Neural Network Sketches

Tuesday, July 30, 2024 8:43 AM

1986 → first full year of RoR for FNDX3

* θ_n → represents param set "n"

1986: $\{\theta_0, \theta_1, \ldots, \theta_n\}$

month = 252//1

engine(data[:month]*)

SHARPE(") → above function

get sharpe from 1986[:3] (month)
for θ_n's → cust_sample_space
↳ ≈350

np.mean()*

*np.std() # creating initial distr. for Gibbs

samples = {'x':[], 'y':[]} → cartesian coordinate[0] = (cur, most_recent) (x, y)

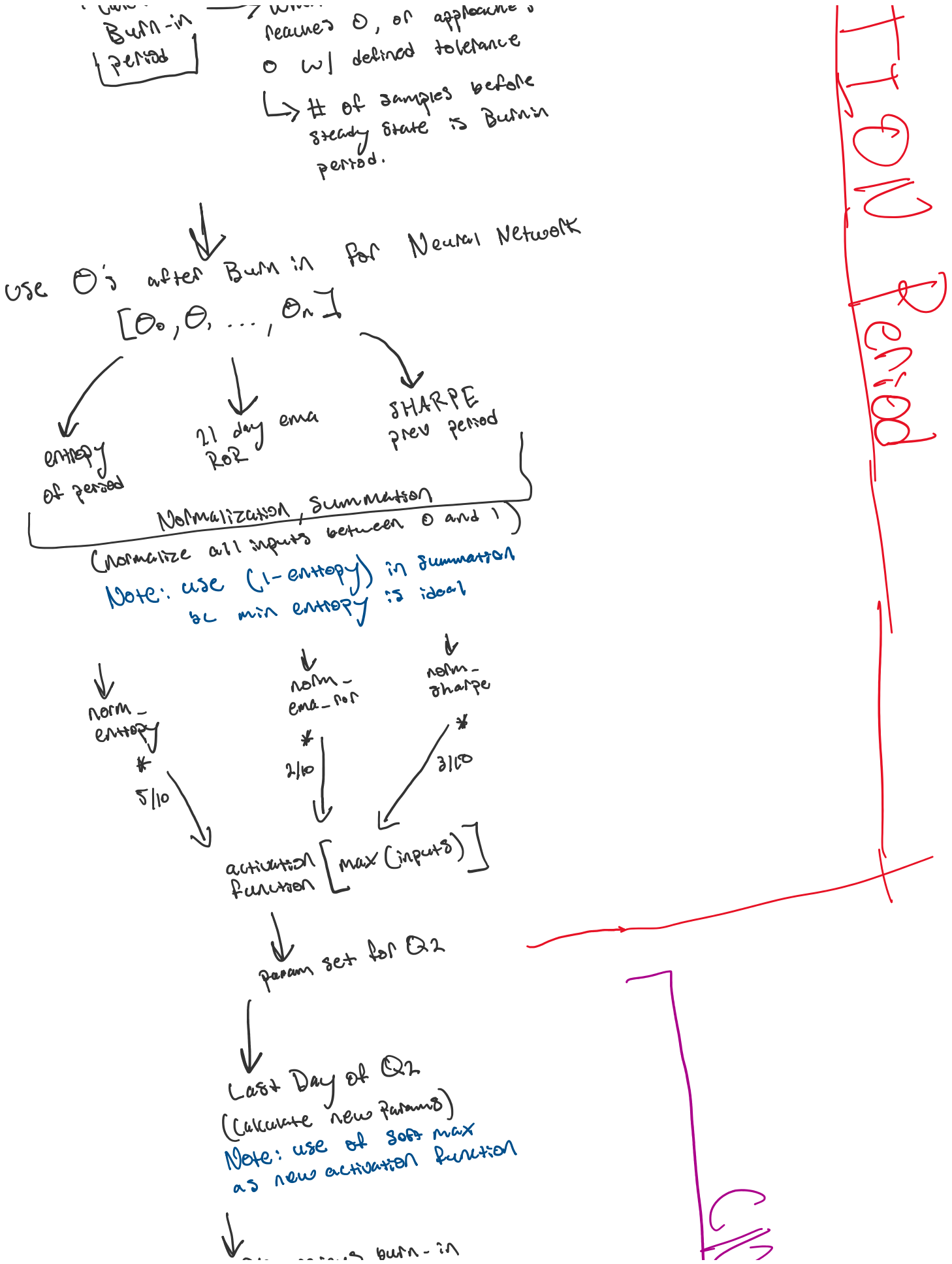
for i in range(initial):
    cur_y = samples['y'][-1]
    new_x = np.random.normal(mean, std)
    new_y = np.rd.rnorm(*, *)
    append(new_x)
    append(new_y)

(gets 200 Markov samples for initial p(x))

+ Another consideration at this point:
- Burning non orthogonal vectors (reduce time complexity)
↳ not currently implemented

Calculate Burn-in period → when auto-correlation reaches 0, or approaches w/ defined tolerance

INITIAL


Burn-in period
reaches 0, or approaches 0 w/ defined tolerance
# of samples before steady state is Burnin period.
use 0's after Burn in for Neural Network
[θ0, θ, ..., θn]
entropy of period
21 day ema RoR
SHARPE prev period
Normalization, summation
Normalize all inputs between 0 and 1
Note: use (1-entropy) in summation bc min entropy is ideal
norm_entropy
norm_ema_ror
norm_sharpe
*
5/10
2/10
3/10
activation function [max (inputs)]
param set for Q2
Last Day of Q2
(Calculate new params)
Note: use of soft max as new activation function
Iteration Period
burn-in

find θ's minus burn-in

1-norm ent

norm-entropy * 8/10

norm_ema_ror * 2/10

norm_sharpe * 8/10

activation:
soft_max( )

choose highest prob θ

repeat this process for NN

⋮

2023

2020 w/ Gibbs

updating