Exercises for Chapter 6

6.1 Consider the data sets $X = \{1, 2, 4, 5\}$, $Y = \{-1, 1, 1, -1\}$.

a) Which function $y = f(x)$ will be found by linear regression?

The data are symmetric with respect to the axis $x = 3$, so the regression function is horizontal at the average of Y, which is $\bar{y} = 1/4 \cdot (-1+1+1-1) = 0$, so the regression model is $f(x) = 0$.

b) When we add an outlier at $x_5 = 3$, $y_5 = 15$, which function $y = f(x)$ will then be found by linear regression?

The data stay symmetric, now with average $\bar{y} = 1/5 \cdot (-1+1+15+1-1) = 3$, so the regression model is $f(x) = 3$.

c) Now we use robust linear regression, $\varepsilon = 4$, with the error functional

$$E_H = \frac{1}{n} \sum_{k=1}^{n} \begin{cases} e_k^2 & \text{if } |e_k| < \varepsilon \\ 2\varepsilon \cdot |e_k| - \varepsilon^2 & \text{otherwise} \end{cases}$$

Which outlier value y_5' would have the same effect as the outlier value y_5 in (b)?

If we use the quadratic error, then the contribution of y_5 to the error sum is $e_5^2 = (15-3)^2 = 12^2 = 144$. If we use the Huber function, then the contribution of y_5' to the error sum is $2\varepsilon \cdot |e_5| - \varepsilon^2 = 8|e_5| - 16$. If we set this to 144, then we obtain $|e_5| = (144+16)/8 = 20$. So, we have the two solutions: $y_5' = 3 + 20 = 23$ or $y_5' = 3 - 20 = -17$.

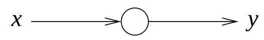
d) How do you interpret these results?

In robust regression the same effect is obtained by a larger outlier, so robust regression is more robust to outliers.

6.2 Sketch an MLP with hyperbolic tangent transfer functions and possible additional constant (bias) inputs at each neuron so that the network (approximately) realizes

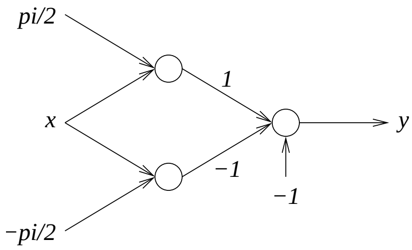
a) a hyperbolic tangent function

We can immediately use the transfer function of one neuron.



b) a cosine function for inputs $x \in [-\pi, \pi]$

We can approximately construct a cosine function in this interval by a hyperbolic tangent function moved left by $\pi/2$ plus a negative hyperbolic tangent function moved right by $\pi/2$ minus a constant 1.

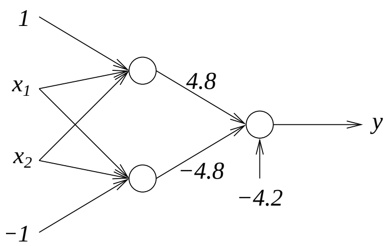


c) an XOR function with inputs x_1, x_2, and output y, so that

x_1	-1	-1	$+1$	$+1$
x_2	-1	$+1$	-1	$+1$
y	-1	$+1$	$+1$	-1

Use the approximations $\tanh(\pm 1) \approx \pm 0.75$, $\tanh(\pm 3) \approx \pm 1$.

We want the same output for the input vectors $(-1, +1)$ and $(+1, -1)$, so we use the sum of both inputs. If this sum is -2 or $+2$, then we want the same output, so we need two paths with opposite signs compensating each other. To achieve the target values ± 1 and ± 3, adding 1 or -1 to the sum of inputs will approximately yield as outputs of the first layer $(0.75, -0.75)$ for inputs $(\pm 1, \mp 1)$, $(-0.75, -1)$ for inputs $(-1, -1)$, and $(1, 0.75)$ for inputs $(+1, +1)$. For opposite signs, taking the differences of the outputs of the first layer yields 1.5 for inputs $(\pm 1, \mp 1)$ and 0.25 for inputs $(\pm 1, \pm 1)$, with a difference of $1.5 - 0.25 = 1.25$. For overall outputs $\tanh \pm 3 \approx \pm 1$. we want the inputs to the second layer to be $+3$ and -3, so we multiply the outputs of the first layer by $(3+3)/1.25 = 4.8$ and then subtract $1.5 \cdot 4.8 - 3 = 0.25 \cdot 4.8 + 3 = 4.2$. This yields $\tanh(4.8 \cdot 1.5 - 4.2) = \tanh(+3) \approx +1$ for inputs $(\pm 1, \mp 1)$ and $\tanh(4.8 \cdot 0.25 - 4.2) = \tanh(-3) \approx -1$ for inputs $(\pm 1, \mp 1)$, as required.



6.3 Consider an MLP with *linear* transfer functions (slope one, offset zero), input layer (neurons 1, 2, and 3), hidden layer (neurons 4, 5, and 6), and output layer (neurons 7 and 8), and the weight matrix

$$W = \begin{pmatrix} 0 & 0 & 0 & 1 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & -1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & a & d \\ 0 & 0 & 0 & 0 & 0 & 0 & b & e \\ 0 & 0 & 0 & 0 & 0 & 0 & c & f \end{pmatrix}$$

with real valued parameters a, b, c, d, e, f.

a) Which function does this MLP realize?

The outputs of the hidden layer are $h_1 = x_1 + x_2$, $h_2 = x_1 - x_2$, and $h_3 = 0$. The outputs of the output layer are $y_1 = ah_1 + bh_2 + ch_3 = (a+b)x_1 + (a-b)x_2$, $y_2 = dh_1 + eh_2 + fh_3 = (d+e)x_1 + (d-e)x_2$.

b) How would you choose the parameters a, b, c, d, e, f so that this MLP can be used as an auto-encoder?

We want $y_1 = x_1$ and $y_2 = x_2$, so $a + b = 1$, $a - b = 0$, $d + e = 0$, and $d - e = 1$, so $a = b = d = \frac{1}{2}$, $e = -\frac{1}{2}$, c and f are arbitrary.

c) What are the advantages and disadvantages of this auto-encoder?

The advantage is that x_1 and x_2 are preserved without loss. The disadvantage is that x_3 gets lost completely.

6.4 What does a small training error and a large validation error indicate?

Overfitting.