6/10.

Discuss: Neural Network. Shallow: 1 layer. 1 neuron.

Deep: many layer many neuron

Conv. Neural Net: ✓ image data

✓ data in 2D or 3D.

Today: Recurrent Neural Network. (RNN).

notion: Long Short-term Memory.

Setup: Markov Chain. ← LSTM (solution).

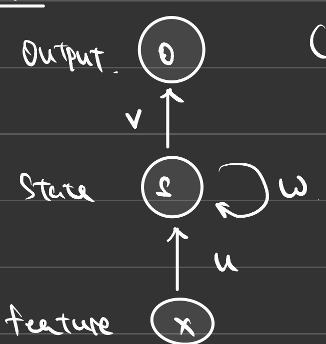
Architecture:

Text data: I love cats. (3 words).

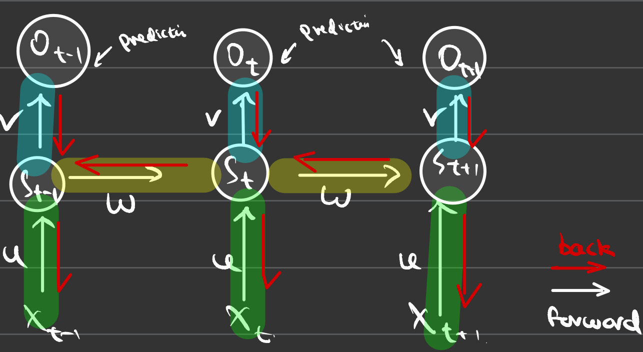
dictionary: { cats, I, love }.

Text to token seq.: $\begin{pmatrix}0\\1\\0\end{pmatrix}, \begin{pmatrix}0\\0\\1\end{pmatrix}, \begin{pmatrix}1\\0\\0\end{pmatrix}$

Simple:



Complicated



Mathematical Form for Forward Propagation:

" ": $S_{t-1} = a_1(X_{t-1} \cdot u)$, ex: pick ReLU ✓

$= \max(X_{t-1} \cdot u, 0)$. "feels like flipping a switch".

" ": $S_t = a_1(S_{t-1} \cdot w + X_t \cdot u)$

↑ non-linear. ← linear transformation.

↑ $ax^2 + b$. $(a \neq 0)$ ↑ from static perspective.

$ax^3 + bx^2 + c$. $a+b$: linear.

$ax+b$: linear.

$= \max(S_{t-1} \cdot w + X_t \cdot u, 0)$. assume ReLU.

" ": $O_{t-1} = a_2(S_{t-1} \cdot v)$. , ex: sigmoid $y = 1/(1+\exp(-x))$

$$= \frac{1}{1+\exp(-S_{t-1} \cdot v)}$$

Mathematical Form for Backward Propagation:

Cool Name: Back Propagation Through Time. (BPTT).

$$\mathcal{L}(O,t) = \sum_{t=1}^{T} \sum_{i=1}^{n} \ell(O_{it}, Y_{it})$$

loss function: your choice.

✓ regression: MSE.

✓ classification: cross-entropy.

i: refers observation

t: time. (or sequential index, position)

O_{it}: educated guess.

Y_{it}: reality (Truth).

n: sample size.

T: Total length of time.

Dataframe: ?

shift by 1 day

One feature: $t = 1, 2 \ldots, T$.

daily data

	Price	$Price_{t-1}$	$Price_{t-2}$
Jun 9	10		
Jun 10	11	10	
Jun 11	12	11	10
Jun 12	13	12	11

← One observation $i = 1, 2 \ldots n$.

Feature:	var. 1.	var 2	var 3.
Obs:	Y = Price	X_1 = Square feet.	X_2 = # of Bedroom.
$i=1$	1 mil.	10000.	5.
$i=2$	800 k.	200.	1.
$\vdots$			
$i=100$.			

Obs: $i = 1, 2, \cdots 100$.

variable: $j = 1, 2$.

Question: What is the loss of this dataframe.

Optimizer: Choose Grad. Desc.

Start: $\mathcal{L}(0, y)$.

At each step s:

compute
$$\nabla \mathcal{L}(0,y) = \begin{array}{r} \\ \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{array} \begin{bmatrix} \frac{\partial}{\partial u} \mathcal{L}(0,y) \\ \frac{\partial}{\partial v} \mathcal{L}(0,y) \\ \frac{\partial}{\partial w} \mathcal{L}(0,y) \end{bmatrix} = \begin{bmatrix} \gamma_1 \\ \gamma_2 \\ \gamma_3 \end{bmatrix}.$$

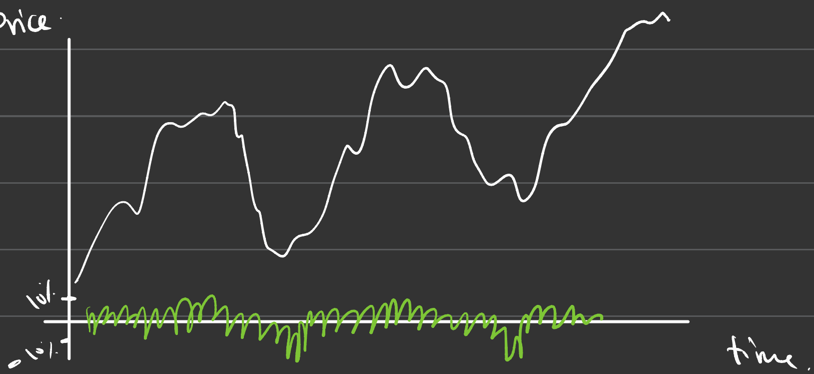
how many mistakes the machine makes with respect to u

update:
$$u_s := u_{s-1} - \eta \cdot \gamma_1$$
$$v_s := v_{s-1} - \eta \gamma_2$$
$$w_s := w_{s-1} - \eta \cdot \gamma_3.$$

η: step size, chosen by us.

Back to Research.

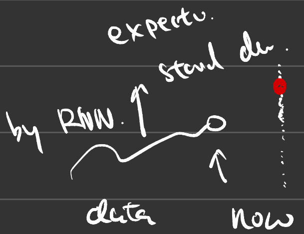
daily chart for one stock.



Monte Carlo:

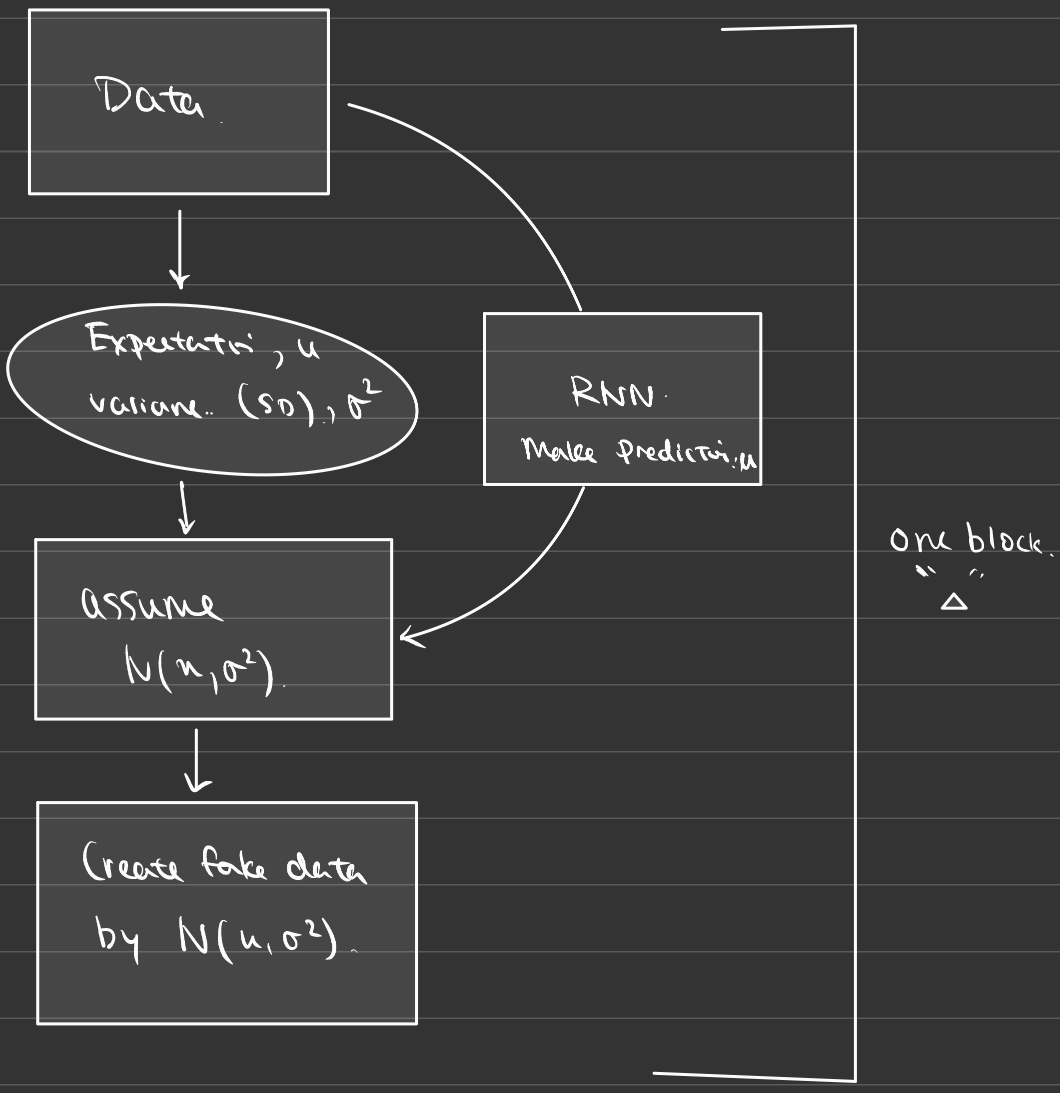
hopefully one guess get it right.

RNN:



hopefully this dot is more accurate

Monte Carlo (1 generation).



Monte Carlo Tree Search: (MCTS).

generation 1. △

$\downarrow$

2. △

$\downarrow$

3. △

$\vdots$

G. △