

词表示 → 词典展开为一个向量 one-hot vector

词性标注任务 \Rightarrow 将词分为名、动……九维向量 one-hot vector
全连接层输入 \rightarrow 多个词 C 个词向量大小为 (字典长 X 1) 参数过多
RNN 动机
前馈网络的一些不足：
1. 连接存在层与层之间，每层的节点之间是无连接的。（无循环）
2. 输入和输出的维数都是固定的，不能任意改变。无法处理变长的序列数据。
3. 假设每次输入都是独立的，也就是说每次网络的输出只依赖于当前的输入。

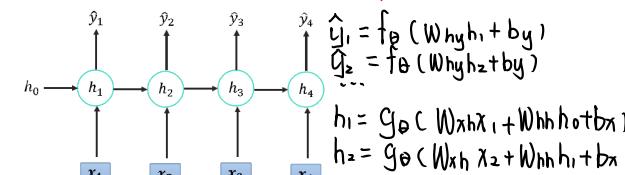
仍无法处理时间序列数据 → 不同时间点收集到的数据，反映事物随时间变化状态程度
前后有关联 视频、音乐、生物序列 DNA、工业传感、NLP

RNN



任务：预测某一时间节点的输出向量 非线性函数

使用一进归函数对时序信息 x 建模 $h_t = \sigma_{\theta}(h_{t-1}, x_t)$ 引入隐状态对序列数据提取特征，再转换为输出
新状态 新旧



Elman/vanilla RNN 经典结构

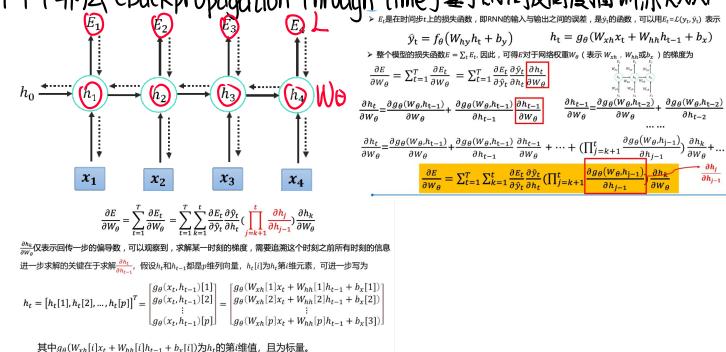
上述均为 many to many 结构

应用示例	
1 to Many	1. 从图像生成文字，输入图像的特征，输出为一段句子 2. 根据某些语音或音乐，输入为图像特征，输出为一段语音或音乐
Many to 1	1. 输入一段文字，判断其所属类别 2. 输入一个句子，判断情感倾向 3. 输入一段视频，判断其新闻类别
Sequence to Sequence	1. 机器翻译，输入一种语言文本序列，输出另外一种语言文本序列 2. 文本摘要，输入文本序列，输出该文本序列摘要 3. 图像理解，输入文字，输出问题答案 4. 语音识别，输入语音序列信息，输出文字序列

RNN 与 CNN 比较

类别	特点描述
相同点	1. 传统神经网络的扩展。 2. 前向计算产生结果，反向计算模型更新。 3. 每层神经网络横向向多个神经元共享，纵向可以有多层神经网络连接。 4. 语音识别，输入语音序列信息，输出文字序列。
不同点	1. CNN 空间扩展，神经元与特征卷积。 2. RNN 可以用于描述时间上连续状态的输出，有记忆功能。CNN 用于静态输出。

BPTT 算法 (backpropagation through time) 基于时间反向传播训练 RNN



$\frac{\partial E}{\partial W_h}$ 只表示回传一步的偏导数，可以推断出，要解某一步的梯度，需要追溯这个时刻之前所有时刻的信息

进一步求解的关键在于求解 $\frac{\partial E}{\partial h_{t-1}}$ ，假设 h_{t-1}, h_t 是两个向量， $h_{t-1}[i]$ 表示第 i 元素，可进一步写为

$$h_t = [h_t[1], h_t[2], \dots, h_t[p]]^T = \begin{bmatrix} g_\theta(x_t, h_{t-1}[1]) \\ g_\theta(x_t, h_{t-1}[2]) \\ \vdots \\ g_\theta(x_t, h_{t-1}[p]) \end{bmatrix}$$

其中 $g_\theta(x_t, h_{t-1}[i]) = g_\theta(x_t, h_{t-1})[i]$ 为 h_t 的第 i 位值，即为标量。

进一步求得 $\frac{\partial E}{\partial h_{t-1}} = \frac{\partial E}{\partial h_{t-1}} = \sum_{i=1}^p \frac{\partial E}{\partial h_{t-1}} \frac{\partial h_t}{\partial h_{t-1}} \frac{\partial g_\theta}{\partial h_{t-1}} \frac{\partial h_{t-1}}{\partial h_{t-1}}$

将上式代入到网络中权重 W_h 的梯度公式中，可得

$$\frac{\partial E}{\partial W_h} = \sum_{t=1}^T \sum_{k=1}^p \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_t} \left(\prod_{j=t+1}^p \text{diag}(g'_\theta(x_j, h_{j-1})) W_{hj} \right) \frac{\partial h_k}{\partial W_h}$$

其中 $\text{diag}(g'_\theta(x_t, h_{t-1}))$ 为对角矩阵，其对角元素 $g'_\theta(x_t, h_{t-1})[i]$ 。

将上式代入到网络中权重 W_h 的梯度公式中，可得

$$\frac{\partial E}{\partial W_h} = \sum_{t=1}^T \sum_{k=1}^p \frac{\partial E}{\partial y_k} \frac{\partial y_k}{\partial h_t} \left(\prod_{j=t+1}^p \text{diag}(g'_\theta(x_j, h_{j-1})) W_{hj} \right) \frac{\partial h_k}{\partial W_h}$$

RNN 通过上式实现沿时间的反向传播。误差随时间从步到步（重叠矩阵相乘）

梯度消失与梯度爆炸

如时间长期依赖问题 \Rightarrow 追溯很久之前的语境

$$|g'_\theta(x_j, h_{j-1})| > 1 \quad \text{在最大值 } |g'_\theta(x_j, h_{j-1})| \leq \gamma \quad \left\{ \begin{array}{l} \text{sigmoid : (0, 0.25]} \\ \text{tanh : (0, 1]} \\ |W_h| \end{array} \right.$$

假设 λ_h 是 W_{hh} 矩阵的奇异值分解后的最大值，如果 $\lambda_h < \frac{1}{\gamma}$ ，可知：

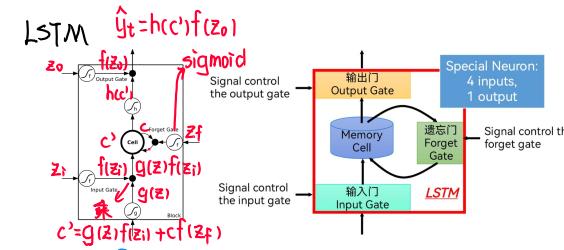
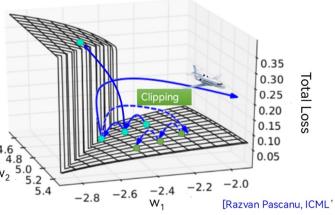
$$|\text{vt}| \cdot \frac{\partial h_t}{\partial h_{t-1}} \leq |\text{diag}(g'_\theta(x_j, h_{j-1}))| \cdot |W_{hh}| = \eta < \gamma^{-1} < 1$$

通过多次连乘，得到

$$\prod_{j=t+1}^p \text{diag}(g'_\theta(x_j, h_{j-1})) W_{hj} \leq \eta^{p-t}$$

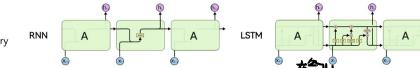
因为 $\eta < 1$ ，当沿着时间方向回传时 η^{p-t} 接近于 0。相反，如果 $\lambda_h > \frac{1}{\gamma}$ ，可证明会出现梯度爆炸。

梯度平面要平坦要么陡峭



LSTM 与 RNN 的区别

Long Short-term Memory (LSTM) 是门控类 RNN 中最著名的一种。它改善了 RNN 的记忆能力并解决了梯度爆炸消失问题。基于 RNN 的基础结构，LSTM 主要将原有的隐层循环单元替换成三个门控单元：输入门（Input Gate）、遗忘门（Forget Gate）和输出门（Output Gate）。通过下图对比看到 RNN 与 LSTM 单元结构的区别。



与 RNN 最主要区别在于加入了三个门控单元，控制信息通过多少

在 t 时刻，LSTM 的循环函数可写为

$$\begin{aligned} c_t &= f_t \otimes \text{tanh}(i_t \otimes W_{xc} x_t + b_c) \\ h_t &= o_t \otimes \text{tanh}(c_t) \\ c_{t+1} &= f_{t+1} \otimes \text{tanh}(i_{t+1} \otimes W_{xc} x_{t+1} + b_c) \end{aligned}$$

$$\begin{aligned} h_t &= o_t \otimes \text{tanh}(c_t) \\ c_t &= f_t \otimes c_{t-1} + i_t \otimes \hat{c}_t \\ \hat{c}_t &= \text{tanh}(W_{ch} x_t + W_{hc} h_{t-1} + b_c) \end{aligned}$$

其中 c_t 为 t 时刻的储存单元状态，存储序列的历史信息。三个门控单元分别为 i_t 、 f_t 、 o_t ，分别称之为输入门、输出门、遗忘门，下面分别解释三种门的工作模式。
遗忘门 $f_t = \sigma(W_f x_t + W_{hf} h_{t-1} + b_f)$ 对上一个单元状态信息选择性遗忘
输入门 $i_t = \sigma(W_i x_t + W_{hi} h_{t-1} + b_i)$ 决定当前时刻隐变量需更新的信息量
输出门 $o_t = \sigma(W_o x_t + W_{ho} h_{t-1} + b_o)$ 从记忆单元 c_t 产生隐层单元 h_t

LSTM 通过这种复杂的循环函数在每个时间步上对当前的输入和记忆的历史信息进行重新的组合。

很大程度上解决了梯度爆炸和梯度消失的问题。LSTM 的优点可以总结为以下几点：

- 通过增加循环函数的阶数程度，从而降低了梯度爆炸发生的可能性。
- 通过遗忘门的使用减少了梯度消失的可能性。遗忘门中的端置项 f_t 在初始时刻设置为一个较大的值，从而使输出接近日 1，即这一时刻的单元状态 c_t 与上一时刻可能接近 c_{t-1} ，因此在训练时，即使更新的神经依然面临梯度消失的风险， c_t 上的梯度能够通过遗忘门的引入一直回传而不会丢失。
- 解决梯度消失和梯度爆炸并不是设计 LSTM 的初衷。通过引入门控单元更新的修改 RNN 的模型结构，能够使得模型由原来的单向信息传递，是 LSTM 能够广泛运用和受到喜爱的重要原因。
- 研究者通过在 RNN 和 LSTM 的网络架构上进行改进，开发了多种 RNN 和 LSTM 的流行变体结构。下面我们将介绍几种经典的流行架构。

peephole 连接：隐状态及连接上单元 c_t 影响 GRU，将遗忘门与输入门合成重置门 r_t ，引入更新门 u_t

训练效果与 LSTM 类似
而提升了训练速度并降低了成本



多层 RNN

如果 RNN 包含多层的循环函数，便能得到多层 RNN，与标准的 RNN 网络相同，第一层输出的输入上一层直接接受前一时刻的输入信息，而其他隐藏层的输入则是上一时刻的隐藏状态和上一层隐藏状态在当前时刻的状态。Deep RNNs 的结构如下图所示：

双向 RNN / LSTM

单向 RNN 由于输入的顺序为按照时间顺序从左至右以此进行编码，每个时刻的隐状态的影响主要来自于当前时刻的输入与之前时刻的隐状态。但在一些任务中，当前时刻和之后时刻的信息均会对于当前时刻的输出产生作用。因此研究者设计了将两层 RNN 相加在一起，构成了双向 RNN，双向 RNN 的隐状态由两个方向的编码所得的隐状态组成。与双向 RNN (Bi-directional RNN) 类似，Bi-directional LSTM 有两层 LSTM。

GRU 结构图

LSTM 结构图

GRU 结构图