Deep Learning (83882) - 2023-2024

Recitation 2: Linear & Logistic Regression

Ran Eisenberg

Bar-Ilan University, Israel

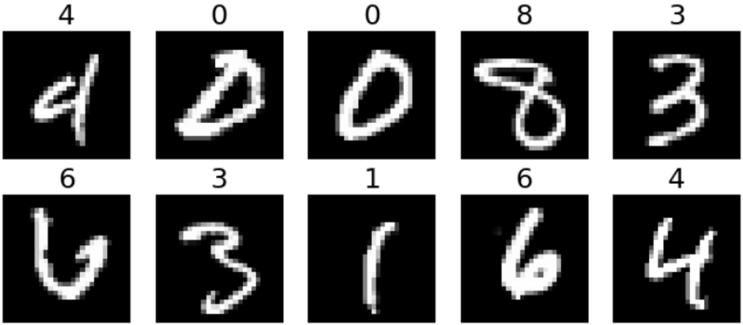
Supervised learning

In supervised learning setups:

- Given a set of input (X) and labels (y) pairs
- We search for a function (model) f that represents the relationship between X and y
- We define a loss function which tells us how well our model approximates the training examples
- Optimization process, a way of finding the parameters of our model that minimize the loss function

Supervised learning - An example

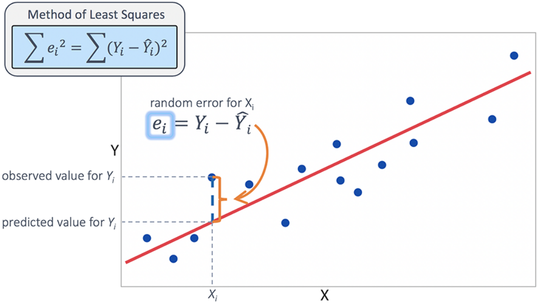
As an example consider the following problem (which we will return to later) of recognizing handwritten digits:



Linear Regression

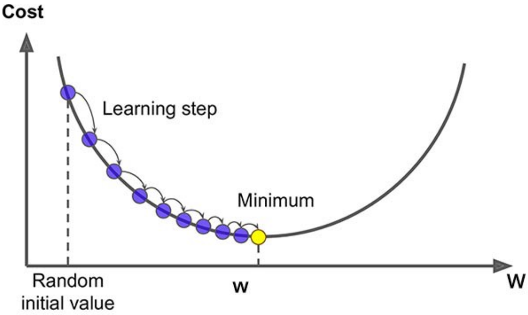
- Given $(\mathrm{x}_1, y_1), \ldots, (\mathrm{x}_\mathrm{n}, y_n)$ where, $\mathrm{x_j} \in \mathbb{R}^d$ and $y_j \in \mathbb{R}$
- We would like to find $\hat{y} = f(\mathrm{x}; w) = \mathrm{w}^T\mathrm{x}$
- We minimize the following loss term:

$$\mathcal{L}(\mathrm{y}, \hat{\mathrm{y}}) = \frac{1}{n}\sum_{j=1}^{n}(\mathrm{y_j} - \hat{\mathrm{y}}_\mathrm{j})^2$$



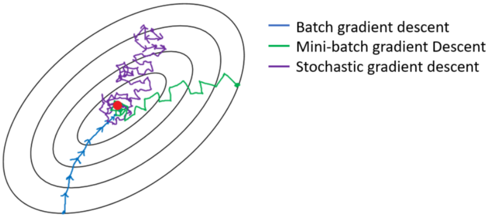
Gradient Descent

- Usually DL models are optimized in an iterative manner
- The most basic algorithm is Gradient Descent (GD):
$$\mathrm{w}^{t+1} = \mathrm{w}^{t} - \eta \frac{\partial \mathcal{L}}{\partial \mathrm{w}}(\mathrm{w}^{t})$$
- In each update we average the gradients for all training examples



Mini-Batch Gradient Descent

- The convergence of Gradient Descent is slow (each update requires processing all the dataset)
- Stochastic Gradient Descent (SGD) - an approximation of the gradient with only one example. Usually very noisy.
- Mini-Batch Gradient Descent - At every iteration average the gradients of k examples sampled i.i.d



Learning the Parameters of Linear Regression

Back to the Linear Regression. Let's find the gradients:

$$
\begin{aligned}
\frac{\partial \mathcal{L}}{\partial \mathrm{w}} &= \frac{\partial(\frac{1}{n}\sum_{j=1}^{n}(\mathrm{y_j} - \hat{\mathrm{y}}_\mathrm{j})^2)}{\partial \mathrm{w}} \\
&= \frac{1}{n}\sum_{j=1}^{n}\frac{\partial(\mathrm{y}_j - \hat{\mathrm{y}}_\mathrm{j})^2}{\partial \mathrm{w}} \\
&= \frac{1}{n}\sum_{j=1}^{n}\frac{\partial(\mathrm{y}_j - \mathrm{w}^T\mathrm{x}_j)^2}{\partial \mathrm{w}} \\
&= \frac{1}{n}\sum_{j=1}^{n} 2\cdot(\mathrm{y}_j - \mathrm{w}^T\mathrm{x}_j)\cdot -\mathrm{x}_j
\end{aligned}
$$

In the code!