CSCE 633: Machine Learning

Lecture 10: Gradient Descent

Texas A&M University
Bobak Mortazavi

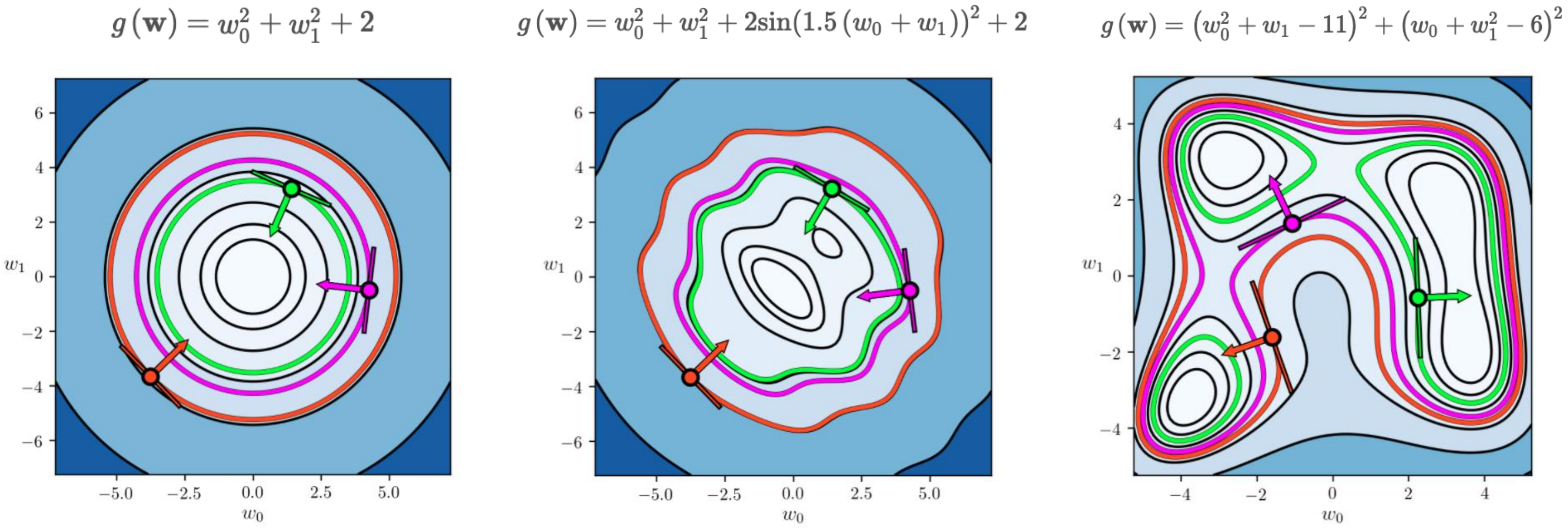
Goals for this Lecture

- First Order Optimization – The Gradient Function
- Understand Gradient Descent
- Understanding Limitations to Gradient Descent
- Second Order Optimization – Convexity/Concavity
- Newton's Method for Descent

Two Natural Weaknesses of Gradient Descent

Problem 1: The 'zig-zagging' behavior of gradient descent

- The (negative) gradient direction points perpendicular to the contours of any function



Two Natural Weaknesses of Gradient Descent

- The negative gradient direction oscillate rapidly or zig-zag
- Example
  - Functions: three N=2N=2 dimensional quadratic

$$g(\mathbf{w}) = a + \mathbf{b}^T\mathbf{w} + \mathbf{w}^T\mathbf{C}\mathbf{w}$$

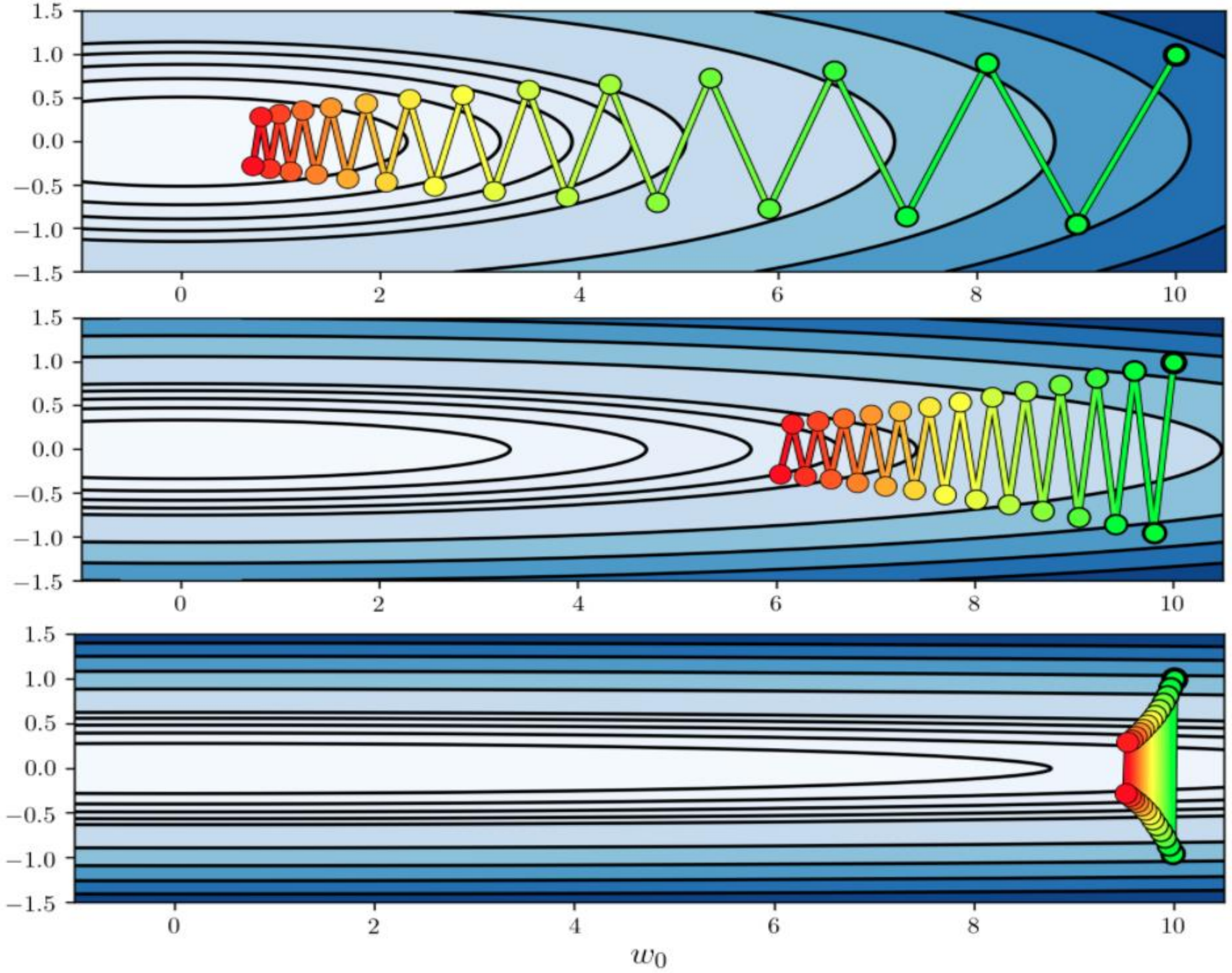
    - The first quadratic: $\mathbf{C} = \begin{bmatrix} 0.5 & 0 \\ 0 & 12 \end{bmatrix}$
    - The second quadratic: $\mathbf{C} = \begin{bmatrix} 0.1 & 0 \\ 0 & 12 \end{bmatrix}$
    - The third quadratic: $\mathbf{C} = \begin{bmatrix} 0.01 & 0 \\ 0 & 12 \end{bmatrix}$
  - Same global minimum: $\mathbf{w} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$ where $g(\mathbf{w}) = 0$
  - Initialization: $\mathbf{w}^{\mathbf{0}} = \begin{bmatrix} 10 \\ 1 \end{bmatrix}$
  - Steplength / learning rate value: $\alpha = 0.1$

Two Natural Weaknesses of Gradient Descent

The first quadratic

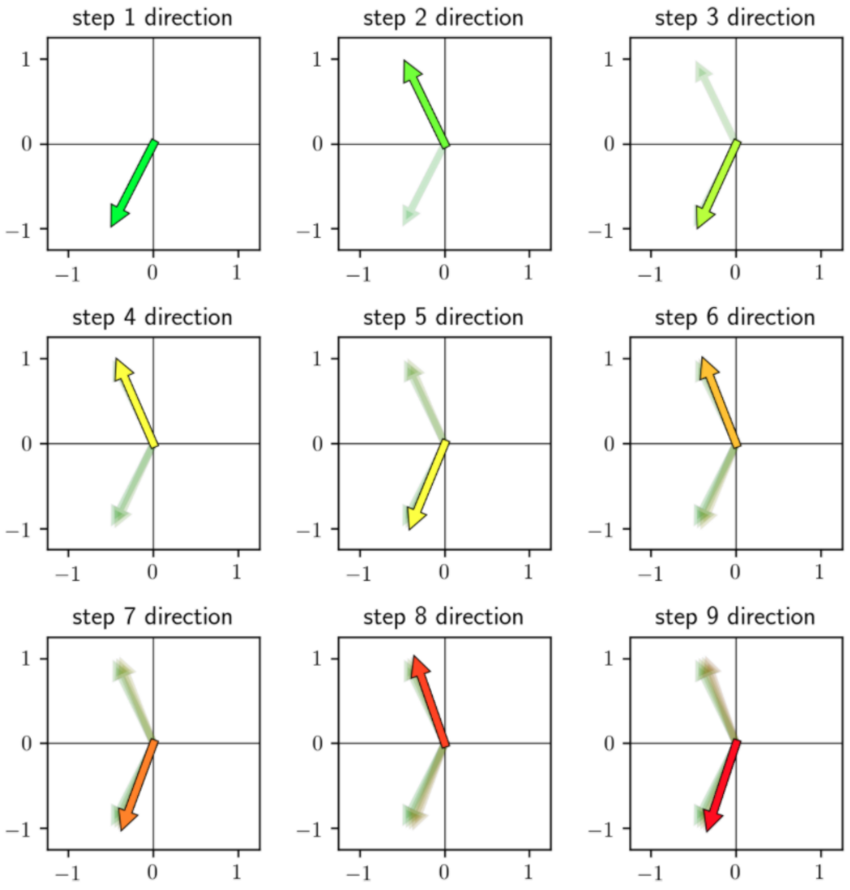
The second quadratic

The third quadratic



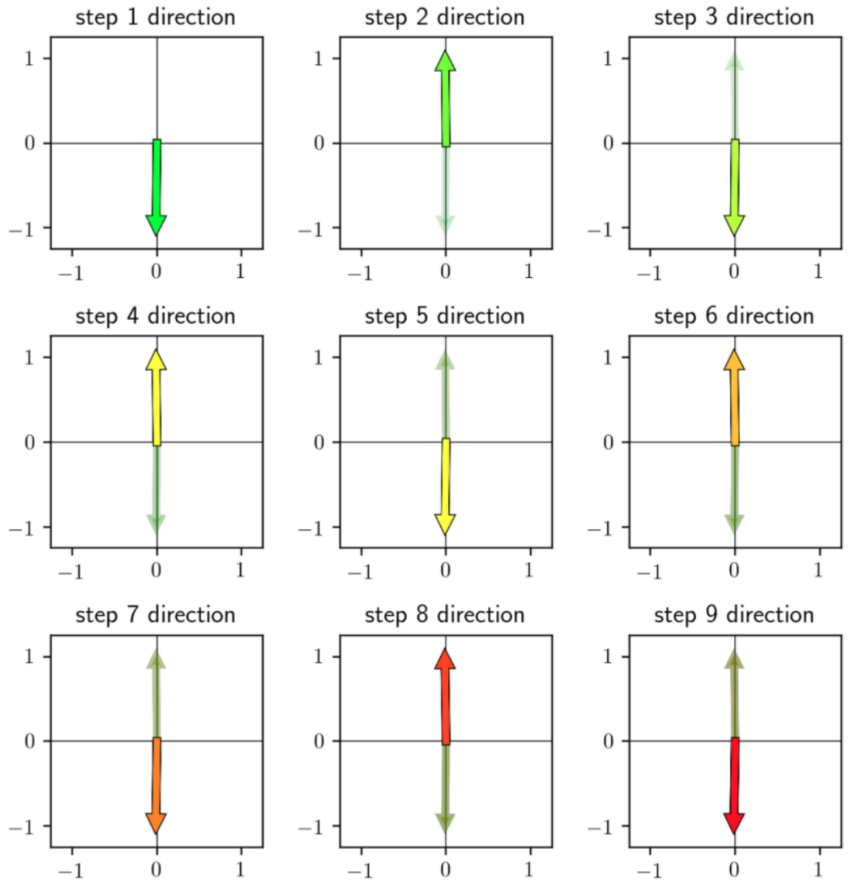
Two Natural Weaknesses of Gradient Descent

- Descent direction on the **first** quadratic



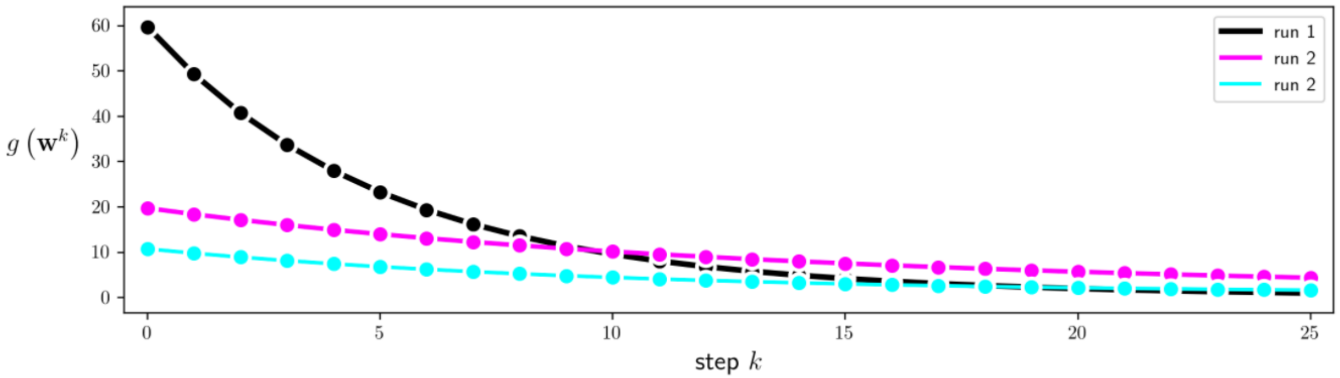
Two Natural Weaknesses of Gradient Descent

- Descent direction on the **third** quadratic

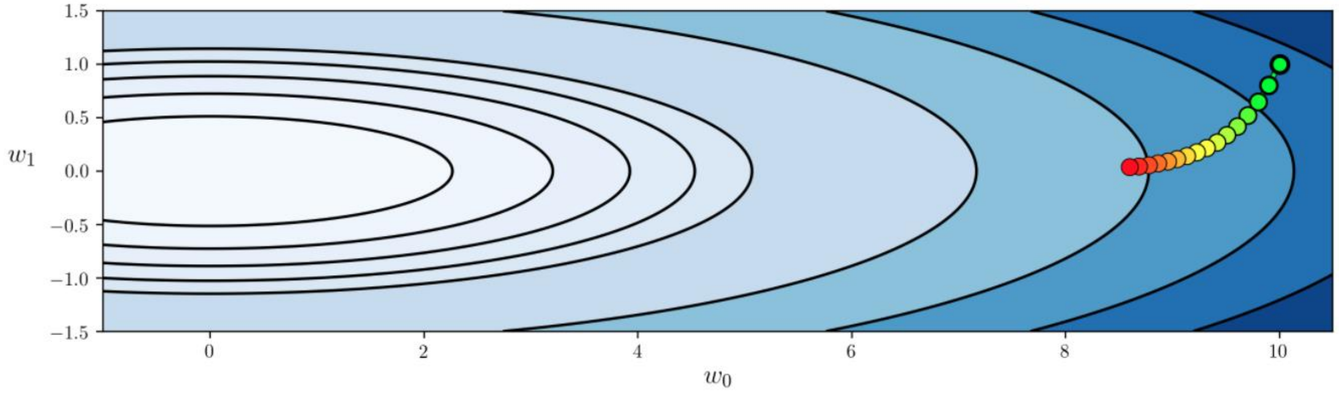


Two Natural Weaknesses of Gradient Descent

- Cost function plot



- Reducing the steplength value can ameliorate this zig-zagging behavior.
- Do not solve the underlying problem that zig-zagging produces - slow convergence



Two Natural Weaknesses of Gradient Descent

Problem 2: The slow-crawling behavior of gradient descent

- The vanishing behavior of the negative gradient magnitude near stationary points has a natural consequence for gradient descent steps - they progress very slowly, or 'crawl', near stationary points.
- Unlike zero order methods, the distance traveled during each step of gradient descent is not completely determined by the steplength/learning rate value α.

Two Natural Weaknesses of Gradient Descent

- The general local optimization step:

$$\mathbf{w}^k = \mathbf{w}^{k-1} + \alpha \mathbf{d}^{k-1}$$

- Zero order: $\mathbf{d}^{k-1}$ is a unit length descent direction

$$\left\|\mathbf{w}^k - \mathbf{w}^{k-1}\right\|_2 = \left\|\left(\mathbf{w}^{k-1} + \alpha \mathbf{d}^{k-1}\right) - \mathbf{w}^{k-1}\right\|_2 = \alpha \left\|\mathbf{d}^{k-1}\right\|_2 = \alpha$$

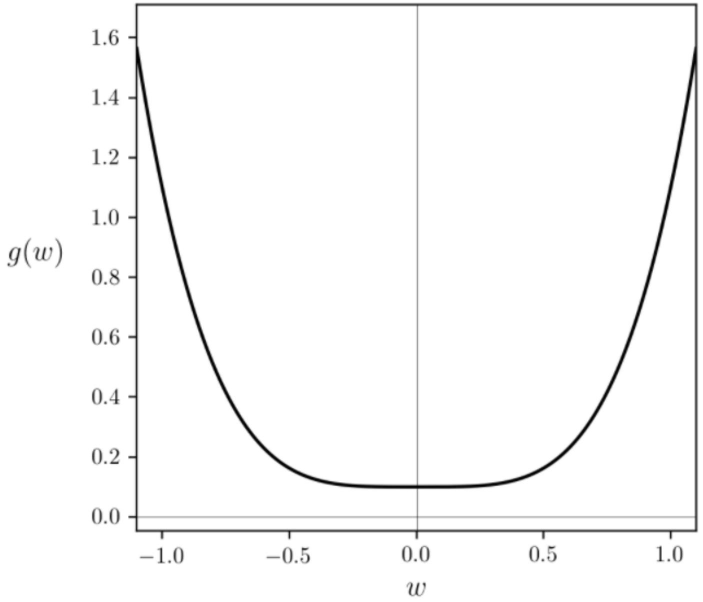
- Gradient descent: $\mathbf{d}^{k-1} = -\nabla g(\mathbf{w}^{k-1})$

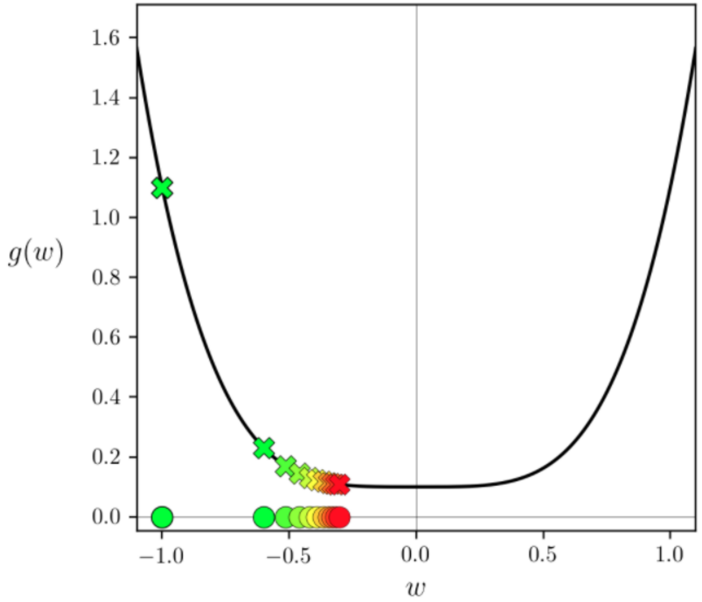
$$\left\|\mathbf{w}^k - \mathbf{w}^{k-1}\right\|_2 = \left\|\left(\mathbf{w}^{k-1} - \alpha \nabla g\left(\mathbf{w}^{k-1}\right)\right) - \mathbf{w}^{k-1}\right\|_2 = \alpha \left\|\nabla g\left(\mathbf{w}^{k-1}\right)\right\|_2$$

Two Natural Weaknesses of Gradient Descent

- Example 1: Slow-crawling behavior of gradient descent near the minimum of a function
  - Function:
  - Minimum: $w = 0$
  - Steplength: $\alpha = 0.1$

$$g(w) = w^4 + 0.1$$



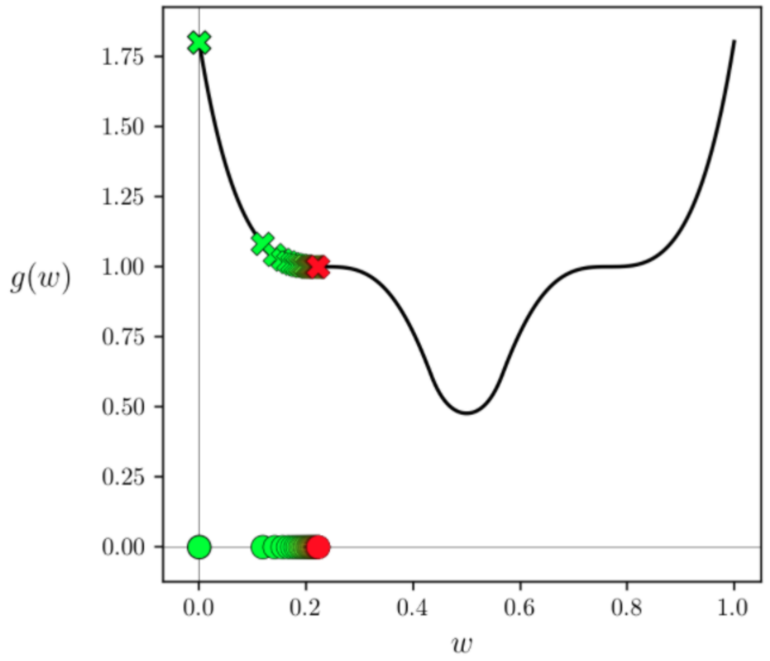


Two Natural Weaknesses of Gradient Descent

- Example 2: Slow-crawling behavior of gradient descent near saddle points
  - Function:

$$g(w) = \text{maximum}(0, (3w - 2.3)^3 + 1)^2 + \text{maximum}(0, (-3w + 0.7)^3 + 1)^2$$

  - Minimum: $w = \frac{1}{2}$
  - Saddle points:
    - $w = \frac{7}{30}$
    - $w = \frac{23}{30}$
  - Gradient descent: 50 steps
  - Steplength: $\alpha = 0.01$
  - Initialization: $w = 0$

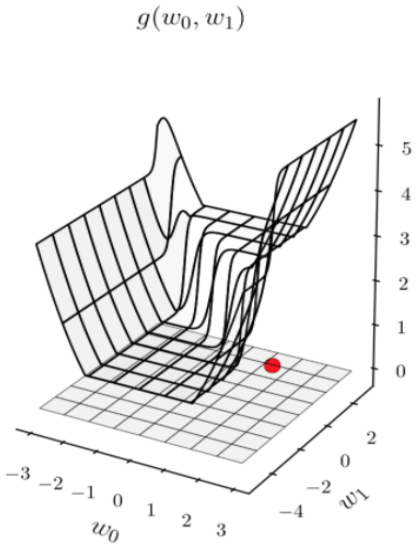


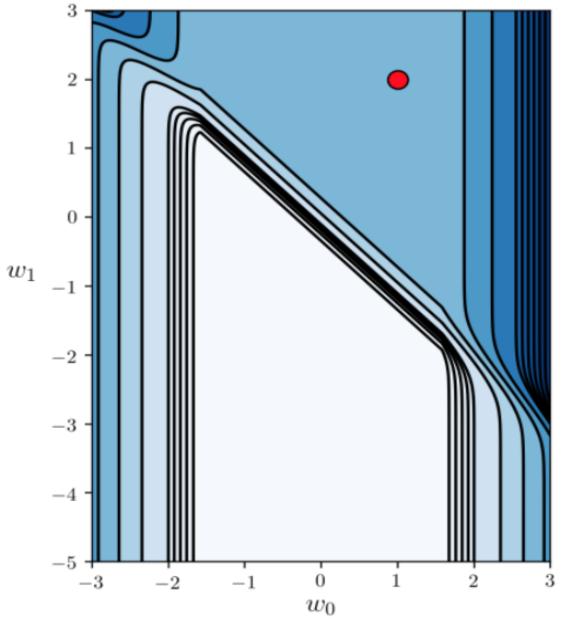
Two Natural Weaknesses of Gradient Descent

- Example 3: Slow-crawling behavior of gradient descent in large flat regions of a function
  - Function:
  - Initialization: $w^0 = \begin{bmatrix} 0 \\ 0 \end{bmatrix}$

$$g(w_0, w_1) = \tanh(4w_0 + 4w_1) + \max(1, 0.4w_0^2) + 1$$

  - 1000 steps of gradient descent with a steplength $\alpha = 0.1$





Takeaways

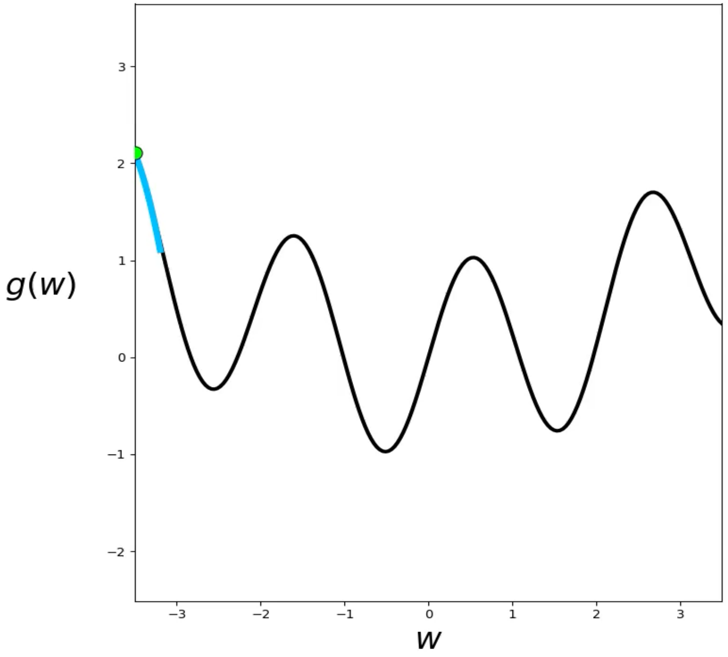
- Limitations of Gradient Descent as a result of Step size
- Understanding what happens when Stepsize is too big
- Understanding what happens when Stepsize is too small

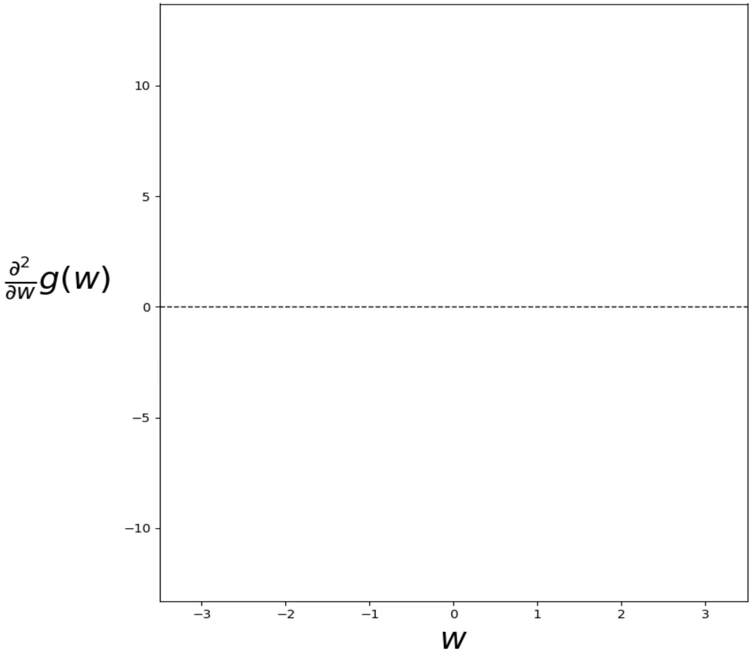
The Second-Order Optimality Condition

Curvature and single-input functions

- The second order Taylor series approximation of function

$$g(w) = \sin(3w) + 0.1w^2$$





The Second-Order Optimality Condition

- The second order approximation appears to match the local convexity/concavity of the underlying function near the point on which it is defined.
  - If at this point the function appears to be convex locally, the second order approximation is too convex and upward facing.
  - If the point is on a part of the function where it is facing downward or concave, the second order approximation is also concave and facing downward.
- The second order Taylor Series is a quadratic built to match a function locally.

The Second-Order Optimality Condition

- Quadratic functions are easy to determine convex or concave
- A general single input quadratic

$$g(w) = a + bw + cw^2$$

  - $c > 0$: convex
  - $c < 0$: concave
  - $c = 0$: both convex and concave (a line)
- The second order Taylor Series $h(w)$ of a single input function $g(w)$ at a point w_0 is:

$$h(w) = g(w^0) + \left(\frac{\mathrm{d}}{\mathrm{d}w} g(w^0)\right)(w - w^0) + \frac{1}{2}\left(\frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^0)\right)(w - w^0)^2$$

The Second-Order Optimality Condition

$$c = \frac{1}{2} \frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^0)$$

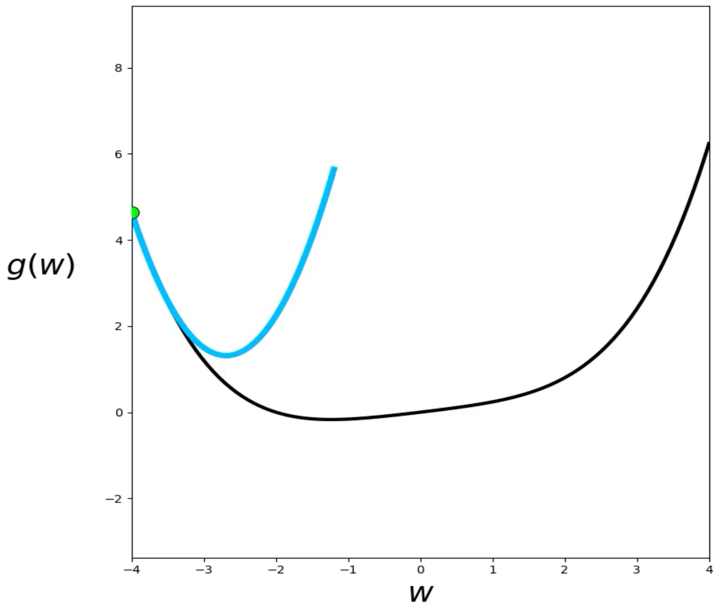
– $\frac{d^2}{dw^2} g(w^0) \geq 0$: convex at w^0
– $\frac{d^2}{dw^2} g(w^0) \leq 0$: concave at w^0

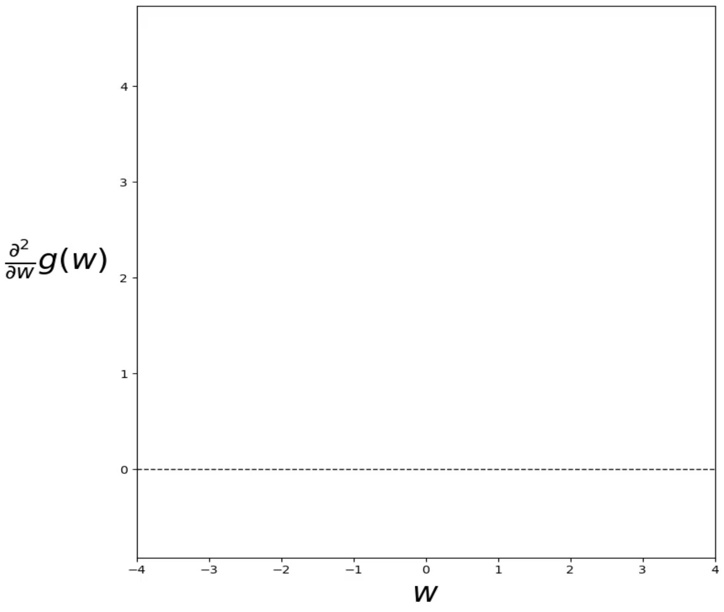
- A function g is convex if it is convex at each of its input points. ($\frac{d^2}{dw^2} g(w^0) \geq 0$ everywhere)
- A function g is concave if it is convex at each of its input points. ($\frac{d^2}{dw^2} g(w^0) \leq 0$ everywhere)

The Second-Order Optimality Condition

- Example: single-input plot
  - Function:

$$g(w) = \frac{1}{50}\left(w^4 + w^2 + 10w\right)$$





The Second-Order Optimality Condition

Curvature and multi-input functions

- The general multi-input quadratic function

$$g(\mathbf{w}) = a + \mathbf{b}^T\mathbf{w} \ + \mathbf{w}^T\mathbf{C}\,\mathbf{w}$$

- The convexity/concavity is determined by the eigenvalues of $\mathbf{C}$
  - The quadratic is convex along its n^{th} input iif its n^{th} eigenvalue $d_n \geq 0$
  - The quadratic is convex along its n^{th} input iif its n^{th} eigenvalue $d_n \leq 0$

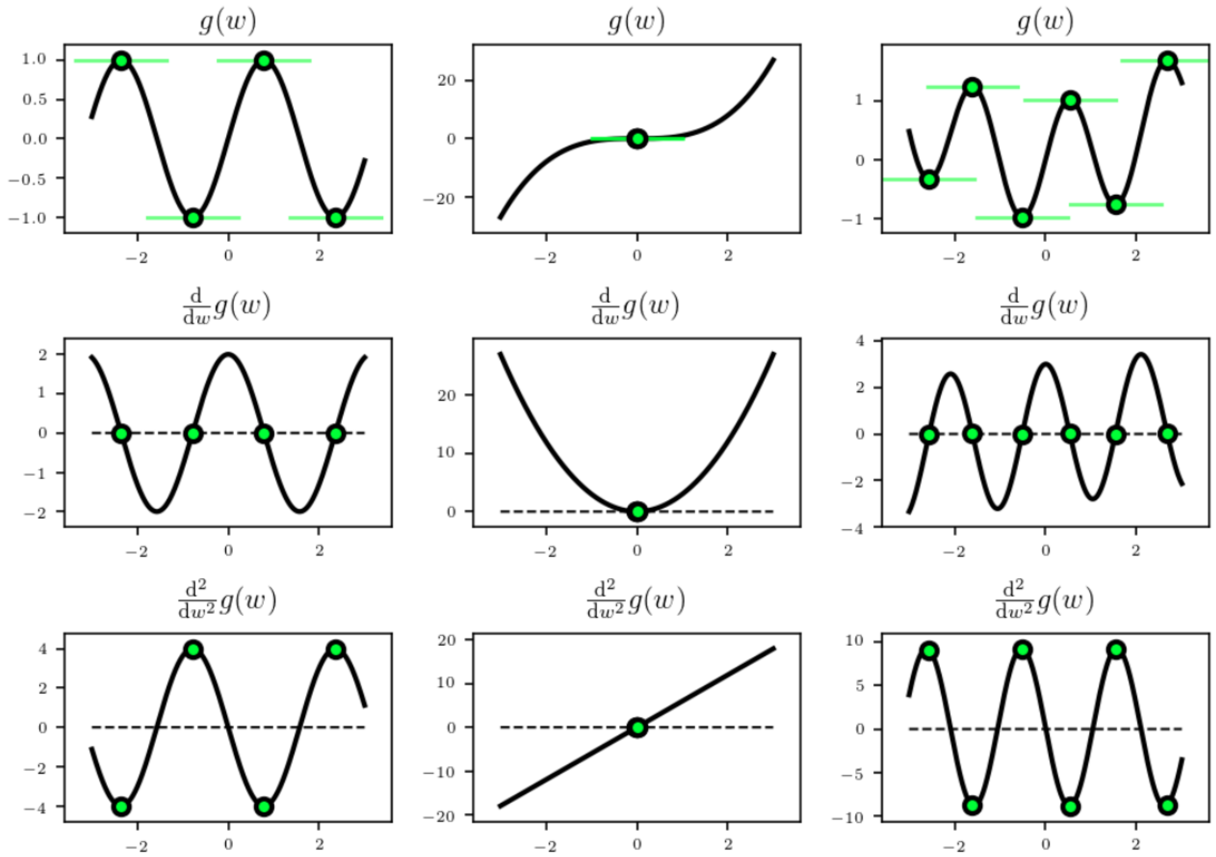
The Second-Order Optimality Condition

- Convexity/concavity at $\mathbf{w}^0$
  - g is convex at $\mathbf{w}^0$ iif the second order Taylor Series approximation is convex in every one of its input dimension, i.e., $\nabla^2 g(\mathbf{w}^0)$ has no negative eigenvalues.
  - g is concave at $\mathbf{w}^0$ iif the second order Taylor Series approximation is concave in every one of its input dimension, i.e., $\nabla^2 g(\mathbf{w}^0)$ has no positive eigenvalues.
- Convex/concave function
  - g is a *convex function* if it is convex everywhere, or if $\nabla^2 g(\mathbf{w}^0)$ has all nonnegative eigenvalues at every input.
  - g is a *concave function* if it is concave everywhere, or if $\nabla^2 g(\mathbf{w}^0)$ has all non-positive eigenvalues at every input.

The Second-Order Optimality Condition

The second order condition

- Comparison of zero, first, second order



The Second-Order Optimality Condition

- Single-input functions
  - Local/global minimum: $\frac{\partial^2}{\partial w^2} g(w) > 0$
  - Local/global maximum: $\frac{\partial^2}{\partial w^2} g(w) < 0$
  - A saddle point: $\frac{\partial^2}{\partial w^2} g(w) = 0$ and $\frac{\partial^2}{\partial w^2} g(w)$ changes sign at w.

- Multi-input functions
  - Local minimum: all eigenvalues of $\nabla^2 g(\mathbf{w}^0)$ are positive
  - Local maximum: all eigenvalues of $\nabla^2 g(\mathbf{w}^0)$ are negative
  - A saddle points: all eigenvalues of $\nabla^2 g(\mathbf{w}^0)$ are mixed (have both positive and negative.

Takeaways

- Properties of Convex/Concave functions
- How to identify a function as Convex/Concave

Newton's Method

- Newton’s method: a local optimization algorithm produced by repeatedly taking steps that are stationary points of the second order Taylor series approximations to a function.

- Method:
  - At k^{th} step move to the stationary point of the quadratic approximation generated at the previous step $\mathbf{w}^{k-1}$:
  - A stationary p

$$h(\mathbf{w}) = g(\mathbf{w}^{k-1}) + \nabla g(\mathbf{w}^{k-1})^T(\mathbf{w} - \mathbf{w}^{k-1}) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^{k-1})^T \nabla^2 g\left(\mathbf{w}^{k-1}\right)\left(\mathbf{w} - \mathbf{w}^{k-1}\right)$$

$$\mathbf{w}^k = \mathbf{w}^{k-1} - \left(\nabla^2 g(\mathbf{w}^{k-1})\right)^{-1} \nabla g(\mathbf{w}^{k-1})$$

Newton's Method

- For single input functions:

$$w^k = w^{k-1} - \frac{\frac{\mathrm{d}}{\mathrm{d}w} g(w^{k-1})}{\frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^{k-1})}$$

- This local optimization fits:

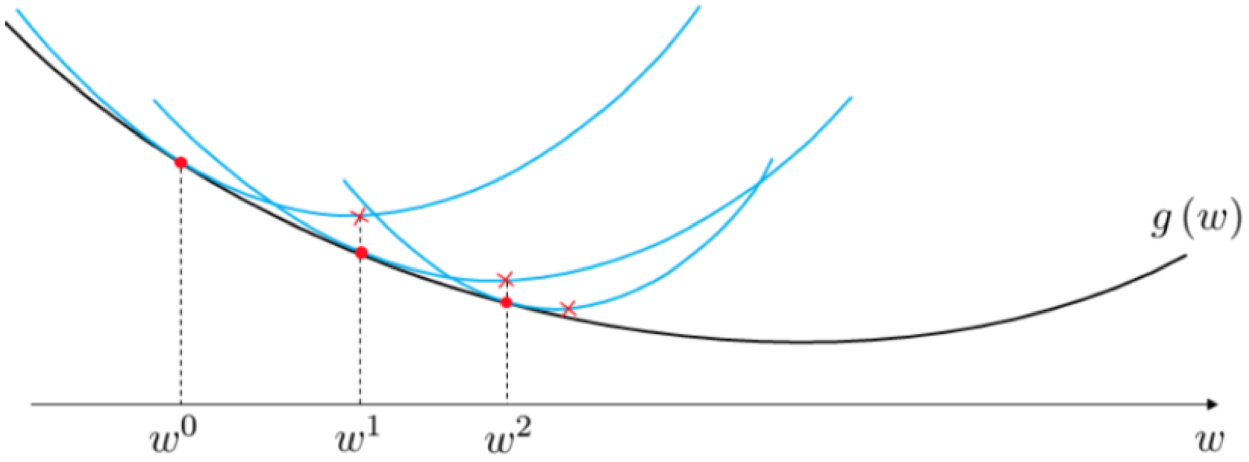
$$\mathbf{w}^k = \mathbf{w}^{k-1} + \alpha \mathbf{d}^k$$

where

$$\mathbf{d}^k = -\left(\nabla^2 g(\mathbf{w}^{k-1})\right)^{-1} \nabla g(\mathbf{w}^{k-1})$$

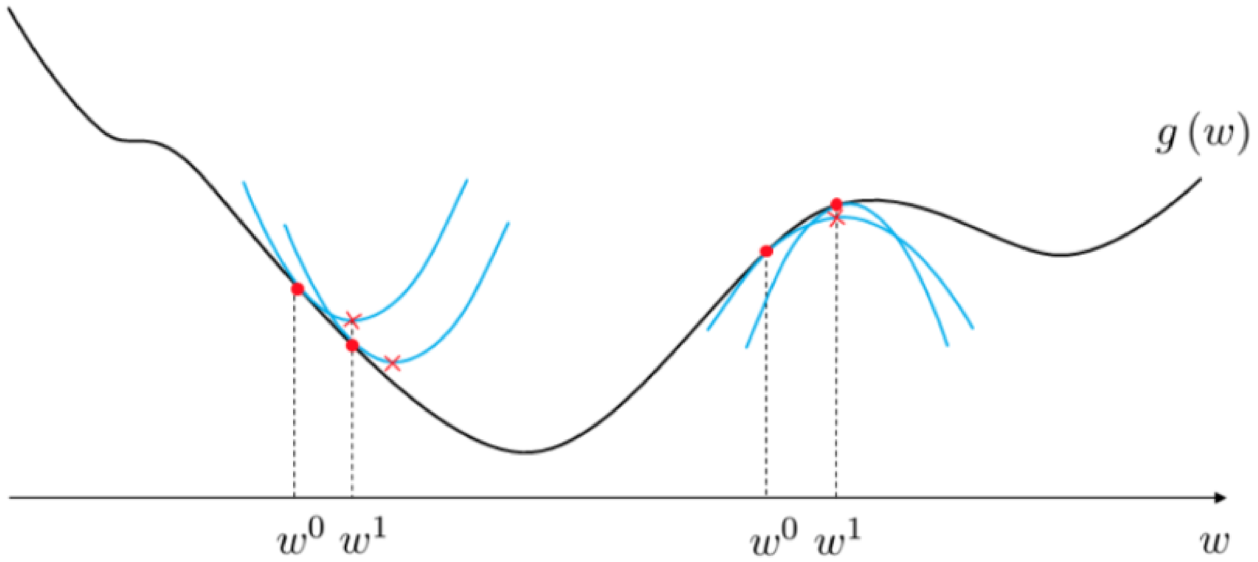
Newton's Method

- Example 1: Newton's method on a convex function
  - The quadratic approximations are themselves always convex
  - The stationary points are minima
  - The sequence leads to a minimum of the original function



Newton's Method

- Example 2: Newton's method on a non-convex function
  - The quadratic approximations can be concave or convex
  - Lead the algorithm to possibly converge to a maximum.



Newton's Method

Ensuring numerical stability

- The single-input Newton step

$$w^k = w^{k-1} - \frac{\frac{\mathrm{d}}{\mathrm{d}w} g(w^{k-1})}{\frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^{k-1})}$$

  – Near flat portions of a function, both $\frac{\mathrm{d}}{\mathrm{d}w} g(w^{k-1})$ and $\frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^{k-1})$ can be nearly zero valued.

- Regularized Newton: add a very small positive value ϵ to the second derivative:

$$w^k = w^{k-1} - \frac{\frac{\mathrm{d}}{\mathrm{d}w} g(w^{k-1})}{\frac{\mathrm{d}^2}{\mathrm{d}w^2} g(w^{k-1}) + \epsilon}$$

Newton's Method

- Multi-input functions
  - Regularized Newton: add $\epsilon \mathbf{I}_{N \times N}$, a $N \times N$ identity matrix scaled by a small positive ϵ value, to the Hessian matrix:

$$\mathbf{w}^k = \mathbf{w}^{k-1} - \left(\nabla^2 g(\mathbf{w}^{k-1}) + \epsilon \mathbf{I}_{N \times N}\right)^{-1} \nabla g(\mathbf{w}^{k-1})$$

  - $\left(\nabla^2 g(\mathbf{w}^{k-1}) + \epsilon \mathbf{I}_{N \times N}\right)$en

$$\left(\nabla^2 g(\mathbf{w}^{k-1}) + \epsilon \mathbf{I}_{N \times N}\right) \mathbf{w} = \left(\nabla^2 g(\mathbf{w}^{k-1}) + \epsilon \mathbf{I}_{N \times N}\right) \mathbf{w}^{k-1} - \nabla g(\mathbf{w}^{k-1})$$

Newton's Method

- Newton's method:

1: **input:** function g, maximum number of steps K, initial point $\mathbf{w}^0$, and regularization parameter ϵ

2: `for` $k = 1...K$

3: $\mathbf{w}^k = \mathbf{w}^{k-1} - \left(\nabla^2 g(\mathbf{w}^{k-1}) + \epsilon \mathbf{I}_{N \times N}\right)^{-1} \nabla g(\mathbf{w}^{k-1})$

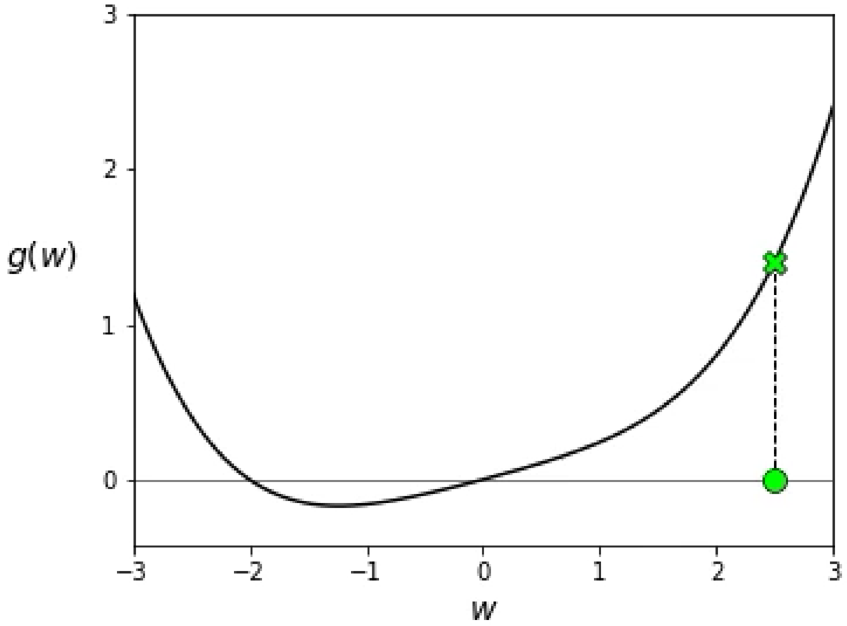
4: **output:** history of weights $\left\{\mathbf{w}^k\right\}_{k=0}^{K}$ and corresponding function evaluations $\left\{g\left(\mathbf{w}^k\right)\right\}_{k=0}^{K}$

Newton's Method

- Example 1: Newton's method applied to a convex single-input function
  - Function:

$$g(w) = \frac{1}{50}\left(w^4 + w^2 + 10w\right) + 0.5$$

  - Initialization: $w = 2.5$

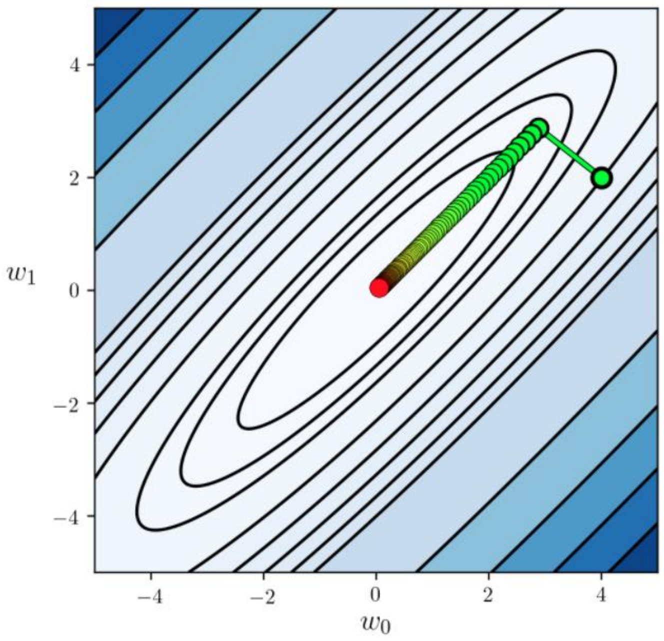


Newton's Method

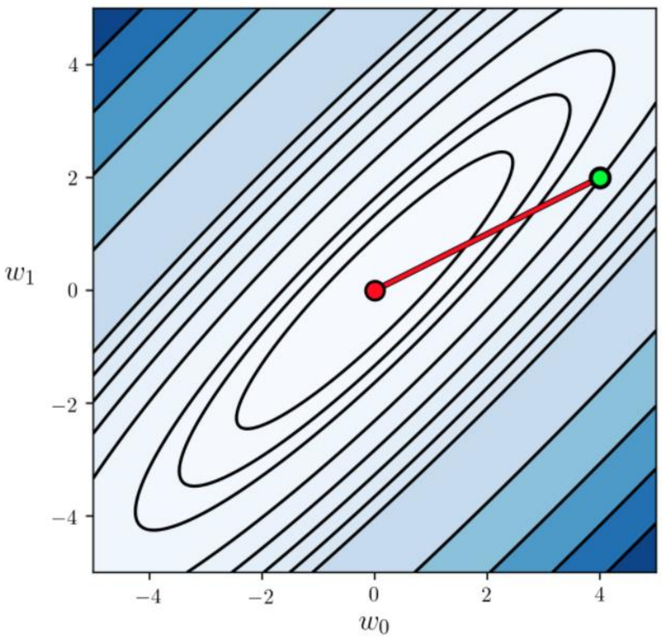
- Example 2: Minimizing a quadratic function with a single Newton step
  - Function:

$$g(w_1, w_2) = 0.26(w_1^2 + w_2^2) - 0.48 w_1 w_2$$

100 steps of gradient descent



A single Newton step



Limitation of Newton's Method

A Newton's method step requires far more in terms of storage and computation than a first order step

- Requires the storage and computation of not just a gradient but an entire $N \times N$ Hessian matrix of second derivative information.
- In machine learning, this can easily have tens of thousands to hundreds of thousands or even hundreds of millions of inputs, making the complete storage of an associated Hessian impossible.

Takeaways

- Understand Gradient Descent
- Understanding Selection of Step Size for Gradient Descent
- Understanding Newton's Method
- Be able to compute Gradients
- Be able to evaluate function optimality based upon Gradient Descent
- **Next Time: Logistic Regression**