Neural Net from scratch

Design: a two-layer network with 3 input and 3 node at the hidden layer, and one node at output layer that output [0, 1]

Usage: Simulate $a \wedge (b \vee c)$

Hidden layer: sigmoid
Output layer: sigmoid

back prop

$$\frac{dLoss}{dW} = 2(y - \hat{y}) \cdot d\_sig(out) \cdot input$$

$$Weights = weights + \alpha \cdot \frac{dLoss}{dW}$$

$$cost = (actual - out)^2$$

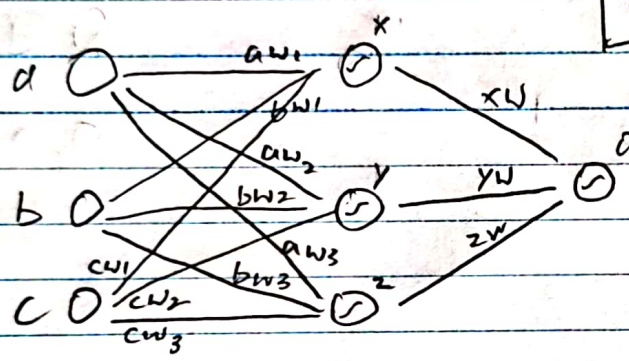
$$d\_cost = 2(actual - out)$$

xn = normalized x

α : learning rate

$$sigmoid(z) = \frac{1}{1 + e^{-z}}$$

$$d\_sigmoid(z) = sig(z)[1 - sig(z)]$$



forward prop

$x = a(aw_1) + b(bw_1) + c(cw_1)$
$y = a(aw_2) + b(bw_2) + c(cw_2)$
$z = a(aw_3) + b(bw_3) + c(cw_3)$
$xn = sigmoid(x)$
$yn = sigmoid(y)$
$zn = sigmoid(z)$
$out = x(xw) + y(yw) + z(zw)$
$out\_n = sigmoid(out)$

back prop

$xw = xw + (d\_cost \times deriv\_sigmoid(out\_n) \times x) \times \alpha$
$yw = yw + (d\_cost \times deriv\_sigmoid(out\_n) \times y) \times \alpha$
$zw = zw + (d\_cost \times deriv\_sigmoid(out\_n) \times z) \times \alpha$
$aw1 = aw1 + [[d\_cost \times deriv\_sig(out\_n) \cdot xw] \times d\_sig(xn) \times a] \times \alpha$
$bw1 = bw1 + [[d\_cost \cdot d\_sig(out\_n) \cdot xw] \cdot d\_sig(xn) \cdot b] \times \alpha$
$cw1 = cw1 + [[[d\_cost \cdot d\_sig(out\_n) \cdot xw] \cdot d\_sig(xn) \cdot c] \times \alpha$
$aw2 = aw2 + [[d\_cost \cdot d\_sig(out\_n) \cdot yw] \cdot d\_sig(yn) \cdot a] \times \alpha$
$bw2 = bw2 + [[d\_cost \cdot d\_sig(out\_n) \cdot yw] \cdot d\_sig(yn) \cdot b] \times \alpha$
$cw2 = cw2 + [[d\_cost \cdot d\_sig(out\_n) \cdot yw] \cdot d\_sig(yn) \cdot c] \cdot \alpha$

back prop

$$aw3 = aw3 + [[d\_cost \cdot d\_sig(out\_n) \cdot zw] \cdot d\_sig(zn) \cdot a] \times \alpha$$
$$bw3 = bw3 + [[d\_cost \cdot d\_sig(out\_n) \cdot zw] \cdot d\_sig(zn) \cdot b] \times \alpha$$
$$cw3 = cw3 + [[d\_cost \cdot d\_sig(out\_n) \cdot zw] \cdot d\_sig(zn) \cdot c] \times \alpha$$