Convergence

- In the discussion so far we have assumed the training arrives at a local minimum

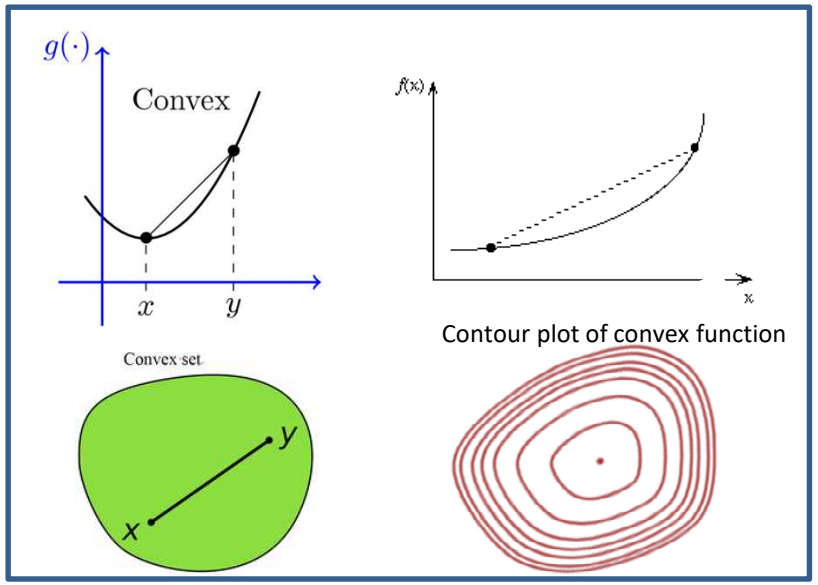
- Does it always converge?
- How long does it take?

- Hard to analyze for an MLP, but we can look at the problem through the lens of convex optimization

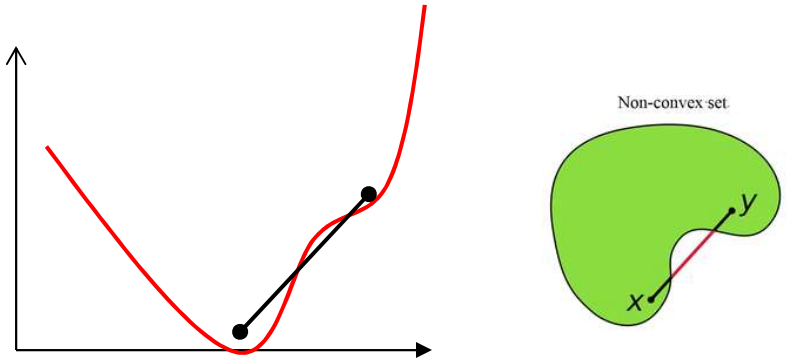
A quick tour of (convex) optimization

Convex Loss Functions

- A surface is “convex” if it is continuously curving upward
  - We can connect any two points above the surface without intersecting it
  - Many mathematical definitions that are equivalent

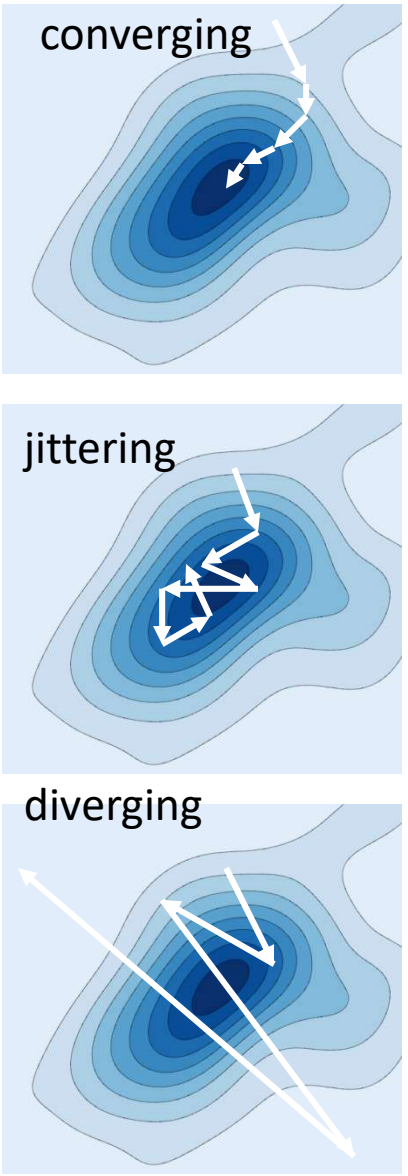


- Caveat: Neural network loss surface is generally not convex
  - Streetlight effect



Convergence of gradient descent

- An iterative algorithm is said to *converge* to a solution if the value updates arrive at a fixed point
  - Where the gradient is 0 and further updates do not change the estimate
- The algorithm may not actually converge
  - It may jitter around the local minimum
  - It may even diverge
- Conditions for convergence?

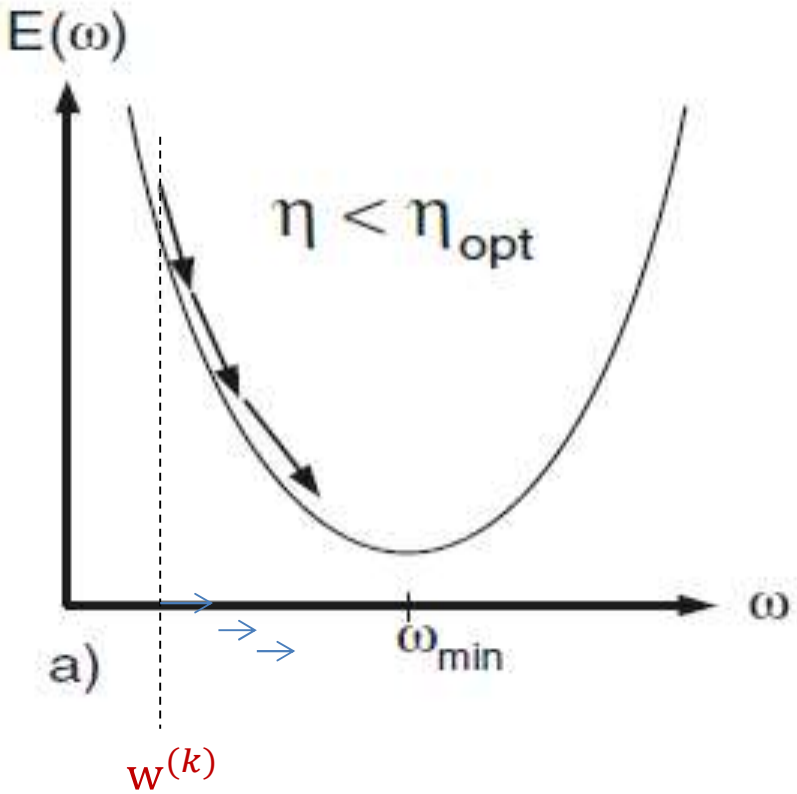


Convergence for quadratic surfaces

$$Minimize\ E = \frac{1}{2}aw^2 + bw + c$$

$$\mathrm{w}^{(k+1)} = \mathrm{w}^{(k)} - \eta \frac{dE\left(\mathrm{w}^{(k)}\right)}{d\mathrm{w}}$$

Gradient descent with fixed step size η to estimate *scalar* parameter w



- Gradient descent to find the optimum of a quadratic, starting from $\mathrm{w}^{(k)}$
- Assuming fixed step size η
- What is the optimal step size η to get there fastest?

Convergence for quadratic surfaces

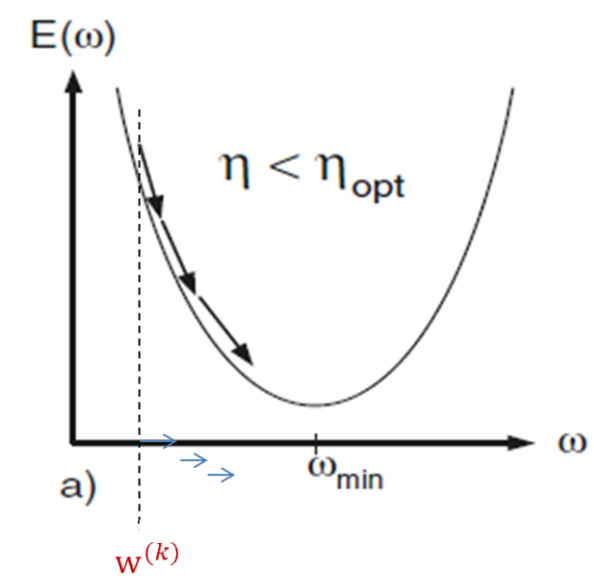
$$E = \frac{1}{2}aw^2 + bw + c$$

$$\mathrm{w}^{(k+1)} = \mathrm{w}^{(k)} - \eta \frac{dE\left(\mathrm{w}^{(k)}\right)}{d\mathrm{w}}$$

- Any quadratic objective can be written as

$$E(w) = E\left(\mathrm{w}^{(k)}\right) + E'\left(\mathrm{w}^{(k)}\right)\left(w - \mathrm{w}^{(k)}\right) + \frac{1}{2}E''\left(\mathrm{w}^{(k)}\right)\left(w - \mathrm{w}^{(k)}\right)^2$$

  - Taylor expansion



- Minimizing w.r.t w, we get (Newton's method)

$$w_{min} = \mathrm{w}^{(k)} - E''\left(\mathrm{w}^{(k)}\right)^{-1} E'\left(\mathrm{w}^{(k)}\right)$$

- Note:

$$\frac{dE\left(\mathrm{w}^{(k)}\right)}{d\mathrm{w}} = E'\left(\mathrm{w}^{(k)}\right)$$

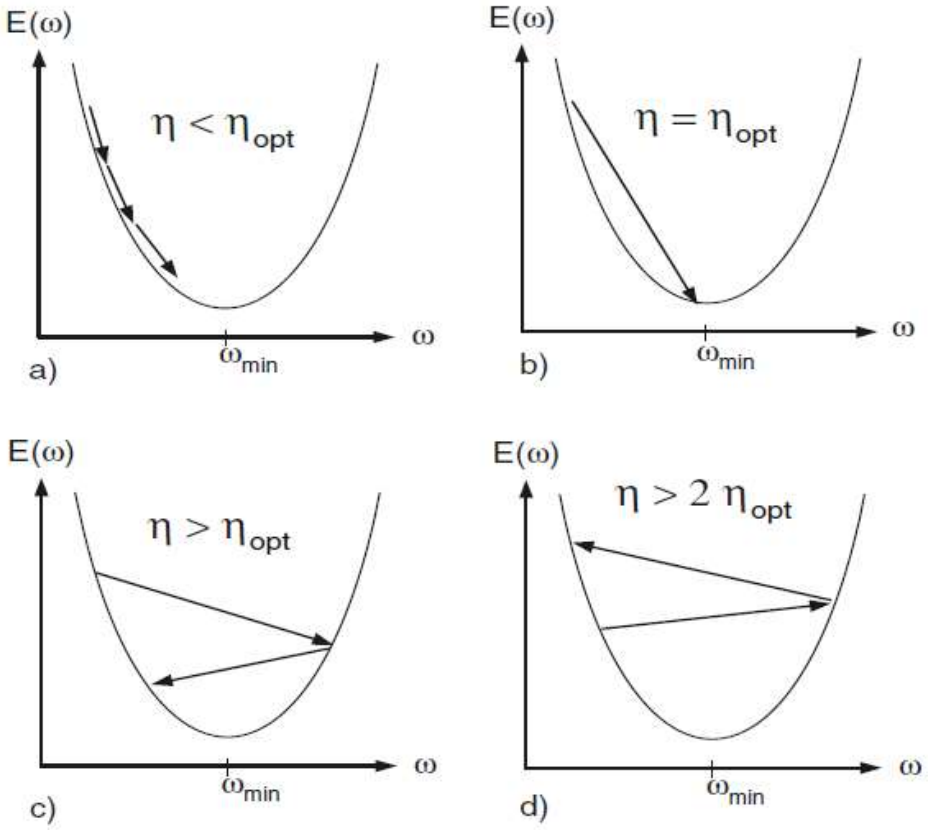
- Comparing to the gradient descent rule, we see that we can arrive at the optimum in a single step using the optimum step size

$$\eta_{opt} = E''\left(\mathrm{w}^{(k)}\right)^{-1} = \boldsymbol{a^{-1}}$$

With non-optimal step size

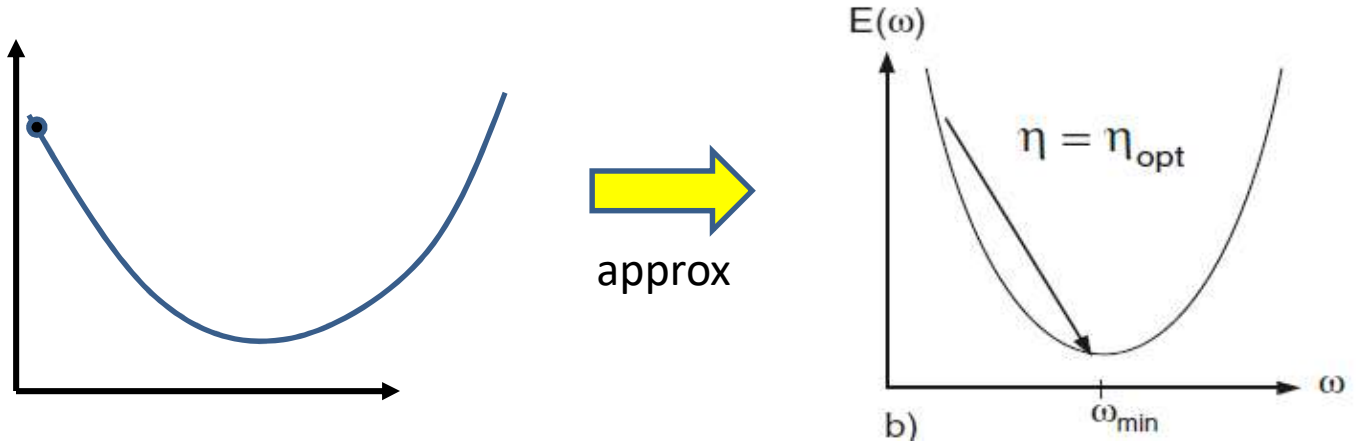
$$\mathrm{w}^{(k+1)} = \mathrm{w}^{(k)} - \eta \frac{dE\left(\mathrm{w}^{(k)}\right)}{d\mathrm{w}}$$

Gradient descent with fixed step size η to estimate scalar parameter w



- For $\eta < \eta_{opt}$ the algorithm will converge monotonically
- For $2\eta_{opt} > \eta > \eta_{opt}$ we have oscillating convergence
- For $\eta > 2\eta_{opt}$ we get divergence

For generic differentiable convex objectives



- Any differentiable convex objective $E(w)$ can be approximated as

$$E \approx E\left(\mathrm{w}^{(k)}\right) + \left(w - \mathrm{w}^{(k)}\right)\frac{dE\left(\mathrm{w}^{(k)}\right)}{dw} + \frac{1}{2}\left(w - \mathrm{w}^{(k)}\right)^2 \frac{d^2E\left(\mathrm{w}^{(k)}\right)}{dw^2} + \cdots$$

  - Taylor expansion
- Using the same logic as before, we get (Newton's method)

$$\eta_{opt} = \left(\frac{d^2E\left(\mathrm{w}^{(k)}\right)}{dw^2}\right)^{-1}$$

- We can get divergence if $\eta \geq 2\eta_{opt}$

For functions of *multivariate* inputs

$E = g(\mathbf{w})$, $\mathbf{w}$ is a vector $\mathbf{w} = [w_1, w_2, \ldots, w_N]$

- Consider a simple quadratic convex (paraboloid) function

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c$$

  - Since $E^T = E$ (E is scalar), $\mathbf{A}$ can always be made symmetric
    - For **convex** E, $\mathbf{A}$ is always positive definite, and has positive eigenvalues

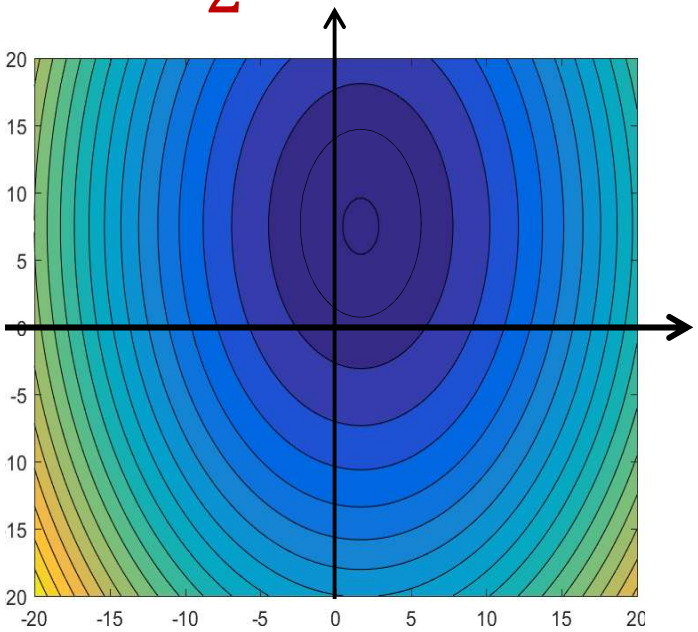
- When $\mathbf{A}$ is diagonal:

$$E = \frac{1}{2}\sum_i \left(a_{ii}w_i^2 + b_i w_i\right) + c$$

  - The w_is are *uncoupled*
  - For *convex* (paraboloid) E, the a_{ii} values are all positive
  - Just an sum of N independent quadratic functions

Multivariate Quadratic with Diagonal $\mathbf{A}$

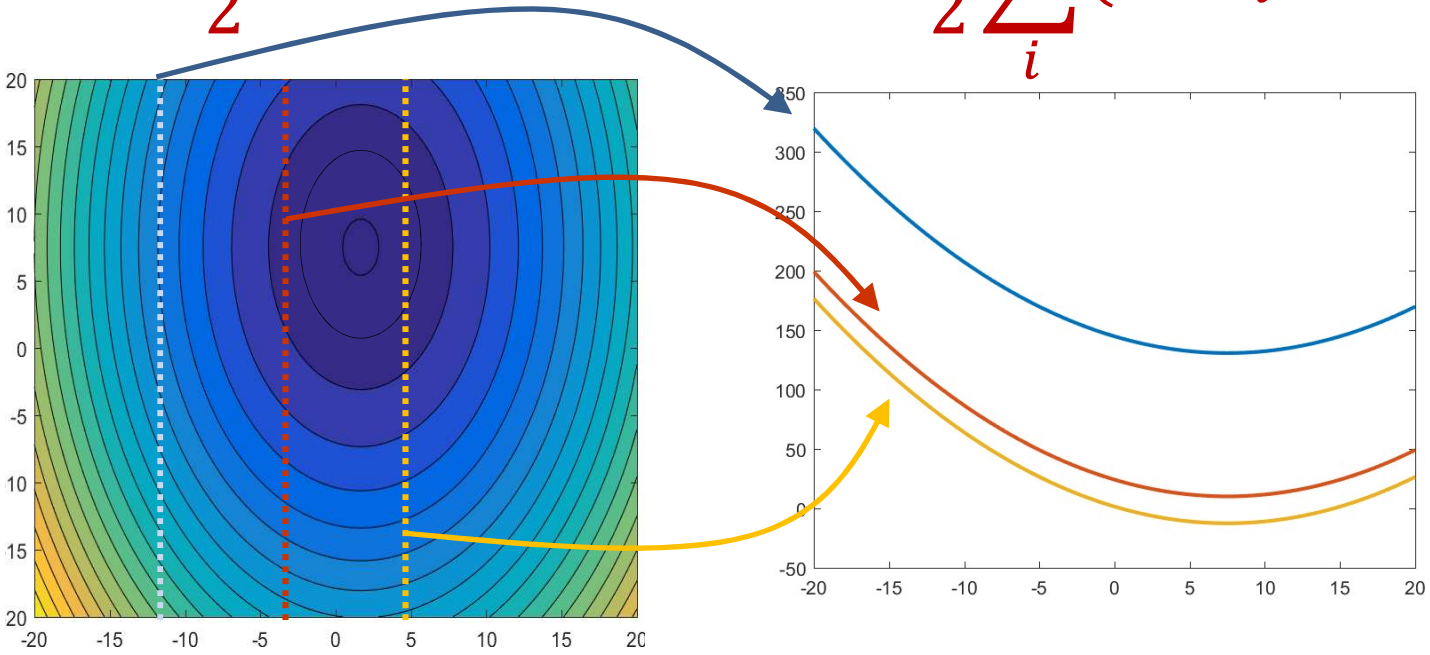
$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c = \frac{1}{2}\sum_i \left(a_{ii}w_i^2 + b_i w_i\right) + c$$



- Equal-value contours will be parallel to the axis

Multivariate Quadratic with Diagonal A

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c = \frac{1}{2}\sum_i \left(a_{ii}w_i^2 + b_i w_i\right) + c$$

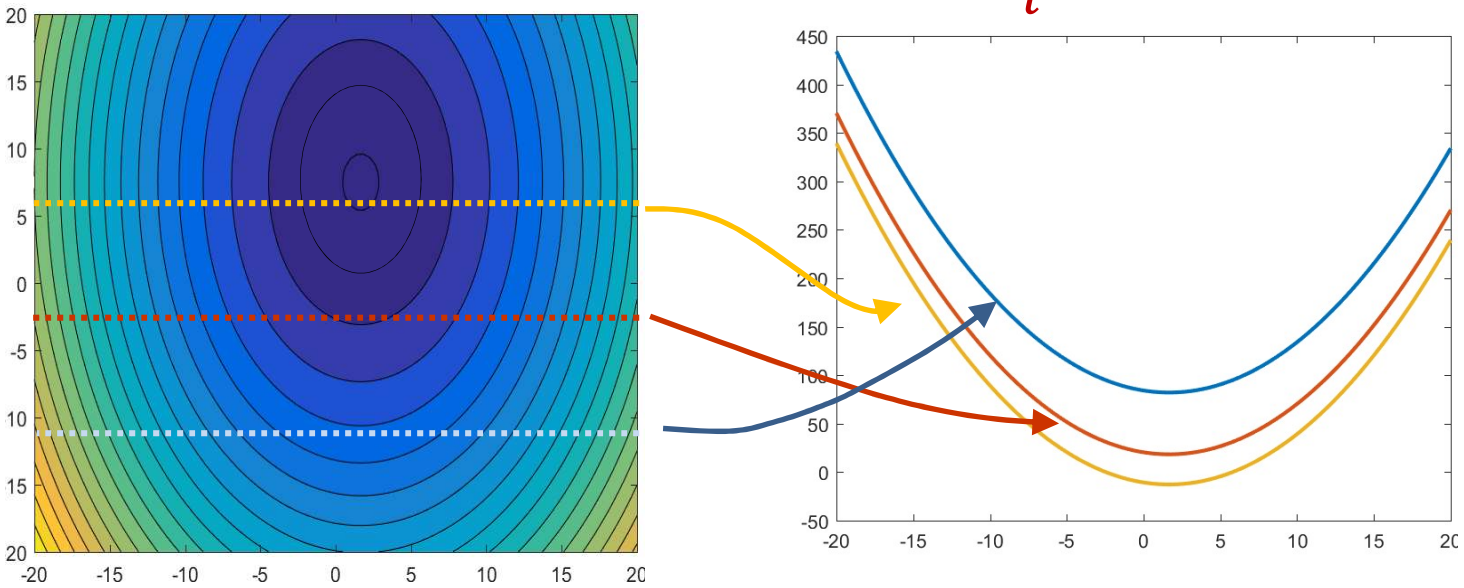


- Equal-value contours will be parallel to the axis
  - All "slices" parallel to an axis are shifted versions of one another

$$E = \frac{1}{2}a_{ii}w_i^2 + b_i w_i + c + C(\neg w_i)$$

Multivariate Quadratic with Diagonal A

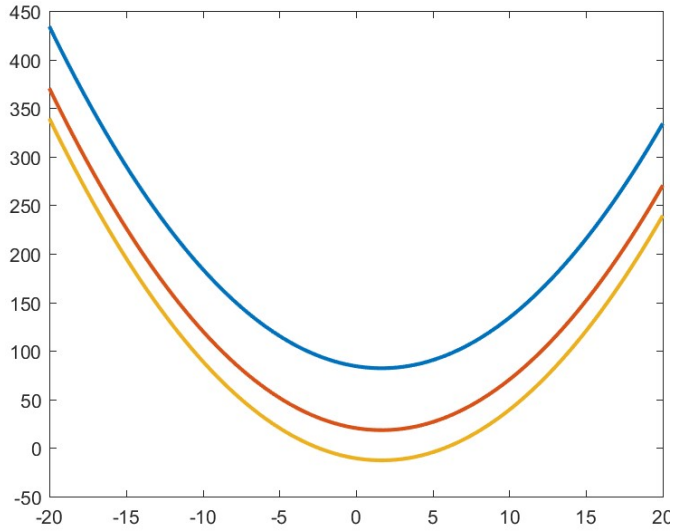
$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c = \frac{1}{2}\sum_i \left(a_{ii}w_i^2 + b_i w_i\right) + c$$

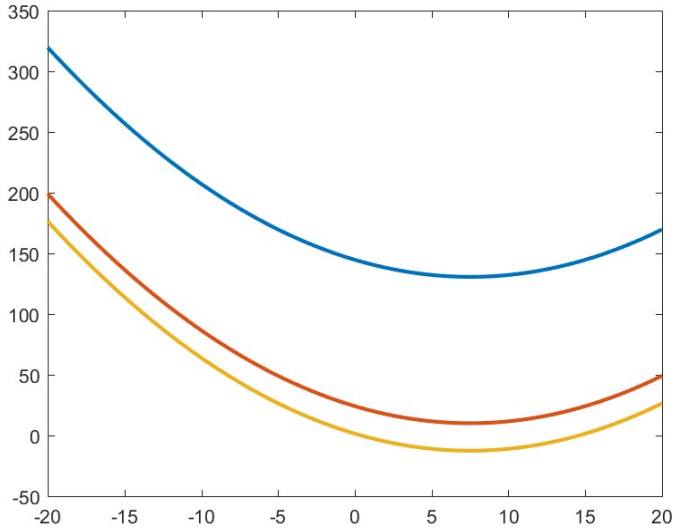


- Equal-value contours will be parallel to the axis
  - All "slices" parallel to an axis are shifted versions of one another

$$E = \frac{1}{2}a_{ii}w_i^2 + b_i w_i + c + C(\neg w_i)$$

“Descents” are uncoupled





$$E = \frac{1}{2}a_{11}w_1^2 + b_1 w_1 + c + C(\neg w_1)$$

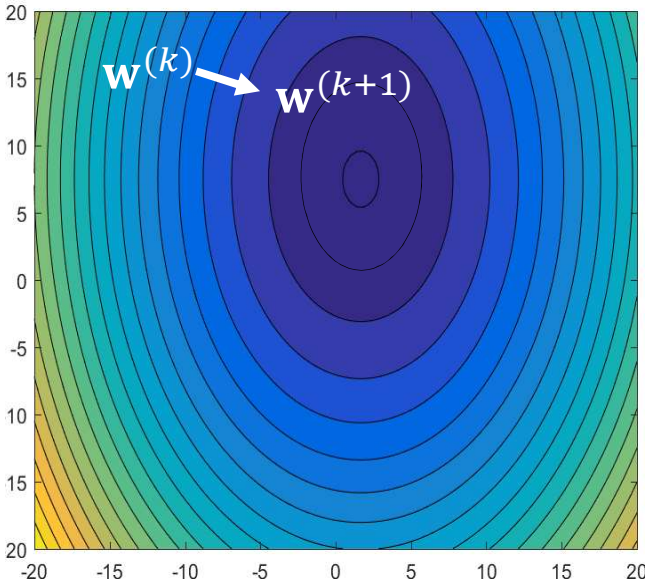
$$\eta_{1,opt} = a_{11}^{-1}$$

$$E = \frac{1}{2}a_{22}w_2^2 + b_2 w_2 + c + C(\neg w_2)$$

$$\eta_{2,opt} = a_{22}^{-1}$$

- The optimum of each coordinate is not affected by the other coordinates
  - I.e. we could optimize each coordinate independently
- **Note: Optimal learning rate is different for the different coordinates**

Vector update rule



$$\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta \nabla_{\mathbf{w}} E$$

$$w_i^{(k+1)} = w_i^{(k)} - \eta \frac{dE\left(w_i^{(k)}\right)}{d\mathrm{w}}$$

- Conventional vector update rules for gradient descent: update entire vector against direction of gradient
  - Note : Gradient is perpendicular to equal value contour
  - The same learning rate is applied to all components

Problem with vector update rule

$$\mathbf{w}^{(k+1)} \leftarrow \mathbf{w}^{(k)} - \eta \nabla_{\mathbf{w}} E^T$$

$$w_i^{(k+1)} = w_i^{(k)} - \eta \frac{dE\left(w_i^{(k)}\right)}{d\mathrm{w}}$$

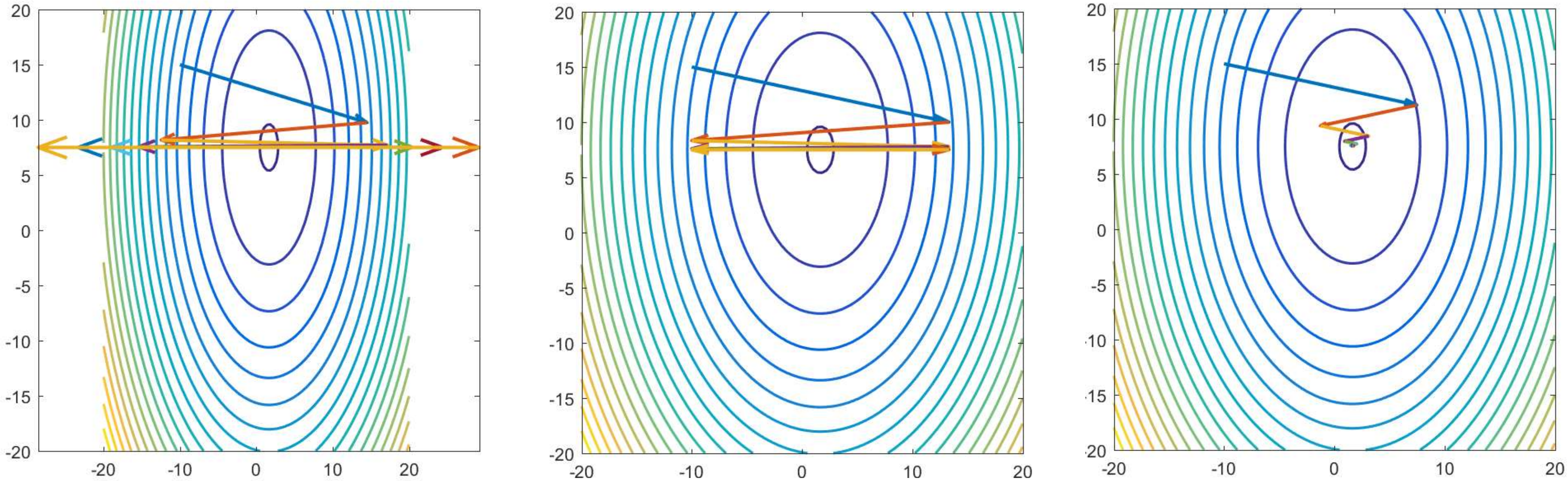
$$\eta_{i,opt} = \left(\frac{d^2 E\left(w_i^{(k)}\right)}{dw_i^2}\right)^{-1} = a_{ii}^{-1}$$

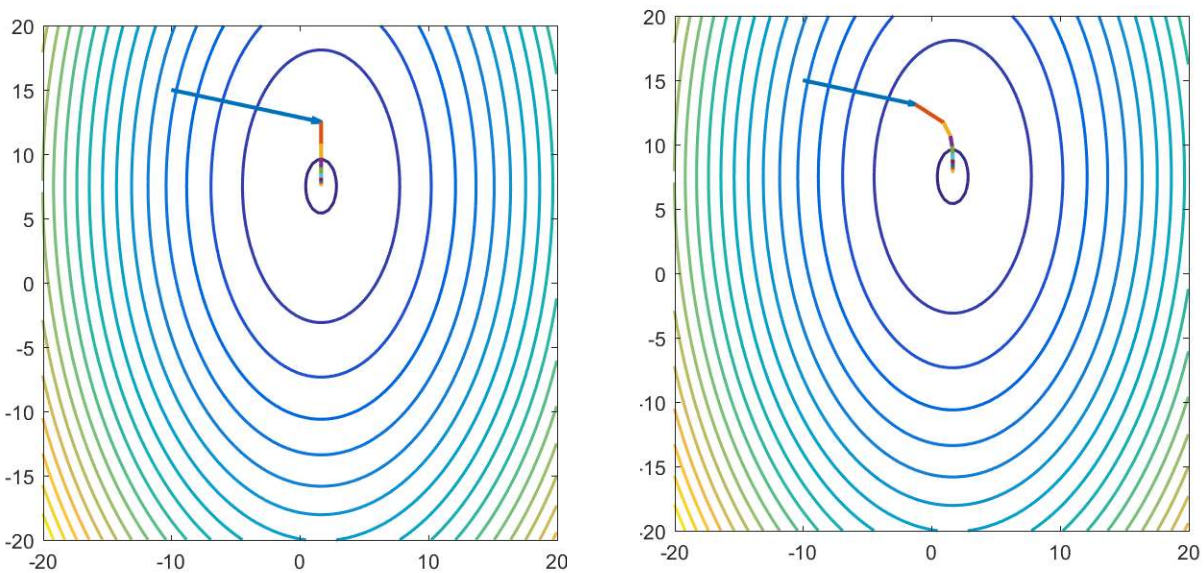
- The learning rate must be lower than twice the *smallest* optimal learning rate for any component

$$\eta < 2 \min_i \eta_{i,opt}$$

  - Otherwise the learning will diverge
- This, however, makes the learning very slow
  - And will oscillate in all directions where $\eta_{i,opt} \leq \eta < 2\eta_{i,opt}$

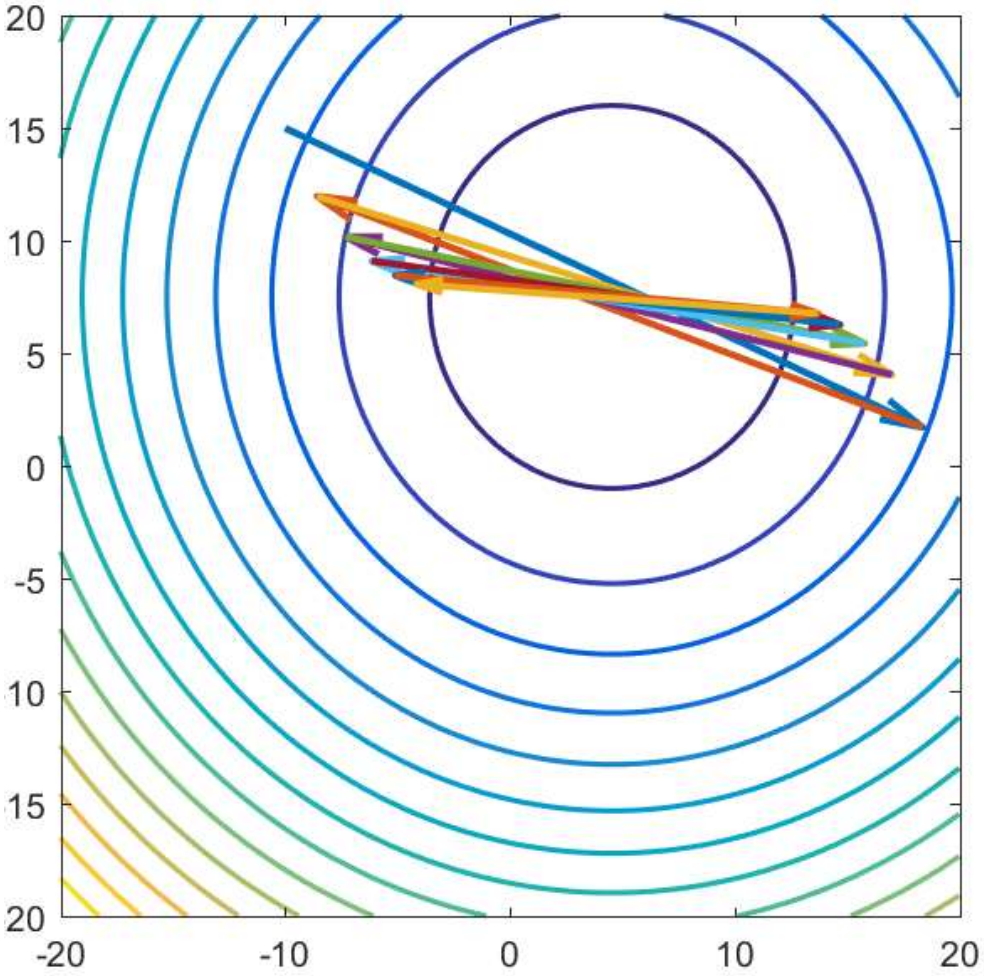
Dependence on learning rate





- $\eta_{1,opt} = 1;\ \eta_{2,opt} = 0.33$
- $\eta = 2.1\eta_{2,opt}$
- $\eta = 2\eta_{2,opt}$
- $\eta = 1.5\eta_{2,opt}$
- $\eta = \eta_{2,opt}$
- $\eta = 0.75\eta_{2,opt}$

Dependence on learning rate



- $\eta_{1,opt} = 1;\ \eta_{2,opt} = 0.91;$ $\eta = 1.9\,\eta_{2,opt}$

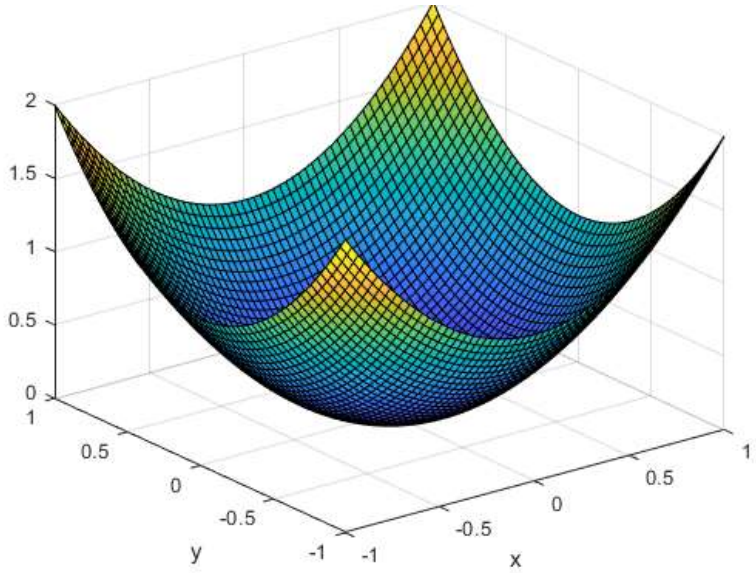
Convergence

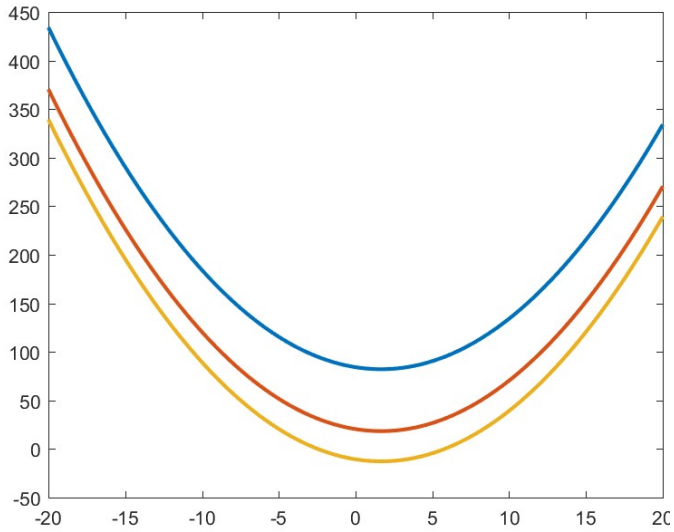
- Convergence behaviors become increasingly unpredictable as dimensions increase

- For the fastest convergence, ideally, the learning rate η must be close to both, the largest $\eta_{i,opt}$ and the smallest $\eta_{i,opt}$
  - To ensure convergence in every direction
  - Generally infeasible

- Convergence is particularly slow if $\frac{\max_i \eta_{i,opt}}{\min_i \eta_{i,opt}}$ is large
  - The "condition" number is small

Quadratic convexity





- A quadratic function has the form $\frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c$
  - Every "slice" is a quadratic bowl
- In some sense, the "standard" for gradient-descent based optimization
  - Others convex functions will be steeper in some regions, but flatter in others
- Gradient descent solution will have linear convergence
  - Take $O(\log 1/\varepsilon)$ steps to get within ε of the optimal solution

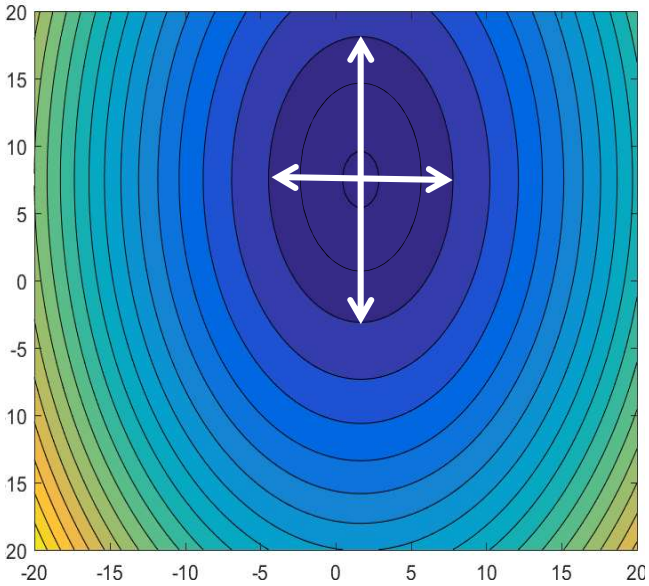
Convergence

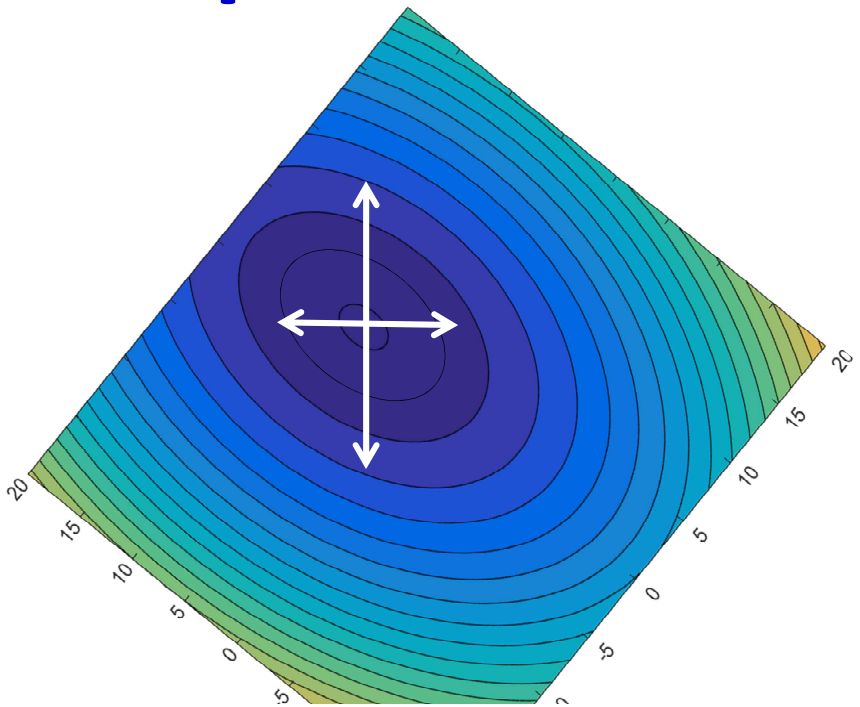
- Convergence behaviors become increasingly unpredictable as dimensions increase

- For the fastest convergence, ideally, the learning rate η must be close to both, the largest $\eta_{i,opt}$ and the smallest $\eta_{i,opt}$
  - To ensure convergence in every direction
  - Generally infeasible

- Convergence is particularly slow if $\frac{\max_i \eta_{i,opt}}{\min_i \eta_{i,opt}}$ is large
  - The "condition" number is small

One reason for the problem

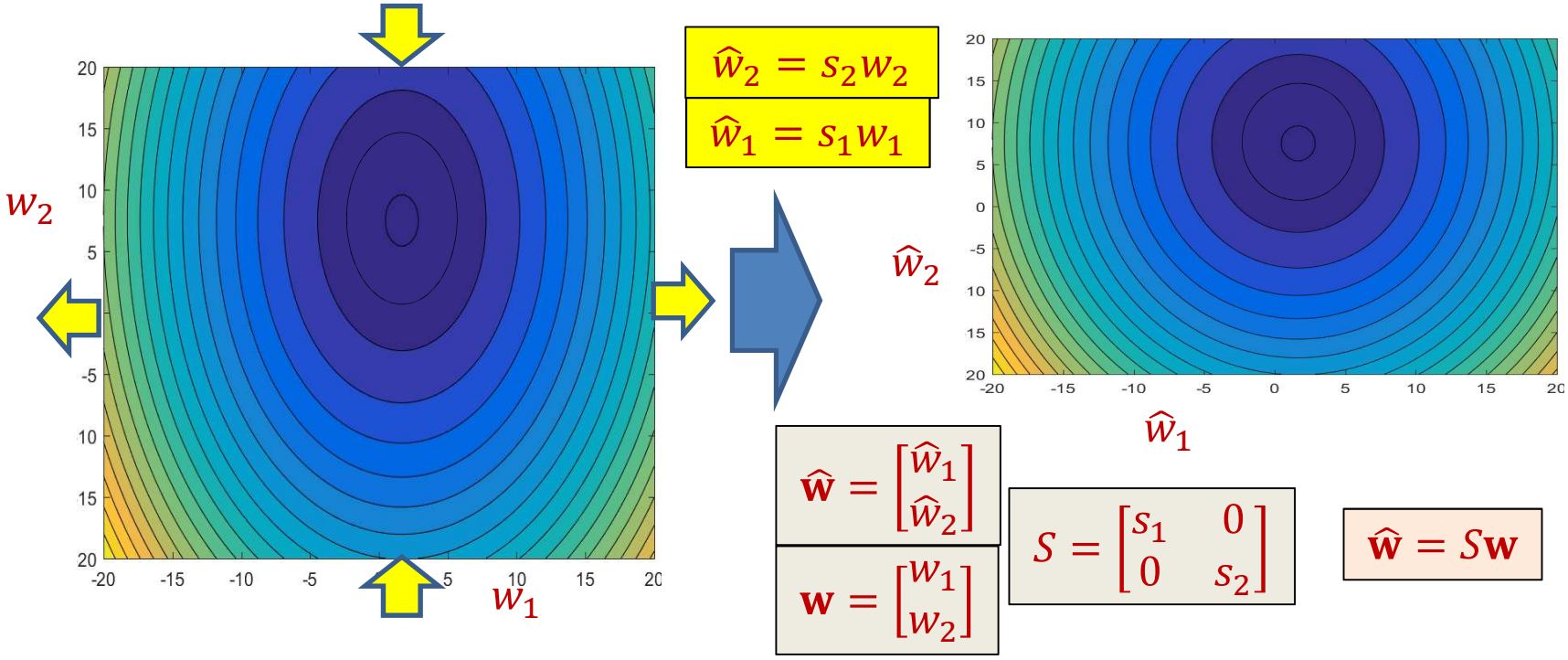




- The objective function has different eccentricities in different directions
  - Resulting in different optimal learning rates for different directions
  - The problem is more difficult when the ellipsoid is not axis aligned: the steps along the two directions are coupled! Moving in one direction changes the gradient along the other

- Solution: *Normalize* the objective to have identical eccentricity in all directions
  - Then all of them will have identical optimal learning rates
  - Easier to find a working learning rate

Solution: Scale the axes



- Scale (and rotate) the axes, such that all of them have identical (identity) "spread"
  - Equal-value contours are circular
  - Movement along the coordinate axes become independent
- **Note:** equation of a quadratic surface with circular equal-value contours can be written as

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + c$$

Scaling the axes

- Original equation:

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

- We want to find a (diagonal) scaling matrix S such that

$$\mathrm{S} = \begin{bmatrix} s_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_N \end{bmatrix}, \qquad \hat{\mathbf{w}} = \mathrm{S}\mathbf{w}$$

- And

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + \mathrm{c}$$

Scaling the axes

- Original equation:

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

- We want to find a (diagonal) scaling matrix S such that

$$S = \begin{bmatrix} s_1 & \cdots & 0 \\ \vdots & \ddots & \vdots \\ 0 & \cdots & s_N \end{bmatrix}, \qquad \hat{\mathbf{w}} = S\mathbf{w}$$

- And

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + c$$

By inspection:
$S = \mathbf{A}^{0.5}$

Scaling the axes

- We have

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

$$\hat{\mathbf{w}} = \mathrm{S}\mathbf{w}$$

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + \mathrm{c}$$

$$= \frac{1}{2}\mathbf{w}^T\mathrm{S}^T\mathrm{S}\mathbf{w} + \hat{\mathbf{b}}^T\mathrm{S}\mathbf{w} + c$$

- Equating linear and quadratic coefficients, we get

$$\mathrm{S}^T\mathrm{S} = \mathbf{A}, \qquad \hat{\mathbf{b}}^T\mathrm{S} = \mathbf{b}^T$$

- Solving: $\mathrm{S} = \mathbf{A}^{0.5}, \qquad \hat{\mathbf{b}} = \mathbf{A}^{-0.5}\mathbf{b}$

Scaling the axes

- We have

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

$$\hat{\mathbf{w}} = \mathrm{S}\mathbf{w}$$

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + \mathrm{c}$$

- Solving for S we get

$$\hat{\mathbf{w}} = \mathbf{A}^{0.5}\mathbf{w}, \qquad \hat{\mathbf{b}} = \mathbf{A}^{-0.5}\mathbf{b}$$

Scaling the axes

- We have

$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

$$\hat{\mathbf{w}} = \mathrm{S}\mathbf{w}$$

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + \mathrm{c}$$

- Solving for S we get

$$\hat{\mathbf{w}} = \mathbf{A}^{0.5}\mathbf{w}, \qquad \hat{\mathbf{b}} = \mathbf{A}^{-0.5}\mathbf{b}$$

The Inverse Square Root of A

- For *any* positive definite $\mathbf{A}$, we can write

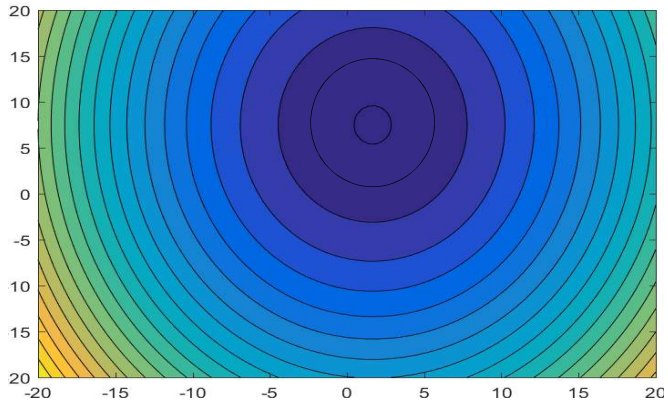
$$\mathbf{A} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^{\mathrm{T}}$$

  - Eigen decomposition
  - $\mathbf{E}$ is an orthogonal matrix
  - $\boldsymbol{\Lambda}$ is a diagonal matrix of non-zero diagonal entries

- Defining $\mathbf{A}^{0.5} = \mathbf{E}\boldsymbol{\Lambda}^{0.5}\mathbf{E}^{\mathrm{T}}$
  - Check $(\mathbf{A}^{0.5})^{\mathrm{T}}\mathbf{A}^{0.5} = \mathbf{E}\boldsymbol{\Lambda}\mathbf{E}^{\mathrm{T}} = \mathbf{A}$

- Defining $\mathbf{A}^{-0.5} = \mathbf{E}\boldsymbol{\Lambda}^{-0.5}\mathbf{E}^{\mathrm{T}}$
  - Check: $(\mathbf{A}^{-0.5})^{\mathrm{T}}\mathbf{A}^{-0.5} = \mathbf{E}\boldsymbol{\Lambda}^{-1}\mathbf{E}^{\mathrm{T}} = \mathbf{A}^{-1}$

Returning to our problem

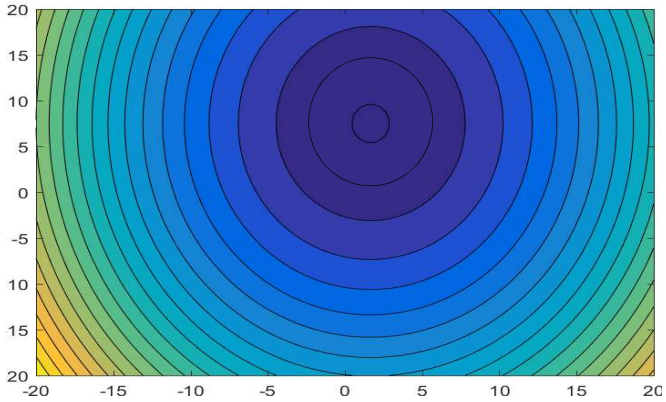


$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + c$$

- Computing the gradient, and noting that $\mathbf{A}^{0.5}$ is symmetric, we can relate $\nabla_{\hat{\mathbf{w}}}E$ and $\nabla_{\mathbf{w}}E$:

$$\nabla_{\hat{\mathbf{w}}}E = \hat{\mathbf{w}}^T + \hat{\mathbf{b}}^T$$
$$= \mathbf{w}^T\mathbf{A}^{0.5} + \mathbf{b}^T\mathbf{A}^{-0.5}$$
$$= (\mathbf{w}^T\mathbf{A} + \mathbf{b}^T)\mathbf{A}^{-0.5}$$
$$= \nabla_{\mathbf{w}}E.\,\mathbf{A}^{-0.5}$$

Returning to our problem

- 

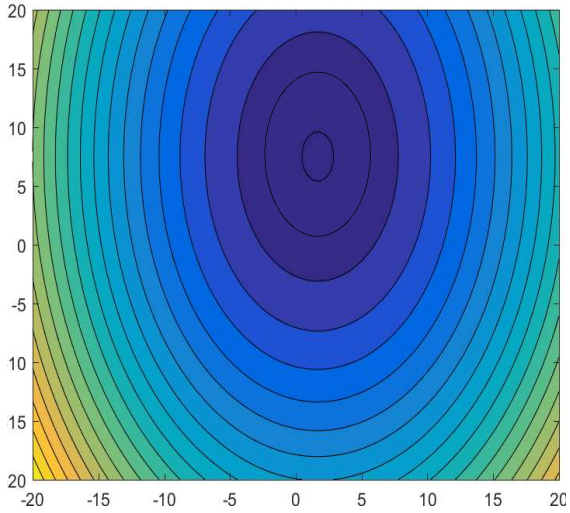
$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + c$$

- Gradient descent rule:
  - $\hat{\mathbf{w}}^{(k+1)} = \hat{\mathbf{w}}^{(k)} - \eta\nabla_{\hat{\mathbf{w}}}E\left(\hat{\mathbf{w}}^{(k)}\right)^T$
  - Learning rate is now independent of direction

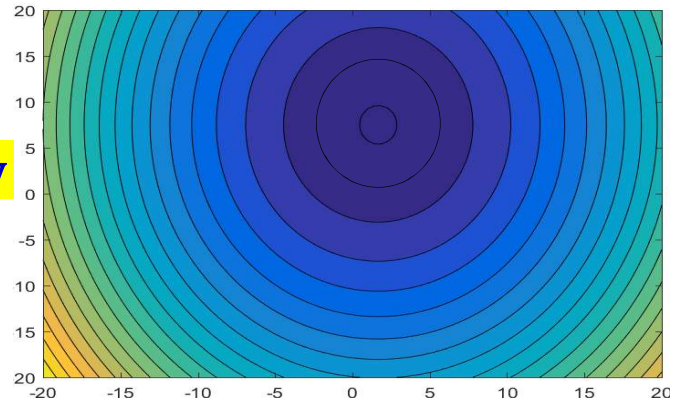
- Using $\hat{\mathbf{w}} = \mathbf{A}^{0.5}\mathbf{w}$, and $\nabla_{\hat{\mathbf{w}}}E(\hat{\mathbf{w}})^T = \mathbf{A}^{-0.5}\nabla_{\mathbf{w}}E(\mathbf{w})^T$

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta\mathbf{A}^{-1}\nabla_{\mathbf{w}}E\left(\mathbf{w}^{(k)}\right)^T$$

Modified update rule



$$\hat{\mathbf{w}} = \mathbf{A}^{0.5}\mathbf{w}$$



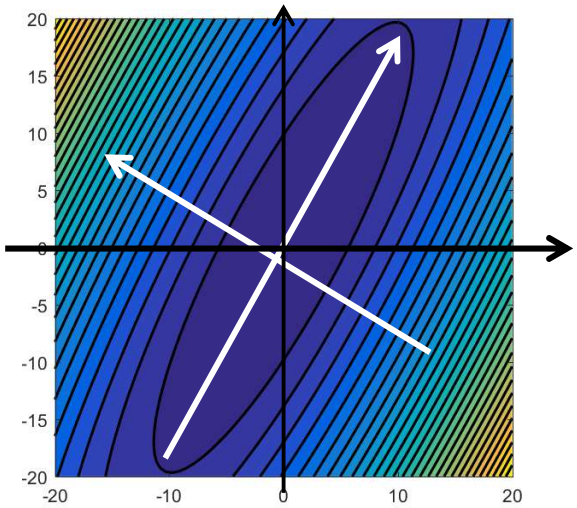
$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{b}^T\mathbf{w} + c$$

$$E = \frac{1}{2}\hat{\mathbf{w}}^T\hat{\mathbf{w}} + \hat{\mathbf{b}}^T\hat{\mathbf{w}} + c$$

- $\hat{\mathbf{w}}^{(k+1)} = \hat{\mathbf{w}}^{(k)} - \eta\nabla_{\hat{\mathbf{w}}}E\left(\hat{\mathbf{w}}^{(k)}\right)^T$
- Leads to the modified gradient descent rule

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta\mathbf{A}^{-1}\nabla_{\mathbf{w}}E\left(\mathbf{w}^{(k)}\right)^T$$

For non-axis-aligned quadratics..

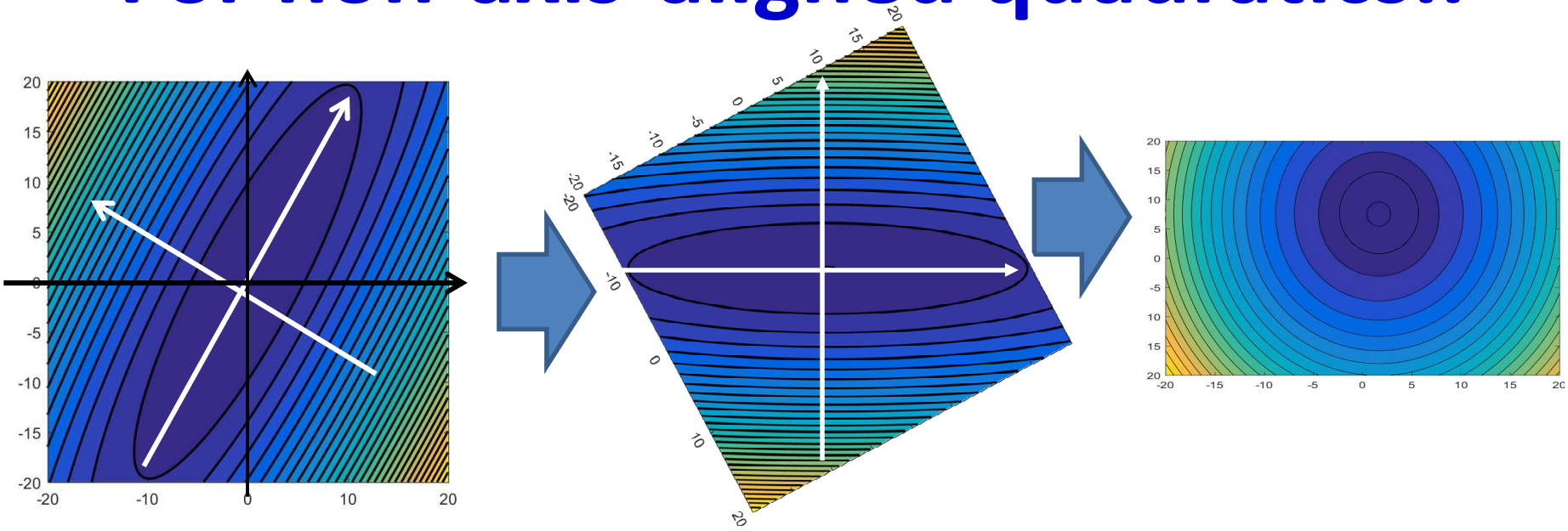


$$E = \frac{1}{2}\mathbf{w}^T\mathbf{A}\mathbf{w} + \mathbf{w}^T\mathbf{b} + c$$

$$E = \frac{1}{2}\sum_i a_{ii}w_i^2 + \sum_{i\neq j} a_{ij}w_i w_j + \sum_i b_i w_i + c$$

- If $\mathbf{A}$ is not diagonal, the contours are not axis-aligned
  - Because of the cross-terms $a_{ij}w_i w_j$
  - The major axes of the ellipsoids are the *Eigenvectors* of $\mathbf{A}$, and their diameters are proportional to the Eigen values of $\mathbf{A}$
- But this does not affect the discussion
  - This is merely a rotation of the space from the axis-aligned case
  - The component-wise optimal learning rates along the major and minor axes of the equal-contour ellipsoids will be different, causing problems
    - The optimal rates along the axes are Inversely proportional to the *eigenvalues* of $\mathbf{A}$

For non-axis-aligned quadratics..



- The component-wise optimal learning rates along the major and minor axes of the contour ellipsoids will differ, causing problems
  - Inversely proportional to the *eigenvalues* of $\mathbf{A}$

- This can be fixed as before by rotating and resizing the different directions to obtain the same *normalized* update rule as before:

$$\mathbf{w}^{(k+1)} = \mathbf{w}^{(k)} - \eta \mathbf{A}^{-1}\mathbf{b}$$