Variation in Gradient Descent

Momentum Gradient Descent :-

Gradient Descent with Momentum considers the past gradients to smooth out the update. It computes an exponentially weighted average of your gradients, and then use that gradient to update your weights instead. It works faster than the standard gradient descent algorithm.

Code :-

```
def do_momentum_gradient_descent():
    w,b,eta,max_epochs = -2, -2, 1.0,10
    prev_v_w,prev_v_b,gamma=0,0,0.9
    for i in range(max_epochs):
        dw,db=0,0
        for x,y in zip(X,Y):
            dw+=grad_w(w,b,x,y)
            db+=grad_b(w,b,x,y)
        v_w=gamma*prev_v_w+eta*dw
        v_b=gamma*prev_v_b+eta*db
        w=w-v_w
        b=b-v_b
        prev_v_w=v_w
        prev_v_b=v_b
        print(error(w,b))
```

Nesterov Accelerated Gradient Descent :-

```
def do_nestrove_accelerated_gradient_descent():
    w,b,eta,max_epochs = -2, -2, 1.0,10
    prev_v_w,prev_v_b,gamma=0,0,0.9
    for i in range(max_epochs):
        dw,db=0,0
        for x,y in zip(X,Y):
            dw+=grad_w(w,b,x,y)
            db+=grad_b(w,b,x,y)
        v_w=gamma*prev_v_w+eta*dw
        v_b=gamma*prev_v_b+eta*db
        w=w-v_w
        b=b-v_b
        prev_v_w=v_w
        prev_v_b=v_b
      print(error(w,b))
```