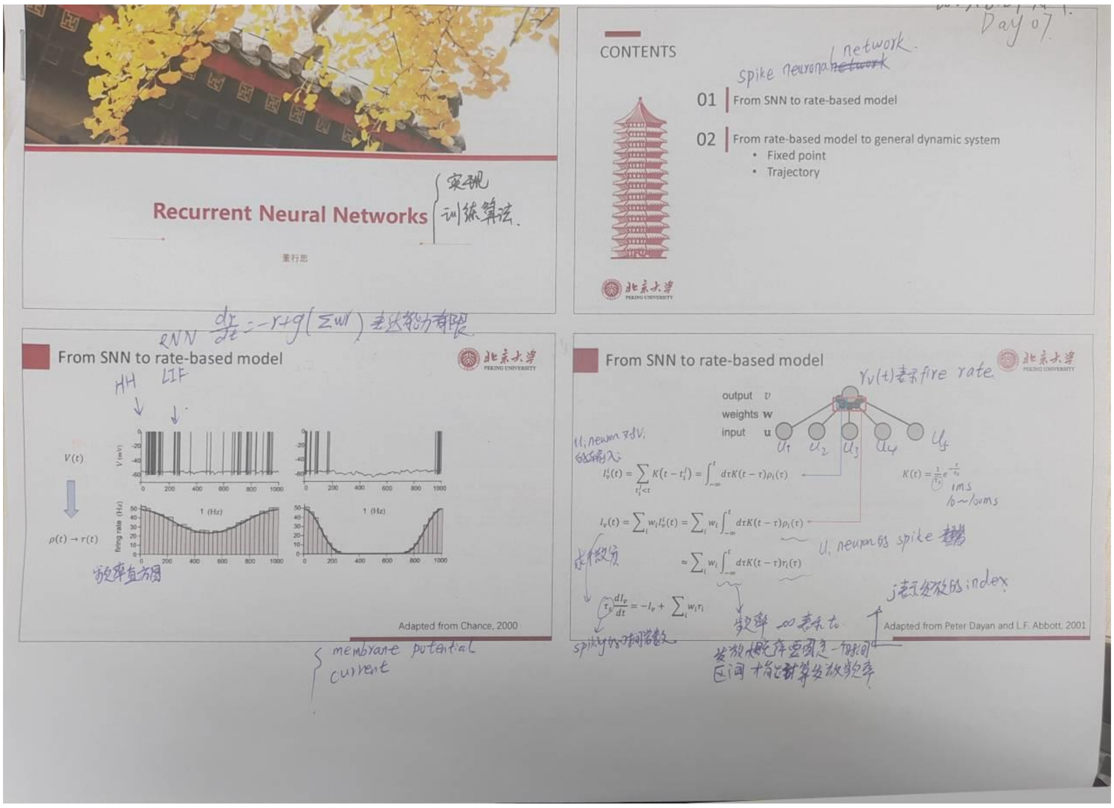


Day 07

Recurrent Neural Networks

实现 / 训练算法

董行思

CONTENTS

spike neuronal network

01 From SNN to rate-based model

02 From rate-based model to general dynamic system
- Fixed point
- Trajectory

From SNN to rate-based model

RNN $\frac{dr}{dt}=-r+g(\Sigma wI)$ 表达能力有限

HH　LIF

$V(t)$

$\rho(t) \to r(t)$

放电直方图

V (mV)　firing rate (Hz)　I (Hz)

Adapted from Chance, 2000

membrane potential / current

From SNN to rate-based model

$r_v(t)$表示fire rate

output v
weights $\mathbf{w}$
input $\mathbf{u}$

u_1 u_2 u_3 u_4 u_5

u_i neuron 对 v_i 的输入：

$$I_v^i(t)=\sum_{t_i^j<t} K(t-t_i^j)=\int_{-\infty}^{t} d\tau K(t-\tau)\rho_i(\tau)$$

$$K(t)=\frac{1}{\tau_s}e^{-\frac{t}{\tau_s}}$$

1ms　10~100ms

$$I_v(t)=\sum_i w_i I_v^i(t)=\sum_i w_i\int_{-\infty}^{t} d\tau K(t-\tau)\rho_i(\tau)$$

u_i neuron 的 spike 情况

$$\approx \sum_i w_i\int_{-\infty}^{t} d\tau K(t-\tau)r_i(\tau)$$

求微分

$$\tau_s\frac{dI_v}{dt}=-I_v+\sum_i w_i r_i$$

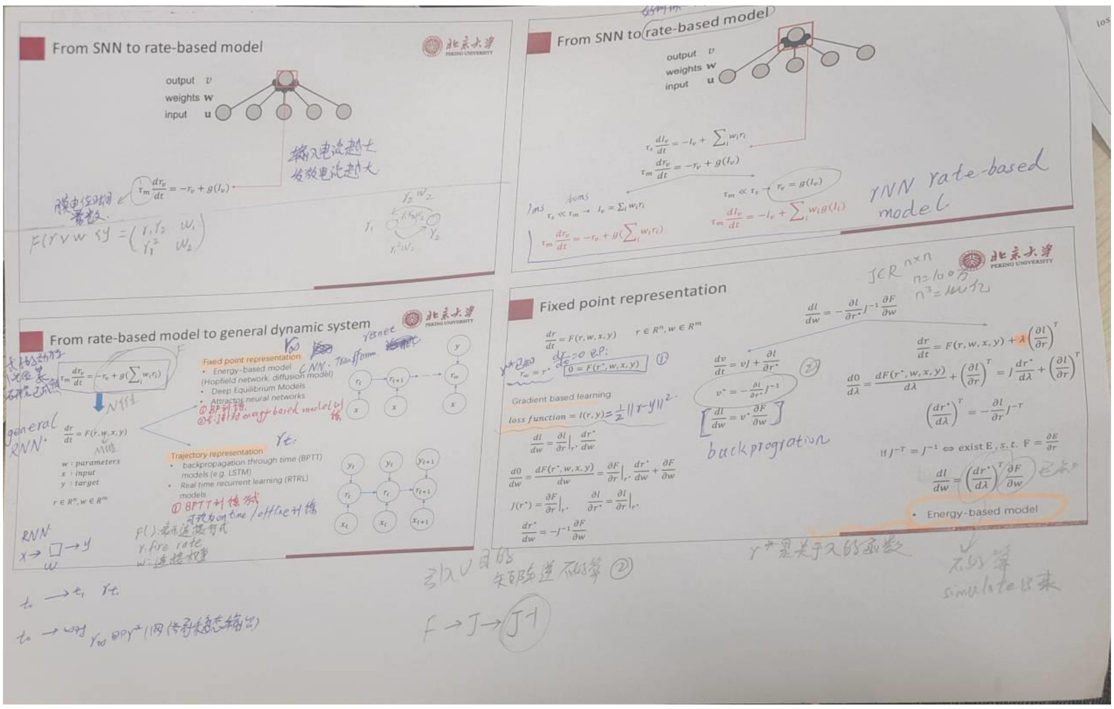
spiking的时间常数

放率 $-\infty$ 表示 t_0

发放概率要固定一个时间区间才能计算发放频率

j表示发放的index

Adapted from Peter Dayan and L.F. Abbott, 2001



From SNN to rate-based model

output v
weights $\mathbf{w}$
input $\mathbf{u}$

输入电流越大 / 发放电流越大

$$\tau_m\frac{dr_v}{dt}=-r_v+g(I_v)$$

膜电位时间常数

$F(r\,v\,w\,\langle y)=\begin{pmatrix} r_1 r_2 & w_1 \\ r_1^2 & w_2 \end{pmatrix}$

From SNN to rate-based model

output v
weights $\mathbf{w}$
input $\mathbf{u}$

$$\tau_s\frac{dI_v}{dt}=-I_v+\sum_i w_i r_i$$

$$\tau_m\frac{dr_v}{dt}=-r_v+g(I_v)$$

1ms　10ms

$\tau_s \ll \tau_m \to I_v=\Sigma_i w_i r_i$

$$\tau_m\frac{dr_v}{dt}=-r_v+g\left(\sum_i w_i r_i\right)$$

$\tau_m \ll \tau_s \to r_v=g(I_v)$

$$\tau_m\frac{dI_v}{dt}=-I_v+\sum_i w_i g(I_i)$$

rNN rate-based model

From rate-based model to general dynamic system

$$\tau_m\frac{dr_v}{dt}=-r_v+g\left(\sum_i w_i r_i\right)$$

N维

general RNN:

$$\frac{dr}{dt}=F(r,w,x,y)$$

w : parameters
x : input
y : target

$r\in R^n, w\in R^m$

Fixed point representation
- Energy-based model (Hopfield network, diffusion model)
- Deep Equilibrium Models
- Attractor neural networks

r_∞　CNN、Transform　resnet

① BP训练
② 针对energy-based model训练

Trajectory representation
- backpropagation through time (BPTT) models (e.g. LSTM)
- Real time recurrent learning (RTRL) models

r_t

① BPTT训练方式 可视为on time / offline训练

RNN
$x \to \square \to y$
w

F( )表示连接方式
r: fire rate
w: 连接权重

$t_0 \to t_1$　r_{t_1}

$t_0 \to \infty$时 r_∞ 即 r^* (网络的稳态输出)

Fixed point representation

$$\frac{dr}{dt}=F(r,w,x,y)\qquad r\in R^n, w\in R^m$$

r^*已知　$r_\infty=r^*$　$\frac{dr}{dt}=0$ 即:

$$\boxed{0=F(r^*,w,x,y)}\quad ①$$

Gradient based learning

loss function $=l(r,y)=\frac{1}{2}\|r-y\|^2$

$$\frac{dl}{dw}=\frac{\partial l}{\partial r}\Big|_{r^*}\frac{dr^*}{dw}$$

$$\frac{d0}{dw}=\frac{dF(r^*,w,x,y)}{dw}=\frac{\partial F}{\partial r}\Big|_{r^*}\frac{dr^*}{dw}+\frac{\partial F}{\partial w}$$

$$J(r^*)=\frac{\partial F}{\partial r}\Big|_{r^*}\qquad \frac{\partial l}{\partial r^*}=\frac{\partial l}{\partial r}\Big|_{r^*}$$

$$\frac{dr^*}{dw}=-J^{-1}\frac{\partial F}{\partial w}$$

$$\frac{dl}{dw}=-\frac{\partial l}{\partial r^*}J^{-1}\frac{\partial F}{\partial w}$$

$J\in R^{n\times n}$　$n=100$万　$n^3=100$亿

$$\frac{dv}{dt}=vJ+\frac{\partial l}{\partial r^*}\quad ②$$

$$v^*=-\frac{\partial l}{\partial r^*}J^{-1}$$

$$\left[\frac{dl}{dw}=v^*\frac{\partial F}{\partial w}\right]$$

backprogration

$$\frac{dr}{dt}=F(r,w,x,y)+\lambda\left(\frac{\partial l}{\partial r}\right)^T$$

$$\frac{d0}{d\lambda}=\frac{dF(r^*,w,x,y)}{d\lambda}+\left(\frac{\partial l}{\partial r}\right)^T=J\frac{dr^*}{d\lambda}+\left(\frac{\partial l}{\partial r}\right)^T$$

$$\left(\frac{dr^*}{d\lambda}\right)^T=-\frac{\partial l}{\partial r}J^{-T}$$

If $J^{-T}=J^{-1} \Leftrightarrow$ exist E, s.t. $F=\frac{\partial E}{\partial r}$

$$\frac{dl}{dw}=\left(\frac{dr^*}{d\lambda}\right)^T\frac{\partial F}{\partial w}$$

已知

- Energy-based model

r^*是关于λ的函数

不好算 simulate出来

引入v目的 矩阵求逆不好算 ②

$F \to J \to J^{-1}$

Trajectory representation

$$\frac{dr}{dt} = F(r, w, x, y) \quad r \in R^n, w \in R^m$$

loss function: $l = \int a_t l_t(r_t, y_t) dt$ $= \frac{1}{2}(r_t - y_t)^2$

$$\frac{dl_t(r_t, y_t)}{dw} = \frac{\partial l_t}{\partial r_t} \frac{dr_t}{dw}$$

$$\frac{dr_t}{dw} = \frac{d}{dw}\int_0^t dr_t = \frac{d}{dw}\int_0^t \frac{dr_\tau}{d\tau} d\tau$$

$$= \frac{d}{dw}\int_0^t F(r, w, x, y) d\tau$$

$$= \frac{d}{dw}\int_0^t F(r, w, x, y) d\tau$$

$$p_t = \int_0^t \frac{dF(r, w, x, y)}{dw} d\tau$$

$$p_t = \frac{dr_t}{dw}$$

$$\frac{dp_t}{dt} = \frac{dF(r, w, x, y)}{dw} = J(r_t)p_t + \frac{\partial F}{\partial w}$$

Real time recurrent learning
Time: $O(n^2 m * T)$
Space: $O(mn + n^2)$

$$\boxed{\frac{dp_t}{dt} = J(r_t)p_t + \frac{\partial F}{\partial w}}$$

$$p_t = [J(r_{t-1})\Delta t + 1]p_{t-1} + \frac{\partial F(r_{t-1})}{\partial w}\Delta t$$ 用离散迭代式

$$p_t = \frac{\partial r_t}{\partial r_{t-1}} p_{t-2} + \frac{\partial F(r_{t-1})}{\partial w}\Delta t$$

$$p_t = \frac{\partial r_t}{\partial r_{t-1}}\frac{\partial r_{t-1}}{\partial r_{t-2}} p_{t-2} + \frac{\partial r_t}{\partial r_{t-1}}\frac{\partial F(r_{t-1})}{\partial w}\Delta t + \frac{\partial F(r_{t-1})}{\partial w}\Delta t$$

$$\frac{dr_t}{dw} = p_t = \int_0^t \frac{\partial r_t}{\partial r_\tau}\frac{\partial F(r_\tau, w, x, y)}{\partial w} d\tau$$

$$\frac{dl_t(r_t, y_t)}{dw} = \int_0^t \frac{\partial l_t}{\partial r_t}\frac{\partial r_t}{\partial r_\tau}\frac{\partial F(r_\tau, w, x, y)}{\partial w} d\tau$$

$1 \times n \quad n \times n$

BPTT
Time: $O(n^2 T + nmT)$
Space: $O(mn + n^2)$

Example

$$\frac{dr}{dt} = -r + wr + b + x$$

For a input sequence $x_{1:T} \in R$, get the target output $y_{1:T} \in R$

Pseudo code of RTRL
1. Initial $r_0 = y_0$, $p_0 = 0$
2. For a given sequence $x_{1:T}$, compute $r_{1:T}, p_{1:T}$ according to $\frac{dr}{dt} = -r + wr + b + x$, $\frac{dp_w}{dt} = (-1 + w)p_w + r$, $\frac{dp_b}{dt} = (-1 + w)p_b + 1$
3. Set $l_t = \frac{1}{2T}(r_t - y_t)^2$, leading to $\Delta w = -\frac{\eta}{T}\sum(r_t - y_t)p_t$, $\Delta b = -\frac{\eta}{T}\sum(r_t - y_t)p_b$

Pseudo code of BPTT
1. Initial $r_0 = y_0$, $p_0 = 0$
2. For a given sequence $x_{1:T}$, compute $r_{1:T}$ according to $\frac{dr}{dt} = -r + wr + b + x$
3. Set $l_t = \frac{1}{2T}(r_t - y_t)^2$, leading to $\Delta w = -\eta\sum_t\sum_\tau \frac{1}{T}(r_t - y_t)\frac{\partial r_t}{\partial r_\tau} r$, $\Delta b = -\eta\sum_t\sum_\tau \frac{1}{T}(r_t - y_t)\frac{\partial r_t}{\partial r_\tau}$

已知 r^*、$\frac{\partial L}{\partial r}\big|_{r^*}$、$\frac{\partial F}{\partial r}\big|_{r^*}$，
对①式关于 w 求导
$0 = F(r^*, w, x, y)$ ---①

从而 $0 = \frac{dF(r^*, w, x, y)}{dw}$

$= \frac{\partial F}{\partial r}\big|_{r^*} \cdot \frac{dr^*}{dw} + \frac{\partial F}{\partial w}$

$= J \cdot \frac{dr^*}{dw} + \frac{\partial F}{\partial w}$

即: $\frac{dr^*}{dw} = -J^{-1} \cdot \frac{\partial F}{\partial w}$

$r_t = F(r_{t-1}) \cdot \Delta t + r_{t-1}$

$\frac{\partial r_t}{\partial r_{t-1}} = \frac{\partial F}{\partial r}\big|_{r_{t-1}}$

关于 v^* 相关

$F = \begin{pmatrix} 1 & 1 \\ 0 & 1 \end{pmatrix}\begin{pmatrix} x_1 \\ x_2 \end{pmatrix}$

aA^{-1} 已知 A, a 求 aA^{-1} 迭代的求逆算法

v, s.t. $\dot{v} = vA - a$

$v^* = aA^{-1}$

$A = P\Sigma P$

已知 a 求 $\frac{1}{a}$. $\dot{v} = va - 1$ 无除法

$v^* = \frac{1}{a}$