Optimization & Deep Learning

Yann Le Cun

Facebook AI Research,

Center for Data Science, NYU

Courant Institute of Mathematical Sciences, NYU

http://yann.lecun.com

The Convergence of Gradient Descent

Y LeCun

$$\omega \leftarrow \omega - \eta \frac{\partial E}{\partial \omega}$$

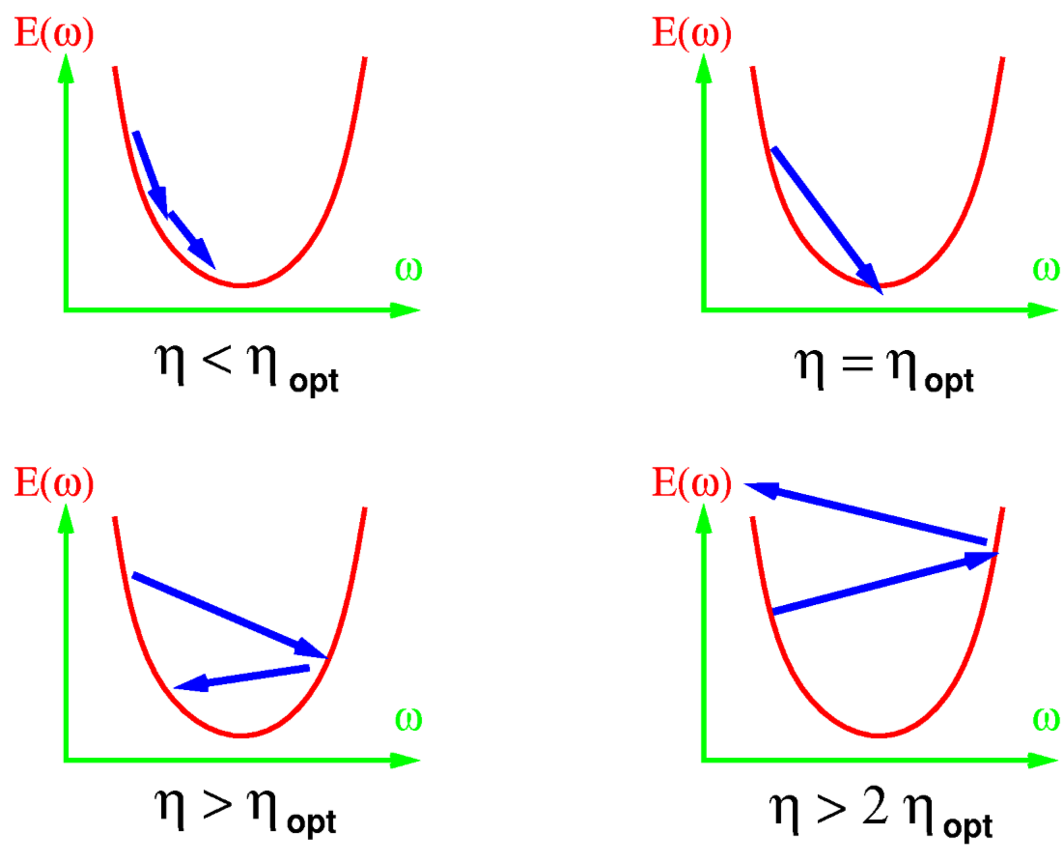
gradient of objective function

weight vector

learning rate

- Batch Gradient
- There is an optimal learning rate
- Equal to inverse 2nd derivative

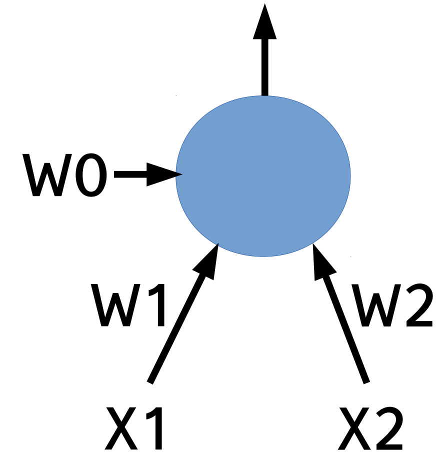
$$\boxed{\eta_{opt} = \left(\frac{\partial^2 E}{\partial \omega^2}\right)^{-1}}$$



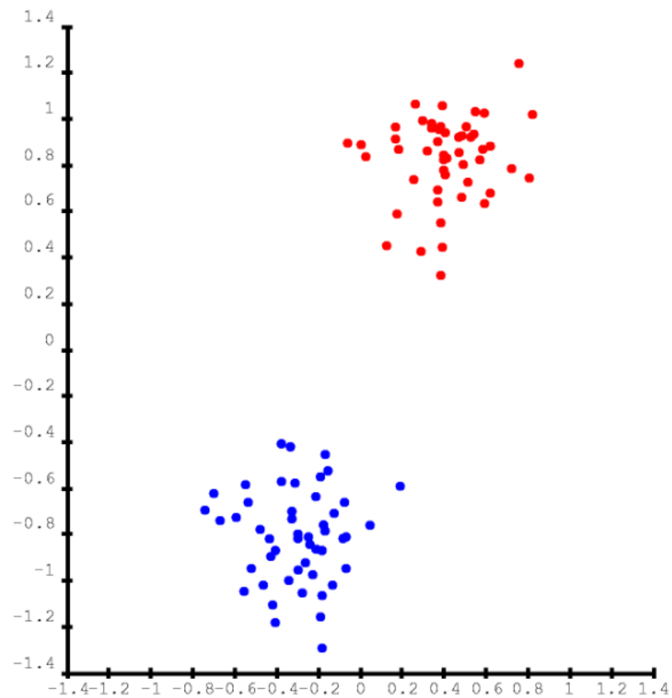
Let's Look at a single linear unit

Y LeCun

- Single unit, 2 inputs



- Quadratic loss
  - $E(W) = 1/p \sum_p (Y - W \cdot X_p)^2$
- Dataset: classification: Y=-1 for blue, +1 for red



- Hessian is covariance matrix of input vectors
  - $H = 1/p \sum X_p X_p^T$
- To avoid ill conditioning: normalize the inputs
  - Zero mean
  - Unit variance for all variable

Convergence is Slow When Hessian has Different Eigenvalues

Y LeCun

Batch Gradient, small learning rate

Learning rate:

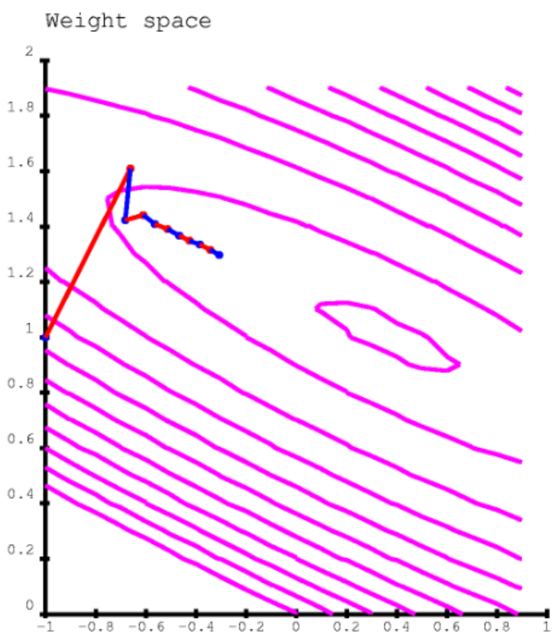
$\eta = 1.5$

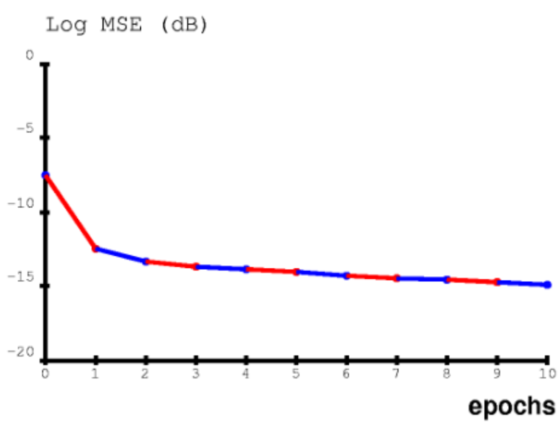
Hessian largest eigenvalue:

$\lambda_{max} = 0.84$

Maximum admissible Learning rate:

$\eta_{max} = 2.38$





Batch Gradient, large learning rate

Learning rate:

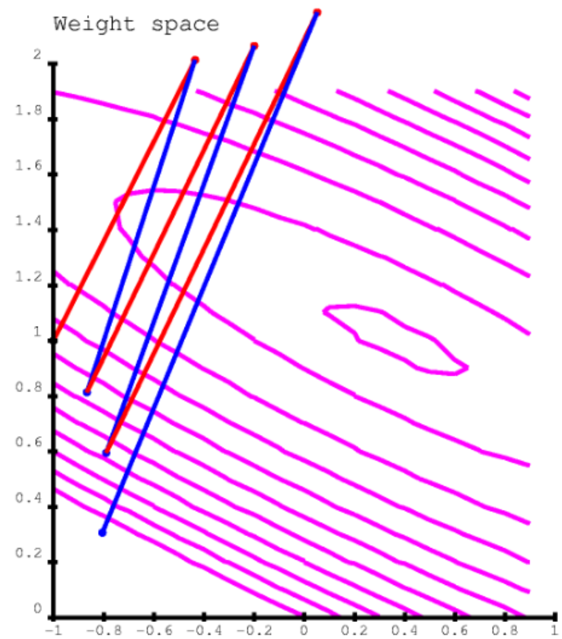
$\eta = 2.5$

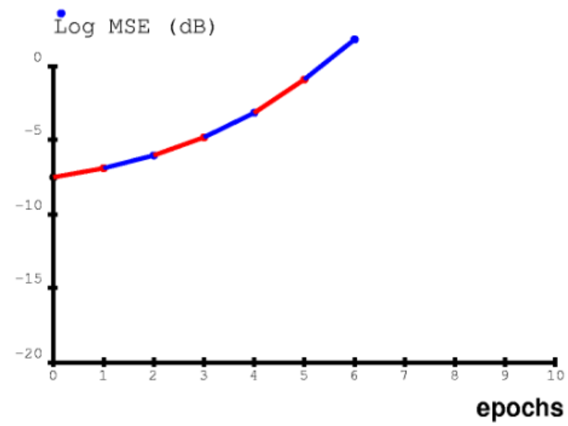
Hessian largest eigenvalue:

$\lambda_{max} = 0.84$

Maximum admissible Learning rate:

$\eta_{max} = 2.38$





Convergence is Slow When Hessian has Different Eigenvalues

- Batch Gradient, small learning rate

Learning rate:

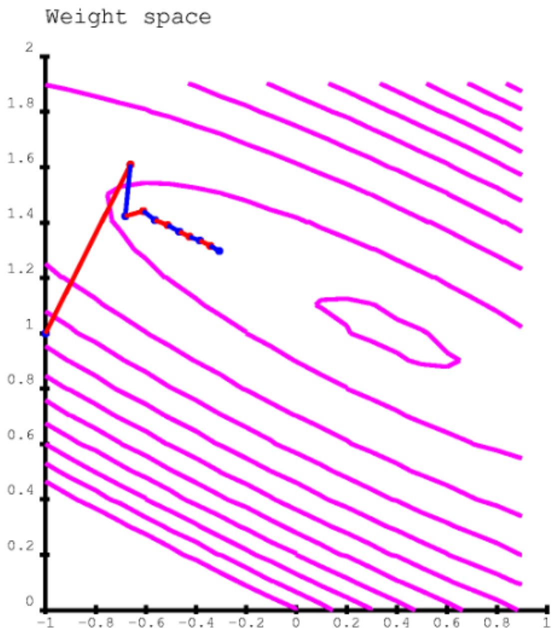
$\eta = 1.5$

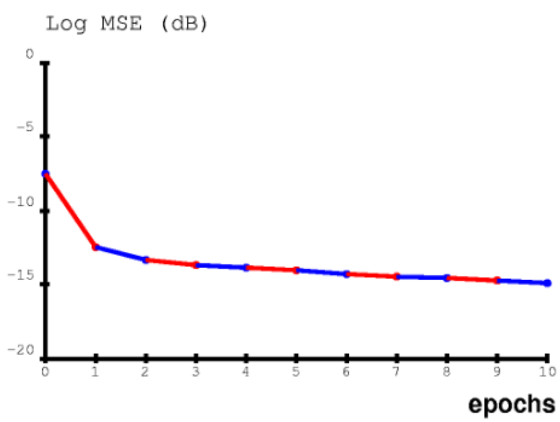
Hessian largest eigenvalue:

$\lambda_{max} = 0.84$

Maximum admissible Learning rate:

$\eta_{max} = 2.38$





- Stochastic Gradient: Much Faster
- But fluctuates near the minimum

Learning rate:

$\eta = 0.2$

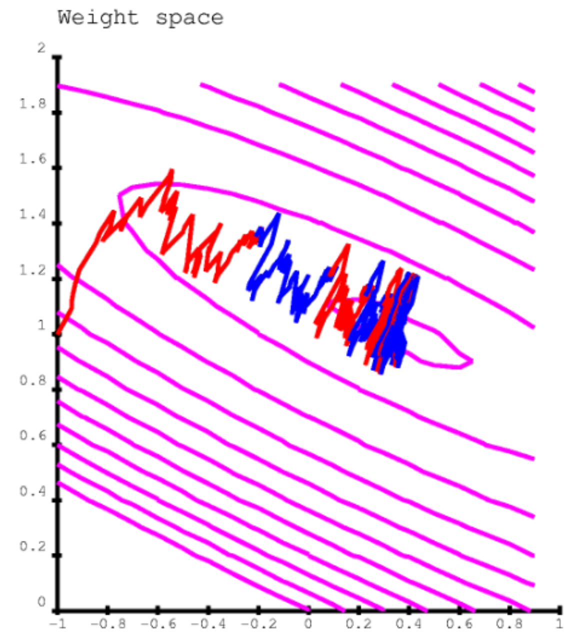
(equivalent to a batch learning rate of 20)

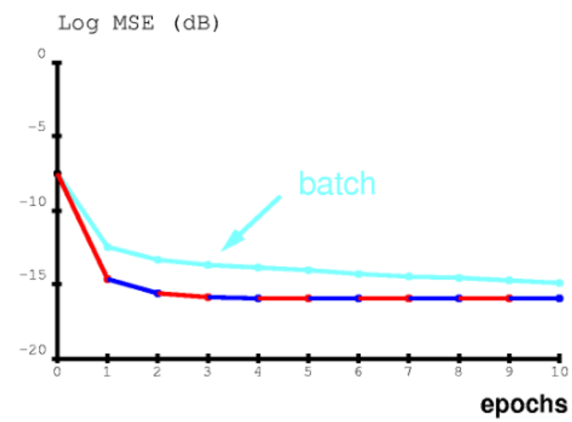
Hessian largest eigenvalue:

$\lambda_{max} = 0.84$

Maximum admissible Learning rate (for batch):

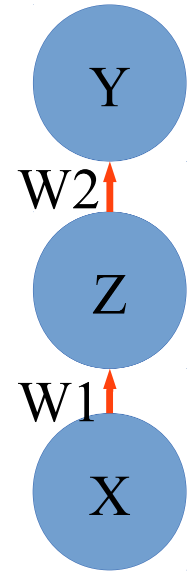
$\eta_{max} = 2.38$





Multilayer Nets Have Non-Convex Objective Functions

Y LeCun



- 1-1-1 network
  - Y = W1*W2*X
- trained to compute the identity function with quadratic loss
  - Single sample X=1, Y=1 L(W) = (1-W1*W2)^2
- Solution: W2 = 1/W2 hyperbola.