!!!! Il sample ogni volta è 1024(mc)x200(time dimension)x1(b.m. dimention) !!!!!

Nota sulla rete: Use mini batches of data to calculate gradients, when training!

SOLVER

Bisogna modificare la funzione delta, perche mette in input solo l'ultimo valore, mentre noi vogliamo i path dependents

NB: bisogna anche modifcare y (il suo output) di conseguenza? Io prenderei l'ultimo valore di Y, e lo comparerei con il valore attualizzato del payoff (tramite x)

SWAP

- Inserire elementi corretti in eqn\_config

- Sample: inserire modello per calcolare il sottstante (rate): Bjork: pag 281. Scegi modello tra quelli

(eventualmente, includere il caso esplicito)

- G\_TF: cambiare 'x' con 'r'

metterei qui il fatto che il payoff è la somma dei due flussi (quindi attualizzare per il tempo) ((la rete impara da sola a seconda di cosa gli diamo noi di parametro da imaprare nella loss))

DA FARE?: mettere una condizione si y che gli fa imparare nel modo corretto

PROBLEMA: la loss dà come risultato “null”

BARRIER

- Inserire elementi corretti in eqn\_config

- Fare lo switch tra i casi DO, UO ecc, e...

- ...Modificare g\_tf per includere il caso di rottura della barriera

- mettere una condizione si y che gli fa imparare nel modo corretto PROBLEMA RISOLTO?

CONTROLLARE I CASI DI BASE!!!!

Americana tutto un altro discorso :(

PRIMA PENSA AL BSDE\_SOLVER, POI AL XVA SOLVER

- Impostare pagamento dei flussi. IDEA: mettere (r - tasso\_fisso) nella g\_tf; nella loss inserire il periodo di tempo

Il fatto è: dobbiamo metterlo solo nella g\_tf, o nella f\_tf(in solver)? Risposta: la f\_tf deriva dalla formula di V in algo\_1

Nella g\_tf, come inseriamo le varie tempistiche dei pagamenti?

Il bsde\_solver, in "train", prende in input i sample di dw e x (valid\_data).

AMERICAN

- Estendere il caso della call: valutare ogni istante di tempo se conviene il pricing della opzione:

Il fatto è: DOVE teniamo in conto il tempo di ogni possibile path?

American: una volta simulati i path per intero, ti calcoli il prezzo a ritroso (guarda appunti Maraz)

Il problema è che (ad ogni train) loss = V - g\_tf: entrambi sono calcolati con il path totale.

Quindi il calcolo a ritroso, per ogni path, andrebbe fatto SIA in call di NonsharedModel, dove calcolo la V

SIA in sample di xvaEQUATION (chiamata in train di BSDEsOLVER), dove calcolo g\_tf