Deep Learning

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Examples of sequence data

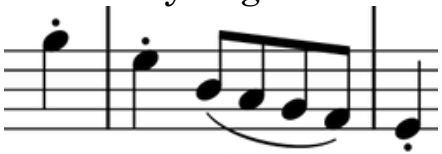
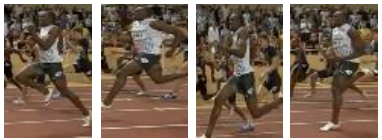
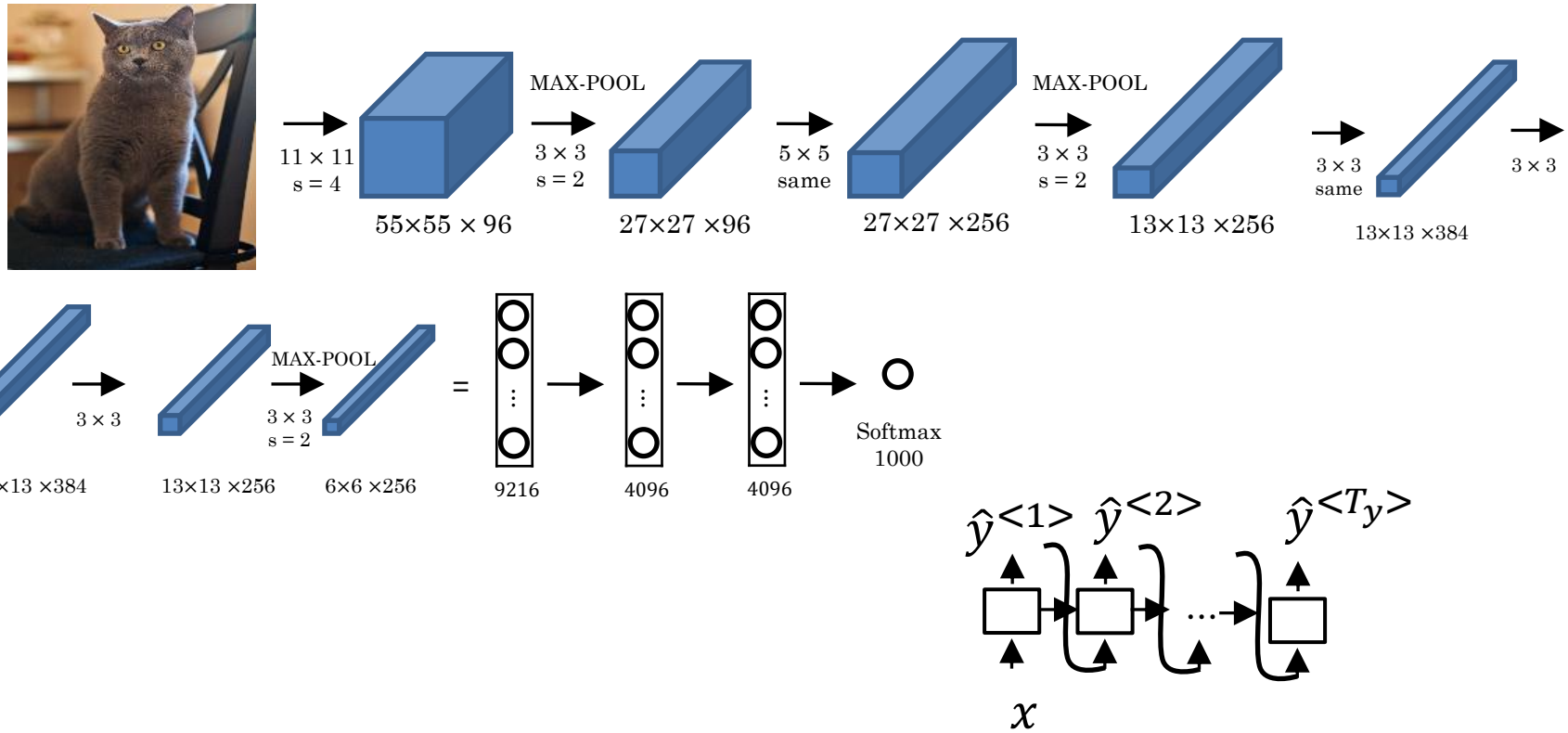
Speech recognition		→	"The quick brown fox jumped over the lazy dog."
Music generation	∅	→	
Sentiment classification	"There is nothing to like in this movie."	→	
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	→	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	→	Do you want to sing with me?
Video activity recognition		→	Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	→	Yesterday, Harry Potter met Hermione Granger.

Image captioning



[Mao et. al., 2014. Deep captioning with multimodal recurrent neural networks]
[Vinyals et. al., 2014. Show and tell: Neural image caption generator]
[Karpathy and Li, 2015. Deep visual-semantic alignments for generating image descriptions]

Recurrent Neural Networks

- Now, given this input X let's say that you want a model to operate Y that has one outputs per input word and the target output the design Y tells you for each of the input words is that part of a person's name.
- X: (Ali Ahmadi)and(Hassan Hamidi) invented a new drug.

$X^{\langle 1\rangle}$ $X^{\langle 2\rangle}$ $X^{\langle 3\rangle}$ ... ... $X^{\langle t\rangle}$ ... ... $X^{\langle 9\rangle}$ $T_x=9$

- y: 1 1 0 1 1 0 0 0 0 $T_y=9$

$y^{\langle 1\rangle}$ $y^{\langle 2\rangle}$ $y^{\langle 3\rangle}$ ... ... $y^{\langle 9\rangle}$

- $X^{(i)\langle t\rangle}$

داده آموزشی i ام

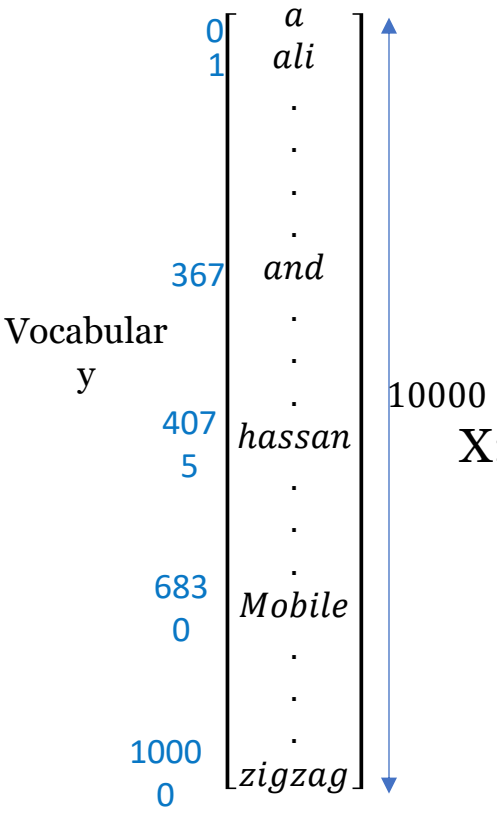
$T_x^{(i)} = 9$

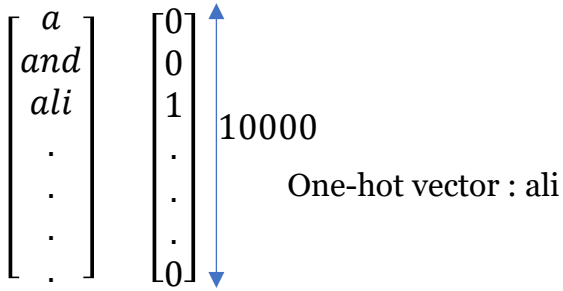
$T_y^{\langle i\rangle}$

- This is our first serious foray into NLP or Natural Language Processing.

Representing words

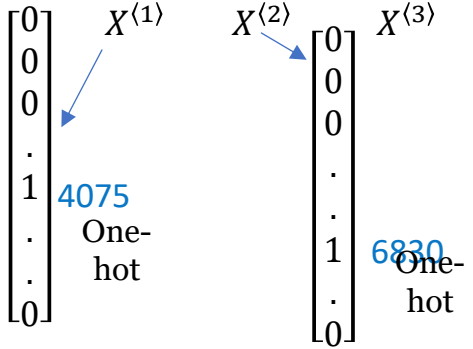
- Dictionary: 30000, 50000





- So, if you have chosen a dictionary of 10,000 words, what you can do is then use one-hot representations to represent each of these words.

X: Ali Ahmadi and Hassan Hamidi invented a new drug.

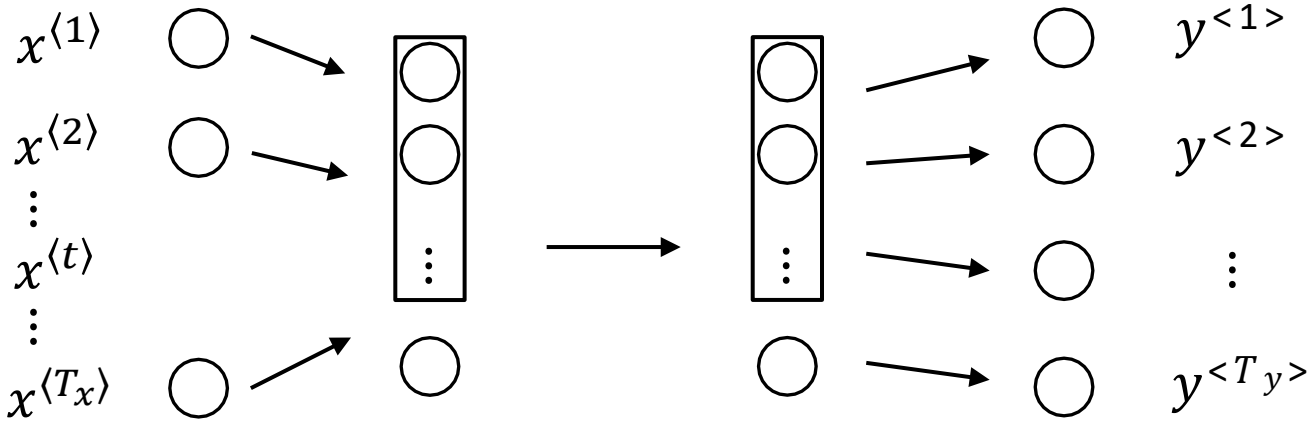


Representing words

- Utilizing a sequence model for supervised learning to map input X to output Y.
- Introduction of an "Unknown Word" token for handling out-of-vocabulary words.
- Describing a notation for training sets in sequence data.

Recurrent Neural Network Model

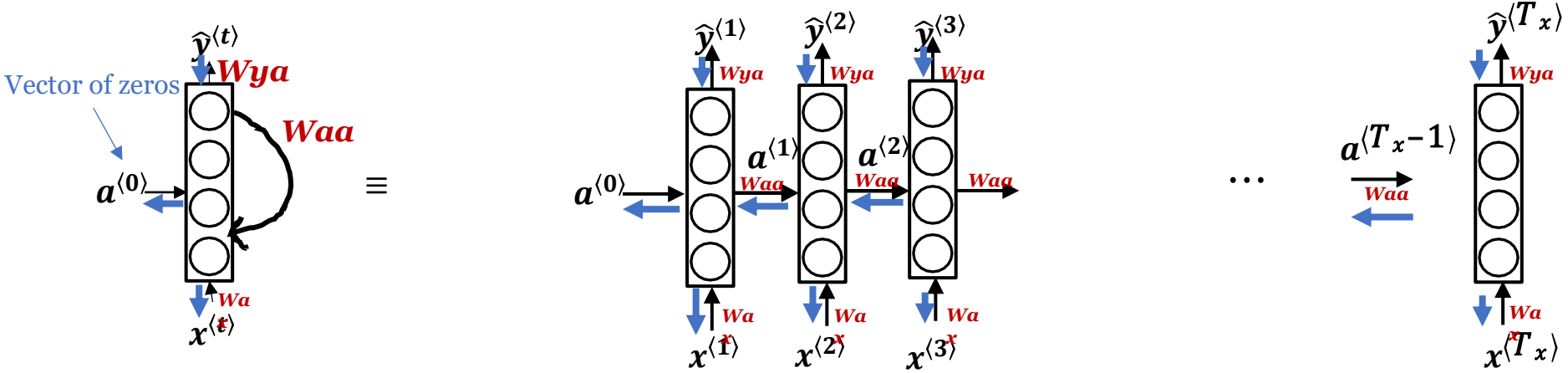
Why not a standard network?



Problems:

- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.

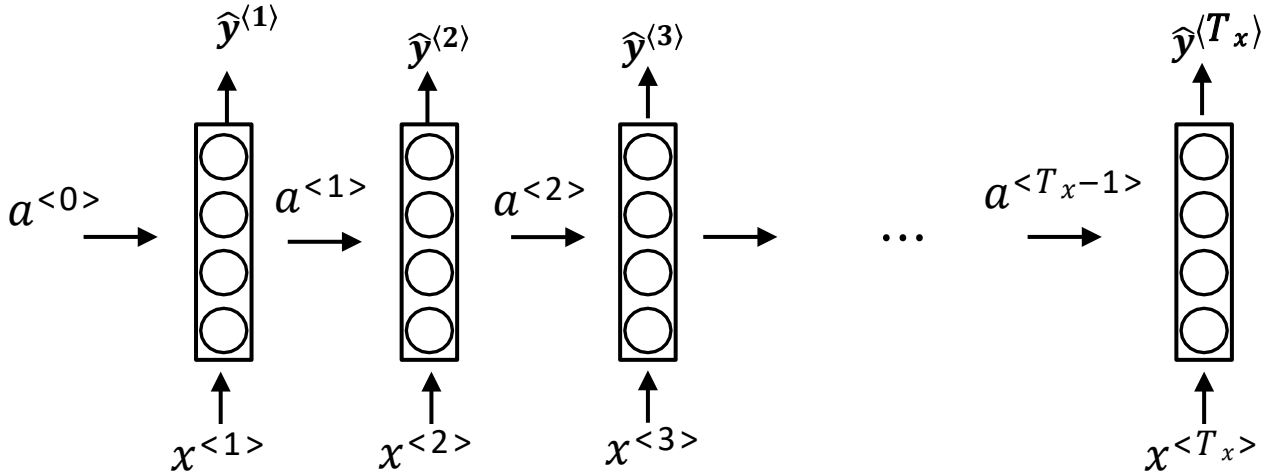
Recurrent Neural Networks



- Activation at time zero in neural networks is often initialized as a vector of zeros.
- Some researchers prefer initializing it as $a^{\langle 0 \rangle}$ randomly.
- There are alternative methods to initialize the activation at time zero

Backward propagation through time

Forward Propagation



$a^{\langle 0 \rangle} = \vec{0}$

$a^{\langle 1 \rangle} = g_1(w_{aa}a^{\langle 0 \rangle} + w_{ax}x^{\langle 1 \rangle} + b_a)$ ← tanh|Relu

$\hat{y}^{\langle 1 \rangle} = g_2(w_{ya}a^{\langle 1 \rangle} + b_y)$ ← sigmoid

$$a^{\langle t \rangle} = g(w_{aa}a^{\langle t-1 \rangle} + w_{ax}x^{\langle t \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = g(w_{ya}a^{\langle t \rangle} + b_y)$$

Simplified RNN notation

$$a^{\langle t \rangle} = g(w_{aa}a^{\langle t-1 \rangle} + w_{ax}x^{\langle t \rangle} + b_a)$$

(100,100) (100,10000) 100 10000 100

$$a^{\langle t \rangle} = g(w_a[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_a)$$

$$\hat{y}^{\langle t \rangle} = g(w_{ya}a^{\langle t \rangle} + b_y)$$

- $a^{\langle t \rangle} = g\left(w_a\left[a^{\langle t-1 \rangle\prime}, u^{\langle t \rangle\prime}\right]' + b_a\right)$

$$\begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix}$$

100 10000 10100

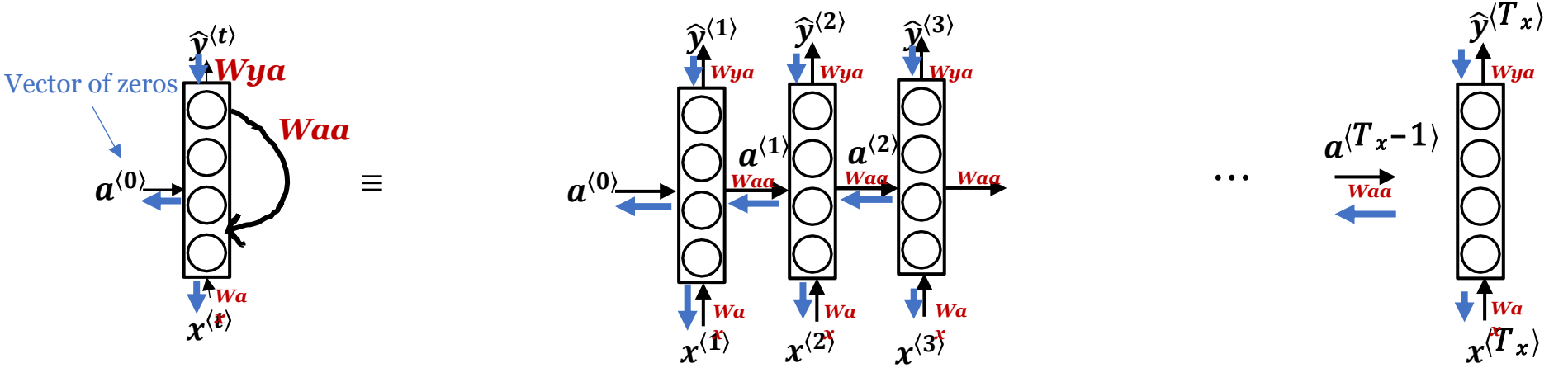
100 10000 100

$$w_a = [w_{aa} \vdots w_{ax}]$$

(100,10100)

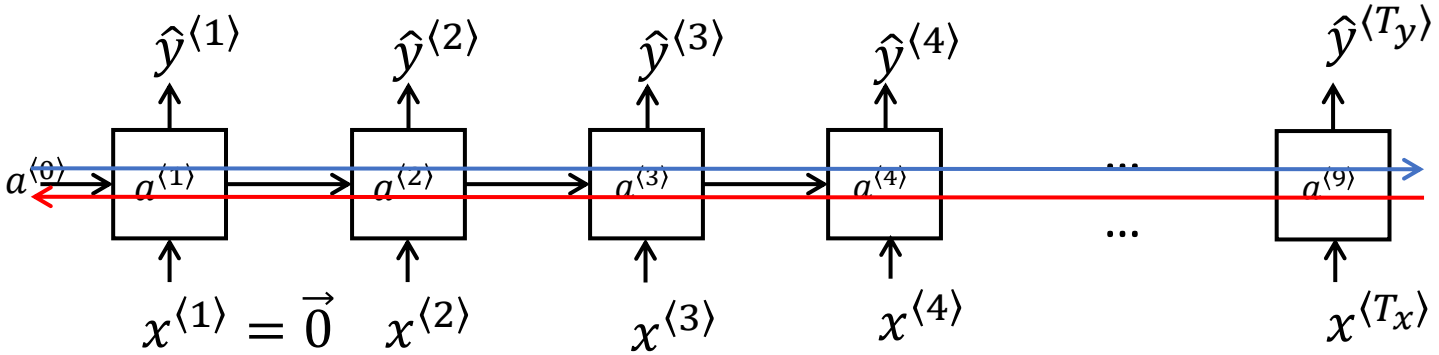
$$[w_{aa} \vdots w_{ax}]\begin{bmatrix} a^{\langle t-1 \rangle} \\ x^{\langle t \rangle} \end{bmatrix} = w_{aa}a^{\langle t-1 \rangle} + w_{ax}x^{\langle t \rangle}$$

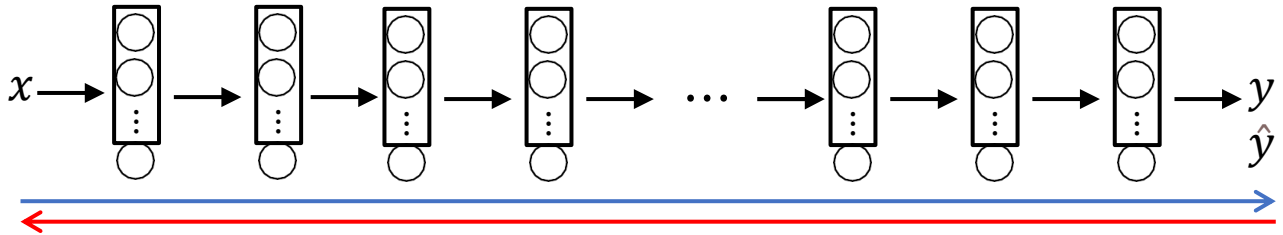
Recurrent Neural Networks



Vanishing gradients with RNNs

- The cat which already ate bunch of food was full
- The cats … were full



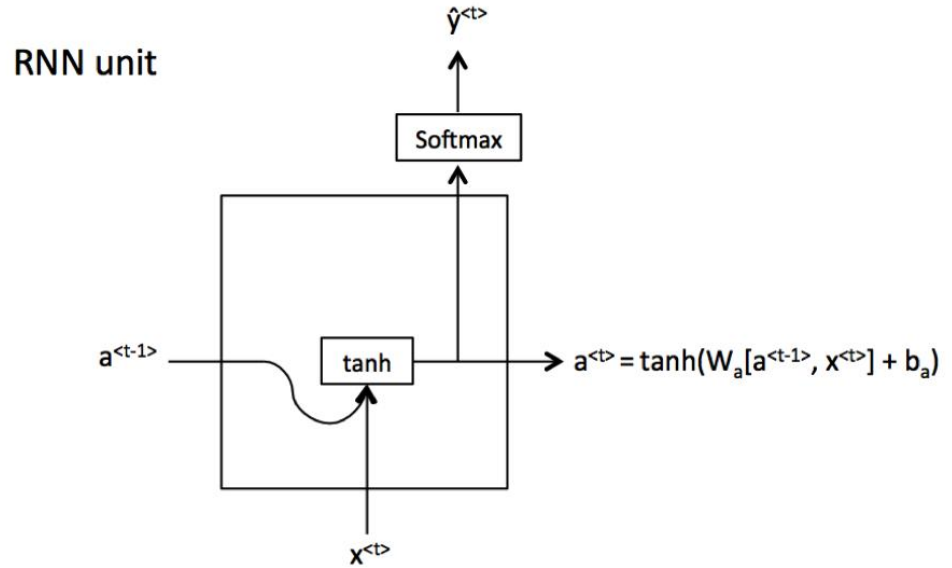


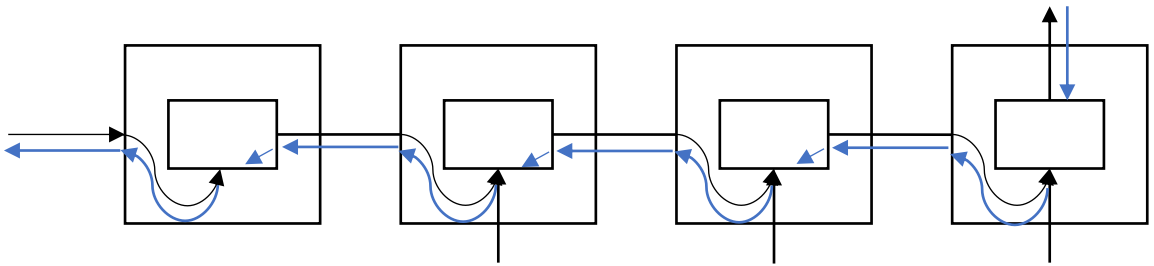
Exploding gradients. NaN gradient clipping

RNN Unit

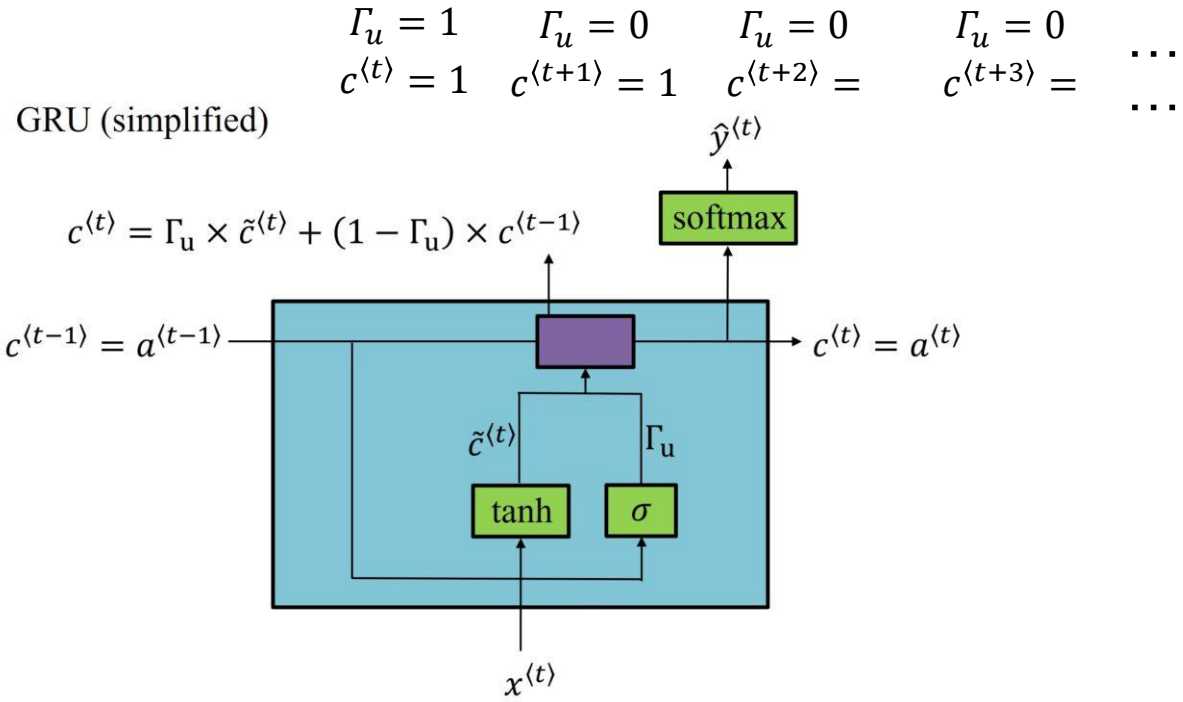
- $a^{\langle t \rangle} = g\left(W_a\left[a^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_a\right)$

tanh



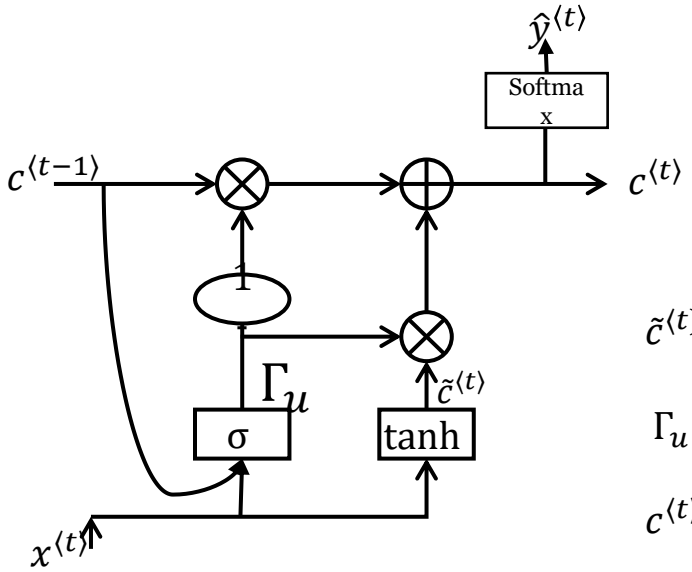


GRU (simplified)



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

GRU (Simplified)



C : memory cell

$$\tilde{c}^{\langle t \rangle} = \tanh\left(\omega_c\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$$

$$\Gamma_u = \sigma\left(w_{\mathrm{u}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{u}}\right)$$

$$c^{\langle t \rangle} = \Gamma_{\mathrm{u}} * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$$

Full GRU

- $\hat{c}^{\langle t \rangle} = \tanh\left(w_c\left[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_c\right)$
- $\Gamma_u = \sigma\left(w_{\mathrm{u}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{u}}\right)$
- $\Gamma_r = \sigma\left(w_{\mathrm{r}}\left[c^{\langle t-1 \rangle}, x^{\langle t \rangle}\right] + b_{\mathrm{r}}\right)$
- $c^{\langle t \rangle} = \Gamma_{\mathrm{u}} * \hat{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle}$