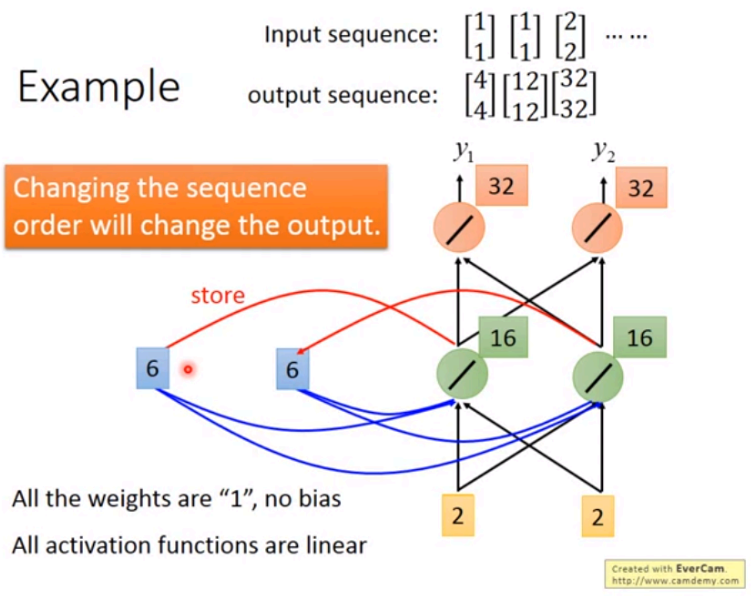
RNN

Recurrent Neural Network, sequence (e.g.text, speech, time series...) as input
RNN can "remember" some part of sequences, thus understanding the context.

How RNN remember and how sequence order effects the output


Example
Input sequence: $\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}1\\1\end{bmatrix}\begin{bmatrix}2\\2\end{bmatrix}$ ……
output sequence: $\begin{bmatrix}4\\4\end{bmatrix}\begin{bmatrix}12\\12\end{bmatrix}\begin{bmatrix}32\\32\end{bmatrix}$
Changing the sequence order will change the output.
store
y_1
y_2
32
32
16
16
6
6
2
2
All the weights are "1", no bias
All activation functions are linear
Created with EverCam. http://www.camdemy.com

RNN retains and updates an internal memory across time steps to remember past information.

- At each time step, the hidden layers learn from the input and also **the internal memory**
- As the internal memory updates, the output of hidden layers varies, even when it takes same input as before

Elman Network & Jordan Network

- Elman Network is able to learn **the influence of incorporating the most recent input into the sequence**.
  - Compared to the vanilla RNN which learn from 2 sources, the hidden layers of Elman Network draw information from 3 sources. The additional source is **the previous internal memory**
  - By comparing the current internal memory with the previous internal memory, the Elman Network can learn the dynamic trends of introducing a new input into the sequence
- Jordan Network
  - Also learn from 3 sources. The additional source is **the previous final output of the network**

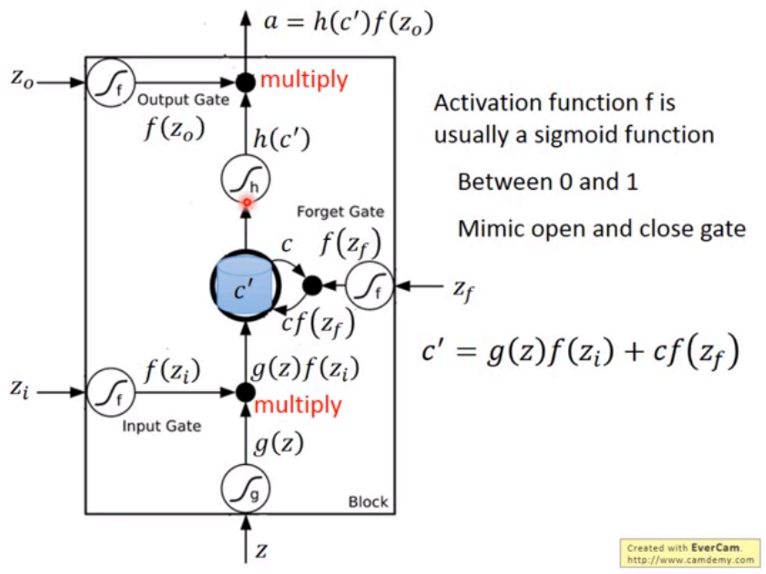
Bidirectional RNN

combine the hidden layers of two RNNs of opposite directions, thus understand the context in two directions

Long Short-term Memory (LSTM)

Conventional internal memory is *updated at every time step* (i.e. short-term memory), which makes it unable to remember important details.
In contrast, LSTM determines the importance of information and selectively retains only the crucial details, enabling it to maintain significant context over long sequences.

3 gates on the memory:

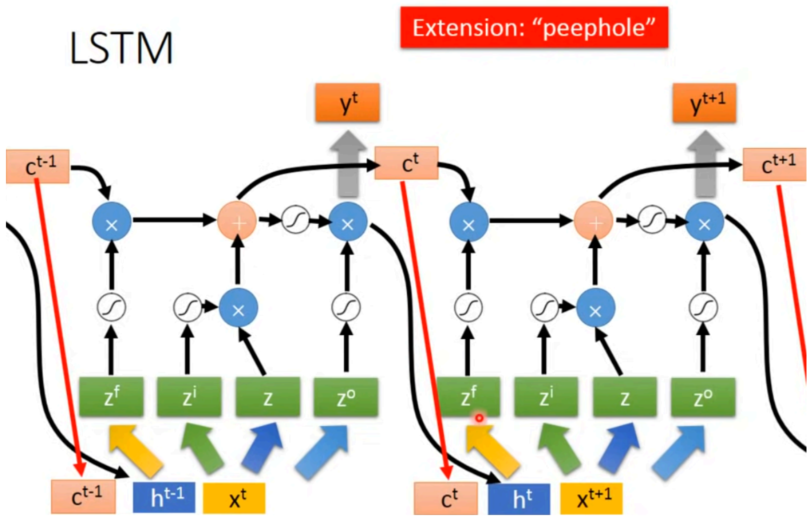


- Input Gate
  - z_i determines how much of the new input is allowed to affect the memory
- Forget Gate
  - z_f determines how much of the previous memory is retained
- Output Gate
  - z_o determines how much of the memory content is exposed to the output layer

LSTM's three gates, along with its cell state, result in the parameter count being roughly 4 times that of a standard RNN unit.
In addition, the functioning of LSTM also considers

- h_t, the final output of the previous LSTM

- c_t, the content in the previous LSTM



Gated Recurrent Unit (GRU) optimized LSTM by only using 2 gates, making it easier for training.

RNN's training issue

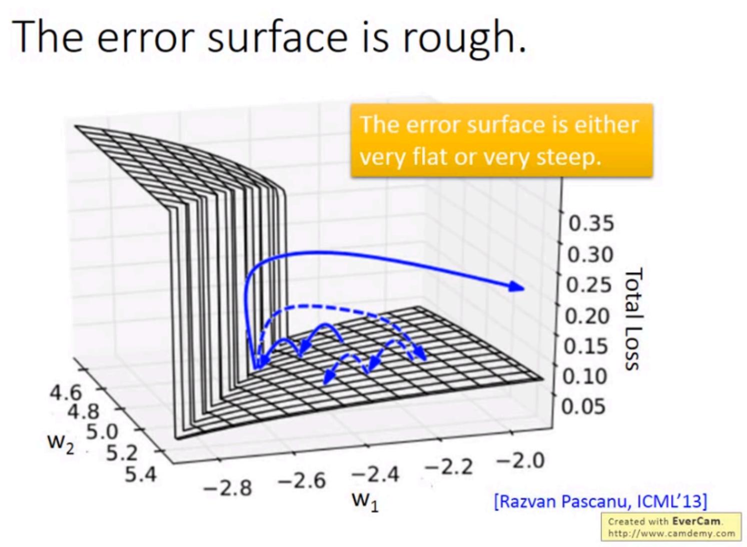
In normal NNs, the BP process updates the parameters layer by layer.
However, RNN's neurons are dependent on the previous neurons. So all the parameters before the time step have to be updated at the same time, which is called **Backpropagation through time (BPTT)**

Gradient Vanishing issue bacomes more significant in deep RNN due to BPTT, and this issue is the reason behind short-term memory.
LSTM can address this issue by controlling the Forget Gate.

Grandient Explode issue is also frequent. e.g. $w = 1, y^{1000} = 1; w = 1.01; y^{1000} \approx 20000$

- Gradient Clipping
- set a small learning rate

Gradient Vanishing performs as the falt areas in the error surface;
Gradient Explode performs as the steep areas.



The error surface is rough.

The error surface is either very flat or very steep.

RNN's applications

- Many to one
  - Key term extraction
- Many to many, output is shorter than input
  - Speech Recognition. Many acoustic features to many characters
- Many to many, no limitation
  - Translation. Many english words to many chinese characters
- Beyond Sequence
  - Syntactic parsing

Sequence-to-sequence Auto-encoder, RNN can be the encoder and decoder

- Text
- Speech
  - e.g. Chat-bot. Encoder:"How are you" --> Decoder:"I am fine"

Attention-based Model, Neural Turing Machine

- reading comprehension
- visual question answering
- speech question answering