RECURRENT NEURAL NETWORKS

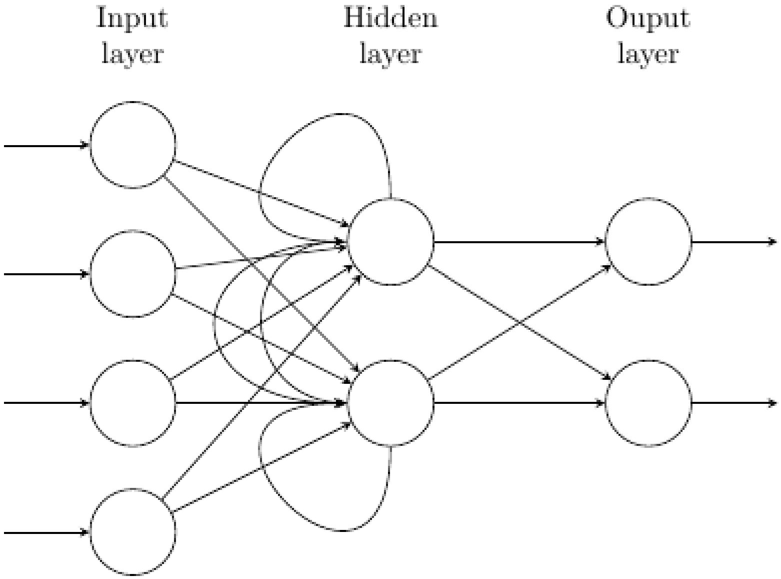
Shingchern D. You

References

- RNN and LSTM, by E. Lobacheva and D. Vetrov
- http://proceedings.mlr.press/v28/pascanu13.pdf

What is a recurrent neural network

- The outputs of the network depend not only on present inputs, but also PAST inputs (picture source: https://tex.stackexchange.com/questions/364413/drawing-an-unfolded-recurrent-neural-network)



What is a recurrent neural network

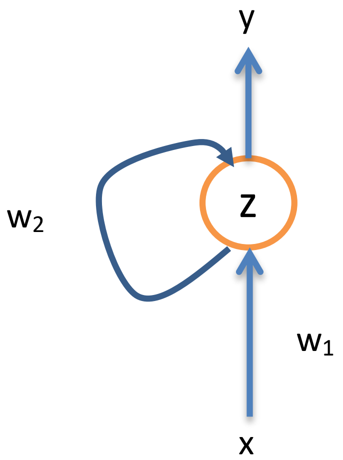
- This type of neural networks can capture temporal dependences of input sequence
- Examples: speech signal, text, handwritten characters, etc.

Training recurrent neural network

- Traditional recurrent neural networks suffer from a problem called vanish of gradient (although explosion of gradient also possible) during training
- Due to this problem, this type of network cannot learn temporal correlations with a long (time) lag

Training recurrent neural network

- To illustrate this problem, we use a simplified version shown below



Training recurrent neural network

- The network has the following equations

$$z_{(t)} = x_{(t)}w_1 + y_{(t-1)}w_2$$

$$y_{(t)} = f\left(z_{(t)}\right) = \frac{1}{1 + \exp(-z_{(t)})}$$

- Note the followings
  - t is a discrete variable, i.e., $t = 0{,}1{,}2, \ldots$
  - We usually let $y_{(-1)} = 0$ as the initial condition
  - Weights are not changed over time

Training recurrent neural network

- □ We define the cost function as the MSE summing over time (also over all outputs if more than one output node is present)

$$\varepsilon_{(t)} = \left(y_{(t)} - d_{(t)}\right)$$

$$J = \sum_t J_t = \frac{1}{2}\sum_t \varepsilon_{(t)}^2 = \frac{1}{2}\sum \left(y_{(t)} - d_{(t)}\right)^2$$

Training recurrent neural network

- □ We want to find the weights for a given time, say t

$$\frac{\partial J_t}{\partial w_2} = \frac{\partial J_t}{\partial y_{(t)}} \frac{\partial y_{(t)}}{\partial w_2}$$

- □ We know $\frac{\partial y_{(t)}}{\partial w_2} = \frac{\partial y_{(t)}}{\partial z_{(t)}} \frac{\partial z_{(t)}}{\partial w_2} = f'(z_{(t)}) \frac{\partial z_{(t)}}{\partial w_2}$

$$= y_{(t)}\left(1 - y_{(t)}\right)\left(y_{(t-1)} + w_2 \frac{\partial y_{(t-1)}}{\partial w_2}\right)$$

Backprop through time

- If we do one more step, we have

$$\frac{dy_{(t-1)}}{dw_2} = f'(z_{(t-1)})\left(y_{(t-2)} + w_2\frac{dy_{(t-2)}}{dw_2}\right)$$

- Therefore,

$$\begin{aligned}\frac{dJ_t}{dw_2} = \varepsilon_{(t)}\Bigg\{f'(z_{(t)})\Bigg[y_{(t-1)} \\ + w_2\left(f'(z_{(t-1)})\left(y_{(t-2)} + w_2\frac{dy_{(t-2)}}{dw_2}\right)\right)\Bigg]\Bigg\}\end{aligned}$$

Backprop through time

- □ If we simplify the math a bit, we have

$$\frac{dJ}{dw_2}=\varepsilon_{(t)}f'\left(z_{(t)}\right)y_{(t-1)}$$
$$+w_2\varepsilon_{(t)}f'\left(z_{(t)}\right)f'\left(z_{(t)}\right)y_{(t-2)}$$
$$+w_2^2\varepsilon_{(t)}f'\left(z_{(t)}\right)f'\left(z_{(t-1)}\right)f'\left(z_{(t-2)}\right)y_{(t-3)}+\cdots$$

- □ Therefore, the term $y_{(t-2)}$ is multiplied with w_2, the term $y_{(t-3)}$ with w_2^2, and so on
- □ In general, $y_{(t-m)}$ is multiplied with $w_2^{(m-1)}$

Backprop through time

- □ Previously, we have

$$\frac{\partial y_{(t)}}{\partial w_2} = \frac{\partial y_{(t)}}{\partial z_{(t)}} \frac{\partial z_{(t)}}{\partial w_2} = f'(z_{(t)}) \frac{\partial z_{(t)}}{\partial w_2}$$

$$= y_{(t)}(1 - y_{(t)}) \left(y_{(t-1)} + w_2 \frac{\partial y_{(t-1)}}{\partial w_2} \right)$$

- □ Rearrange the terms, we have

$$\frac{\partial y_{(t)}}{\partial w_2} = f'(z_{(t)}) y_{(t-1)} + f'(z_{(t)}) w_2 \frac{\partial y_{(t-1)}}{\partial w_2}$$

Backprop through time

- It is easy to show that

$$f'\left(z_{(t)}\right)w_2\frac{\partial y_{(t-1)}}{\partial w_2}=\frac{\partial y_{(t)}}{\partial y_{(t-1)}}\frac{\partial y_{(t-1)}}{\partial w_2}$$

- If we define $\frac{\partial^+ y_{(t)}}{\partial w_2}=f'\left(z_{(t)}\right)y_{(t-1)}$

We can simplify the expression into a sum-of-product form

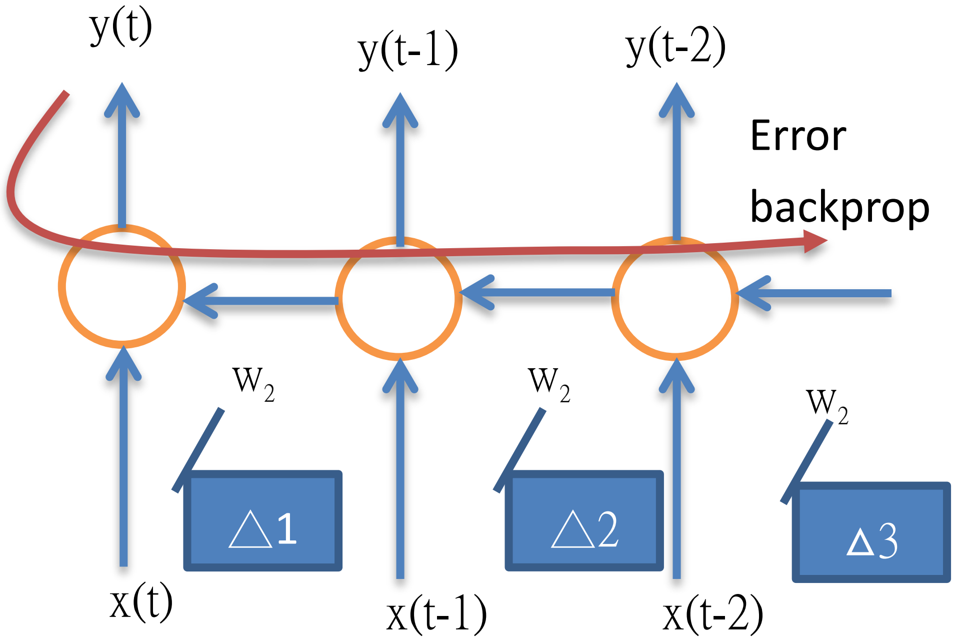
Backprop through time

- □ Therefore, the partial derivatives can be expressed as follow (used in http://proceedings.mlr.press/v28/pascanu13.pdf)

$$\frac{\partial y_{(t)}}{\partial w_2} = \frac{\partial^+ y_{(t)}}{\partial w_2} + \frac{\partial y_{(t)}}{\partial y_{(t-1)}} \frac{\partial^+ y_{(t-1)}}{\partial w_2} + \frac{\partial y_{(t)}}{\partial y_{(t-1)}} \frac{\partial y_{(t-1)}}{\partial y_{(t-2)}} \frac{\partial^+ y_{(t-2)}}{\partial w_2} + \cdots$$

Backprop through time

- One way to interpret the equation in previous slide is the unfolded model



Backprop through time

- Updating of w_2 is $\Delta 1 + \Delta 2 + \Delta 3 + \cdots$

$$\Delta 1 = \frac{\partial^+ y_{(t)}}{\partial w_2} \varepsilon_{(t)}$$ (1st term)

$$\Delta 2 = \frac{\partial y_{(t)}}{\partial y_{(t-1)}} \frac{\partial^+ y_{(t-1)}}{\partial w_2} \varepsilon_{(t)}$$ (2nd term)

$$\Delta 3 = \frac{\partial y_{(t)}}{\partial y_{(t-1)}} \frac{\partial y_{(t-1)}}{\partial y_{(t-2)}} \frac{\partial^+ y_{(t-2)}}{\partial w_2} \varepsilon_{(t)}$$ (3rd term)

Etc.

Backprop through time

- □ The approach we are doing is like we unfold the network over time into a very large network
- □ Usually, we need to restrict the number of unfolding m in order to make training possible
- □ Because $y_{(t-m)}$ is multiplied with $w_2^{(m-1)}$, if w_2 is less than 1, $w_2^{(m-1)} \ll 1$ is very likely (called vanish of gradient)
- □ Therefore, it is difficult to train recurrent neural networks (cf. http://proceedings.mlr.press/v28/pascanu13.pdf)

Basic idea of LSTM

- If we want to solve this problem, we have to restrict the following conditions
  - $w_2 = 1$
  - $f(z)$ must be “like” a linear function
- In addition, to make sure the training can be accomplished with reasonable complexity, **truncation** is used (i.e., no backprop through time)

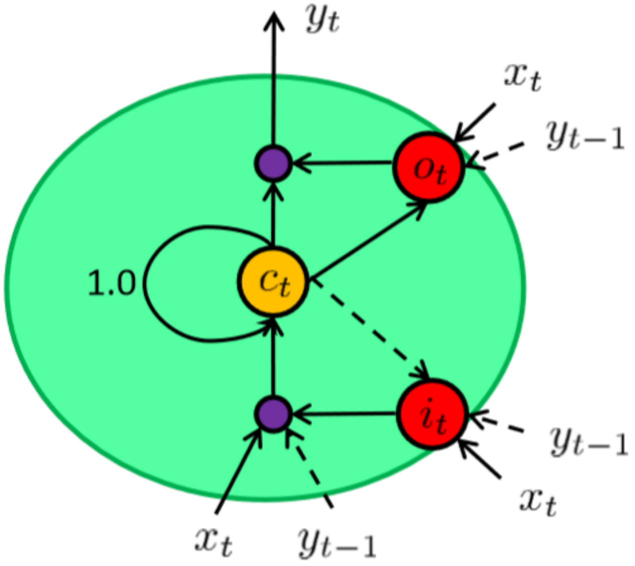
Basic idea of LSTM

- A node configured in this way is called a (memory) cell
- Learning is accomplished elsewhere, but not within the cell
- The proposed model is called LSTM, standing for long short-term memory
- Meaning that their model can have long-term or short-term memory

LSTM overview

- □ Original LSTM (From RNN and LSTM, by E. Lobacheva and D. Vetrov)

Version 0



i_t, o_t - input and output gates from 0 to 1

c_t - memory

x_t - input, y_t - output

$$i_t = \sigma(w_{ix}x_t + w_{ic}c_{t-1} + w_{iy}y_{t-1} + b_i)$$

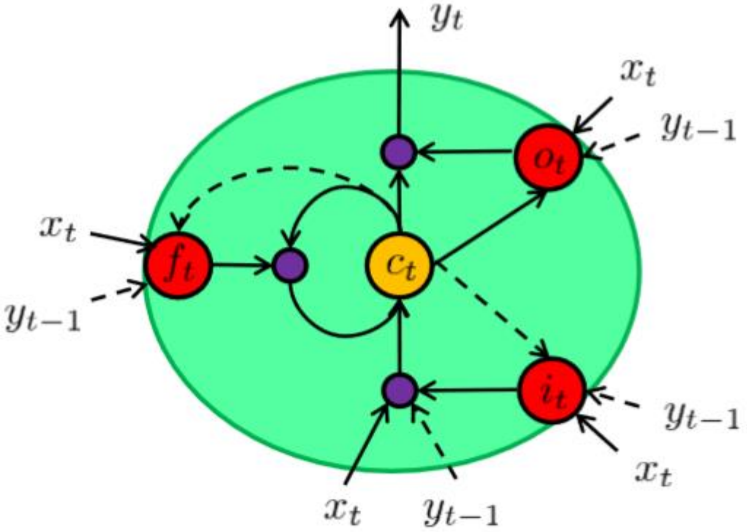
$$o_t = \sigma(w_{ox}x_t + w_{oc}c_t + w_{oy}y_{t-1} + b_o)$$

$$c_t = c_{t-1} + i_t \cdot tanh(w_{cx}x_t + w_{cy}y_{t-1}) \qquad y_t = o_t \cdot tanh(c_t)$$

LSTM overview

- □ **Modified version** (From RNN and LSTM, by E. Lobacheva and D. Vetrov)

Version 1



i_t, f_t, o_t - input, forget and output gates from 0 to 1

c_t - memory

x_t - input, y_t - output

$$i_t = \sigma(w_{ix}x_t + w_{ic}c_{t-1} + w_{iy}y_{t-1} + b_i)$$

$$f_t = \sigma(w_{fx}x_t + w_{fc}c_{t-1} + w_{fy}y_{t-1} + b_f)$$

$$o_t = \sigma(w_{ox}x_t + w_{oc}c_t + w_{oy}y_{t-1} + b_o)$$

$$c_t = f_t c_{t-1} + i_t \cdot tanh(w_{cx}x_t + w_{cy}y_{t-1}) \qquad y_t = o_t \cdot tanh(c_t)$$

LSTM overview

- □ Version 1 adds a “forget” gate so that “reset” memory cell is possible
- □ Because we have several gates, their weights are also trained by using backprop
- □ In the previous slides, the σ function stands for sigmoid function
- □ As the details derivation of back prop for LSTM is tedious (actually not extremely difficult), we omit the details, cf. the original LSTM paper at https://www.researchgate.net/publication/13853244_Long_Short-term_Memory

LSTM overview

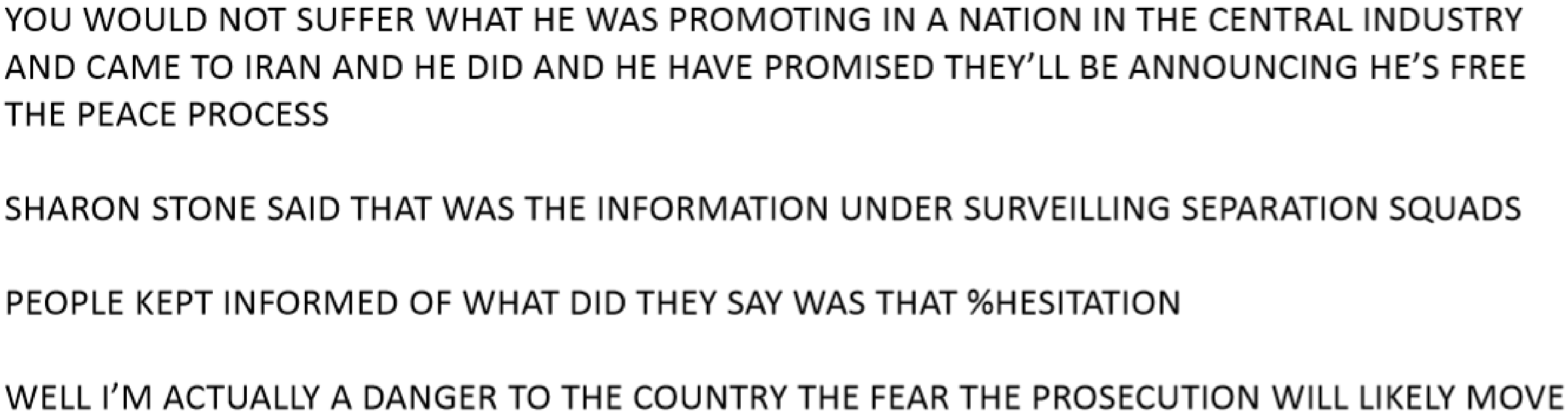
- It is not so easy to imagine the global picture of the LSTM network
- Remember all inputs and outputs connect to all gates with weights
- Some LSTM variations share some of the weights across different LSTM modules (one module is represented by a green circle given previously)

Using LSTM

- We use automatic music composition as an example
- Training
  - Input to the network is musical note k
  - Desired output is note (k+1)
- Generation
  - Randomly give one note to LSTM
  - Use the produced note k as input at time (k+1)

LSTM applications

- Here are some more applications also from RNN and LSTM, by E. Lobacheva and D. Vetrov
- Text generation using word-wise case (i.e., one input for a given time is one word, not one alphabet)



YOU WOULD NOT SUFFER WHAT HE WAS PROMOTING IN A NATION IN THE CENTRAL INDUSTRY AND CAME TO IRAN AND HE DID AND HE HAVE PROMISED THEY'LL BE ANNOUNCING HE'S FREE THE PEACE PROCESS

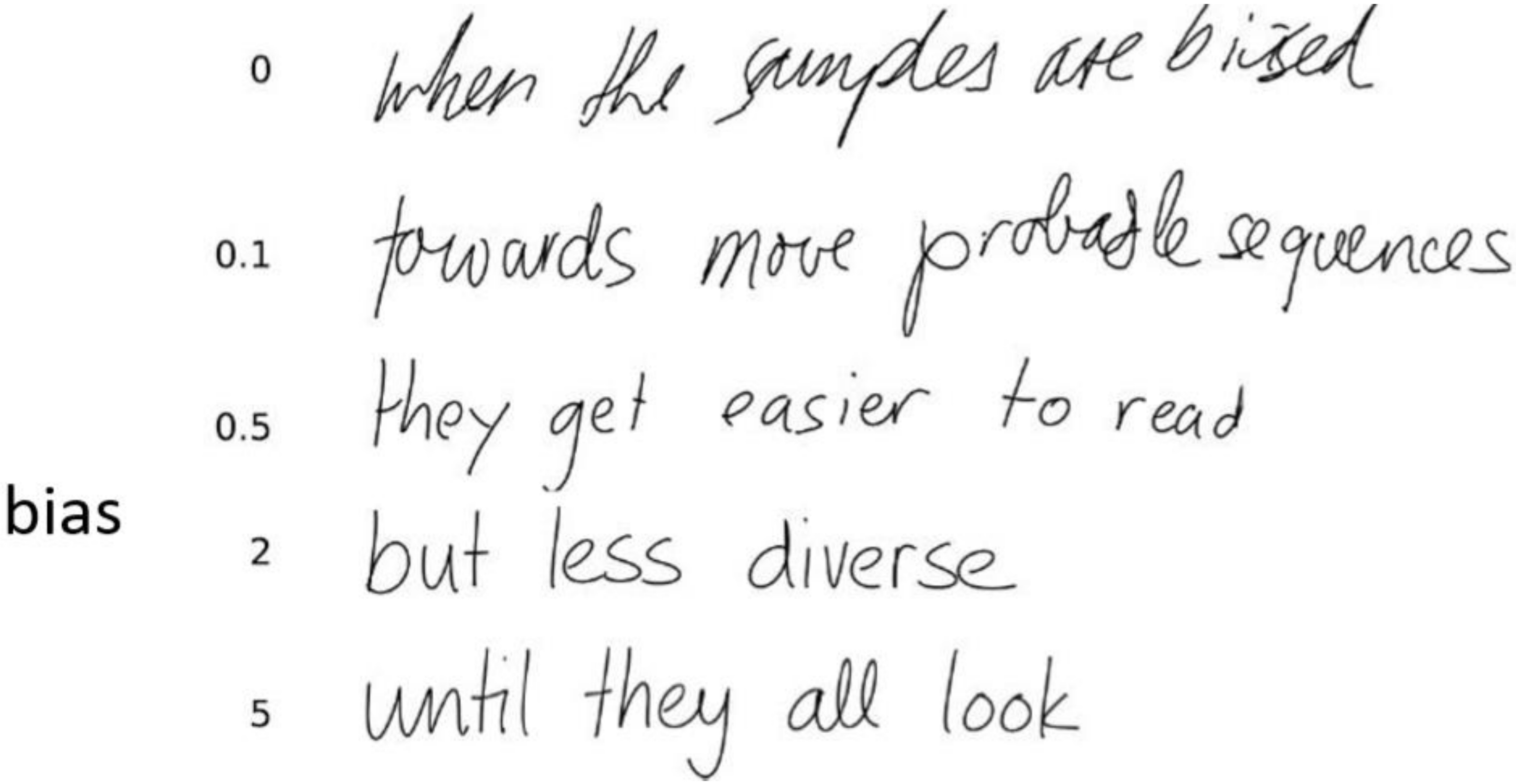
SHARON STONE SAID THAT WAS THE INFORMATION UNDER SURVEILLING SEPARATION SQUADS

PEOPLE KEPT INFORMED OF WHAT DID THEY SAY WAS THAT %HESITATION

WELL I'M ACTUALLY A DANGER TO THE COUNTRY THE FEAR THE PROSECUTION WILL LIKELY MOVE

LSTM applications

- Handwriting generation


bias
0 when the samples are biased
0.1 towards more probable sequences
0.5 they get easier to read
2 but less diverse
5 until they all look