Recurrent Neural Network

Tianchu.Zhao@uts.edu.au

Why Recurrent

Sequence data

- Sequence data means that there exists a relation between data points
- The relation can be time, spatial, semantic ...
- Each sequence consists of multiple data points can have vary length

Neural network on sequence data

- For vary length input, we need vary length neuron (parameters) in each layer
- If we capture a certain feature in one area, this feature can not be broadcasted to another area
- Can not accommodate the large amount of parameters

- To address the above problems, we use Recurrent Neural Network(RNN)

Applications of RNN

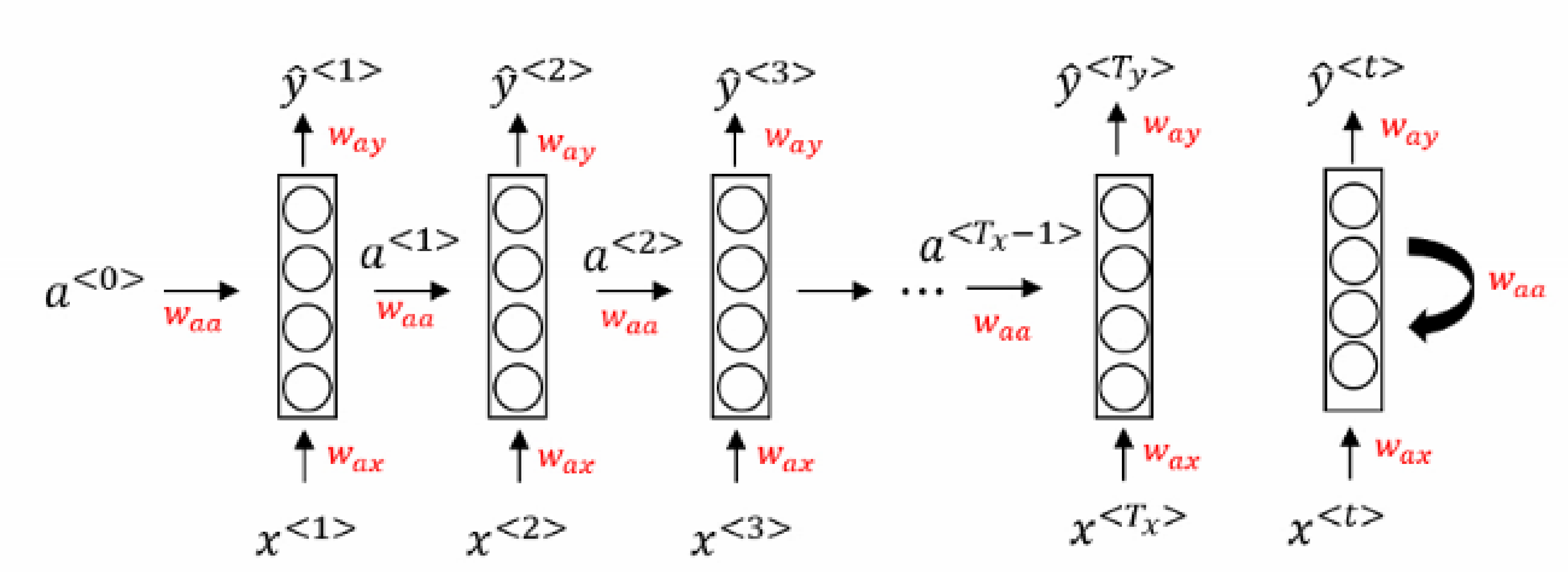
- name entity recognition: sentence to location of noun, action
  - eg, what is the noun(machine learning) action(training)
- sentiment analysis: sentence to score
- speech recognition: audio to text
- machine translation: French to English
- video activity recognition: video to "running"
- music generation: notes to music sheet/audio
- DNA/Protein analysis: sequence to property score

How Recurrent

Mathematic notation

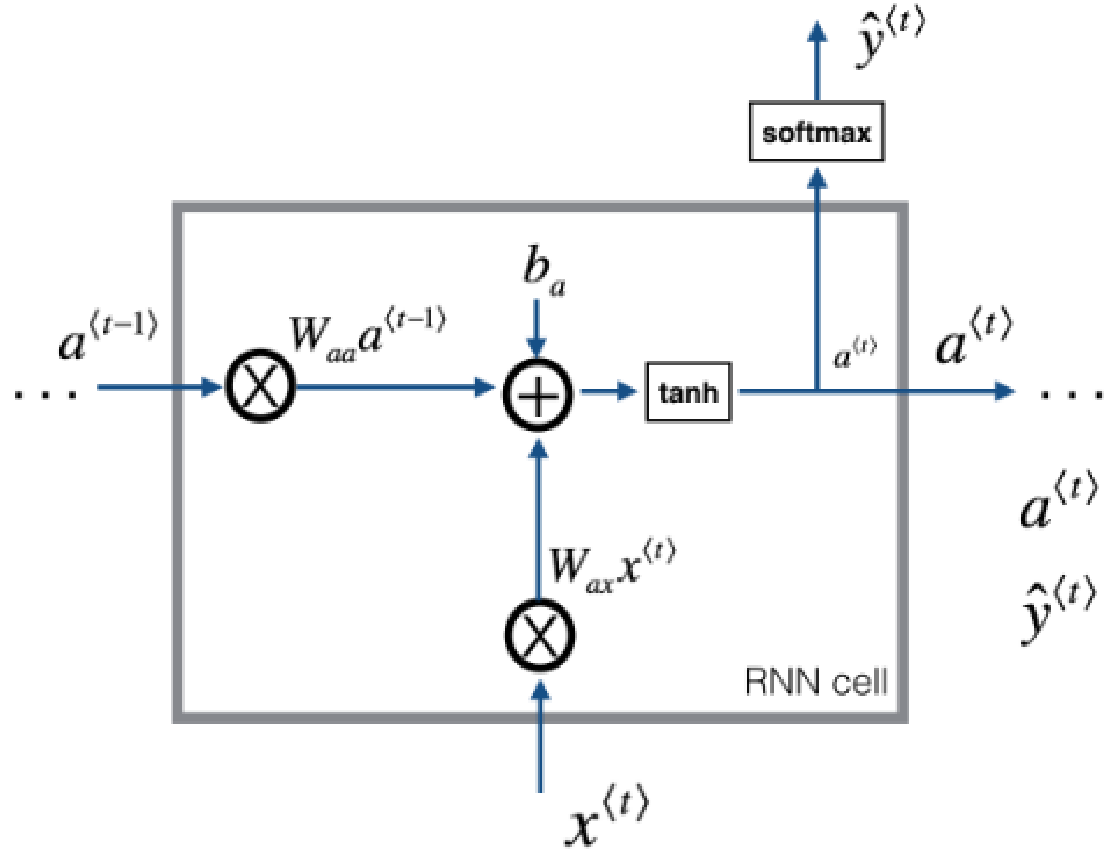
- x, data point, each data point is a sequence
- $x^{(t)}$, the t-th element in a sequence
- $y^{(t)}$, the t-th label in a sequence
- T_x , the input length
- T_y , the output length
- put together we have
- $x^{(i)(t)}$, i-th sequence t-th element
- $y^{(i)(t)}$, i-th sequence t-th label
- $T_x^{(i)}$, input length of i-th sequence

RNN Structure



- The hidden layer of each corresponding time step will receive the activation value $a^{(t-1)}$, as a input, $a^{(0)}$ is usually 0
- The parameters are input W_{ax}, W_{aa}, W_{ay} where we share parameters at every timestamp

Zoom in each neuron



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

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

Forward pass

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

$$a^{\langle t \rangle} = g_1(W_{aa}a^{\langle t-1 \rangle} + W_{ax}x^{\langle t \rangle} + b_a)$$

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

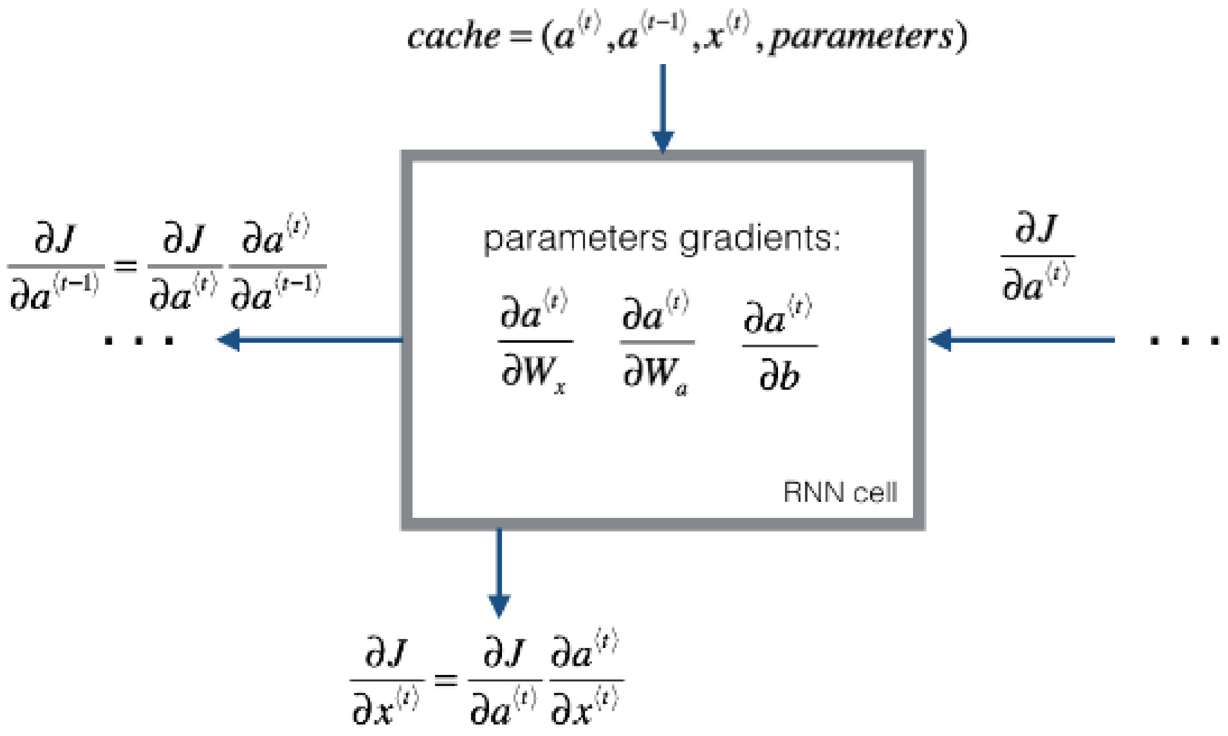
- To simplify the calculation we can combine (append) W_{ax}, W_{aa} to W_a
- combine $a^{(t-1)}$ and $x^{(t)}$

$$W_a = [W_{ax}, W_{aa}]$$

$$a^{\langle t \rangle} = g_1(W_{aa} a^{\langle t-1 \rangle} + W_{ax} x^{\langle t \rangle} + b_a) \quad \Longrightarrow \quad a^{\langle t \rangle} = g_1(W_a [a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_a)$$

$$\hat{y}^{\langle t \rangle} = g_2(W_{ya} a^{\langle t \rangle} + b_y) \qquad \hat{y}^{\langle t \rangle} = g_2(W_y a^{\langle t \rangle} + b_y)$$

Backpropagation



$$a^{\langle t \rangle} = \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)$$

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh(x)^2$$

$$\frac{\partial a^{\langle t \rangle}}{\partial W_{ax}} = (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2) x^{\langle t \rangle T}$$

$$\frac{\partial a^{\langle t \rangle}}{\partial W_{aa}} = (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2) a^{\langle t-1 \rangle T}$$

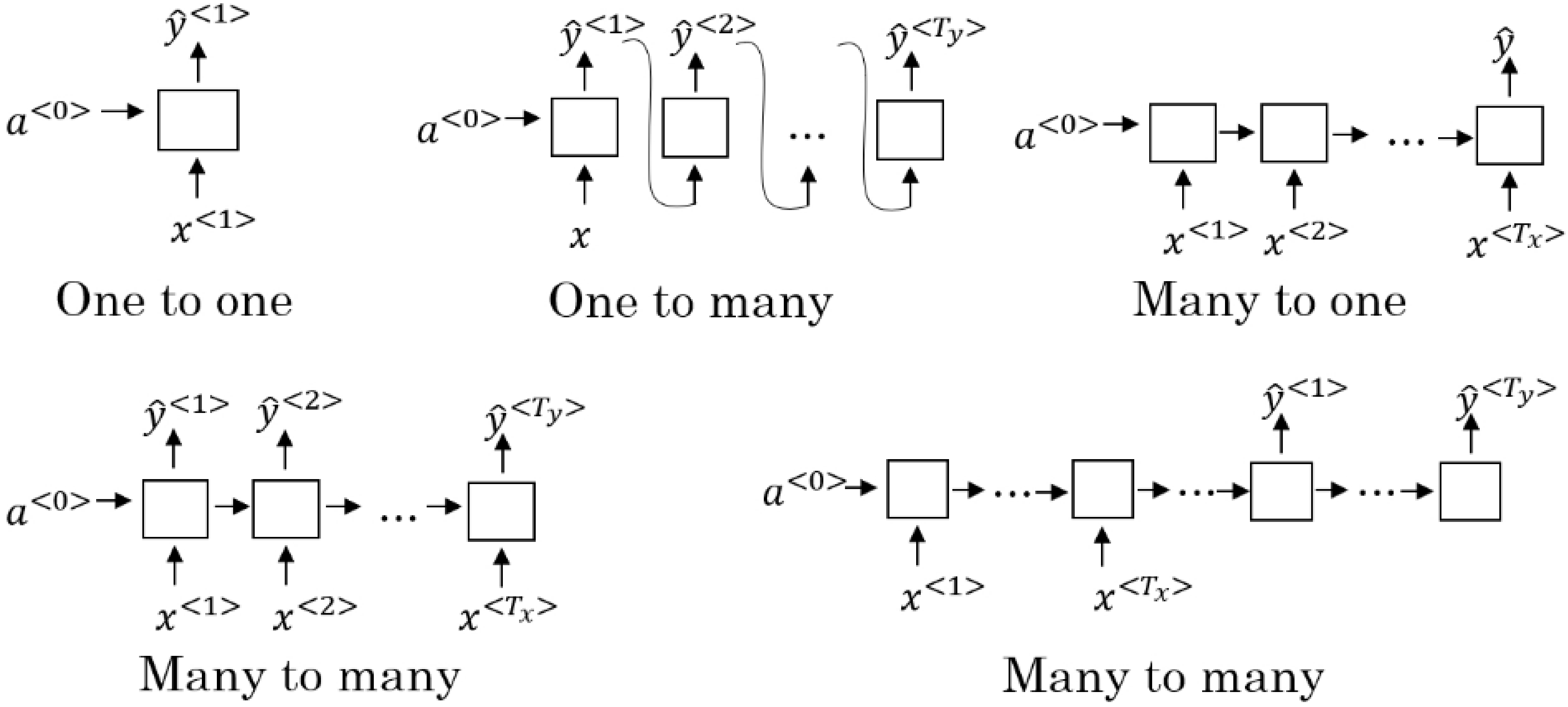
$$\frac{\partial a^{\langle t \rangle}}{\partial b} = \sum_{batch} (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

$$\frac{\partial a^{\langle t \rangle}}{\partial x^{\langle t \rangle}} = W_{ax}{}^T . (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

$$\frac{\partial a^{\langle t \rangle}}{\partial a^{\langle t-1 \rangle}} = W_{aa}{}^T . (1 - \tanh(W_{ax} x^{\langle t-1 \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

RNN Structuressssss

- we can have vary input output structure



RNN Issues

- The cat, which already ate a bunch of food, was full
- The cats, which already ate a bunch of food, were full
- The verb at the end is dependent on the noun at the beginning (subject-verb agreement)
- but RNN is not good at learning this long term dependent relationship

- As we go deeper, vanishing gradient

- To combat these two issues we have GRU and LSTM

Gated Recurrent Units (GRU)

- introduce additional variable call c in the recurrent unit which stands for memory cell
- We have the following new variables
- $c^{(t)}$, output activation value $a^{(t)}$
- $\bar{c}^{(t)}$, the next activation value $a^{(t)}$ candidate
- Γ_u , the Update Gate, it tells when to update memory cell c value

$$a^{\langle t \rangle} = g_1(W_a[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_a) \Longrightarrow$$

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

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

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

$$a^{\langle t \rangle} = c^{\langle t \rangle}$$

- Complete version also includes a Relevance gate Γ_r to represent the relevance between $c^{-(t)}$ and $c^{(t)}$

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

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

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

$$a^{\langle t \rangle} = c^{\langle t \rangle}$$

$$\Longrightarrow$$

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

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

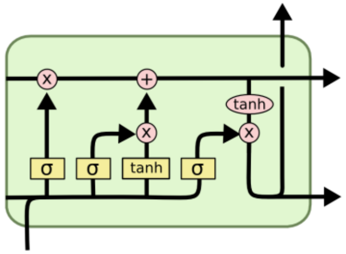
$$\Gamma_r = \sigma(W_r[c^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_r)$$

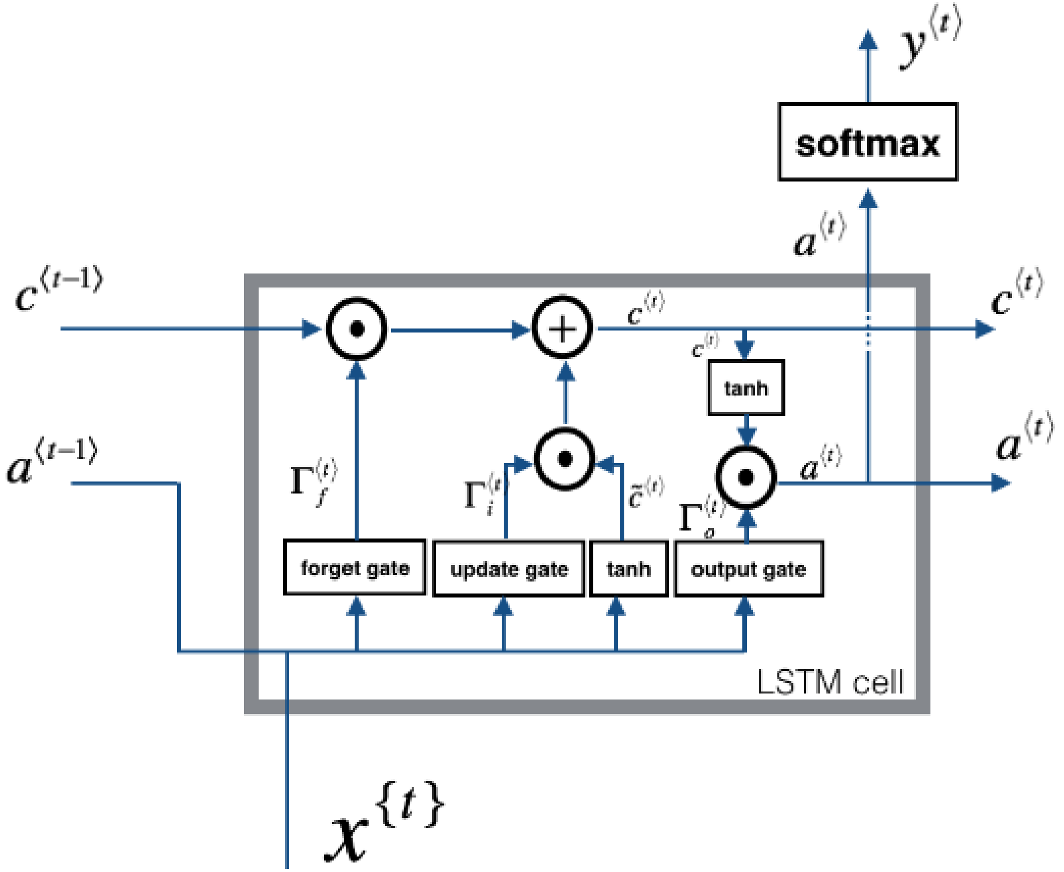
$$c^{\langle t \rangle} = \Gamma_u \times \tilde{c}^{\langle t \rangle} + (1 - \Gamma_u) \times c^{\langle t-1 \rangle}$$

$$a^{\langle t \rangle} = c^{\langle t \rangle}$$

Long Short Term Memory (LSTM)

- more flexible and powerful than GRU
- it involves Forget Gate Γ_f and output gate Γ_o





. . .

$$\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$$

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

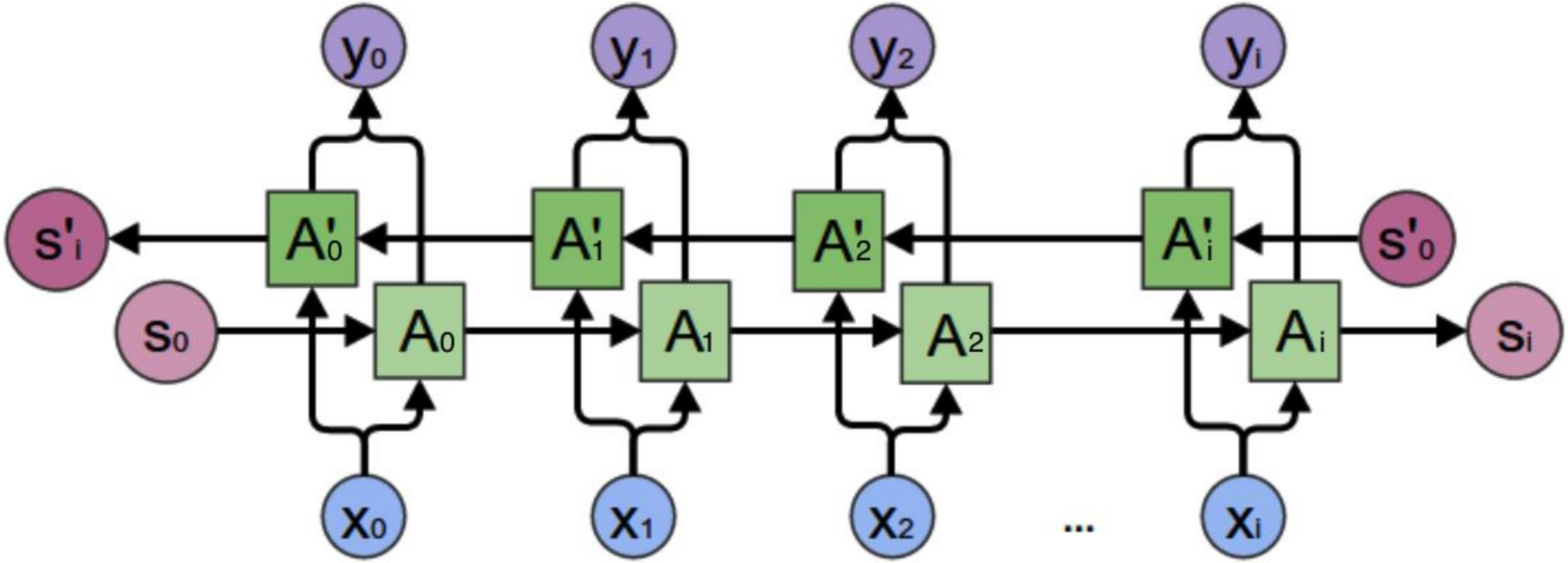
$$\tilde{c}^{\{t\}} = \tanh(W_C[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_C)$$

$$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$$

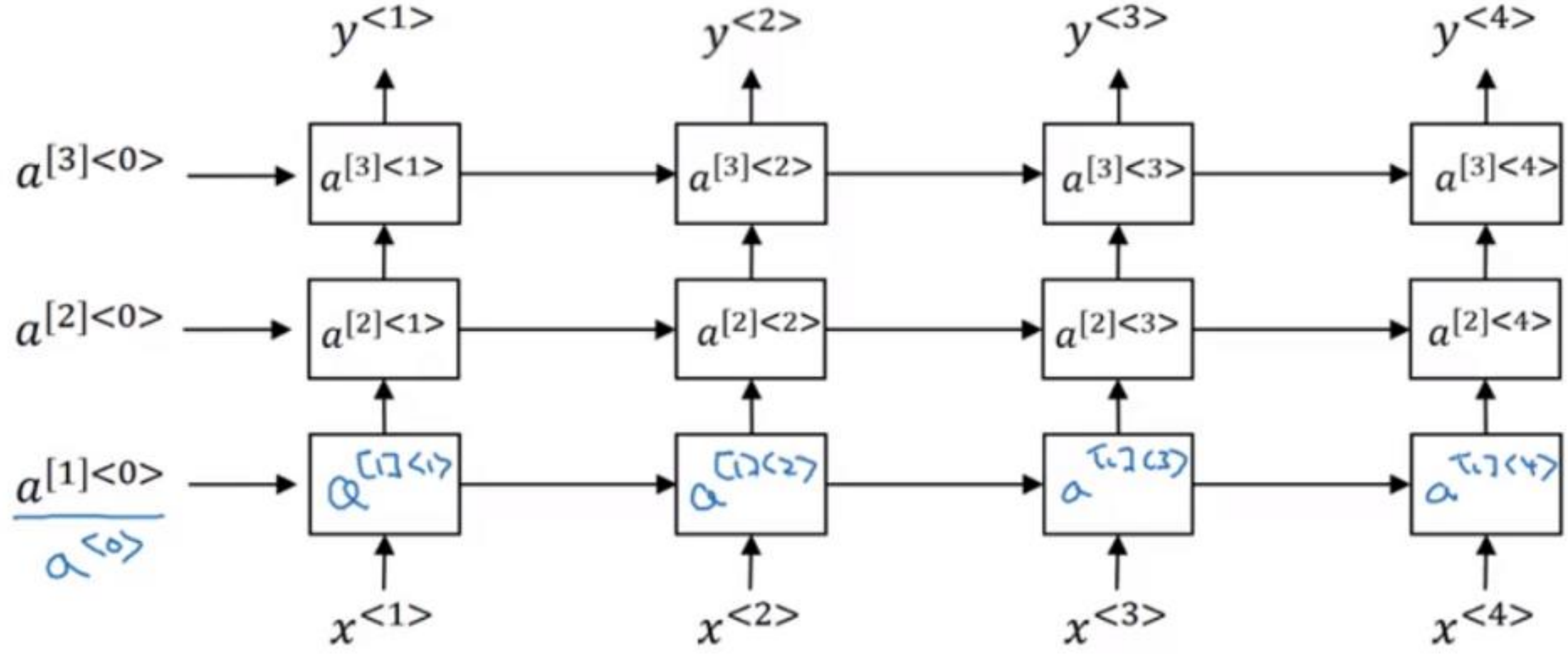
$$\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$$

$$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$$

Bidirectional RNN (BRNN)

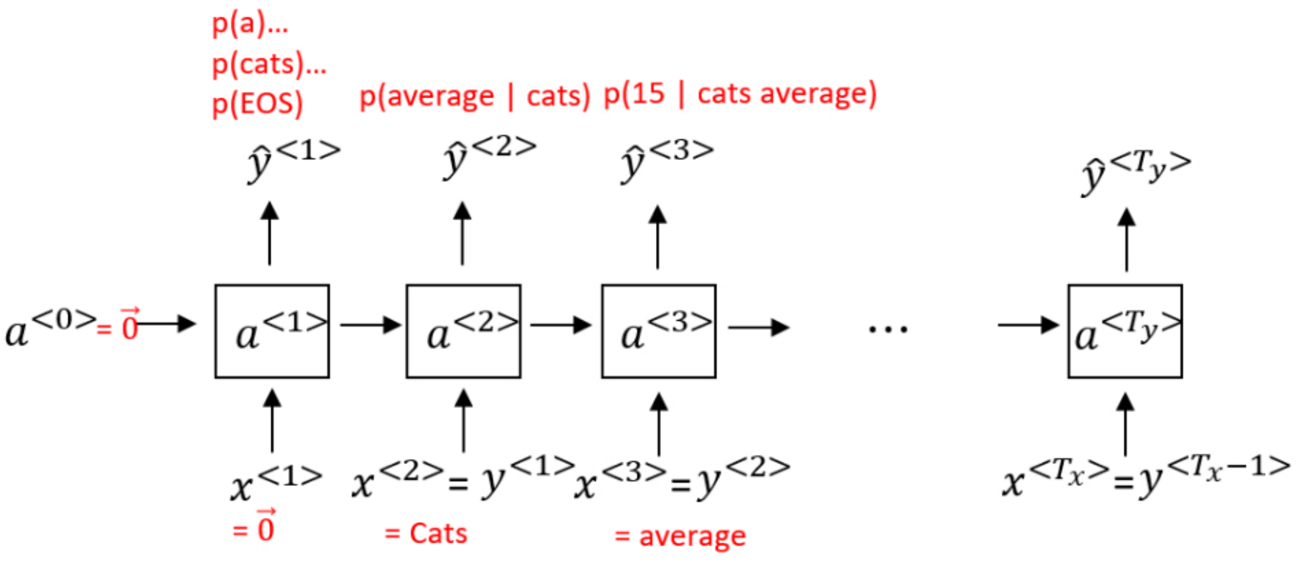


Deep RNN (DRNN)



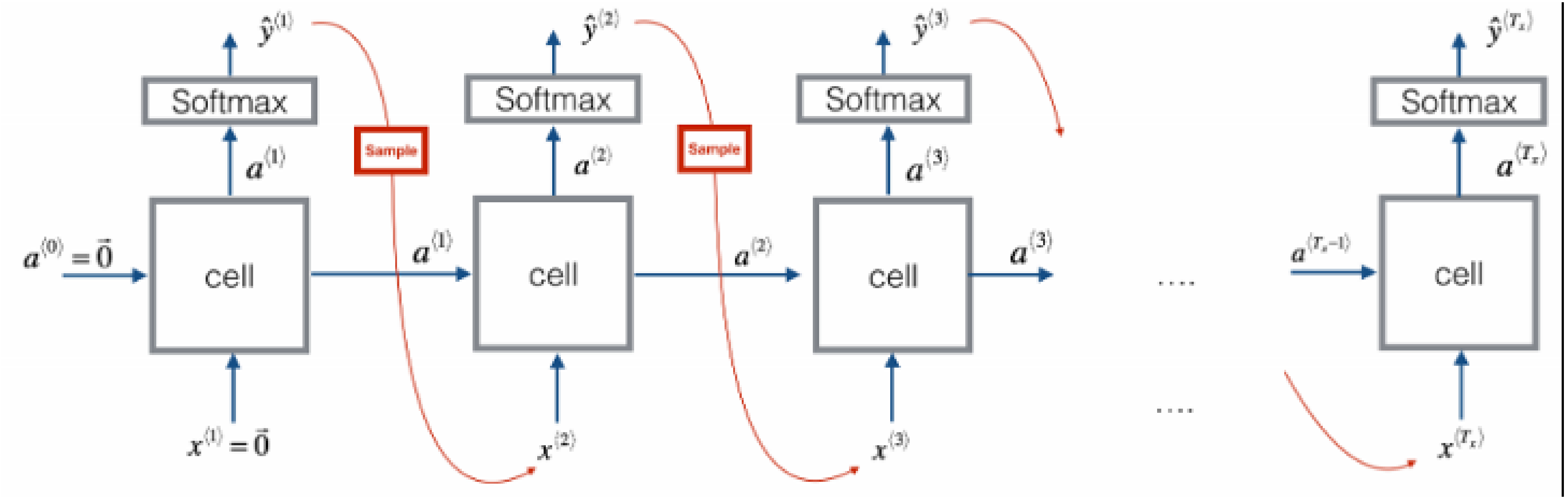
Simple RNN example

- Language Model is to describe fact base on language to model language using mathematic, it is able to predict the probability of element occurring in a sequence
- we use corpus (text compose of many sentences)
- the first step to build the model is tokenize (dictionaries words)
- we then encode each word into word vector and add EOS (End of Sentence) to signify the end of sentence



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

- after we trained a language model, we can use sampling to learn what we've learnt
- first input $a^{(0)}$ $x^{(1)}$ are 0, we get a distribution of softmax
- then we sample a random word
- we keep sampling till EOS
- then we get some sentence



Grand Plan (besides the lecture material)

- (✓) Neural Network Foundation
- (✓) Ensemble/Optimise your neural network
- (✓) Convolution Neural Network
- (✓) Principal Component Analysis / Big data
- (✓) Reinforcement Learning
- (✓) Recurrent Neural Network
- (☐) Natural Language Processing
- (☐) Generative Adversarial Network