ED5340 - Data Science: Theory and Practise

L15 - Optimization for multiple variable

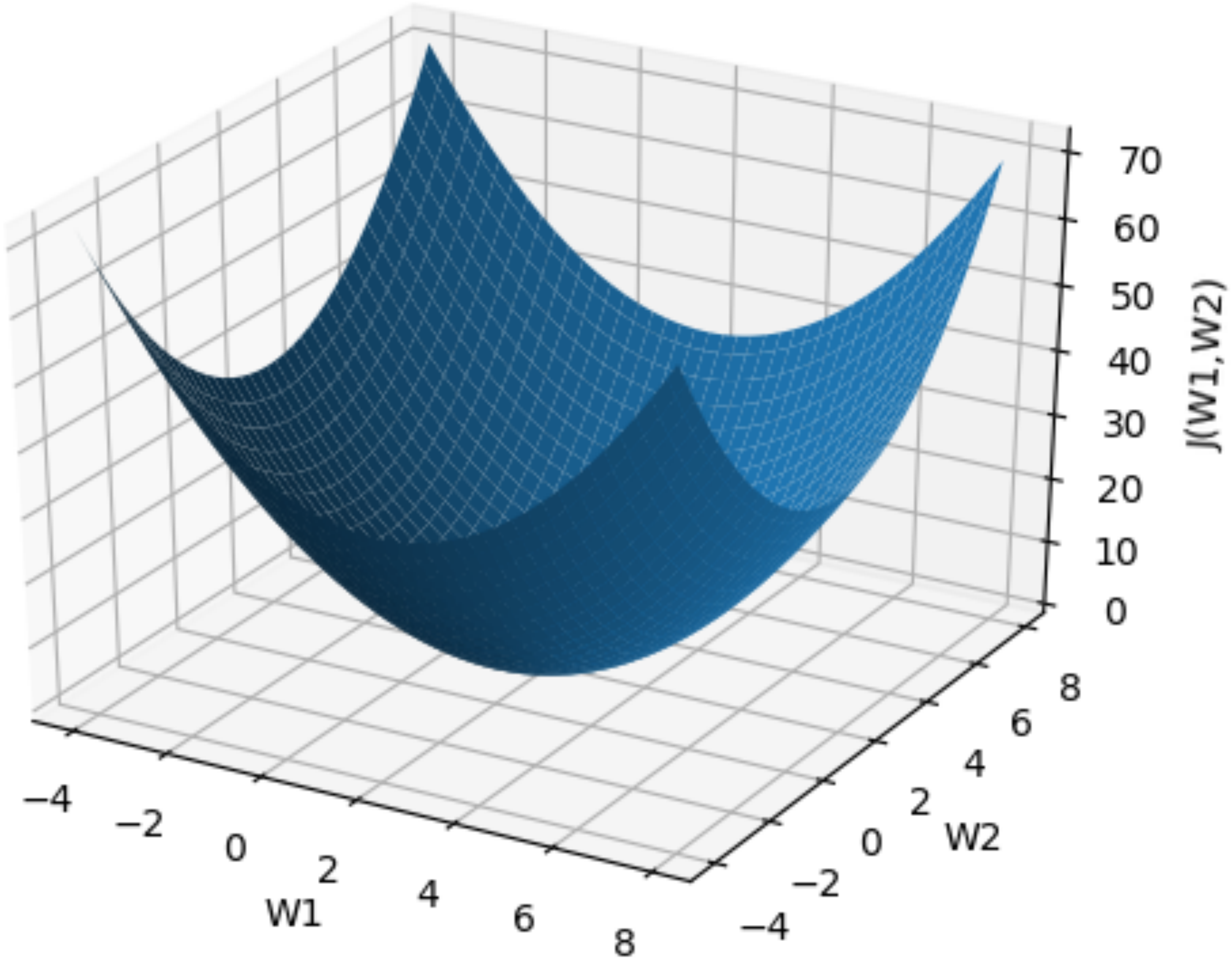
Ramanathan Muthuganapathy (https://ed.iitm.ac.in/~raman)
Course web page: https://ed.iitm.ac.in/~raman/datascience.html
Moodle page: Available at https://courses.iitm.ac.in/

Unconstrained optimization

- Single variable (e.g. min $J(w)$, e.g $J(w) = w^2, J(w) = w^3$, $J(w) = w^2 + 54/w$)
- multivariable (e.g. $min\, J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$)
- n-dimensional multivariable (e.g. $J(w_1, w_2, \ldots\ldots, w_n) = (w_1 - 2)^2 + (w_2 - 2)^2 + \ldots. + (w_n - 2)^2$))
- $min\, J(w_1, w_2, \ldots\ldots, w_n)$

Surface plot

$J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$

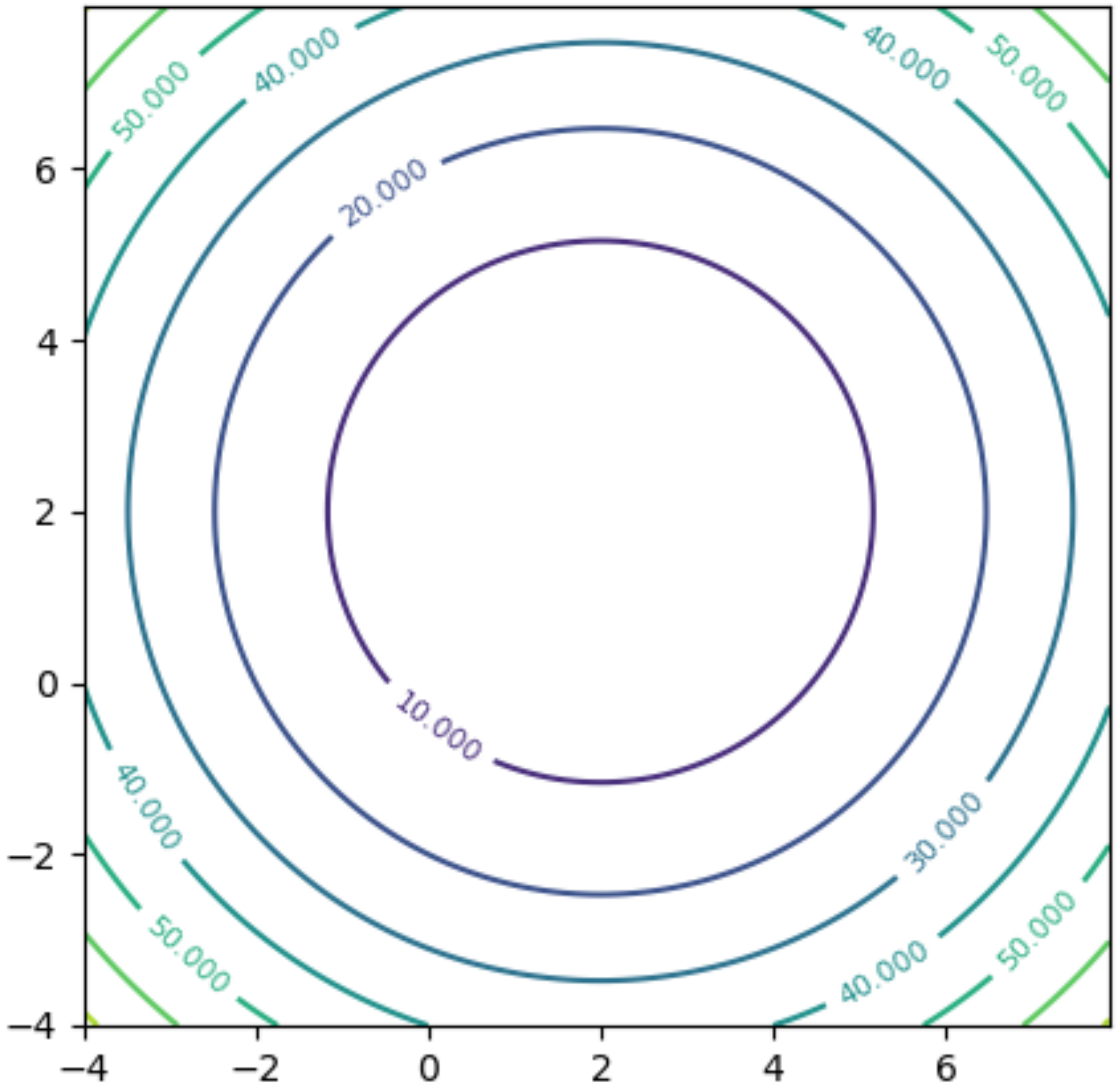


Contour plot / level set / height function

$J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$

- Two points in a contour have the same J value

- Imaging cutting with J-plane at different J-values



Demo using SrfPlots.py

Optimality criteria - multiple variables

- min $J(w)$
  - The value of w for which the function $J(w)$ has the least (minimum) value
  - Local minimum

Gradient - multiple variables

$min\, J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$ **- Partial derivatives**

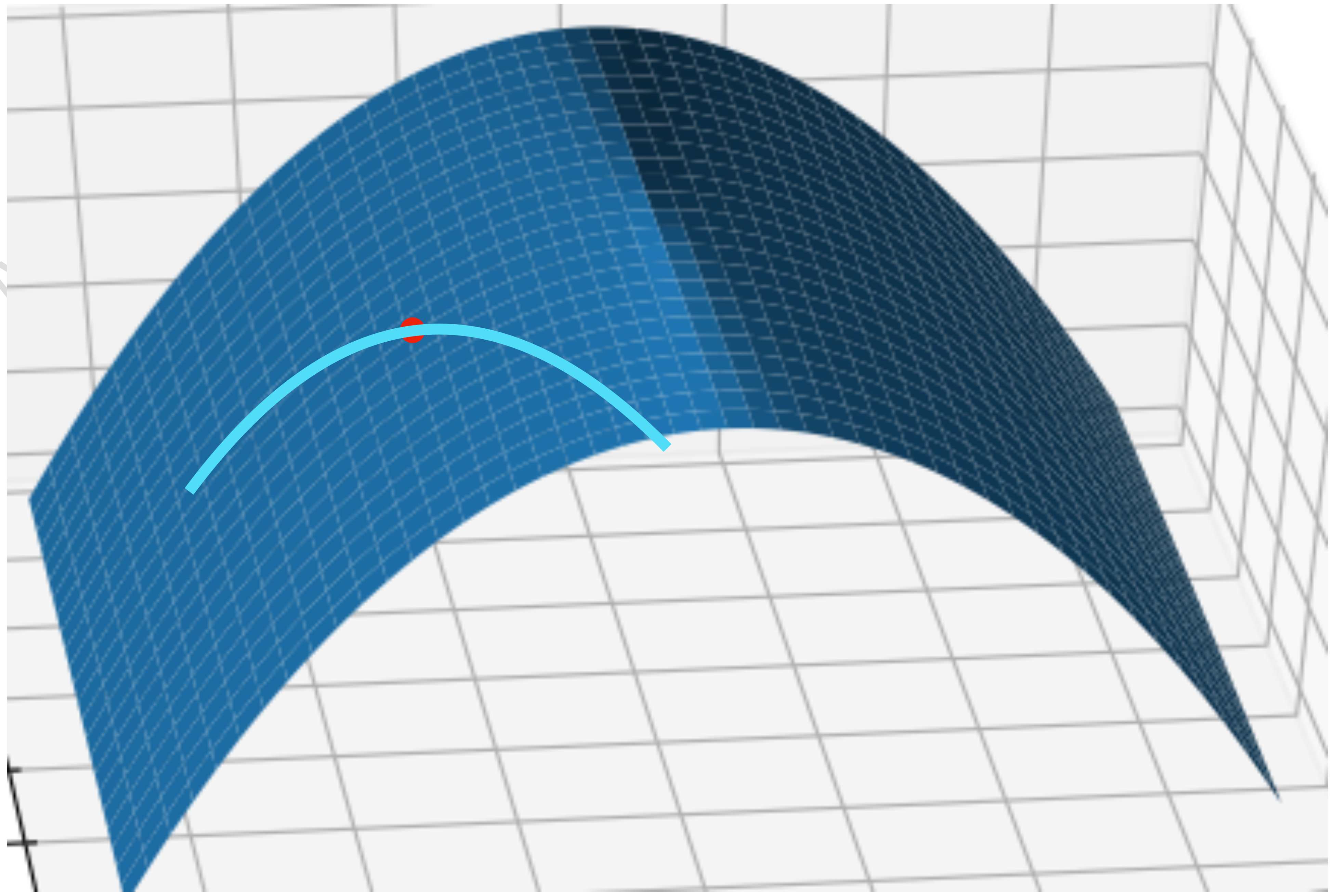
- $J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$
- $\dfrac{\partial J}{\partial w_1}$ - Partial derivation of $J(w_1, w_2)$ wrt w_1
- $\dfrac{\partial J}{\partial w_2}$ - Partial derivation of $J(w_1, w_2)$ wrt w_2
- $\nabla J(w_1, w_2) = \left(\dfrac{\partial J}{\partial w_1}, \dfrac{\partial J}{\partial w_2}\right)$, where $\nabla J(w_1, w_2)$ or grad. J
- NOTE: grad. J is a vector.

Gradient - multiple variables

What is $\nabla J(w_1, w_2)$ or grad. J?

- Surface f(x, y, z) = c
- Any curve f(x(t), y(t), z(t))
- $\dfrac{\partial f}{\partial x}\dfrac{dx}{dt} + \dfrac{\partial f}{\partial y}\dfrac{dy}{dt} + \dfrac{\partial f}{\partial z}\dfrac{dz}{dt} = 0$
- $\left(\dfrac{\partial f}{\partial x}, \dfrac{\partial f}{\partial y}, \dfrac{\partial f}{\partial z}\right) \cdot \left(\dfrac{dx}{dt}, \dfrac{dy}{dt}, \dfrac{dz}{dt}\right) = 0$
- $\nabla f \,.\, (x'(t), y'(t) \,.\, z'(t)) = 0$

f(x, y, z)

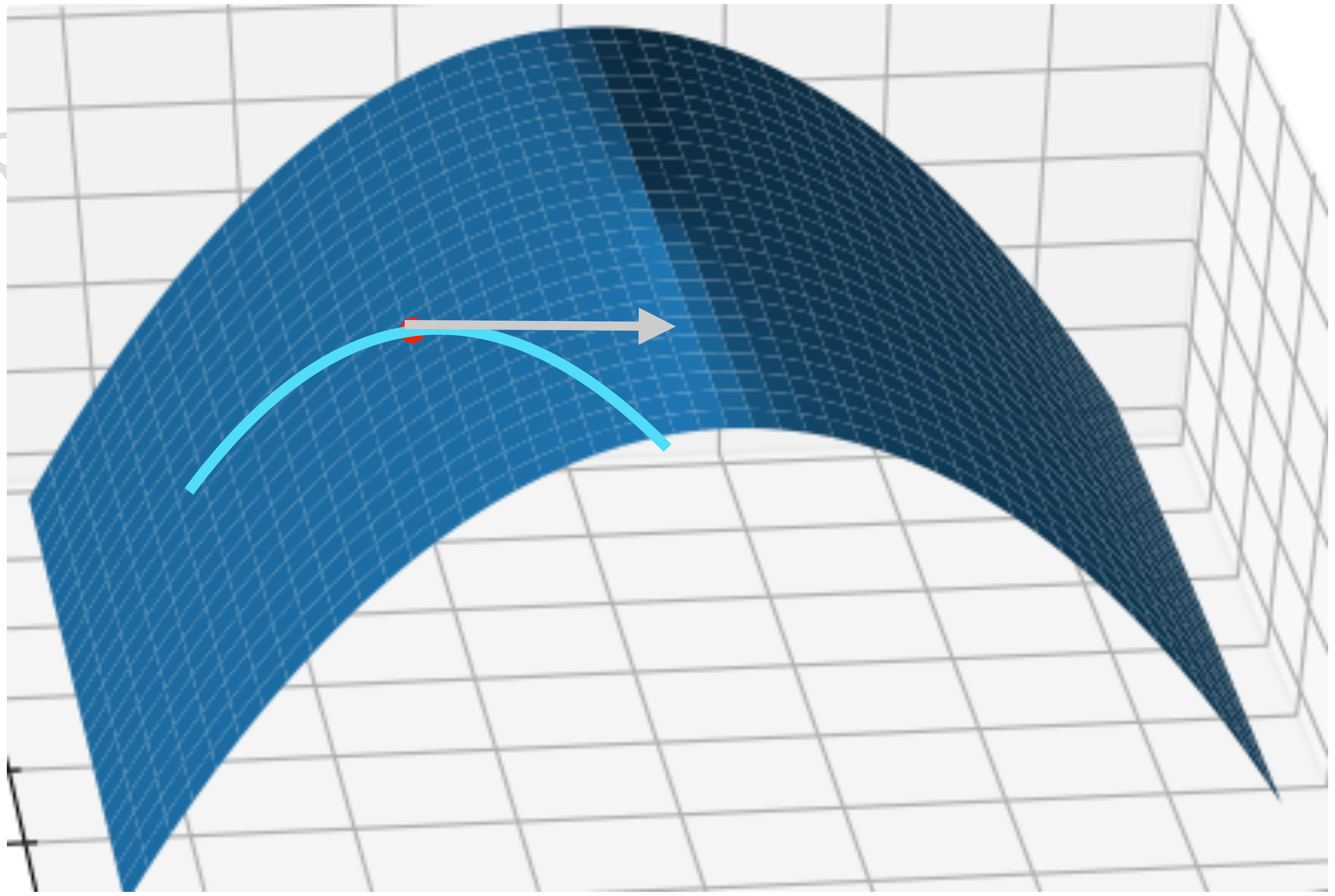


Gradient - multiple variables

What is $\nabla J(w_1, w_2)$ or grad. J?

- ∇f is the grad. f and $(x^{'}(t), y^{'}(t)\,.\,z^{'}(t))$ is the tangent vector.

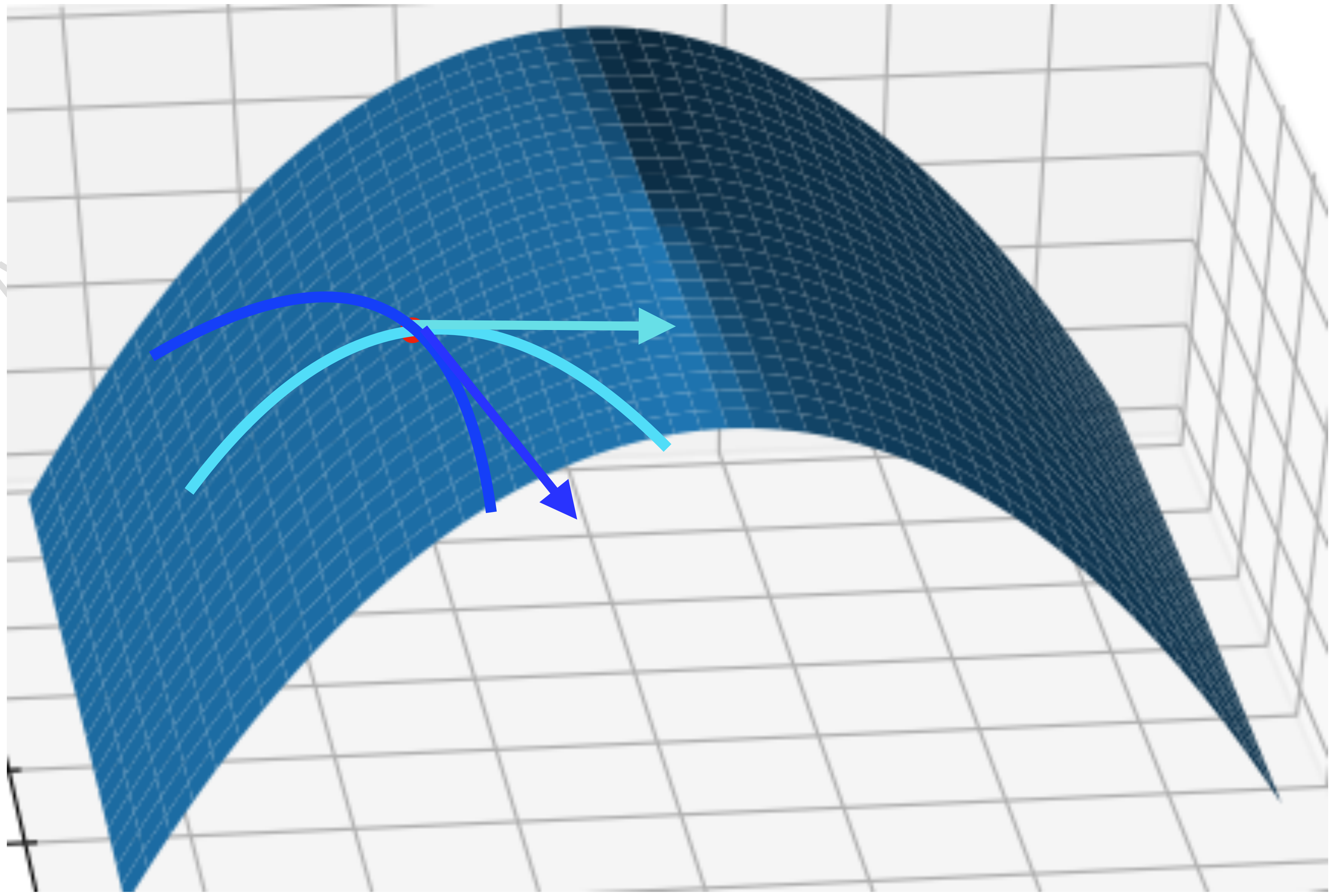
f(x, y, z)



Gradient - multiple variables

What is $\nabla J(w_1, w_2)$ or grad. J?

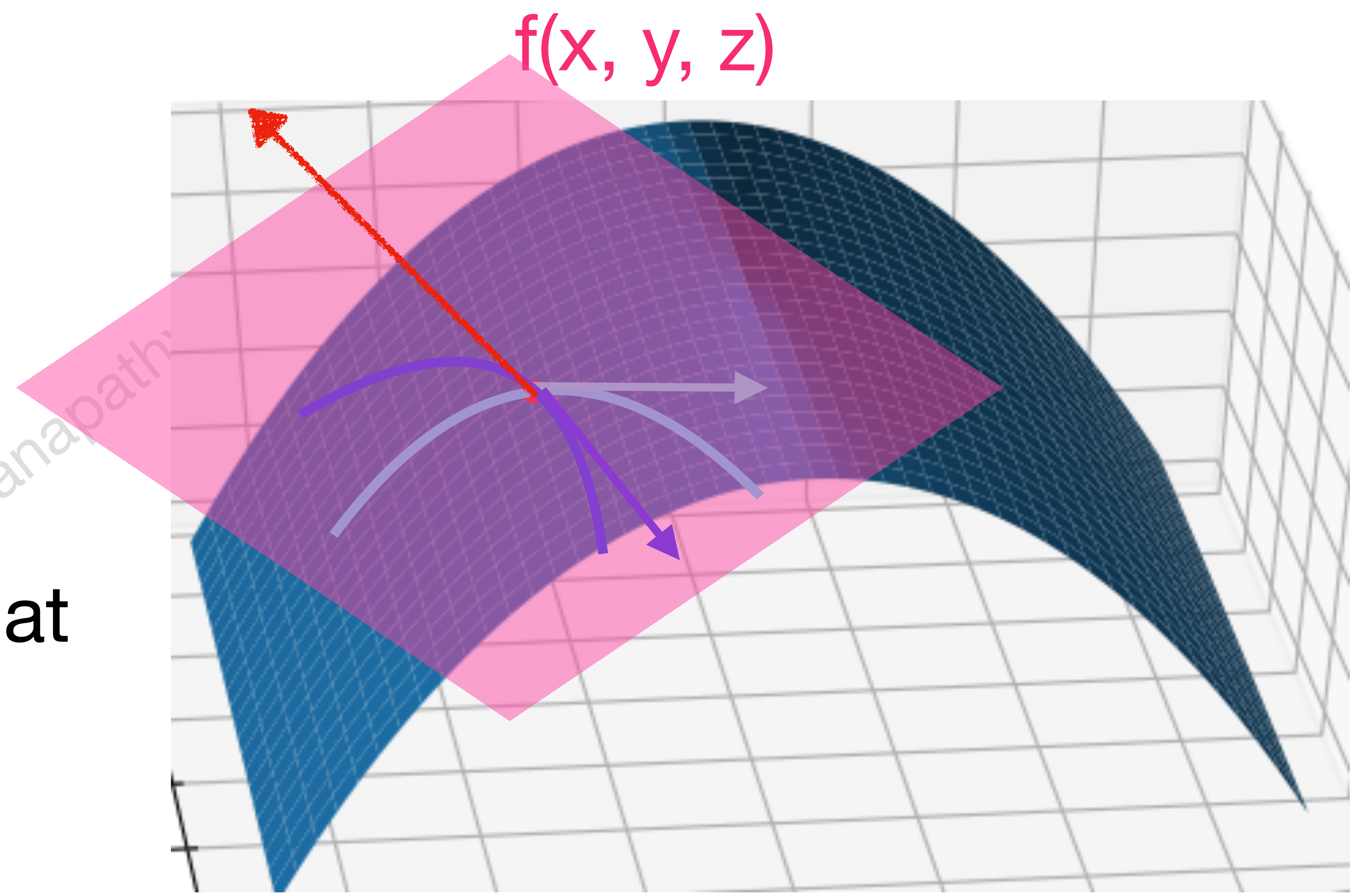
- Take another curve (blue)
- $\nabla f \,.\, (x^{'}(t), y^{'}(t) \,.\, z^{'}(t)) = 0$
- Dot product
- ∇f is perpendicular to set of tangents at that point.



Gradient - multiple variables

What is $\nabla J(w_1, w_2)$ or grad. J?

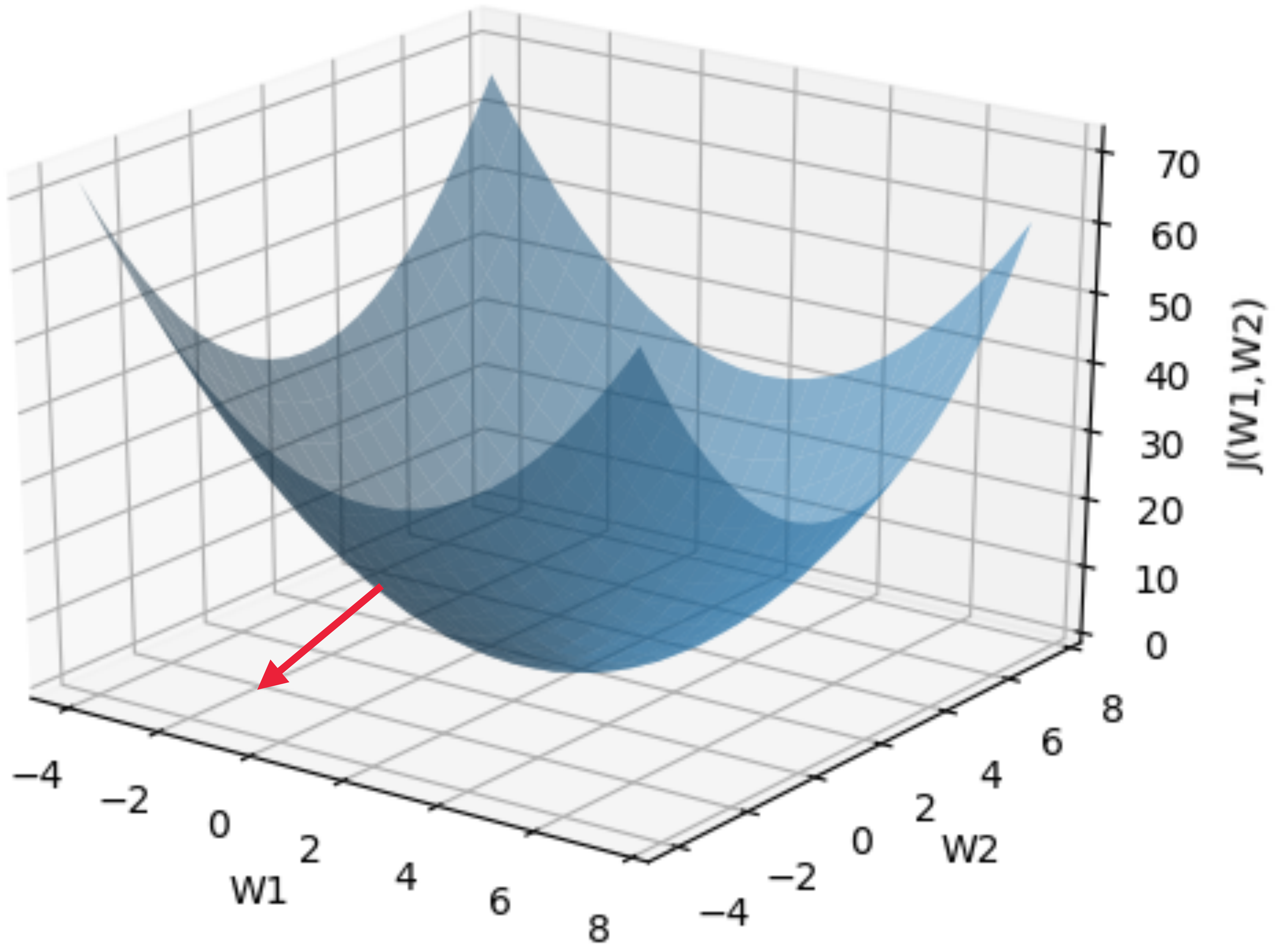
- $\nabla f \,.\, (x^{'}(t), y^{'}(t) \,.\, z^{'}(t)) = 0$
- Dot product
- ∇f is perpendicular to set of tangents at that point.
- ∇f is the Normal vector at a point.



Gradient at a point

What is $\nabla J(w_1, w_2)$ or grad. J? - Back to our notation

- $\nabla J(w_1, w_2)$ is a normal vector



Hessian Matrix

$min\, J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$ **- Second partial derivatives**

- $\dfrac{\partial^2 J}{\partial w_1^2} = \dfrac{\partial}{\partial w_1}\left(\dfrac{\partial J}{\partial w_1}\right)$
- $\dfrac{\partial^2 J}{\partial w_2^2} = \dfrac{\partial}{\partial w_2}\left(\dfrac{\partial J}{\partial w_2}\right)$
- $\dfrac{\partial^2 J}{\partial w_1 \partial w_2} = \dfrac{\partial}{\partial w_1}\left(\dfrac{\partial J}{\partial w_2}\right)$
- $\dfrac{\partial^2 J}{\partial w_2 \partial w_1} = \dfrac{\partial}{\partial w_2}\left(\dfrac{\partial J}{\partial w_1}\right)$

Hessian Matrix

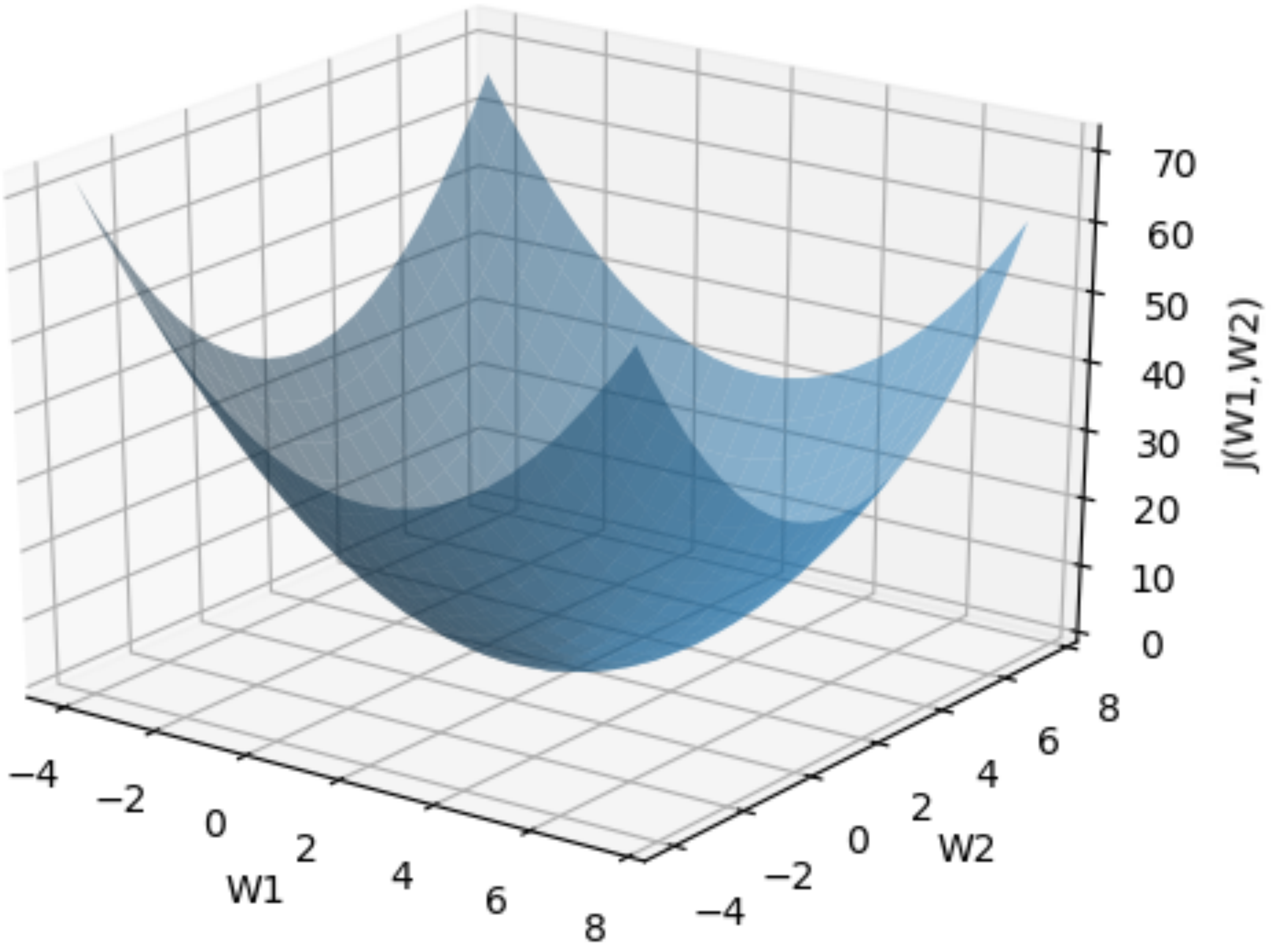
Matrix of second partial derivatives

$$H = \begin{bmatrix} \dfrac{\partial^2 J}{\partial w_1^2} & \dfrac{\partial^2 J}{\partial w_1 \partial w_2} \\ \dfrac{\partial^2 J}{\partial w_2 \partial w_1} & \dfrac{\partial^2 J}{\partial w_2^2} \end{bmatrix}$$

Optimality Criteria for Multiple Variables

$$min\, J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$$

- $\nabla J(w_1, w_2)$ = 0, Get $w^* = (w_1^*, w_2^*)$
- Hessian H should be positive definite at w^* for min
- Hessian H should be negative definite at w^* for max
- At a saddle point, H is indefinite



Optimality Criteria for Multiple Variables

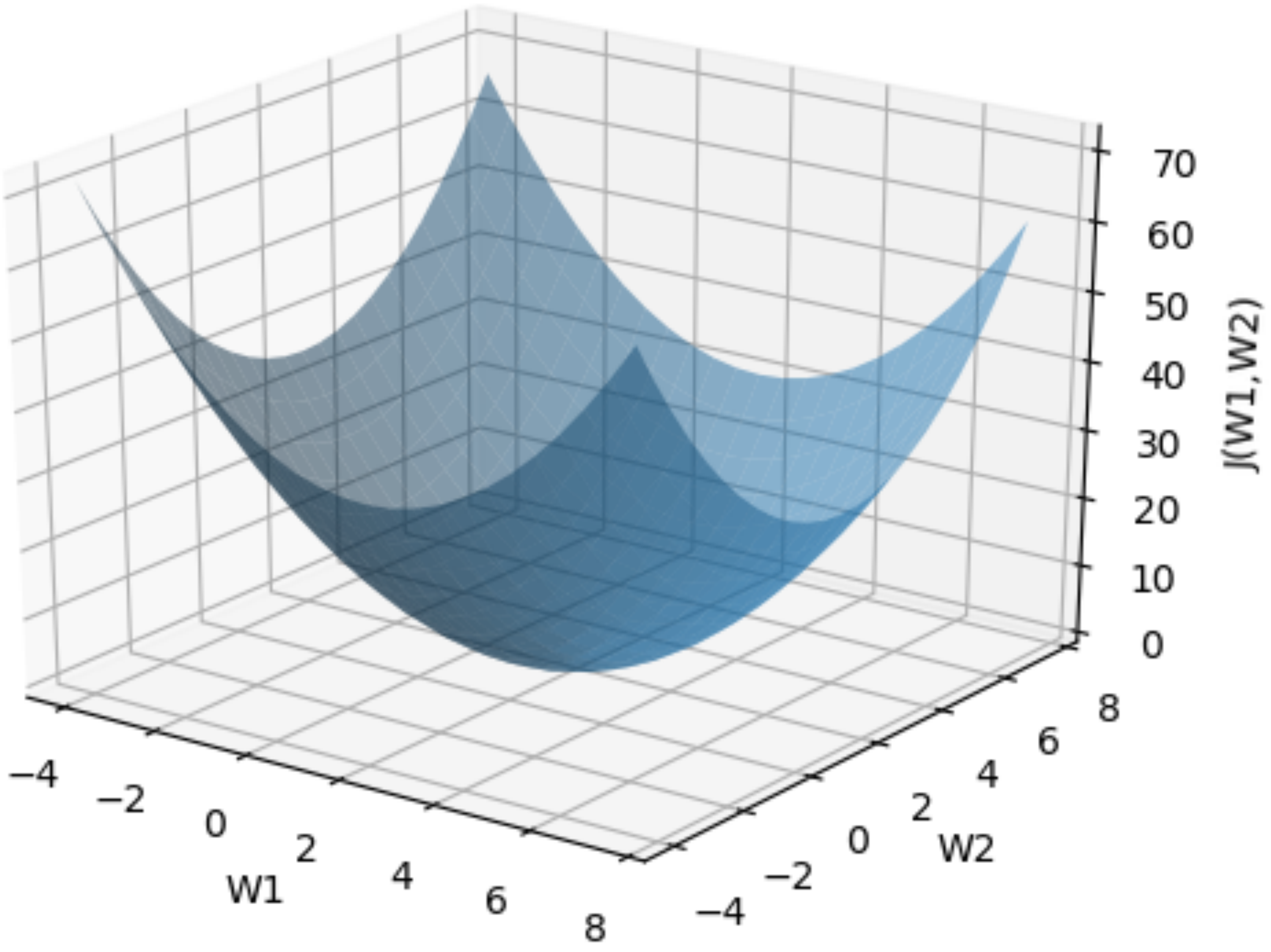
How to find the type for H? (Use LA)

- H is positive definite if all the Eigenvalues are > 0 (All $\lambda_i's > 0$)
- H is negative definite if all the Eigenvalues are < 0 (All $\lambda_i's < 0$)
- H is indefinite if some Eigenvalues are > 0 and some are < 0 (All $\lambda_i's > 0$)

Example

$min\, J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$

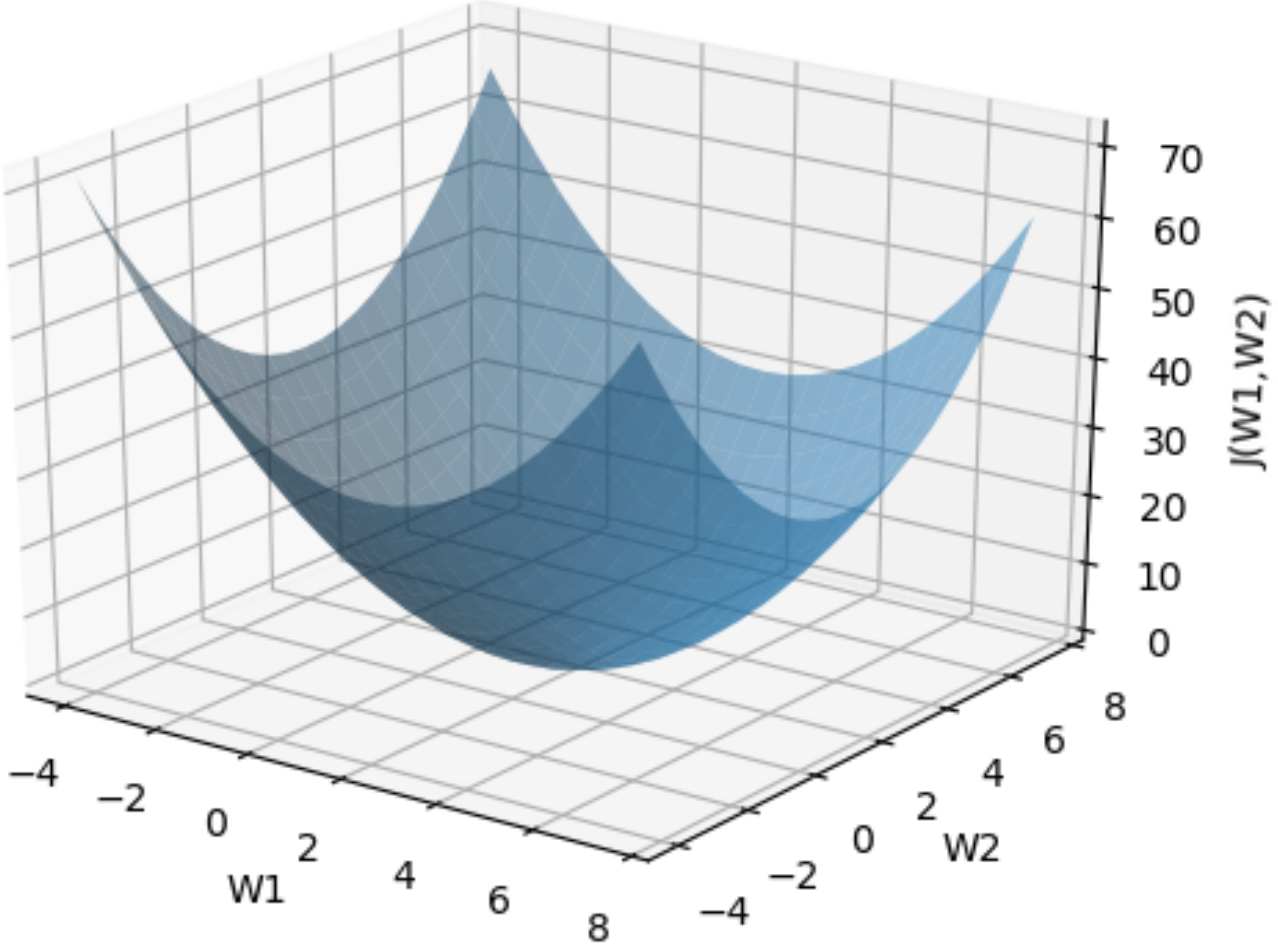
- $\nabla J(w_1, w_2)$ = 0, Get $w^* = (w_1^*, w_2^*)$
- $\dfrac{\partial J}{\partial w_1} = 2(w_1 - 2)$
- $\dfrac{\partial J}{\partial w_2} = (2w_2 - 2)$
- Critical point
  $w^* = (w_1^*, w_2^*) = (2, 2)$



Example

Compute Hessian at (2, 2)

- $$\frac{\partial^2 J}{\partial w_1^2} = 2$$
- $$\frac{\partial^2 J}{\partial w_2^2} = 2$$
- $$\frac{\partial^2 J}{\partial w_1 \partial w_2} = 0$$
- $$\frac{\partial^2 J}{\partial w_2 \partial w_1} = 0$$



Hessian Matrix

Matrix of second partial derivatives

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 2 \end{bmatrix}$$

Eigen values ?

H is then ____________ definite and hence the point $w^* = (w_1^*, w_2^*) = (2, 2)$ is local ________________

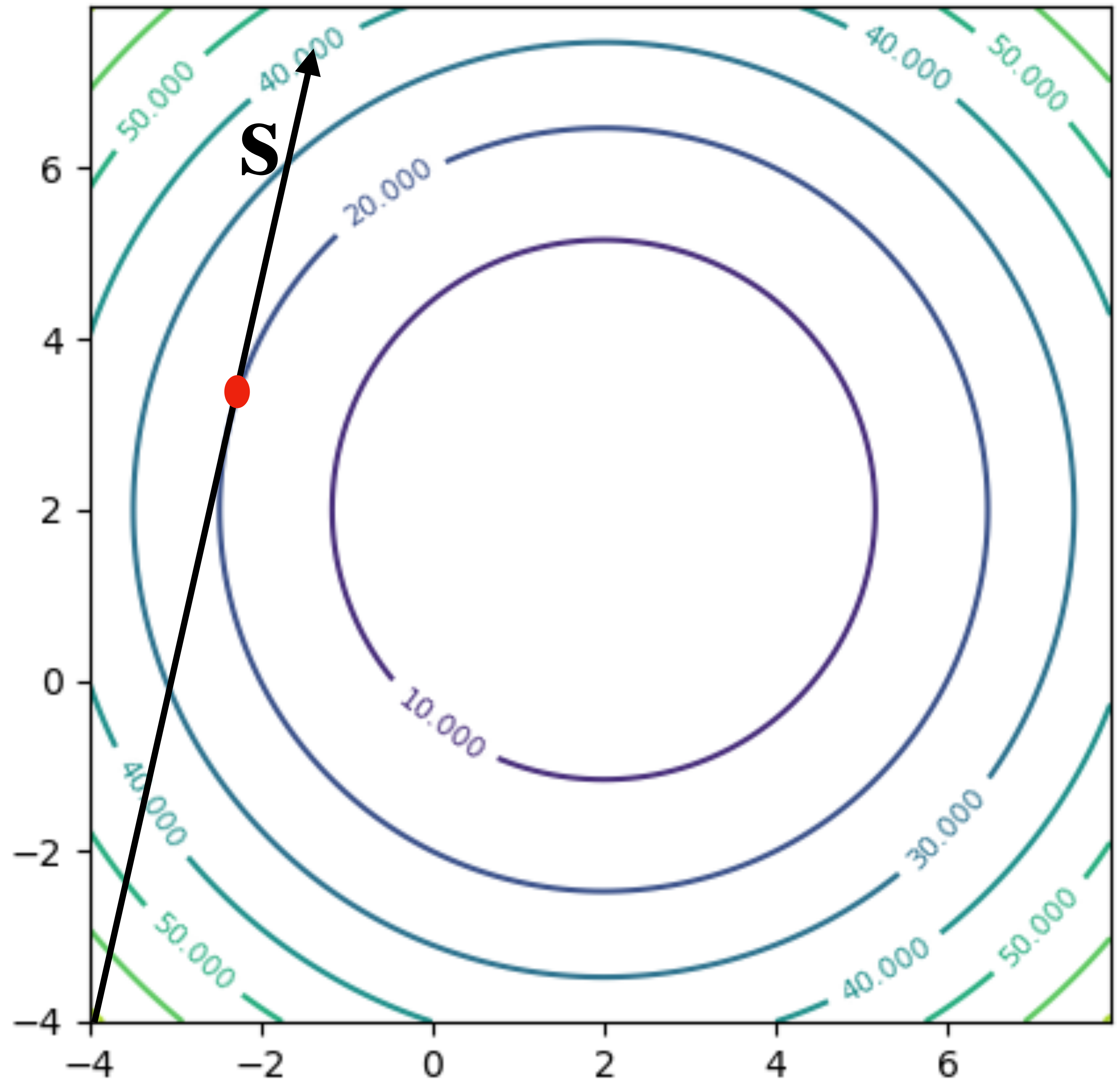
CW: Do a similar exercise for

$$J(w_1, w_2) = w_1^2 - w_2^2$$

Unidirectional search

$J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$

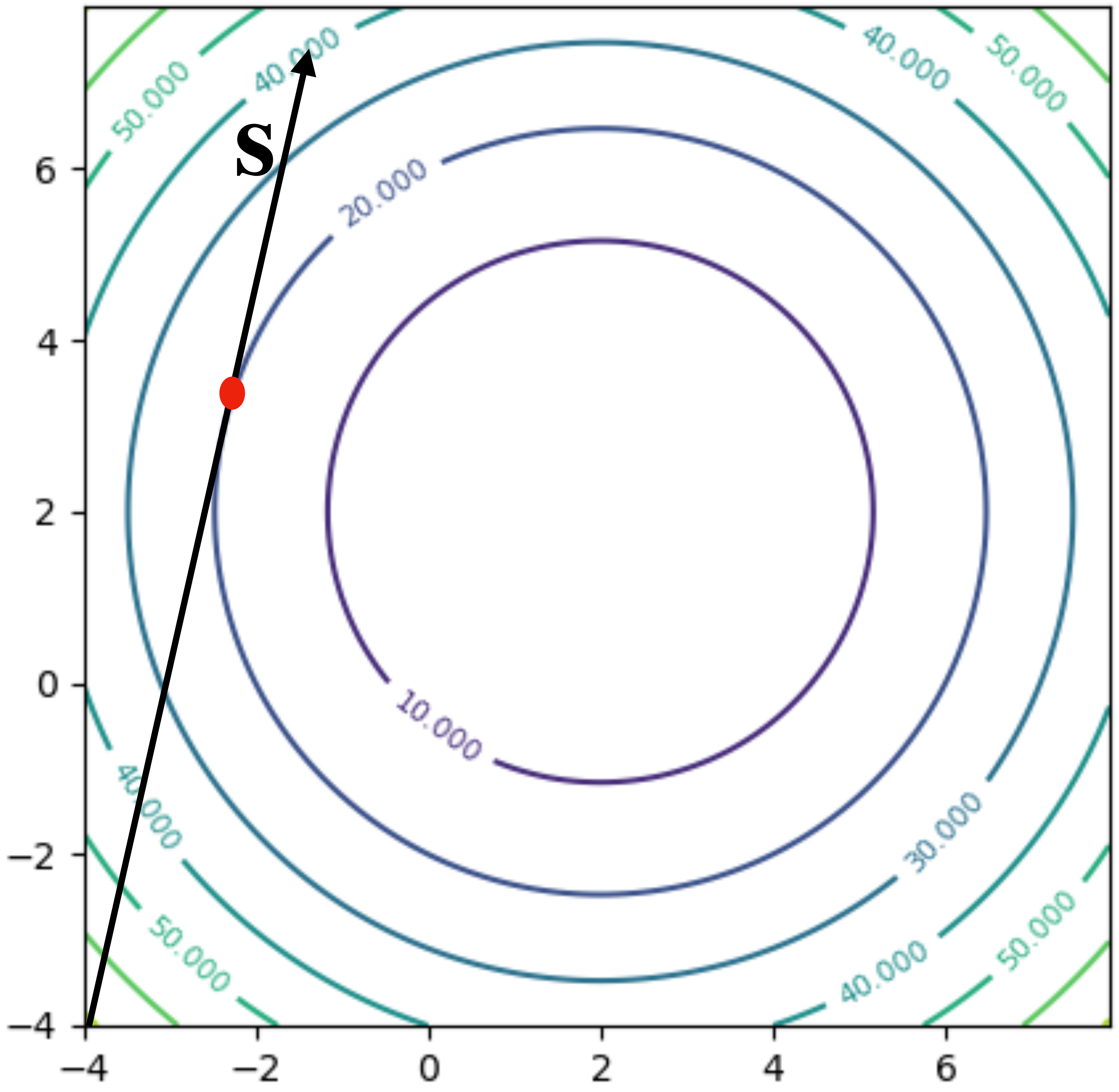
- Starting point
  $w^s = (w_1^s, w_2^s) = (-4, \ -4)$
- Search direction $\mathbf{s}$ (vector)
- $w^* = w^s + \alpha \mathbf{S}$
- Bracketing method to find α
- Fine tuning with interval halving (or golden search etc.)



Unidirectional search - Issues

$J(w_1, w_2) = (w_1 - 2)^2 + (w_2 - 2)^2$

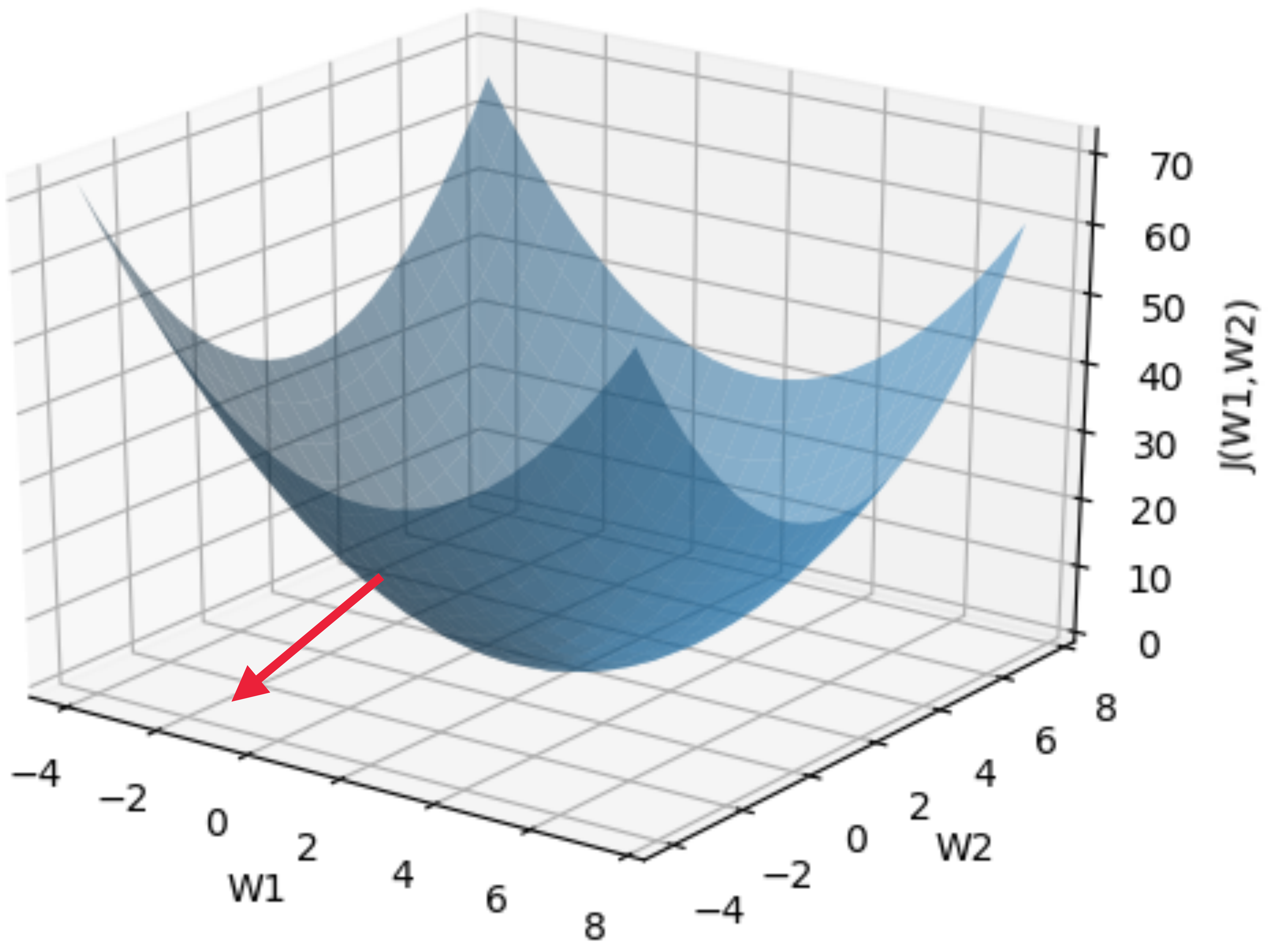
- Starting point
- Search direction $\mathbf{s}$ (vector)

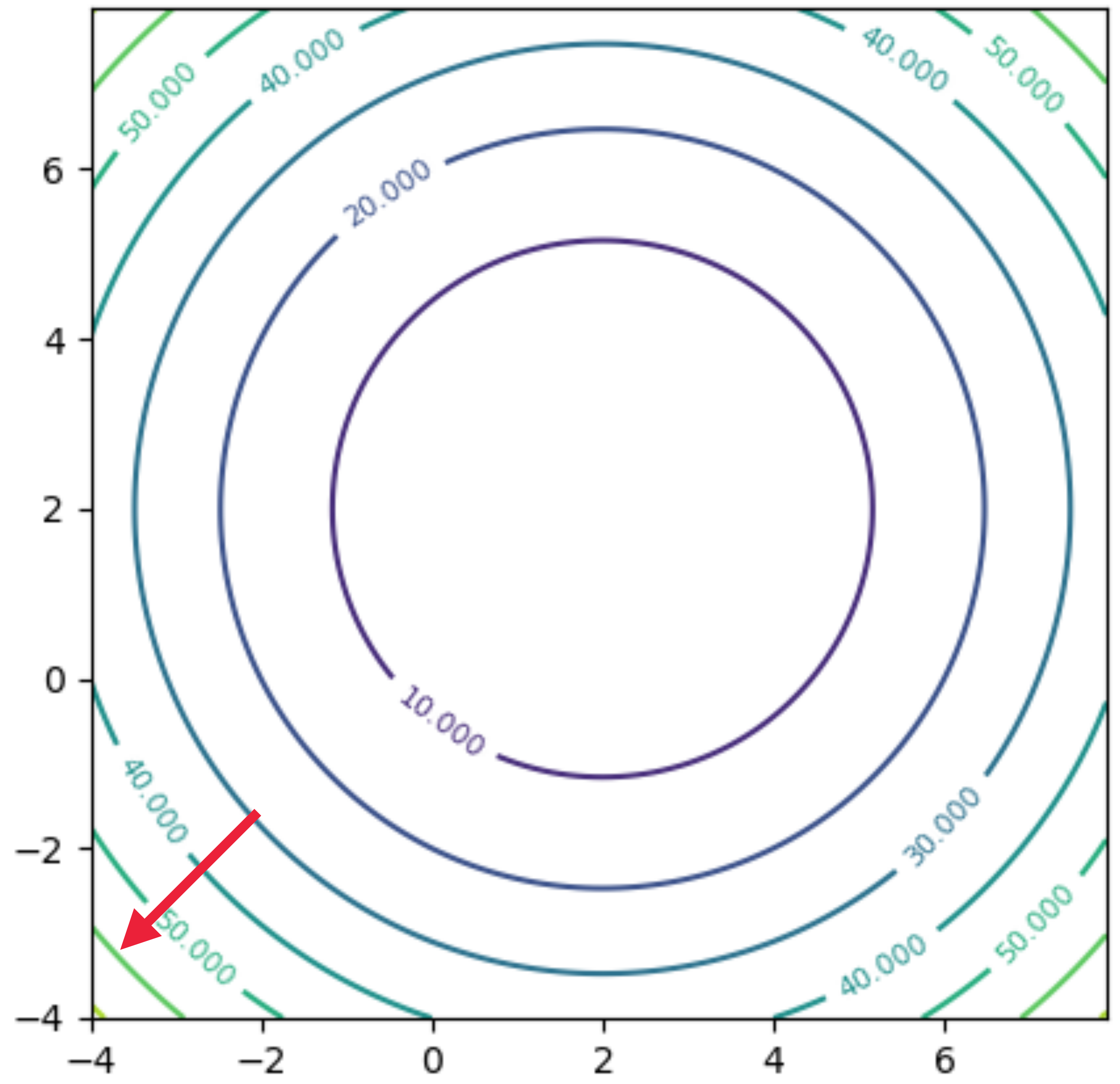


Gradient at a point

What is $\nabla J(w_1, w_2)$ or grad. J? - Back to our notation

- $\nabla J(w_1, w_2)$ is a normal vector



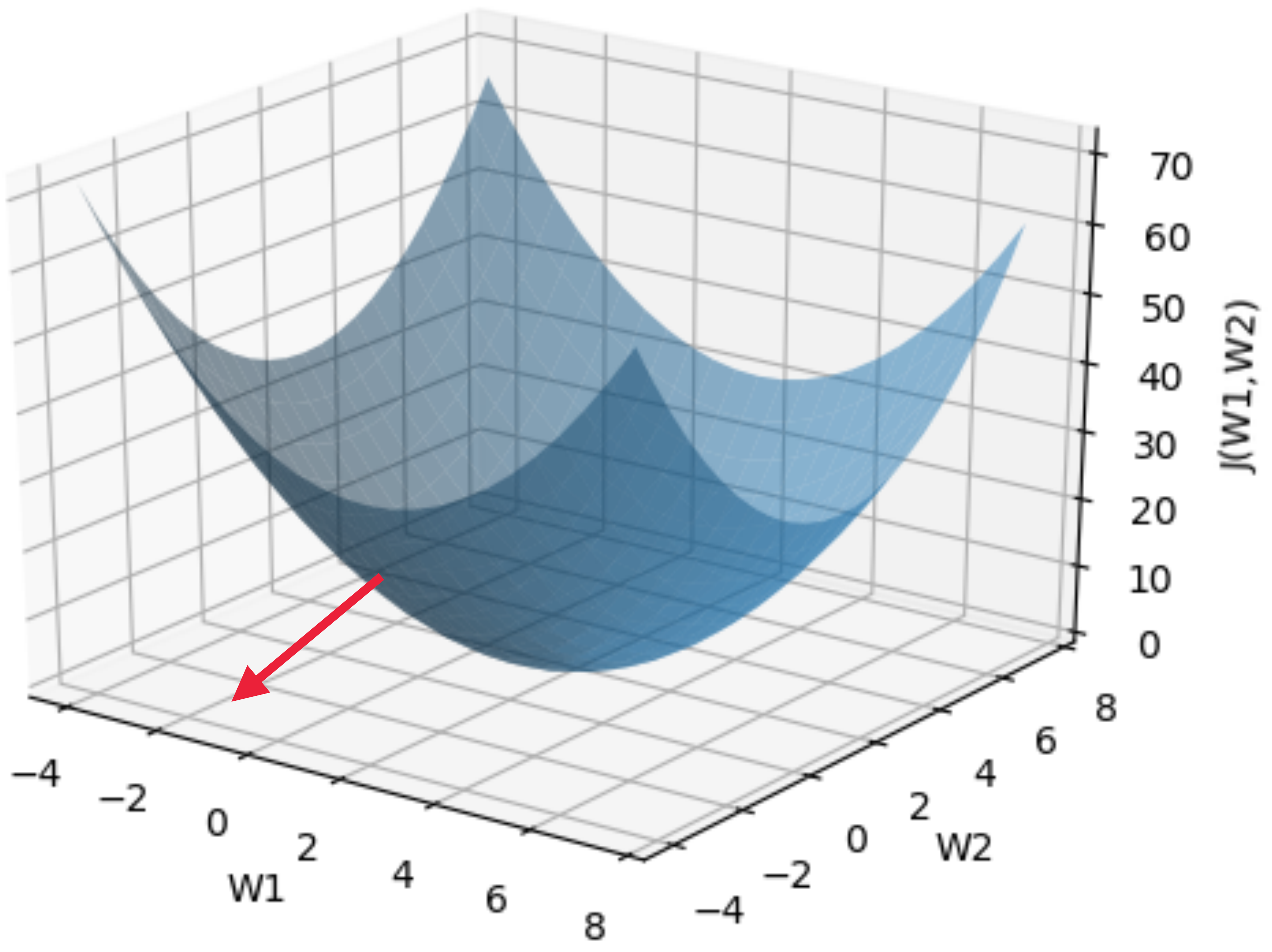


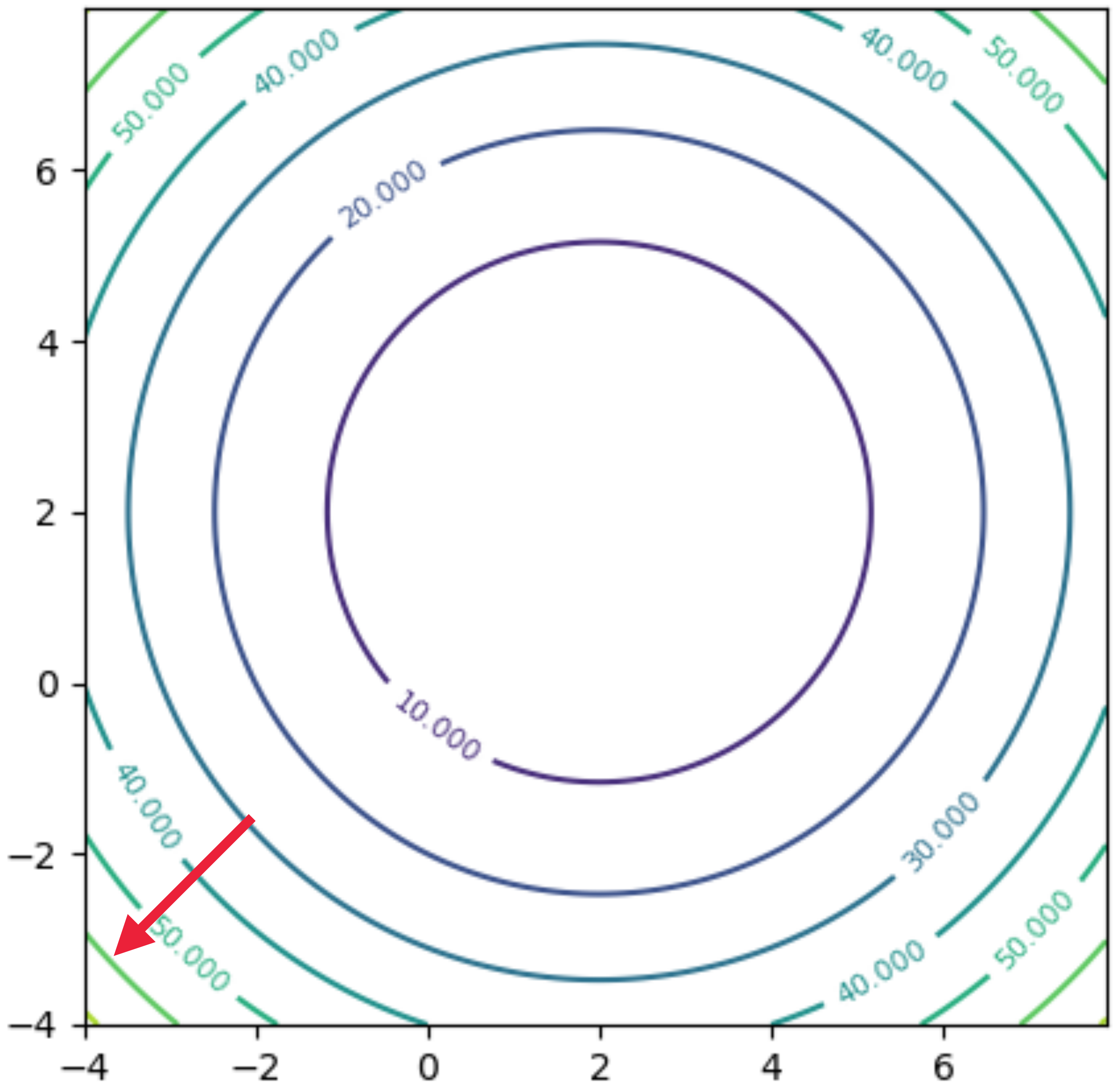
Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Gradient at a point

Traveling along grad. J

- If you travel along the direction of the grad. J, what happens to J?

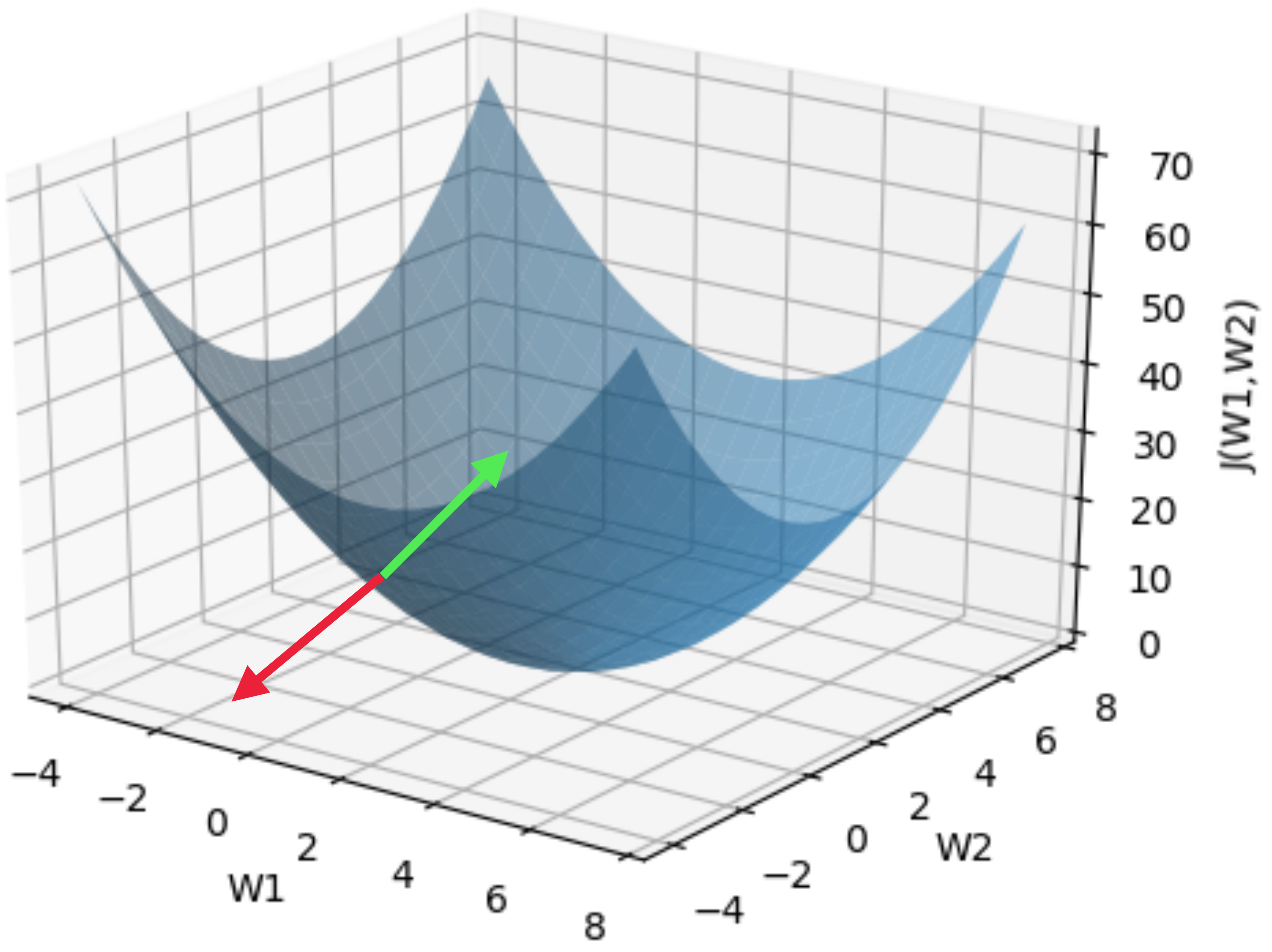


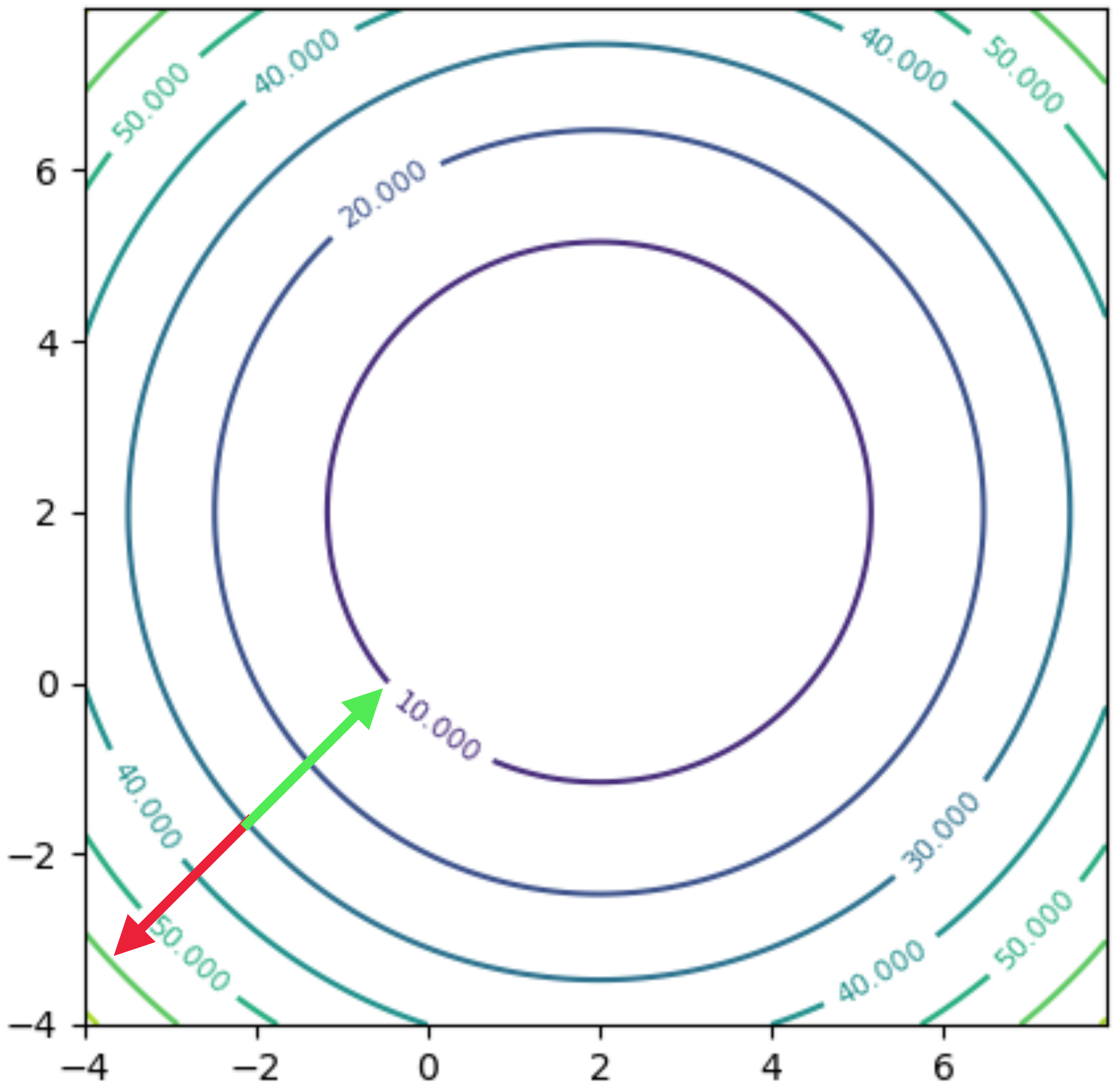


Gradient descent

Traveling along -grad. J or $-\nabla J(w_1, w_2)$

- We should travel along $-\nabla J$



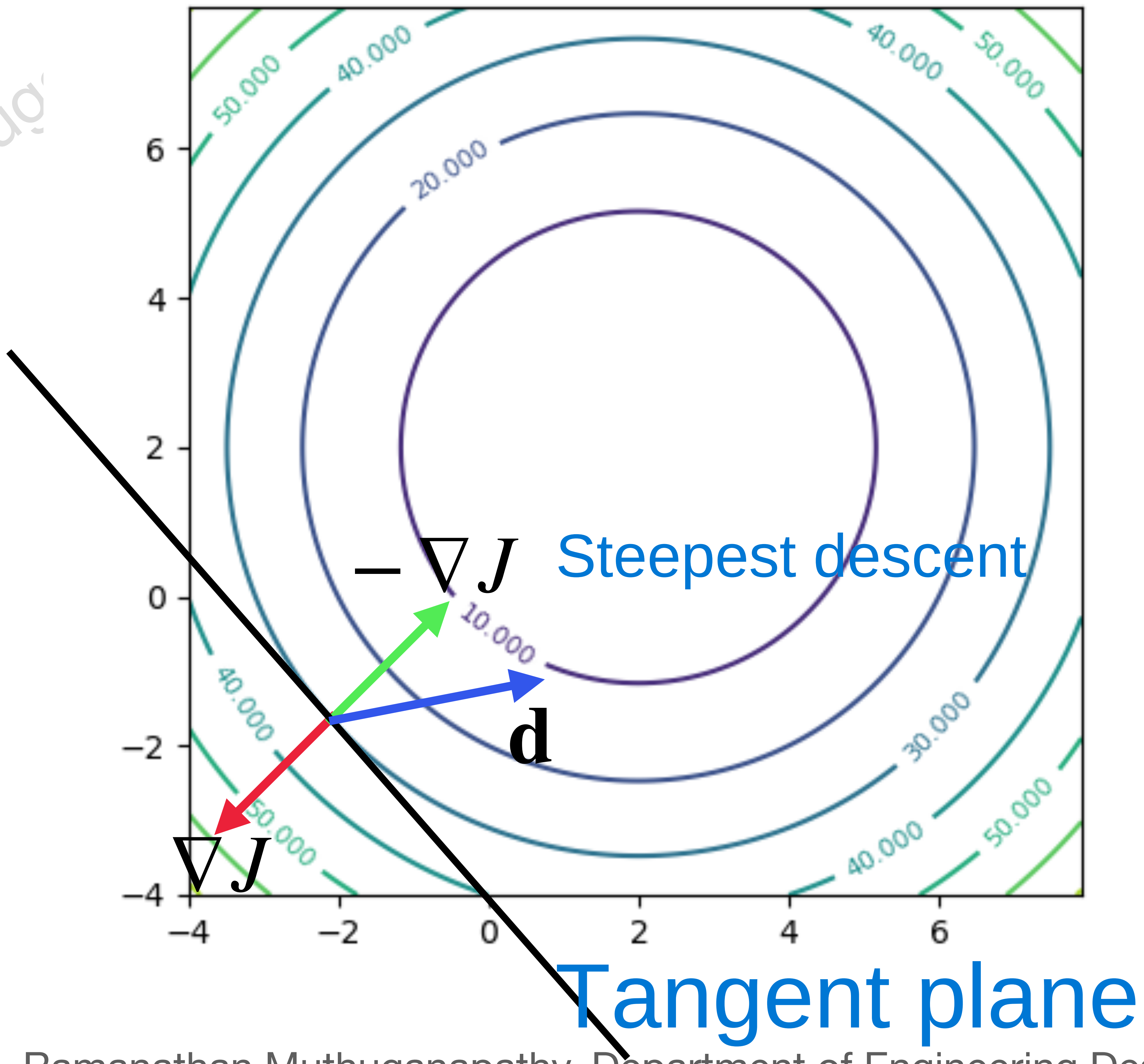


Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Potential directions and steepest descent

Traveling along -grad. J **or** $-\nabla J(w_1, w_2)$

- Let $\mathbf{d}$ be such that $\nabla J\,.\,\mathbf{d}$ is -ve.
- Let $\mathbf{d}$ be such that $\mathbf{d} = -\nabla J$
- $\nabla J. -\nabla J = -1$
- Hence $-\nabla J$ is the steepest!
- Steepest (Cauchy's) Gradient Descent

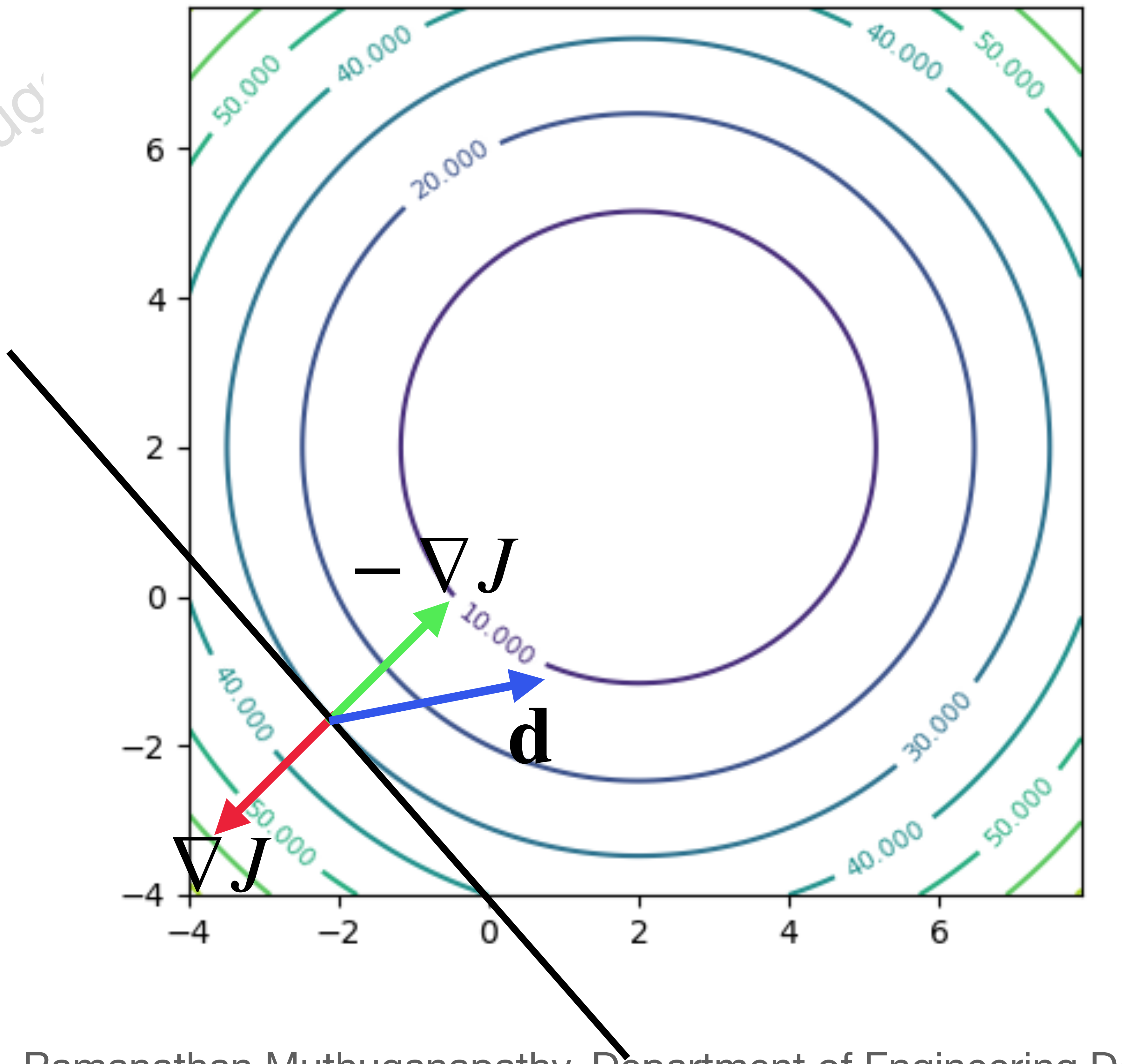


Algorithm - Gradient descent

Traveling along -grad. J or $-\nabla J(w_1, w_2)$

- Starting point $w* = (w_1^*, w_2^*)$
- Compute $J, \ -\nabla J$ at $w_k^* = w*$.
- Update w's
  - $w_{k+1}^* = w_k^* - \alpha_k \nabla J$
- Check for stopping criteria
- Else continue the iteration

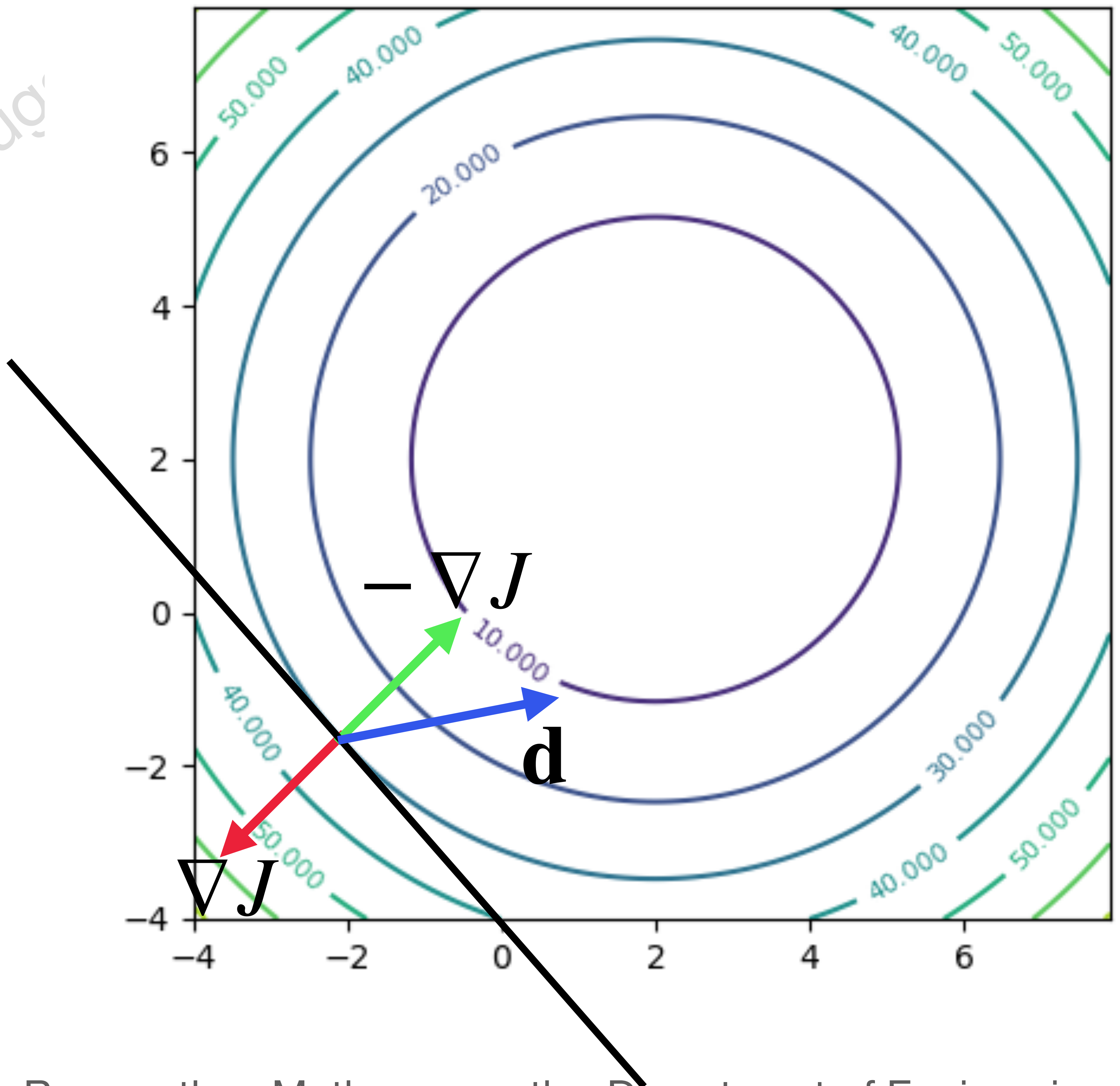
Del_J - compute using central diffeeerence



Algorithm - Update step

$w^*_{k+1} = w^*_k - \alpha_k \nabla J$

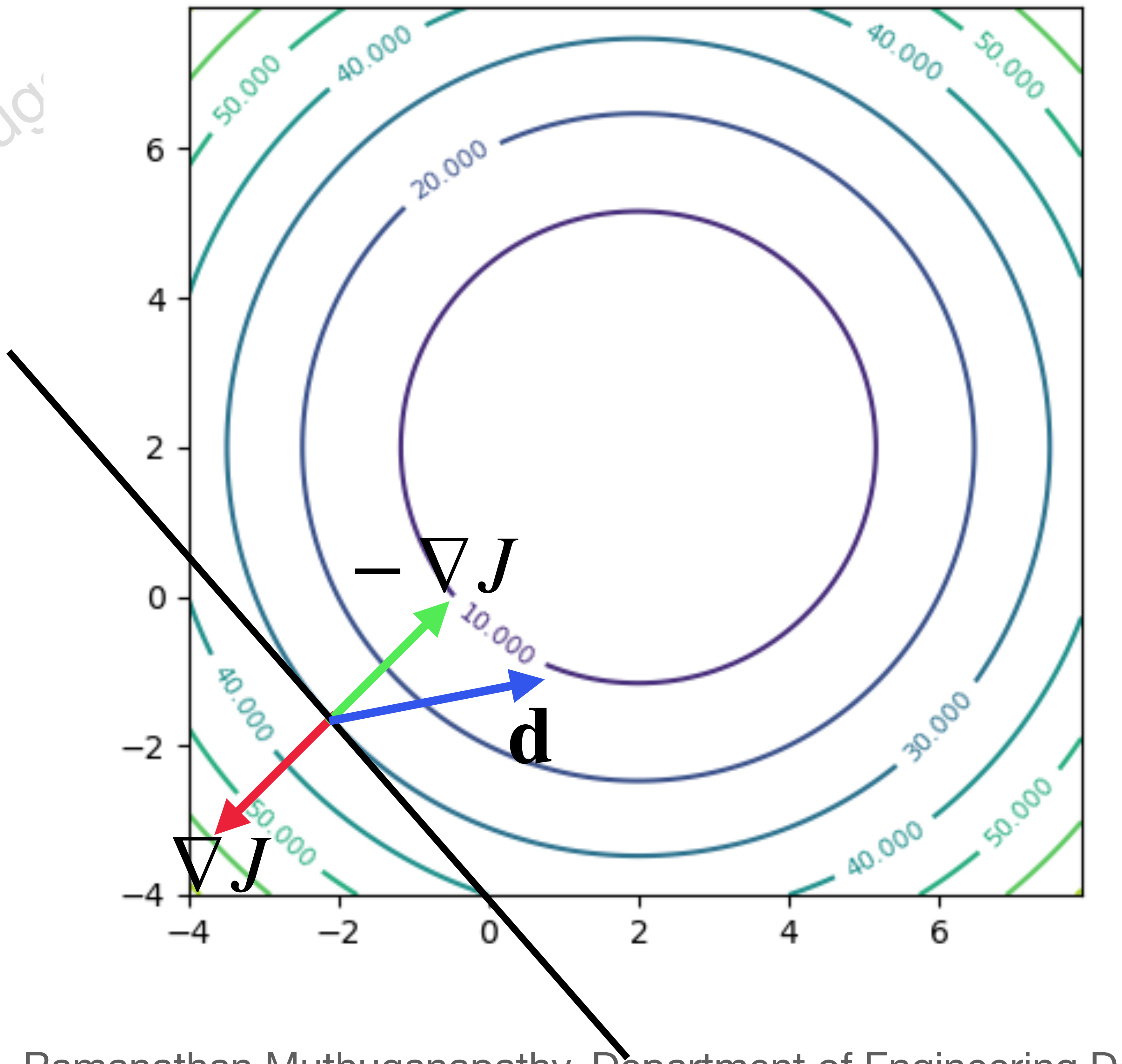
- Update w's
  - $w_1^{k+1} = w_1^k - \alpha_k \nabla J$
  - $w_2^{k+1} = w_2^k - \alpha_k \nabla J$
  - Compute $J, -\nabla J$ at w^*_{k+1}.
- Finding α_k
  - Unidirectional search (or)
  - Make it a constant (Learning rate in ML)



Algorithm - Stopping criteria

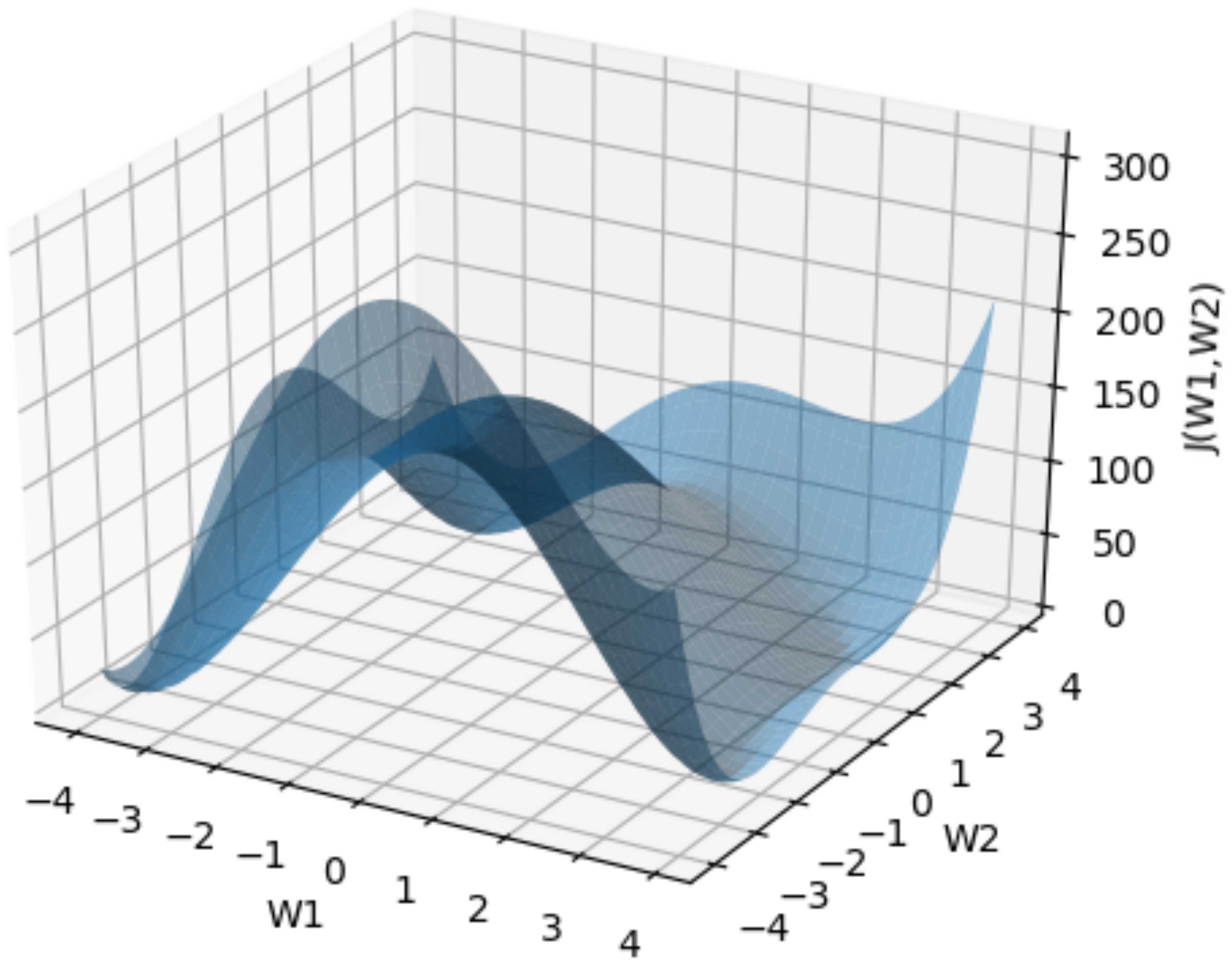
1. if $||\nabla J(w_k^*)|| \leq \epsilon_1$
2. if $|\nabla J(w_{k+1}^*) \, . \, \nabla J(w_k^*)| \leq \epsilon_2$
3. if $\dfrac{||w_{k+1}^* - w_k^*||}{||w_k^*||} \leq \epsilon_1$
4. if number of iterations exceeds a predefined constant ($k > 100$, say)

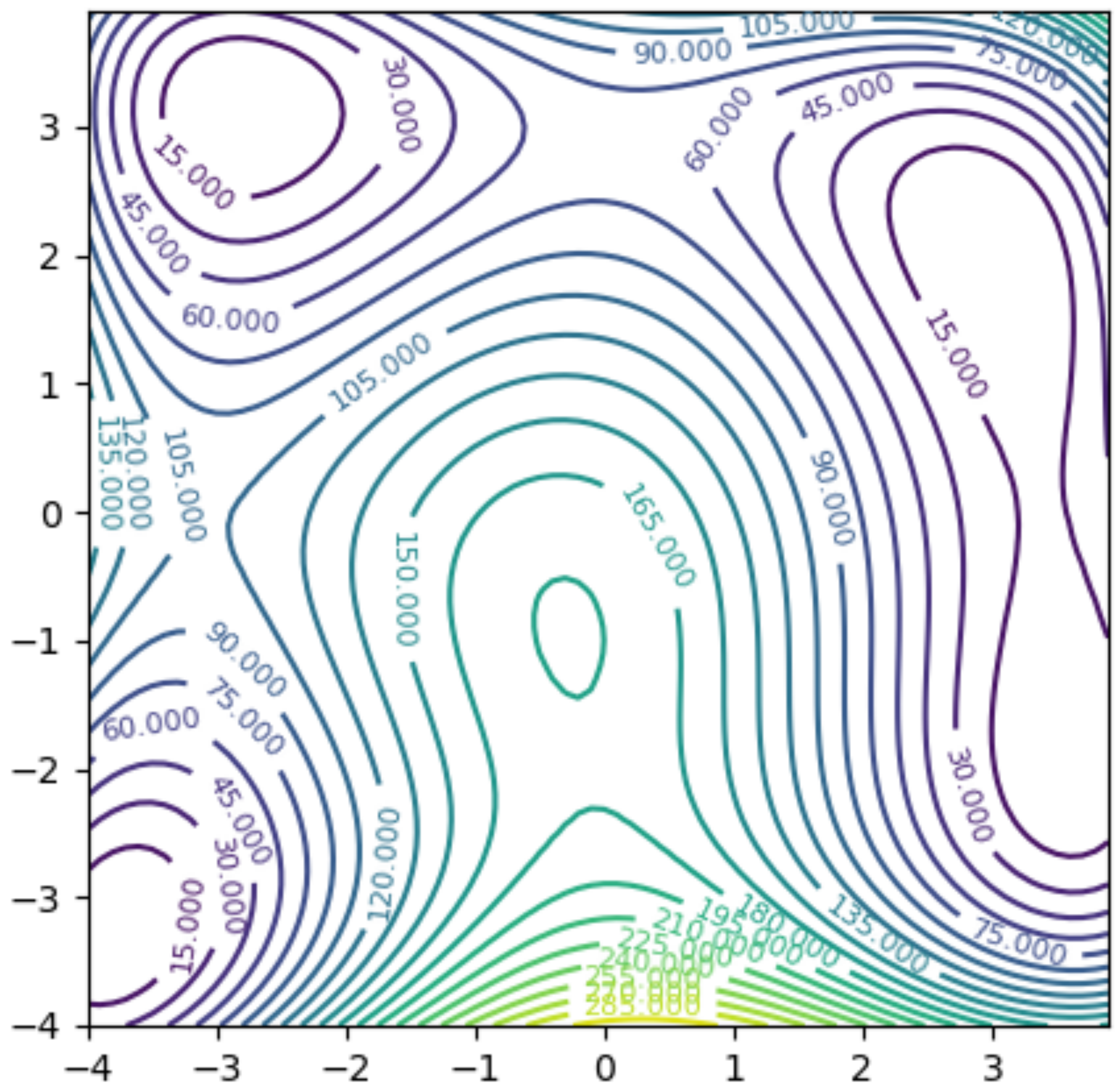
- NOTE: Compute 1 or 4 before update and 2 or 3, after



Himmelblau function

$$J(w_1, w_2) = (w_1^2 + w_2 - 11)^2 + (w_1 + w_2^2 - 7)^2$$

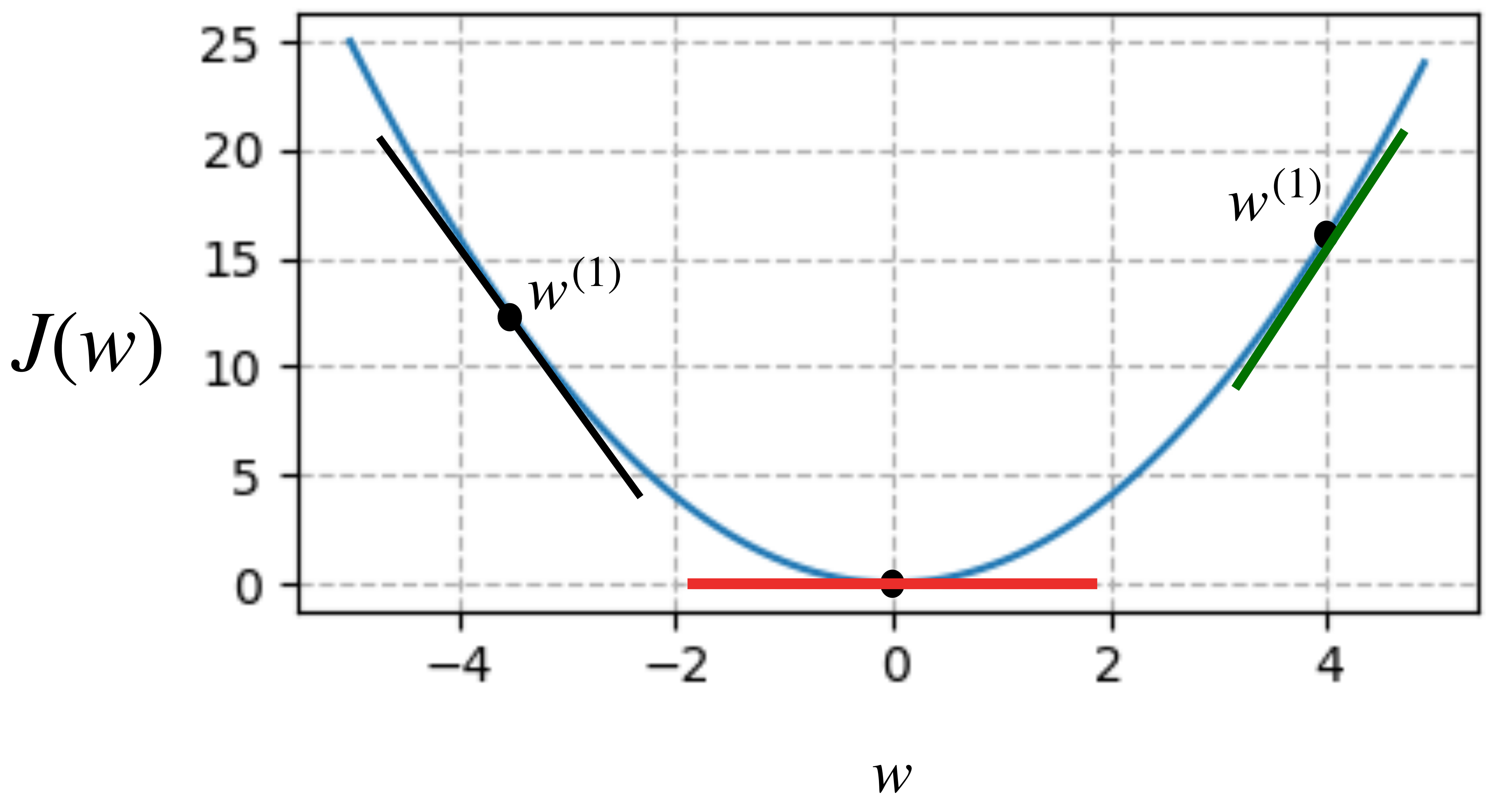


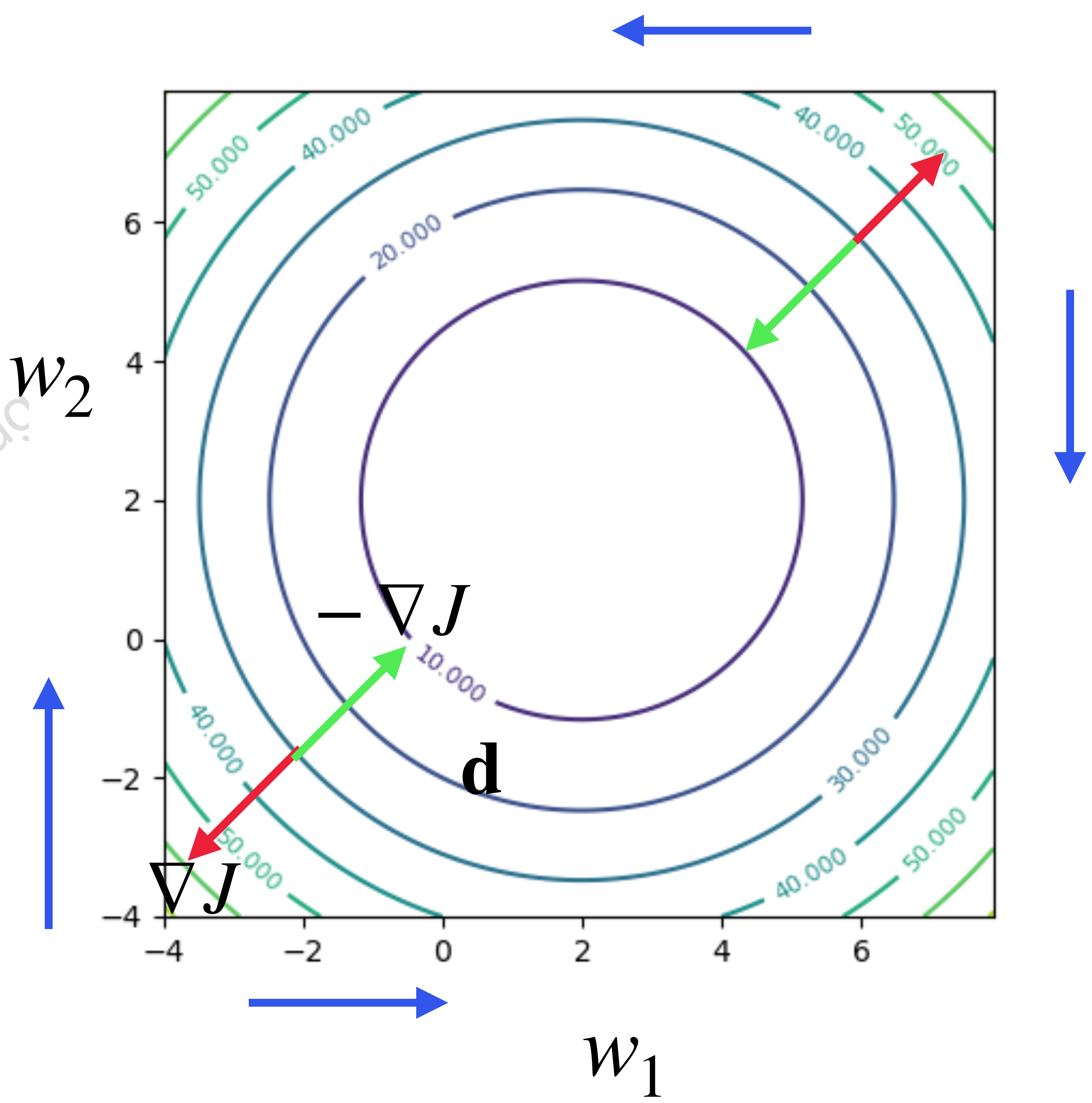


Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Recap - Single vs Multiple

$$w^*_{k+1} = w^*_k - \alpha_k \nabla J$$





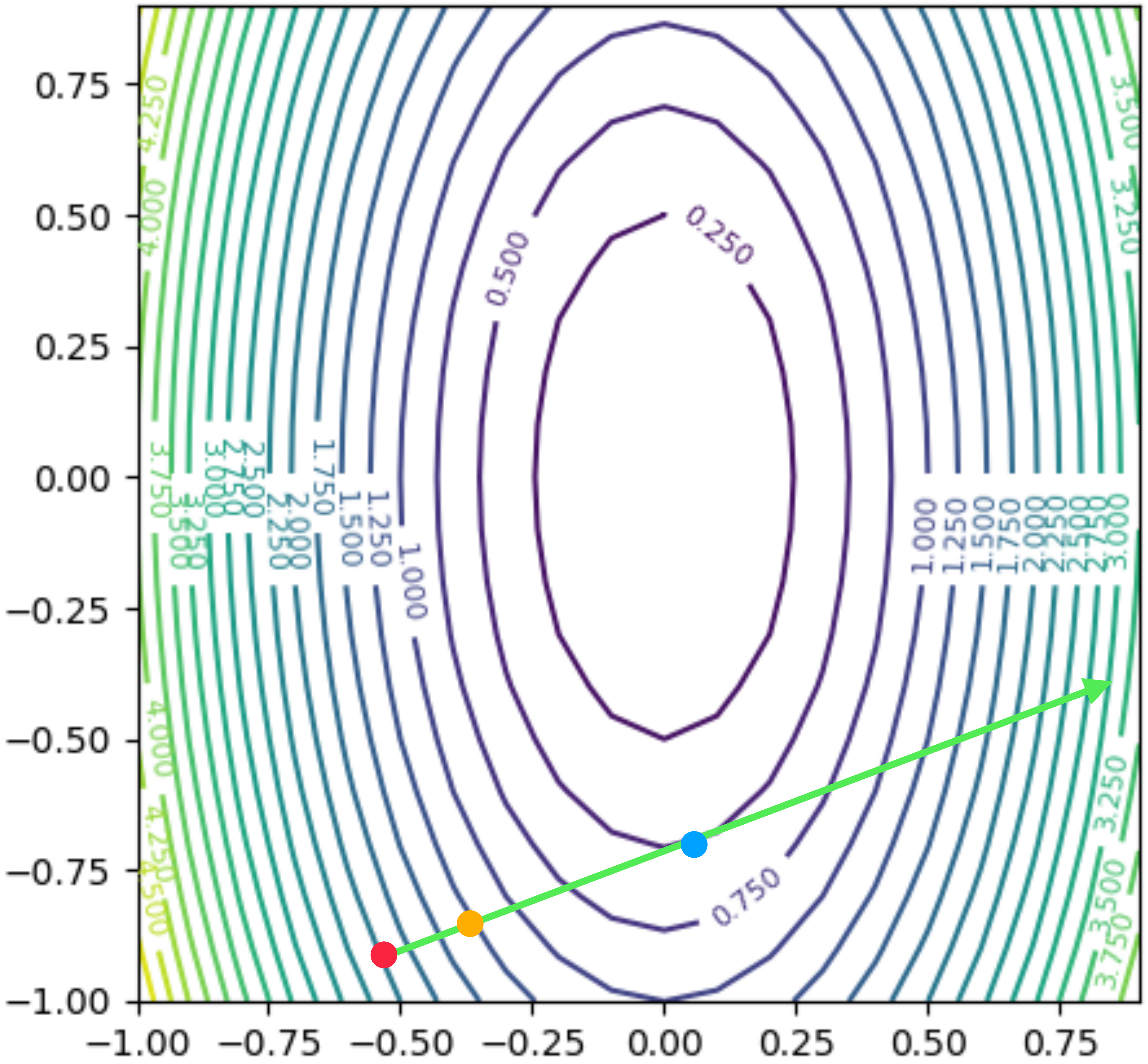
Ramanathan Muthuganapathy, Department of Engineering Design, IIT Madras

Optimization strategies

Variations in (steepest) gradient descent

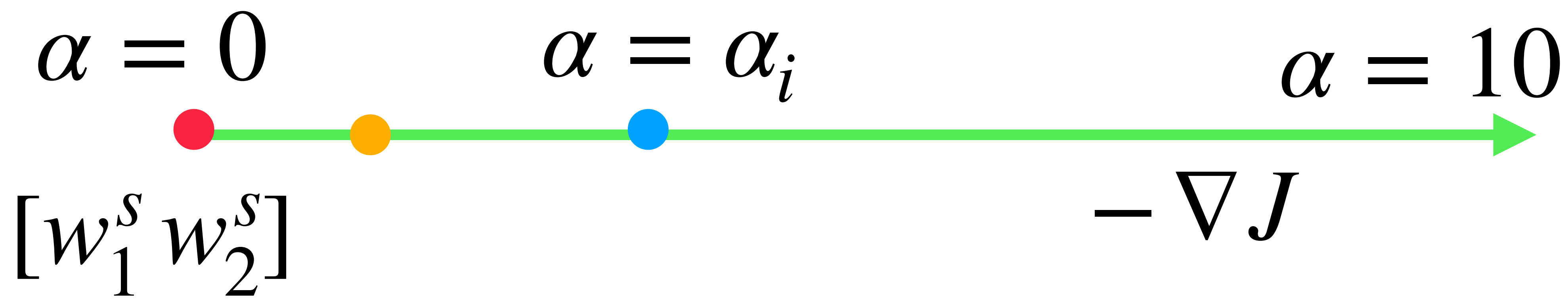
- Constant step length α
- Adaptive step length (α_k) using line search
- Stochastic gradient descent

Line Search



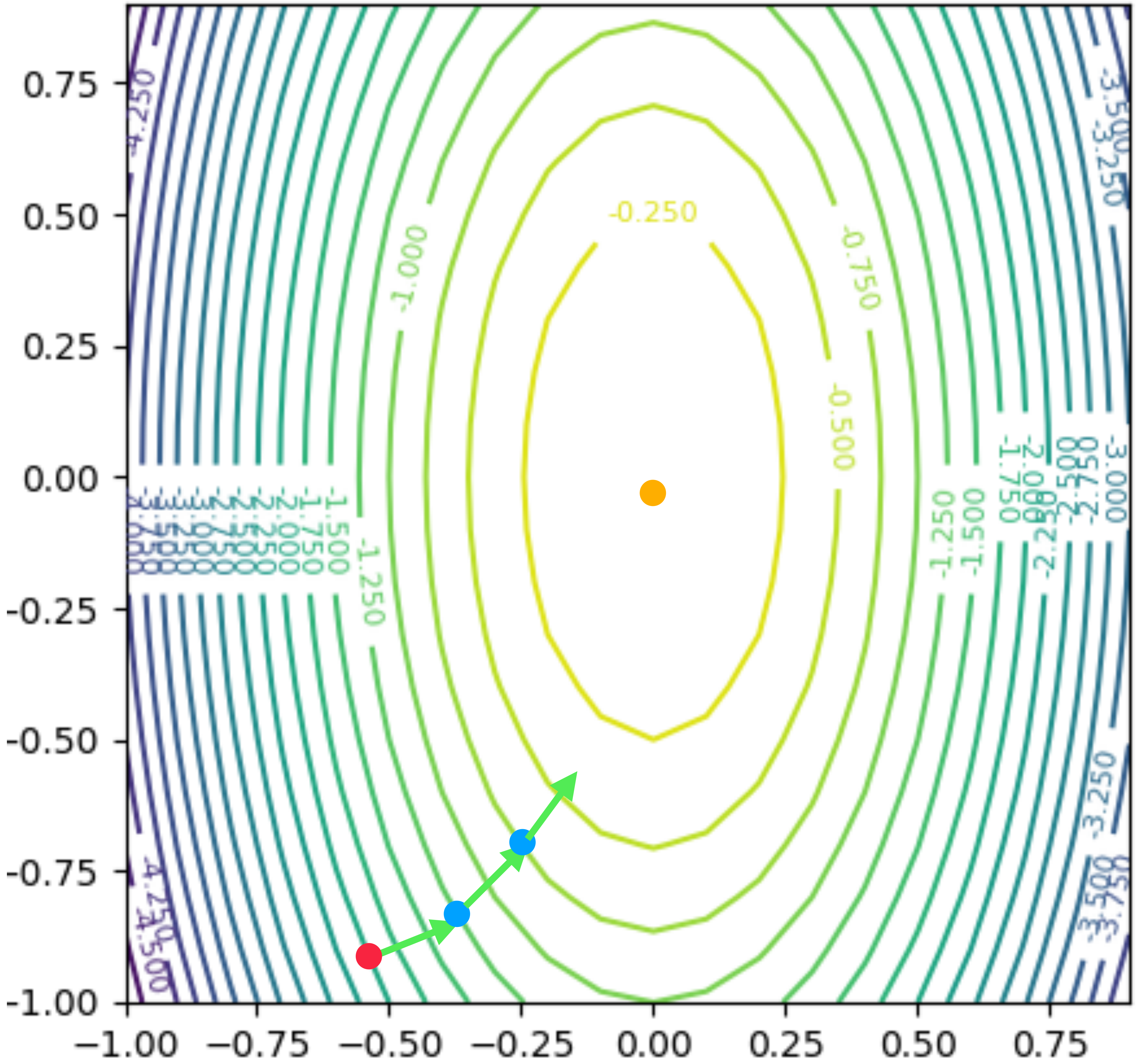
$$[w_1^* \, w_2^*] = [w_1^s \, w_2^s] + \alpha \mathbf{S}$$

$$[w_1^* \, w_2^*] = [w_1^s \, w_2^s] + \alpha(-\nabla J)$$



$$[w_1^s \, w_2^s] + \alpha(-\nabla J)$$

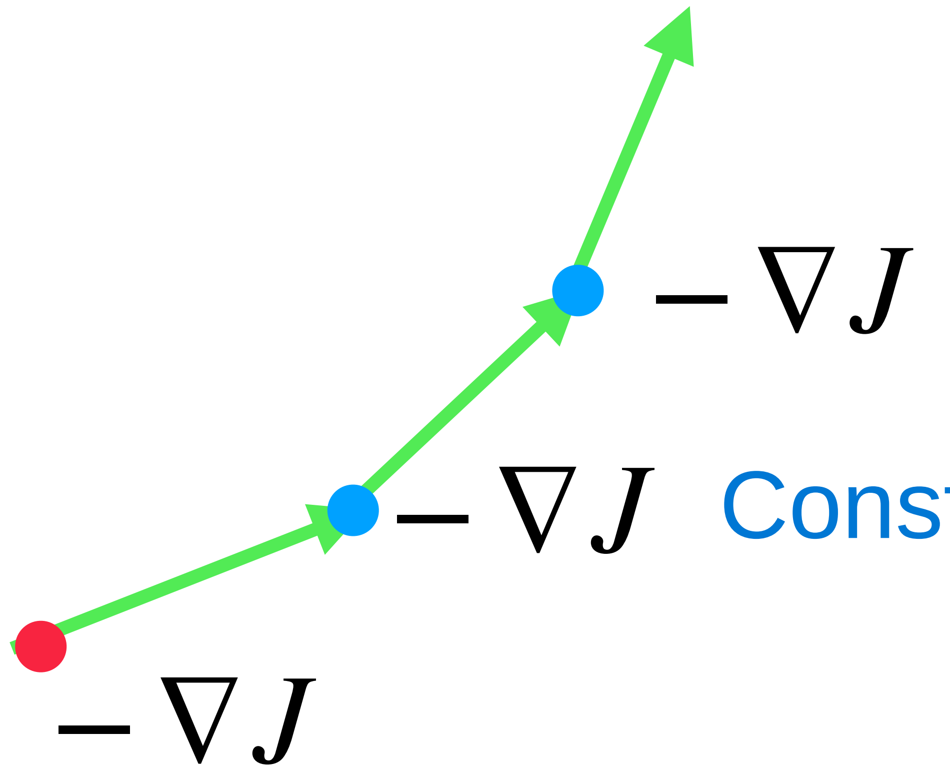
S. Grad. Des.



$[w_1^s \, w_2^s]$

$\alpha = 0.01$

alpha is no longer parameter, it is called as Hyperparameter or learning rate



Constant learning rate

Terminating criteria Del_J at minima is 0