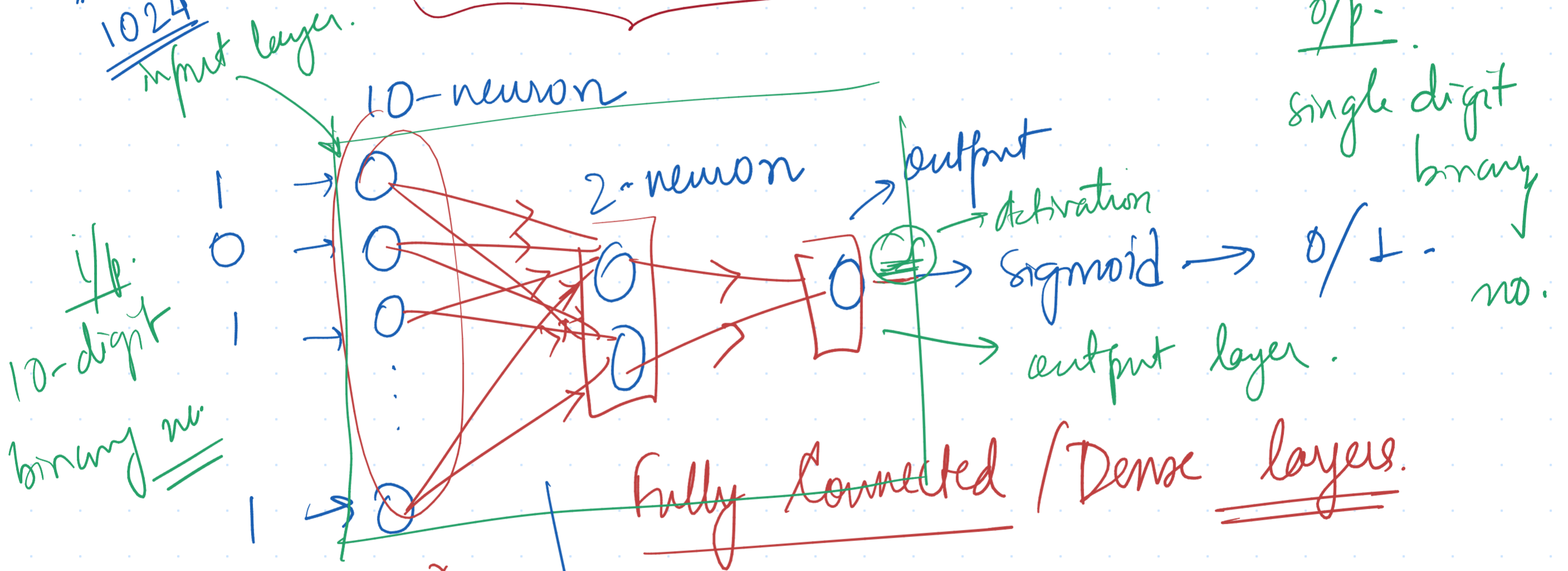
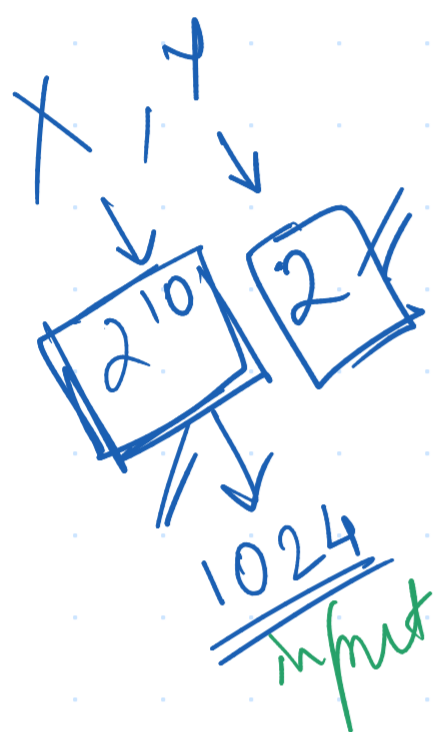


# Designing classifier Pipeline using Pytorch.

16/02/2025

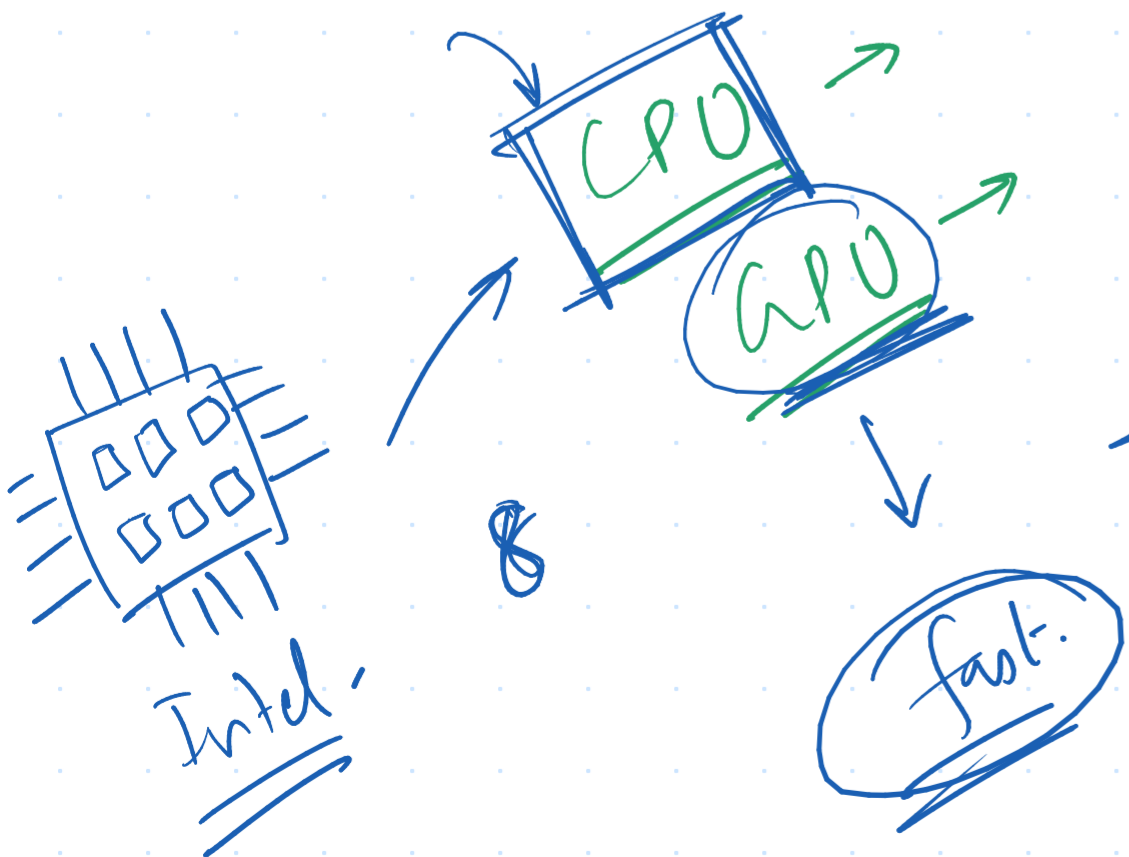
Task: Create a simulated dataset for binary digits.  
10-digit binary numbers.

# 1s > # 0's then  $\rightarrow 1$   
 $\begin{matrix} 0101011111 & \rightarrow & 1 \\ 0000001111 & \rightarrow & 0 \\ 1010101010 & \rightarrow & 0 \\ 1111110000 & \rightarrow & 1 \end{matrix}$



Pytorch.

2-layer neural network.  $\rightarrow$  for this task.

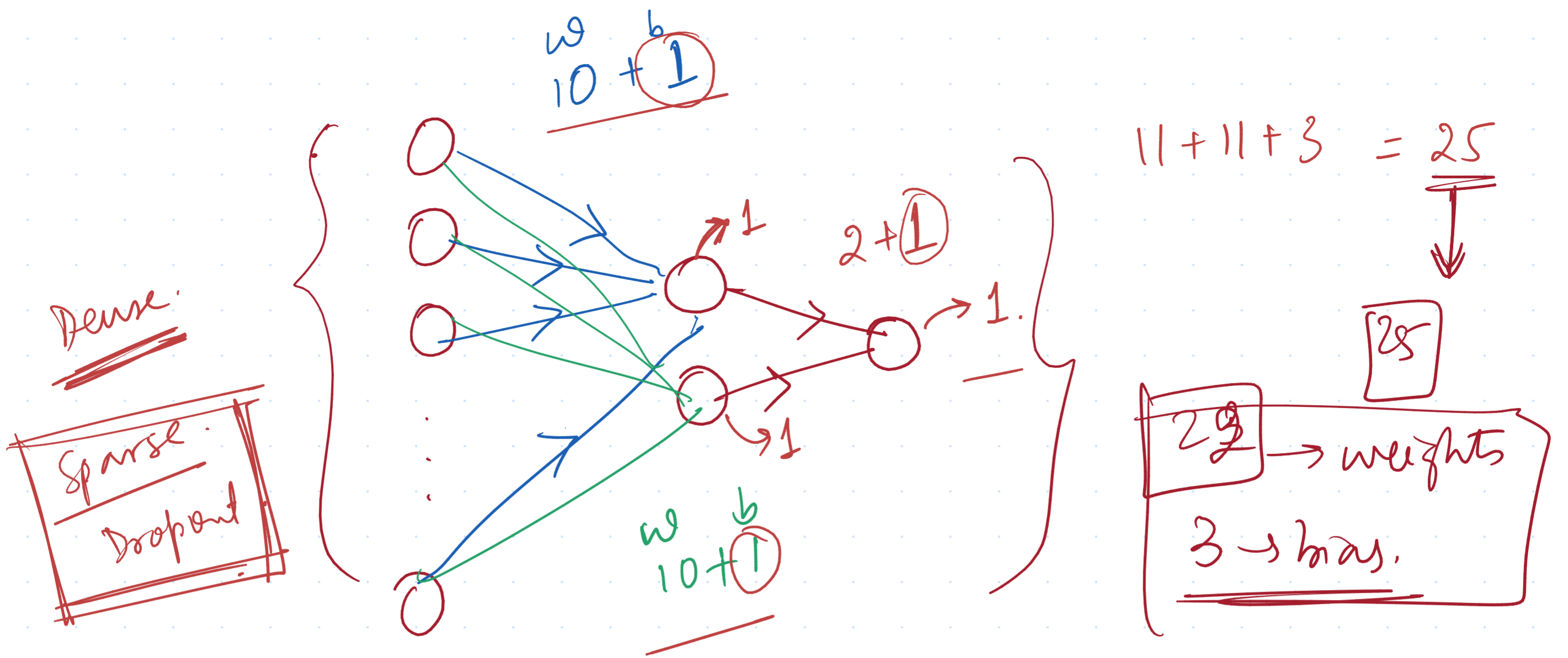


2-layer # param  $\rightarrow$  less.

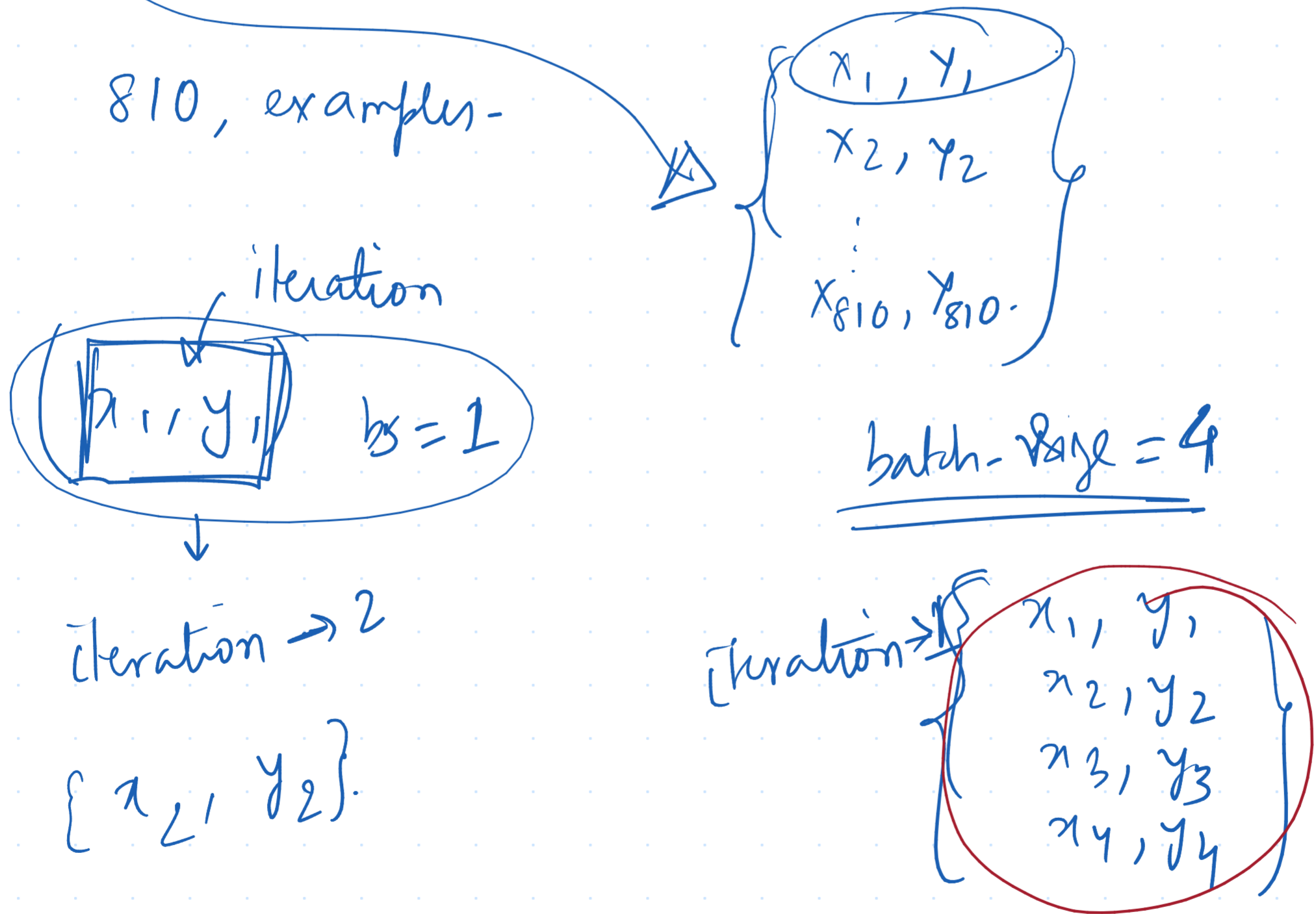
epoch  $\rightarrow$  metrics & loss.

same checkpoint etc.

- Dataloader
- Training pipeline
- Validation pipeline
- Test pipeline  $\rightarrow$  inference
- train-val-test epoch



train-epoch ( ) / val-epoch ( ) / test-epoch ( )  
 ↓  
 train-loader, model, optimizer.



$bs \Rightarrow 16$   
 $\{x_1, \dots, x_{16}\} \rightarrow iter-1$   
 $\{x_{17}, \dots, x_{32}\} \rightarrow iter-2$

iteration 2  $\{x_5, y_5, \dots, x_9, y_9\}$

model.

gradient accumulat  
mitigation

0/1

eval() valid.  
model.train() → model's param are updated by backprop.

x, y → device CPU/GPU

optimizer.zero\_grad() → ?

y ← model ← x

pred-y →  $\hat{y}$

loss → BCE. (y,  $\hat{y}$ )

loss.backward() → backpropagate.

optimizer.step() → params update via optimizer.

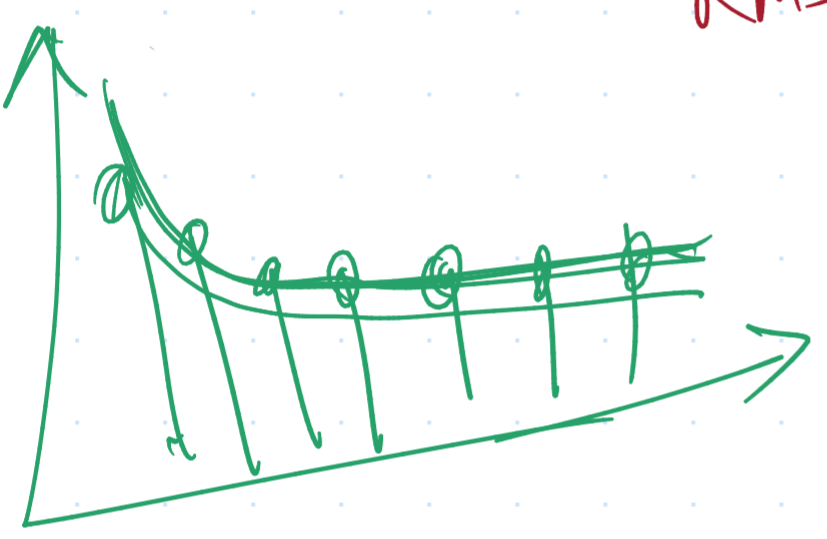
SAD/Adam  
RMSprop etc.

dump loss.

print.

loss add

total-loss → dump.



redundant example  
#1 > #0's

weights / bias.

$\begin{pmatrix} 0 \\ 0 \\ 0 \\ 0 \end{pmatrix}$

Model. → ↓

if #1 > #0's  
print(1)  
else  
print(0)

General.

AI

Model is simulating a program which is a complex function by learning (weights & bias) → approximate

model  $\rightarrow$  functional Approximator.

