

# Lecture 24: Deep neural networks

Professor Ilias Bilonis

## Training regression networks - Loss function minimization

# The square error loss function for regression

$$x_{1:n} = (x_1, \dots, x_n) ; y_{1:n} = (y_1, \dots, y_n)$$

$$y = f(x; \theta) : \text{DNN model}$$

parameters

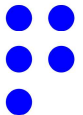
Loss function (Square Error)

$$L(\theta) = \frac{1}{n} \sum_{i=1}^n \left( \underbrace{y_i}_{\text{observation}} - \underbrace{f(x_i; \theta)}_{\text{prediction}} \right)^2$$

error

$$\theta^* = \operatorname{argmin} L(\theta)$$

Non-linear optimization method.  
analytical solution does not exist



# From Gaussian likelihood to loss function

Likelihood :  $p(y_i | x_i, \theta, \sigma) = N(y_i | f(x_i; \theta), \sigma^2)$

$$p(y_{1:n} | x_{1:n}, \theta, \sigma) = \prod_{i=1}^n p(y_i | x_i, \theta, \sigma) = N(y_{1:n} | f(x_{1:n}; \theta), \sigma^2 \mathbb{I}_n)$$

Max like :  $\max_{\theta} \log p(y_{1:n} | x_{1:n}, \theta, \sigma)$

$$= -\frac{1}{2\sigma^2} \sum_{i=1}^n (y_i - f(x_i; \theta))^2 + \dots$$

This process & way of thinking can be generalized for use in other settings

$$\min_{\theta} L(\theta)$$