RNN

Recurrent Neural Network

Handle Sequential data
(sentence, Stocks price, Time Series)

Memory
(remember the what happend before)

input at time t

$X(t) \xrightarrow{W}$ RNN Cell $\rightarrow h(t) \xrightarrow{W} y(t)$

$h(t-1) \xrightarrow{W}$ RNN Cell

(updated memory)
current hidden state

output at time t

$h(t-1)$
previous hidden state
(memory from last time step)

$$h_t = \tanh\left(W_x x_t + W_h h_{t-1} + b\right)$$

$$y_t = \left(W_{hy} h_t + b_y\right)$$ Scalar

W_x = weight matrix for input

W_h = weight matrix of Previous hidden

W_{hy} = weight matrix of current hidden

b, b_y = bias

tanh = activation function to introduce non-linearity

Sample RNN with Numerical input

Assume

Input size = 1

Hidden size = 2

Initial hidden state = $h_0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ 2×1

Input $x_1 = 1$

2×2

Random Assumption

$$\overset{2\times1}{W_x} = \begin{bmatrix}0.5\\0.1\end{bmatrix} \qquad b = \begin{bmatrix}0\\0\end{bmatrix}$$

$$\overset{2\times2}{W_h} = \begin{bmatrix}0.4 & 0.2\\0.3 & 0.7\end{bmatrix}$$

$$\overset{1\times2}{W_{hy}} = \overset{1\times2}{\begin{bmatrix}1 & -1\end{bmatrix}} \qquad b_y = 0$$

Step 1 - Calculations for hidden State

$$h_1 = \tanh\left(W_x x_1 + W_h h_0 + b\right)$$

$$W_x x_1 = \begin{bmatrix}0.5\\0.1\end{bmatrix} \times 1 = \begin{bmatrix}0.5\\0.1\end{bmatrix}$$

$$W_h h_0 = \underbrace{\begin{bmatrix}0.4 & 0.2\\0.3 & 0.7\end{bmatrix}}_{2\times2} \begin{bmatrix}0\\0\end{bmatrix} = \begin{bmatrix}0\\0\end{bmatrix}$$

$$h_1 = \tanh\left(\begin{bmatrix} 0.5 \\ 0.1 \end{bmatrix}\right)$$

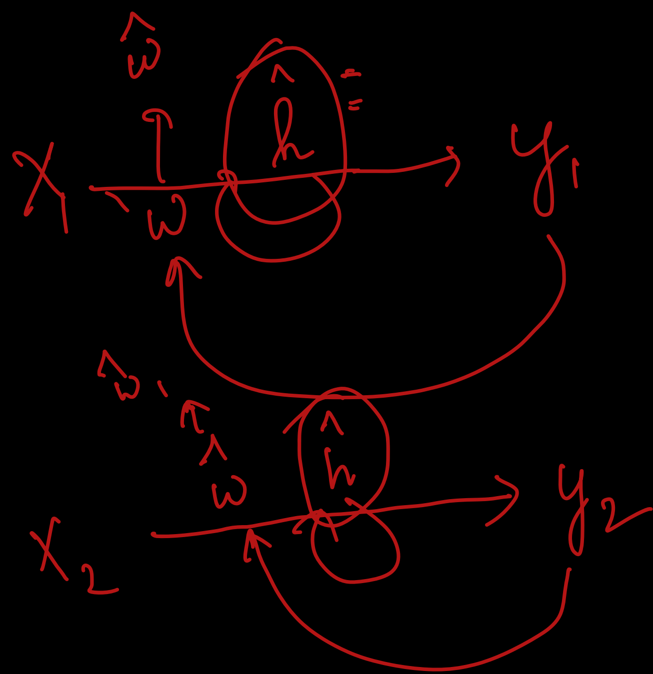
$$\boxed{h_1 = \begin{bmatrix} 0.46 \\ 0.09 \end{bmatrix}}$$

Calculate y_1

$$y_1 = W_{hy} h_1 + b_y$$

$$= \begin{bmatrix} 1 & -1 \end{bmatrix} \begin{bmatrix} 0.46 \\ 0.09 \end{bmatrix} + 0$$

$$\boxed{y_1 = 0.362}$$



Analogy

Input x_t	Vector at time t	Reading a word
Previous hidden state h_{t-1}	Memory from previous step	your memory of what you read before
New hidden state h_t	Updated memory with new input	your updated understanding after reading current word
Output y_t	Prediction of next word	your answer for next word or plot

X

2×1

$W \times [\ ]$

$2 \times 2 \quad 2 \times 1$

2×1

$3 \times 2 \quad 2 \times 5$

3×5

$\cancel{2 \times 1}$

$1 \times 2 \quad 2 \times 1$

$= 1 \times 1$

Limitation

Vanishing & Exploding,

Loosing Long Term Memory

Long Short term Memory

(LSTM)

kind of RNN

Capturing long term dependencies

memory cell

gates

Keep, forget, output

Components :

Forgot Gate - Decides what to forgot from previous state

Input Gate - Decides the new information to add to cell state

Candidate Cell State - Creates new candidate values to add

Output Gate - Decides what part of cell state to output as hidden state

Analogy LSTM

Input x_t = you receive an email from your boss

Previous hidden state h_{t-1} = your recent verbal summary to your boss - short

Previous cell state C_{t-1} = your detailed notebook - long

Each day you decide

Forget gate - what old notes to erase from your notebook

Input gate - what new notes to add based on new mail

Candidate cell state - Draft new information to add

Update Notebook - Combine what you keep and add to your notebook

Output gate - How you respond based on input and hidden state

LSTM mathematical Formulas

Forgot gate = $f_t = \sigma\left(W_f \cdot \overbrace{[\underbrace{h_{t-1}}_{\text{short}}, x_1]}^{\text{concat}} + b_f\right)$

Input gate = what new information to store

$$i_t = \sigma\left(W_i\,[h_{t-1}, x_1] + b_i\right)$$

Candidate cell state   New Candidate values to add

$$\boxed{\tilde{C}_t = \tanh\left(W_c\,[h_{t-1}, x_1] + b_c\right)}$$

Update the cell state

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

Output gate

$$o_t \Rightarrow \sigma\left(W_o\,[h_{t-1}, x_t] + b\right)$$

Hidden state

$$h_t = o_t \odot \tanh(C_t)$$