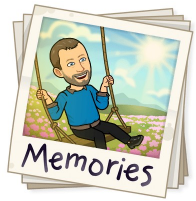


Machine Learning

Textbook: Machine Learning Refined, by Watt, Borhani, and Katsaggelos, 2nd edition:
MLRefined.com

Previous Lecture

- ◆ For a quadratic, gradient descent can be analyzed by tracking the error from the minimum.
- ◆ Convergence requires the error multiplier to have magnitude less than 1.
- ◆ Small learning rates can converge slowly; larger stable rates may oscillate but converge faster.
- ◆ If the learning rate is too large, gradient descent can diverge.
- ◆ Different curvature in different directions can force a small step size and slow convergence.
- ◆ A line search chooses the step size adaptively rather than fixing one value forever.



§4.1, 4.2 - Second-Order Approximation

Last time:

curvature controlled the behavior of gradient descent.

Steep directions forced smaller stable step sizes.

But we described curvature only indirectly.

Today:

use second derivatives to measure curvature directly.

Start Again in One Variable

First-order approximation near w_0 :

$$g(w_0 + \Delta w) \approx g(w_0) + g'(w_0)\Delta w.$$

A line approximates the graph locally.

Add one more Taylor term:

$$g(w_0 + \Delta w) \approx g(w_0) + g'(w_0)\Delta w + \frac{1}{2}g''(w_0)(\Delta w)^2.$$

This is the **second-order approximation**.

Line $\rightarrow$ parabola.

What Does the Second Derivative Tell Us?

If

$g''(w_0) > 0$,
the graph bends upward near w_0 .

If

$g''(w_0) < 0$,
the graph bends downward.

If $|g''(w_0)|$
is large,
curvature is strong.

The first derivative gives slope.

The second derivative gives curvature.

A Nonlinear Example

Let $g(w) = e^w$,
and approximate near $w_0 = 0$.

We have
 $g(0) = 1$,
 $g'(0) = 1$,
 $g''(0) = 1$.

First order:
 $e^{\Delta w} \approx 1 + \Delta w$.

Second order:
 $e^{\Delta w} \approx 1 + \Delta w + \frac{1}{2}(\Delta w)^2$.

For $\Delta w = 0.2$:
first order: 1.2,
second order: 1.22,
exact: $e^{0.2} \approx 1.2214$.

Second-order information improves the local model.

For a Quadratic, Second Order Is Exact

Let $g(w) = (w - 3)^2$.

Then $g''(w) = 2$.

There are no third- or higher-order terms.

Therefore the second-order Taylor model is not an approximation:
it is the function exactly.

This is why quadratics are especially useful in optimization.

Now Several Parameters

Suppose
 $g = g(w_1, w_2).$

First derivatives were collected into the gradient:

$$\nabla g = \begin{bmatrix} g_{w_1} \\ g_{w_2} \end{bmatrix}.$$

Now collect all second partial derivatives:

$$H(\mathbf{w}) = \begin{bmatrix} \frac{\partial^2 g}{\partial w_1^2} & \frac{\partial^2 g}{\partial w_1 \partial w_2} \\ \frac{\partial^2 g}{\partial w_2 \partial w_1} & \frac{\partial^2 g}{\partial w_2^2} \end{bmatrix}.$$

This matrix is the **Hessian**.

The Hessian in d Variables

For
 $\mathbf{w} = (w_1, \dots, w_d),$

the Hessian is the $d \times d$ matrix

$$H_{ij}(\mathbf{w}) = \frac{\partial^2 g}{\partial w_i \partial w_j}.$$

For a sufficiently smooth function,

$$\frac{\partial^2 g}{\partial w_i \partial w_j} = \frac{\partial^2 g}{\partial w_j \partial w_i}.$$

So the Hessian is symmetric.

Symmetric matrices have real eigenvalues
and orthogonal eigenvectors.

That will matter.

The Multivariable Quadratic Approximation

Let
 $\Delta \mathbf{w} = \begin{bmatrix} \Delta w_1 \\ \Delta w_2 \end{bmatrix}.$

Then
 $g(\mathbf{w} + \Delta \mathbf{w}) \approx g(\mathbf{w}) + \nabla g(\mathbf{w})^T \Delta \mathbf{w} + \frac{1}{2} \Delta \mathbf{w}^T H(\mathbf{w}) \Delta \mathbf{w}.$

Constant term:
current height.

Linear term:
current slope.

Quadratic term:
current curvature.

What Is Inside the Quadratic Term?

In two variables,

$$H = \begin{bmatrix} g_{11} & g_{12} \\ g_{21} & g_{22} \end{bmatrix}.$$

Then

$$\begin{aligned} &\Delta \mathbf{w}^T H \Delta \mathbf{w} \\ &= g_{11}(\Delta w_1)^2 + (g_{12} + g_{21})\Delta w_1 \Delta w_2 + g_{22}(\Delta w_2)^2. \end{aligned}$$

For a smooth function,

$$g_{12} = g_{21}.$$

So

$$\begin{aligned} &\frac{1}{2} \Delta \mathbf{w}^T H \Delta \mathbf{w} \\ &= \frac{1}{2} g_{11}(\Delta w_1)^2 + g_{12} \Delta w_1 \Delta w_2 + \frac{1}{2} g_{22}(\Delta w_2)^2. \end{aligned}$$

Off-diagonal Hessian entries measure coupled curvature.

Return to Our Familiar Cost

Use

$$g(w_1, w_2) = (w_1 - 2)^2 + 4(w_2 + 1)^2.$$

Gradient:

$$\nabla g = \begin{bmatrix} 2(w_1 - 2) \\ 8(w_2 + 1) \end{bmatrix}.$$

Hessian:

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}.$$

The Hessian is constant.

Again: quadratic function $\Rightarrow$ exact second-order model.

Revisit the Move From Lesson 1.4

At

$$\begin{aligned} &\mathbf{w} = (0, 0), \\ &g(\mathbf{w}) = 8, \\ &\nabla g(\mathbf{w}) = \begin{bmatrix} -4 \\ 8 \end{bmatrix}. \end{aligned}$$

Take

$$\Delta \mathbf{w} = \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix}.$$

First-order prediction:

$$8 - 0.4 - 0.8 = 6.8.$$

Now add the quadratic term.

Second-Order Prediction

Compute

$$\frac{1}{2} \Delta \mathbf{w}^T H \Delta \mathbf{w}.$$

First,

$$H \Delta \mathbf{w} = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix} \begin{bmatrix} 0.1 \\ -0.1 \end{bmatrix} = \begin{bmatrix} 0.2 \\ -0.8 \end{bmatrix}.$$

Then

$$\frac{1}{2} \begin{bmatrix} 0.1 & -0.1 \end{bmatrix} \begin{bmatrix} 0.2 \\ -0.8 \end{bmatrix} = \frac{1}{2} (0.10) = 0.05.$$

Therefore

$$g(0.1, -0.1) \approx 6.8 + 0.05 = 6.85.$$

Exact value:

$$6.85.$$

For this quadratic, second order is exact.

Curvature Depends on Direction

Choose a unit direction $\mathbf{u}$.

Move along the line

$$\mathbf{w}(t) = \mathbf{w}_0 + t\mathbf{u}.$$

The second derivative along this line is

$$\frac{d^2}{dt^2} g(\mathbf{w}_0 + t\mathbf{u})_{t=0} = \mathbf{u}^T H(\mathbf{w}_0) \mathbf{u}.$$

So

$$\mathbf{u}^T H \mathbf{u}$$

measures curvature in direction $\mathbf{u}$.

The Hessian contains curvature information in every direction.

Directional Curvature Example

For

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix},$$

in direction

$$\mathbf{u}_1 = \begin{bmatrix} 1 \\ 0 \end{bmatrix},$$

curvature is

$$\mathbf{u}_1^T H \mathbf{u}_1 = 2.$$

In direction

$$\mathbf{u}_2 = \begin{bmatrix} 0 \\ 1 \end{bmatrix},$$

curvature is

$$\mathbf{u}_2^T H \mathbf{u}_2 = 8.$$

The w_2 direction is four times as curved.

Exactly the difference we saw in Lesson 1.6.

Eigenvectors Give Principal Curvature Directions

Suppose

$$H\mathbf{v} = \lambda\mathbf{v},$$

with $\|\mathbf{v}\| = 1$.

Then

$$\begin{aligned} \mathbf{v}^T H \mathbf{v} &= \mathbf{v}^T (\lambda \mathbf{v}) \\ &= \lambda \mathbf{v}^T \mathbf{v} = \lambda. \end{aligned}$$

Therefore:

eigenvectors of H give principal curvature directions;

eigenvalues give the corresponding curvatures.

For our Hessian:

eigenvalues are 2 and 8.

Curvature and Conditioning

For a positive-definite quadratic,

let

$$0 < \lambda_{\min} \leq \lambda_{\max}$$

be the smallest and largest Hessian eigenvalues.

A large ratio

$$\kappa = \frac{\lambda_{\max}}{\lambda_{\min}}$$

means very different curvature in different directions.

κ is a **condition number** for the quadratic.

For our familiar example:

$$\kappa = \frac{8}{2} = 4.$$

For

$$g(w_1, w_2) = w_1^2 + 100w_2^2,$$

$$H = \begin{bmatrix} 2 & 0 \\ 0 & 200 \end{bmatrix},$$

so

$$\kappa = 100.$$

Large $\kappa \Rightarrow$ narrow valley $\Rightarrow$ slower gradient descent.

Second-Order Test at a Stationary Point

Suppose

$$\nabla g(\mathbf{w}^*) = \mathbf{0}.$$

Then near $\mathbf{w}^*$,

$$g(\mathbf{w}^* + \Delta \mathbf{w}) \approx g(\mathbf{w}^*) + \frac{1}{2} \Delta \mathbf{w}^T H(\mathbf{w}^*) \Delta \mathbf{w}.$$

The Hessian decides whether nearby directions curve up or down.

Positive Definite

A symmetric matrix H is **positive definite** if

$$\mathbf{v}^T H \mathbf{v} > 0$$

for every nonzero vector $\mathbf{v}$.

Equivalent:

all eigenvalues of H are positive.

If

$$\nabla g(\mathbf{w}^*) = \mathbf{0}$$

and

$H(\mathbf{w}^*)$ is positive definite,

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

Every direction curves upward.

Negative Definite and Indefinite

If every Hessian eigenvalue is negative:

H is **negative definite**.

At a stationary point:
local maximum.

If the Hessian has both positive and negative eigenvalues:
 H is **indefinite**.

At a stationary point:
saddle point.

Some directions curve upward.

Others curve downward.

Classify Our Familiar Minimum

For
 $g(w_1, w_2) = (w_1 - 2)^2 + 4(w_2 + 1)^2,$

stationary point:
 $(2, -1).$

Hessian:
$$H = \begin{bmatrix} 2 & 0 \\ 0 & 8 \end{bmatrix}.$$

Eigenvalues:
2, 8.

Both positive.

Therefore
 $(2, -1)$ is a strict local minimum.

For this quadratic it is also the global minimum.

Classify a Saddle Point

Recall
 $g(w_1, w_2) = w_1^2 - w_2^2.$

At $(0, 0):$
 $\nabla g = \mathbf{0}.$

Hessian:
$$H = \begin{bmatrix} 2 & 0 \\ 0 & -2 \end{bmatrix}.$$

Eigenvalues:
2, -2.

One positive. One negative.

Therefore the Hessian is indefinite.

(0,0) is a saddle point.

What if an Eigenvalue Is Zero?

Suppose the Hessian is only positive semidefinite or negative semidefinite.

Then the second-order test may be inconclusive.

Example:

$$g(w) = w^4.$$

At $w = 0$:

$$g'(0) = 0,$$

$$g''(0) = 0.$$

But $w = 0$ is still a minimum.

Second-order information can sometimes be insufficient.

A Hessian With Cross Terms

Let

$$g(w_1, w_2) = w_1^2 + 2w_1w_2 + 3w_2^2.$$

Gradient:

$$\nabla g = \begin{bmatrix} 2w_1 + 2w_2 \\ 2w_1 + 6w_2 \end{bmatrix}.$$

Hessian:

$$H = \begin{bmatrix} 2 & 2 \\ 2 & 6 \end{bmatrix}.$$

The off-diagonal entries come from the interaction term

$$2w_1w_2.$$

Curvature directions need not line up with the coordinate axes.

Eigenvalues of the Cross-Term Example

For

$$H = \begin{bmatrix} 2 & 2 \\ 2 & 6 \end{bmatrix},$$

solve

$$\det(H - \lambda I) = 0.$$

So

$$(2 - \lambda)(6 - \lambda) - 4 = 0,$$

$$\lambda^2 - 8\lambda + 8 = 0.$$

Thus

$$\lambda = 4 \pm 2\sqrt{2}.$$

Both are positive.

Therefore the quadratic curves upward in every direction.

The origin is a strict minimum.

The Hessian in Python

Represent a Hessian as a NumPy array:

```
H = np.array([[2.0, 2.0],
               [2.0, 6.0]])
```

Find its eigenvalues:

```
np.linalg.eigvals(H)
```

Output is approximately

```
[1.1716  6.8284]
```

Both positive.

New Python idea:

`np` computes matrix eigenvalues.

Evaluate the Quadratic Term in Python

Let

```
delta = np.array([0.1, -0.1])
```

```
H = np.array([[2.0, 0.0],
               [0.0, 8.0]])
```

Then

```
0.5 * delta @ H @ delta
```

returns

```
0.05.
```

New Python idea:

the symbol `@` performs matrix multiplication.

Thus

`delta @ H @ delta`

represents

$$\Delta \mathbf{w}^T H \Delta \mathbf{w}.$$

First Order Versus Second Order

First-order model:

$$g(\mathbf{w} + \Delta\mathbf{w}) \approx g(\mathbf{w}) + \nabla g^T \Delta\mathbf{w}.$$

Uses:

slope.

Second-order model:

$$g(\mathbf{w} + \Delta\mathbf{w}) \approx g(\mathbf{w}) + \nabla g^T \Delta\mathbf{w} + \frac{1}{2} \Delta\mathbf{w}^T H \Delta\mathbf{w}.$$

Uses:

slope + curvature.

More information.

More computation.

Why This Matters for Optimization

Gradient descent uses only

$$\nabla g.$$

It knows which direction is downhill,
but not how the surface bends.

The Hessian tells us:

which directions are steep;

which directions are shallow;

whether a stationary point is locally a bowl, hill, or saddle;

how to build a local quadratic model.

Next question:

can we minimize that quadratic model directly?

What did we learn?

- ◆ Second-order Taylor approximation adds a curvature term to the local linear model.
- ◆ The Hessian contains all second partial derivatives of the cost function.
- ◆ $\frac{1}{2} \Delta\mathbf{w}^T H \Delta\mathbf{w}$ is the quadratic curvature correction.
- ◆ $\mathbf{u}^T H \mathbf{u}$ measures curvature in direction $\mathbf{u}$.
- ◆ Hessian eigenvectors give principal curvature directions;
eigenvalues give the corresponding curvatures.
- ◆ At a stationary point, positive-definite Hessian $\Rightarrow$ local minimum; indefinite Hessian $\Rightarrow$ saddle.



Next

- The gradient gives slope.
- The Hessian gives curvature.
- Together they define a local quadratic model.
- Next: minimize that quadratic model to choose the step.
- **1.8 - Newton's Method in Several Variables**