循环神经网络 RNN & 在新奇检测中的运用 - Part 1

与卷积网络和多层感知机不同，循环神经网络（Recurrent Neural Network）为了更好地处理时序信息（Time Series）而设计的。它的特征在于利用状态变量来储存过去的信息，并和当前时间的输入一起共同决定输出。

语言模型就是我们生活中最常见的时序信息。一句话里的每一个字都是按时间顺序出现的，之前说的话一定会决定当前说出口的单词。所以RNN被广泛用于语音识别、语言模型。文本型数据也是同理。但RNN不局限于语言和文字，只要是按时间采样的数据都能很好的被RNN处理，比如市场价格。我们可以用RNN来辨别一个时间段中，市场价格的非常规波动点，也就是新奇检测。本文的运用部分也会侧重于介绍RNN在新奇检测中的运用。

本文一共分为两个部分：

- **Part 1 主要讲述RNN和其衍生LSTM的理论基础**
- Part 2 是一个RNN在新奇检测场景中的运用和代码实现

1. RNN的基础结构

循环神经网络，顾名思义，就是在同一个神经元上循环计算的神经网络。假设$X_t \in \mathbb{R}^{n\times d}$ 是序列中时间步的 t 中小批量输入，H_t 是 t 时间的隐藏状态（hidden state）。与多层感知机最大的不同是：RNN要用上一个时间步 $t-1$ 的隐藏状态 H_{t-1} 来计算 H_t：

$$H_t = \phi(X_t W_{xh} + H_{t-1} W_{hh} + b_h)$$

上式的几个要点：

- 维度分析：$H_t \in \mathbb{R}^{n\times h}$，所以 $W_{xh} \in \mathbb{R}^{d\times h}$, $W_{hh} \in \mathbb{R}^{h\times h}$, $b_h \in \mathbb{R}^h$
- ϕ 是激活函数。相比感知机，这里多了一个权重参数 W_{hh} 来描述如何使用上一时间隐藏状态 H_{t-1}。
- 上式是个循环计算，用 H_{t-1} 计算 H_t。所以 H_t 也捕捉了截至当前时间 t 的历史信息，是RNN的“记忆”
- 权重参数 W_{xh}、W_{hh} 和偏差 b_h 不会随着时间序列 t 而改变。

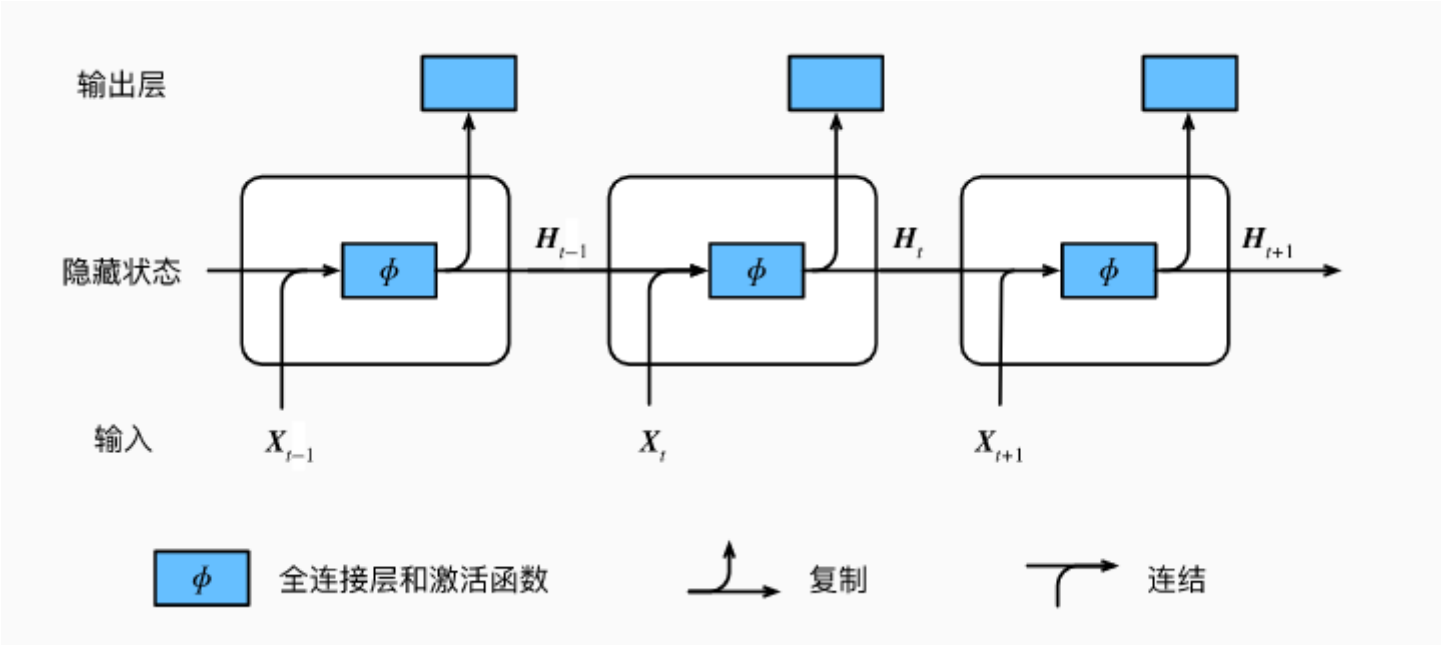
上式是RNN的一种常见的基本构造，后文会接着介绍衍生结构。在时间步 t，输出为 $O_t \in \mathbb{R}^{n*q}$：

$$O_t = W_{hq} H_t + b_q$$

- 和感知机类似。维度分析：$W_{qh} \in \mathbb{R}^{h \times q}$，$b_q \in \mathbb{R}^q$

要点： 权重参数 W_{xh}、W_{hh}、W_{qh} 和偏差 b_h、b_q 不会随着时间序列 t 而改变。在不同时间步，RNN始终使用这些参数。所以，**RNN的参数数量不随时间步的增长而增长。**

我们用一张图来解释我们刚才说的这个RNN的结构：



该图解释了3个相邻的时间步在RNN中的运作模式，非常直观地展示了时序信息在RNN中通过隐藏状态 H_t 来进行传递：。所以说隐藏状态的计算是RNN的核心也不为过。

2. 长短期记忆 LSTM

长短期记忆，又称 LSTM（Long Short Term Memory）是上述RNN的一个衍生。LSTM除了隐藏状态 H_t 以外，还会计算一个和 H_t 形态相同的 **记忆细胞** C_t，记录一些额外的信息。并用不同的“门”来控制信息的流入和流出。所以LSTM也被叫做一种门控循环神经网络。大家要记住的是，LSTM本身循环计算的模式和上述RNN是一样的，变的只是隐藏状态（H_t 和 C_t）的计算方式。我们接下来就一步步地来分析LSTM的结构。

2.1 输入门、遗忘门、输出门

LSTM用“门”来控制信息的流向，三种不同的门在不同阶段扮演了不同的角色。LSTM计算三种门的输入均为当前时间步 t 的输入 X_t 和上一个时间步 $t-1$ 的隐藏状态 H_{t-1}。门的输出由激活函数sigmoid的全连接层计算得到。具体情况如下：

假设隐藏单元数为 h，时间 t 的输入为 $X_t \in \mathbb{R}^{n \times d}$，$t-1$ 的隐藏状态是 H_{t-1}。则时间 t 的输入门 $I_t \in \mathbb{R}^{n \times h}$，遗忘门 $F_t \in \mathbb{R}^{n \times h}$ 和输出门 $O_t \in \mathbb{R}^{n \times h}$ 分别为：

$$
\begin{aligned}
I_t &= \sigma(X_t W_{xi} + \boldsymbol{H}_{t-1} W_{hi} + b_i),\\
F_t &= \sigma(X_t W_{xf} + \boldsymbol{H}_{t-1} W_{hf} + b_f),\\
O_t &= \sigma(X_t W_{xo} + \boldsymbol{H}_{t-1} W_{ho} + b_o)
\end{aligned}
$$

- 所有的 W 都是权重参数，所有的 b 都是偏差参数。这里不再做维度分析，和上文十分类似。
- 和RNN一样，所有的 W 和 b 均不受时间步的影响，是独立的参数。
- 因为用了sigmoid函数作为激活函数，**所有门的取值均在 $[0,1]$ 之间。**

2.2 记忆细胞

接下来，LSTM需要先计算候选记忆细胞 $\tilde{C}_t$，计算方式和门类似，不过这次激活函数是tanh，所以值域变成了 $[-1,1]$：

$$
\tilde{C}_t = tanh(X_t W_{xc} + \boldsymbol{H}_{t-1} W_{hc} + b_c)
$$

然后，我们可以通过值域在 $[0,1]$ 中的输入门、遗忘门和输出门来控制隐藏状态中信息的流动，一般是通过元素相乘（$\odot$）来实现。那首先就是当前时间步的记忆细胞 $C_t \in \mathbb{R}^{n \times h}$。他的计算组合了**上一个时间步的记忆细胞** C_{t-1} 和 **当前时间步的候选记忆细胞** $\tilde{C}_t$ 的信息，并通过输入门和遗忘门控制信息的流动。

$$
C_t = F_t \odot C_{t-1} + I_t \odot \tilde{C}_t
$$

我们针对上式可以做出以下分析：

- 遗忘门控制 $t-1$ 的记忆细胞，即过去的信息如何流入当前；输入门控制 t 的候选记忆细胞，即当前的输入信息如何使用
- 在 $t-n$ 到 t 这个时间段里，如果遗忘门 F 一直近似于 1 且输入门 I 一直近似于 0，则 C_{t-n} 的信息将会一直保留到 C_t。由此可见，记忆细胞可以捕捉时间步距离较大的依赖关系。
- 该设计可以有效应对RNN中梯度衰减的问题

2.3 隐藏状态

最后就是计算隐藏状态 H_t 以及当前时间步骤的输出了。隐藏状态的计算就涉及到了用之前算出来的输出门来控制当前时间步的记忆细胞里的信息。

$$
H_t = O_t \odot tanh(C_t)
$$

针对上式，我们可以做出以下分析：

- $O_t \in [0,1]$，$tanh(C_t) \in [-1,1]$，所以隐藏状态的取值范围是 $H_t \in [-1,1]$
- 若 O_t 近似于1，则记忆细胞将所有信息传递给隐藏状态，使用在输出上；反之若 O_t 近似于0，则记忆细胞的信息自己保留。

输出和RNN是相同的：

$$y = H_t W_{hq} + b_q$$

最后根据场景需要，如果是分类问题就输出 $softmax(y)$。

2.4 LSTM的从零实现

从零实现不在这里赘述，基本就是以上的计算步骤。具体可以参考这篇文章。

Reference

1. https://zh.d2l.ai/chapter_recurrent-neural-networks