Deep Learning - Theory and Practice

Linear Regression, Least Squares
Classification and Logistic Regression

27-02-2020

Assignment 2
Exam-1.

http://leap.ee.iisc.ac.in/sriram/teaching/DL20/

deeplearning.cce2020@gmail.com

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Midterm Exam-2.

- → March 5 (6:00pm)
  - $\boxed{\text{1 hour}}$
  - In ~~class~~ for online
- → Open book, notes etc (no electronics)
- → MCQ but with reasoning
  - [Till 8pm on 27-2-2020]
- → Absence on 5th [Reason by email]
  - → Separate exam.

Logistic Regression

- 2- class logistic regression

$$p(\mathcal{C}_1|\phi) = y(\phi) = \sigma\left(\mathbf{w}^{\mathrm{T}}\phi\right)$$

- Maximum likelihood solution

$$\nabla E(\mathbf{w}) = \sum_{n=1}^{N}(y_n - t_n)\phi_n$$

- K-class logistic regression

$$p(\mathcal{C}_k|\phi) = y_k(\phi) = \frac{\exp(a_k)}{\sum_j \exp(a_j)}$$

$$a_k = \mathbf{w}_k^{\mathrm{T}}\phi.$$

- Maximum likelihood solution

$$\nabla_{\mathbf{w}_j} E(\mathbf{w}_1, \ldots, \mathbf{w}_K) = \sum_{n=1}^{N}(y_{nj} - t_{nj})\,\phi_n$$

Bishop - PRML book (Chap 3)

Established 1911

INDIAN INSTITUTE OF SCIENCE
भारतीय विज्ञान संस्थान

Logistic Regression $(K > 2)$

$$\boxed{p(C_k / \phi(x))} = \frac{p(\phi(x)/C_k)\, p(C_k)}{\sum_{j=1}^{K} p(\phi(x)/C_j)\, p(C_j)}$$

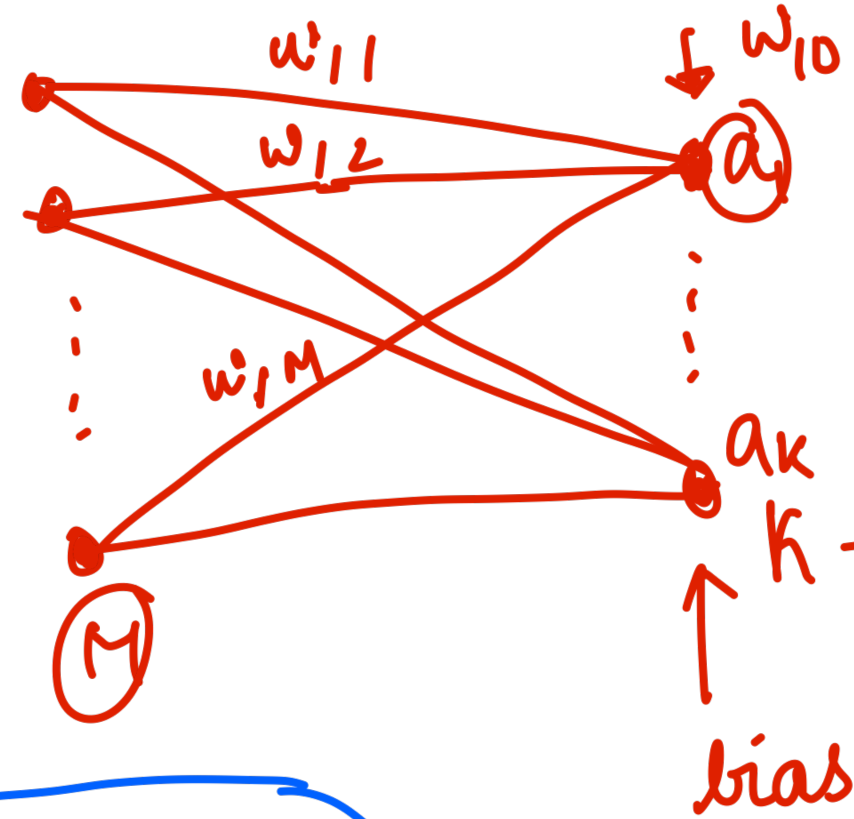
$$\log p(\underline{\phi(x)}, C_k) = a_k = \tilde{w}_k^T \tilde{\phi}(x)$$

Approximation

$$p(C_k / \phi(x)) = \frac{\underline{\exp}(a_k)}{\sum_{j=1}^{K} \exp(a_j)} = \underline{\text{softmax}(a_k)}$$

$$\tilde{W} = \begin{bmatrix} \tilde{w}_1^T \\ \tilde{w}_2^T \\ \vdots \\ \tilde{w}_K^T \end{bmatrix}$$

$\boxed{\phi(x)} \rightarrow$



$\rightarrow \boxed{\text{Softmax}()} \rightarrow \begin{pmatrix} y_1 \\ \vdots \\ y_k \\ \vdots \\ y_K \end{pmatrix}$

$$\tilde{w}_k = \begin{bmatrix} w_{k0} \rightarrow 1 \\ w_{k1} \\ \vdots \\ w_{kM} \end{bmatrix}$$

Properties of $\boxed{y_k = \dfrac{e^{a_k}}{\sum_j e^{a_j}}}$

(i) $\sum_k y_k = 1$

(ii) $0 \le y_k \le 1 \,;\, \forall k.$

$\boxed{\phi(\underline{x}) \in \mathbb{R}^M}$

$a_k = \tilde{w}_k^T \tilde{\phi}(x)$

MNIST $\quad M = 784 \ (28^2) \quad K = 10$

Training

$$\tilde{W} = [\tilde{w}_1 \ldots \tilde{w}_k]$$

Likelihood

$$T = [\underline{t}_1 \ldots \underline{t}_N]$$

$\underline{t}_n$ as one-hot encoding $\in \mathbb{R}^K$

$$P(T/\tilde{w}_1 \ldots \tilde{w}_k) = \prod_{n=1}^{N} \prod_{k=1}^{K} \Big(\underbrace{P(C_k/\phi(x_n))}_{y_k(\phi(x_n))}\Big)^{t_{nk}}$$

$\vdots$

$$\frac{\partial L}{\partial w_j} = \boxed{-\sum_{n=1}^{N} (y_{nj} - t_{nj})\, \phi(x_n)}$$

Cross entropy $= -\log P(T/\ldots)$

Soln is derived in a iterative fashion

$$\tilde{W}^{t+1} = \tilde{W}^{t} - \eta \frac{\partial L}{\partial \tilde{W}}\Big|_{\tilde{W} = \tilde{W}^{t}}$$

[Gradient Descent]

Improving Learning $\rightarrow$ Batchwise Training

$\rightarrow$ Stochastic Gradient Descent (SGD)

Gradient $\rightarrow$ $\dfrac{\partial L}{\partial W}$ is used $-\eta \times \dfrac{\partial L}{\partial W}$.

All samples are ind. $\rightarrow \displaystyle\sum_{n=1}^{N} \frac{\partial L_n}{\partial W} \rightarrow \frac{1}{N} \sum_{n=1}^{N} \frac{\partial L_n}{\partial W}$

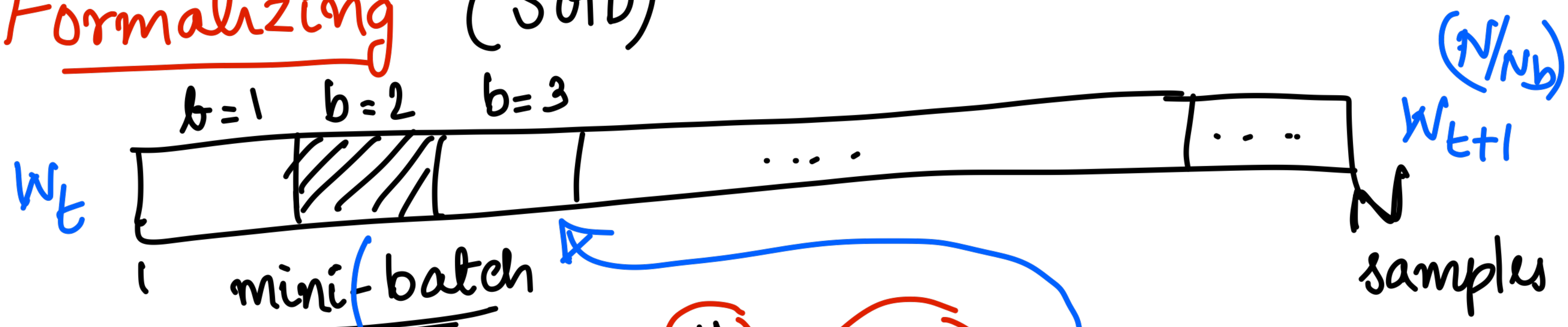
and take factor N into l.r.

Can we approximate average with smaller sample size $N \rightarrow N_b$ [Batch]

Typical batch size 128, 64, ... (few hundreds).

$\hookrightarrow$ Trade off between speed v/s accuracy.

Formalizing (SGD)



$$W^{t+1} = W^t - \eta \frac{1}{N_b} \sum_{\substack{n=1 \\ n \in b}}^{N_b} \left.\frac{\partial L_n}{\partial W}\right|_{W=W^t} \rightarrow W^{t+1}$$

$N_b = 100$

For eg:

$N = 60{,}000$

Assumption

* Each mini batch is a good representation of full batch
* Random shuffling of data before choosing mini-batch
* One run over all minibatches → iteration/epoch.

Epoch 1

W^0 | $b=1$ | $b=2$ | ... | $b = N/N_b$ | W^{599}

W^1 W^2 W^{35}

$N_b = 100$

$N = 60000$

$N/N_b = 600$

Epoch 2 (No more shuffling)

W^{599} . . . . W^{1199}

* After each epoch, model validation

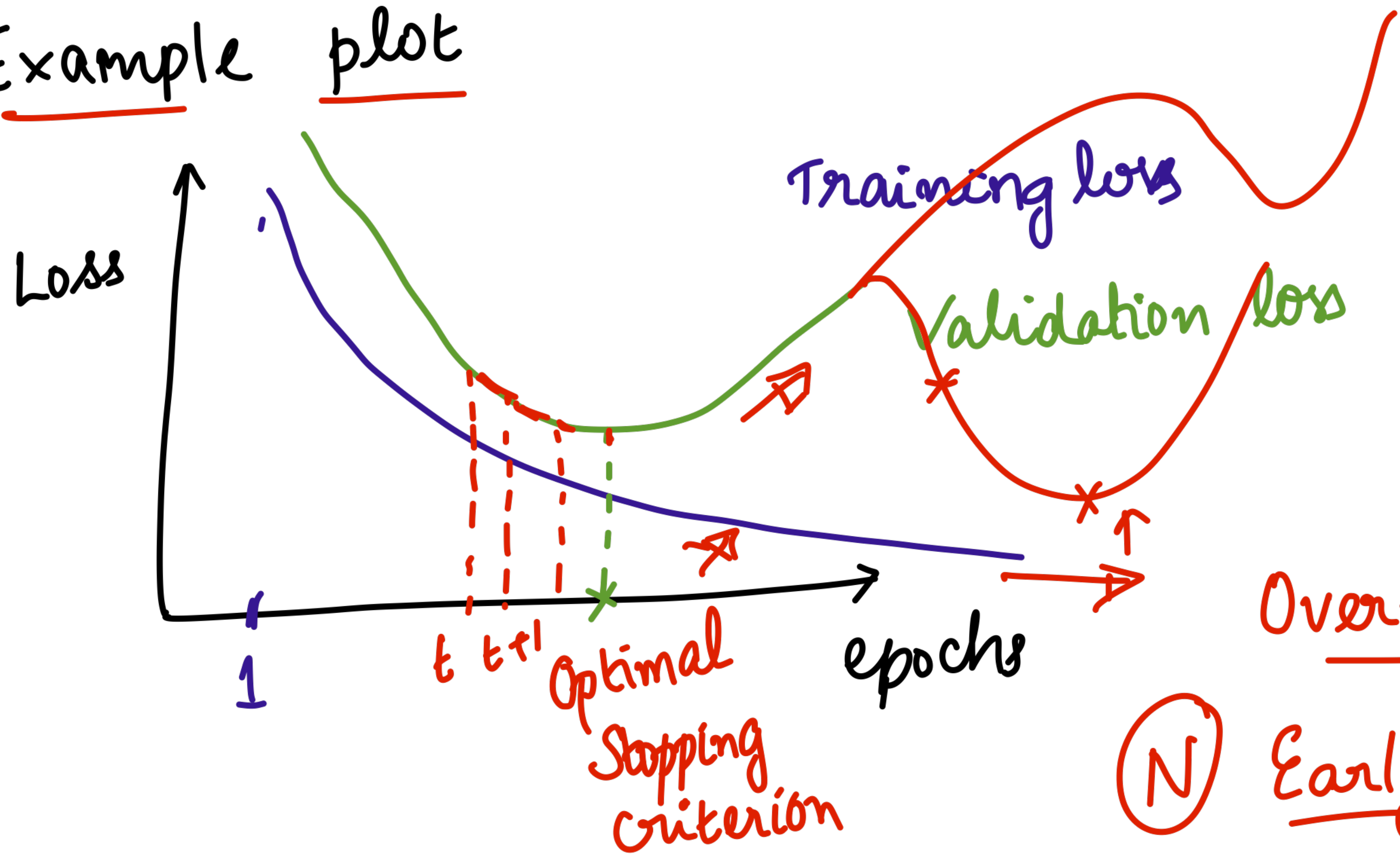
Labeled Data (Shuffling)

90%	10%
1 — Training — N	Held-out Validation

↳ Evaluate Training, Validation loss [Accuracy]

For ex: Cross entropy loss.

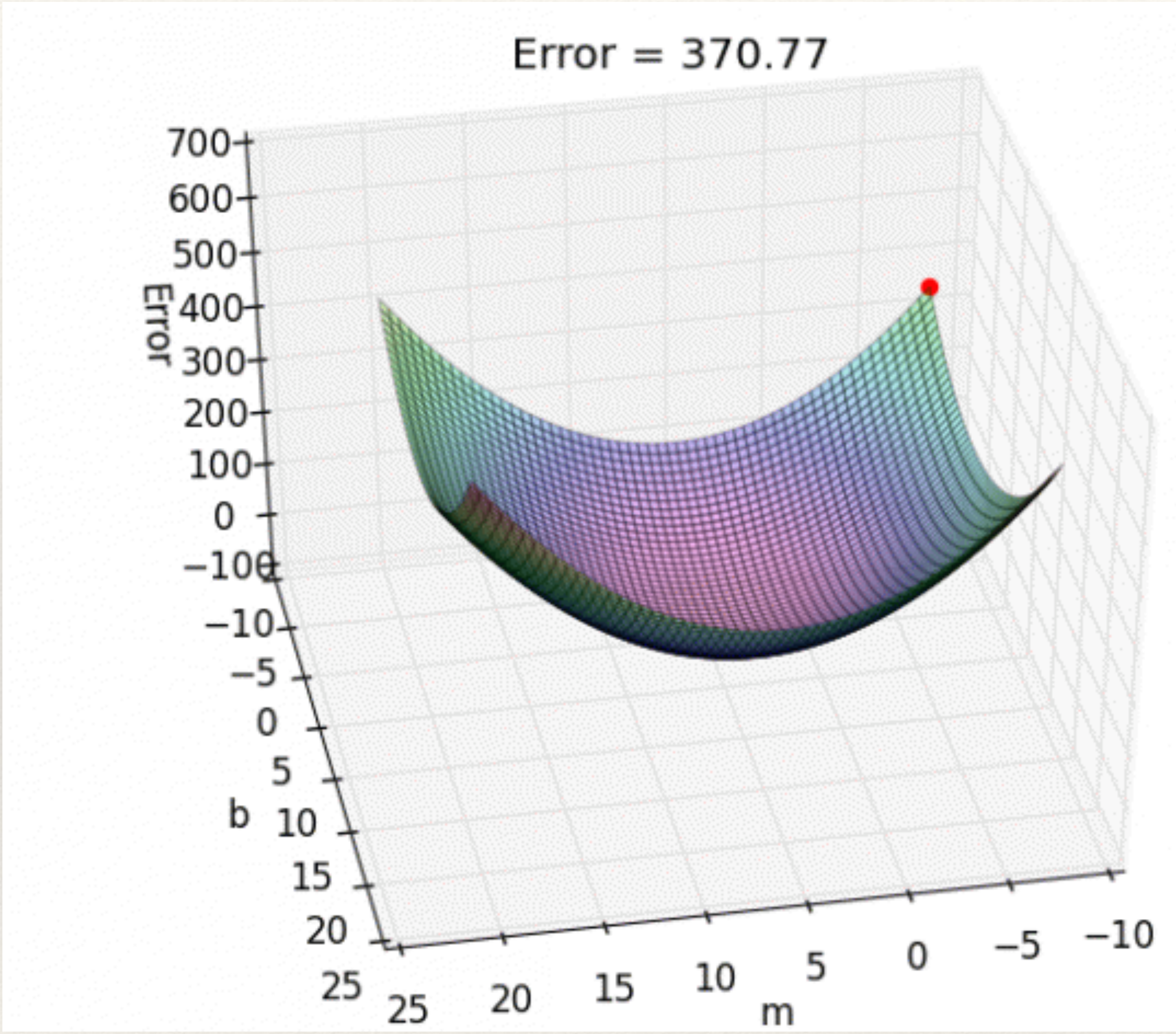
Example plot

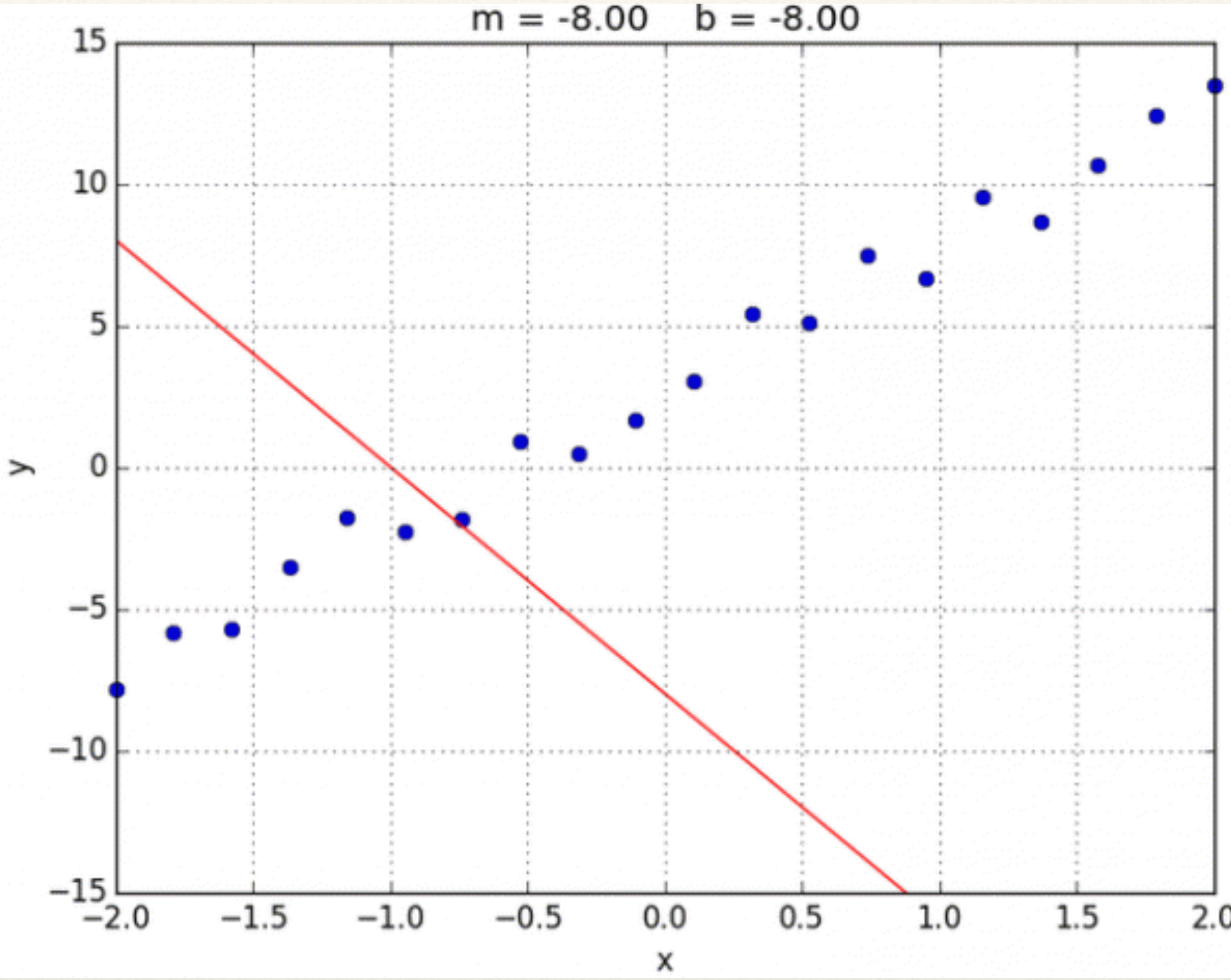


Over-fitting

(N) Early stopping

Learning Using Gradient Descent

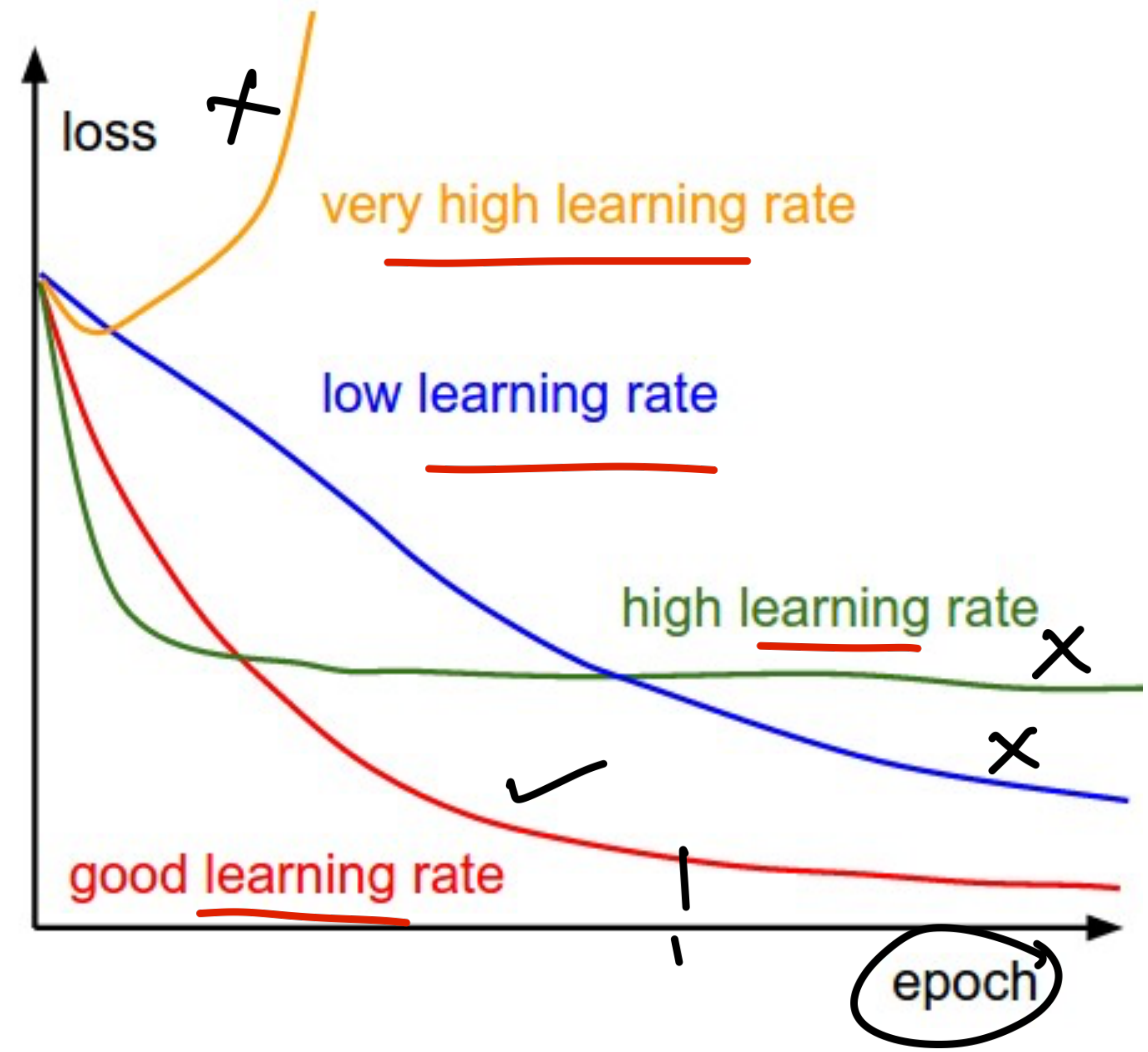




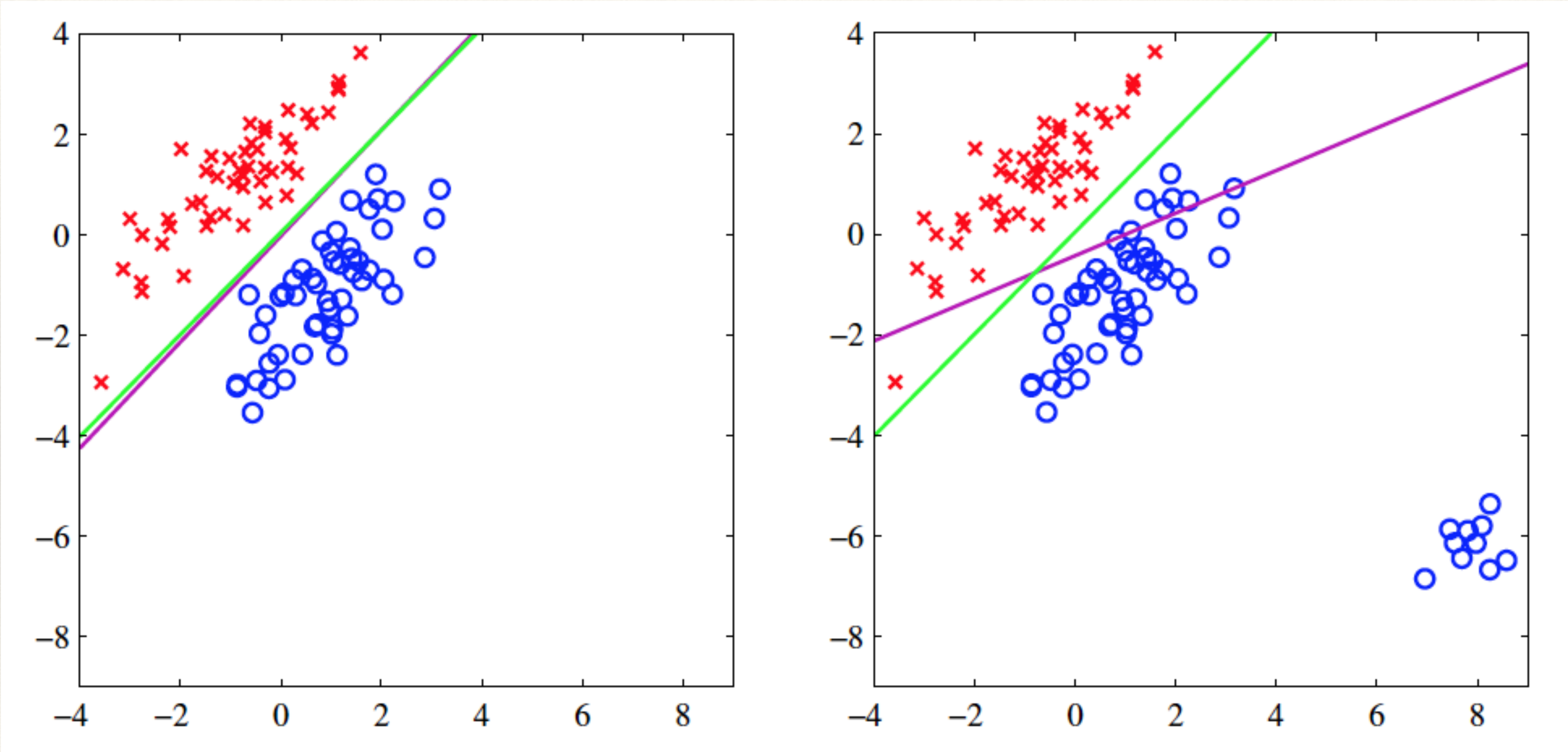
Assignment — Back propagation code it by hand. $\phi(x) = \underline{\underline{x}}$

Parameter Learning

- Solving a non-convex optimization.
- Iterative solution.
- Depends on the initialization.
- Convergence to a local optima.
- Judicious choice of learning rate



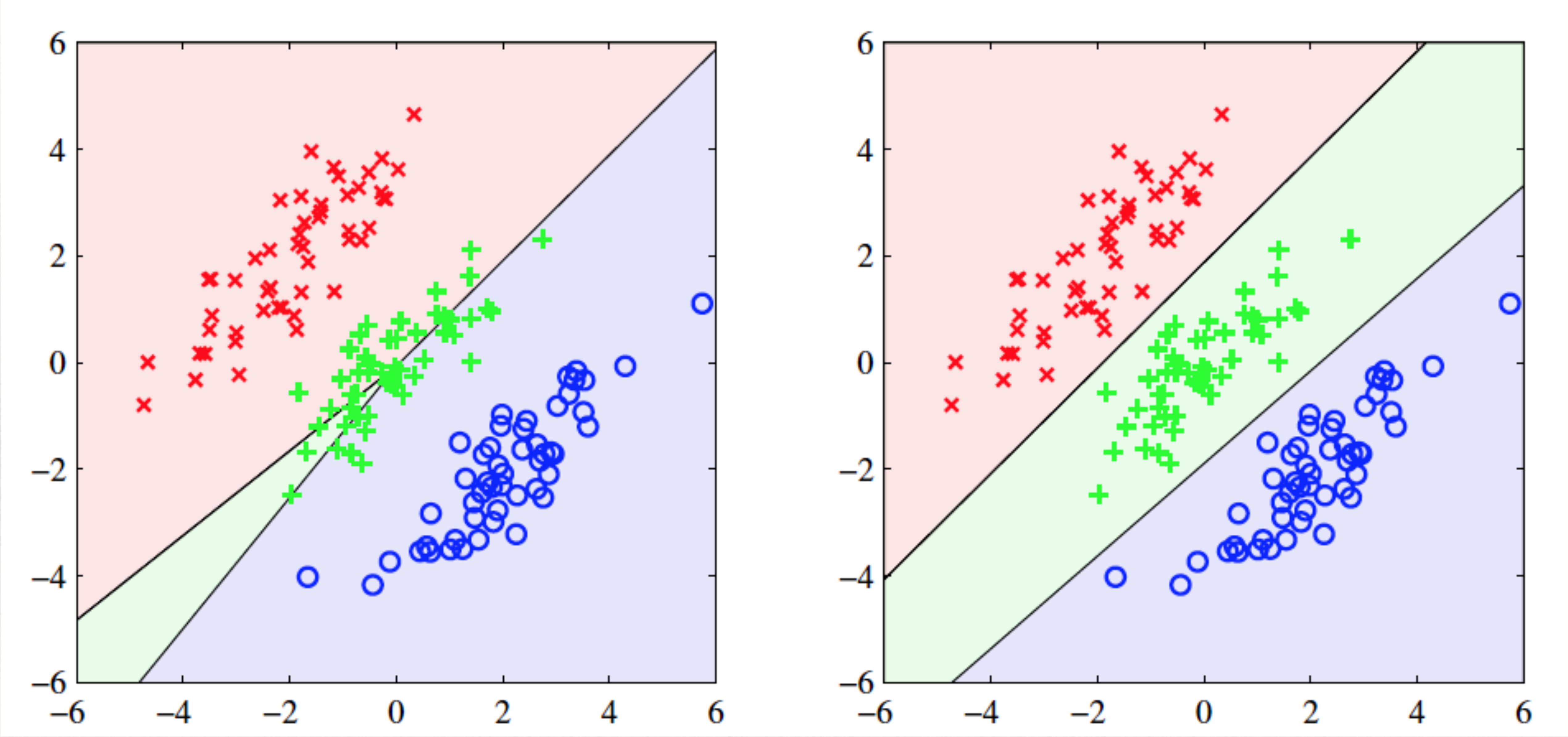
Least Squares versus Logistic Regression



Bishop - PRML book (Chap 4)

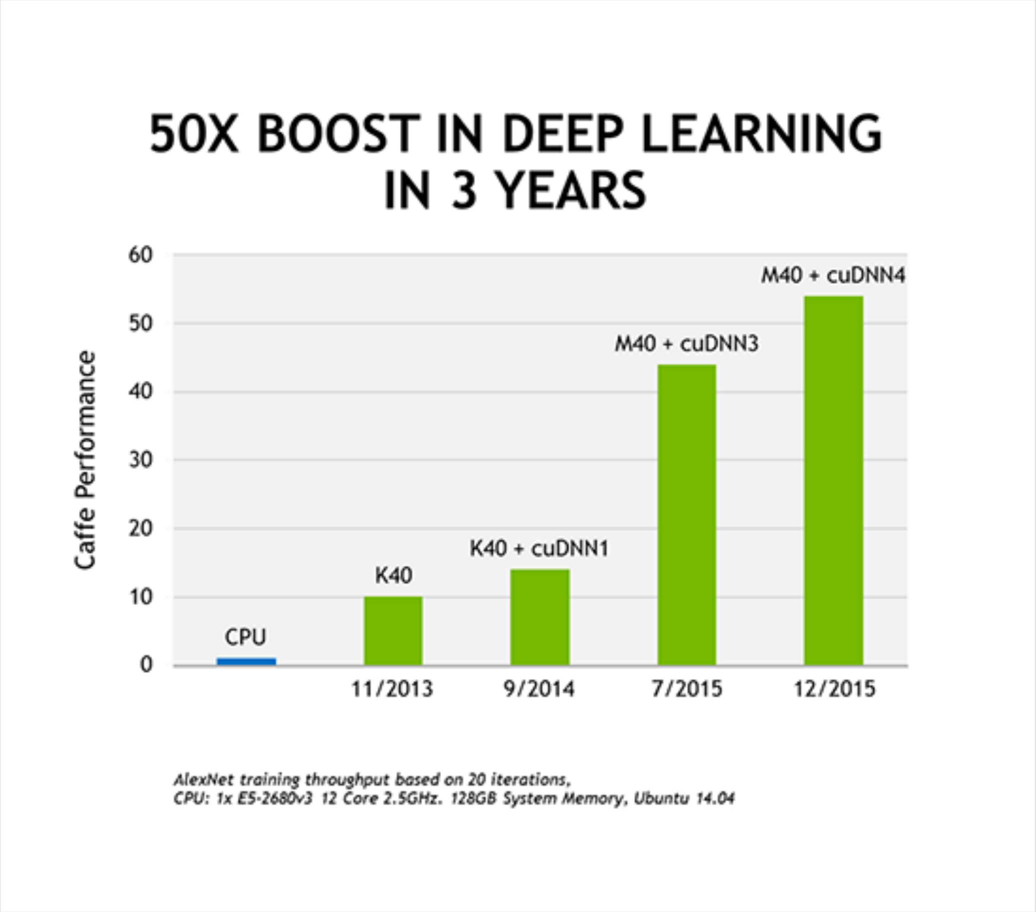
Established 1911

Least Squares versus Logistic Regression



Bishop - PRML book (Chap 4)

Deep Networks



- Are these networks trainable ?

  - Advances in computation and processing
  - Graphical processing units (GPUs) performing multiple parallel multiply accumulate operations.
  - Large amounts of supervised data sets