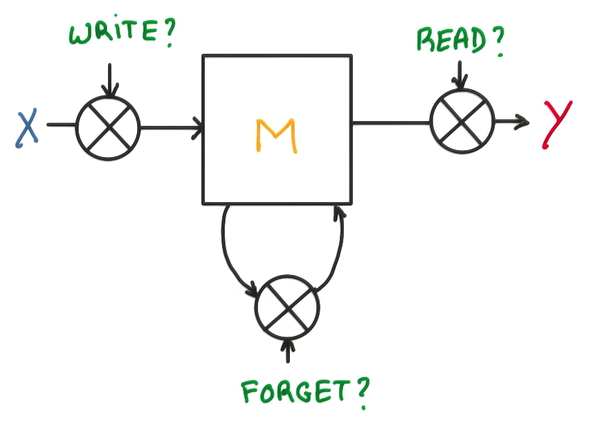
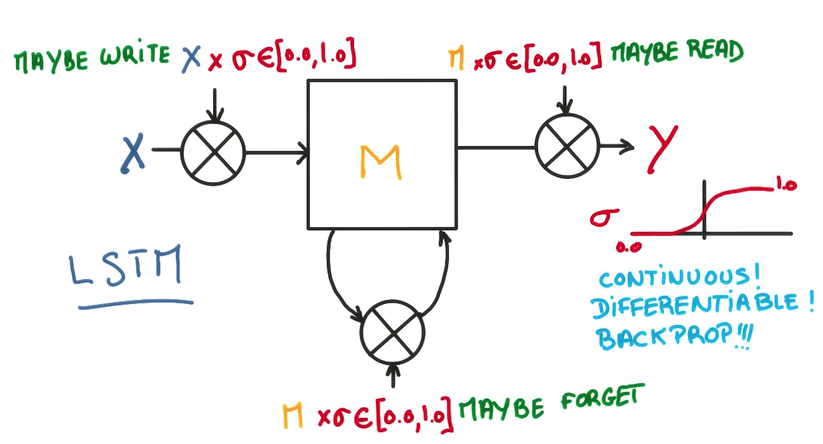


* 中间是一个简单的神经网络
* 将中间的部分换成LSTM-cell就能解决梯度消失问题
* 我们的目的是提高RNN的记忆能力
* Memory Cell

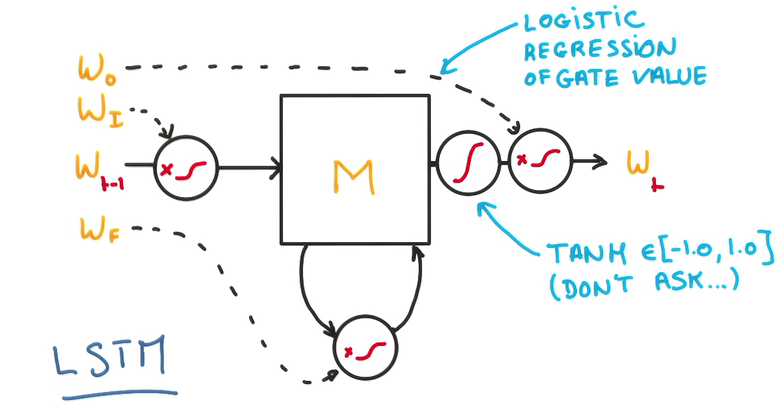


三个门，决定是否写/读/遗忘/写回

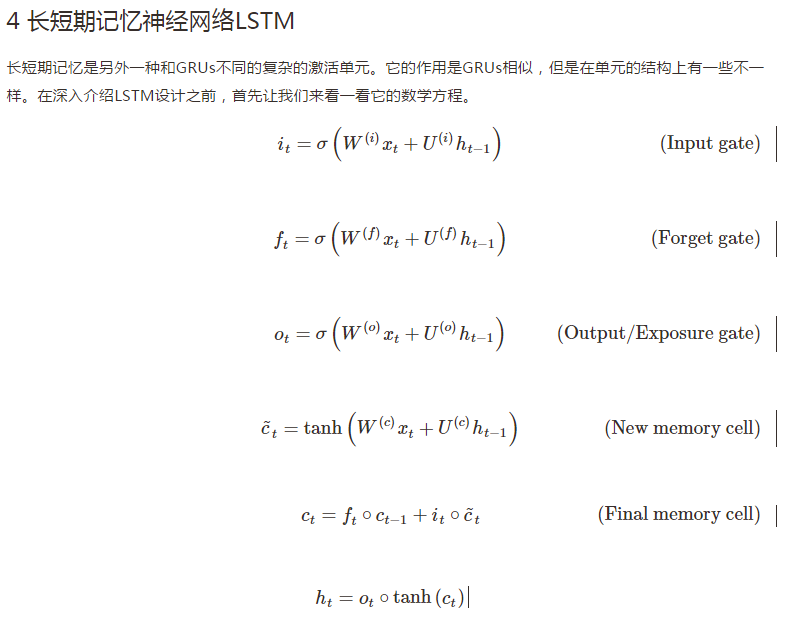
* 在每个门上，不单纯做yes/no的判断，而是使用一个权重，决定对输入的接收程度
* 这个权重是一个连续的函数，可以求导，也就可以进行训练，这是LSTM的核心

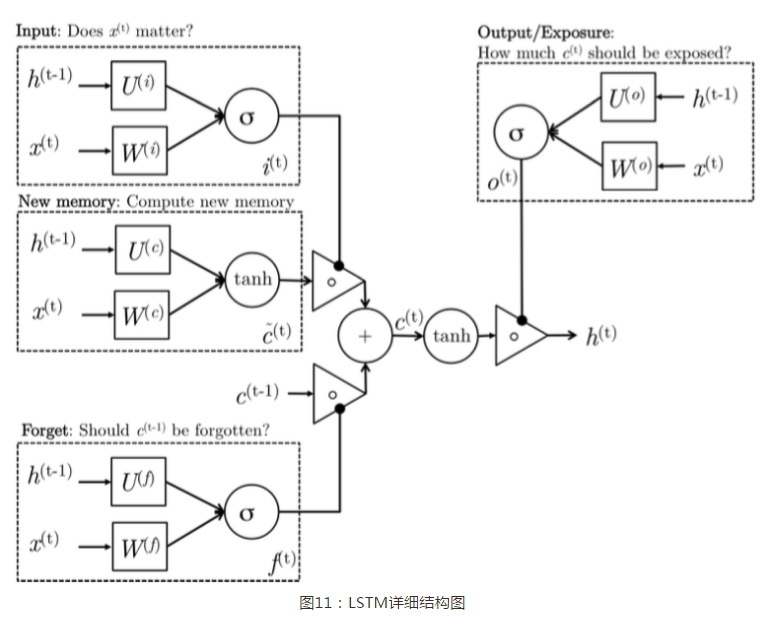


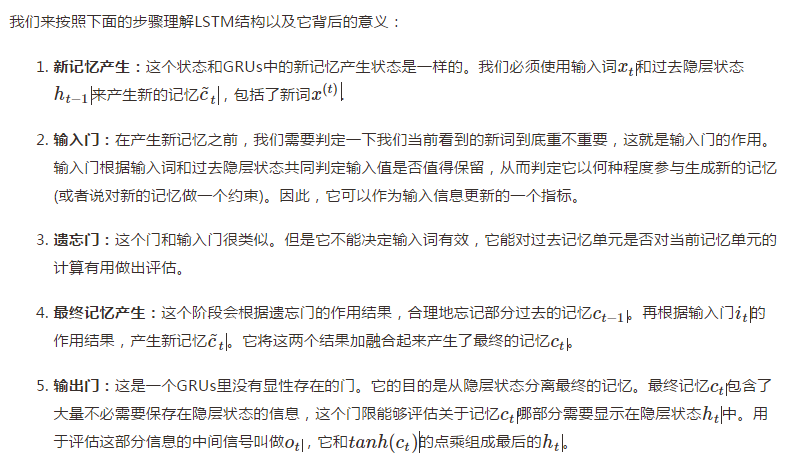
* 用一个逻辑回归训练这些门，在输出进行归一化

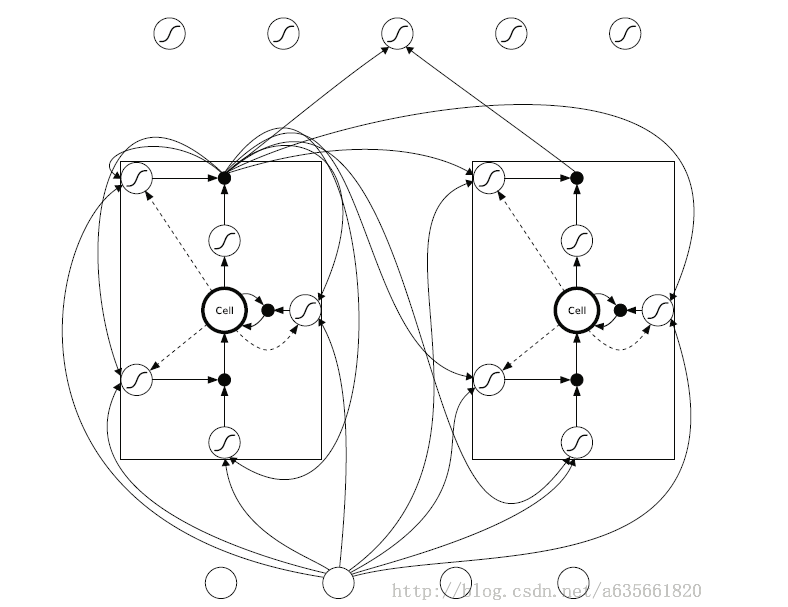


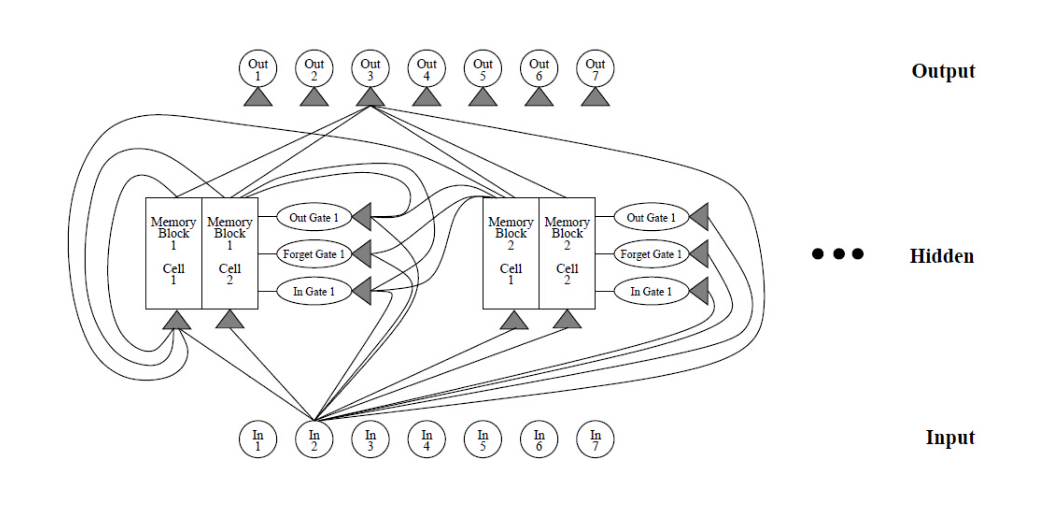
* 这样的模型能让整个cell更好地记忆与遗忘
* 由于整个模型都是线性的，所以可以方便地求导和训练

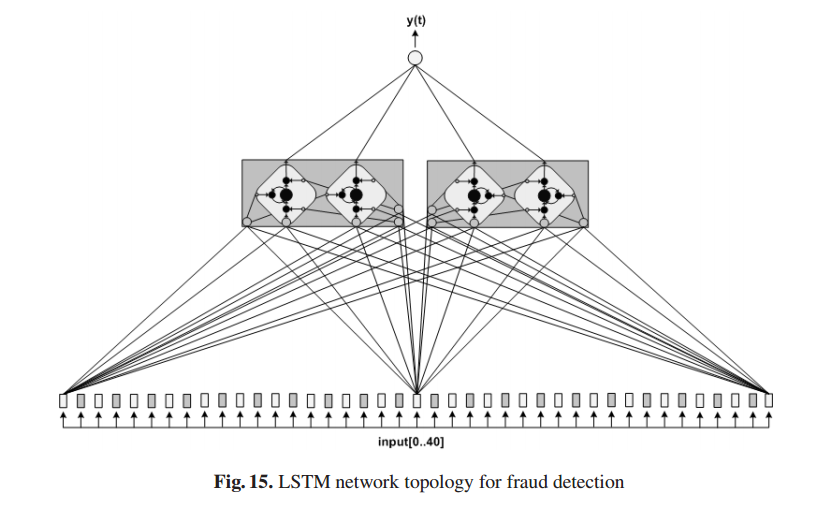


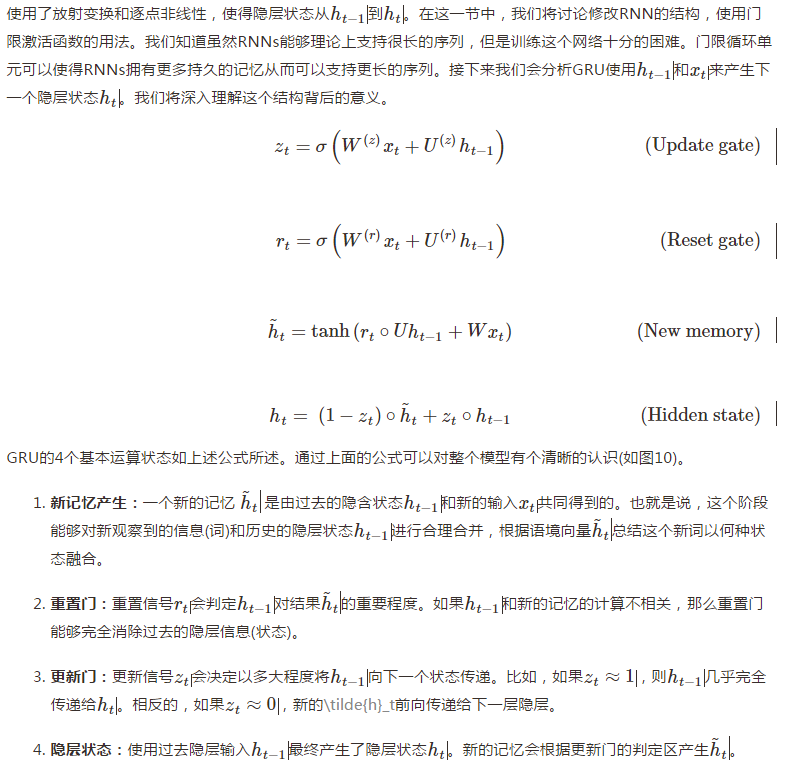
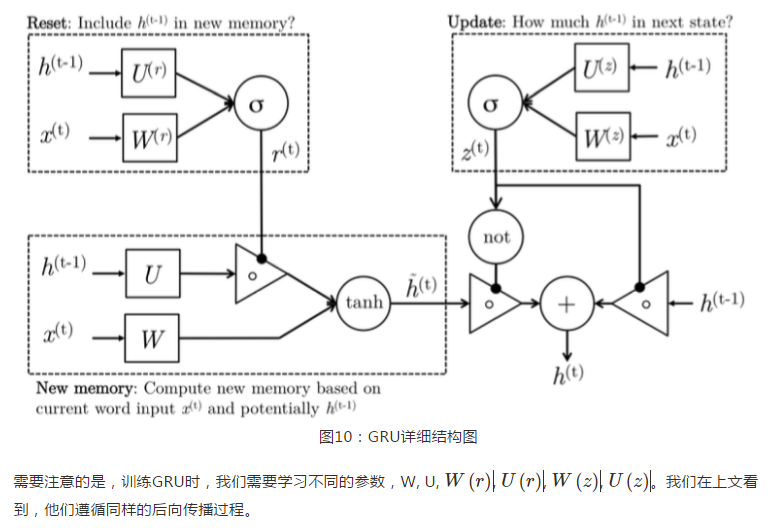


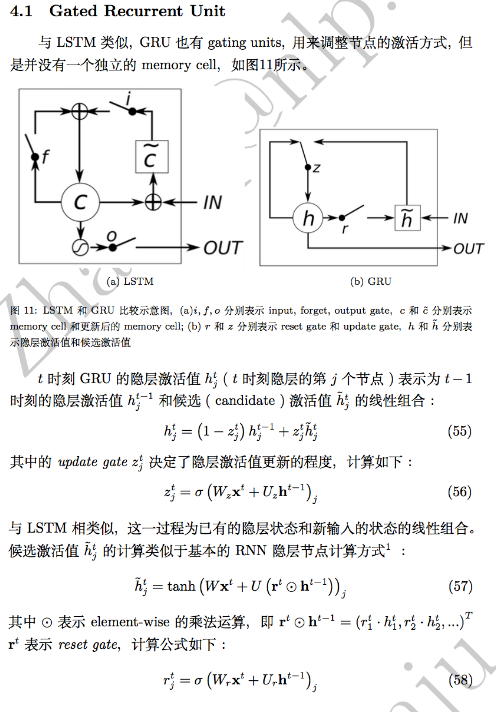










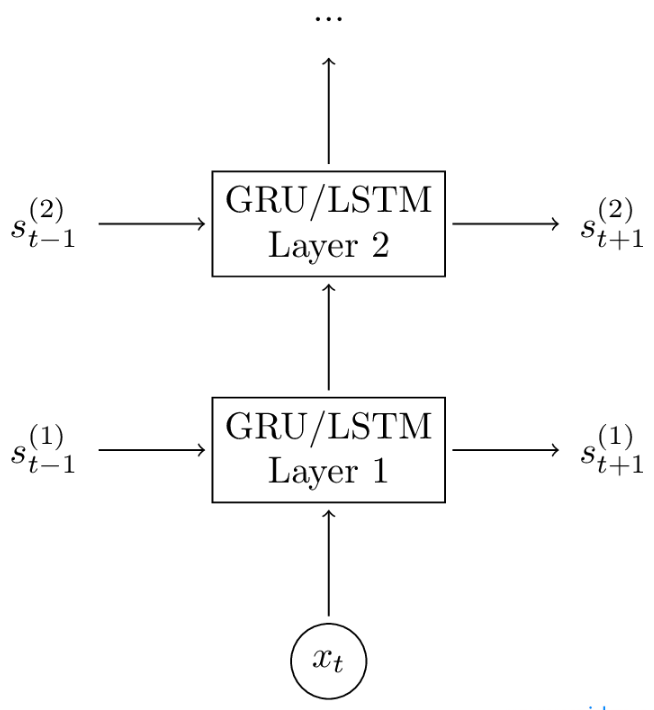


**重要**



Adding a second GRU layer

Adding a second layer to our network allows our model to capture higher-level interactions. You could add additional layers, but I didn’t try that for this experiment. You’ll likely see diminishing returns after 2-3 layers and unless you have a huge amount of data (which we don’t) more layers are unlikely to make a big difference and may lead to overfitting.



#### LSTM（Long Short-Term Memory）

梯度消失会导致分类器只对最近的消息的变化有反应，淡化以前训练的参数，也不能用比值的方法来解决   
- 一个RNN的model包含两个输入，一个是过去状态，一个是新的数据，两个输出，一个是预测，一个是将来状态

