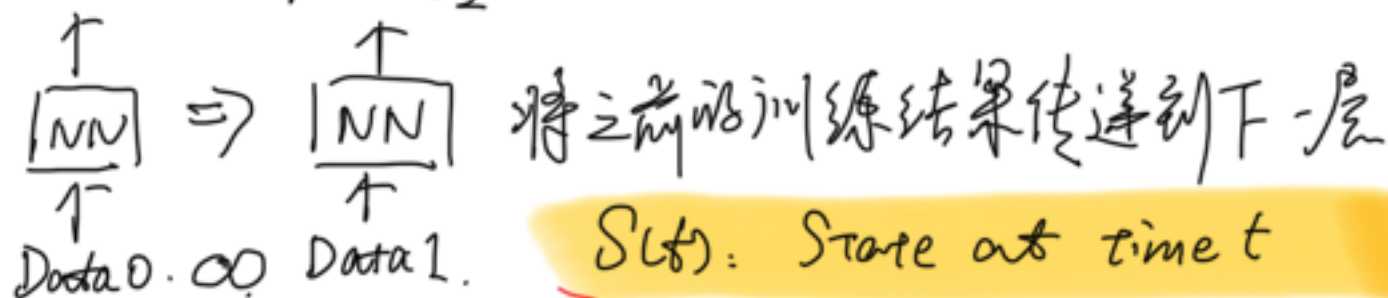


循环神经网络: RNN Recurrent Neural Network.

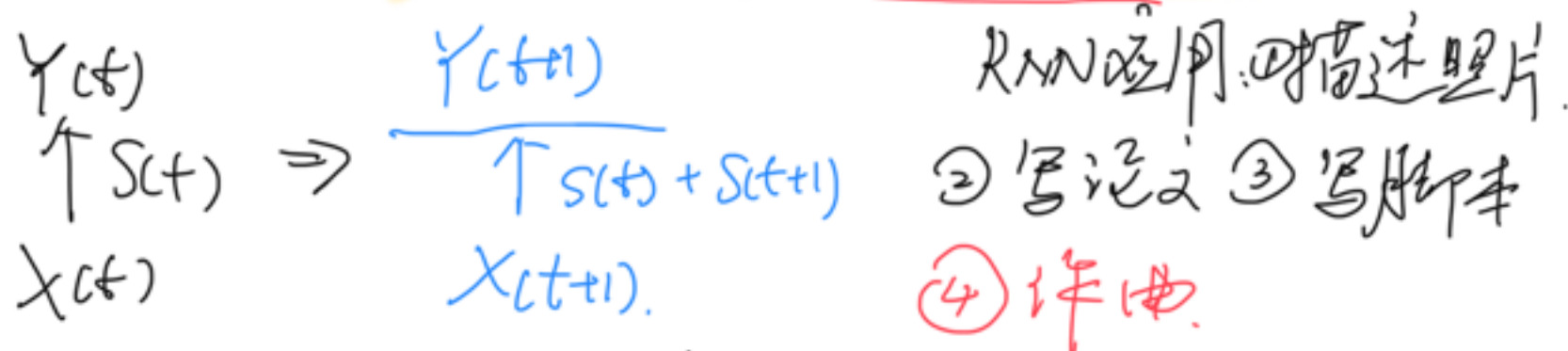
语言分析. Language analysis. 强调顺序在学习中的重要性.

result 0 result 1

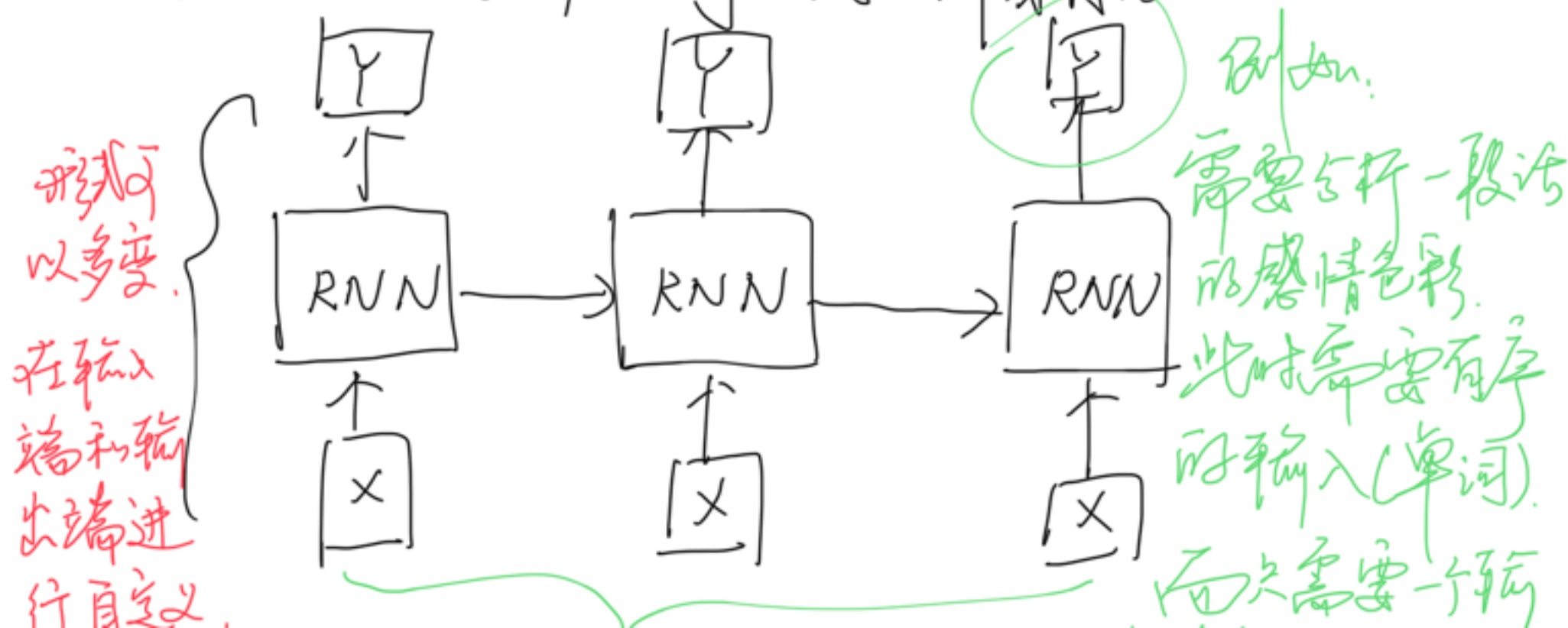


累积之前的记忆

$S(t)$: State at time t $S(t) + S(t+1)$

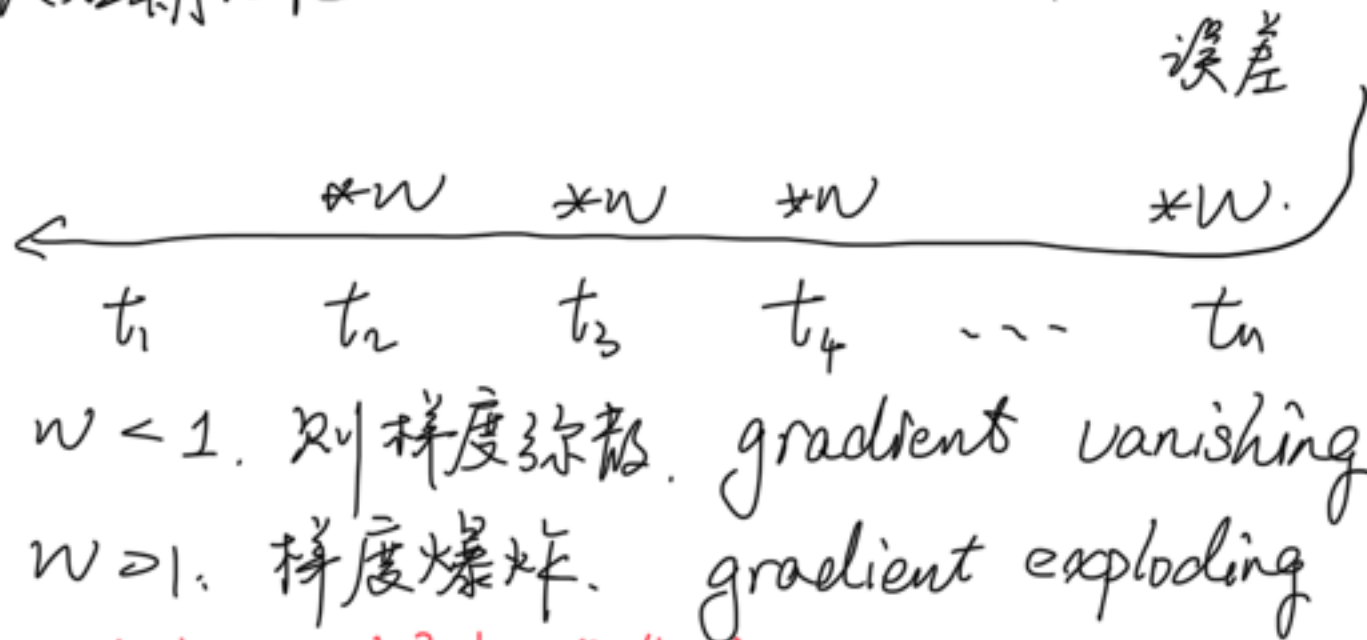
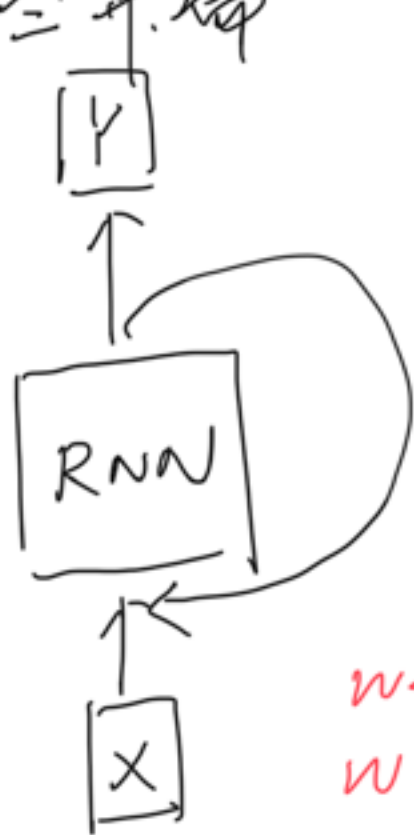


不同形式的RNN可以从事不同的训练目的.



出端: 情绪.

LSTM: Long-short term memory. 用来解决RNN的一些弊端 长短期记忆.



$w < 1$: 在初始时刻, 误差接近 0.
 $w > 1$: RNN 无法回忆久远记忆

three controllers.

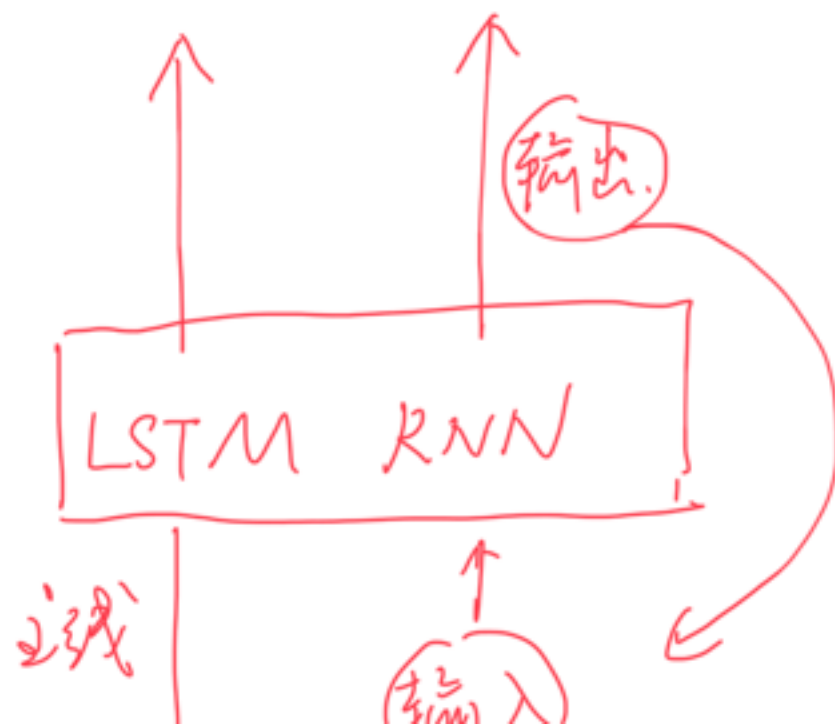
① 输入控制器:

根据输入的重要程度来划分. 然后将其写入主线.



② 忘记控制器:

如果主线剧情改变了之前的想



③ 延缓记忆衰退.

法、则会忘记直线. 故比例皆
换为直线

New tensorflow import:

```
import tensorflow.compat.v1 as tf
```

function rename:

tf.sub \Rightarrow tf.subtract

tf.mul \Rightarrow tf.multiply

激励函数. activation function

Linear v.s. Non linear

$y = Wx \Rightarrow y = AF(Wx)$

^{卷积} CNN: (Convolutional) ^{循环} RNN: (Recurrent)

CNN: relu RNN: relu or tanh.

激励函数并不是随便选的. 用来解决非线性问题.

AF: 非线性化方程.
将线性变为非线性.

① relu

② sigmoid

③ tanh

④ 自定义

激励函数.
但必须可以
微分

在复杂NN下不可.

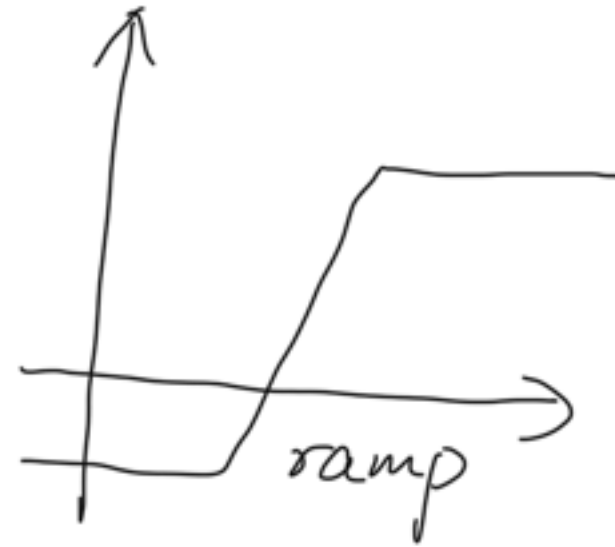
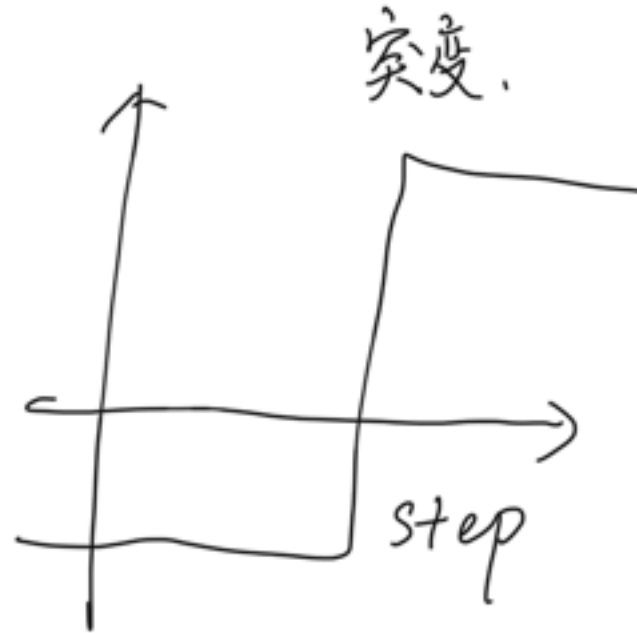
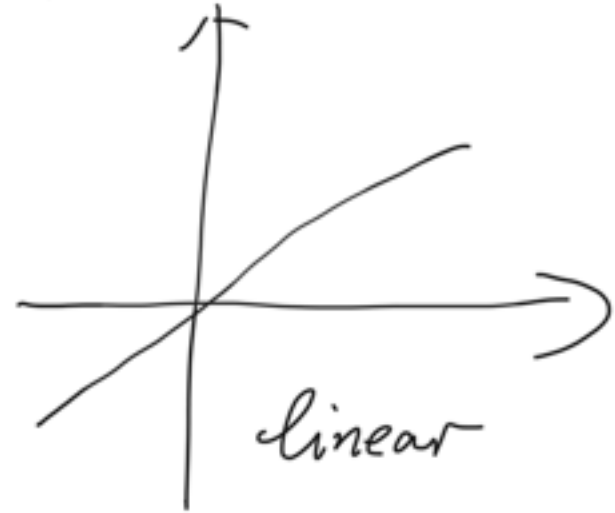
\Rightarrow 可微
梯度

少量层 NN: 则可以随意选择.

爆炸或
梯度消失.



常见的激励函数:



activation function 出线在哪里?

tensorflow structure:

input \rightarrow Layer 1.

\rightarrow Layer 2. activation func \rightarrow output
筛选 \uparrow

classification:

用于判断是否“是”的问题.

常见:

`tf.nn.relu`

`tf.nn.softmax(classification)`
`tf.nn.sigmoid`

每一个激励函数都有适用的场景.

1