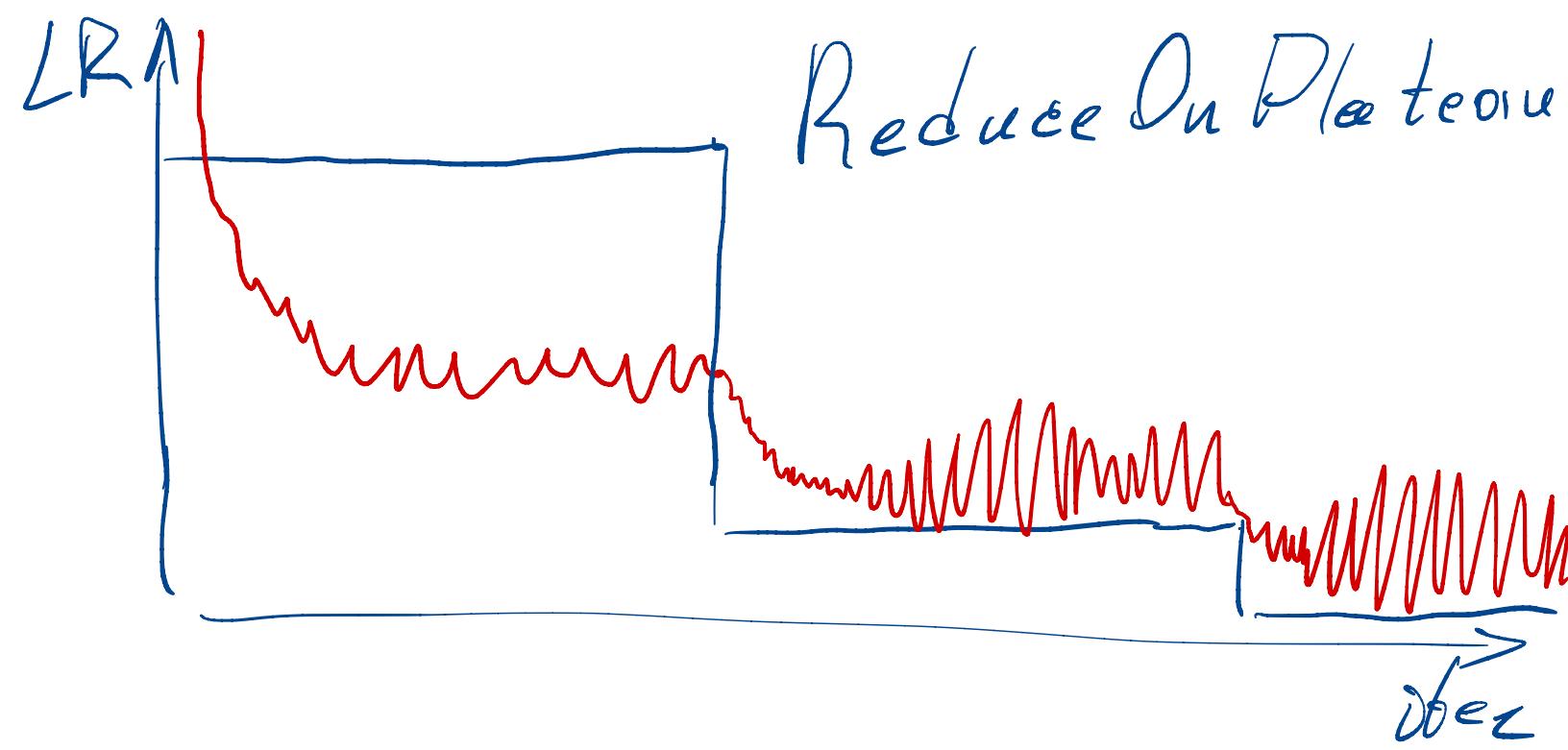
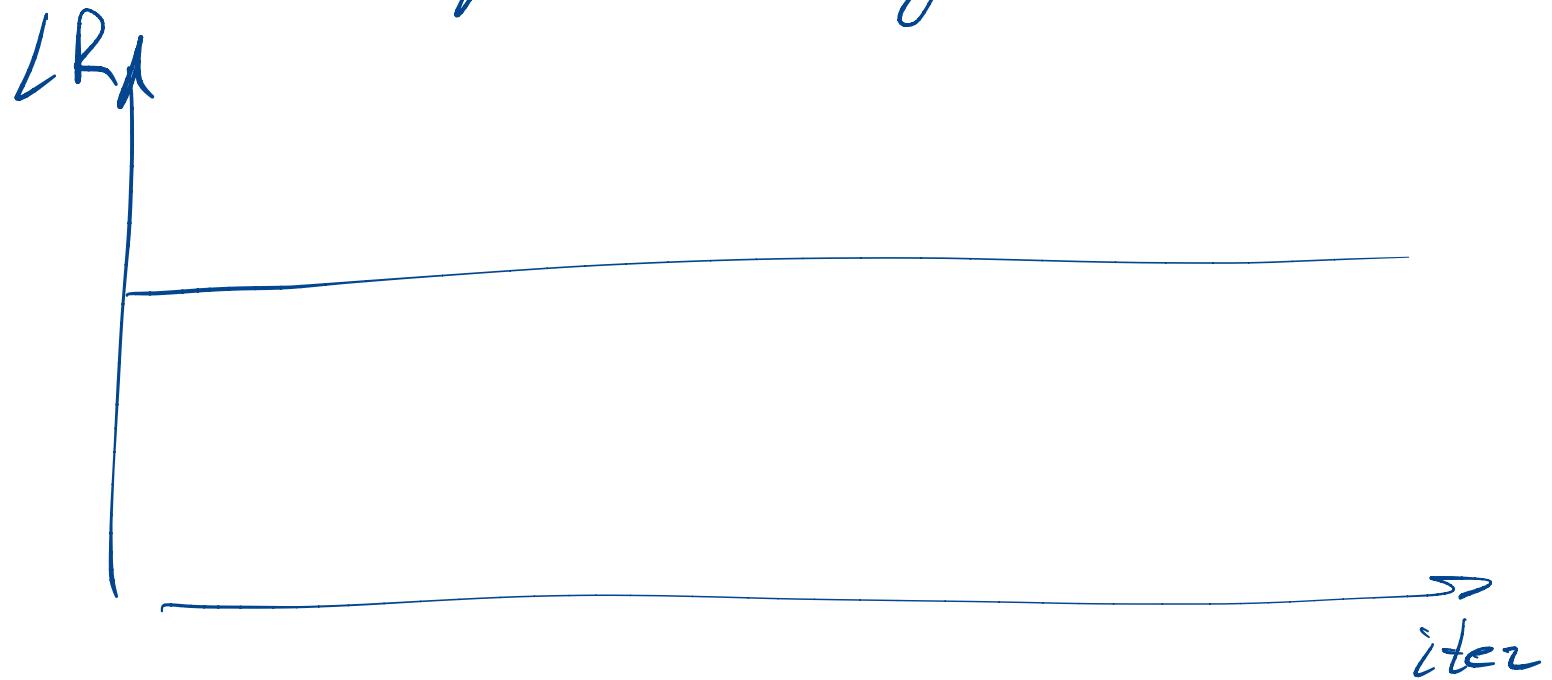
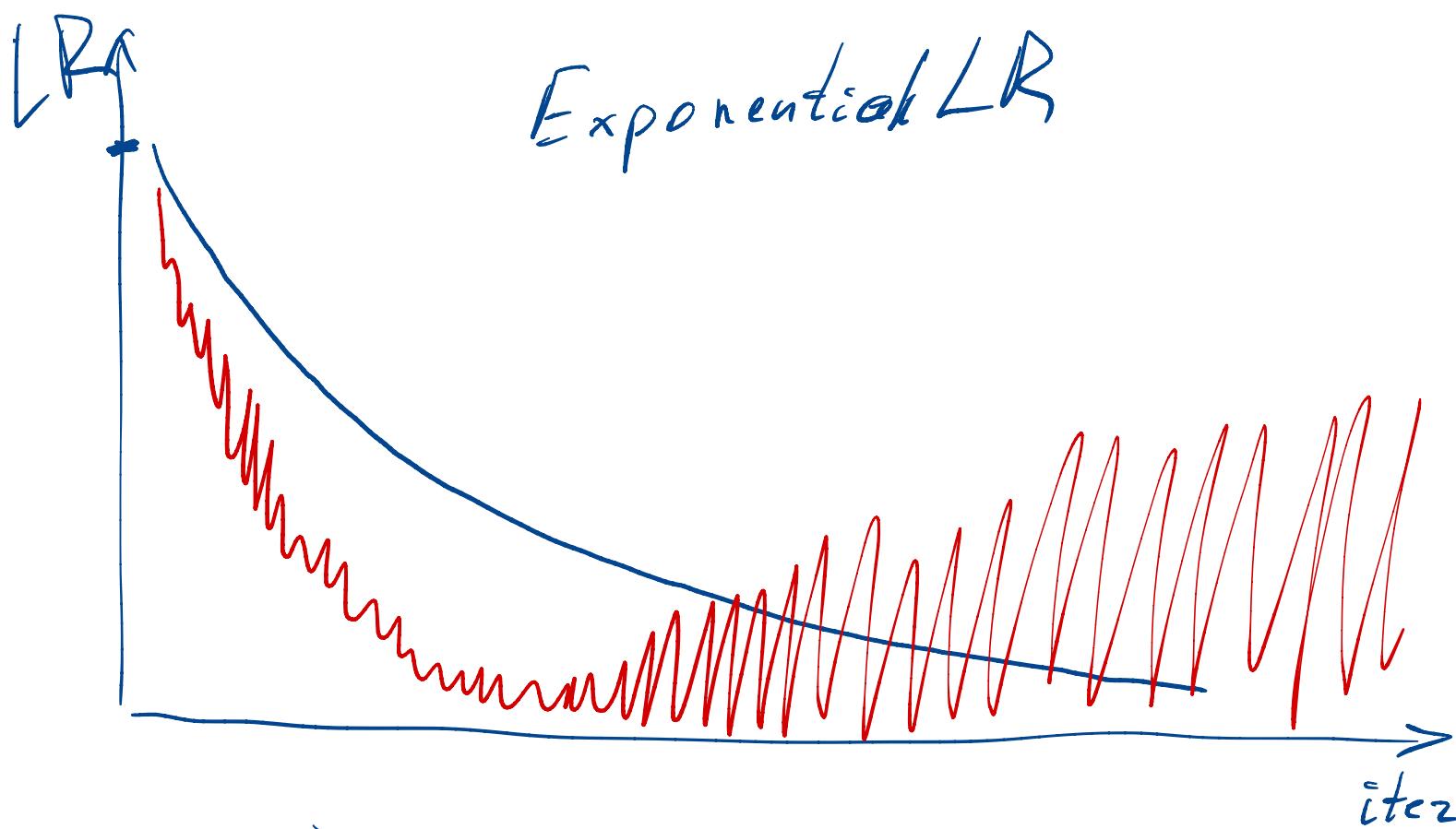


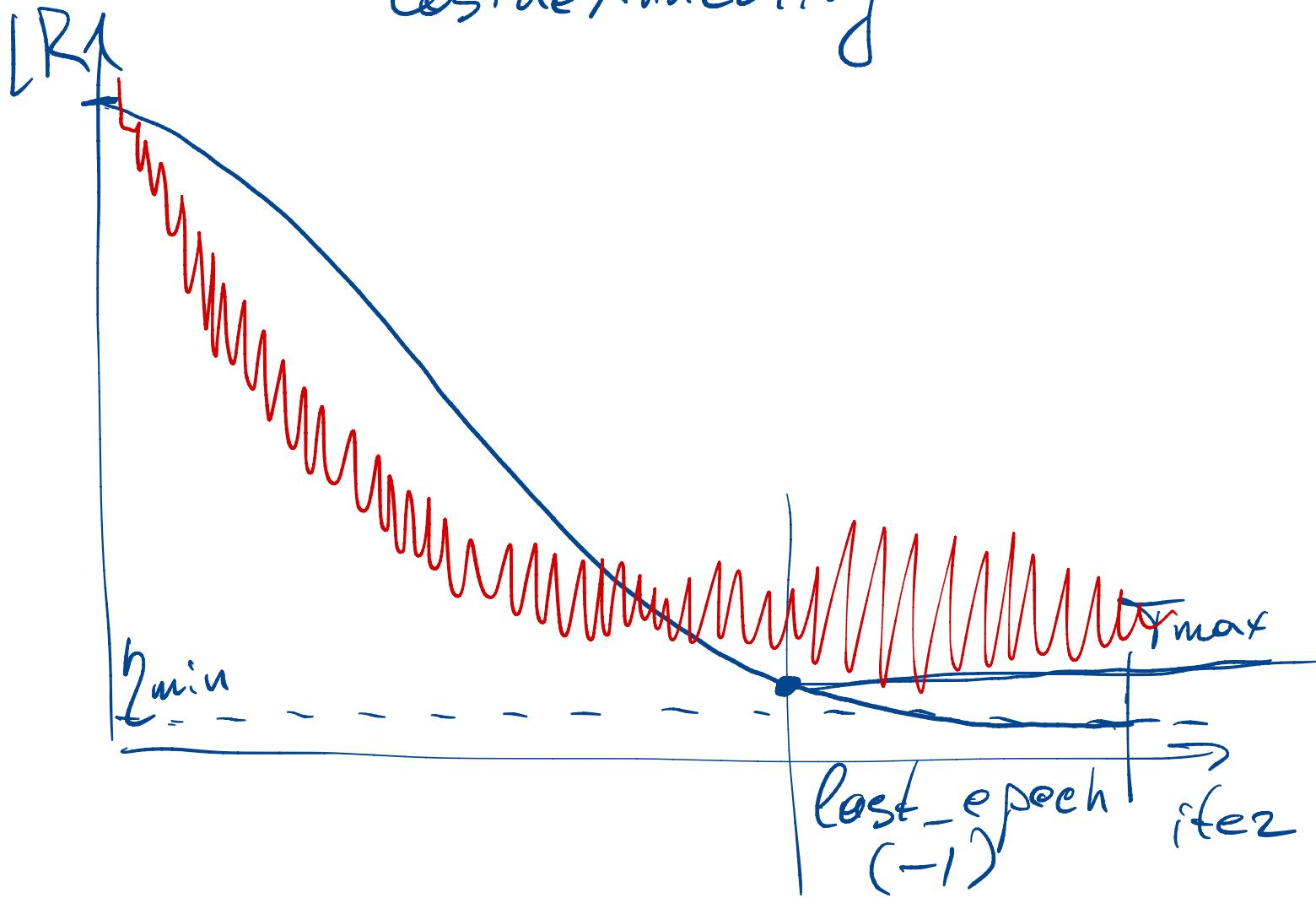
Равномерное LR
без процесса обучения NN





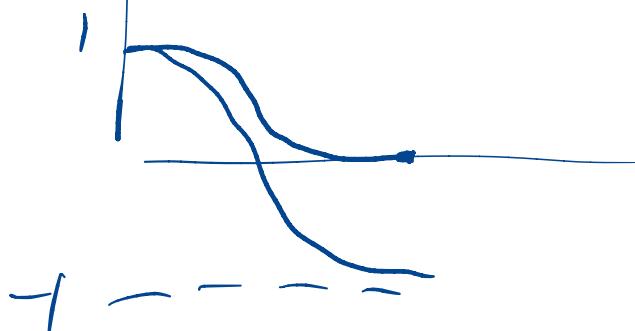
$$\ell_2^{(t+1)} = \gamma \ell_2^{(t)} \quad \gamma = 0,95$$

Cosine Annealing LR



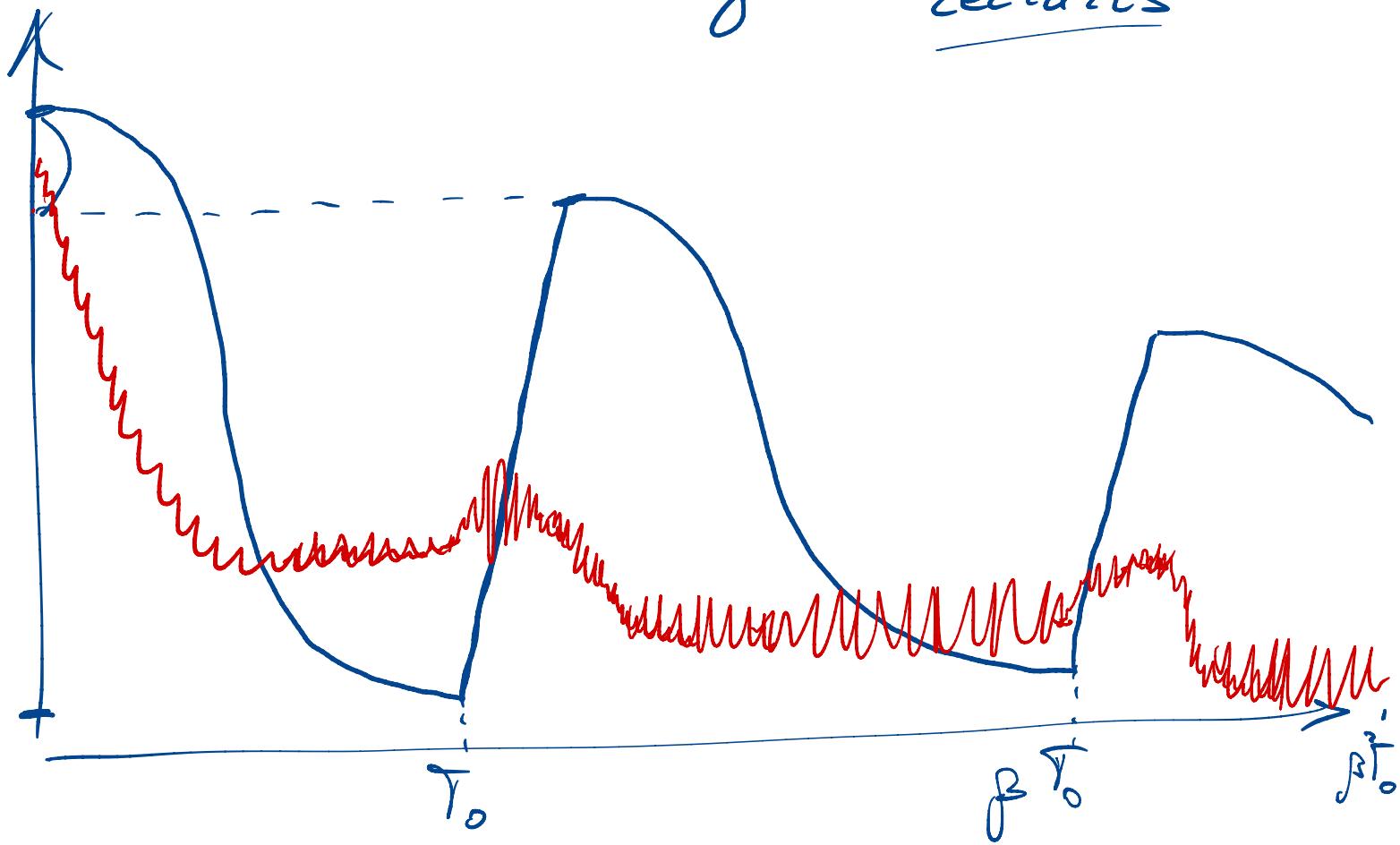
$$h^{(t)} = h_{\min} + \frac{1}{2} (h_0 - h_{\min}) \left(1 + \cos \frac{t \pi}{T_{\max}} \right)$$

$$\propto \frac{1}{2}(1 + \cos)$$



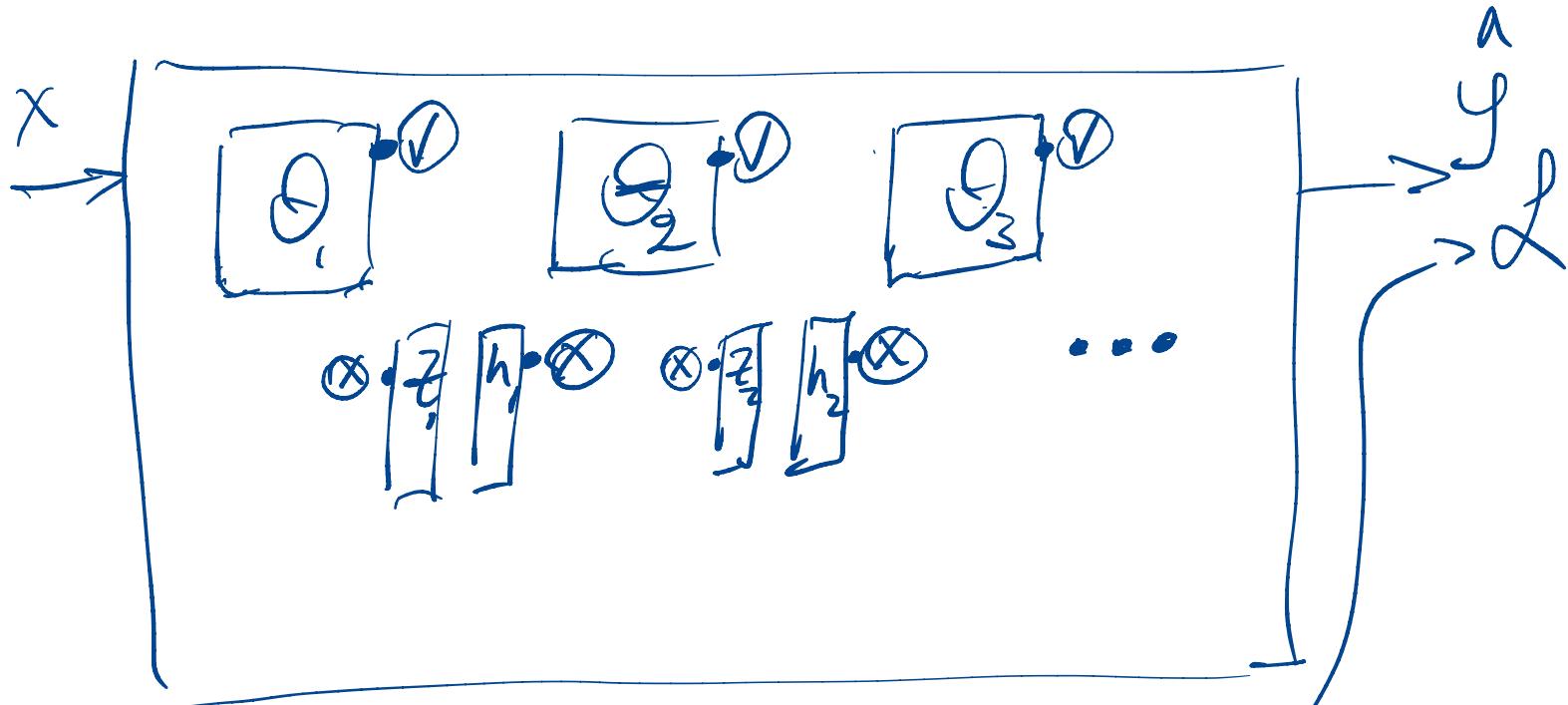
iter

Cosine Annealing with many vectors



$$\beta \sim 1.5 - 2$$

$$\zeta^{\text{new}}_0 = \zeta_0 \gamma \quad \gamma \sim 0.7 - 0.9$$



$$\bar{z} = \theta_x \quad h = \varphi(\bar{z})$$

$$\bar{z} = \theta_h$$

$$\hat{y} = \psi(\dots \varphi(\theta_3 \psi(\theta_2 \psi(\theta_1 g(\theta_0 x)))))$$

① x, y (batch)

$\theta_0 x = z$,
 $\varphi(z) = h$

② $\hat{y} = F_{NN}(x)$

output = model(Batch-data)

③ $L = L(\hat{y}, y)$

⑤ $\theta^{(t+1)} = \theta^{(t)} - \eta g$

④ $\vec{g} = \nabla_{\theta} L$

Argумент requires_grad

④ $g = \nabla_{\theta} L$ - ~~for~~ курса
PyTorch

$x \xrightarrow{\theta_1} z_1 \xrightarrow{h_1} z_2 \xrightarrow{\theta_2} h_2 \xrightarrow{\theta_3} \dots \xrightarrow{\hat{y}} L$

$L.$.backward()

PyTorch курса $\nabla_{\theta} L$, если
 $t.$.requires_grad = True

⑤ Указ requires_grad оптимайзеру:
optimizer $t^{(t+1)} = t^{(t)} - \eta \nabla_t L$