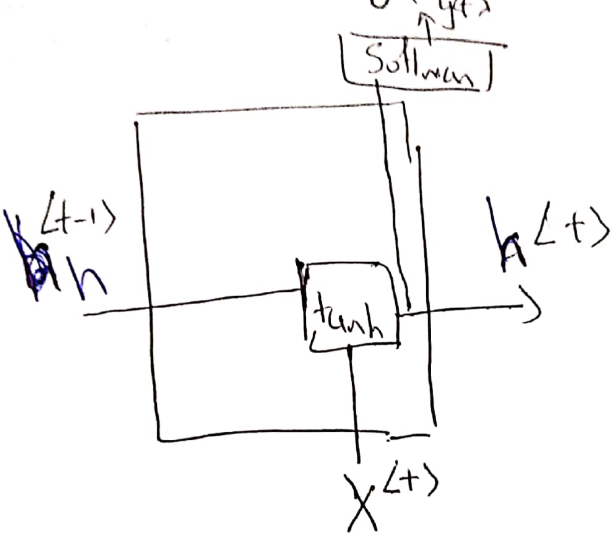
GRU

Recap: RNN → tanh

$$a^{<t>} = g(W_a [a^{<t-1>}, x^{<t>}] + b_a)$$



1) Reset gate: Influence of the previous hidden state
2) Update Gate: Influence of a newly computed update
3) Proposing an update hidden state
4) Computing Updated hidden state

Sample Sentence:

The $\boxed{\text{cat,}}$ which already ate, ..., $\boxed{\text{was}}$ full.

GRU unit will have new memory unit called c = memory cell.
will be used to remember if cat is singular or plural

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

Candidate for $c^{<t>}$ for replacement

$$\tilde{c}^{<t>} = \tanh(W_c [c^{<t-1>}, x^{<t>}] + b_c) \quad \text{—(I}$$

Page 2

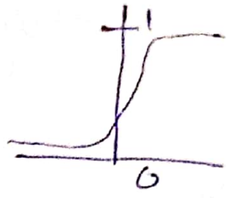
GRU → gamma u $\Gamma_u = 0 \longrightarrow 1$

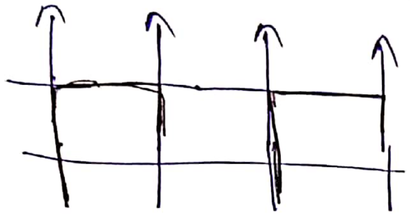
gate

Γ_u ↙ update

assume always 0 or 1

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$ Eq (2)





Gate

G for gamma

gate will decide wether we update it

$\tilde{c}^{<t>}$ will be the candidate value

Assume ~~Cat~~ singular = 1, plural = 0

GRU will memoize the value until was

job of gate is to decide when to update the value.

When we see The cat → subject

forget when we ~~see~~ was

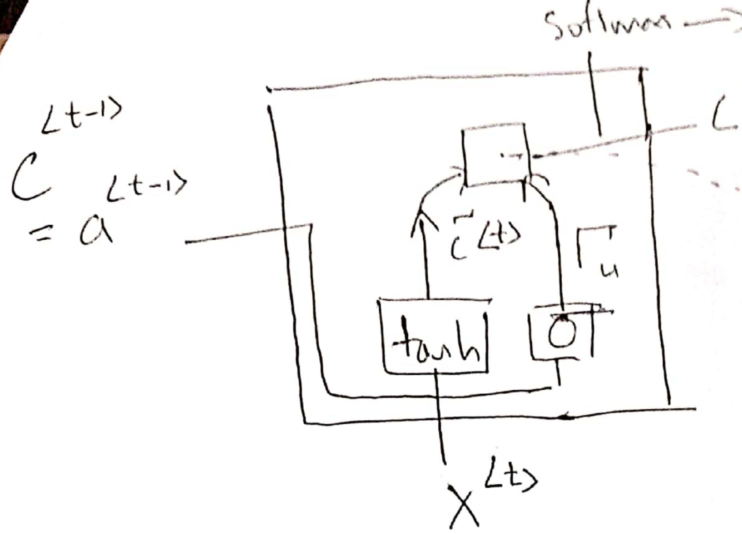
$$c^{<t>} = \underbrace{\Gamma_u}_{\text{gate}} * \underbrace{\tilde{c}^{<t>}}_{\text{candidate value}} + (1 - \Gamma_u) * \underbrace{c^{<t-1>}}_{\text{old value}}$$ Eq (3)

E.g. if $\Gamma_u = 1$ then set the candidate value. and update

for all values in the middle Γ_u should $= 0$

The cat, which already ate...., was full.

if $1-\Gamma_u = 1$; $\Gamma_u = 0$; $c^{<t>} = c^{<t-1>}$



new value for memory cell

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

element wise multiplication

$\Gamma_u = 0.00001$

does not suffer from vanishing gradient problem b/c if $\Gamma_u \simeq 0$ then $1-\Gamma_u = 1$ and $c^{<t>} = c^{<t-1>}$

allowed NN to train over long range dependencies

$c^{<t>}$ can be a vector. e.g. 100-dimension vector. then $\tilde{c}^{<t>}$ will also be same dimension and Γ_u will also be same dimension.

if gate is a 100-dimensional then it ~~we~~ will decide which bits we would like to update. Γ_u can have middle values.

Element-wise multiplication allows which bits of the memory cell needs to be updated it can help in keeping some bits constant while updating other bits.

E.g

The cat, which already ate ... was full

Some bits for singular plural

other bits to remember food etc

use only a subset of bits

Eq ① ② and ③

Full GRU

Introduce $\Gamma_r \rightarrow$ how relevant is $c^{<t-1>}$ to compute the next candidate $\tilde{c}^{<t>}$

Reset gate: Influence of previous hidden state

$$r \rightarrow \Gamma_r = \sigma(W_r[c^{<t-1>}, X^{<t>}] + b_r \quad \text{Eq } (4)$$

Researchers have evaluated different combinations to assess long range affects

$$\tilde{h} \rightarrow \tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c) \quad (1)$$

Update Gate: Determines influence of an update proposal on the new hidden state

$$u \rightarrow \Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u) \quad (2)$$

$$h \rightarrow c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1-\Gamma_u) * c^{<t-1>} \quad (3)$$

Common version.