PadhAI: Variants of Gradient Descent

One Fourth Labs

Running and visualising nesterov accelerated gradient descent

Let's execute the code for this

1. Here is the Python code for NAG, it is an improvement on the MGD

```
X = [0.5, 2.5]
Y = [0.2, 0.9]

def f(w, b, x):
    # sigmoid with parameters w, b
    return 1.0/ (1.0 + np.exp(-(w*x + b)))

def error(w, b):
    err = 0.0
    for x, y in zip(X, Y):
        fx = f(w, b, x)
        err += 0.5 * (fx - y) ** 2
    return err

def grad_b(w, b, x, y):
    fx = f(w, b, x)
    return (fx-y) * fx * (1 - fx)

def grad_w(w, b, x, y):
    fx = f(w, b, x)
    return (fx-y) * fx * (1 - fx) * x

def do_nag_gradient_descent():
    w, b, eta, max_epochs = -2, -2, 1.0, 1000
    v_w, v_b = 0, 0
    gamma = 0.7
    for i in range(max_epochs):
        dw, db = 0, 0

        #Compute the lookahead value
        w = w - gamma*v_w   # this is w_temp
        b = b - gamma*v_b   # this is b_temp

        for x, y in zip(X, Y):
            #Compute the derivatives using the lookahead value
            dw += grad_w(w, b, x, y)
            db += grad_b(w, b, x, y)

        #Now move further in the direction of that gradient
        w = w - eta*dw
        b = b - eta*db

        v_w = gamma * v_w + eta * dw
        v_b = gamma * v_b + eta * db
```

2. Here we have a comparison between NAG and MGD