QR decomposition based methods.

Problem: $f: M \to M$, $M \subseteq \mathbb{R}^m$, $t = 0, 1, 2, \cdots$

$$x \mapsto f(x)$$

Given $x_0 \in M$, we can get $x_1 = f(x_0) \in M$, $x_2 = f(x_1) \in M$, $\cdots$

$$Y^0 = \frac{df(x_0)}{dx_0}, \quad Y^1 = \frac{df(x_1)}{dx_1}, \cdots \quad \text{(Jacobi matrixes)}$$

$\Rightarrow$ Define $\Lambda_x := \lim\limits_{t \to +\infty} \left[Y(x;t)^T Y(x;t) \right]^{\frac{1}{2t}}$

And m Lyapunov exponents:

$$\lambda_i = \lim_{t \to +\infty} \frac{1}{t} \ln \| Y(x;t) \underbrace{u_i}_{\Lambda_x \text{的第}i\text{个特征值}} \| \quad (1 \le i \le m)$$

Then how to calculate $\lambda_1, \lambda_2, \cdots, \lambda_m$?

Solution (QR decomposition based methods):

首先定义矩阵 $P = (\underbrace{P^1}_{p^i(t) = D_x \phi^t(O^i) = YO^i}, P^2, \cdots, P^k)$ $(1 \le k \le m)$ $\in \mathbb{R}^{m \times k}$

其中 $O^1, O^2, \cdots, O^k$ 为 T_xM 切空间的一组正交基

$k = \dim(T_xM)$

然后对P进行QR分解:

$$P = \underbrace{(Q^1, Q^2, \cdots, Q^k)}_{m \times k} \underbrace{\begin{pmatrix} R_{11} & & * \\ & R_{22} & \\ & & \ddots \\ 0 & & R_{kk} \end{pmatrix}}_{k \times k}$$

得到 t 时刻的 $R_{11}\ R_{22},\ \dots,\ R_{kk}$

多次计算 $(t=0,1,2,\cdots)$. 取极限:

$$\lambda_i = \underbrace{\frac{1}{\Delta t}}_{} \lim_{n\to+\infty} \frac{1}{n}\sum_{j=0}^{n-1} \ln\left(R_{ii}^{j}\right)$$

默认应该是取 1吧?

Remind: 只需计算 $R_{11}^{j}, R_{22}^{j}, \dots, R_{kk}^{j}$ 而并不需要去做 QR分解.

Treppen - Iteration Algorithm:

$$\underbrace{Y^{j-1}}_{m\times m}\underbrace{Q^{j-1}}_{m\times k} = \underbrace{Q^{j}}_{m\times k}\underbrace{R^{j-1}}_{k\times k} \quad (1\le j\le n)$$

具体进行方法:

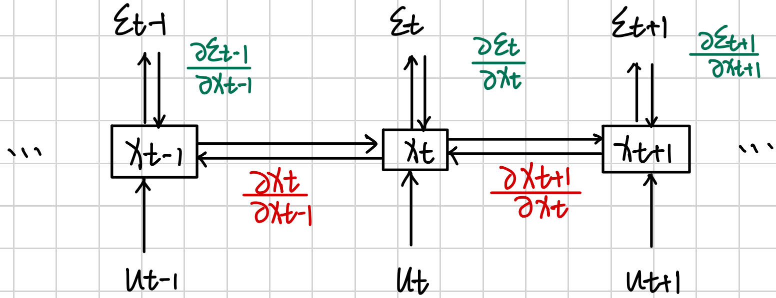
假设已有 $Y^1, Y^2, \dots, Y^n$, $Q^1, Q^2, \dots, Q^n$, $R^1, R^2, \dots, R^n$ 的值:

则 $Q_{n+1}' = Y_{n+1} Q_n$, $Q_{n+1}' = Q_{n+1} R_{n+1}$ (QR)

即得到 $Q_{n+1}.\ R_{n+1}$ 的值

这样迭代可避免较大误差

放在RNN中，需要计算什么呢？



给定输入 $(u_1, u_2, \cdots, u_n)$

每一层中: $x_t = \sigma(W_{uu}u_t + W_{xx}x_{t-1} + b_x)$

激活函数 权重矩阵 偏置项

(tanh / ReLU)

$\varepsilon_t = \phi(W_{xy}x_t + b_y)$

权重矩阵 偏置

输出激活函数 (e.g. softmax)

因此所有的参数为: (所有层都共用的):

- W_{uu}, W_{xx}, W_{xy}: 随机初始化 (Gauss 分布 / 均匀分布)
  Xavier 初始化或 He 初始化
- b_x, b_y: 初始化为0

参数更新的过程为:

$$\begin{cases} W \leftarrow W - \eta \frac{\partial L}{\partial W} \\ b \leftarrow W - \eta \frac{\partial L}{\partial b} \end{cases}$$

给定输入 $(u_1, u_2, \cdots, u_n)$. 一次训练为:

前向传播: $x_t = \sigma(W_{uu} u_t + W_{xx} x_t + b_x)$

$$y_t = \phi(W_{xy} x_t + b_y)$$

计算损失: $L = \sum_{t=1}^{n} \mathcal{L}(y_t, \hat{y}_t)$?

Backpropagation: $\delta L_t = \frac{\partial \mathcal{L}(y_t, \hat{y}_t)}{\partial y_t}$

$$\begin{cases} \frac{\partial L}{\partial W_{uu}} = \sum_{t=1}^{n} \delta L_t \frac{\partial y_t}{\partial W_{uu}} \\ \frac{\partial L}{\partial W_{xx}} = \sum_{t=1}^{n} \delta L_t \frac{\partial y_t}{\partial W_{xx}} \\ \frac{\partial L}{\partial W_{xy}} = \sum_{t=1}^{n} \delta L_t \frac{\partial y_t}{\partial W_{xy}} \\ \frac{\partial L}{\partial b_x} = \sum_{t=1}^{n} \delta L_t \frac{\partial y_t}{\partial b_x} \\ \frac{\partial L}{\partial b_y} = \sum_{t=1}^{n} \delta L_t \frac{\partial y_t}{\partial b_y} \end{cases}$$

然后: $\begin{cases} W = W - \eta \frac{\partial L}{\partial W} \\ b = b - \eta \frac{\partial L}{\partial b} \end{cases}$

这样就完成一次训练. 相当于:

$$(W', b') = f(W, b)$$

进行 $n \to +\infty$ 次训练 即可算 (W, b) 处 f 的 Lyapunov 指数了