


# Machine Learning for Signal Processing

*[5LSL0]*

Rik Vullings  
Ruud van Sloun  
Nishith Chennakeshava  
Hans van Gorp

**Answers: Optimum Linear and Adaptive Filters,  
and Familiarisation with Pytorch**

April 2023

This document outlines all the answers to the questions that are textual and/or graphical. As a companion to this answer sheet, we also provide the code answers separately. If an answer is (partially) provided as code we use a  icon.

The stochastic nature of some of the optimizers will mean that your images might differ slightly from the images we show here. However, in general their overall form and properties should be the same.

## Exercise 1: Wiener Filter

(a) [1 pt]

$$J = E\{y^2\} - \underline{\mathbf{w}}^t \underline{\mathbf{r}}_{yx} - \underline{\mathbf{r}}_{yx}^t \underline{\mathbf{w}} + \underline{\mathbf{w}}^t \underline{\mathbf{R}}_x \underline{\mathbf{w}} \quad (1)$$

$$\underline{\nabla} J = \frac{dJ}{d\underline{\mathbf{w}}} = -2 (\underline{\mathbf{r}}_{yx} - \underline{\mathbf{R}}_x \underline{\mathbf{w}}) = \underline{0} \quad (2)$$

$$\underline{\mathbf{w}}_0 = \underline{\mathbf{R}}_x^{-1} \underline{\mathbf{r}}_{yx} = \begin{pmatrix} 0.2 \\ 1 \\ -0.5 \end{pmatrix}. \quad (3)$$

(b) [1 pt]

$$\underline{\mathbf{r}}_{xe} = E\{\underline{\mathbf{x}}(y - \underline{\mathbf{x}}^t \underline{\mathbf{w}}_0)\} \quad (4)$$

$$= \underline{\mathbf{r}}_{yx} - \underline{\mathbf{R}}_x \underline{\mathbf{w}}_0 \quad (5)$$

$$= \underline{\mathbf{r}}_{yx} - \underline{\mathbf{R}}_x \underline{\mathbf{R}}_x^{-1} \underline{\mathbf{r}}_{yx} = \underline{0} \quad (6)$$

$$(7)$$

The optimum filter decorrelates the input signals. The portion of  $y$  that is correlated to  $x$  is reconstructed in  $\hat{y}$  and subsequently subtracted from  $y$ .

(c) [1 pt]

$$\hat{\underline{\mathbf{R}}}_x = \frac{1}{L} \sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i] \underline{\mathbf{x}}^t[k-i] \quad (8)$$

$$\hat{\underline{\mathbf{r}}}_{yx} = \frac{1}{L} \sum_{i=0}^{L-1} \underline{\mathbf{x}}[k-i] y[k-i] \quad (9)$$

The trade-off in this calculation is the choice of a proper  $L$ . When  $L$  is too short, the statistics cannot be determined reliably. When  $L$  is too long, the statistics might as well be unreliable in case of non-stationary statistics of the inputs.

## Exercise 2: Steepest Gradient Descent

(a) [1 pt] The steady state is reached when

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha (\underline{\mathbf{r}}_{yx} - \underline{\mathbf{R}}_x \underline{\mathbf{w}}[k]) = \underline{\mathbf{w}}[k] \quad (10)$$

$$2\alpha (\underline{\mathbf{r}}_{yx} - \underline{\mathbf{R}}_x \underline{\mathbf{w}}[k]) = \underline{0} \quad (11)$$

$$\underline{\mathbf{w}}[k] = \underline{\mathbf{R}}_x^{-1} \underline{\mathbf{r}}_{yx} = \underline{\mathbf{w}}_0. \quad (12)$$

In other words, steady state is reached when the filter coefficients have converged to the Wiener solution.

- (b) **[1.5 pt]** The eigenvalues  $\lambda$  can be obtained by setting the determinant of  $\mathbf{R}_x - \lambda \mathbf{I}$  equal to zero.


$$\begin{aligned} \begin{vmatrix} 5-\lambda & -1 & -2 \\ -1 & 5-\lambda & -1 \\ -2 & -1 & 5-\lambda \end{vmatrix} &= (5-\lambda)(24-10\lambda+\lambda^2) + (-7+\lambda) + (-22+4\lambda) \\ &= (120-74\lambda+15\lambda^2-\lambda^3) + (-29+5\lambda) \\ &= -\lambda^3 + 15\lambda^2 - 69\lambda + 91 \end{aligned}$$

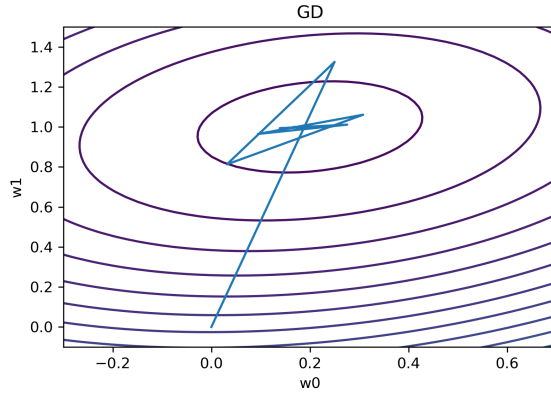
With the hint that one solution is  $\lambda = 7$ , we can divide the term above by  $(\lambda - 7)$ , which yields a factorization as

$$(\lambda - 7)(-\lambda^2 + 8\lambda - 13), \quad (13)$$

giving the solutions  $\lambda_1 = 7$ ,  $\lambda_2 = 4 + \sqrt{3}$ , and  $\lambda_3 = 4 - \sqrt{3}$ . Hence  $\lambda_{max} = 7$  and so stability is obtained for

$$0 \leq \alpha \leq \frac{1}{7}. \quad (14)$$

- (c) **[1 pt]** 



### Exercise 3: Newton's Method

- (a) **[1 pt]**

$$\underline{\mathbf{w}}[k+1] = \underline{\mathbf{w}}[k] + 2\alpha \mathbf{R}_x^{-1} (\underline{\mathbf{r}}_{yx} - \mathbf{R}_x \underline{\mathbf{w}}[k]) \quad (15)$$

Changing variables to  $\underline{\mathbf{d}}[k] = \underline{\mathbf{w}}[k] - \underline{\mathbf{w}}[0]$  yields

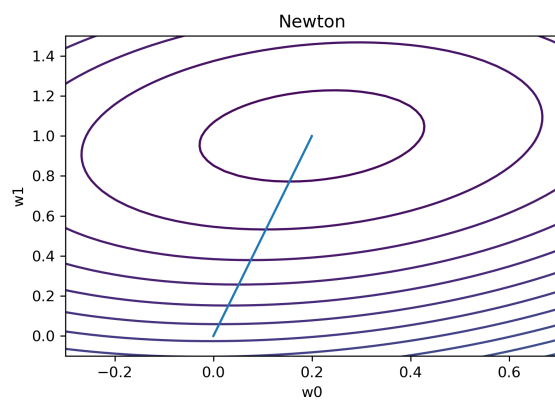
$$\underline{\mathbf{d}}[k+1] = (\mathbf{I} - 2\alpha \mathbf{R}_x^{-1} \mathbf{R}_x)^{k+1} \underline{\mathbf{d}}[0] \quad (16)$$

$$= (\mathbf{I} - 2\alpha)^{k+1} \underline{\mathbf{d}}[0] \quad (17)$$

This equation indicates that the convergence is the same for all filter coefficients  $\underline{\mathbf{w}}$ .

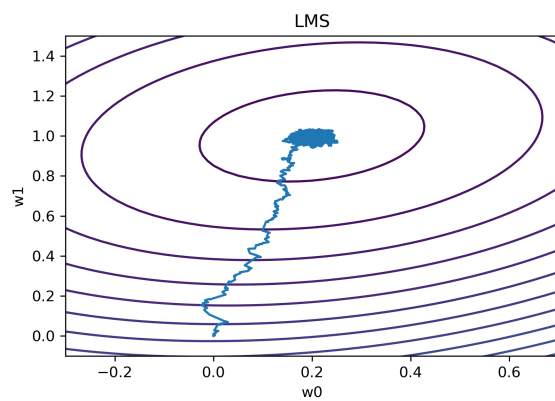
- (b) **[1 pt]** When  $|\mathbf{I} - 2\alpha| < 1$  the method converges. It is stable for  $|\mathbf{I} - 2\alpha| \leq 1$ . This implies that the method is stable for  $0 \leq \alpha \leq 1$ .

(c) [0.5 pt] 🍄

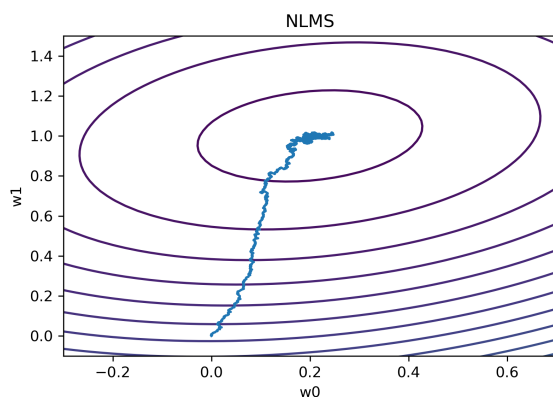


#### Exercise 4: LMS and NLMS

- (a) [1 pt] 🍄 For the choice of  $\alpha$ ,  $\alpha$  should be large enough to have sufficiently fast convergence. At the same time,  $\alpha$  must be sufficiently small to have a stable filter/-convergence.

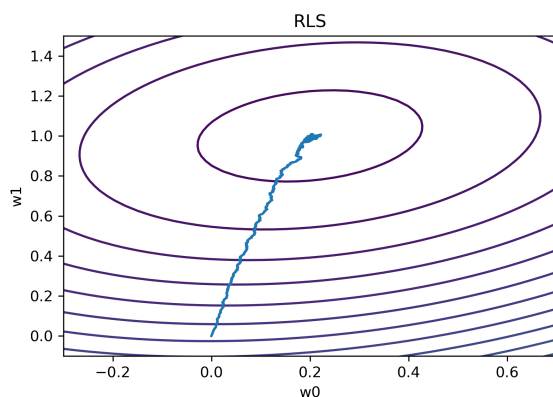


(b) [0.5 pt] 🍄



### Exercise 5: RLS

- (a) [1 pt] Unlike LMS, RLS performs decorrelation leading to improved (fast) convergence. It minimizes a weighted linear least squares cost function, instead of a mean squared error. In RLS the input signals are considered deterministic instead of stochastic (like in (N)LMS).
- (b) [1 pt] 🍄 The influence of  $\gamma$  is that a larger  $\gamma$  reflects a larger forget factor. The memory of the algorithm (i.e. the effective window length of the data that is considered in the filter update) increases with increasing  $\gamma$ . A smaller  $\gamma$  makes the filter more sensitive to recent samples, yielding more fluctuations in the filter coefficients.



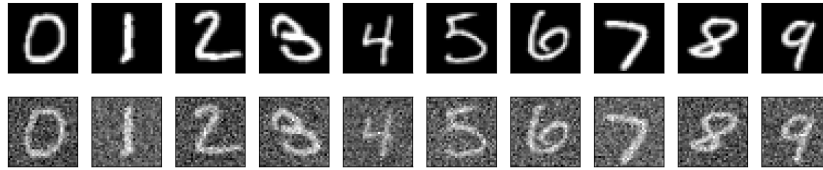
- (c) [1 pt] If we have ample computation power, RLS is the preferred choice having a higher accuracy than (N)LMS. For energy critical solutions, (N)LMS could be the preferred choice.

Filter	Accuracy	Complexity
LMS	worst	best
NLMS	between	between
RLS	best	worst

## Familiarisation with Pytorch

### Exercise 6

- (a) [0.5 pt] ♣

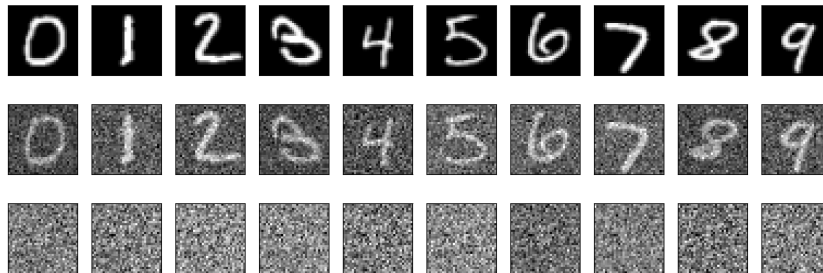


- (b) [1 pt] ♣ A too complex network architecture can lead to over-fitting

- (c) [0.5 pt] ♣ The goal of an optimizer is to update the weights and biases of the model using the gradient of the cost function via application of backpropagation.

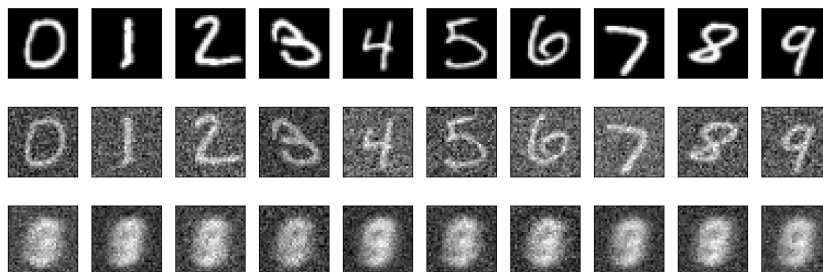
The difference is the use of mini-batches, which effectively subsamples the data for each gradient update. This leads to city in model parameters and is more efficient when using larger datasets, as the computational complexity scales with the size of the mini-batches and not with the size of the entire dataset.

- (d) [0.5 pt] ♣ The prediction is very poor because the weights are initialised randomly.



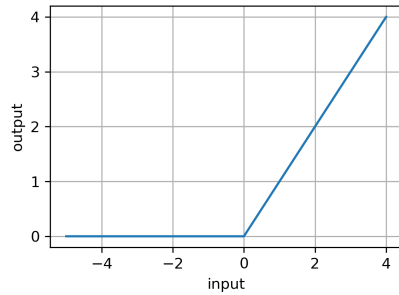
- (e) [1 pt] ♣ If there is a large gap between train and validation loss you might have encountered over-fitting; if the validation loss is much larger than the training loss. If the reverse happened: lucky you! If both training and validation loss are high you are under-fitting. Model complexity could be added in an effort to up the performance.

- (f) [1 pt] ♣ The prediction by the trained model is still very poor because it is linear in nature (as there is no activation function) and cannot capture the non-linearity structure within the features of the noise. It has learned that the content of the digits is in the centre of the figure, while the rest is usually background.

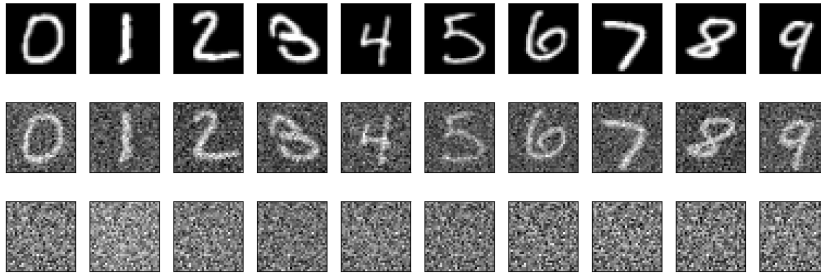


## Exercise 7

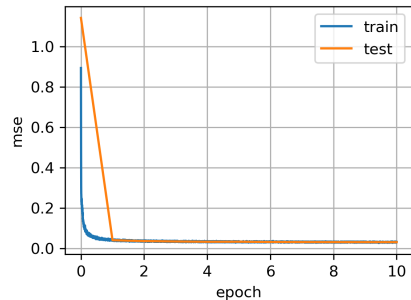
- (a) **[1 pt]** ♣ Key Points: output from the activation function looks like a ReLU (0.75).  
Accurate assessment that the activation function needs to be included after every hidden layer (0.25).



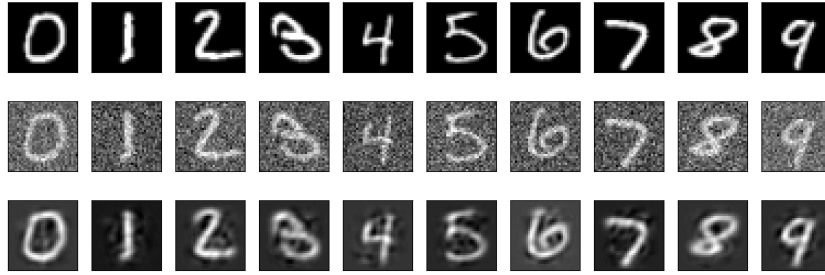
- (b) **[1 pt]** ♣ Key points: They will most likely choose the Adam optimiser.  
Discussion points: Momentum (0.5), adaptive learning rate (0.5).  
If they choose a different optimiser for some reason, look for at least two distinct theory/practice based reason as to why they think it is a better choice.
- (c) **[0.5 pt]** ♣ Key points: Same as 6d



- (d) **[1 pt]** ♣ Key points: Required plots



- (e) **[1.5 pt]** ♣ Key points: shape/clarity/sharpness/contrast of the predicted number (0.5). Inclusion of non-linearities in the network (0.85). Use of a more sophisticated optimiser (0.15).



- (f) **[1 pt]** Key points: Additional layers will not add anything to linear layers. Without activation functions, it will all act like one layer.  
 0.5 for correct answer, 0.5 for the correct reasoning.  
 Calculations are a plus.