NLP Seminar

Recurrent Modeling

2022. 04. 15

KISTI - UST IKJE CHOI

CONTENTS

01 Bidirectional RNNs

02 Encoder-Decoder Seq2Seq

03 Deep Recurrent Networks

04 Long-Term Dependencies

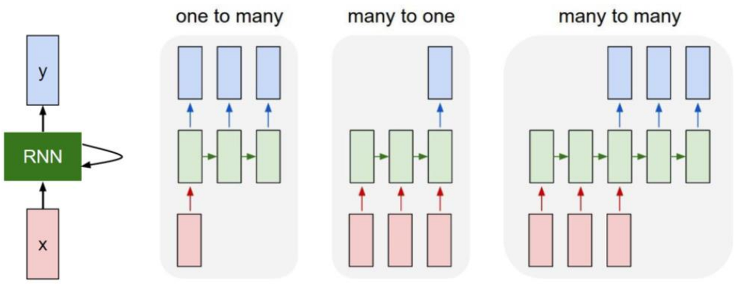
05 Skip Connections and Leaky Units

06 Gradient Clipping

07 Reference

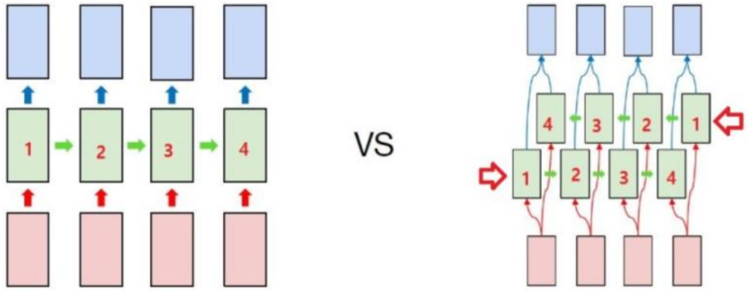
Bidirectional RNNs

Recurrent Neural Networks(RNNS)



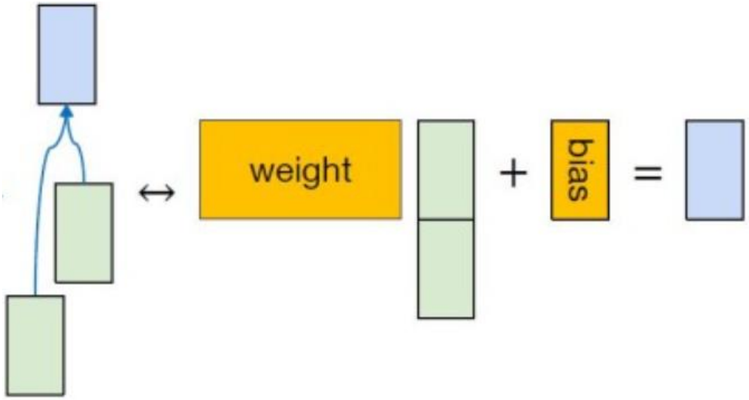
- ✓ Language Model : to predict the next word
- ✓ RNNs : Using sequence inputs, it can predict output(s)

Bidirectional RNNs



- ✓ RNNs : move forward through time
- ✓ Bidirectional RNNs : move forward & backward through time

Any Advantages of Bidirectional RNNs?

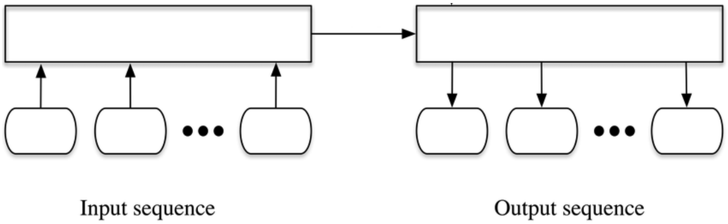


- ✓ Update weight with considering past & future sequence of inputs
  E.G) Twinkle, twinkle, ________ star

Encoder–Decoder Sequence–to–Sequence Architecture

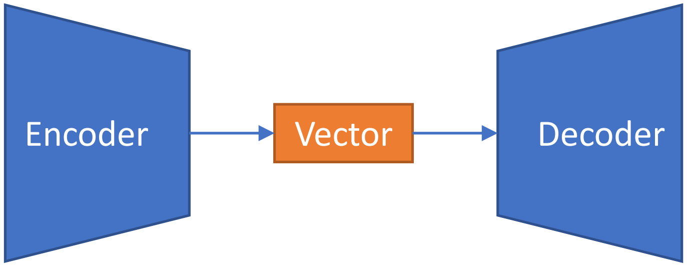
Sequence-to-Sequence

- ✓ Predict Sequence outputs by Using Sequence inputs



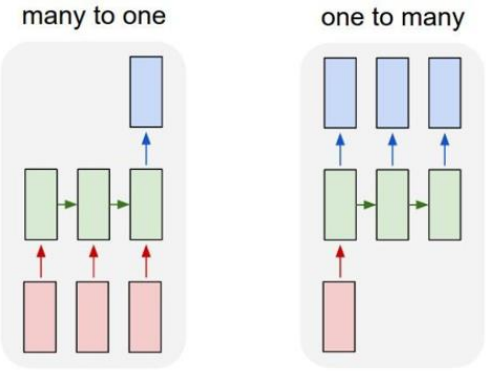
Encoder-Decoder

- ✓ Encoder : To transform variable-length sequence input to fixed shape of vector
- ✓ Decoder : Covert encoded one to output which is variable-length sequence



How it works?

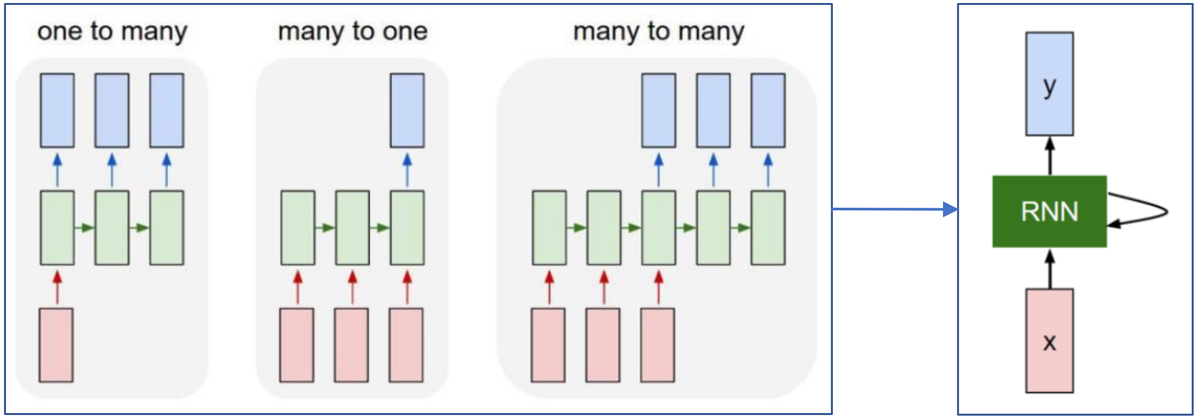
- ✓ Two RNNs architectures
- ✓ Maximize $\log P(y^{(1)}, \ldots, y^{(n_y)} | x^{(1)}, \ldots, x^{(n_x)})$ over all the pairs of x and y sequences in the training set (when x = input & y = output)



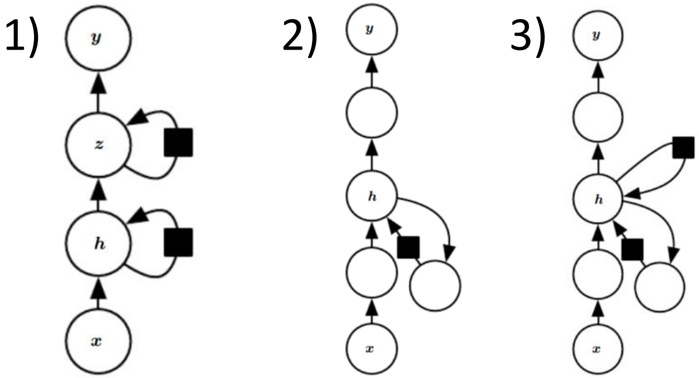
Deep Recurrent Networks

The computation in most recurrent neural networks can be decomposed into three.

1) From the input to the hidden state
2) From the previous hidden state to the next hidden state
3) From the hidden state to the output



1) Recurrent states broken down in groups
2) Deeper computation in hidden-to-hidden
3) Multilayer Perceptron with a single hidden layer



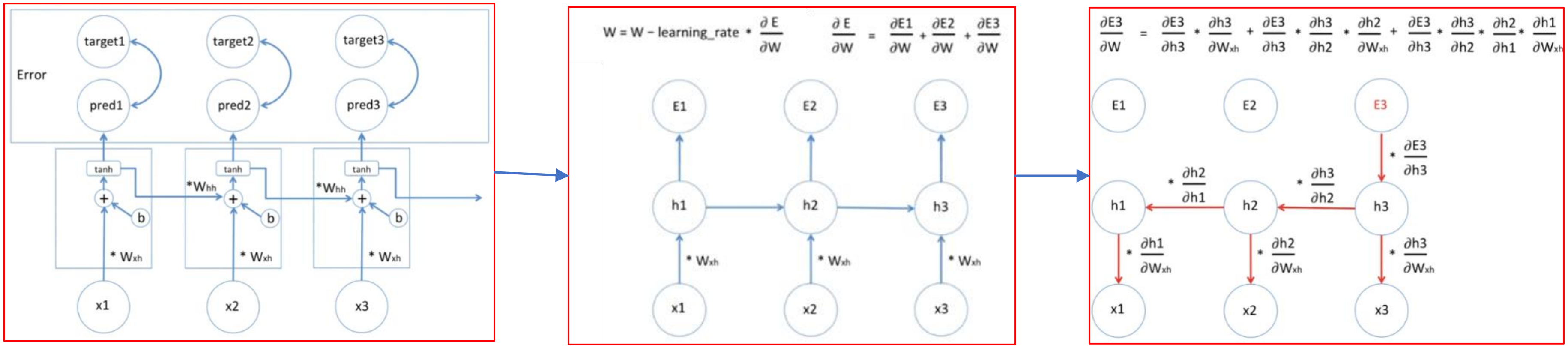
Long–Term Dependencies

Can you predict masked word?

- ✓ E.g.)

  1. The clouds are in the <masked>.
  2. I lived in <masked>.

  ➢ To predict (2.)'s masked word, we need to look at previous words or sentences.

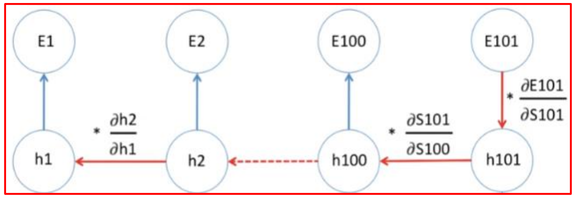


$$W = W - \text{learning\_rate} * \frac{\partial E}{\partial W} \qquad \frac{\partial E}{\partial W} = \frac{\partial E1}{\partial W} + \frac{\partial E2}{\partial W} + \frac{\partial E3}{\partial W}$$

$$\frac{\partial E3}{\partial W} = \frac{\partial E3}{\partial h3} * \frac{\partial h3}{\partial W_{xh}} + \frac{\partial E3}{\partial h3} * \frac{\partial h3}{\partial h2} * \frac{\partial h2}{\partial W_{xh}} + \frac{\partial E3}{\partial h3} * \frac{\partial h3}{\partial h2} * \frac{\partial h2}{\partial h1} * \frac{\partial h1}{\partial W_{xh}}$$

$* \frac{\partial E3}{\partial h3}$ $* \frac{\partial h3}{\partial h2}$ $* \frac{\partial h2}{\partial h1}$ $* \frac{\partial h1}{\partial W_{xh}}$ $* \frac{\partial h2}{\partial W_{xh}}$ $* \frac{\partial h3}{\partial W_{xh}}$

IF



$* \frac{\partial E101}{\partial S101}$ $* \frac{\partial S101}{\partial S100}$ $* \frac{\partial h2}{\partial h1}$

, then multiplied derivative values become very large or small.

At some point, does not update weight well.

- ✓ Vanishing gradients : it is hard to improve cost
- ✓ Exploding gradients : it makes learning unstable

Skip Connections and Leaky Units

Goal : to deal with long-term dependencies before LSTM (Long Short-Term Memory)

- ✓ Skip Connections
  - ➢ Skips some of the layers in the neural network and feeds the output of one layer as the input to the next layers.

- ✓ Leaky Units
  - ➢ The product of derivatives close to one is to have units with linear self-connections and a weight near one on these connections.

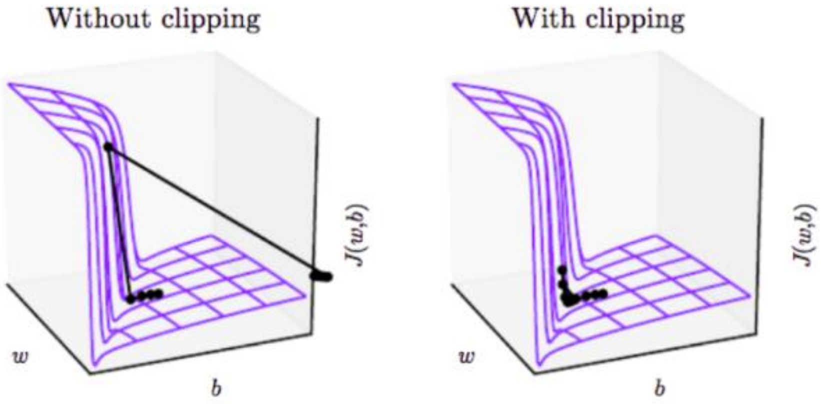
Gradient Clipping

Goal : to avoid Gradient Exploding

- ✓ During Backpropagation, avoid over-update Gradient

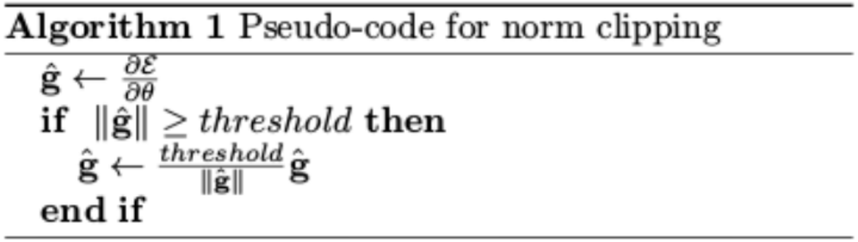
- ✓ Advantage
  - ➢ We can give higher learning rate. -> reducing learning time
  - ➢ Avoid Local minimum

- ✓ Disadvantage
  - ➢ We manually set the threshold



Without clipping

With clipping



Algorithm 1 Pseudo-code for norm clipping

```
$\hat{\mathbf{g}} \leftarrow \frac{\partial \mathcal{E}}{\partial \theta}$
if $\|\hat{\mathbf{g}}\| \geq threshold$ then
    $\hat{\mathbf{g}} \leftarrow \frac{threshold}{\|\hat{\mathbf{g}}\|}\hat{\mathbf{g}}$
end if
```

Reference

- ✓ Sequence Modeling: Recurrent and Recursive Neural Nets
- ✓ ratsgo's blog (for textmining)
- ✓ https://www.analyticsvidhya.com/blog/2021/08/all-you-need-to-know-about-skip-connections/
- ✓ https://lswook.tistory.com/105
- ✓ https://eehoeskrap.tistory.com/582

THANK YOU