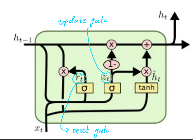
GRU (Gated Recurrent Unit)

→ Simple architecture than LSTM.
→ In certain datasets GRU performs better than LSTM.
→ GRU is simple architecture with fewer amount of trainable parameters.



→ in some cases GRU may work better or in some cases LSTM may work better, so it is recommended to try both.

→ GRU don't have cell state, they only have hidden state.

Q. What is the goal behind this architecture....?

→ for any time stemp t, we provides h_{t-1} and x_t as input.
→ Based on inputs, GRU will calculate h_t for current timestemp t.

Some terminologies

- h_{t-1} : Previous hidden state.
- h_t : current hidden state
- x_t : Input at current time stemp.
- r_t : Reset gate
- Z_t : Update gate
- $\tilde{h}_t$: Candidate hidden state

- σ
- σ
- tanh

Neural Network Architectures (FCL)

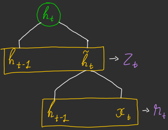
- × : Point-wise multiplication
- + : Point-wise addition
- 1- : Point-wise 1-

→ Mathematically h_{t-1}, h_t, $\tilde{h}_t$, x_t, r_t and z_t, all are vectors.
→ All, except x_t will be of same dimentional vectors.
→ Number of units will always same in all Neural Network Architectures.
→ Number of units in Neural Network Architecure will be equal to the dimentions of h_{t-1}, h_t, $\tilde{h}_t$, Z_t and r_t.

Input
h_{t-1}
x_t

Output
h_t

Steps
1. Calculate r_t
2. Calculate $\tilde{h}_t$
3. Calculate Z_t
4. Calculate h_t



Created with Notewise

→ h_{t-1} will work like a memory to store the context of content till now.

→ Suppose h_{t-1} is 4D vector like [0.4, 0.21, 0.12, 0.01], than this 4D vector represents context of provided content / story in which each individual number holds some aspects.

Example

Suppose we have a story as content like given below.

1. There was a king Vikram very strong and powerful
2. There was an enemy king kaali
3. Both had a war and kaali killed Vikram
4. Vikram had a son Vikram Jr who grew up he to become very strong just like his father
5. He also attacked Kaali But got killed
6. Vikram Jr too had a son called Vikram super Jr and when he grew up he also fought kaali
7. And he killed kaali and took revenge of his father and grand father

And suppose h_t of this store is [0.5, 0.11, 0.21, 0.92]

than those numbers may hold the aspect of story like.

[Power, Conflict, Tragedy, Revenge]

→ Those aspects may be anything, because in deep learning some processes are black box.

Example

Sentence_number	Assumed_h_t ([power, conflict, tragedy, revenge])
1	[0.4, 0, 0, 0]
2	[1, 0.5, 0, 0]
3	[0.7, 0.7, 0.4, 0]
4	[0.9, 0.6, 0.3, 0]
5	[0.8, 0.8, 0.6, 0.5]
6	[1, 0.9, 0.5, 0.7]
7	[0.7, 0.8, 0.3, 1] → final h_t

Reset gate and update gate are causing these changes in h_t.

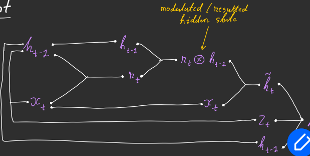
more powerfull, more conflict, less tragedy, full of revenge

$$h_{t-1} \underbrace{\xrightarrow{\text{step-1}}}_{\text{reset gate}} \tilde{h}_t \underbrace{\xrightarrow{\text{step-2}}}_{\text{update gate}} h_t$$

→ Current $\tilde{h}_t$ will be heavily influenced towards current input x_t. hence we can't directly update h_t. hence we need to make balance between h_{t-1} and $\tilde{h}_t$.

→ in simple words, if x_t is more important based on story perspective, than we will give more importance to $\tilde{h}_t$ otherwise h_{t-1}.

Simple flowchart



#1 Calculating the reset gate

→ Dimention of r_t gate will same as h_{t-1} gate.
→ r_t is a gate, hence the value will be between 0 to 1.
as seen previous example.

[Power, Conflict, Tragedy, Revenge]

→ Suppose r_t at 1^{st} dimention is 0, it means that "r_t is closed for that perticular dimention".
→ if r_t at perticular dimention (position) is 1, it means that "r_t is opened for that perticular dimention".

→ So, reset gate will reset necessary dimentions of h_{t-1} based on current input x_t.

based on given story, suppose first 3 event occured & currently we're at 4^{th} event.

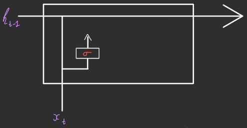
- There was a king Vikram very strong and powerful
- There was an enemy king kaali
- Both had a war and kaali killed Vikram
- Vikram had a son Vikram Jr who grew up he to become very strong just like his father
- He also attacked Kaali But got killed
- Vikram Jr too had a son called Vikram super Jr and when he grew up he also fought kaali
- And he killed kaali and took revenge of his father and grand father

→ Current for x_t suppose
$h_{t-1} = [0.6, 0.6, 0.7, 0.1]$ based on past 3 events.
→ but based on current event (4^{th}) we will update h_{t-1}, now new update may look like this. $r_t = [0.8, 0.2, 0.1, 0.9]$.

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

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

more power raised | conflict decreased | tragedy decreased | revenge raised highly



Suppose our input word are represented in 3D.

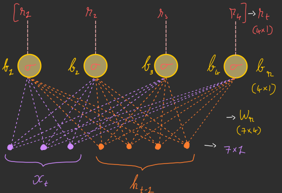
since h_{t-1} is 4D, our NN will have 4 neurons.

$$r_t = \sigma\left(\underbrace{W_r}_{4\times7} \cdot \underbrace{[h_{t-1}, x_t]}_{7\times1} + \underbrace{b_r}_{4\times1}\right)$$

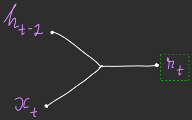
4×1

$\sigma(4\times1)$

4×1



till now, we have done...



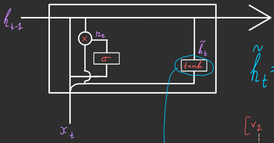
• Now our next task is.

elementwise multiplication

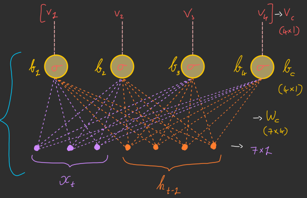
$r_t \otimes h_{t-1}$ (modulated / resetted state)

$$h_{t-1} = [0.6, 0.6, 0.7, 0.1] \otimes r_t = [0.8, 0.2, 0.1, 0.9]$$

$$h_{t-1} \otimes r_t = [0.48, 0.12, 0.07, 0.09] \rightarrow \text{Modulated / Resetted state}$$

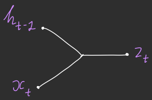


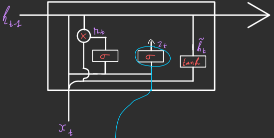
$$\tilde{h}_t = \tanh\left(W_c \cdot [h_{t-1} \otimes r_t, x_t] + b_c\right)$$



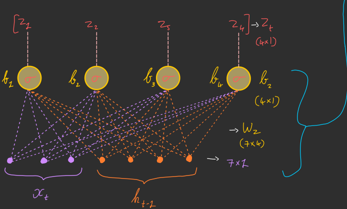
$\tilde{h}_t$ is kind of a proposal, which says that that should be the h_t.

#2 Update Gate





$$\underset{4\times1}{z_t} = \sigma\left(W_z \cdot [h_{t-1}, x_t] + b_z\right)$$



→ $\tilde{h}_t$ is heavily skewed with current x_t hence, we can't declare it as h_t.

→ z_t knows the balance between h_{t-1} and $\tilde{h}_t$.

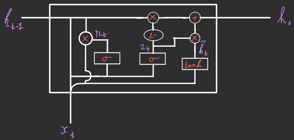
Now we have

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

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

Assumed $z_t = [0.1, 0.7, 0.8, 0.2]$

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



$z_t \uparrow$ = more importance of candidate ($\tilde{h}_t$).

$z_t \downarrow$ = more importance to past memory (h_{t-1}).

$(1-z_t) \otimes h_{t-1} \Rightarrow [0.9, 0.3, 0.2, 0.8] \otimes [0.6, 0.6, 0.7, 0.1]$

$= [0.54, 0.18, 0.14, 0.08]$

$z_t \otimes \tilde{h}_t = [0.07, 0.14, 0.08, 0.04]$

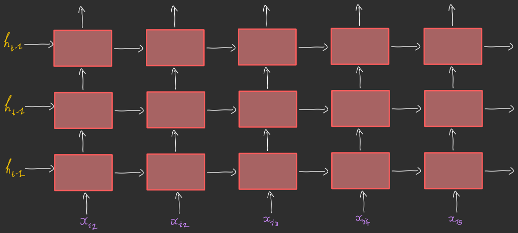
$\oplus \rightarrow [0.61, 0.32, 0.22, 0.12]$

Deep RNNs

Architecture

→ Suppose we have a dataset for sentiment analysis. where each sentence has maximum 5 words.

→ Below is the structure of deep/stacked RNN.



Unfolding stacks in time.....

Let's revise the basic rnn architecture on given below dataset.

Sample data

Reviews	Sentiment
cat mat rat	1
rat rat mat	0
mat cat cat	1

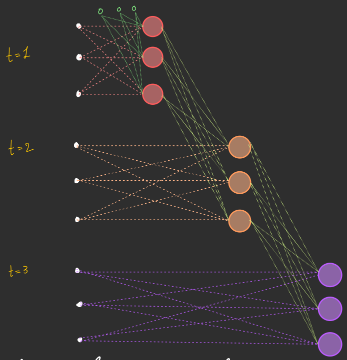
Vocabulary

cat : [1 0 0]

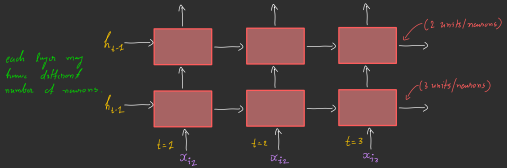
mat : [0 1 0]

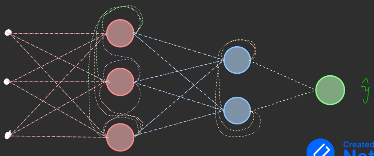
rat : [0 0 1]

Let's see the architecture for $t=3$ timestamp with 1 hidden layer.

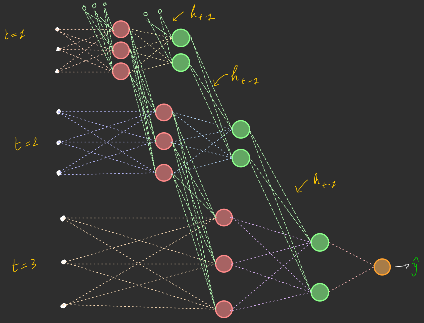


Now let's see for 2 RNN hidden layers.

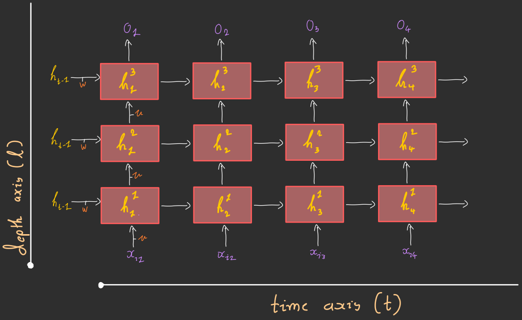




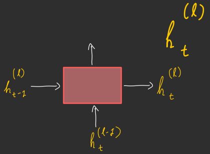
Now let's unfolding them on timestamp.



Notations



Structure of single RNN cell.



t: time stamp
l: layer number

$$h_t^{(l)} = \tanh\left(W^{(l)} \cdot h_{t-1}^{(l)} + U^{(l)} \cdot h_t^{(l-1)} + b^{(l)}\right)$$

Q. Why & when to use....?

1. Hierarchical Representation
2. Customization for Advanced Tasks

→ The core idea behind deep RNNs is to catch the complex patterns of data.
→ Initial layer of this RNN will catch the premitive features. it will try to extract sentiment word by word.
→ At middle level sentiment analysis will be on sentence level.
→ Top layers will analyze overall sentence.

Situations.....

→ Complex tasks.
→ Machine translation.
→ Large dataset.
→ Sufficient computational facilities.
→ Not satisfied with simple models.

→ Stacking is also applicable on Deep LSTM and Deep GRU.

Bidirectional RNNs

→ What if we face such scenarious in which future input affects past output.
→ Suppose we're performing an NLP task NER (Named Entity Recognition)
↳ breaking the sentence word-by-word & extracting entities.

Example
- My name is punth. ← person
- I live in surat. ← place

Example text (for NER)
- I love amazon, it's a great website.
- I love amazon, it's a longest river.

→ At uni-directional process in NER, it may declare Amazon as an Organization or a location, because it has not read the next sentence yet which causes ambiguity.
→ In this situation, i/p of next sentence affects output of current one.
So, currently we're at "I love amazon" hence we don't have idea about "It's a great website" or "It is a beautiful river".
→ In this situation, we can use bi-directional RNNs, which are capable to process input from both sides.
→ It is just one scenario, there are many scenarious in which bi-directional RNN can be used like **Machine Translation**.

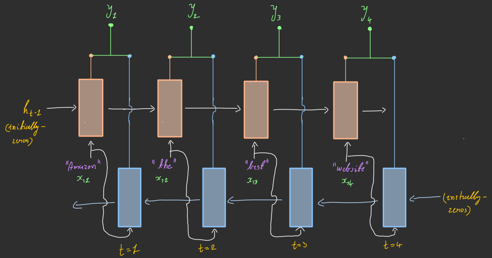
Bi-directional RNN Architecture

1. Amazon, the best website.
2. Amazon, the beautiful river.

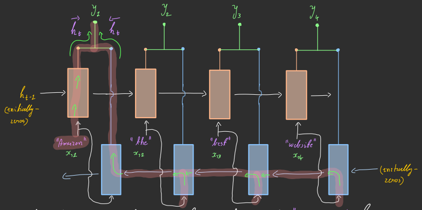
→ Bi-directional RNN uses 2 different RNNs forward RNN ($\overrightarrow{RNN}$) and backward RNN ($\overleftarrow{RNN}$).

forward RNN ($\overrightarrow{RNN}$) : Reads sentence left to right.
backward RNN ($\overleftarrow{RNN}$) : Reads sentence right to left.
} At each time-stamp we will concatenate the o/p of both RNNs.

Let's design RNN structure for above sample text of 4 words.



→ Now suppose we want to know whether "Amazon" word is any Organization or any Place, if we focus on forward RNN, there is no any previous incomming word. but if we focus on backward RNN, "Amazon" word combined with 3 outputs of previous words "the", "best" and "website".



→ Due to backward RNN iteration, we know the word "website" hence we can decide that "Amazon" is a "website" not a "River" or place. or organization

$$\vec{h}_t = \tanh\left(\vec{W} \cdot \vec{h}_{t-1} + \vec{u} \cdot x_t + \vec{b}\right)$$

$$\overleftarrow{h}_t = \tanh\left(\overleftarrow{W} \cdot \overleftarrow{h}_{t+1} + \overleftarrow{u} \cdot x_t + \overleftarrow{b}\right)$$

$$y_t = \sigma\left(W \cdot [\vec{h}_t, \overleftarrow{h}_t] + b\right)$$

Applications of bi-directional RNN/GRU/LSTM

- Named Entity Recognition (NER)
- Parts of Speech (POS) tagging
- Time-series forecasting
- Machine Translation
- Sentiment Analysis

Drawbacks

- Complexity
  - overfitting issue may occur.
- Requires entire data.
  - Not applicable on real-time speech recognition like system, where data is comming on real time.