**Trl changes**

**Major**

Integrated text world repo for text adventure with trl repo for PPO.

Separate Value Head from LM. Fixes model loading errors and makes it far easier to swap models. Tested with GPT2, GPT2-XL, GPT-Neo-1.3B. Any HuggingFace model that can output hidden states should work.

KL divergence term with exact gradients on the current model. Importance sampling corrected for the distribution of the last model.

Implemented Rejection Sampling. Either keep top % or top N trajectories.

**Optimizations**

Pytorch Lightning Deepspeed for PPO, and HuggingFace Trainer Deepspeed for Rejection Sampling. Enables fitting much larger models and larger batch sizes into the same GPU RAM. (Untested multi gpu support)

Batching for both the environment and text gen forward passes in gameplay. Batching for all Forward and backward passes in PPO. Loss calculations from the model outputs are still per sample.

Caching logprobs and values from gameplay loop. Eliminates 1 forward pass over every batch at beginning of training loop. Still need to do a reference model forward pass over all data in this spot.

Don’t save reference model and optimizer state with pytorch lightning. Was using over 20 GB and several minutes for each saved checkpoint.

**Minor**

Fixed discount rate order of ops error. Reward + gamma \* (v\_next – v) instead of reward + gamma \* v\_next – v. It appears the base repos solution to this issue was a default value of 1 for gamma.

Added a field for next value to allow discounting across multiple experiences in the same game. Separate discount rate for cross token and cross step discounting.

Changed format to python files instead of notebooks.

Changed control flow to have trainer run the gameplay loop instead of the gameplay loop running training.

Environment files for both conda and pip only.

String buffer memory that puts previous states and actions into the current context.

Change prompt formats. State, info, You <output action> causes the model to repeat the sentence “You are carrying nothing” from the info. “What do you do?” appears to work better, but the model is not selecting one of the commands in the list.

**Future**

Tune KL value. Too high and the model learns to output empty strings. At zero the model collapses to spamming the word shape. These results were trained on a reward function of the number of Es within a maximum of 20 tokens.

Add KL for Rejection Sampling.

Add Forward KL

Use only the last 3rd of the model for training and take the intermediate output of reference model as input. Could save GPU Ram and decrease training time.

Add examples of valid or correct actions to the context window. Both allows the model to explore more states before learning how to solve the first one and gives it an example it can use. Either an option for human input or a random sample from admissible commands.

Use more complex games, larger models, chain of thought prompting, and GEM.

Still running out of RAM for larger context lengths. Tune Deepspeed params, and batch size to fix.