复习

- 列举几种卷积神经网络架构的经典骨干网络（Backbone）。
- ResNet主要解决了深层网络的什么问题？是如何解决的？
- 如果你正在构建一个基于卷积神经网络的图像分类器，输入图像为RGB彩色图像，尺寸为224×224，类别数量为10，请完成网络架构的设计：
  - 用图示或表格形式进行表达；
  - 给出每层的输入、输出特征图大小；
  - 给出卷积、池化等模块的必要超参数（例如卷积核数量、通道数、大小、步长、填充等等）。

《神经网络与深度学习》

循环神经网络

https://nndl.github.io/

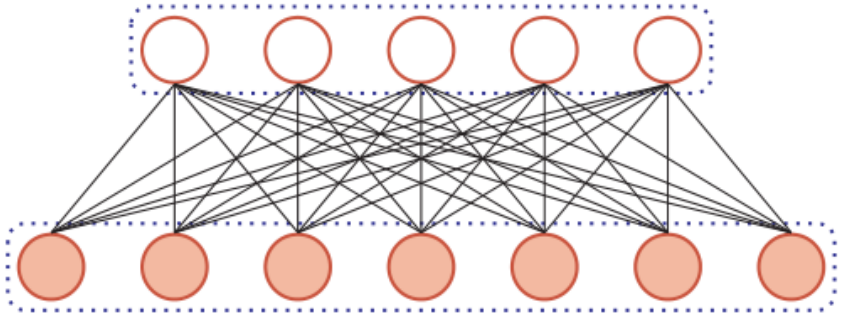
序列数据


新华网 > 财经 > 正文
2024
03/21
10:13:55
来源：经济参考报
新一轮工业设备更新或开启四万亿市场
字体： 小 中 大
分享到：
近段时间，多方正加快部署，聚焦钢铁、有色、石化、化工、建材等重点行业，开列项目清单，加大财税金融支持力度，完善用地用能要素保障，推动工业领域大规模设备更新。业内研究预计，随着政策利好释放，我国工业领域设备更新年规模约在4万亿元左右。
接受《经济参考报》记者采访的多位专家表示，工业既是各类设备的供给方，也是设备的需求方。推动工业设备向高端、智能、绿色、安全方向更新升级，将进一步拉动有效投资、提升发展质效，以设备升级带动我国制造业整体竞争力提升。
工业领域设备更新潜力足
日前，国务院印发《推动大规模设备更新和消费品以旧换新行动方案》（以下简称《行动方案》），明确"聚焦钢铁、有色、石化、化工、建材、电力、机械、航空、船舶、轻纺、电子等重点行业，大力推动生产

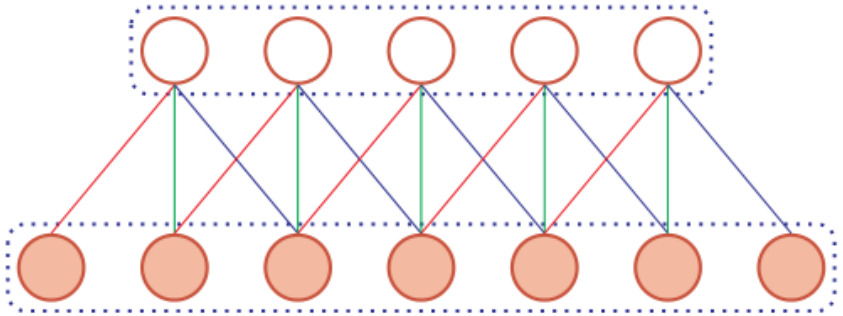

分时 五日 日K 周K 月K 季K 年K 120分 60分 30分 15分 5分 1分
区间统计 全屏显示
均线 MA5:3072.51 MA10:3061.50 MA20:3038.34 MA30:2974.25 MA60:2930.40
前复权
3111.08
3090.05
2945.41
2862.57
2779.73
2696.89
2614.06
2635.09
成交量 3.09亿手 MA5:3.62亿手 MA10:3.64亿手
5.47亿
MACD (12,26,9) DIF:42.03 DEA:42.68 MACD:-1.30
55.14
5.04
-45.05
2024-01
2024-02
2024-03

前馈网络

- 连接存在层与层之间，每层的节点之间是无连接的。（无循环）
  - 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。



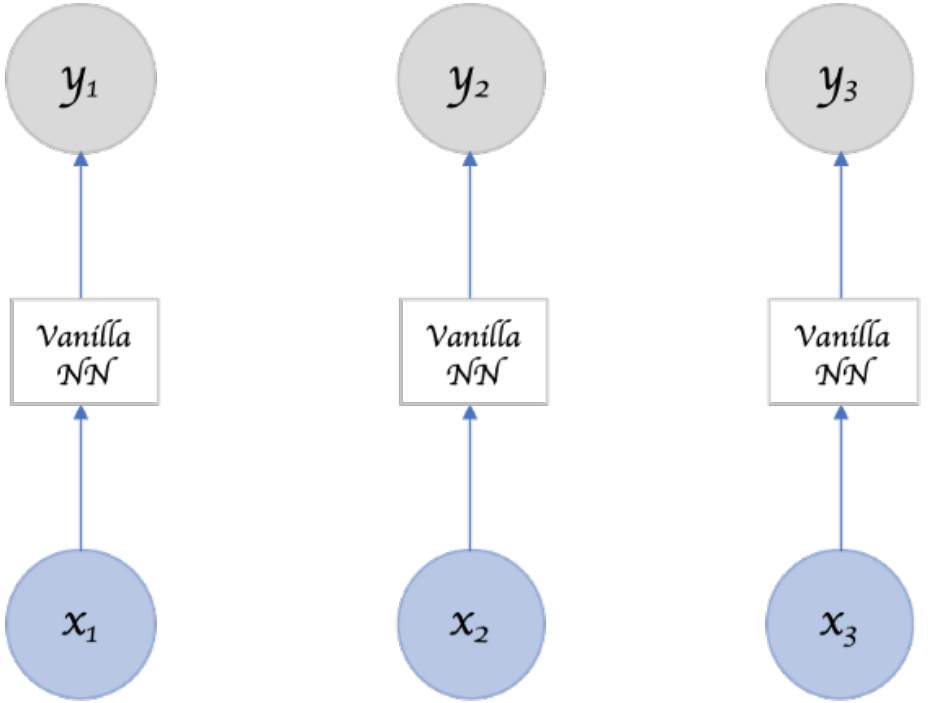
(a) 全连接层



(b) 卷积层

前馈网络

- 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。

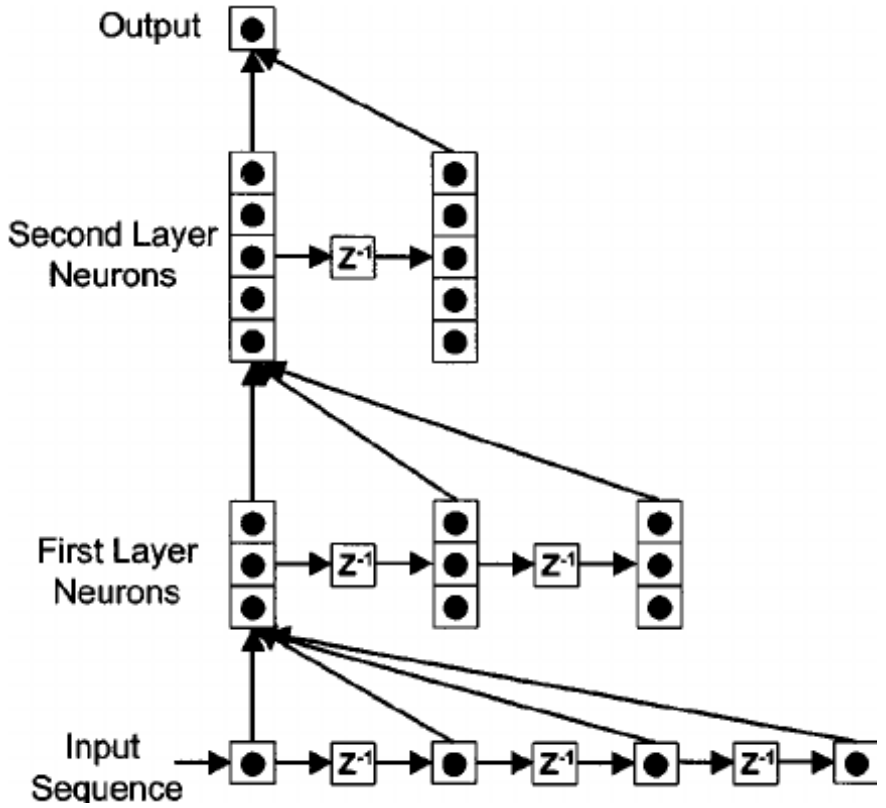


如何给网络增加记忆能力？

- 延时神经网络（Time Delay Neural Network，TDNN）
  - 建立一个额外的延时单元，用来存储网络的历史信息（可以包括输入、输出、隐状态等）

$$\boldsymbol{h}_t^{(l)} = f\left(\boldsymbol{h}_t^{(l-1)}, \boldsymbol{h}_{t-1}^{(l-1)}, \cdots, \boldsymbol{h}_{t-K}^{(l-1)}\right)$$

  - 这样，前馈网络就具有了短期记忆的能力。



https://www.researchgate.net/publication/12314435_Neural_system_identification_model_of_human_sound_localization

序列数据

- 在时间t观察到x_t，那么得到T个不独立的随机变量：

$$(x_1, \ldots x_T) \sim p(\mathbf{x})$$

- 使用条件概率展开：

$$p(\mathbf{x}) = p(x_1) \cdot p(x_2 \mid x_1) \cdot p(x_3 \mid x_1, x_2) \cdot \ldots p(x_T \mid x_1, \ldots x_{T-1})$$

$$p(\mathbf{x}) = p(x_T) \cdot p(x_{T-1} \mid x_T) \cdot p(x_{T-2} \mid x_{T-1}, x_T) \cdot \ldots p(x_1 \mid x_2, \ldots x_T)$$

序列建模

- 方案1：马尔科夫假设
  - 假设当前数据只跟过去τ个数据点有关

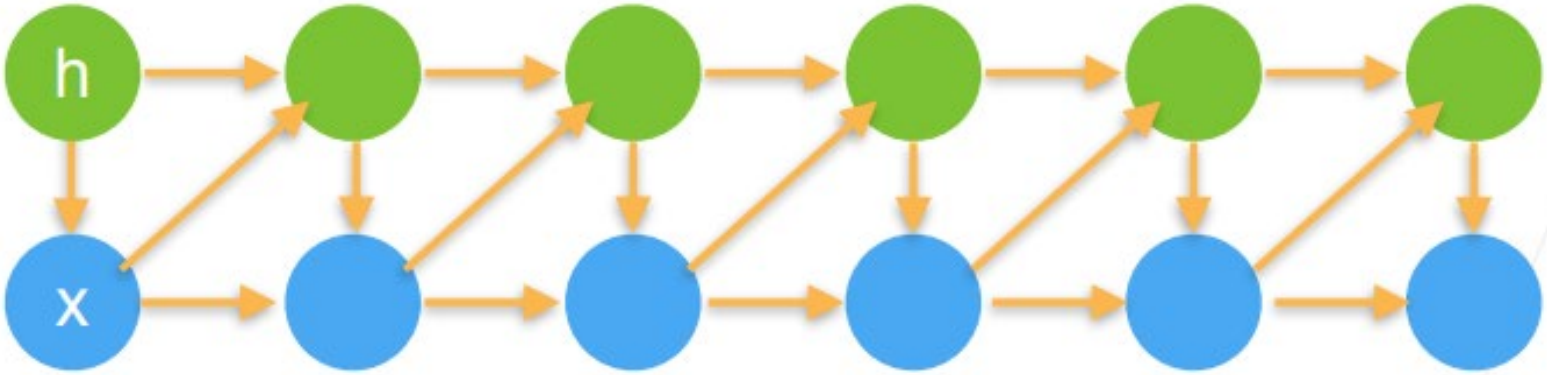
$$p(x_t \mid x_1, \ldots x_{t-1}) = p(x_t \mid x_{t-\tau}, \ldots x_{t-1}) = p(x_t \mid f(x_{t-\tau}, \ldots x_{t-1}))$$

在过去τ个数据上建模，例如在过去的数据上建立一个MLP模型

序列建模

- 方案2：潜变量模型（Latent Variable Model）
  - 引入潜变量h_t来表示过去的信息：$h_t = f(x_1, \ldots x_{t-1})$

这样：$x_t = p(x_t | h_t)$

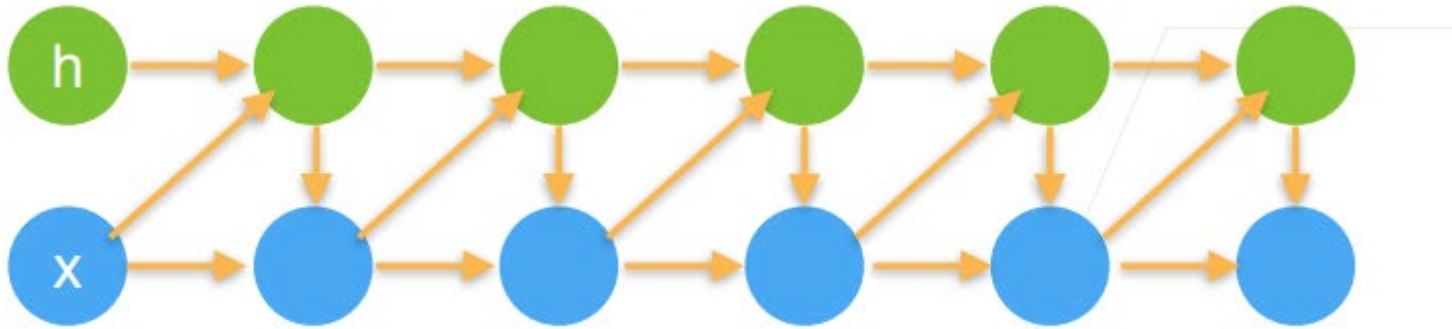


序列建模

- 潜变量自回归模型
  - 使用潜变量h_t总结过去的信息

$$p(h_t \mid h_{t-1}, x_{t-1})$$

$$p(x_t \mid h_t, x_{t-1})$$



序列建模

- 自回归模型（Autoregressive Model，AR）
  - 一类时间序列模型，用变量y_t的历史信息来预测自己

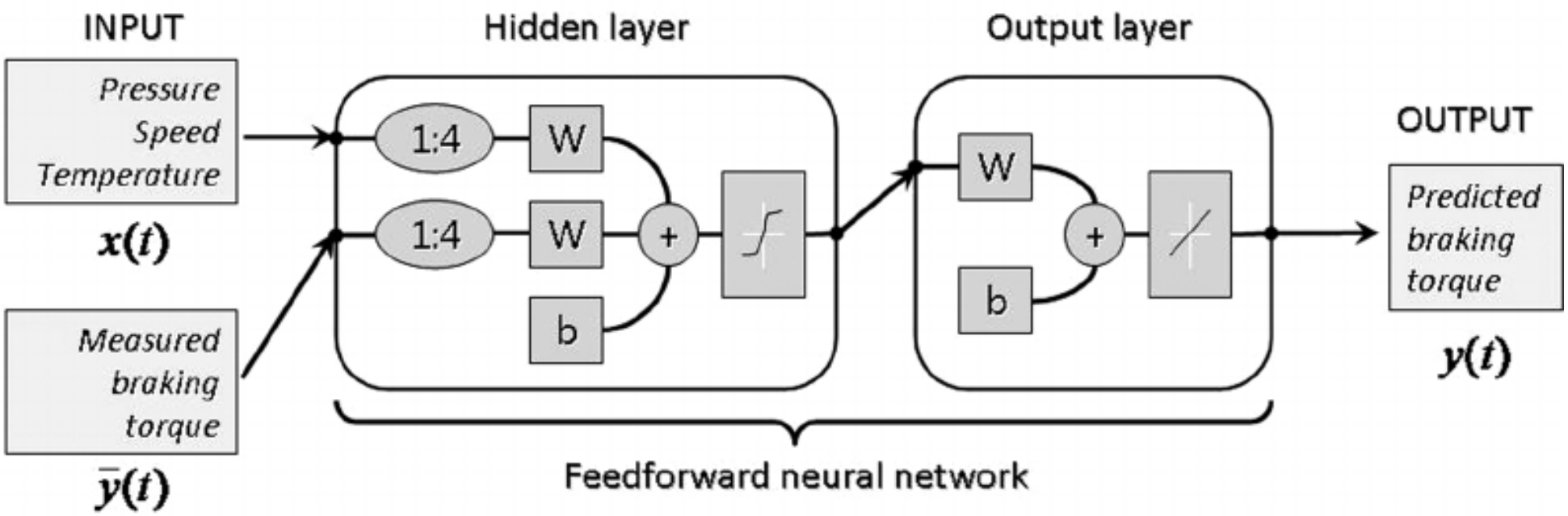
$$\boldsymbol{y}_t = w_0 + \sum_{k=1}^{K} w_k \boldsymbol{y}_{t-k} + \epsilon_t$$

  - $\epsilon_t \sim N(0, \sigma^2)$为第t个时刻的噪声
- 有外部输入的非线性自回归模型（Nonlinear Autoregressive with Exogenous Inputs Model，NARX）

$$\boldsymbol{y}_t = f(\boldsymbol{x}_t, \boldsymbol{x}_{t-1}, \cdots, \boldsymbol{x}_{t-K_x}, \boldsymbol{y}_{t-1}, \boldsymbol{y}_{t-2}, \cdots, \boldsymbol{y}_{t-K_y})$$

  - 其中 $f(\cdot)$ 表示非线性函数，可以是一个前馈网络，K_x 和 K_y 为超参数.

非线性自回归模型



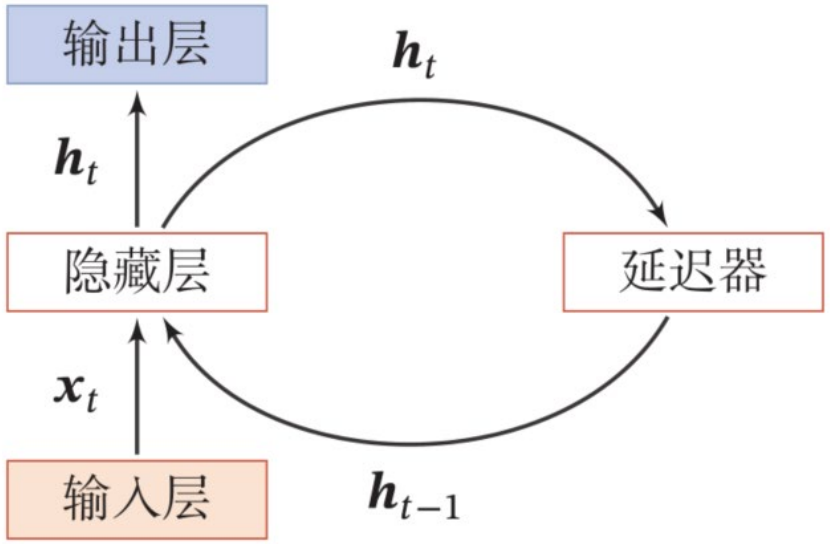
https://www.researchgate.net/publication/234052442_Braking_torque_control_using_reccurent_neural_networks

循环神经网络（Recurrent Neural Network，RNN）

- 循环神经网络通过使用带自反馈的神经元，能够处理任意长度的时序数据。

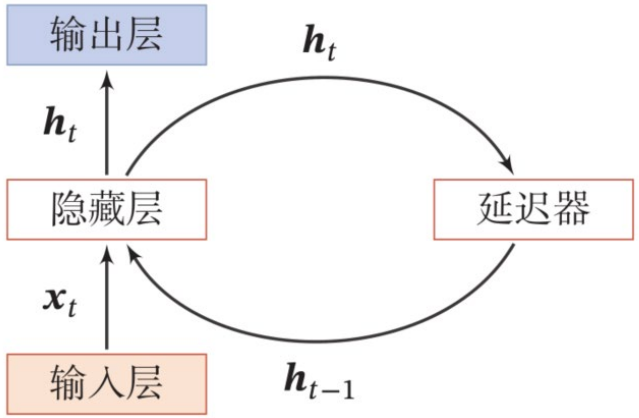
$$\boldsymbol{h}_t = f(\boldsymbol{h}_{t-1}, \boldsymbol{x}_t)$$

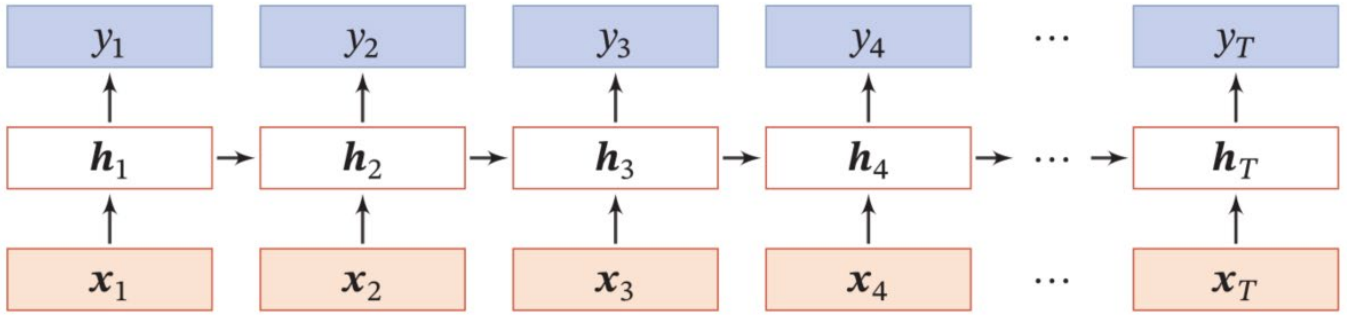
活性值
状态



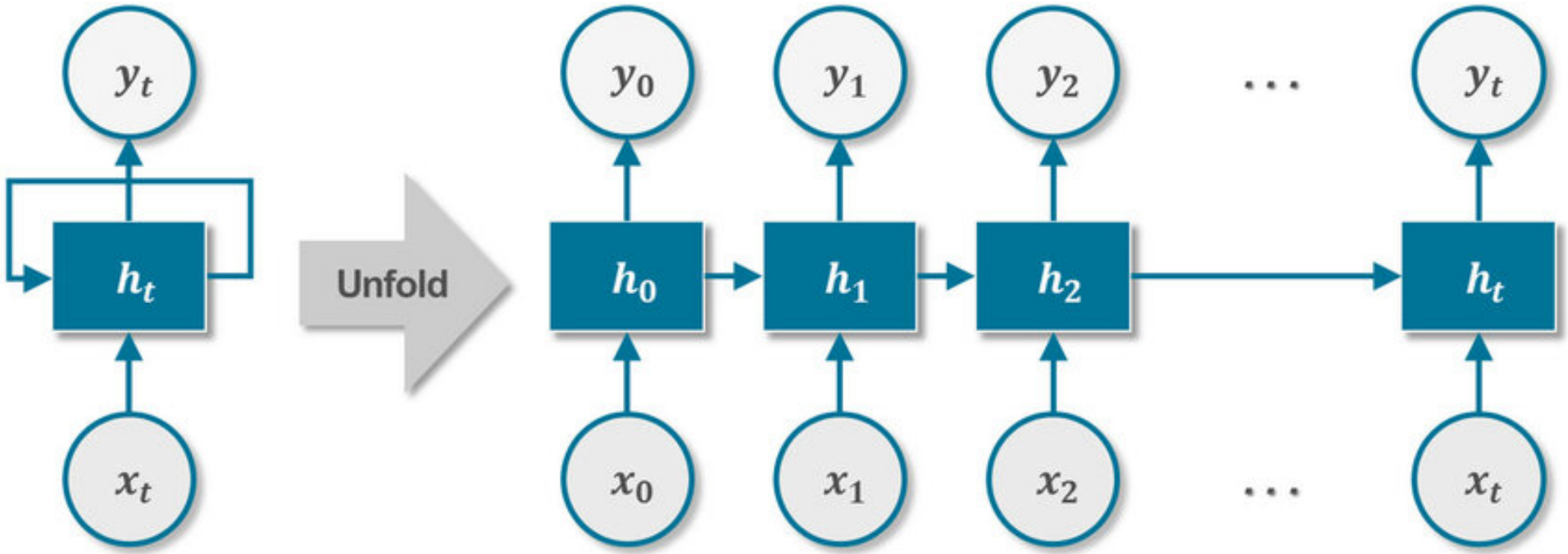
- 循环神经网络比前馈神经网络更加符合生物神经网络的结构。
- 循环神经网络已经被广泛应用在语音识别、语言模型以及自然语言生成等任务上

按时间展开

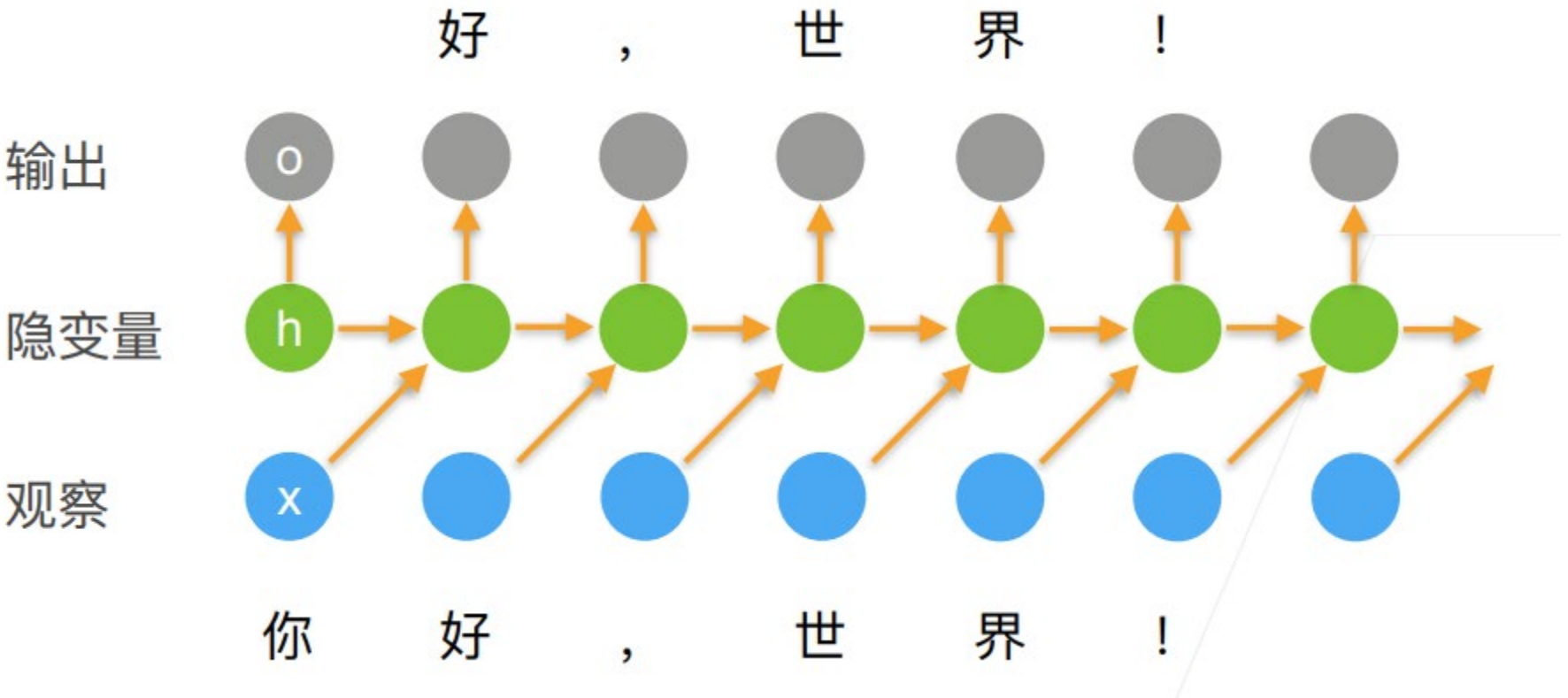

输出层
h_t
h_t
隐藏层
延迟器
x_t
输入层
h_{t-1}


y_1
y_2
y_3
y_4
y_T
h_1
h_2
h_3
h_4
h_T
x_1
x_2
x_3
x_4
x_T

简单循环网络（Simple Recurrent Network， SRN）



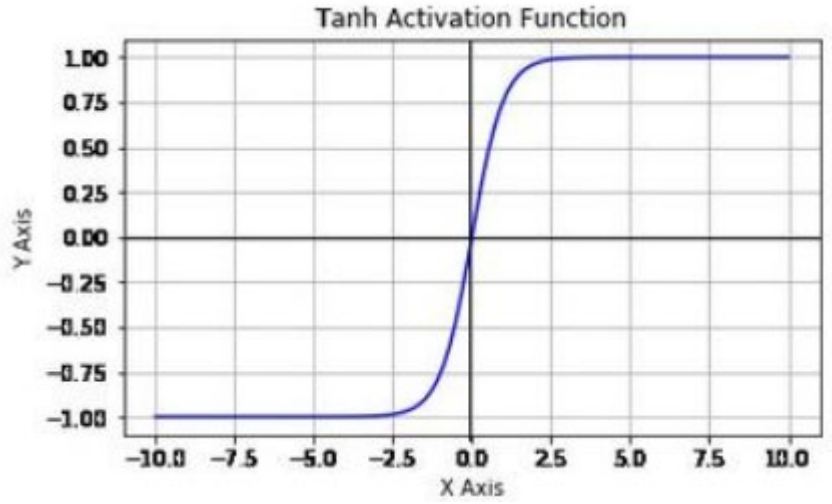
简单循环网络（Simple Recurrent Network， SRN）



循环神经网络中的激活函数

- RNN通常可使用tanh激活函数：

$$f(x) = tanh(x) = \frac{e^x - e^{-x}}{e^x + e^{-x}}$$



- 也可以使用ReLU激活函数（但是不良的参数初始化容易导致梯度爆炸）。

简单循环网络（Simple Recurrent Network， SRN）

- 状态更新：

$$\boldsymbol{h}_t = f(\boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{b})$$

- 一个完全连接的循环网络是任何非线性动力系统的近似器。

定理 6.1 – 循环神经网络的通用近似定理 [Haykin, 2009]： 如果一个完全连接的循环神经网络有足够数量的 sigmoid 型隐藏神经元，它可以以任意的准确率去近似任何一个非线性动力系统

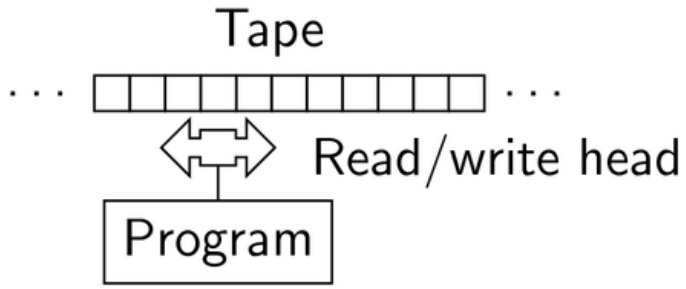
$$\boldsymbol{s}_t = g(\boldsymbol{s}_{t-1}, \boldsymbol{x}_t), \tag{6.10}$$

$$\boldsymbol{y}_t = o(\boldsymbol{s}_t), \tag{6.11}$$

其中 $\boldsymbol{s}_t$ 为每个时刻的隐状态，$\boldsymbol{x}_t$ 是外部输入，$g(\cdot)$ 是可测的状态转换函数，$o(\cdot)$ 是连续输出函数，并且对状态空间的紧致性没有限制.

图灵完备

- 图灵完备（Turing Completeness）是指一种数据操作规则，比如一种计算机编程语言，可以实现图灵机的所有功能，解决所有的可计算问题。



定理 6.2 – 图灵完备 [Siegelmann et al., 1991]： 所有的图灵机都可以被一个由使用 Sigmoid 型激活函数的神经元构成的全连接循环网络来进行模拟.

- 一个完全连接的循环神经网络可以近似解决所有的可计算问题。

循环神经网络

- 作用
  - 输入-输出映射
    - 机器学习模型（本节主要关注这种情况）
  - 存储器
    - 联想记忆模型

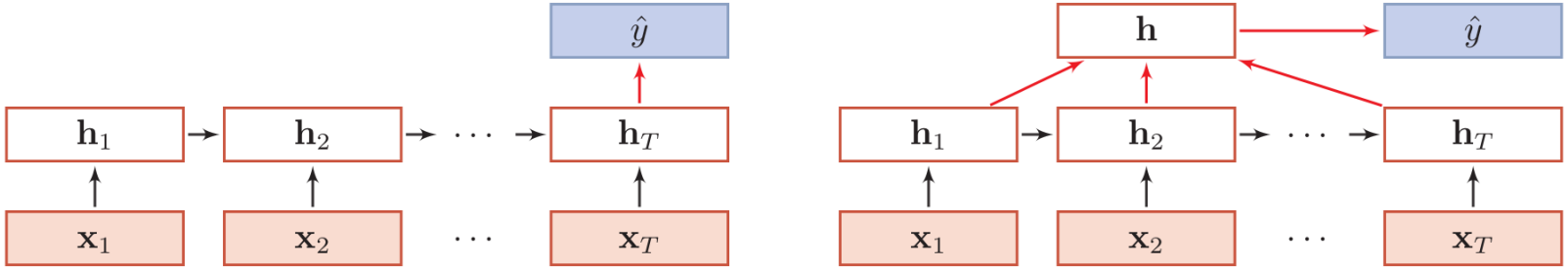
应用到机器学习

应用到机器学习

- 情况1：序列到类别
- 情况2：同步的序列到序列模式
- 情况3：异步的序列到序列模式

应用到机器学习

序列到类别



(a) 正常模式　　(b) 按时间进行平均采样模式

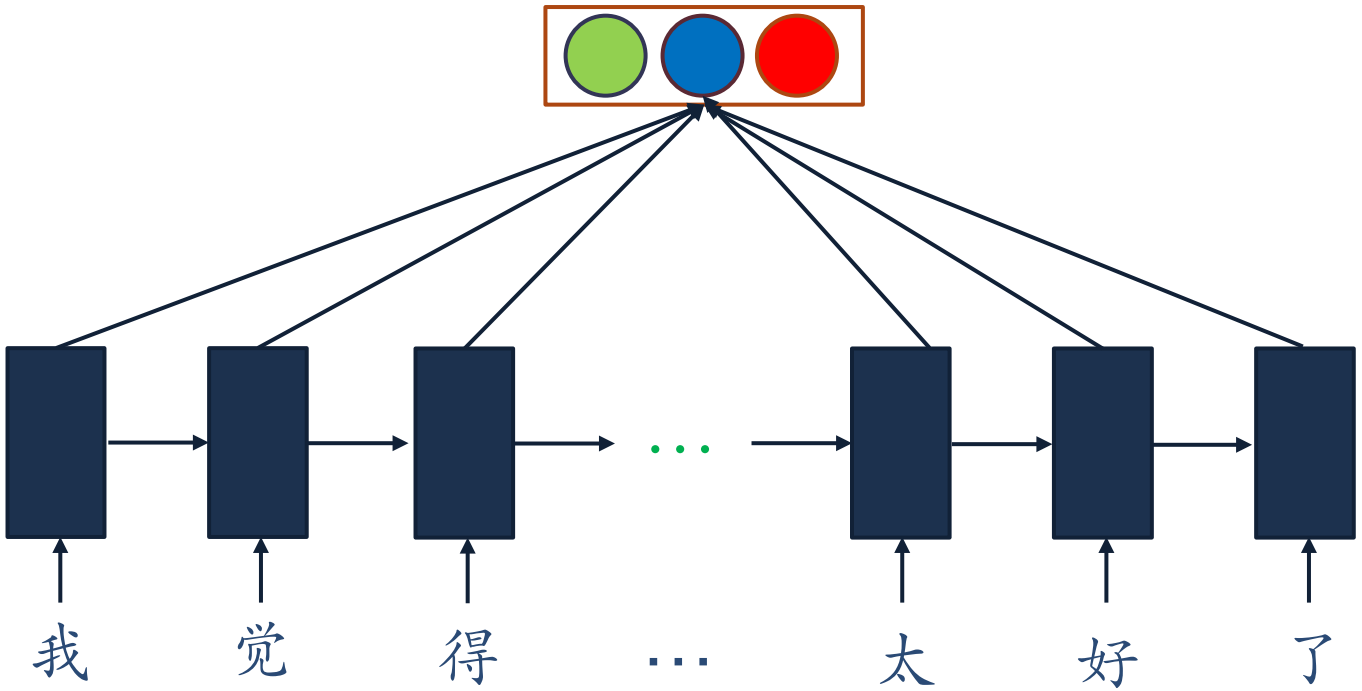
序列到类别

- 情感分类

带着愉悦的心情看了这部电影 — Positive (正面)

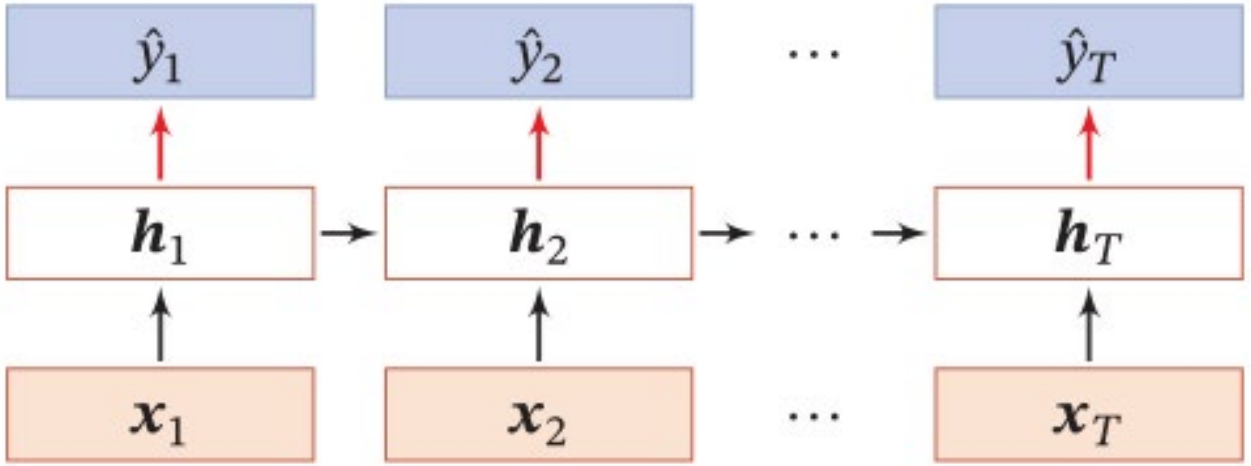
这部电影太糟了 — Negative (负面)

这部电影很棒 — Positive (正面)



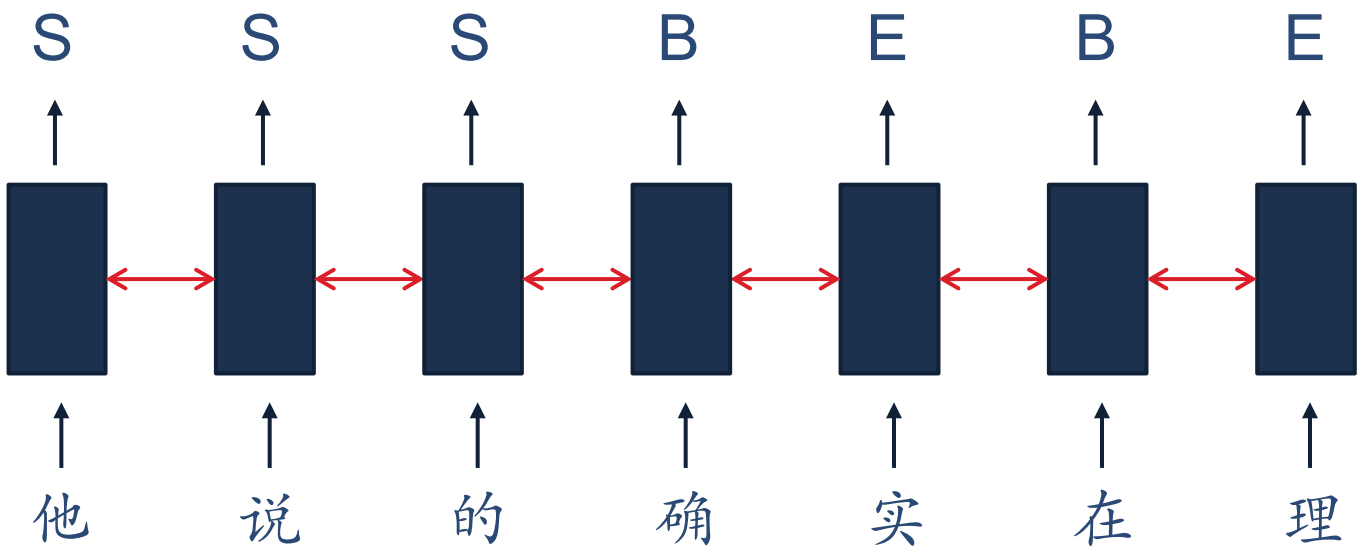
应用到机器学习

- 同步的序列到序列模式



同步的序列到序列模式

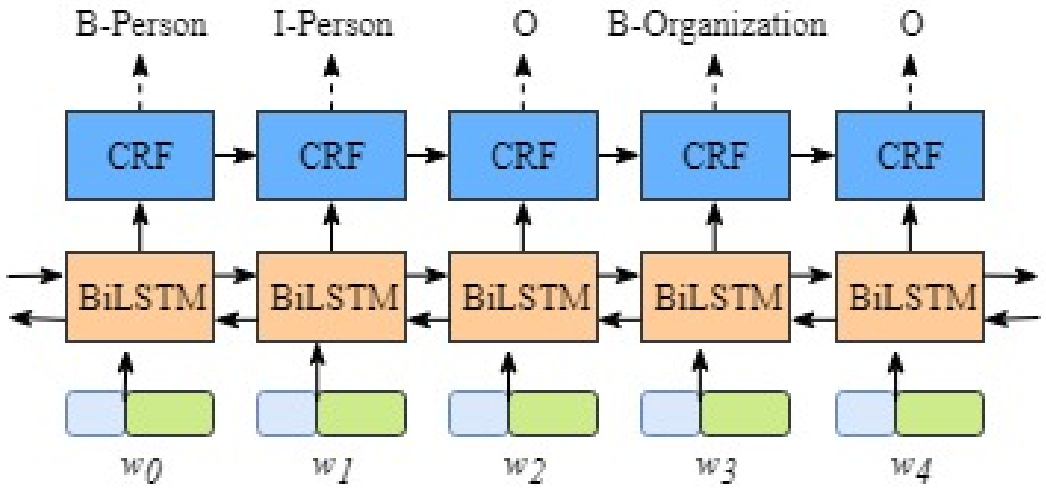
- 中文分词
- 英文词性标注
- Token分析



同步的序列到序列模式

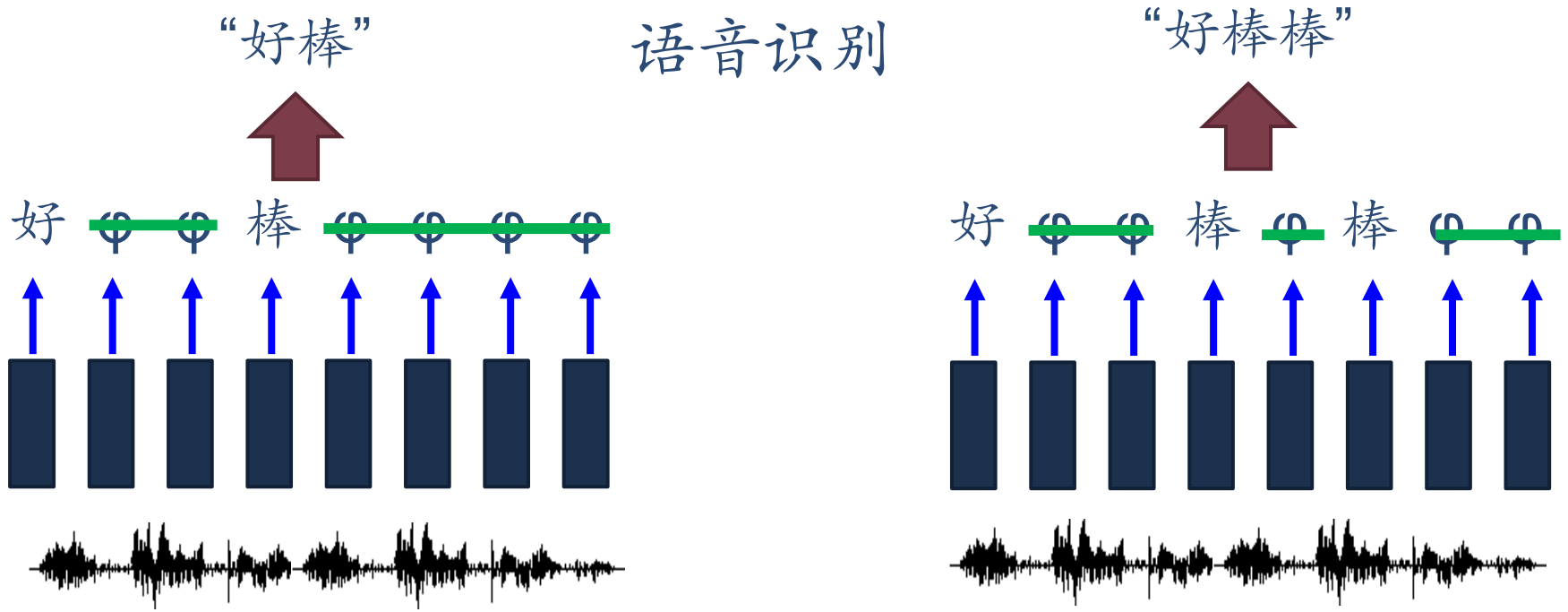
- 信息抽取(Information Extraction，IE)
  - 从无结构的文本中抽取结构化的信息，形成知识

小米创始人雷军表示，该公司2015年营收达到780亿元人民币，较2014年的743亿元人民币增长了5%。



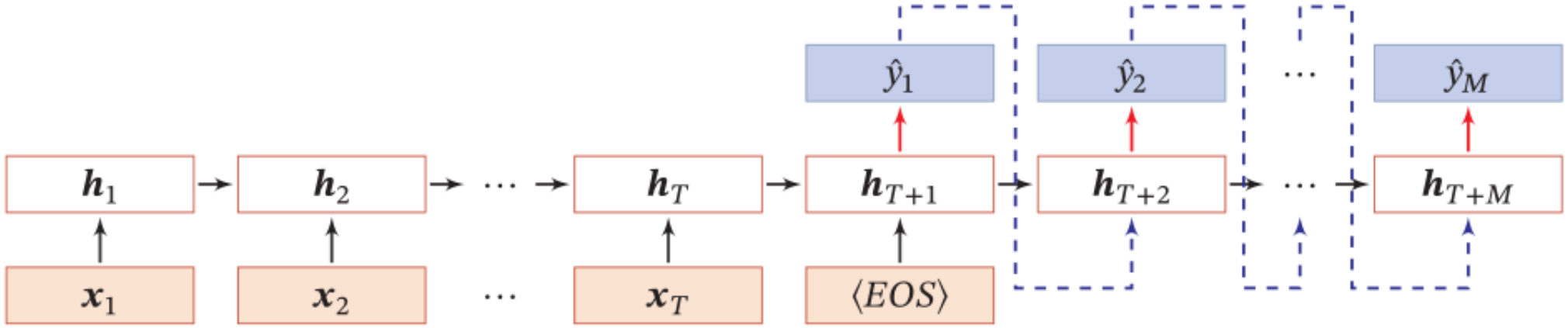
同步的序列到序列模式

- 语音识别——连接时序分类
  - Connectionist Temporal Classification (CTC) [Alex Graves, ICML'06][Alex Graves, ICML'14][Haşim Sak, Interspeech'15][Jie Li, Interspeech'15][Andrew Senior, ASRU'15]



应用到机器学习

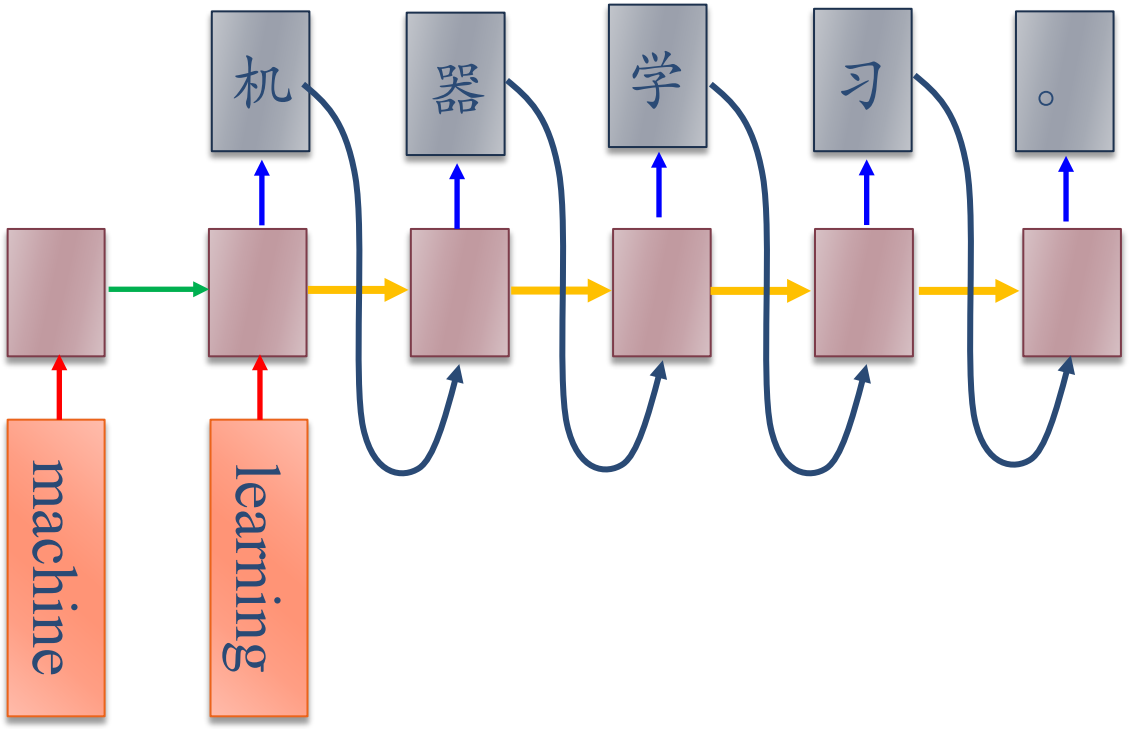
- 异步的序列到序列模式



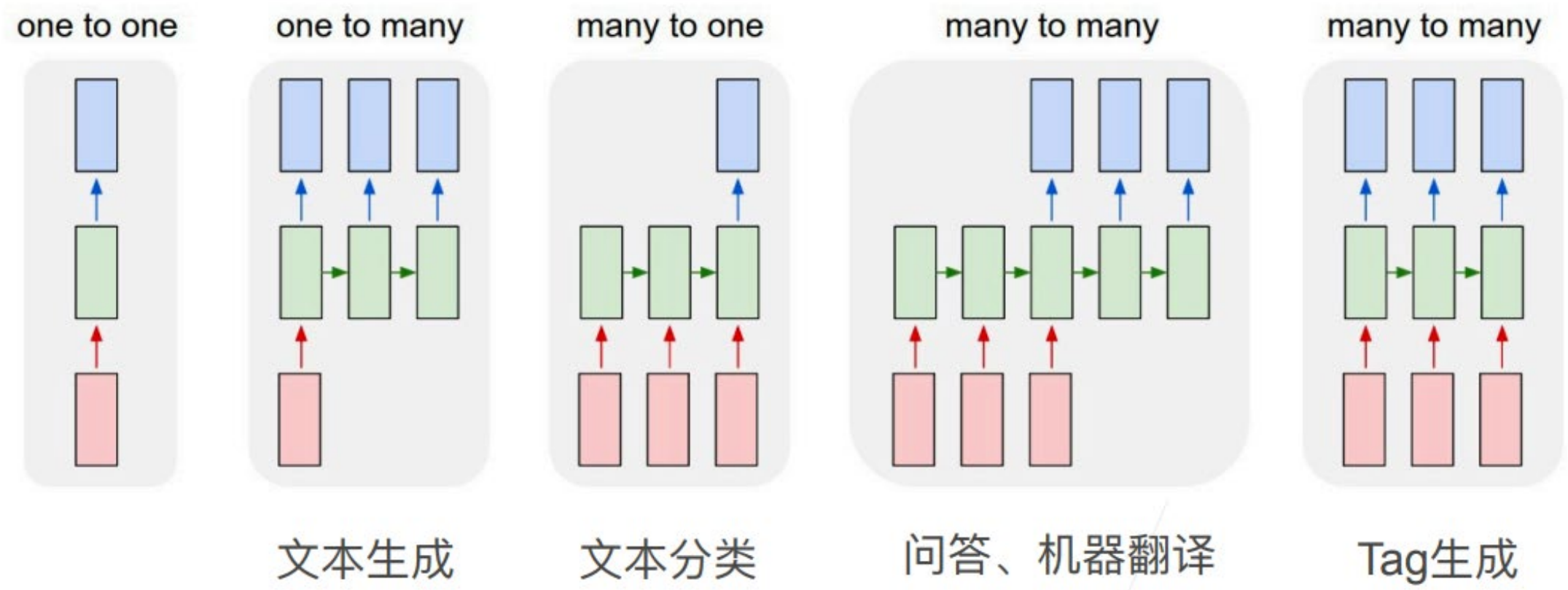
Encoder (编码器)　　Decoder (解码器)

异步的序列到序列模式

- 机器翻译



不同模式的总结


one to one
one to many
many to one
many to many
many to many
文本生成
文本分类
问答、机器翻译
Tag生成

参数学习

- 机器学习
  - 给定一个训练样本(x,y)，其中
    - $x=(x_1,\ldots,x_T)$为长度是T的输入序列，
    - $y=(y_1,\ldots,y_T)$是长度为T的标签序列。

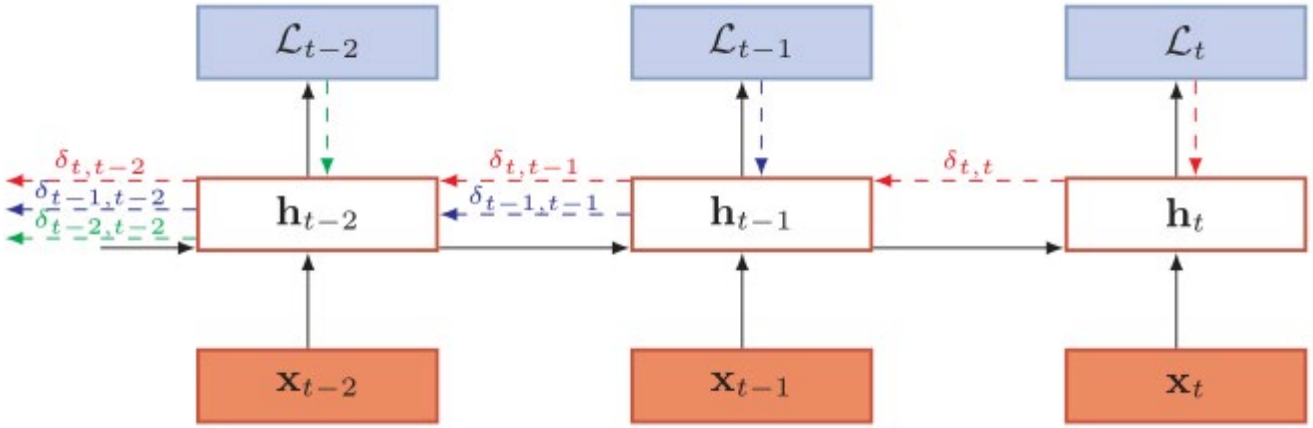
- 时刻t的瞬时损失函数为 $\mathcal{L}_t = \mathcal{L}(\mathbf{y}_t, g(\mathbf{h}_t)),$

- 总损失函数 $\mathcal{L} = \sum_{t=1}^{T} \mathcal{L}_t.$

梯度

- 随时间反向传播算法

$$\mathbf{h}_{t+1} = f(\mathbf{z}_{t+1}) = f(U\mathbf{h}_t + W\mathbf{x}_{t+1} + \mathbf{b})$$



$$\frac{\partial \mathcal{L}}{\partial U} = \sum_{t=1}^{T}\sum_{k=1}^{t} \delta_{t,k}\mathbf{h}_{k-1}^{\mathrm{T}}$$

$$\delta_{t,k} = \prod_{\tau=k}^{t-1}\Big(\mathrm{diag}(f'(\mathbf{z}_\tau))U^{\mathrm{T}}\Big)\delta_{t,t}$$

$\delta_{t,k}$为第t时刻的损失对第k步隐藏神经元的净输入z_k的导数

梯度消失/爆炸

- 梯度

$$\frac{\partial \mathcal{L}}{\partial U} = \sum_{t=1}^{T} \sum_{k=1}^{t} \delta_{t,k} \mathbf{h}_{k-1}^{\mathrm{T}}$$

- 其中

$$\delta_{t,k} = \prod_{\tau=k}^{t-1} \left(\frac{\mathrm{diag}(f'(\mathbf{z}_\tau)) U^{\mathrm{T}}}{\lambda} \right) \delta_{t,t}$$

$$\delta_{t,k} \cong \gamma^{t-k} \delta_{t,t}$$

由于梯度爆炸或消失问题，实际上只能学习到短周期的依赖关系。这就是所谓的长程依赖问题。

长程依赖问题

- 循环神经网络在时间维度上非常深！
  - 梯度消失或梯度爆炸

- 如何改进？
  - 梯度爆炸问题
    - 权重衰减——如添加正则化项
    - 梯度截断——设置梯度上限θ

$$\mathbf{g} \leftarrow \min\left(1, \frac{\theta}{\|\mathbf{g}\|}\right)\mathbf{g}$$

  - 梯度消失问题
    - 改进模型

长程依赖问题

- 改进方法
  - 循环边改为线性依赖关系

$$\mathbf{h}_t = \mathbf{h}_{t-1} + g(\mathbf{x}_t; \theta),$$

$$\boldsymbol{z}_t = \boldsymbol{U}\boldsymbol{h}_{t-1} + \boldsymbol{W}\boldsymbol{x}_t + \boldsymbol{b},$$

$$\boldsymbol{h}_t = f(\boldsymbol{z}_t),$$

$$\delta_{t,k} = \prod_{\tau=k}^{t-1} \left(\mathrm{diag}(f'(\mathbf{z}_\tau))U^{\mathrm{T}} \right) \delta_{t,t}$$

  - 增加非线性

$$\mathbf{h}_t = \mathbf{h}_{t-1} + g(\mathbf{x}_t, \mathbf{h}_{t-1}; \theta),$$

残差网络?

GRU和LSTM

门控机制

- 长程依赖问题

- 门控机制
  - 控制信息的累积速度，包括有选择地加入新的信息，以及有选择地遗忘之前累积的信息。

- 基于门控的循环神经网络（Gated RNN）
  - 门控循环单元（GRU）
  - 长短期记忆神经网络（LSTM）

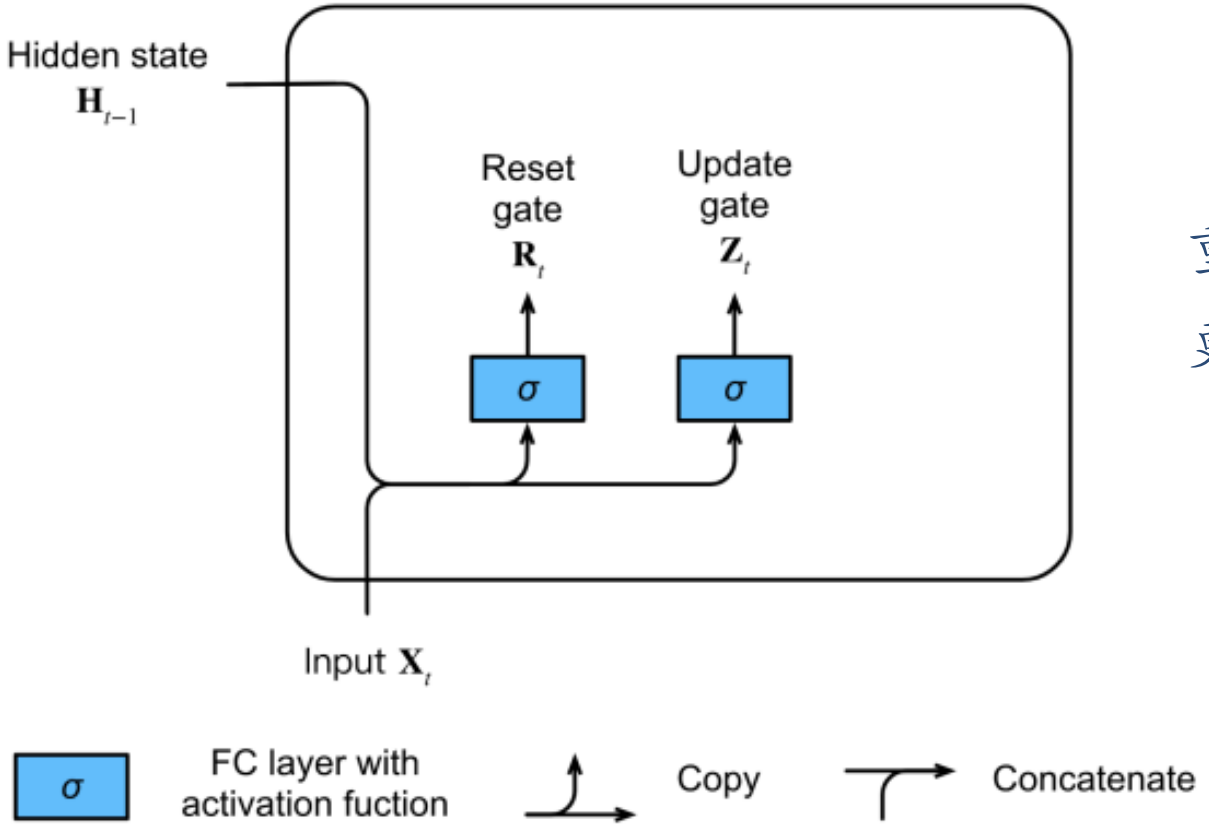
门控循环单元（GRU）

门控循环单元（GRU）

- 不是每个观测值都同等重要

- 只需要记住重要观测：
  - 能关注的机制（更新门）
  - 能遗忘的机制（重置门）

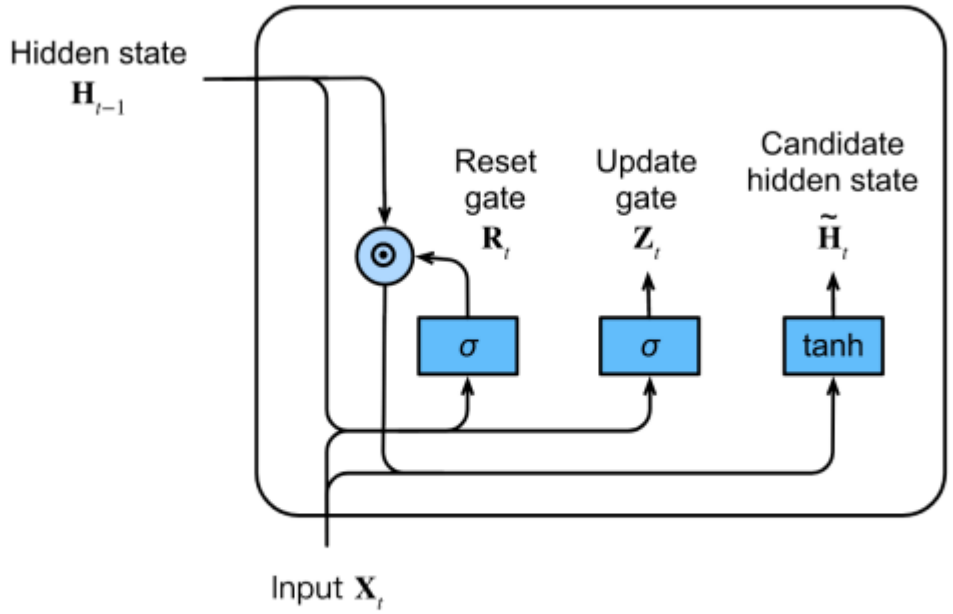
门控循环单元（GRU）

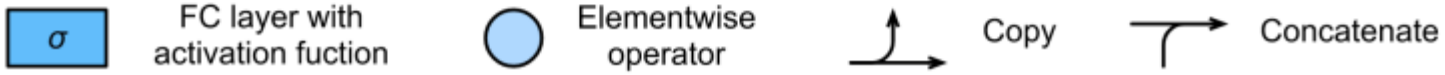


重置门 $\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hr} + \boldsymbol{b}_r),$

更新门 $\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hz} + \boldsymbol{b}_z)$

门控循环单元（GRU）

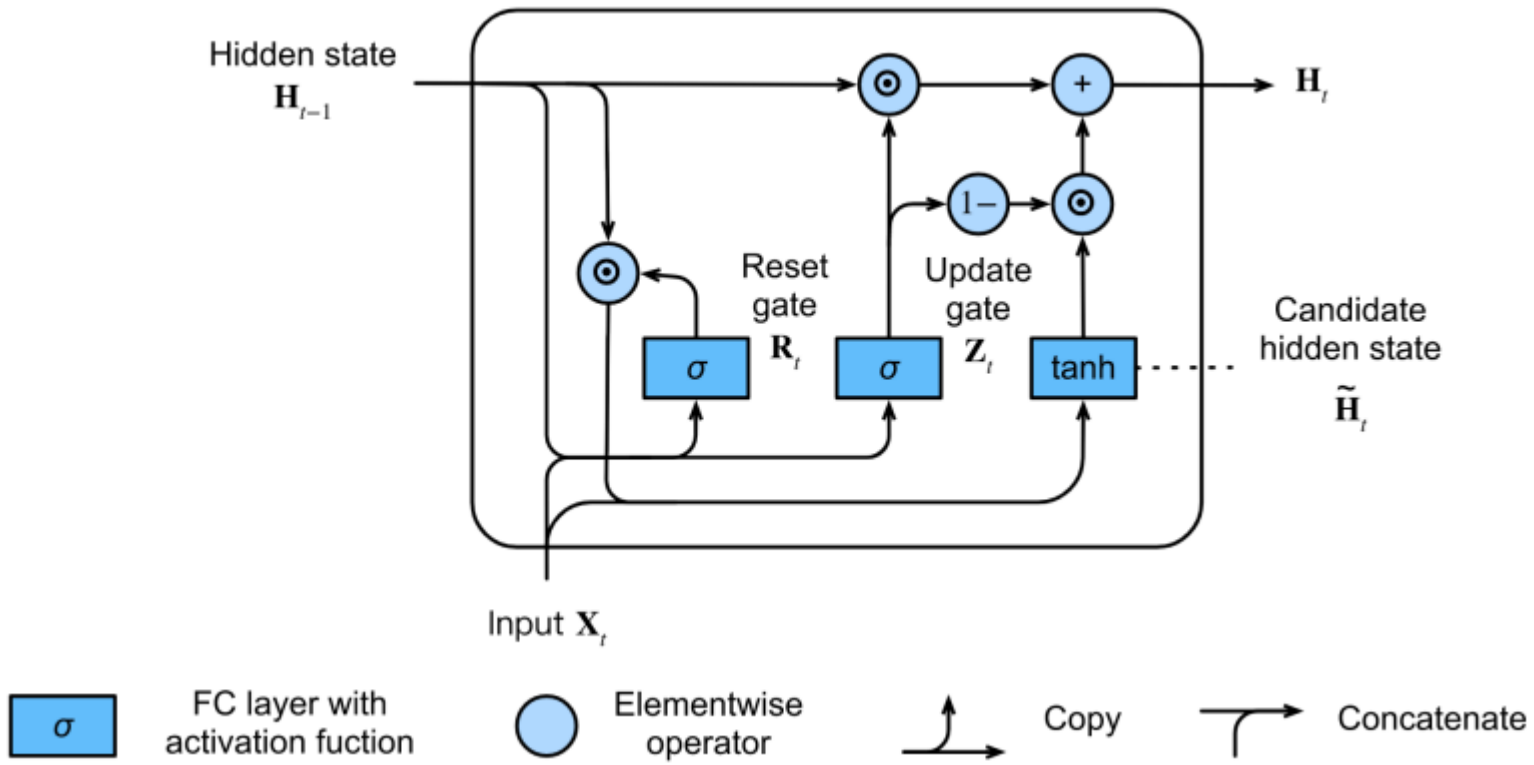




候选隐状态

$$\tilde{\boldsymbol{H}}_t = \tanh(\boldsymbol{X}_t \boldsymbol{W}_{xh} + (\boldsymbol{R}_t \odot \boldsymbol{H}_{t-1}) \boldsymbol{W}_{hh} + \boldsymbol{b}_h)$$

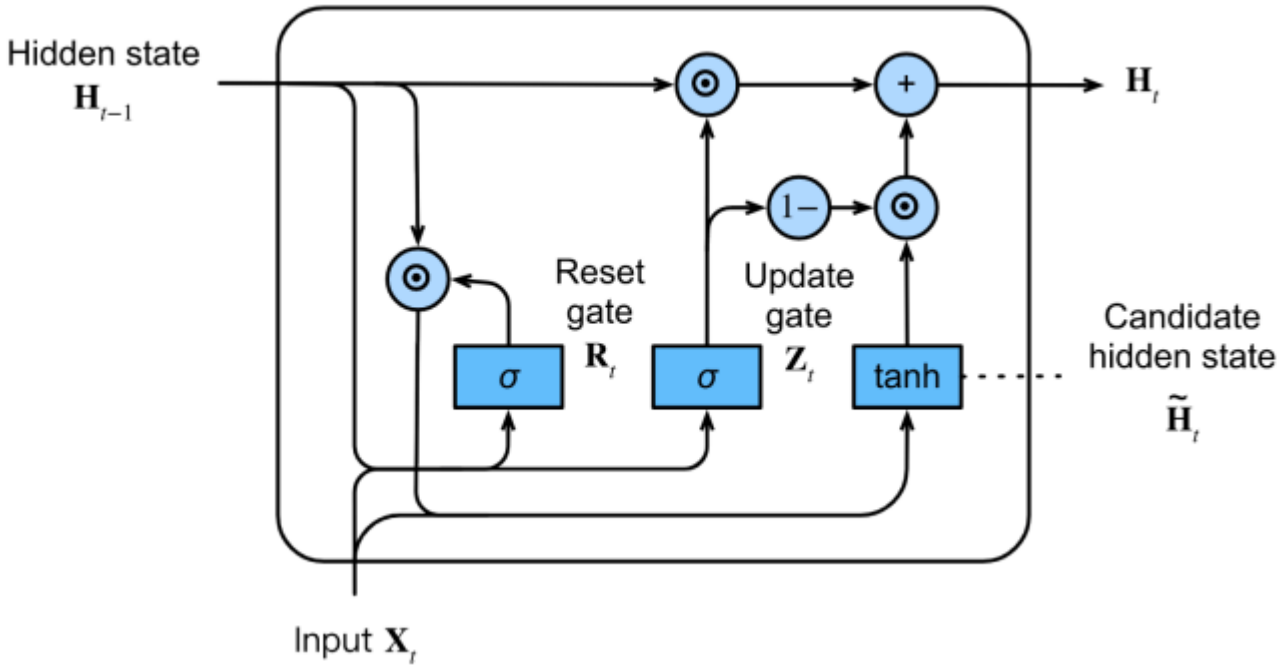
门控循环单元（GRU）



隐状态

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{H}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \tilde{\boldsymbol{H}}_t$$

门控循环单元（GRU）



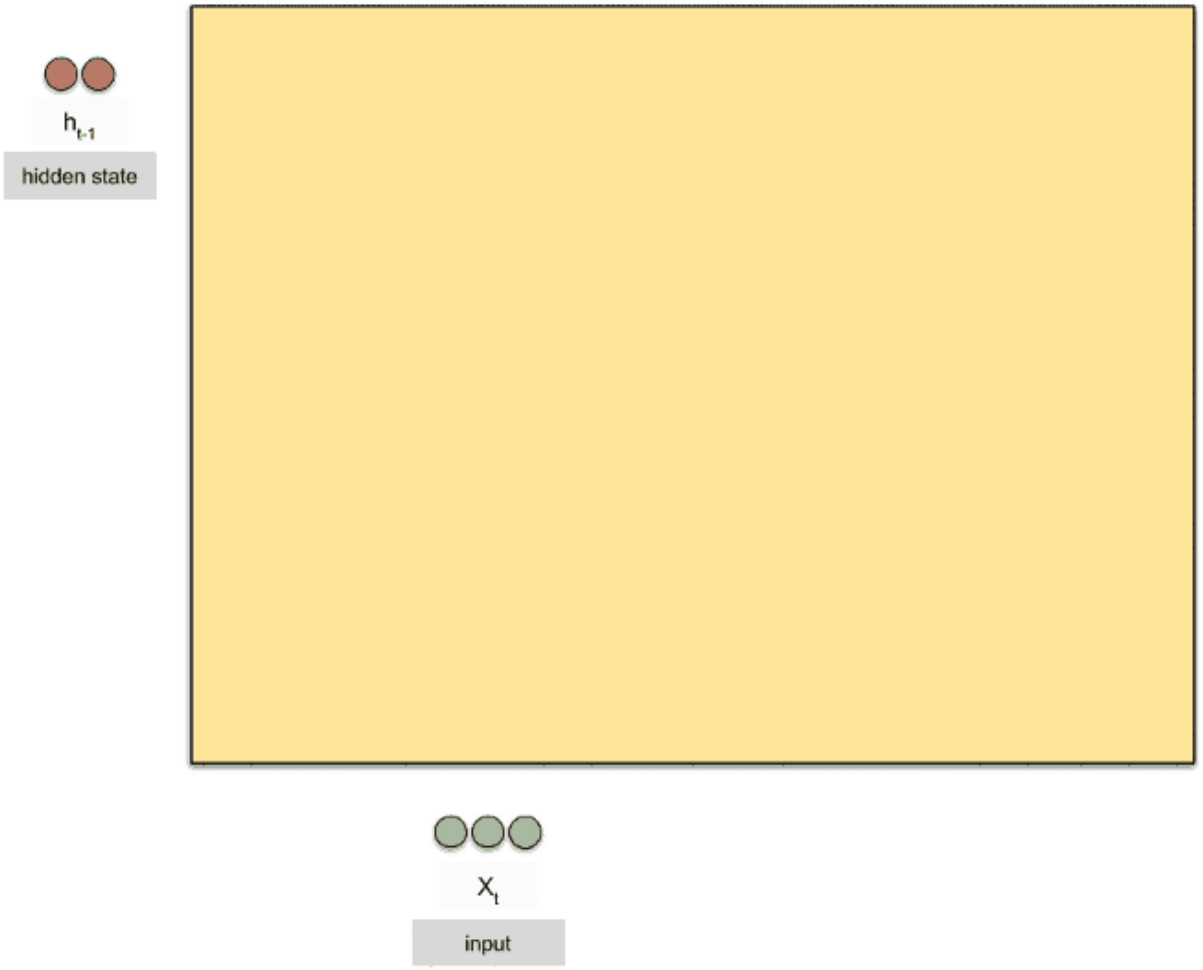
$$\boldsymbol{R}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xr} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hr} + \boldsymbol{b}_r),$$

$$\boldsymbol{Z}_t = \sigma(\boldsymbol{X}_t \boldsymbol{W}_{xz} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hz} + \boldsymbol{b}_z)$$

$$\tilde{\boldsymbol{H}}_t = \tanh(\boldsymbol{X}_t \boldsymbol{W}_{xh} + \left(\boldsymbol{R}_t \odot \boldsymbol{H}_{t-1}\right) \boldsymbol{W}_{hh} + \boldsymbol{b}_h)$$

$$\boldsymbol{H}_t = \boldsymbol{Z}_t \odot \boldsymbol{H}_{t-1} + (1 - \boldsymbol{Z}_t) \odot \tilde{\boldsymbol{H}}_t$$

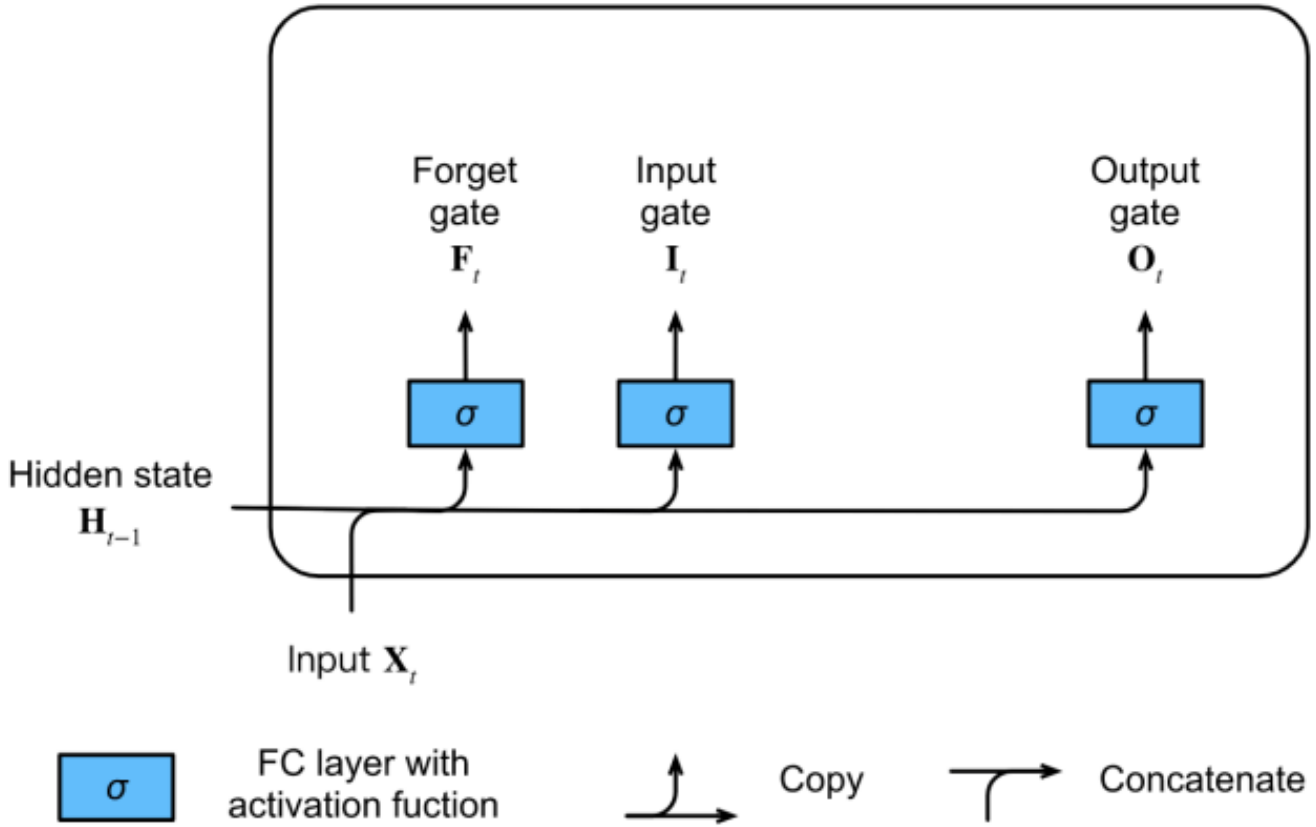
门控循环单元（GRU）



长短期记忆神经网络（LSTM）

- 遗忘门：将传递的信息减少
- 输入门：决定以多大比例忽略输入数据
- 输出门：决定以多大比例使用隐状态

长短期记忆神经网络（LSTM）

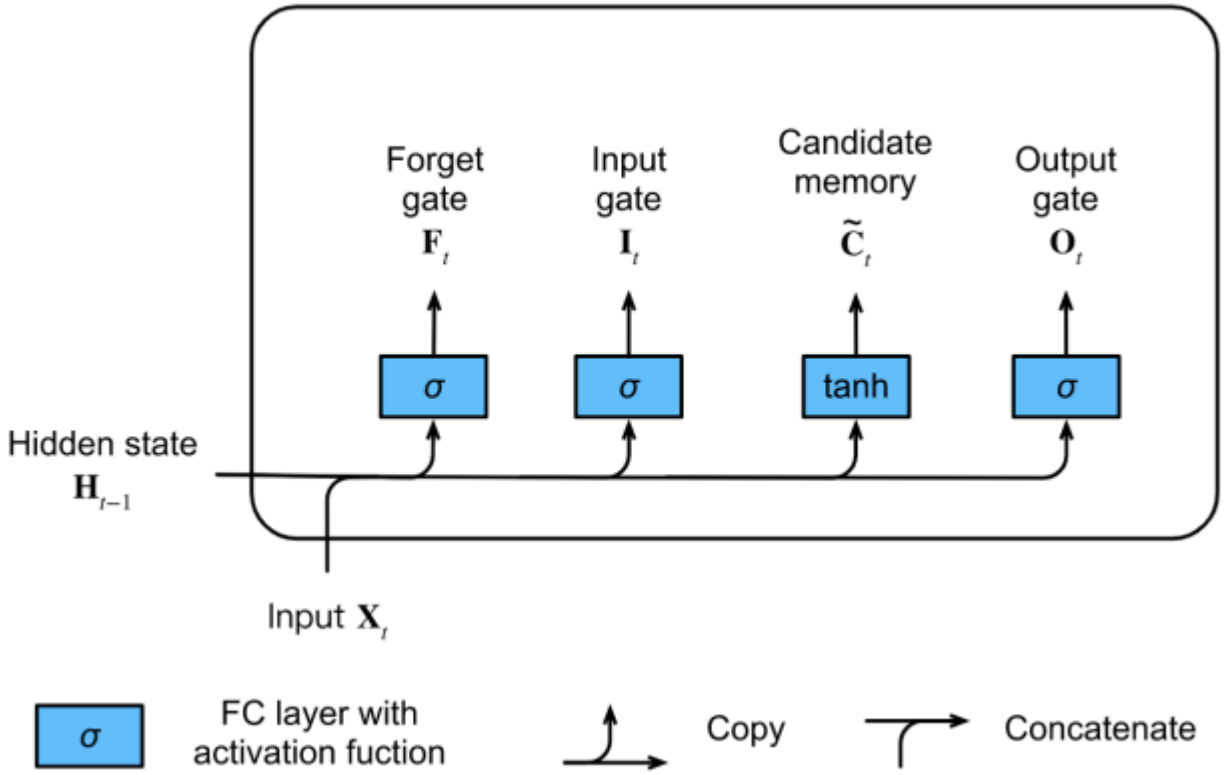


$$\boldsymbol{I}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hi} + \boldsymbol{b}_i)$$

$$\boldsymbol{F}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hf} + \boldsymbol{b}_f)$$

$$\boldsymbol{O}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xo} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{ho} + \boldsymbol{b}_o)$$

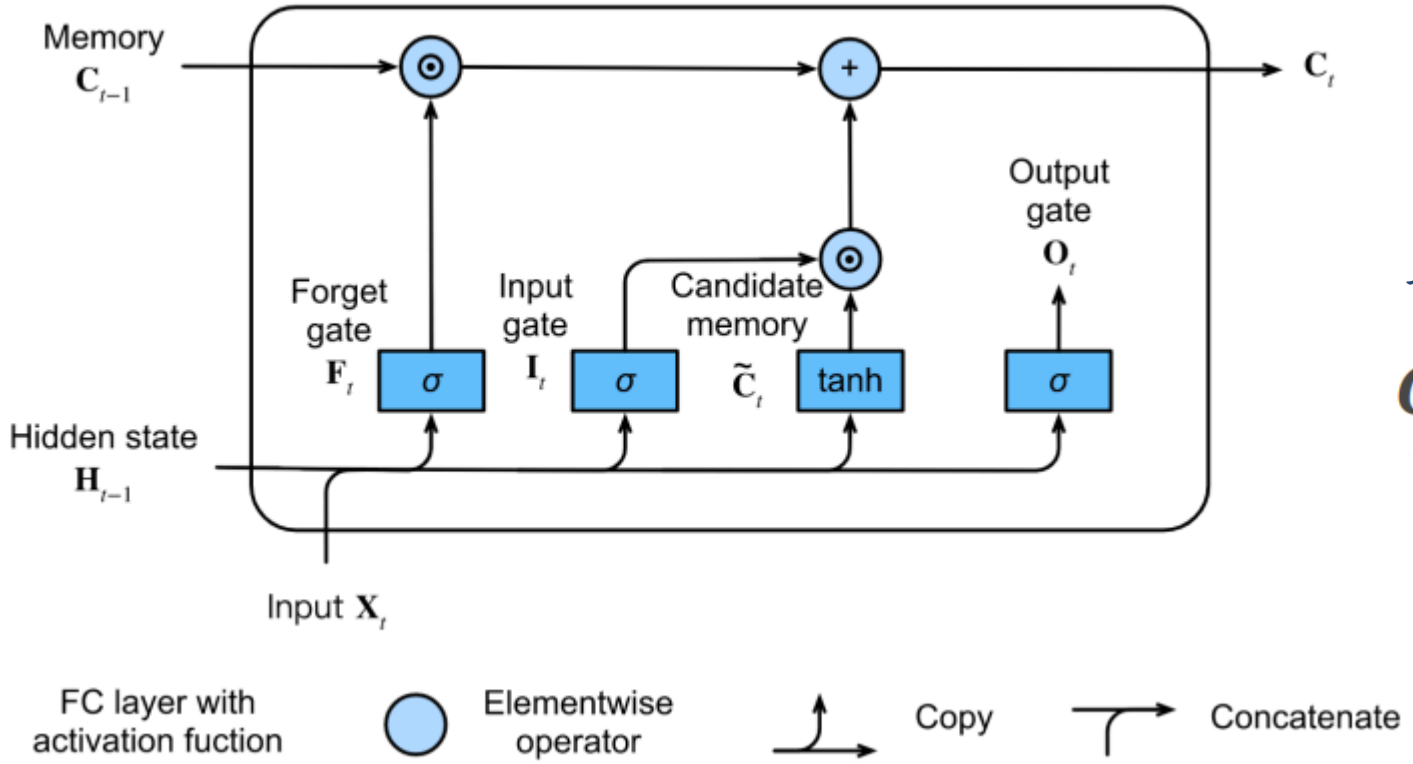
长短期记忆神经网络（LSTM）



候选记忆单元

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{X}_t \boldsymbol{W}_{xc} + \boldsymbol{H}_{t-1} \boldsymbol{W}_{hc} + \boldsymbol{b}_c)$$

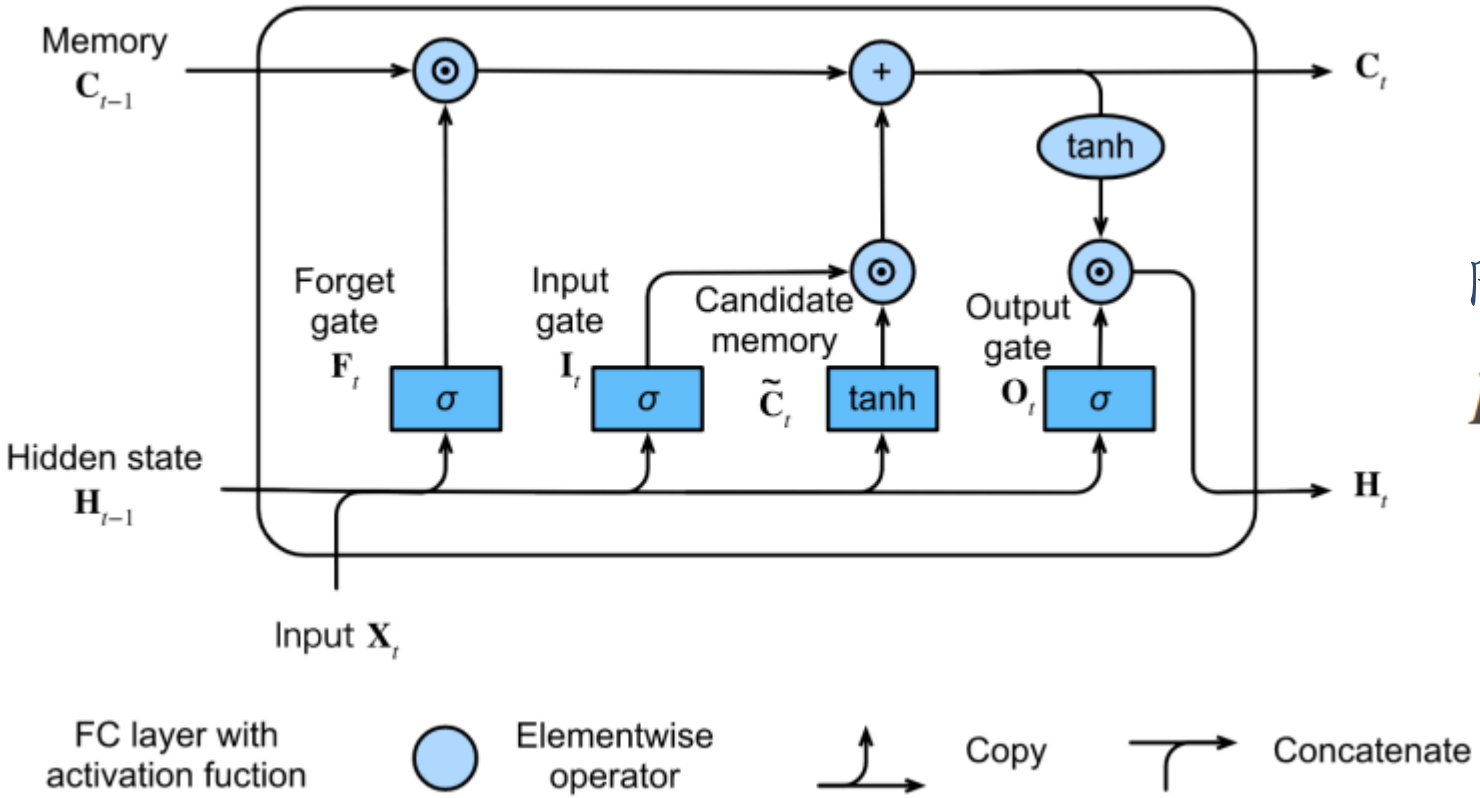
长短期记忆神经网络（LSTM）



记忆单元

$$\boldsymbol{C}_t = \boldsymbol{F}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \odot \tilde{\boldsymbol{C}}_t$$

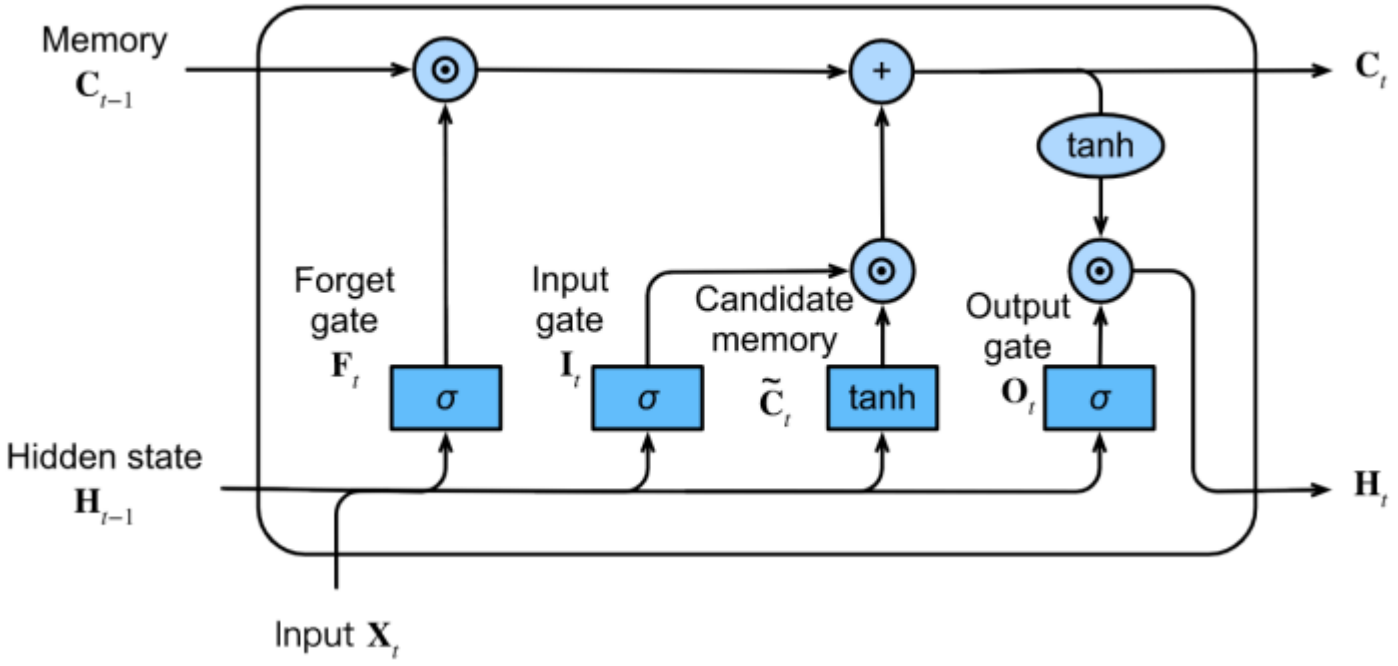
长短期记忆神经网络（LSTM）



隐状态

$$\boldsymbol{H}_t = \boldsymbol{O}_t \odot \tanh(\boldsymbol{C}_t)$$

长短期记忆神经网络（LSTM）



$$\boldsymbol{I}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xi} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hi} + \boldsymbol{b}_i)$$

$$\boldsymbol{F}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xf} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hf} + \boldsymbol{b}_f)$$

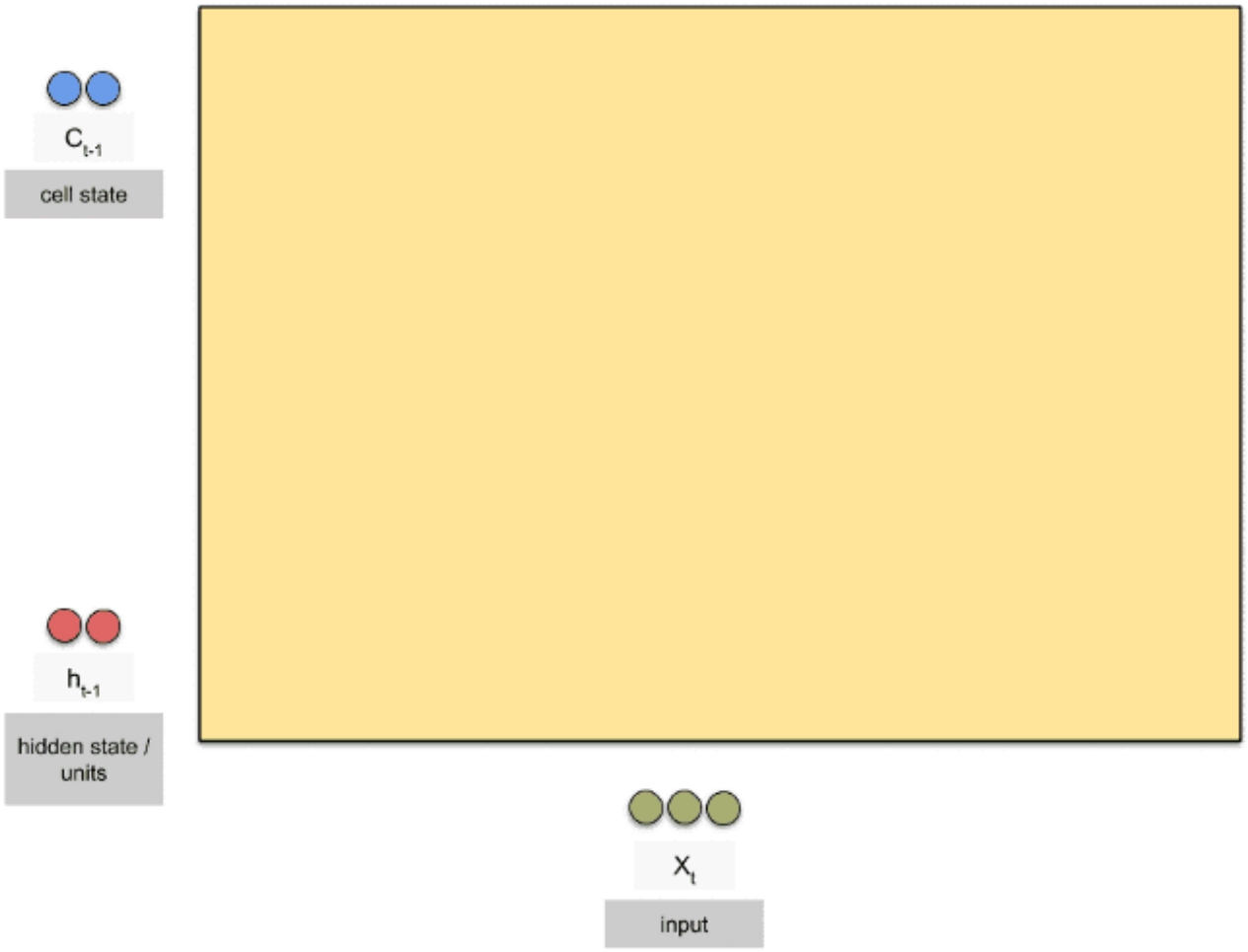
$$\boldsymbol{O}_t = \sigma(\boldsymbol{X}_t\boldsymbol{W}_{xo} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{ho} + \boldsymbol{b}_o)$$

$$\tilde{\boldsymbol{C}}_t = \tanh(\boldsymbol{X}_t\boldsymbol{W}_{xc} + \boldsymbol{H}_{t-1}\boldsymbol{W}_{hc} + \boldsymbol{b}_c)$$

$$\boldsymbol{C}_t = \boldsymbol{F}_t \odot \boldsymbol{C}_{t-1} + \boldsymbol{I}_t \odot \tilde{\boldsymbol{C}}_t$$

$$\boldsymbol{H}_t = \boldsymbol{O}_t \odot \tanh(\boldsymbol{C}_t)$$

长短期记忆神经网络（LSTM）



LSTM的各种变体

- 没有遗忘门（早期的LSTM）

$$\mathbf{c}_t = \mathbf{c}_{t-1} + \mathbf{i}_t \odot \tilde{\mathbf{c}}_t.$$

- 耦合输入门和遗忘门

$$\mathbf{f}_t + \mathbf{i}_t = \mathbf{1}.$$

$$\boldsymbol{c}_t = (\mathbf{1} - \boldsymbol{i}_t) \odot \boldsymbol{c}_{t-1} + \boldsymbol{i}_t \odot \tilde{\boldsymbol{c}}_t$$

- peephole连接

$$\mathbf{i}_t = \sigma(W_i \mathbf{x}_t + U_i \mathbf{h}_{t-1} + V_i \mathbf{c}_{t-1} + \mathbf{b}_i),$$

$$\mathbf{f}_t = \sigma(W_f \mathbf{x}_t + U_f \mathbf{h}_{t-1} + V_f \mathbf{c}_{t-1} + \mathbf{b}_f),$$

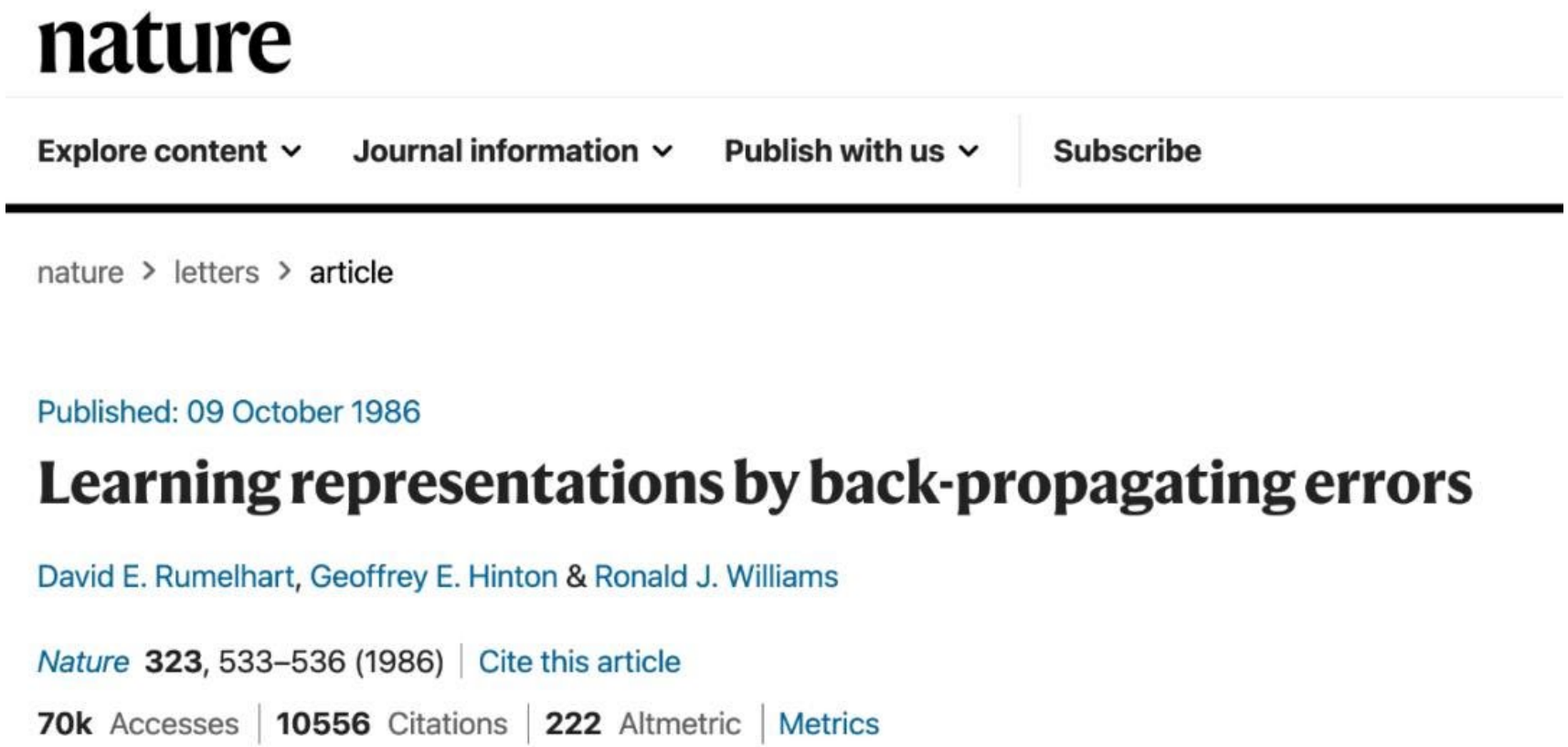
$$\mathbf{o}_t = \sigma(W_o \mathbf{x}_t + U_o \mathbf{h}_{t-1} + V_o \mathbf{c}_t + \mathbf{b}_o),$$

LSTM

Geoffrey Hinton

Jürgen Schmidhuber

LSTM


nature
Explore content
Journal information
Publish with us
Subscribe
nature > letters > article
Published: 09 October 1986
Learning representations by back-propagating errors
David E. Rumelhart, Geoffrey E. Hinton & Ronald J. Williams
Nature 323, 533–536 (1986) | Cite this article
70k Accesses | 10556 Citations | 222 Altmetric | Metrics

LSTM

Gradient Theory of Optimal Flight Paths

HENRY J. KELLEY[1]

Grumman Aircraft Engineering Corp.
Bethpage, N. Y.

An analytical development of flight performance optimization according to the method of gradients or "method of steepest descent" is presented. Construction of a minimizing sequence of flight paths by a stepwise process of descent along the local gradient direction is described as a computational scheme. Numerical application of the technique is illustrated in a simple example of orbital transfer via solar sail propulsion. Successive approximations to minimum time planar flight paths from Earth's orbit to the orbit of Mars are presented for cases corresponding to free and fixed boundary conditions on terminal velocity components.

LSTM

LONG SHORT-TERM MEMORY

NEURAL COMPUTATION 9(8):1735–1780, 1997

Sepp Hochreiter
Fakultät für Informatik
Technische Universität München
80290 München, Germany
hochreit@informatik.tu-muenchen.de
http://www7.informatik.tu-muenchen.de/~hochreit

Jürgen Schmidhuber
IDSIA
Corso Elvezia 36
6900 Lugano, Switzerland
juergen@idsia.ch
http://www.idsia.ch/~juergen