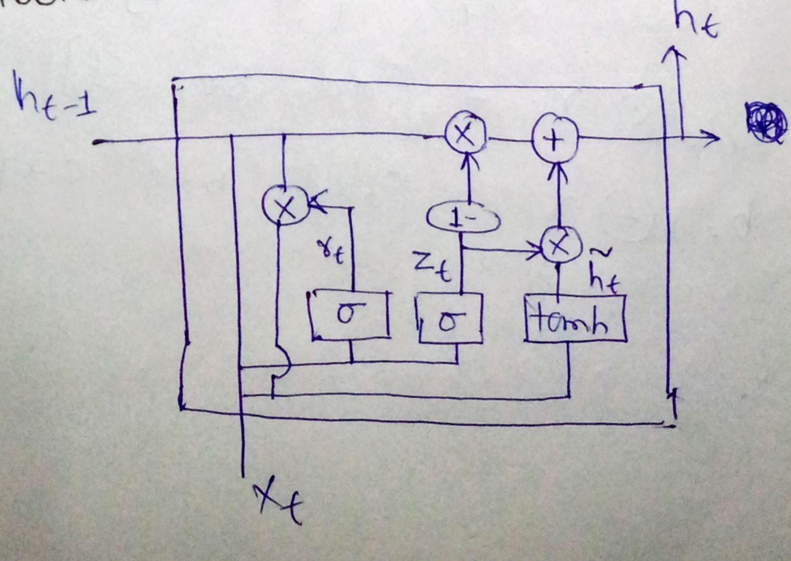
22 Oct 2024

GATED RECURRENT UNIT (GRU)

GRU:

- Gated Recurrent Unit (GRU) is a type of recurrent neural network (RNN) that was introduced by Cho. a in 2014
- that uses gates to control the flow of information in its hidden unit.
- it contain only two gain. reset gate and update gate which. control the flow of information in the hidden unit.
- GRUs have fewer parameters than LSTM networks because they lack a context vector or output gate.
- GRUs use less memory than LSTM networks.
- The reset gate controls how much of the previous hidden state is reset or forgotten before incorporating new information.

Architecture:



→ h_{t-1}, h_t, x_t, r_t, z_t, $\tilde{h}_t$ all are vectors with same dimension except x_t

h_{t-1} → previous hidden state

h_t → current hidden state

x_t → input

r_t → reset gate

z_t → update gate

$\tilde{h}_t$ → candidate hidden state

→ $\boxed{\sigma}$ and $\boxed{\tanh}$ are ANN with activation function sigmoid and tangent hyperbolic function respectively here the number of node is equal to the dimension of vectors.

→ ⊗ ⊕ these are pointwise multiplication and pointwise addition operation respectively.

example $h_{t-1} = [a \quad b \quad c]$

$r_t = [d \quad e \quad f]$

$h_{t-1} \otimes r_t = [ad \quad be \quad cf]$

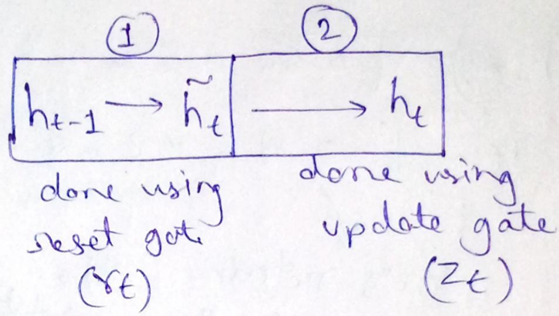
$h_{t-1} \oplus r_t = [a+d \quad b+e \quad c+f]$

input $(h_{t-1}, x_t) \longrightarrow$ output (h_t)

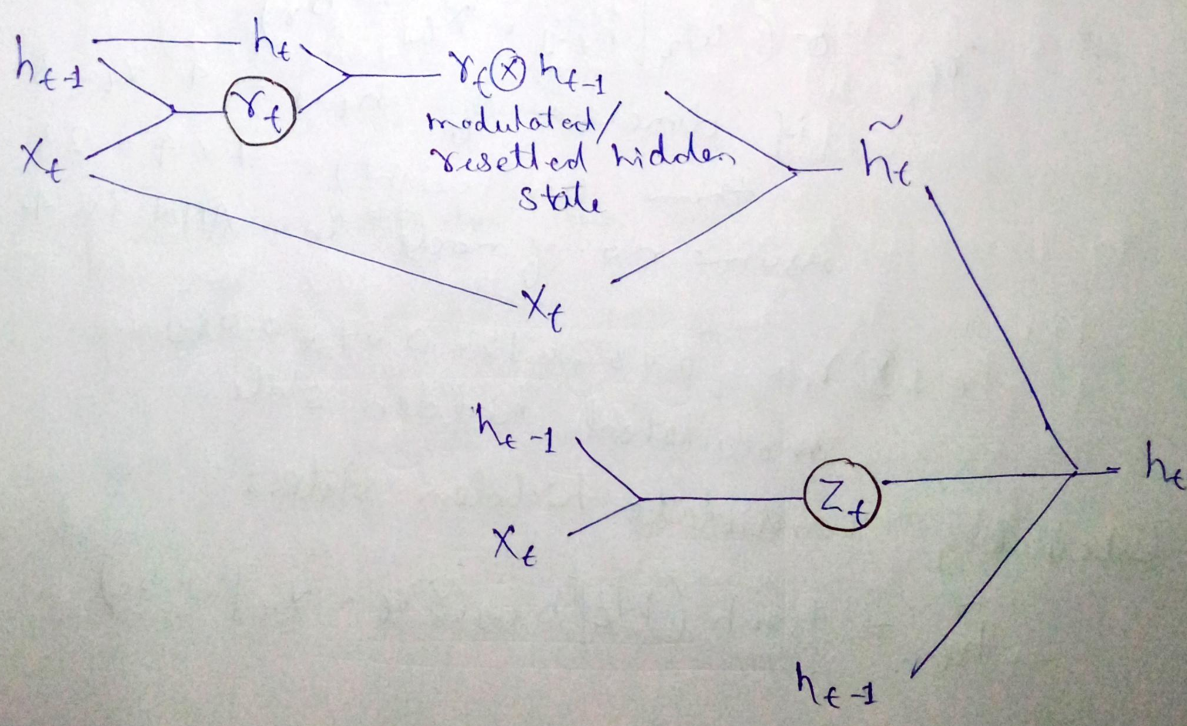
steps:

- calculate r_t (reset gate)
- calculate $\tilde{h}_t$ (candidate hidden state)
- calculate z_t (update gate)
- Calculate h_t (current hidden state)

GRU work in two phases with the two gates.



Flow Diagram/working:



Calculating the Reset gate:

- Reset gate (r_t) is a vecto which has the same dimension as h_{t-1}
- in the vector the values are range from 0 to 1
- it reset the values of h_{t-1} vector's value using the reset gate. based on the context example [power, conflict, tragedy, revenge]

past context $(h_{t-1}) = [0.6, 0.6, 0.7, 0.1]$

$r_t = [0.8, 0.2, 0.1, 0.9]$

↓

80% retain the memory in the ~~existing~~ past context

similar for other 0.2, 0.1, 0.9

$$r_t = \sigma(W_r[h_{t-1}, X_t] + b_r)$$

if dimension of $h_{t-1} = 4$ $X_t = 3$

then W_r (weight) $= 7 \times 4 = 28$

assume no. of node in ANN is 4

$h_{t-1} \otimes r_t = [0.48, 0.12, 0.07, 0.09]$

modulated hidden state

Calculating candidate hidden state:

$$\tilde{h}_t = \tanh(W_c[h_{t-1} \otimes r_t, X_t] + b_c)$$

Calculating the update gate:

$$\begin{matrix} h_{t-1} \searrow & \\ & \longrightarrow z_t \\ x_t \nearrow & \end{matrix}$$

$$z_t = \sigma\left(W_z [h_{t-1}, x_t] + b_z \right)$$

a neural network with activation function is sigmoid

- z_t is a vector and also it is a gate

let assume $z_t = [0.1, 0.7, 0.8, 0.2]$

$$\tilde{h}_t = [0.7, 0.2, 0.1, 0.2]$$

$$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$$

$$h_t = \left((1 - z_t) \otimes h_{t-1}\right) \oplus \left(z_t \otimes \tilde{h}_t\right)$$

here $1 - z_t = [0.9 \quad 0.3 \quad 0.2 \quad 0.8]$

Difference between the LSTMs and GRUs:

1. Number of Gates:
   - LSTM has three gates input (update) gate, forget gate and output gate
   - GRU has two gates reset gate and update gate.

2. Memory Units:

- LSTM uses two separate ~~gate~~ state, the cell state and hidden state. The cell state acts as an "internal memory" and is crucial for carrying long-term dependencies.
- GRU simplifies this by using a single hidden state (ht) to both capture and output the memory.

3. Parameter Unit:

- LSTM Generally has more parameters than a GRU because of its additional gates and separate cell state. for an input size of a d and a hidden size of h, the LSTM has $4\times((d\times h)+(h\times h)+h)$ $4\times((d\times h)+(h\times h)+h)$ parameters.
- GRU has fewer parameters for the same sizes, the GRU has $3\times((d\times h)+(h\times h)+h)$ $3\times((d\times h)+(h\times h)+h)$ parameters.

4. Computational Complexity:

- LSTM, due to the extra gate the cell state, LSTM are typically more computationally intensive than GRUs.
- GRUs is simpler and can be faster to compute, especially on smaller datasets or when computational resources are limited.

5. Empirical Performance:

- LSTM, in many tasks, especially more complex ones, LSTMs have been observed to perform slightly better than GRUs.

- GRUs can perform comparably to LSTMs on certain tasks, especially when data is limited or tasks are simpler. They can also train faster due to fewer parameters.

6. Choice in Parctice:

- The choice between LSTM and GRU often comes down to empirical testing. Depending on the dataset and task, one might out-perform the other. However, GRUs, due to their simplicity are often the first choice when starting out.