Logistic Regression

Part II

Logistic Regression and Square Error

Step One: $$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$$

Step Two: Training data: $(x^n, \hat{y}^n)$, $\hat{y}^n$: 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2}\sum_n \left(f_{w,b}(x^n) - \hat{y}^n\right)^2$$

Step Three:

$$\frac{\partial\,(f_{w,b}(x) - \hat{y})^2}{\partial w_i} = 2\left(f_{w,b}(x) - \hat{y}\right) \frac{\partial f_{w,b}(x)}{\partial z} \frac{\partial z}{\partial w_i}$$

$$= 2\left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

$\hat{y}^n = 1$

If $f_{w,b}(x^n) = 1$ (close to target) $\Rightarrow$ $\partial L / \partial w_i = 0$

If $f_{w,b}(x^n) = 0$ (far from target) $\Rightarrow$ $\partial L / \partial w_i = 0$

Logistic Regression and Square Error, Continued

Step One: $f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$

Step Two: Training data: $(x^n, \hat{y}^n)$, $\hat{y}^n$: 1 for class 1, 0 for class 2

$$L(f) = \frac{1}{2}\sum_n \left(f_{w,b}(x^n) - \hat{y}^n\right)^2$$

Step Three:

$$\frac{\partial\,(f_{w,b}(x) - \hat{y})^2}{\partial w_i} = 2\left(f_{w,b}(x) - \hat{y}\right)\frac{\partial f_{w,b}(x)}{\partial z}\frac{\partial z}{\partial w_i}$$

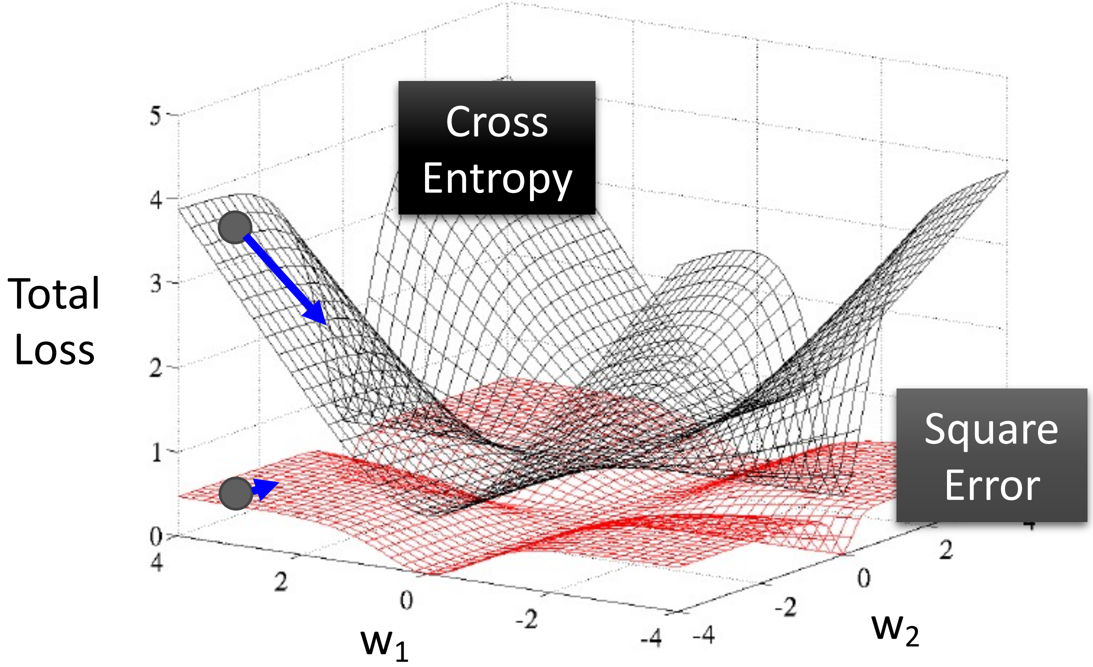
$$= 2\left(f_{w,b}(x) - \hat{y}\right) f_{w,b}(x)\left(1 - f_{w,b}(x)\right) x_i$$

$\hat{y}^n = 0$

If $f_{w,b}(x^n) = 1$ (far from target) $\Rightarrow$ $\partial L / \partial w_i = 0$

If $f_{w,b}(x^n) = 0$ (close to target) $\Rightarrow$ $\partial L / \partial w_i = 0$

Cross Entropy Versus Square Error



(Glorot & Bengio, 2010)

Logistic Regression

Step One:

$$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$$

Output: between 0 and 1

Step Two:

Step Three:

Linear Regression

$$f_{w,b}(x) = \sum_i w_i x_i + b$$

Output: any value

Logistic Regression	Linear Regression
Step One: $f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Training data: $(x^n, \hat{y}^n)$ Step Two: $\hat{y}^n$: 1 for class 1, 0 for class 2 $L(f) = \sum_n l(f(x^n), \hat{y}^n)$	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: a real number $L(f) = \frac{1}{2}\sum_n (f(x^n) - \hat{y}^n)^2$

Cross-entropy:

$$l(f(x^n), \hat{y}^n) = -\left[\hat{y}^n ln f(x^n) + (1 - \hat{y}^n) ln\left(1 - f(x^n)\right)\right]$$

	Logistic Regression	Linear Regression
Step One:	$f_{w,b}(x) = \sigma\left(\sum_i w_i x_i + b\right)$ Output: between 0 and 1	$f_{w,b}(x) = \sum_i w_i x_i + b$ Output: any value
Step Two:	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: 1 for class 1, 0 for class 2 $L(f) = \sum_n l(f(x^n), \hat{y}^n)$	Training data: $(x^n, \hat{y}^n)$ $\hat{y}^n$: a real number $L(f) = \frac{1}{2}\sum_n (f(x^n) - \hat{y}^n)^2$

Step Three:

Logistic regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Linear regression: $w_i \leftarrow w_i - \eta \sum_n -\left(\underline{\hat{y}^n - f_{w,b}(x^n)}\right) x_i^n$

Optimization Algorithm

Optimization Algorithms

- Gradient descent
- Conjugate gradient
- BFGS
- L-BFGS

Advantages and Disadvantages

- Advantages:
  - No need to manually pick learning rate
  - Often faster than gradient descent.
- Disadvantages:
  - More complex

Reference

- Glorot, X., & Bengio, Y. (2010). Understanding the difficulty of training deep feedforward neural networks. *Journal of Machine Learning Research*, 9, pp. 249-256.