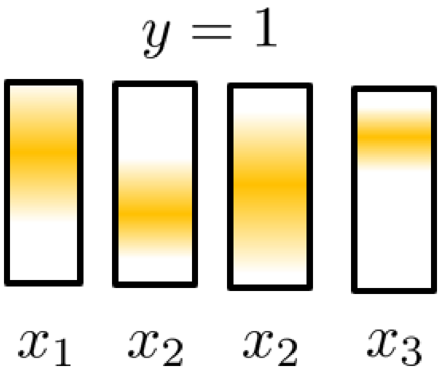
Photogrammetry & Robotics Lab

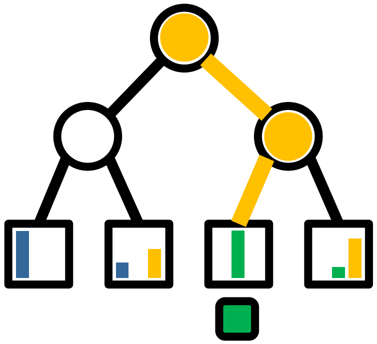
Machine Learning for Robotics and Computer Vision Tutorial

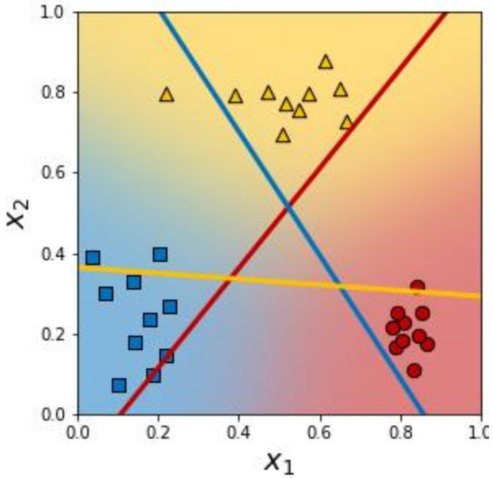
More on Classification

Jens Behley

Topics of this week lecture







- Classification models
  - Naïve Bayes (Generative Model)
  - Decision Tree (Discriminative Model)
  - Logistic/Softmax Regression (Discriminative Model)

- Optimization with Gradient Descent

Logistic Regression

- For **binary** classification $y = \{0, 1\}$, we want:

$$P(y=0|\mathbf{x}) = 1 - P(y=1|\mathbf{x})$$

- In **Logistic Regression** we define our model as:

$$P(y=1|\mathbf{x}) = \sigma(\theta^T \mathbf{x})$$
$$= \frac{1}{1+\exp(-\theta^T \mathbf{x})}$$
$$P(y=0|\mathbf{x}) = 1 - P(y=1|\mathbf{x})$$

- (We used again as in the Linear Regression: $\mathbf{x} := (1, \mathbf{x}^T)^T$)

[CVMLI Chap. 9.1]

Recap: Gradient of NLL

- For the gradient follows:

$$\frac{\partial \mathcal{L}}{\partial \theta} = \frac{1}{\partial \theta}\left(-\sum_{i=1}^{N}(\mathbf{1}\{y_i = 1\} - 1)\theta^T\mathbf{x} - \log\left(1 + \exp(-\theta^T\mathbf{x})\right)\right)$$

$$= -\sum_{i=1}^{N}(\mathbf{1}\{y_i = 1\} - 1)\mathbf{x} - \frac{1}{1+\exp(-\theta^T\mathbf{x})}\exp(-\theta^T\mathbf{x})(-\mathbf{x})$$

$$= -\sum_{i=1}^{N}(\mathbf{1}\{y_i = 1\} - 1)\mathbf{x} + \underbrace{\frac{\exp(-\theta^T\mathbf{x})}{1 + \exp(-\theta^T\mathbf{x})}}_{1-\sigma(\theta^T\mathbf{x})}\mathbf{x}$$

$$= \sum_{i=1}^{N}(\sigma(\theta^T\mathbf{x}) - \mathbf{1}\{y_i = 1\})\mathbf{x}$$

$$= \sum_{i=1}^{N}(P(y_i = 1|\mathbf{x}) - \mathbf{1}\{y_i = 1\})\mathbf{x}$$

- **Problem:** Setting this to zero, no closed form solution!

[CVMLI Chap. 9.1]

How do you check the gradient?

Numerical Gradient

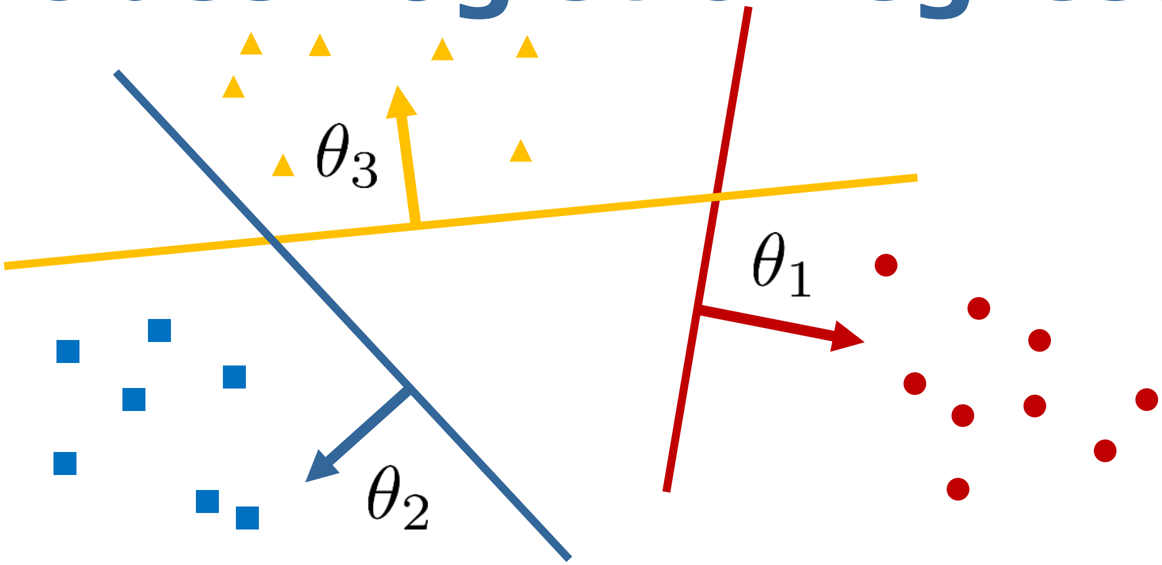
- Compute numerical gradient (h = 1e-7)

$$\frac{\hat{df}}{d\mathbf{x}_i} \approx \frac{f((x_0,\ldots,x_i+h,\ldots,x_D)^T)-f((x_0,\ldots,x_i-h,\ldots,x_D)^T)}{2\cdot h}$$

- Relative error should be small (e.g., 1e-5):

$$\text{rel\_error} = \frac{|\frac{\hat{df}}{d\mathbf{x}_i} - \frac{df}{d\mathbf{x}_i}|}{|\frac{\hat{df}}{d\mathbf{x}_i}|+|\frac{df}{d\mathbf{x}_i}|+1e-12}$$

Multi-class Logistic Regression



- **Idea:** class with largest $\mathbf{x}^T\theta_k$ should have highest confidence $P(y = k|\mathbf{x})$
- Want to find parameters such that distance is maximize for correct class
- How to turn “distances” into probability distribution?

Recap: Softmax

- Softmax function $\mathrm{softmax}: \mathbb{R}^D \mapsto [0,1]^D$ is given by

$$\mathrm{softmax}(\mathbf{x}) = \mathbf{s}$$

with

$$s_i = \frac{\exp(x_i)}{\sum_{d=1}^{D} \exp(x_d)}$$

- Properties of softmax function:
  - $s_i \in [0,1]$
  - $\sum_{i=1}^{D} s_i = 1$

Intuition of Softmax

- Some examples for intuition about the output of softmax function:
  - softmax(10,10,10) = (1/3, 1/3, 1/3)
  - softmax(10,11,10) = (0.21, 0.58, 0.21)
  - softmax(10,13,10) = (0.045, 0.91, 0.045)
  - softmax(9, 11,10) = (0.09, 0.67, 0.24)

More on the softmax intuition

- softmax(9, 11,10) = (0.09, 0.67, 0.24)
- softmax(109, 111,110) = ?

More on the softmax intuition

- softmax(9, 11,10) = (0.09, 0.67,   0.24)
- softmax(109, 111,110) = (0.09, 0.67, 0.24)
- softmax(1009, 1011,1010) = ?

How to fix numerical overflow?

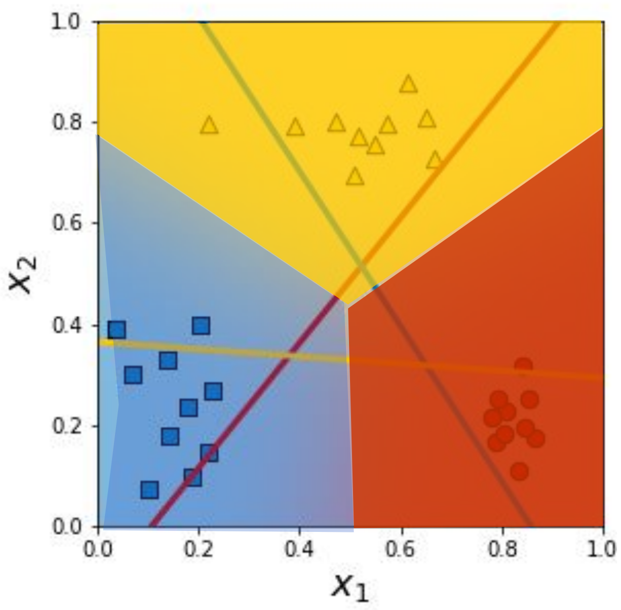
The softmax trick

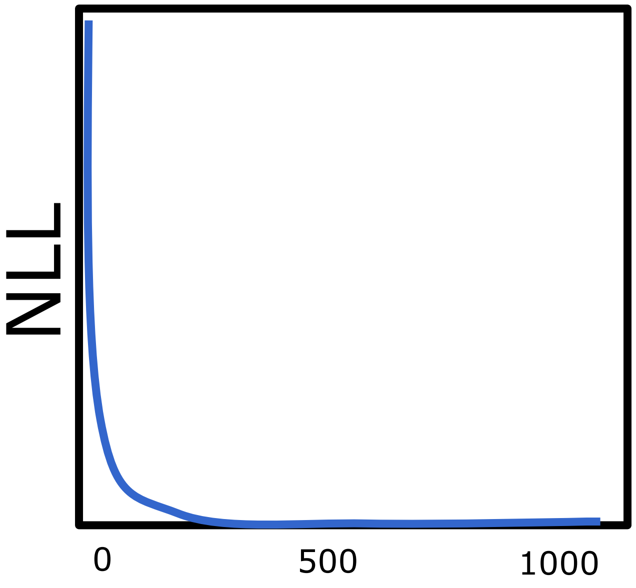
$$\frac{\exp\left(\boldsymbol{a}^{(i)}\right)}{\exp\left(\sum_j \boldsymbol{a}^{(j)}\right)} = \frac{\exp\left(\boldsymbol{a}^{(i)}\right)\cdot\exp(z)}{\exp\left(\sum_j \boldsymbol{a}^{(j)}\right)\cdot\exp(z)}$$

$$= \frac{\exp\left(\boldsymbol{a}^{(i)} + z\right)}{\exp\left(\sum_j \boldsymbol{a}^{(j)} + z\right)}.$$

- Thus, we can use $z = -\max_j a^{(j)}$ to get smaller arguments in the exponention.

Training long enough…

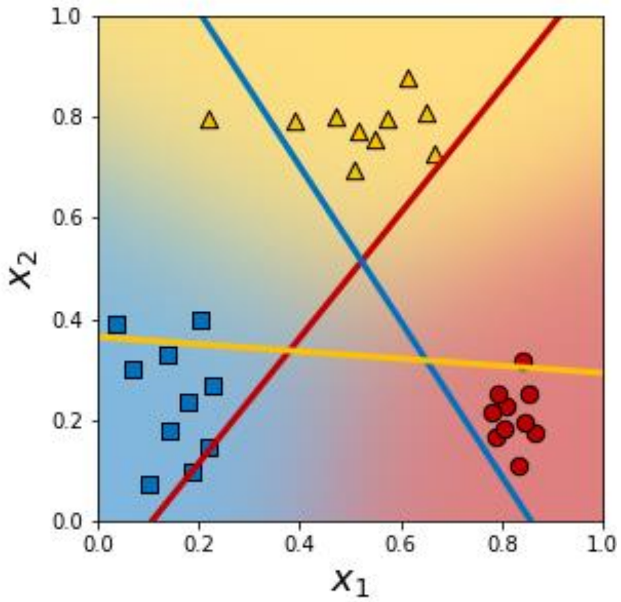


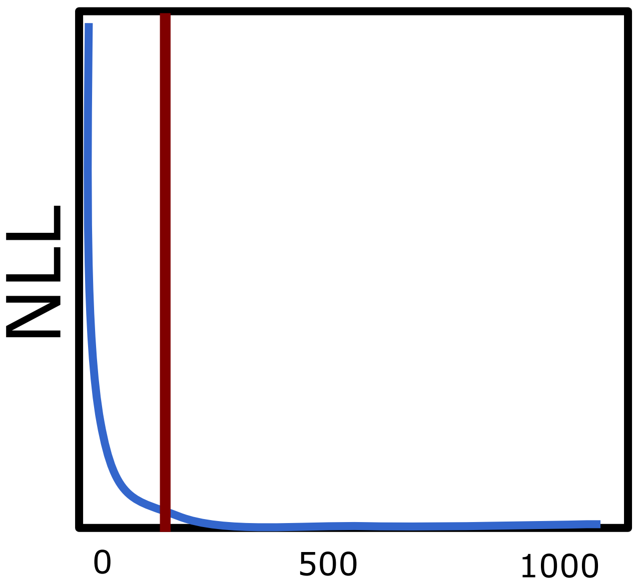


- Train long enough and get "hard" boundaries

How to avoid overfitting?

Early Stopping





- Easy solution: stop early!
- Often used in NN training

See you next week!