1. Pytorch recap
2. Custom layer
   1. Not fully connected
   2. No biases
   3. Binary weights (if(abs(wegiht) < thresh) then weight 🡺 0, else(weight 🡺 1))
      1. STE method
3. Custom loss function
   1. Penalty & penalty strength

Next steps:

* Look into the data generation process, I think it would be better to explicitly generate the data from a multivariate normal distribution and set the correlation of the predictors to some low value
* And then just add the noisy predictors
* Before the simulations write all the layers and custom loss functions into a script and import that script (think about setting a seed)
* Simulate
  + Don’t change any hyperparameters yet, just try out different optimizers with different options
  + Then simulate for different values of the cutoff parameter and the penalty strength
  + Also look at the architecture after the masking layer