Lecture January 13, 2022

RNNs

simple example

$x(t=0) \wedge v(t=0) = v_0$

$= x_0 \qquad t=0, \; t_0$

$$m \frac{d^2x}{dt^2} + \eta \frac{dx}{dt} + x(t) = F(t)$$

$$v(t) = \frac{dx}{dt}$$

$$m \frac{dv}{dt} + \eta v + x(t) = F(t)$$

$$\frac{dv}{dt} + \underbrace{\frac{\eta}{m}}_{\alpha} v + \underbrace{\frac{x}{m}}_{\delta} = \underbrace{\frac{F}{m}}_{\tilde{F}}$$

$$\frac{dv}{dt} = \tilde{F} - \alpha v - \delta x$$

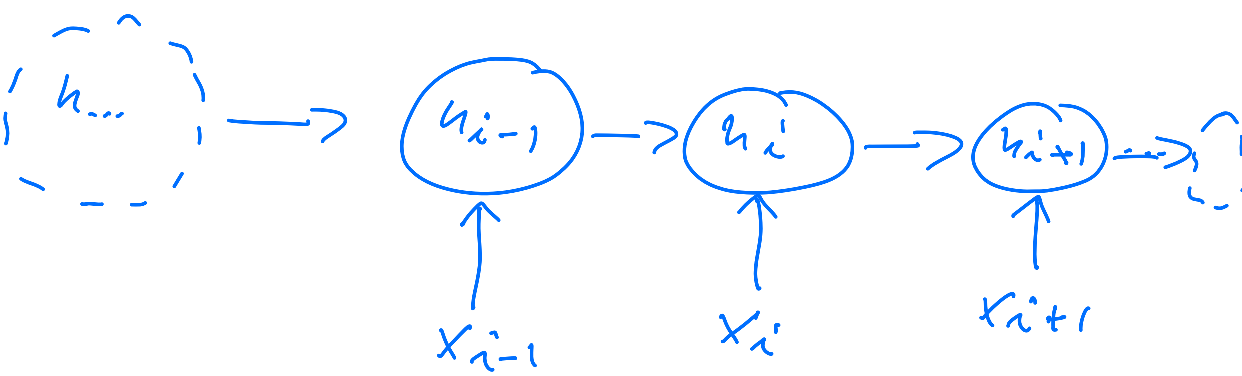
Euler's method:

$$v(t_i + \Delta t) = v_{i+1} = (\tilde{F}_i - \alpha v_i - \delta x_i)\Delta t + v_i$$

$$\left(y_{i+1} = y_i + \Delta t \left.\frac{dy}{dt}\right|_{y=y_i} \right)$$

$$v_{i+1} = h(v_i, \tilde{F}_i, x_i, t_i)$$

$$= \underbrace{h(\underline{h(v_{i-1}}, \tilde{F}_{i-1}, x_{i-1}, t_{i-1}), \tilde{F}_i \ldots)}_{h_{i+1}(h_i,\, x_i,\, t_i)}$$



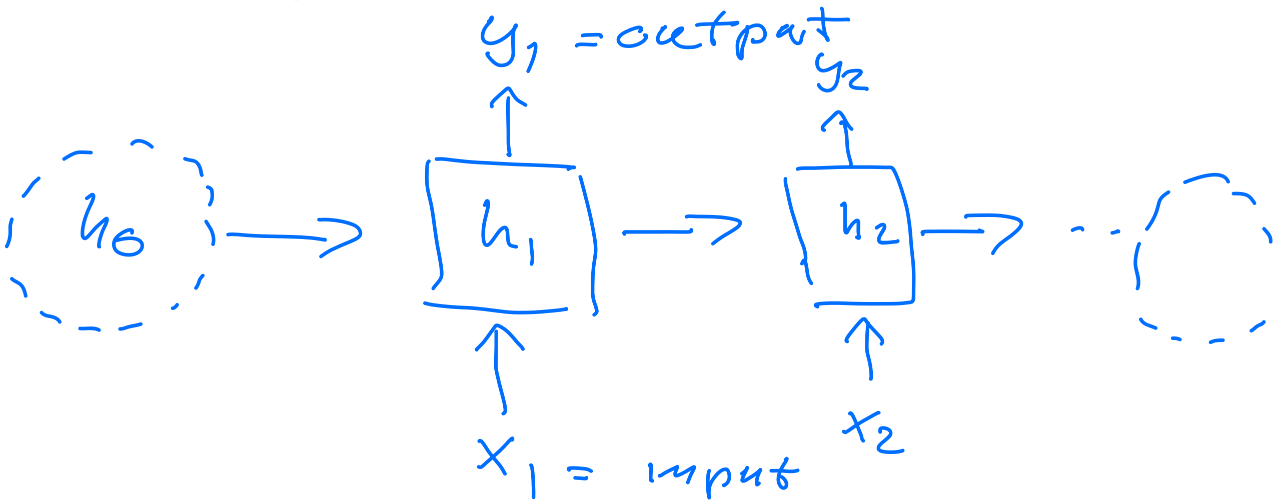
- NNs work well when the input data are well structured

- CNNs work well with images

- what does not work well?
  processing data with unknown length

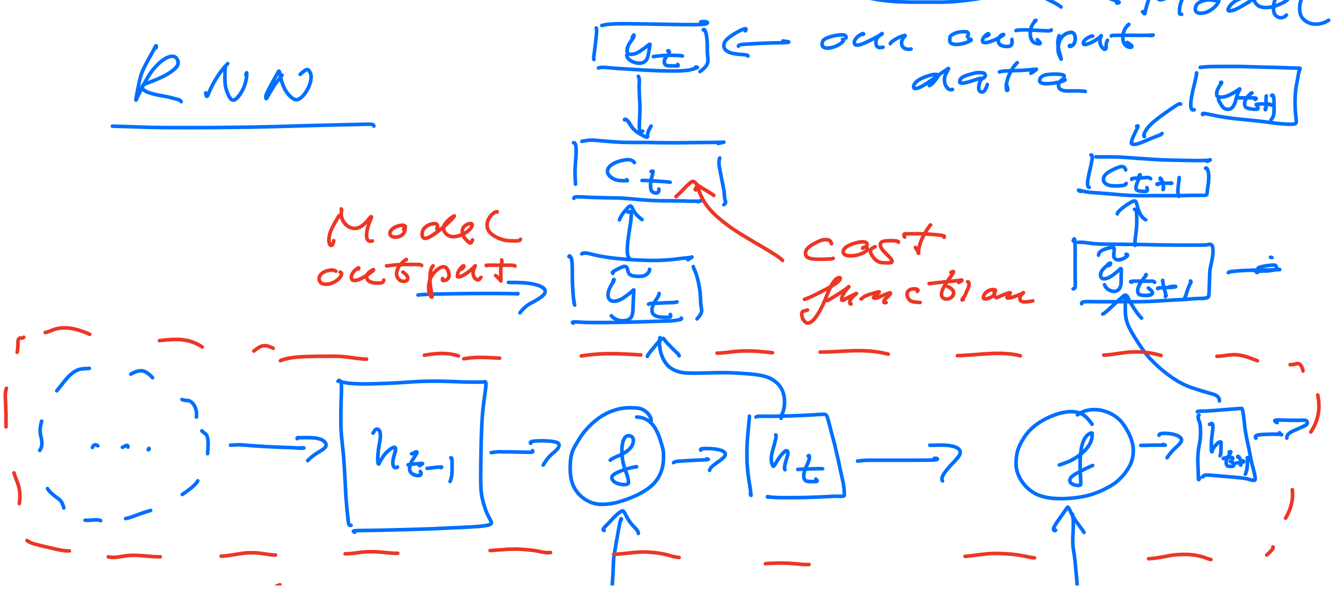
RNNs

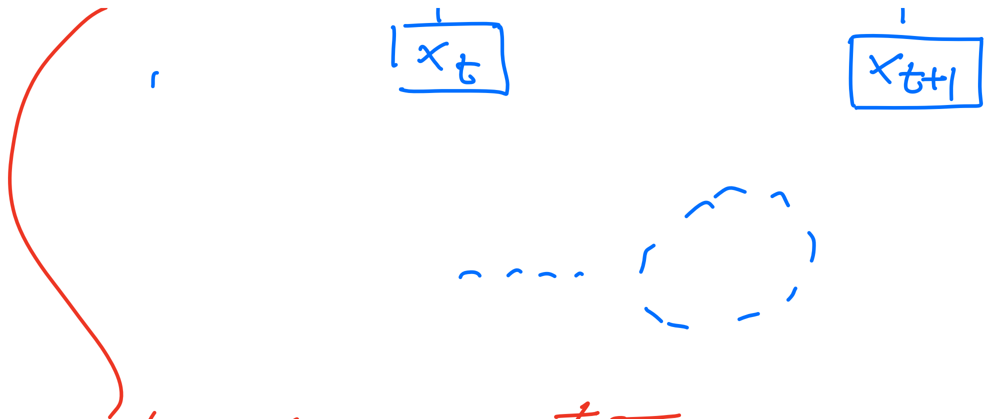
- new inputs
- manipulates the state
- reuse weights
- gives new outputs.



$$h_t = \underbrace{f(h_{t-1}, x_t; \Theta)}_{\text{Model}}$$

RNN



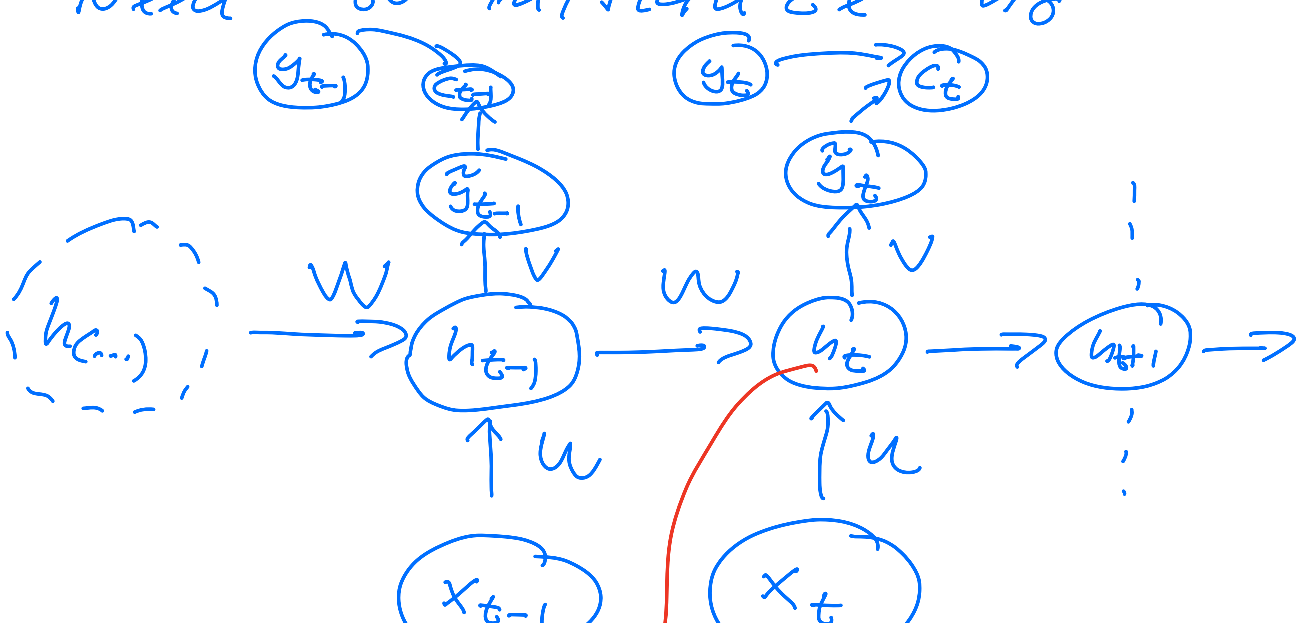


hidden units

RNNs produce an output at each time step and have recurrent connections between hidden units.

For each (time) step, we have a forward propagation.

Need to initialize h_0



bias

$$z_t = b + W h_{t-1} + U x_t$$

output is $h_t = f(z_t)$

$$\tilde{y}_t = g(h_t V + c)$$

bias

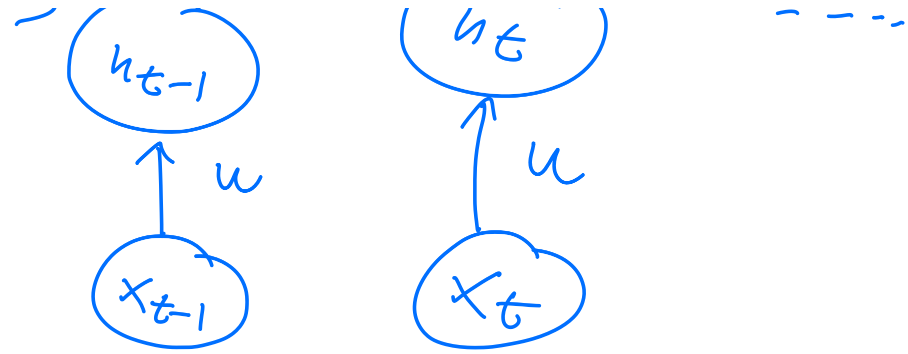
BPTT = Back propagation through time.

∇C_t which depends

$\nabla_b C_t$, $\nabla_V C_t$, $\nabla_c C_t$

$\nabla_W C_t$, $\nabla_U C_t$

a simplification is to only connect the output from one layer to the next.



Differential equations

$$\frac{dy}{dt} = \underbrace{f\left(y, t, \frac{dy}{dt}\right)}_{\text{known equation}}$$

$$m\frac{d^2y}{dt^2} = -ky$$

$$\frac{d^2y}{dt^2} = -\frac{k}{m}y$$

$$v = \frac{dy}{dt}$$

$$a = \frac{dv}{dt} = \frac{d^2y}{dt^2} = -\frac{k}{m}y$$

$$\frac{k}{m} = \omega_0^2$$

$$\frac{dv}{\;} = \left(-\omega_0^2 y\right)$$

$$\boxed{\frac{dy}{dt} = v}$$

$$y = A\cos(\omega_0 t) + B\sin(\omega_0 t)$$

$t_0 = 0 \quad v_0 \wedge y_0$

Euler's method:

$$v_{i+1} = v_i + \Delta t \left.\overbrace{\frac{dv}{dt}}^{a_i}\right|_{v=v_i} + O(\Delta t^2)$$

$$y_{i+1} = y_i + \Delta t\, v_i$$

$$v_{i+1} = v_i + \Delta t\, a_i$$

$$y_{i+1} = y_i + \Delta t \cdot v_i$$

Algo

Def initial conditions:-

t_0, v_0, y_0

FOR $t = t_0, t_{final}$

$$v_{t+1} = v_t + \Delta t\, a_t$$

$$y_{t+1} = y_t + \Delta t\, v_t$$

when using NNs :

- respect initial functions
- attempt a solution

$$g_{Trial}(t) = h_1(t) + h_2(t, NN(t,P))$$

P = Biases and weights of a neural network

$$\frac{dy}{dt} = f\left(y, \frac{dy}{dt}, t\right)$$

Cost function

(i) $\left| \frac{dy}{dt} - f \right| \leq \varepsilon \sim 10^{-6}$

(ii) MSE of this difference.