THE UNIVERSITY OF TENNESSEE KNOXVILLE

AICIP RESEARCH

# ECE599/692 - Deep Learning

## Lecture 14 – Recurrent Neural Network (RNN)

Hairong Qi, Gonzalez Family Professor
Electrical Engineering and Computer Science
University of Tennessee, Knoxville
http://www.eecs.utk.edu/faculty/qi
Email: hqi@utk.edu



AICIP RESEARCH

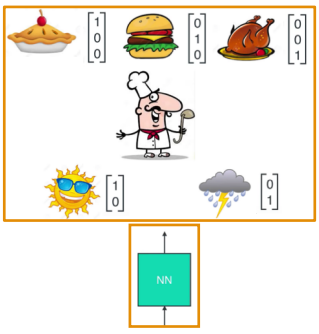
## Outline

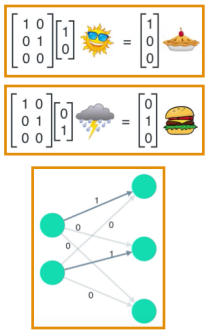
- Introduction
- LSTM vs. GRU
- Applications and Implementations
- References:
  - [1] Luis Serrano, A Friendly Introduction to Recurrent Neural Networks, https://www.youtube.com/watch?v=UNmqTiOnRfg, Aug. 2018
  - [2] Brandon Rohrer, Recurrent Neural Networks (RNN) and Long, Short-Term Memory (LSTM), https://www.youtube.com/watch?v=WCUNPb-5EYI, Jun. 2017
  - [3] Denny Britz, Recurrent Neural Networks Tutorial, http://www.wildml.com/2015/09/recurrent-neural-networks-tutorial-part-1-introduction-to-rnns/, Sept. 2015 (Implementation)
  - [4] Colah's blog, Understanding LSTM Networks, http://colah.github.io/posts/2015-08-Understanding-LSTMs/, Aug. 2015



AICIP RESEARCH

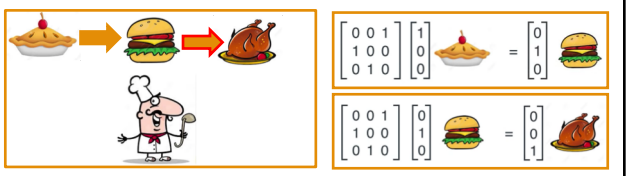
## A friendly introduction to NN [1]



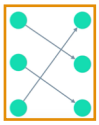
## A friendly introduction to RNN

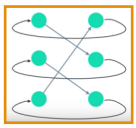
AICIP RESEARCH







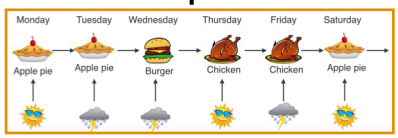



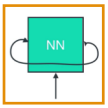


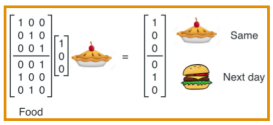


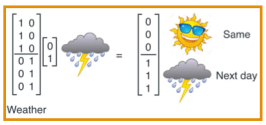
## A more complicated case

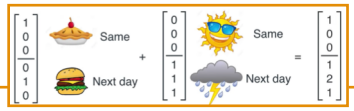
AICIP RESEARCH



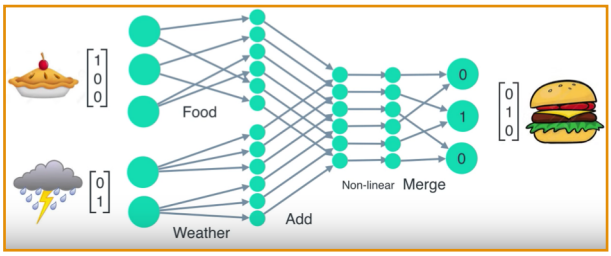



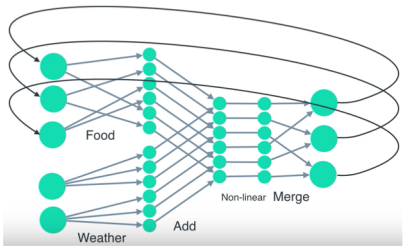



## A more complicated case (cont'd)

AICIP RESEARCH



## A more complicated case (cont'd)






## The children's book example from Brandon Rohrer [2]

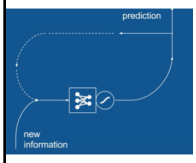
Doug saw Jane.
Jane saw Spot.
Spot saw Doug.

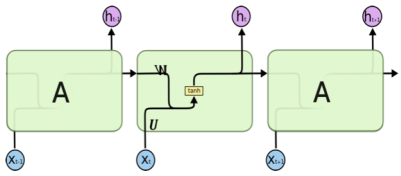
Dictionary = {Doug, Jane, Spot, saw, .}

Potential mistakes:
Doug saw Doug.
Doug saw Jane saw Spot saw Doug…
Doug.



## The standard RNN module



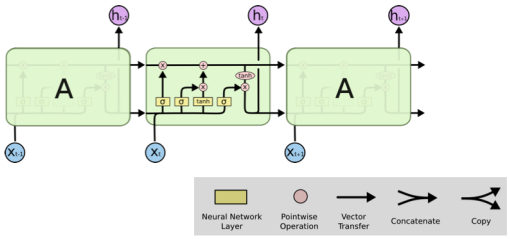

$$h_t = f(Ux_t + Wh_{t-1} + b)$$

**Gradient vanishing issue: by the end of the RNN, the data from the first timestep has very little impact on the output of the RNN. An example of word prediction, "I grew up in France… I speak fluent *French*."**

## The long-short term memory (LSTM) module

AICIP RESEARCH

**LSTMs are explicitly designed to avoid the long-term dependency problem.**

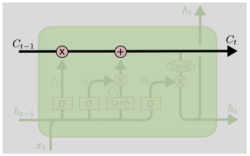




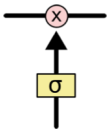

## Two keys of LSTM

AICIP RESEARCH

**"Cell state" which works like a conveyor belt runs straight down the entire chain, easy for information to flow along without changes.**
**"Gates" which control or decide what kind of information could go or throw away from the cell state.**



**Cell state**



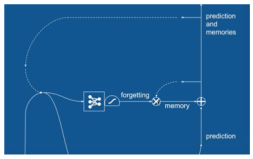

**Gate**

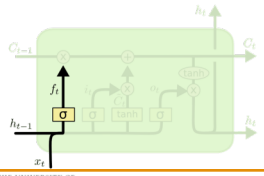


## LSTM – forget gate

AICIP RESEARCH

Take the example of a language model trying to predict the next word based on all the previous ones. In such a problem, the cell state might include the gender of the present subject, so that the correct pronouns can be used. When we see a new subject, we want to forget the gender of the old subject.



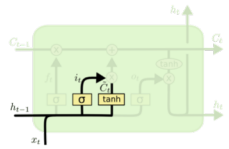




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$

## LSTM – input gate

AICIP RESEARCH

It decides what new information we're going to store in the cell state. It has two parts. First, a sigmoid layer called the "input gate layer" decides which values we'll update. Next, a tanh layer creates a vector of new candidate values, $\tilde{C}_t$ , that could be added to the state. In the next step, we'll combine these two to create an update to the state.



$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$

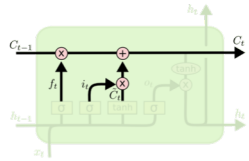
THE UNIVERSITY OF TENNESSEE KNOXVILLE

13

## LSTM – cell state update

AICIP RESEARCH

It actually drops the information about the old subject's gender and add the new information, as we decided in the previous steps.



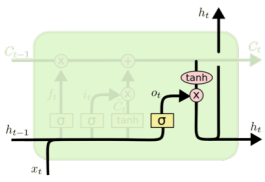
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$

THE UNIVERSITY OF TENNESSEE KNOXVILLE

14

## LSTM – output gate

AICIP RESEARCH



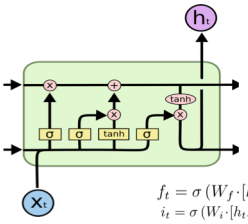
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

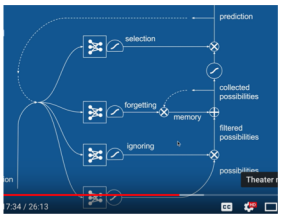
THE UNIVERSITY OF TENNESSEE KNOXVILLE

15

## LSTM - revisit

AICIP RESEARCH



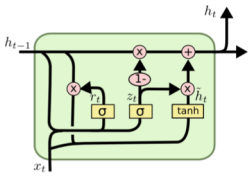




$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$



## The gated recurrent units (GRUs) module

AICIP RESEARCH



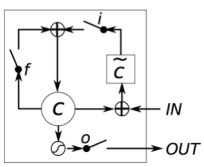

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

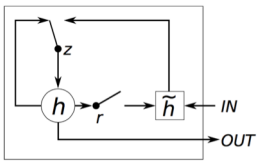
**Similar with LSTM but with only two gates and less parameters.**
**The "update gate" determines how much of previous memory to be kept.**
**The "reset gate" determines how to combine the new input with the previous memory.**

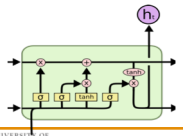


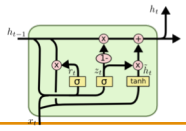
## Comparison of the gating mechanism

AICIP RESEARCH

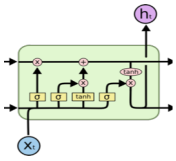


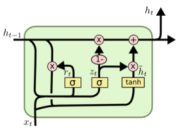





## LSTM vs. GRU





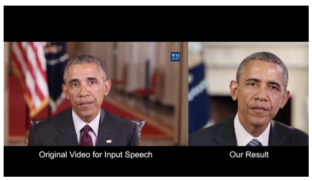
$$f_t = \sigma\left(W_f \cdot [h_{t-1}, x_t] \; + \; b_f\right)$$
$$i_t = \sigma\left(W_i \cdot [h_{t-1}, x_t] \; + \; b_i\right)$$
$$\tilde{C}_t = \tanh(W_C \cdot [h_{t-1}, x_t] \; + \; b_C)$$
$$C_t = f_t * C_{t-1} + i_t * \tilde{C}_t$$
$$o_t = \sigma\left(W_o \; [h_{t-1}, x_t] \; + \; b_o\right)$$
$$h_t = o_t * \tanh\left(C_t\right)$$

$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$
$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$
$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$



## Application example: The talking face

Goal: Given an arbitrary audio clip and a face image, automatically generate realistic and smooth face video with accurate lip sync.



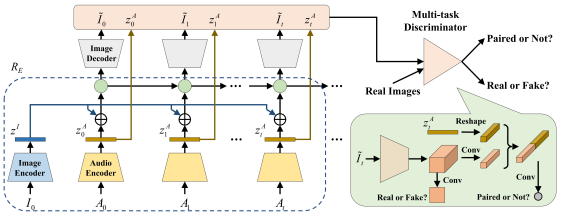

[Suwajanakorn et al., 2017]

Application: Face animation, entertainment, video bandwidth reduction, etc.



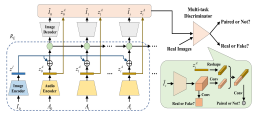
## The proposed framework

The proposed method: conditional video generation



## The proposed framework

AICIP RESEARCH



$$\mathcal{L}_G = \underbrace{\mathbb{E}_{I\sim\mathbb{P}_r}\left[\log(1-D(I))\right]}_{\text{Unconditioned}} + \underbrace{\mathbb{E}_{\tilde{I}\sim\mathbb{P}_g}\left[\log(1-D(\tilde{I},z_I))\right]}_{\text{Conditioned}} \tag{2}$$

$$\mathcal{L}_D = \underbrace{-\mathbb{E}_{I\sim\mathbb{P}_r}\left[\log D(I)\right] - \mathbb{E}_{\tilde{I}\sim\mathbb{P}_g}\left[\log(1-D(\tilde{I}))\right]}_{\text{Unconditioned}} - \tag{3}$$

$$\underbrace{\mathbb{E}_{I\sim\mathbb{P}_r}\left[\log(1-D(I,z_I))\right] - \mathbb{E}_{I\sim\mathbb{P}_r}\left[\log(1-D(I,\tilde{z}_I))\right] - \mathbb{E}_{\tilde{I}\sim\mathbb{P}_g}\left[\log(1-D(\tilde{I},z_I))\right]}_{\text{conditioned}}$$

$$\mathcal{L}_{per} = \sum_{i=1}^{C}\left\|\phi_i(I) - \phi_i(\tilde{I})\right\|, \tag{4}$$

The final objective function is

$$\min_{G,E_A,E_I,R_E}\max_{D}\mathcal{F}(G,E_A,E_I,R_E,D), \tag{5}$$

where $\mathcal{F}(G,E_A,E_I,R_E,D) = L_{rec} + \gamma L_{per} + \lambda L_G + \lambda L_D$.

THE UNIVERSITY OF TENNESSEE KNOXVILLE