Recurrent Neural Network

PHYS591000 Spring 2021

Reading:

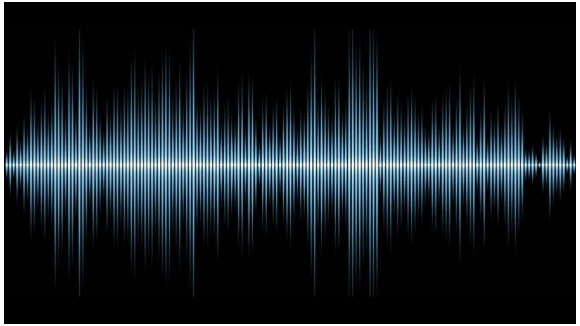
- Stanford cs-230 RNN
- Colah’s blog

Sequence Data

Image

Sequence Data

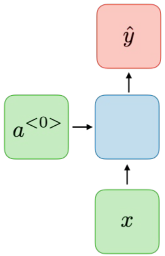
Image



Sequence - Audio

RNN

One-to-one

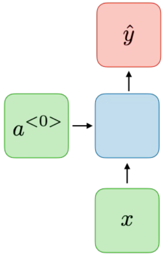


Binary Classification

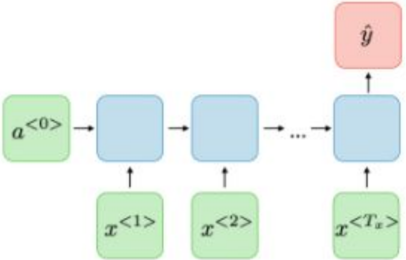
Pass vs Fail

References

One-to-one



Many-to-One



Binary Classification

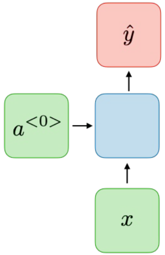
Pass vs Fail

Sentiment Classification

“There is nothing to like in this movie.”

References

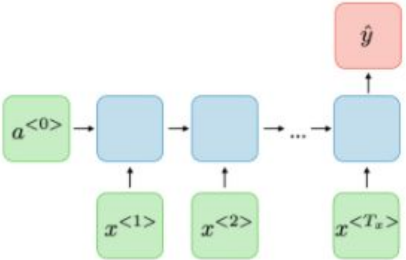
One-to-one



Binary Classification

Pass vs Fail

Many-to-One



Sentiment Classification

“There is nothing to like in this movie.”

One-to-Many

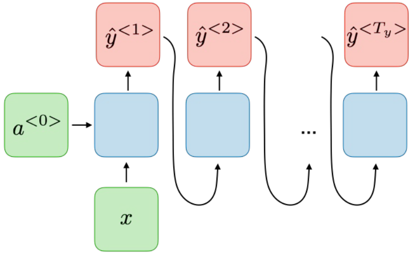
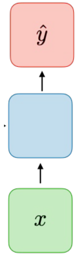


Image Captioning

A man is running.

References

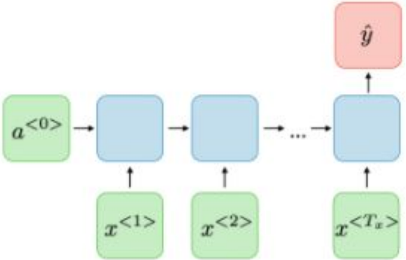
One-to-one



Binary Classification

Pass vs Fail

Many-to-One



Sentiment Classification

“There is nothing to like in this movie.”

One-to-Many

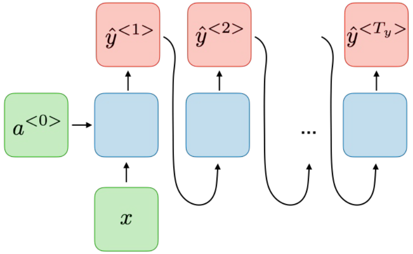
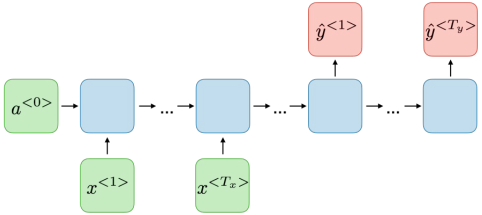


Image Captioning

A man is running.

Many-to-Many



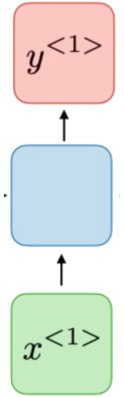
Machine Translation

«Hey Siri, où puis-je acheter une Tesla?»

“Hey Siri,
where can I buy a Tesla?”

Neurons with Recurrence

Output
vector

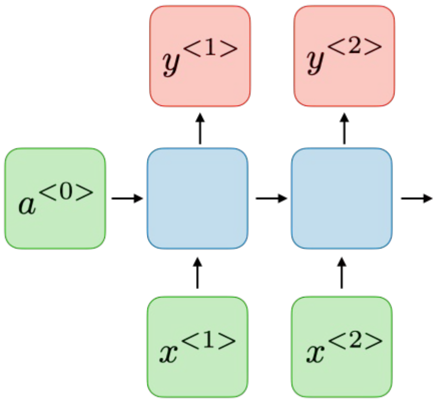


Input
vector

$$\boxed{y^{<1>}} = f(\boxed{x^{<1>}})$$

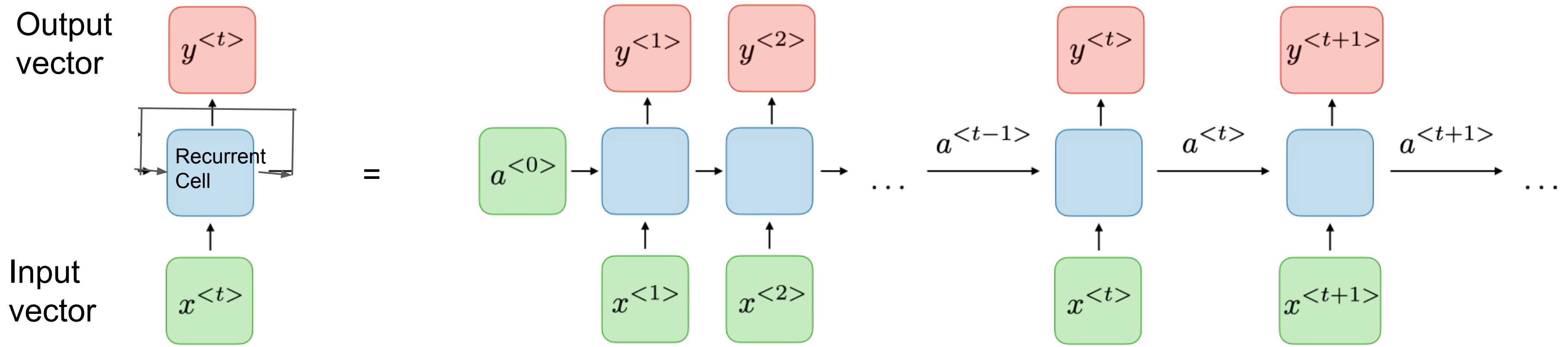
Output Input

Output vector



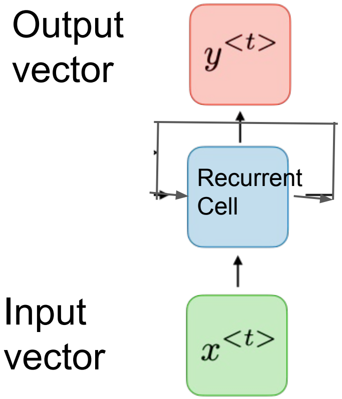
Input vector

$$\underset{\text{Output}}{\boxed{y^{<t>}}} = f(\underset{\text{Input}}{\boxed{x^{<t>}}}, \underset{\text{Past memory}}{\boxed{a^{<t-1>}}})$$



$$\underbrace{\boxed{y^{<t>}}}_{\text{Output}} = f(\underbrace{\boxed{x^{<t>}}}_{\text{Input}}, \underbrace{\boxed{a^{<t-1>}}}_{\text{Past memory}})$$

Recurrent Neural Network



Apply a recurrence relation at every step to process a sequence:

Function with weights W

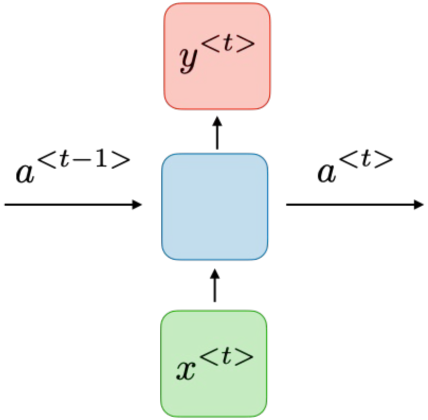
$$a^{<t>} = f_W(\boxed{x^{<t>}}, \boxed{a^{<t-1>}})$$

Cell State Input Old State

The same function and set of parameters are used at every time step

Cell state $a^{<t>}$ is updated at each time step t as a sequence is processed.

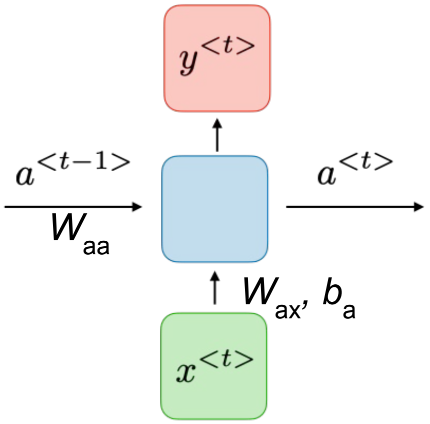
RNN State Update and Output



Input Vector

$x^{<t>}$

RNN State Update and Output



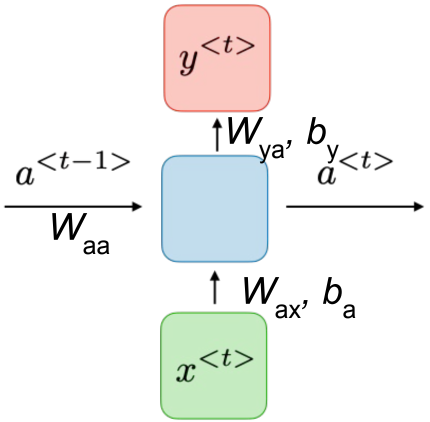
Update Hidden State

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)}$$

Input Vector

$$x^{<t>}$$

RNN State Update and Output



Output Vector

$$\boxed{y^{<t>} = g_2(W_{ya}a^{<t>} + b_y)}$$

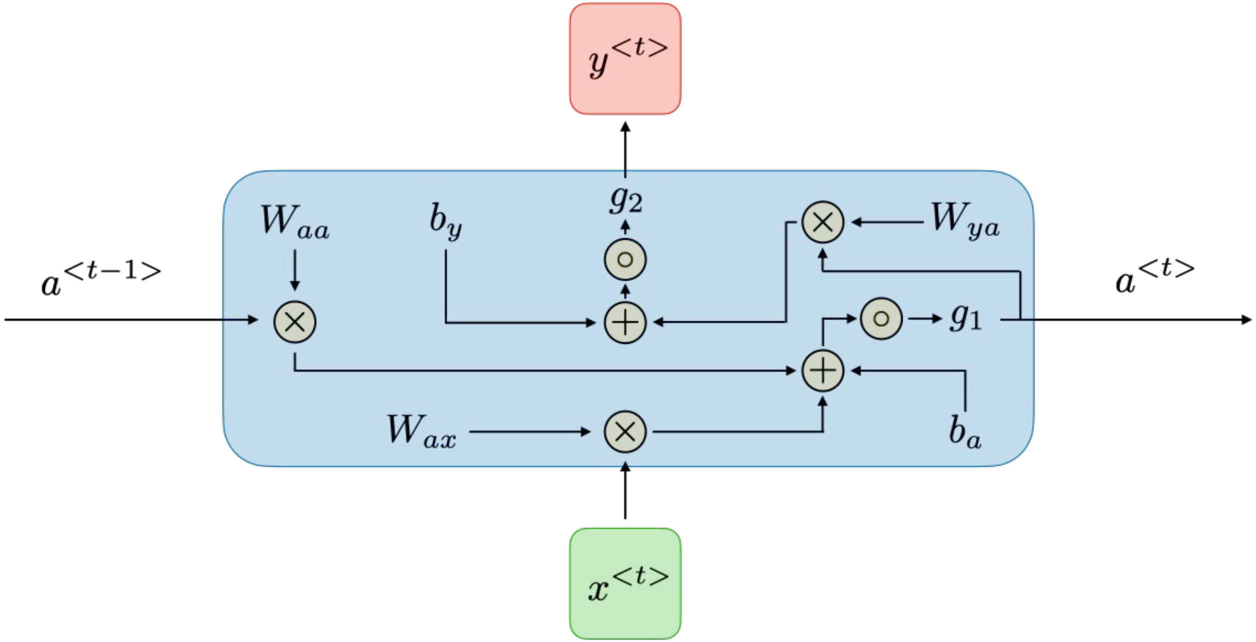
Update Hidden State

$$\boxed{a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)}$$

Input Vector

$$x^{<t>}$$

For each timestep t, the activation $a^{<t>}$ and the output $y_{<t>}$



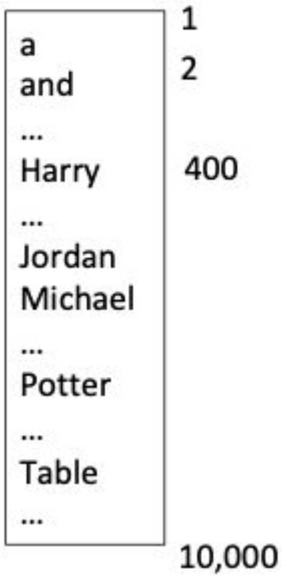
W_{aa}, W_{ax}, b_a,W_{ya}, b_y are coefficients that are shared temporally and g_1 , g_2 activation functions.

Pros and Cons of RNN

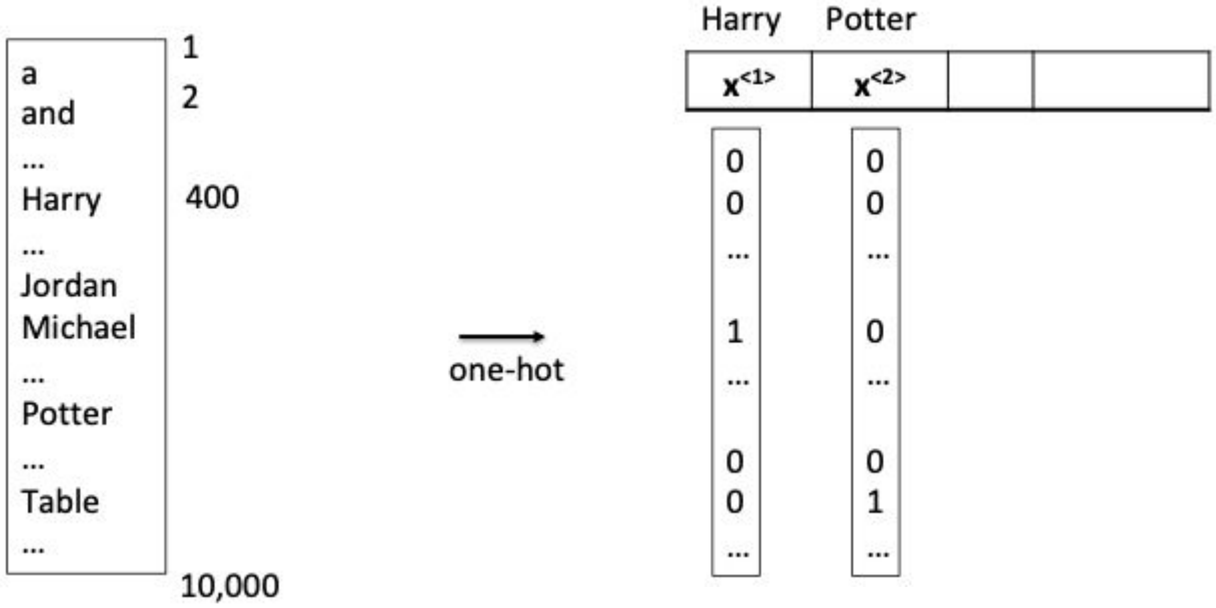
Advantages	Drawbacks
• Possibility of processing input of any length • Model size not increasing with size of input • Computation takes into account historical information • Weights are shared across time	• Computation being slow • Difficulty of accessing information from a long time ago • Cannot consider any future input for the current state

RNN in Practice

Encoding for numerical representation

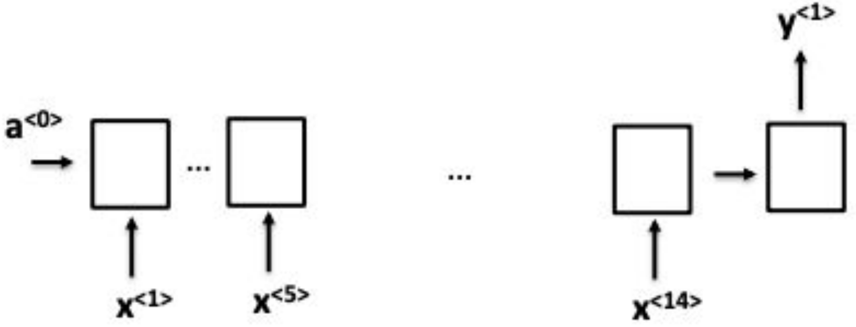


One-hot encoding



Backpropagation Through Time

“I grew up in France and moved to United States, therefore I speak ____”



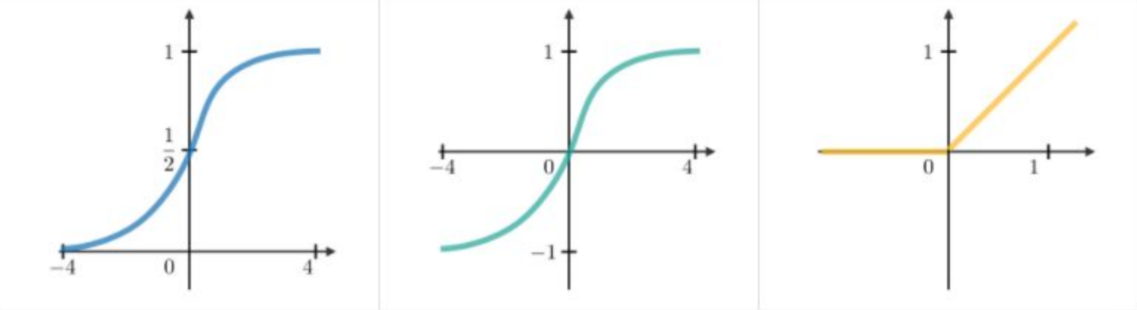
$$\mathcal{L}(\hat{y}, y) = \sum_{t=1}^{T_y} \mathcal{L}(\hat{y}^{<t>}, y^{<t>})$$

$$\frac{\partial \mathcal{L}^{(T)}}{\partial W} = \sum_{t=1}^{T} \frac{\partial \mathcal{L}^{(T)}}{\partial W}\bigg|_{(t)}$$

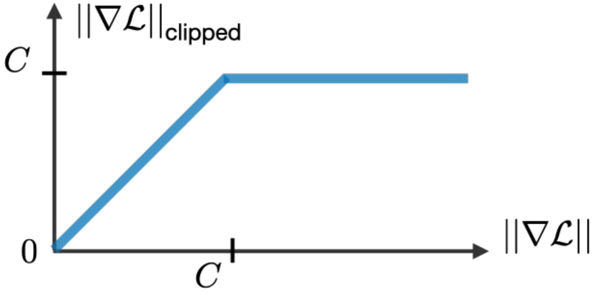
Vanishing/Exploding Gradients

Vanishing gradients
(propagation of small gradients):
Gated Structure

Sigmoid	Tanh	RELU
$g(z) = \frac{1}{1+e^{-z}}$	$g(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}}$	$g(z) = \max(0, z)$



Exploding gradients
(propagation of big gradients):
Gradient Clipping



Types of Gates

$$\boxed{\Gamma = \sigma(Wx^{<t>} + Ua^{<t-1>} + b)}$$

W,U,b are coefficients specific to the gate and σ is the sigmoid function.

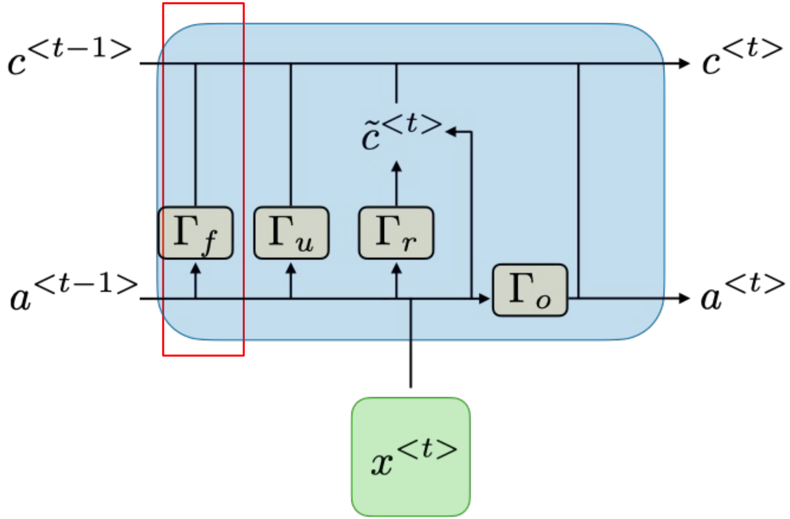
Type of gate	Role
Update gate Γ_u	How much past should matter now?
Relevance gate Γ_r	Drop previous information?
Forget gate Γ_f	Erase a cell or not?
Output gate Γ_o	How much to reveal of a cell?

Long-Short Term Memory (LSTM)

LSTM

1) Forget 2) Update 3) Store 4) Output

LSTM forget irrelevant part of the previous state.

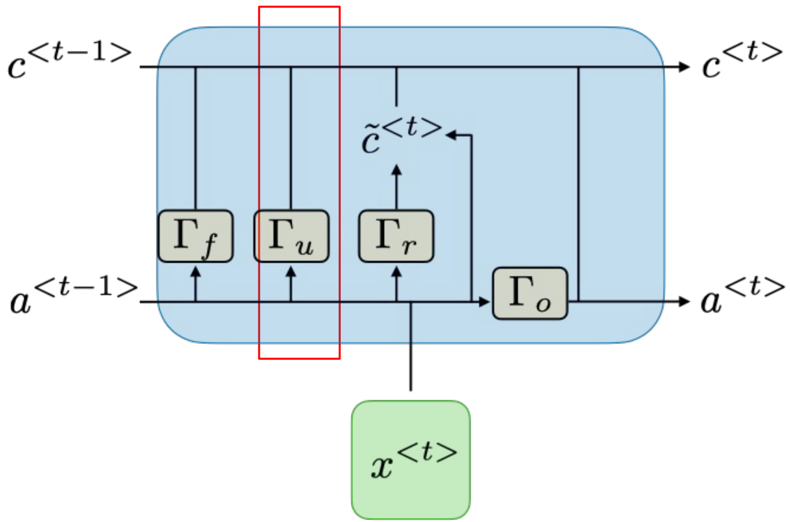


$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$

LSTM

1) Forget 2) Update 3) Store 4) Output

LSTM selectively update cell state value

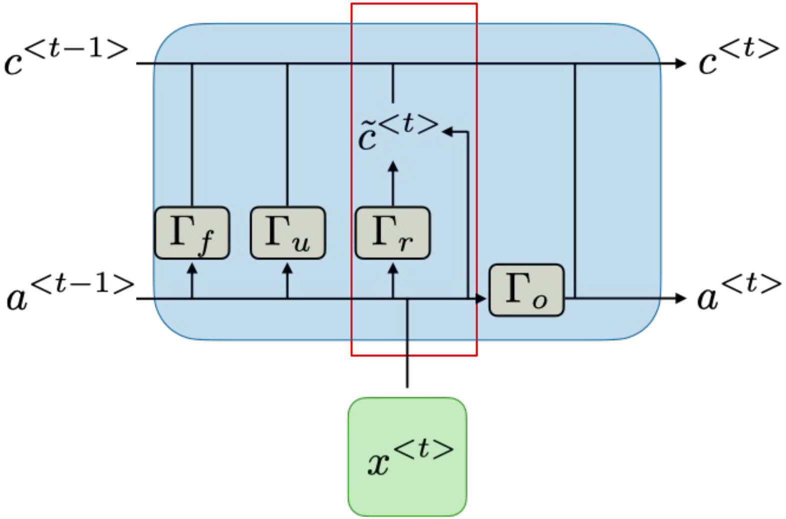


$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$

LSTM

1) Forget 2) Update 3) Store 4) Output

LSTM store relevant new information into the cell state

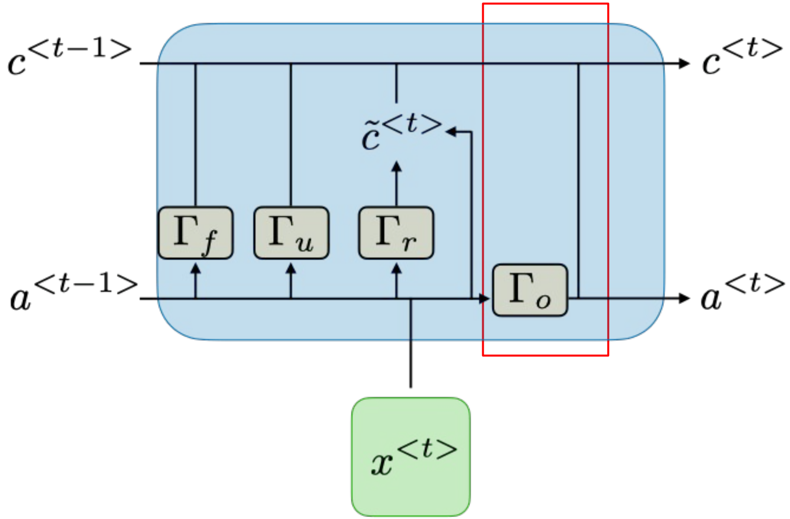


$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$\Gamma_o \star c^{<t>}$

LSTM

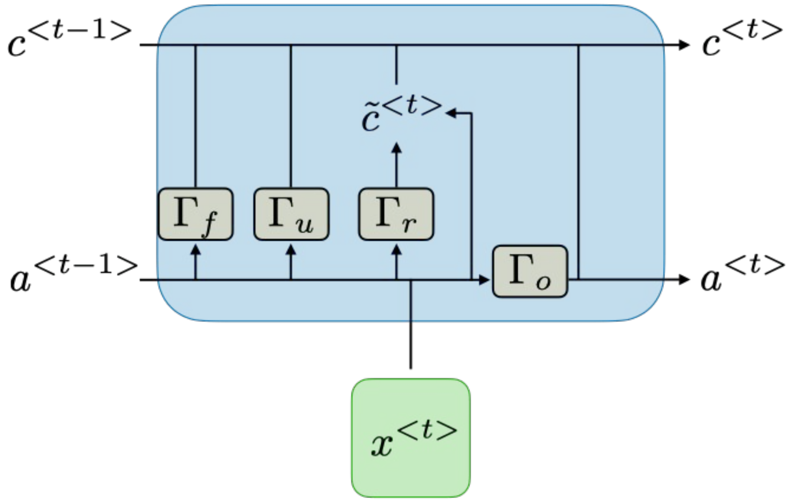
1) Forget 2) Update 3) Store 4) Output

The output ate controls what information is sent to the next time step.



$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$\Gamma_o \star c^{<t>}$

Long Short-Term Memory units (LSTM) deal with the vanishing gradient problem encountered by traditional RNN



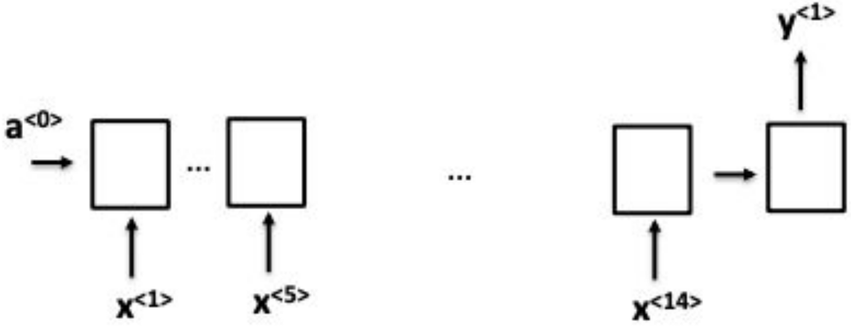
$\tilde{c}^{<t>}$	$\tanh(W_c[\Gamma_r \star a^{<t-1>}, x^{<t>}] + b_c)$
$c^{<t>}$	$\Gamma_u \star \tilde{c}^{<t>} + \Gamma_f \star c^{<t-1>}$
$a^{<t>}$	$\Gamma_o \star c^{<t>}$

LSTM Key-Concepts

1. Maintain a separate cell state from what is outputted
2. Use gates to control the flow information
   a. **Forget** gate gets rid of irrelevant information
   b. **Update** selected cell state
   c. **Store** relevant information from current input
   d. **Output** gate returns a filtered version of the cell state
3. Backpropagation through time with uninterrupted gradient flow

Backpropagation Through Time

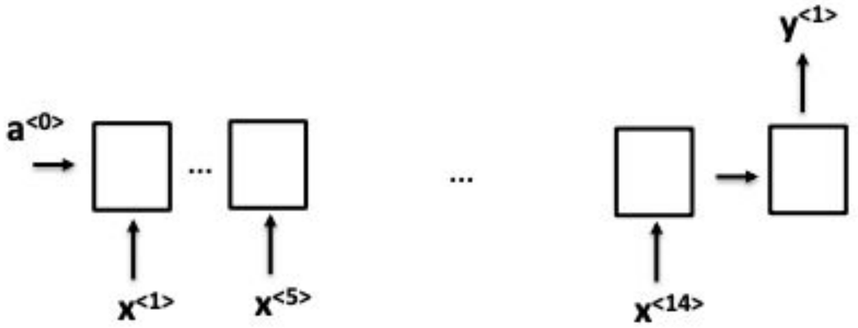
“I grew up in France and moved to United States, therefore I speak ____”



Backpropagation Through Time

Less relevant

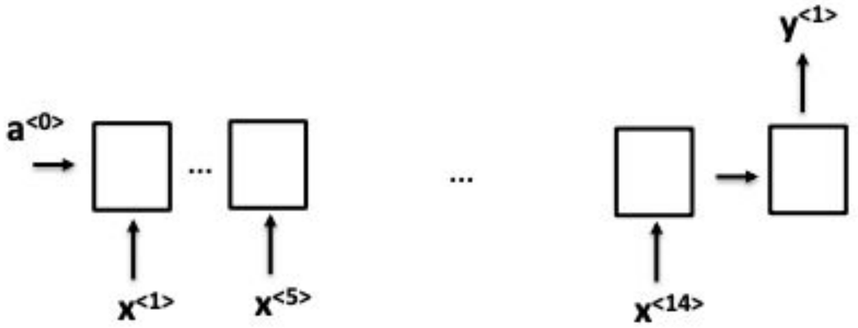
"I grew up in France and moved to United States, therefore I speak _____"



Backpropagation Through Time

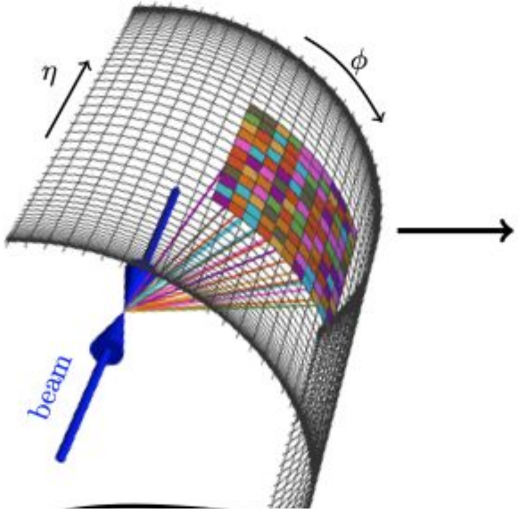
More relevant

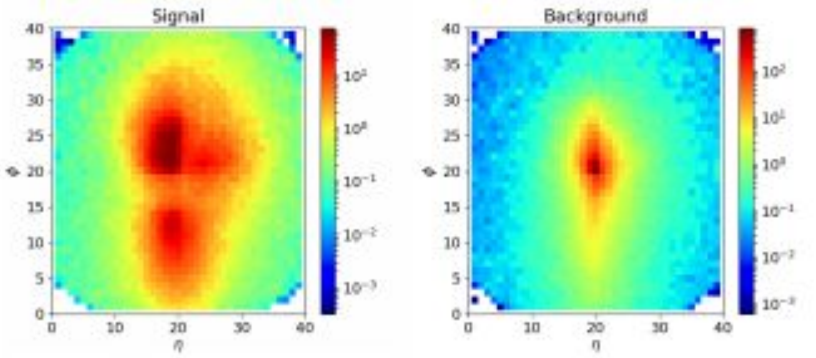
"I grew up in France and moved to United States, therefore I speak ____"



In-class Quiz and Lab

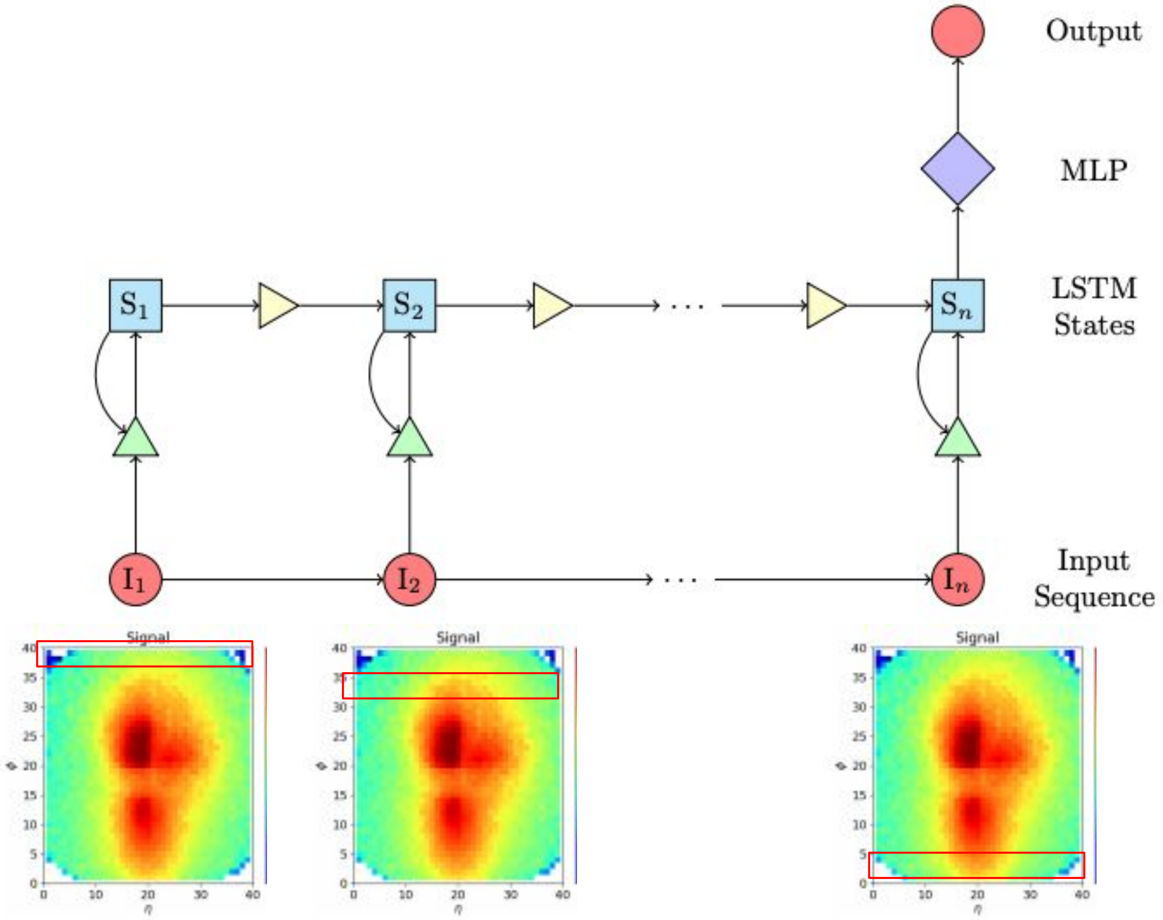
Jet Tagging as Images



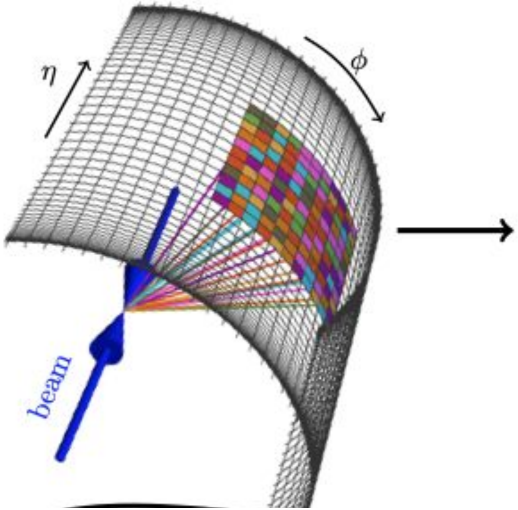


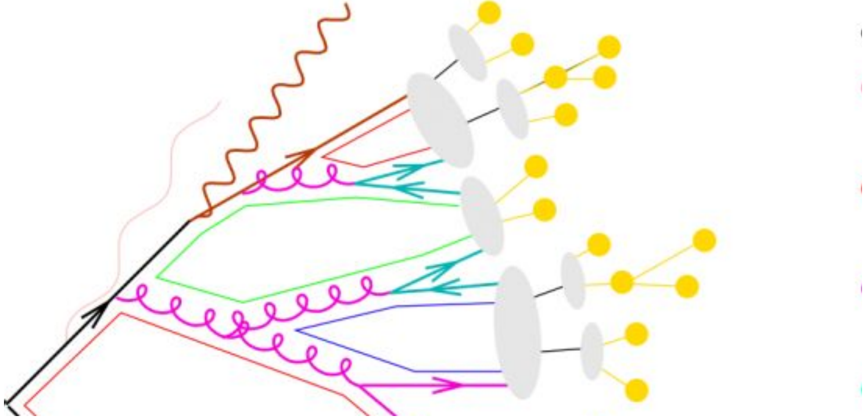
https://arxiv.org/pdf/1902.09914

https://arxiv.org/pdf/1607.08633.pdf



Jet Tagging as Images

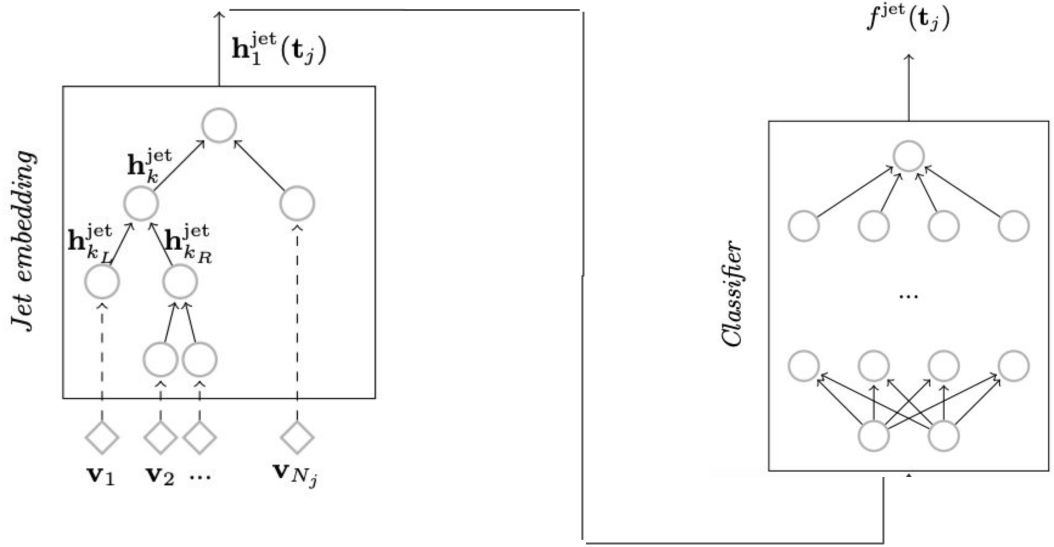




Credit: Angelo Monteux [URL]

Credit: Angelo Monteux [URL]

https://arxiv.org/pdf/1702.00748.pdf



$\boldsymbol{v}_i$= properties of each hadrons, e.g. 4-vector, charge.

References

- Standord S230 Amidi, Recurrent Neural Network
- MIT 6.S191 Soleiman, Deep Sequence Modeling
- LSTM Colah’s blog