

# Week 2 Optimisation for Machine Learning

Neimhin Robinson Gunning, 16321701

March 11, 2024

Let

$$f(x, y) = 3(x - 5)^4 + 10(y - 9)^2 \quad (1)$$

and

$$g(x, y) = \max(x - 5, 0) + 10|y - 9| \quad (2)$$

Using `sympy` we find the derivatives:

$$\nabla f = \left[ \frac{df}{dx}, \frac{df}{dy} \right] = [12(x - 5)^3, 20y - 180]$$

$$\nabla g = \left[ \frac{dg}{dx}, \frac{dg}{dy} \right] = [\text{Heaviside}(x - 5), 10\text{sign}(y - 9)]$$

Clearly, the minimum of both  $f(x, y)$  is 0 and they is minimized by  $x = 5, y = 9$ . The other function  $g(x, y)$  also has minimum 0 but is minized by any of  $x \in [-\infty, 5]$  and  $y = 9$ .

The Polyak step size is

$$\alpha_{\text{Polyak}} = \frac{f(x) - f^*}{\nabla f(x)^T \nabla f(x)} \quad (3)$$

where  $x$  is the parameter vector,  $f(x)$  is the function to optimise, and  $f^* \approx \min_x f(x)$ .

```
funcs.txt           Wed Feb 21 15:03:56 2024           1  
function: 3*(x-5)^4+10*(y-9)^2  
function: Max(x-5, 0)+10*|y-9|
```

Figure 1: Two bivariate functions downloaded from <https://www.scss.tcd.ie/Doug.Leith/CS7DS2/week4.php>

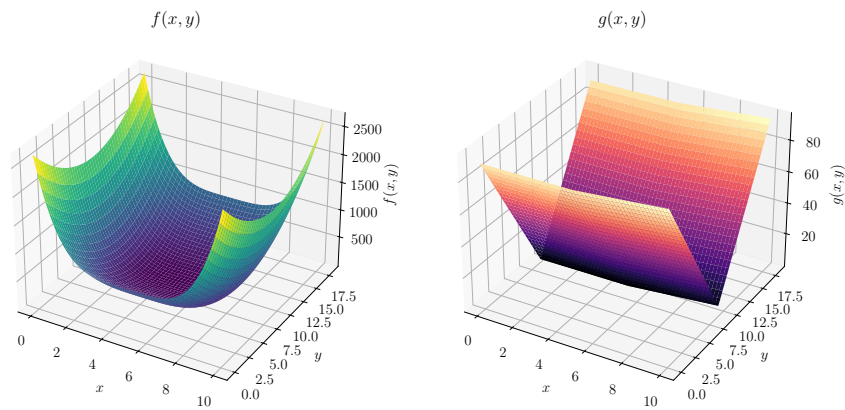


Figure 2

Listing 1: A python function to calculate the Polyak step size on a `sympy` function.

```
src/polyak_step_size.py      Wed Feb 14 15:12:30 2024      1
1: import numpy as np
2:
3:
4: def polyak_step_size(self, sp_func, sp_x, x, f_star):
5:     assert len(sp_x) == len(x)
6:     subs = {sp_xi: xi for sp_xi, xi in zip(sp_x, x)}
7:     fx = sp_func.subs(subs)
8:     grad = [sp_func.diff(sp_xi).subs(subs) for sp_xi in sp_x]
9:     grad = np.array(grad)
10:    denominator = sum(grad * grad)
11:    numerator = fx - f_star
12:    return numerator / denominator
```