Note about solving

$u' = f(u,t)$ with an NN — Demo

Flux.~~adam~~Adam(0.01) worked so much better than plain old Flux.descent(0.01)

Plain old gradient descent

$$p_{k+1} = p_k - s_k \nabla_p \text{loss}$$

(p: parameters)

Fancier methods compute s_k + the step using memory of previous step size s

Art not a science on what to pick

Hooke's law with Adam

Demo

Dynamical Systems – a little math, a little performance optimization

$$u_{n+1} = \text{function}(u_n, \text{parameters}) \text{ — etc}$$

logistic map — $u_{n+1} = r u_n (1-u_n)$

recurrent neural network — $u_{n+1} = u_n + f(u_n, \theta)$ — cont form $u' \sim$ NN with parameters θ

might have noise — $u_{n+1} = \alpha u_n + \varepsilon_n$

might have delays — $u_{n+1} = \sum_{j=0}^{k} \alpha_j u_{n-j} + \varepsilon_n$

can be multidimensional — Lorenz

Simplest Model

$u_{n+1} = \alpha u_n \qquad u_0 = \text{given} \in \mathbb{R}$

Solution: $u_n = \alpha^n u_0$

$|\alpha| < 1 \qquad u_n \to 0$

$|\alpha| > 1 \qquad u_n \to \infty$

$|\alpha| = 1$ depends

For vectors

$u_{n+1} = A u_n \qquad u_0 \in \mathbb{R}^n$

$|\lambda_{max}| < 1 \qquad u_n \to 0$

$|\lambda_{max}| > 1 \qquad u_n \to \infty$

Banach Fixed Point

If $\| f(x) - f(y) \| \leq q \| x - y \|$ for some $q \leq 1$

then

$\exists \; f(x^*) = x^*$

+ the sequence $x_{n+1} = f(x_n)$

converges to the limit

Efficient Implementation

DEMO

(8)

Analysis of $u_{n+1} = u_n^2 - p u_n$

Let $u_n = x_n + (p+1)$ x_n small

$$x_{n+1} + (p+1) = x_n^2 + 2x_n(p+1) + (p+1)^2 - p x_n - p(p+1)$$

$$x_{n+1} = x_n (p+2)$$

$u_n \approx 0$ $u_{n+1} \approx -p u_n$ so for $|p|<1$ converges to 0