Exercise 1

Starting from the sum-of-squares error function

$$E_D(w) = \frac{1}{2}\sum_{n=1}^{N}\{t_n - w^T\phi(x_n)\}^2$$

derive the maximum likelihood solution for the parameters

$$w_{ML} = (\Phi^T\Phi)^{-1}\Phi^T t$$

where

$$\Phi = \begin{pmatrix} \phi_0(x_1) & \phi_1(x_1) & ... & \phi_{M-1}(x_1) \\ \phi_0(x_2) & \phi_1(x_2) & ... & \phi_{M-1}(x_2) \\ \vdots & \vdots & \ddots & \vdots \\ \phi_0(x_N) & \phi_1(x_N) & ... & \phi_{M-1}(x_N) \end{pmatrix}$$

is the design matrix with basis functions $\phi_j(x_i)$, $X = \{x_1, ...x_n\}$ the vectors of input training data and $t = \{t_1, ...t_n\}$ corresponding output training values.

i)

A very clear and brief explanation to this topic can be found in the *Deep Learning Book*, page 109: Since the energy function is quadratic, the optimization solution for one dimension is convex and can be anallitically found by equalling the derivative to zero. But furthermore, a linear combination of convex optimization problems is itself convex too, since multiplication by a scalar and addition of linearly independent terms doesn't affect the outcome: the multiplicating scalars can be ignored (In fact, the usual normalization factor $\frac{1}{M}$ was here disregarded), and the different problems optimized separately.
This is the case when performing parametric multivariate linear regression, with the L_2 energy function: each weight is linearly independent from all others, and contributes to E in a convex way.

ii)

In this terms the optimization objective can be formuled as follows:

$$w_{ML} = \underset{w}{\text{minimize}} \quad E(w, \Phi, t) = \|\Phi w - t\|_2^2 = (\Phi w - t)^T(\Phi w - t)$$

And the analytical way to find it:

$$\begin{aligned}\frac{\partial}{\partial w_{ML}} E(w_{ML}, \Phi, t) = 0 &\iff \frac{\partial}{\partial w_{ML}} \left((\Phi w_{ML} - t)^T (\Phi w_{ML} - t)\right) = 0 \\ &\iff \frac{\partial}{\partial w_{ML}} \left(w_{ML}^T \Phi^T \Phi w_{ML} - 2 w_{ML}^T \Phi^T t + t^T t\right) = 0 \\ &\implies 2\Phi^T \Phi w_{ML} - 2 w_{ML}^T \Phi^T t = 0 \\ &\implies w_{ML} = (\Phi^T \Phi)^{-1} \Phi^T t\end{aligned}$$

□

Exercise 2

Consider a data set in which each data point (x_n, t_n) is associated with a weighting factor $r_n > 0$, so that the sum-of-squares error function becomes

$$E_D(w) = \frac{1}{2} \sum_n r_n (t_n - w^T \phi(x_n))^2$$

Find an expression for the solution $w*$ that minimizes this error function. Give two alternative interpretations of the weighted sum-of-squares error function in terms of (i) data dependent noise variance and (ii) replicated data points.

i)

The energy function can be reformulated as follows:

$$\begin{aligned}\frac{1}{2} \sum_n r_n (t_n - w^T \phi(x_n))^2 &= \frac{1}{2} \sum_n {}_+\sqrt{r_n}^2 (t_n - w^T \phi(x_n))^2 \\ &= \frac{1}{2} \sum_n ({}_+\sqrt{r_n} t_n - {}_+\sqrt{r_n} w^T \phi(x_n))^2\end{aligned}$$

Which brings up the very same optimization objective as the one shown in Exercise 1 (assuming that the weighting factors are given and not learned, that is). Therefore, just two pre-processing calculations are needed:

$$\begin{aligned}t_{ML} &= t \circ {}_+\sqrt{r} \\ \Phi_{ML} &= \Phi \odot {}_+\sqrt{r}\end{aligned}$$

Whereas ${}_+\sqrt{r}$ is the element-weise positive square root of the r vector in $\mathbb{R}^N$, $a \circ b$ represents the element-weise multiplication of two vectors of same dimensionality, and $a \odot b$ abuses this notation, to represent the element-weise multiplication of each vector

in the matrix a with the vector b. In This terms, the solutions is simply to apply the normal equations to the preprocessed data:

$$w_{ML} = (\Phi_{ML}^T \Phi_{ML})^{-1} \Phi_{ML}^T t_{ML}$$

□

Exercise 3

Generate own data sets, e.g. using $t = f(x) + 0.2\epsilon$ *with* $f(x) = sin(2\pi x)$ *and* $\epsilon \sim \mathcal{N}(0, 1)$, *and illustrate the bias-variance decomposition by fitting a polynomial model* $y(x; w) = \sum_{i=0}^{r} w^i x^r$ *to many different data sets* $D_1, ..., D_L$ *, each of length* N.
Let w^D *denote the parameters minimizing the mean squared error on dataset* D. *Then,*

$$bias^2 \approx \frac{1}{L} \sum_l \frac{1}{N} \sum_n (\overline{y}(x) - f(x))^2$$

$$variance \approx \frac{1}{L} \sum_l \frac{1}{N} \sum_n (y(x; w*^{D_t}) - \overline{y}(x)^2$$

where $\overline{y}(x) = \frac{1}{L} \sum_t y(x; w*^{D_t})$

Solution:

See/execute Python2 script*fernandez_blatt5.py* for the details. As explained in the *lecture's slides*), the L_2 loss function can be decomposed into **bias**^{2} **+ variance + noise**. For a given dataset of limited size, only limited assumptions can be done: if only the variance term is taken into account (that is, no regularization term is provided), the hypothesis will maximize its adaptation to the dataset, potentially fitting perfectly to it, but it will fail to generalize, that is: to capture the underlying features. On the other side, a model excesively based on the bias term (that is, with a very high regularization index), will penalize every hypothesis that goes too far away from some given assumptions (in this case, the overall distance to the zero-vector). This assumptions may be unrealistic and relying heavily on them may be therefore a bad strategy.
Both terms are based on the same input parameters, but represent opposite ideas. The bottom line behind this explanation is that a **bias-variance tradeoff** takes always place for datasets of limited size, and, unless the size of the dataset can be increased, a compromise between both of them must be achieved. This is typically achieved by testing many different regularization factors, and cross-validating the results, as shown in the Figure 1:

From these plots, we note that an over-regularized model (large λ) will have a high bias, while an under-regularized model (small λ) will have a high variance.

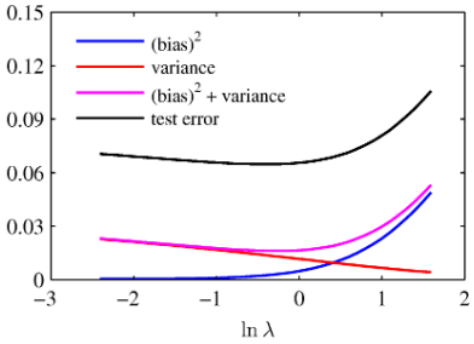

Figure 1: test error and its decomposition for different reg. factors (lecture slides)

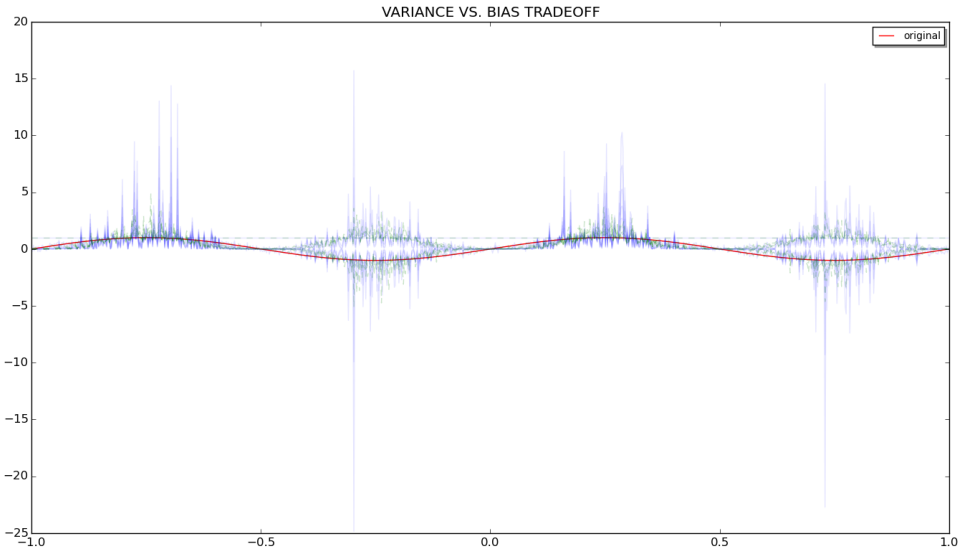

Figure 2: generated example illustrating a case of variance (blue) vs. bias (green) tradeoff