Deep Learning

Dr. Mehran Safayani

safayani@iut.ac.ir

safayani.iut.ac.ir

https://www.aparat.com/mehran.safayani

https://github.com/safayani/deep_learning_course

Department of Electrical and computer engineering, Isfahan university of technology, Isfahan, Iran

Hyperparameter tuning

Tuning process

Hyper parameters

① α

momentum $\beta = 0.9$

Adam $\underbrace{\beta_1}_{0.9}$, $\underbrace{\beta_2}_{0.999}$, $\underbrace{\varepsilon}_{10^{-8}}$

③ #layers

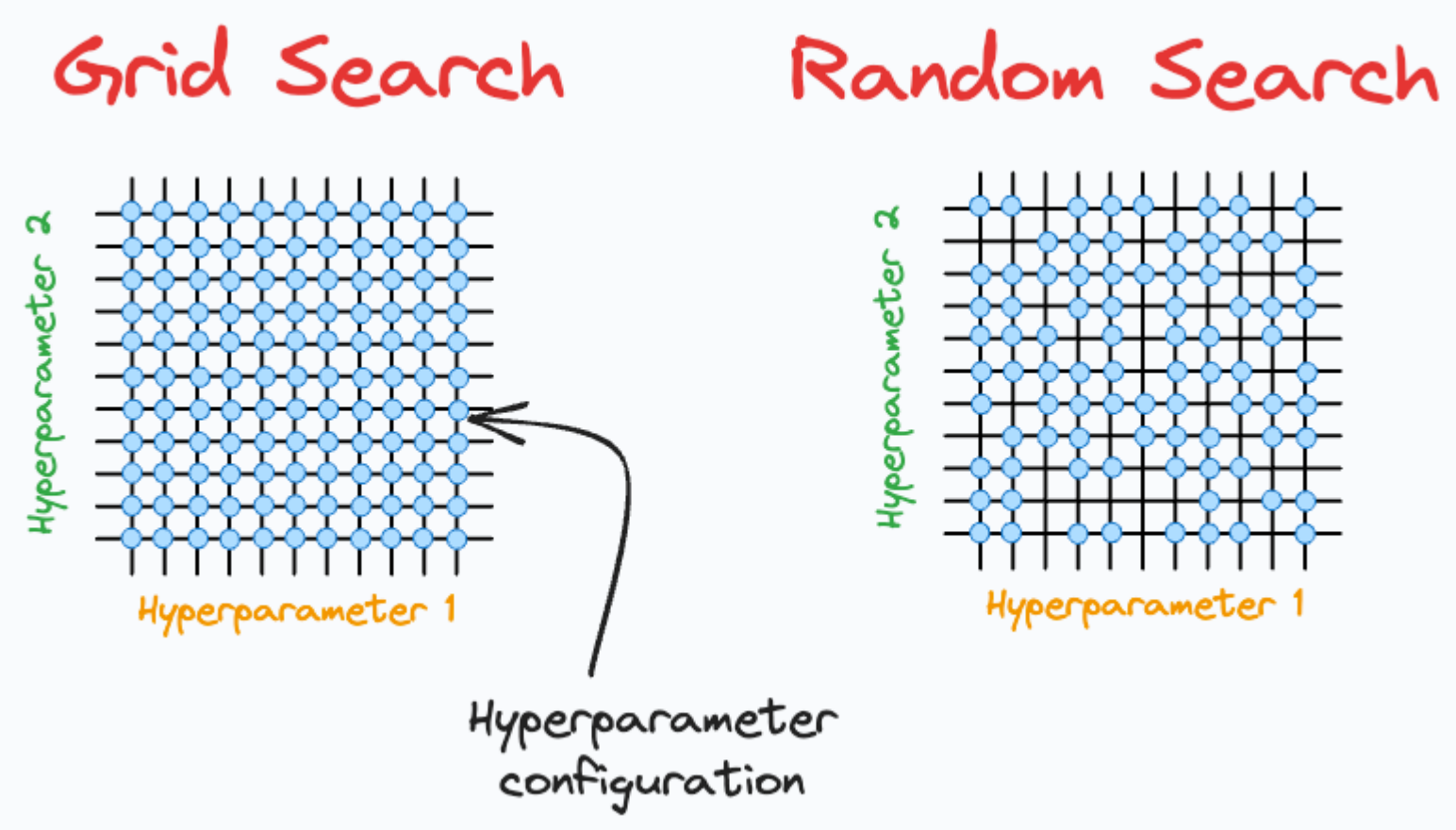
② #hidden units

③ Learning rate decay

② mini-batch size

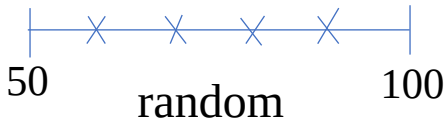
Hyperparameter tunning

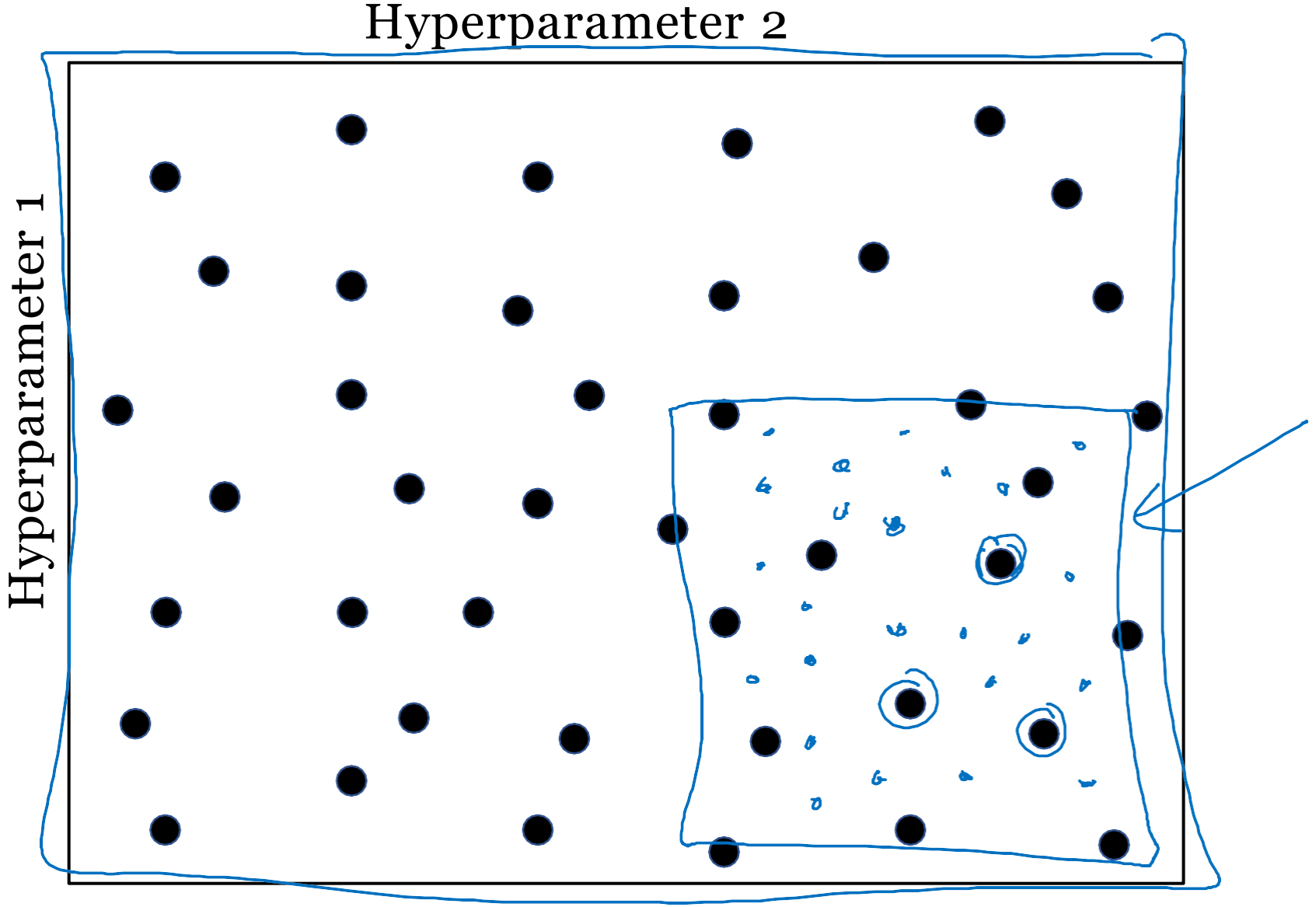
- Grid Search
- Random search



Coarse to fine

- $n^{[l]} = 50, \ldots, 100$





Andrew Ng

Using an appropriate scale to pick hyperparameters

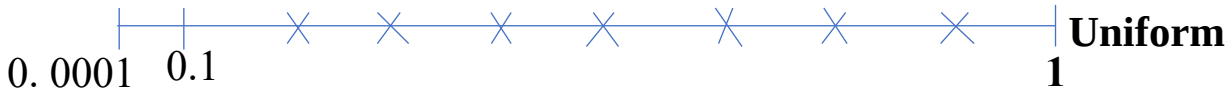
$n^{[l]} = 50, \ldots, 100$

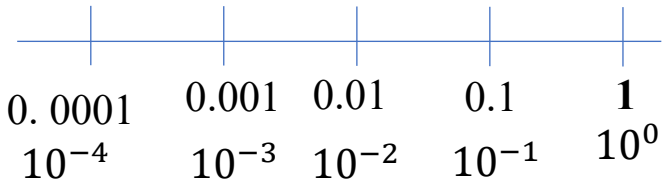
- #Layers

  L: 2 – 4

  2,3,4

- $\alpha = 0.0001, \ldots, 1$ $\alpha = 10^r \; r = -1 * np \cdot random \cdot randint(4)$

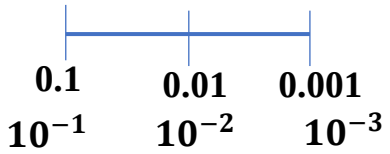




Andrew Ng

Using an appropriate scale to pick hyperparameters

- $r = -4 * np \cdot random \cdot rand()$ $\quad r \in [-4, 0]$
- $\alpha = 10^r$
- $\beta = 0.9 \cdots 0.999$
- $1 - \beta = 0.1 \cdots 0.001$

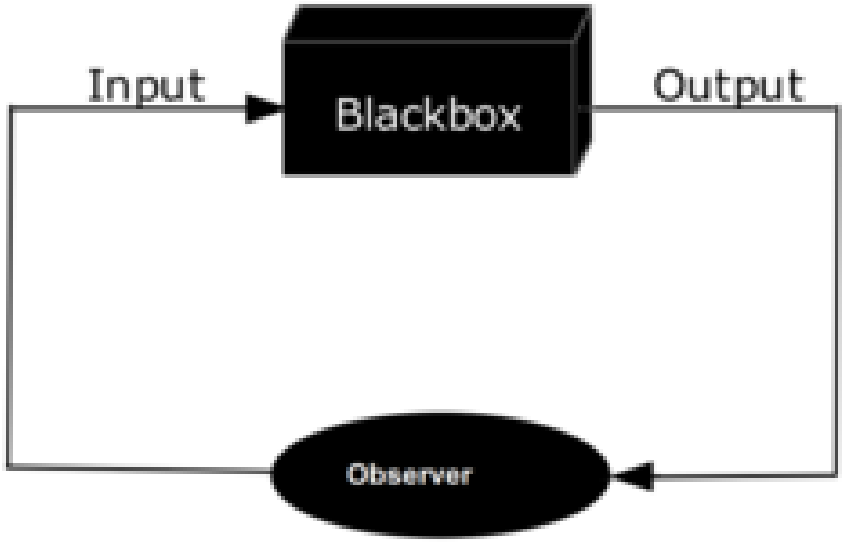


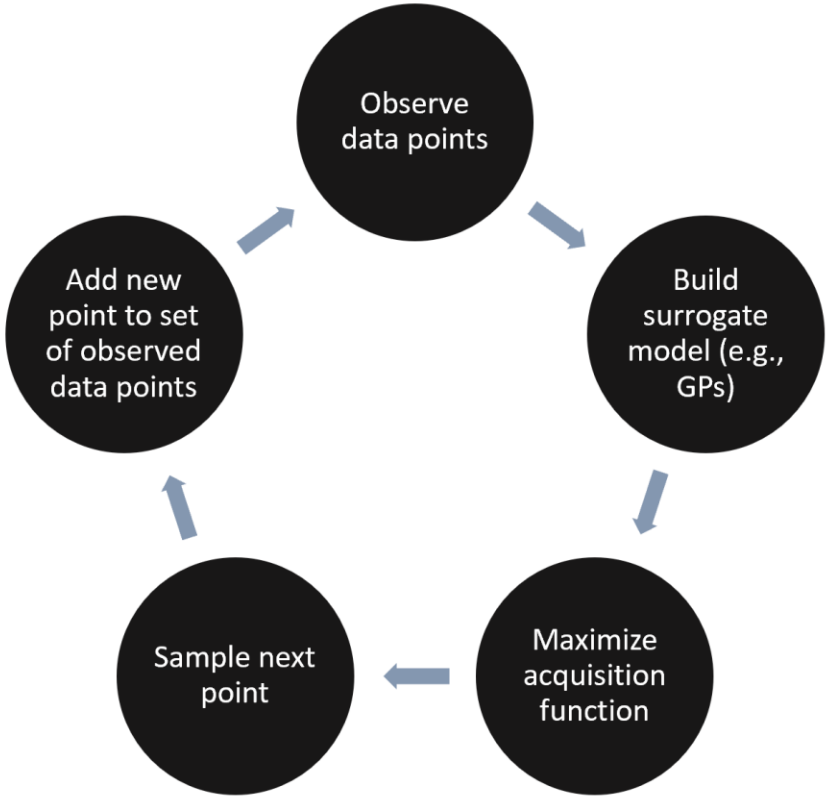
- $r \in [-3, -1]$

  $1 - \beta = 10^r$

  $\beta = 1 - 10^r$

Bayesian Optimization

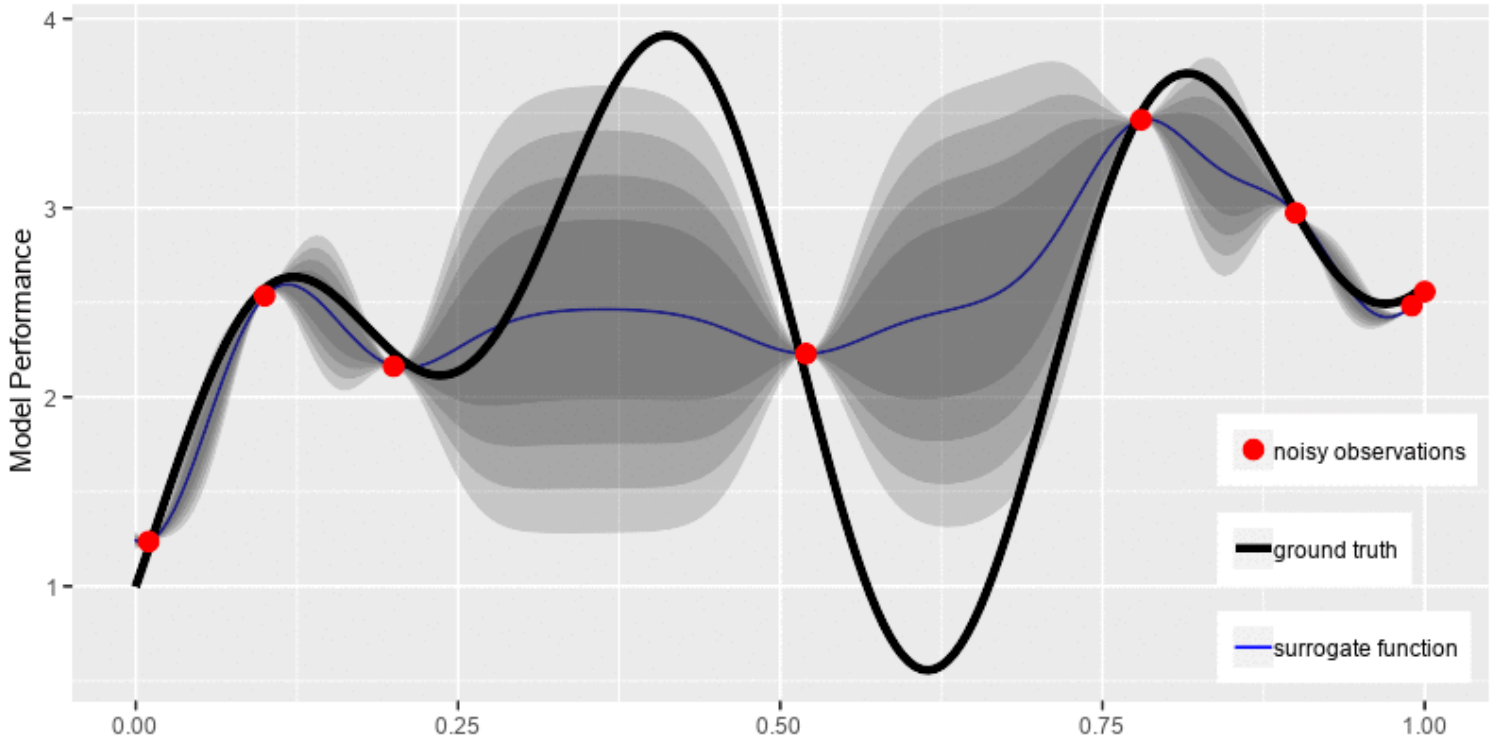


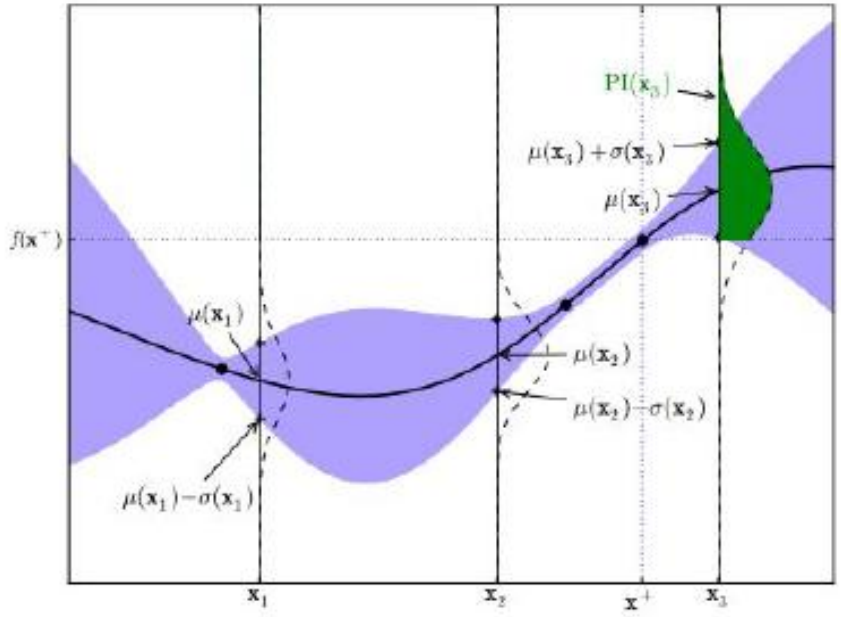


Bayesian optimization algorithm

https://ekamperi.github.io/machine%20learning/2021/06/11/acquisition-functions.html

Expected Improvement function





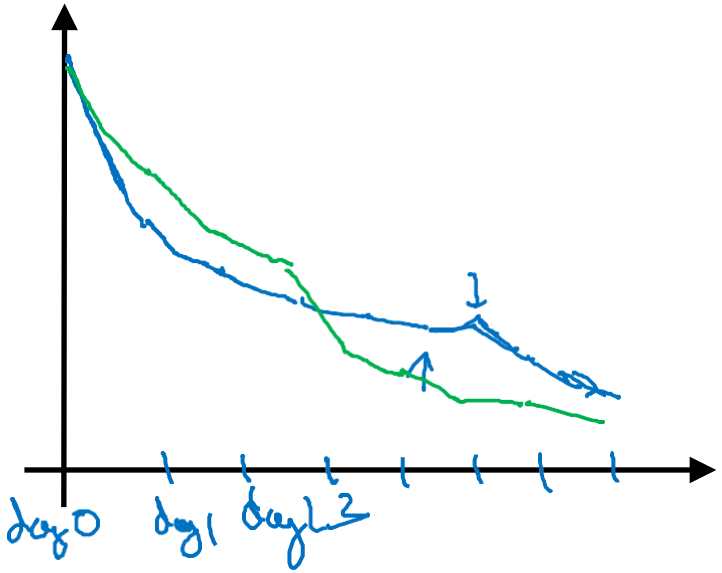
Suppose that we'd like to maximize $f(x)$, and the best solution we have so far is $x^$

$$\mathrm{EI}(x) = (\mu - f(x^\star))\,\Phi\left(\frac{\mu - f(x^\star)}{\sigma}\right) + \sigma\varphi\left(\frac{\mu - f(x^\star)}{\sigma}\right)$$

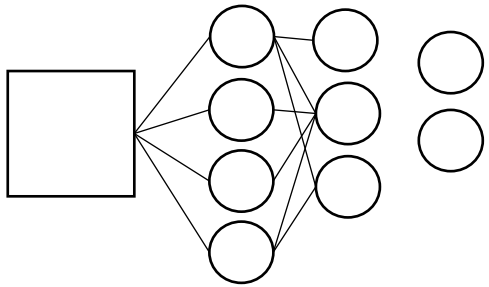
Where $\varphi(z)$ is the probability density function of the normal distribution $\mathcal{N}(0,1)$,

$\Phi(z) \equiv \mathrm{CDF}(z)$

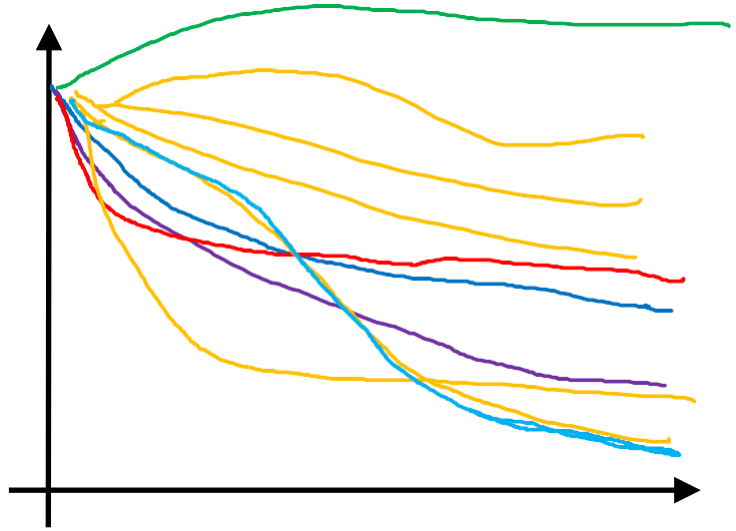
Babysitting one model



Panda



Training many models in parallel



Caviar