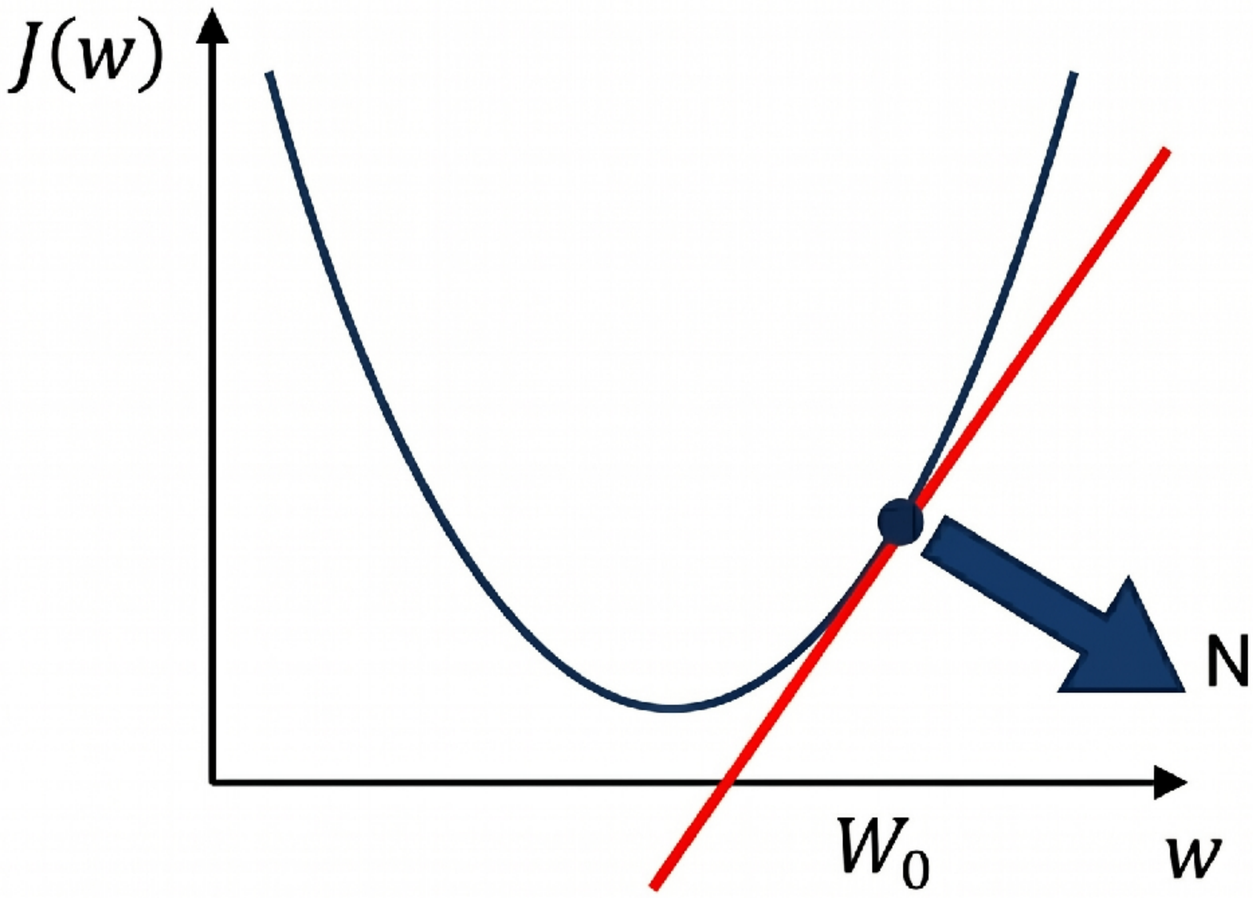
CSCI1470

Deep Learning

Randall Balestriero

Recap

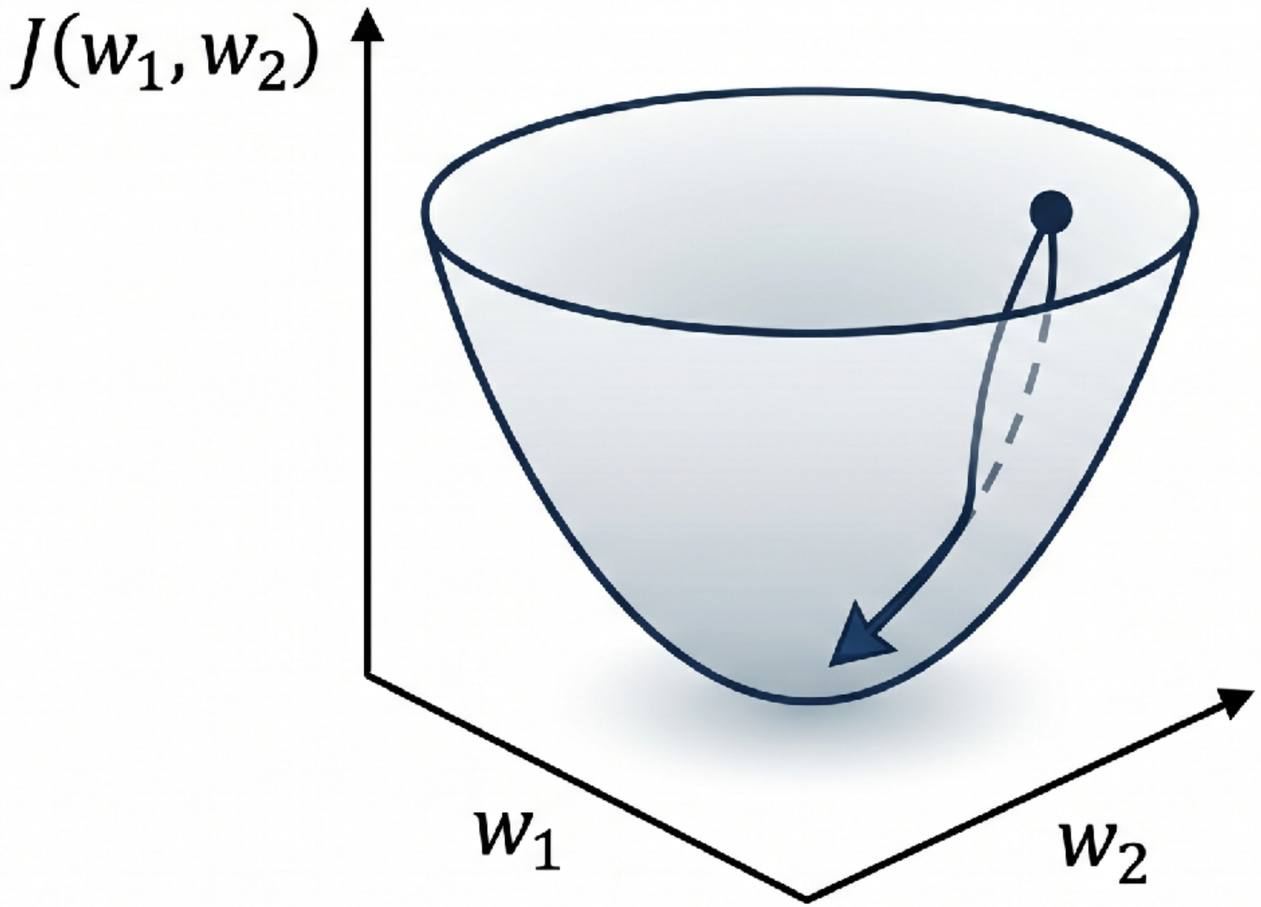
Visualizing the Gradient (2D)



Gradient at W_0 is the vector perpendincular to the contour (tangent line in 2D) pointing uphill

Negative Gradient Direction

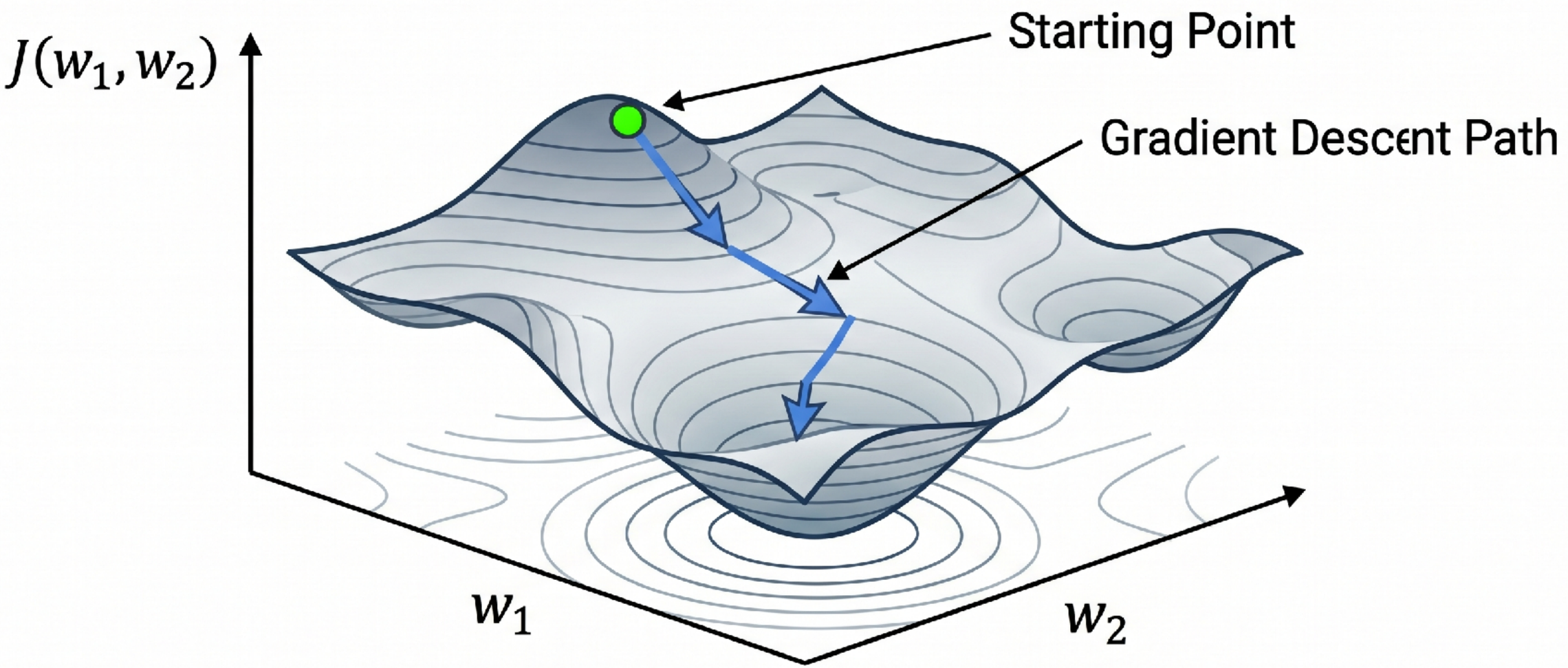
Moving to 3D Landscapes



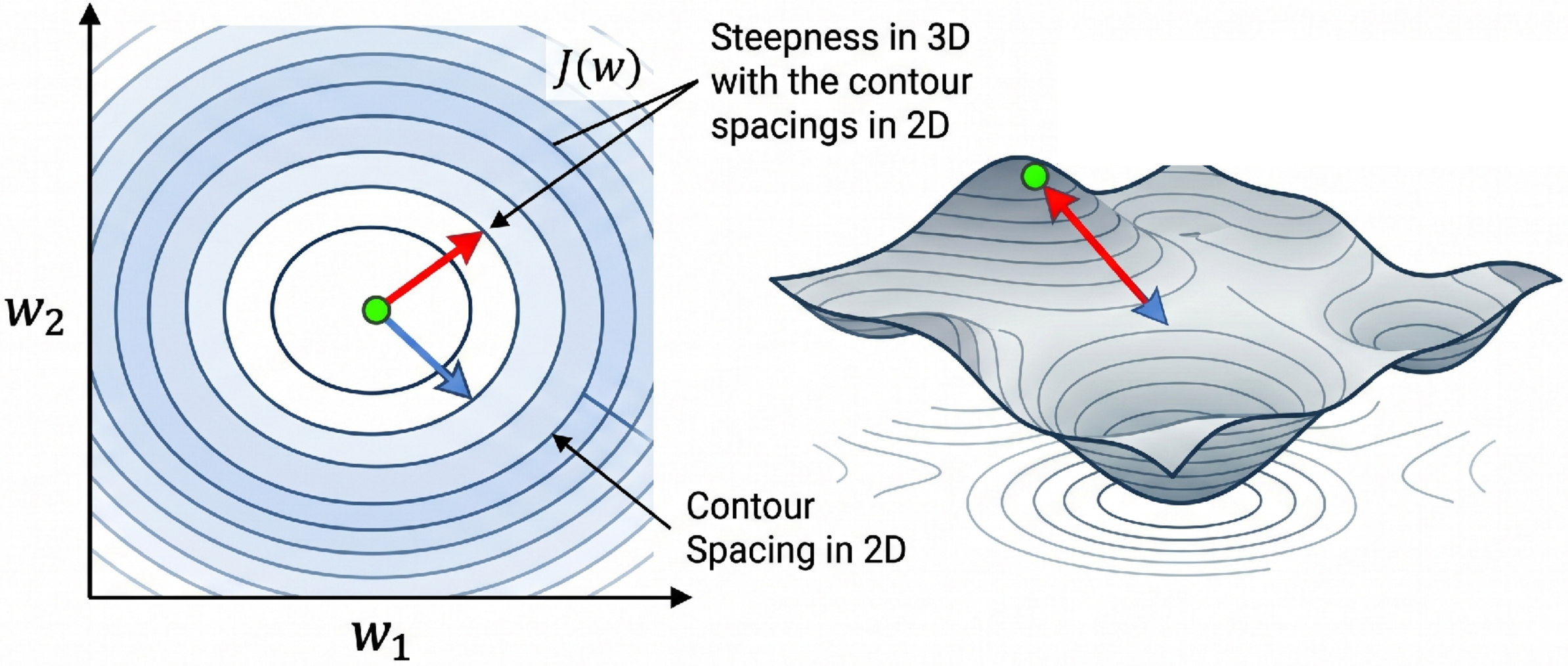
Cost Function with Two Parameters

Gradient points to the steepest descent direction from *any* point point (w_1, w_2)

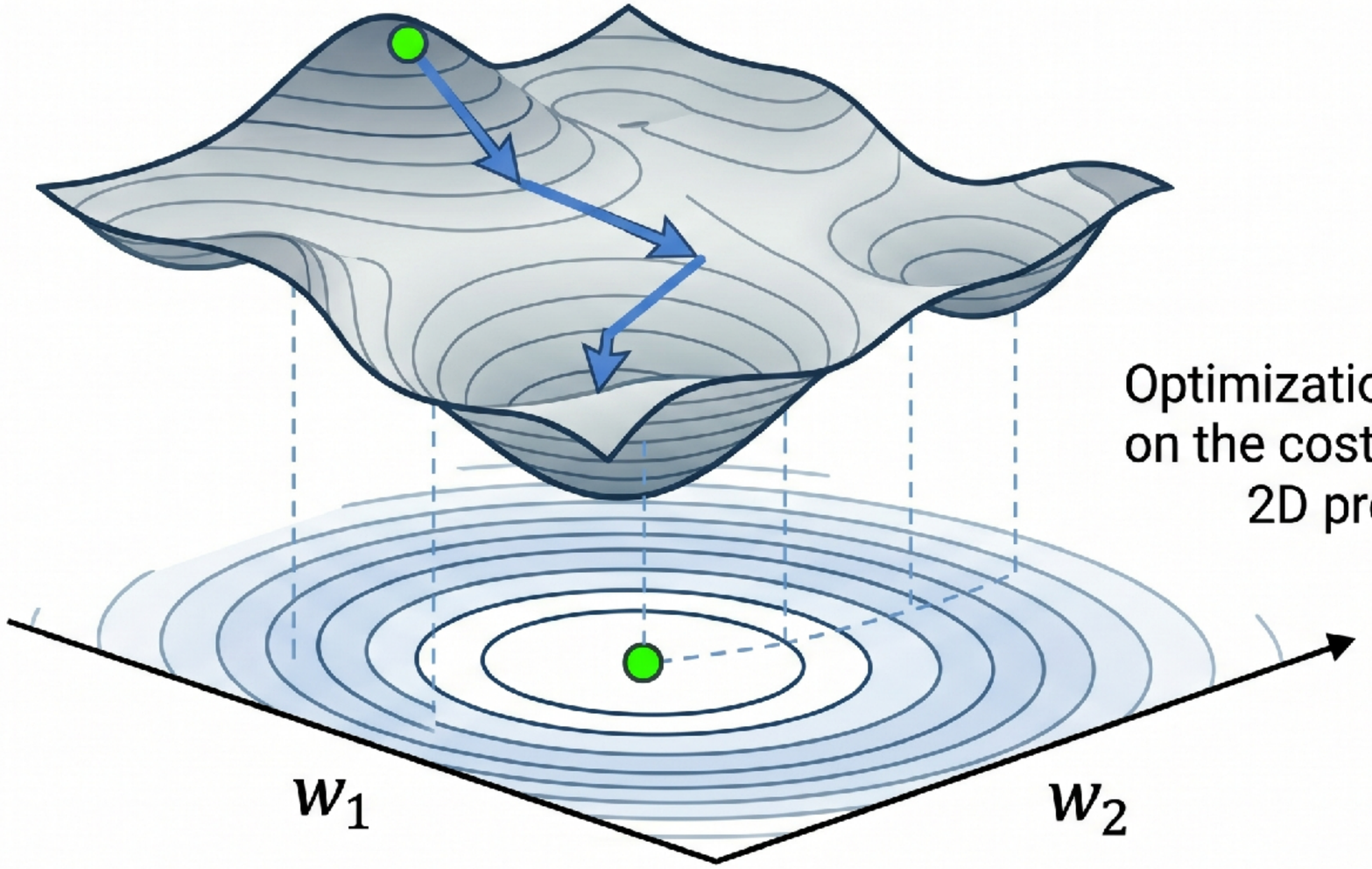
The Cost Function Surface



Understanding Contours and Gradients

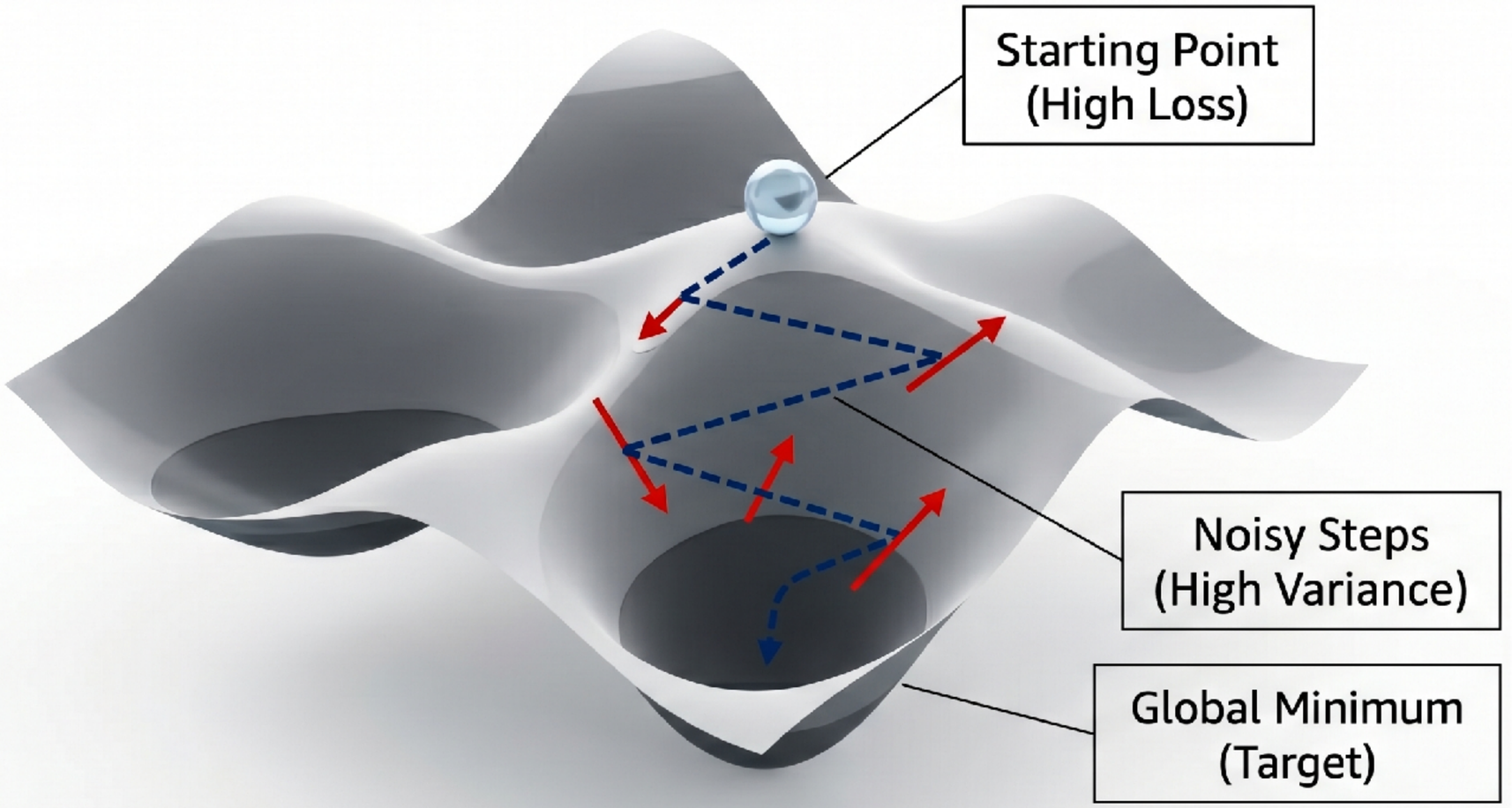


Iterative Path in 3D and Projection

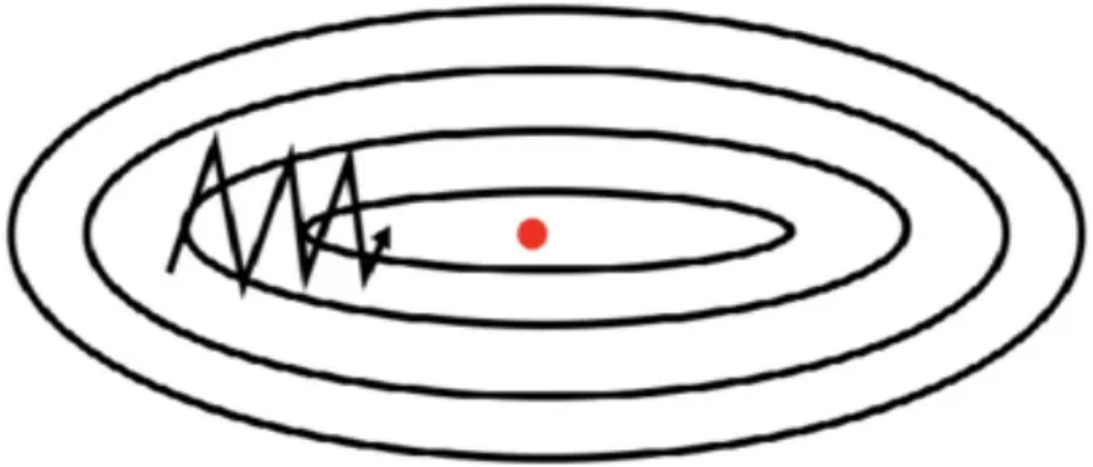


Optimization path shown on the cost surface and its 2D projection

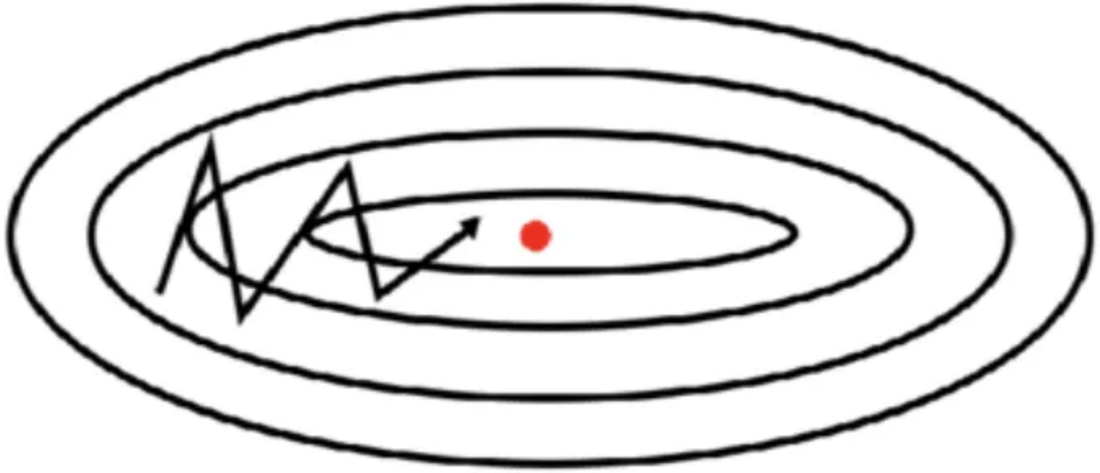
SGD: The Base Optimizer



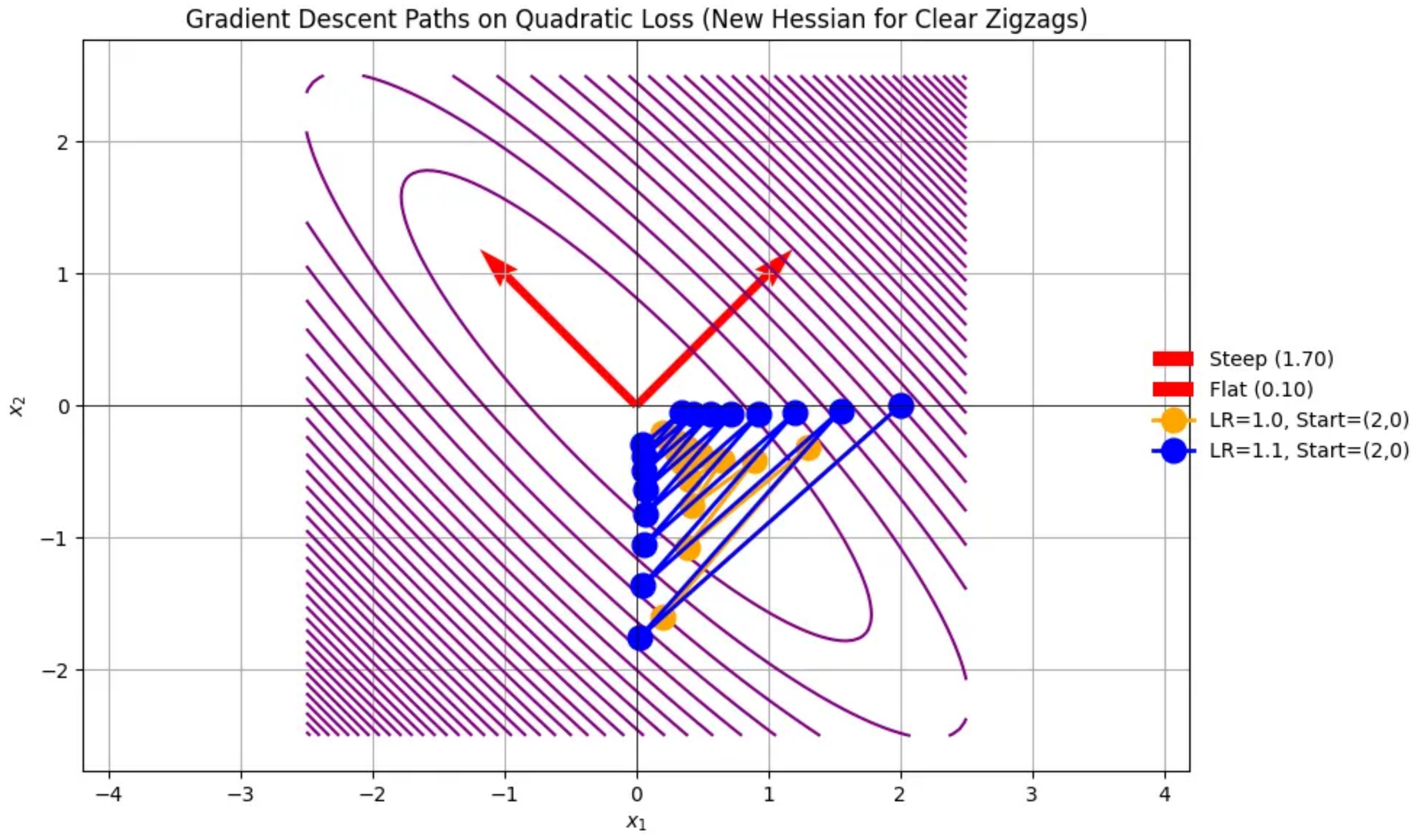
SGD without momentum



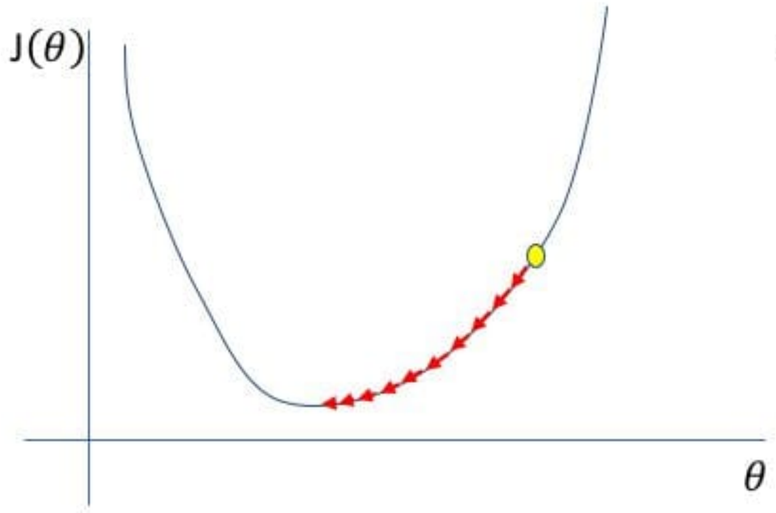
SGD with momentum



Case where gradient is “bad”?

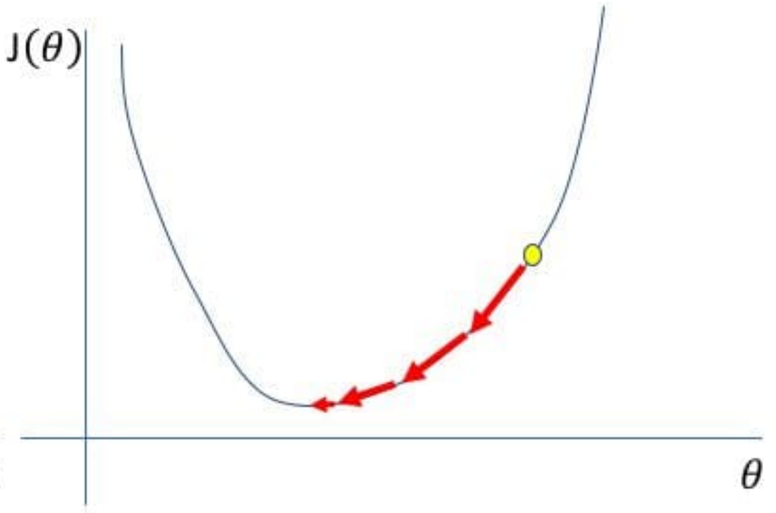

Gradient Descent Paths on Quadratic Loss (New Hessian for Clear Zigzags)
x_1
x_2
Steep (1.70)
Flat (0.10)
LR=1.0, Start=(2,0)
LR=1.1, Start=(2,0)

Too low



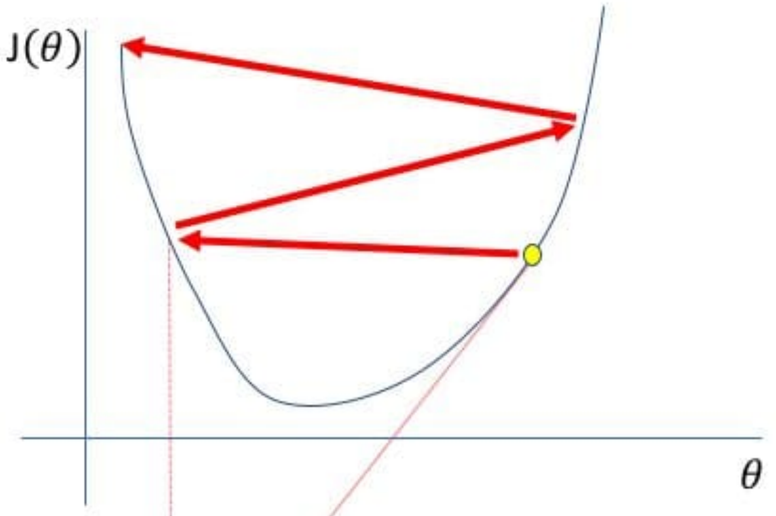
A small learning rate requires many updates before reaching the minimum point

Just right



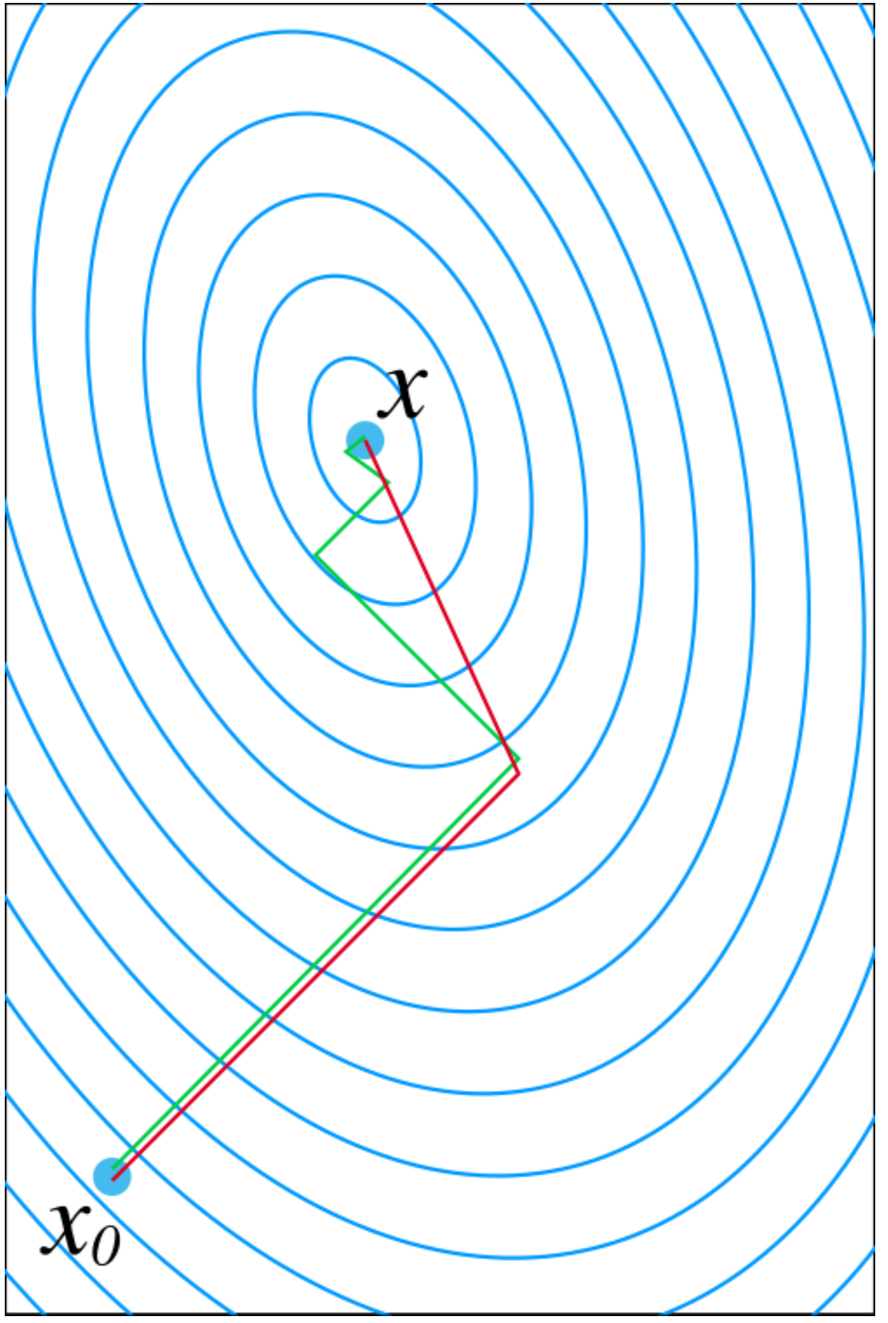
The optimal learning rate swiftly reaches the minimum point

Too high



Too large of a learning rate causes drastic updates which lead to divergent behaviors

How to resolve that?


x
x_0

Algorithm 1: *Adam*, our proposed algorithm for stochastic optimization. See section 2 for details, and for a slightly more efficient (but less clear) order of computation. g_t^2 indicates the elementwise square $g_t \odot g_t$. Good default settings for the tested machine learning problems are $\alpha = 0.001$, $\beta_1 = 0.9$, $\beta_2 = 0.999$ and $\epsilon = 10^{-8}$. All operations on vectors are element-wise. With β_1^t and β_2^t we denote β_1 and β_2 to the power t.

Require: α: Stepsize
Require: $\beta_1, \beta_2 \in [0, 1)$: Exponential decay rates for the moment estimates
Require: $f(\theta)$: Stochastic objective function with parameters θ
Require: θ_0: Initial parameter vector
  $m_0 \leftarrow 0$ (Initialize 1^{st} moment vector)
  $v_0 \leftarrow 0$ (Initialize 2^{nd} moment vector)
  $t \leftarrow 0$ (Initialize timestep)
  while θ_t not converged **do**
    $t \leftarrow t + 1$
    $g_t \leftarrow \nabla_\theta f_t(\theta_{t-1})$ (Get gradients w.r.t. stochastic objective at timestep t)
    $m_t \leftarrow \beta_1 \cdot m_{t-1} + (1 - \beta_1) \cdot g_t$ (Update biased first moment estimate)
    $v_t \leftarrow \beta_2 \cdot v_{t-1} + (1 - \beta_2) \cdot g_t^2$ (Update biased second raw moment estimate)
    $\widehat{m}_t \leftarrow m_t / (1 - \beta_1^t)$ (Compute bias-corrected first moment estimate)
    $\widehat{v}_t \leftarrow v_t / (1 - \beta_2^t)$ (Compute bias-corrected second raw moment estimate)
    $\theta_t \leftarrow \theta_{t-1} - \alpha \cdot \widehat{m}_t / (\sqrt{\widehat{v}_t} + \epsilon)$ (Update parameters)
  end while
  return θ_t (Resulting parameters)

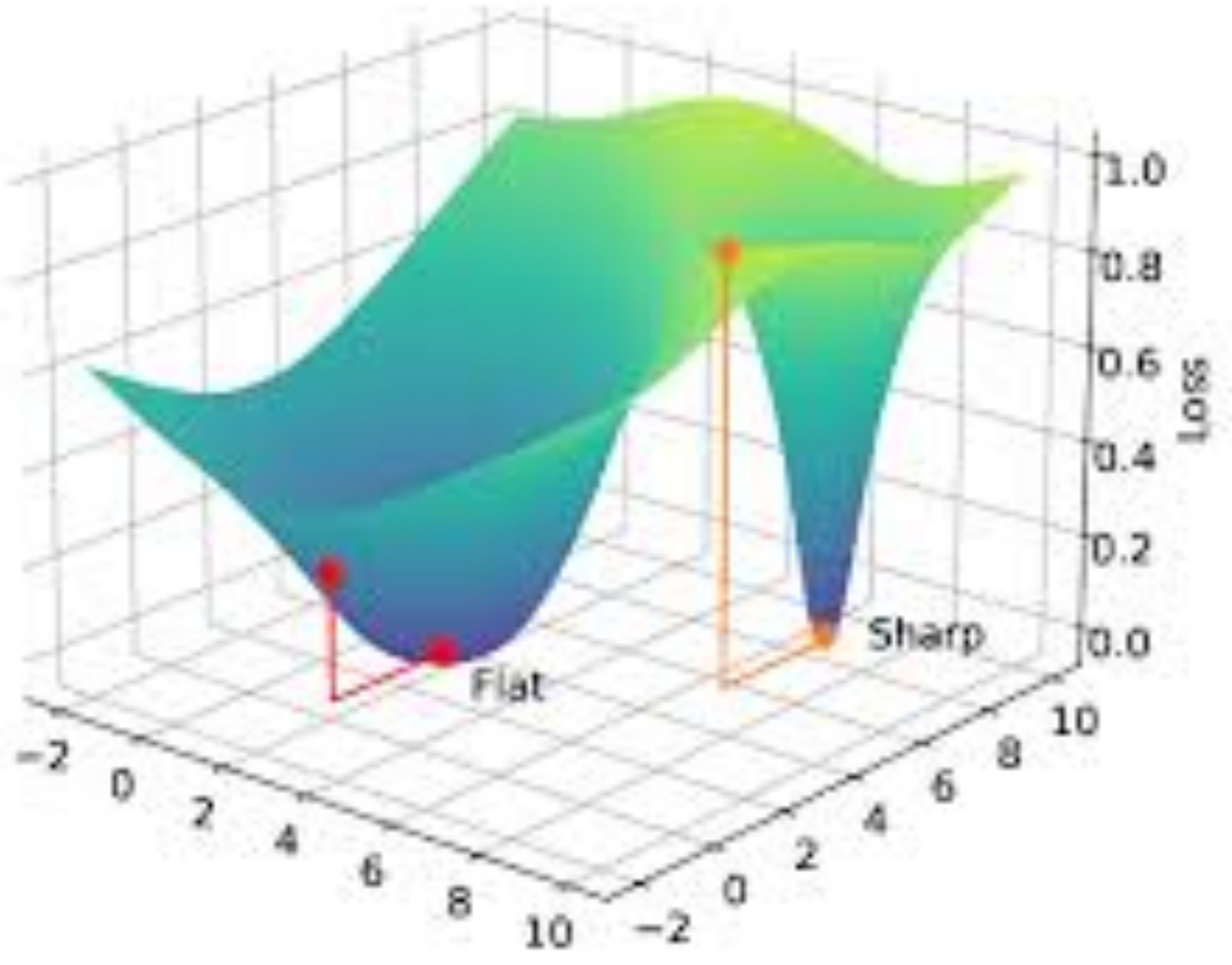
Issue with Adam?

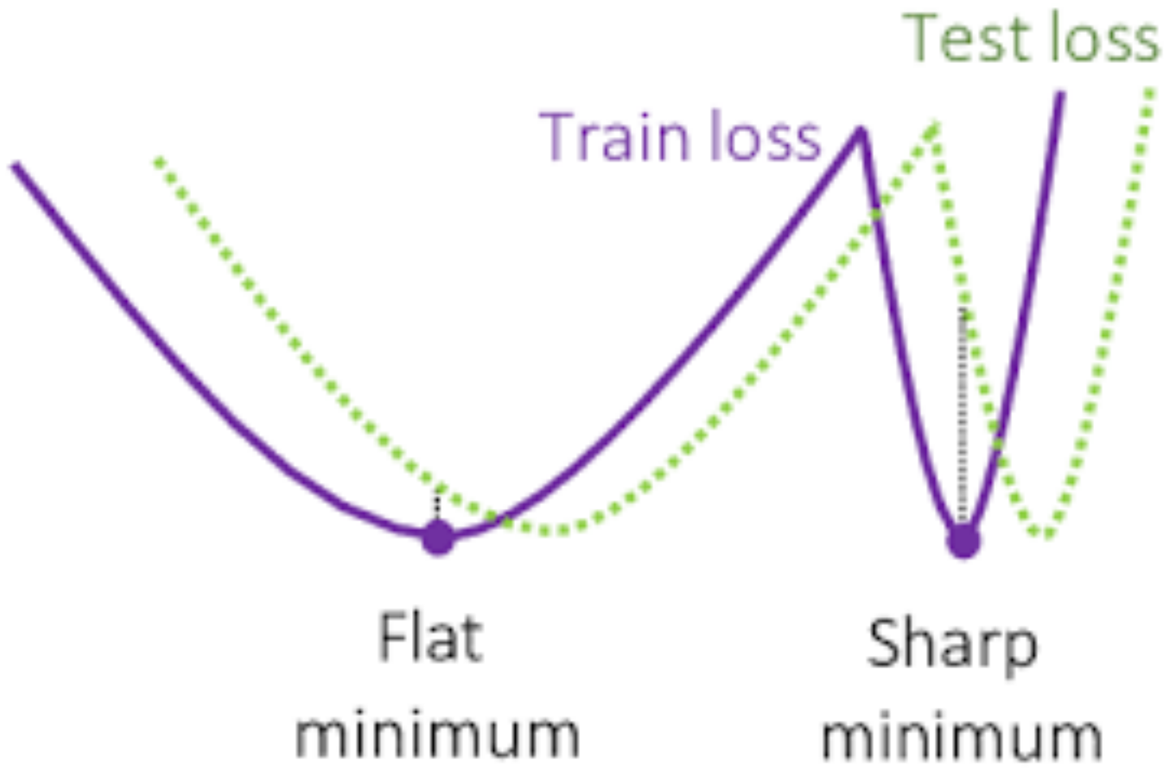
Memory! How to resolve?

Algorithm 2 Adam for a matrix parameter X with factored second moments and first moment decay parameter $\beta_1 = 0$.

Inputs: initial point $X_0 \in \mathbb{R}^{n \times m}$, step sizes $\{\alpha_t\}_{t=1}^T$, second moment decay β_2, regularization constant ϵ
Initialize $R_0 = 0$ and $C_0 = 0$
for $t = 1$ **to** T **do**
    $G_t = \nabla f_t(X_{t-1})$
    $R_t = \beta_2 R_{t-1} + (1 - \beta_2)(G_t^2)1_m$
    $C_t = \beta_2 C_{t-1} + (1 - \beta_2)1_n^\top (G_t^2)$
    $\hat{V}_t = (R_t C_t / 1_n^\top R_t)/(1 - \beta_2^t)$
    $X_t = X_{t-1} - \alpha_t G_t / (\sqrt{\hat{V}_t} + \epsilon)$
end for

Optimization for Better Generalization?


Flat
Sharp
Loss
1.0
0.8
0.6
0.4
0.2
0.0
−2
0
2
4
6
8
10


Test loss
Train loss
Flat
minimum
Sharp
minimum

the terms from the bound, we propose to select parameter values by solving the following Sharpness Aware Minimization (SAM) problem:

$$\min_{\boldsymbol{w}} L_S^{SAM}(\boldsymbol{w}) + \lambda ||\boldsymbol{w}||_2^2 \quad \text{where} \quad L_S^{SAM}(\boldsymbol{w}) \triangleq \max_{||\boldsymbol{\epsilon}||_p \leq \rho} L_S(\boldsymbol{w} + \boldsymbol{\epsilon}), \tag{1}$$

the terms from the bound, we propose to select parameter values by solving the following Sharpness Aware Minimization (SAM) problem:

$$\min_{\boldsymbol{w}} L_S^{SAM}(\boldsymbol{w}) + \lambda ||\boldsymbol{w}||_2^2 \quad \text{where} \quad L_S^{SAM}(\boldsymbol{w}) \triangleq \max_{||\boldsymbol{\epsilon}||_p \leq \rho} L_S(\boldsymbol{w} + \boldsymbol{\epsilon}), \tag{1}$$

Remind you of anything?

Thank you!
See you Monday!