

## 1 Gradient descent in TF

Let a vector-scalar function  $f$  be defined as  $f(\vec{x}) = \sum_i (x_i - t_i)^2$ , with  $\vec{t}$  being an arbitrary vector of target values.

a) What loss function does this resemble?

b) For  $\vec{t} = (1, 2, 3, 4)^T$ : write a Python program that performs gradient descent on  $f$  **using TensorFlow** from a starting point of  $\vec{x}(0) = (-1, -1, -1, -3)$ . Choose a suitable step size and perform 10,100,1000 iterations and print  $\vec{x}$  and  $f$ . What do you observe?

c) For  $\vec{t} = (1, 2, 3, 4)^T$ : write a Python program that performs gradient descent on  $f$  **using numpy** from a starting point of  $\vec{x}(0) = (-1, -1, -1, -3)$ . In this case, you need to implement the gradient computation manually: compute it on paper first!

(d) If you set the step size rather high: try to observe how GD oscillates around the minimum of  $f$ !