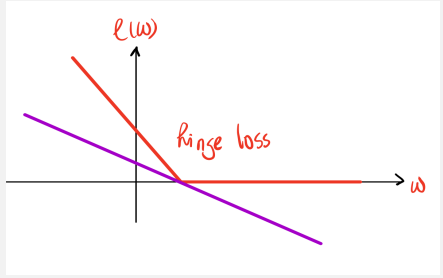
CS/ECE/ME 532
Matrix Methods in Machine Learning

Welcome!

Activity 20

Sub-gradients



Hinge loss:
Convex
Non differentiable

***Sub-gradient**: any plane that lies below function.*

any $\boldsymbol{v}$ such that $\ell(\boldsymbol{w}) \geq \ell(\boldsymbol{w}_0) + (\boldsymbol{w} - \boldsymbol{w}_0)^T \boldsymbol{v}$

Classifying new data:

$\boldsymbol{x} =$ 

$\widehat{y} = \text{sign}(\boldsymbol{x}^T \boldsymbol{w})$

if $\widehat{y} = 1$ then dog

if $\widehat{y} = -1$ then cat

$(\boldsymbol{x}_i, y_i), i = 1, ..., \text{a million}$

Training a classifier:

$$\min_{\boldsymbol{w}} \sum_{i=1}^{\text{a million}} (\boldsymbol{x}_i^T \boldsymbol{w} - y_i)^2$$

Problem: computing the loss is too slow.

Stochastic Gradient Descent

$$\min_{\boldsymbol{w}} \sum_{i=1}^{\text{a million}} \ell_i(\boldsymbol{w})$$

$$\boldsymbol{w}^{(k+1)} = \boldsymbol{w}^{(k)} - \tau \nabla \ell(\boldsymbol{w}^k)$$

(Gradient Descent)

Main idea

Do gradient descent, but on a random subset of training examples at each iteration.

$$\boldsymbol{w}^{(1)} = \boldsymbol{w}^{(0)} - \tau \sum_{i=1}^{100} \nabla \ell_i(\boldsymbol{w}^{(0)})$$

"mini batch"

$$\boldsymbol{w}^{(2)} = \boldsymbol{w}^{(1)} - \tau \sum_{i=101}^{200} \nabla \ell_i(\boldsymbol{w}^{(1)})$$

Good for ML problems that involve lots of training data:

- Image/video classification and recognition
- ML translation
- Large scale prediction and regression tasks