

# Perceptron Algorithm

1. Start with random weights  $w_1, \dots, w_n, b$
2. for every misclassified point  $(x_1, \dots, x_n)$ :
  - if prediction == 0:
    - for  $i=1 \dots n$  : update  $w_i = w_i + \alpha x_i$
    - update  $b$  to  $b + \alpha$
  - if prediction == 1:
    - for  $i=1 \dots n$  : update  $w_i = w_i - \alpha x_i$
    - update  $b$  to  $b - \alpha$

\* Perceptron algorithm has to be generalized for non-linear stuff.

\* Error function = Distance : how far are we from the solution?

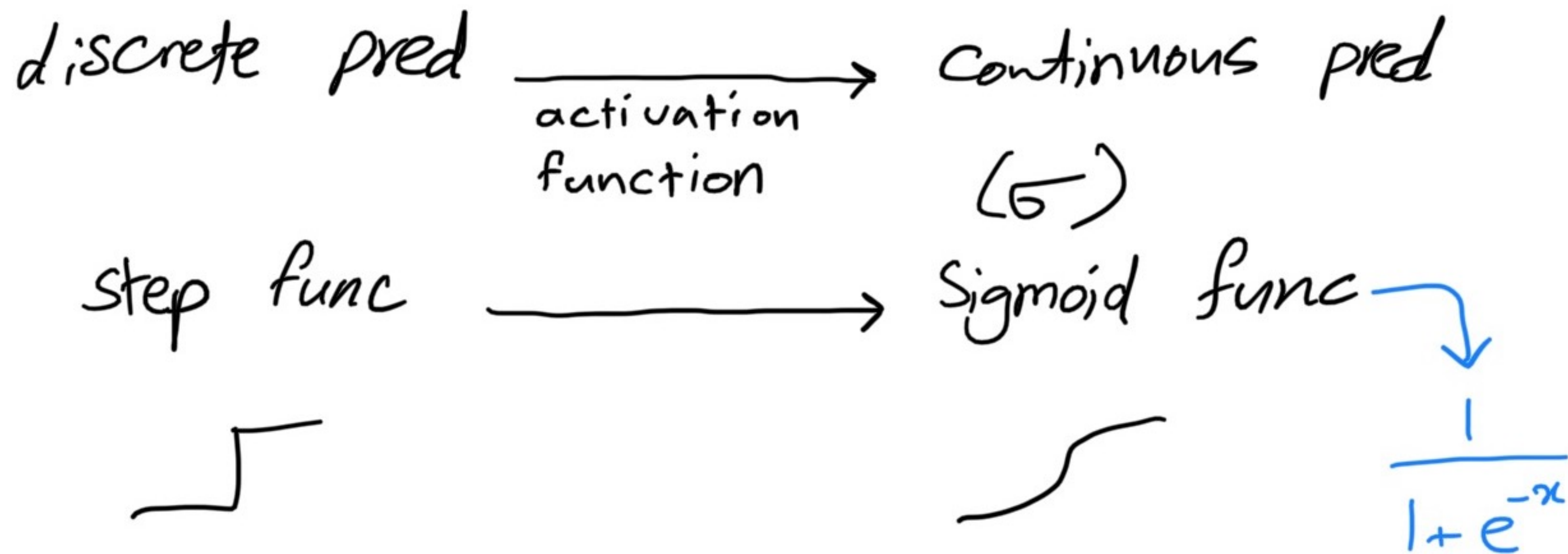


## Gradient Descent:

minimizing the error.

error function must be differentiable  
continuous

So, how do we go from discrete to continuous?



$wz + b$  gives us  $\hat{y}$ . we have to apply  $\sigma()$  to  $\hat{y}$ .

when the score is zero, probability is 50%.