What the project does

Spread deep learning task to multiple clients in order to potentially speed up the calculation. The server sends weight of model to clients, clients recreates the model, calculate weight increment, then sends increments back to server. Server adds increment to model weights.

Why I chose it:

Distributed training is a tool used to speed up deep learning. I want to explore it more.

Technical challenges:

1) serialization of weights. Weights are Pytorch tensors that needs to be transformed into binaries. Multiple solutions are explored. First I tried str() and ast.literal_eval(), then repr(), finally decided to use pickle package and its dumps() and loads() method (reason for this decision see 2), which writes python objects directly into binary.
2) Recv() buffer size. I initially set buffer size for recv() to be 32, which is too small for the weights of the model. I did not realize this when debugging so I thought str() and repr() is buggy. Then I figured out this buffer size problem and changed it to 10000 and the problem was solved.