

# 第十讲 循环神经网络

2019年5月13日 16:08

## 1. 梯度流管理

从Alexnet开始往后网络越来越深，越深的网络在分类任务中的表现也会越好，但是深层的网络梯度很容易消失，训练时不容易收敛，在BN出现之前VGG和Googlenet都是通过一些技巧来使梯度容易流向底层。

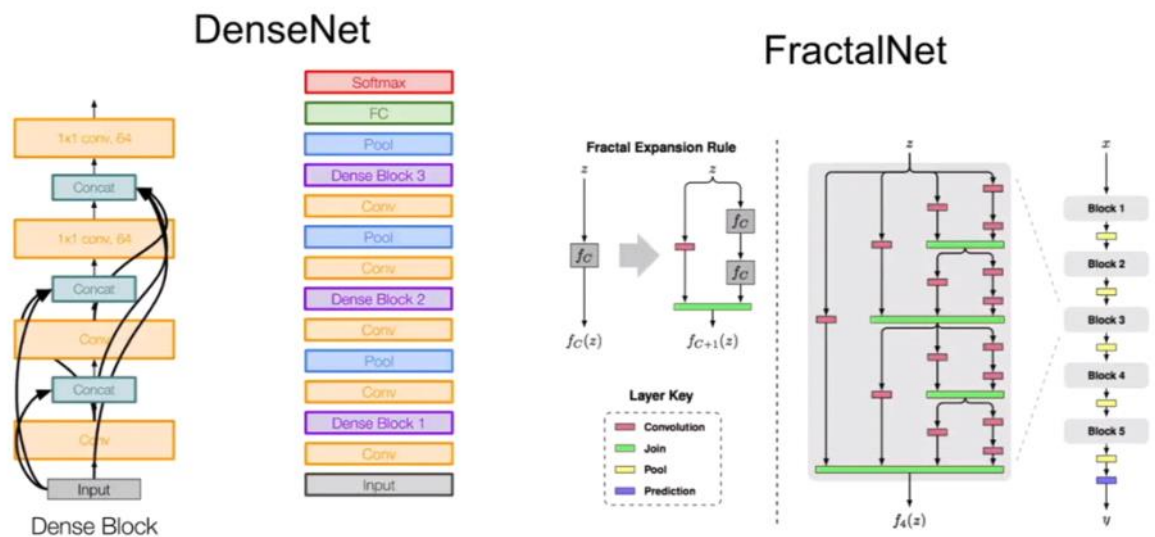


图10.1.1 DenseNet和FractalNet

从梯度流管理的角度来看这两个网络，通过加入恒等链接和捷径来使梯度更容易的流向底层，更快更容易的收敛。 ???

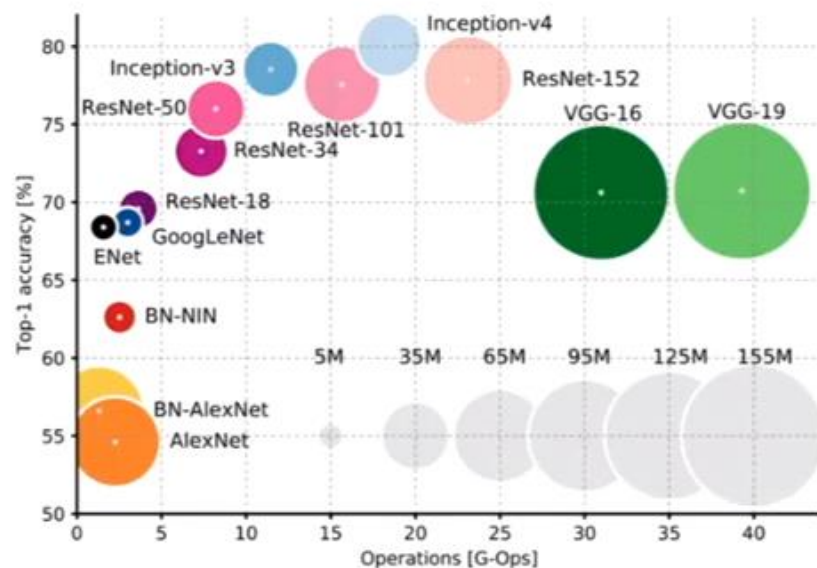


图 10.1.2 各个模型的运算量、参数量、准确率对比

## 2.RNN

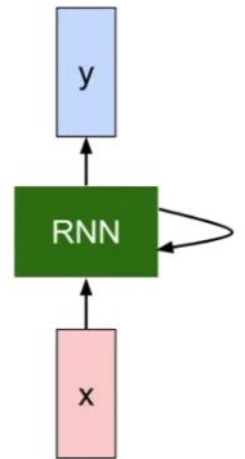
1.在之前学习的网络输入和输出大小都是固定的，但是当输入 网络的内容大小是不固定(相当于输入层shape不确定)的时候就要用到RNN。

# Recurrent Neural Network

We can process a sequence of vectors  $\mathbf{x}$  by applying a **recurrence formula** at every time step:

$$\boxed{h_t} = \boxed{f_W}(\boxed{h_{t-1}}, \boxed{x_t})$$

new state / old state input vector at some time step  
some function with parameters  $W$

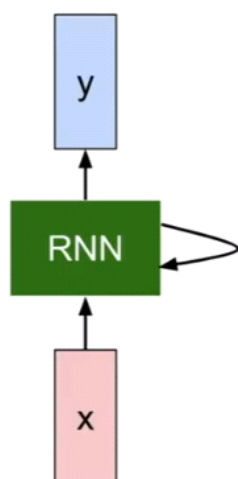


$f_W$ 可以理解为一个递归函数，参数为新的 $x_t$ 和上次计算得出的返回值，根据这两个参数得到新的返回值并作为参数传递给下次递归。

这样输入 $x$ 大小不固定， $x$ 每一个子元素输入都会生成一个 $h_t$ 并对之后的输出产生影响。（这是所谓的有时序吗？）

## (Vanilla) Recurrent Neural Network

The state consists of a single “hidden” vector  $\mathbf{h}$ :



$$h_t = f_W(h_{t-1}, x_t)$$



$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

与传统的网络相比，在RNN的递归层有两个参数矩阵，分别计算隐藏层和输入元素。

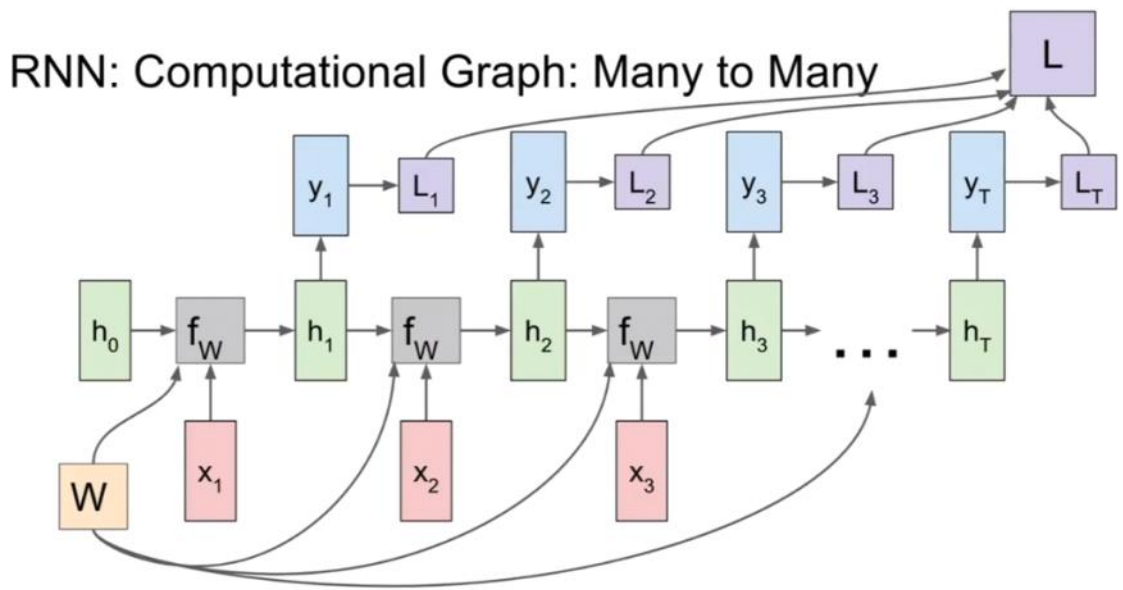


图10.1.3 输入和输出都不确定的RNN网络

### RNN: Computational Graph: Many to One

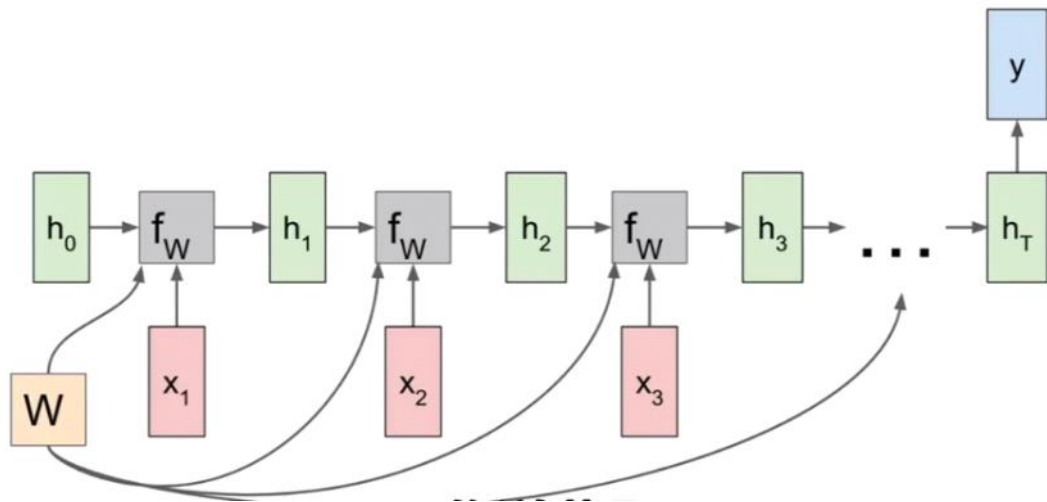


图10.1.4 输入不确定输出确定的RNN网络

### RNN: Computational Graph: One to Many

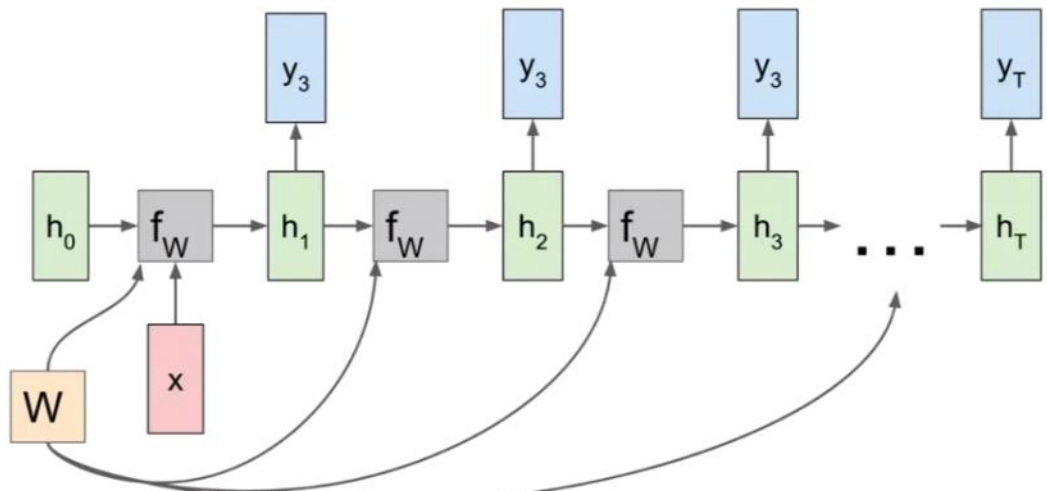


图10.1.5 输入确定输出不确定的RNN网络

## Sequence to Sequence: Many-to-one + one-to-many

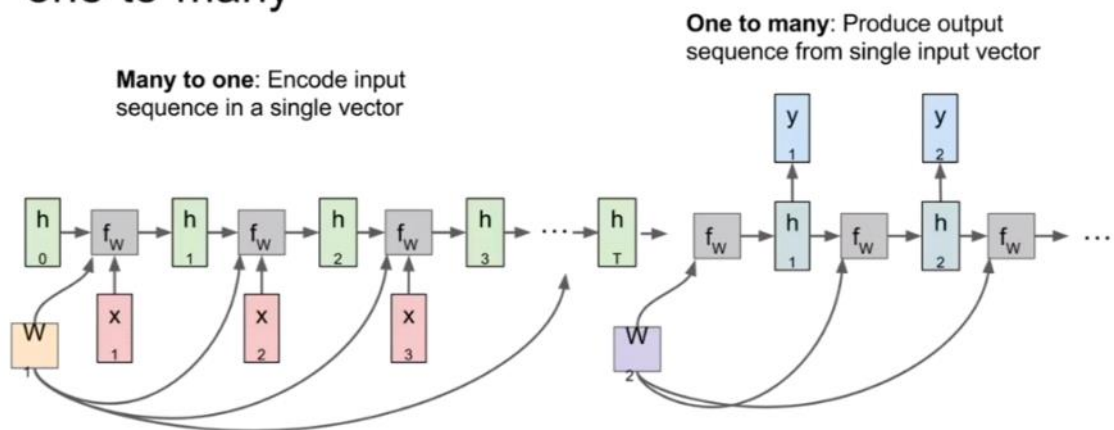


图10.1.6 Sequence to Sequence Model

机器翻译中常用的模型，输入不定长序列，输出不定长序列。整个过程可以看作是多对一和一对多的模型组合 (encode和decode)。

### 3. Language modeling

#### 3.1 RNN对语言的建模

##### Example: Character-level Language Model

Vocabulary:  
[h,e,l,o]

Example training  
sequence:  
"hello"

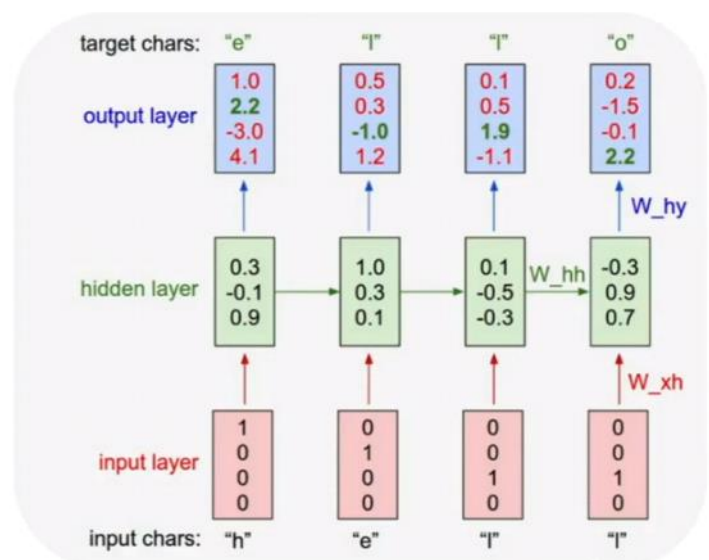


图10.3.1 RNN在Character级别的语言模型

目的：输入若干个字母，输出预测的下一个字母。

训练过程：两个参数矩阵 $W_{xh}$ 、 $W_{hy}$ ，根据训练句子逐个输入字母，得到输出之后计算softmax损失，梯度下降求得最优参数矩阵。

## Truncated Backpropagation through time

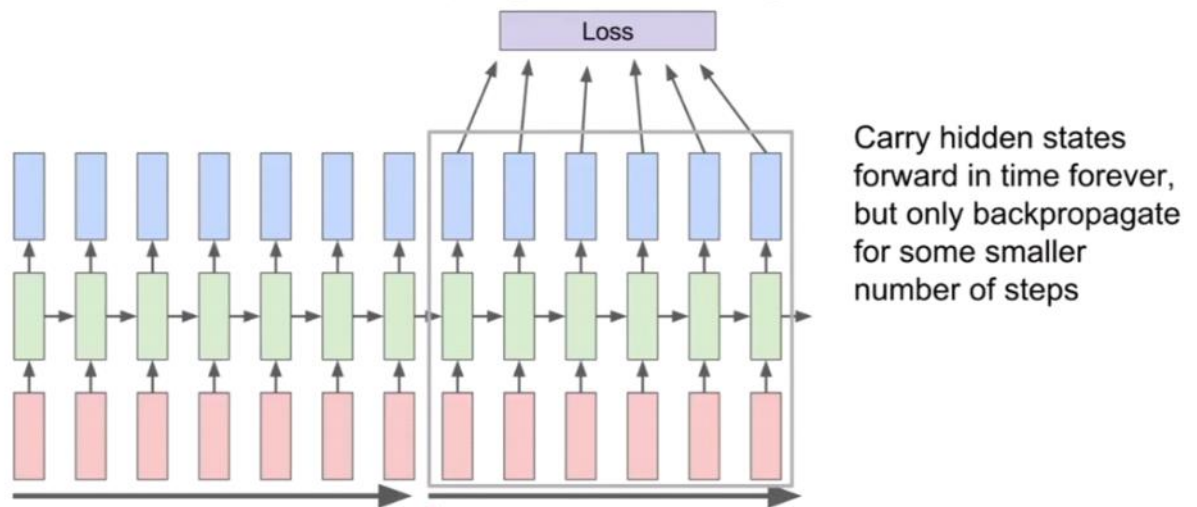


图10.3.2 语言的训练数据比较长时沿着时间一段一段的前馈和反向传播

一个RNN对语言建模的demo:

<https://gist.github.com/karpathy/d4dee566867f8291f086>

## 4.Other RNN modeling

### 4.1 Image Captioning

目的: 根据图片生成一段描述该图片的文字

模型: (来自讲师自己实验室的模型)

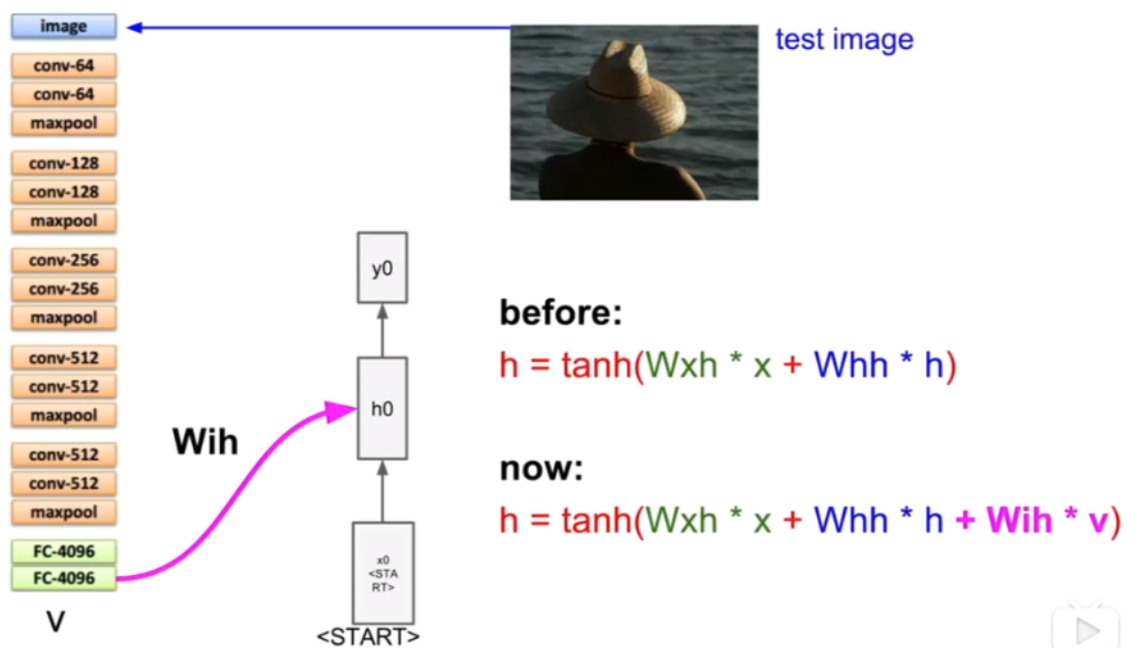


图10.4.1 Image Captioning 模型

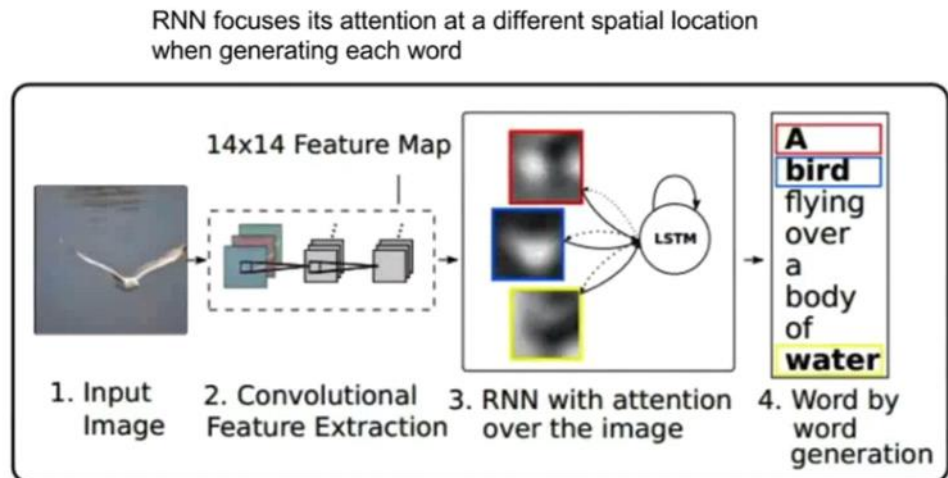
除了传统的连个参数矩阵, 该模型还加入了第三个参数矩阵 $w_{ih} * v$ (讲师只提了一句这个是图像信息, 后面没解释具体怎样的)

该模型是一个CNN加RNN的集成模型, 去掉CNN的最后一层全连接层, 直接将倒数第二层的输出作为RNN的输入, 每预测一个词后将该词作为输入输入到RNN中预测下一个词直到预测到句号。

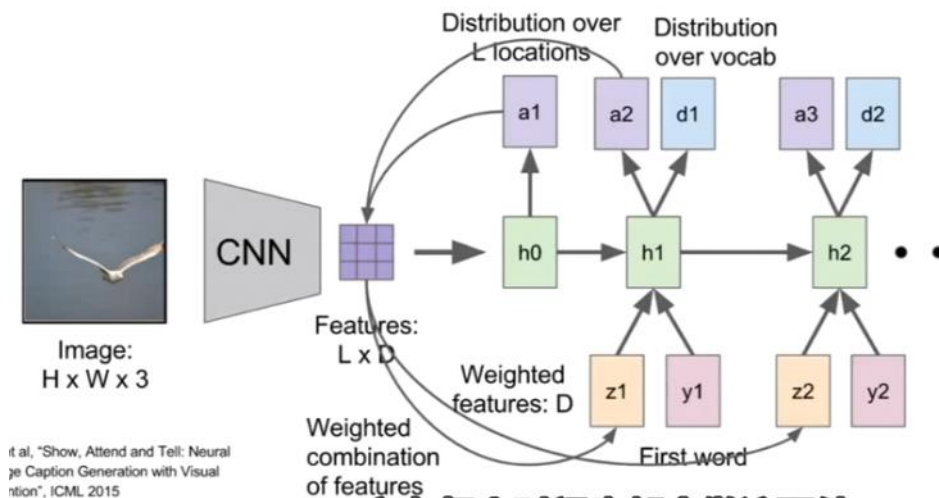


## 4.2 Image Captioning With Attention

### Image Captioning with Attention



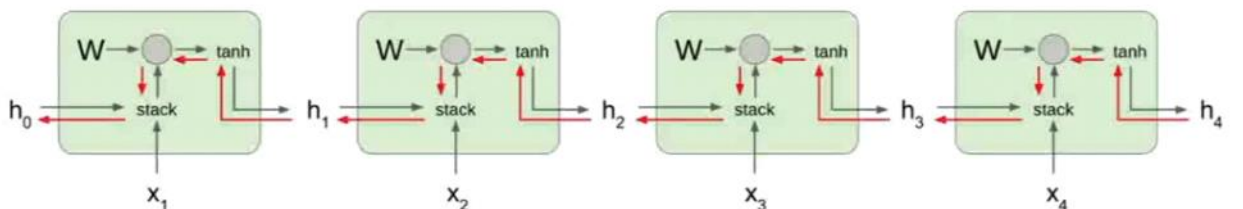
### Image Captioning with Attention



## 4.3 RNN梯度流

### Vanilla RNN Gradient Flow

Bengio et al., "Learning long-term dependencies with gradient descent is difficult", IEEE Transactions on Neural Networks, 1994  
Pascanu et al., "On the difficulty of training recurrent neural networks", ICML 2013



Computing gradient of  $h_0$  involves many factors of  $W$  (and repeated  $\tanh$ )

Largest singular value  $> 1$ :  
**Exploding gradients**

Largest singular value  $< 1$ :  
**Vanishing gradients**

**Gradient clipping:** Scale gradient if its norm is too big

```
grad_norm = np.sum(grad * grad)
if grad_norm > threshold:
    grad *= (threshold / grad_norm)
```

**梯度爆炸:**

RNN递归神经元梯度在反向传播时如果其值大于1, 那么在递归n次之后就会产生梯度爆炸, 解决办法是梯度截断,  $\text{if}(\text{grad\_norm} > \text{threshold}) \text{ grad} *= (\text{threshold} / \text{grad\_norm})$

**梯度消失:**

RNN递归神经元梯度在反向传播时如果其值小于1, 那么在递归n次之后就会产生梯度消失, 解决办法时更换更加复杂的RNN模型。

为了解决RNN梯度爆炸与梯度消失的问题引出LSTM.

## 4.4 LSTM