Natural Language Processing (NLP) from Bidirectional Recurrent Neural Networks

By Dr. Ganesh, IIT Delhi & Harvard Stanford

Recap: ① RNN (Recurrent Neural Network)

↳ ② Feedforward

- ③ Back Propagation → does flow in future
- ④ Vanishing gradient problem ✓ bad
- ⑤ Exploding gradient → bad
- ⑥ Results ok

weakness

text → Vector — [10, 11, ...]

Bag of words → Bag of words (counting)

3 low, 1 mid, 1 high

vocab size = 10,000

Bag of words / Bag of bigrams

Bag-of-words / Bag-of-bigrams

NLP → regular expressions

① # ! - + - ? ^ ' ( ) → sentiment info.

Data processing → Bain training

input ok

data size = m, long [0, 1, 2, 3, 4, ...]

② text → $\begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}$ → NLP models deep nets

③ word → $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}$ → NLP model

④ sent → $\begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}$ → NLP model

text grow

vocabsize grow → $\begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}_{10}$ deep nodes

Disadvantage of RNN

① Shorter sequences is a nice way out

I am → I grow up

danger sequence → long term dependency problem.

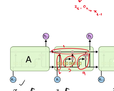
I was born in France ..... I speak French fluently

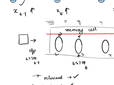
② Vanishing gradient problem (✓ NLP)

③ LSTM (long short term memory).

I was born in France ..... I speak French fluent

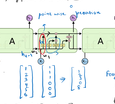
NLP → feature engineering from personal way





Parts

① memory cell
② Forget gate
③ Input gate
④ Output gate



→ relevant ✓

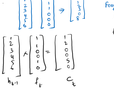
→ irrelevant ✗

memory cell & forget → memory & forget → relevant info / irrelevant info

Generating LSTM

I was born in France

→ my parents are at home



① ② Pointwise operation

⊤ → bigram fn

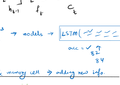
⊤ → concatenate

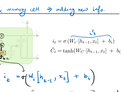
⊤ → copying

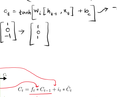
Forget Gate

$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f, w_{t-1}) + b)$

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





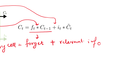


Keep ↳ models → LSTM (✓ ✗ ✗ ✗ ✗) ✓

acc → ✓ 97

87

Input & memory cell → adding new info



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

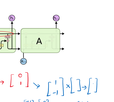
$\tilde{C}_t = \tanh(W_c[h_{t-1}, x_t] + b_c)$

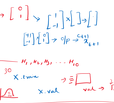
$f_t = \sigma(W_f[h_{t-1}, x_t] + b_f)$

I was born in France

$C_t = \tanh\left[W_c[h_{t-1}, x_t] + b_c\right]$ → -1 to +1

$\begin{bmatrix} 0.5 \\ 0.1 \\ 0.2 \end{bmatrix} \times \begin{bmatrix} 0 \\ 0.1 \\ 0.2 \end{bmatrix} \Rightarrow \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}$





memory cell = forget + relevant info

Output gate

$o_t = \sigma(W_o[h_{t-1}, x_t] + b_o)$

$h_t = o_t * \tanh(C_t)$

$\begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \times \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} = \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix}$

Review of LSTM

$\sigma\left(W\left(h_t, x_{t+1}\right) + b\right) \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix} \times \begin{bmatrix} 0 \\ 1 \end{bmatrix} \rightarrow \begin{bmatrix} 0 \\ 1 \end{bmatrix}$

$\begin{bmatrix} 0.5 \\ 0.1 \\ 0.2 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix} = \begin{bmatrix} 0.5 \\ 0.1 \\ 0.2 \end{bmatrix}$

$\begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \rightarrow \sigma/p \rightarrow \begin{bmatrix} 0.1 \\ 0.2 \end{bmatrix} \rightarrow \hat{x}_{t+1}$

(10) Dataset $M_1, M_2, M_3 \cdots M_{10}$

→ 70, 30 training

30 → X test

→ 5 → test → 5

X test

→ X test

M_1, M_2, M_3

Bad dataset

data distribution

→ 70 → 20 → 10

unit = 10,000