Introduction to Deep Learning

Machine Learning

Linear Regression

$$\hat{y}_i = \sum_{j=1}^{d} x_{ij}\theta_j$$

$d =$ input dimension
$x =$ input data
$\theta =$ weights
Loss function:

$$J(\theta) = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2$$

in matrix notation:

$$J(\theta) = (\mathbf{X}\theta - \mathbf{y})^T(\mathbf{X}\theta - \mathbf{y})$$

differentiate:

$$\frac{\partial J(\theta)}{\partial \theta} = 2\mathbf{X}^T\mathbf{X}\theta - 2\mathbf{X}^T\mathbf{y} = 0$$

$$\theta = (\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T\mathbf{y}$$

Analytical solution for a convex problem.

Logistic Regression

$$\hat{\mathbf{y}} = p(\mathbf{y} = 1|\mathbf{X}, \theta) = \prod_{i=1}^{n} p(y_i = 1|\mathbf{x}_i, \theta)$$

where

$$\hat{y}_i = \sigma(\mathbf{x}_i\theta)$$

σ is the sigmoid/logistic function.
Loss function (binary cross entropy loss):

$$L(\hat{y}_i, y_i) = -[y_i \, log \, \hat{y}_i + (1 - y_i) \, log(1 - \hat{y}_i]$$

No analytical solution. Iterative method needed (gradient descent).

Loss Functions

Measure the goodness of the prediction.
Large loss indicates bad performance.
Choice of loss function depends on problem.

Regression Loss

L1 Loss:

$$L(y, \hat{y}; \theta) = \frac{1}{n} \sum_{i}^{n} ||y_i - \hat{y}_i||_1$$

MSE Loss:

$$L(y, \hat{y}; \theta) = \frac{1}{n} \sum_{i}^{n} ||y_i - \hat{y}_i||_2^2$$

where
L1 norm (Manhattan distance):

$$||x||_1 = \sum |x_i|$$

- Promotes sparsity
- More robust to outliers

L2 norm (Euclidean distance):

$$||x||_2 = \sqrt{\sum x_i^2}$$

- Differentiable
- Penalizes large errors more heavily

Squared L2 norm:

$$||x||_2^2 = \sum x_i^2$$

- Differentiable

- Computationally efficient

Classification Loss

Binary Cross Entropy loss (binary classification):

$$L(y, \hat{y}; \theta) = -\frac{1}{n}\sum_{i}^{n}[y_i log \hat{y}_i + (1 - y_i) log(1 - \hat{y}_i)]$$

y is a probability output ($[0, 1]$) of the first class. Therefore, $1 - y$ is the probability of the second class. Works well with a sigmoind activation in the last layer.
Log penalizes bad predictions: $log\, 0 = -\infty$, $log\, 1 = 0$ (sort of inverts the prediction $\hat{y}$).
Encourages the model to make confident predictions close to 0 or 1.
Cross Entropy loss (multi-class classification):

$$L(y, \hat{y}; \theta) = -\sum_{i=1}^{n}\sum_{k=1}^{k}(y_{ik} \cdot log\, \hat{y}_{ik})$$

The second sum is a generalization of the binary cross entropy case for multiple classes.