1. Regularization is a technique to prevent overfitting by penalizing complex models.
Weights that close to 0 is prefered, and it panalizes those weights have high absolute value since $\lambda > 0$.

2. When $\lambda = 0$, it ignores to panalize w
when $\lambda \to +\infty$, it much prefer to panalize w
When $\lambda \to -\infty$, it prefer a w that much more fit the given training data.

3. $\frac{\partial J}{\partial w_0} = \frac{1}{n} \frac{\partial}{\partial w_0} (y_i - hw(x_i))^2 + \frac{\partial}{\partial w_0} \lambda \sum_{j=1}^{p} w_j^2$

$= \frac{1}{n} \sum_{i=1}^{n} -2(y_i - hw(x_i)) + 0$

$= \frac{2}{n} \sum_{i=1}^{n} (y_i - hw(x_i))$

$\frac{\partial J}{\partial w_i} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w_i} (y_i - hw(x_i))^2 + \frac{\partial}{\partial w_i} \lambda \sum_{j=1}^{p} w_j^2$

$\frac{\partial J}{\partial w_i} = \frac{1}{n} \sum_{i=1}^{n} \frac{\partial}{\partial w_i} (y_i - hw(x_i))^2 + \frac{\partial}{\partial w_i} \lambda \sum_{j=1}^{p} w_j^2$

$= \frac{1}{n} \sum_{i=1}^{n} -2 (y_i - hw(x_i)) (-x_i) + \frac{\partial}{\partial w_i} \lambda 2 w_i$

4. The different between this gradient descent and regularized linear gradient descent, it has one more

3. $\frac{\partial J}{\partial w_0} = \frac{1}{n}\frac{\partial}{\partial w_0}\sum(y_i - h_w(x_i))^2 + \frac{\partial}{\partial w_0}\lambda\sum w_i^2$

$= \frac{1}{n}(-2)\sum_{i=1}^{n}(y_i - h_w(x_i))$

$= -\frac{2}{n}\sum_{i=1}^{n}(y_i - h_w(x_i))$

$\frac{\partial J}{\partial w_j} = \frac{1}{n}\sum_{i=1}^{n}(y_i - h_w(x_i))(-x_i) + 2\lambda\sum_{j=1}^{p} w_j$

~~$= \frac{-x_i}{n}\sum(y_i - h_w(x_i)) + 2\lambda w_j$~~ $= \frac{1}{n}\sum_{i=1}^{n}(y_i - h_w(x_i))(-x_i) + 2\lambda\sum_{j=1}^{p} w_j$

4. The difference between two gradient descent is

$w_0 \leftarrow w_0 + \alpha\sum_{i=1}^{n}(y_i - h_w(x_i))$

$w_j \leftarrow w_j + \alpha\left[\sum_{i=1}^{n}(y_i - h_w(x_i)) \cdot x_i\right] + \alpha\left[2\lambda\sum_{j=1}^{p} w_j\right]$

which has one more element considered in w_j

When update w_0, there is no difference,

but update w_j will in become larger since $\lambda > 0$.

When w is positive, the loss function will get a large value

when w is negative, the loss function decreased which means

more overrate.

5. If designer want the agent be more accurate on given training dataset, it is better to set 0 for λ

If designer doesn't want overfitting, it is better to choose a λ larger than 0.

Perceptron

① $z = hw_1 = w_1x_1 + w_2x_2 + w_3x_3 + bx_0 = 0 \Rightarrow 1$

$w_1, w_2, w_3, b = 0$

② $z = hw_2 = w_1x_1 + w_2x_2 + w_3x_3 + bx_0 = 0 \Rightarrow 1$

$w_1, w_2, w_3, b = 0$

③ $z = 0 \Rightarrow 1$

$w_1, w_2, w_3, b = 0$

④ $z = 0 \Rightarrow 1$

$w_1, w_2, w_3, b = 0$

⑤ $z = 0 \Rightarrow 1$

$w_1 = 0.5(0-1)\times 1 + 0 = -0.5 \quad w_2 = 0.5(0-1)\times 0 + 0 = 0$

$w_3 = 0.5(0-1)\times(-2) + 0 = 1 \quad b = 0.5(0-1) + 0 = -0.5$

⑥ $z = -0.5\times(-1) + 0 + 1 + 0.5 = 2 \Rightarrow 1$

$w_1 = 0.5(0-1)\times -1 + -0.5 = 0 \quad w_2 = 0.5(0-1)\times -1 = 0.5$

$w_3 = 0.5(0-1)\times 1 + 1 = 0.5 \quad b = 0.5(0-1)\times 1 + 0.5 = -1$

⑦ $z = 0 - 4\times 0.5 - 1 = -3 \Rightarrow 0$

$w_1 = 0.5(0-0)\times 0 + 0.5 = 0 \quad w_2 = 0.5(0-0)\times -4 + 0.5 = 0.5$

$w_3 = 0.5(0-0)\times 0 + 0.5 = 0.5 \quad b = 0.5(0-0) - 1 = -1$

⑧ $z = 1\times 0 + 0\times 0.5 - 3\times 0.5 - 1 = -2.5 \Rightarrow 0$

$w_1 = 0.5(0-0)\times 1 + 0 = 0 \quad w_2 = 0.5(0-0)\times 0 + 0.5 = 0.5$

$w_3 = 0.5(0-0)\times -3 + 0.5 = 0.5 \quad b = 0.5(0-0) - 1 = -1$

① $z = 4\times 0.5 + 0.5 - 1 = 1.5 \Rightarrow 1$

$w_1 = 0 \quad w_2 = 0.5 \quad w_3 = 0.5 \quad b = -1$

② $z = 2\times 0.5 + 0.5\times 3 - 1 = 1.5 \Rightarrow 1$

$w_1 = 0 \quad w_2 = 0.5 \quad w_3 = 0.5 \quad b = -1$

③ $z = 0.5\times 1 = 1 - 0.5 \Rightarrow 0$

$w_1 = 0.5(1-0)\times 0 + 0 = 0 \quad w_2 = 0.5(1-0)\times 0 + 0.5 = 0.5$

$w_3 = 0.5(1-0)\times 1 + 0.5 = 1 \quad b = 0.5(1-0) - 1 = -0.5$

~~④ $z = 0\times 1 + 0\times 0.5 - 3\times 1 + -0.5 = -3.5 \Rightarrow 0$~~

~~$w_2 = 0.5$ $w_1 = 0.5$~~

④ $z = 0 - 1\times 0 + 4\times 0.5 + 0\times 1 - 0.5 = 1.5 \Rightarrow 1$

$w_1 = 0 \quad w_2 = 0.5 \quad w_3 = 1 \quad b = -0.5$

⑤ $z = 1\times0 + 0\times0.5 - 2\times1 - 0.5 = -2.5 \Rightarrow 0$

$w_1=0 \quad w_2=0.5 \quad w_3=1 \quad b=-0.5$

⑥ $z = -1\times0 + 1\times0.5 - 1 - 0.5 = -2 \Rightarrow 0$

$w_1=0 \quad w_2=0.5 \quad w_3=1 \quad b=-0.5$

⑦ $z = 0\times0 - 1\times0.5 - 0.5 = -2.5 \Rightarrow 0$

$w_1=0 \quad w_2=0.5 \quad w_3=1 \quad b=-0.5$

⑧ $z = 1\times0 - 0\times0.5 - 3\times1 - 0.5 = -3.5 \Rightarrow 0$

$w_1=0 \quad w_2=0.5 \quad w_3=1 \quad b=-0.5$

3.

Sample	N_1	N_2	N_3	N_4	N_5	N_6	N_7	Output
x_1	0	1	1	0	1	1	0	0
x_2	0	1	1	1	0	0	0	0
x_3	1	0	1	1	0	1	0	1
x_4	1	0	0	0	0	1	1	0

Where ~~there~~ N_i is ith cell, totally 7 cells.
empty cell (cell with white color) is 0,
colored cell (cell with gray color) is 1.
~~total~~ one gray area is defined by all connected cells colored. If the ~~total~~ number of gray areas are more than 2, the output is 1, otherwise, 0,
PS: N_1 and N_7 will sign as connected.
learning rate $\alpha = 0.5$, starts ~~as~~ with initial weights of 0,
bias neuron $x_0 = 1$