ALMA MATER STUDIORUM
UNIVERSITÀ DI BOLOGNA

DIPARTIMENTO DI
SCIENZE STATISTICHE "PAOLO FORTUNATI"

3 - Gated Recurrent Unit

Giovanni Della Lunga

giovanni.dellalunga@unibo.it

Halloween Conference in Quantitative Finance

Bologna - October 25-26, 2023

Introduction

Introduction

- Introduced by Cho, et al. in 2014, GRU (Gated Recurrent Unit) aims to solve the vanishing gradient problem which comes with a standard recurrent neural network.
- GRU can also be considered as a variation on the LSTM because both are designed similarly and, in some cases, produce equally excellent results.

Introduction

- To solve the vanishing gradient problem of a standard RNN, GRU uses, so-called, update gate and reset gate.
- Basically, these are two vectors which decide what information should be passed to the output.

Introduction

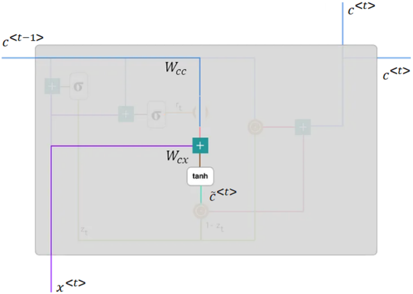
- The special thing about them is that they can be trained to keep information from long ago, without washing it through time or remove information which is irrelevant to the prediction.
- For simplicity we first describe the operation of a network with a single gate, the update gate.
- In this presentation we are going to use the following notation:

+	σ	⊙	tanh
"plus" operation	"sigmoid" function	"Hadamard product" operation	"tanh" function

Memory Cell and Equations

GRU Memory Cell Simplified: Input Data

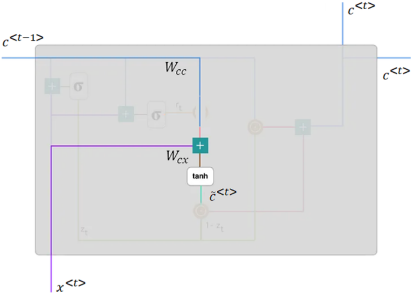
- At each time step t, the GRU network takes an input vector $x(t)$.
- This input could be a single element from a sequence, such as a word in a sentence for natural language processing tasks, a data point in a time series for forecasting, or any other sequential data.



$$\tilde{c}^{<t>} = tanh(W_{cc}c^{<t-1>} + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell Simplified: Previous Hidden State

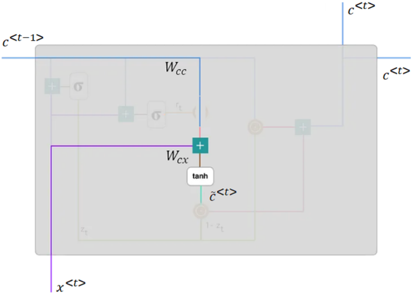
- The GRU network also maintains a hidden state vector $c^{<t-1>}$, which represents the network's memory of the past information.
- Initially, this hidden state can be set to zeros or initialized randomly.



$$\tilde{c}^{<t>} = tanh(W_{cc}c^{<t-1>} + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell Simplified: Current Memory Content

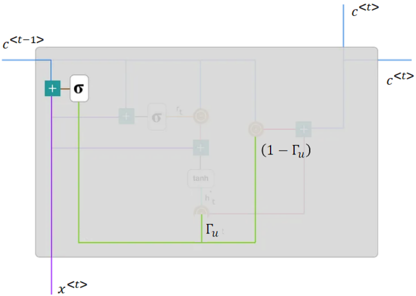
- A candidate hidden state, $\tilde{c}^{<t>}$, is computed as a preliminary update based on the current input $x(t)$ ...



$$\tilde{c}^{<t>} = tanh(W_{cc}c^{<t-1>} + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell Simplified: The Update Gate

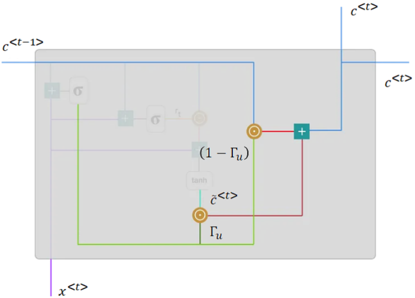
- The update gate helps the model to determine how much of the past information (from previous time steps) needs to be passed along to the future.
- That is really powerful because the model can decide to copy all the information from the past and eliminate the risk of vanishing gradient problem.
- We will see the usage of the update gate in the next slide...



$$\Gamma_u = \sigma(W_{uc}c^{<t-1>} + W_{ux}x^{<t>} + b_u)$$

GRU Memory Cell Simplified: Final Hidden State

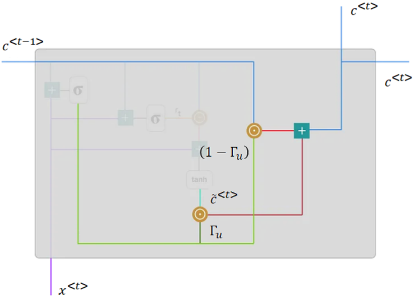
- As the last step, the network needs to calculate $c^{<t>}$, vector which holds information for the current unit and passes it down to the network.
- In order to do that the update gate is needed. It determines what to collect from the current memory content $\tilde{c}^{<t>}$ and what from the previous steps $c^{<t>}$.



$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1-\Gamma_u) * c^{<t-1>}$$

GRU Memory Cell Simplified: Final Hidden State

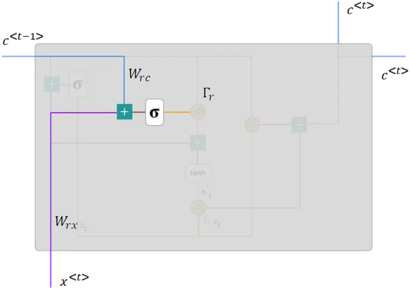
- As you can see, the final hidden state $h(t)$ is a combination of the previous hidden state $c^{<t-1>}$ and the candidate hidden state $\tilde{c}^{<t>}$, weighted by the update gate Γ_u.
- This determines how much of the previous state is retained and how much is updated.



$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1-\Gamma_u) * c^{<t-1>}$$

GRU Memory Cell: The Reset Gate

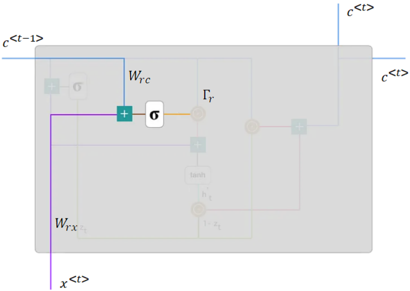
- Essentially, **this gate is used from the model to decide how much of the past information to forget**.
- The formula is the same as the one for the update gate.
- The values in the vector Γ_r are bounded between 0 and 1 by a sigmoid function and depend on the hidden state $c^{<t-1>}$ of the previous time step and the current input $x^{<t>}$.



$$\Gamma_r = \sigma(W_{rc}c^{<t-1>} + W_{rx}x^{<t>} + b_r)$$

GRU Memory Cell: The Reset Gate

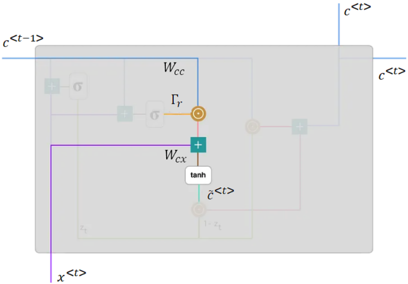
- $c^{<t-1>}$ and the current input $x^{<t>}$ are weighted using the weight matrices W_r.
- Furthermore, a bias b_r is added.
- The difference comes in the weights and the gate‘s usage



$$\Gamma_r = \sigma(W_{rc}c^{<t-1>} + W_{rx}x^{<t>} + b_r)$$

GRU Memory Cell: Use of the Reset Gate

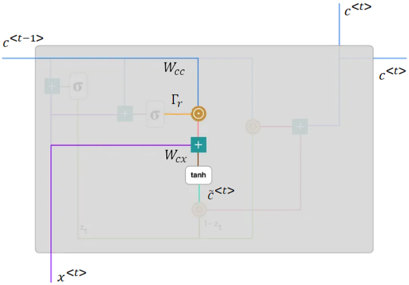
- Using the reset gate the formula for the candidate state vector change as shown in figure;
- The difference with respect to the simplified version is the computation of the Hadamard (element-wise) product between the reset gate Γ_r and $c^{<t-1>}$.
- That will determine what to remove from the previous time steps.



$$\tilde{c}^{<t>} = tanh(W_{cc}[\Gamma_r * c^{<t-1>}] + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell: Use of the Reset Gate

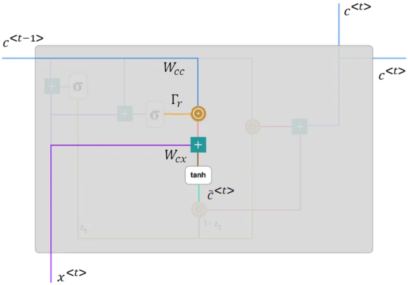
- Let‘s say we have a sentiment analysis problem for determining one‘s opinion about a book from a review he wrote.
- The text starts with "This is a fantasy book which illustrates..." and after a couple paragraphs ends with "I didn"t quite enjoy the book because I think it captures too many details."



$$\tilde{c}^{<t>} = tanh(W_{cc}[\Gamma_r * c^{<t-1>}] + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell: Use of the Reset Gate

- To determine the overall level of satisfaction from the book we only need the last part of the review. In that case as the neural network approaches to the end of the text it will learn to assign Γ_r vector values close to 0, washing out the past and focusing only on the last sentences.



$$\tilde{c}^{<t>} = tanh(W_{cc}[\Gamma_r * c^{<t-1>}] + W_{cx}x^{<t>} + b_c)$$

GRU Memory Cell: Recap

GRU

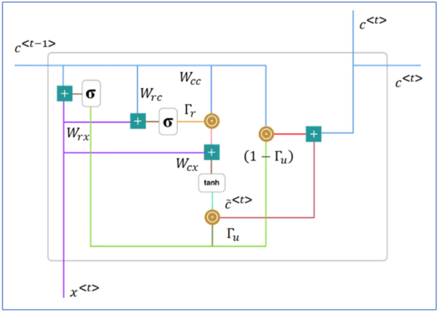
$$\tilde{c}^{<t>} = tanh(W_{cc}[\Gamma_r * c^{<t-1>}] + W_{cx}x^{<t>} + \ b_c)$$

$$\Gamma_u = \sigma(W_{uc}c^{<t-1>} + W_{ux}x^{<t>} + \ b_u)$$

$$\Gamma_r = \sigma(W_{rc}c^{<t-1>} + W_{rx}x^{<t>} + \ b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \ (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$


$c^{<t>}$
$c^{<t-1>}$
$c^{<t>}$
W_{cc}
σ
W_{rc}
Γ_r
σ
W_{rx}
$(1 - \Gamma_u)$
W_{cx}
tanh
$\tilde{c}^{<t>}$
Γ_u
$x^{<t>}$