Lecture 24: Deep neural networks

Professor Ilias Bilionis

Training regression networks - Loss function minimization

PREDICTIVE
SCIENCE LABORATORY

The square error loss function for regression

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

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

Loss function (Square Error)

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

$$\theta^* = \arg\min L(\theta)$$

Non-linear optimization method.

analytical solution does not exist

From Gaussian likelihood to loss function

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

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

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

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

$$\Downarrow$$

$$\min_\theta L(\theta)$$

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