Week 2 Optimisation for Machine Learning

Neimhin Robinson Gunning, 16321701

March 11, 2024

```
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`

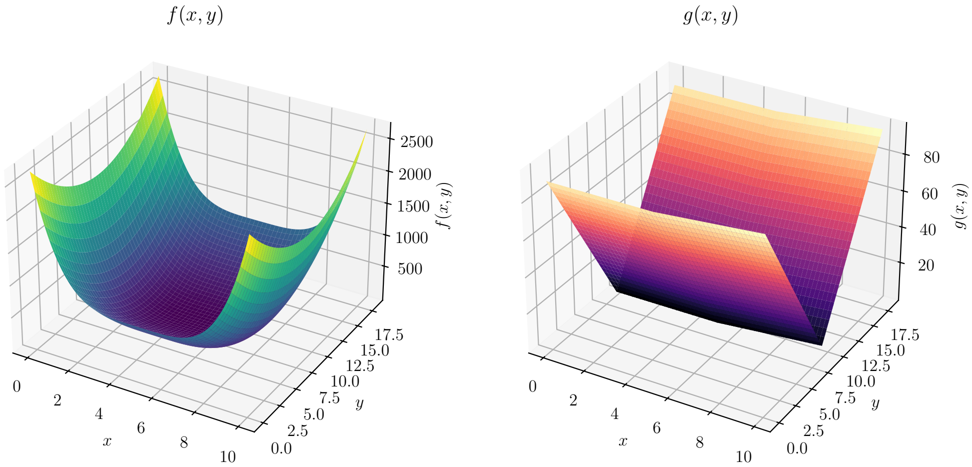

Figure 2

Let

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

and

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

Using `sympy` we find the derivatives:

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

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

Clearly, the minimum of both $f(x,y)$ and $g(x,y)$ is 0 and they are both minimized by $x = 5$, $y = 9$.

The Polyak step size is

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

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

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

```python
import numpy as np


def polyak_step_size(self, sp_func, sp_x, x, f_star):
    assert len(sp_x) == len(x)
    subs = {sp_xi: xi for sp_xi, xi in zip(sp_x, x)}
    fx = sp_func.subs(subs)
    grad = [sp_func.diff(sp_xi).subs(subs) for sp_xi in sp_x]
    grad = np.array(grad)
    denominator = sum(grad * grad)
    numerator = fx - f_star
    return numerator / denominator
```