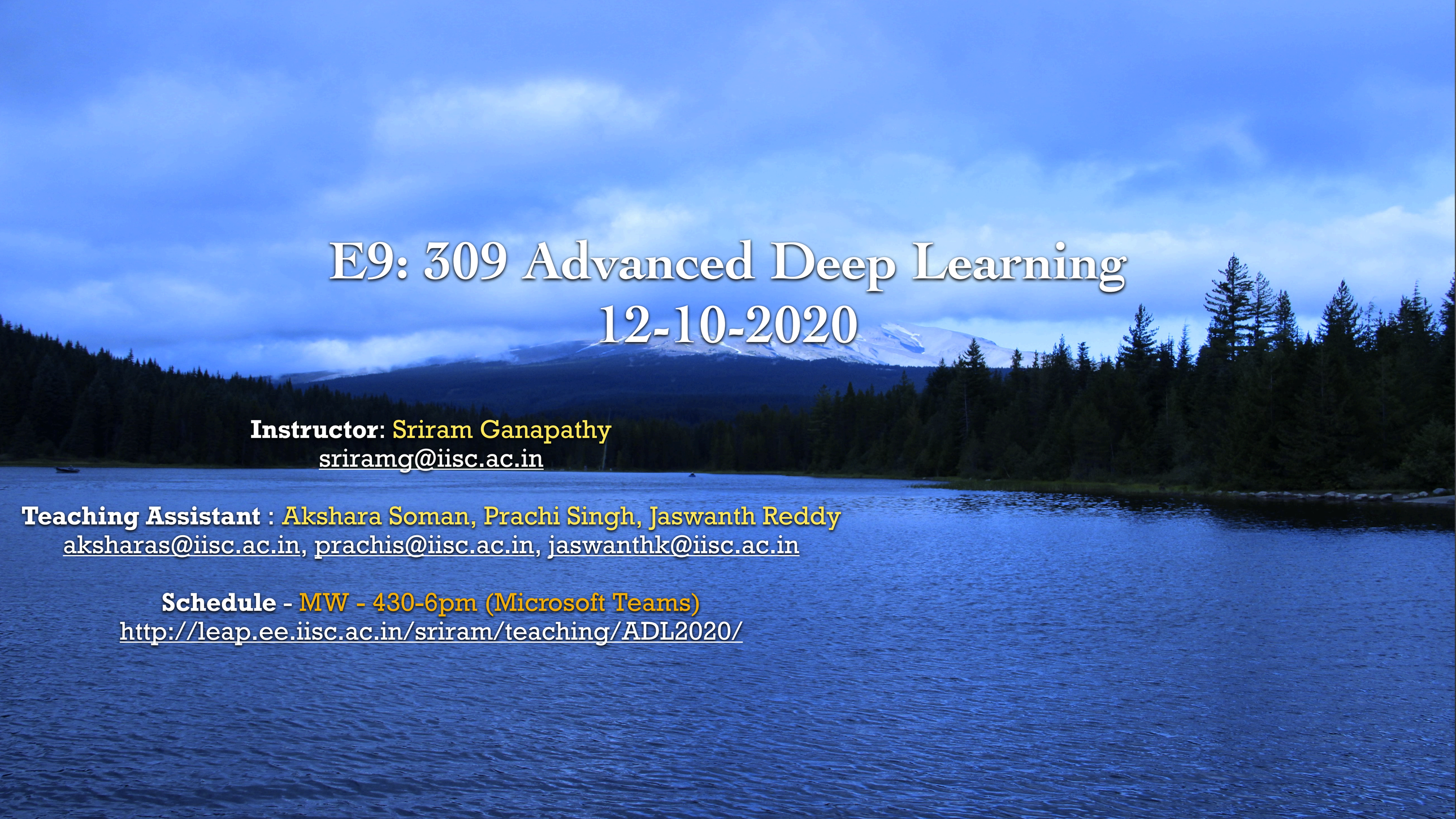


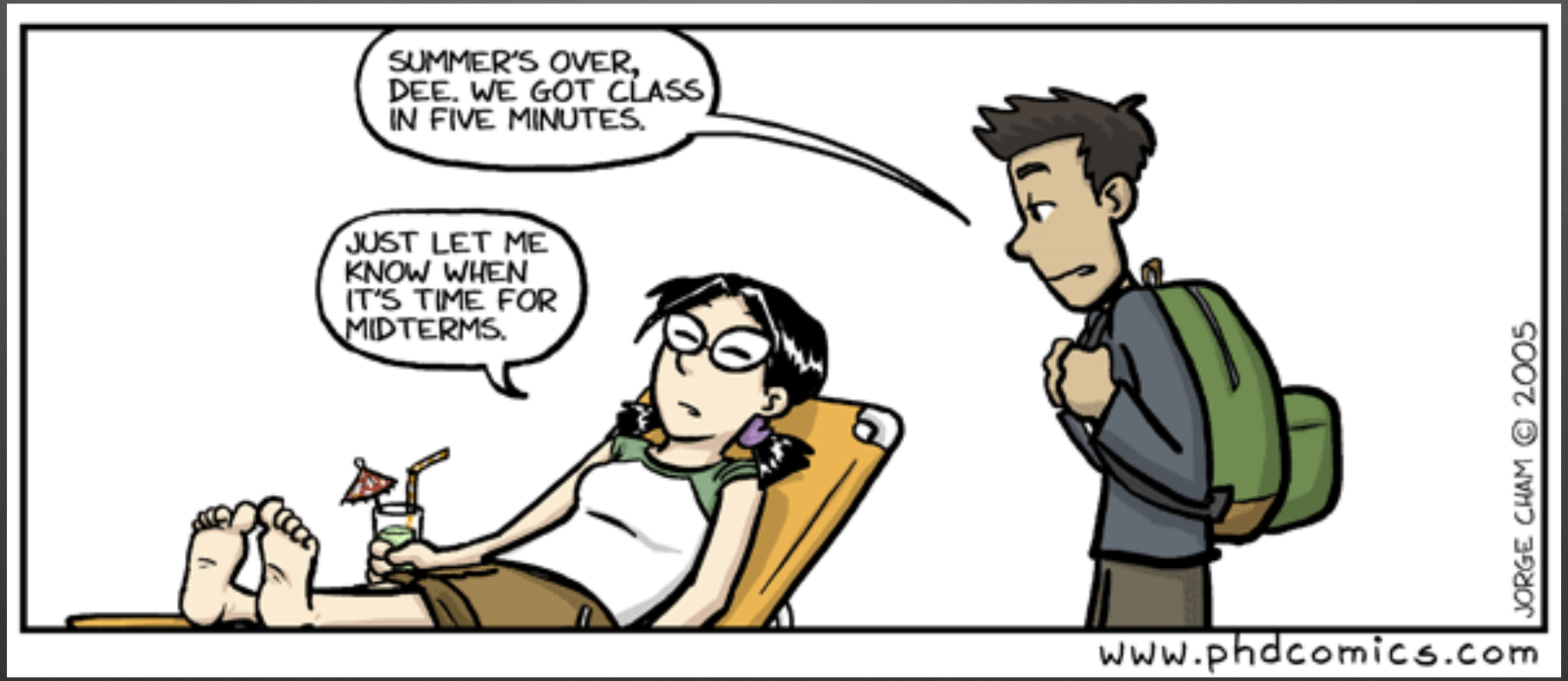
E9: 309 Advanced Deep Learning
12-10-2020

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/

The bell has rung



http://phdcomics.com/

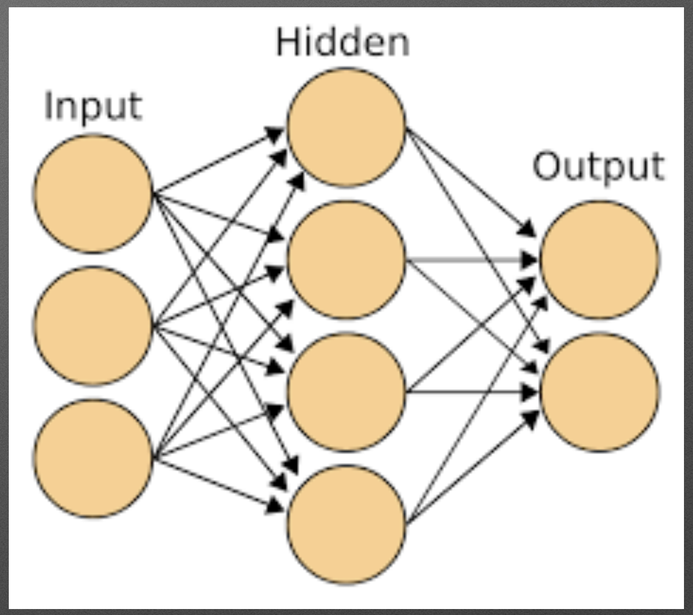
Recap of deep learning

Housekeeping

- Filling the google form in the webpage
  - ➡ Contents will be made available to the folks in the 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?)

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.
- Θ - collection of learnable parameters in the model.
- $E(\boldsymbol{y}, \hat{\boldsymbol{y}})$ - error function used in the model training.

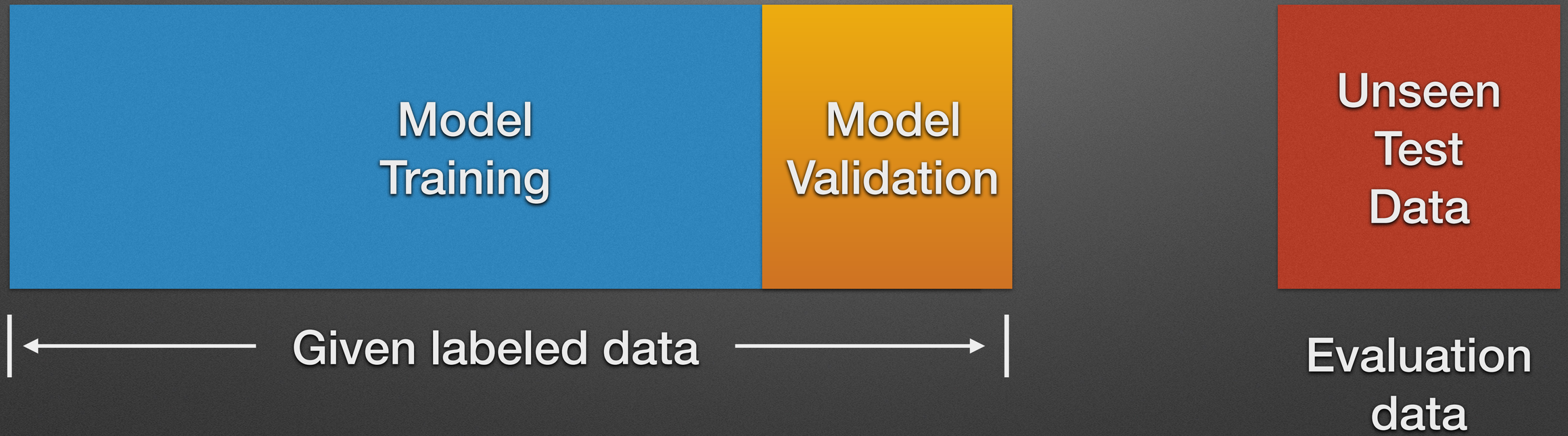


Some notations

- $\{\boldsymbol{x}_1, ..., \boldsymbol{x}_N, \boldsymbol{y}_1, ..., \boldsymbol{y}_N\}$ - 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.

Premise

- Training data, validation data, test data.



- Model training data - used for parameter learning.
- Validation data - used for hyper-parameter tuning (cross validation CV).

Last lecture

- Feedforward networks
- Convolutional networks
- Learning in deep networks
  - Parameters, hyper-parameters etc
  - SGD learning rule
  - Adam optimization, batch normalization.
- Reading assignments.

Module - I Visual and Time Series Modeling

Why do need recurrent models

- Learn from ordered pairs of x, y
  - ✓ All the data samples are treated independently.
    - ⋆ Data are shuffled before mini-batch formation
- If the input data and output labels are time-series data $x(t), y(t)$
  - DNNs/CNNs may fail to model the correlation of the data across the time
  - Question - how can we build models that capture the time evolution of the data and the labels.

Why do need recurrent models

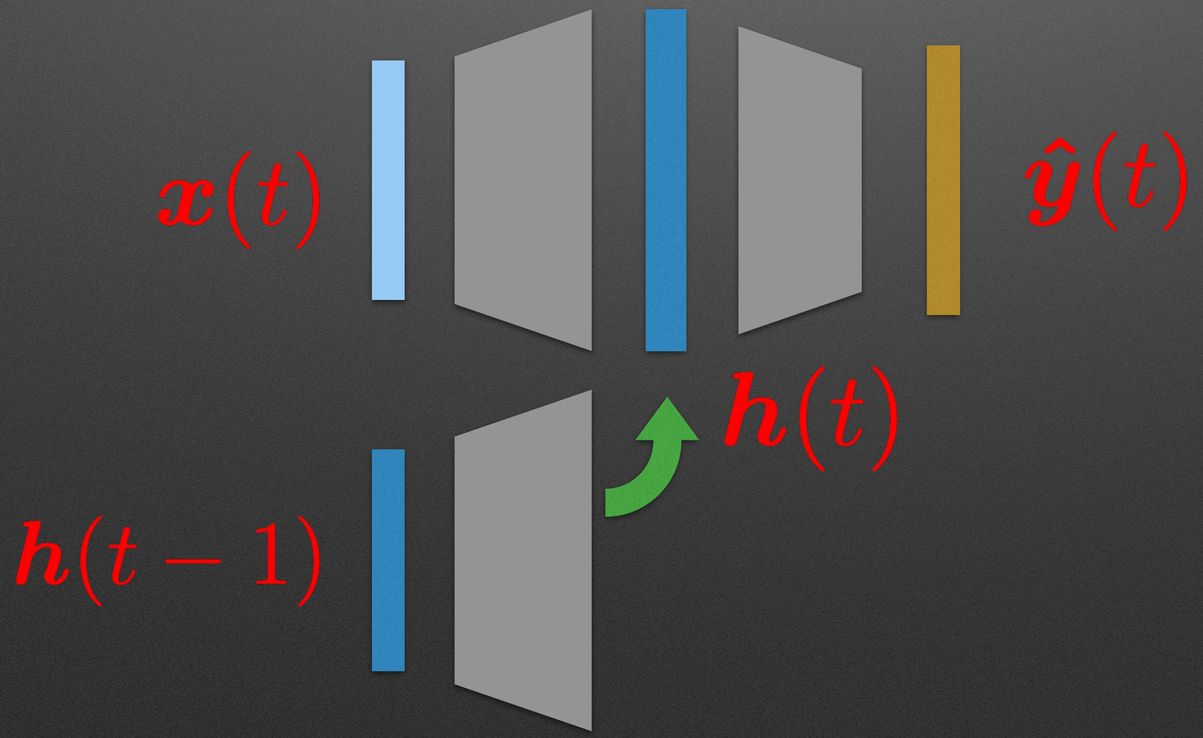
- An interesting subset of this problem is where the input alone is a time series $\boldsymbol{x}(t), \boldsymbol{y}$ or have different indices $\boldsymbol{x}(t), \boldsymbol{y}(u)$

- Examples
  - ✓ Text sequences
  - ✓ Speech and audio
  - ✓ Video sequences

First order recurrence - hidden layer

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

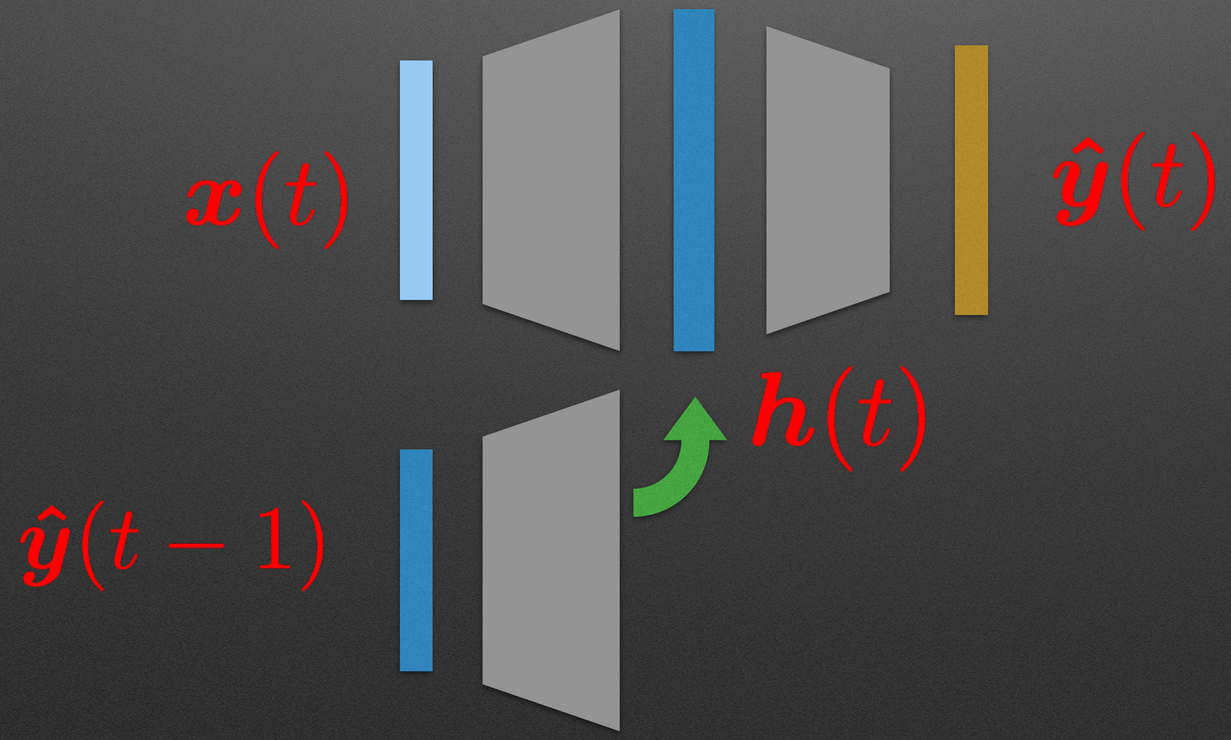
$$\boldsymbol{h}(t) = f\big(\boldsymbol{h}(t-1), \boldsymbol{x}(t)\big)$$



First order recurrence - output layer

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

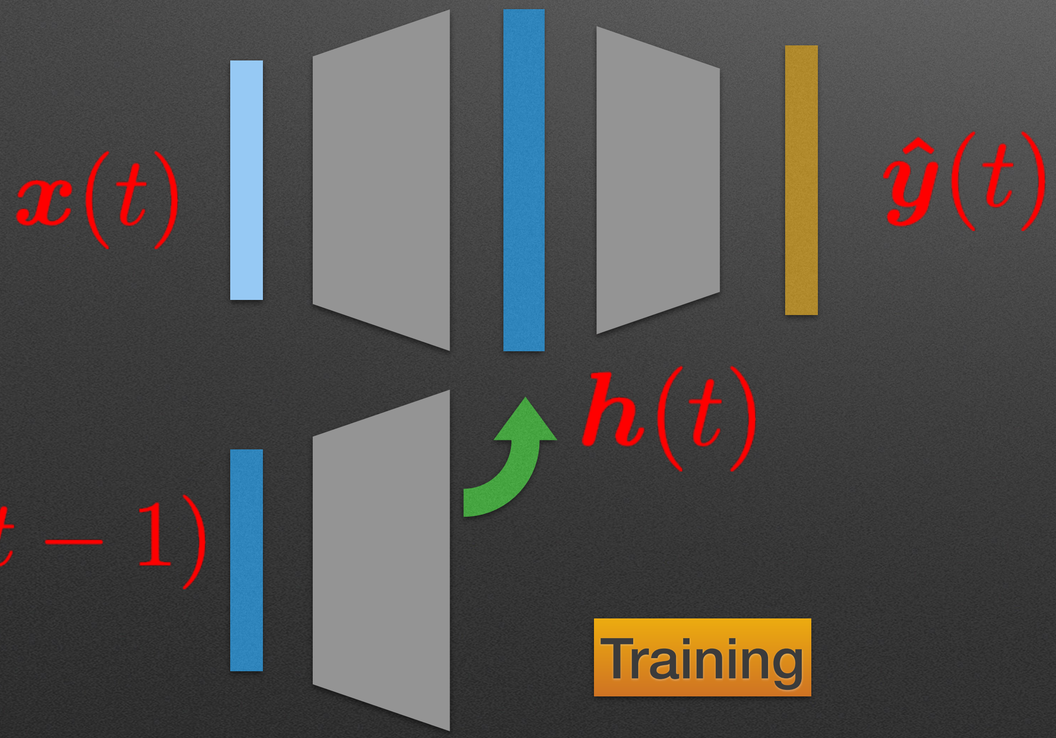
$$\boldsymbol{h}(t) = f\left(\hat{\boldsymbol{y}}(t-1), \boldsymbol{x}(t)\right)$$

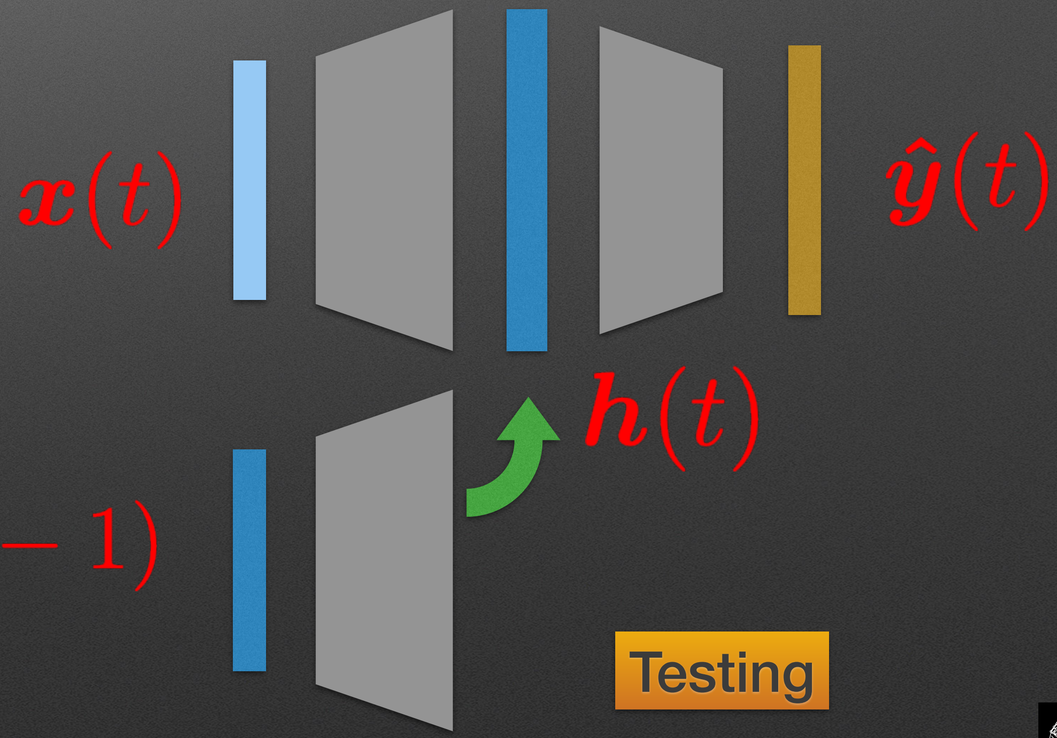


First order recurrence - output layer

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

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



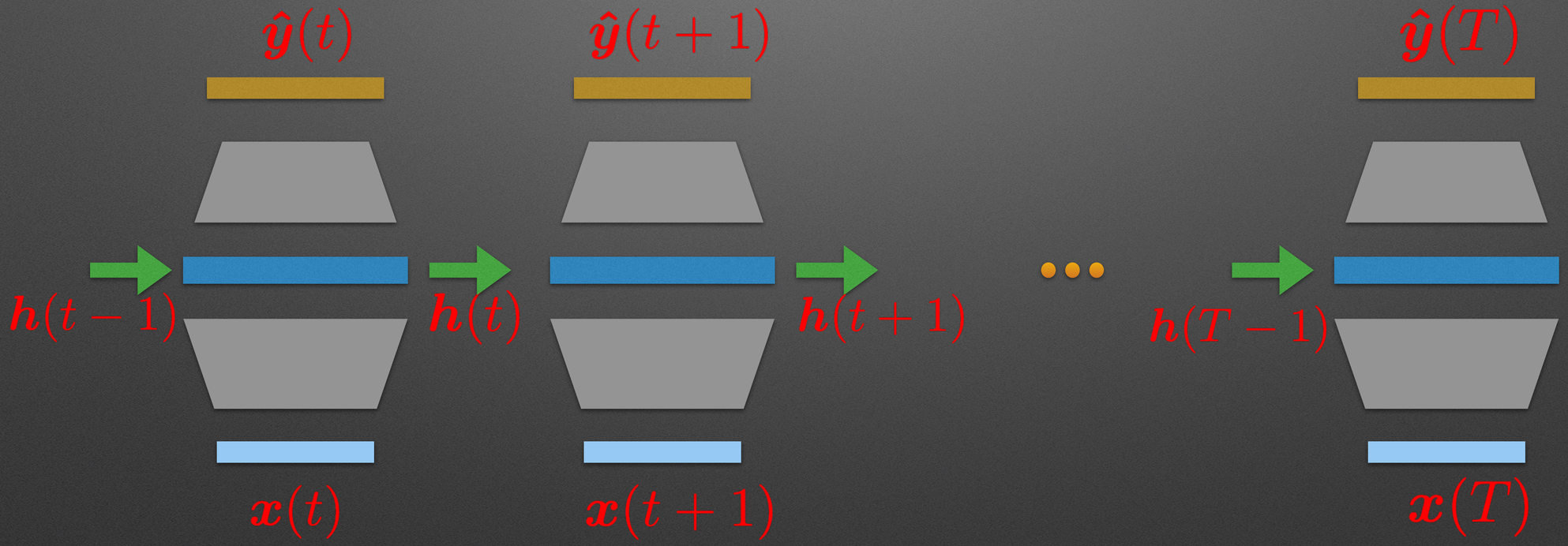


Coming up …

- Learning in recurrence networks: Back-propagation in time.
- Learning considerations : Issues with forgetting and long-short-term memory networks

First order recurrence - hidden layer

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

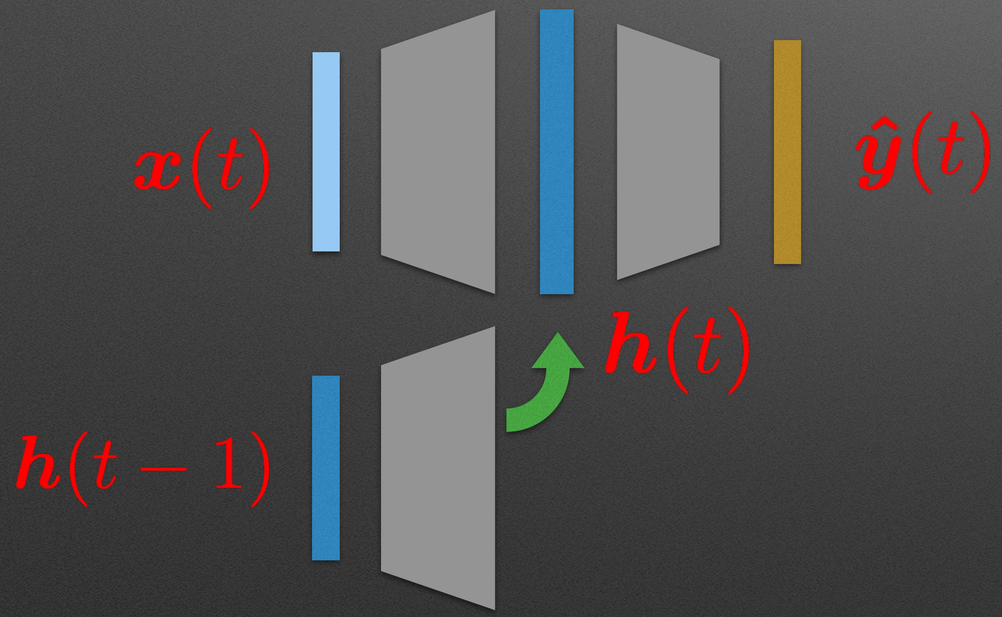


- Makes the hidden layer dependent of previous layer outputs in a recurring fashion.

First order recurrence - hidden layer

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

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



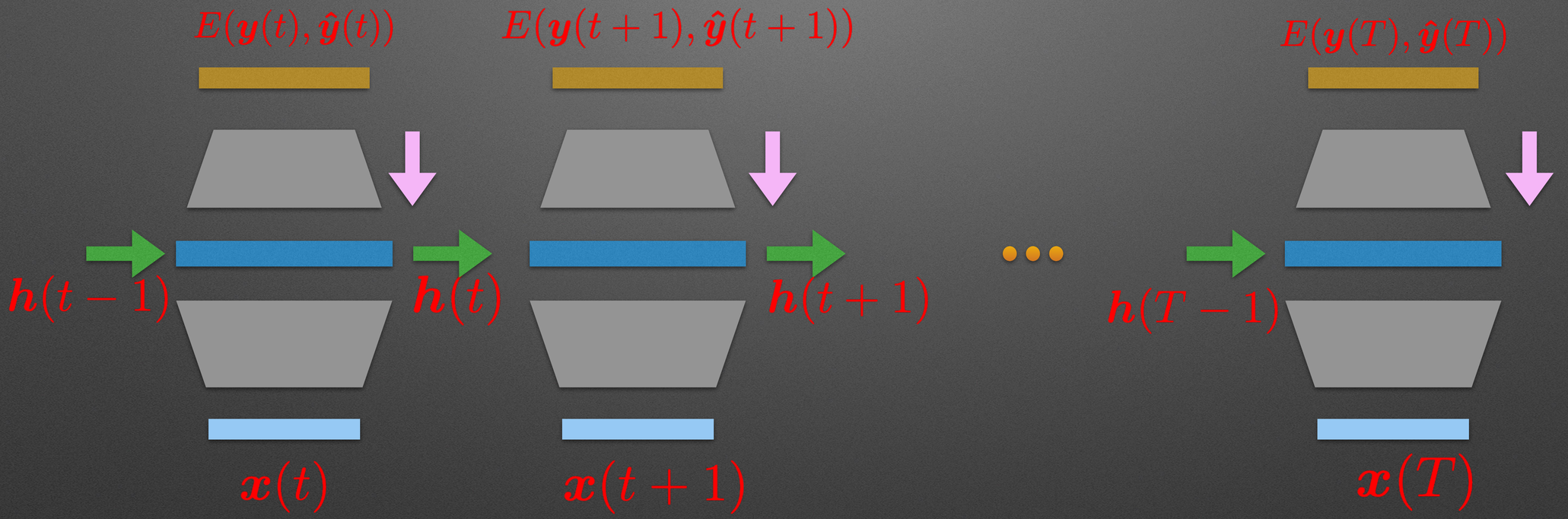
Model Forward Pass - 1 hidden layer

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

- Makes the hidden layer dependent of previous layer outputs in a recurring fashion.

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

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

- ✓ Output activations

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

- ✓ 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

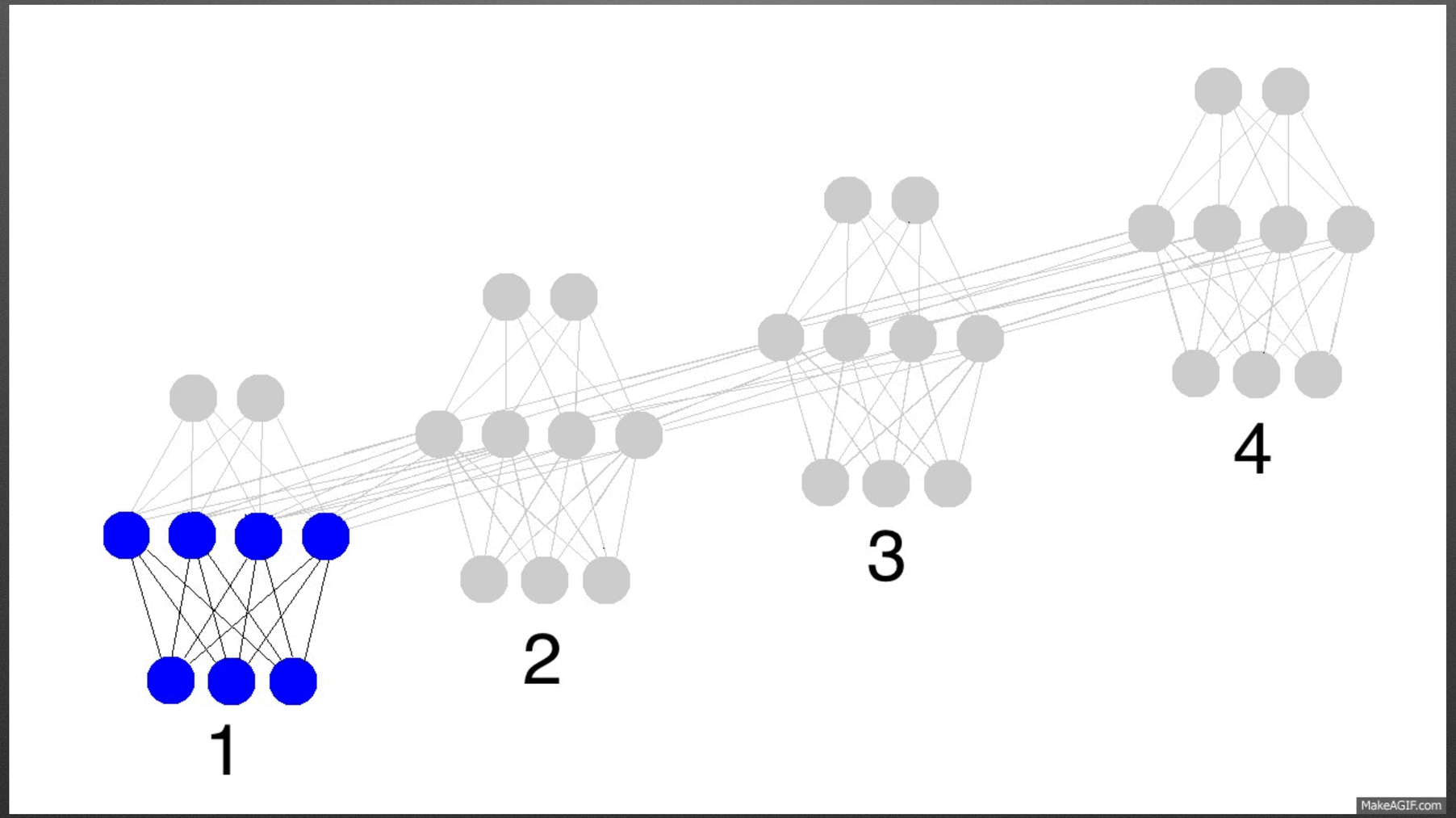
- Hidden activations for previous instances t = T-1,… 1

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

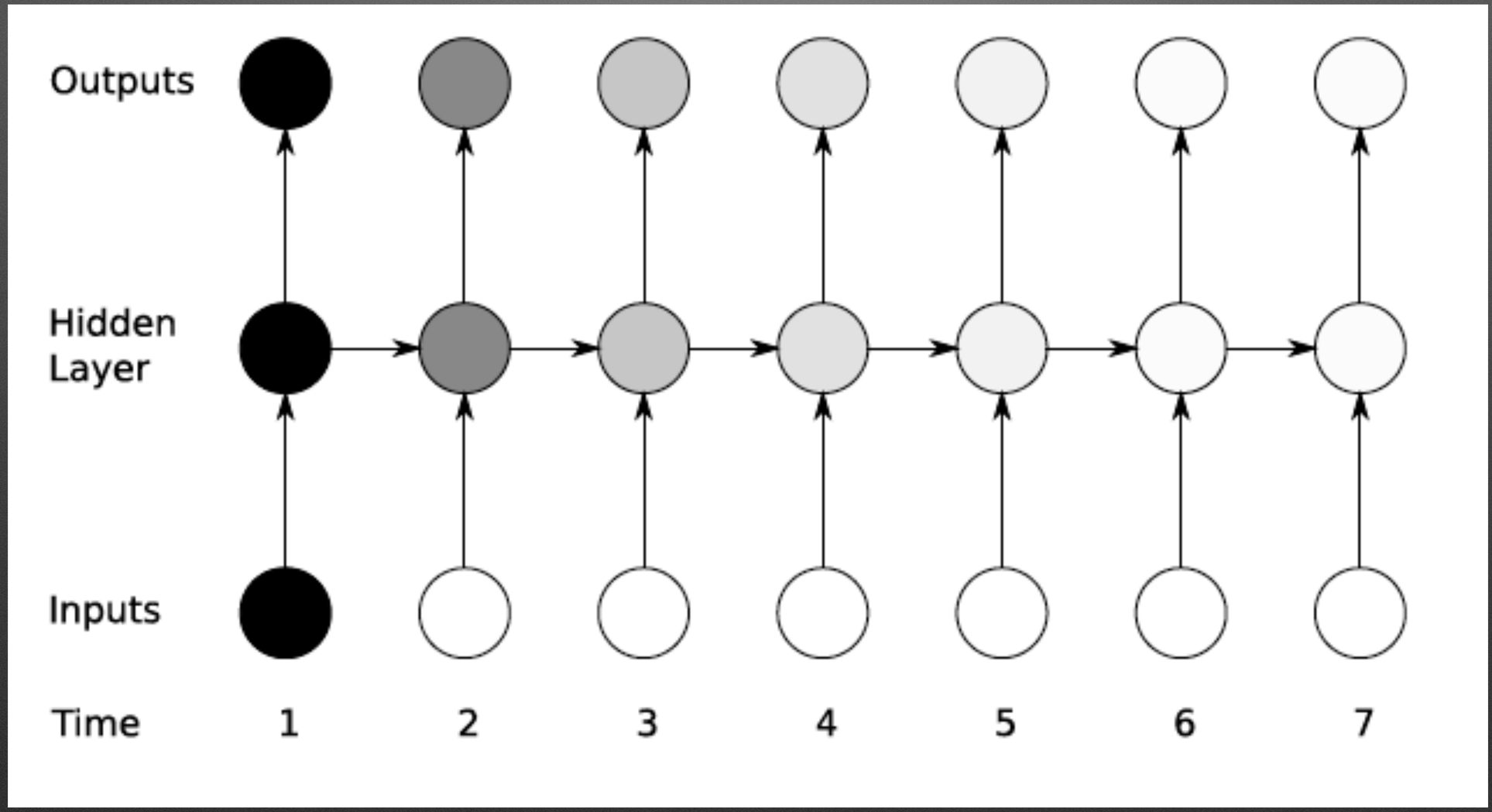
$$\frac{\partial E}{\partial \boldsymbol{h}^1(t)} = (\boldsymbol{W}^2)^T \frac{\partial E}{\partial \boldsymbol{a}^2(t)} + (\boldsymbol{U}^1)^T diag(1 - (\boldsymbol{h}^1(t+1)).^2) \frac{\partial E}{\partial \boldsymbol{h}^1(t+1)}$$

- Using the above gradient the rest of the gradients can be computed …

Back propagation through time

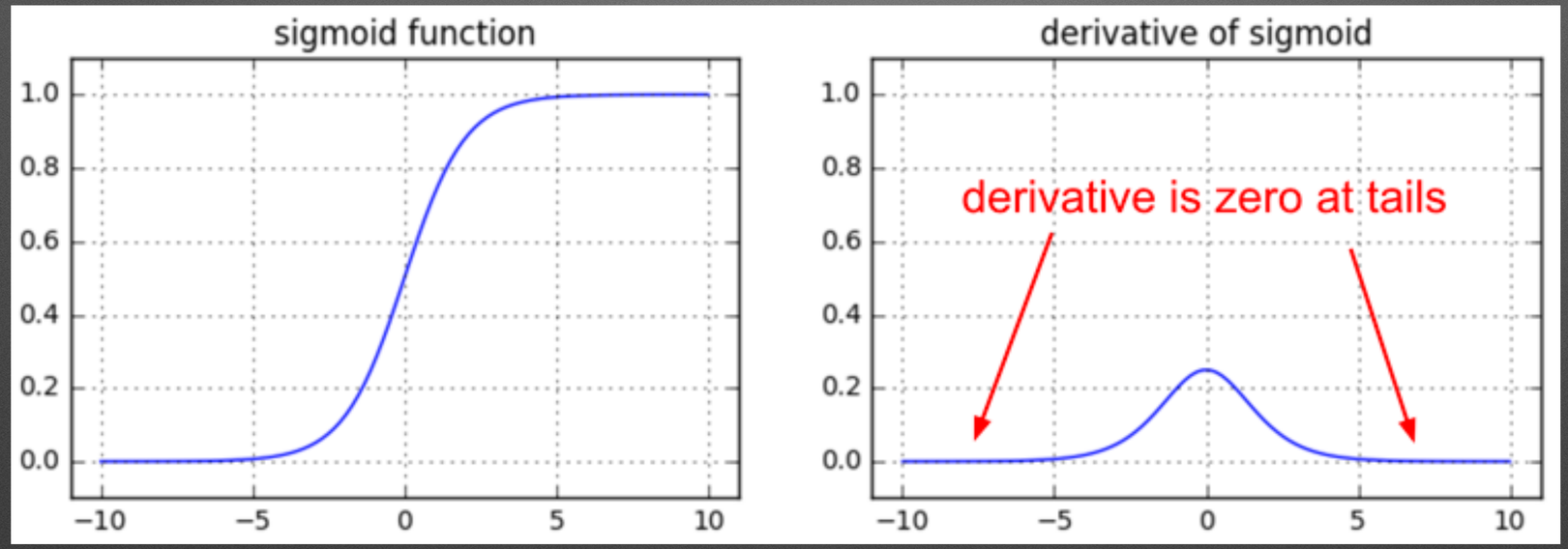


Long-term dependency issues



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