Machine Learning Outline: Neural Network and Digit Recognition

Yantao Wu

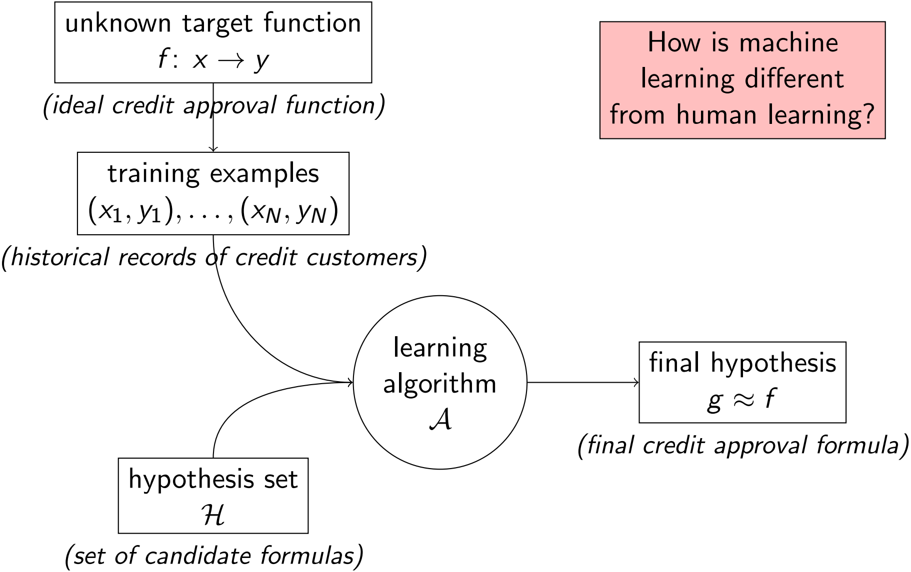
Syracuse University

ywu206@syr.edu

Wednesday 17$^{\text{th}}$ April, 2019

Overview

1. Feasibility of Learning: Outline of Machine Learning



How is machine learning different from human learning?

1. Feasibility of Learning: How approximate it can be?

Theorem (from Hoeffding Inequality)

$$\mathbf{P}[|E_{in}(h) - E_{out}(h)| > \epsilon] \leq 2e^{-2\epsilon^2 N}$$

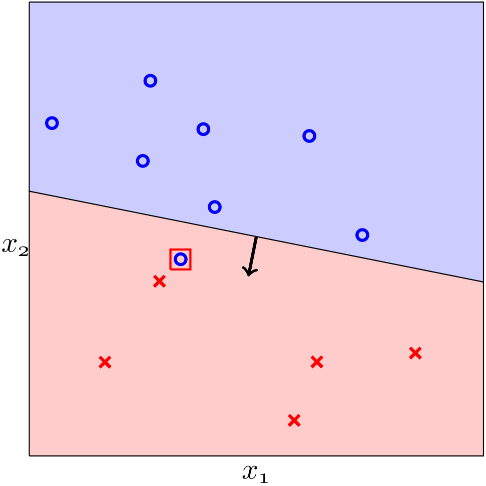
Theorem (VC generalization bound)

$$E_{out}(g) \leq E_{in}(g) + \sqrt{\frac{8}{N} \ln \frac{4m_H(2N)}{\delta}}$$

Conclusion: More data means more certainty on test examples.

Note1: Learning is only feasible in probabilistic sense.
Note2: Decreasing training examples error is the remaining work.

2.1. Linear Classification

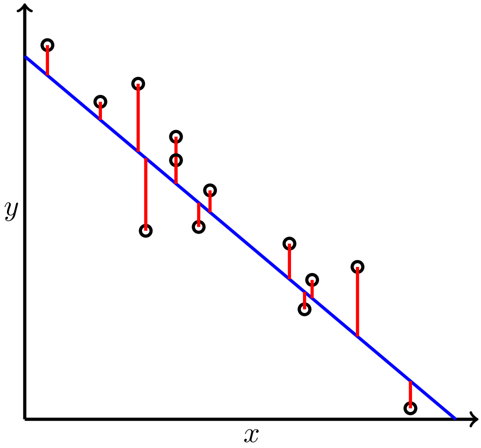


Perceptron Learning Algorithm

$\mathbf{w}(t+1) = \mathbf{w}(t) + y(t)\mathbf{x}(t)$

Pocket Algorithm

$$\min_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N}[\mathrm{sign}(\mathbf{w}^{\mathrm{T}}\mathbf{x}_n) \neq y_n]$$



Linear Regression

$$\min_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N}(\mathbf{w}^T\mathbf{x}_n - y_n)^2$$

$$\nabla E_{in}(\mathbf{w}) = \frac{2}{N}(\mathrm{X}^{\mathrm{T}}\mathrm{X}\mathbf{w} - X^T\mathbf{y})$$

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

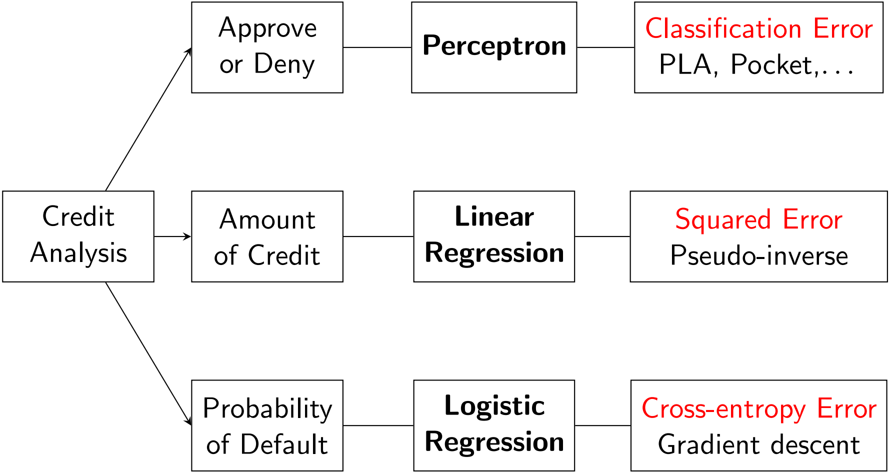
Logistic Regression

$$\min_{\mathbf{w}\in\mathbb{R}^{d+1}} \frac{1}{N}\sum_{n=1}^{N}\theta(\mathbf{w}^T\mathbf{x}_n - y_n)$$

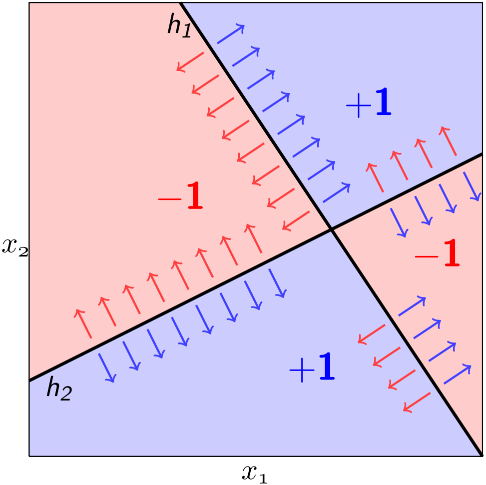
$$\theta(x) = \frac{1}{1+e^{-x}}$$

$$\nabla E_{in}(\mathbf{w}) = -\frac{1}{N}\sum_{n=1}^{N}\frac{y_n\mathbf{x}_n}{1+e^{y_n\mathbf{w}^{\mathrm{T}}(t)\mathbf{x}_n}}$$

2. Summary of One-layer Models



$$f = \mathrm{XOR}(h_1, h_2) = h_1\bar{h}_2 + \bar{h}_1 h_2$$



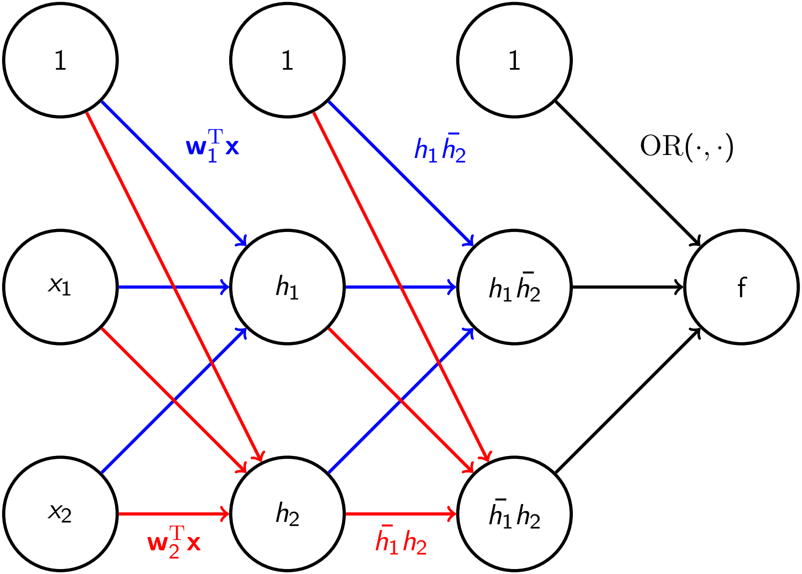
$$\mathrm{OR}(x, y) = \mathrm{sign}(x + y + 1.5)$$
$$\mathrm{AND}(x, y) = \mathrm{sign}(x + y - 1.5)$$

$$\mathrm{Step1}: h_1 = \mathrm{sign}(\mathbf{w}_1^{\mathrm{T}}\mathbf{x})$$
$$h_2 = \mathrm{sign}(\mathbf{w}_2^{\mathrm{T}}\mathbf{x})$$

$$\mathrm{Step2}: h_1\bar{h}_2 = \mathrm{AND}(h_1, -h_2)$$
$$= \mathrm{sign}(h_1 - h_2 - 1.5)$$
$$\bar{h}_1 h_2 = \mathrm{AND}(-h_1, h_2)$$
$$= \mathrm{sign}(-h_1 + h_2 - 1.5)$$

$$\mathrm{Step3}: f = \mathrm{OR}(h_1\bar{h}_2, \bar{h}_1 h_2)$$

3.2 Neural Network Architecture



3.2 Neural Network Architecture

Forward Propagation to compute $\mathrm{h}(\mathrm{x})$:

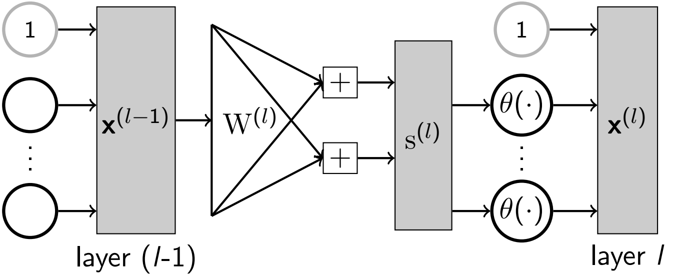
1: $\mathbf{x}^{(0)} \leftarrow \mathbf{x}$ **[Initialization]**

2: **for** $l = 1$ to L **do** **[Forward Propagation]**

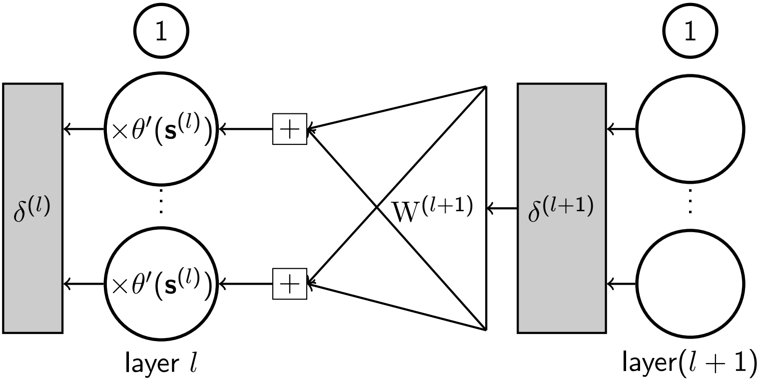
3: $\mathbf{s}^{(l)} \leftarrow (\mathrm{W}^{(l)})^{\mathrm{T}}\mathbf{x}^{(l-1)}$

4: $\mathbf{x}^{(l)} \leftarrow \begin{bmatrix} 1 \\ \theta(\mathbf{s}^{(l)}) \end{bmatrix}$

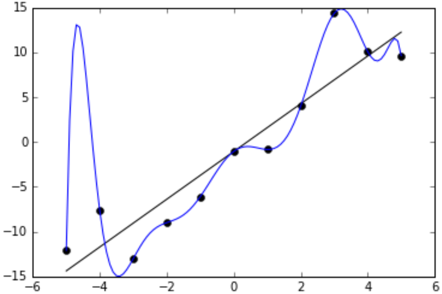
5: $h(\mathbf{x}) = \mathbf{x}^{(L)}$ **[Output]**

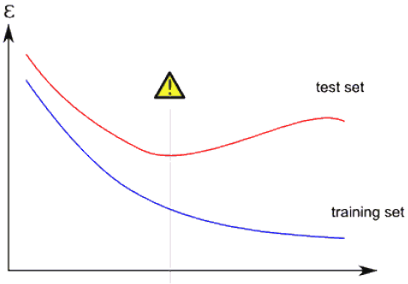


3.3 Back Propagation



4.1 Solving Over-fitting

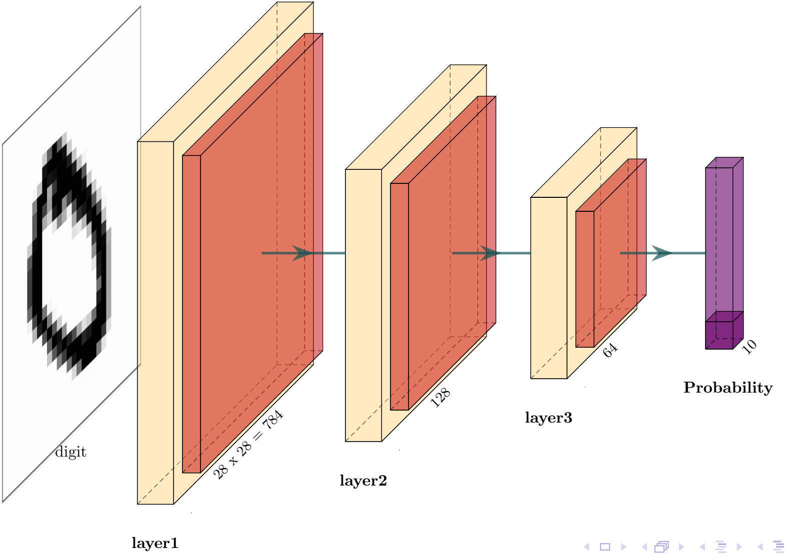




- E_{in} is small, and E_{out} is great.
- $\frac{\partial E}{\partial d_{\mathrm{VC}}} > 0$, where VC-dimension d_{VC} measures complexity of model.
- Regularization: $E_{\mathrm{aug}}(\mathbf{w}) = E_{\mathrm{in}}(\mathbf{w} + \lambda \mathbf{w}^{\mathrm{T}}\mathbf{w})$

Product: Digit Recognition

personal website: `https://williamwuyantao.github.io`



References

Yaser S. Abu-Mostafa (2012)
Learning From Data
AMLBook Ch. 1,3,4,7.

The End