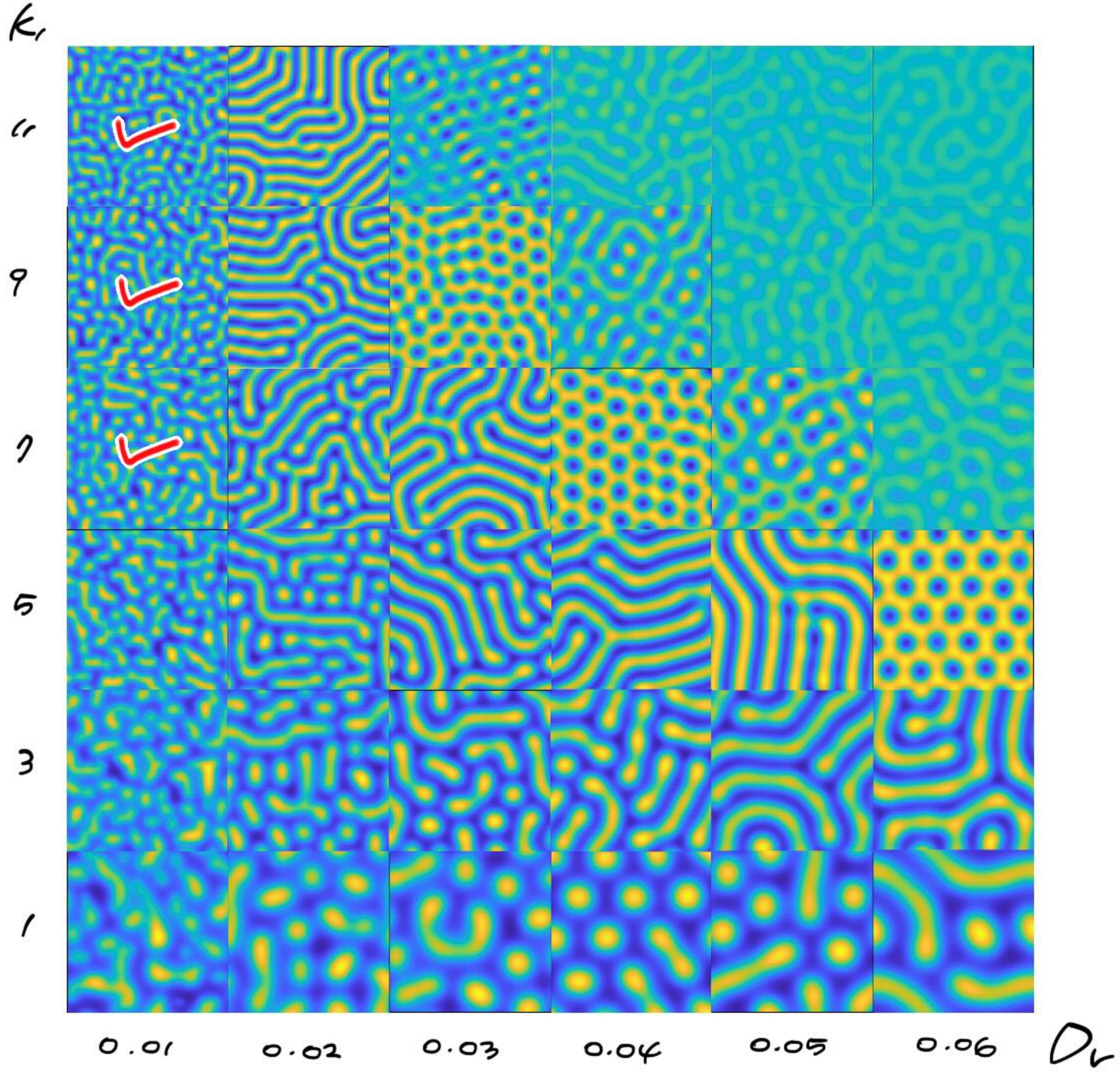
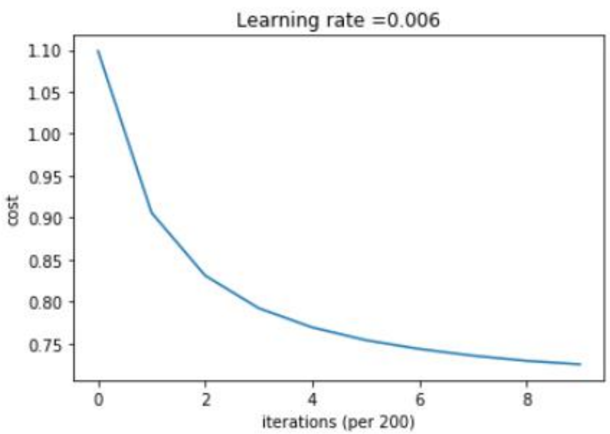


k_1
11
9
7
5
3
1
0.01
0.02
0.03
0.04
0.05
0.06
D_r

- 2000 iteration

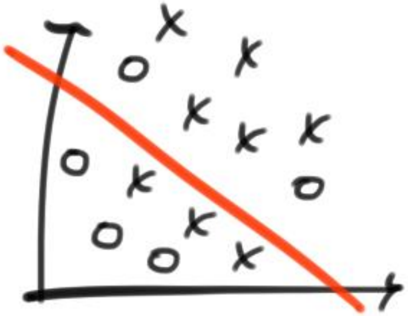
ADAM

```
Cost after iteration 0: 1.098627
Cost after iteration 200: 0.906004
Cost after iteration 400: 0.831171
Cost after iteration 600: 0.792285
Cost after iteration 800: 0.769404
Cost after iteration 1000: 0.754191
Cost after iteration 1200: 0.743681
Cost after iteration 1400: 0.735736
Cost after iteration 1600: 0.729602
Cost after iteration 1800: 0.725414
```



train accuracy : 0.86
test accuracy : 0.34

overfitting

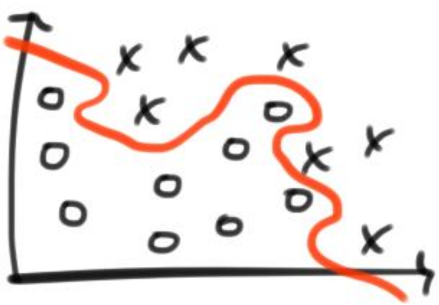


- underfitting

$\iff$ high bias

(1) pick network

(2) train longer



- overfitting

$\iff$ high variance

(1) more data

(2) regularization

$$w = w - \alpha\, dw$$

$$J = \frac{1}{m} \sum L(\hat{y}, y)$$

$\downarrow$ L2 regularization

$$w = w - \alpha\left(dw + \frac{\lambda}{m} w\right)$$

$$J = \frac{1}{m} \sum L(\hat{y}, y) + \underbrace{\frac{\lambda}{2m} \|w\|_2^2}_{\text{penalty}}$$

⊕ Adam

$$
\begin{cases}
v_{dW^{[l]}} = \beta_1 v_{dW^{[l]}} + (1-\beta_1)\frac{\partial \mathcal{J}}{\partial W^{[l]}} \\
v^{corrected}_{dW^{[l]}} = \frac{v_{dW^{[l]}}}{1-(\beta_1)^t} \\
s_{dW^{[l]}} = \beta_2 s_{dW^{[l]}} + (1-\beta_2)(\frac{\partial \mathcal{J}}{\partial W^{[l]}})^2 \\
s^{corrected}_{dW^{[l]}} = \frac{s_{dW^{[l]}}}{1-(\beta_2)^t} \\
W^{[l]} = W^{[l]} - \alpha \frac{v^{corrected}_{dW^{[l]}}}{\sqrt{s^{corrected}_{dW^{[l]}}}+\varepsilon}
\end{cases}
$$

"decoupled weight decay regularization"

① Adam은 손실함수에 L2 norm을 더하여 최적화 해도 일반화 효과를 덜 받게 된다

② weight 식에 직접 weight decay term을 추가하여 이 문제를 해결

Algorithm 2 Adam with L_2 regularization and Adam with decoupled weight decay (AdamW)

given $\alpha = 0.001, \beta_1 = 0.9, \beta_2 = 0.999, \epsilon = 10^{-8}, \lambda \in \mathbb{R}$

initialize time step $t \leftarrow 0$, parameter vector $\boldsymbol{\theta}_{t=0} \in \mathbb{R}^n$, first moment vector $\boldsymbol{m}_{t=0} \leftarrow \boldsymbol{0}$, second moment vector $\boldsymbol{v}_{t=0} \leftarrow \boldsymbol{0}$, schedule multiplier $\eta_{t=0} \in \mathbb{R}$

repeat

$\quad t \leftarrow t + 1$

$\quad \nabla f_t(\boldsymbol{\theta}_{t-1}) \leftarrow \text{SelectBatch}(\boldsymbol{\theta}_{t-1})$ ▷ select batch and return the corresponding gradient

$\quad \boldsymbol{g}_t \leftarrow \nabla f_t(\boldsymbol{\theta}_{t-1}) + \lambda\boldsymbol{\theta}_{t-1}$

$\quad \boldsymbol{m}_t \leftarrow \beta_1 \boldsymbol{m}_{t-1} + (1 - \beta_1)\boldsymbol{g}_t$ ▷ here and below all operations are element-wise

$\quad \boldsymbol{v}_t \leftarrow \beta_2 \boldsymbol{v}_{t-1} + (1 - \beta_2)\boldsymbol{g}_t^2$

$\quad \hat{\boldsymbol{m}}_t \leftarrow \boldsymbol{m}_t / (1 - \beta_1^t)$ ▷ β_1 is taken to the power of t

$\quad \hat{\boldsymbol{v}}_t \leftarrow \boldsymbol{v}_t / (1 - \beta_2^t)$ ▷ β_2 is taken to the power of t

$\quad \eta_t \leftarrow \text{SetScheduleMultiplier}(t)$ ▷ can be fixed, decay, or also be used for warm restarts

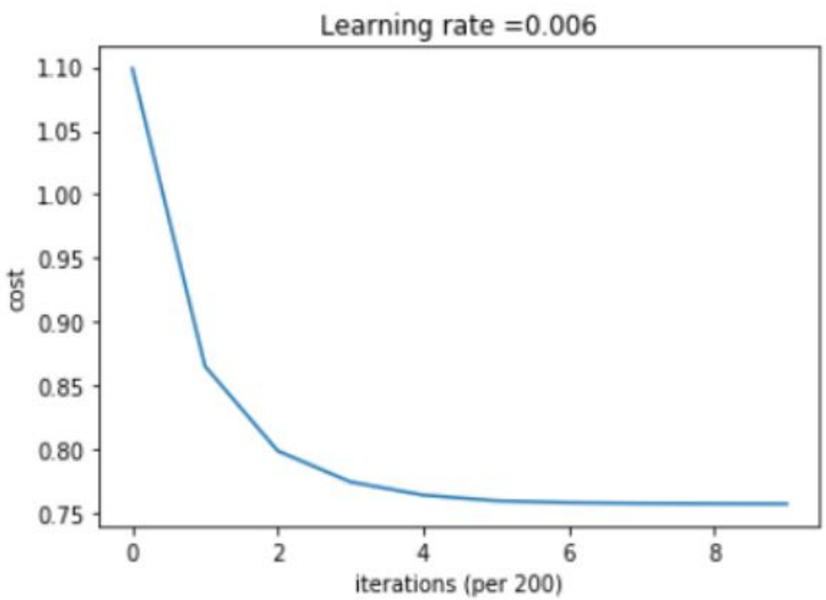
$\quad \boldsymbol{\theta}_t \leftarrow \boldsymbol{\theta}_{t-1} - \eta_t \left(\alpha \hat{\boldsymbol{m}}_t / (\sqrt{\hat{\boldsymbol{v}}_t} + \epsilon) + \lambda\boldsymbol{\theta}_{t-1} \right)$

until *stopping criterion is met*

return optimized parameters $\boldsymbol{\theta}_t$

```
Cost after iteration 0: 1.098858
Cost after iteration 200: 0.865402
Cost after iteration 400: 0.799049
Cost after iteration 600: 0.774712
Cost after iteration 800: 0.764440
Cost after iteration 1000: 0.759916
Cost after iteration 1200: 0.758445
Cost after iteration 1400: 0.757904
Cost after iteration 1600: 0.757688
Cost after iteration 1800: 0.757495
```

```
train accuracy :  0.8583333333333333
test accuracy :  0.32
```

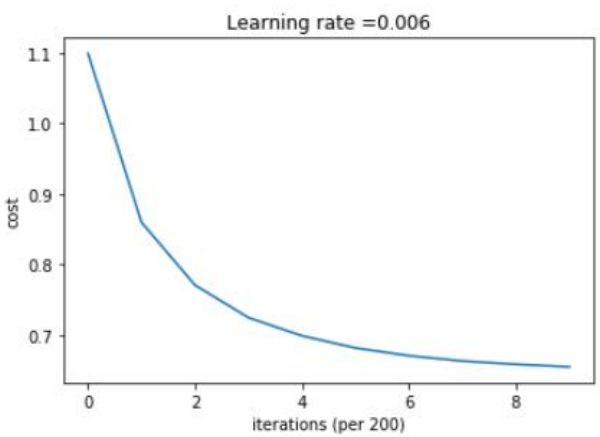


"Double backpropagation"

- **DataGrad** (Double Backpropagation): penalize the $L2$ norm of the gradient of the original loss term with respect to the inputs.

$$L_{DG}(x, y, \Theta) = L(x, y, \Theta) + \lambda \|(\frac{\partial}{\partial x} L(x, y, \Theta))\|_2$$

```
Cost after iteration 0: 1.098838
Cost after iteration 200: 0.859740
Cost after iteration 400: 0.770849
Cost after iteration 600: 0.724785
Cost after iteration 800: 0.699078
Cost after iteration 1000: 0.681866
Cost after iteration 1200: 0.670880
Cost after iteration 1400: 0.663346
Cost after iteration 1600: 0.658710
Cost after iteration 1800: 0.655125
```




```
train accuracy :  0.85
test accuracy :  0.36
```