University of Amsterdam
Informatics Institute

AMLAB
Amsterdam
Machine Learning Lab

Machine Learning 1

Lecture 8.4 - Supervised Learning
Neural Networks - Training

Erik Bekkers

(Bishop 5.2)

Slide credits: Patrick Forré and Rianne van den Berg

Image credit: Kirillm | Getty Images

Neural Networks: Parameter Optimization

- For each task a different loss function $E(\mathbf{w})$
- Optimal parameters $\mathbf{w}^* = \arg\min_{\mathbf{w}} E(\mathbf{w})$
- Problem: E(**w**) is not convex in **w,** so several local minima can exist.
- How to reach the global minimum?

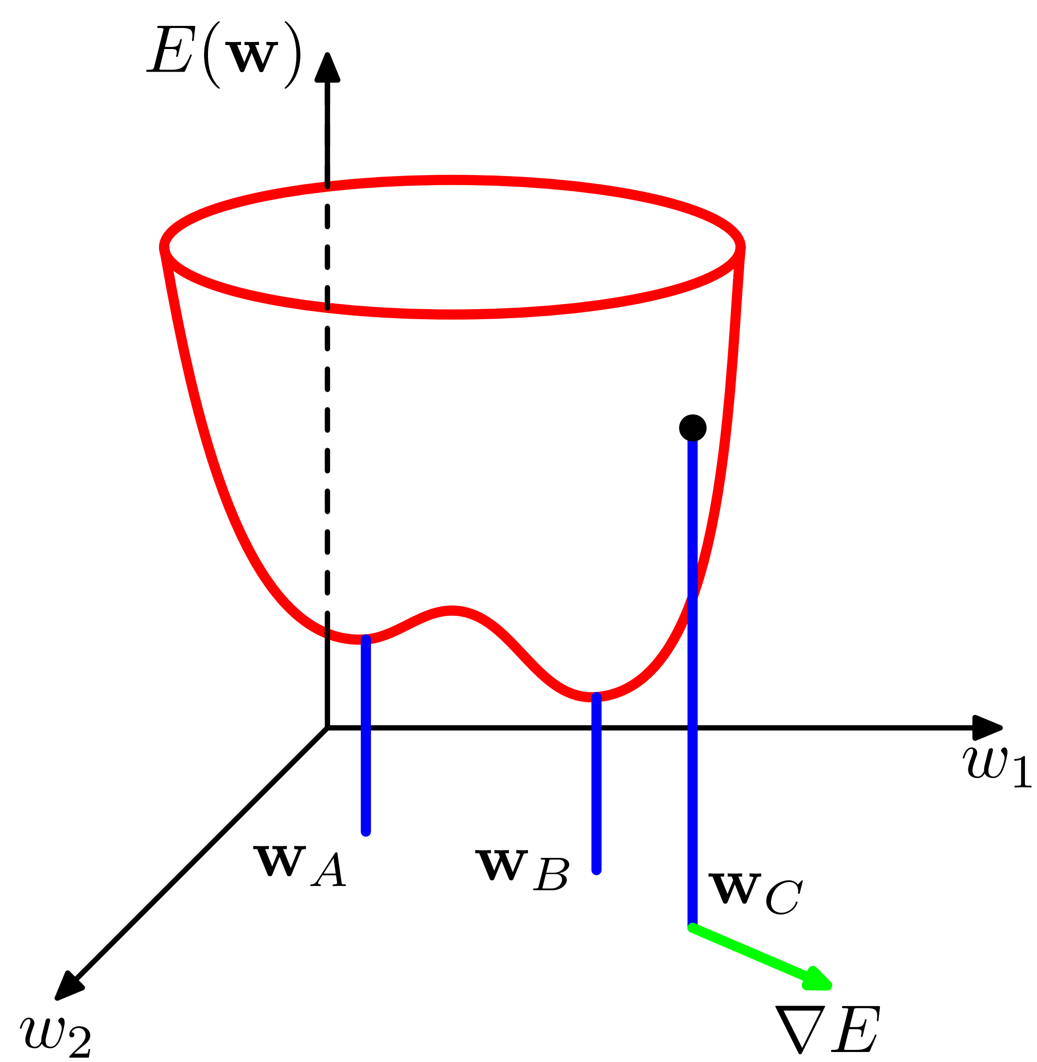


Figure: E(**w**) as surface in weight space (Bishop 5.4)

Gradient Descent vs. Stochastic Gradient Descent

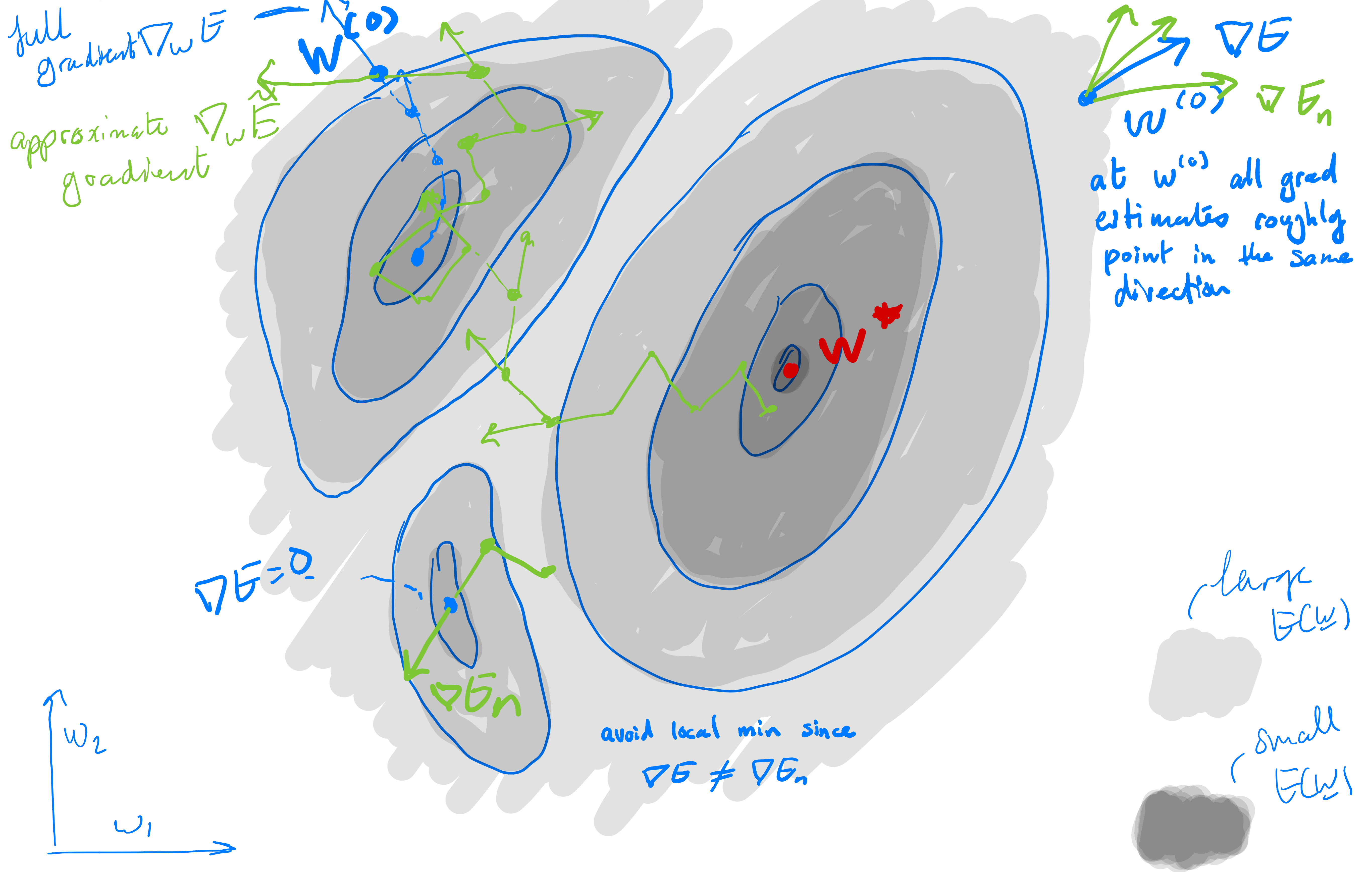
- Gradient Descent: $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E(\mathbf{w}^{(\tau)})$
- Will easily get stuck in local minimum where $\nabla E(\mathbf{w}) = 0$
- Use $E(\mathbf{w}) = \sum_{n=1}^{N} E_n(\mathbf{w})$ to implement SGD:
  - Carefully choose learning rate $\eta > 0$
  - Randomly initialize $\mathbf{w}^{(0)}$
  - Randomly/sequentially choose $\mathbf{x}_n$ and update $\mathbf{w}$:

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)}) \quad = \nabla \tilde{E}$$

  - Convergence to area around local minimum

- Can also use minibatches $\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla \sum_{i=1}^{M} E_i(\mathbf{w}^{(\tau)})$

Gradient Descent vs. Stochastic Gradient Descent



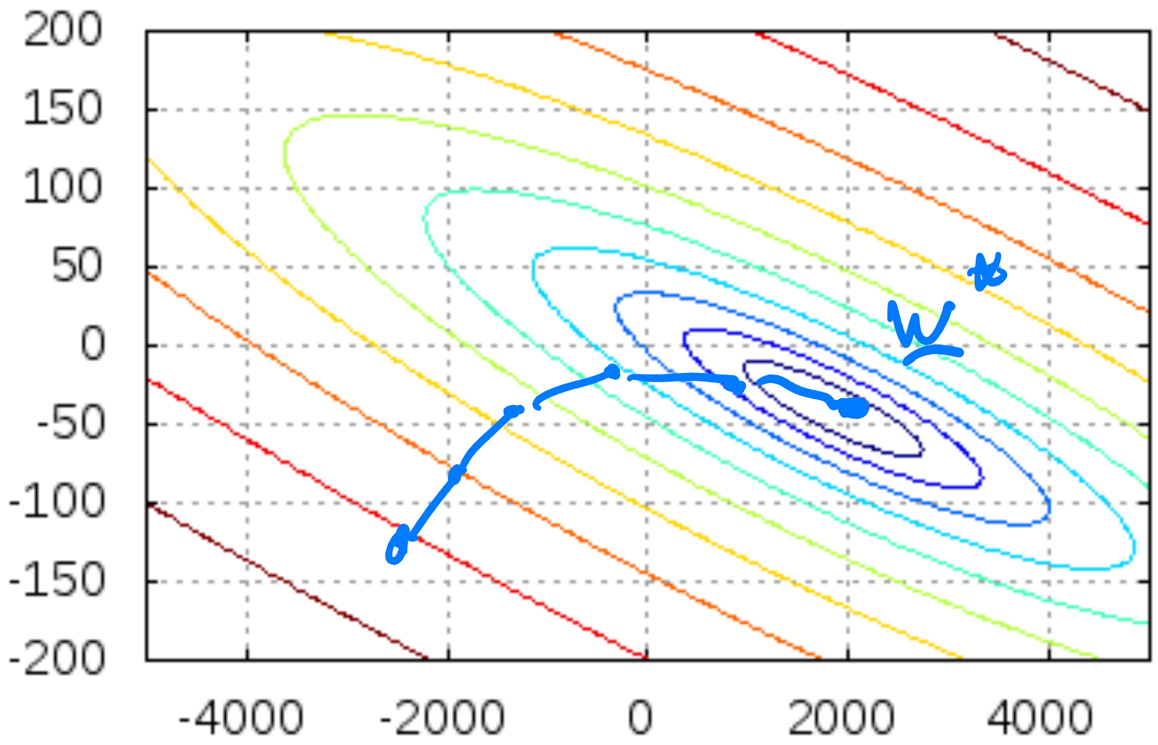
Gradient Descent vs. Stochastic Gradient Descent

- If learning rate is too small: slow convergence
- If learning rate is too large: oscillations around local minimum
- Use learning rate scheduling with smaller learning rate over time
- At the beginning of learning all gradients $\nabla E_n(\mathbf{w})$ will roughly point in the same general direction. Full batch gradient descent computes redundant number of gradients!
- SGD is more likely to escape a local minimum since

$\nabla E(\mathbf{w}) = 0$ does **not** necessarily imply $\nabla E_n(\mathbf{w}) = 0$!

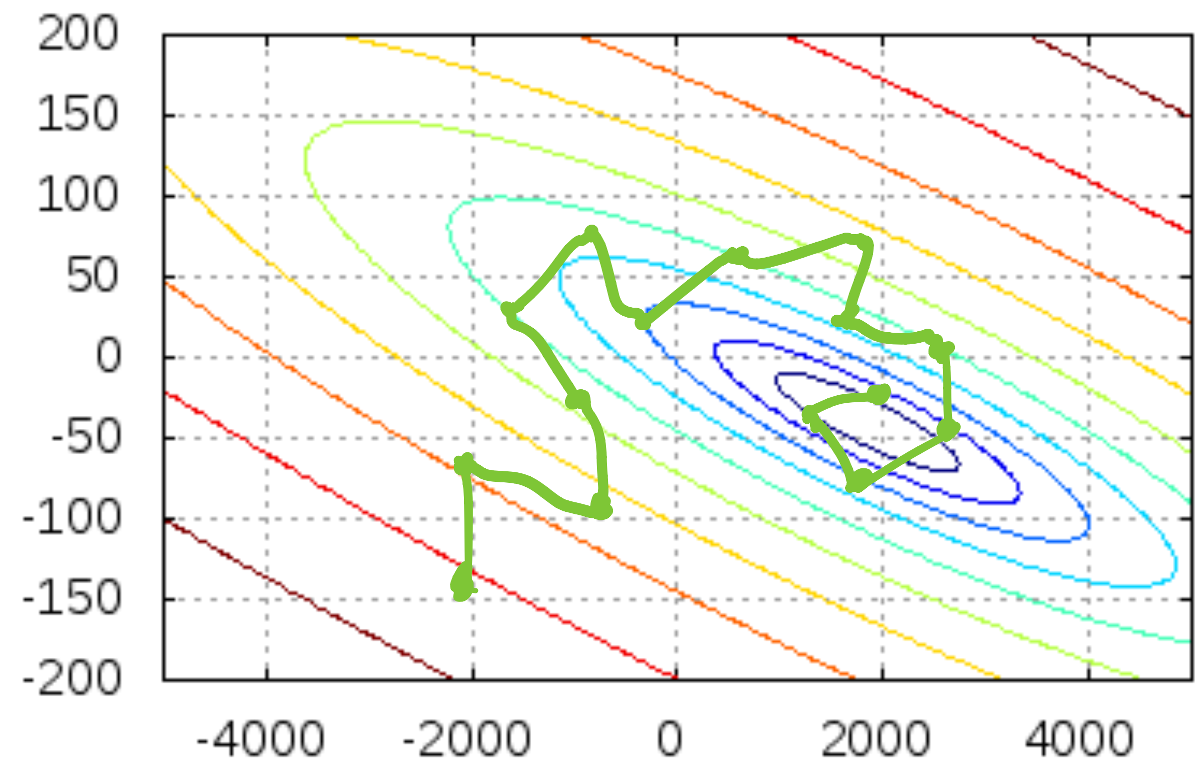
Gradient Descent vs. Stochastic Gradient Descent

GD



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

SGD



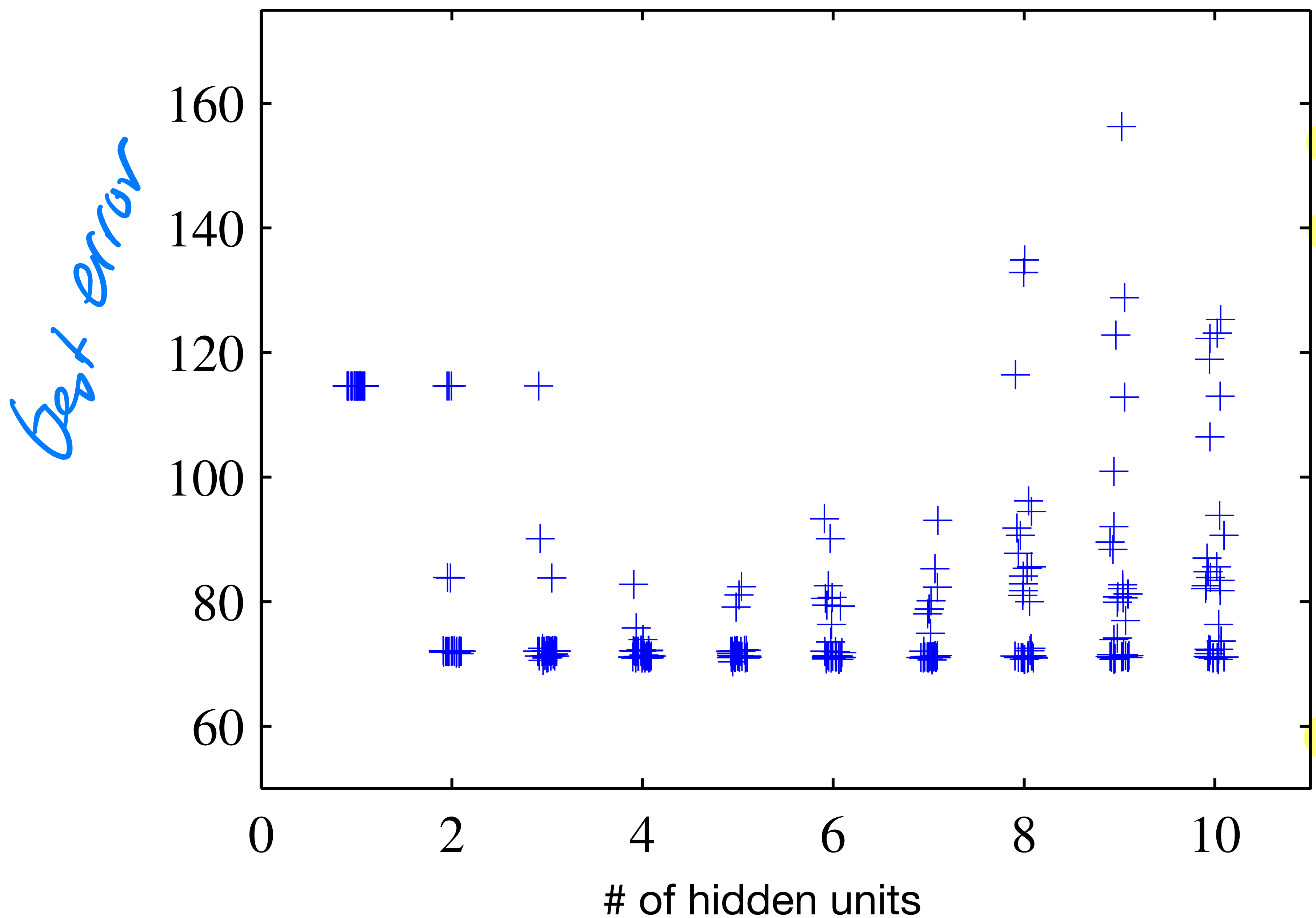
requires more steps

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \nabla E_n(\mathbf{w}^{(\tau)})$$

but ∇E_n is fast to compute!

Example: Test Errors and Local Minima

Restart for different random initial $\mathbf{w}^{(0)}$ to end up in different local minima



Always run with several inits

⇓

uncertainty on performance

Figure: sum-of-squares test error vs. network size (# of hidden units) for 30 random starts each (Bishop 5.10)