Practical Methodology

Lecture slides for Chapter 11 of *Deep Learning*

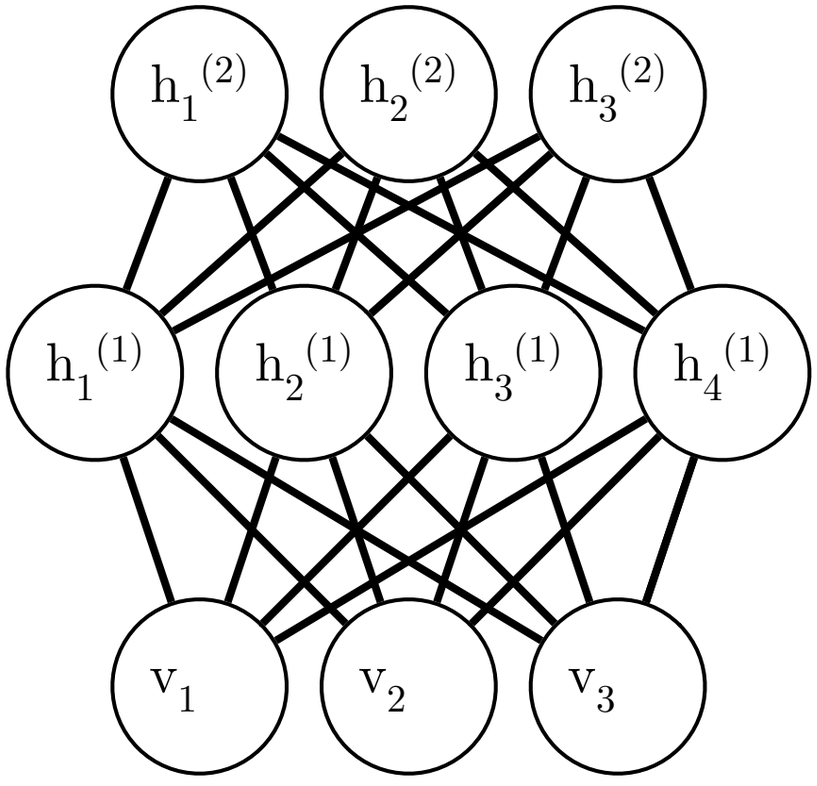
www.deeplearningbook.org

Ian Goodfellow

2016-09-26

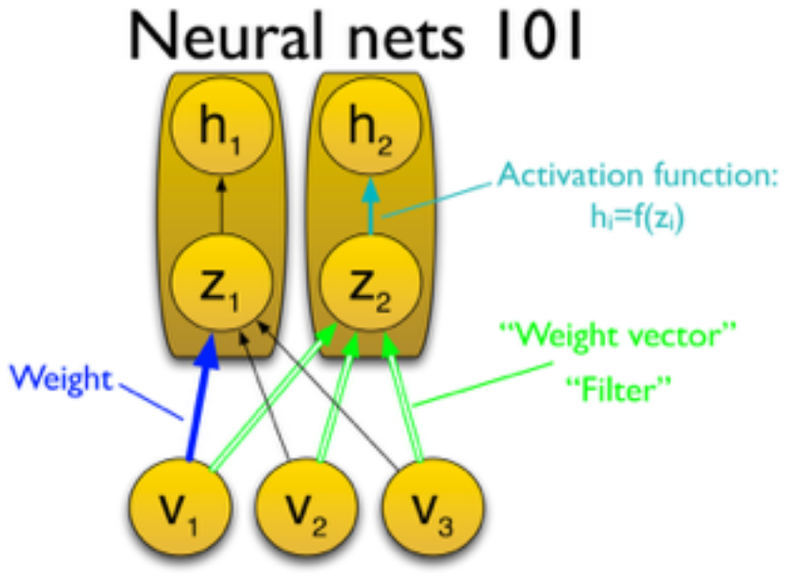
What drives success in ML?

Arcane knowledge of dozens of obscure algorithms?

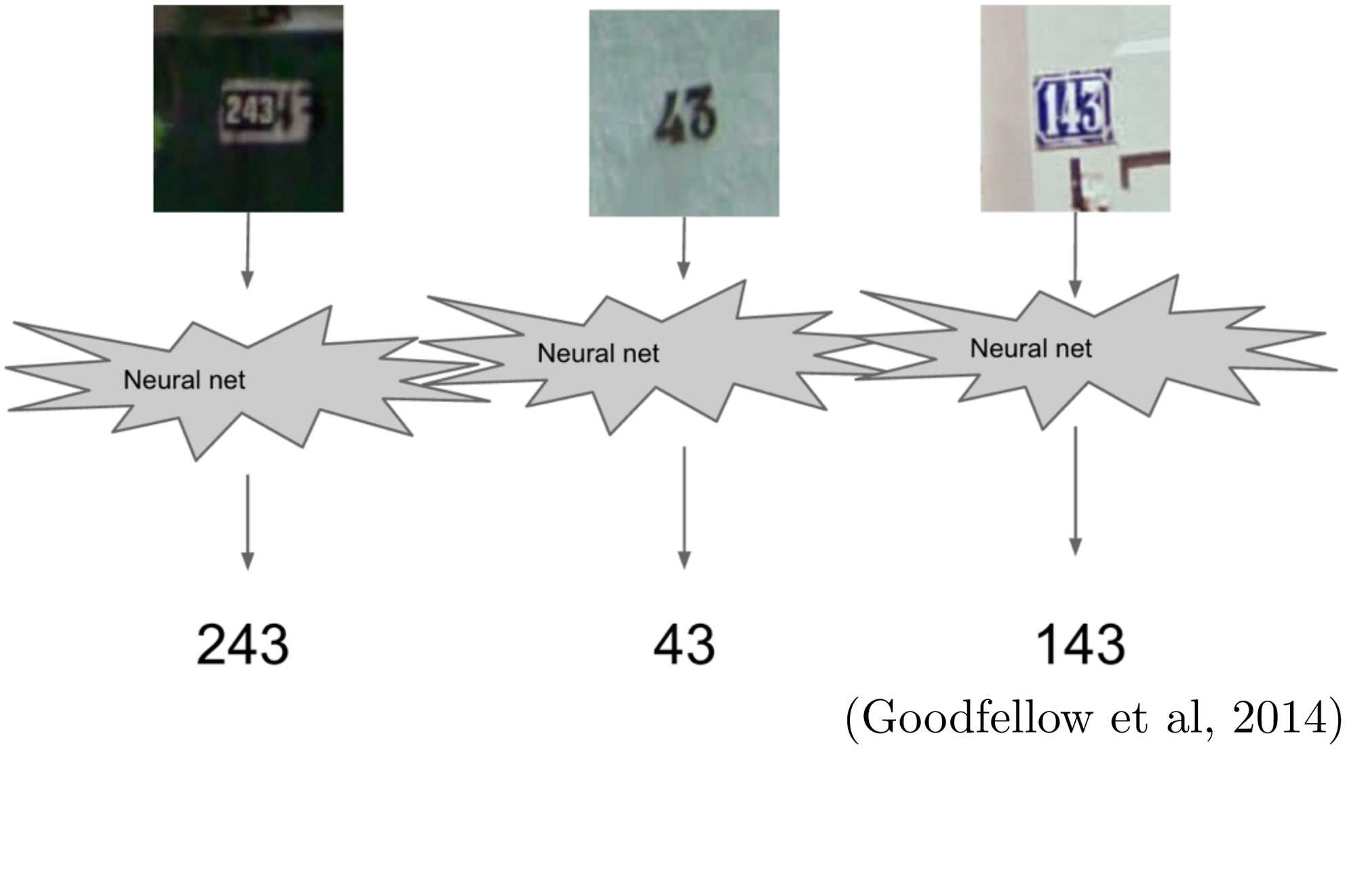


Mountains of data?

Knowing how to apply 3-4 standard techniques?



Example: Street View Address Number Transcription



(Goodfellow et al, 2014)

Three Step Process

- Use needs to define metric-based goals
- Build an end-to-end system
- Data-driven refinement

Identify Needs

- High accuracy or low accuracy?
- Surgery robot: high accuracy
- Celebrity look-a-like app: low accuracy

Choose Metrics

- Accuracy? (% of examples correct)
- Coverage? (% of examples processed)
- Precision? (% of detections that are right)
- Recall? (% of objects detected)
- Amount of error? (For regression problems)

End-to-end System

- Get up and running ASAP
- Build the simplest viable system first
- What baseline to start with though?
  - Copy state-of-the-art from related publication

Deep or Not?

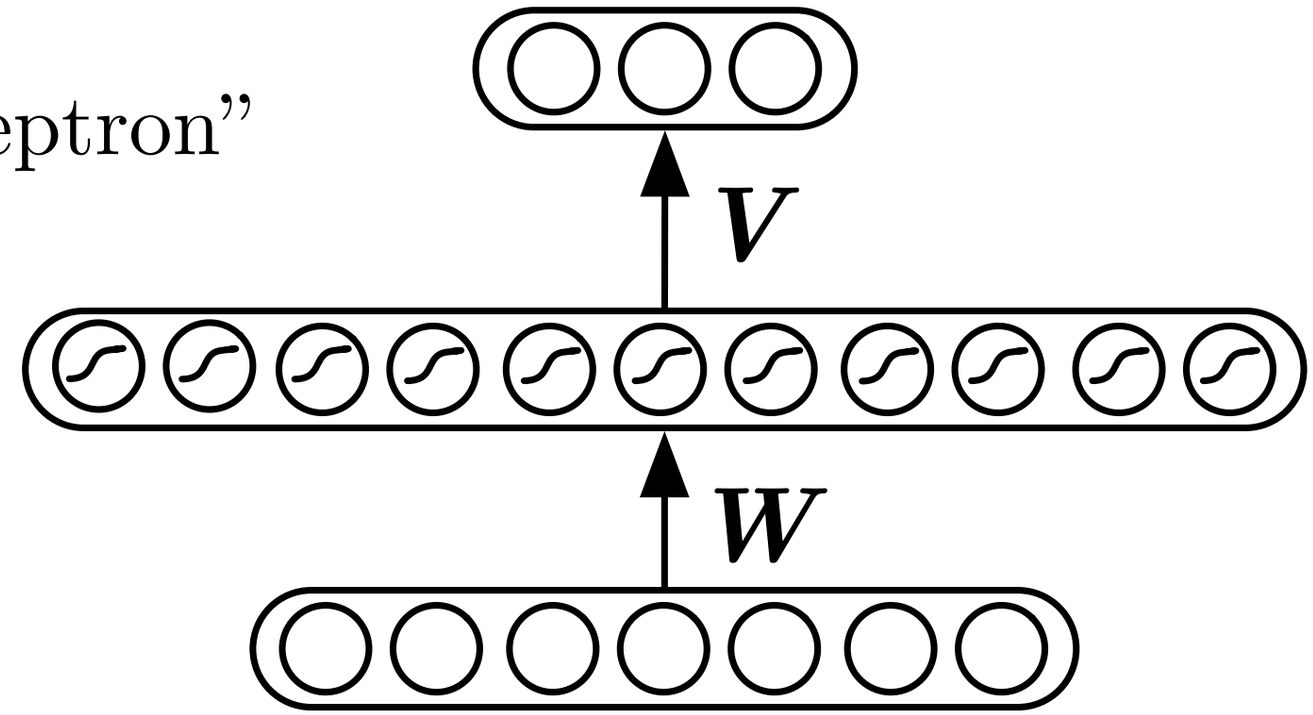
- Lots of noise, little structure -> not deep
- Little noise, complex structure -> deep
- Good shallow baseline:
  - *Use what you know*
  - Logistic regression, SVM, boosted tree are all good

Choosing Architecture Family

- No structure -> fully connected
- Spatial structure -> convolutional
- Sequential structure -> recurrent

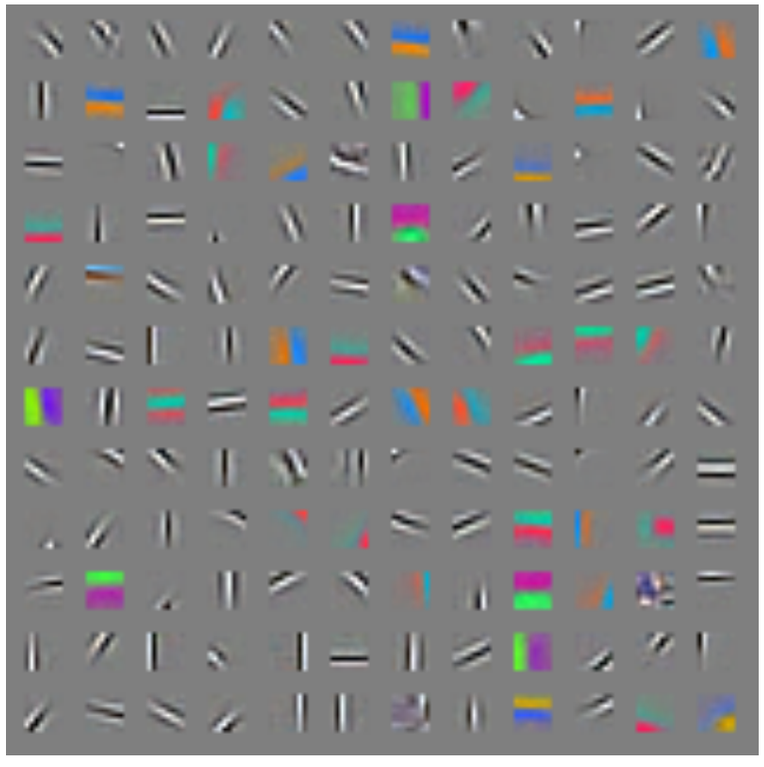
Fully Connected Baseline

- 2-3 hidden layer feed-forward neural network
  - AKA "multilayer perceptron"
- Rectified linear units
- Batch normalization
- Adam
- Maybe dropout



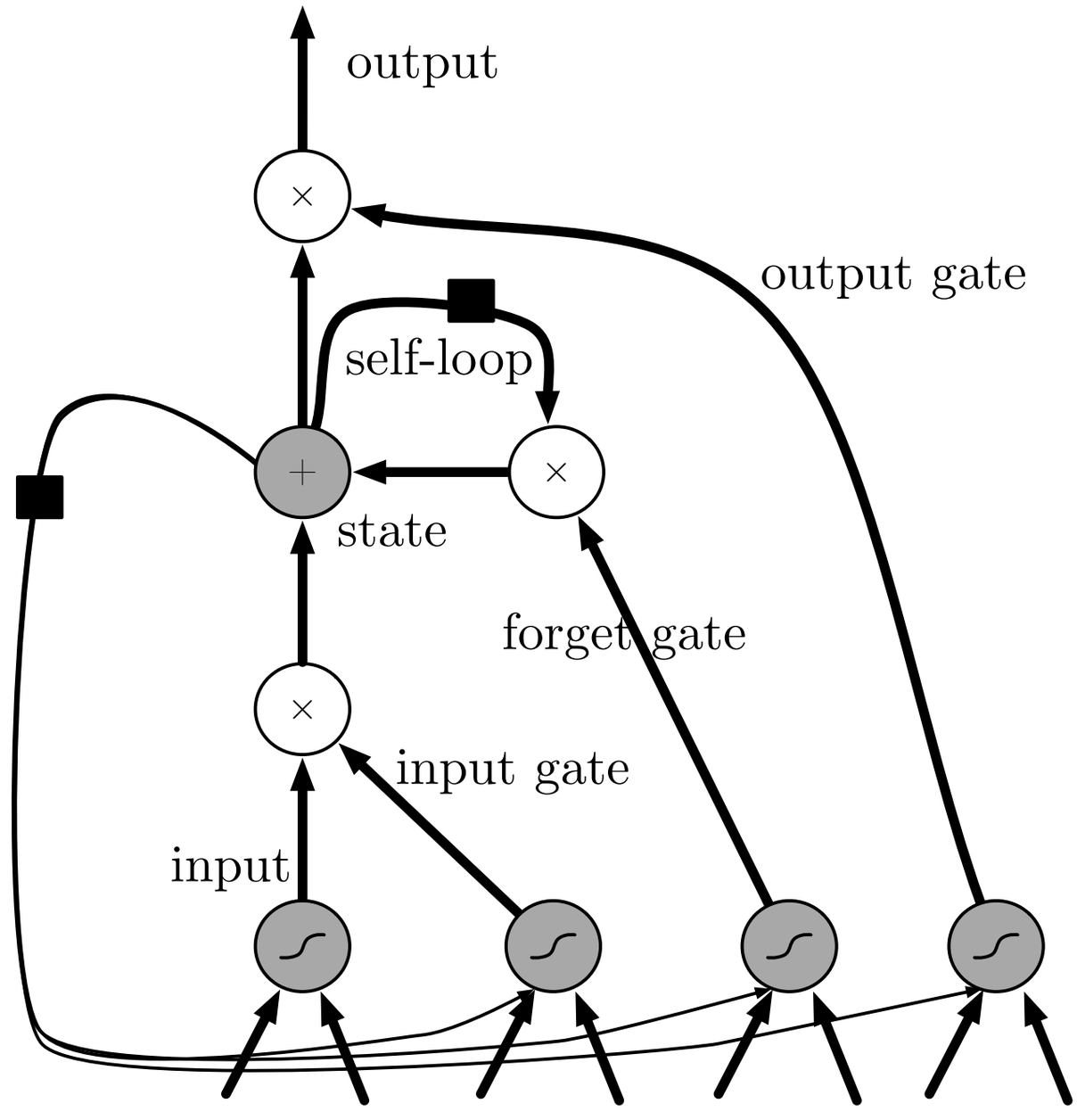
Convolutional Network Baseline

- Download a pretrained network
- Or copy-paste an architecture from a related task
- Or:
  - Deep residual network
  - Batch normalization
  - Adam



Recurrent Network Baseline

- LSTM
- SGD
- Gradient clipping
- High forget gate bias



Data-driven Adaptation

- Choose what to do based on data
- Don't believe hype
- Measure train and test error
  - "Overfitting" versus "underfitting"

High Train Error

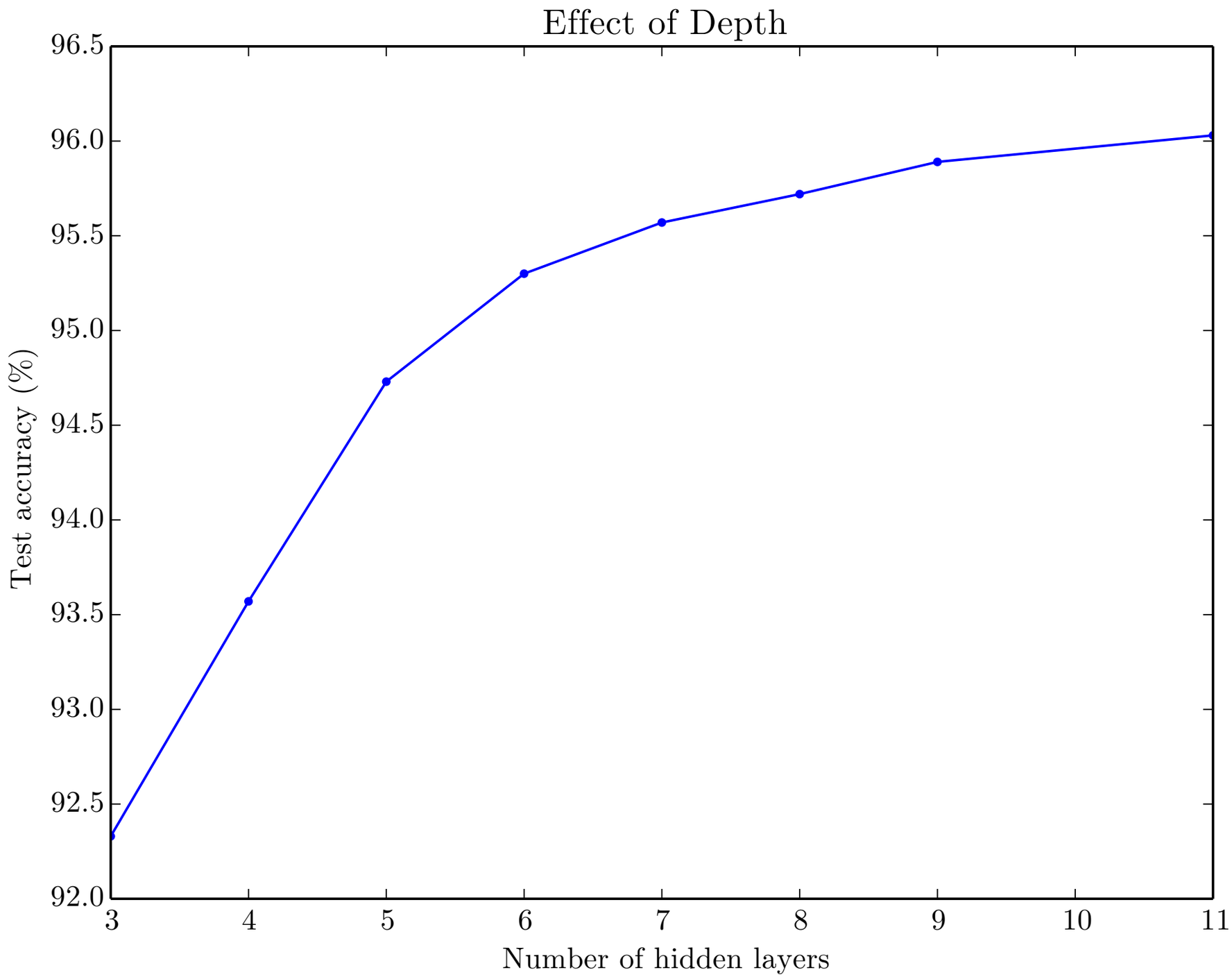
- Inspect data for defects
- Inspect software for bugs
  - Don't roll your own unless you know what you're doing
- Tune learning rate (and other optimization settings)
- Make model bigger

Checking Data for Defects

- Can a human process it?

26624

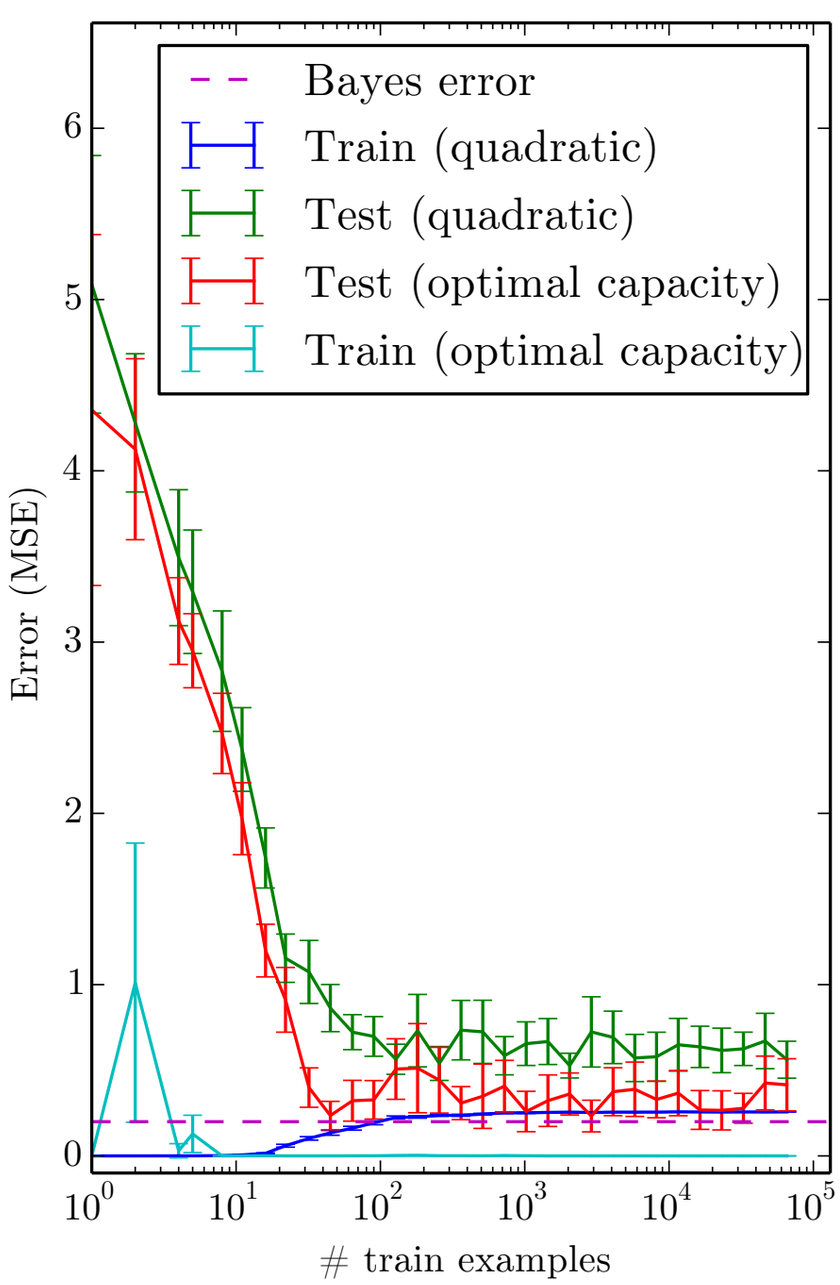
Increasing Depth

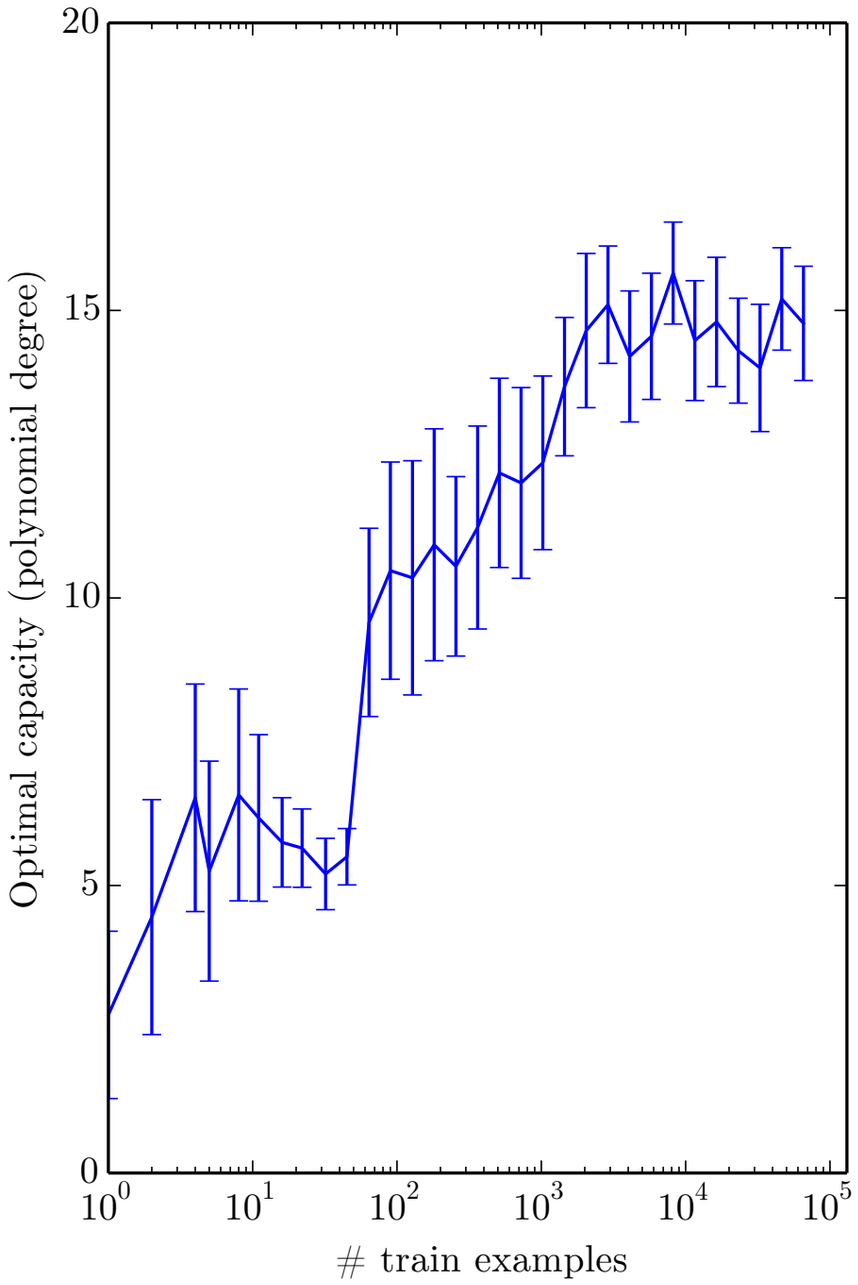


High Test Error

- Add dataset augmentation
- Add dropout
- Collect more data

Increasing Training Set Size





Tuning the Learning Rate

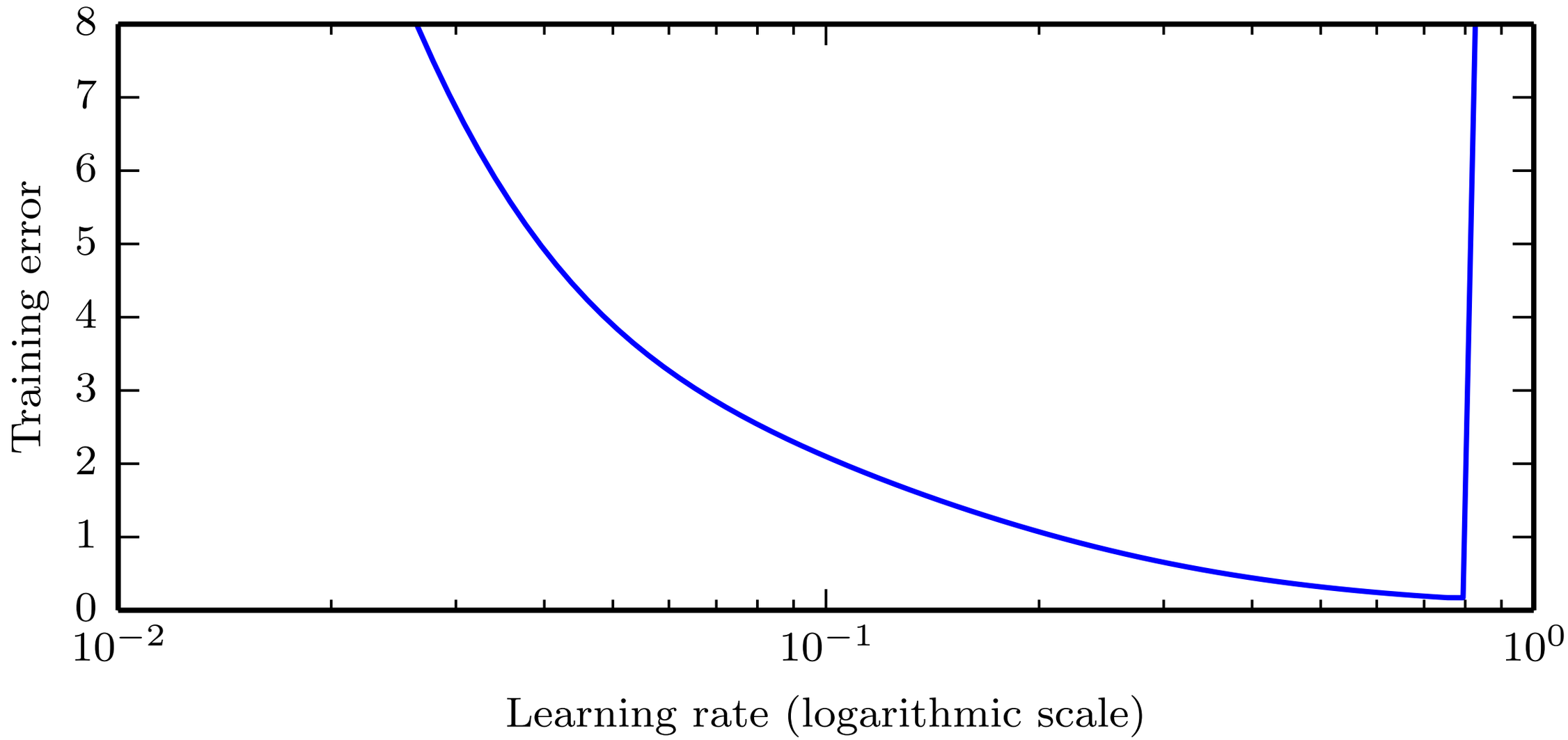


Figure 11.1

Reasoning about Hyperparameters

Hyperparameter	Increases capacity when...	Reason	Caveats
Number of hidden units	increased	Increasing the number of hidden units increases the representational capacity of the model.	Increasing the number of hidden units increases both the time and memory cost of essentially every operation on the model.

Table 11.1

Hyperparameter Search

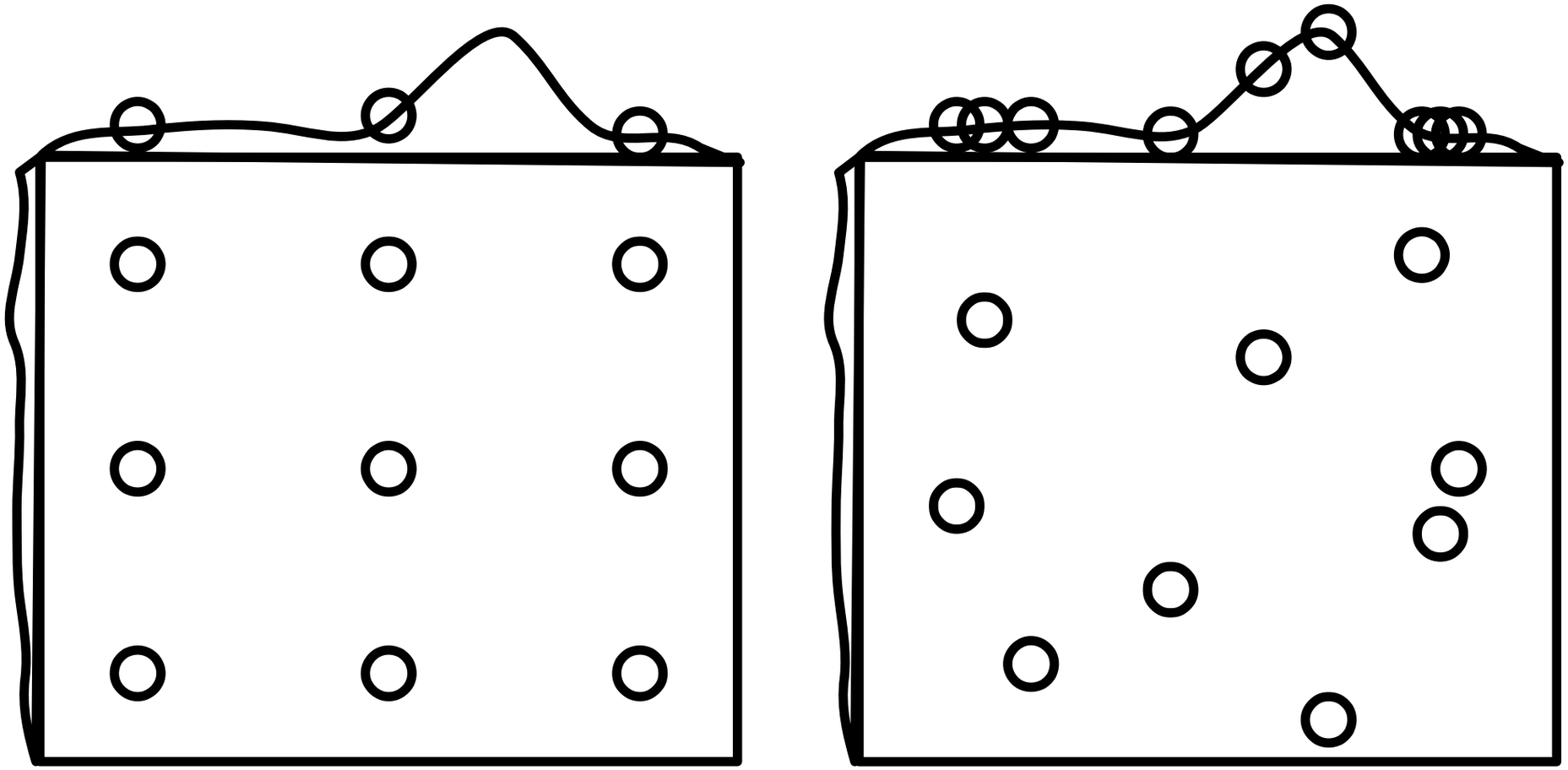


Figure 11.2