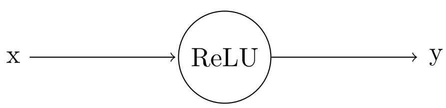
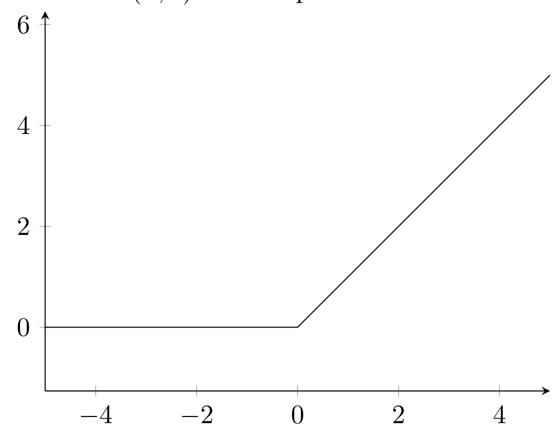
Exercise 1: Function approximation

Name: .... UTID:................

In this exercise you'll use linear functions and Rectified Linear Units (ReLUs) to build simple function approximators. You may work in teams of up to 3 students.

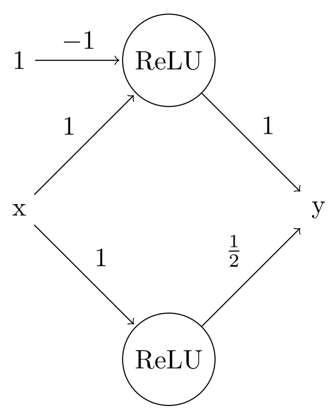
In all examples, we will approximate one dimensional functions: the input will be a single real number, the output another single real number. Formally, a ReLU is defined as $\max(x, 0)$ for an input x. It looks like this:

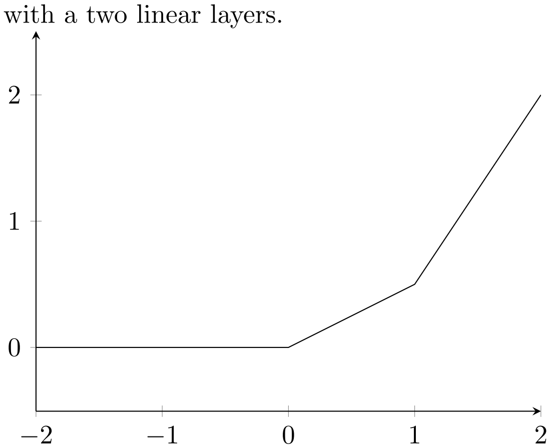




On the right you can see the corresponding network.

Now, let's see what happend if we combine a ReLU layer, with a two linear layers.





Here we subtract one from the input of the first ReLU, and divide the output of the second relu by two. The entire network corresponds to

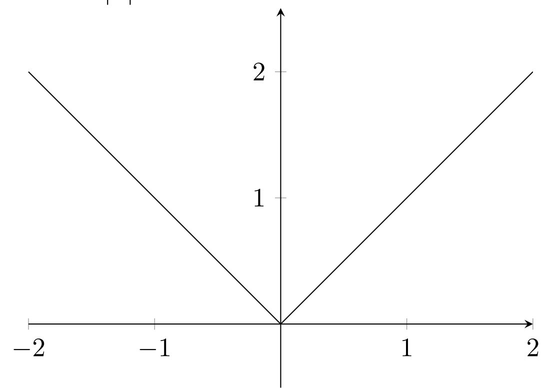
$$y = \max(x - 1, 0) + \frac{1}{2}\max(x, 0).$$

Note that we used a bias term in the linear layer here to add and subtract values.

Not let's start approximating functions.

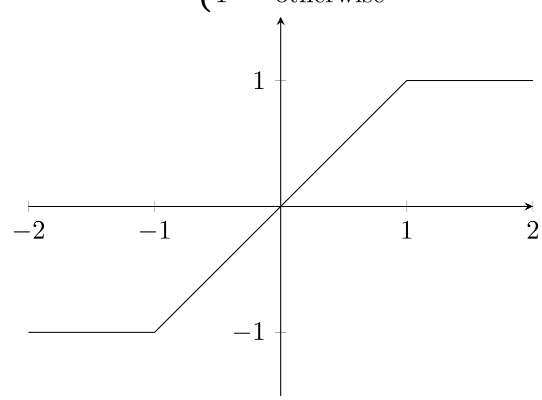
1 abs

Build a two layer ReLU-network that approximates the absolute value $|x|$.



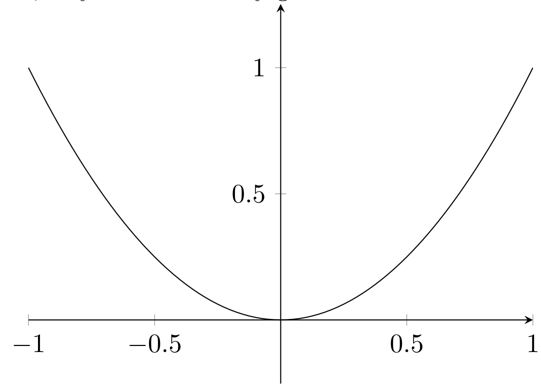
2 soft step

Build a two layer ReLU-network that approximates the soft step function $y = \begin{cases} -1 & \text{if } x < -1 \\ x & \text{if } -1 \le x \le 1 \\ 1 & \text{otherwise} \end{cases}$.



3 Square

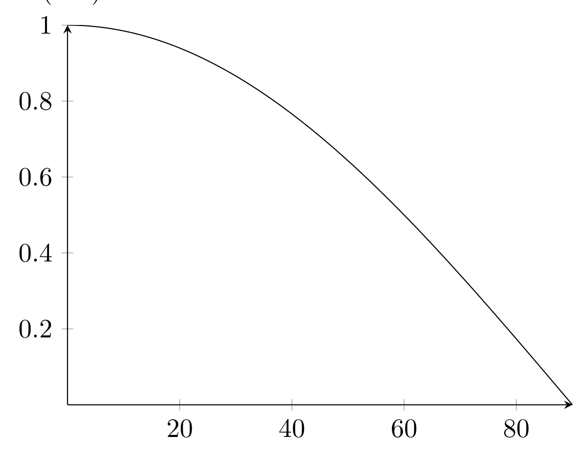
Build a two layer ReLU-network that approximates the soft step function $y = x^2$ in the range $[-1, 1]$ using 8 hidden units. This is the first function you'll have to approximate, as you will unlikely get an exact solution.



Bonus question: Can you build the same network using three layers and 6 *ReLUs?*

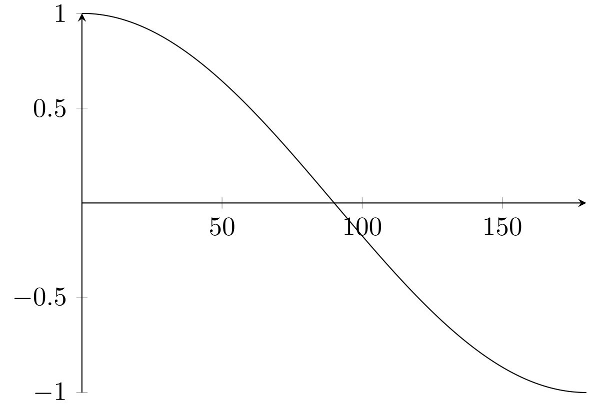
4 Cosine I

Build a two layer ReLU-network that approximates the cosine function $y = \cos(x)$ in the range $[0°, 90°]$ using 3 hidden units. *Use* $\cos(22.5°) \approx 0.92$*,* $\cos(45°) \approx 0.7$*,* $\cos(90°) = 0$ *and a calculator.*



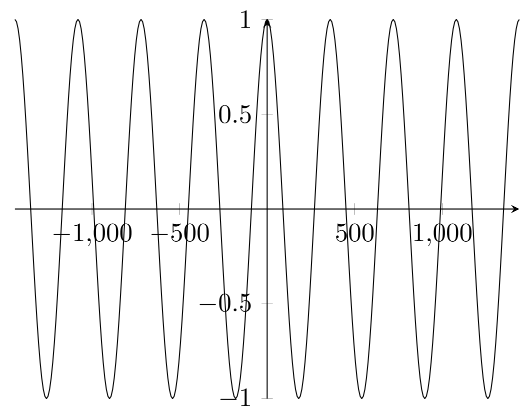
5 Cosine II

Build a two layer ReLU-network that approximates the cosine function $y = \cos(x)$ in the range $[0°, 180°]$ using 5 hidden units. *You may use the above function, no need to copy the network.*



6 Cosine III

Build a multi layer ReLU-network that approximates the cosine function $y = \cos(x)$ in the range $[-1440°, 1440°]$ using at most 5 hidden units per layer. *Hint: You can reuse previous networks directly, no need to copy them here. E.g. abs*(x).



Bonus question: How many hidden units would you need to compute this using a two layer network?