Recurrent Neural Network Workshop

Austin Juhl

Resources

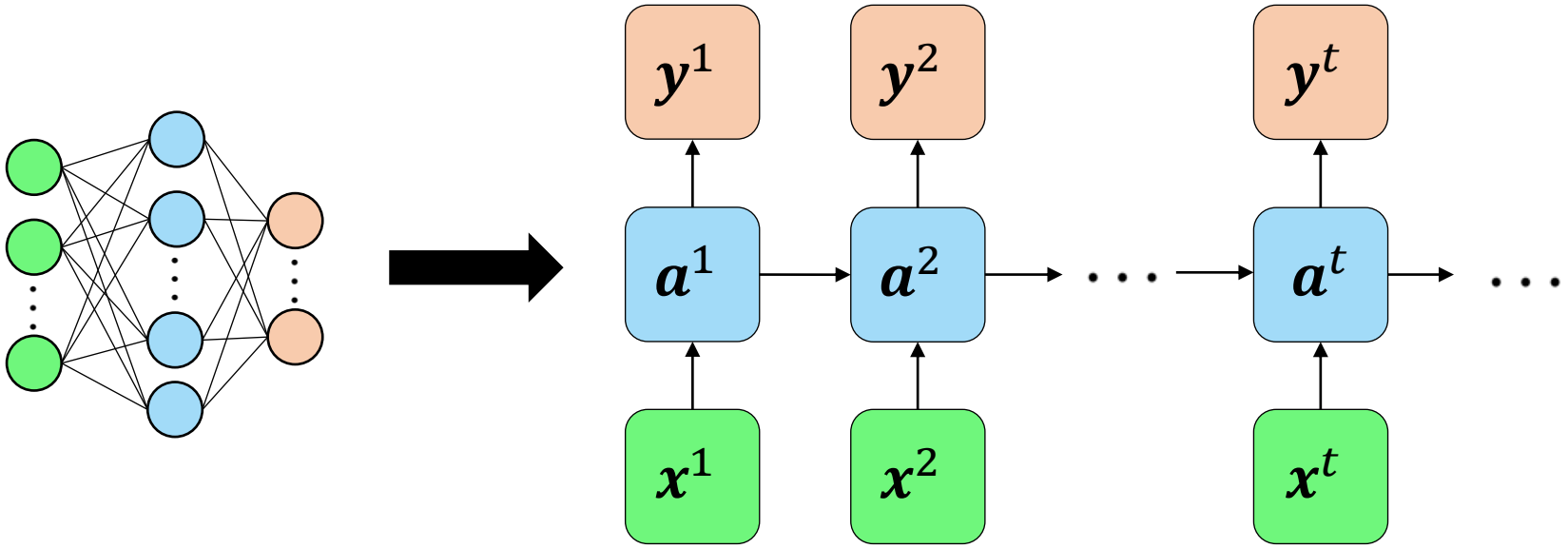
- Andrew Ng – Coursera
  - https://www.coursera.org/learn/nlp-sequence-models
- Shervine Amidi – Stanford
  - https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks
- Judahsemi
  - https://github.com/judahsemi/Dino-Name-Generator

Sequence Models

- Machine Learning models that can input and/or output sequence data.
- Examples:
  - Sentiment Classification "This movie is very bad" → ★☆☆
  - Music Generation  → 
  - Translation "Good morning" → "Buongiorno"
  - Video Activity Recognition  → Dancing

Why not use a standard network?

- Multi-Layer Perceptron (MLP)
  - Cannot processes input of varying lengths.
  - Does not consider historical (or future) information.



Notation and Review

$x^t \in R^k$ - t^{th} training element with k features.

$y^t \in R^l$ - t^{th} output element with l features.

$a^t \in R^m$ - t^{th} activation value with m features.

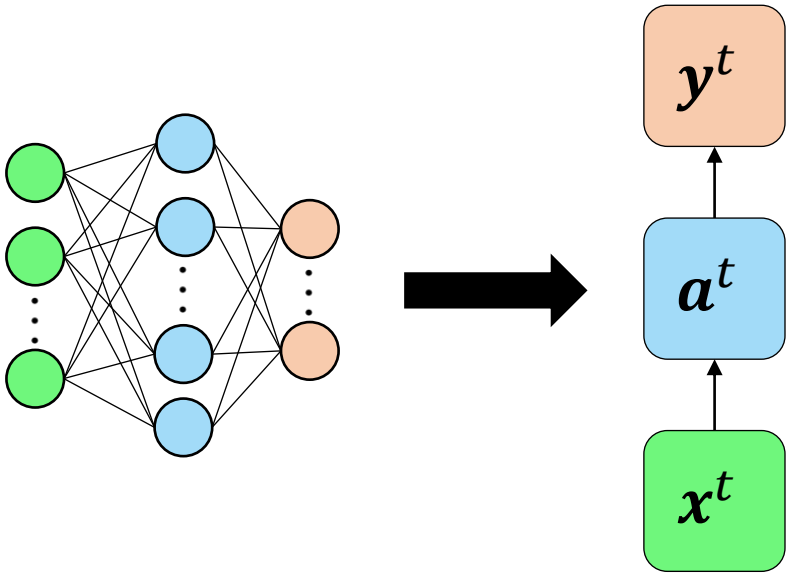
$b_a \in R^m$ - activation bias.

$W_a \in R^{m,k}$ - activation weight.

$b_y \in R^l$ - output bias.

$W_y \in R^{l,m}$ - output weight.

ϕ - arbitrary activation function



MLP:

$$a^t := \phi_1(W_a x^t + b_a)$$
$$y^t := \phi_2(W_y a^t + b_y)$$

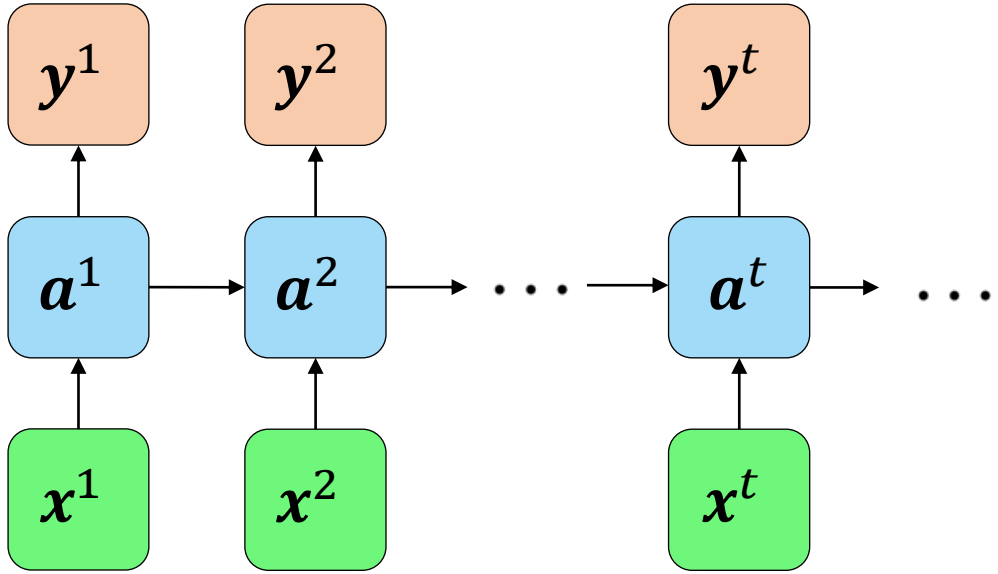
Recurrent Neural Network (RNN)

$x^{i,t} \in R^k$ - t^{th} input element of i^{th} sequence with k features.

$y^{i,t} \in R^l$ - t^{th} output element of i^{th} sequence with l features.

T_x - input sequence length.

T_y - output sequence length.



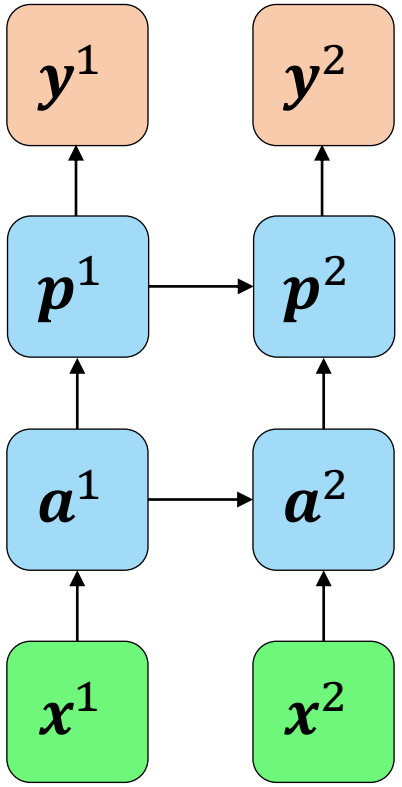
RNN:

$$a^t := \phi(W_{aa}a^{t-1} + W_{ax}x^t + b_a)$$
$$y^t := \phi(W_y a^t + b_y)$$

Two layered RNN

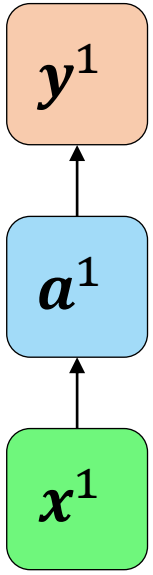
RNN:

$$a^t := \phi(W_{aa}a^{t-1} + W_{ax}x^t + b_a)$$
$$p^t := \phi(W_{aa}p^{t-1} + W_{ax}a^t + b_a)$$
$$y^t := \phi(W_y p^t + b_y)$$



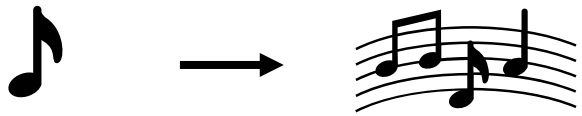
Types of RNN's

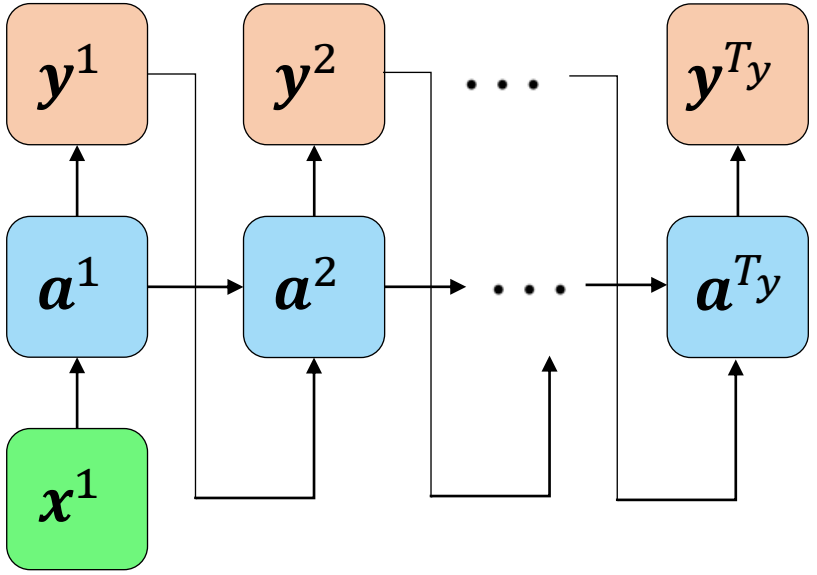
- One-to-One
  - $T_x = T_y$=1
  - MLP



Types of RNN's

- One-to-Many
  - $T_x = 1, T_y > 1$
  - Music generation.

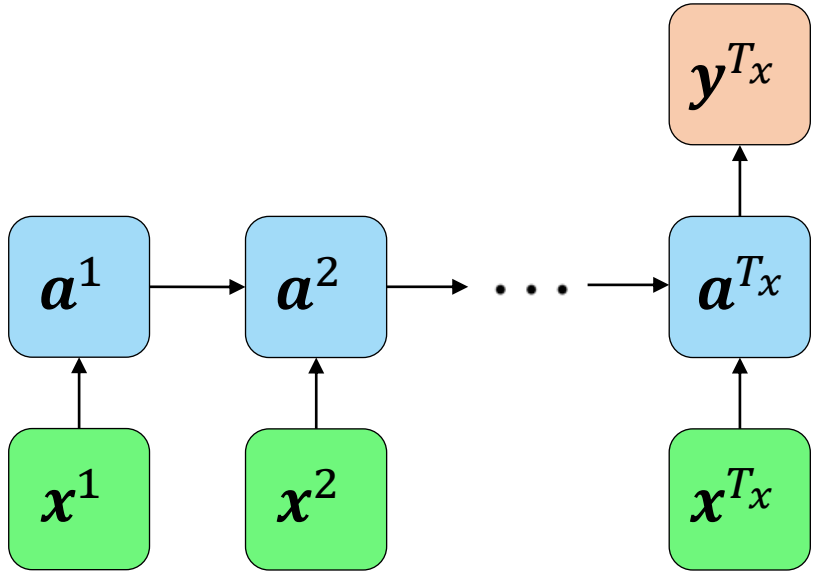




Types of RNN's

- Many-to-One
  - $T_x > 1, T_y = 1$
  - Sentiment Classification.

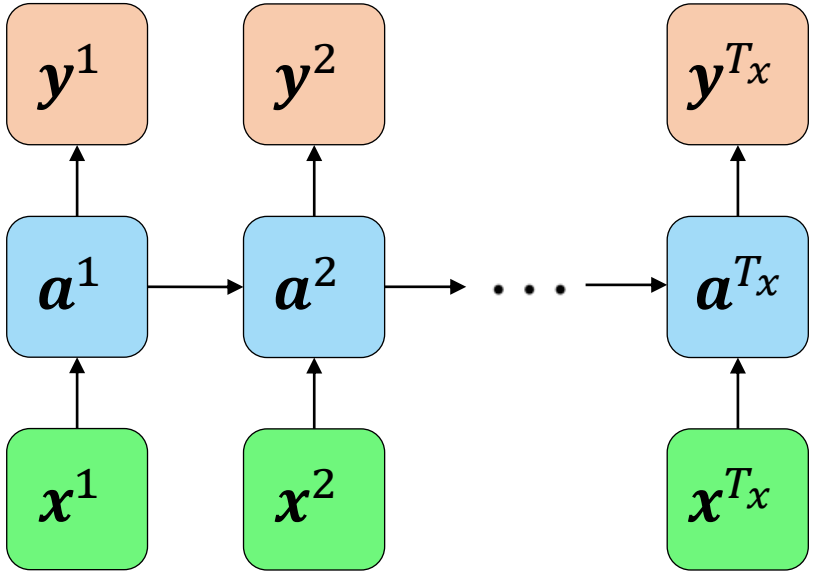
"This movie is very bad" ⟶ ★☆☆



Types of RNN's

- Many-to-Many
  - $T_x = T_y$
  - Name recognition.

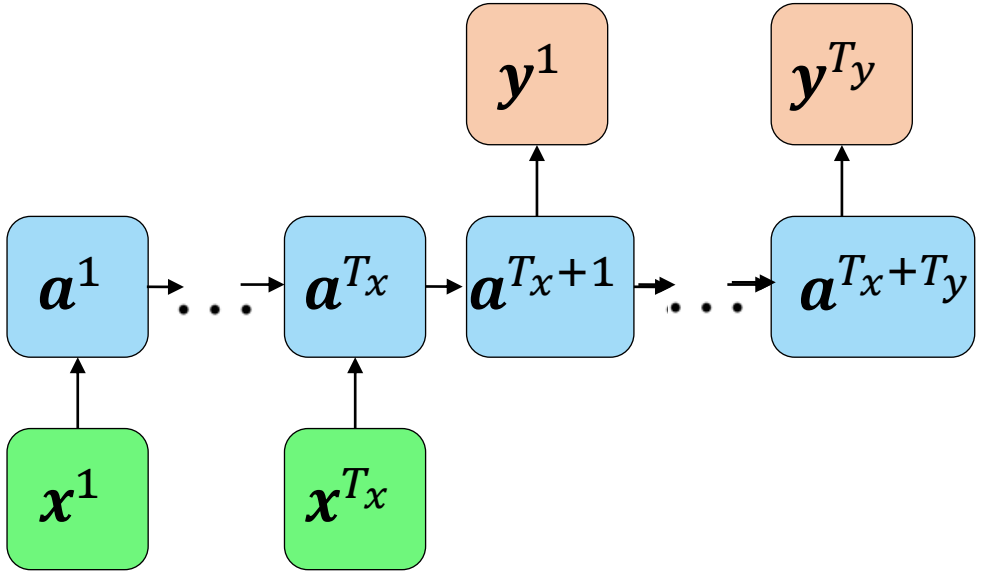
"My name is Austin." ⟶ 0 0 0 1



Types of RNN's

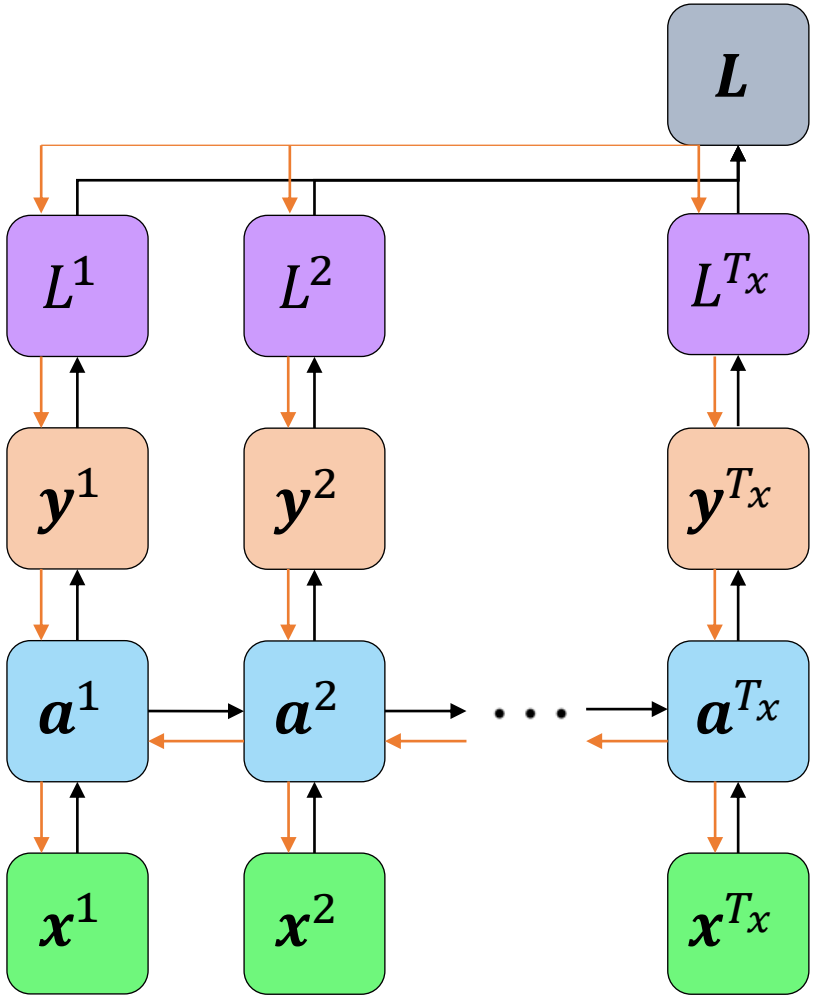
- Many-to-Many
  - $T_x \neq T_y$
  - Translation.

"Good morning" ⟶ "Buongiorno"



Backpropagation Through Time

- Loss for the Sequence:
  - $L(\hat{y}, y) = \sum_{t=1}^{T_y} L^t(\hat{y}^t, y^t)$
  - MSE, Cross Entropy, etc.
- Backpropagation:
  - Calculate the derivative of the loss $\boldsymbol{L}$ with respect to the parameters W_a, $\mathrm{b_a}$, $\mathrm{W_y}$, and b_y.
  - Gradient decent is used to update the parameters.

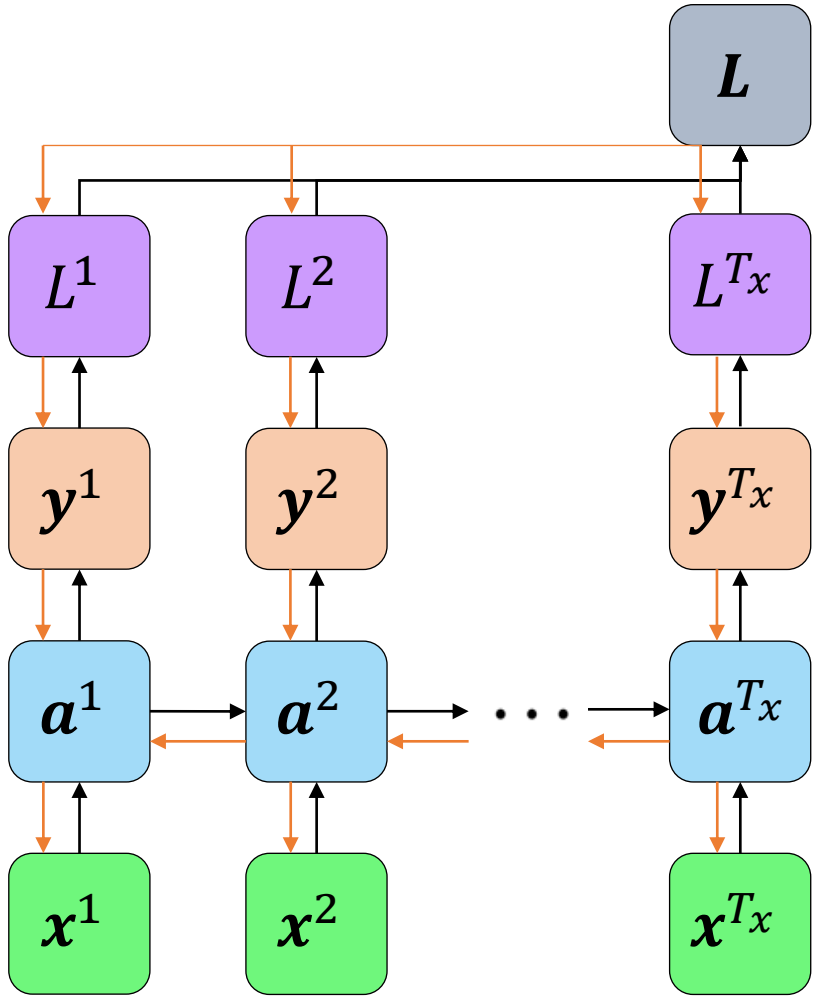


Vanishing Gradient

$$\frac{\partial \boldsymbol{L}}{\partial W_{aa}} = \frac{\partial L^1}{\partial W_{aa}} + \cdots + \frac{\partial L^{Tx}}{\partial W_{aa}}$$

$$\frac{\partial L^1}{\partial W_{aa}} = \frac{\partial L^1}{\partial y^1}\left(\frac{\partial y^1}{\partial a^1}\left(\frac{\partial a^1}{\partial W_{aa}} + \frac{\partial a^1}{\partial a^0}\right)\right)$$
$$= \frac{\partial L^1}{\partial y^1}\frac{\partial y^1}{\partial a^1}\frac{\partial a^1}{\partial W_{aa}} + \frac{\partial L^1}{\partial y^1}\frac{\partial y^1}{\partial a^1}\phi'(W_{aa}a^0 \cdots)W_{aa}$$

$$\frac{\partial L^2}{\partial W_{aa}} = \frac{\partial L^2}{\partial y^2}\left(\frac{\partial y^2}{\partial a^2}\left(\frac{\partial a^2}{\partial W_{aa}} + \frac{\partial a^2}{\partial a^1}\left(\frac{\partial a^1}{\partial W_{aa}} + \frac{\partial a^1}{\partial a^0}\right)\right)\right)$$
$$= \frac{\partial L^2}{\partial y^2}\frac{\partial y^2}{\partial a^2}\frac{\partial a^2}{\partial W_{aa}} + \frac{\partial L^2}{\partial y^2}\frac{\partial y^2}{\partial a^2}\frac{\partial a^1}{\partial W_{aa}}\phi'(W_{aa}a^1 \cdots)W_{aa} + \frac{\partial L^2}{\partial y^2}\frac{\partial y^2}{\partial a^2}\phi'(W_{aa}a^1 \cdots)\phi'(W_{aa}a^0 \cdots)W_{aa}^2$$



RNN elements are affected mostly locally.

Alternate Form

Let $W_a = [W_{aa}\ W_{ax}]$.

$$a^t := \phi(W_{aa}a^{t-1} + W_{ax}x^t + b_a)$$
$$y^t := \phi(W_y a^t + b_y)$$

$$a^t := \phi(W_a[a^{t-1}; x^t] + b_a)$$
$$y^t := \phi(W_y a^t + b_y)$$

Gated Recurrent Unit (GRU)

- Overcomes the vanishing gradient problem by using gates to determine which information should be retained and updated.

$$\Gamma_r = \sigma(W_r[a^{t-1}; x^t] + b_r)$$

Retain previous information?

$$\Gamma_u = \sigma(W_u[a^{t-1}; x^t] + b_u)$$

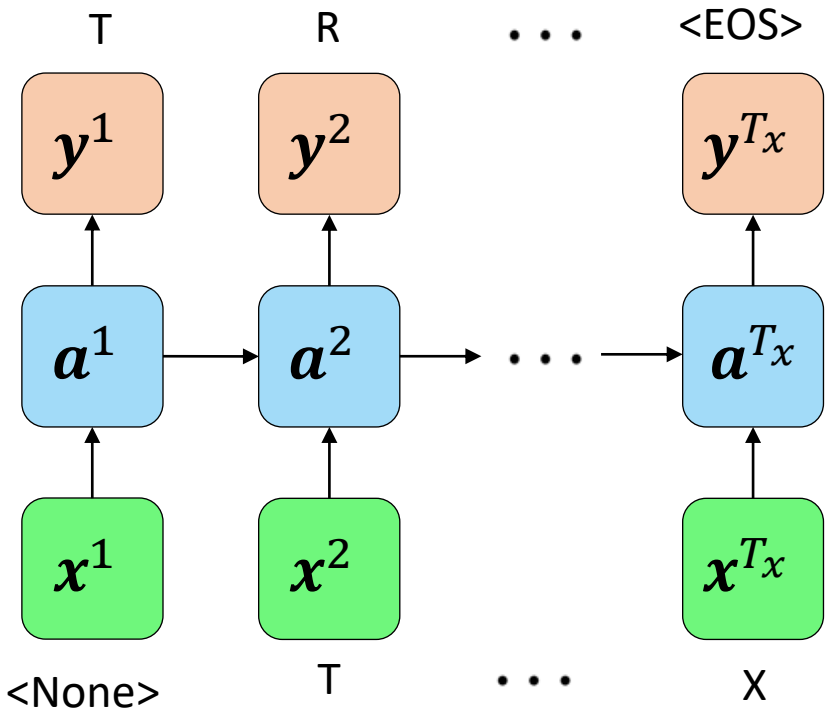
How much of the past should matter now?

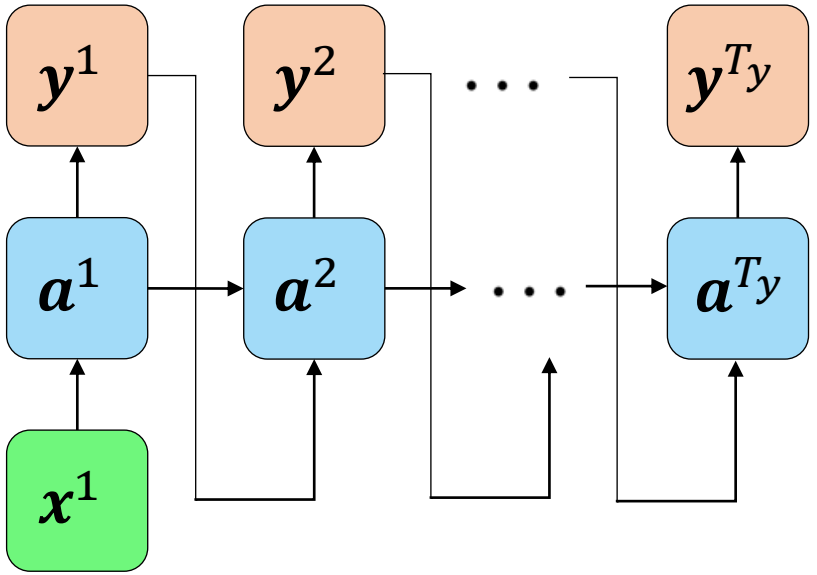
$$\tilde{a}^t = \phi(W_a[\Gamma_r * a^{t-1}; x^t] + b_a)$$

$$a^t = \Gamma_u * \tilde{a}^t + (1 - \Gamma_u) * a^{t-1}$$

Dino-Name Generator

- Model: GRU or LSTM
- Training set: List of dinosaur names
- Goal: Generate new dinosaur names





Resources

- Andrew Ng – Coursera
  - https://www.coursera.org/learn/nlp-sequence-models
- Shervine Amidi – Stanford
  - https://stanford.edu/~shervine/teaching/cs-230/cheatsheet-recurrent-neural-networks
- Judahsemi
  - https://github.com/judahsemi/Dino-Name-Generator