Exercise sheet 2: Backprop and parametric ReLUs

- This exercise requires the code '2_candle.py' from the Moodle page of the course.
- Please submit the solution as a single file, using the naming convention 'your_last_name.py', on the Moodle page of the course.
- Keep in mind to comment your code sufficiently and to include a demonstration.
- You should not import any packages that violate the spirit of the exercise.
- It should suffice to execute your Python script once to verify that the exercise has been solved.

Parametric leaky ReLU activation layers

A layer of parametric rectified linear units $\mathrm{PReLU} : \mathbb{R}^n \to \mathbb{R}^n$ acts, element-wise, as

$$\mathrm{PReLU}(x)_i = \begin{cases} x_i \text{ if } x_i > 0 \\ \alpha_i x_i \text{ otherwise} \end{cases}$$

where $\alpha_i \in \mathbb{R}$. Hence the PReLU-layer is characterized by the parameter-vector $\alpha \in \mathbb{R}^n$.

Parametric ReLU layers where introduced in He et al. (2015): Delving Deep into Rectifiers – Surpassing Human-Level Performance on ImageNet Classification.

1 Implement parametric ReLU layers

Implement parametric leaky ReLU layers in the provided Python module 'candle'.

Hint: Since the PReLU-layer applies one-dimensional functions to its entries, as do other activation layers, its gradients have a simple form.

2 Numerical experimentation

Show that a feedforward network with 20 hidden layers of width 10 and parametric ReLU-activations can learn the XOR function (for example with learning rate 0.01). Use your own implementation.

x_1	x_2	$\mathrm{XOR}(x_1, x_2)$
0	0	0
0	1	1
1	0	1
1	1	0