Lecture 24: Deep neural networks

Professor Ilias Bilionis

The stochastic gradient decent and variants

The Robbins-Monro algorithm

$$\min_{\vartheta} \mathbb{E}_z[\ell(\vartheta; z)] \quad (*)$$

1. Initialize ϑ_0.
2. Iterate:

$$\vartheta_{t+1} = \vartheta_t - \alpha_t \nabla_{\vartheta} \ell(\vartheta; z_t)$$

(α_t: learning rate (scalar); z_t: a sample of z.)

where z_t is a sample of z.

RM theorem: the algorithm above will converge to a (local) minimum of the st. opt. $(*)$ if

$$\sum_{t=1}^{\infty} \alpha_t = +\infty \; ; \; \sum_{t=1}^{\infty} \alpha_t^2 < \infty$$

(α_t goes to zero but not too fast)

$$\alpha_t = \frac{A}{(Bt+C)^p}, \quad 0.5 < p < 1$$

Application of Robbins-Monro to loss minimization

$$L(\vartheta) = \frac{1}{n}\sum_{i=1}^{n}\left(y_i - f(x_i;\vartheta)\right)^2$$

m (batch size), $I_1, \dots, I_m \sim \text{Categorical}(1/n, \dots, 1/n)$ ind.

$$\ell_m(\vartheta; I_{1:m}) = \frac{1}{m}\sum_{j=1}^{m}\left(y_{I_j} - f(x_{I_j};\vartheta)\right)^2$$

$\min_\vartheta L(\vartheta) \Leftarrow$ $\min_\vartheta \mathbb{E}_{I_{1:m}}\left[\ell_m(\vartheta; I_{1:m})\right]$

R.M.

- Sample indices $i_1, \dots, i_m$ from $I_1, \dots, I_m$
- $\vartheta_{t+1} = \vartheta_t - \alpha_t \nabla_\vartheta \ell_m(\vartheta_t; i_{1:m})$

$$\to \vartheta_{t+1} = \vartheta_t - \alpha_t \nabla_\vartheta \sum_{j=1}^{m}\left\{y_{i_j} - f(x_{i_j};\vartheta_t)\right\}^2$$

$$= \vartheta_t - \alpha_t \sum_{j=1}^{m} 2\left\{y_{i_j} - f(x_{i_j};\vartheta_t)\right\} \cdot \nabla_\vartheta f(x_{i_j};\vartheta_t)$$

error

gradient of the model wrt to all parameters

Backprop = Automatic Differentiation in PyTorch

Advanced variants of stochastic gradient descent

- Stochastic gradient descent with momentum.
- AdaGrad.
- Adam (adaptive moment estimation).