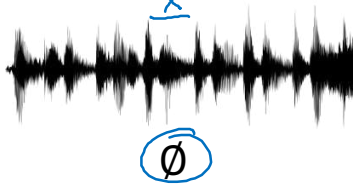
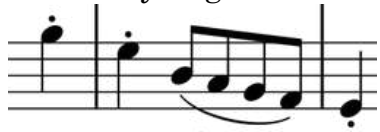
deeplearning.ai

Recurrent Neural Networks

Why sequence models?

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.

Andrew Ng

Notation used in Sequence Models

An motivating example

Goal: Find persons' name in the sentence below. (Name Entity Recognition)

Input X:	Harry	Potter	and	Hermione	Granger	invented	a	new	spell.
$T_x = 9$	$x^{\langle 1 \rangle}$	$x^{\langle 2 \rangle}$	$x^{\langle 3 \rangle}$	...		$x^{\langle t \rangle}$	...		$x^{\langle 9 \rangle}$
Desired output y:	1	1	0	1	1	0	0	0	0
$T_y = 9$	$y^{\langle 1 \rangle}$	$y^{\langle 2 \rangle}$	$y^{\langle 3 \rangle}$	...		$y^{\langle t \rangle}$	...		$y^{\langle 9 \rangle}$

The $\langle t \rangle$ is used to index the position of a word in a sequence.

T_x: the length of the input sequence. T_y: the length of the output sequence.
T_x and T_y are not necessarily the same value.

$x^{(i)\langle t \rangle}$: the t-th element in the input sequence of the training sample i.

$y^{(i)\langle t \rangle}$: the t-th element in the output sequence of the training sample i.

$T_x^{(i)}$: the input sequence length for the training sample i.

$T_y^{(i)}$: the output sequence length for the training sample i.

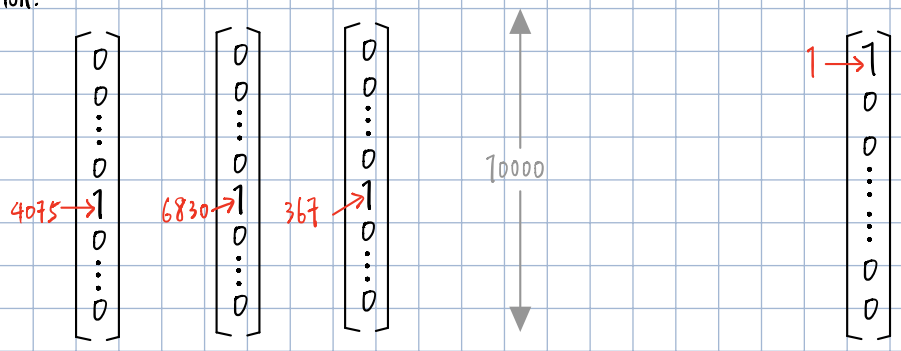
Vocabulary: Build a list with 10000 most used English words. What if encountering a word that is not in the vocabulary. I can create a new token <UNK>.

a	1
aaron	2
⋮	
and	367
⋮	
harry	4075
⋮	
potter	6830
⋮	
zulu	10,000

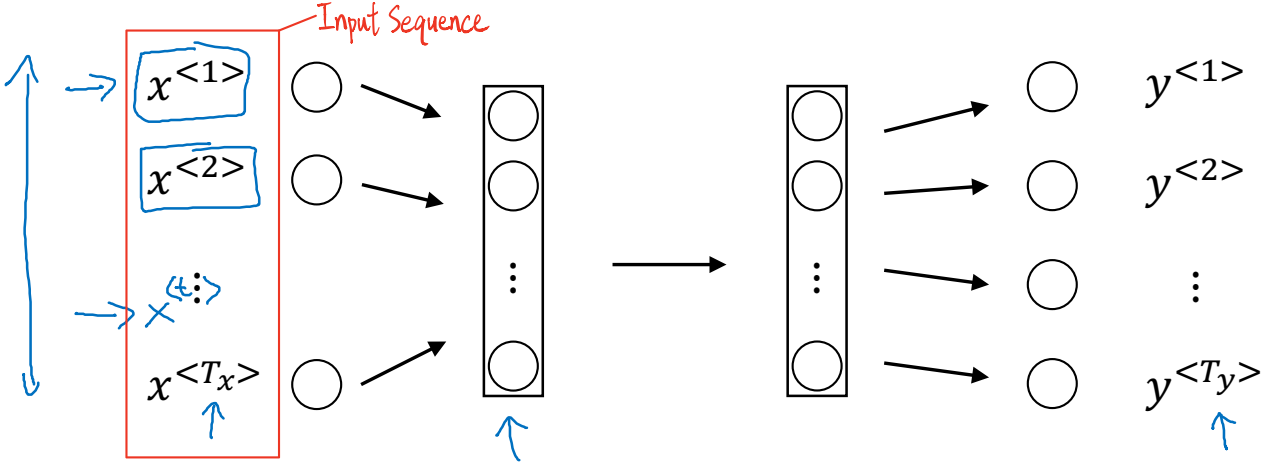
Representing Words

Input X: Harry $X^{\langle 1 \rangle}$ Potter $X^{\langle 2 \rangle}$ and $X^{\langle 3 \rangle}$ Hermione ... Granger $X^{\langle t \rangle}$ ... $X^{\langle 7 \rangle}$ invented a new spell. $X^{\langle 9 \rangle}$

One-Hot representation:



Why not a standard network?



Problems:

Every sentence has different lengths.

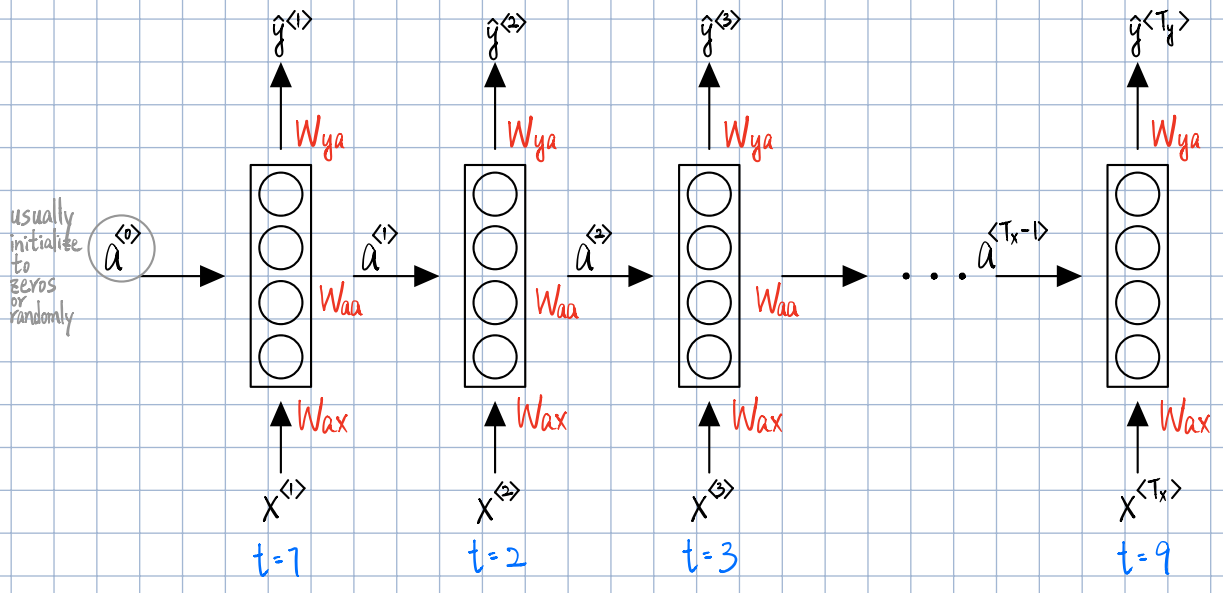
- Inputs, outputs can be different lengths in different examples.
- Doesn't share features learned across different positions of text.

Recurrent Neural Network Model

Input X: Harry Potter and Hermione Granger invented a new spell. $T_x = T_y$

$X^{\langle 1 \rangle}$ $X^{\langle 2 \rangle}$ $X^{\langle 3 \rangle}$ ... $X^{\langle t \rangle}$ ... $X^{\langle 7 \rangle}$ $X^{\langle 8 \rangle}$ $X^{\langle 9 \rangle}$

Previously, $X^{(i)\langle t \rangle}$ can be seen as the t-th element in the input sequence of the training sample i. Here, it can also be seen as the time step t.



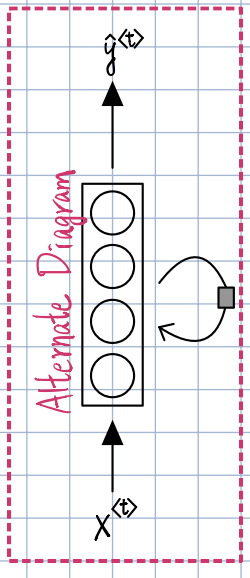
$a^{\langle 1 \rangle}$: the activiation value of the time step 1. $a^{\langle 2 \rangle}$: the activiation value of the time step 2.

At each time step t, the RNN passes on activation to the next time step for it to use.

The RNN scans through the data from left to right. The parameters, Wax, Waa, Wya, it uses for each time step are shared.

In this RNN, when making the prediction for $\hat{y}^{\langle 3 \rangle}$, it gets the information not only from $X^{\langle 3 \rangle}$ but also from $X^{\langle 1 \rangle}$ and $X^{\langle 2 \rangle}$.

Weakness: This RNN only uses the information that is earlier in the sequence to make a prediction. In particular, when predicting $\hat{y}^{\langle 3 \rangle}$, it doesn't use information about words $X^{\langle 4 \rangle}$, $X^{\langle 5 \rangle}$ and so on.



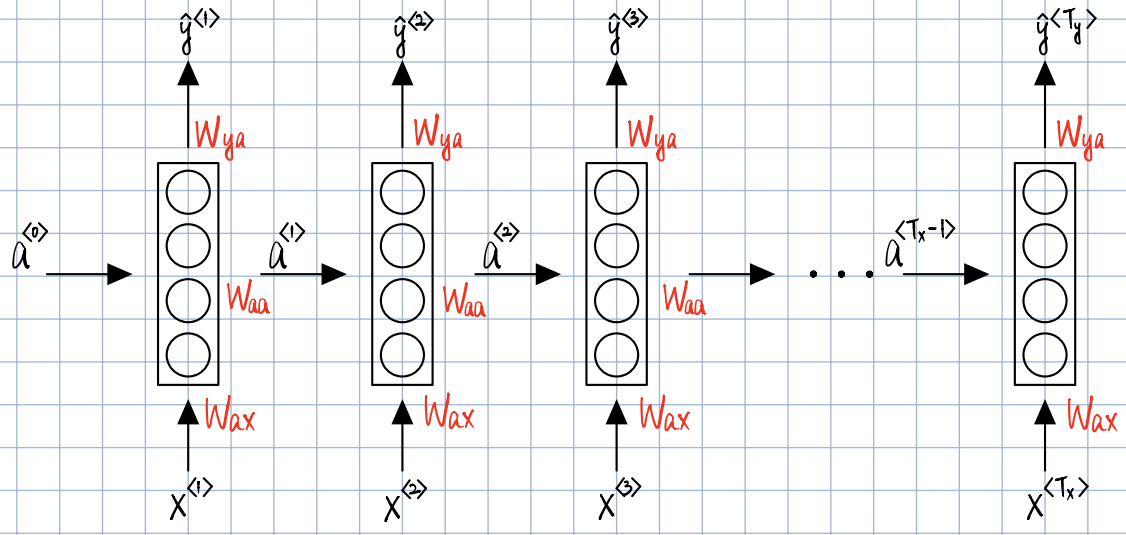
E.g.:

He said, "Teddy (a name) Roosevelt was a great president."

He said, "Teddy (not a name) bears are on sale."

Using unidirectional RNN is just impossible to say if Teddy is a name or not.

Forward Propagation



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

$a^{\langle 1\rangle} = g_1(W_{aa}\, a^{\langle 0\rangle} + W_{ax}\, x^{\langle 1\rangle} + b_a)$ The activiation function g_1 is often tanh or ReLU

$\hat{y}^{\langle 1\rangle} = g_2(W_{ya}\, a^{\langle 1\rangle} + b_y)$

g_2: sigmoid for binary class. (Name Entity recognition)
softmax for k class.

$$a^{\langle t\rangle} = g_1(W_{aa}\, a^{\langle t-1\rangle} + W_{ax}\, x^{\langle t\rangle} + b_a)$$
$$\hat{y}^{\langle t\rangle} = g_2(W_{ya}\, a^{\langle t\rangle} + b_y)$$

$$\begin{bmatrix} W_{aa} \mid W_{ax} \end{bmatrix} \begin{bmatrix} a^{\langle t-1\rangle} \\ \hline x^{\langle t\rangle} \end{bmatrix} = W_{aa}\, a^{\langle t-1\rangle} + W_{ax}\, x^{\langle t\rangle}$$

Simplifying RNN notation

$$a^{\langle t\rangle} = g_1(W_a [a^{\langle t-1\rangle}, x^{\langle t\rangle}] + b_a)$$
$$\hat{y}^{\langle t\rangle} = g_2(W_y\, a^{\langle t\rangle} + b_y)$$

Stack W_{aa} and W_{ax} horizontally $[W_{aa} \mid W_{ax}] = W_a$

$[a^{\langle t-1\rangle}, x^{\langle t\rangle}]$ means stack vertically $\begin{bmatrix} a^{\langle t-1\rangle} \\ \hline x^{\langle t\rangle} \end{bmatrix}$

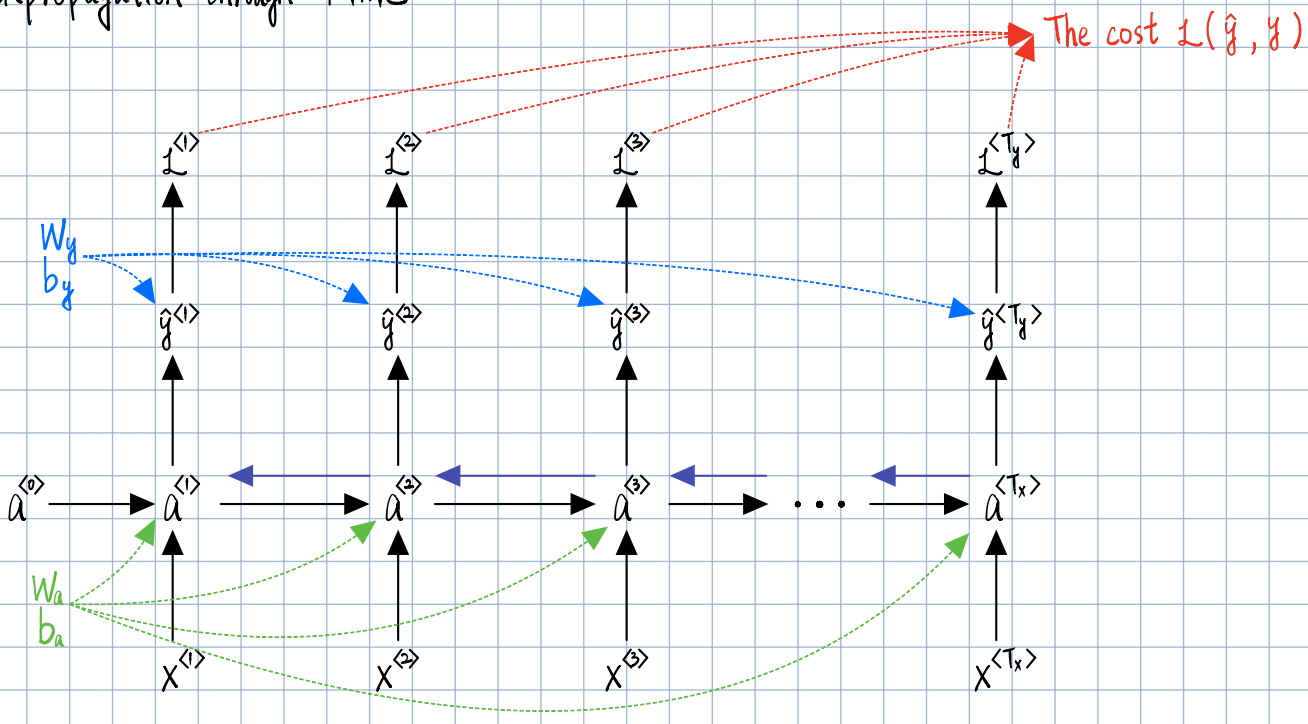
E.g.: $a^{\langle t-1\rangle}$ is (100,). $x^{\langle t\rangle}$ is (10000,)

Then W_{aa} is (100, 100) and W_{ax} is (100, 10000)

$$\underset{100}{\begin{bmatrix} W_{aa} \\ 100 \end{bmatrix}} \mid \underset{10000}{\begin{bmatrix} W_{ax} \\ 10000 \end{bmatrix}} = \underset{(100,\,10100)}{W_a}$$

$$[a^{\langle t-1\rangle}, x^{\langle t\rangle}] = \begin{bmatrix} a^{\langle t-1\rangle} \\ \hline x^{\langle t\rangle} \end{bmatrix} \begin{matrix} 100 \\ 10000 \end{matrix} \quad \updownarrow\ 10100$$

Backpropagation through Time



Backpropagation through Time

$$\mathcal{L}^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle}) = -y^{\langle t \rangle} \log \hat{y}^{\langle t \rangle} - (1 - y^{\langle t \rangle}) \log (1 - \hat{y}^{\langle t \rangle})$$

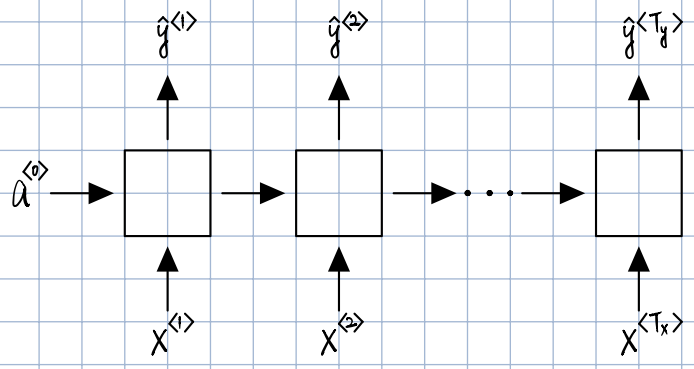
$$\text{The cost } \mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{\langle t \rangle}(\hat{y}^{\langle t \rangle}, y^{\langle t \rangle})$$

Different Types of RNNs

Inspired by Andrey Karpathy "The Unreasonable Effectiveness of Recurrent Neural Networks"

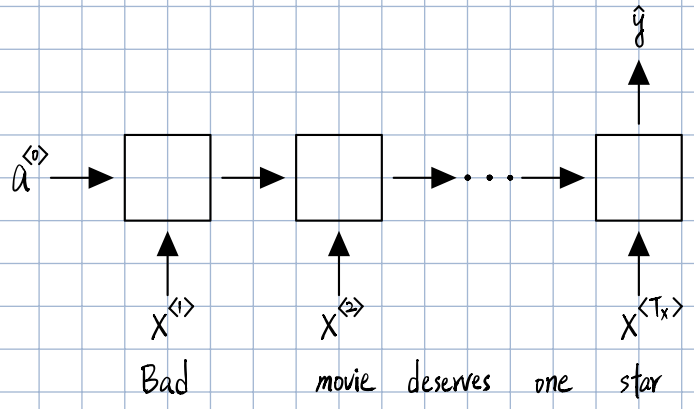
Many-to-many. $T_x = T_y$

Eg: Name Entity Recognition



Many-to-one . $T_x = T_y$

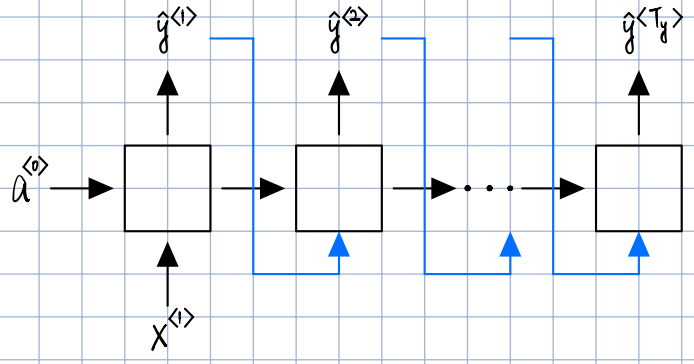
Eg: Sentiment Classification
x: text
y: 1...5



One-to-one.
The standard neural network.

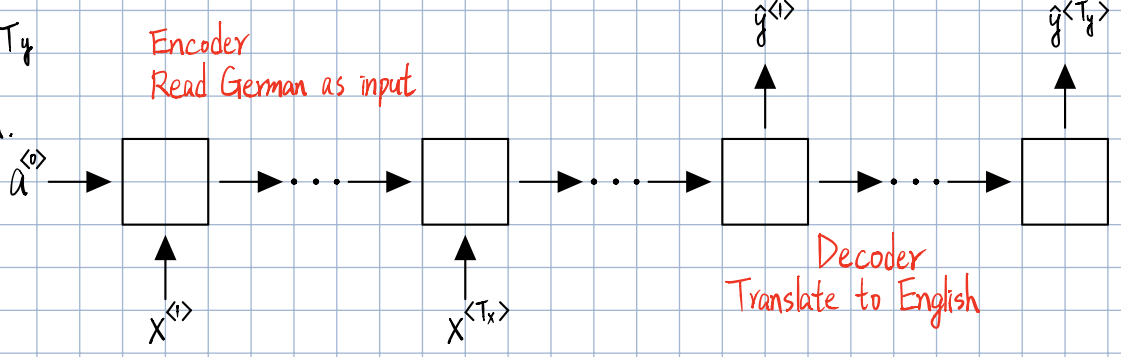
One-to-many .

Eg: Music generation
x: genre, $\emptyset$
y: music



Many-to-many. $T_x \neq T_y$

Eg: Machine translation.
x: German
y: English



Language Model and Sequence Generation

What is language model ?

The output of speech recognition:

The apple and pair salad.

The apple and pear salad.

Which sentence has higher probability? P(sentence)

$P(\text{The apple and pair salad.}) = 3.2 \times 10^{-13}$

$P(\text{The apple and pear salad.}) = 5.7 \times 10^{-10}$

Given a sentence, a sequence of words $y^{<1>}, y^{<2>}, \dots, y^{<T_k>}$, a language model will estimate the probability of the particular sequence of words. $P(y^{<1>}, y^{<2>}, \dots, y^{<T_k>})$

Now, the question is how to build a language model ? Let's build a language model with RNN.

Training set : large corpus of English text

Give a sentence: Cats average 15 hours of sleep a day. <EOS>

End of Sentence

Tokenize : $y^{<1>}$ $y^{<2>}$ . . . $y^{<8>}$ $y^{<9>}$

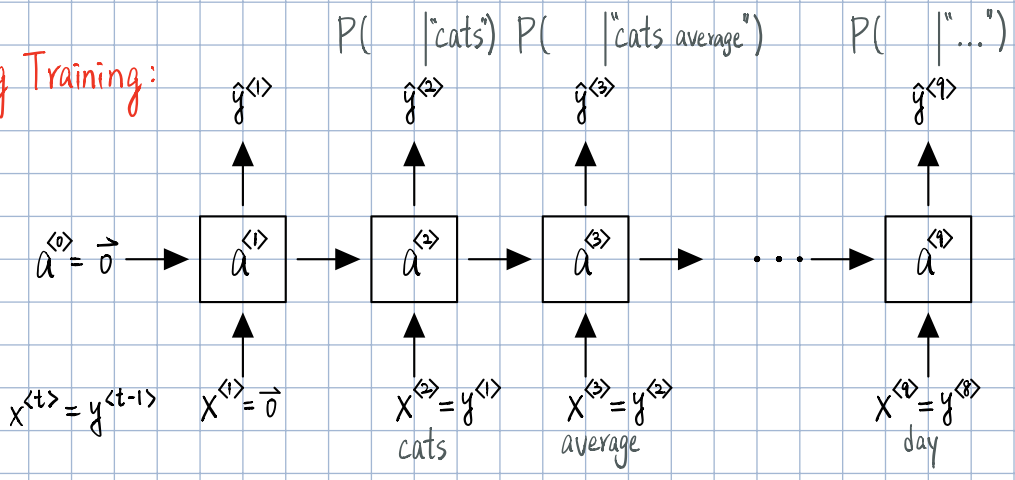
<UNK> unknown word. Not in vocabulary.

The Egyptian Mau is a bread of cat. <EOS>

RNN model:

Using "Cats average 15 hours of sleep a day. <EOS>" as training sample.

During Training:



$\hat{y}^{\langle 1\rangle}$: the prob. of the first prediction being any one of the word in the vocabulary.

$\hat{y}^{\langle 2\rangle}$: the prob. of the 2nd prediction being any one of the word in the vocabulary given cats.

$\mathcal{L}^{\langle t\rangle}(\hat{y}^{\langle t\rangle}, y^{\langle t\rangle}) = -\sum_i y_i^{\langle t\rangle} \log \hat{y}_i^{\langle t\rangle}$ softmax loss function

The cost $\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}^{\langle t\rangle}(\hat{y}^{\langle t\rangle}, y^{\langle t\rangle})$

During Inference:

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

Given first three words, "cats", "average", "15", it can predict the next word.

$P(y^{\langle 1\rangle}, y^{\langle 2\rangle}, y^{\langle 3\rangle}) = P(y^{\langle 1\rangle})\, P(y^{\langle 2\rangle} | y^{\langle 1\rangle})\, P(y^{\langle 3\rangle} | y^{\langle 1\rangle}, y^{\langle 2\rangle})$

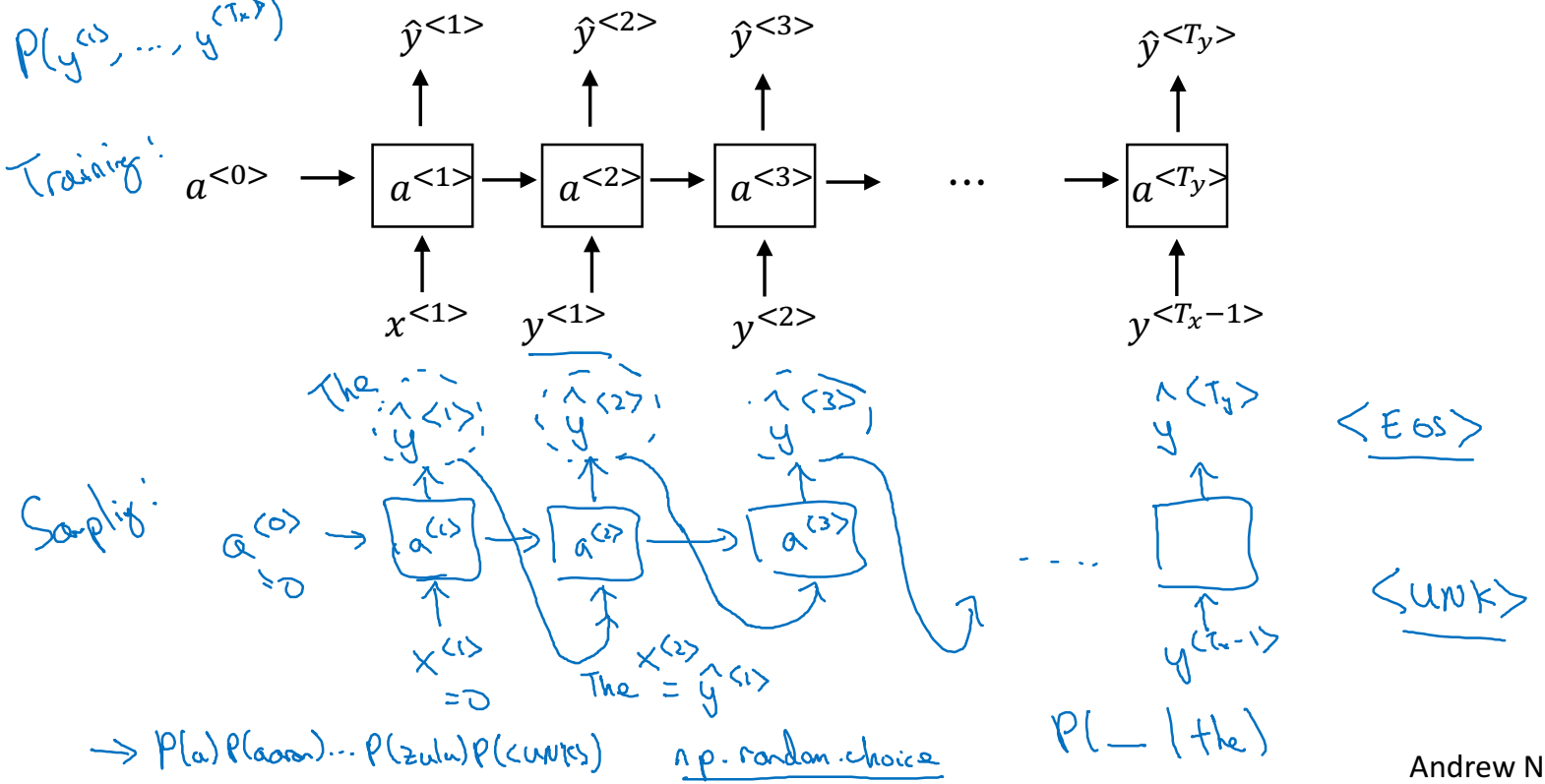
It calculates the probability of "cats average 15" being a sentence.

deeplearning.ai

Recurrent Neural Networks

Sampling novel sequences

Sampling a sequence from a trained RNN



Andrew Ng

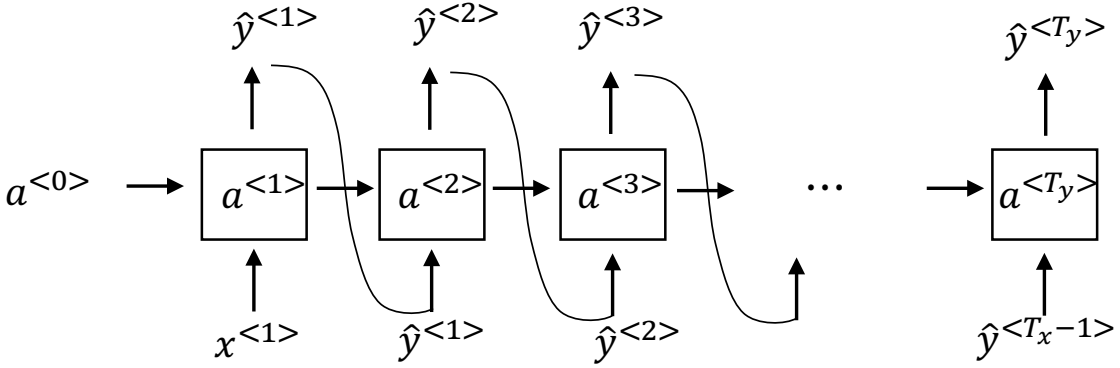
Character-level language model

→ Vocabulary = [a, aaron, …, zulu, <UNK>] ←

→ Vocabulary = [a, b, c, … , z, ␣, ., ,, ;, 0, …, 9, A, …, Z]

$y^{<1>}$ $y^{<2>}$ $y^{<3>}$ $y^{<4>}$

Cat average … May



Sequence generation

News

President enrique peña nieto, announced sench's sulk former coming football langston paring.

"I was not at all surprised," said hich langston.

"Concussion epidemic", to be examined.

The gray football the told some and this has on the uefa icon, should money as.

Shakespeare

The mortal moon hath her eclipse in love.

And subject of this thou art another this fold.

When besser be my love to me see sabl's.

For whose are ruse of mine eyes heaves.

Vanishing Gradients with RNNs

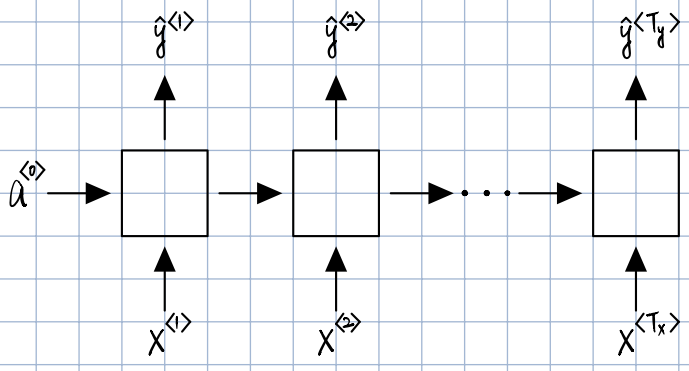
Example sentences:

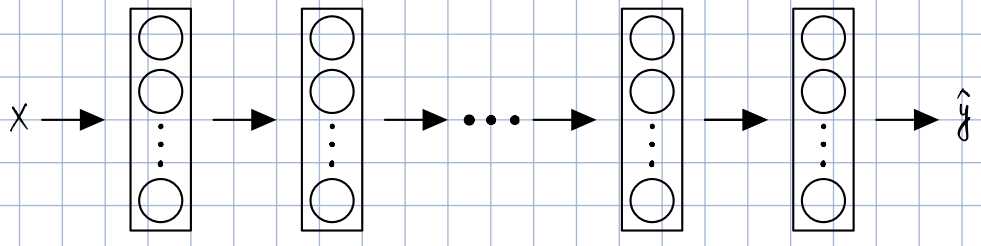
This can be arbitrary long.

The cat, which already ate apple, pear, banana, ... , was full.

The cats, which already ate apple, pear, banana, ... , were full.

The basic RNN is not good at capturing very long term dependency.





Vanishing gradients is hard to solve.

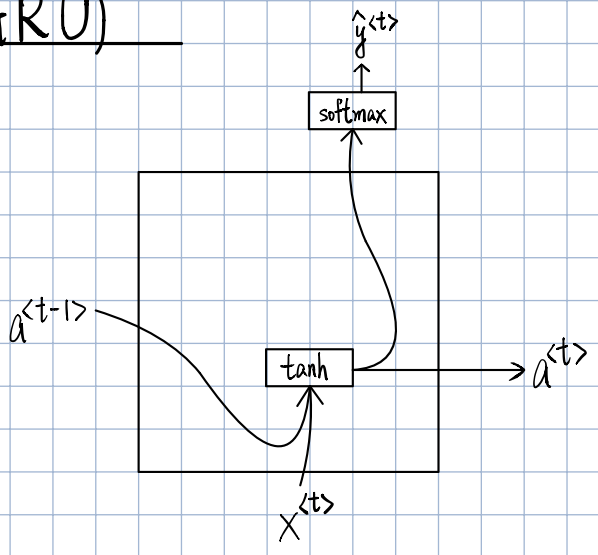
In contrast, exploding gradients is easier to solve.

Apply gradient clipping to clip very large gradients > threshold.

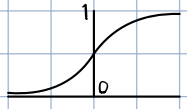
Gated Recurrent Unit (GRU)

RNN unit:

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



σ: sigmoid function



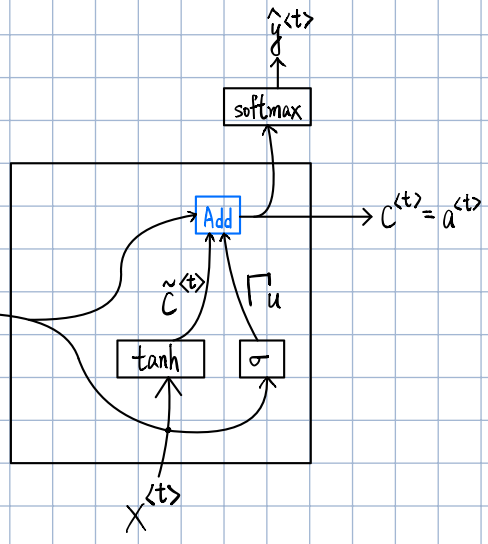
GRU (simplified) C: memory cell $c^{\langle t \rangle} = a^{\langle t \rangle}$

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

Gate $\Gamma_u = \sigma(W_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$ (update)

Imagine Γ_u is either 0 or 1 most of the time.

$$c^{\langle t \rangle} = \Gamma_u * \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) * c^{\langle t-1 \rangle} \quad *: \text{element-wise}$$



If Γ_u is very close to 0, eg, $\Gamma_u = 10^{-6}$, $c^{\langle t \rangle}$ will be assigned to $c^{\langle t-1 \rangle}$, which can mitigate the vanishing gradient problem. The value of $c^{\langle t \rangle}$ is maintained.

Intuition: Γ_u is like a binary gate (bit: 0 / 1), which can control if the information is updated. The element-wise multiplication can keep some bits as constant while updating other bits.

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

Full GRU

$$\tilde{c}^{\langle t \rangle} = \tanh(W_c[\Gamma_r * c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$$

(Γ_r: relevance)

$$\Gamma_u = \sigma(W_u[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u) \quad \text{update gate}$$

$$\Gamma_r = \sigma(W_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_r) \quad \text{relevance gate}$$

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

Read "On the properties of neural machine translation: Encoder-decoder approaches"
"Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling"

Long Short Term Memory (LSTM)

Read 1997 "Long Short Term Memory"

Read Christopher Olah's "Understanding LSTM Networks"

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

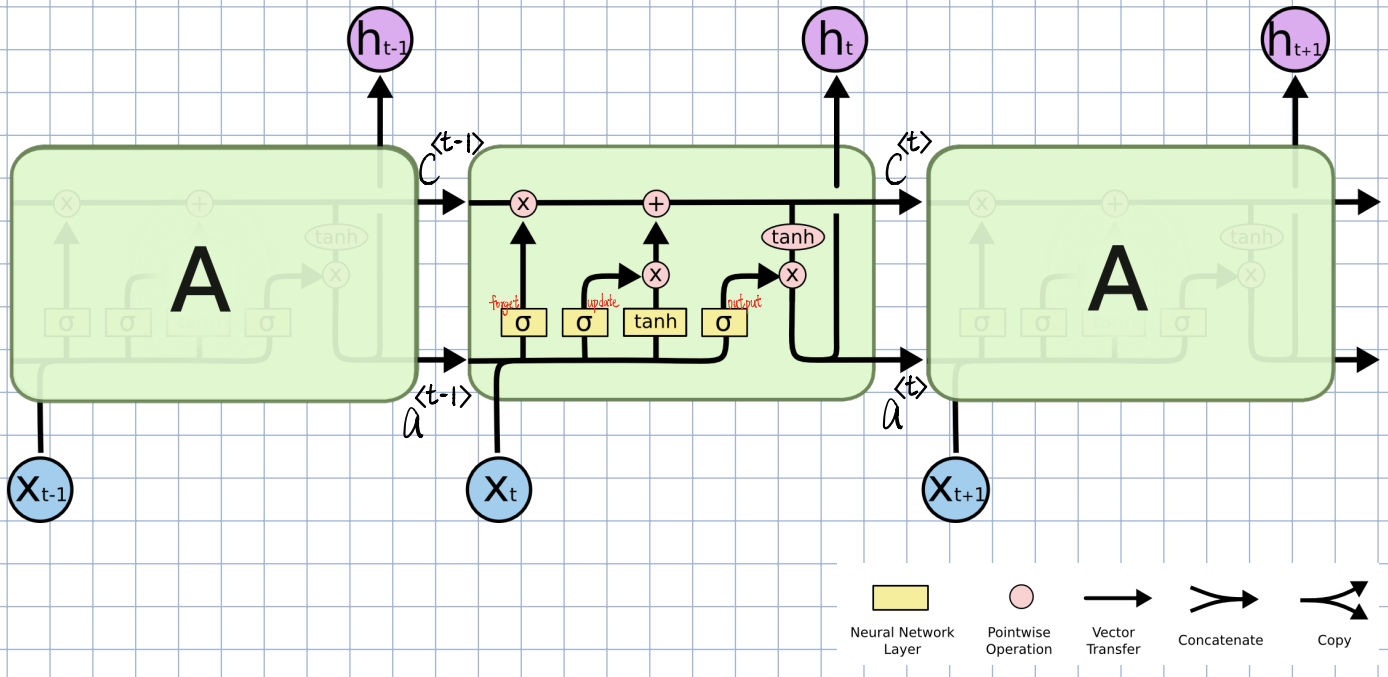
update gate $\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$

forget gate $\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$

output gate $\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$

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

$$a^{<t>} = \Gamma_o * \tanh(c^{<t>})$$



Bidirectional RNN (BRNN)

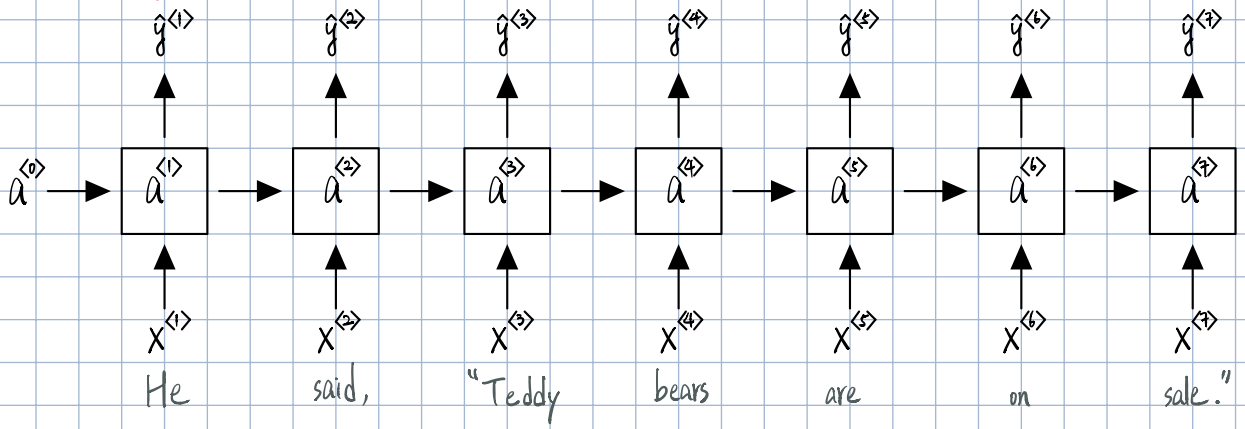
We need to get information from the future to predict the current output.

E.g.:

He said, "Teddy (a name) Roosevelt was a great president."

He said, "Teddy (not a name) bears are on sale."

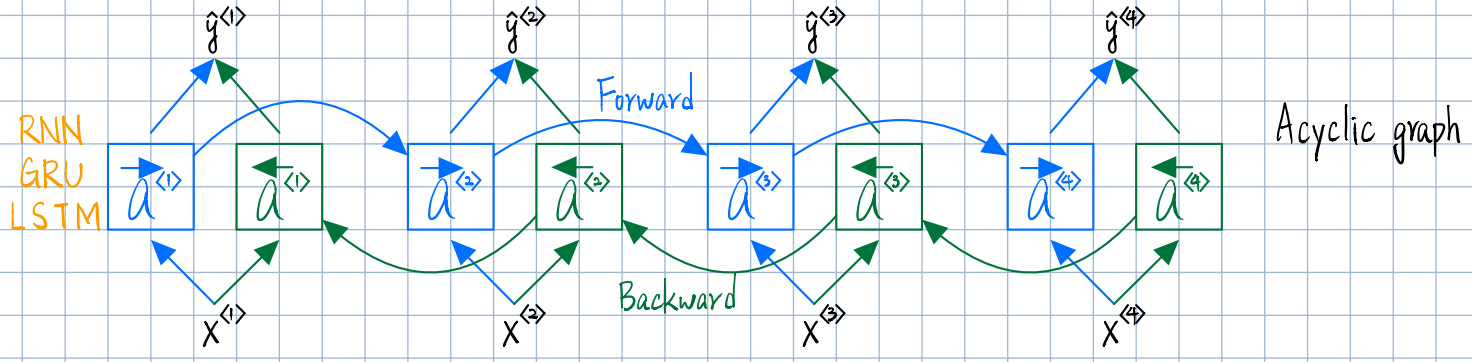
Using unidirectional RNN is just impossible to say if Teddy is a name or not.



Unidirectional structure doesn't work well regardless of using standard RNN, GRU or LSTM or not.

Bidirectional RNN

$$\text{Prediction: } \hat{y}^{\langle t\rangle} = g\left(W_y\left[\overrightarrow{a}^{\langle t\rangle}, \overleftarrow{a}^{\langle t\rangle}\right] + b_y\right)$$



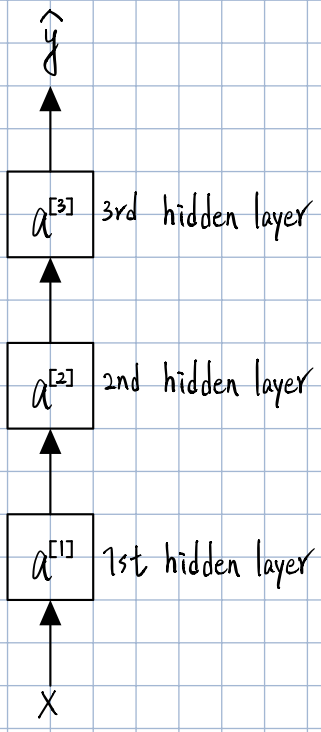
Acyclic graph

Notice that Forward and Backward connection are both forward propagation. Backward connection is not backpropagation. In other words, forward propagation composes of both Forward and Backward connection.

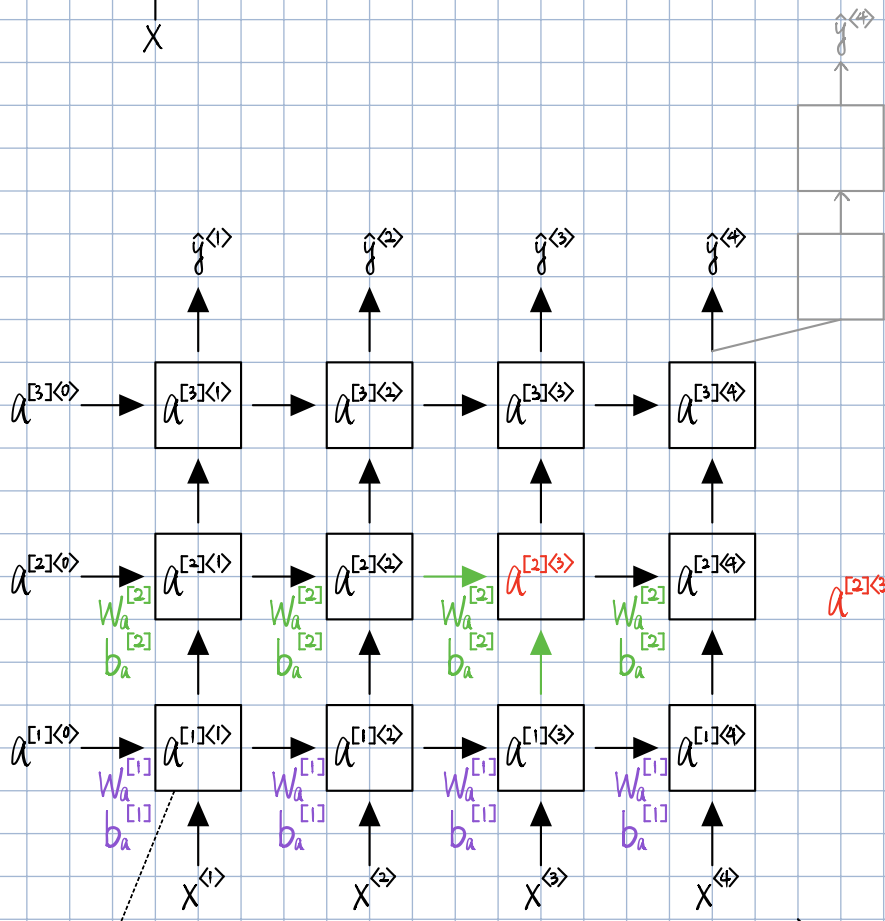
In prediction, to compute $\hat{y}^{\langle 3\rangle}$, information from $x^{\langle 1\rangle}$ to $x^{\langle 4\rangle}$ are used.

Deep RNNs

Recall



$a^{[l]\langle t\rangle}$: l is associated with the layer l.
t is associated with the time t.



$$a^{[2]\langle 3\rangle} = g\left(W_a^{[2]}\left[a^{[2]\langle 2\rangle}, a^{[1]\langle 3\rangle}\right] + b_a^{[2]}\right)$$

Blocks could be standard RNNs, GRU, LSTM

The overall structure could also be BRNN.