GRUs

Pavlos Protopapas

Outline

- RNN shortcomings
- PRU
- Gated Recurrent Unit (GRU)

Outline

- **RNN shortcomings**
- PRU
- Gated Recurrent Unit (GRU)

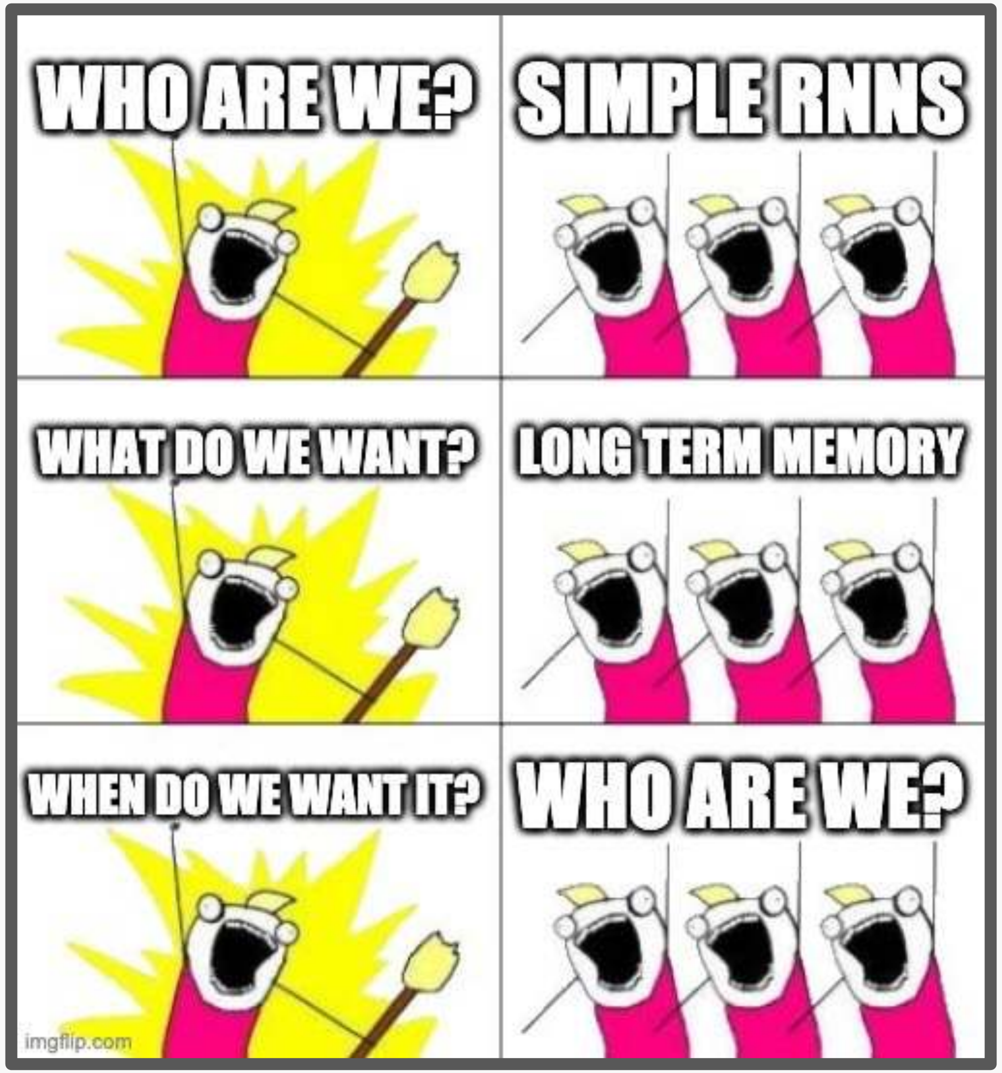
RNN shortcomings

RNNs Wishlist

RNNs should exhibit the following advantages for sequence modelling:

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

RNN shortcomings



RNN shortcomings: Problem 1

$$\boxed{h_t = \tanh(V X_t + U h_{t-1} + \beta_1)}$$

Current Input

Previous hidden state

RNN shortcomings: Problem 1

$$\boxed{h_t = \tanh(V X_t + U h_{t-1} + \beta_1)}$$

Input weights

Previous hidden state weights

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

RNN shortcomings: Problem 1

A very long email:

Hello Professor Protopapas,
My name is Germán 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.

RNN shortcomings: Problem 1

A very long email:

Hello Professor Protopapas,
My name is Germán 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.

RNN shortcomings: Problem 1

A very long email:

Hi Pavlos,
Varshini 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.

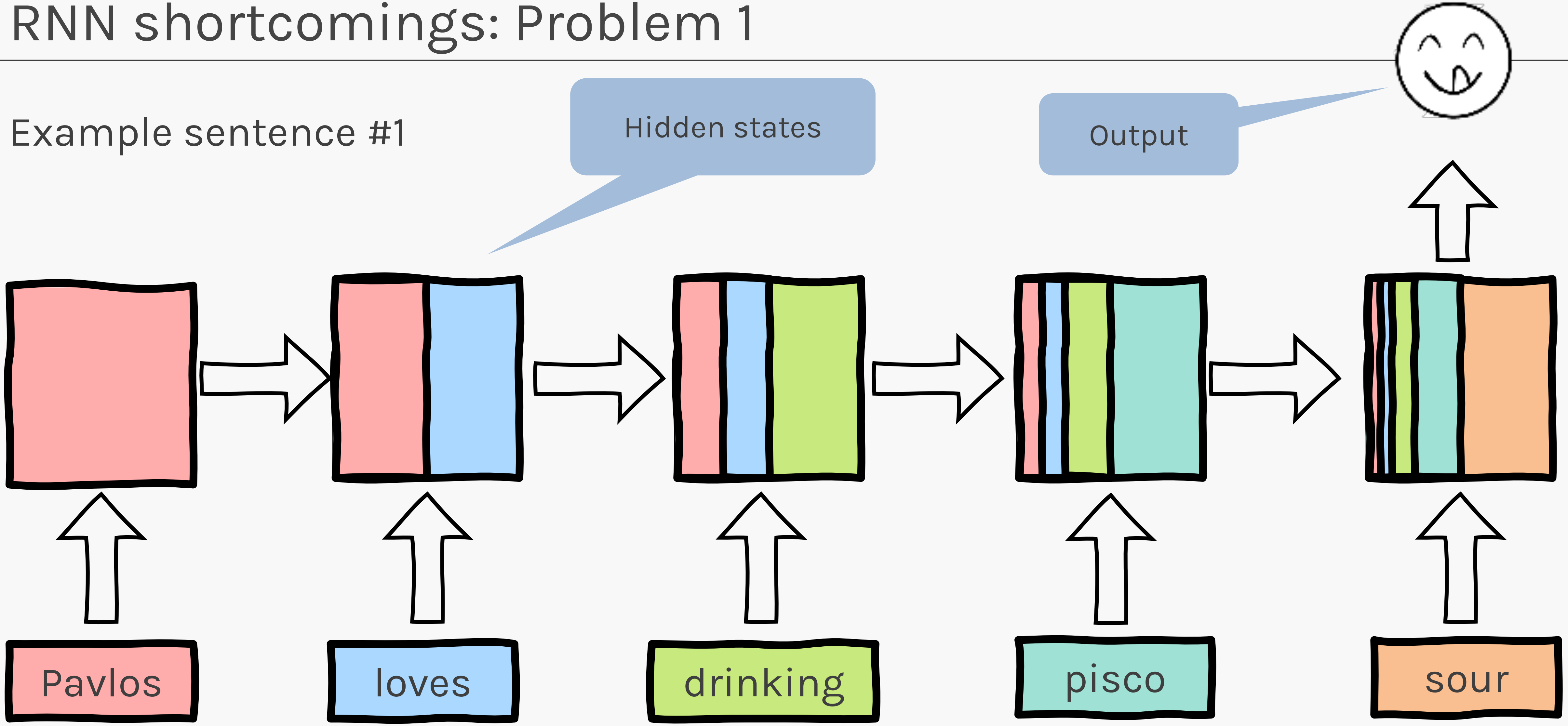
RNN shortcomings: Problem 1

A very long email:

Hi Pavlos,
Varshini 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.

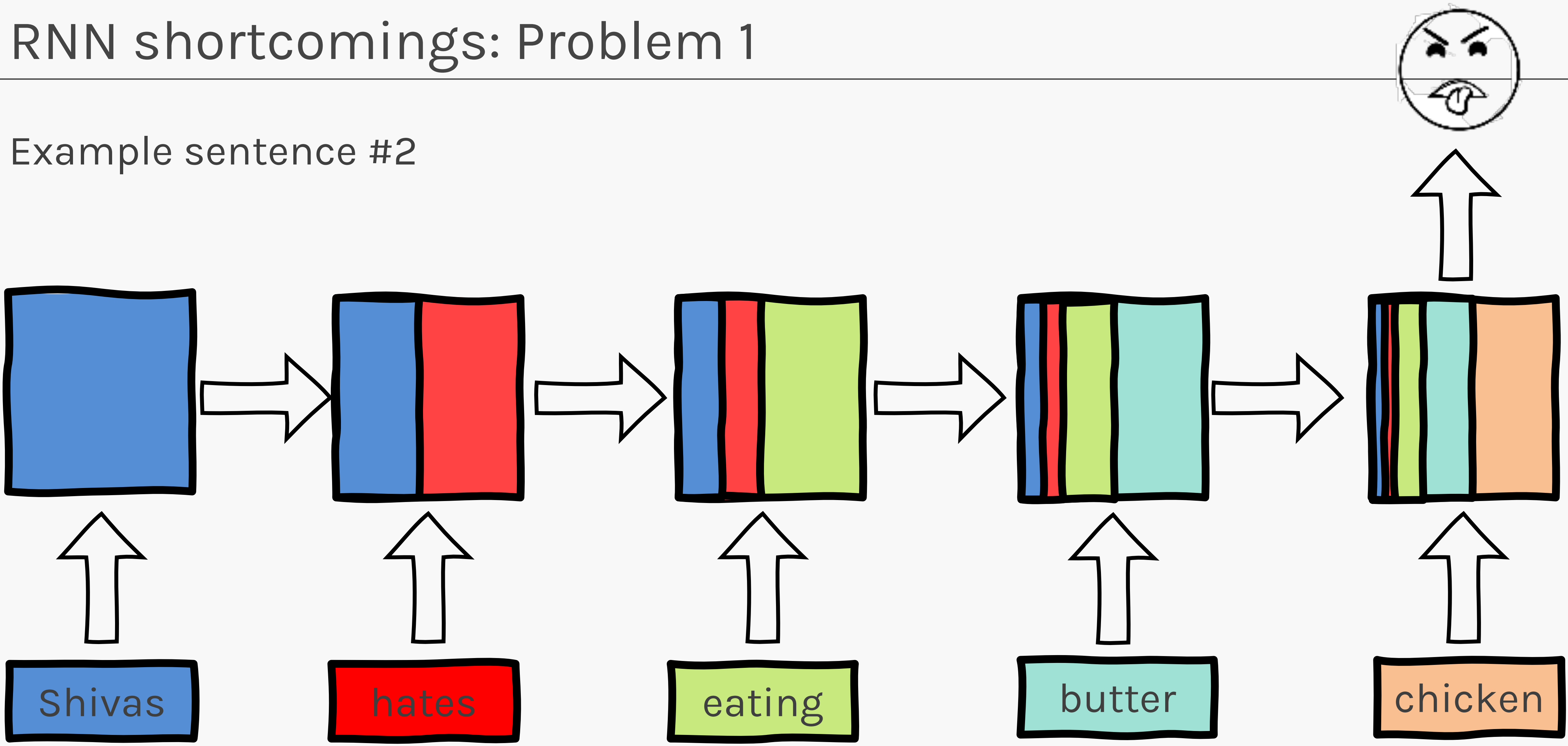
RNN shortcomings: Problem 1

Example sentence #1



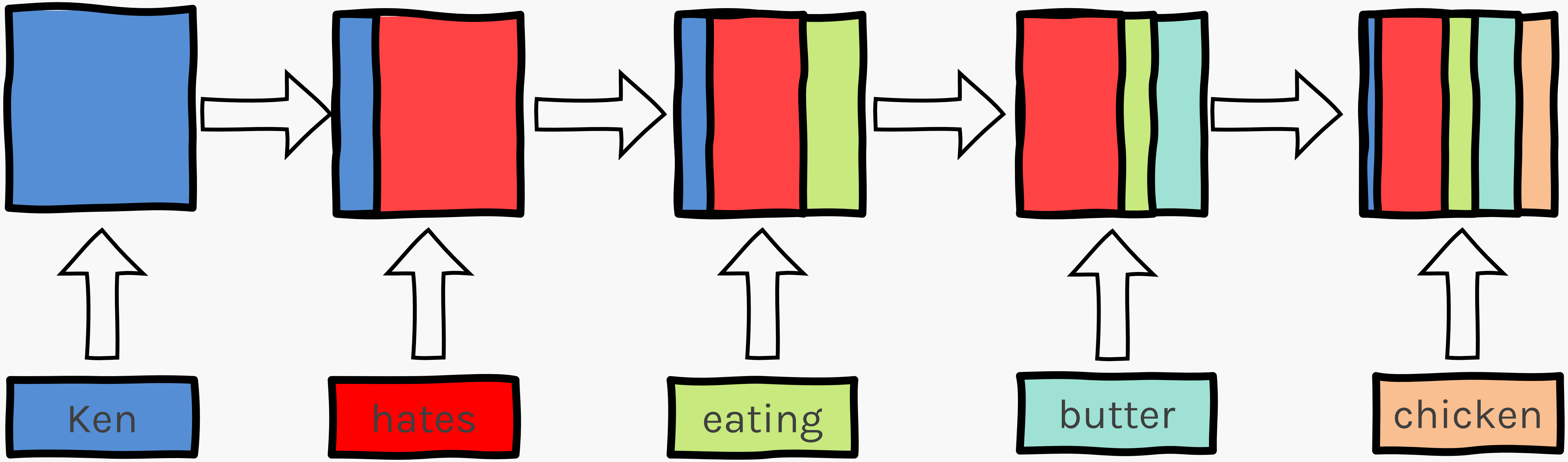
RNN shortcomings: Problem 1

Example sentence #2



RNN shortcomings: Problem 1

Example sentence #2 when U and V depends on x_t and h_{t-1}.

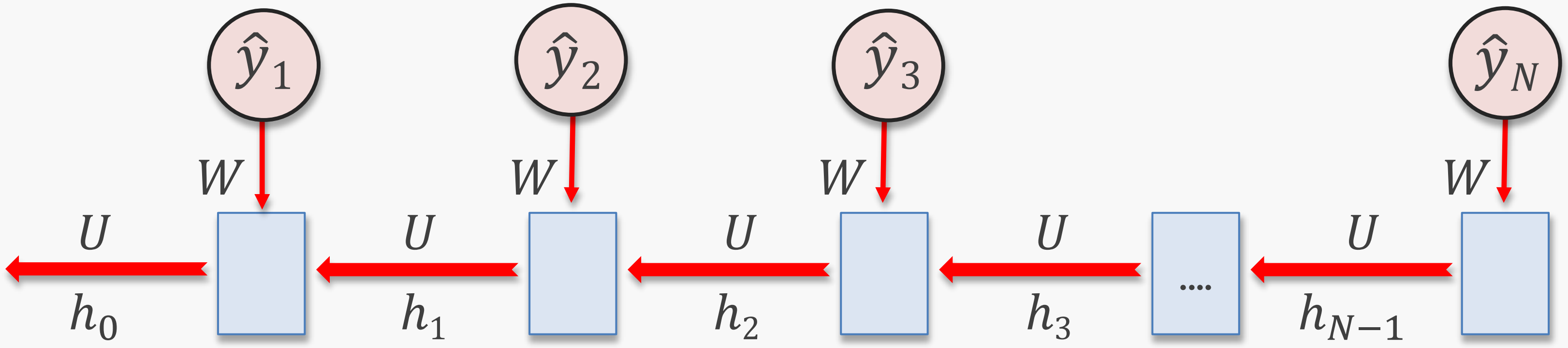


RNN shortcomings: Problem 1

$$\boxed{h_t = \tanh(V(h_{t-1}, X_t)X_t + U(h_{t-1}, X_t)h_{t-1} + \beta_1)}$$

We want our trainable weights V, U to somehow incorporate the input X_t and the previous state h_{t-1}.

RNN shortcomings: Problem 2



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.

Outline

- RNN shortcomings
- **PRU**
- Gated Recurrent Unit (GRU)

PRU

We wanted U and V to be a function of input X_t or previous state h_{t-1} :

$$\boxed{h_t = \tanh(V(h_{t-1}, X_t)X_t + U(h_{t-1}, X_t)h_{t-1} + \beta_1)}$$

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

$$\boxed{h_t = \tanh(VX_t + U(h_{t-1}, X_t)h_{t-1} + \beta_1)}$$

PRU

Idea #1:

$$\boxed{h_t = \tanh(VX_t + U(h_{t-1}, X_t)h_{t-1} + \beta_1)}$$

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

$$\boxed{h_t = \tanh(VX_t + U[PP(h_{t-1}, X_t)h_{t-1}] + \beta_1)}$$

PRU

Idea #2:

$$h_t = \tanh(VX_t + U[PP(h_{t-1}, X_t)h_{t-1}] + \beta_1)$$

Idea #3: Use **element wise multiplication** so we not mix different hidden state elements

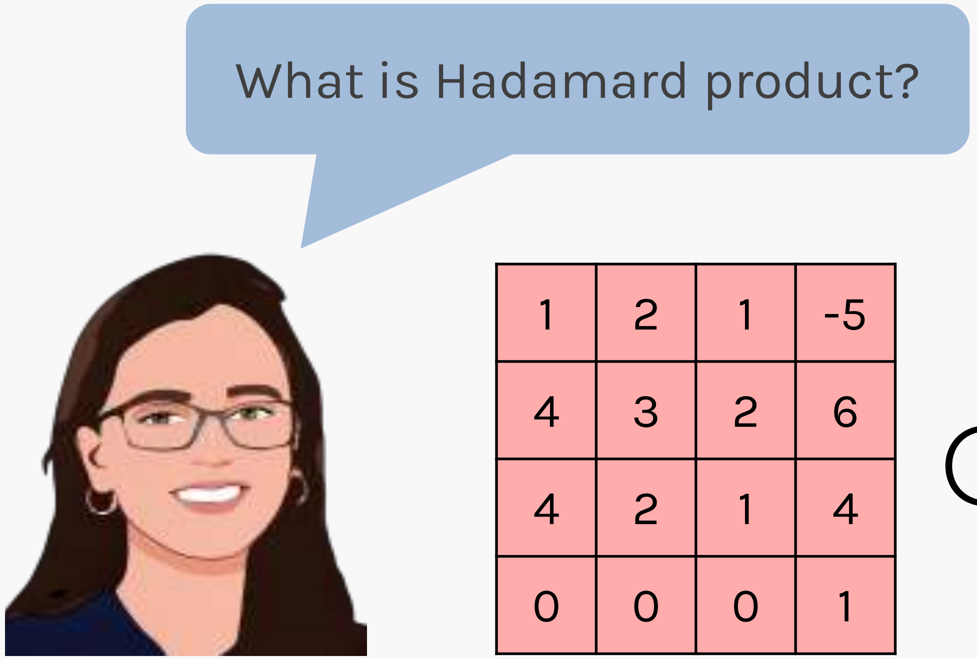
Hadamard product

$$h_t = \tanh(VX_t + U[PP(h_{t-1}, X_t) \odot h_{t-1}] + \beta_1)$$

PRU

Idea #3:

$$\boxed{h_t = \tanh(VX_t + U[PP(h_{t-1}, X_t) \odot h_{t-1}] + \beta_1)}$$



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

$$\begin{bmatrix} 1 & 2 & 1 & -5 \\ 4 & 3 & 2 & 6 \\ 4 & 2 & 1 & 4 \\ 0 & 0 & 0 & 1 \end{bmatrix} \odot \begin{bmatrix} 0 & 0 & 4 & 1 \\ -1 & 1 & 2 & 1 \\ 0 & 2 & 4 & 0 \\ 1 & 4 & 7 & 4 \end{bmatrix} = \begin{bmatrix} 0 & 0 & 4 & -5 \\ -4 & 3 & 4 & 6 \\ 0 & 4 & 4 & 0 \\ 0 & 0 & 0 & 4 \end{bmatrix}$$

PRU

Idea #3:

$$h_t = \tanh(VX_t + U[PP(h_{t-1}, X_t) \odot h_{t-1}] + \beta_1)$$

Now, let's give a name to the PP variable: PP-gate and shorter the notation.
PP gate decides the amount of past information to be considered for the hidden state.

$$h_t = \tanh(VX_t + U[PP_t \odot h_{t-1}] + \beta_1)$$

PRU

But what is this PP gate?

For a given timestep t, if:

- input $X_t \in \mathbb{R}^{d \times 1}$
- hidden state in the previous timestep, $h_{t-1} \in \mathbb{R}^{h \times 1}$

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

$$\boxed{PP_t = \sigma(V_{pp} X_t + U_{pp}\, h_{t-1} + \beta_{pp})}$$

PRU

Sigmoid activation makes output between 0 and 1

$$\boxed{PP_t = \sigma(\underbrace{V_{pp}}_{\mathbb{R}^{\boldsymbol{h \times d}}} \overbrace{X_t}^{\mathbb{R}^{\boldsymbol{d \times 1}}} + \underbrace{U_{pp}}_{\mathbb{R}^{\boldsymbol{h \times h}}}\ \overbrace{h_{t-1}}^{\mathbb{R}^{\boldsymbol{h \times 1}}} +\ \overbrace{\beta_{pp}}^{\mathbb{R}^{\boldsymbol{h \times 1}}})}$$

PRU

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

$$
\begin{array}{ccccc}
\boldsymbol{h_{t-1}} & & \boldsymbol{PP_t} & & \textbf{Out} \\
\begin{bmatrix} 4 \\ 2 \\ 7 \\ 5 \\ -6 \\ -4 \\ 0 \\ 2 \end{bmatrix} & \odot & \begin{bmatrix} 0.1 \\ 0.1 \\ 0 \\ 0.2 \\ 0.1 \\ 0.2 \\ 0.1 \\ 0 \end{bmatrix} & = & \begin{bmatrix} 0.4 \\ 0.2 \\ 0 \\ 1 \\ -0.6 \\ 0.8 \\ 0 \\ 0 \end{bmatrix}
\end{array}
$$

$$
\begin{array}{ccccc}
\boldsymbol{h_{t-1}} & & \boldsymbol{PP_t} & & \textbf{Out} \\
\begin{bmatrix} 4 \\ 2 \\ 7 \\ 5 \\ -6 \\ -4 \\ 0 \\ 2 \end{bmatrix} & \odot & \begin{bmatrix} 0.9 \\ 0.9 \\ 0.8 \\ 0.9 \\ 0.7 \\ 0.8 \\ 0.8 \\ 0.9 \end{bmatrix} & = & \begin{bmatrix} 3.6 \\ 1.8 \\ 5.6 \\ 4.5 \\ -4.2 \\ -3.2 \\ 0 \\ 1.8 \end{bmatrix}
\end{array}
$$

PRU

Now we have:

$$\boxed{h_t = \tanh(\boxed{V} X_t + \boxed{U}[PP_t \odot h_{t-1}] + \boxed{\beta_1})}$$

$$\boxed{PP_t = \sigma(\boxed{V_{pp}} X_t + \boxed{U_{pp}} h_{t-1} + \boxed{\beta_{pp}})}$$

GAME Time

Quiz Time

What does the name PRU stand for?

A. Progressive Recurrent Unit
B. Passive Recurrent Unit
C. Pavlos Recurrent Unit
D. Pathetic Recurrent Unit

Quiz Time

What does the name PRU stand for?

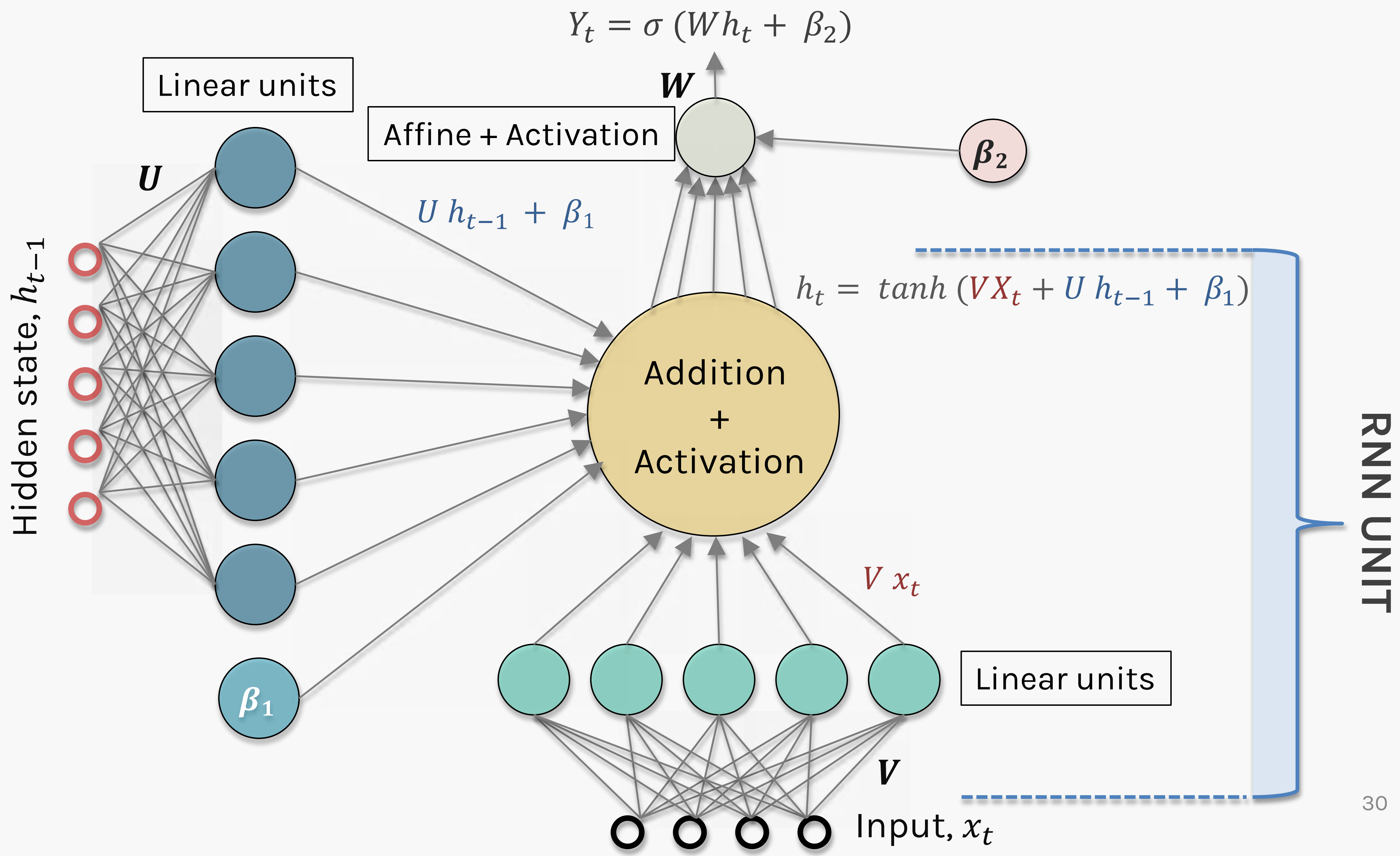
A. Progressive Recurrent Unit

B. Passive Recurrent Unit

C. Pavlos Recurrent Unit

D. Pathetic Recurrent Unit

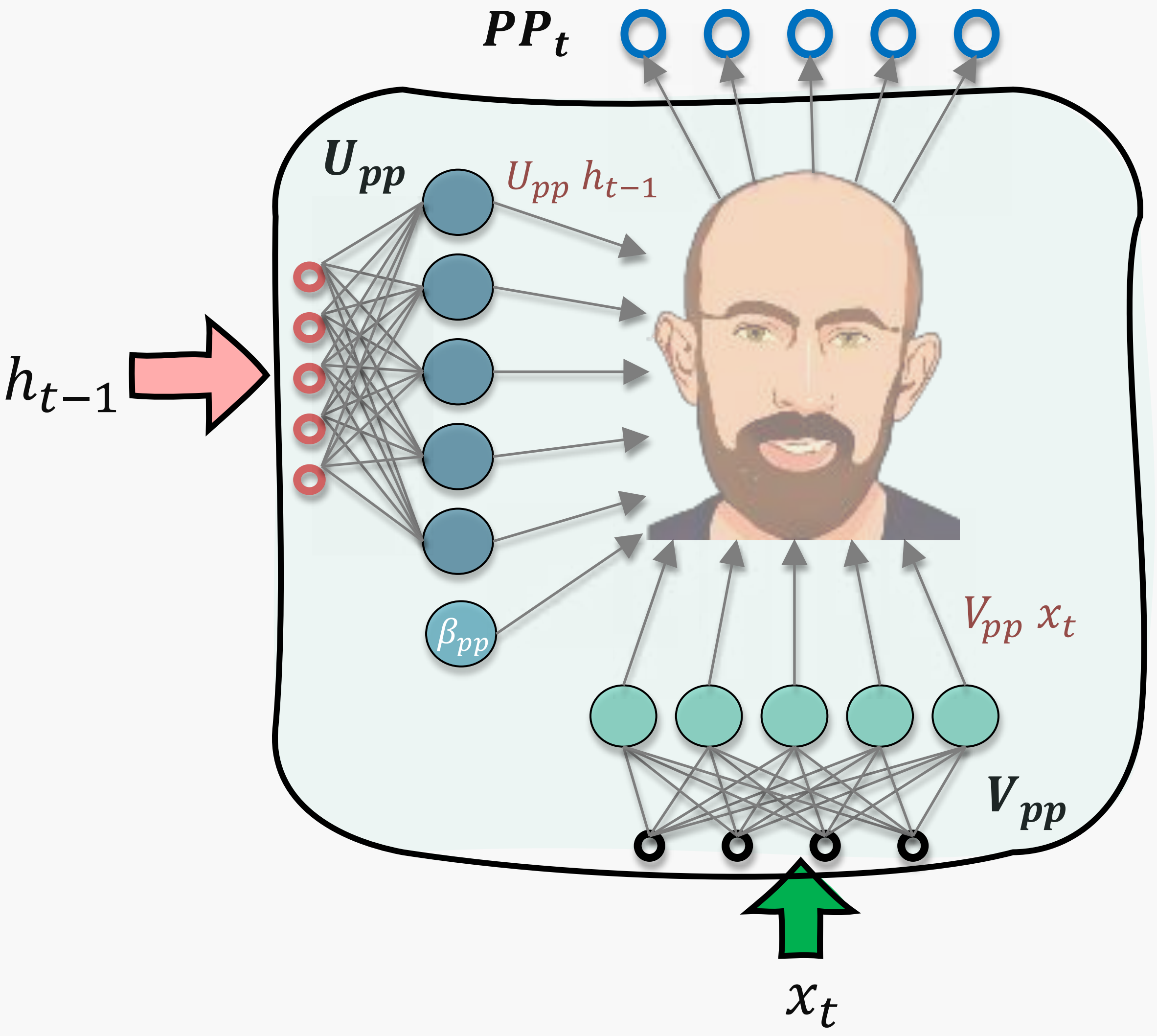
PRU



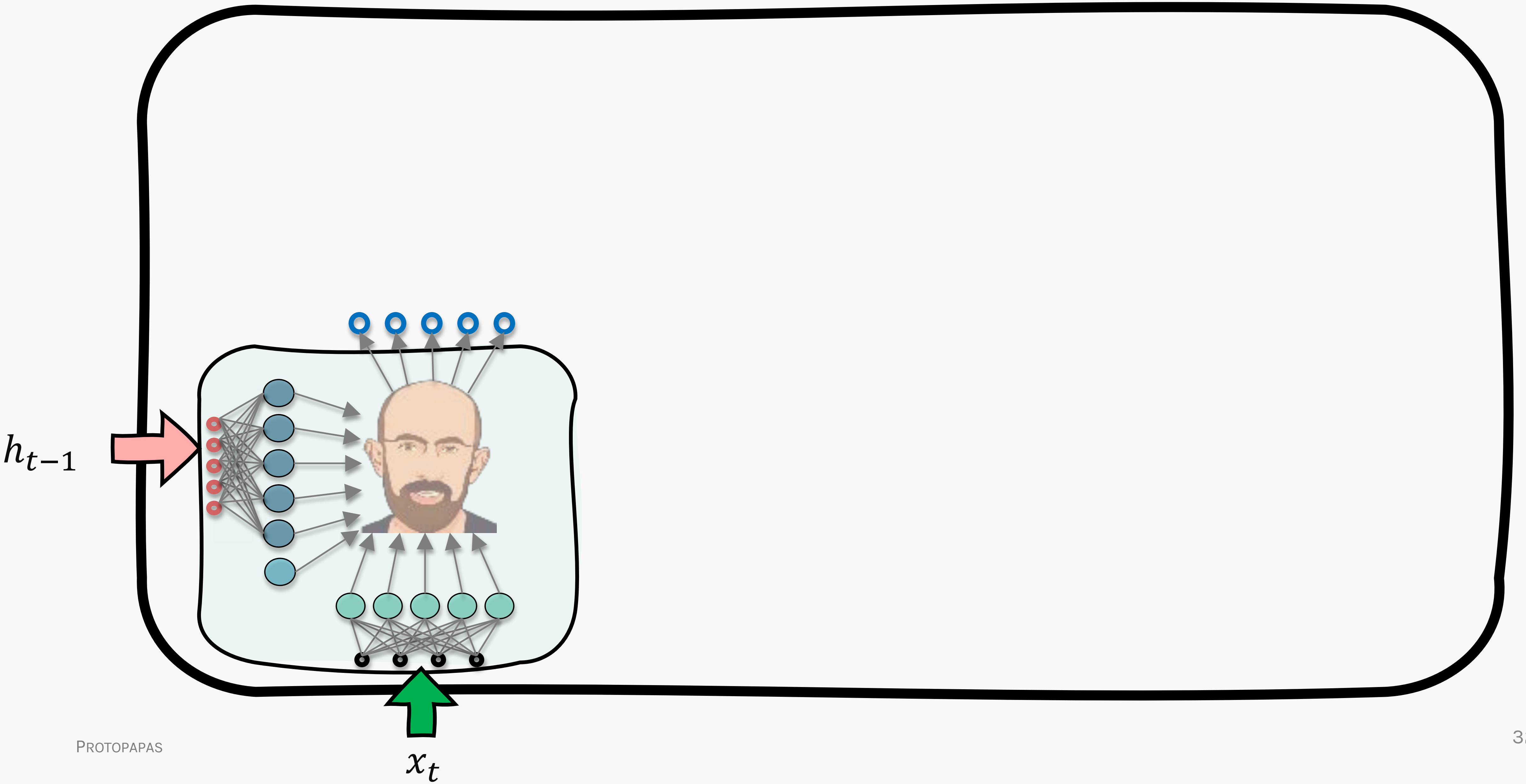
$Y_t = \sigma\,(W h_t + \beta_2)$
Linear units
W
Affine + Activation
β_2
U
$U\, h_{t-1} + \beta_1$
$h_t = tanh\,(V X_t + U\, h_{t-1} + \beta_1)$
Hidden state, h_{t-1}
Addition
+
Activation
RNN UNIT
$V\ x_t$
β_1
Linear units
V
Input, x_t

PRU

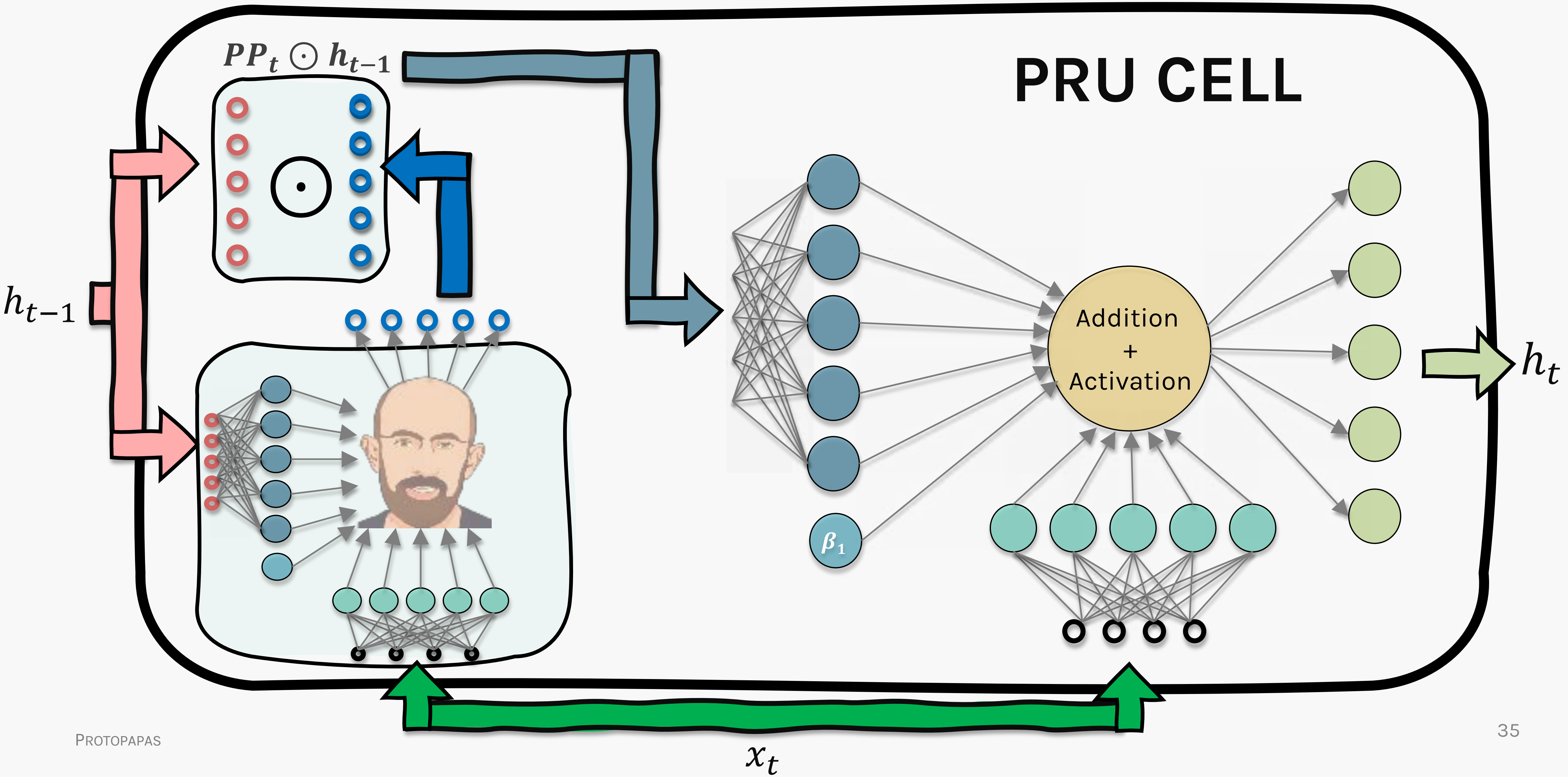
$$PP_t = \sigma(V_{pp}X_t + U_{pp}\ h_{t-1} + \beta_{pp})$$

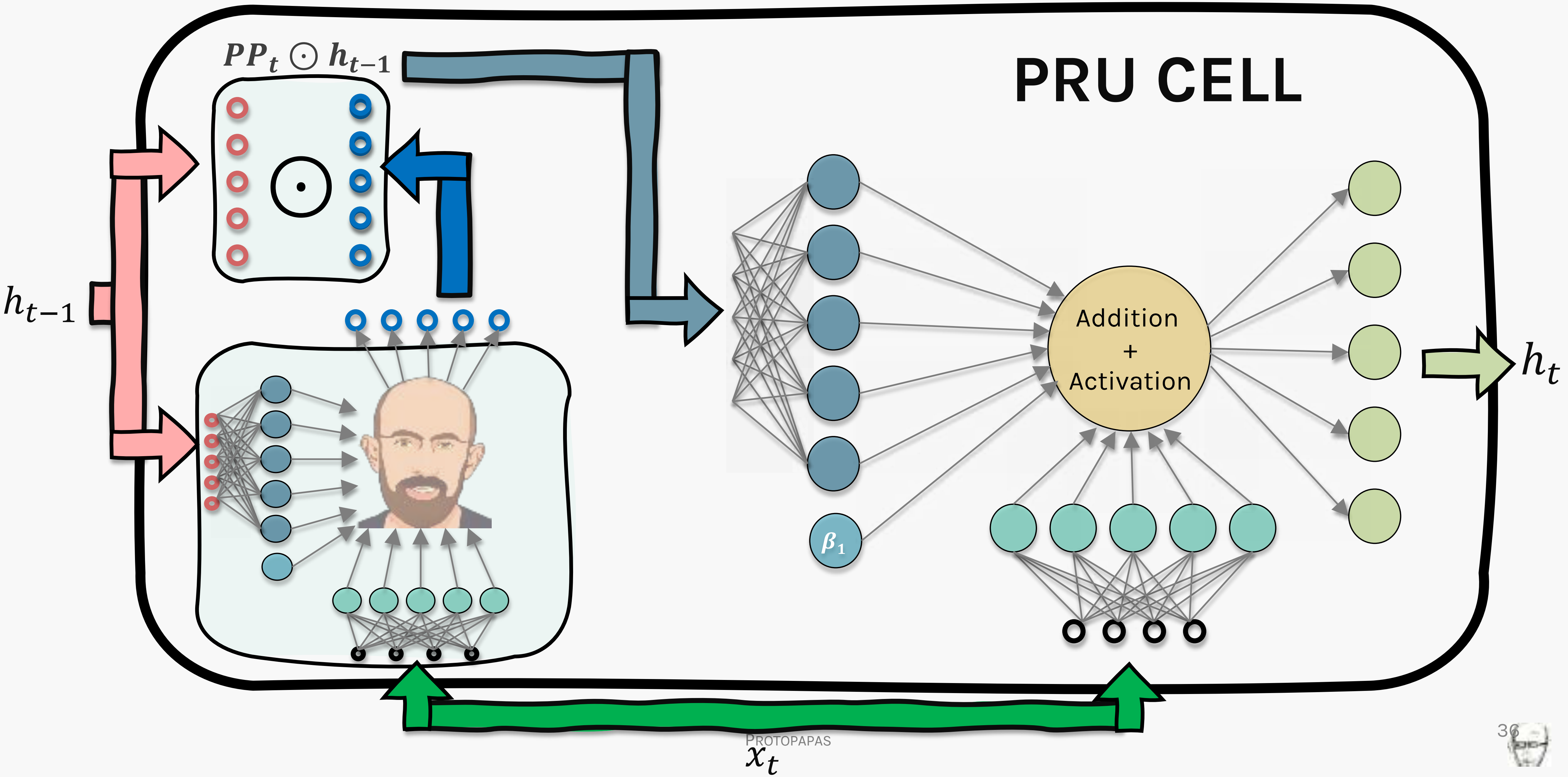


Pavlos Recurrent Unit (PRU)



Pavlos Recurrent Unit (PRU)




PRU CELL
$PP_t \odot h_{t-1}$
h_{t-1}
h_t
x_t
β_1
Addition
+
Activation

Pavlos Recurrent Unit (PRU)

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

Pavlos Recurrent Unit (PRU)

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

$$\tilde{h}_t = \tanh(VX_t + U[PP_t \odot h_{t-1}] + \beta_1)$$

$$h_t = \alpha h_{t-1} + (1-\alpha)\tilde{h}_t$$

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

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 shortcomings
- Pavlos Recurrent Unit (PRU)
- **Gated Recurrent Unit (GRU)**

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}?

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

Gated Recurrent Unit (GRU)

$$\boxed{\tilde{h}_t = \tanh(VX_t + U[\,PP_t \odot h_{t-1}] + \beta_1)}$$

Change PP_t to more conventional name.

Gated Recurrent Unit (GRU)

$$\boxed{\tilde{h}_t = \tanh(VX_t + U[\,R_t \odot h_{t-1}] + \beta_1)}$$

$$\boxed{h_t = \alpha h_{t-1} + (1-\alpha)\tilde{h}_t}$$

Change α to learnable => a gate.

Gated Recurrent Unit (GRU)

$$\boxed{\tilde{h}_t = \tanh(VX_t + U[\,R_t \odot h_{t-1}] + \beta_1)}$$

$$\boxed{h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot \tilde{h}_t}$$

$$\boxed{R_t = \sigma(V_R X_t + U_R h_{t-1} + \beta_R)}$$

$$\boxed{Z_t = \sigma(V_Z X_t + U_Z h_{t-1} + \beta_Z)}$$

Gated Recurrent Unit (GRU)

$$\tilde{h}_t = \tanh(VX_t + U[\,R_t \odot h_{t-1}] + \beta_1)$$

$$h_t = Z_t \odot h_{t-1}$$

Reset gate,
equivalent to PP gate

$$R_t = \sigma(V_R X_t + U_R h_{t-1} + \beta_R)$$

$$Z_t = \sigma(V_Z X_t + U_Z h_{t-1} + \beta_Z)$$

Update gate

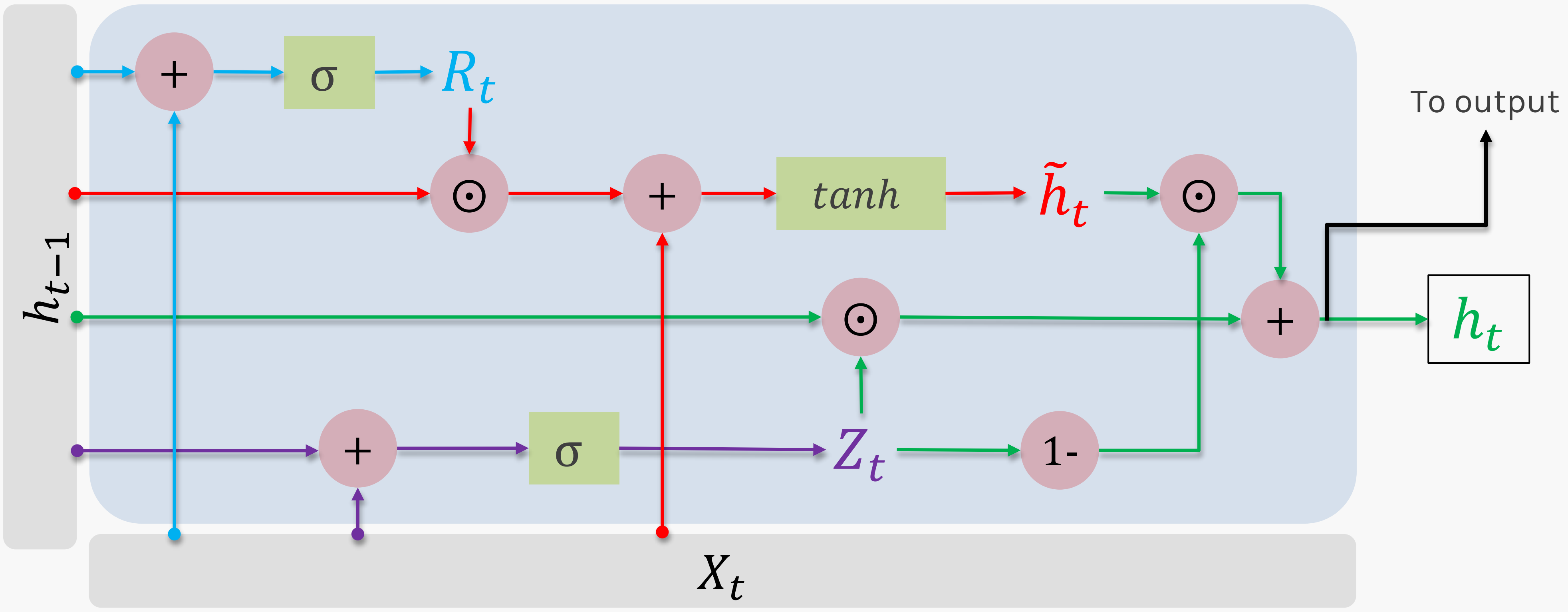
Gated Recurrent Unit (GRU)

$$\tilde{h}_t = \tanh(VX_t + U[\,R_t \odot h_{t-1}] + \beta_1)$$

$$h_t = Z_t \odot h_{t-1} + (1 - Z_t) \odot \tilde{h}_t$$

$$R_t = \sigma(V_R X_t + U_R h_{t-1} + \beta_R)$$

$$Z_t = \sigma(V_Z X_t + U_Z h_{t-1} + \beta_Z)$$



Final Remarks

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

Final Remarks

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

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

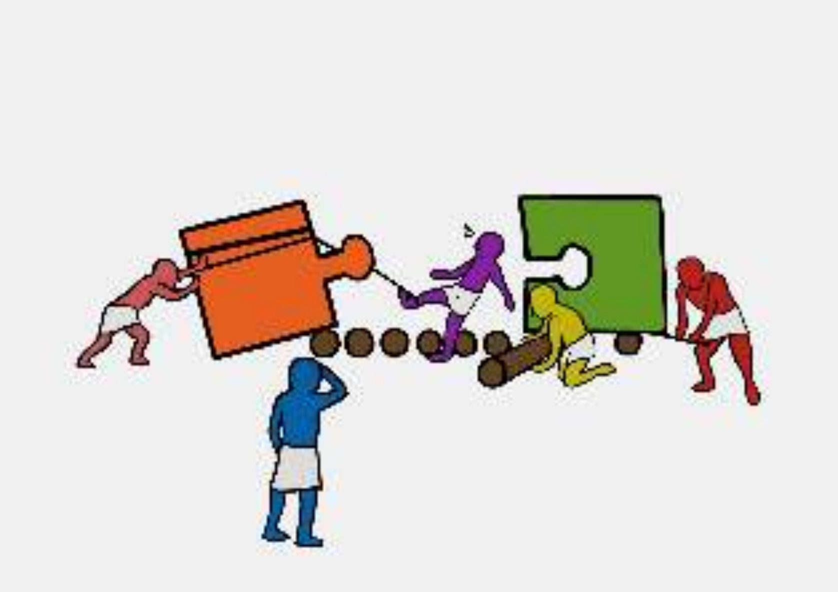
Final Remarks

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

- 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

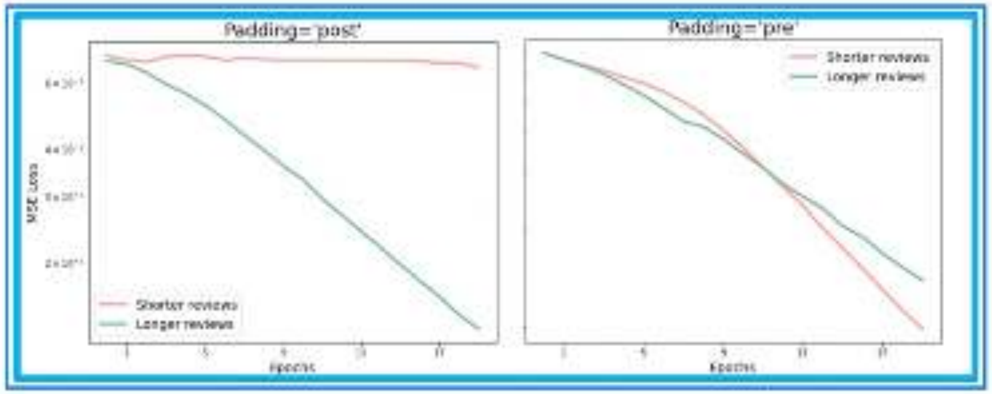


🏋 Exercise: Vanishing Gradients

The goal of this exercise is to understand the vanishing gradient problem in training RNNs and using various methods to improve training.

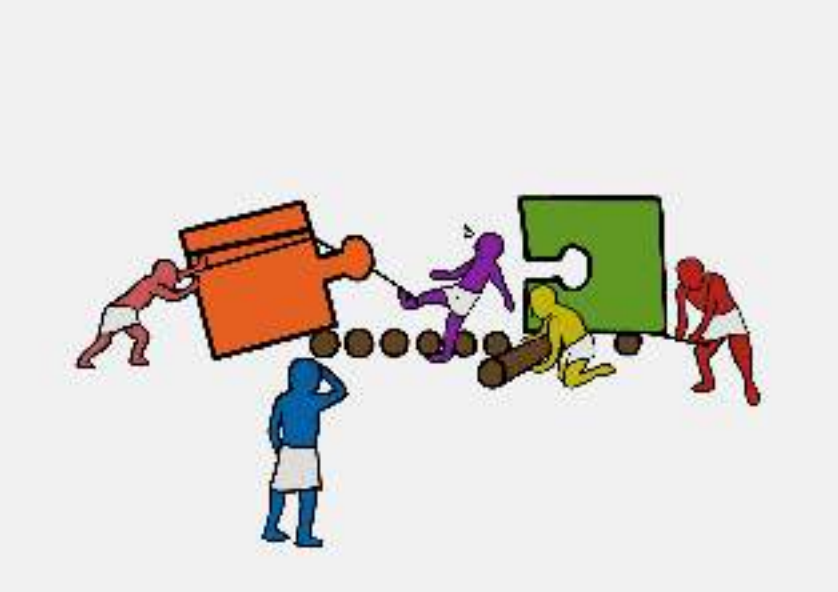
In order to do this exercise, we will use the IMDB movie review dataset to perform sentiment analysis.

Your final comparison for the trace plot may look something like this:



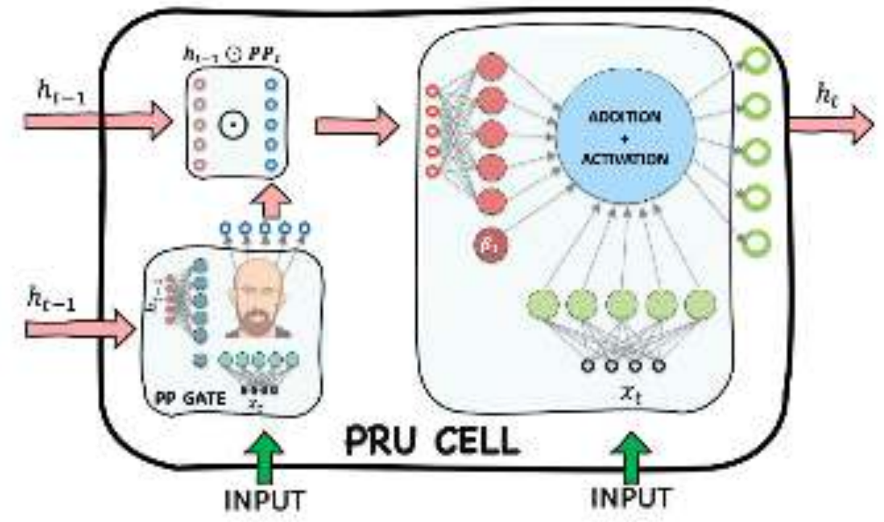
Instructions:

- Read the IMDB dataset from the helper code given.
- Take a quick look at your training inputs and labels.
- Pad the values to a fixed number `max_words` in-order to have sequences of the same size.
- First *post* pad the inputs with `padding='post'` i.e sequences smaller than `max_words` will be followed by zero padding.
- Build, compile and fit a Vanilla RNN and evaluate it on the test set.



Exercise: Pavlos Recurrent Unit

The goal of this exercise is to build the **Pavlos Recurrent Unit** discussed in class.



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

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

Alternative notation used in the exercise:

$$\mathbf{H}_t = \tanh\left(\mathbf{X}_t\mathbf{W}_{xh} + \left(\mathbf{PP}_t \odot \mathbf{H}_{t-1}\right)\mathbf{W}_{hh} + \mathbf{b}_h\right)$$

$$\mathbf{PP}_t = \sigma(\mathbf{X}_t\mathbf{W}_{xpp} + \mathbf{H}_{t-1}\mathbf{W}_{hpp} + \mathbf{b}_{pp})$$

THANK YOU