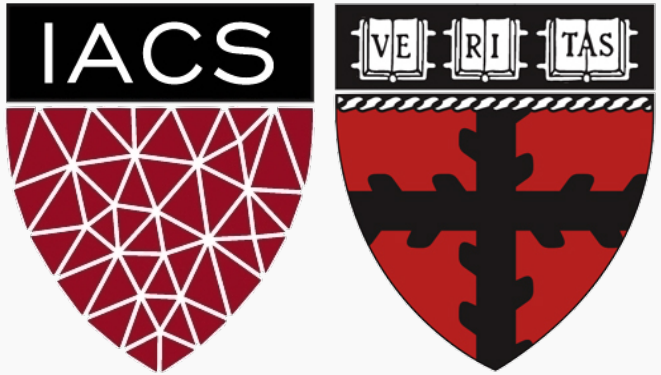
Gated Recurrent Units

CS109B Data Science 2

Pavlos Protopapas, Mark Glickman, and Chris Tanner



Outline

RNN review

RNN shortcomings

Pavlos Recurrent Unit(PRU)

Gated Recurrent Unit (GRU)

Outline

RNN review

RNN shortcomings

Pavlos Recurrent Unit(PRU)

Gated Recurrent Unit (GRU)

RECAP: RNNs

RNNs exhibit the following advantages for sequence modeling:

- Handle variable-length sequences
- Keep track of long-term dependencies
- Maintain information about the order as opposed to FFNN
- Share parameters across the network

THE END

Outline

RNN review

RNN shortcomings

Pavlos Recurrent Unit(PRU)

Gated Recurrent Unit (GRU)

RNN ISSUES?

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\mathbf{h}_{t-1} + \beta_1\right)$$

RNN ISSUES?

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\mathbf{h}_{t-1} + \beta_1\right)$$

Current Input

Previous hidden state

RNN ISSUES?

$$\mathbf{h}_t = \tanh(\mathbf{V}\mathbf{X}_t + \mathbf{U}\mathbf{h}_{t-1} + \beta_1)$$

Current Input

Previous hidden state

The trainable weights V, U are constants and they are not a function of input X_t or previous state h_{t-1}.

RNN ISSUES?

$$\mathbf{h}_t = \tanh(\mathbf{V}\mathbf{X}_t + \mathbf{U}\mathbf{h}_{t-1} + \beta_1)$$

Current Input

Previous hidden state

The trainable weights V, U are constants, and they are not a function of input X_t or previous state h_{t-1}.

The simple repeated structure suffers from vanishing/exploding gradients as we move farther away from the target, and hence weights do not learn from initial inputs.

A very long email

Hello Professor Protopapas,
My name is Jim and I am writing to you for an opportunity to work with your research group.
I am a very motivated person and love playing football, I also love dancing and having a good time, but I am also dedicated to conducting research. I spent the last three months sincerely completing the coursera course on introduction to machine learning by Andrew Ng, and I feel like now I completely understand all the techniques of data science and that makes me a prime candidate for your research group. So please consider my request.

A very long email

Hello Professor Protopapas,
My name is Jim and I am writing to you for an opportunity to work with your research group.
I am a very motivated person and I love playing football, I also love dancing and having a good time, but I am also dedicated to conducting research. I spent the last three months sincerely completing the coursera course on introduction to machine learning by Andrew Ng, and I feel like now I completely understand all the techniques of data science and that makes me a prime candidate for your research group. So please consider my request.

Another very long email

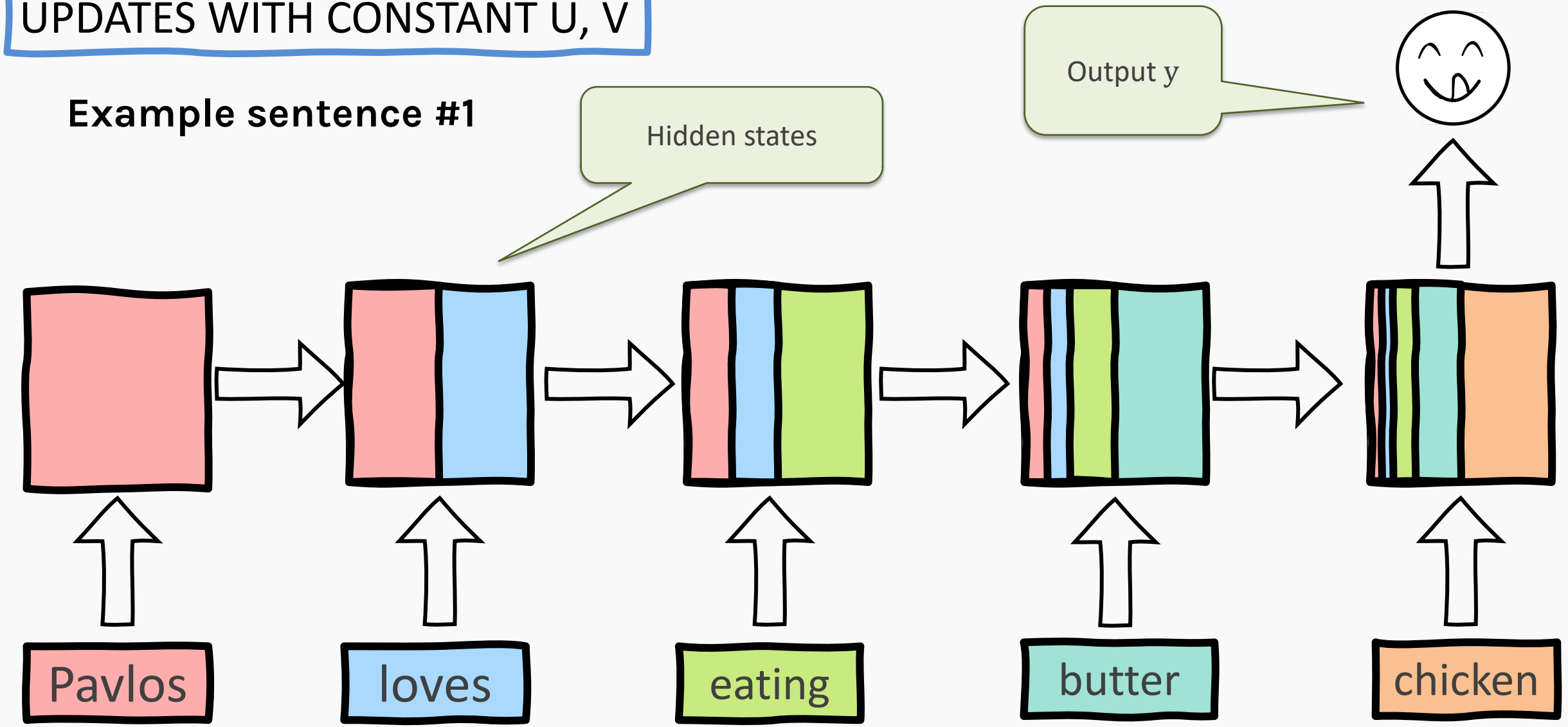
Hi Pavlos,
Rashmi here. I was recently working on the new exercise you proposed last night but unfortunately the dataset I am using is too big for Ed. I think you'll need to ask Alex to upload that dataset directly from backend. I know you don't like such workarounds and I specifically remember you asking me to work with something smaller, but I just don't think the exercise would be as nice if we use a smaller dataset, because the language model is not training very well. With this new dataset, I'm sure the students will connect the dots better and have more clarity in how rnns work. So please consider my request.

Another very long email

Hi Pavlos,
Rashmi here. I was recently working on the new exercise you proposed last night but unfortunately the dataset I am using is too big for Ed. I think you'll need to ask Alex to upload that dataset directly from backend. I know you don't like such workarounds and I specifically remember you asking me to work with something smaller, but I just don't think the exercise would be as nice if we use a smaller dataset, because the language model is not training very well. With this new dataset, I'm sure the students will connect the dots better and have more clarity in how rnns work. So please consider my request.

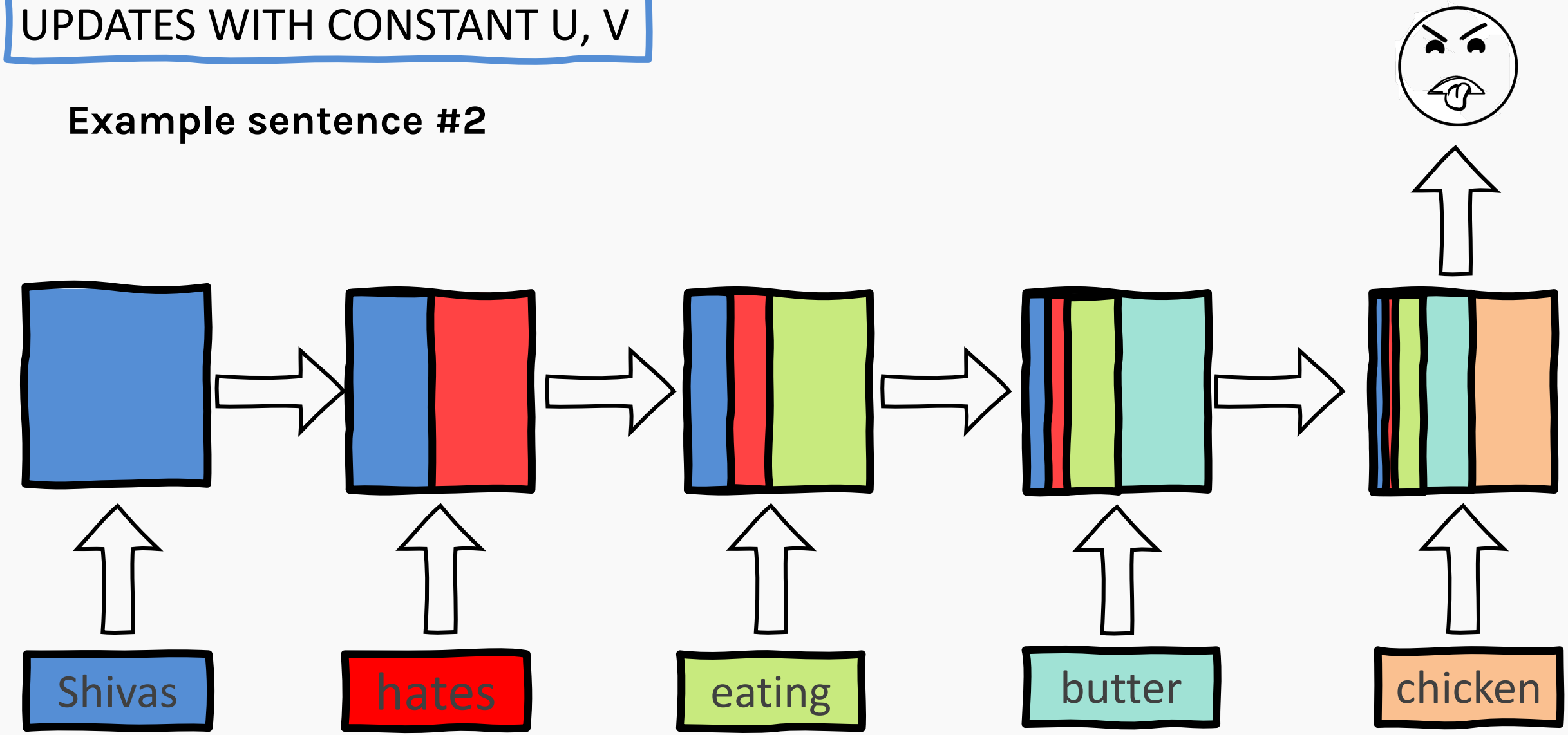
UPDATES WITH CONSTANT U, V

Example sentence #1



UPDATES WITH CONSTANT U, V

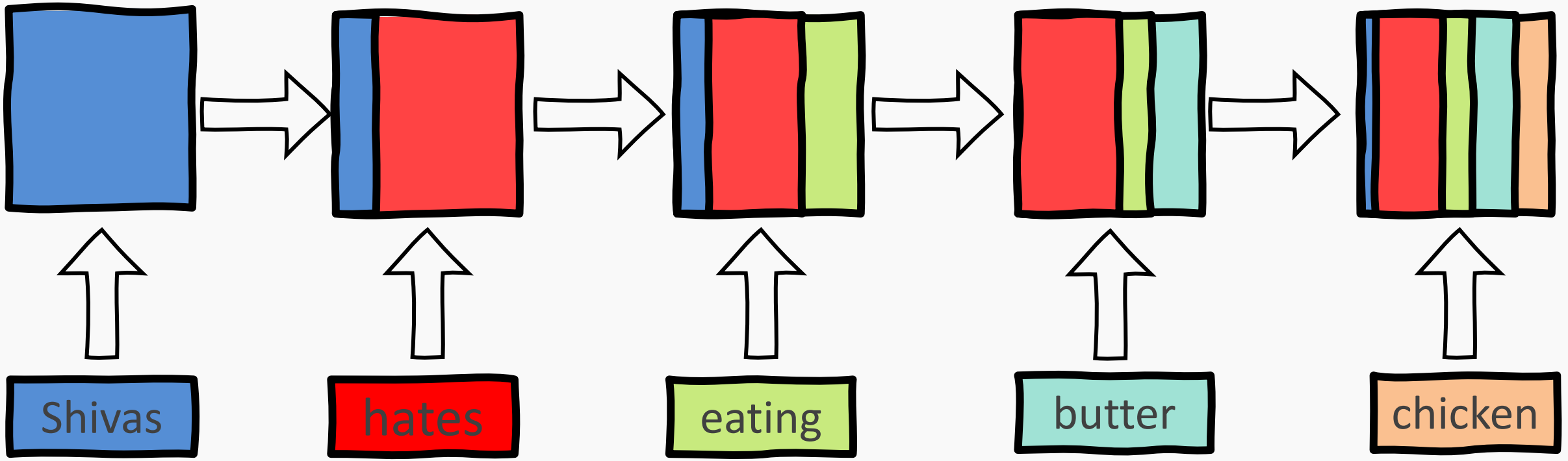
Example sentence #2



What we want

Example sentence #2

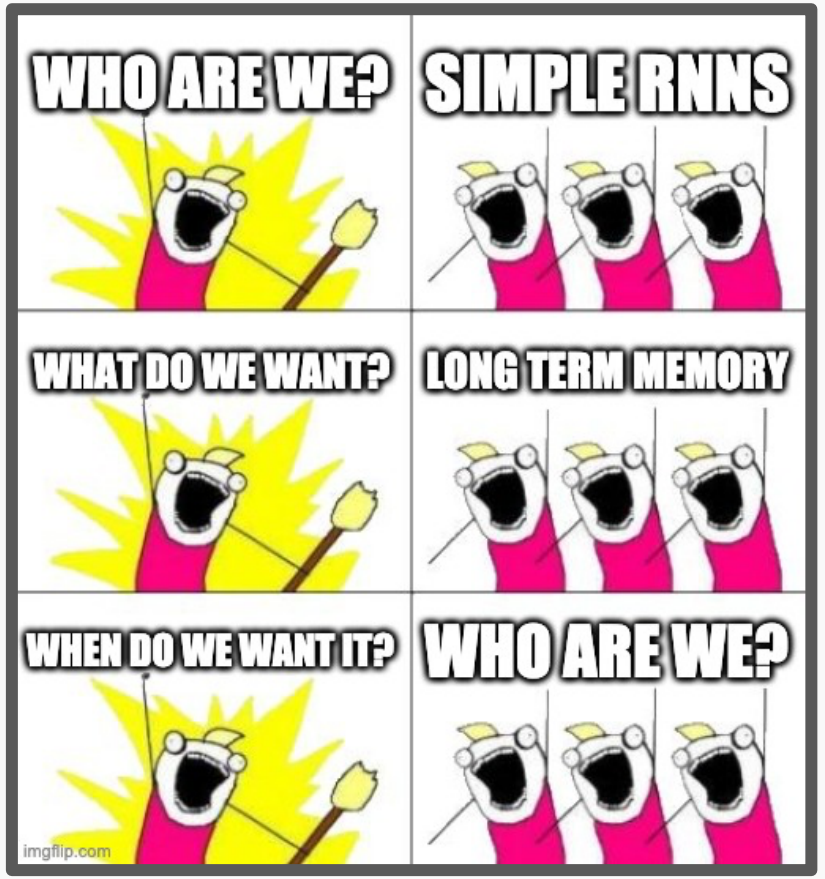
UPDATES WHEN U, V DEPEND ON X_t and h_{t-1}



RNN Wishlist?

$$\mathbf{h}_t = \tanh\left(\mathbf{V}(h_{t-1}, X_t)\mathbf{X}_t + \mathbf{U}(h_{t-1}, X_t)\mathbf{h}_{t-1} + \beta_1\right)$$

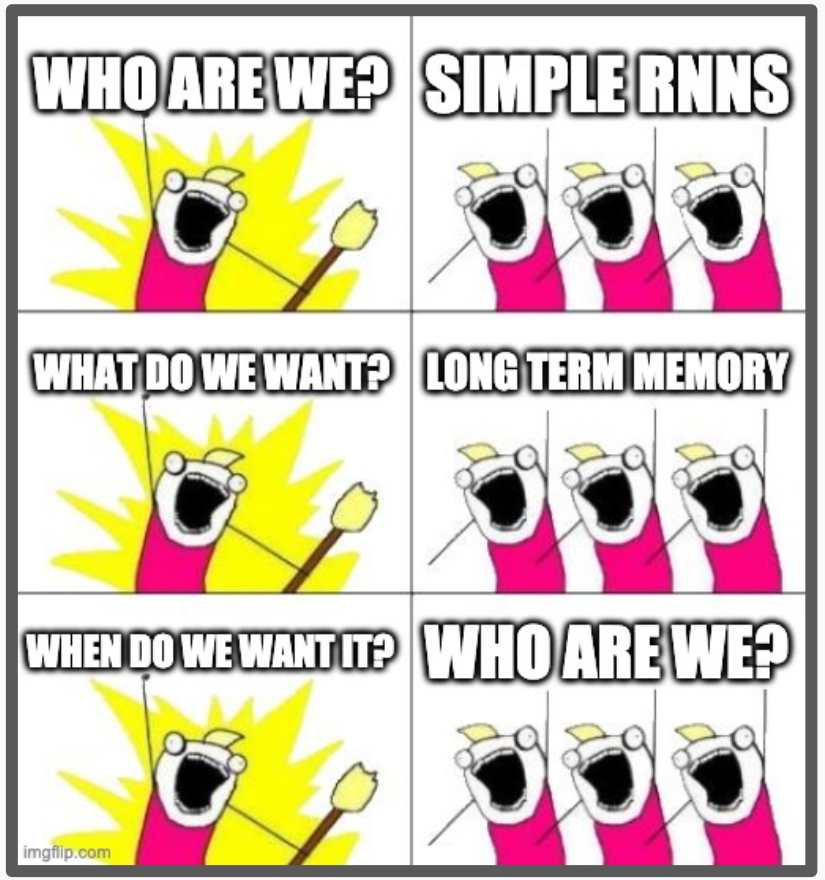
- We want our trainable weights V, U to somehow incorporate the input X_t and the previous state h_{t-1}.
- We need to solve the vanishing gradient problem so that our network also learns from inputs at the beginning of a sequence.



RNN Wishlist?

$$\mathbf{h}_t = \tanh\left(\mathbf{V}(h_{t-1}, X_t)\mathbf{X}_t + \mathbf{U}(h_{t-1}, X_t)\mathbf{h}_{t-1} + \beta_1\right)$$

- We want our trainable weights V, U to somehow incorporate the input X_t and the previous state h_{t-1}.
- We need to solve the vanishing gradient problem so that our network also learns from inputs at the beginning of a sequence.



Outline

RNN review

RNN shortcomings

Pavlos Recurrent Unit(PRU)

Gated Recurrent Unit (GRU)

Pavlos Recurrent Unit (PRU)

$$\mathbf{h}_t = \tanh\left(\mathbf{V}(h_{t-1}, X_t)\mathbf{X}_t + \mathbf{U}(h_{t-1}, X_t)\mathbf{h}_{t-1} + \beta_1\right)$$

Idea #1: Keep V as a constant and let only U be a function of X_t, h_{t-1}.

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}(h_{t-1}, X_t)\mathbf{h}_{t-1} + \beta_1\right)$$

Idea #2: Keep U as a constant too, and introduce a new variable, PP, that is a function of X_t, h_{t-1}

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{PP}(h_{t-1}, X_t)\,\mathbf{h}_{t-1}\right] + \beta_1\right)$$

Pavlos recurrent unit

Idea #4: Use element wise multiplication so we not mix different hidden state elements::

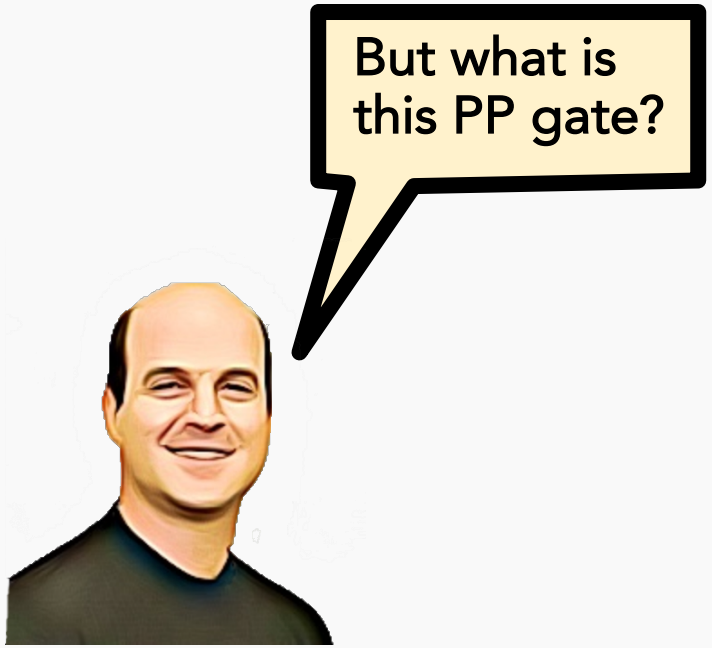
$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{P}\mathbf{P}(h_{t-1}, X_t) \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

The Hadamard product is an element wise operation between two matrices of the same size.

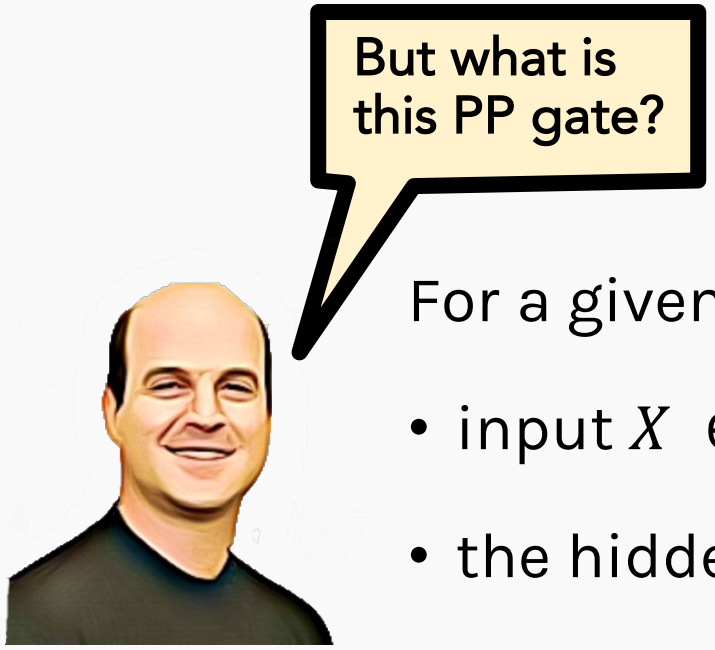
Use shorter notation. Also, we give a name to the PP variable: PP-gate. The PP gate decides the amount of past information to be considered for the hidden state.

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{P}\mathbf{P}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

PRU continued...



PRU continued...

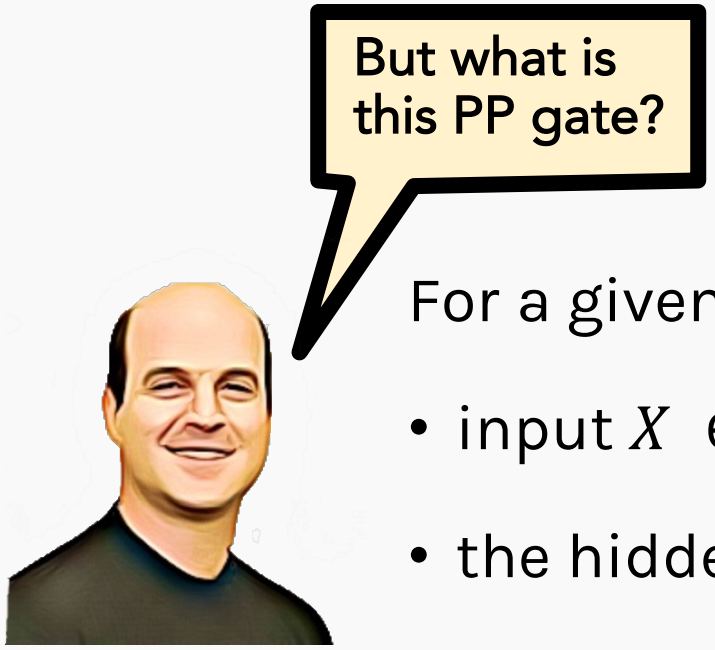


For a given timestep t:

- input $X \in \mathbb{R}^{1\times d}$, (number of inputs: d)
- the hidden state in the previous timestep, $h_{t-1} \in \mathbb{R}^{1\times h}$, (number of hidden units: h)

then the PP gate $PP_t \in \mathbb{R}^{1\times h}$ is given as:

PRU continued…



For a given timestep t:

- input $X \in \mathbb{R}^{d \times 1}$, (number of inputs: d)
- the hidden state in the previous timestep, $h_{t-1} \in \mathbb{R}^{h \times 1}$, (number of hidden units: h)

then the PP gate $PP_t \in \mathbb{R}^{h \times 1}$ is given by:

$$\mathbf{PP}_t = \sigma\left(\mathbf{V}_{pp}\mathbf{X}_t + \mathbf{U}_{pp}\mathbf{h}_{t-1} + \beta_{pp}\right)$$

PRU continued…

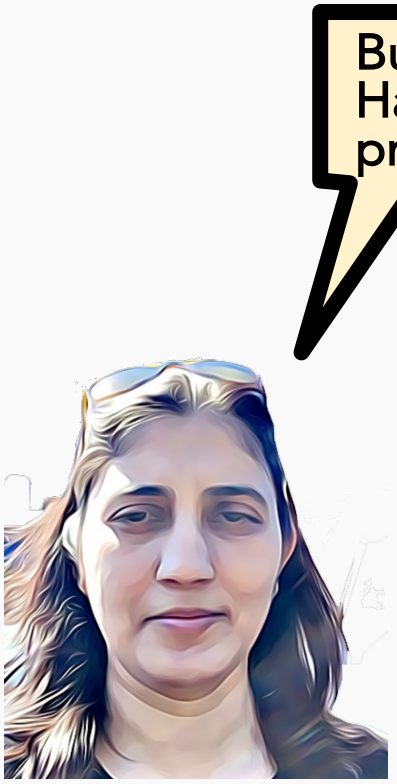
$$\mathbf{PP}_t = \sigma\left(\mathbf{V}_{pp}\mathbf{X}_t + \mathbf{U}_{pp}\mathbf{h}_{t-1} + \beta_{pp}\right)$$

$\mathbf{V}_{pp} \in \mathbb{R}^{h \times d}$, $\mathbf{X}_t \in \mathbb{R}^{d \times 1}$, $\mathbf{U}_{pp} \in \mathbb{R}^{h \times h}$, $\mathbf{h}_{t-1} \in \mathbb{R}^{h \times 1}$, $\beta_{pp} \in \mathbb{R}^{h \times 1}$

Sigmoid activation

- The PP gate matrix $PP_t \in \mathbb{R}^{h \times 1}$, will have the same dimensions as the hidden state of the previous timestep, $h_{t-1} \in \mathbb{R}^{h \times 1}$.
- It will have values between 0 & 1 because of the sigmoid activation.
- Its values depend on the input, X_t, and the previous hidden layer output, h_{t-1}.

PRU continued...



- The Hadamard Product is simply the element-wise multiplication of two matrices of the same dimensions.

1	2	1	-5
4	3	2	6
4	2	1	4
0	0	0	1

$\odot$

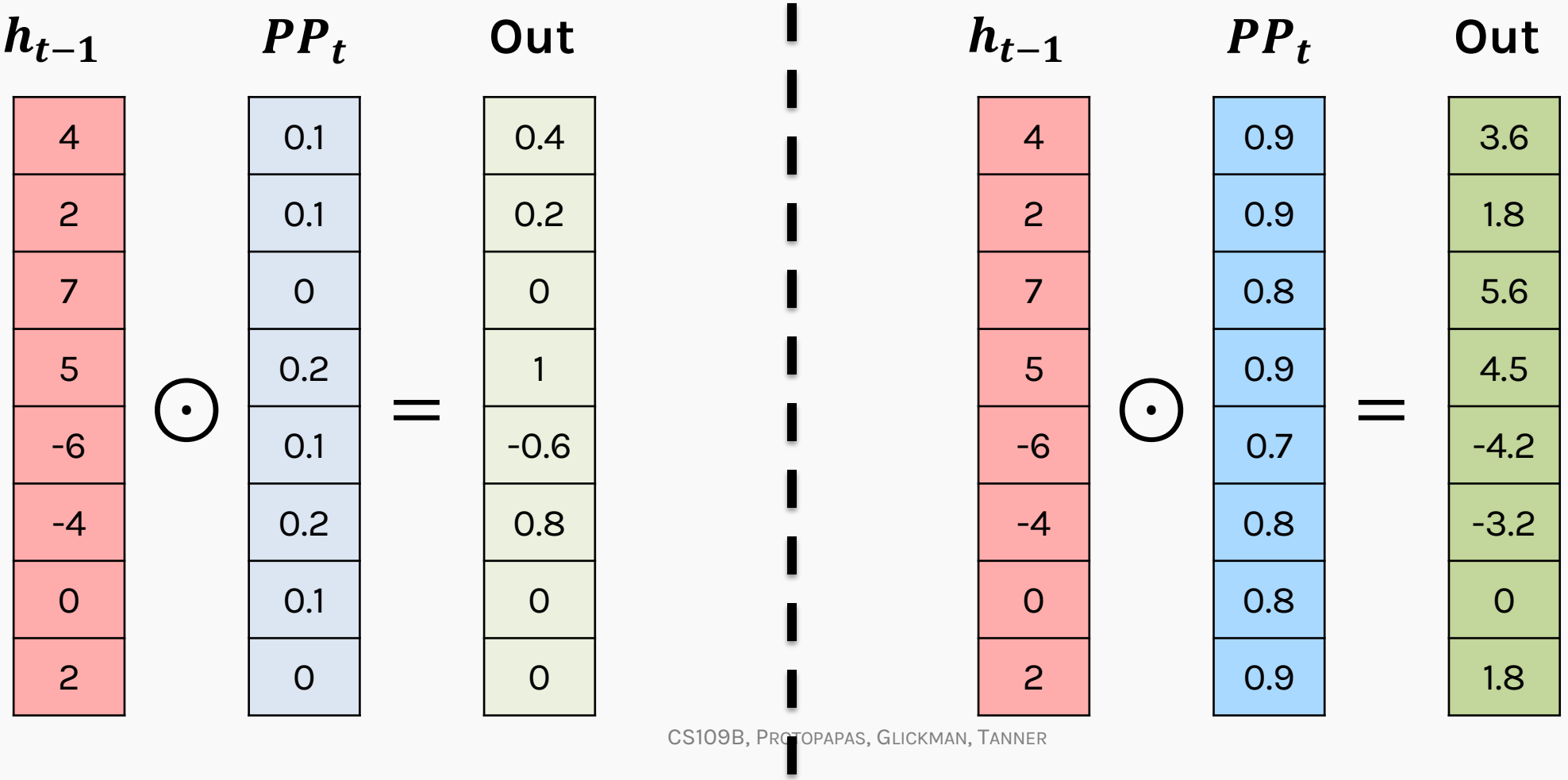
0	0	4	1
-1	1	2	1
0	2	4	0
1	4	7	4

$=$

0	0	4	-5
-4	3	4	6
0	4	4	0
0	0	0	4

Gates in RNNs

If the PP_t values are low, then h_t will depend mostly on current information (X_t), else it will consider the past information (h_{t-1}) as well



Pavlos Recurrent Unit: PRU

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{P}\mathbf{P}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

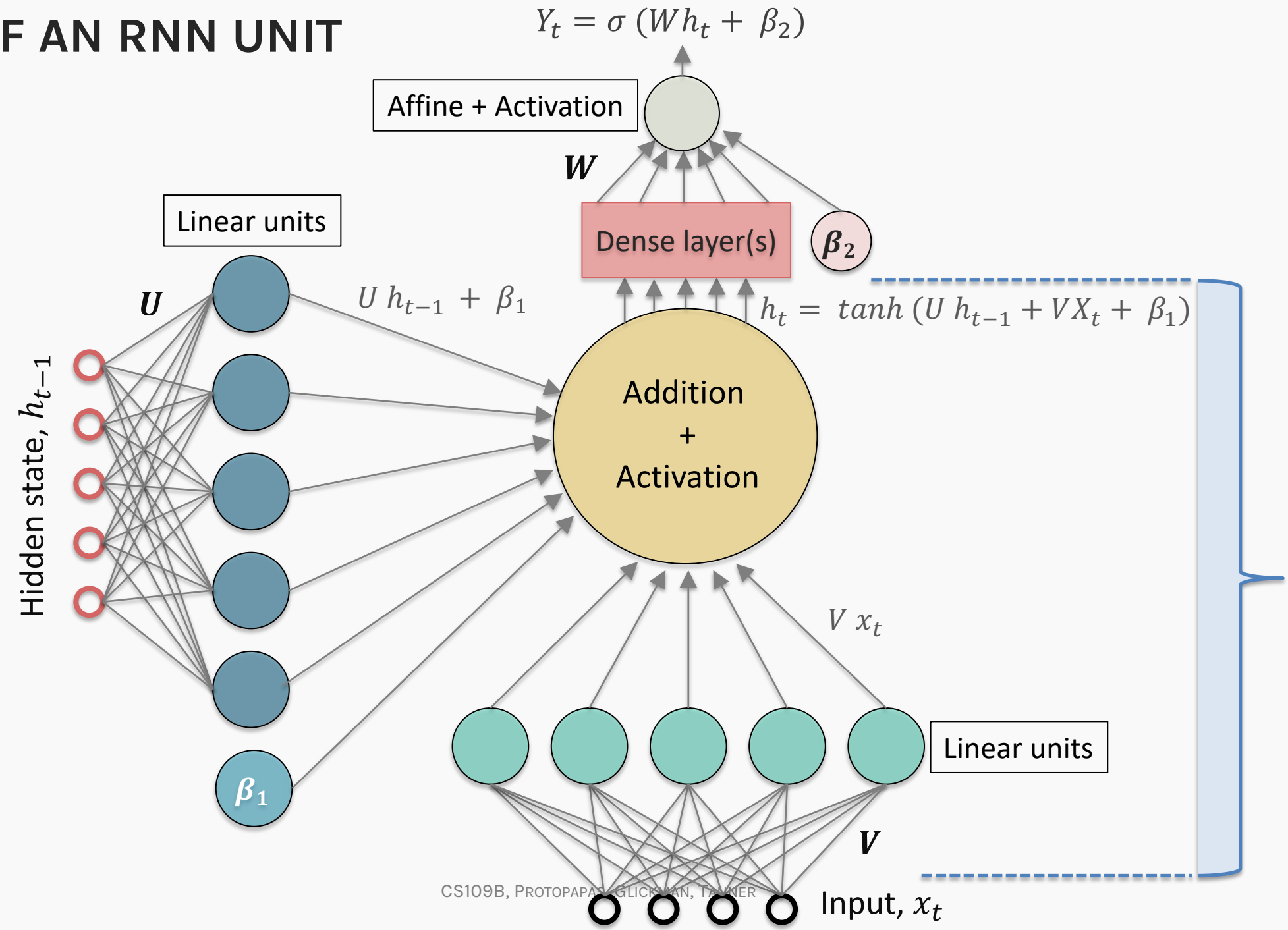
$$\mathbf{P}\mathbf{P}_t = \sigma\left(\mathbf{V}_{pp}\mathbf{X}_t + \mathbf{U}_{pp}\mathbf{h}_{t-1} + \beta_{pp}\right)$$

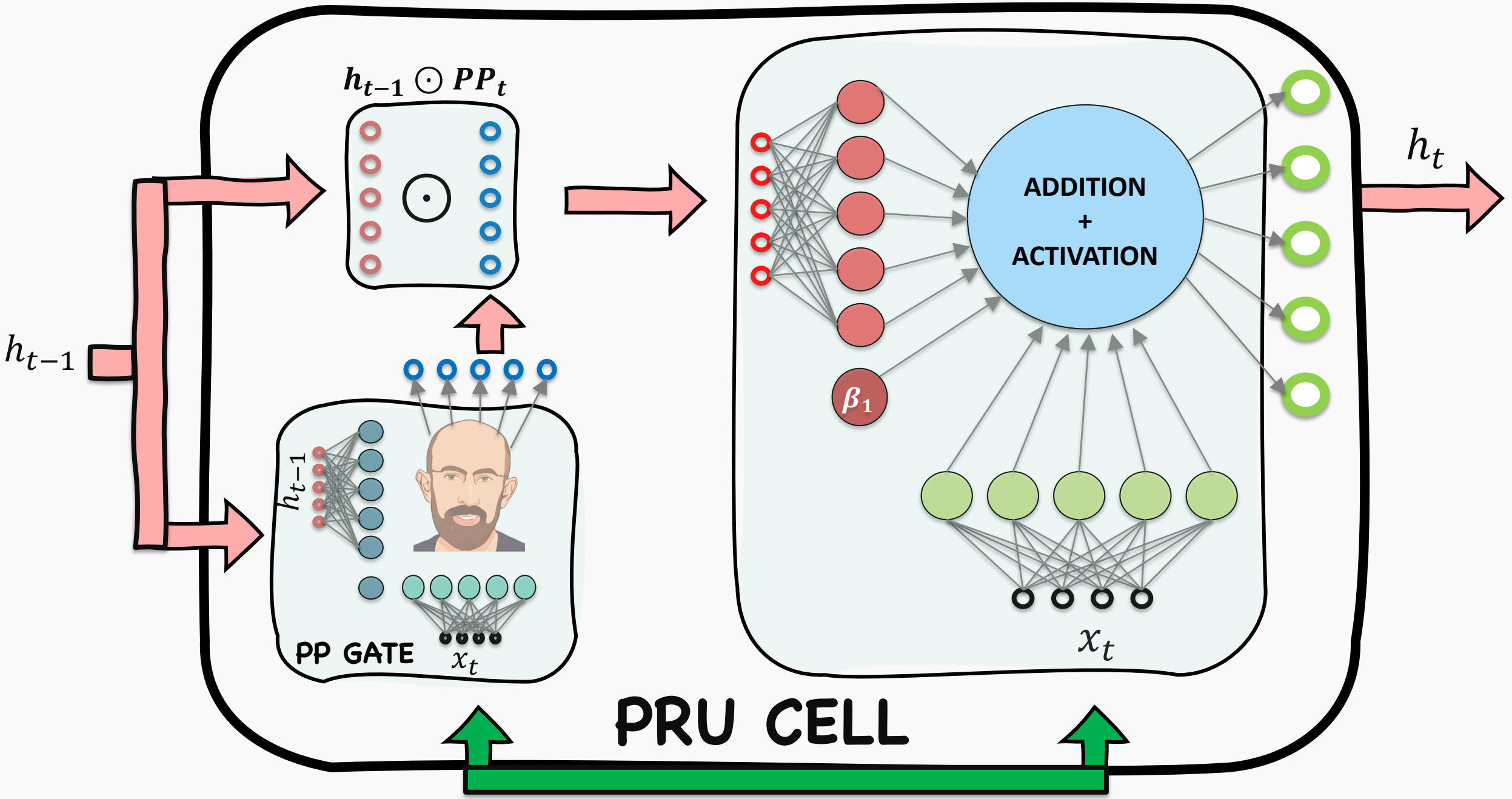
PRU's learnable parameters

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{PP}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

$$\mathbf{PP}_t = \sigma\left(\mathbf{V}_{pp}\mathbf{X}_t + \mathbf{U}_{pp}\mathbf{h}_{t-1} + \beta_{pp}\right)$$

ANATOMY OF AN RNN UNIT




$h_{t-1} \odot PP_t$
h_{t-1}
h_t
ADDITION
+
ACTIVATION
β_1
h_{t-1}
PP GATE
x_t
x_t
PRU CELL
INPUT

PRU final review **

$$\mathbf{h}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{P}\mathbf{P}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

$$\mathbf{P}\mathbf{P}_t = \sigma\left(\mathbf{V}_{pp}\mathbf{X}_t + \mathbf{U}_{pp}\mathbf{h}_{t-1} + \beta_{pp}\right)$$

PRU STRENGTHS?

- Current input can affect how much of the past information to consider
- This means we now can forget *irrelevant* past information

PRU ISSUES?

- Noisy inputs can severely affect the hidden memory
- Can still suffer from vanishing/exploding gradients

Leaky PRU

Idea #5: Use skip connections, aka as leaky units. Gradients can flow through the skip connection.

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{P}\mathbf{P}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

$$\mathbf{h}_t = \alpha \mathbf{h}_{t-1} + (1-\alpha)\tilde{\mathbf{h}}_\mathbf{t}$$

$\alpha \in [0,1]$ decides the amount of past information to carry over.

More details in a-sec2

Leaky PRU ***

Leaky PRU STRENGTHS?

- Vanishing gradient problem is diminished because of *skip* connections
- Hidden state more robust to outlier inputs because of α hyper-parameter

Leaky PRU ISSUES?

- The network performance is heavily dependent on the choice of the hyper-parameter α
- A fixed value of α restricts network from adaptively learning long term dependencies

Outline

RNN review

RNN shortcomings

Pavlos Recurrent Unit(PRU)

Gated Recurrent Unit (GRU)

What if we could adaptively learn α based on the input X_t and the previous hidden state h_{t-1}?

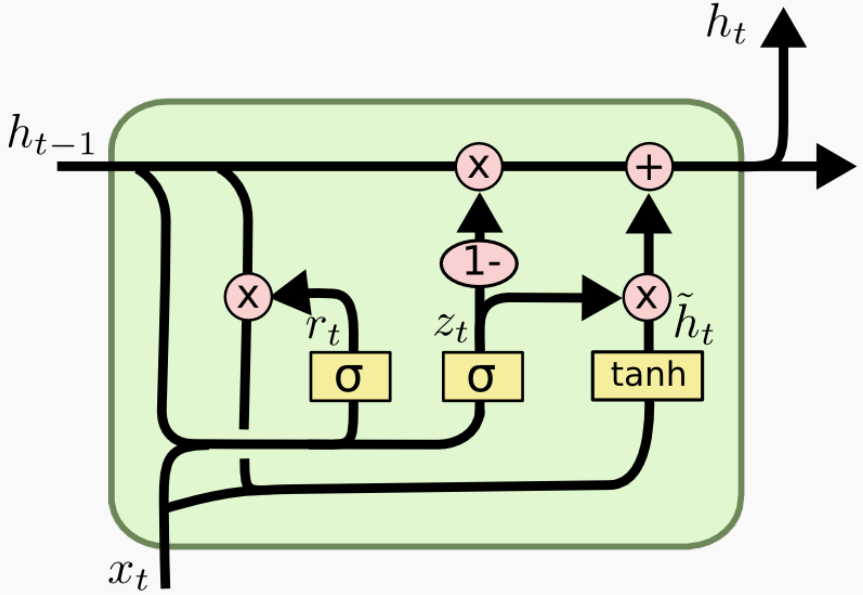
What if we could adaptively learn α based on the input X_t and the previous hidden state h_{t-1}?

Don't worry Pavlos, my minions will fix it!

Gated Recurrent Unit (GRU)

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{R}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

$$\mathbf{h}_t = \mathbf{Z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{h}}_\mathbf{t}$$



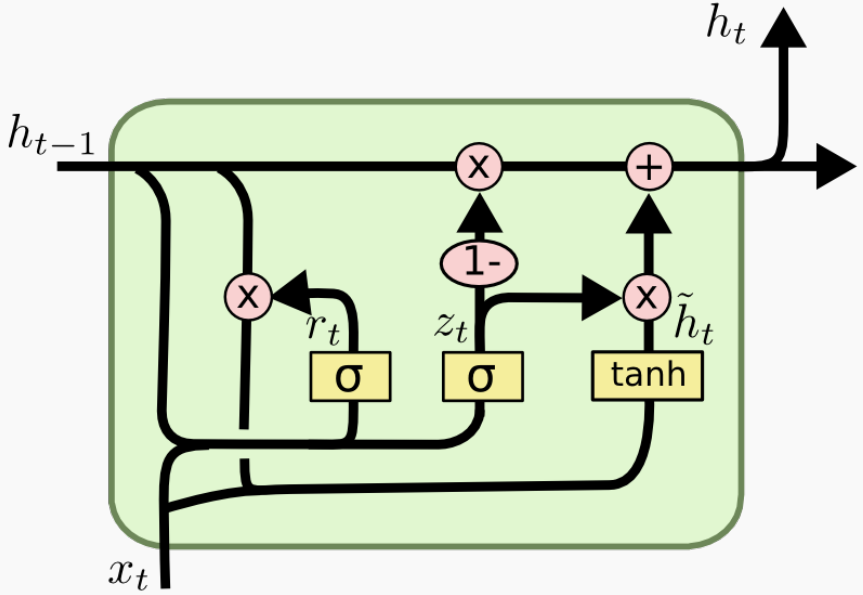
$$\mathbf{R}_t = \sigma\left(\mathbf{V}_R\mathbf{X}_t + \mathbf{U}_R\mathbf{h}_{t-1} + \beta_R\right)$$

$$\mathbf{Z}_t = \sigma\left(\mathbf{V}_Z\mathbf{X}_t + \mathbf{U}_Z\mathbf{h}_{t-1} + \beta_Z\right)$$

Gated Recurrent Unit (GRU): Learnable parameters

$$\tilde{\mathbf{h}}_t = \tanh\left(\mathbf{V}\mathbf{X}_t + \mathbf{U}\left[\mathbf{R}_t \odot \mathbf{h}_{t-1}\right] + \beta_1\right)$$

$$\mathbf{h}_t = \mathbf{Z}_t \odot \mathbf{h}_{t-1} + (1 - \mathbf{Z}_t) \odot \tilde{\mathbf{h}}_\mathbf{t}$$



Reset Gate (equivalent to PP gate)

$$\mathbf{R}_t = \sigma\left(\mathbf{V}_R\mathbf{X}_t + \mathbf{U}_R\mathbf{h}_{t-1} + \beta_R\right)$$

$$\mathbf{Z}_t = \sigma\left(\mathbf{V}_Z\mathbf{X}_t + \mathbf{U}_Z\mathbf{h}_{t-1} + \beta_Z\right)$$

Update Gate

Final Remarks

- We will investigate the specific architecture of Vanilla LSTM in the next class

Final Remarks

- We will investigate the specific architecture of Vanilla LSTM in the next class

- However, the central ideas revolve around:
  - Making trainable weights sensitive to inputs to improve context
  - Creating skip-connections to minimize vanishing gradients

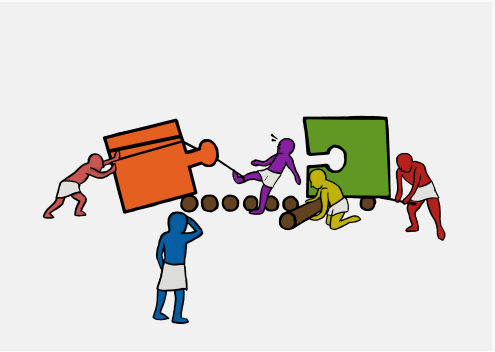
Final Remarks

- We will investigate the specific architecture of Vanilla LSTM in the next class

- However, the central ideas revolve around:
  - Making trainable weights sensitive to inputs to improve context
  - Creating skip-connections to minimize vanishing gradients

- The various architectures & variants aim to achieve these two goals

Exercise: Vanishing Gradients

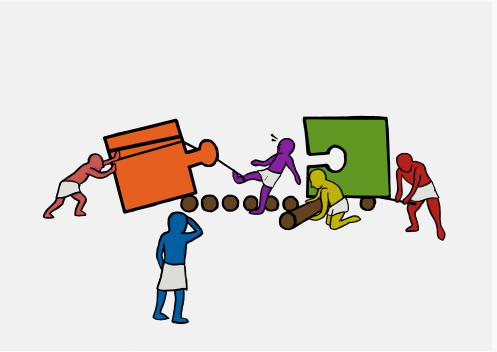


Is vanishing gradients real or a hoax?

The goal of this exercise is to decide yourself if it is for real or not!

We use IMDB movie review dataset to perform sentiment analysis and check post and pre padding and examine the effect of vanishing gradients in these two cases.

Exercise: Vanishing Gradients



The goal of this exercise is to build the Pavlos Recurrent Unit.

You will for the first time in CS109 learn how to build a custom layer.

Knowing how to build custom layers is yet another important skill.

Another key to the kingdom of deep learning.