Datasets and Losses

Model Fitting

Goal: find parameters θ

$$f_\theta(\mathbf{x}) = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Failed to load file "/home/philkr/workspace/classes/deeplearning_v2/content/first_example/tasks/figures.ipynb.exec

Components:

- dataset
- loss function

Dataset

Recap:

- samples from a data generating distribution
- we use **labels** to train the model

$$\mathcal{D} = \{(\mathbf{x}_i, \mathbf{y}_i)\}_{i=1}^{N} \text{ where } (\mathbf{x}_i, \mathbf{y}_i) \sim P(X, Y)$$

Loss Function

A **loss function** measures the quality of the model

Failed to load file "/home/philkr/workspace/classes/deeplearning_v2/content/making-it-work/overfitting/figures.ipynb.exec"

Loss function:

$$l(\theta|\mathbf{x}_i, \mathbf{y}_i)$$

Expected loss:

$$L(\theta|\mathcal{D}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[l(\theta|\mathbf{x}, \mathbf{y})]$$

Properties of a Loss Function

- Low loss - good ☺
- High loss - bad ☹
- Loss function over the full dataset $\mathcal{D}$

$$L(\theta) = L(\theta|\mathcal{D}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[l(\theta|\mathbf{x},\mathbf{y})]$$

Loss Function: Examples

Linear regression:

$$\hat{\mathbf{y}} = \mathbf{W}\mathbf{x} + \mathbf{b}$$

Binary classification

$$\hat{\mathbf{y}} = \sigma(\mathbf{W}\mathbf{x} + \mathbf{b})$$

Multi-class classification:

$$\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{W}\mathbf{x} + \mathbf{b})$$

L2 loss

$$l(\theta|\mathbf{x}, \mathbf{y}) = \frac{1}{2}||\hat{\mathbf{y}} - \mathbf{y}||_2^2$$

Binary cross entropy loss

$$l(\theta|\mathbf{x}, \mathbf{y}) = -\mathbf{y}\log(\hat{\mathbf{y}}) - (1-\mathbf{y})\log(1-\hat{\mathbf{y}})$$

Cross entropy loss

$$l(\theta|\mathbf{x}, \mathbf{y}) = -\sum_{c=1}^{C} 1_{[\mathbf{y}=c]}\log(\hat{\mathbf{y}}_c)$$

Expected Loss

Converts a sample-based loss l into dataset-based loss L

$$L(\theta|\mathcal{D}) = \mathbb{E}_{(\mathbf{x},\mathbf{y})\sim\mathcal{D}}[l(\theta|\mathbf{x},\mathbf{y})]$$

- $l(\theta|\mathbf{x},\mathbf{y})$ function of θ, $\mathbf{x}$, $\mathbf{y}$
- $L(\theta|\mathcal{D})$ function of θ

Next - Model Fitting

Goal: find parameters θ

$$\theta^* = \arg\min_{\theta} L(\theta|\mathcal{D})$$

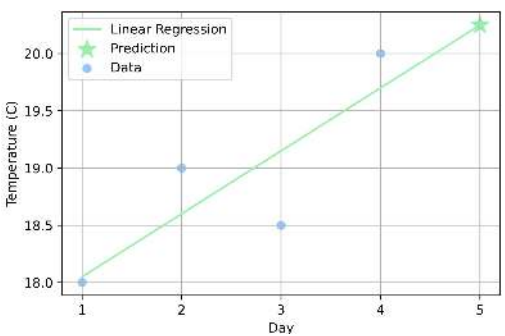
Deep learning uses **gradient descent**:

- requires gradient $\nabla_\theta L(\theta|\mathcal{D})$
- requires optimizers to update θ

Datasets and Losses TL;DR

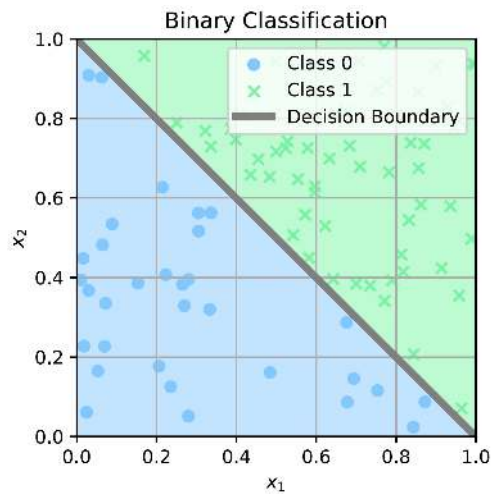
Regression

L2 loss



Binary classification

Sigmoid + binary cross entropy



Multi-class classification

Softmax + cross entropy