11/15: final results

daily plan

after a day off yesterday, it's time to get back on that grind. next off day is probably only after thanksgiving, when this project is hopefully done and when the rest of my projects are around 3/4 through.

today is basically the final model iteration. writing a FCNN/CNN combo model which can take the player count for each side and do a logreg on that, plus the distances - that means hopefully it will guaranteed outperform baseline, since it's only adding things to a LR -, removing the mandatory early stopping and making it a CLI argument, and then running some of the best models from last week for a lot of epochs without early stopping.

hopefully all this will give an improvement over baseline. finally, i'll take the best performing model and train it on train+val, then run it on the test set and see its out-of-sample performance. if time allows, i'll also start the writeup today, at least to define the general structure i want to follow.

- [x] cli arg for early stopping
- [x] no early stopping experiments with best model
- [x] build fcnn/cnn combo model
- [x] run fcnn/cnn combo model
- [ ] analyze results/run best model on test set
- [ ] start writeup

early stopping cli arg

this shouldn't prove too difficult, it's just wrapping the early stopping stuff in an if structure.

as expected, that took me like 2 lines of code, haha. now: running a few of the best models with no early stopping (just in case there's some noise in the validation process). i'm going with the `CSGO-WP results 2` excel sheet, `2nd iter`, rows 2, 6, 12, 22, 25, all for 100 epochs. with the exception of row 6, these all had early stops, and were doing pretty decently on AUC/log loss. row 6 only had 25 epochs but it also did pretty well. in the meantime, i'll start building the fcnn/cnn combo model.

lol, apparently argparser's bool stuff doesn't work like i expected. passing in anything turns it to True, so the only way to make it work the way i intended is setting the default to False. this should probably be fixed in some way that argparser intended bools to be passed in (`store_true` or something?) but i don't think i care enough right now.

fcnn/cnn combo

this is gonna be an interesting one. i think it's easy to set up a combo conceptually, but i'm not sure i can achieve the exact same as a LR just by setting up *any* FCNN.

setting up the forward pass seems easy enough. let's start there. divide the 2 channels with player alive-ness and pass that through linear blocks, pass the rest through CNN.

forward pass works fine. the setup also didn't involve too much work (mostly just renaming vars because they collided), but getting the number of elements in the end of the conv block is a little weird. this is gonna have to be hardcoded in (but the input is anyway; so it should be fine).

ok well now it works. i forgot that i needed one final linear layer (2 elements -> 1 element), and that i needed to change the linear input size (6 channels -> 2 channels), as well as the fact that i didn't have to change the cnn stuff (it was all based on H/W anyway, not on C).

now to make it better: a lot of the data in the fully connected section is all 0's anyway and only the diagonal matters. so let me go extract that and change the input size to 10.

ok cool, it works. that took a little bit longer than expected because... pytorch documentation sucks. as usual. but now it's working fine. i'm still running the 100epoch stuff without early stopping so it'll be a while before i can run this but i'm hopeful it will work great.

i'm also gonna hardcode in the dropout/batch norm stuff because it's different on the cnn vs the linear level.

finally, let's add the new model to the cli args.

actually, this might have to wait. i don't want the training of the long-running models to fail because of some syntaxerror because the `train.py` file is being edited.

ok i got tired of waiting because i did the mistake *again* of running with early stopping. so i added it in to `train.py` already. i hope it's not broken. anyway - it'll be a while before i can get back to active work on this, it'll just be passively working in the background training the models.

experiments

the results will be in `CSGO-WP results 2`, sheet `3rd iter`, for both the no early stop stuff and the combo model.

---

summary

things done

- implemented cli arg for early stopping
- new LR-CNN combo model
- ran experiments without early stopping and for LR-CNN model

questions for peterx

- n/a

next steps/remaining action items

- run final hyperparameter set for determining # of epochs
- run final hyperparameter set on train+val and see test performance
- writeup