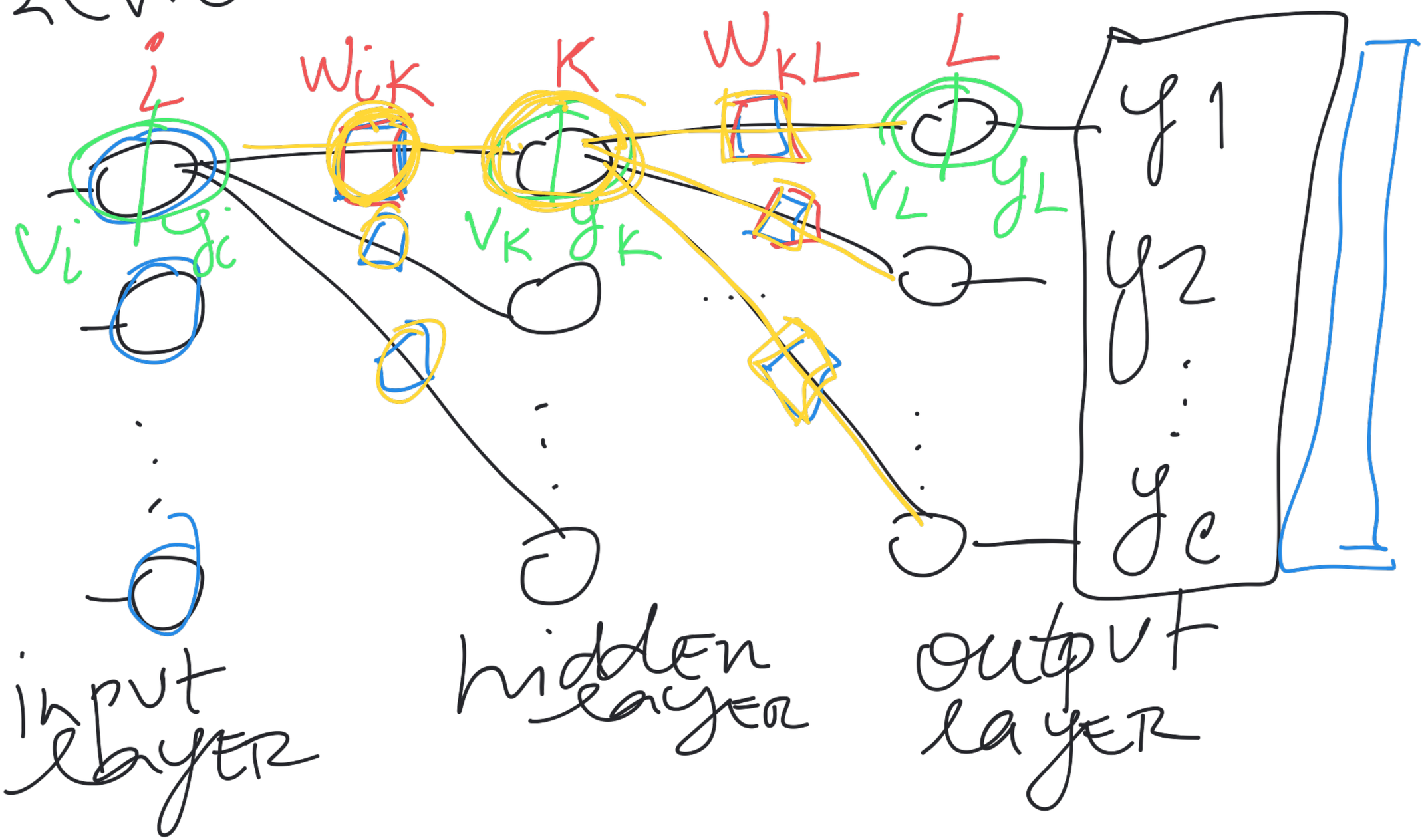
LECTURE 18

REVIEW BACKPROPAGATION



$$X = \{x_1, x_2, \dots, x_N\} \in \mathbb{R}^D$$

$$t = \{t_1, t_2, \dots, t_N\}$$

Could have ONE-hot ENCODING

$t_1 \in C_2$

$$t_1 = \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}$$

$$J = \frac{1}{2} \sum_{i=1}^{N} e_i^2$$

$$= \frac{1}{2} \sum_{i=1}^{N} (t_i - y_i)^2$$

Chain Rule : $-\delta_L$

$$\frac{\partial J}{\partial w_{KL}} = \frac{\partial J}{\partial e_L} \cdot \frac{\partial e_L}{\partial y_L} \cdot \frac{\partial y_L}{\partial v_L} \cdot \frac{\partial v_L}{\partial w_{KL}}$$

$$e_L = t_L - y_L$$

$$J = \frac{1}{2} \sum_{i=1}^{N} e_i^2$$

$$y_L = \phi\left(\sum_K w_{KL} \cdot y_K \right) = \phi(v_L)$$

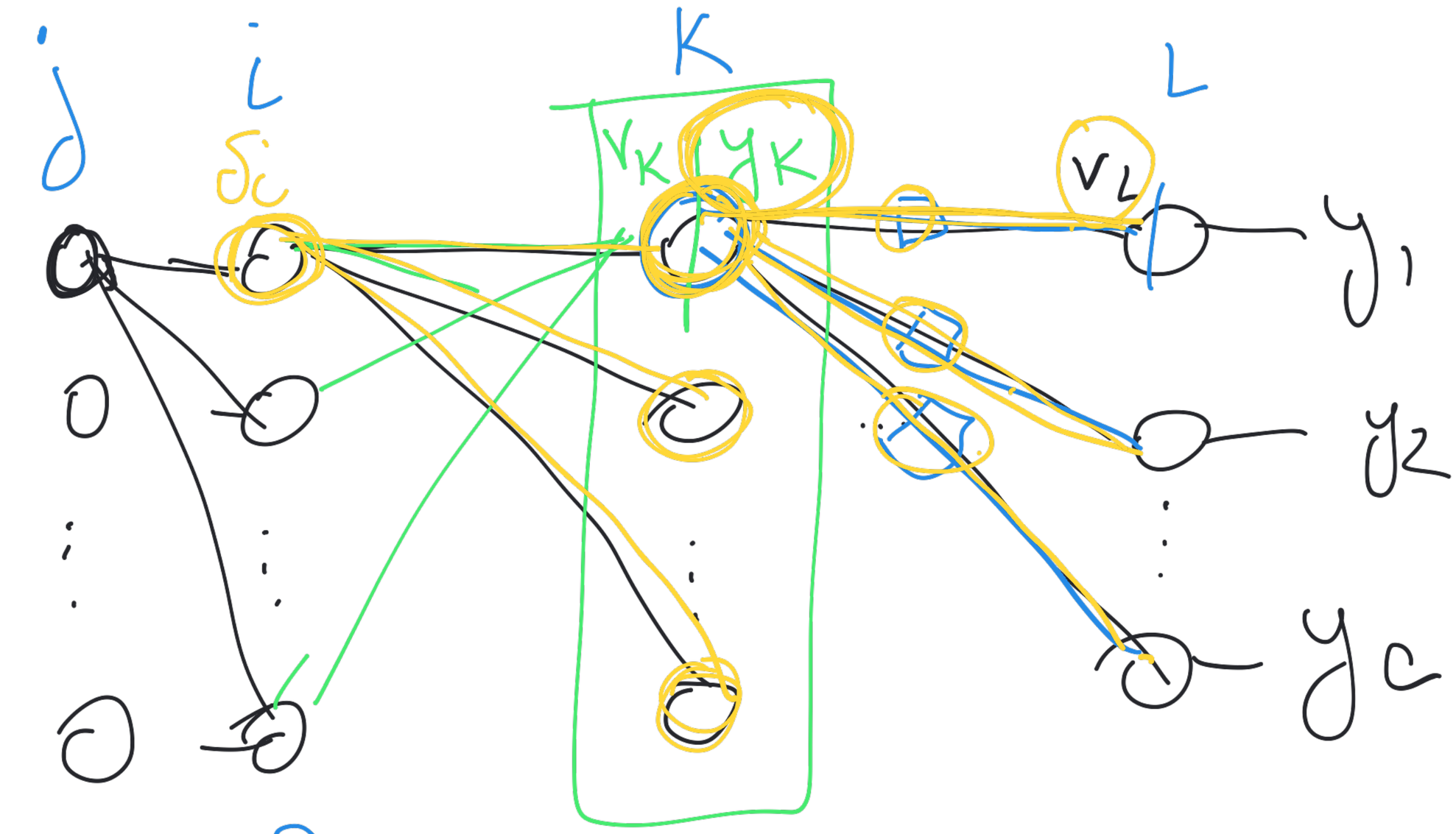
(v_L = the sum inside ϕ)

$$\frac{\partial J}{\partial w_{KL}} = e_L \cdot (-1) \cdot \phi'(v_L) \cdot y_K$$

$$w_{KL}^{(t+1)} \leftarrow w_{KL}^{(t)} - \eta \cdot \frac{\partial J}{\partial w_{KL}}$$

$(t) \leftarrow$ iteration

$\eta \leftarrow$ learning Rate



$$\delta_i = -\frac{\partial J}{\partial v_i} \qquad \delta_K = -\frac{\partial J}{\partial v_K} \qquad \delta_L = -\frac{\partial J}{\partial v_L}$$

Local gradient

$$\delta_k = -\frac{\partial J}{\partial v_k}$$

$$= -\frac{\partial J}{\partial y_k} \cdot \frac{\partial y_k}{\partial v_k}$$

$$\frac{\partial y_k}{\partial v_k} = \phi'(v_k)$$

$$\frac{\partial J}{\partial y_k} = \sum_L \frac{\partial J}{\partial e_L} \cdot \frac{\partial e_L}{\partial y_L} \cdot \frac{\partial y_L}{\partial v_L} \cdot \frac{\partial v_L}{\partial y_k}$$

$$\frac{\partial J}{\partial v_L} = -\delta_L$$

$$\frac{\partial e_L}{\partial y_L} = e_L \qquad \frac{\partial y_L}{\partial v_L} = (-1) \qquad \frac{\partial v_L}{\partial y_k} = \phi'(v_L)$$

$$= \sum_L (-\delta_L) \cdot w_{kL}$$

$$\delta_k = -\frac{\partial J}{\partial y_k} \cdot \frac{\partial y_k}{\partial v_k}$$

$$= \left(\sum_L \delta_L \cdot w_{kL}\right) \cdot \phi'(v_k)$$

Weight Update:

$$\Delta w_{ik}^{(t)} = -\eta \cdot \delta_k \cdot y_i$$

$$w_{ik}^{(t+1)} = w_{ik}^{(t)} + \Delta w_{ik}^{(t)}$$

Output Layer

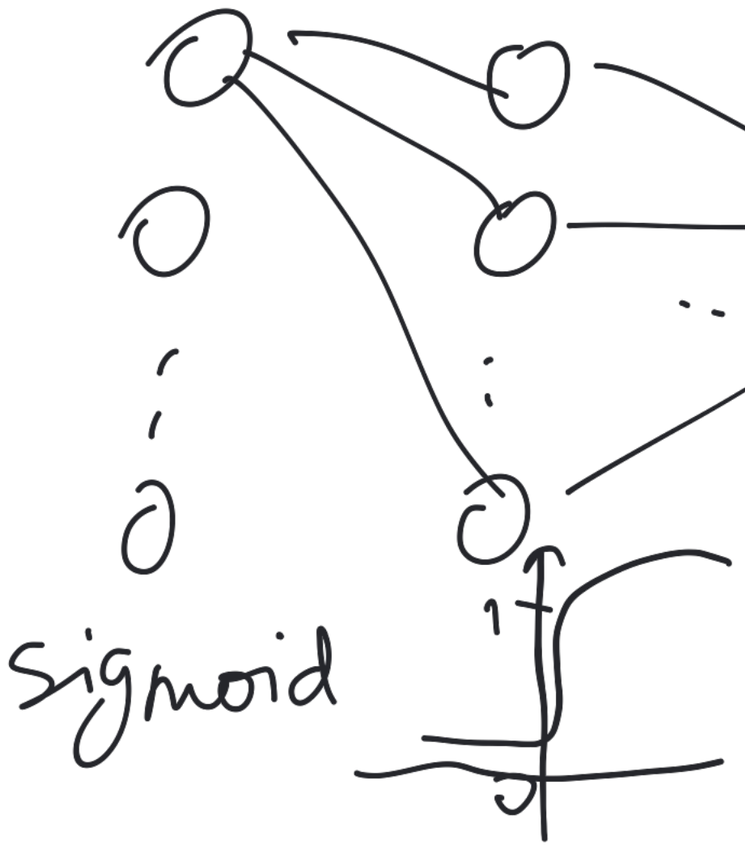
- Integer Encoding :

{Cat, dog, bird}

2 1 3

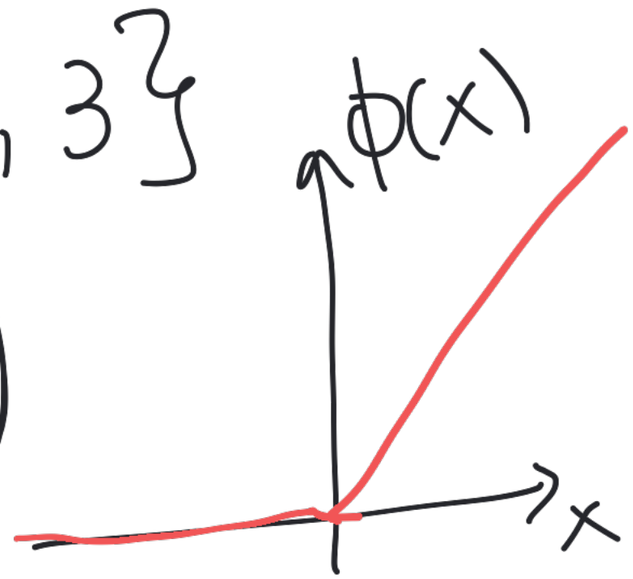
Relu

$$\phi(x) = \begin{cases} 0, & x \le 0 \\ x, & x > 0 \end{cases}$$



$y = \{1, 2, 3\}$

$y = \phi(v)$

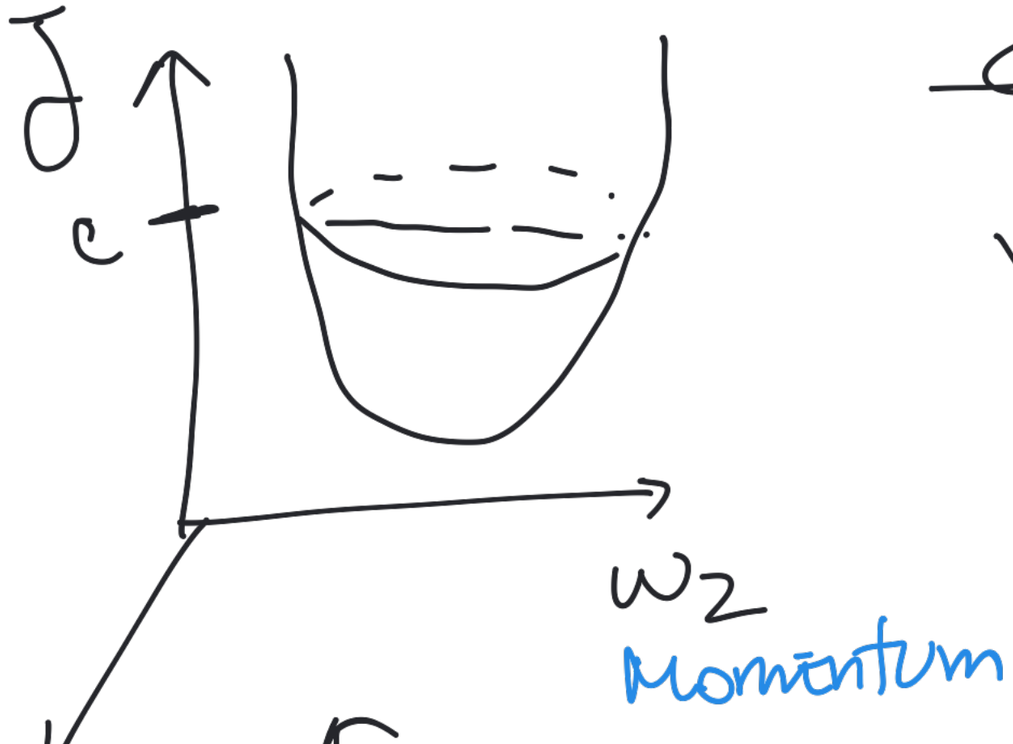


RELU : – handles vanishing gradient better than sigmoid or tanh.

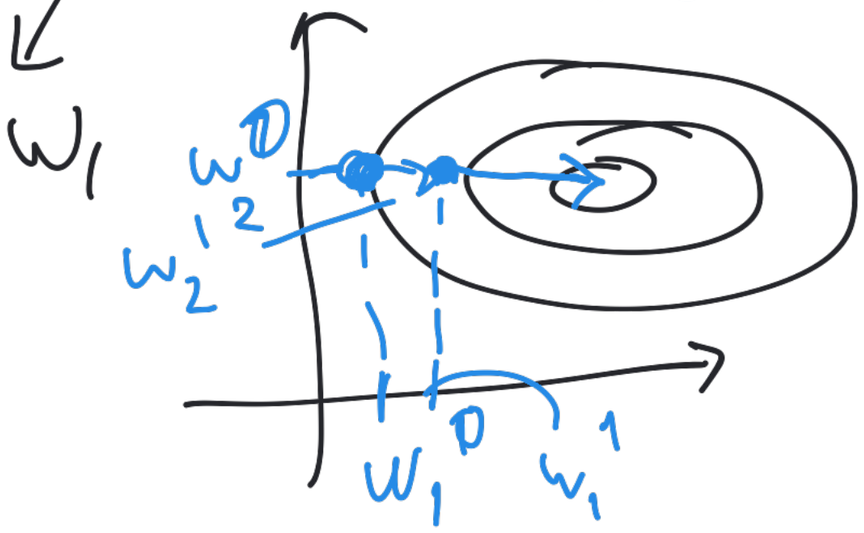
Common uses:
sigmoid or tanh at hidden layers followed by RELU at output layer

Accelerated Gradient

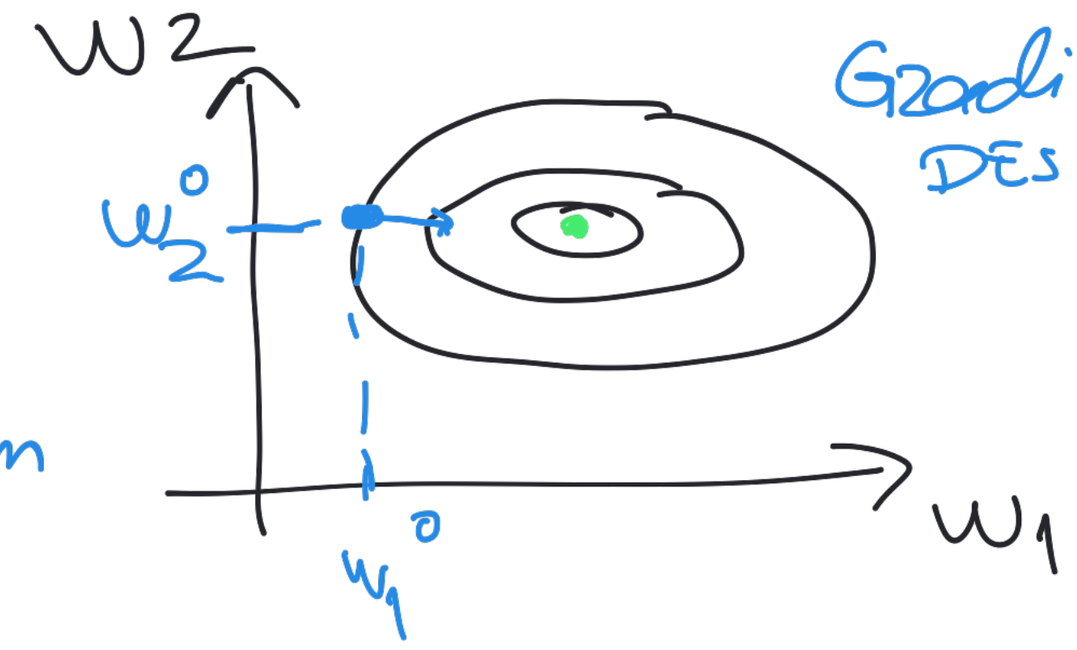
J = C : surface level



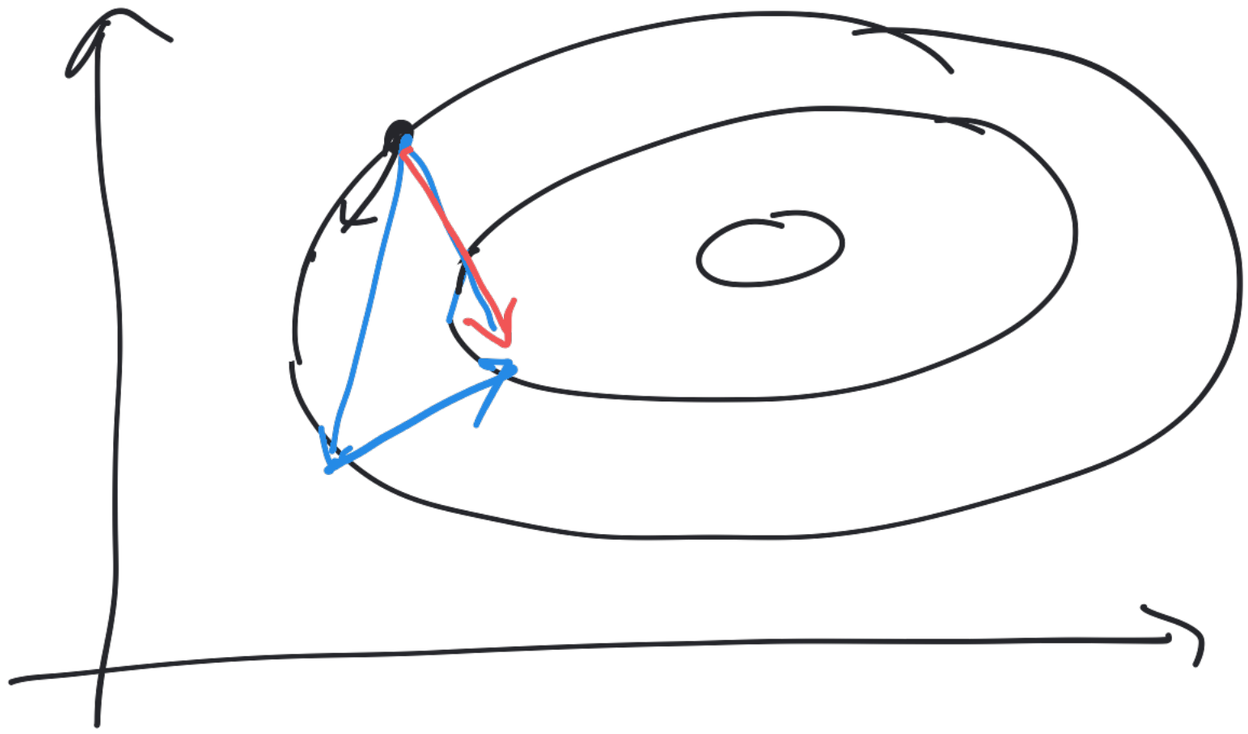
Momentum



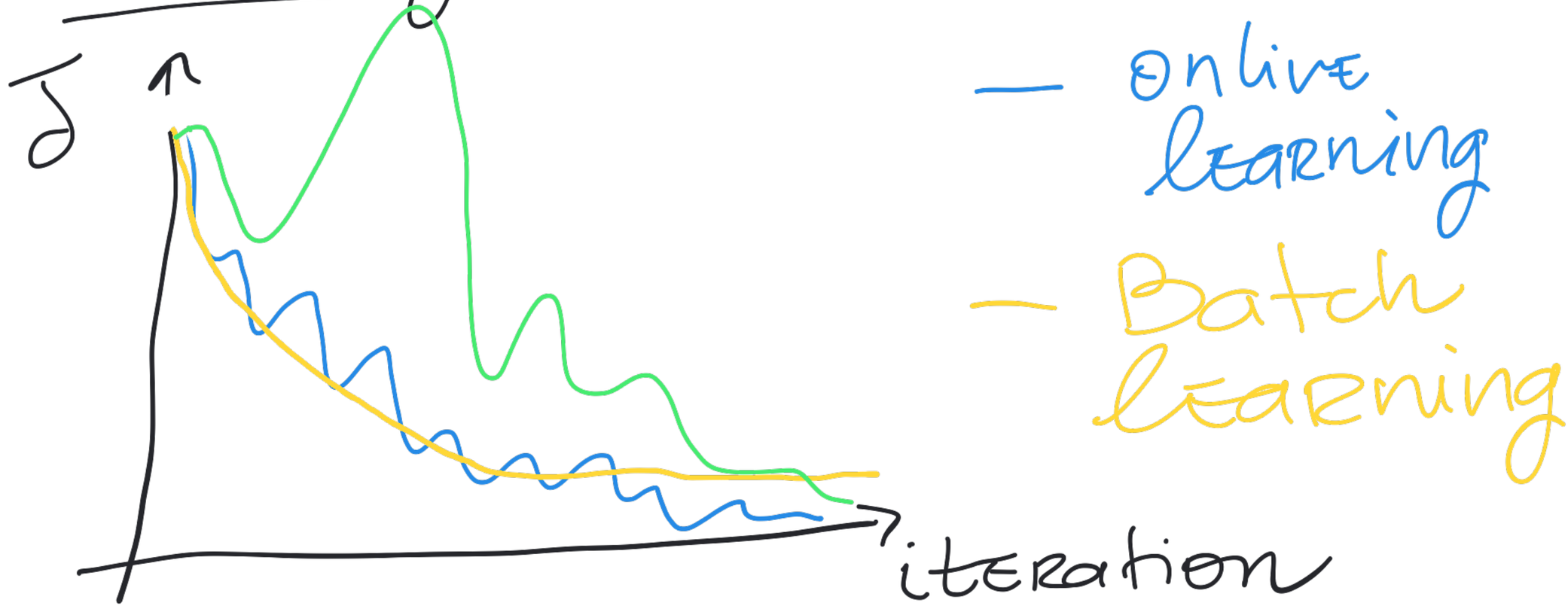
Gradients Descent



NAG (Nesterov's)



Learning Curves



— online learning

— Batch learning

Online/Mini-Batch Learning

- suffle data
- if using mini-batch,
  partition data into batches
- for N epochs:
  - compute local gradient
    (iteration)