

# Lecture Notes for **Machine Learning in Python**



Professor Eric Larson  
**Optimizing Neural Networks**

# Class Logistics and Agenda

- Logistics
  - Grading
  - Team canvas assignment turn in
- Agenda:
  - Practical Multi-layer Architectures
  - Programming Examples and Adaptive Eta's
- Next Time: More MLPs

# Class Overview, by topic

Table Data  
Visualization

Numpy, Pandas, Seaborn  
Overviews with some in-depth discussion

Dimension  
Reduction and  
Image Processing

Scikit-learn, Scikit Image,  
Intuition only, Some mathematics

Linear and  
Logistic  
Regression

Numpy, Recreate API for Scikit-learn  
Detailed mathematics for simple optimization  
intuition for advanced optimization

Neural Networks  
and Back Prop.

Numpy  
Detailed mathematics for NN operations

Wide and Deep  
Networks

Convolutional  
Networks

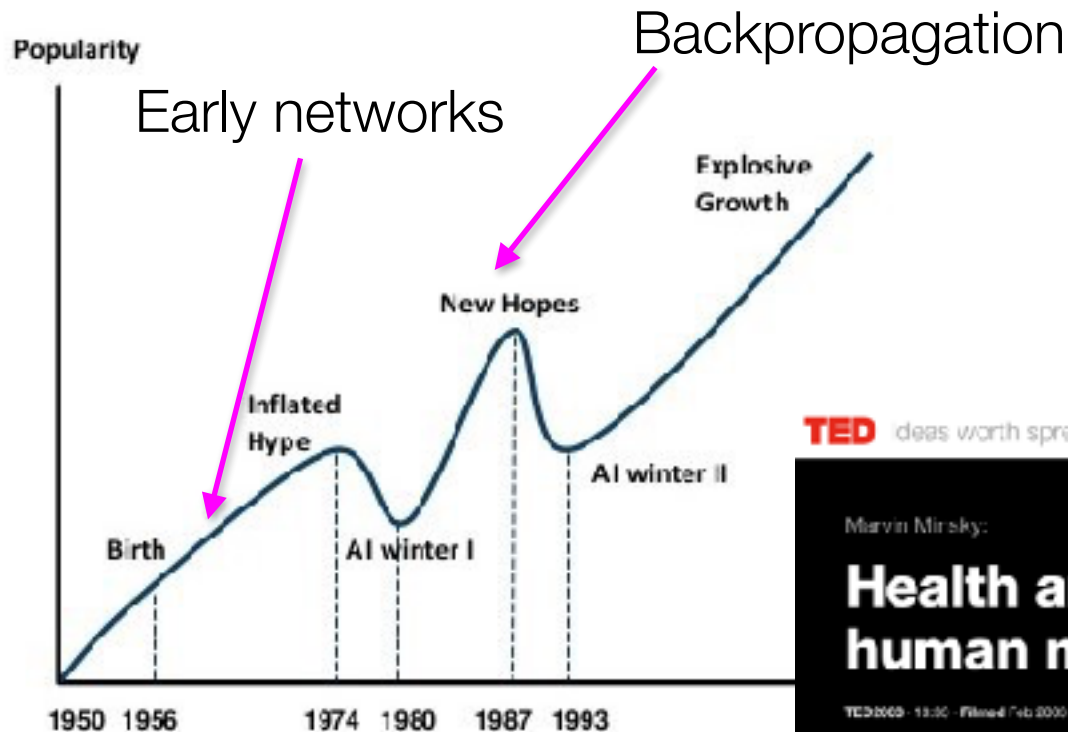
Recurrent  
Networks

Keras, Tensorflow  
Intuition, Detailed implement.

Ethics in  
Language Models

ConceptNet  
Case studies

# The First AI Winter (if not covered already)



TED ideas worth spreading

WATCH DISCOVER ATTEMPT PARTICIPATE

Marvin Minsky:

## Health and the human mind

TED2009 - 13:30 - Filmed Feb 2009

21 subtitle languages

View interactive transcript



# The Rosenblatt-Widrow-Hoff Dilemma

- 1960's: Rosenblatt got into a public academic argument with Marvin Minsky and Seymour Papert

"Given an elementary  $\alpha$ -perceptron, a stimulus world  $W$ , and any classification  $C(W)$  for which a solution exists; let all stimuli in  $W$  occur in any sequence, provided that each stimulus must reoccur in finite time; then beginning from an arbitrary initial state, an error correction procedure will always yield a solution to  $C(W)$  in finite time..."
- Minsky and Papert publish limitations paper, 1969:

"the style of research being done on the perceptron is doomed to failure because of these limitations."
- Widrow and Rosenblatt try to build bigger networks without limitations and fail
  - ★ Neural Networks research **basically stops** for **17 years**
- **Until:** researchers revisit training bigger networks
  - ★ Neural Networks with multiple layers

# Stable Training of Multi-layer Architectures: history

- 1986: *Rumelhart, Hinton, and Williams* popularize gradient calculation for multi-layer network
  - *technically* introduced by Werbos in 1982
- **difference:** Rumelhart *et al.* validated ideas with a computer
- until this point no one could train a multiple layer network consistently
- algorithm is popularly called **Back-Propagation**
- wins pattern recognition prize in 1993, becomes de-facto machine learning algorithm until: SVMs and Random Forests in ~2004
- would eventually see a resurgence for its ability to train algorithms for Deep Learning applications: **Hinton is widely considered the founder of deep learning**

David Rumelhart

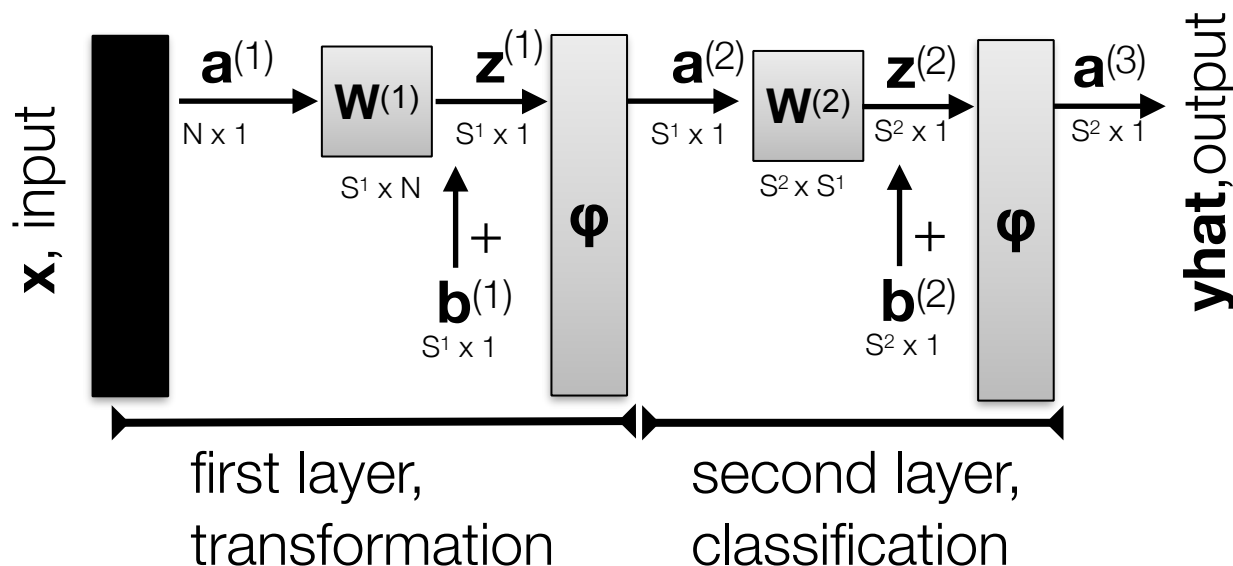


Geoffrey Hinton



# Review: Back propagation

- Optimize all weights of network at once
- Steps:
  1. Forward propagate to get all  $\mathbf{Z}^{(l)}$ ,  $\mathbf{A}^{(l)}$
  2. Get final layer gradient
  3. Back propagate sensitivities
  4. Update each  $\mathbf{W}^{(l)}$

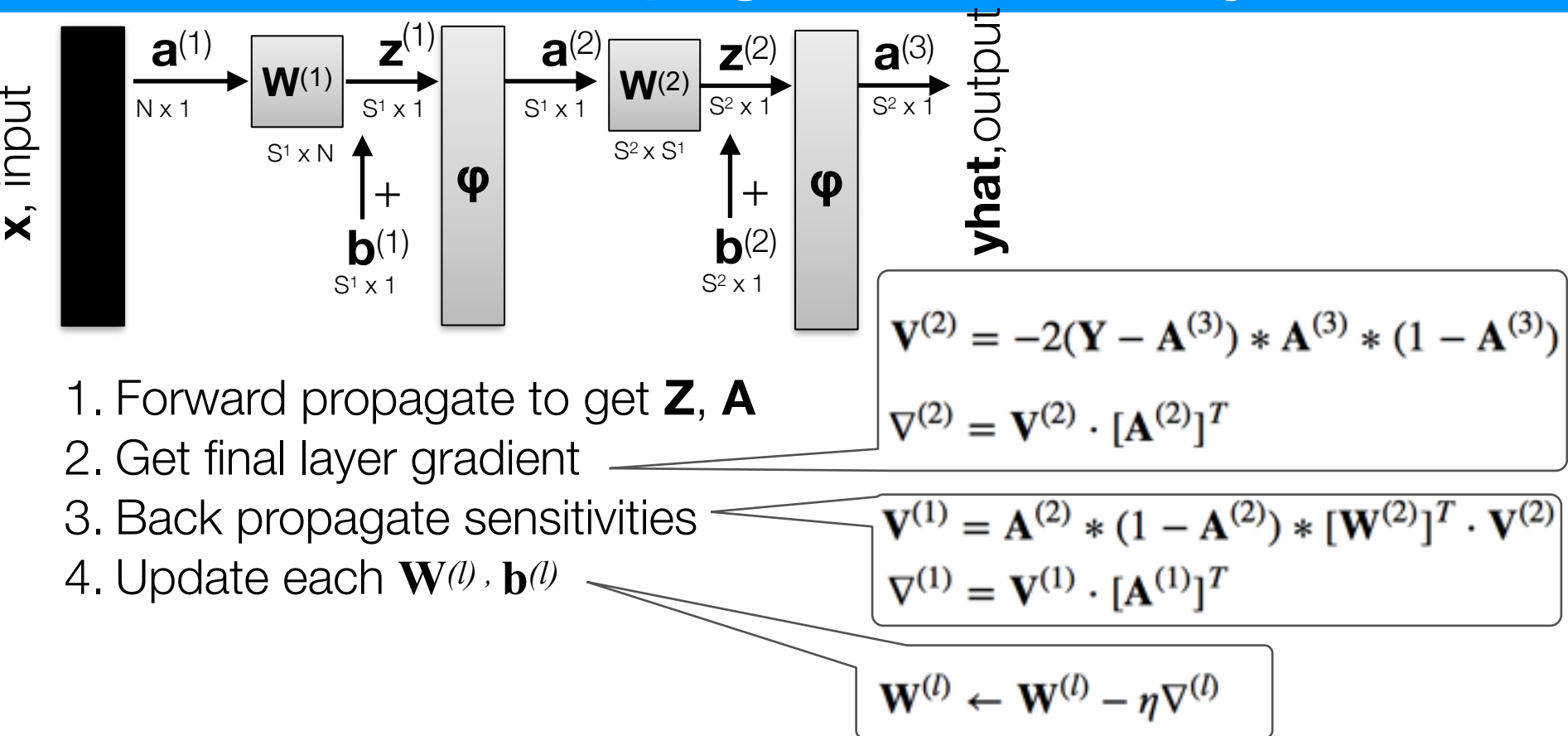


$$J(\mathbf{W}) = \| \mathbf{Y} - \hat{\mathbf{Y}} \|^2$$

$$w_{i,j}^{(l)} \leftarrow w_{i,j}^{(l)} - \eta \frac{\partial J(\mathbf{W})}{\partial w_{i,j}^{(l)}}$$

\*\*Recall from Flipped Assignment!

# Review: Back Propagation Summary



Where is the problem of **vanishing gradients** introduced?

\*\*Recall from Flipped Assignment!



## 07a. MLP Neural Networks with bias.ipynb

same as Flipped Assignment!  
with regularization  
and vectorization  
and mini-batching

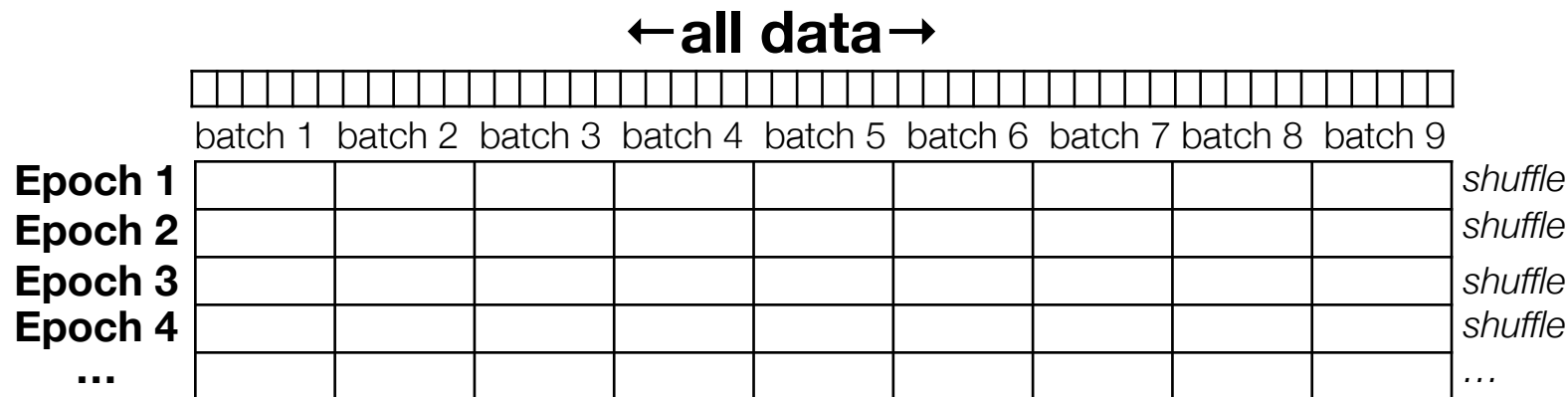


A.  $\mathbf{z} = \mathbf{W} \cdot \mathbf{a}_{bias}$  old notebooks

B.  $\mathbf{z} = \mathbf{W} \cdot \mathbf{a} + \mathbf{b}$  new notebook!

# Mini-batching

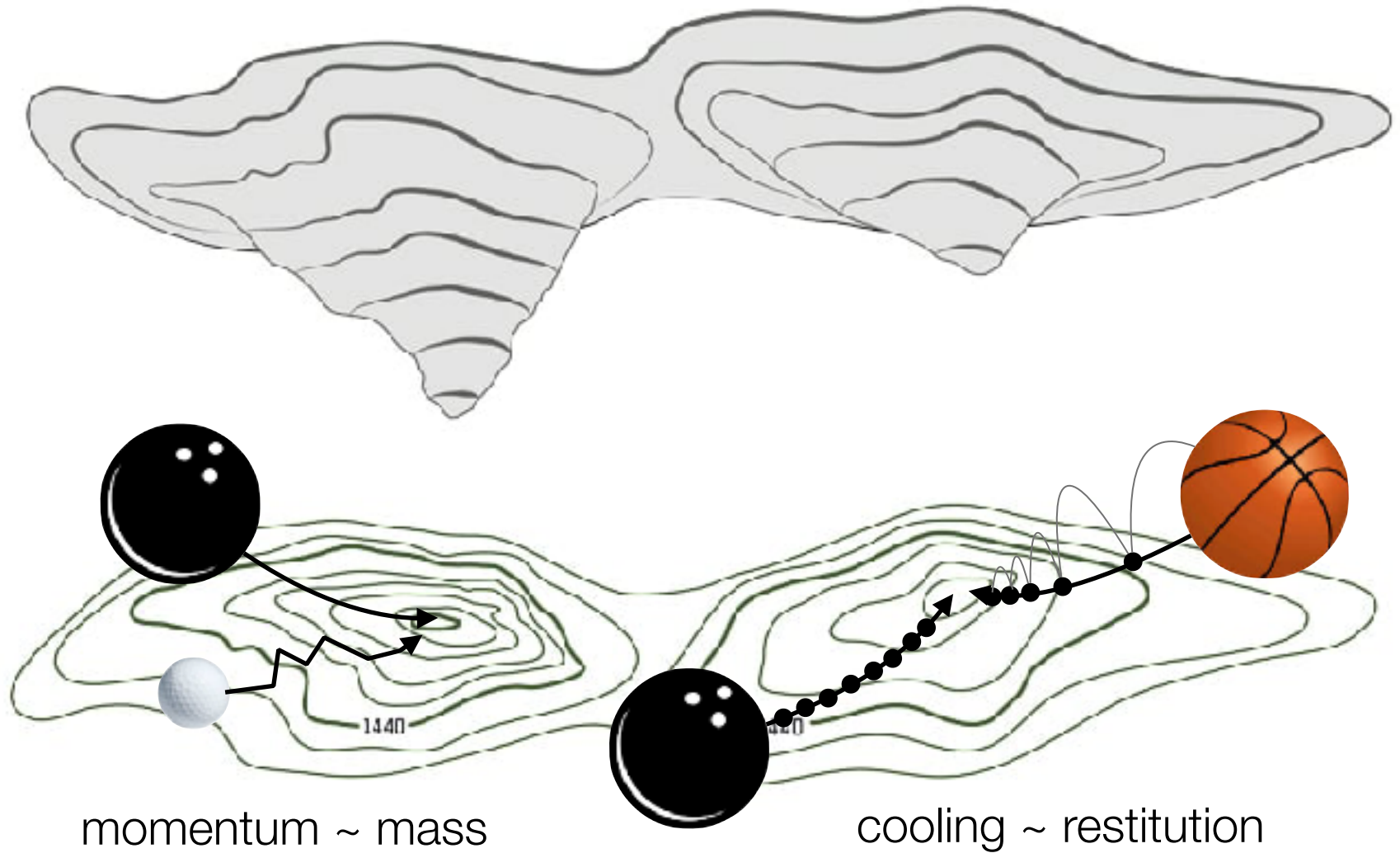
- Numerous instances to find one gradient update
  - **solution:** mini-batch



*shuffle ordering **each epoch** and update  $W$ 's after **each batch***

- **new problem:** mini-batch gradient updates can be erratic and there might be many local optima...
  - **solutions:**
    - momentum
    - adaptive learning rate (cooling)

# Momentum and Cooling Intuition



# Momentum

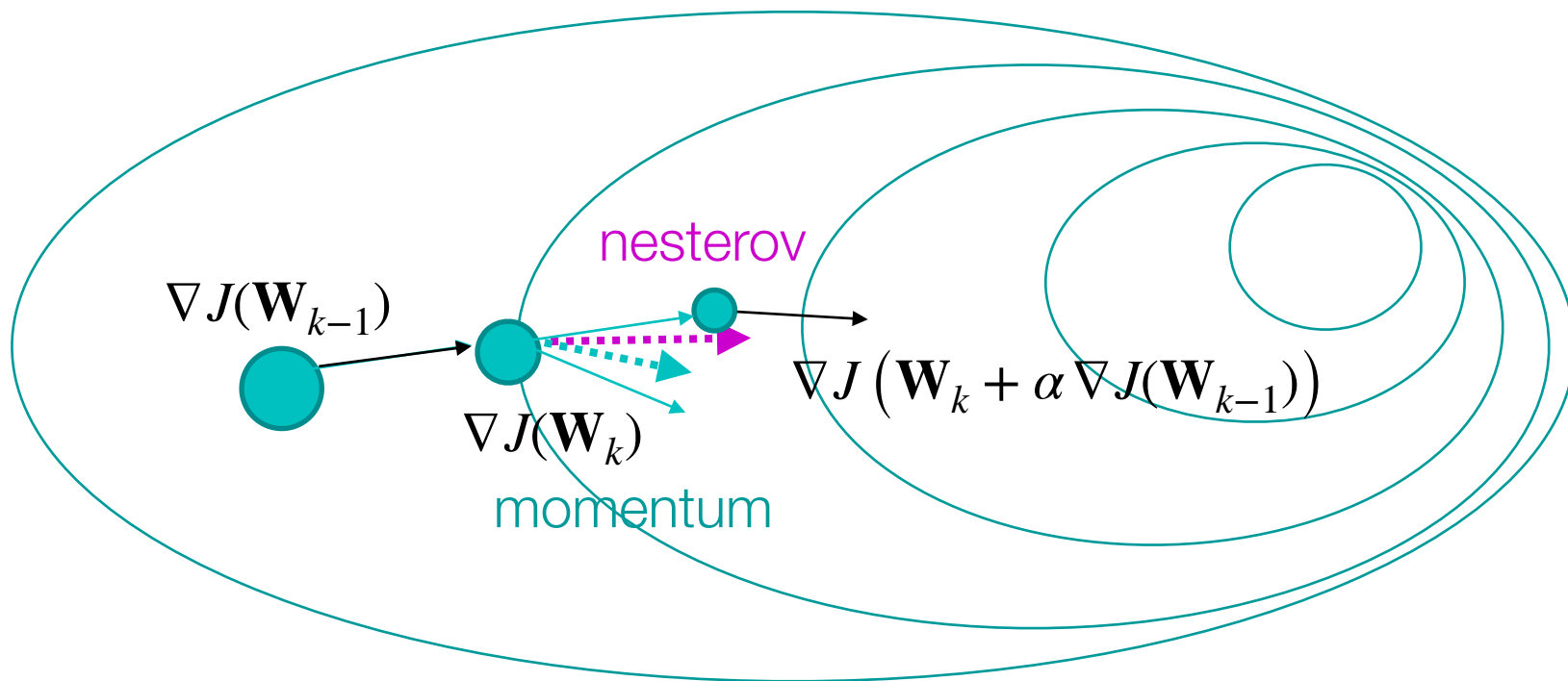
$$\mathbf{W}_{k+1} = \mathbf{W}_k - \rho_k$$

- Momentum

$$\rho_k = \eta \cdot \nabla J(\mathbf{W}_k) + \alpha \cdot \rho_{k-1}$$

- Nesterov's Accelerated Gradient

$$\rho_k = \underbrace{\beta \nabla J(\mathbf{W}_k + \alpha \nabla J(\mathbf{W}_{k-1})) + \alpha \nabla J(\mathbf{W}_{k-1})}_{\text{step twice}}$$



# Cooling (Learning Rate Reduction)

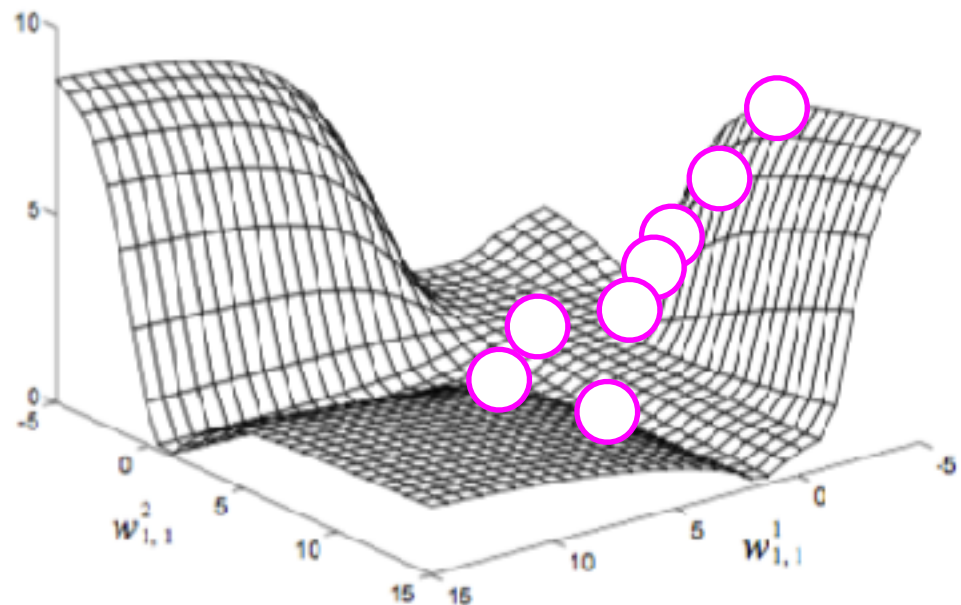
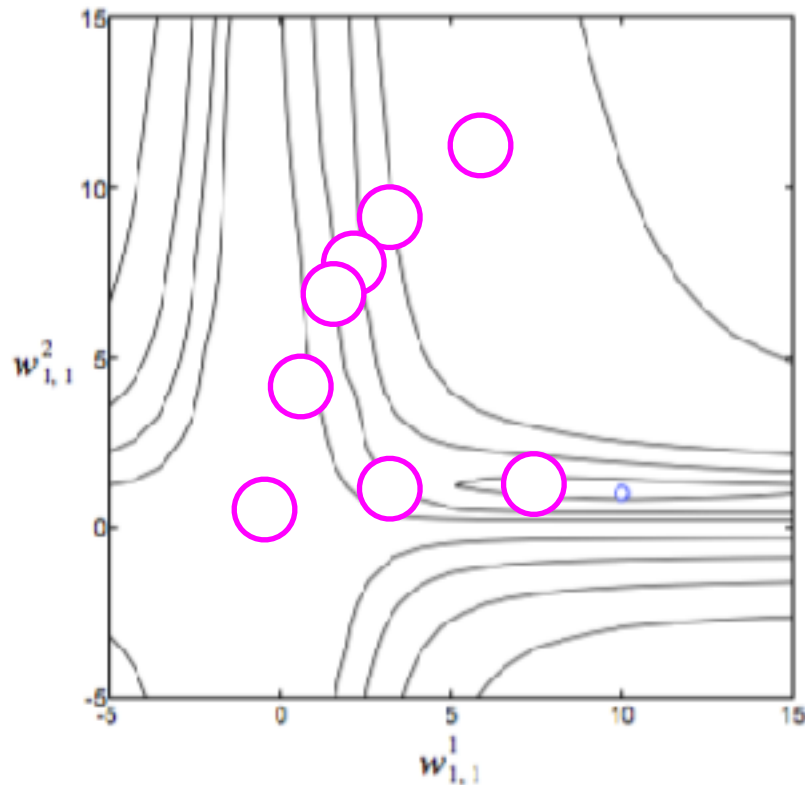
- Fixed Reduction at Each Epoch,  $k$

$$\eta_k = \eta_0 \cdot d^{\lfloor \frac{k_{max}}{k} \rfloor} \quad \text{drop by } d \text{ every } k_d \text{ epochs}$$

- Adjust on Plateau

- make smaller when  $J$  rapidly changes
- make bigger when  $J$  not changing much

$$\eta_k = \eta_0^{(1+k \cdot d)} \quad \text{drop a little every epoch}$$



## 07. MLP Neural Networks.ipynb

### **comparison:**

mini-batch

momentum

adaptive learning rate

L-BFGS (if time)

$$\rho_k = \eta \cdot \nabla J(\mathbf{W}_k) + \alpha \cdot \rho_{k-1}$$

$$\eta_k = \eta_0^{(1+k \cdot d)}$$

