BRSU

Neural Networks
Assignment 5

Bastian Lang

November 7, 2015

This report contains the summary, ex3.1 and ex3.2 without any programming/plotting.

1 SUMMARY

Please see figures 1.1 - 1.6.

2 EX3.1

See figure 2.1

3 EX3.2

3.1 A

See figure 3.1

3.2 B

See figure 3.2 for formulas.

Haykin Chapter 3

- Introduction (3.1)
  - Formative years & outstanding researchers
    - McCulloch & Pitts: Introduction NN as computing machines
    - Hebb: First rule for learning
    - Rosenblatt: Perceptron first model for learning with a teacher
  - Rosenblatt introduction of Perceptron for classification of linear separable patterns
  - Perceptron Convergence Theorem
  - Use of perceptrons in adaptive filtering
- Adaptive Filtering Problem (3.2)
  - Dynamical system that does input-output mapping from input vector to output scalar.
  - Training data given
  - Input-Data may
    - be a snapshot of data, where every element originates at a different points in space
    - represent uniformly distributed spaced in time
  - How to design this model around a single neuron?
    - Start with arbitrary weight setting
    - Adjustments made to weights on continuous basis
    - Computations completed within one time interval
  - Neuronal model ~~refferd~~ referred to as Adaptive Filter
  - Two processes within adaptive filtering
    - Filtering
      - Compute $y(i)$ given $x(i)$
      - Compute error signal $e(i)$ given $y(i)$ and $d(i)$
    - Adaptive Process
      - Adjust weights in accordance with $e(i)$
  - linear Neuron → Output equals induced local field

$$y(i) = v(i) = \sum x_k(i)\, w_k(i)$$

Figure 1.1: summary 1

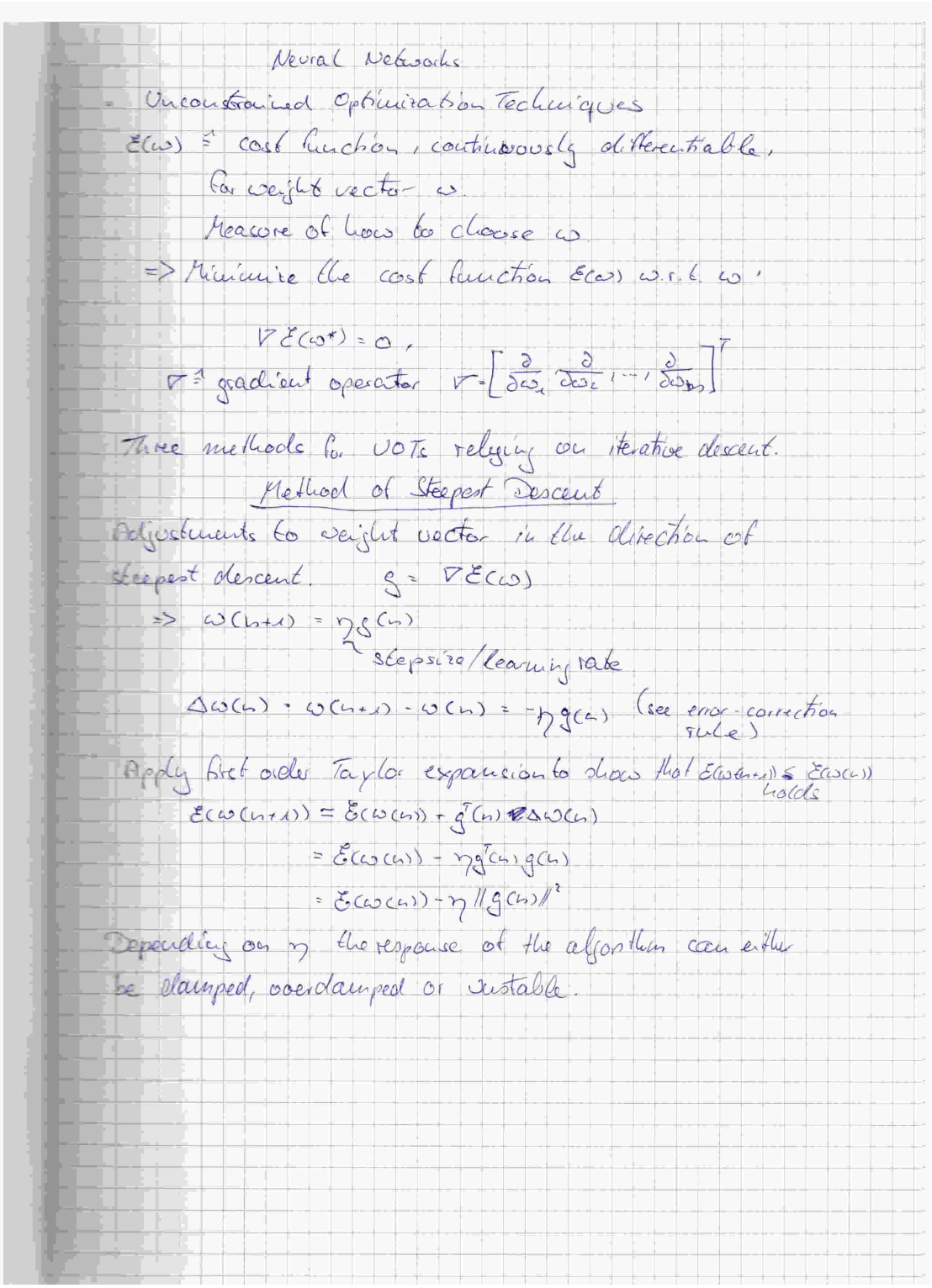

Figure 1.2: summary 2

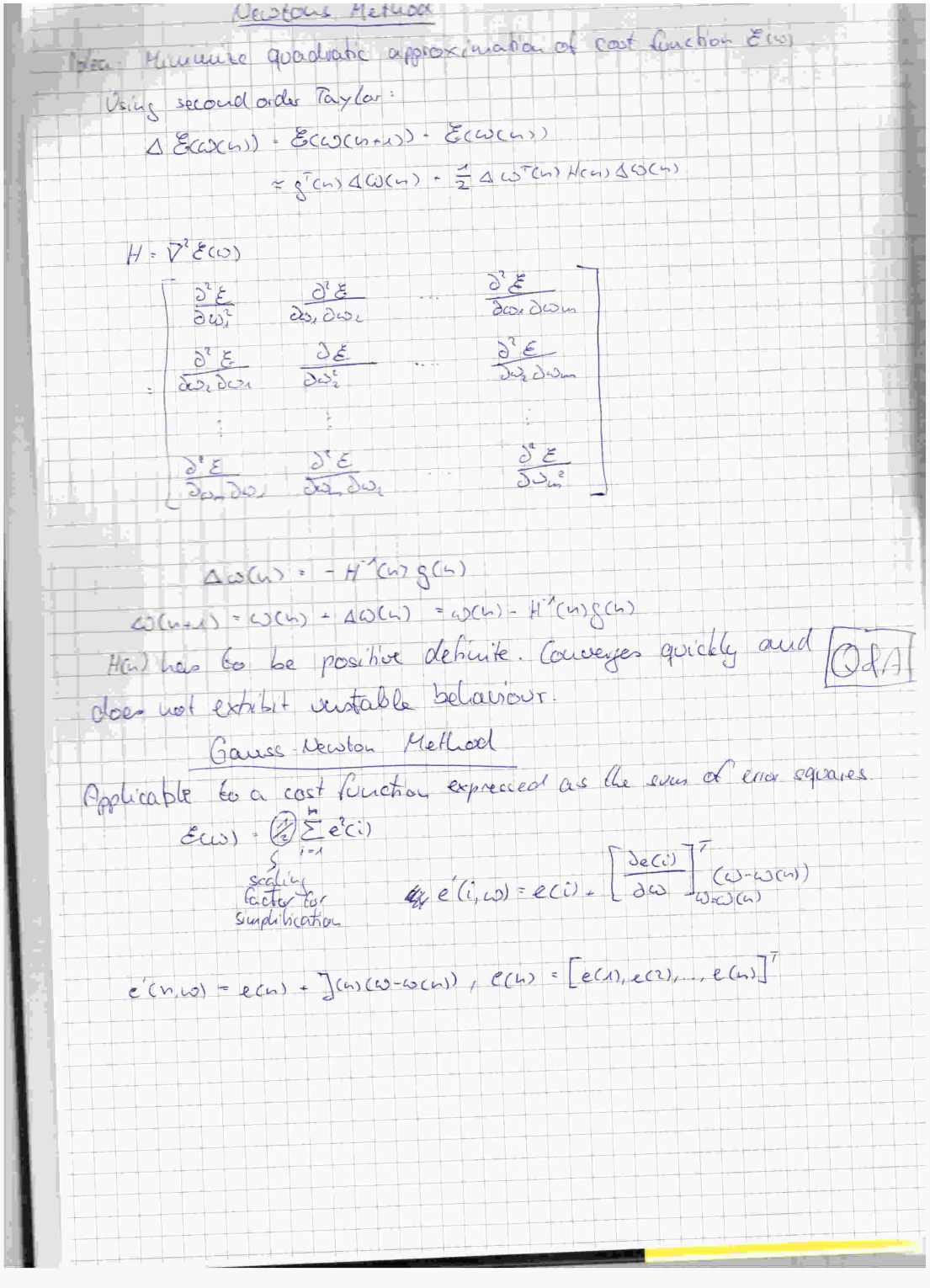

Figure 1.3: summary 3

$$J(n) = \begin{bmatrix} \frac{\partial e(1)}{\partial w_1} & \frac{\partial e(1)}{\partial w_2} & \cdots & \frac{\partial e(1)}{\partial w_m} \\ \frac{\partial e(2)}{\partial w_1} & \frac{\partial e(2)}{\partial w_2} & & \frac{\partial e(2)}{\partial w_m} \\ \vdots & \vdots & & \vdots \\ \frac{\partial e(n)}{\partial w_1} & \frac{\partial e(n)}{\partial w_2} & & \frac{\partial e(n)}{\partial w_m} \end{bmatrix}$$

Jacobian is the transpose of the n-by-m gradient matrix $\nabla e(n)$.

$\Rightarrow w(n+1) = \arg\min_{w} \{ \frac{1}{2} \| e'(n, w) \|^2 \}$

$\Rightarrow J^T(n) e(n) + J^T(n) J(n) (w - w(n)) = 0$

$\Rightarrow w(n+1) = w(n) + (J^T(n) J(n))^{-1} J^T(n) e(n)$

Gauss-Newton only requires the Jacobian of the error vector $e(n)$.

Therefore $J^T(n) J(n)$ must be nonsingular. Q&A

$\Rightarrow$ row rank of $J(n)$ must be $n \triangleq n$ rows must be linearly independent.

To guarantee that, customary practice is to add diagonal matrix δI to ensure positive definite matrix.

$w(n+1) = w(n) - (J^T(n) J(n) + \delta I)^{-1} J^T(n) e(n)$

Linear Least-Squares Filter (3.4)

Two characteristics:

(1) Single neuron is linear

(2) Cost function $\mathcal{E}(w)$ consists of sum of error squares

$e(n) = d(n) - X(n) w(n)$

$\nabla e(n) = -X^T(n)$ $\quad J(n) = -X(n)$

Figure 1.4: summary 4

Applying Gauss-Newton

$w(n+1) = (X^T(n)X(n))^{-1}X^T(n)d(n)$, where $(X^T(n)X(n))^{-1}X^T(n)$ is the pseudo-inverse of $X(n)$.

$\Rightarrow w(n+1) = X^+(n)d(n)$

Wiener Filter: Limiting form of the Linear Least Squares Filter for an Ergodic Environment.

When $x(i)$ and $d(i)$ are drawn from an ergodic, the linear least squares filter asymptotically approaches the Wiener filter.

Requires knowledge: correlation matrix, cross-correlation vector. Not always available $\Rightarrow$ use of linear adaptive filter.

Least-Mean-Square algorithm (3.5)

$\mathcal{E}(n) = \frac{1}{2}e^2(n)$ $\quad \frac{\partial \mathcal{E}(n)}{\partial w} = e(n)\frac{\partial e(n)}{\partial w}$

$e(n) = d(n) - x^T(n)w(n)$

$\frac{\partial e(n)}{\partial w(n)} = -x(n)$ $\quad \frac{\partial \mathcal{E}(n)}{\partial w(n)} = -x(n)e(n)$

Estimate for gradient vector:

$\hat{g}(n) = -x(n)e(n)$

$\hat{w}(n+1) = \hat{w}(n) + \eta x(n)e(n)$

The smaller η, the more past data is remembered. The inverse of η may thus be seen as a measure of memory. QQA

LMS algorithm does not require knowledge of the statistics of the environment.

Convergence Considerations of the LMS Algorithm

Convergence in the mean square

$E[e^2(n)] \to$ constant as $n \to \infty$

Figure 1.5: summary 5

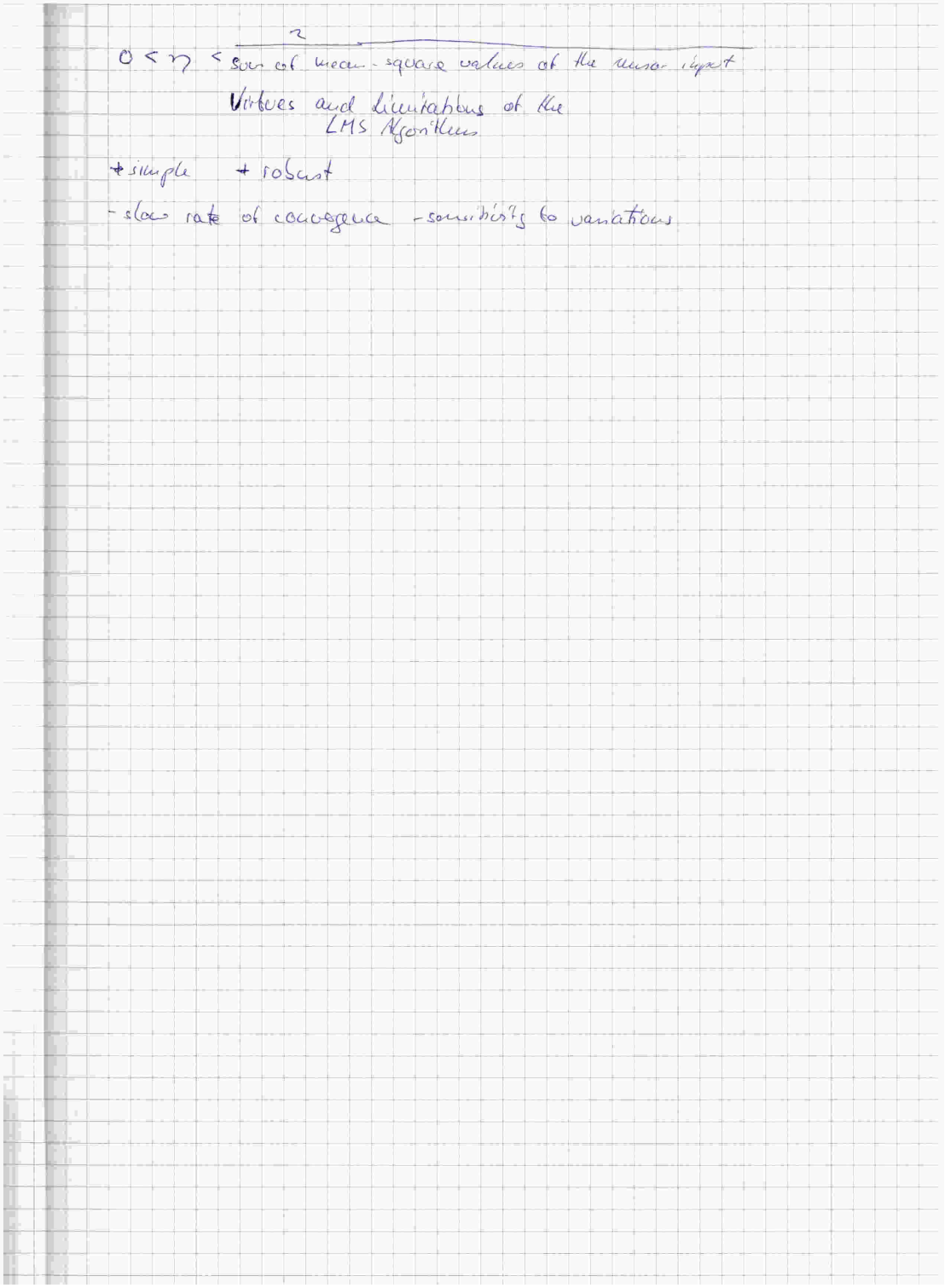

Figure 1.6: summary 6

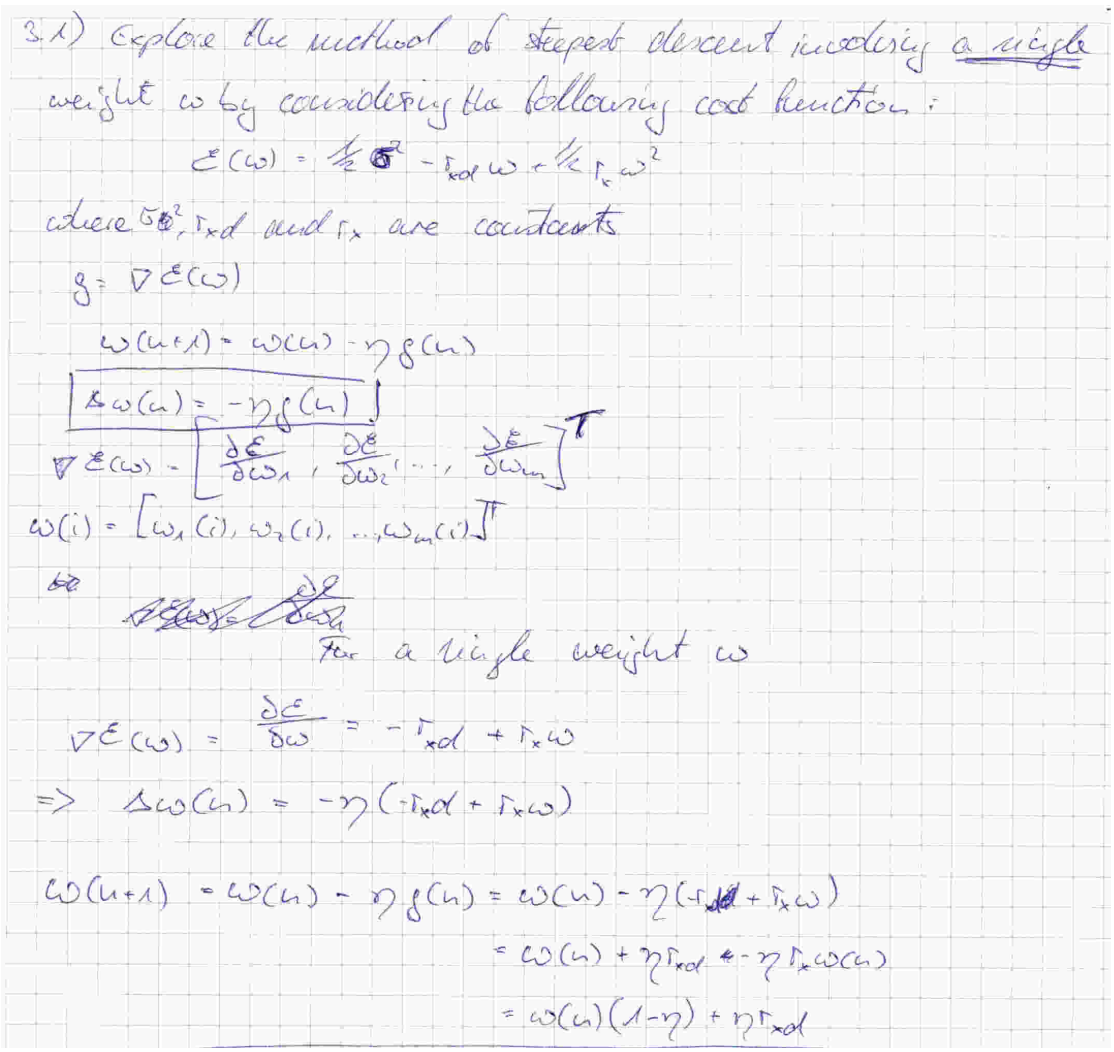

Figure 2.1: ex3.1

8.2) Consider the cost-function

$$\mathcal{E}(w) = \frac{1}{2}\sigma^2 - r_{xd}^T w + \frac{1}{2} w^T R_x w$$

where σ^2 is some constant and

$$r_{xd} = \begin{bmatrix} 0.8182 \\ 0.354 \end{bmatrix}$$

$$R_x = \begin{bmatrix} 1 & 0.8182 \\ 0.8182 & 1 \end{bmatrix}$$

(a) Find the optimum value w^* for which $\mathcal{E}(w)$ reaches its minimum value

$$\mathcal{E}(w) = \frac{1}{2}\sigma^2 - \begin{bmatrix} 0.8182 \\ 0.354 \end{bmatrix}^T \begin{bmatrix} w_1 \\ w_2 \end{bmatrix} + \frac{1}{2} \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} 1 & 0.8182 \\ 0.8182 & 1 \end{bmatrix} \begin{bmatrix} w_1 \\ w_2 \end{bmatrix}$$

$$= \frac{1}{2}\sigma^2 - 0.8182 w_1 - 0.354 w_2 + \frac{1}{2} \begin{bmatrix} w_1 & w_2 \end{bmatrix} \begin{bmatrix} w_1 + 0.8182 w_2 \\ 0.8182 w_1 + w_2 \end{bmatrix}$$

$$= \frac{1}{2}\sigma^2 - 0.8182 w_1 - 0.354 w_2 + \frac{1}{2}\left(w_1^2 + 0.8182 w_1 w_2 + 0.8182 w_1 w_2 + w_2^2\right)$$

$$= \frac{1}{2}\sigma^2 - 0.8182 w_1 - 0.354 w_2 + \frac{1}{2} w_1^2 + \frac{1}{2} w_2^2 + 0.8182 w_1 w_2$$

$$\frac{\partial \mathcal{E}}{\partial w_1} = -0.8182 + w_1 + 0.8182 w_2$$

$$\frac{\partial \mathcal{E}}{\partial w_2} = -0.354 + w_2 + 0.8182 w_1$$

$$\Delta w_1(n) = -\eta\left(-0.8182 + w_1 + 0.8182 w_2\right)$$

$$\Delta w_2(n) = -\eta\left(-0.354 + w_2 + 0.8182 w_1\right)$$

$\nabla \mathcal{E}(w^*) \doteq 0$

$$w_1 = -0.8182 w_2 + 0.8182 \quad (1)$$

$$w_2 = -0.8182 w_1 + 0.354 \quad (2)$$

$$w_1 = -0.8182\left(-0.8182 w_1 + 0.354\right) + 0.8182 \quad (2) \text{ in } (1)$$

$$w_1 = +0.8182^2 w_1 - \left(0.8182 \cdot 0.354\right) + 0.8182$$

$$0.331\, w_1 = 0.52856$$

$$w_1 \approx 1.597$$

$$w_2 = -0.8182 \cdot 1.597 + 0.354$$

$$= -0.9527$$

Figure 3.1: ex3.2a

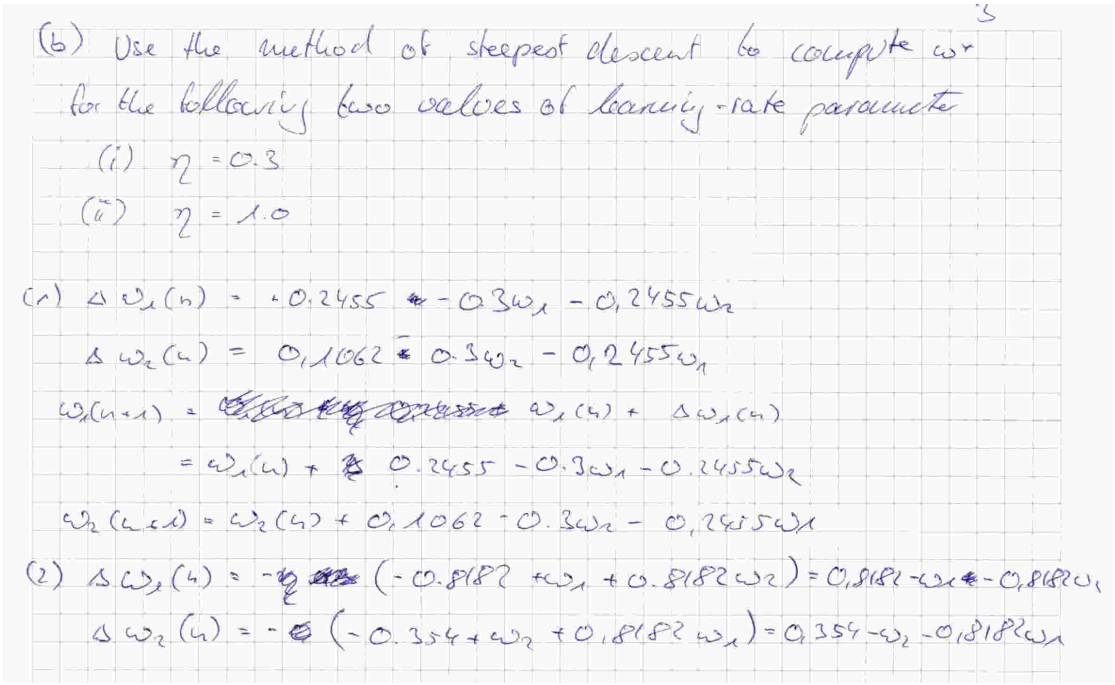

Figure 3.2: ex3.2b