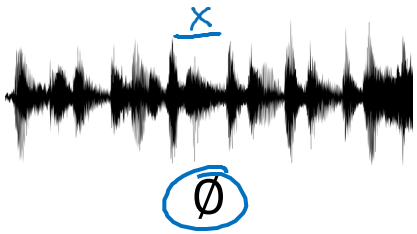
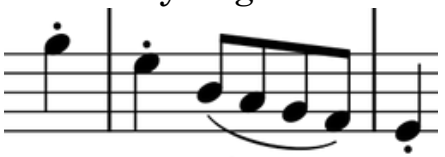
循环神经网络(RNN)

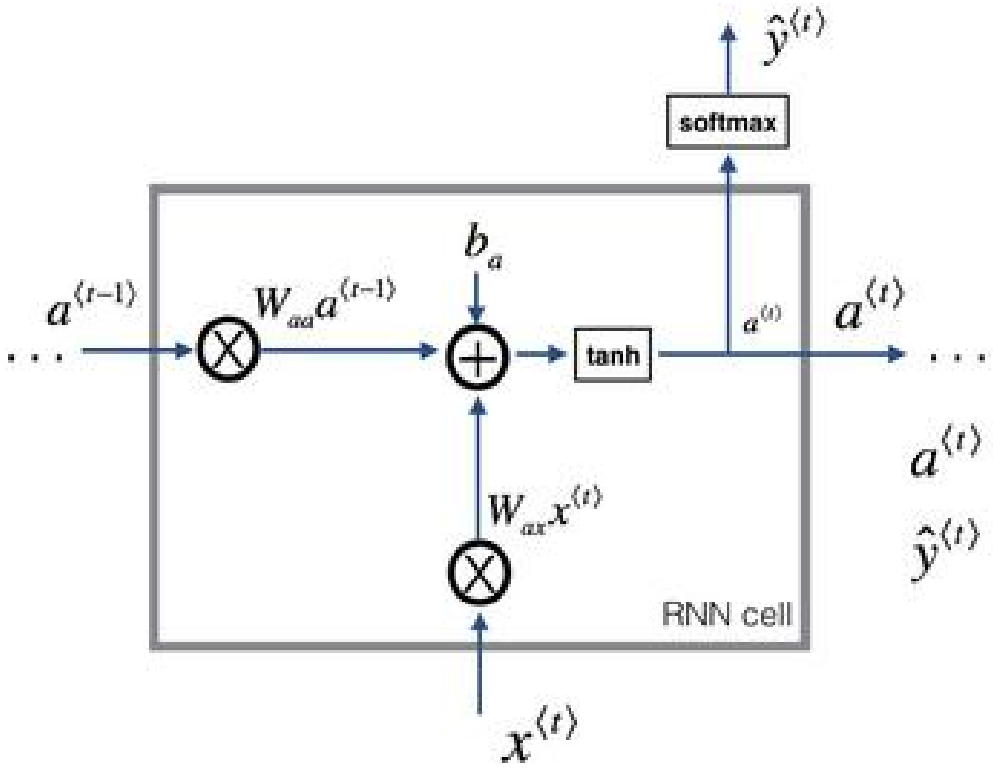
主要内容

- 基本循环神经网络
- GRU网络
- LSTM网络
- 双向RNN网络
- 典型应用

Examples of sequence data

Speech recognition		"The quick brown fox jumped over the lazy dog."
Music generation		
Sentiment classification	"There is nothing to like in this movie."	★☆☆☆☆
DNA sequence analysis	AGCCCCTGTGAGGAACTAG	AGCCCCTGTGAGGAACTAG
Machine translation	Voulez-vous chanter avec moi?	Do you want to sing with me?
Video activity recognition		Running
Name entity recognition	Yesterday, Harry Potter met Hermione Granger.	Yesterday, Harry Potter met Hermione Granger.

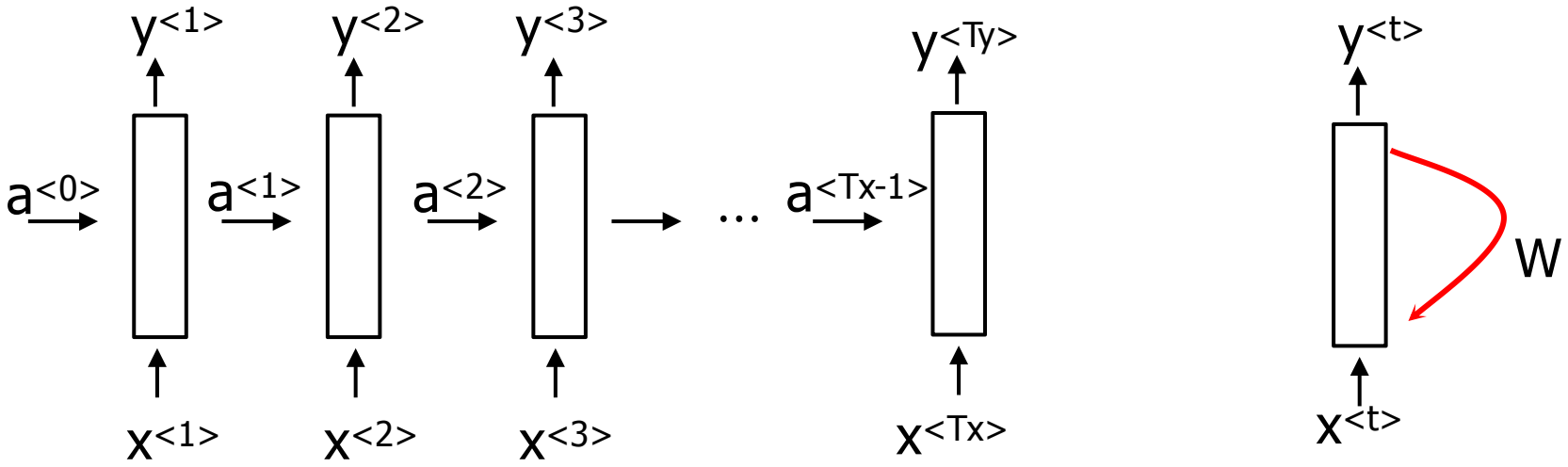
基本的RNN单元



$$a^{\langle t \rangle} = \tanh(W_{ax}x^{\langle t \rangle} + W_{aa}a^{\langle t-1 \rangle} + b_a)$$

$$\hat{y}^{\langle t \rangle} = soft\max(W_{ya}a^{\langle t \rangle} + b_y)$$

RNN前向传播



$$a^{<1>} = g_1(W_{aa}a^{<0>} + W_{ax}x^{<1>} + b_a)$$

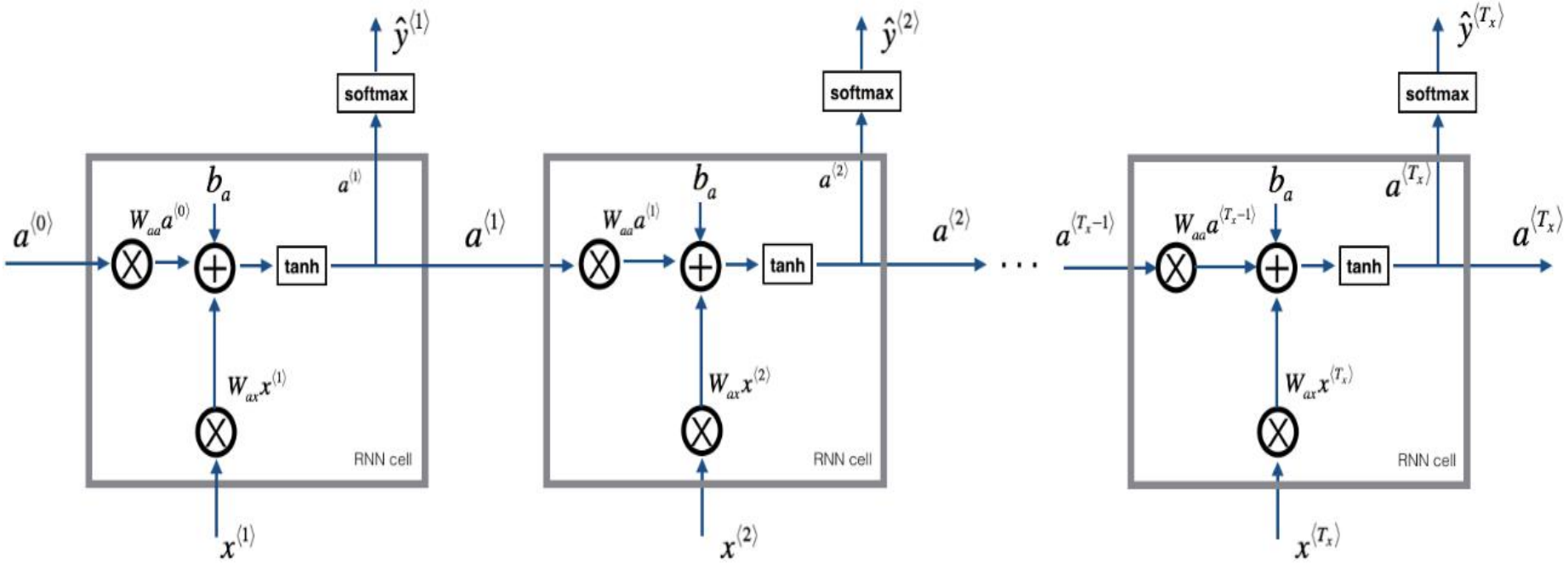
$$\hat{y}^{<1>} = g_2(W_{ya}a^{<1>} + b_y)$$

$$a^{<t>} = g_1(W_{aa}a^{<t-1>} + W_{ax}x^{<t>} + b_a)$$

$$\hat{y}^{<t>} = g_2(W_{ya}a^{<t>} + b_y)$$

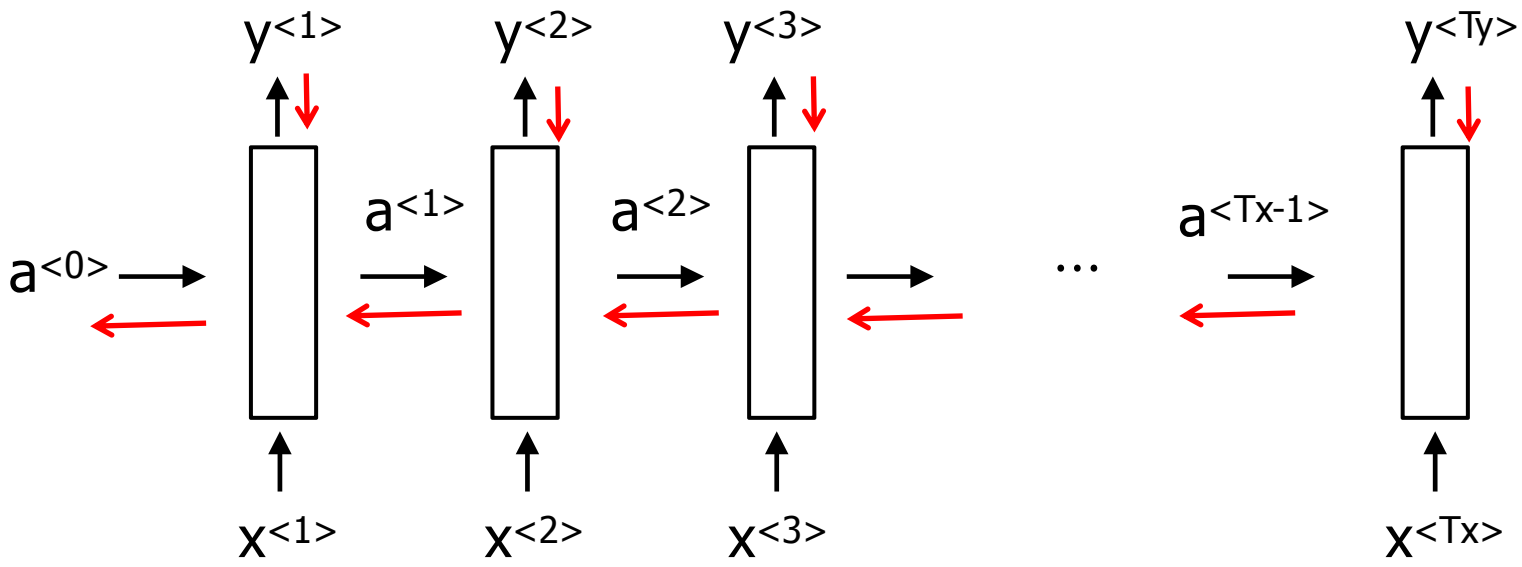
- g_1一般选择tanh/ReLU
- g_2一般选择Sigmoid/Softmax

RNN前向传播示意图



Notice: the same function and the same set of parameters are used at every time step.

通过时间的反向传播Backpropagation through time



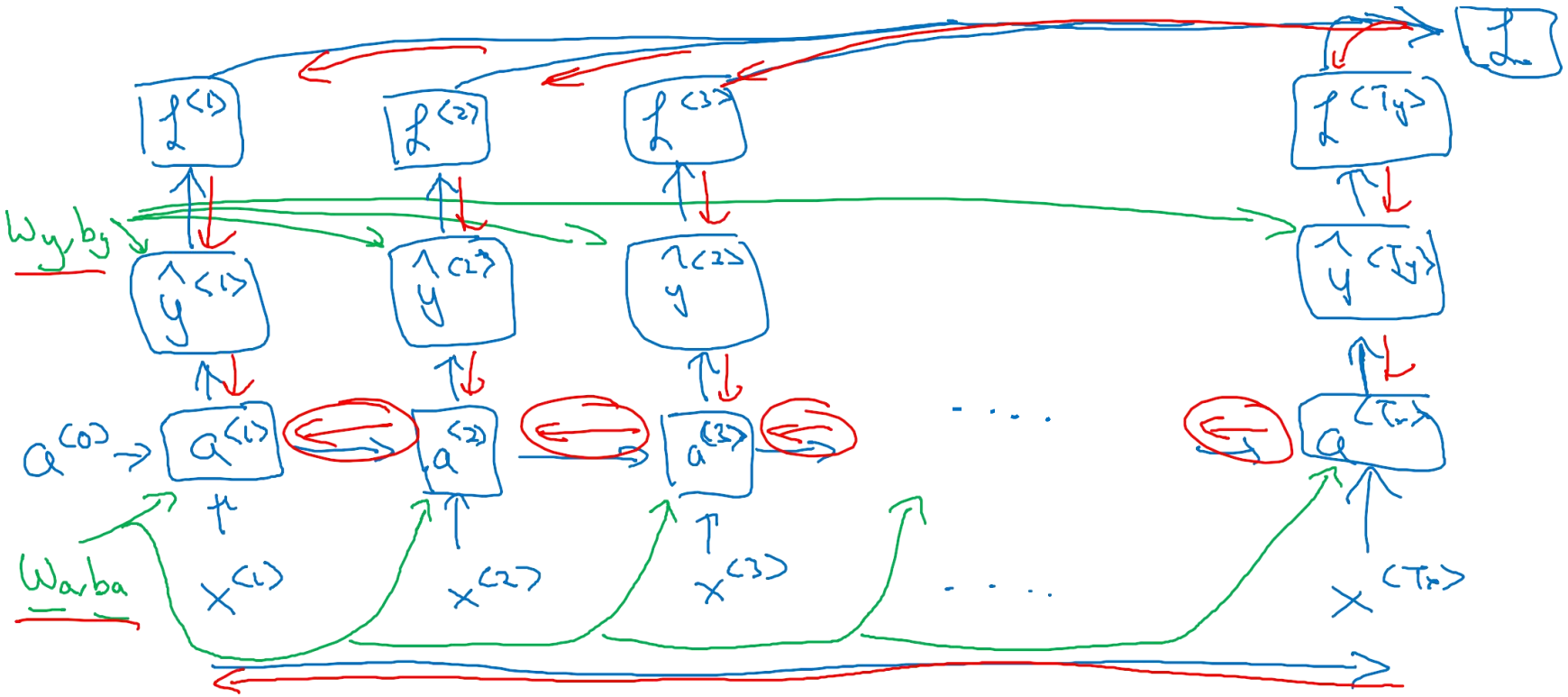
为了计算反向传播，需要一个损失函数。先定义一个样本的损失函数，即关于单个位置上或者说某个时间步t上某个样本的预测值的损失函数：

$$L^{<t>}(\hat{y}^{<t>}, y^{<t>}) = -y^{<t>}\log\hat{y}^{<t>} - (1-\hat{y}^{<t>})log(1-\hat{y}^{<t>})$$

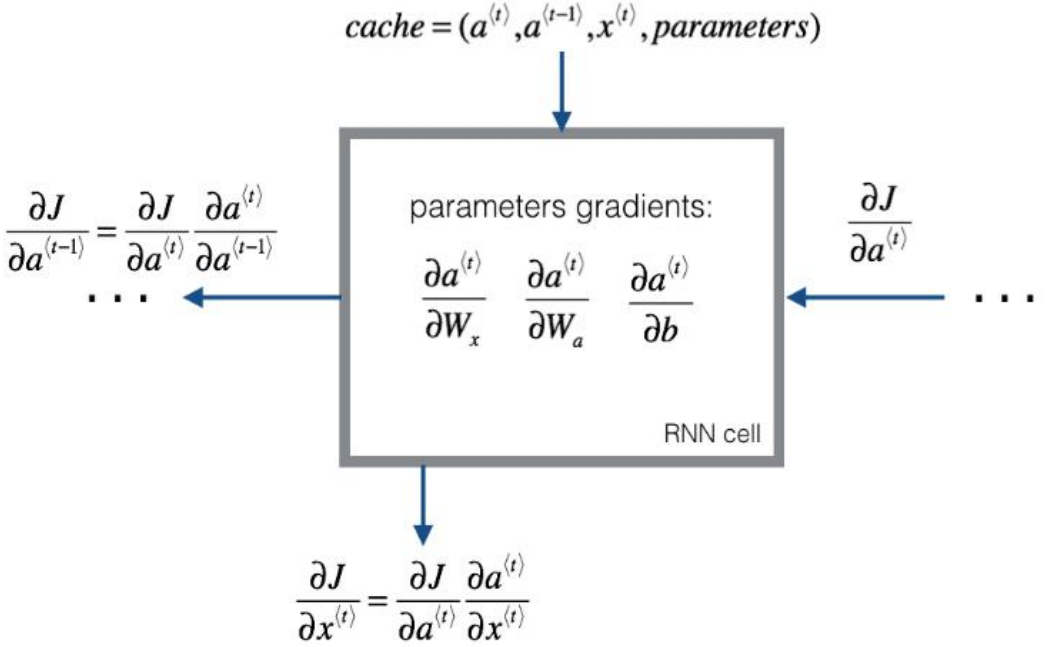
通过时间的反向传播Backpropagation through time

整个序列的损失函数L定义为：

$$L(\hat{y},y)=\sum_{t=1}^{T_x}L^{<t>}(\hat{y}^{<t>},y^{<t>})$$



RNN反向传播示意图



$$a^{\langle t \rangle} = \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)$$

$$\frac{\partial \tanh(x)}{\partial x} = 1 - \tanh(x)^2$$

$$\frac{\partial a^{\langle t \rangle}}{\partial W_{ax}} = (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2) . x^{\langle t \rangle T}$$

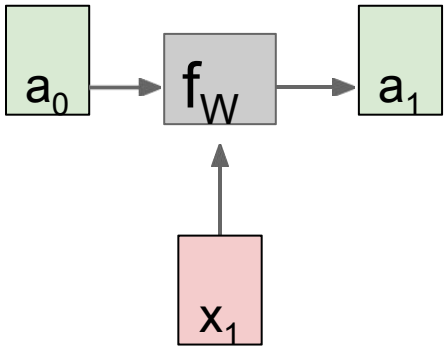
$$\frac{\partial a^{\langle t \rangle}}{\partial W_{aa}} = (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2) . a^{\langle t-1 \rangle T}$$

$$\frac{\partial a^{\langle t \rangle}}{\partial b} = \sum_{batch} (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

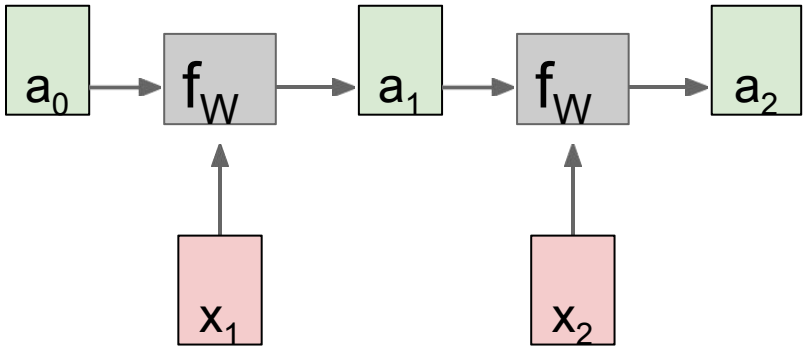
$$\frac{\partial a^{\langle t \rangle}}{\partial x^{\langle t \rangle}} = W_{ax}{}^T . (1 - \tanh(W_{ax} x^{\langle t \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

$$\frac{\partial a^{\langle t \rangle}}{\partial a^{\langle t-1 \rangle}} = W_{aa}{}^T . (1 - \tanh(W_{ax} x^{\langle t-1 \rangle} + W_{aa} a^{\langle t-1 \rangle} + b)^2)$$

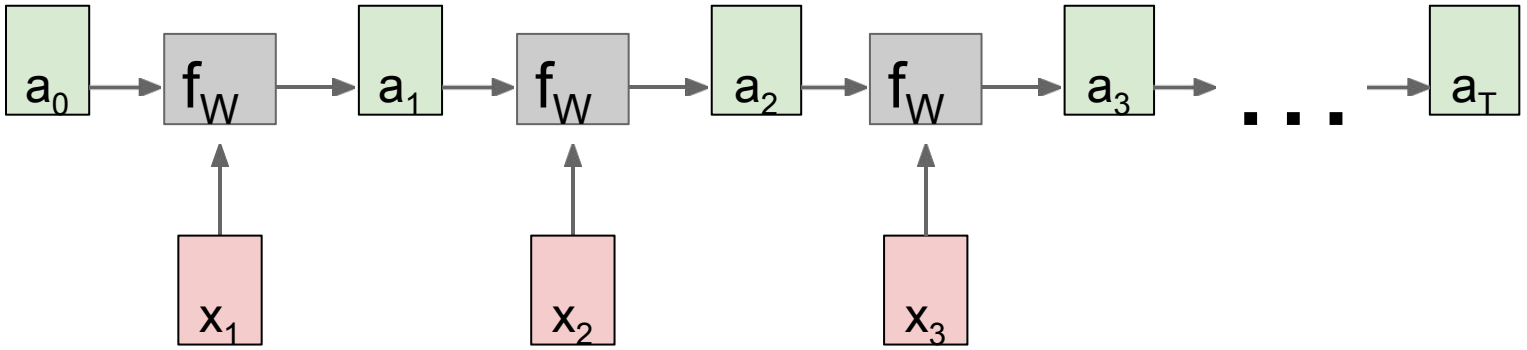
RNN: Computational Graph



RNN: Computational Graph

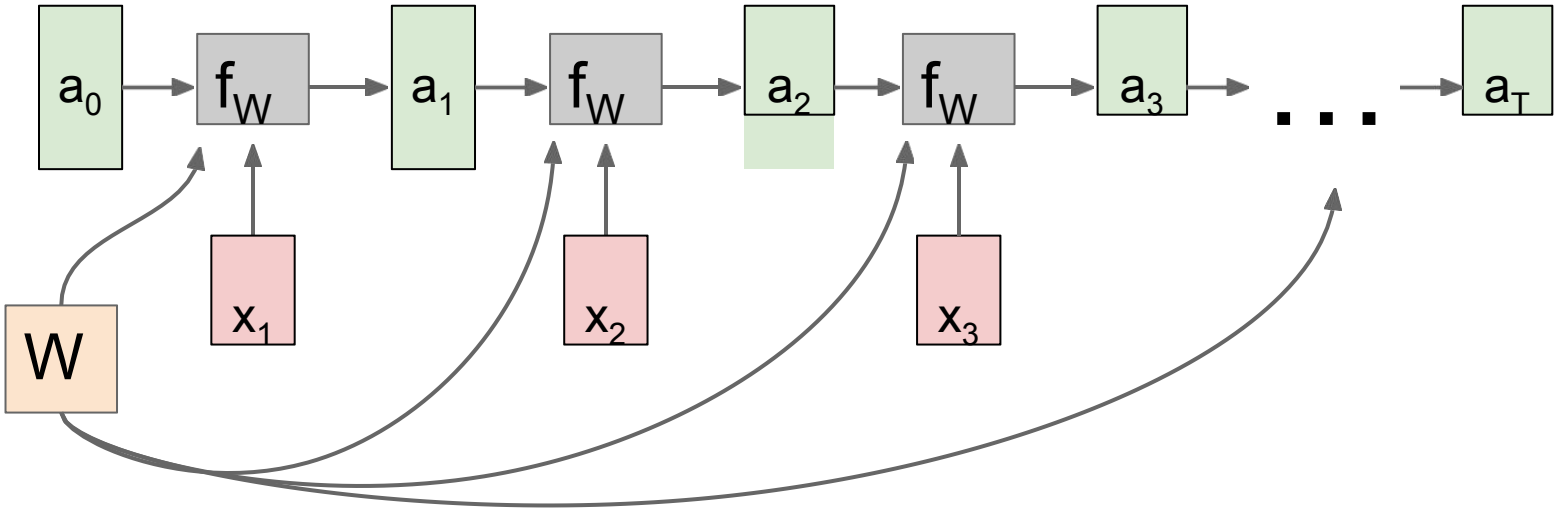


RNN: Computational Graph

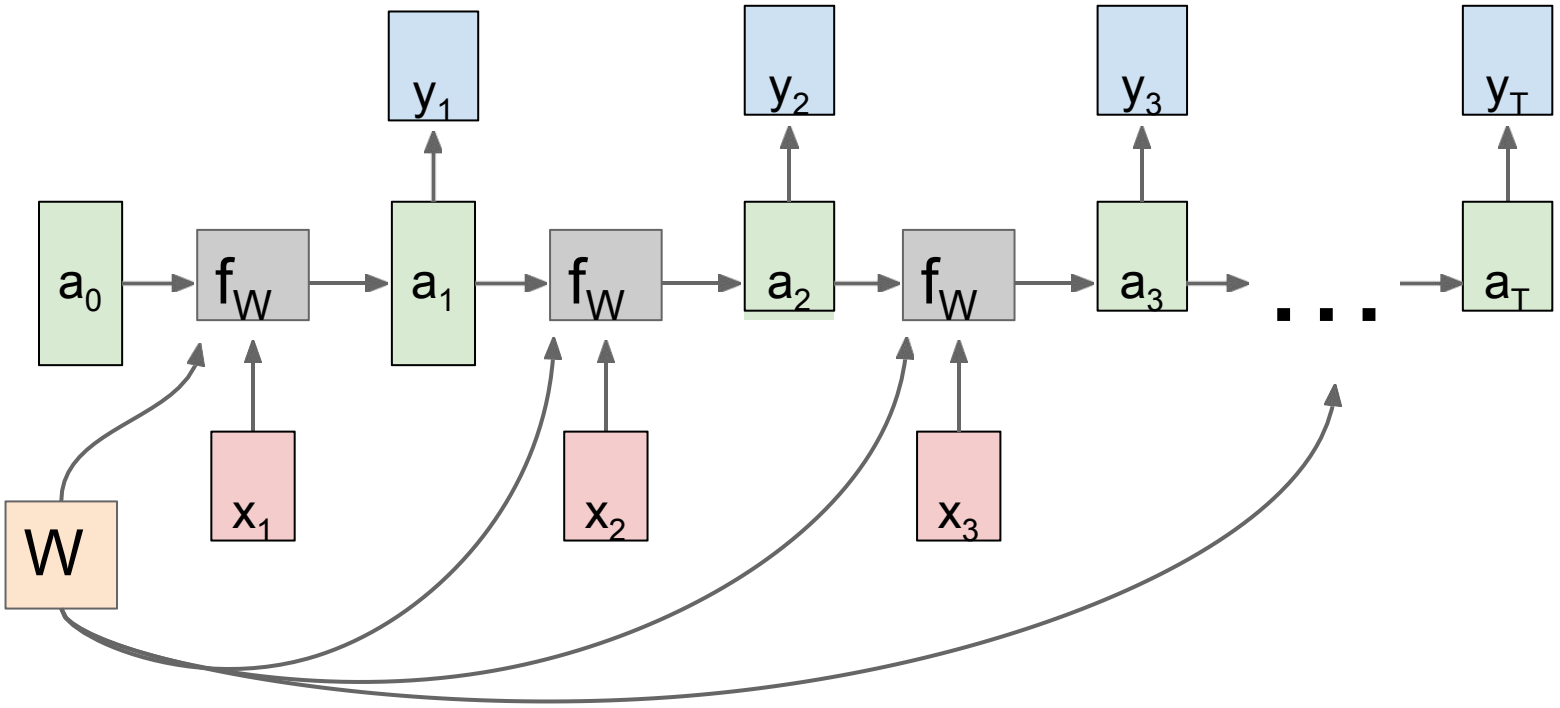


RNN: Computational Graph

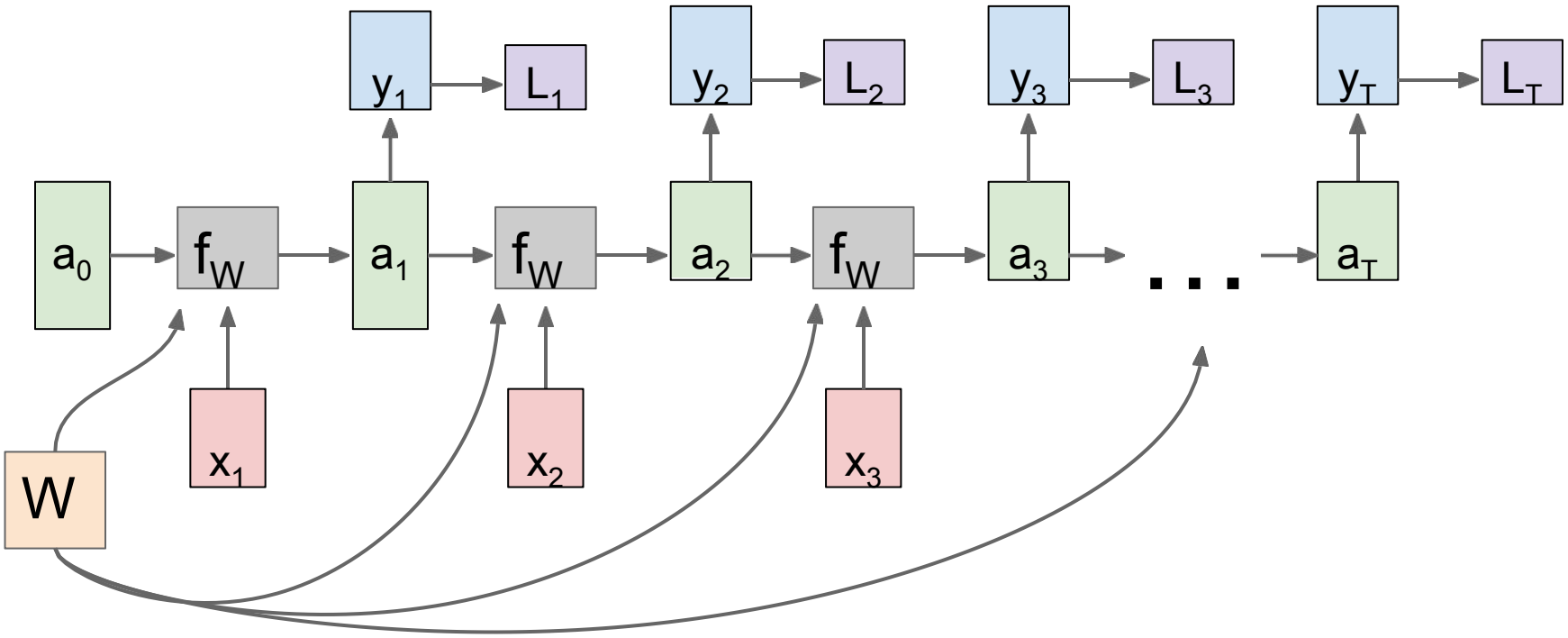
Re-use the same weight matrix at every time-step



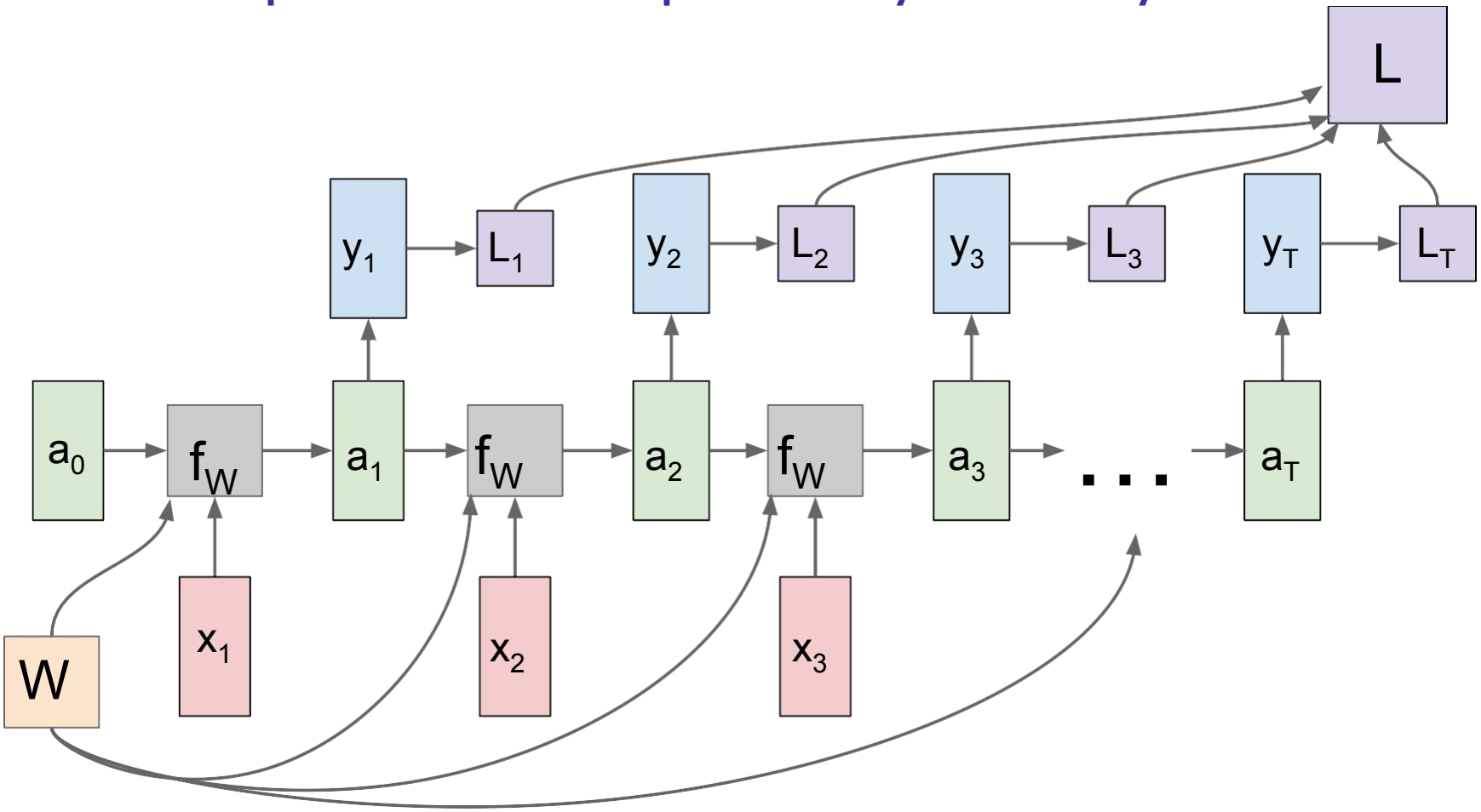
RNN: Computational Graph: Many to Many



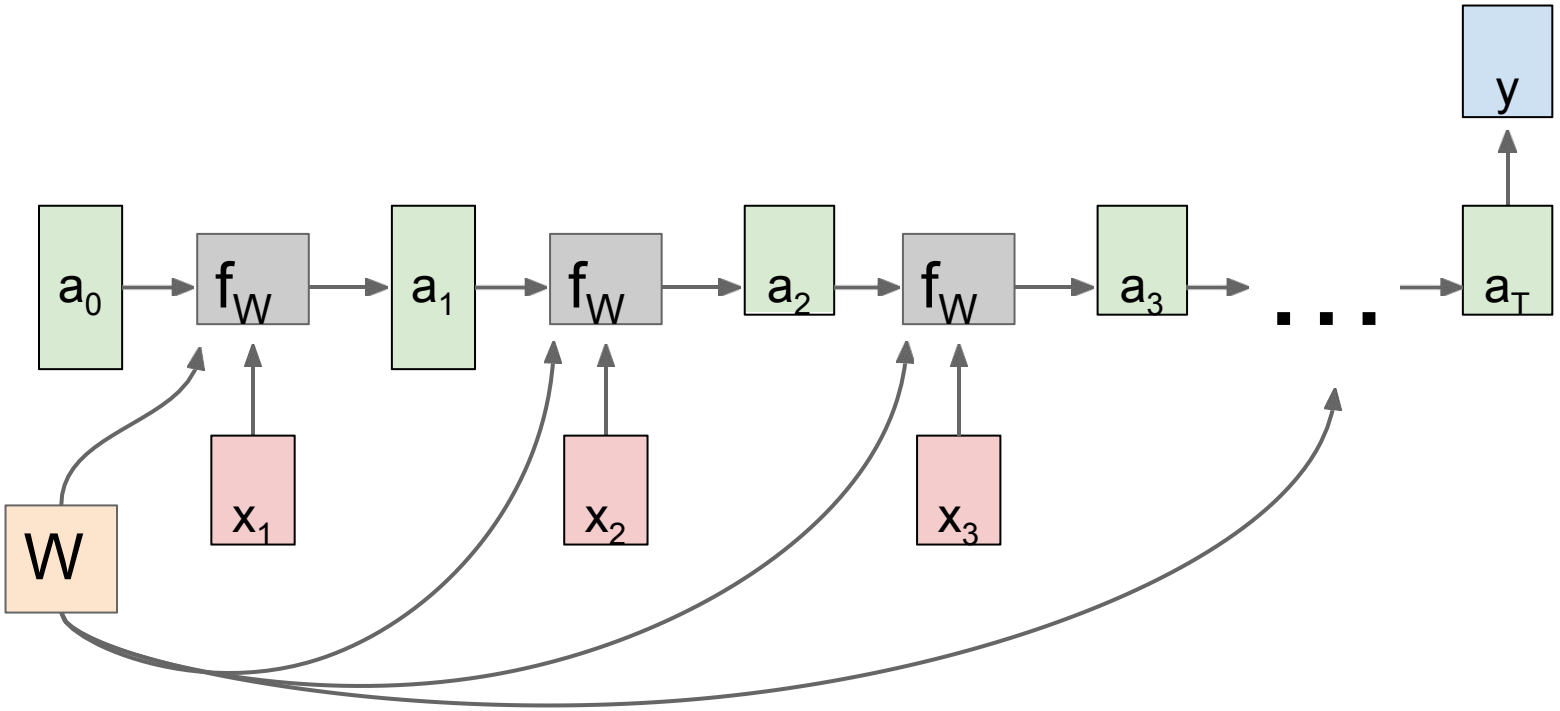
RNN: Computational Graph: Many to Many



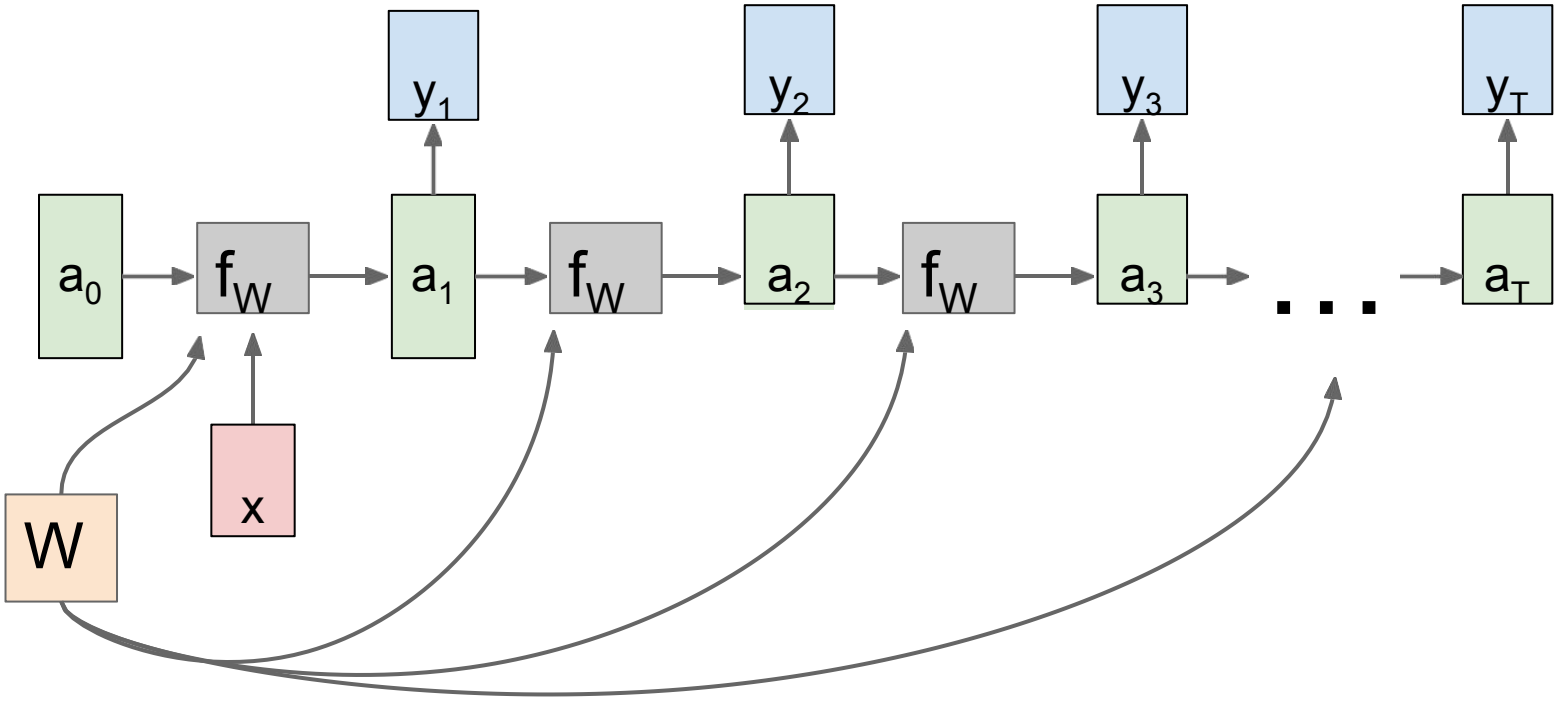
RNN: Computational Graph: Many to Many



RNN: Computational Graph: Many to One



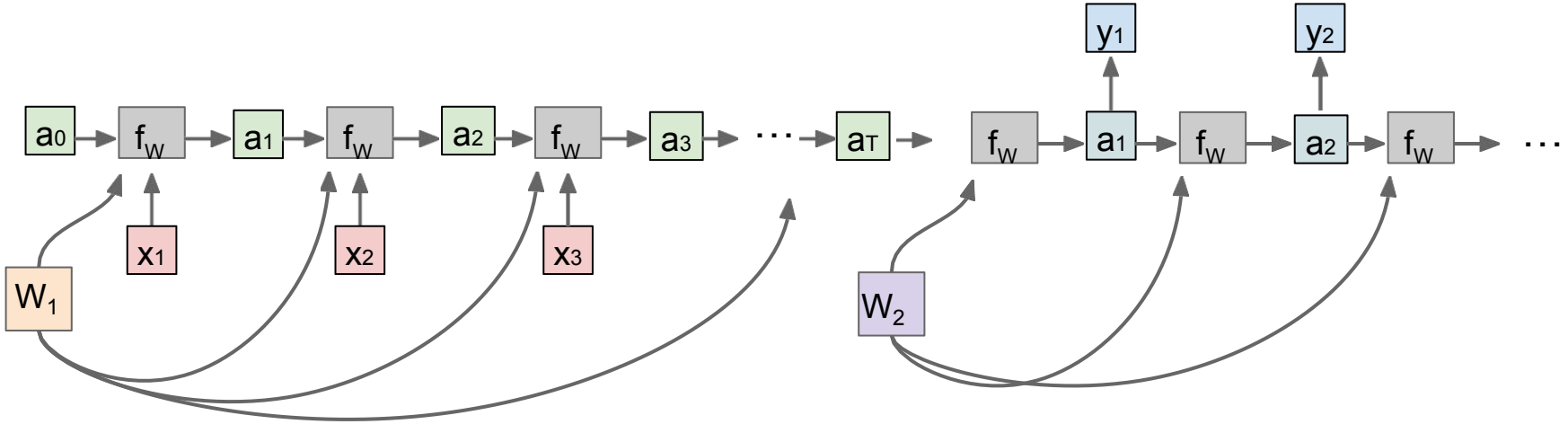
RNN: Computational Graph: One to Many



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

Many to one: Encode inputsequence in a single vector

One to many: Produce output sequence from single input vector

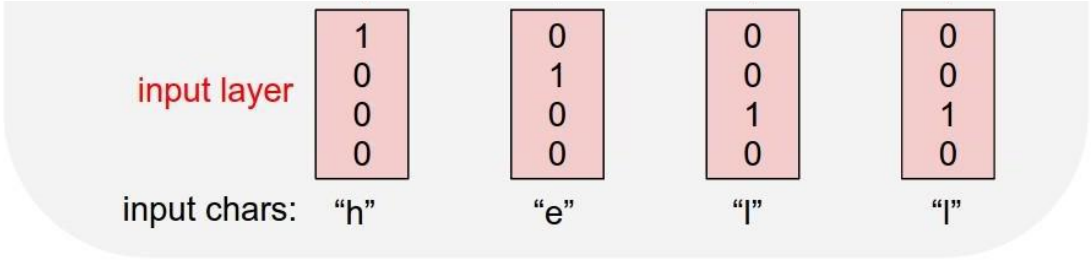


Sutskever et al, “Sequence to Sequence Learning with Neural Networks”, NIPS 2014

Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

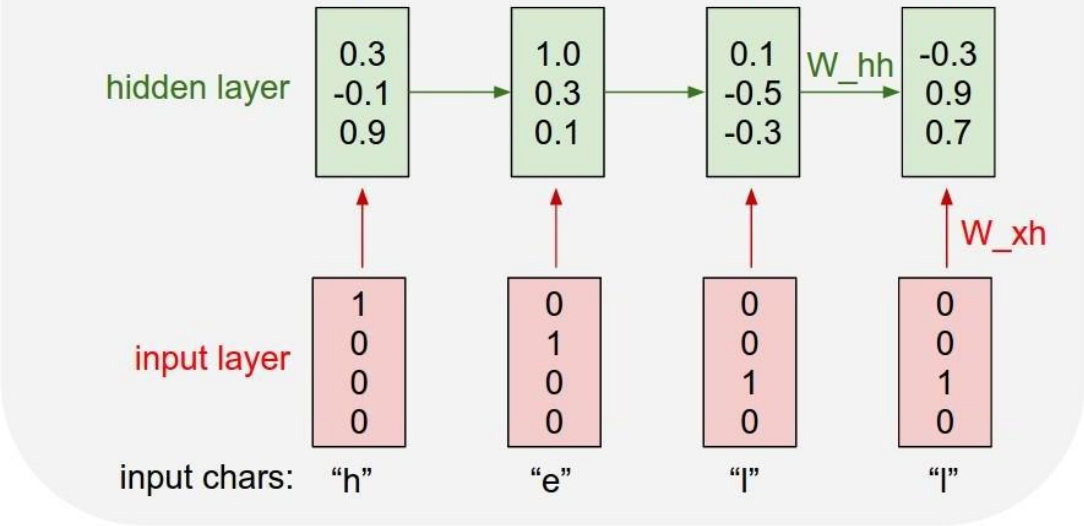


Example:
Character-level
Language Model

Vocabulary:
[h,e,l,o]

Example training sequence:
“hello”

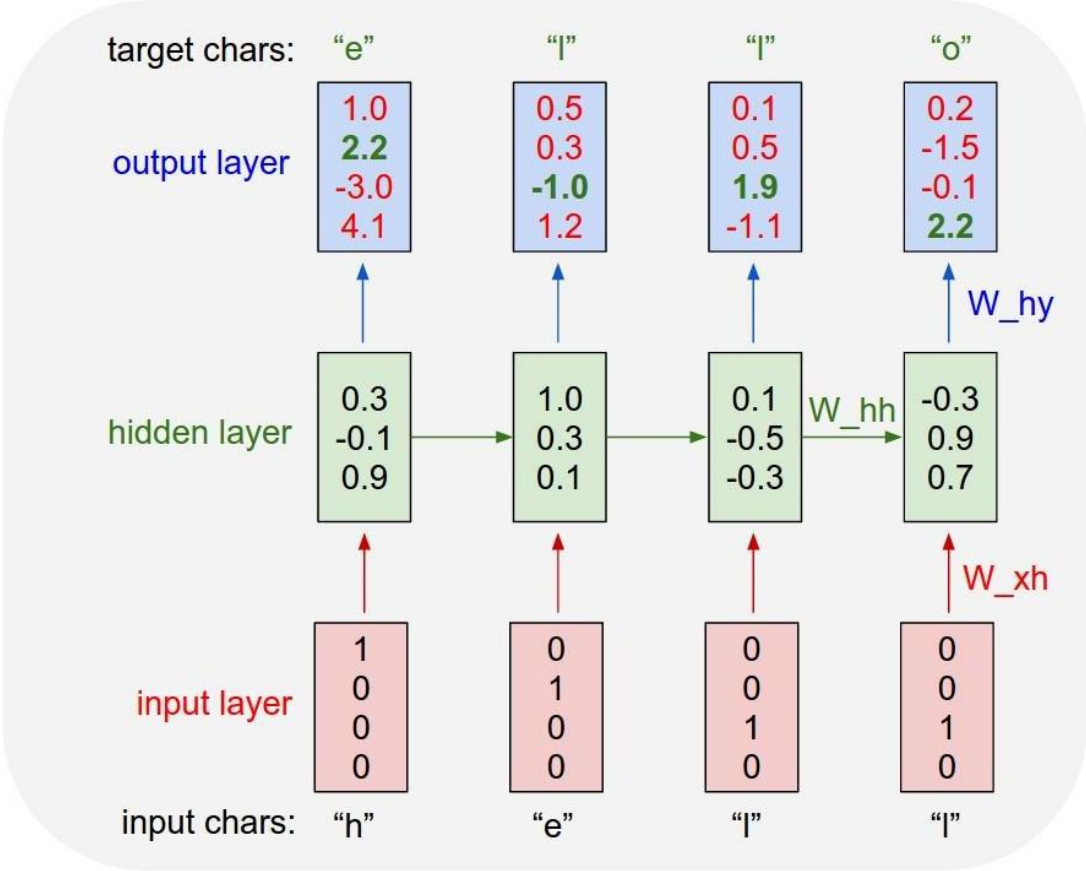
$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$



Example: Character-level Language Model

Vocabulary:
[h,e,l,o]

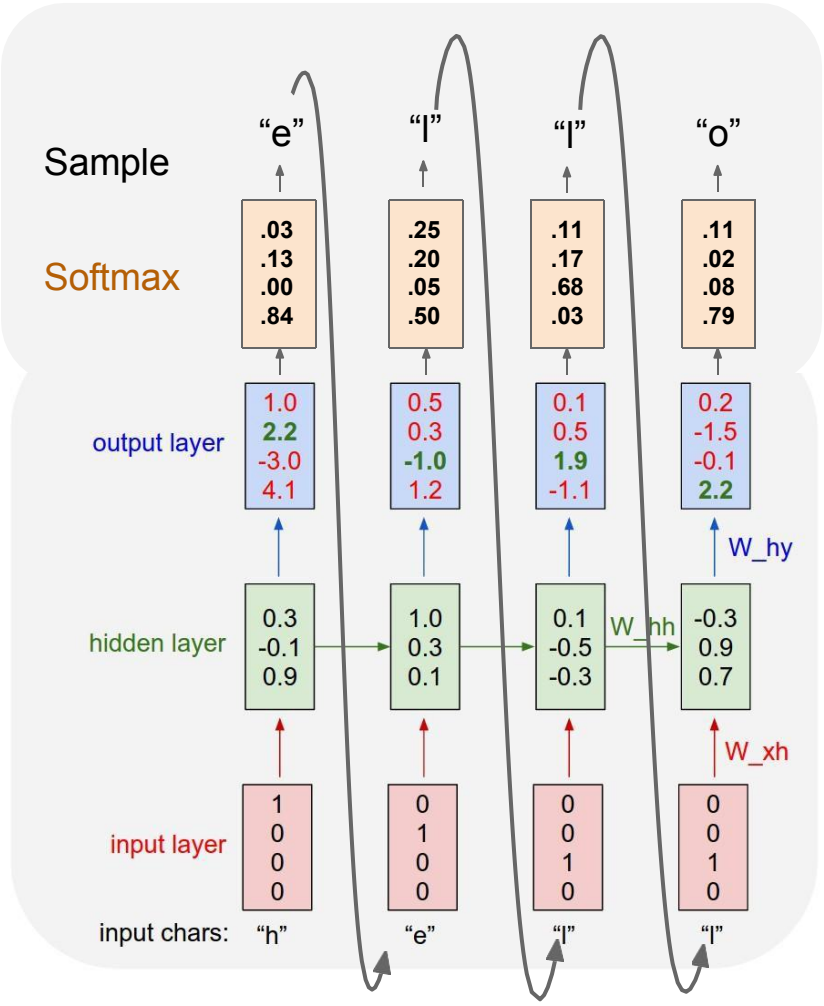
Example training sequence:
“hello”



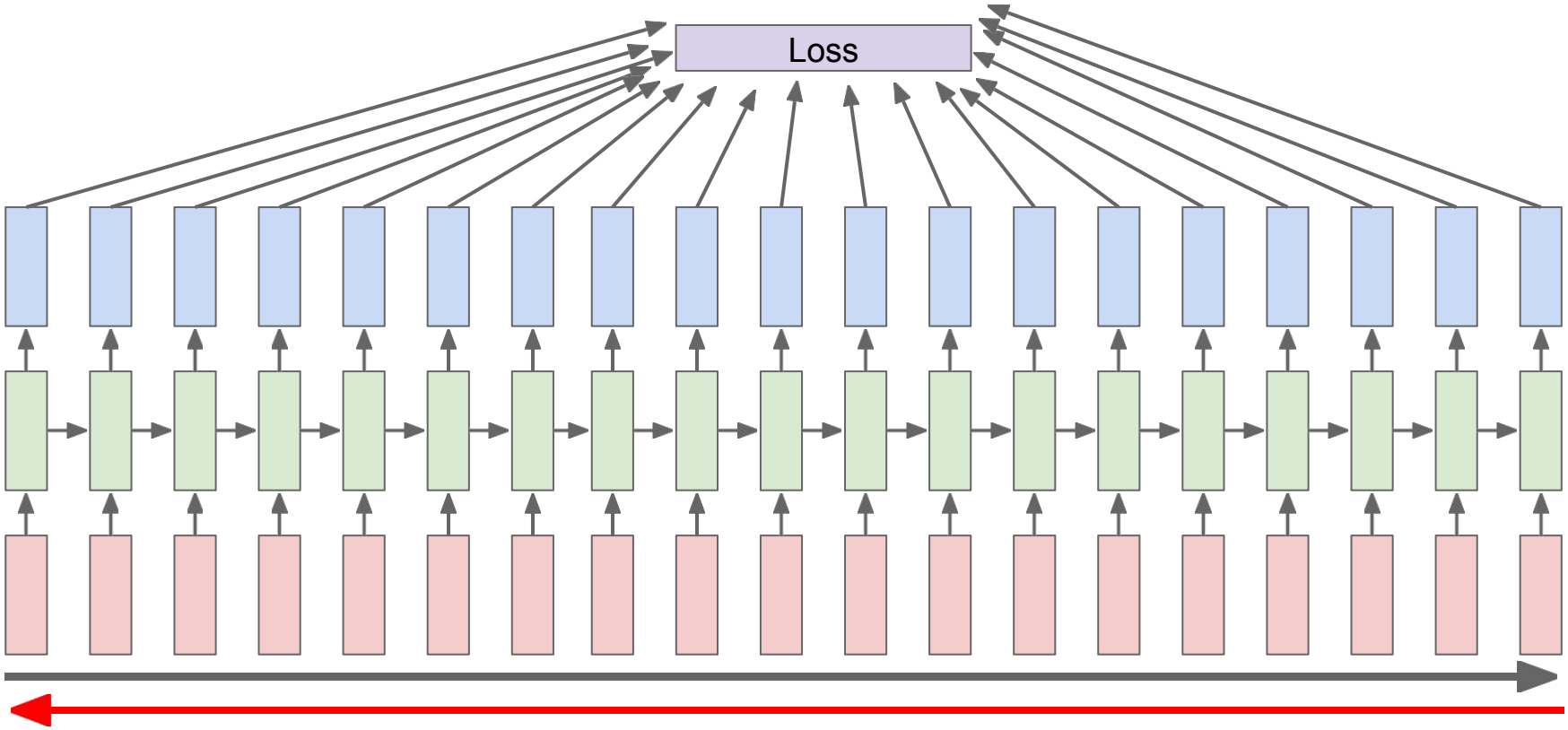
Example: Character-level Language Model Sampling

Vocabulary: [h,e,l,o]

At test-time sample characters one at a time, feed back to model

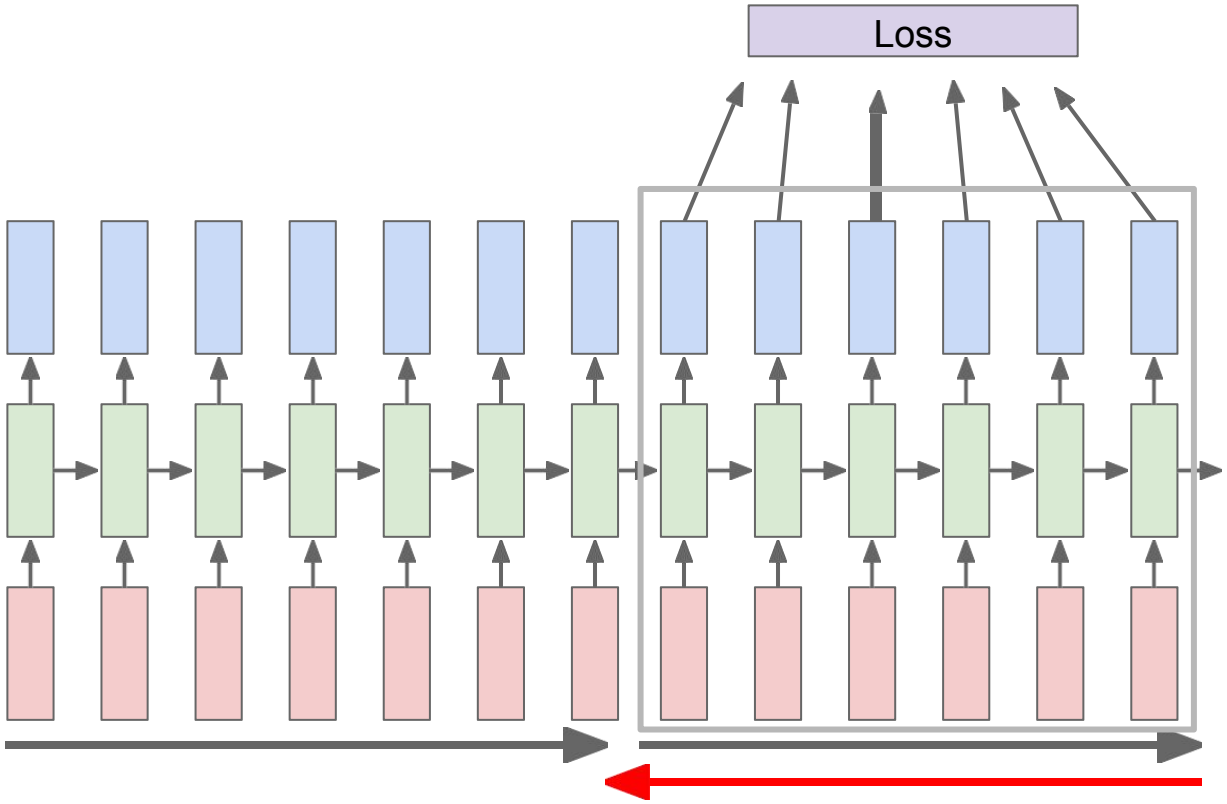


Backpropagation through time



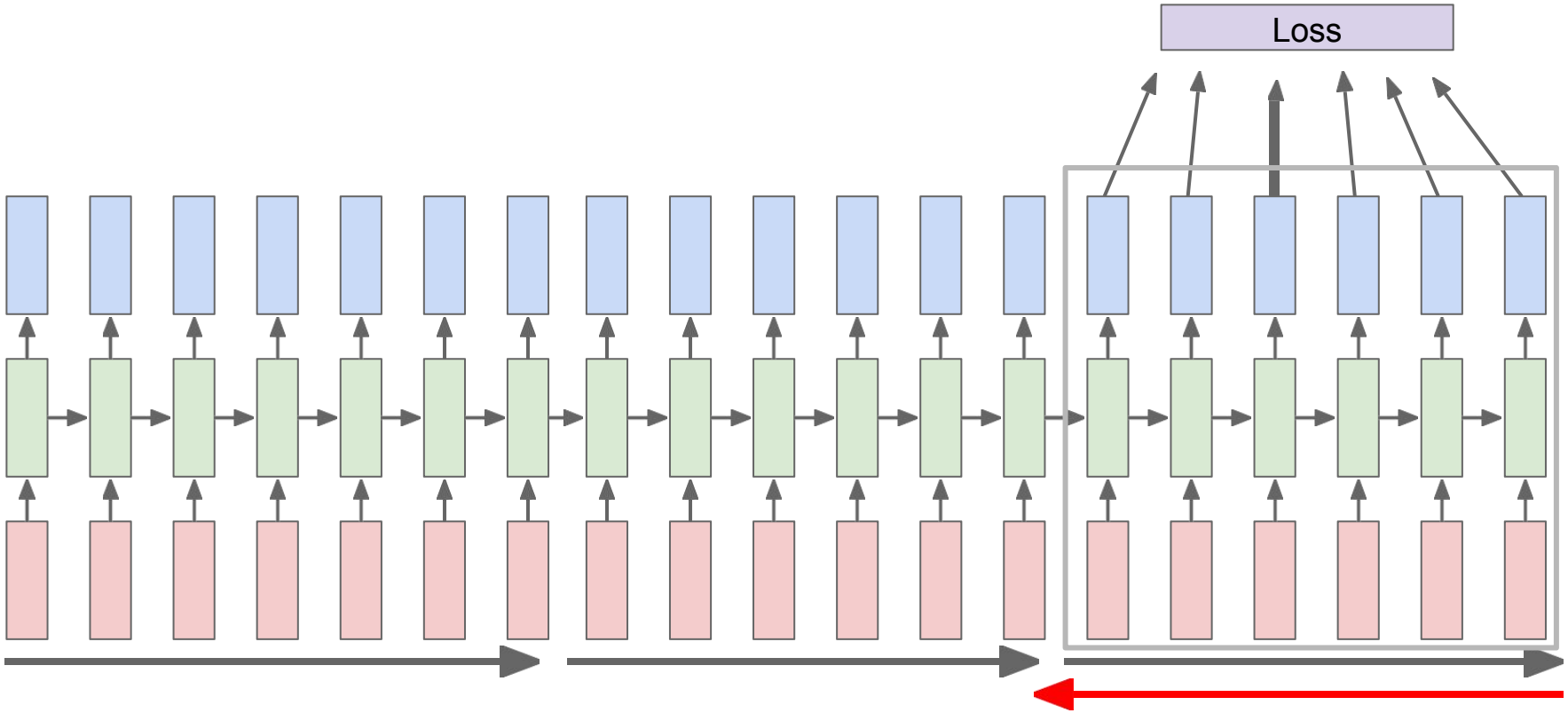
Forward through entire sequence to compute loss, then backward through entire sequence to compute gradient

Truncated Backpropagation through time



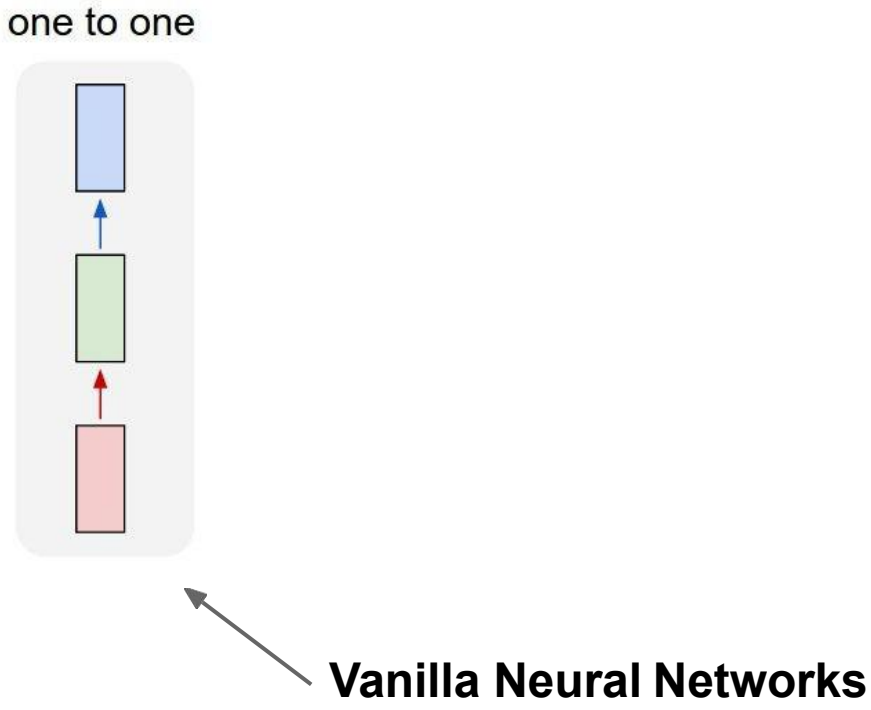
Carry hidden states forward in time forever, but only backpropagate for some smaller number of steps

Truncated Backpropagation through time

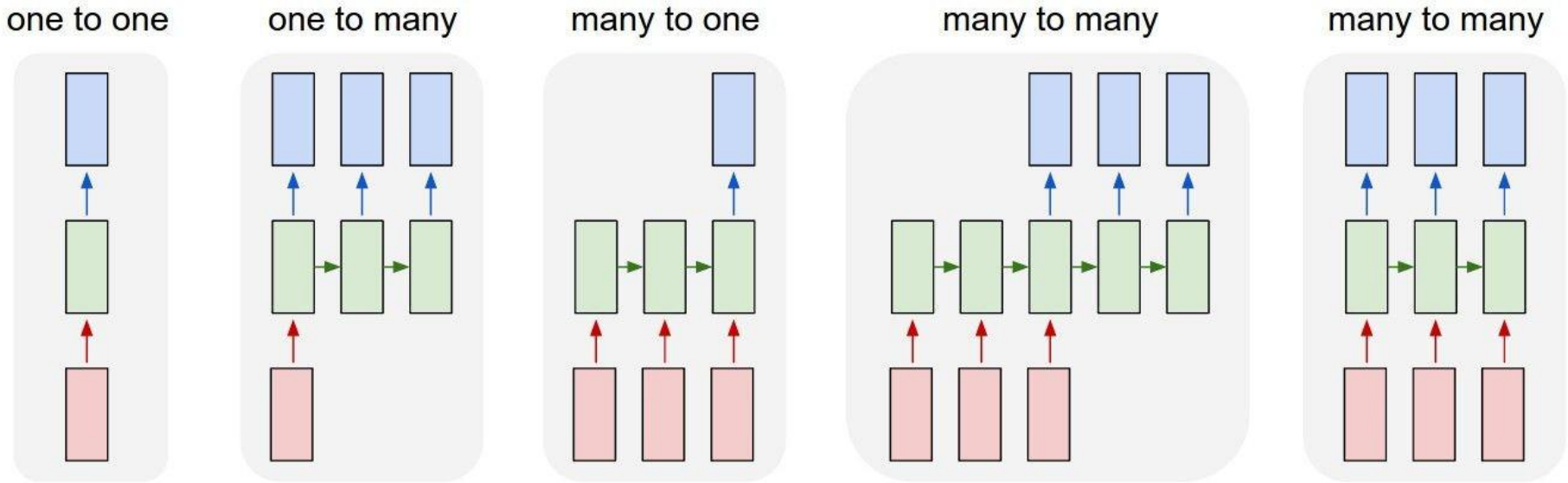


I want a glass of orange juice
I think he wants a glass of apple juice

“Vanilla” Neural Network

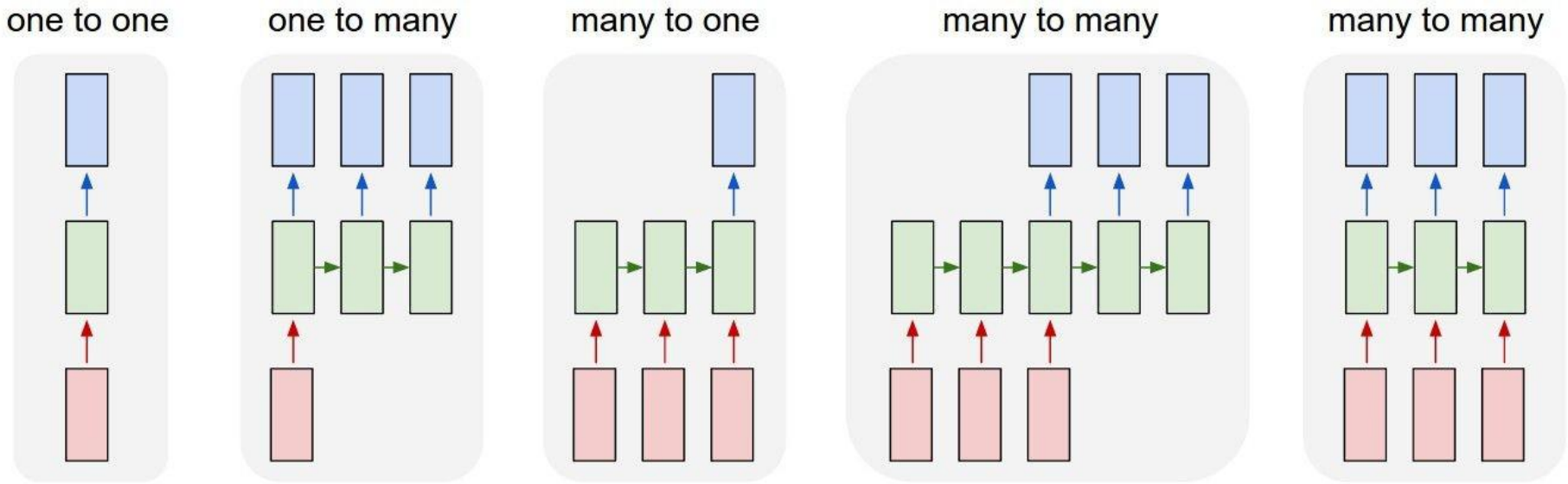


Recurrent Neural Networks: Process Sequences



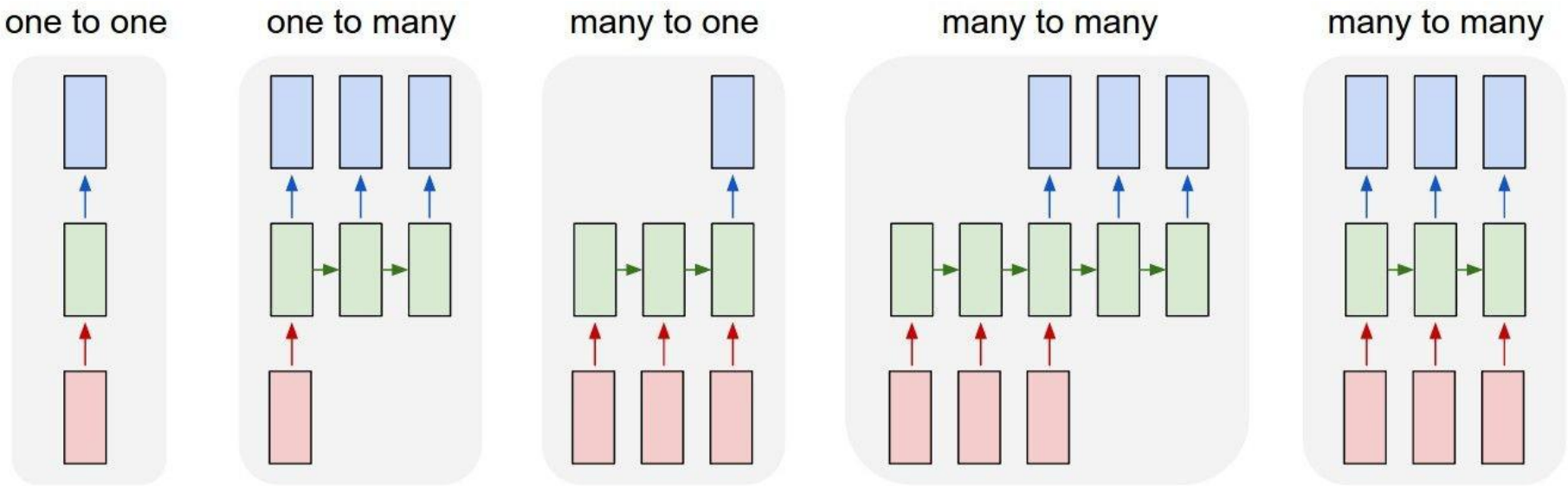
e.g. **Image Captioning**
image -> sequence of words

Recurrent Neural Networks: Process Sequences



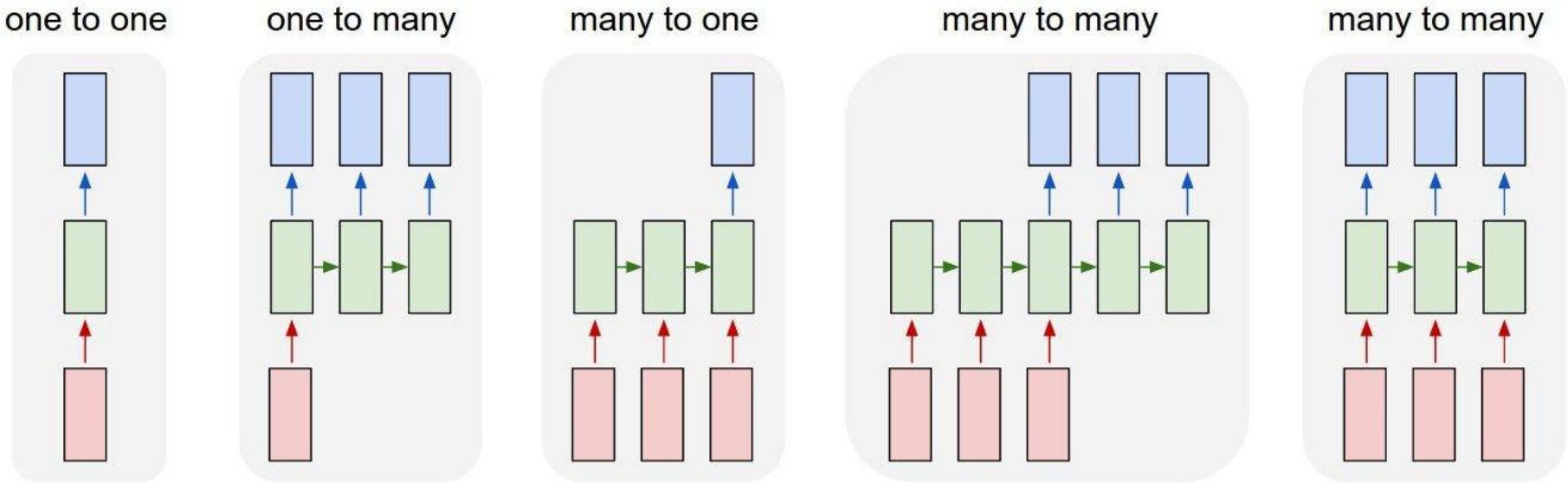
e.g. **Sentiment Classification**
sequence of words -> sentiment

Recurrent Neural Networks: Process Sequences



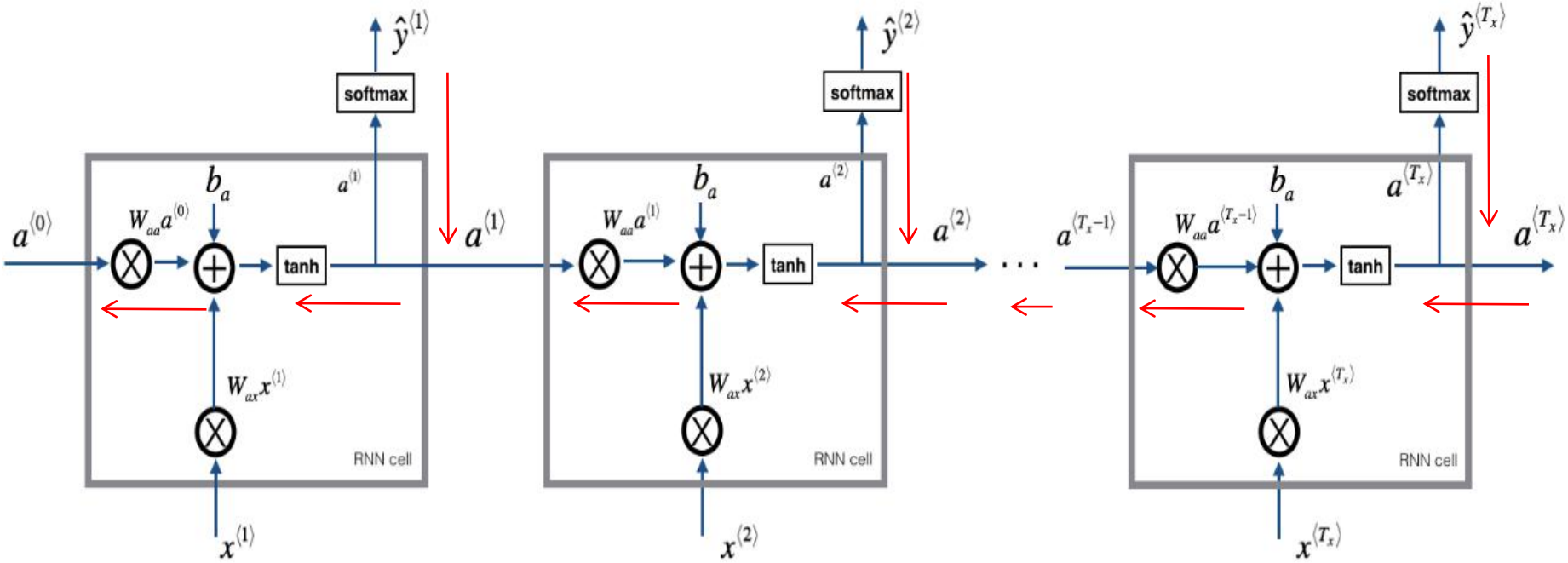
e.g. **Machine Translation**
seq of words -> seq of words

Recurrent Neural Networks: Process Sequences



e.g. **Video classification on frame level**

基本RNN的梯度消失



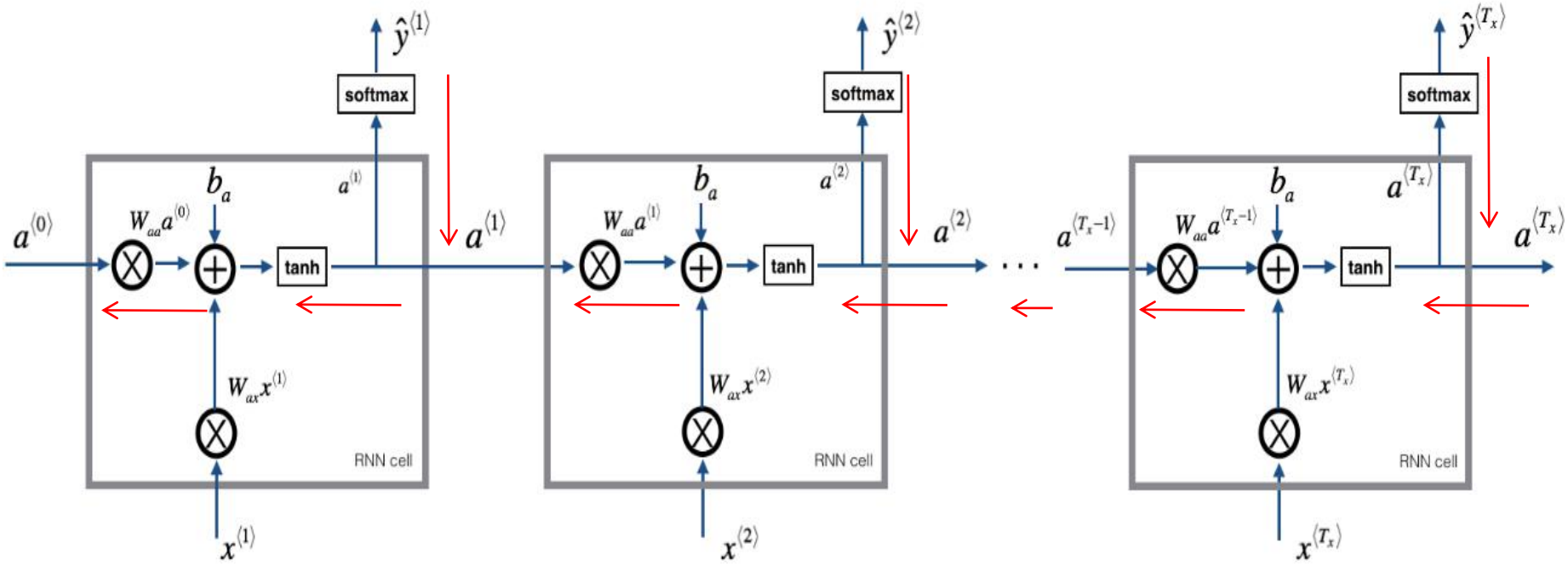
Computing gradient of a_0 involves many factors of W (and repeated tanh)

"The cat, which already ate ……, was full."

"The cats, which ate ……, were full."

对于基本的RNN网络存在梯度消失和梯度爆炸问题

基本RNN的梯度消失

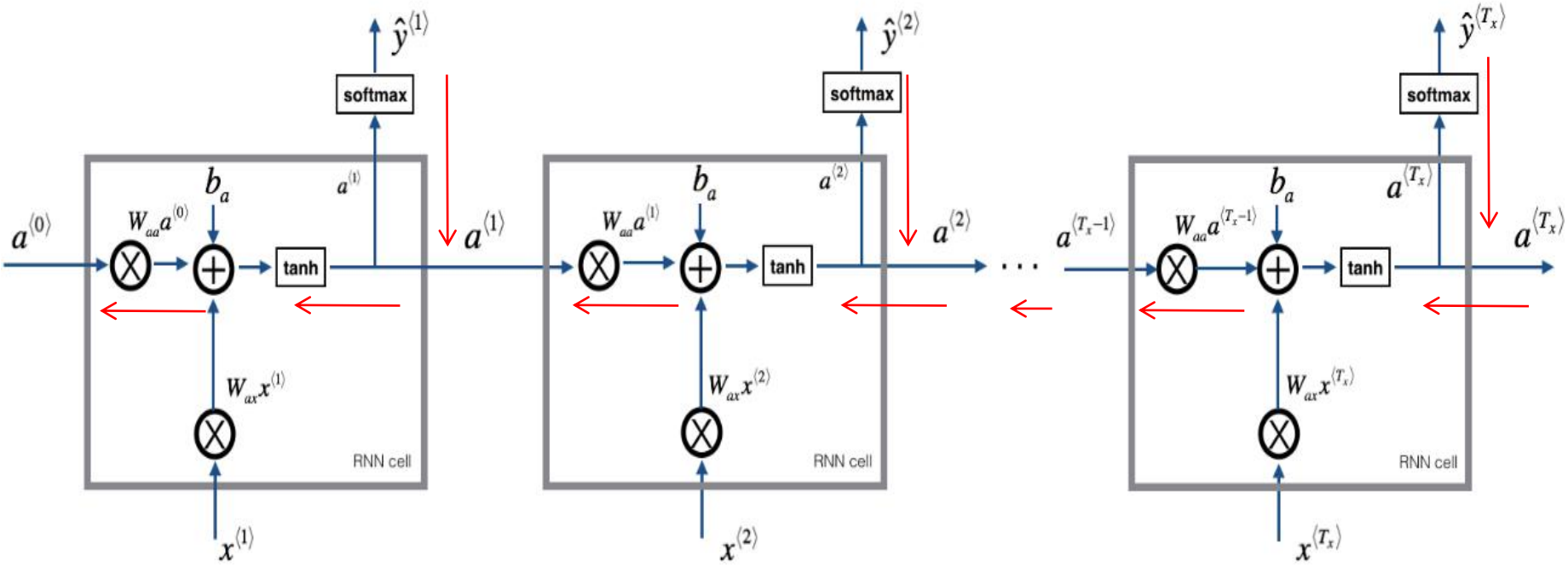


Computing gradient of a_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
呈指数增长，出现梯度爆炸

Largest singular value < 1:
呈指数递减，出现梯度消失

基本RNN的梯度消失



Computing gradient of a_0 involves many factors of W (and repeated tanh)

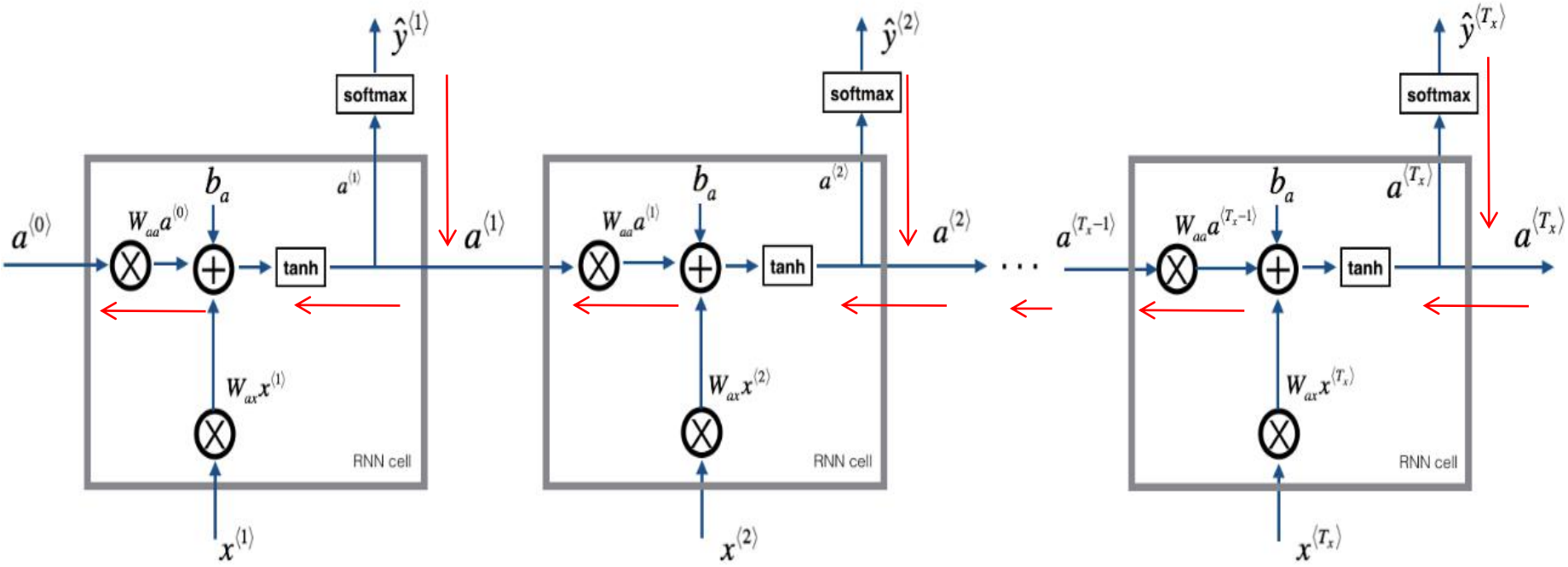
Largest singular value > 1:
呈指数增长，出现梯度爆炸

→ 梯度裁剪: Scale gradient if its norm is too big

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

Largest singular value < 1:
呈指数递减，出现梯度消失

基本RNN的梯度消失



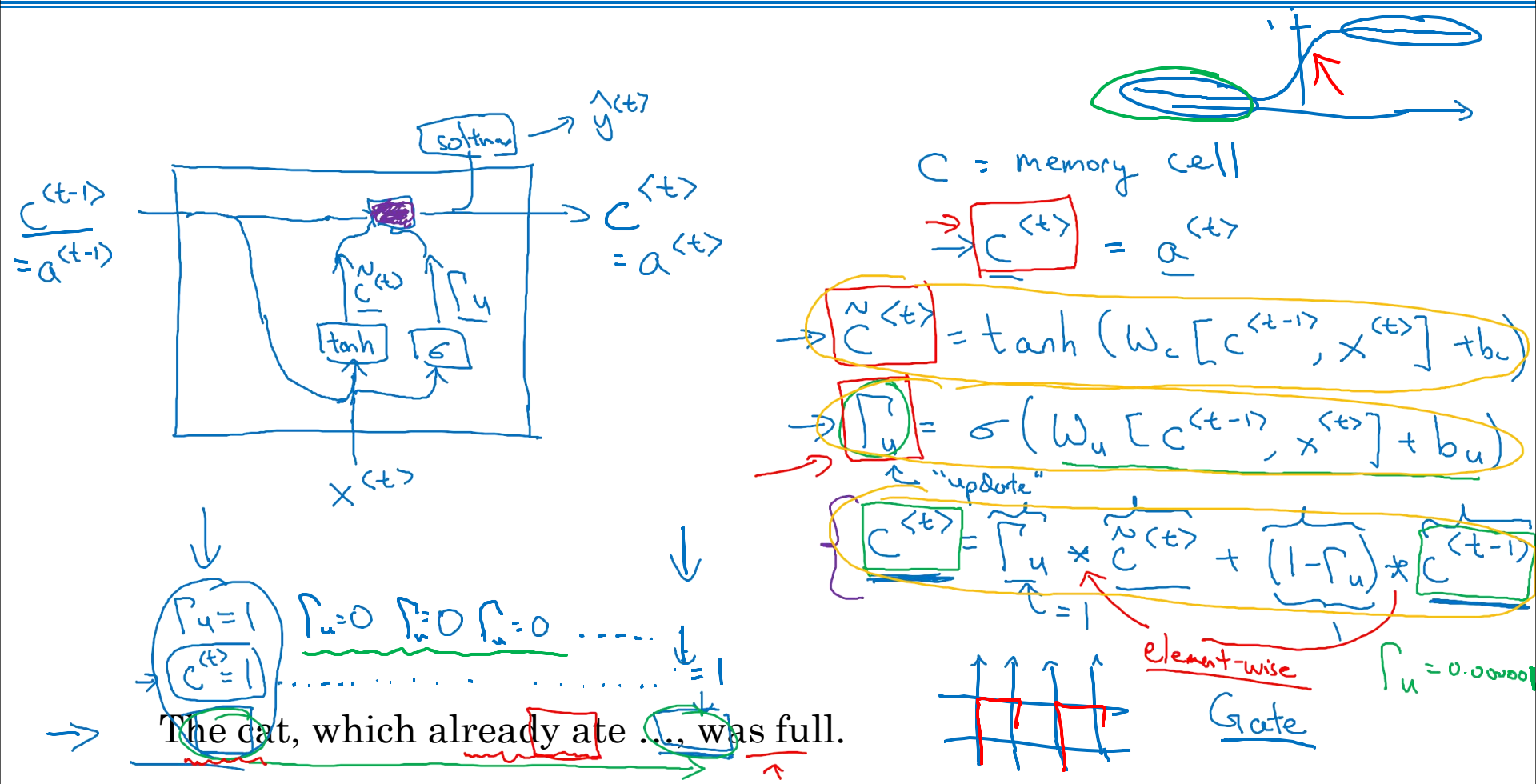
Computing gradient of a_0 involves many factors of W (and repeated tanh)

Largest singular value > 1:
呈指数增长，出现梯度爆炸

Largest singular value < 1:
呈指数递减，出现梯度消失

→ 改变基本RNN结构

门控循环单元GRU (simplified)



[Cho et al., 2014. On the properties of neural machine translation: Encoder-decoder approaches]
[Chung et al., 2014. Empirical Evaluation of Gated Recurrent Neural Networks on Sequence Modeling]

完整的GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$ 更新门

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$ 相关门

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

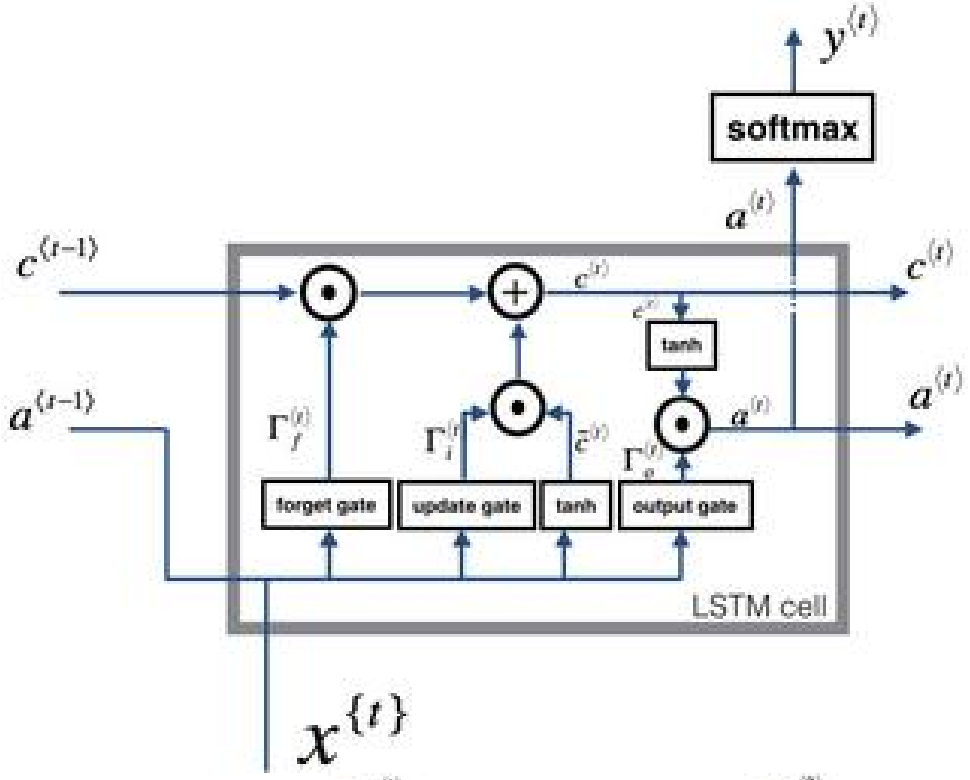
$$a^{<t>} = c^{<t>}$$

LSTM

The cat, which ate already, was full.

Andrew Ng

LSTM示意图



遗忘门 $\Gamma_f^{\langle t \rangle} = \sigma(W_f[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_f)$

更新门 $\Gamma_u^{\langle t \rangle} = \sigma(W_u[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_u)$

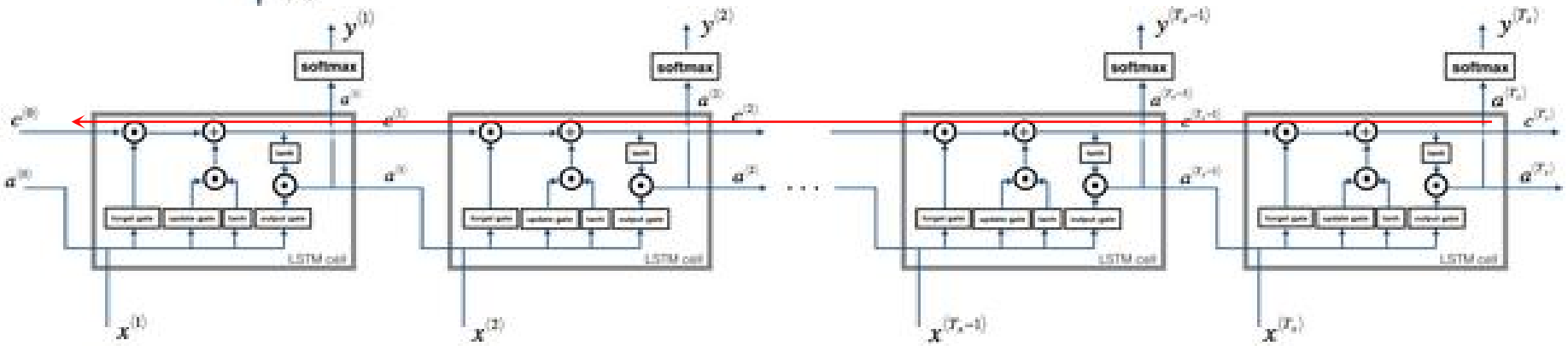
$\tilde{c}^{\langle t \rangle} = \tanh(W_c[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_c)$

$\cdots$

$c^{\langle t \rangle} = \Gamma_f^{\langle t \rangle} \circ c^{\langle t-1 \rangle} + \Gamma_u^{\langle t \rangle} \circ \tilde{c}^{\langle t \rangle}$

输出门 $\Gamma_o^{\langle t \rangle} = \sigma(W_o[a^{\langle t-1 \rangle}, x^{\langle t \rangle}] + b_o)$

$a^{\langle t \rangle} = \Gamma_o^{\langle t \rangle} \circ \tanh(c^{\langle t \rangle})$



GRU vs LSTM

GRU

$$\tilde{c}^{<t>} = \tanh(W_c[\Gamma_r * c^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[c^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_r = \sigma(W_r[c^{<t-1>}, x^{<t>}] + b_r)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + (1 - \Gamma_u) * c^{<t-1>}$$

$$a^{<t>} = c^{<t>}$$

LSTM

$$\tilde{c}^{<t>} = \tanh(W_c[a^{<t-1>}, x^{<t>}] + b_c)$$

$$\Gamma_u = \sigma(W_u[a^{<t-1>}, x^{<t>}] + b_u)$$

$$\Gamma_f = \sigma(W_f[a^{<t-1>}, x^{<t>}] + b_f)$$

$$\Gamma_o = \sigma(W_o[a^{<t-1>}, x^{<t>}] + b_o)$$

$$c^{<t>} = \Gamma_u * \tilde{c}^{<t>} + \Gamma_f * c^{<t-1>}$$

$$a^{<t>} = \Gamma_o * c^{<t>}$$

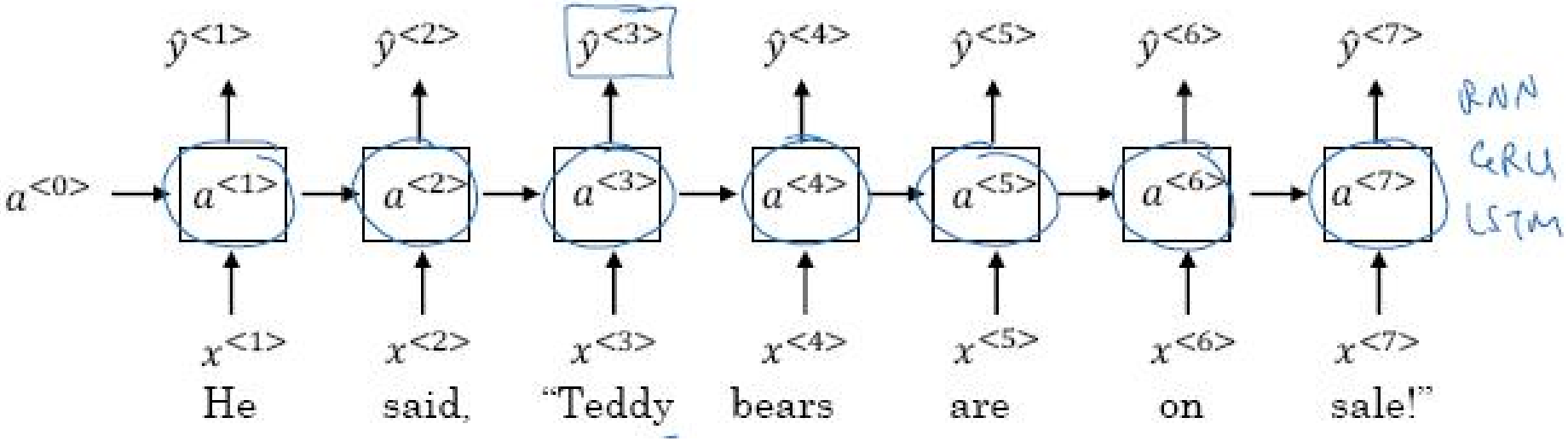
- GRU更简单，只有两个门，计算性更快，更易构建更大网络
- LSTM有三个门，更强大灵活，大部分会把LSTM作为默认选择尝试
- 选择哪一个没有统一准则

双向RNN (BRNN)

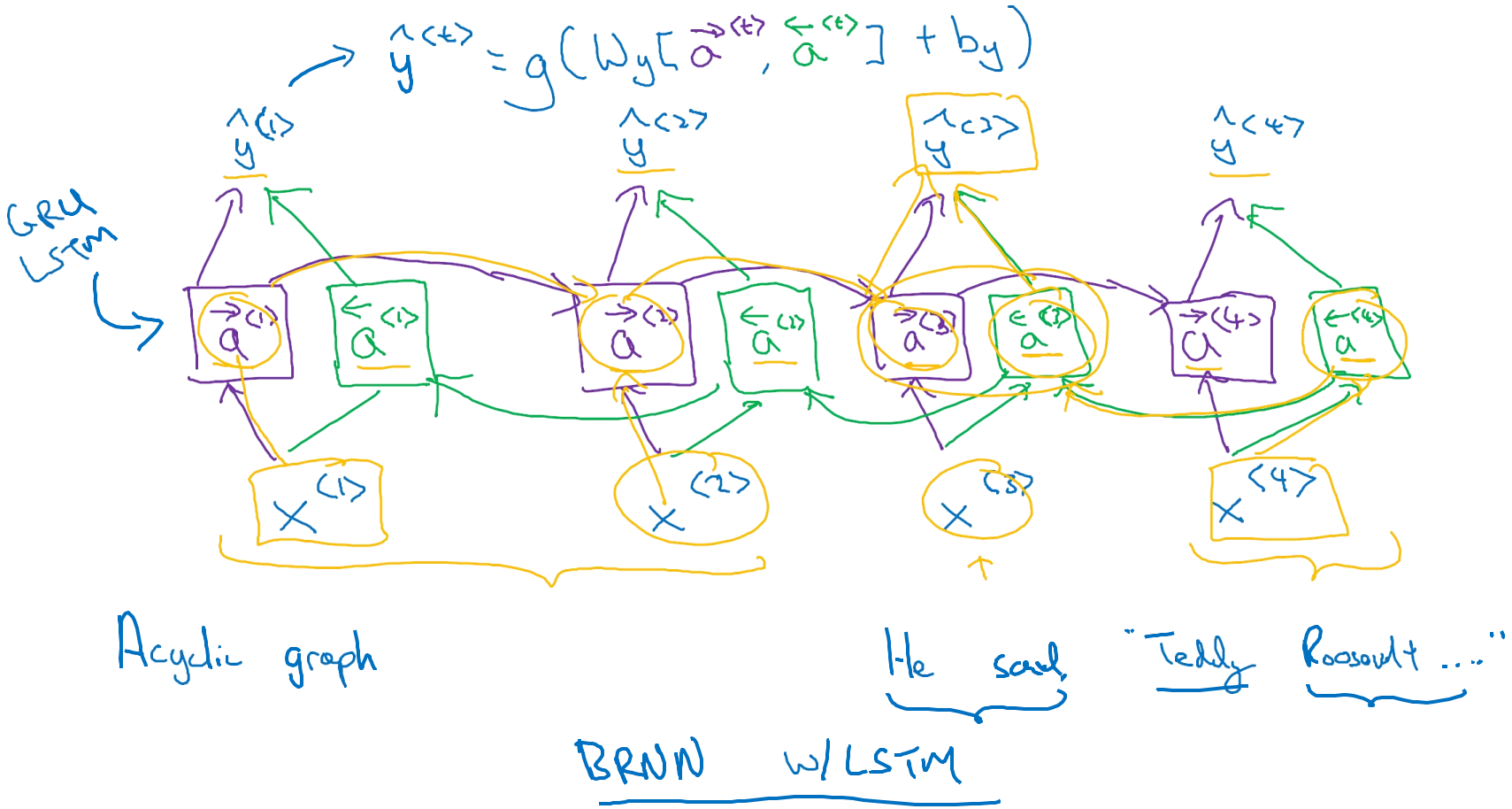
Getting information from the future

He said, "Teddy bears are on sale!"

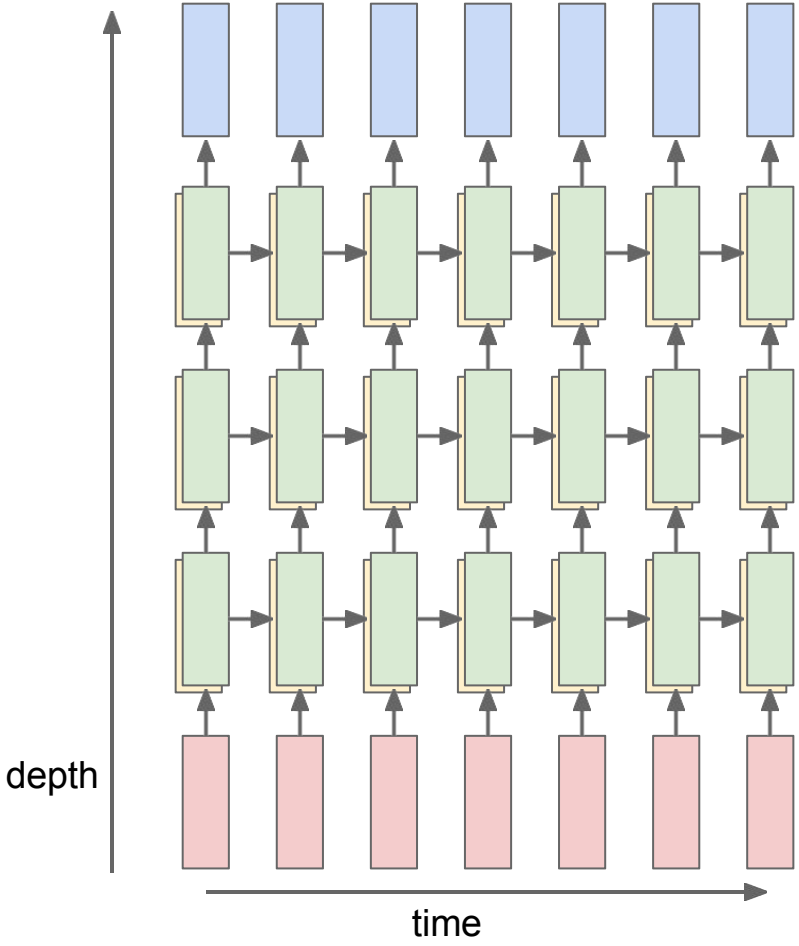
He said, "Teddy Roosevelt was a great President!"



双向RNN (BRNN)


$\hat{y}^{<t>} = g(W_y[\overrightarrow{a}^{<t>}, \overleftarrow{a}^{<t>}] + b_y)$
GRU
LSTM
Acyclic graph
He said, "Teddy Roosevelt ..."
BRNN w/LSTM

Deep RNN example



应用

什么是情感分析

情感分析是对人、物以及事件的态度倾向的分析

情感分析根据复杂程度可以分为三种：

- 简单的任务，把文本分成正、负、中三种
- 稍微复杂一点的任务，考虑情感的强弱程度，比如把文本的情感进行评分，1-5分，分越高情感越偏正面
- 更复杂的任务，细粒度的情感分析，考虑特定目标的不同程度的情感，分析情感分析的目标

◆Positive: 我爱北京天安门

◆Negative: 政府采取的这个政策很不合理啊

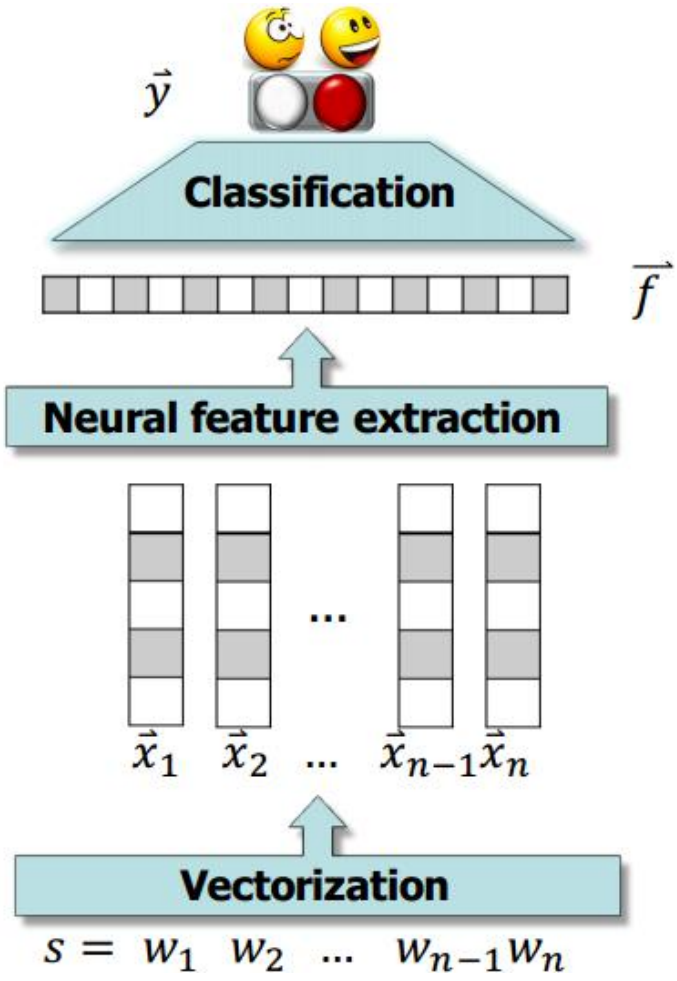
◆Neutral: 北京是中国首都

应用

基于深度学习的情感分析

- ◆分类问题
- ◆ Classification layer
- ◆Input: 包含n个 词的句子
- ◆Output: 情感标签

$$\vec{y} = softmax(\boldsymbol{W}_o\vec{f} + \vec{b}_o)$$



应用

中文古诗词自动生成

啤酒 Beer	冰心 Xin Bing
今宵啤酒两三缸， I drink glasses of beer tonight,	一片冰心向月明， I open up my pure heart to the moon,
杯底香醇琥珀光。 With the bottom of the glass full of aroma and amber light.	千山春水共潮生。 With the spring river flowing past mountains.
清爽金风凉透骨， Feeling cold as the autumn wind blows,	繁星闪烁天涯路， Although my future is illuminated by stars,
醉看明月挂西窗。 I get drunk and enjoy the moon in sight by the west window.	往事萦怀梦里行。 The past still lingers in my dream.

请填入内容<最多八个字>：北京航空航天大学 再来一首 重置

字数：◉五言 ○七言

藏的位置：◉藏头 ○藏尾 ○藏中 ○递增 ○递减

押韵方式：◉双句一压 ○双句押韵 ○一三四押

北岸掩柴关， 京师阳春晚。 空伴夜泉清。 天半锁重关。 大鹏浮通川， 学剑已应晚。	北岸掩柴关， 京口青山远。 空悲撤瑟晨。 天宝中进士。 大宾威仪肃， 学得中州语。	北场芸蓼罢， 京坻长有积。 空悲逝水流。 天白水如练。 大才生间气， 学书徒弄笔。

应用

对联自动生成

请输入或修改上联：

东风有意绿野舒青

自动生成下联

上联：东风有意绿野舒青

下联：春雨多情蓝天展黑

上联：东风有意绿野舒青

下联：明月存心蓝天展黑

应用

图片描述生成，本质是视觉到语言（Visual-to-Language，即V2L）的问题，解释起来很简单，就是四个字：**看图说话**。就像老师要求小朋友们在看图说话作业中完成的任务一样，我们也希望算法能够**根据图像给出能够描述图像内容的自然语言语句**。

一个女运动员在球场上打网球

一个孩子拿着一把叉子

应用

一群年青男人在踢足球

三只狗在草地上奔跑

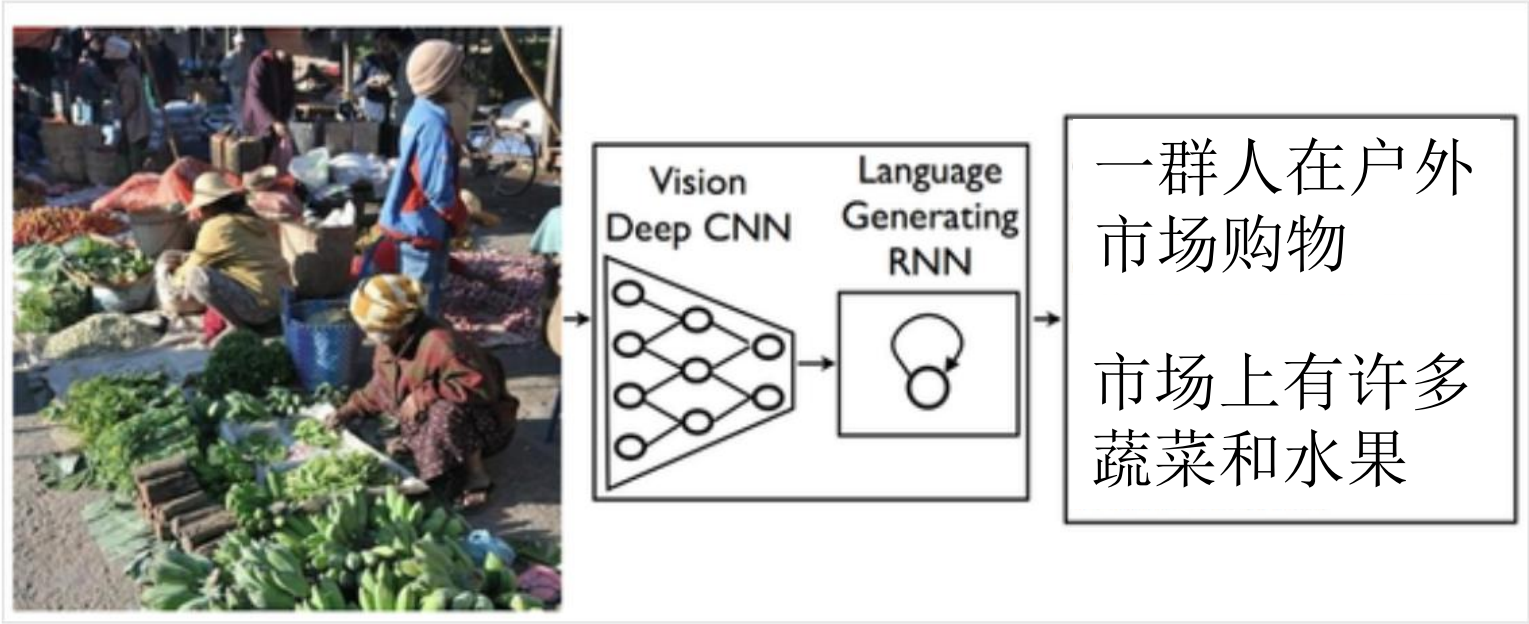
一辆粉红色摩托车停在骑车旁边

一个戴红帽子小孩和另一个小孩在说话

应用

原理

CNN 图像识别 › 语义关联分析 › RNN 语言生成



总结

- RNN在架构设计中具有很大的灵活性
- 基本的RNN很简单，但效果不太好
- 目前使用LSTM或GRU很常见：改善了梯度流
- RNN中的梯度回流可能出现梯度爆炸或梯度消失
- 梯度爆炸可由梯度裁剪控制
- 梯度消失可由LSTM或GRU控制
- 当前研究的热点是构建更好/更简单的网络结构

实验平台说明

- 上机实践平台：
  - **http://10.134.171.240**
  - 用户名：学号
  - 初始密码：**123456**
  - 在“我的课程”进入本课程，第六次实验“RNN实验”
  - 按要求扩展实验，并提交实验报告
  - 实验报告统一发助教邮箱，实验报告命名“第n次实验-学号-姓名”