

Discussion 6

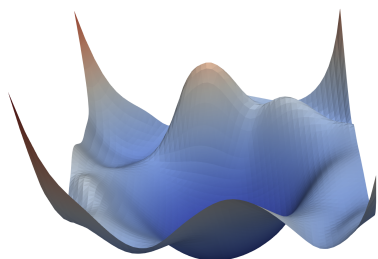


fig: A neural network's loss surface
(source: University of Maryland)

This week's resources:

– CMU Computer Vision Lecture: Visualizing Quadratics

[link](#)

– 3Blue1Brown: Gradient descent, how neural networks learn

[video](#)

1 Taylor Expansion and Quadratic Forms

In Lecture 13, we saw that the local behavior of an error surface $E(\mathbf{w})$ around a stationary point $\mathbf{w}^*$ can be approximated using its second-order Taylor expansion:

$$\tilde{E}_{\mathbf{w}^*}(\mathbf{w}) = E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^\top \mathbf{H}(\mathbf{w} - \mathbf{w}^*),$$

where $\mathbf{H} = \nabla^2 E(\mathbf{w}^*)$ is the Hessian matrix describing the local curvature. The eigenvalues and eigenvectors of $\mathbf{H}$ determine the curvature along each principal direction and whether $\mathbf{w}^*$ is a minimum, maximum, or saddle point.

Consider the quadratic function:

$$E(\mathbf{w}) = \frac{1}{2}w_1^2 + 2w_2^2 - 2w_2 + 1.$$

(a) Find the stationary point $\mathbf{w}^*$ of $E(\mathbf{w})$.

Hint: Compute the gradient $\nabla E(\mathbf{w}) = \begin{bmatrix} \frac{\partial E}{\partial w_1} \\ \frac{\partial E}{\partial w_2} \end{bmatrix}$ and set it equal to zero.

Solution: Computing the partial derivatives:

$$\frac{\partial E}{\partial w_1} = w_1, \quad \frac{\partial E}{\partial w_2} = 4w_2 - 2.$$

Setting $\nabla E = \mathbf{0}$ gives

$$\begin{cases} w_1 = 0, \\ 4w_2 - 2 = 0 \end{cases} \implies w_2 = \frac{1}{2}.$$

Thus, the stationary point is:

$$\mathbf{w}^* = \begin{bmatrix} 0 \\ 1/2 \end{bmatrix}.$$

- (b) Compute the Hessian matrix $\mathbf{H} = \nabla^2 E(\mathbf{w})$ and determine its eigenvalues λ_1, λ_2 and corresponding eigenvectors $\mathbf{u}_1, \mathbf{u}_2$. Based on the signs of the eigenvalues, determine whether $\mathbf{w}^*$ is a local minimum, local maximum, or saddle point.

Hint: The Hessian is the matrix of second partial derivatives: $H_{ij} = \frac{\partial^2 E}{\partial w_i \partial w_j}$.

Solution: The Hessian is:

$$\mathbf{H} = \begin{bmatrix} \frac{\partial^2 E}{\partial w_1^2} & \frac{\partial^2 E}{\partial w_1 \partial w_2} \\ \frac{\partial^2 E}{\partial w_2 \partial w_1} & \frac{\partial^2 E}{\partial w_2^2} \end{bmatrix} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}.$$

Its eigenvalues and eigenvectors are:

$$\lambda_1 = 1, \quad \mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_2 = 4, \quad \mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}.$$

Since both eigenvalues are positive, the quadratic surface is **locally convex** at $\mathbf{w}^*$.

$\mathbf{w}^*$ is a local minimum.

- (c) Use the eigen-decomposition of the Hessian to express the 2nd-order Taylor expansion of $E(\mathbf{w})$ around $\mathbf{w}^*$ in the eigenvector basis.

Solution: Near a stationary point $\mathbf{w}^*$, the second-order Taylor approximation of the error surface is

$$\tilde{E}_{\mathbf{w}^*}(\mathbf{w}) = E(\mathbf{w}^*) + \frac{1}{2}(\mathbf{w} - \mathbf{w}^*)^\top \mathbf{H}(\mathbf{w} - \mathbf{w}^*),$$

where $\mathbf{H}$ is the Hessian evaluated at $\mathbf{w}^*$.

We diagonalize the Hessian using its eigen-decomposition:

$$\mathbf{H} = \mathbf{U} \mathbf{\Lambda} \mathbf{U}^\top, \quad \text{where} \quad \mathbf{H} \mathbf{u}_i = \lambda_i \mathbf{u}_i.$$

Let $\boldsymbol{\alpha} = \mathbf{U}^\top(\mathbf{w} - \mathbf{w}^*)$ denote the coordinates of $\mathbf{w}$ in the eigenvector basis. Then

$$(\mathbf{w} - \mathbf{w}^*)^\top \mathbf{H}(\mathbf{w} - \mathbf{w}^*) = \boldsymbol{\alpha}^\top \mathbf{\Lambda} \boldsymbol{\alpha} = \sum_i \lambda_i \alpha_i^2.$$

Thus, the quadratic form simplifies to:

$$\tilde{E}_{\mathbf{w}^*}(\boldsymbol{\alpha}) = E(\mathbf{w}^*) + \frac{1}{2} \sum_i \lambda_i \alpha_i^2.$$

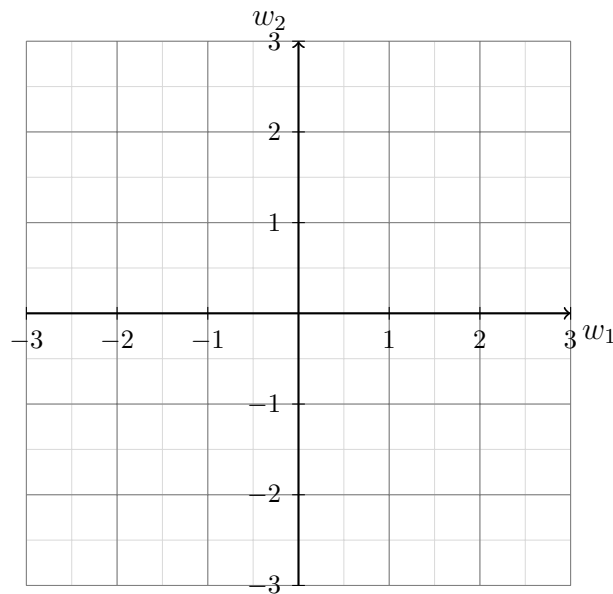
For this example,

$$\mathbf{H} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}, \quad \mathbf{U} = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix}, \quad \mathbf{\Lambda} = \begin{bmatrix} 1 & 0 \\ 0 & 4 \end{bmatrix}.$$

Substituting gives

$$\tilde{E}_{\mathbf{w}^*}(\boldsymbol{\alpha}) = \frac{1}{2} + \frac{1}{2}\alpha_1^2 + 2\alpha_2^2.$$

- (d) Sketch the isocontours of $E(\mathbf{w})$ and mark $\mathbf{w}^*$. Indicate the eigenvector directions $\mathbf{u}_1$ and $\mathbf{u}_2$, labeling which has the greater curvature (larger λ_i).



Solution: The isocontours of $E(\mathbf{w})$ are ellipses centered at

$$\mathbf{w}^* = \begin{bmatrix} 0 \\ \frac{1}{2} \end{bmatrix}.$$

From part (b), eigen-decomposition of the Hessian gives:

$$\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix}, \quad \lambda_1 = 1; \quad \mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix}, \quad \lambda_2 = 4.$$

- The direction of $\mathbf{u}_1$ (horizontal) corresponds to the smaller eigenvalue $\lambda_1 = 1$, indicating the **least curvature**. This forms the **major axis** of the ellipses.
- The direction of $\mathbf{u}_2$ (vertical) corresponds to the larger eigenvalue $\lambda_2 = 4$, indicating the **greatest curvature**. This forms the **minor axis** of the ellipses.

Hence, the ellipses are elongated in the horizontal direction ($\mathbf{u}_1$) and compressed in the vertical direction ($\mathbf{u}_2$).

2 Momentum vs. SGD

Consider a loss/error function $E(\mathbf{w})$ with parameters $\mathbf{w} \in \mathbb{R}^D$. Recall the stochastic gradient descent (SGD) update rule is

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} - \eta \nabla E(\mathbf{w}^{(\tau-1)}),$$

where $\eta > 0$ is the learning rate and τ is the iteration number (starting at 0).

We saw in lecture that we can improve upon vanilla SGD by adding momentum, such that

$$\Delta \mathbf{w}^{(\tau-1)} = -\eta \nabla E(\mathbf{w}^{(\tau-1)}) + \boxed{\mu \Delta \mathbf{w}^{(\tau-2)}},$$

$$\mathbf{w}^{(\tau)} = \mathbf{w}^{(\tau-1)} + \Delta \mathbf{w}^{(\tau-1)},$$

where $0 \leq \mu < 1$ is the momentum parameter and $\Delta \mathbf{w}^{(0)} = 0$.

- (a) In a nearly flat region where the gradient direction stays almost constant, how does momentum affect learning compared to plain SGD?
- (i) It slows learning down to prevent overshooting.
 - (ii) It keeps the same pace as SGD.
 - (iii) It speeds learning up by accumulating updates in the same direction.
 - (iv) It randomly perturbs the updates to explore more directions.

Solution:(iii): In flat regions, the gradient direction barely changes. Each step reinforces the previous one, so momentum “builds speed” along that direction, effectively increasing how far the parameters move each iteration.

Example: Consider a nearly flat loss surface $E(w) = \frac{1}{200}(w-5)^2$ and start from $w_0 = 0$ with $\eta = 0.1$. Then $\nabla E(w) = \frac{1}{100}(w-5)$, which changes very little as w moves.

- SGD: $w_1 = 0.005$, $w_2 = 0.010$, $w_3 = 0.015, \dots$
- Momentum ($\mu = 0.9$): $w_1 = 0.005$, $w_2 = 0.0145$, $w_3 = 0.0280, \dots$

Momentum increases the rate of convergence toward $w = 5$ by adding up past gradients.

- (b) In a steep or highly curved region where the gradient direction alternates sign each step (causing oscillations), how does momentum affect the updates?
- (i) It amplifies oscillations by adding more energy to the updates.
 - (ii) It cancels out part of each new step, smoothing and stabilizing motion.
 - (iii) It leaves the oscillations unchanged.
 - (iv) It reverses the update direction completely.

Solution:(ii): Momentum blends the oscillatory directions over time, causing partial cancellation. This reduces the size of the back-and-forth motion, helping the optimizer move more smoothly.

Example: Consider a narrow valley

$$E(w_1, w_2) = w_1^2 + 10w_2^2,$$

with learning rate $\eta = 0.1$. The gradient is

$$\nabla E = (2w_1, 20w_2),$$

so the slope is gentle in w_1 but steep in w_2 . Starting from $(w_1, w_2) = (1, 1)$:

- **SGD:**

$$w_2^{(1)} = 1 - 0.1(20)(1) = -1.0,$$

$$w_2^{(2)} = -1.0 - 0.1(20)(-1.0) = 1.0,$$

$$w_2^{(3)} = -1.0, \quad \text{and so on.}$$

The w_2 coordinate oscillates between 1 and -1 and will not converge.

- **Momentum ($\mu = 0.9$):**

$$\Delta w_2^{(1)} = -0.1 \cdot 20 \cdot 1 + 0.9 \cdot 0 = -2.0,$$

$$w_2^{(1)} = 1 + \Delta w_2^{(1)} = 1 - 2 = -1.0.$$

$$\Delta w_2^{(2)} = -0.1 \cdot 20 \cdot (-1.0) + 0.9 \cdot (-2.0) = 2.0 - 1.8 = 0.2,$$

$$w_2^{(2)} = -1.0 + 0.2 = -0.8.$$

$$\Delta w_2^{(3)} = -0.1 \cdot 20 \cdot (-0.8) + 0.9 \cdot 0.2 = 1.6 + 0.18 = 1.78,$$

$$w_2^{(3)} = -0.8 + 1.78 = 0.98.$$

Now the w_2 sequence still oscillates between positive and negative values but with damped amplitude, so it will eventually converge (albeit still slowly).