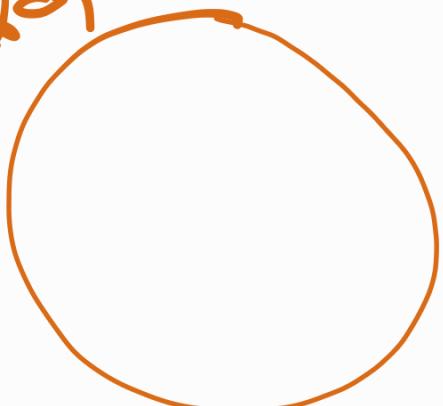
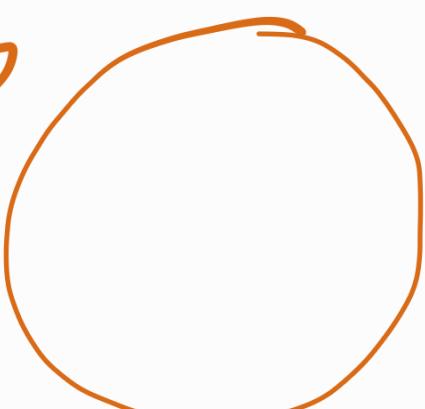


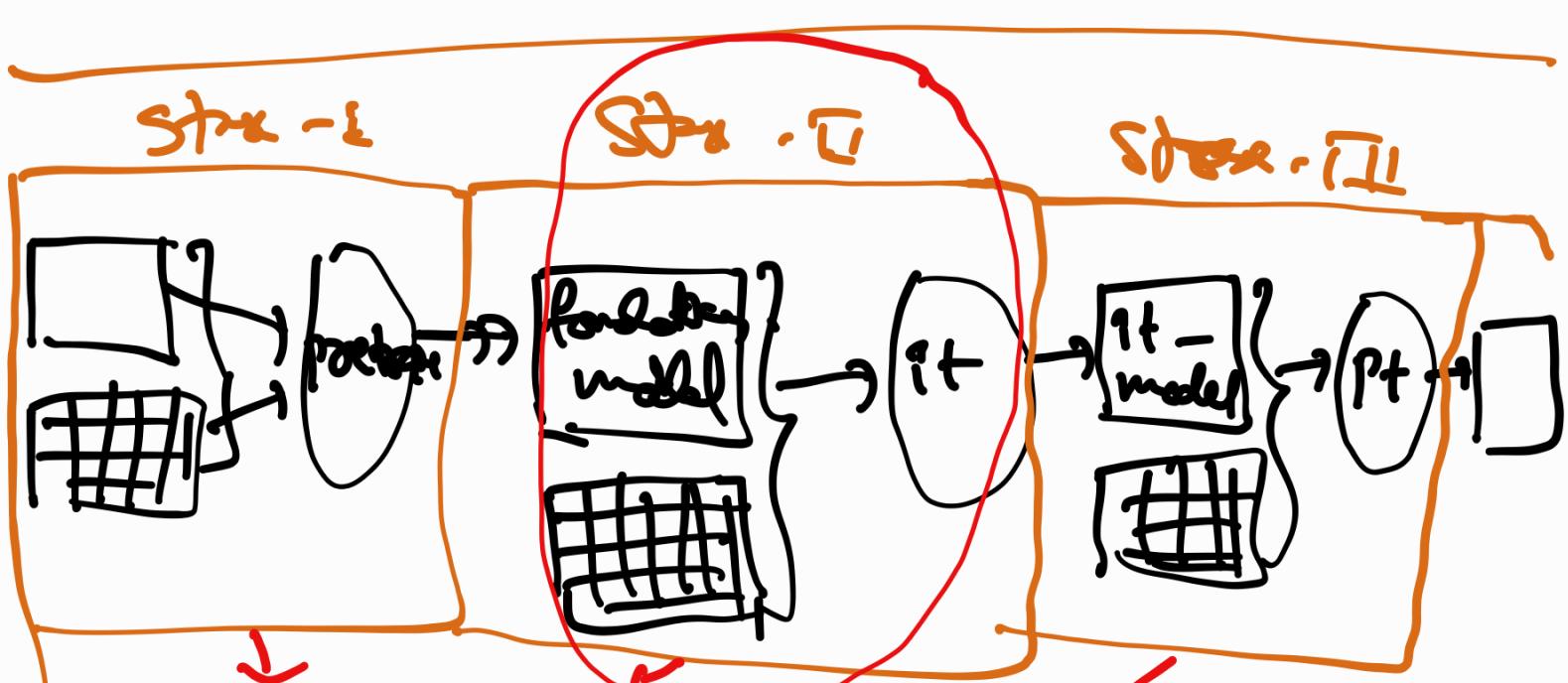
application



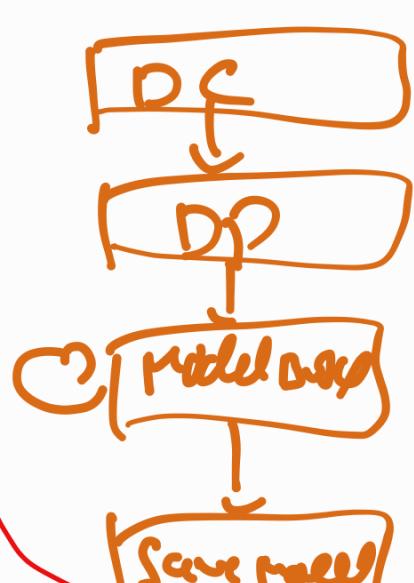
building
Bustine



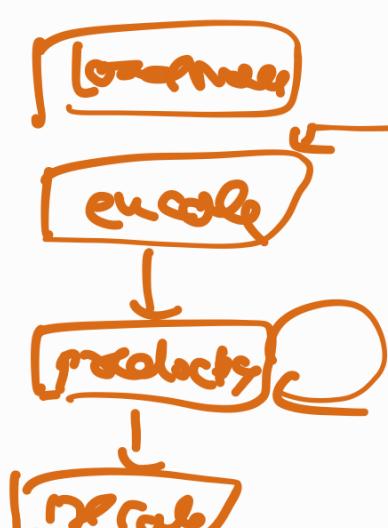
Centr.
Scrypt



tokens



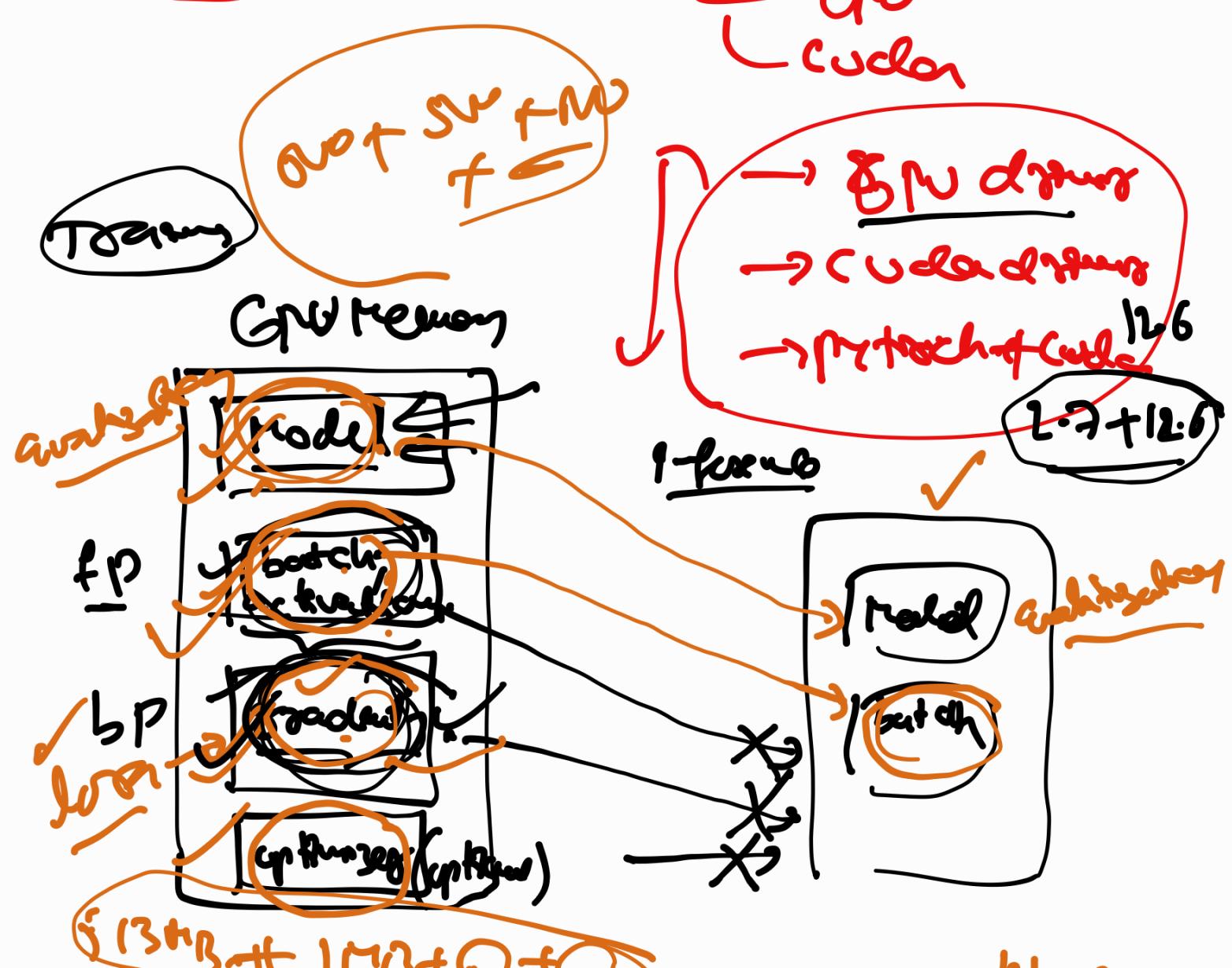
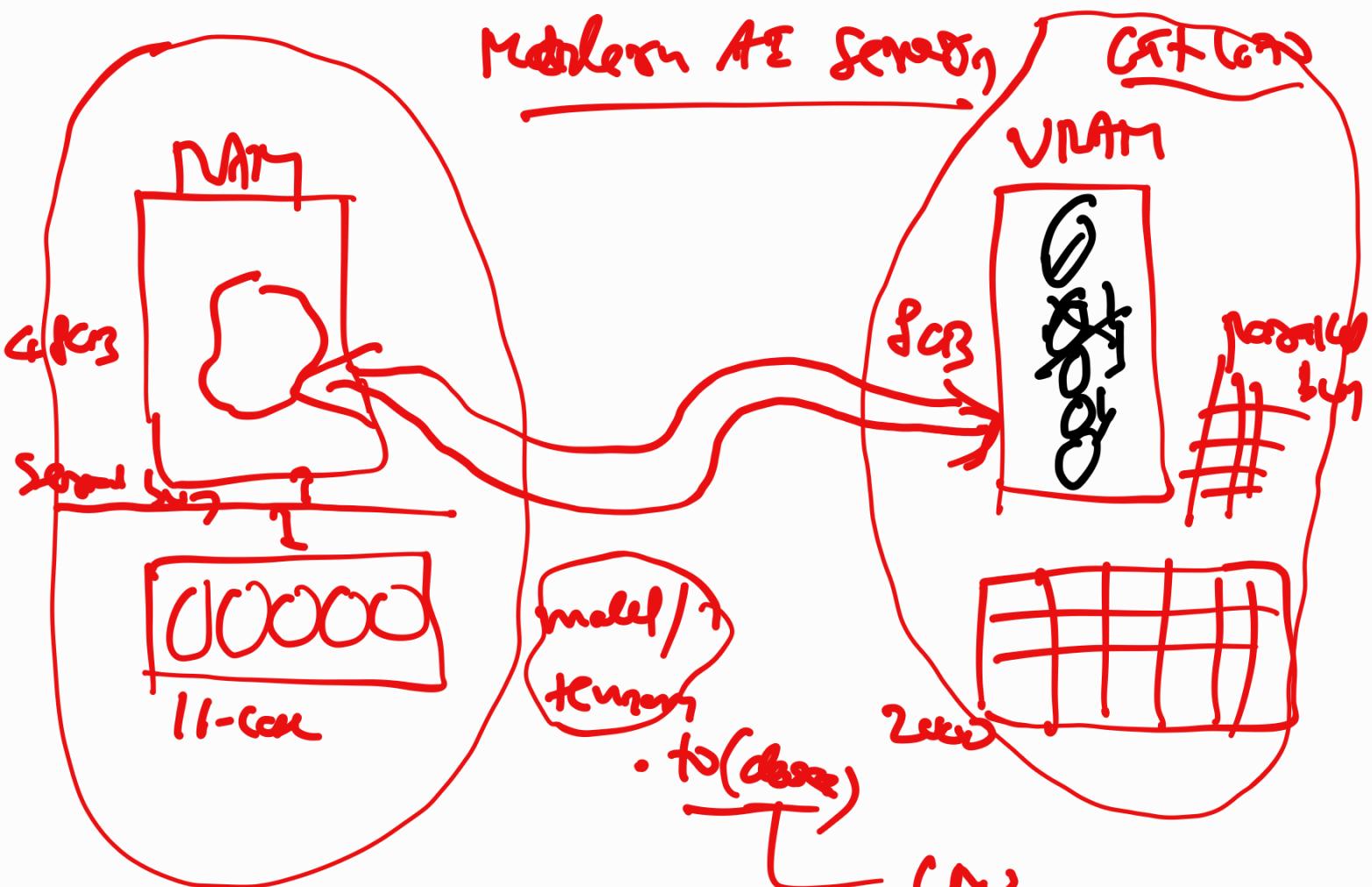
Inference



-> pytorch
-> OHF

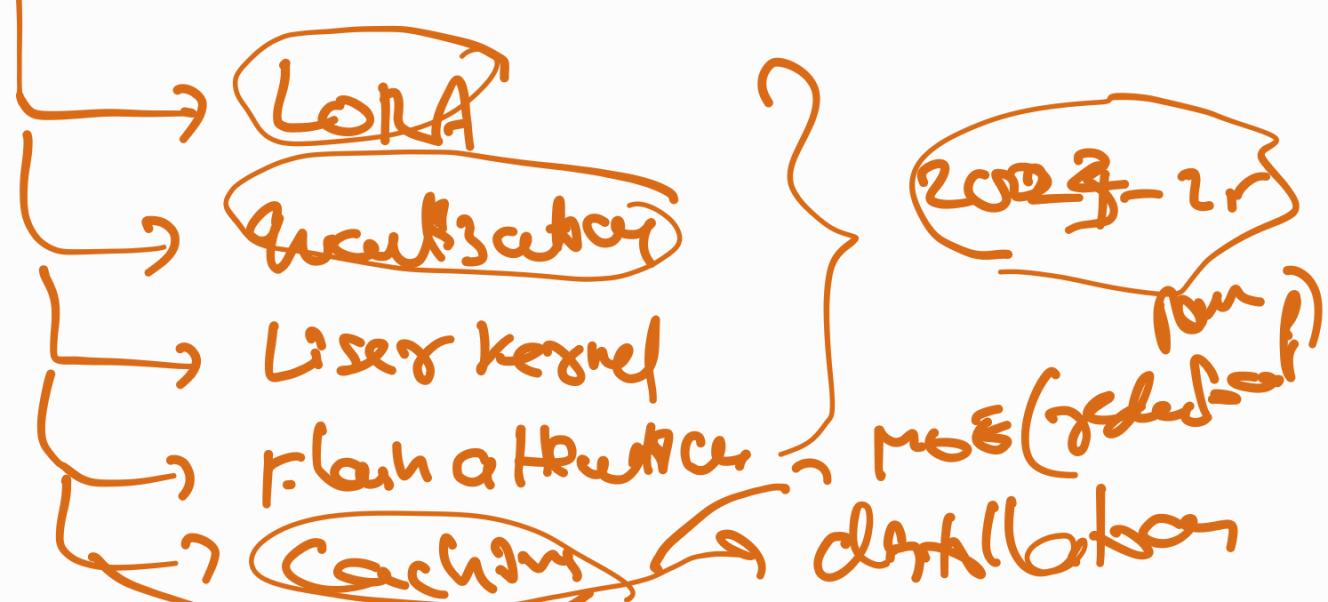
-> =

64GB
multi GPU → 1000
algo



Efficiency techniques to optimize

GPU memory effectively



| <gradient> | w1

$$2 \times 10 \times 4$$

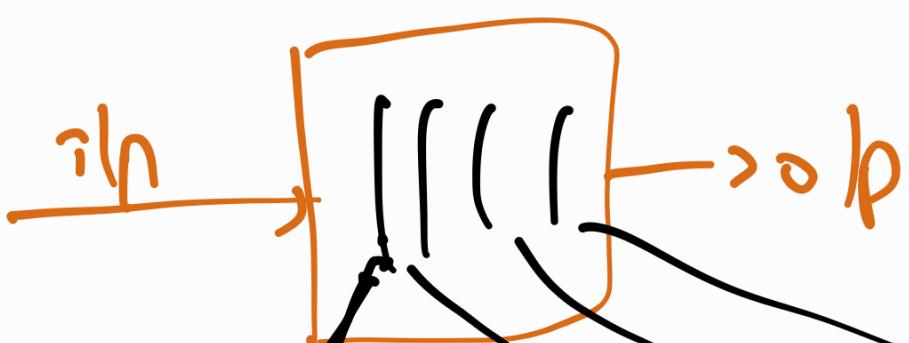
$$8 \times 6 \times 3 = 8 \text{ GB}$$

$$2 \times 10 \times 4 \times 4$$

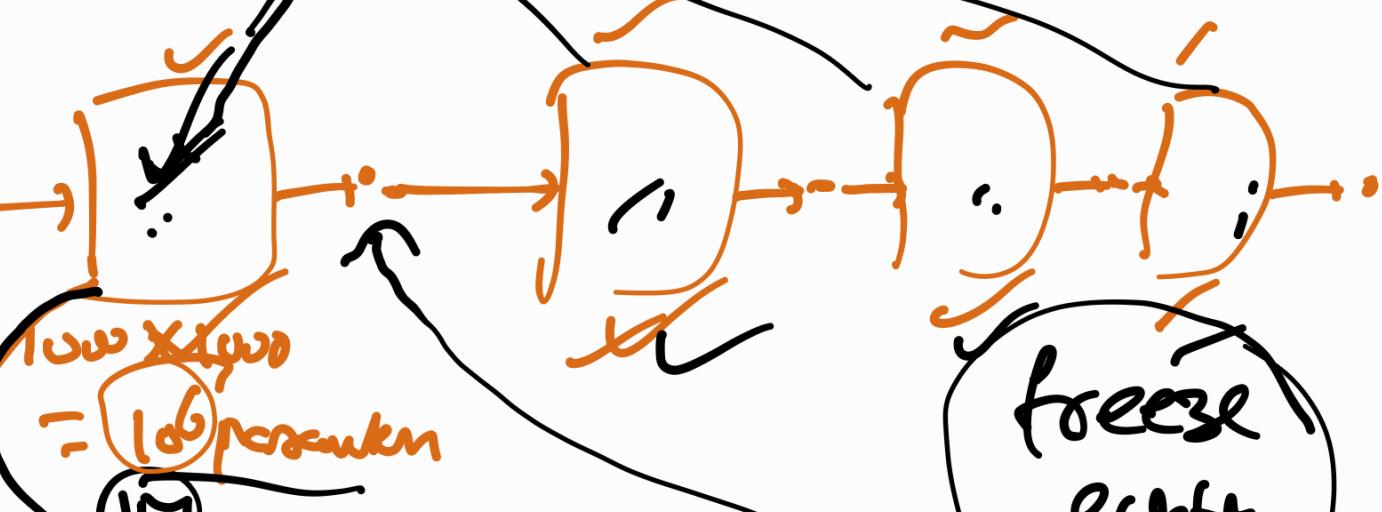
$$8 \text{ GB}$$

↓ Cost 3 -
45

Low Rank Adapter (LoRA)



Learnable
naturally
avoids + blocks



1000 X 1000
= 100 parameters
1M

freeze
embed
layer

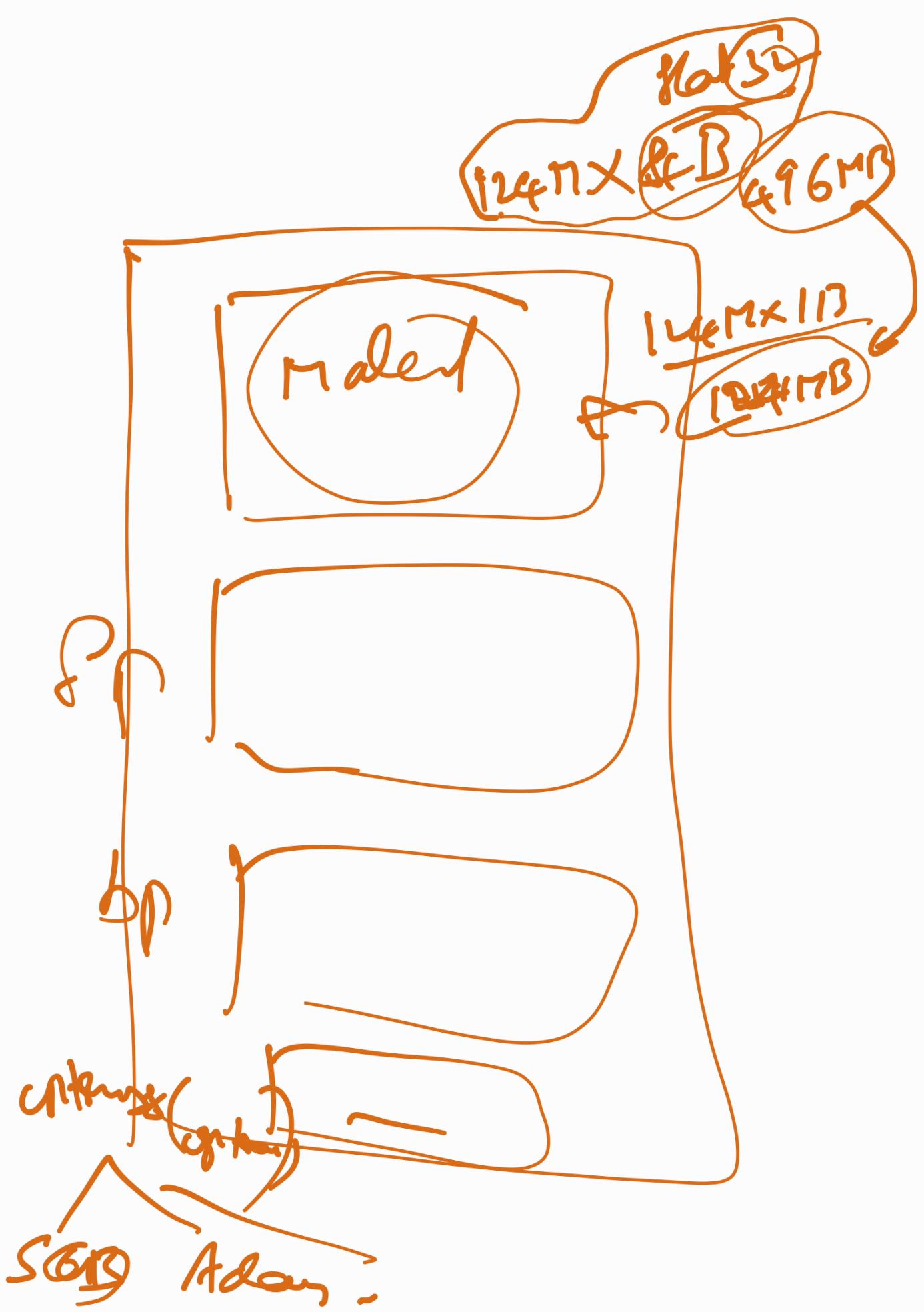
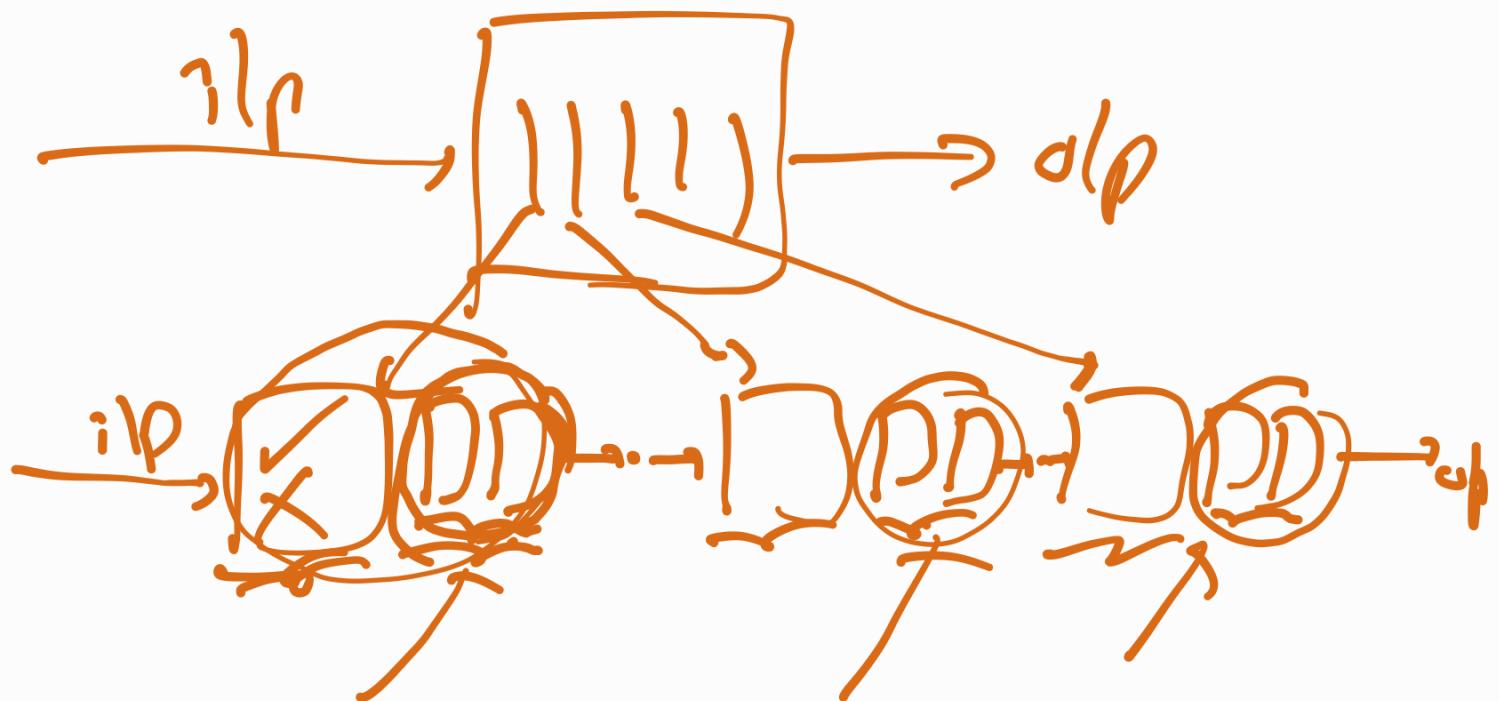
train
loss
layer

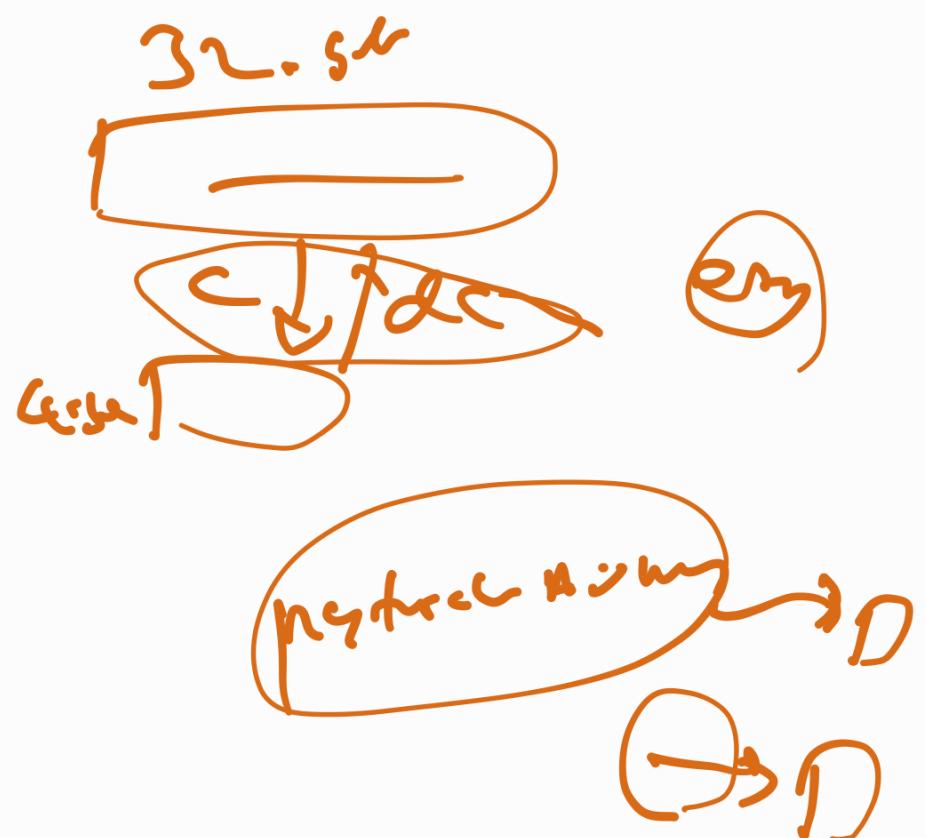
$$8000 + 1000 = 16000 \text{ parameters}$$

16k

- faster to train
- less memory footprint

only linear
layer





mother - k - 2

