# 人人都能看懂的GRU

陈诚

努力修炼内力

> 接续上一次介绍的LSTM，这里我又很不要脸地使用“人人都能看懂的xxx”来作为标题，来将对GRU进行介绍。同样这里的内容是对台大李宏毅老师课程视频的一些记录以及自己的一些整理和思考。对于不懂基础RNN和LSTM的同学可以先看看我的上一篇文章 *人人都能看懂的LSTM*。有任何疑问欢迎交流。

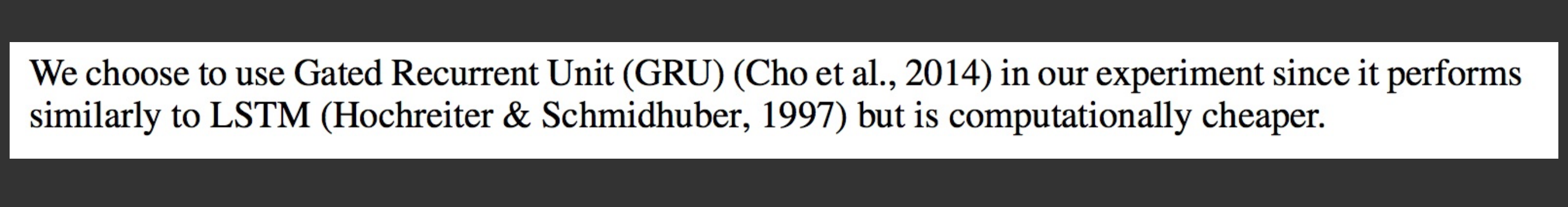
---

## 1. 什么是GRU

GRU（Gate Recurrent Unit）是循环神经网络（Recurrent Neural Network, RNN）的一种。和LSTM（Long-Short Term Memory）一样，也是为了解决长期记忆和反向传播中的梯度等问题而提出来的。

GRU和LSTM在很多情况下实际表现上相差无几，那么为什么我们要使用新人GRU（2014年提出）而不是相对经受了更多考验的LSTM（1997提出）呢。

下图1-1引用论文中的一段话来说明GRU的优势所在。



We choose to use Gated Recurrent Unit (GRU) (Cho et al., 2014) in our experiment since it performs similarly to LSTM (Hochreiter & Schmidhuber, 1997) but is computationally cheaper.

图1-1 R-NET: MACHINE READING COMPREHENSION WITH SELF-MATCHING NETWORKS（2017）

> 简单译文：我们在我们的实验中选择GRU是因为它的实验效果与LSTM相似，但是更易于计算。

**简单来说就是贫穷限制了我们的计算能力...**

相比LSTM，使用GRU能够达到相当的效果，并且相比之下更容易进行训练，能够很大程度上提高训练效率，因此很多时候会更倾向于使用GRU。

OK，那么为什么说GRU更容易进行训练呢，下面开始介绍一下GRU的内部结构。

---

## 2. GRU浅析

### 2.1 GRU的输入输出结构

GRU的输入输出结构与普通的RNN是一样的。

有一个当前的输入 $x^t$ ，和上一个节点传递下来的隐状态（hidden state） $h^{t-1}$ ，这个隐状态包含了之前节点的相关信息。

结合 $x^t$ 和 $h^{t-1}$ ，GRU会得到当前隐藏节点的输出 $y^t$ 和传递给下一个节点的隐状态 $h^t$ 。

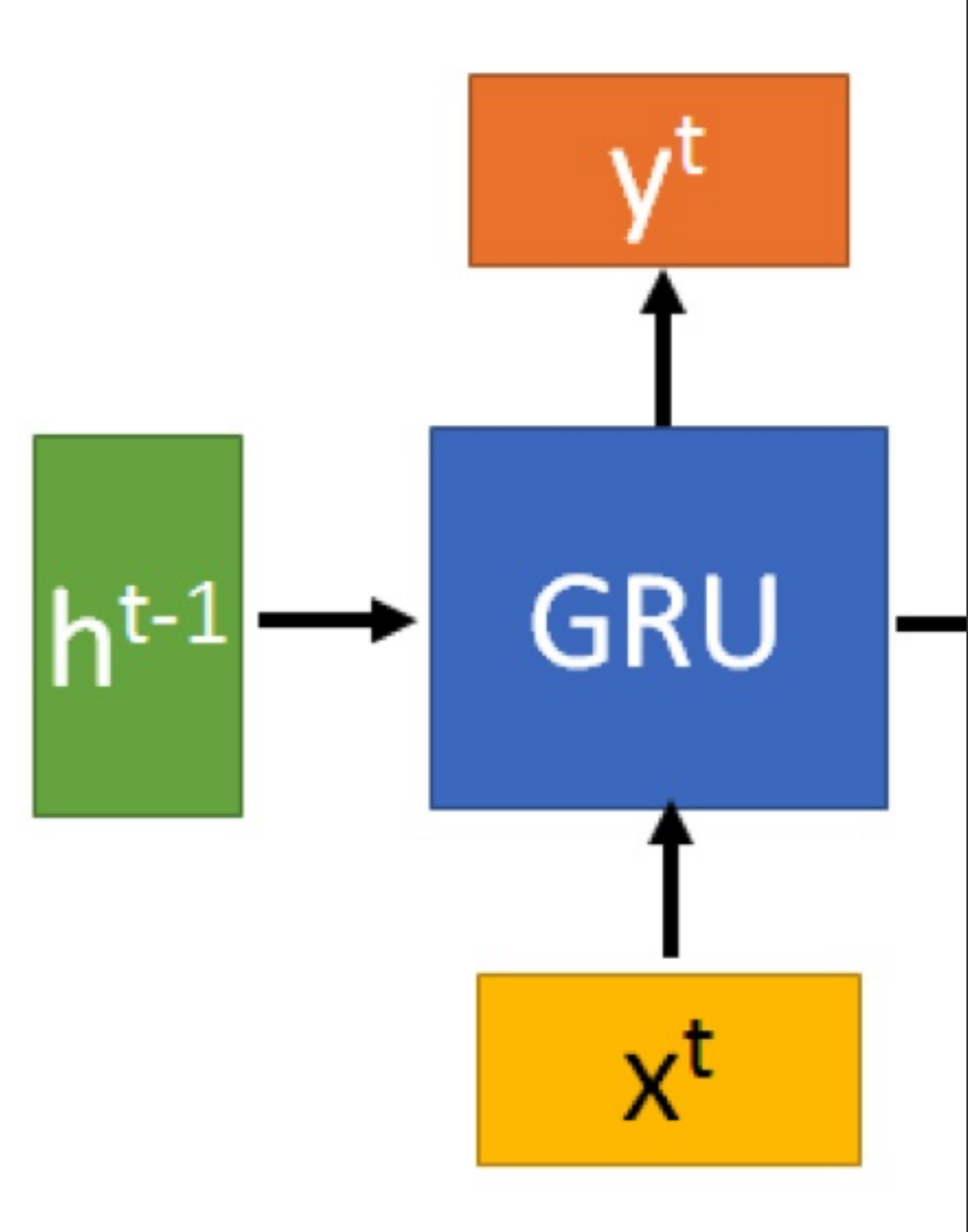


图2-1 GRU的输入输出结构

那么，GRU到底有什么特别之处呢？下面来对它的内部结构进行分析！

---

### 2.2 GRU的内部结构

首先，我们先通过上一个传输下来的状态 $h^{t-1}$ 和当前节点的输入 $x^t$ 来获取两个门控状态。如下图2-2所示，其中 $r$ 控制重置的门控（reset gate）， $z$ 为控制更新的门控（update gate）。

> Tips： $\sigma$ 为*sigmoid*函数，通过这个函数可以将数据变换为0-1范围内的数值，从而来充当门控信号。

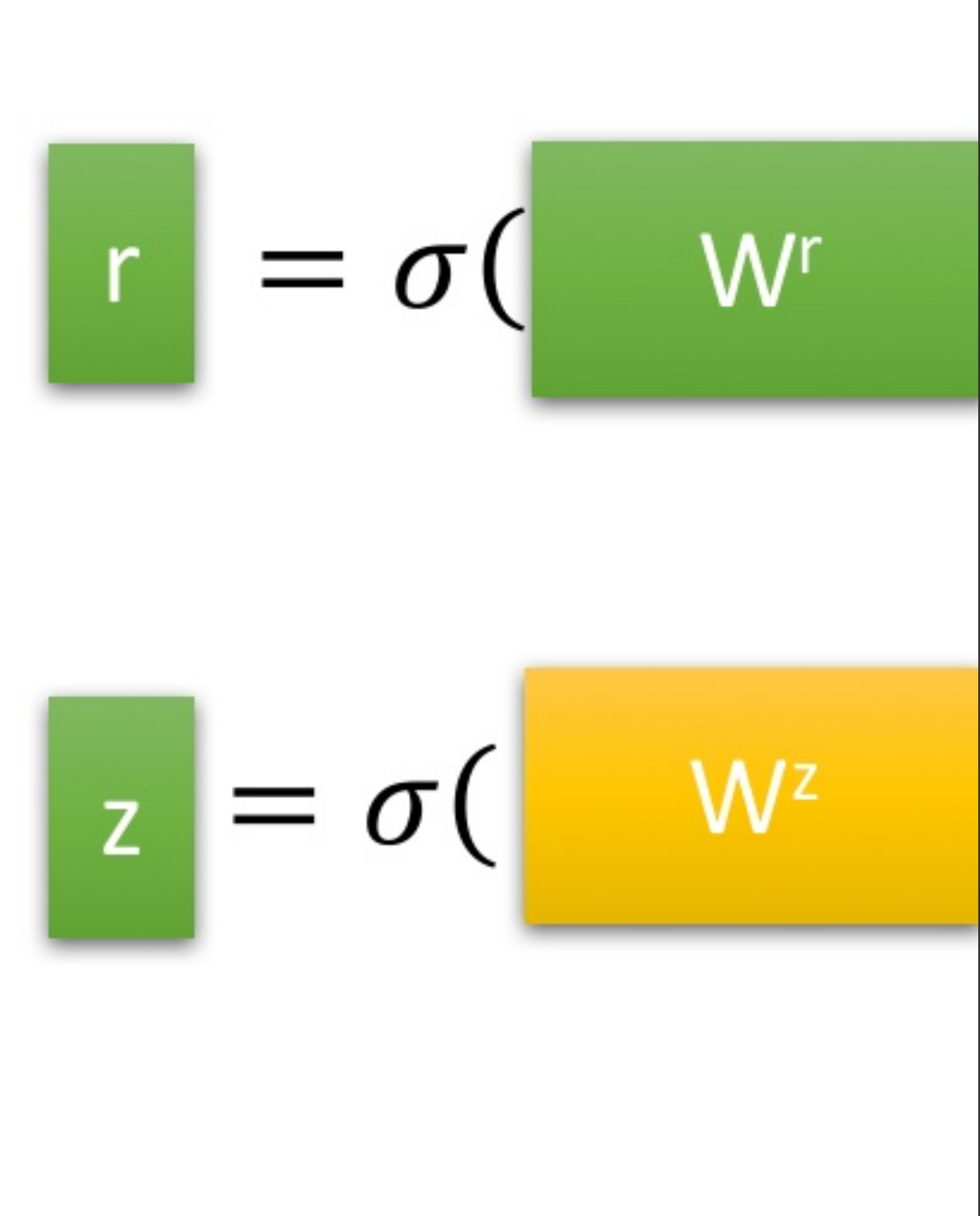


图2-2 r，z门控

**与LSTM分明的层次结构不同，下面将对GRU进行一气呵成的介绍~~~ 请大家屏住呼吸，不要眨眼。**

得到门控信号之后，首先使用重置门控来得到“重置”之后的数据 $h^{t-1'} = h^{t-1} \odot r$ ，再将 $h^{t-1'}$ 与输入 $x^t$ 进行拼接，再通过一个tanh激活函数来将数据放缩到**-1~1**的范围内。即得到如下图2-3所示的 $h'$ 。

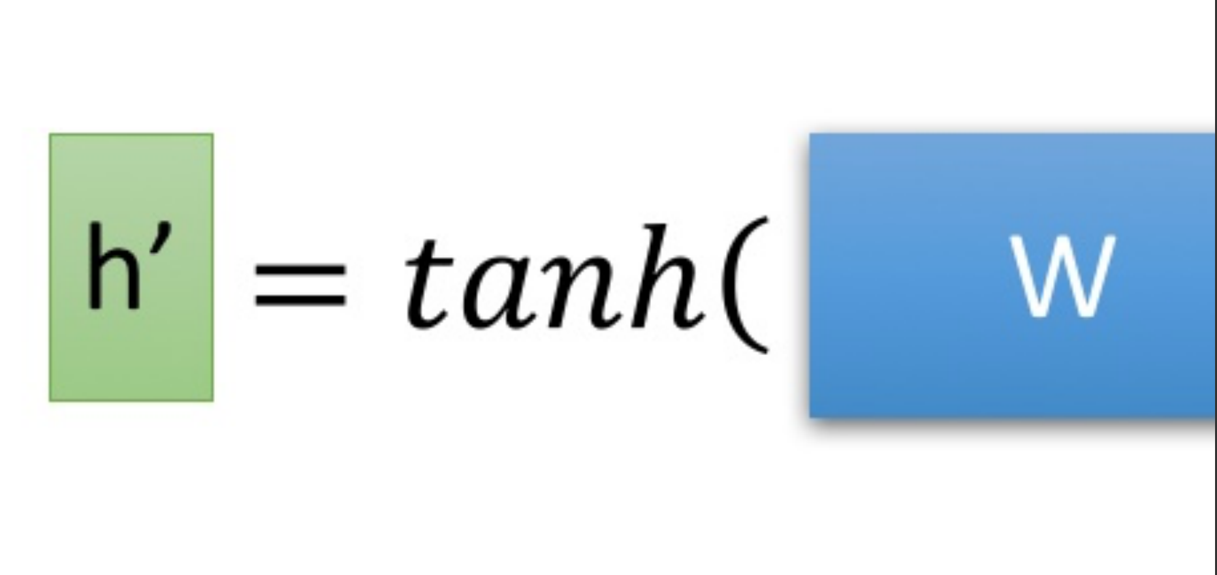


图2-3 包含当前输入数据的h&amp;amp;#39;

这里的 $h'$ 主要是包含了当前输入的 $x^t$ 数据。有针对性地对 $h'$ 添加到当前的隐藏状态，相当于”记忆了当前时刻的状态“。类似于LSTM的选择记忆阶段（参照我的上一篇文章）。

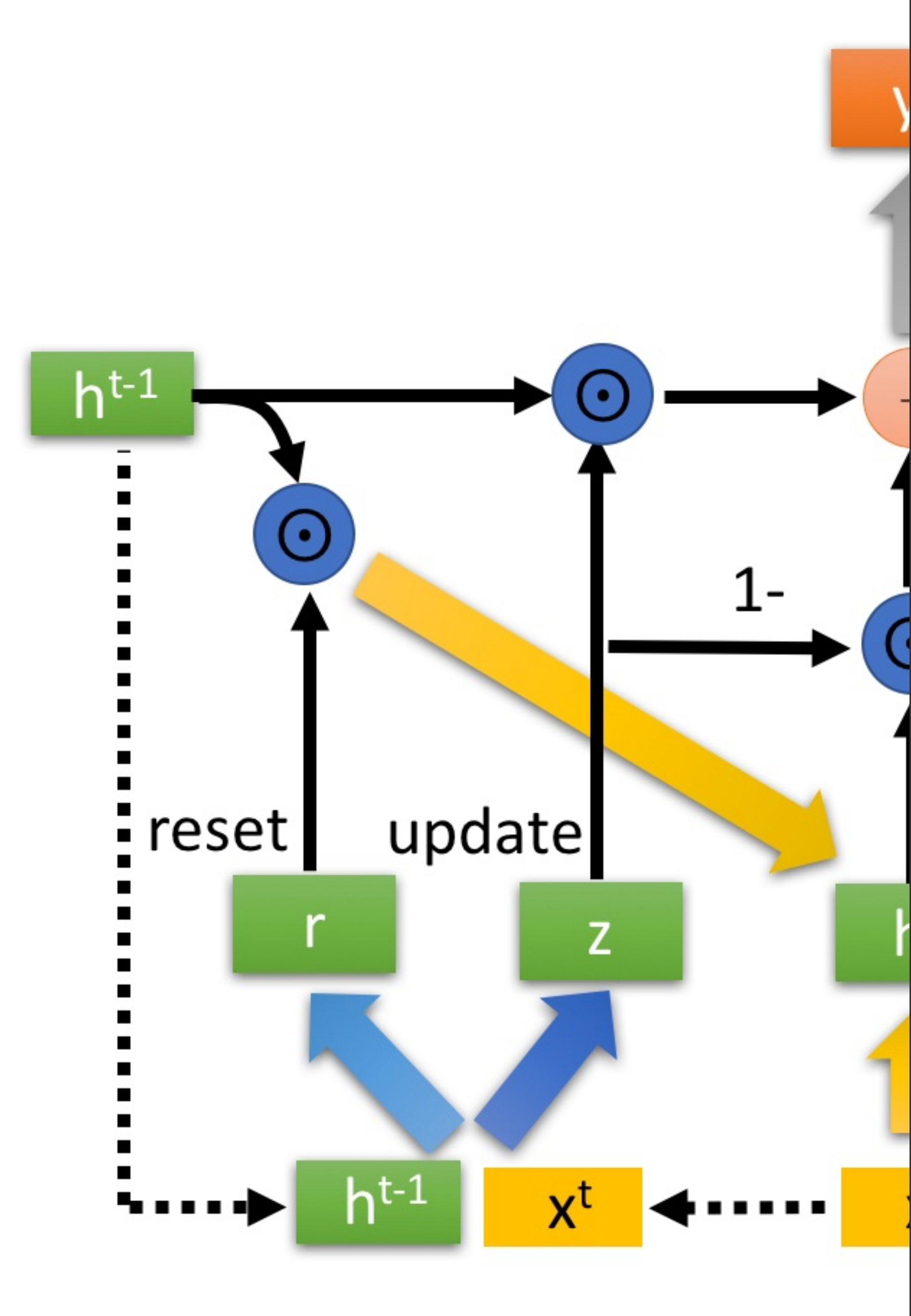


图2-3 GRU的内部结构

> 图2-3中的 $\odot$ 是Hadamard Product，也就是操作矩阵中对应的元素相乘，因此要求两个相乘矩阵是同型的。 $\oplus$ 则代表进行矩阵加法操作。

---

最后介绍GRU最关键的一个步骤，我们可以称之为”**更新记忆**“阶段。

在这个阶段，我们同时进行了遗忘了记忆两个步骤。我们使用了先前得到的更新门控 $z$ （update gate）。

**更新表达式**： $h^t = z \odot h^{t-1} + (1-z) \odot h'$

首先再次强调一下，门控信号（这里的 $z$ ）的范围为0~1。门控信号越接近1，代表”记忆“下来的数据越多；而越接近0则代表”遗忘“的越多。

GRU很聪明的一点就在于，**我们使用了同一个门控 $z$ 就同时可以进行遗忘和选择记忆（LSTM则要使用多个门控）**。

- $z \odot h^{t-1}$ ：表示对原本隐藏状态的选择性“遗忘”。这里的 $z$ 可以想象成遗忘门（forget gate），忘记 $h^{t-1}$ 维度中一些不重要的信息。
- $(1-z) \odot h'$ ： 表示对包含当前节点信息的 $h'$ 进行选择性”记忆“。与上面类似，这里的 $(1-z)$ 同理会忘记 $h'$ 维度中的一些不重要的信息。或者，这里我们更应当看做是对 $h'$ 维度中的某些信息进行选择。
- $h^t = z \odot h^{t-1} + (1-z) \odot h'$ ：结合上述，这一步的操作就是忘记传递下来的 $h^{t-1}$ 中的某些维度信息，并加入当前节点输入的某些维度信息。

> 可以看到，这里的遗忘 $z$ 和选择 $(1-z)$ 是联动的。也就是说，对于传递进来的维度信息，我们会进行选择性遗忘，则遗忘了多少权重 （ $z$ ），我们就会使用包含当前输入的 $h'$ 中所对应的权重进行弥补 $(1-z)$ 。以保持一种”恒定“状态。

---

## 3. LSTM与GRU的关系

GRU是在2014年提出来的，而LSTM是1997年。他们的提出都是为了解决相似的问题，那么GRU难免会参考LSTM的内部结构。那么他们之间的关系大概是怎么样的呢？这里简单介绍一下。

大家看到 $r$ (reset gate)实际上与他的名字有点不符。我们仅仅使用它来获得了 $h'$ 。

那么这里的 $h'$ 实际上可以看成对应于LSTM中的hidden state；上一个节点传下来的 $h^{t-1}$ 则对应于LSTM中的cell state。z对应的则是LSTM中的 $z^f$ forget gate，那么 $(1-z)$ 我们似乎就可以看成是选择门 $z^i$ 了。大家可以结合我的两篇文章来进行观察，这是非常有趣的。

## 4. 总结

GRU输入输出的结构与普通的RNN相似，其中的内部思想与LSTM相似。

与LSTM相比，GRU内部少了一个”门控“，参数比LSTM少，但是却也能够达到与LSTM相当的功能。考虑到硬件的**计算能力和时间成本**，因而很多时候我们也就会选择更加”实用“的GRU啦。

编辑于 2018-10-16

「真诚赞赏，手留余香」

2 人已赞赏

RNN LSTM 深度学习（Deep Learning）