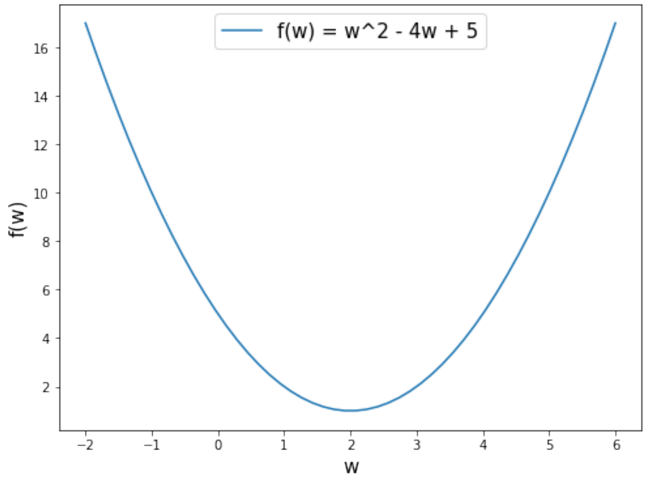
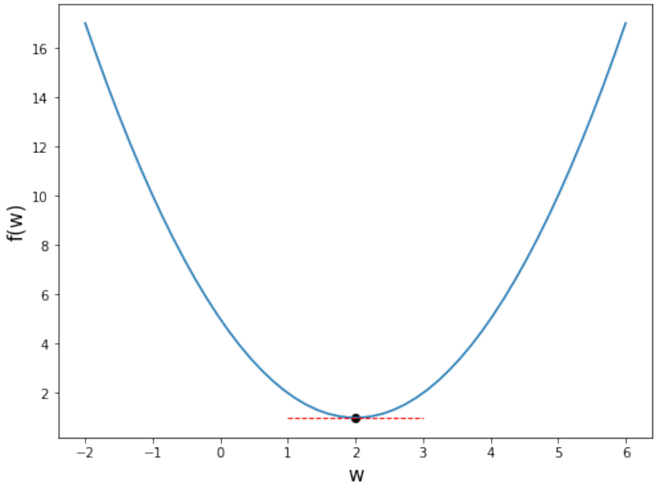
4. GRADIENT DESCENT

Gradent descent is a method of finding a minimum of some function.

Consider the following function $f(w) = w^2 - 4w + 5$. From its graph, we see that the function has the minimum at $w = 2$. How would we find this value if we did not have the graph and we only had the formula of this function?



We see that the minimum is at the bottom of the "valley" in the graph. Also observe that if a function has a mimimum at some point, then the tangent line at that point must be flat. In other words, the slope of that tangent line is 0 or, in other words, the derivative must be zero.



That means that to find the value where f has the minimum, we must find the points where the derivative is zero. Using calculus, we find the derivative to be

$$f'(w) = 2 * w - 4$$

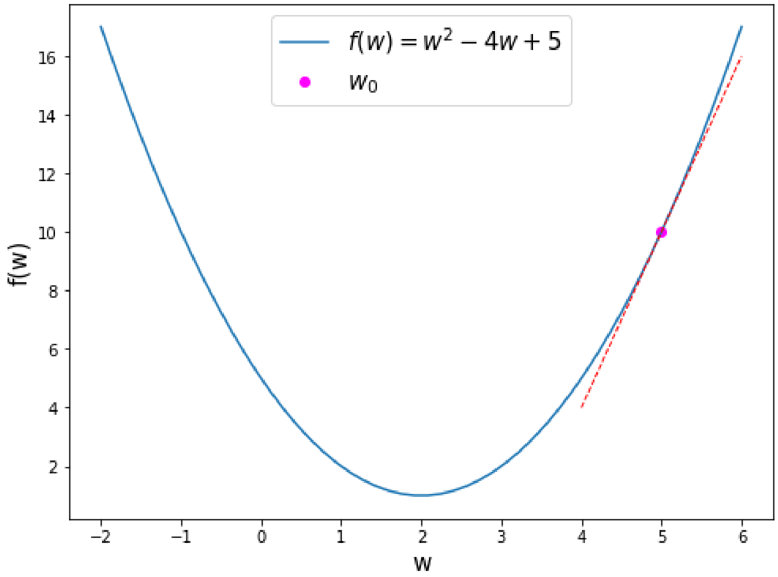
Sometimes we can see what value of *w* would give the value of 0 for the derivative. Here, it is easy to see that *w=2* makes the derivative zero, but sometimes the equation

$$f'(w) = 0$$

is very complicated and it is impossible to solve. The method that we could use to find the minimum in this case is the **gradient descent**.

Note: The values of w where $f'(w) = 0$ are called critical points. To determine that $f(w)$ has minimum at $f(w)$, we need to confirm that the second derivative $f''(w)$ is postive. However, since the function $f(w) = w^2 - 4w + 5$ is convex, this must be a minimum. Most of the functions in ML that we will be minimizing will be convex and the zeroes of the derivative (or the gradient, in multidimensions) will be points where the function has minumum.

We start with a random initial guess, such as $w_0 = 5$ and we compute the derivative. The derivative is $f'(5) = 2 * 5 - 4 = 6$ implying that the tangent line has slope 6. Since the tangent line is not flat, this point is not where *f* has a minimum. We can also visualize this.



The derivative tells us whether the graph is going up or down. Since $f'(5) = 6$ is a positive number, the graph is going up and to find the minimum, we should move to the left of 5. In other words, we need to move in the opposite direction of the sign of the derivative. We should not move too much to the left and we control by how much we will move using the learning rate α. We define the new point by

$$w_1 = w_0 - \alpha * f'(w_0)$$

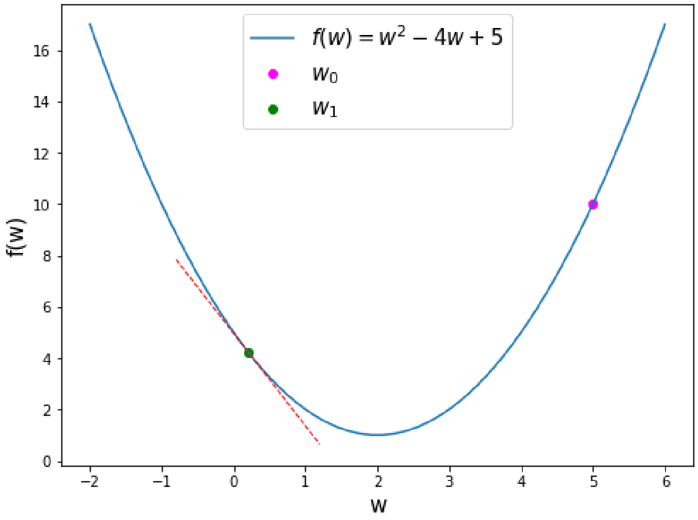
Let's set α = 0.8. Then we get

$$w_1 = 5 - 0.8 * 6 = 0.2$$

To see if w_1 is a point where *f* achieves its minimum, we need to find $f'(w_1)$ and check if it is 0. We calculate

$$f'(0.2) = 2 * 0.2 - 4 = -3.6$$

Since this derivative is not zero, 0.2 is not where *f* has a minimum and we are not done. Since the derivative is -3.6 we know that both the graph and the tangent line are going down. The next figure shows it as well.

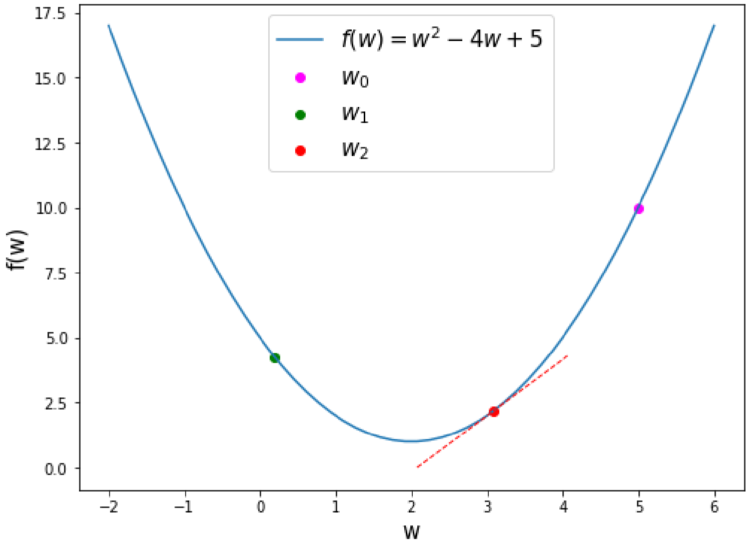


To get to the minimum ("the bottom of the valley"), we should move now to the right, which again is in the opposite direction of the derivative. We define the new point by

$$w_2 = w_1 - \alpha * f'(w_1)$$

$$w_2 = 0.2 - 0.8 * (-3.6) = 3.08$$

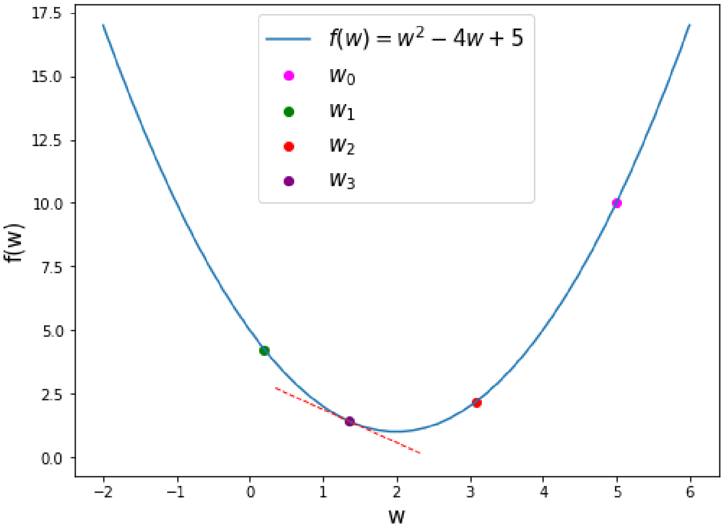
Now, we find the derivative at 3.08 to see if it is 0 or if we need to find the next point w_3.

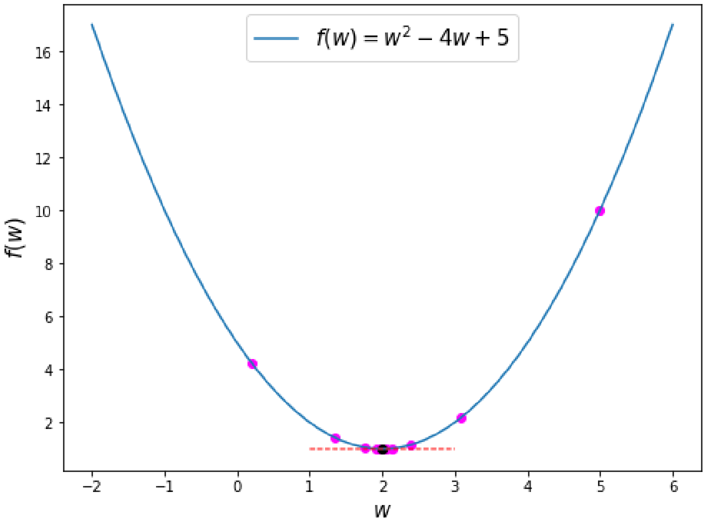


Observe that the general formula for updating the point is

$$w_{n+1} = w_n - \alpha * f'(w_n)$$

and that by this method we are moving closer and closer to the "bottom of the valley" which is where *f* has a minimum.





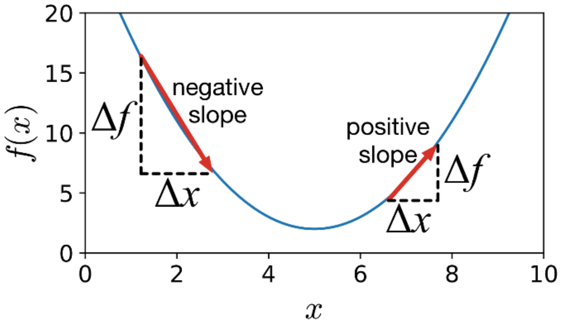
We stop once the derivative becomes very small (practically 0).

Remark:

1) Note again that the formula

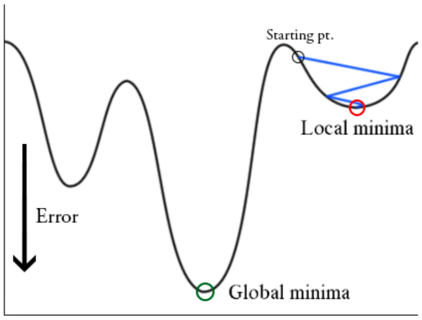
$$w_{n+1} = w_n - \alpha * f'(w_n)$$

implies we are moving towards the minimum ("bottom of the valley").



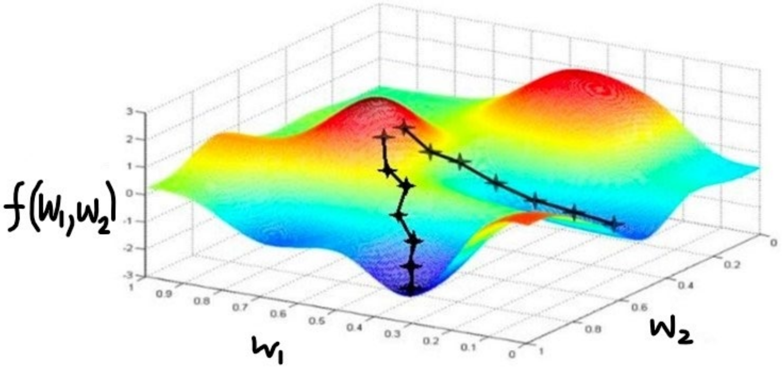
https://explained.ai/gradient-boosting/descent.html

2) Sometimes the gradient descent will lead us only to a local minimum.



https://www.researchgate.net/figure/Gradient-Descent-Stuck-at-Local-Minima-18_fig4_338621083

3) If we have a function of several variables, the idea of the gradient descent still applies. We start from a random point and move towards the "bottom of the valley".



https://www.analyticsvidhya.com/blog/2017/03/introduction-to-gradient-descent-algorithm-along-its-variants/

Assume we are given a function $f(w_1, w_2, \ldots, w_d)$. To find its minimum, we use the gradient descent.

First, we need to find derivative of f with respect to each of the variables $w_1, w_2, \ldots, w_d$. These derivatives are combined in a vector that is called the **gradient of f**:

$$\nabla f(w_1, \ldots, w_d) = \left(\frac{\partial f}{\partial w_1}, \ldots, \frac{\partial L}{\partial w_d} \right)$$

The method of gradient descent is as follows:

1. We start with the random choice of initial values for $w_1, w_2, \ldots, w_d$.
2. We make the updates using the following formulas for each variable

$$w_1(new) = w_1(old) - \alpha \cdot \frac{\partial f}{\partial w_1}(old)$$
$$\cdots\cdots\cdots$$
$$w_d(new) = w_d(old) - \alpha \cdot \frac{\partial f}{\partial w_d}(old)$$

If we use vector notation $W = (w_1, w_2, \ldots, w_d)$, the formula for the updates can be written as

$$W_{n+1} = W_n - \alpha \cdot \nabla f(W_n)$$

3. We repeat step 2 until the gradient of f becomes close to zero.

Python code: Lecture_4_Gradient_Descent.ipynb

References and Reading Material:

[1] MIT notes, Chapter 6, Sections 1-2
https://openlearninglibrary.mit.edu/assets/courseware/v1/d81d9ec0bd142738b069ce601382fdb7/asset-v1:MITx+6.036+1T2019+type@asset+block/notes_chapter_Gradient_Descent.pdf

[2] For those interested more in optimization, see

J. Nocedal, S. J. Wright, Numerical Optimization, 2nd edition, Springer (2006).

http://egrcc.github.io/docs/math/numerical-optimization.pdf