10/21: data loading/transform check

general notes

notebook: `data-transform-checking.ipynb`

must read from df instead of frameparser. only using first 5000 rows to start with since 1.2gb csv

data contained in `G:/datasets/csgo/example_frames.csv`

best idea is probably to set up a pytorch dataset/dataloader for this. inspiration code from capstone project. starting from `https://github.com/cs3026/nyu-cds-capstone-baseball-2020/blob/events/event_classification/data_helper.py`, commit `53a3c5a`

a few changes since last time:

- different column names
- player name not available (should i be using playerid instead? or steamid? not sure if playerid is consistent across games)
- data now has matchid/map name in it, instead of being supplied via parser object

column names are easy to take care of. playerid is hopefully consistent across games (but must be verified with steamid). matchid/map name should be fine, i can iterate over unique combinations.

playerid consistency verification

> have to read in entire dataframe and collect playerid/steamid

> we have 7.8m rows. lol. verifying that there's only one steamid per playerid via groupby on unique steamid

> great. looks like in the first 5 rows there are already two steamids for playerid 3. verifying

> playerid 3 has steamids 76561198113666193, 76561198017671061 . looking at the steam profiles, they're completely different players (the first is french, the second is finnish). not sure what causes playerid to change, so probably best to use steamid in the data transform in case playerid is not consistent in a game

> would this info be in the csgo library anywhere? doesn't look like it in csgo/parser/frameparser.py (commit 903f9e2)

> just gonna use steamid i guess. sounds more consistent than playerid. must confirm with peterx

concat both the distance matrix and the extra data? e.g. (N, 10, 12) shape. i think this makes sense and probably works

running into a couple "setting with copy" issues with `df['pos']` and it looks like autoreload extension doesn't work for this. not sure this matters, but would be nice to fix eventually. added TODO

looks like as long as `game_map` is given to `data_transform.transform_data` it still works fine after changing colnames. will combine distances with extra data via `stack`, then time to write dataset

stacking was a lot more difficult than imagined, lol. `cat` + reshaping before and after concatenation. distance matrix still accessible through `[:, :, :10]` (so can be used for cnn)

how do i separate train/val/test?

> can this be random and then placed in different folders? -> this is what i did, essentially separating by map/match. looks like there's only like 50 map/match combos though? -> reading directly from file, indeed, there's only 55 combos... maybe split should be per round? -> this yields 1295 rounds, probably better to use this. though i would still prefer more rounds. i thought we had hundreds of matches? did peterx ever get to putting that stuff on the google cloud db he wanted to?

how do i get the target?

> maybe look at peterx's paper/code? can probably take the `RoundWinnerSide` column from rounds to do this. i'm not sure whether this would work directly but hopefully.

seems like the dataset/dataloader work, although kind of slowly. probably because of the IO requirement. it might be better to store the transformed samples in train/val/test instead of the originals? though that would be annoying to re-do if transform changes, and it would be difficult to run different transforms (or add stuff like armor values).

for train i got 508444 samples, in total there are 844207. that's 60.2% which is pretty spot on to what i wanted. for val/test i got 168060, 167703 (~20% each)

possible issue: correlated samples (like chess WP model) is this a problem? should maybe test out with the chess data?

looks like the concat/stacking isn't working properly. getting some very weird behavior. gotta look into that again. however, targets are now working by using rounds data. almost done

easy fix. just switch view(10, 12) to view(12, 10) and (optionally) transpose. distance matrix is now [:10, :] obviously.

now that i can load the data in, i should be able to train any nets/algos pretty easily. one worry that i have is that it takes a little long to do the transform (5s per sample?), but i'm not sure what the limiting factor is here. i am also still pretty annoyed at the warning it keeps spitting out.

removing the actual area calculation changes from 4.74s per sample to 4.37s. so not the problem. probably the pivot itself which calls this function pairwise. not sure how to fix

looks like this isn't working with batch sizes greater than 1. lol. gives weird numbers for the size of distance matrix. looks like dataloader passes several items at once to the transform...

actually, looks like some assumption was violated. match 58, de_inferno, round 17 has 5496 rows. why? shouldn't this have 10 players per tick? so much work to transform this data...

wow. looks like some of these games are 8v8 and not 10v10. great. now what? another peterx question. for now, i think i can count distinct steamid's per side and use that in the transform instead of 10

ok, fixed... but now the batch size > 1 still doesn't work, because the tensor sizes are mismatched (since some games have 10 players, some have 8, etc...). not sure what to do here. maybe the best call is to just run with `batch_size=1`? other option would be padding with 0's maybe? padding feels wrong. also not sure if it even makes sense to have the distance matrix now. also just realizing that inputs will be different to the net, so this entire thing is kind of broken.

:put_litter_in_its_place:

sounds like i already have a delay to my schedule, and it's just the first day. lol.

---

summary

things done

- adapted code to new csgo library version/data given
- wrote pytorch Dataset and transform function to go from row-wise per-tick per-player to (player x player) distance matrix + additional data
- verified that unique players are indeed unique
- reshaped data to be (N_samples, 10, 12) (i.e. concatenated the additional data to the distance matrix)
- train/val/test split by having a folder for each + files for each round in them (60-20-20)
- target generation via the `example_rounds.csv` file, using `RoundWinnerSide`

questions for peterx

- games with fewer than 10 players? what do we do in this situation? padding? (how?)
    - discard, they must be warmup rounds/artifacts of the parser
- steamid consistent across games? (should be, but to confirm)

    - they are indeed consistent/unique
- i thought there were hundreds+ of games? there are only 55 matches in this data (1295 rounds). did he ever get to putting stuff up on the google cloud sql server he wanted?
    - this is sample data, there's a lot more that he can send me

next steps/remaining action items

- fix data size for games with fewer than 10 players (or change data format for it to somehow be consistent)
- basic baseline model (per map? like peterx paper? maybe a tuned logistic regression?)