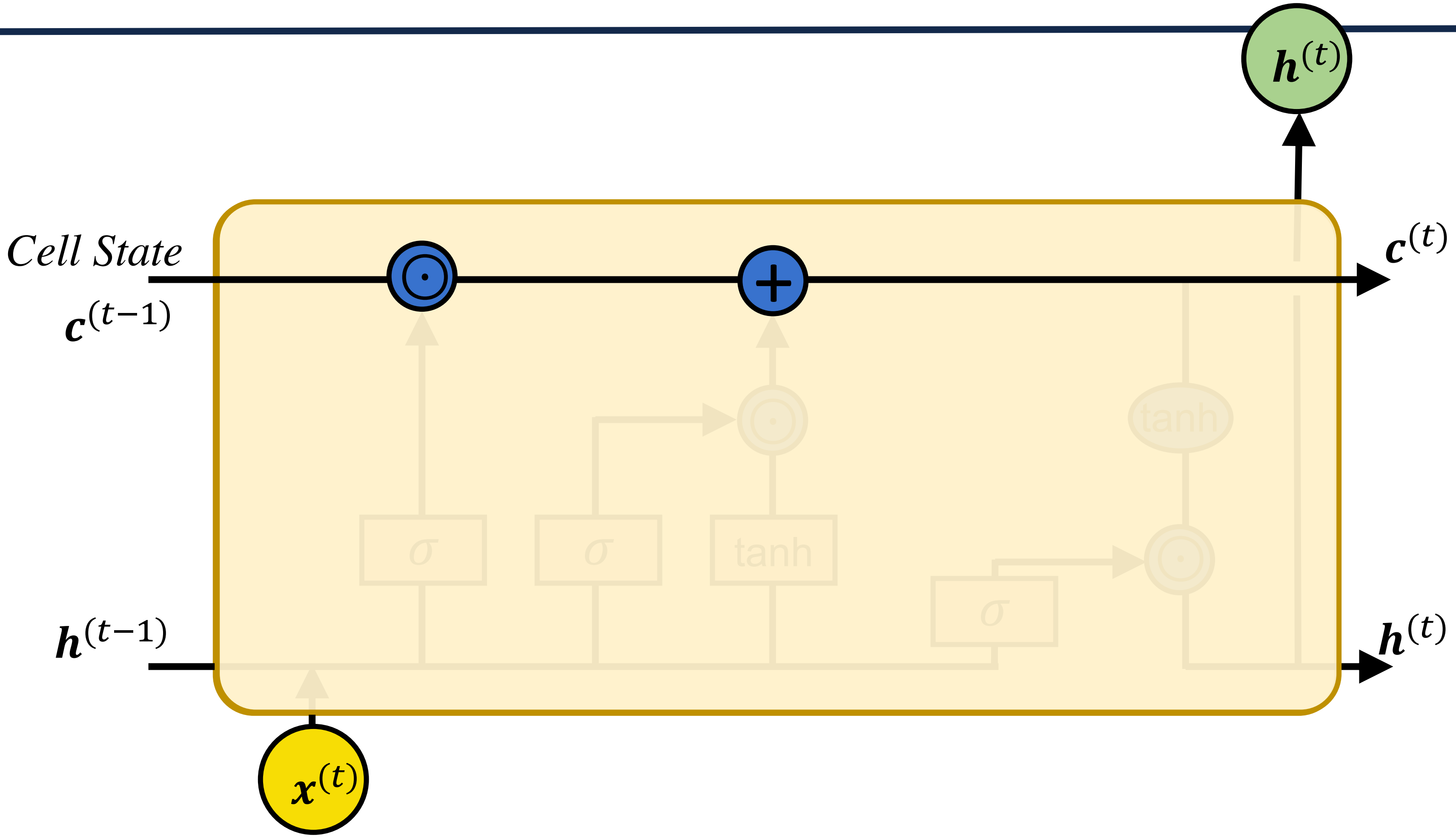


- Standard RNN has a simple computation *cell* from input x to latent state h
- LSTM provides a more sophisticated *cell*

LSTM: Cell Structure

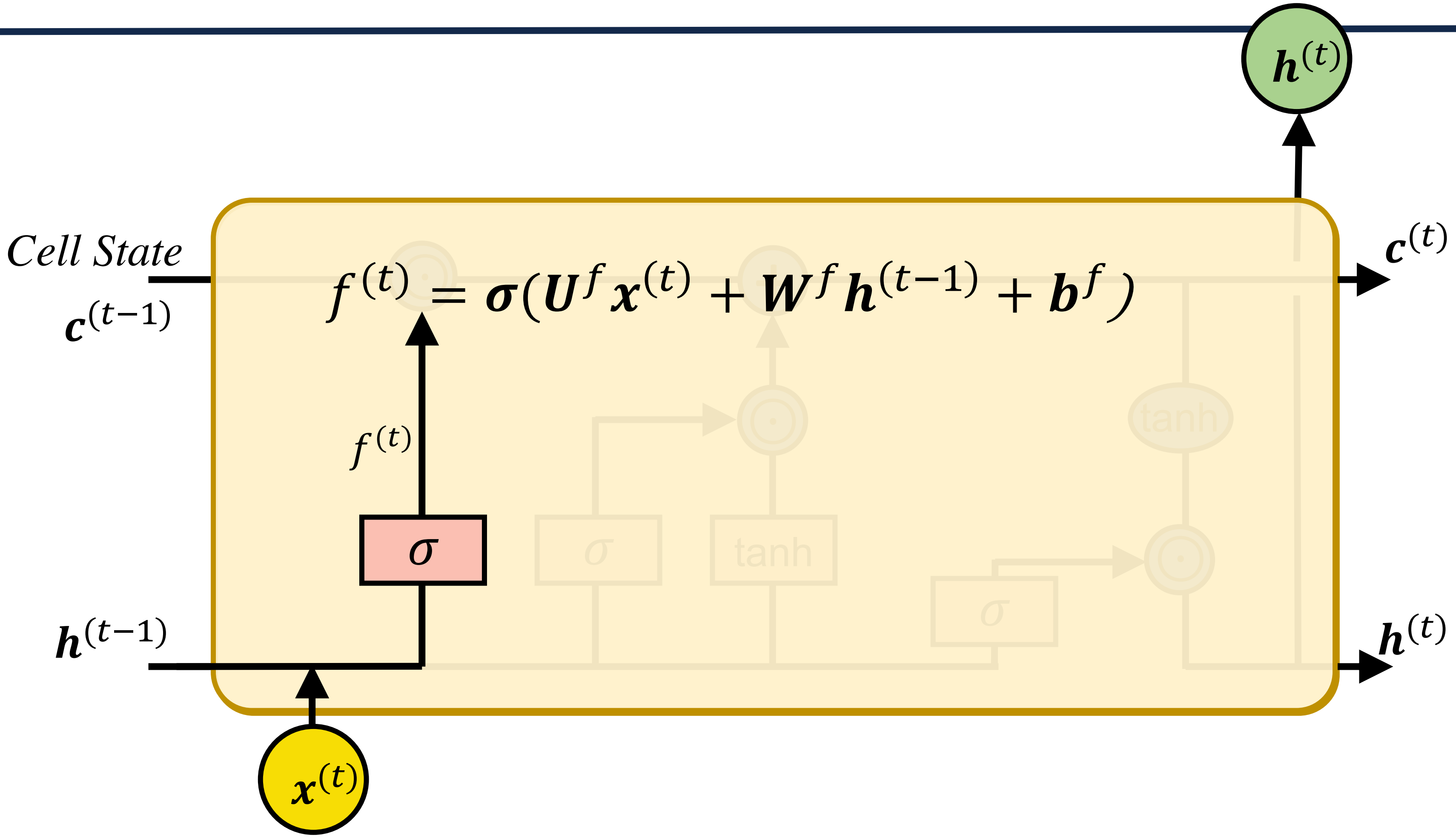


LSTM: Cell state



LSTM: Forget Gate

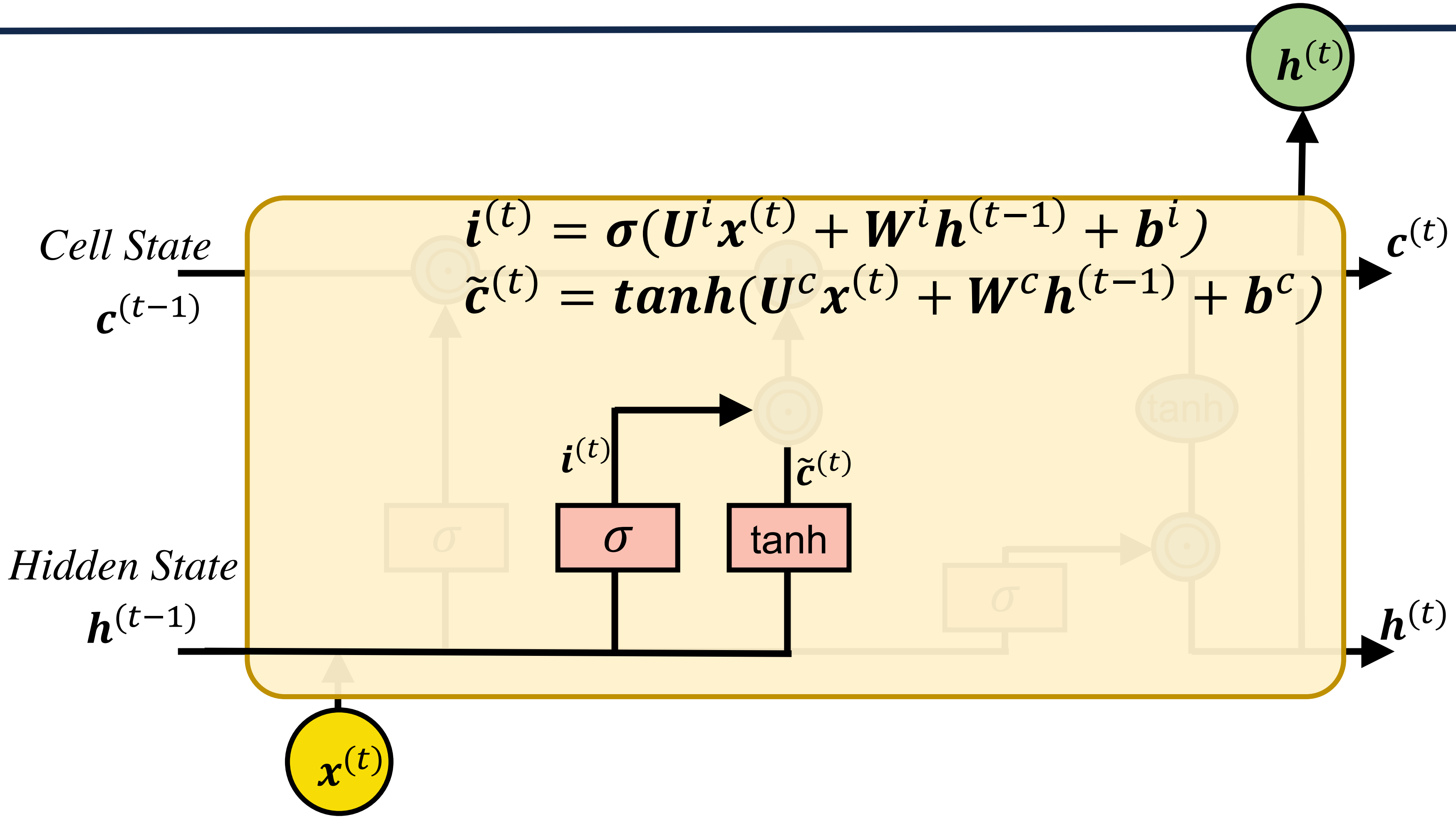
$$f^{(t)} = \boldsymbol{\sigma}(\boldsymbol{U}^f \boldsymbol{x}^{(t)} + \boldsymbol{W}^f \boldsymbol{h}^{(t-1)} + \boldsymbol{b}^f)$$



LSTM: Input Gate

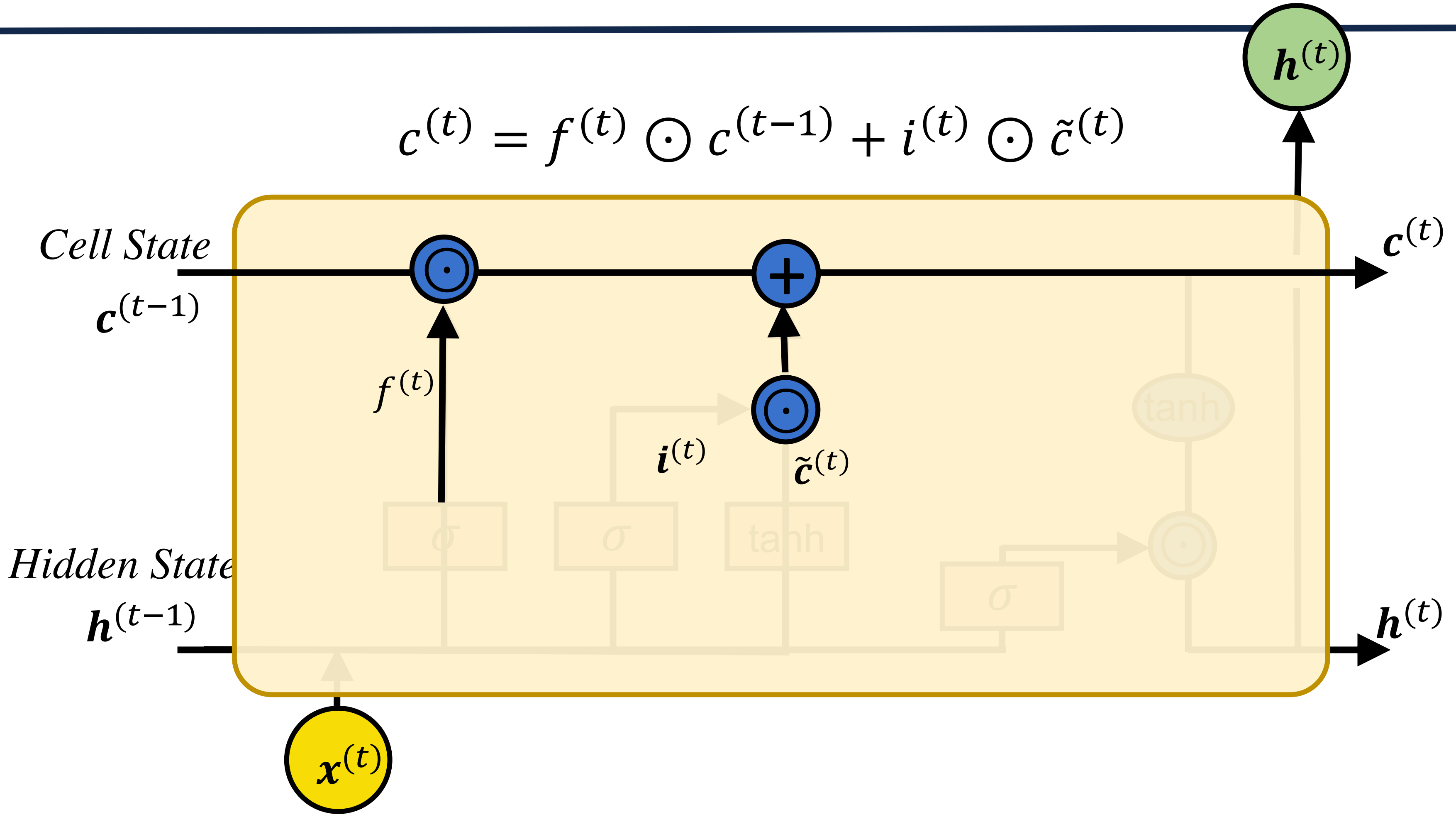
$$\boldsymbol{i}^{(t)} = \boldsymbol{\sigma}(\boldsymbol{U}^{i}\boldsymbol{x}^{(t)} + \boldsymbol{W}^{i}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{i})$$

$$\tilde{\boldsymbol{c}}^{(t)} = \boldsymbol{tanh}(\boldsymbol{U}^{c}\boldsymbol{x}^{(t)} + \boldsymbol{W}^{c}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}^{c})$$

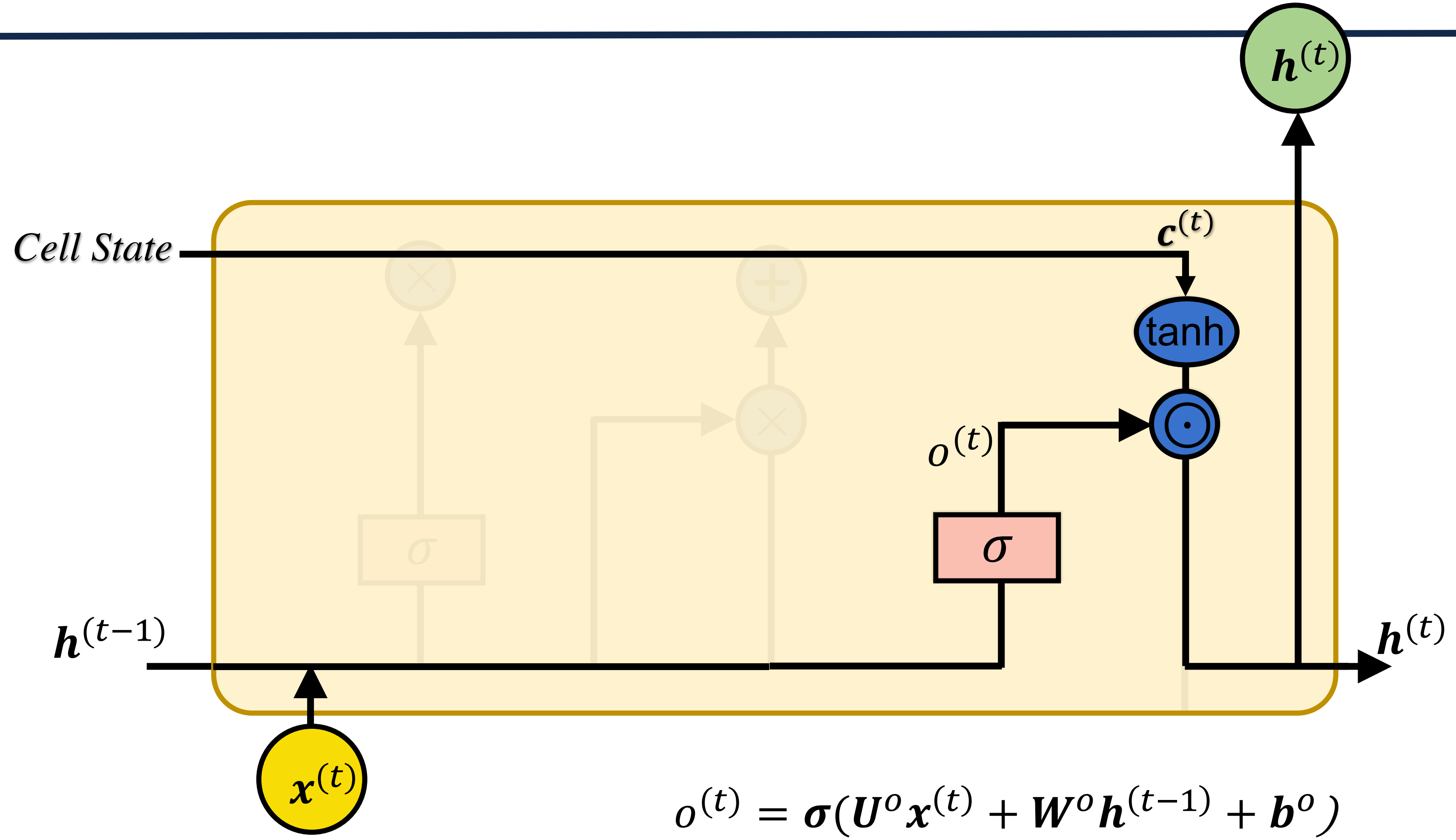


LSTM: Update cell state

$$c^{(t)} = f^{(t)} \odot c^{(t-1)} + i^{(t)} \odot \tilde{c}^{(t)}$$



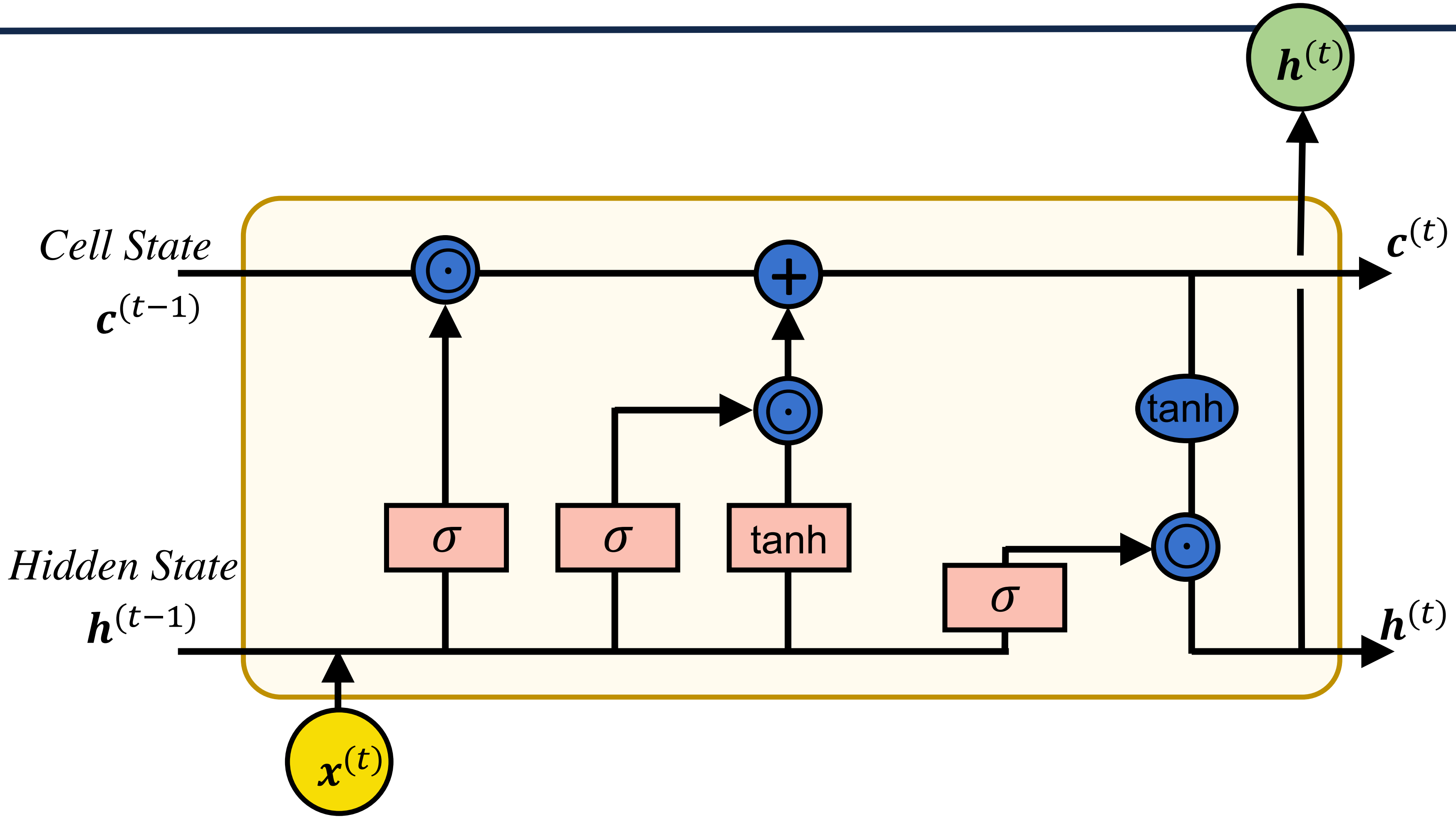
LSTM: Output Gate



$$o^{(t)} = \boldsymbol{\sigma}(\boldsymbol{U}^o \boldsymbol{x}^{(t)} + \boldsymbol{W}^o \boldsymbol{h}^{(t-1)} + \boldsymbol{b}^o)$$
$$\boldsymbol{h}^{(t)} = o^{(t)} \odot \tanh(c^{(t)})$$

ILLINOIS

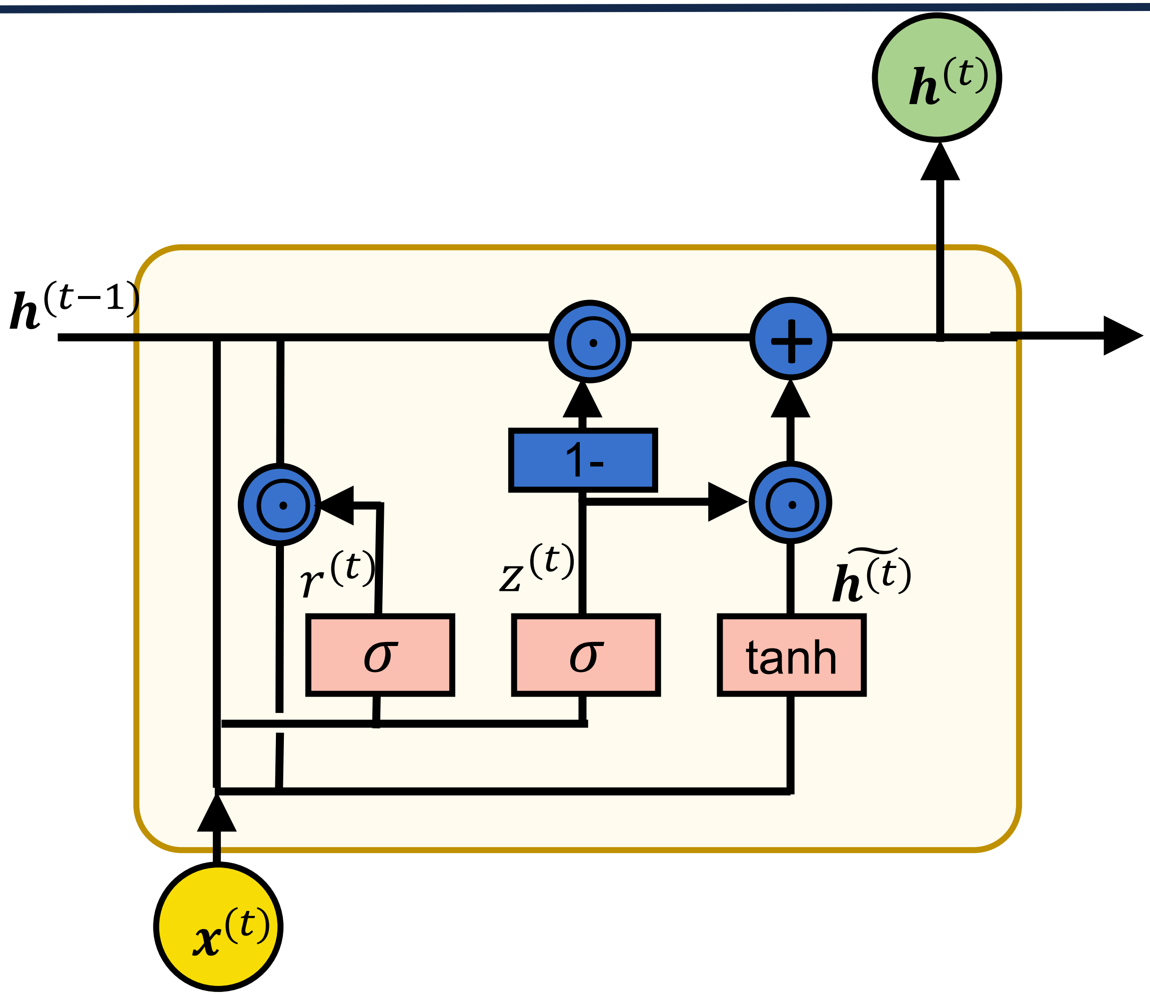
LSTM: Cell Structure



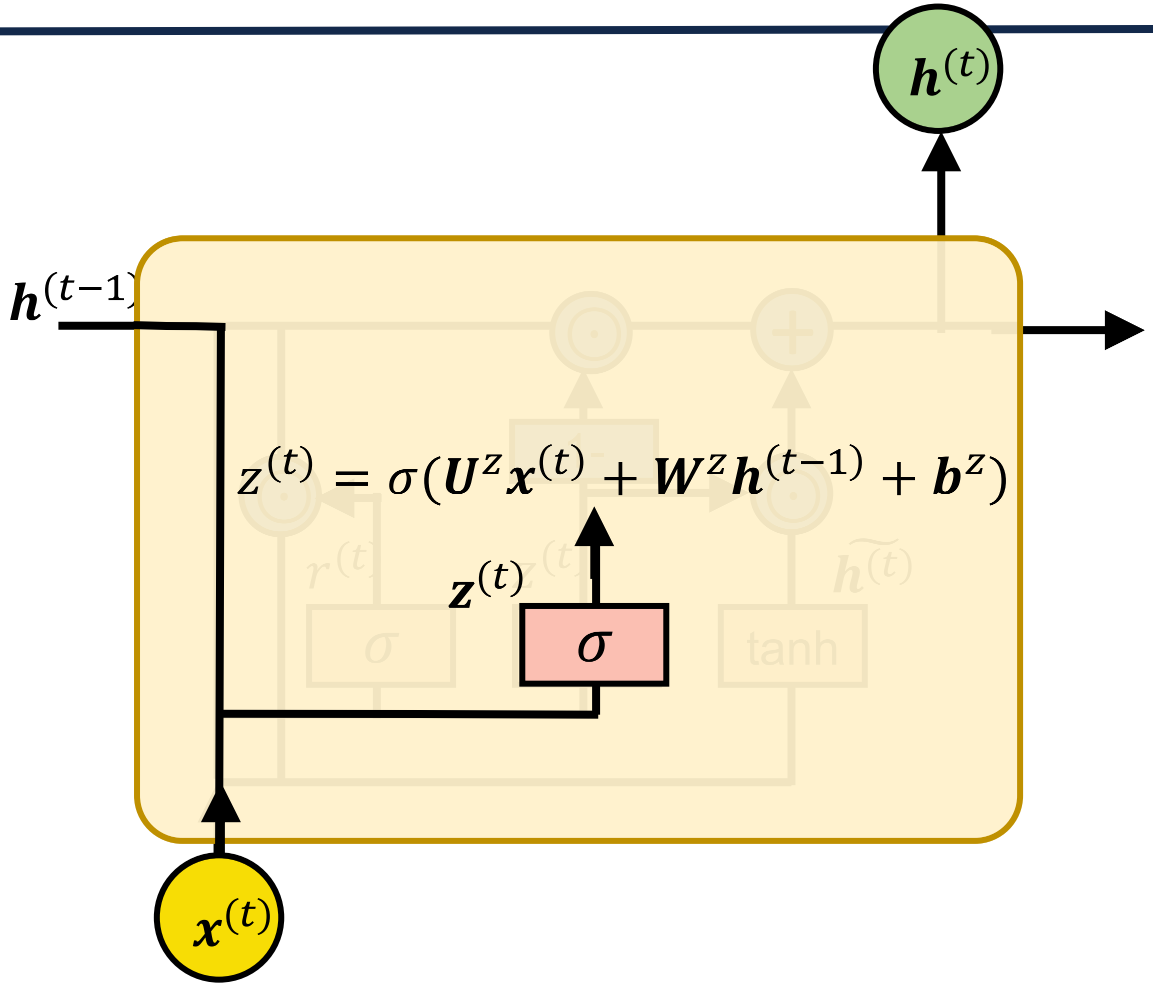
Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

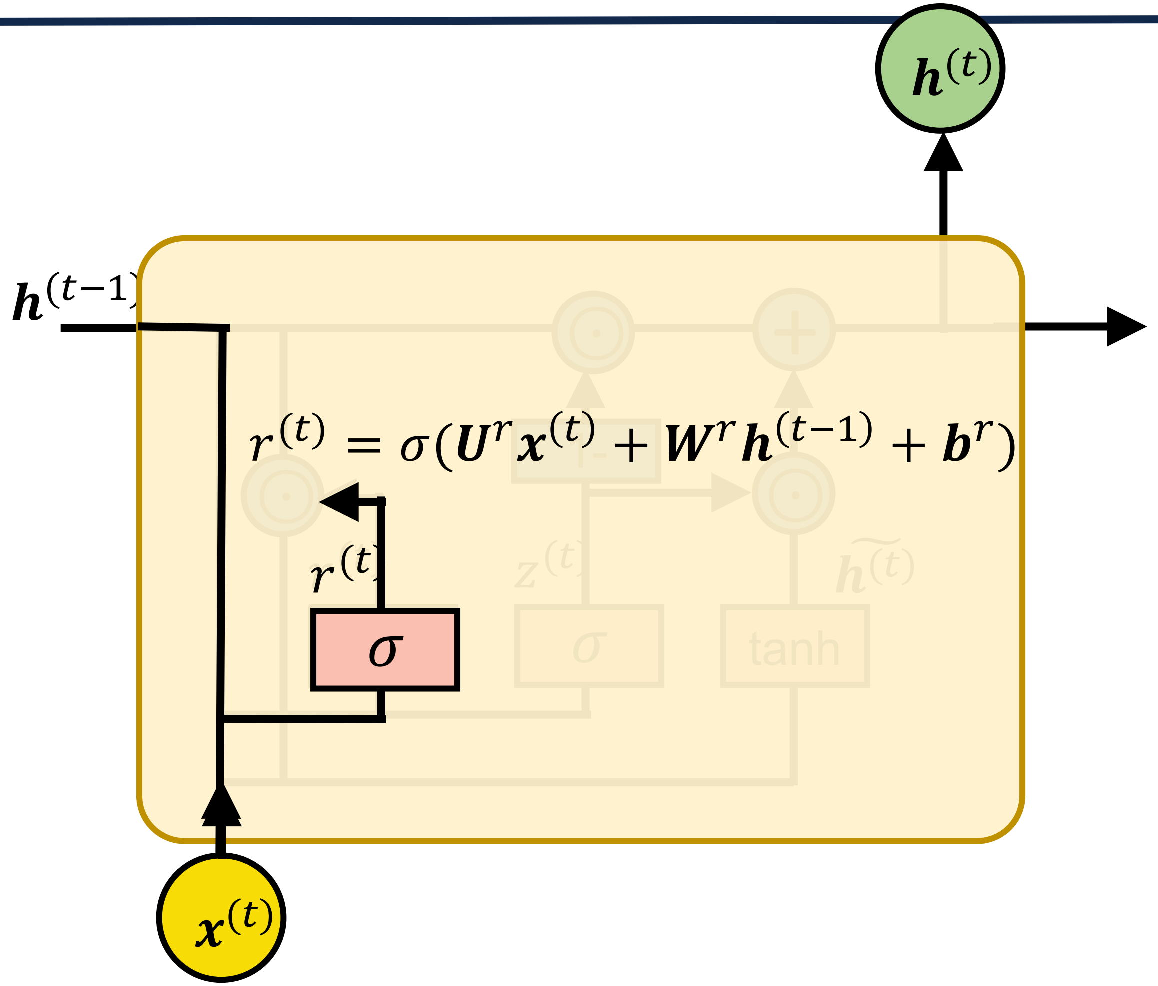
GRU : Gated Recurrent Unit



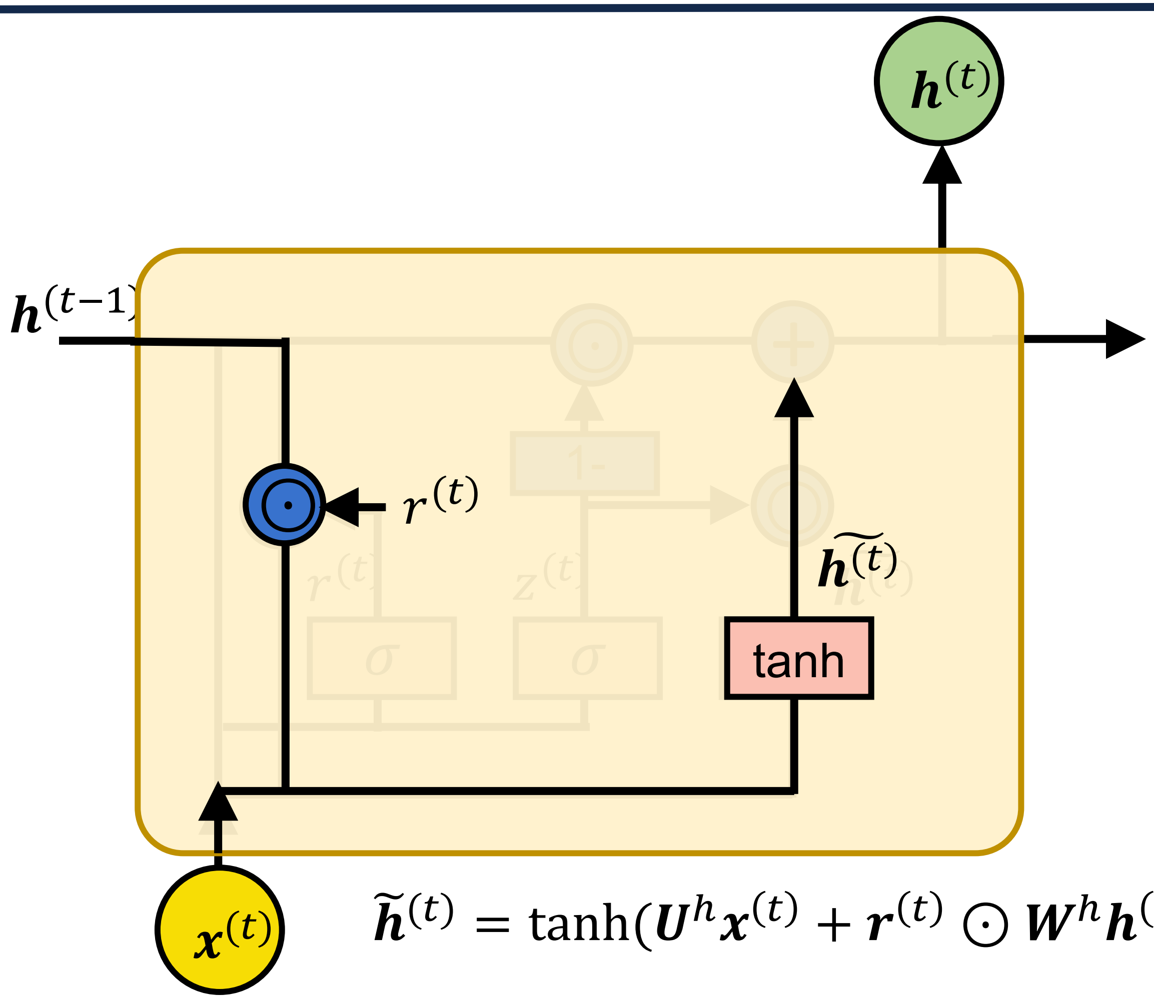
GRU: Update Gate



GRU: Reset Gate



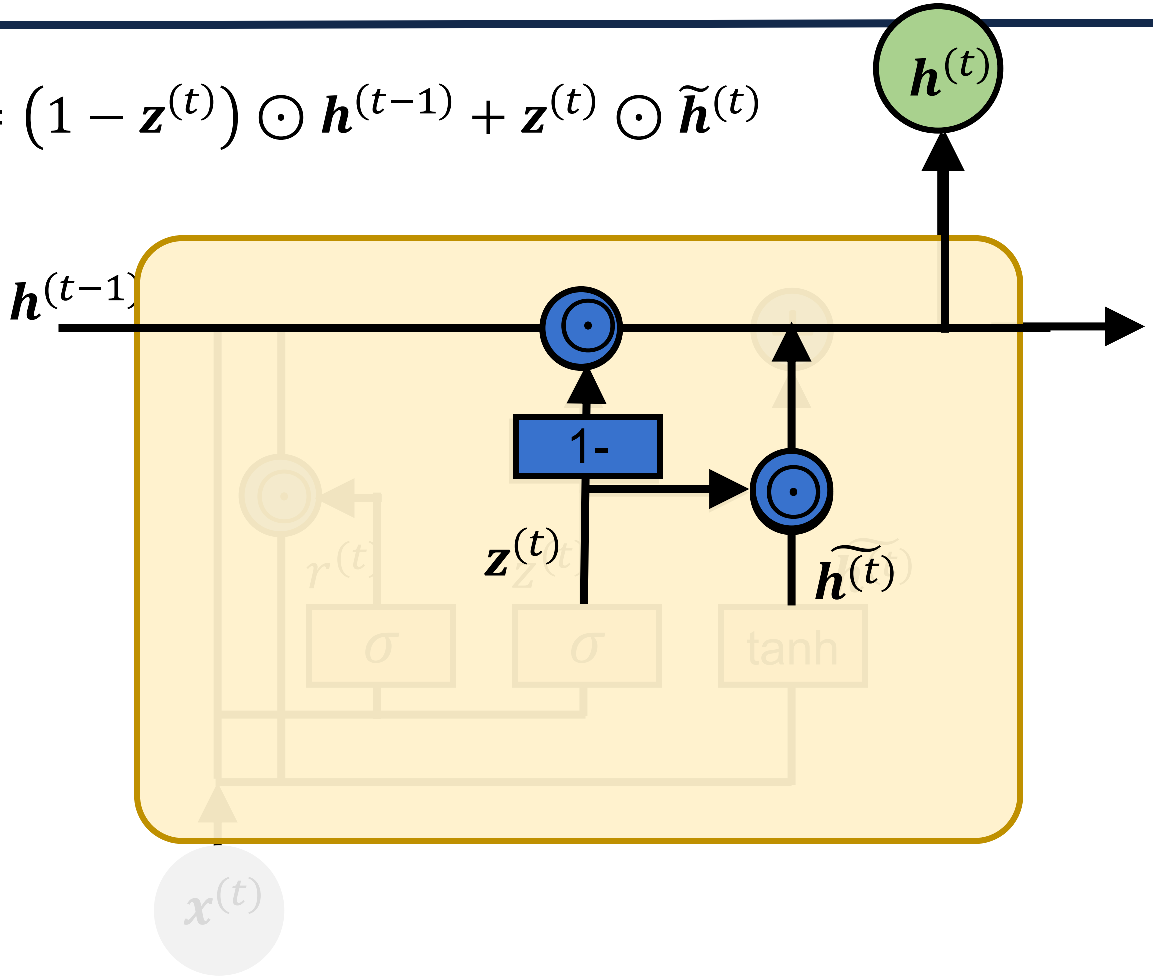
GRU: New information to the hidden state



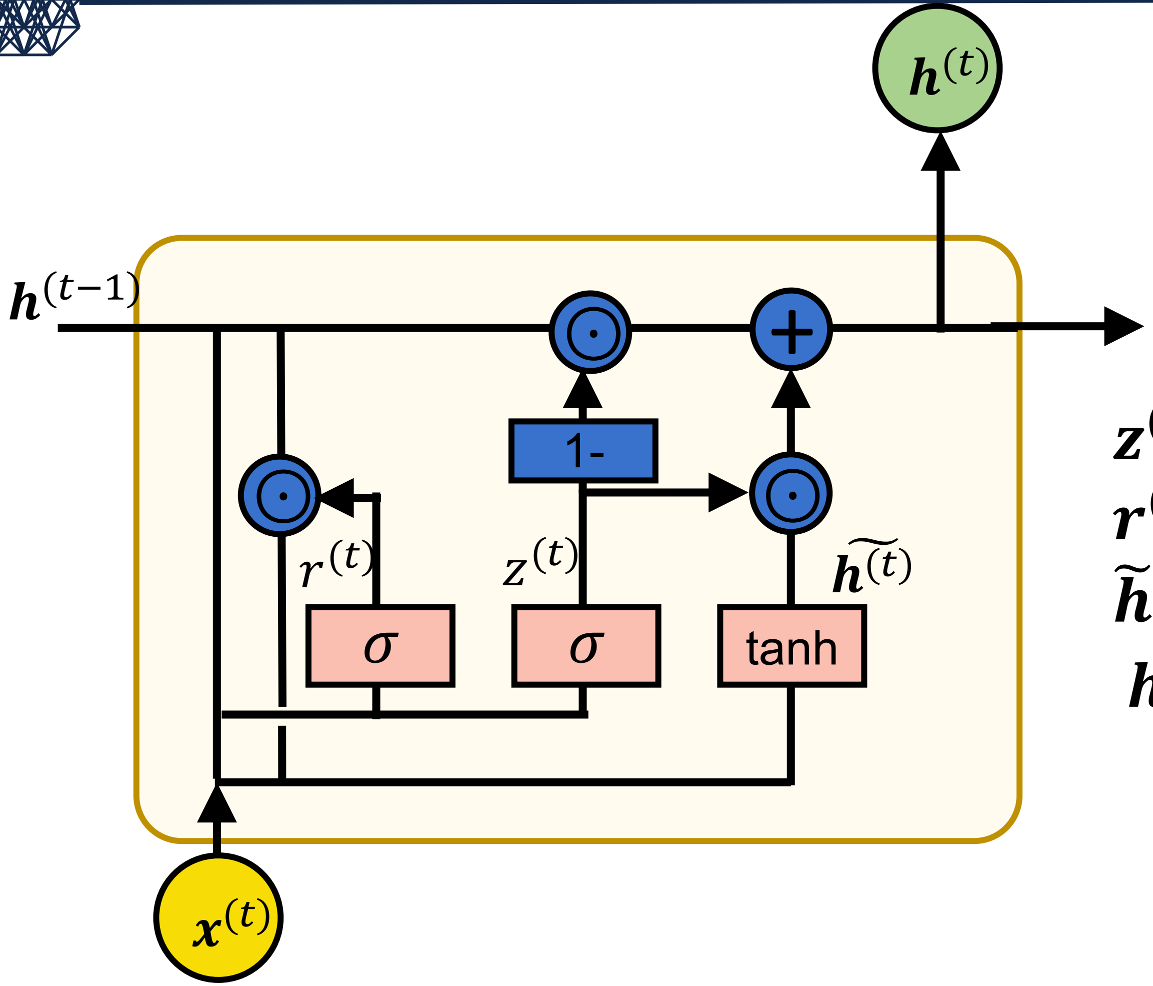
$$\widetilde{\boldsymbol{h}}^{(t)} = \tanh(\boldsymbol{U}^h \boldsymbol{x}^{(t)} + \boldsymbol{r}^{(t)} \odot \boldsymbol{W}^h \boldsymbol{h}^{(t-1)} + \boldsymbol{b}^h)$$

GRU: Final New Hidden State

$$\boldsymbol{h}^{(t)} = \left(1 - \boldsymbol{z}^{(t)}\right) \odot \boldsymbol{h}^{(t-1)} + \boldsymbol{z}^{(t)} \odot \tilde{\boldsymbol{h}}^{(t)}$$



GRU: Summary

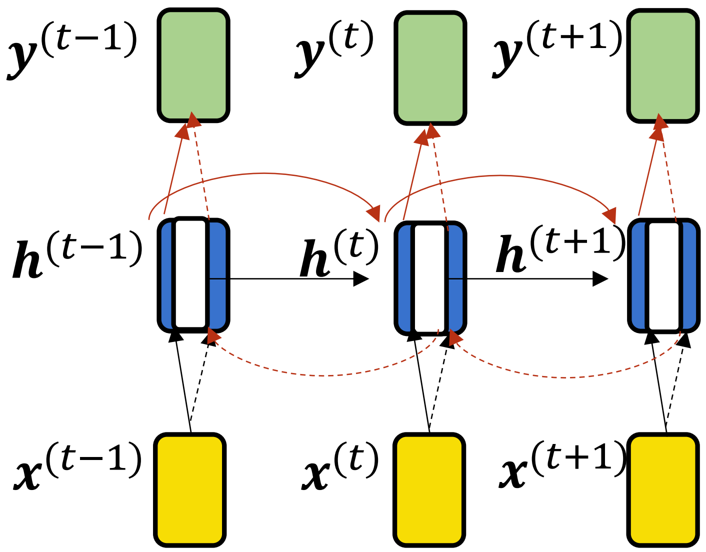


$$z^{(t)} = \sigma(U^z x^{(t)} + W^z h^{(t-1)} + b^z)$$
$$r^{(t)} = \sigma(U^r x^{(t)} + W^r h^{(t-1)} + b^r)$$
$$\widetilde{h}^{(t)} = \tanh(U^h x^{(t)} + r^{(t)} \odot W^h h^{(t-1)} + b^h)$$
$$h^{(t)} = \left(1 - z^{(t)}\right) \odot h^{(t-1)} + z^{(t)} \odot \widetilde{h}^{(t)}$$

Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

Bidirectional RNN



$$\overrightarrow{\boldsymbol{h}}^t = f(\boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t-1)} + \boldsymbol{b}_1)$$

$$\overleftarrow{\boldsymbol{h}}^t = f(\boldsymbol{U}\boldsymbol{x}^{(t)} + \boldsymbol{W}\boldsymbol{h}^{(t+1)} + \boldsymbol{b}_1)$$

$$y^{(t)} = g(\boldsymbol{V}[\overrightarrow{\boldsymbol{h}}^t; \overleftarrow{\boldsymbol{h}}^t] + \boldsymbol{b}_2)$$

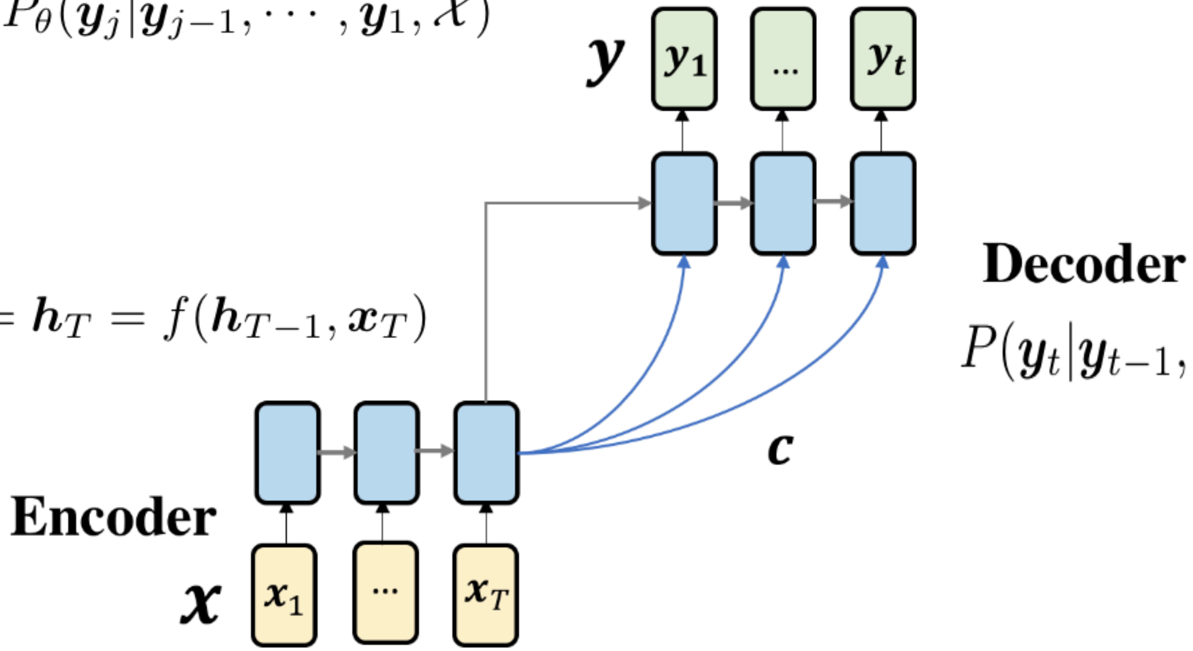
Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

Encoder-Decoder Sequence-to-Sequence Model

$$P_\theta(\mathcal{Y}|\mathcal{X}) = \prod_{j=1}^{J+1} P_\theta(\boldsymbol{y}_j|\boldsymbol{y}_{j-1}, \cdots, \boldsymbol{y}_1, \mathcal{X})$$

$$\boldsymbol{c} = \boldsymbol{h}_T = f(\boldsymbol{h}_{T-1}, \boldsymbol{x}_T)$$



$$P(\boldsymbol{y}_t|\boldsymbol{y}_{t-1}, \cdots, \boldsymbol{y}_1, \boldsymbol{c}) = g(\boldsymbol{h}_t, \boldsymbol{y}_{t-1}, \boldsymbol{c})$$

Agenda

- RNN Basics
- Learning RNN with Backpropagation Through Time (BPTT)
- Long-Short Term Memory Networks (LSTM)
- Gated Recurrent Unit (GRU)
- Bidirectional RNN
- Sequence-to-Sequence RNN
- Healthcare Applications

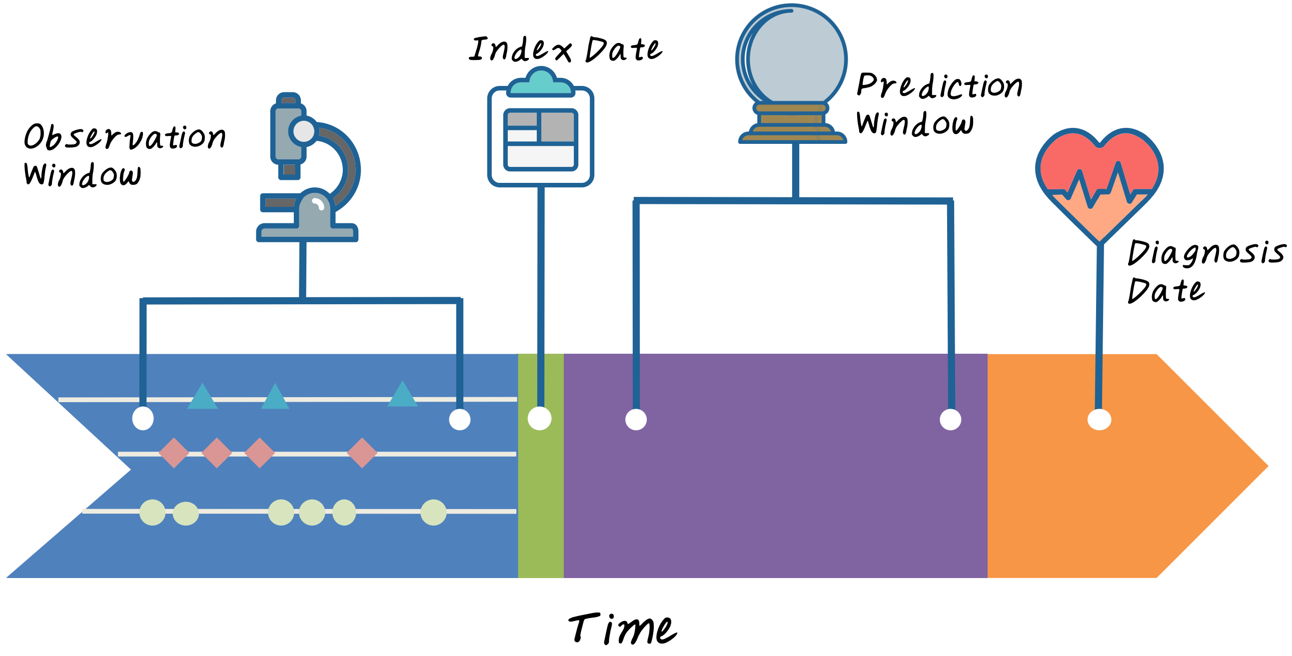
Using Recurrent Neural Network Models For Early Detection Of Heart Failure Onset

How to model temporal relations in EHR

Edward Choi, Andy Schuetz, Walter Stewart, Jimeng Sun. Using Recursive Neural Network Models for Early Detection of Heart Failure Onset, JAMIA 2017

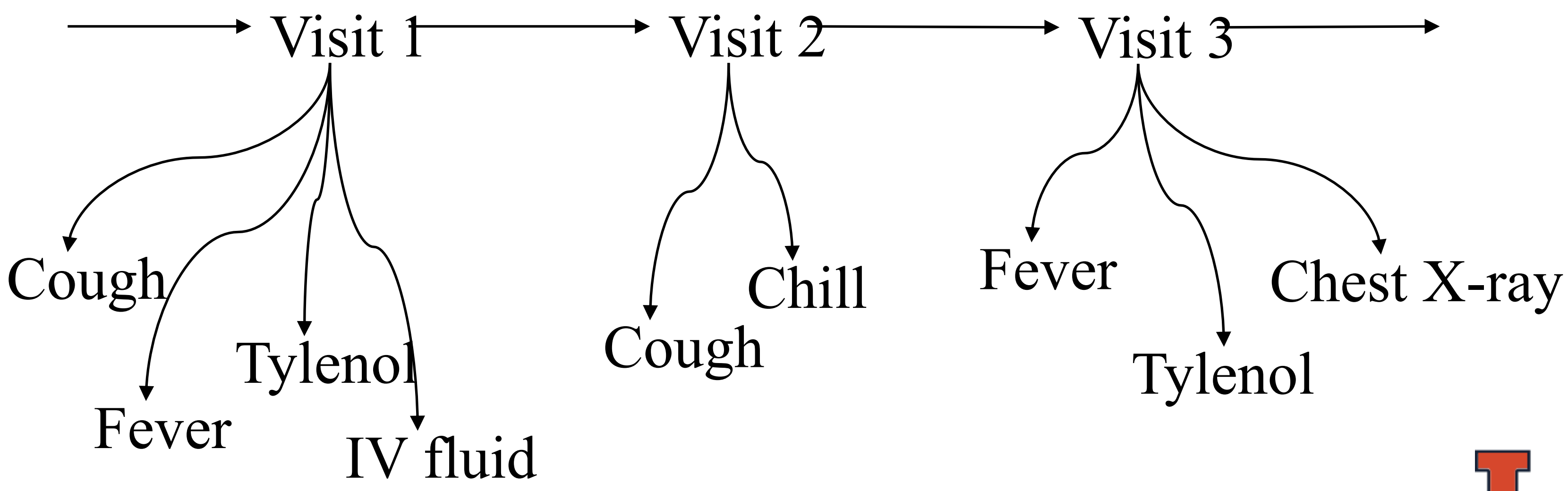
Heart Failure Prediction with RNN

- Given a patient record, predict if the patient will be diagnosed with heart failure in the future



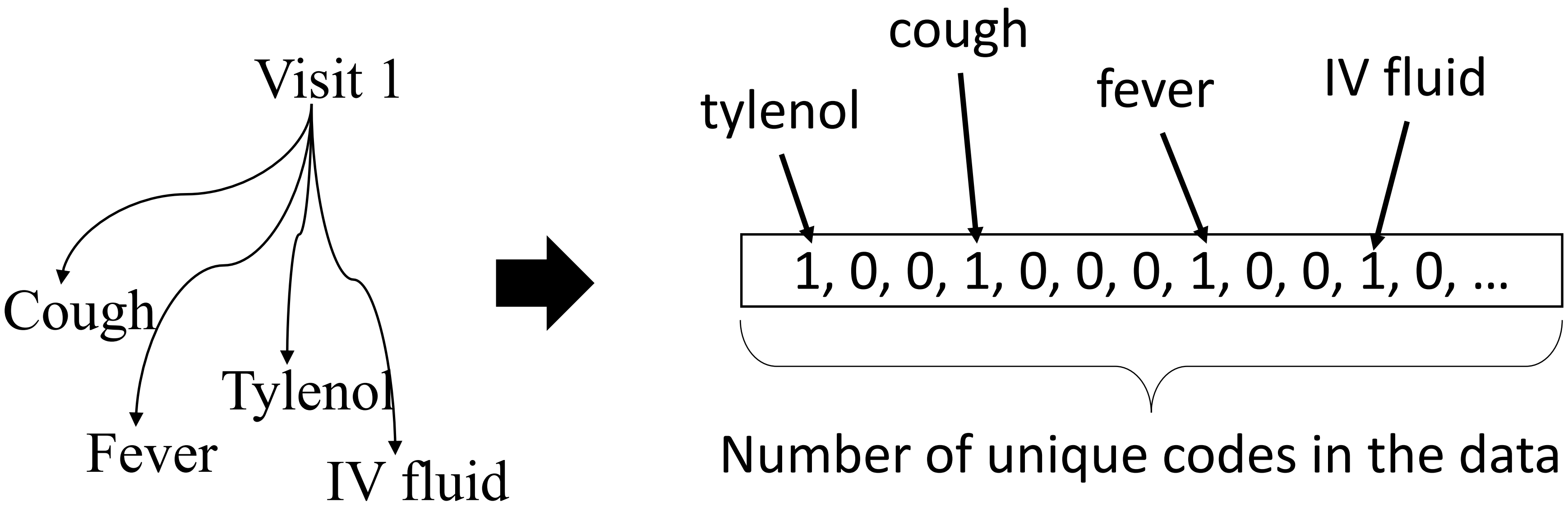
Heart Failure Prediction with RNN

- Input sample **x**
  - Patient record over time
  - Diagnosis codes, medication codes, procedure codes



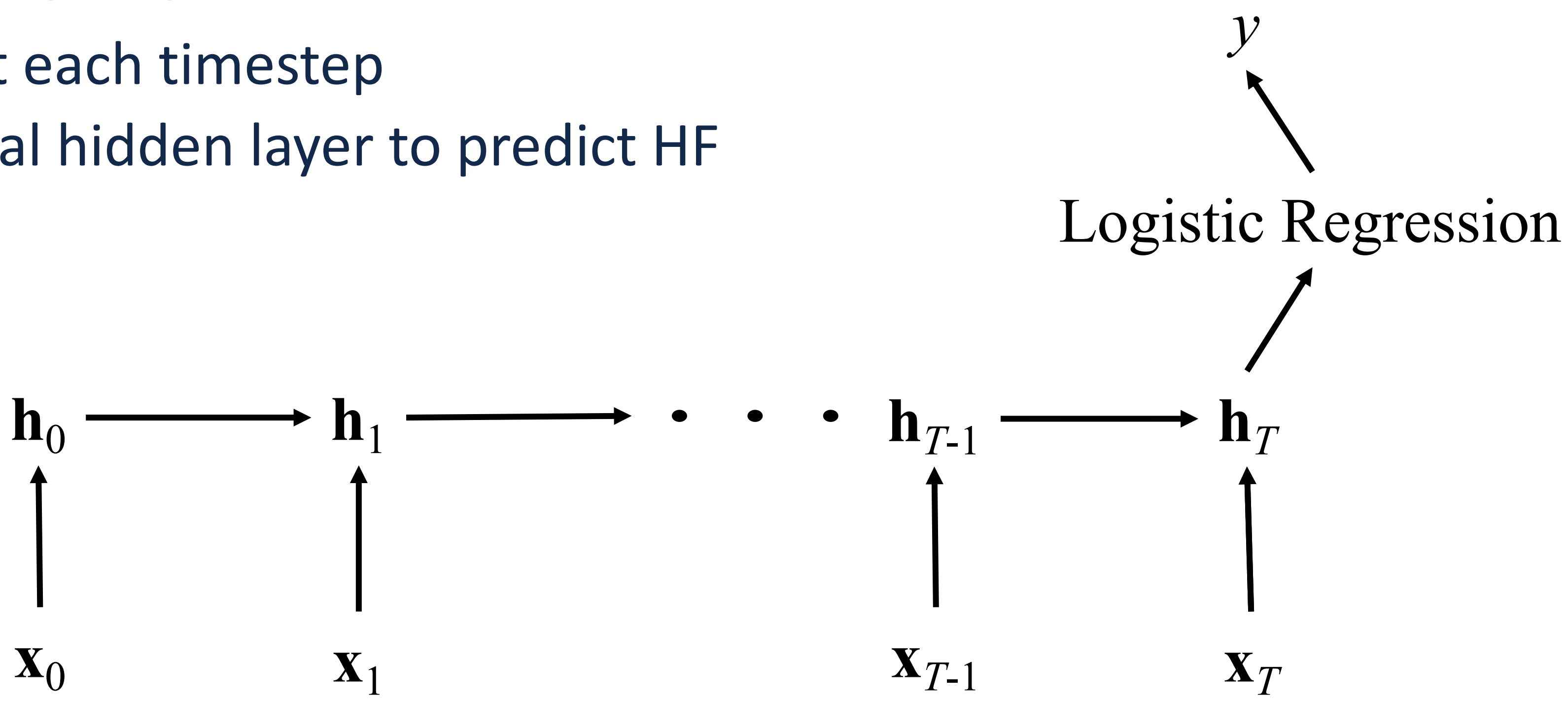
Heart Failure Prediction with RNN

- Input sample **x**
  - Patient record over time
  - Diagnosis codes, medication codes, procedure codes



Heart Failure Prediction with RNN

- Feed visits into the RNN
  - One visit at each timestep
  - Use the final hidden layer to predict HF



Heart Failure Prediction with RNN

Data

34K patients from Sutter Health

4K cases, 30K controls

18-months observation window

Case-control selection criteria

Age (40-85)

Types of diagnoses received

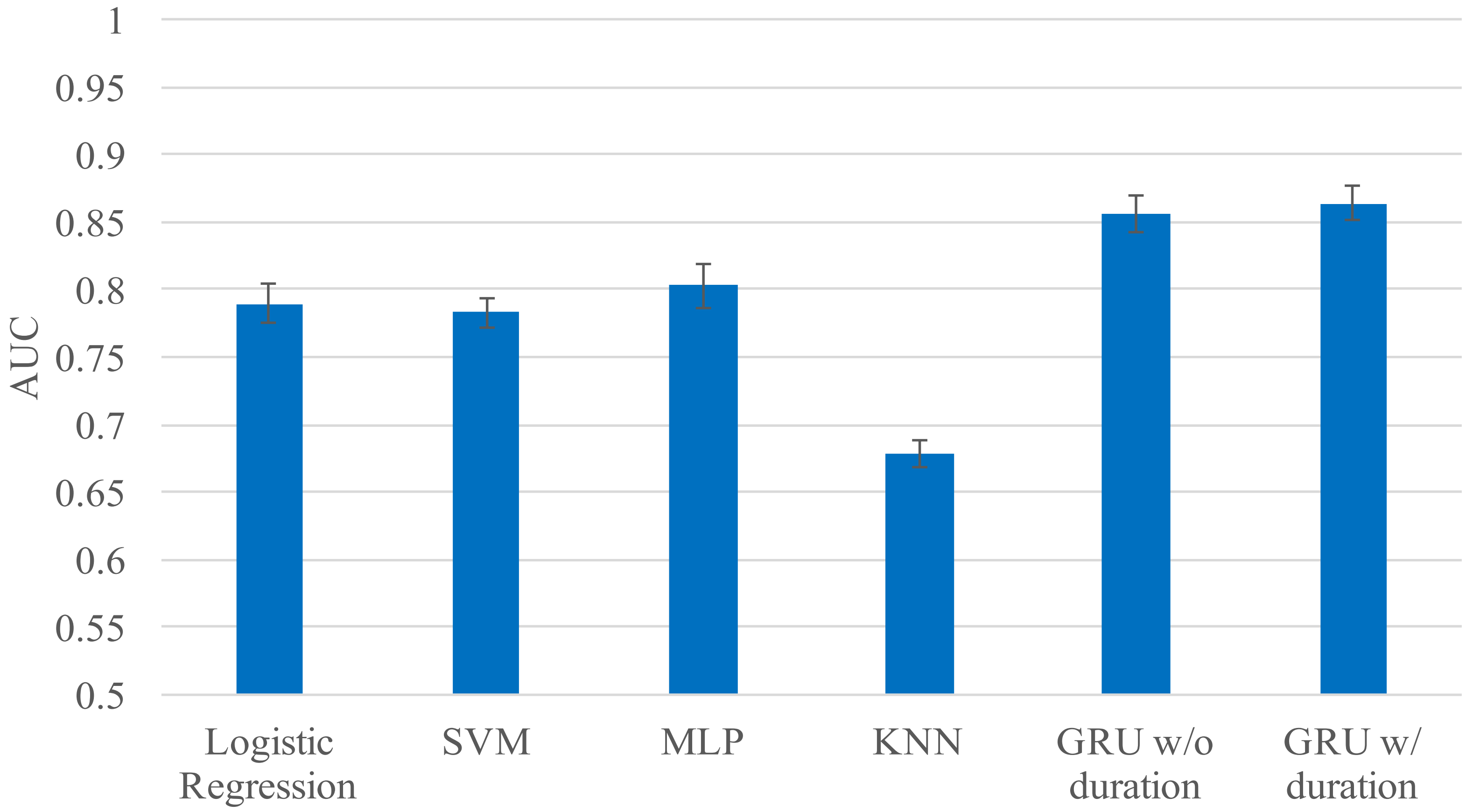
Number of hospital visits

Time span between diagnoses

ILLINOIS

Heart Failure Prediction with RNN

- Prediction performance



Doctor AI: Predicting Clinical Events via Recurrent Neural Networks

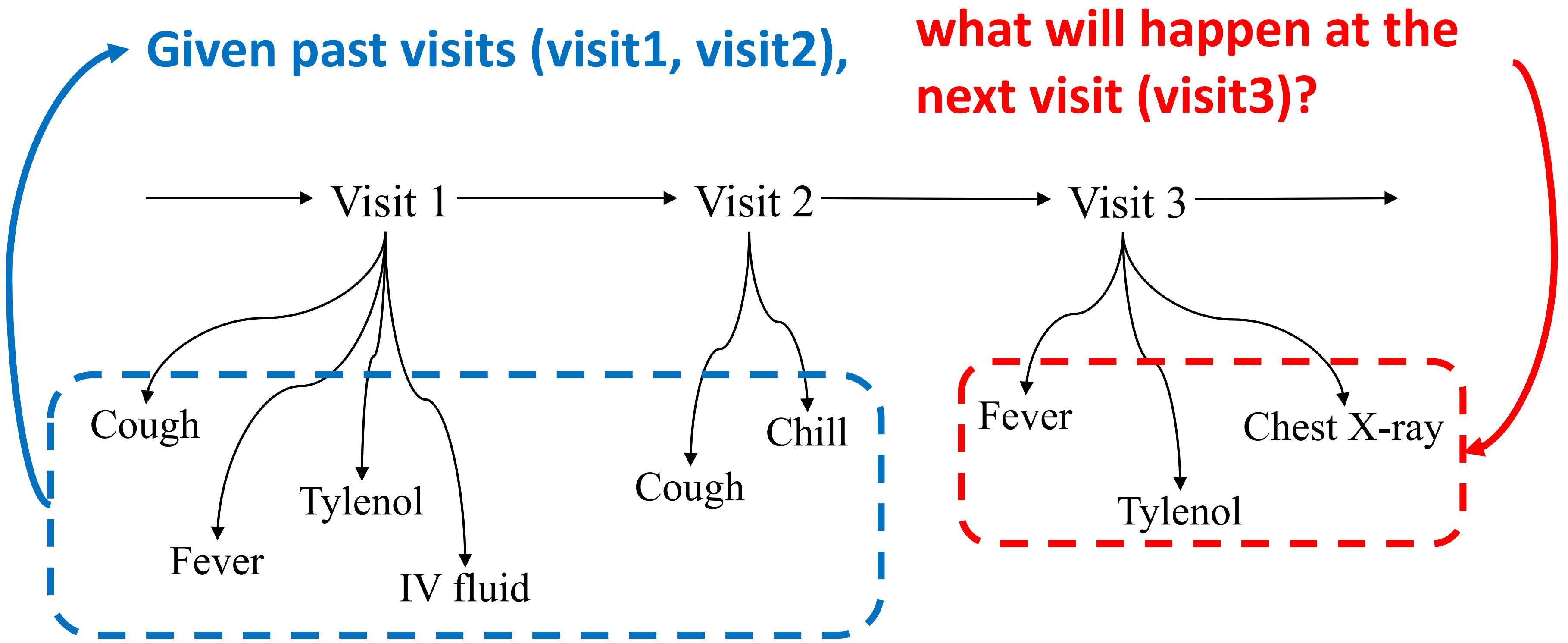
Edward Choi, Mohammad Taha Bahadori, Andy Schuetz, Walter F. Stewart, Jimeng Sun

Machine Learning for Healthcare Conference, 2016

ILLINOIS

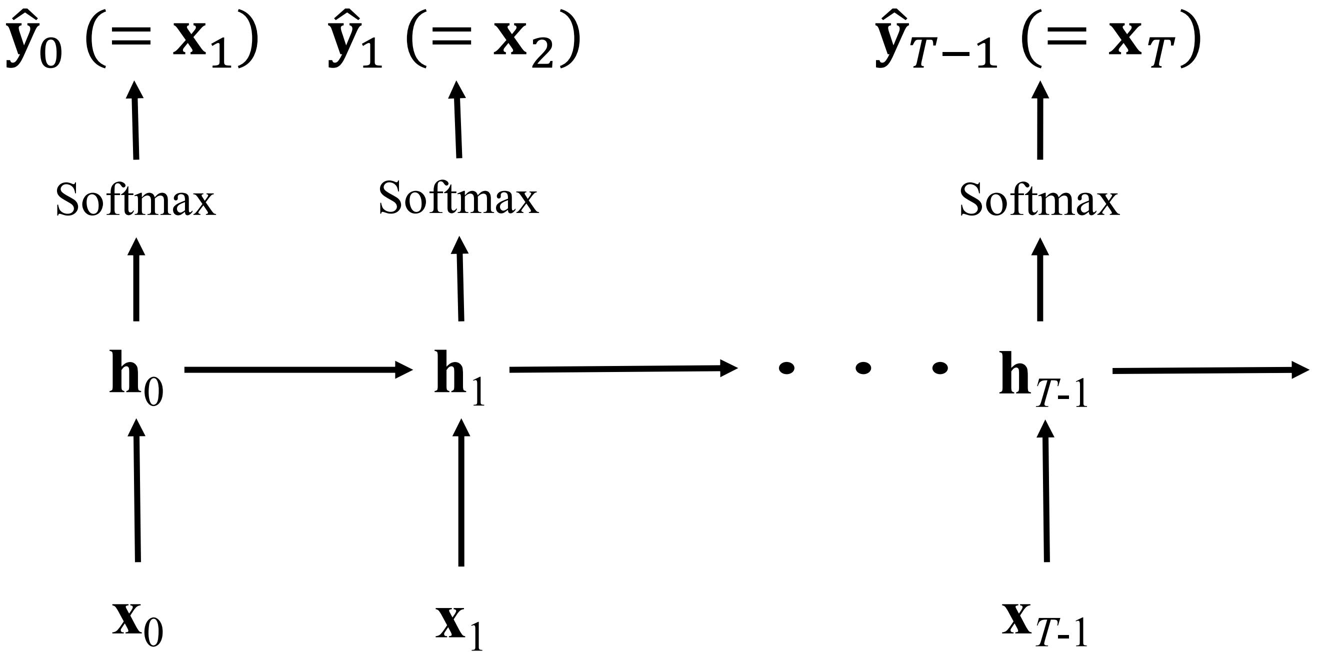
Doctor AI: Background

- Disease progression modeling



Doctor AI: Model

- Feed visits into the RNN
  - One visit at each timestep.
  - Predict next events at each timestep.

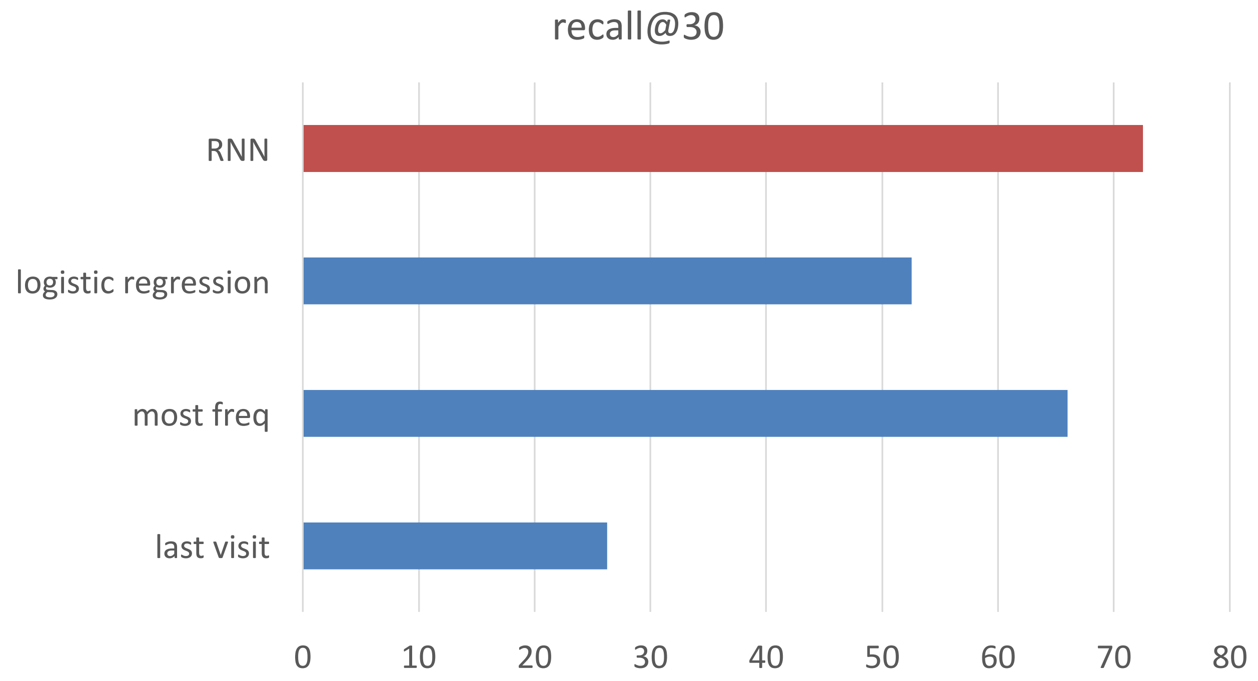


Doctor AI: Data

- 260K patients from Sutter Health
- Patient records over 10 years
- Input codes
  - Diagnosis codes, medication codes, procedure codes (38,000 codes)
- Output labels
  - 1,183 diagnosis codes

Doctor AI: Sequential Prediction

- Predicting diagnoses in the next visit

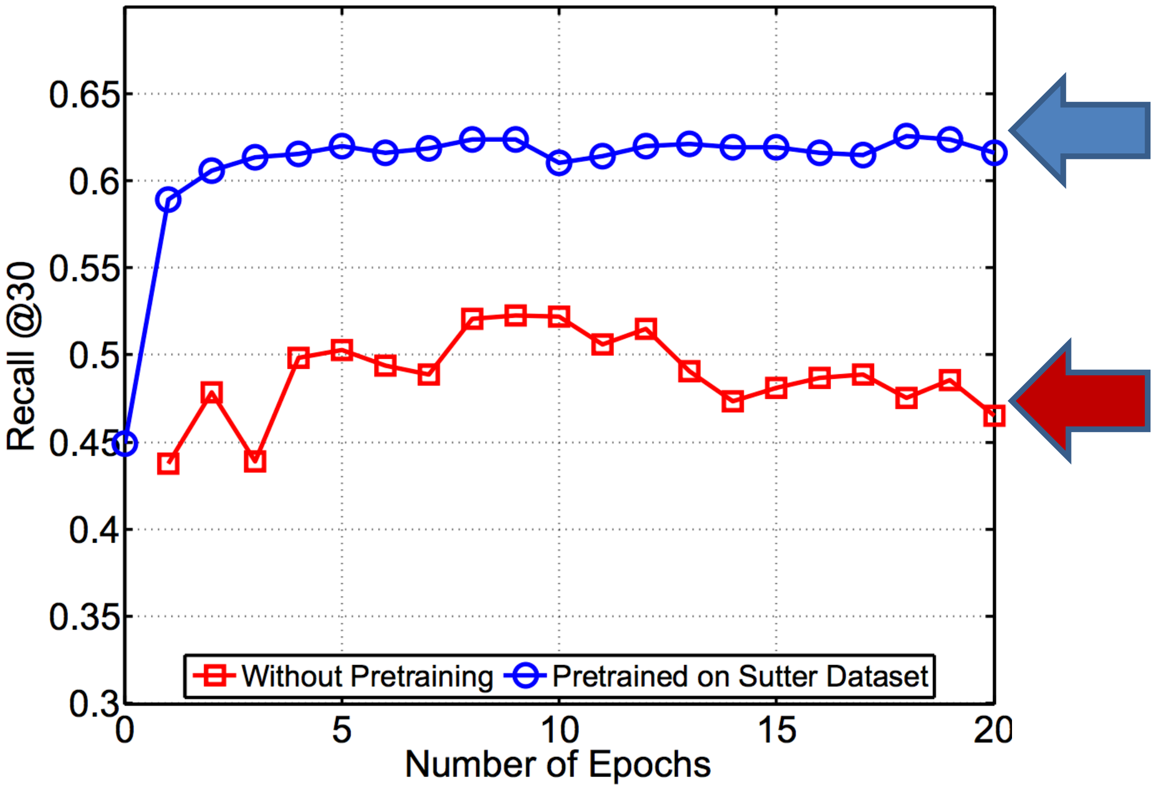


$$\text{top-}k\text{ recall} = \frac{\#\text{ of true positives in the top }k\text{ predictions}}{\#\text{ of true positives}}$$

ILLINOIS

Doctor AI: Knowledge Transfer

- Generalize RNN model from one hospital to another



Perform better when warm start the existing model

Perform worse when cold-start from random initialization