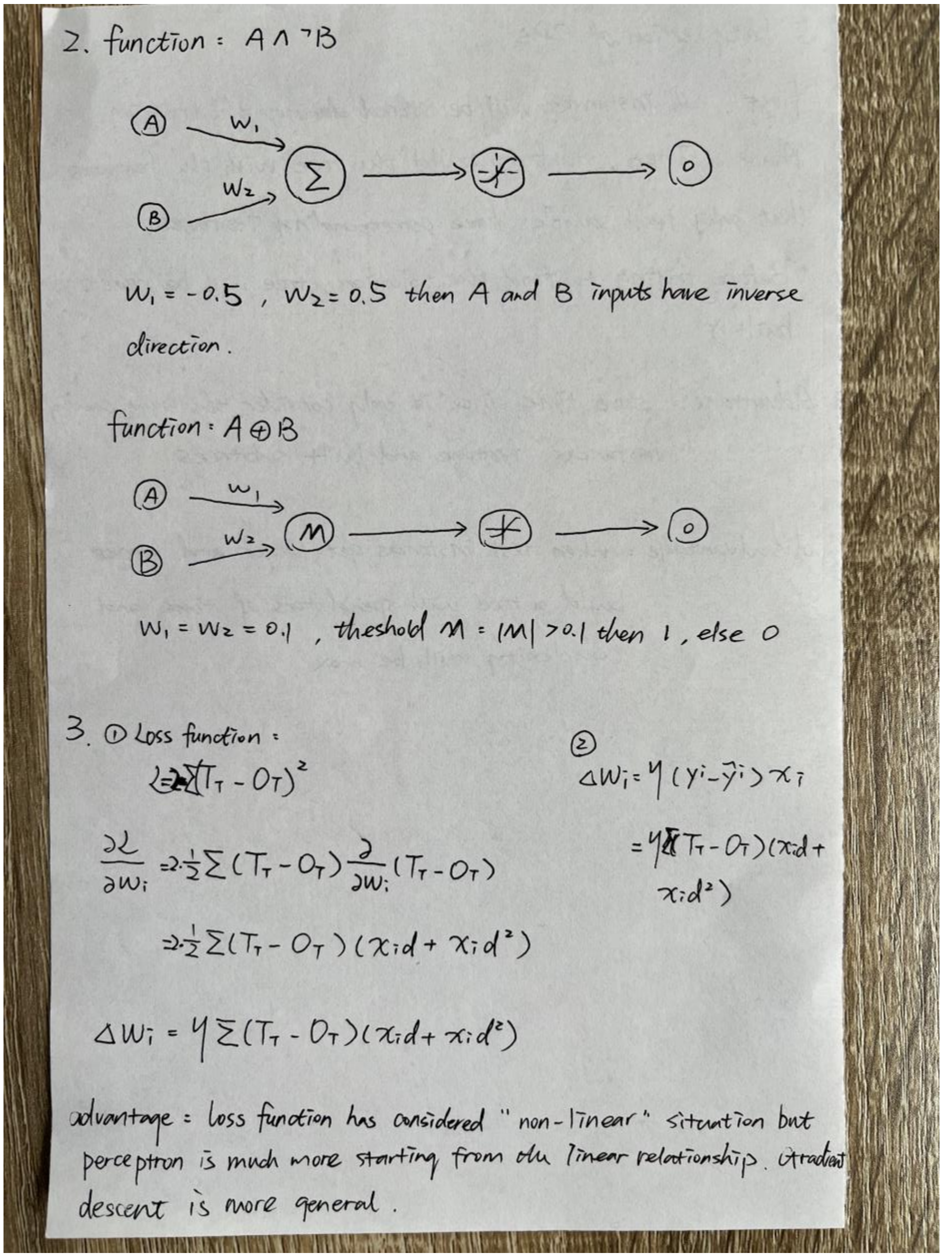


2. function : $A \wedge \neg B$

A
w₁
B
w₂
Σ
o

$w_1 = -0.5$, $w_2 = 0.5$ then A and B inputs have inverse direction.

function : $A \oplus B$

A
w₁
B
w₂
M
o

$w_1 = w_2 = 0.1$, theshold M : $|M| > 0.1$ then 1 , else 0

3. ① Loss function :

$$\mathcal{L} = \frac{1}{2}\sum(T_T - O_T)^2$$

$$\frac{\partial \mathcal{L}}{\partial w_i} = 2\cdot\frac{1}{2}\sum(T_T - O_T)\frac{\partial}{\partial w_i}(T_T - O_T)$$

$$= 2\cdot\frac{1}{2}\sum(T_T - O_T)(x_i d + x_i d^2)$$

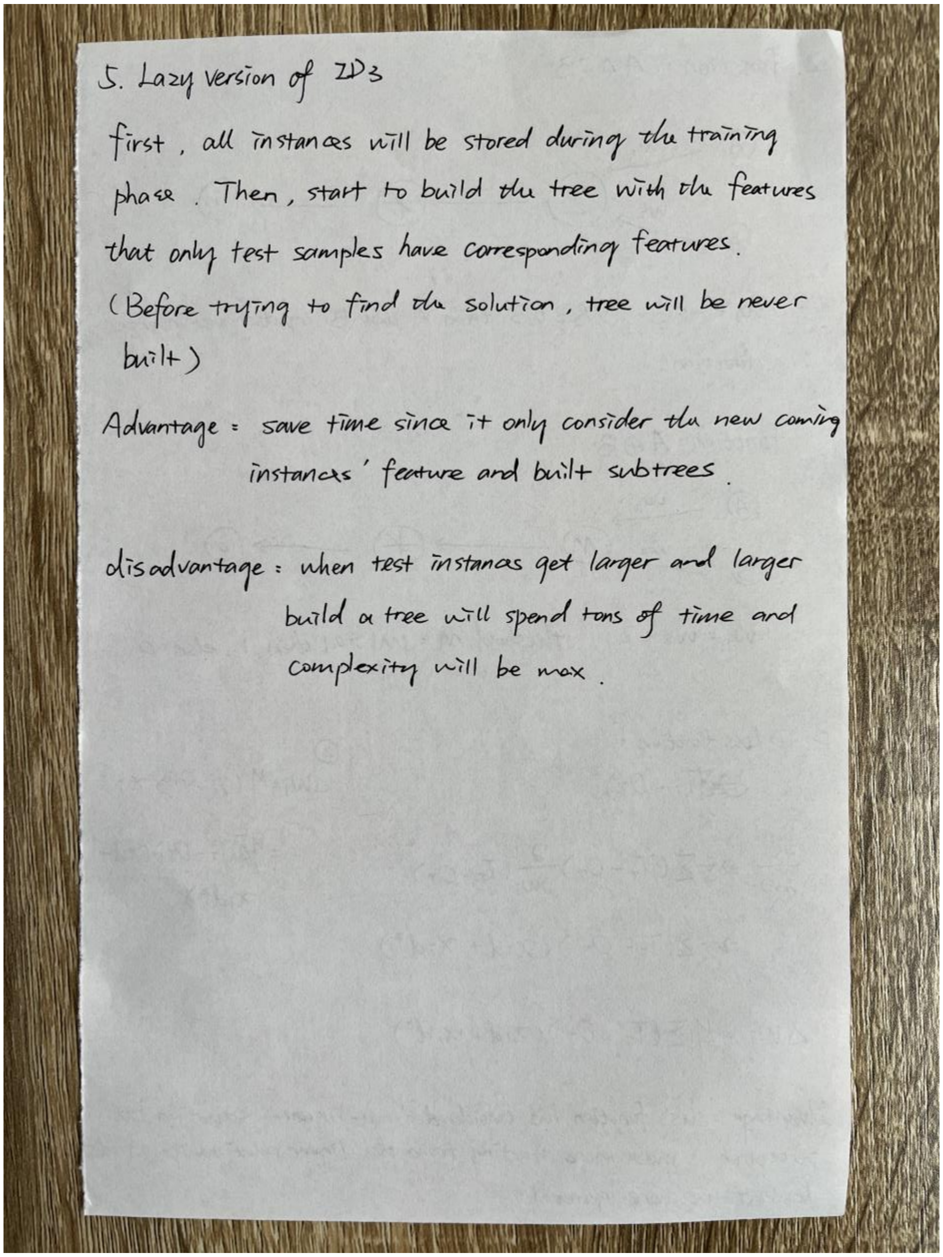
$$\Delta w_i = \eta\sum(T_T - O_T)(x_i d + x_i d^2)$$

②

$$\Delta w_i = \eta(y^i - \hat{y}^i)x_i$$

$$= \eta\sum(T_T - O_T)(x_i d + x_i d^2)$$

advantage : loss function has considered "non-linear" situation but perceptron is much more starting from the linear relationship. Gradient descent is more general.



5. Lazy version of ID3

first, all instances will be stored during the training phase. Then, start to build the tree with the features that only test samples have corresponding features.
(Before trying to find the solution, tree will be never built)

Advantage: save time since it only consider the new coming instances' feature and built subtrees.

disadvantage: when test instances get larger and larger build a tree will spend tons of time and complexity will be max.