PHYS 139/239: Machine Learning in Physics

Lecture 10: More on RNNs

Javier Duarte — February 9, 2023

Final groups projects!

- Groups have been formed for final projects!
- Encourage you to create a (public) channel in Slack for communication and meet (frequently) to decide on a project and collaborate
- Final project should be an attempt at replicating (or extending) a (portion of) a published paper on ML in physics
    - Syllabus contains some candidates for papers to choose from with accessible datasets
    - If code exists, you can look at it as a reference (and if so cite it!), but your group should implement it yourself

Advice on proposal

- Proposal due Friday 2/24 5pm
- Recommend using Google Docs or Overleaf to collaborate on writing
- Approximately 2 pages containing
  - Background, motivation, and dataset
  - Main task/challenge/problem
  - Method(s) to be used/explore (e.g. RNNs, GNNs, …)
  - Expected outcomes/deliverables (e.g. jet tagging AUC > X)
  - Weekly/daily schedule (with person power assigned), e.g.
    - Week 8: create GitHub (A), preprocess data (B), exploratory data analysis (C+D)
    - Week 9: define metrics (A), create simple benchmark (B), train, and evaluate (C+D)
    - Week 10: create more advanced model (A+B), train, and compare performance to simple benchmark (C+D)
    - Finals Week: clean up code (C+D), write presentation and final 4-page report (A+B+C+D)

Code + GitHub

- Cookiecutter data science is a nice project structure: https://drivendata.github.io/cookiecutter-data-science/

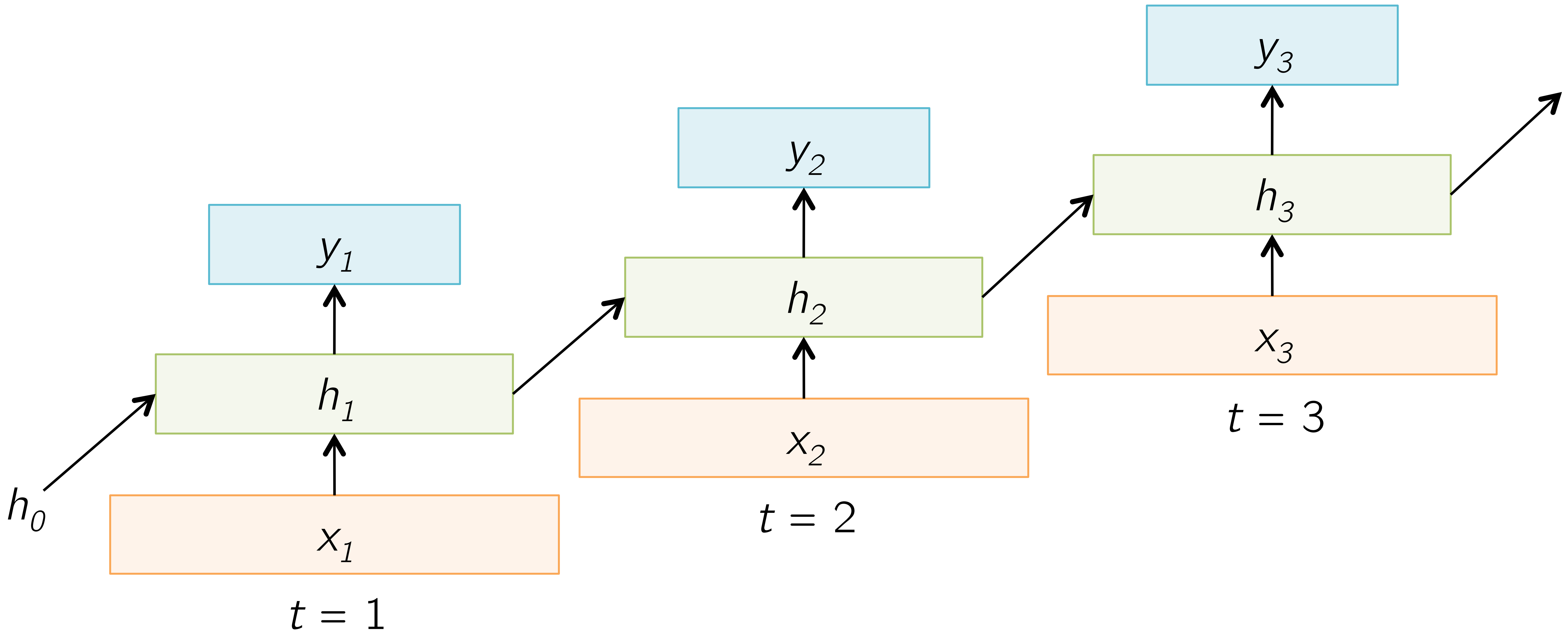
  - Find out more here: https://github.com/afraenkel/DSC180A-DS-Methodology/blob/master/week04/Lecture%2003.pdf

- GitHub tutorial: https://docs.github.com/en/get-started/quickstart/hello-world

Homeworks 3 + 4

- Will both be very short (to give you time to focus on projects)
- Definitely prioritize projects

Recap: Recurrent neural network

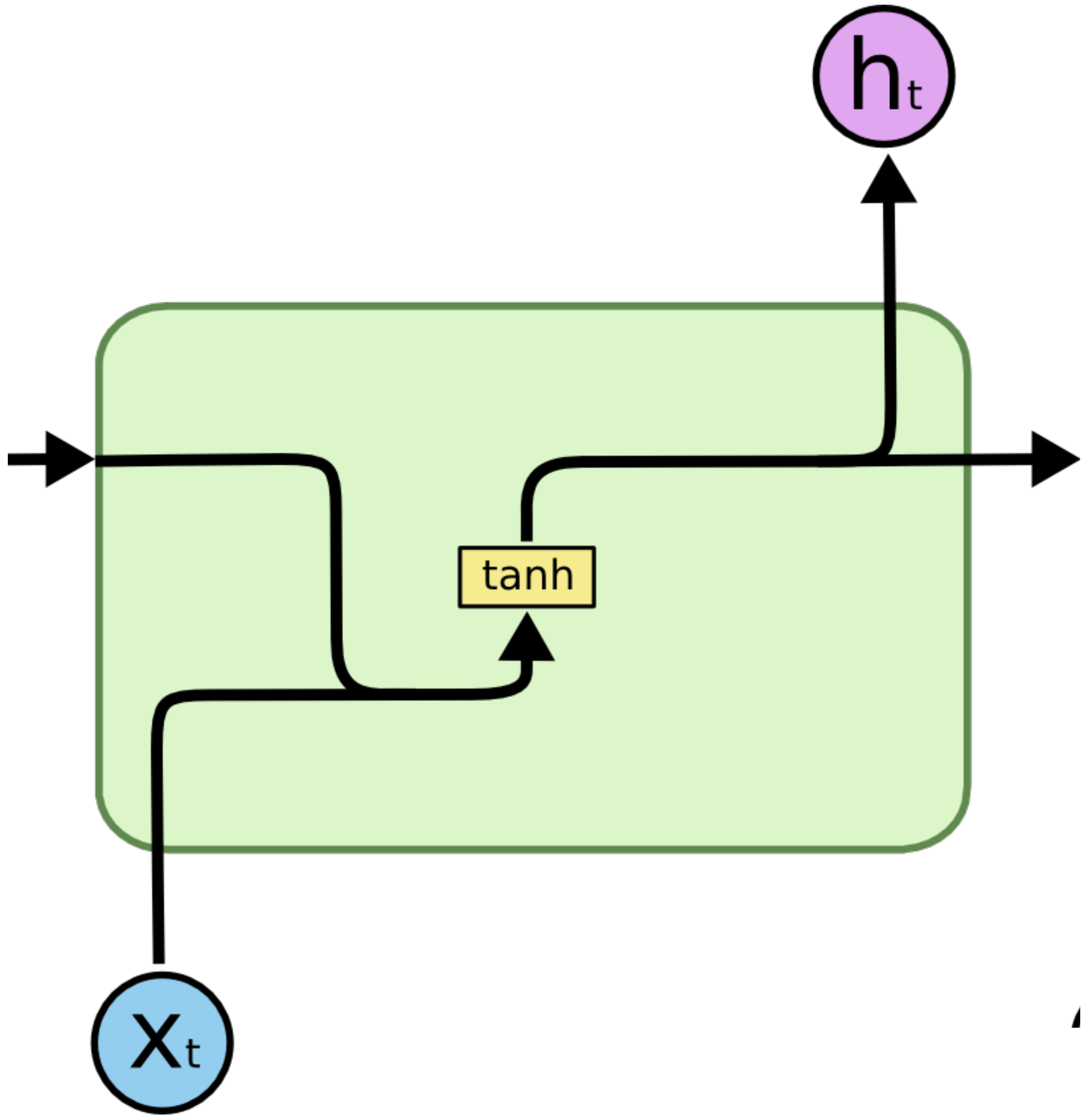


- Recurrent NN considers current input **and** previous hidden state
- Trained using backpropagation through time (unfolding network)

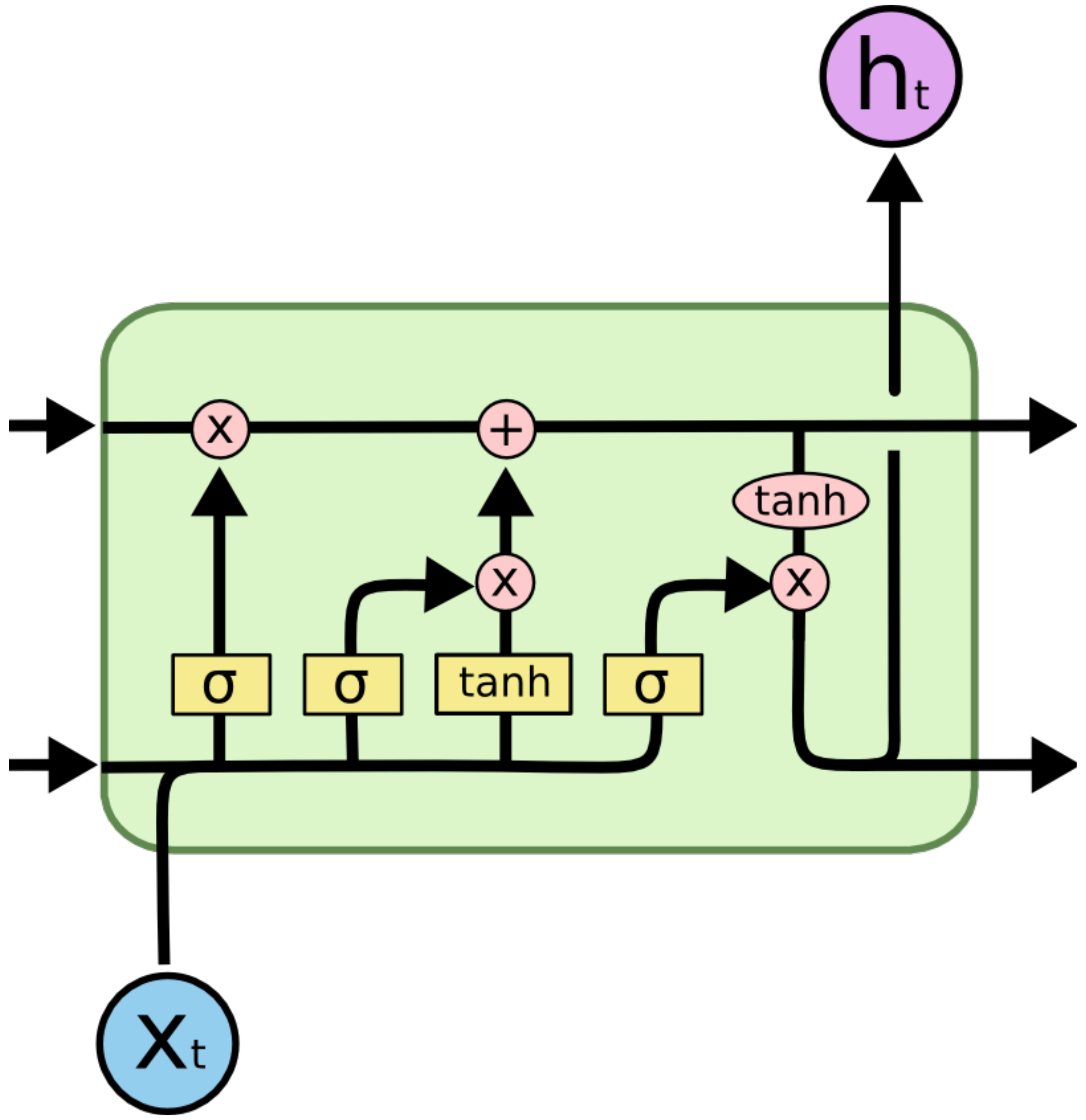
Recap: Long short-term memory (LSTM)

- Developed to cope with the issue of long-term dependencies [10.1162/neco.1997.9.8.1735]
- LSTM uses this idea of “constant error flow” to ensure that gradients don’t decay
- Key components are
  - an internal memory (“cell state”)
  - gates that control the cell state actively
- Variant of LSTM simplify operations while maintaining key pieces

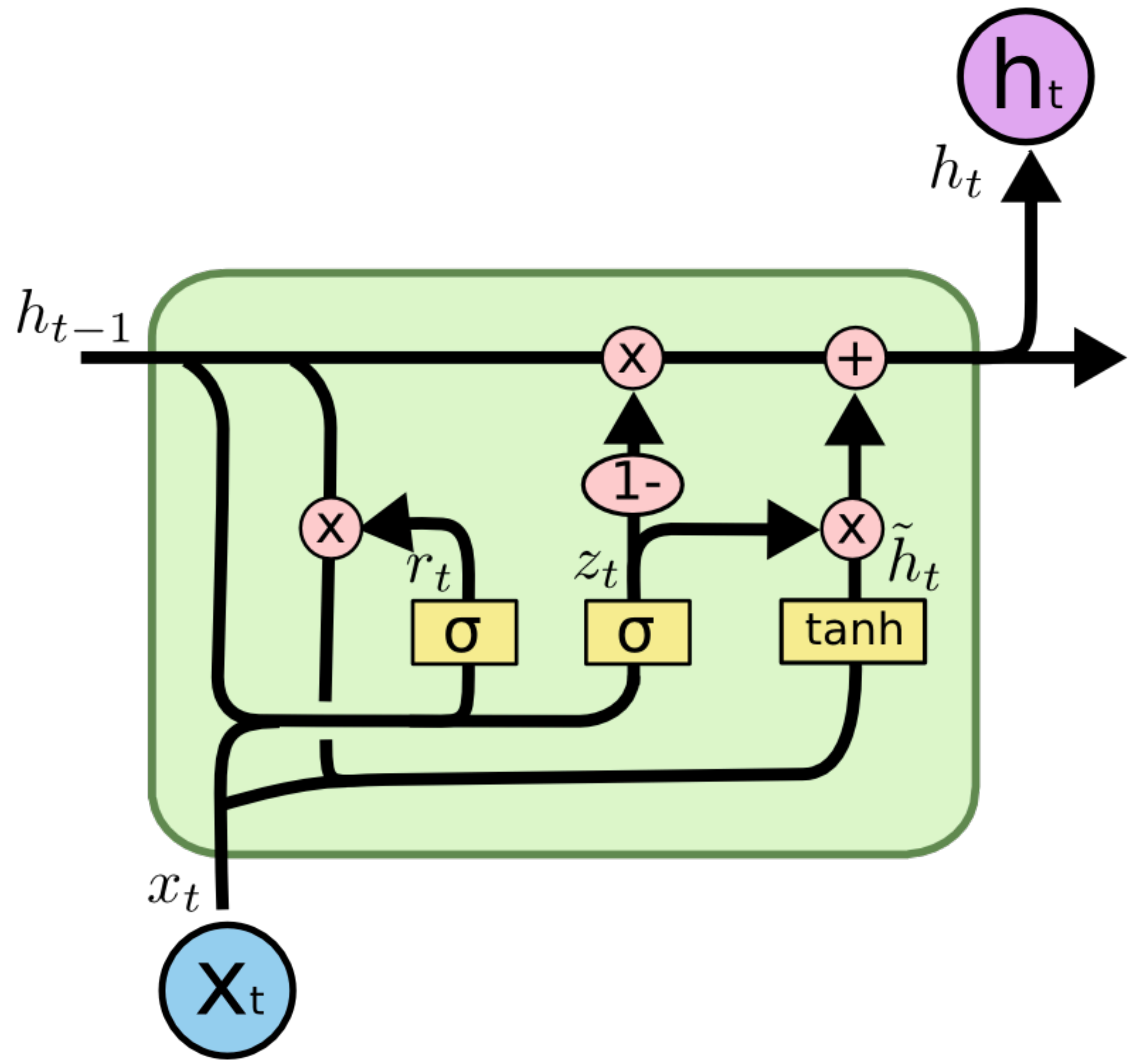
Recap: Simple RNN vs. LSTM vs. GRU



$$h_t = \tanh(W_f x_t + U_f h_{t-1} + b_f)$$



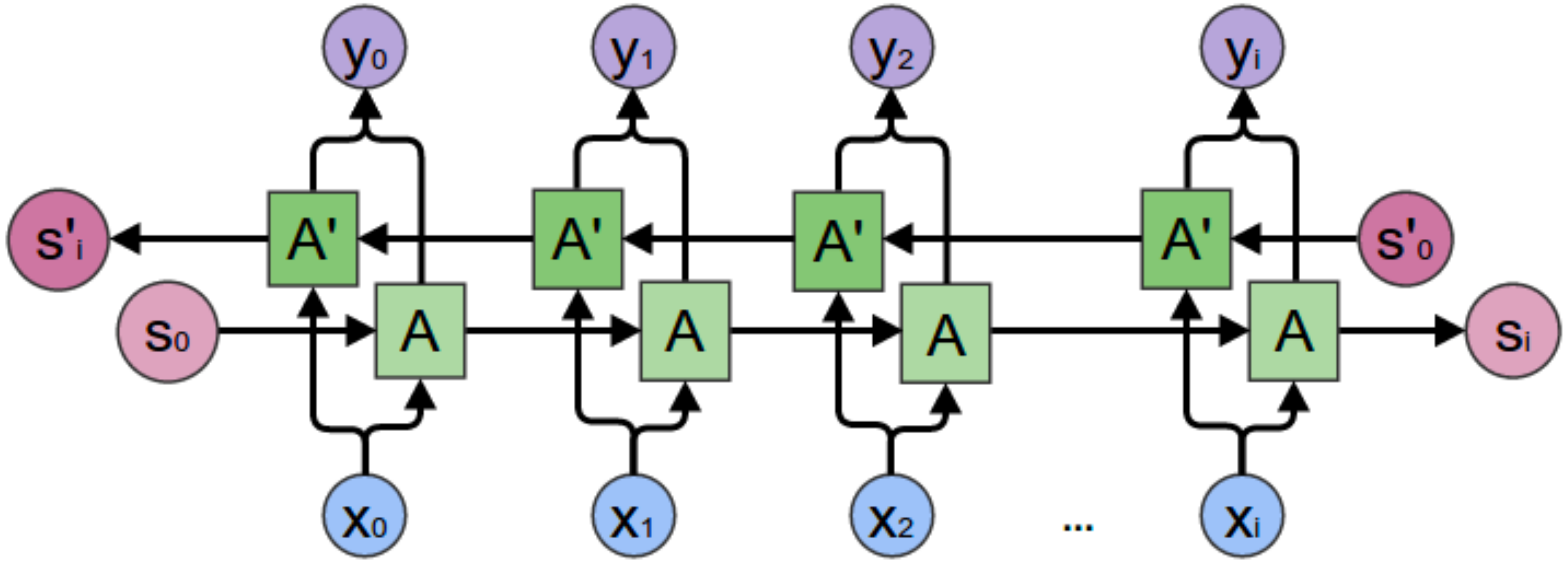
$$f_t = \sigma(W_f x_t + U_f h_{t-1} + b_f)$$
$$i_t = \sigma(W_i x_t + U_i h_{t-1} + b_i)$$
$$\tilde{C}_t = \tanh(W_c x_t + U_c h_{t-1} + b_c)$$
$$C_t = f_t \cdot C_{t-1} + i_t \cdot \tilde{C}_t$$
$$o_t = \sigma(W_o x_t + U_o h_{t-1} + b_o)$$
$$h_t = o_t \cdot \tanh(C_t)$$



$$r_t = \sigma(W_r x_t + U_r h_{t-1} + b_r)$$
$$z_t = \sigma(W_z x_t + U_z h_{t-1} + b_z)$$
$$\tilde{h}_t = \tanh(W_z x_t + U_h(r_t \cdot h_{t-1}) + b_h)$$
$$h_t = (1 - z_t) \cdot h_{t-1} + z_t \cdot \tilde{h}_t$$

Bidirectional RNNs

- A typical state in an RNN (simple RNN, GRU, or LSTM) relies on the past and the present events
- However, there can be situations where a prediction depends on the past, present, and future events
- Bidirectional RNN (BRNN) is a combination of two RNNs—one RNN moves forward, beginning from the start of the data sequence, and the other, moves backward, beginning from the end of the data sequence
  - Note: not be appropriate for forecasting!



$$h_t = \tanh(W_f x_t + U_f h_{t-1} + b_f)$$

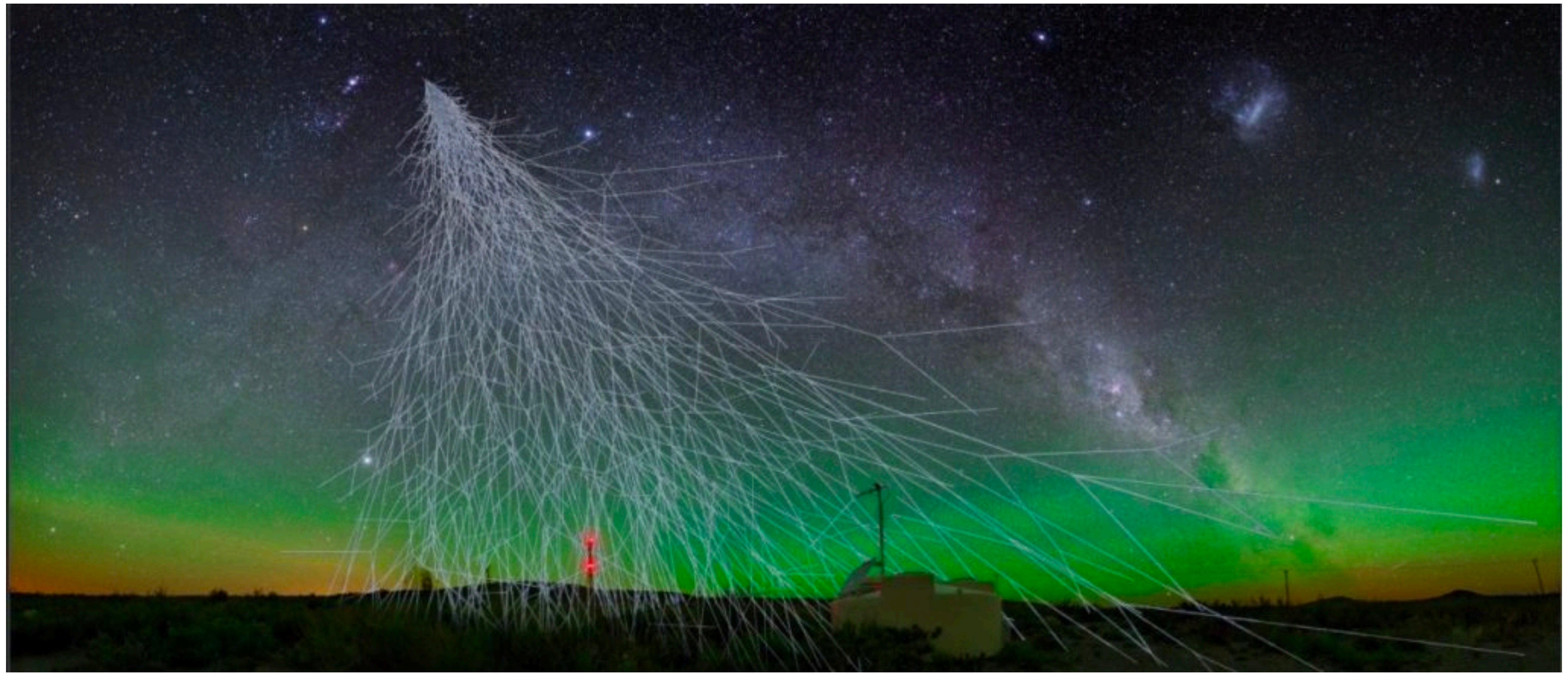
$$h'_t = \tanh(W'_f x_t + U'_f h_{t+1} + b'_f)$$

Hands-on: Classify cosmic ray radio signals

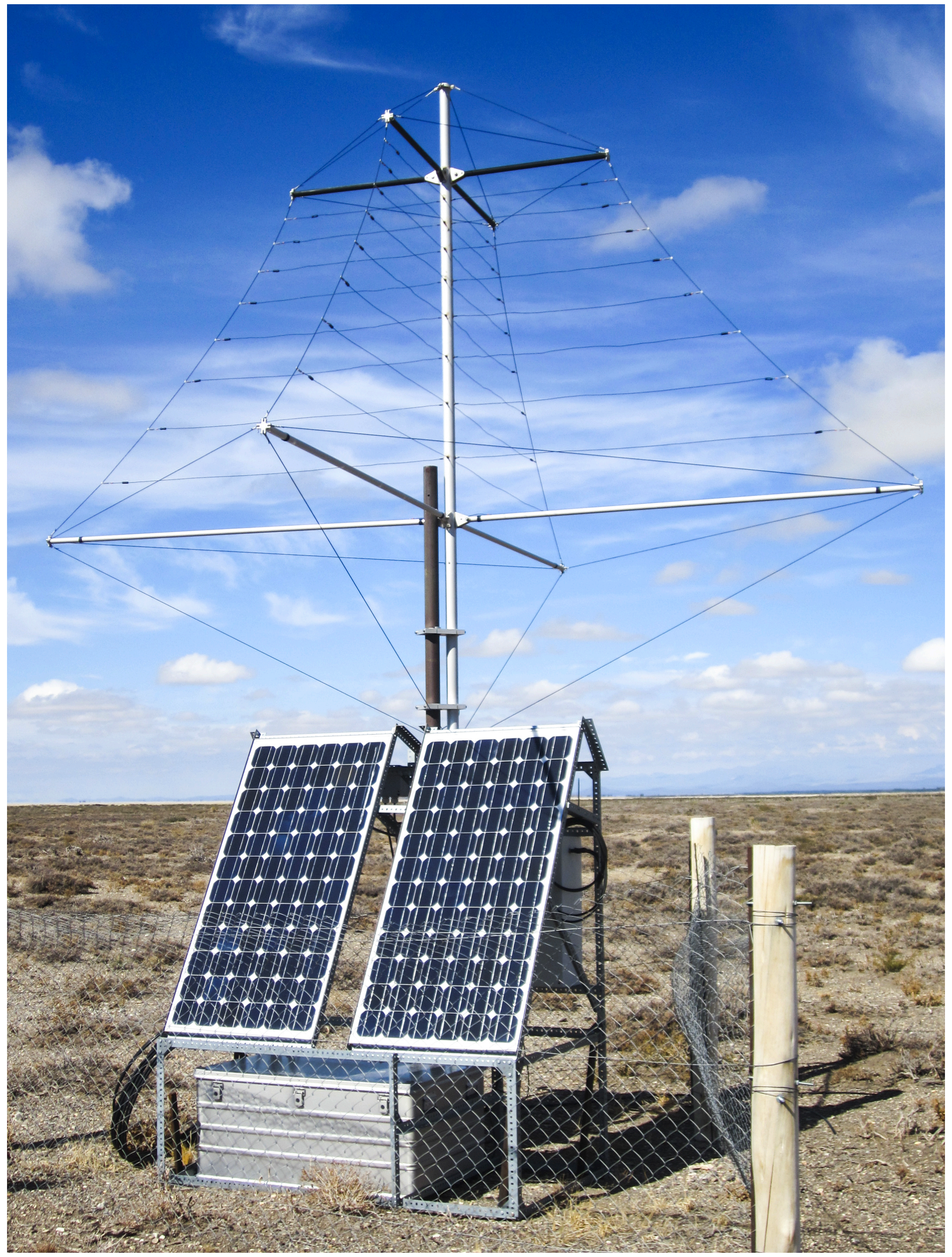
- Auger Engineering Radio Array (AERA) is a system of radio antennas installed at the Pierre Auger Observatory [42], measuring pulses of a few nanoseconds in length emitted by cosmic ray air showers with energies above 10^{17} eV

- Goal is to identify radio signals emitted by ultra-high energy cosmic rays (UHECRs) that initiate extensive particle showers in the atmosphere

- UHECRs are likely protons and nuclei with energies extending from 10^{18} eV to above 10^{20} eV



arXiv:1901.04079

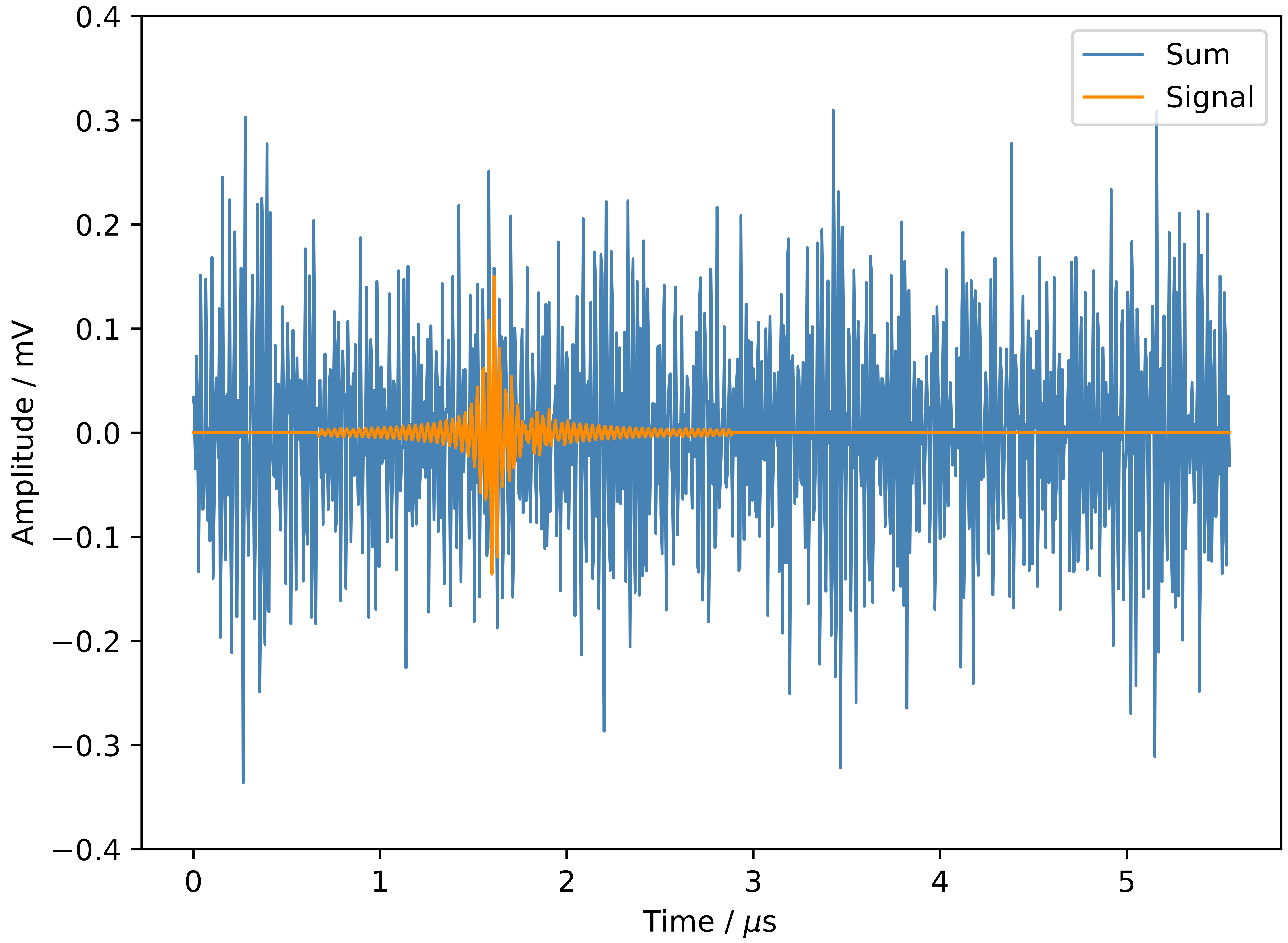


Hands-on: Classify cosmic ray radio signals

- Characterize strength of a signal in a noisy trace using signal-to-noise (SNR) ratio

$$\mathrm{SNR} = \frac{\max\ \mathrm{signal}}{\mathrm{RMS}_{\mathrm{noise}}} = \frac{A^S_{\max}}{\sqrt{\frac{1}{N}\sum_{i=1}^{N} A_i^2}}$$

- Simulated data produced with SNR values between 0.5 and 5.0
  - SNR = 1.5 shown
  - Preprocessing applied: normalize each generated time trace by dividing each bin by the standard deviation of the whole trace



- Note: more advanced "signal recovery" task can be performed as well, but we will focus on simpler classification task

Next time

- Graphs!