

CS194-10 Fall 2011

Introduction to Machine Learning

***Machine Learning: An Overview***

# Course outline

- Overview of machine learning (today)
- Classical supervised learning
  - Linear regression,
  - perceptrons,
  - neural nets,
  - SVMs,
  - decision trees,
  - nearest neighbors
- Learning probabilistic models
  - Probabilistic classifiers (logistic regression, etc.)
  - Unsupervised learning, density estimation, EM
  - Bayes net learning
  - Time series models

# Learning is....

... a computational process for improving performance based on experience

# Learning: Why?

# Learning: Why?

- *The baby, rushed by eyes, ears, nose, skin, and insides at once, feels it all as one great blooming, buzzing confusion ...*
  - [William James, 1890]

# Learning: Why?

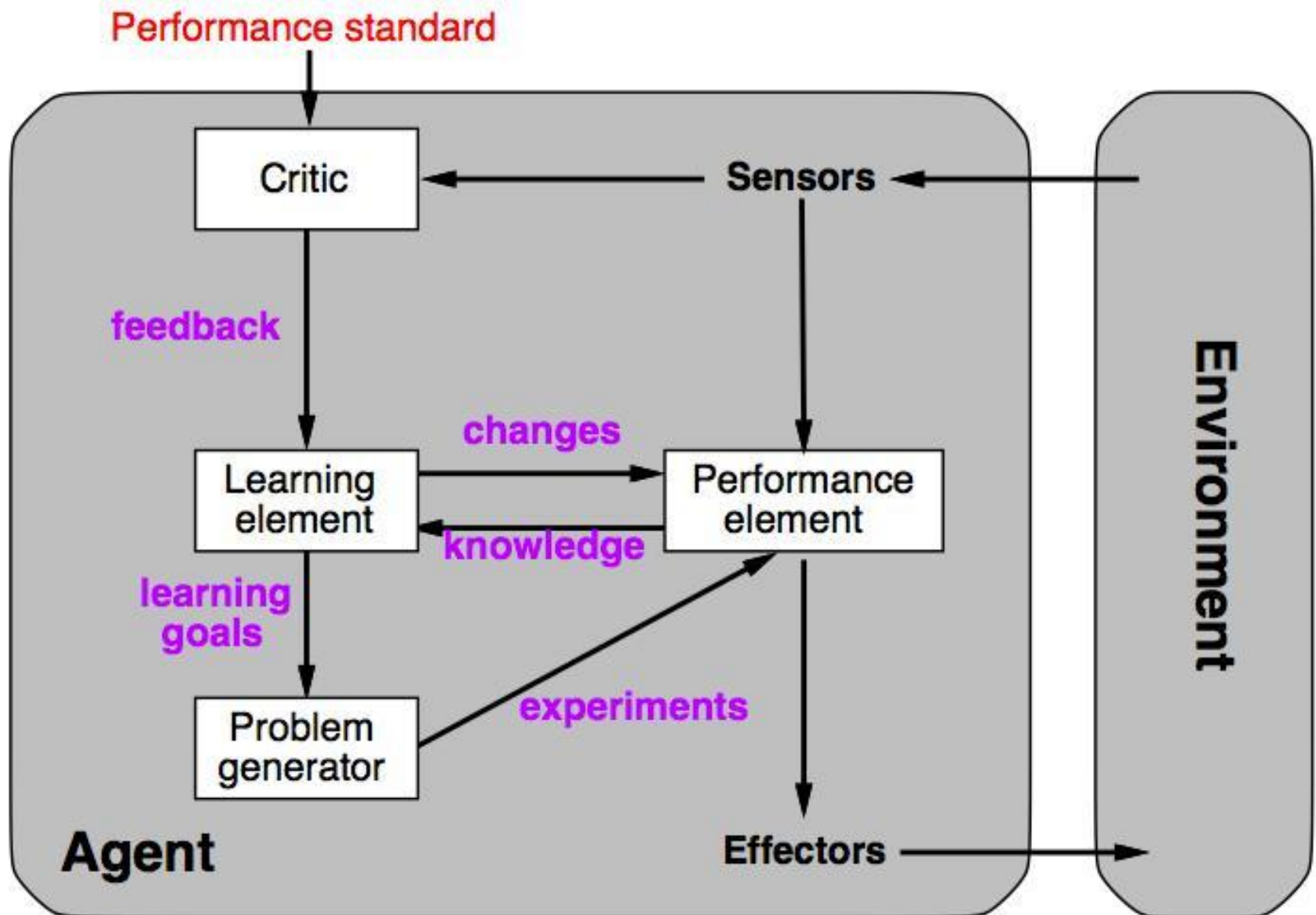
- *The baby, assailed by eyes, ears, nose, skin, and entrails at once, feels it all as one great blooming, buzzing confusion ...*
  - [William James, 1890]

Learning is *essential* for unknown environments, i.e., when the designer lacks knowledge

# Learning: Why?

- *Instead of trying to produce a programme to simulate the adult mind, why not rather try to produce one which simulates the child's? If this were then subjected to an appropriate course of education one would obtain the adult brain. Presumably the child brain is something like a notebook as one buys it from the stationer's. Rather little mechanism, and lots of blank sheets.*
  - [Alan Turing, 1950]
- Learning is *useful* as a system construction method, i.e., expose the system to reality rather than trying to write it down

# Structure of a learning agent





# Design of learning element

- Key questions:
  - What is the **agent design** that will implement the desired performance?
  - Improve the performance of **what piece** of the agent system and **how is that piece represented**?
  - **What data** are available relevant to that piece? (In particular, do we know the right answers?)
  - **What knowledge** is already available?

# Examples

Agent design	Component	Representation	Feedback	Knowledge
Alpha-beta search	Evaluation function	Linear polynomial	Win/loss	Rules of game; Coefficient signs
Logical planning agent	Transition model (observable envt)	Successor-state axioms	Action outcomes	Available actions; Argument types
Utility-based patient monitor	Physiology/sensor model	Dynamic Bayesian network	Observation sequences	Gen physiology; Sensor design
Satellite image pixel classifier	Classifier (policy)	Markov random field	Partial labels	Coastline; Continuity scales
Supervised learning: correct answers for each training instance				

Reinforcement learning: reward sequence, no correct answers

Unsupervised learning: “just make sense of the data”

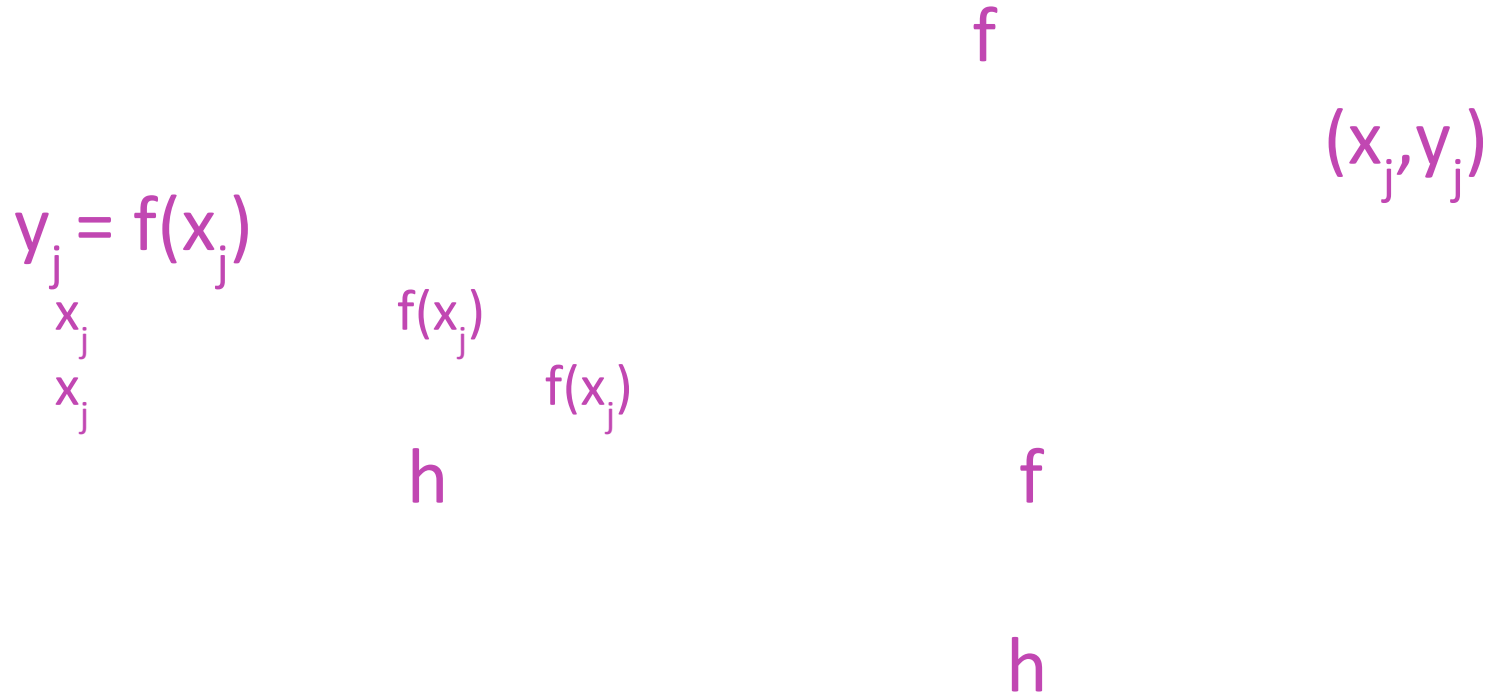
# Supervised learning

- To learn an unknown *target function*  $f$
- Input: a *training set* of *labeled examples*  $(x_j, y_j)$  where  $y_j = f(x_j)$ 
  - E.g.,  $x_j$  is an image,  $f(x_j)$  is the label “giraffe”
  - E.g.,  $x_j$  is a seismic signal,  $f(x_j)$  is the label “explosion”
- Output: *hypothesis*  $h$  that is “close” to  $f$ , i.e., predicts well on unseen examples (“*test set*”)
- Many possible hypothesis families for  $h$ 
  - Linear models, logistic regression, neural networks, decision trees, examples (nearest-neighbor), grammars, kernelized separators, etc etc

# Supervised learning

- To learn an unknown *target function*
- Input: a *training set* of *labeled examples* where
  - E.g.,  $x$  is an image,  $y$  is the label “giraffe”
  - E.g.,  $x$  is a seismic signal,  $y$  is the label “explosion”
- Output: *hypothesis* that is “close” to  $f$ , i.e., predicts well on unseen examples (“*test set*”)
- Many possible hypothesis families for
  - Linear models, logistic regression, neural networks, decision trees, examples (nearest-neighbor), grammars, kernelized separators, etc etc

# Supervised learning



# Example: object recognition

x



f(x)

giraffe

giraffe

giraffe

llama

llama

llama

# Example: object recognition

x



f(x)

giraffe

giraffe

giraffe

llama

llama

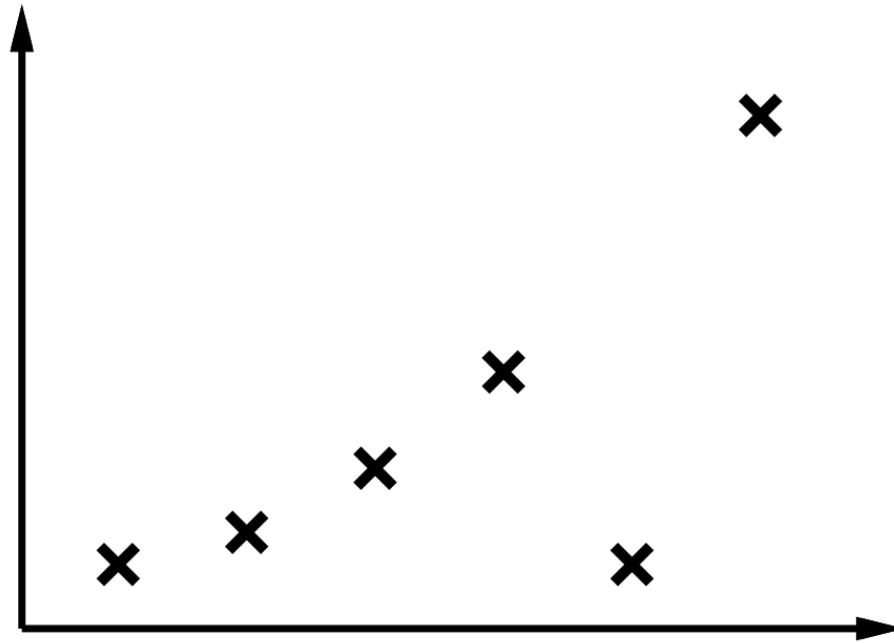
llama

x=



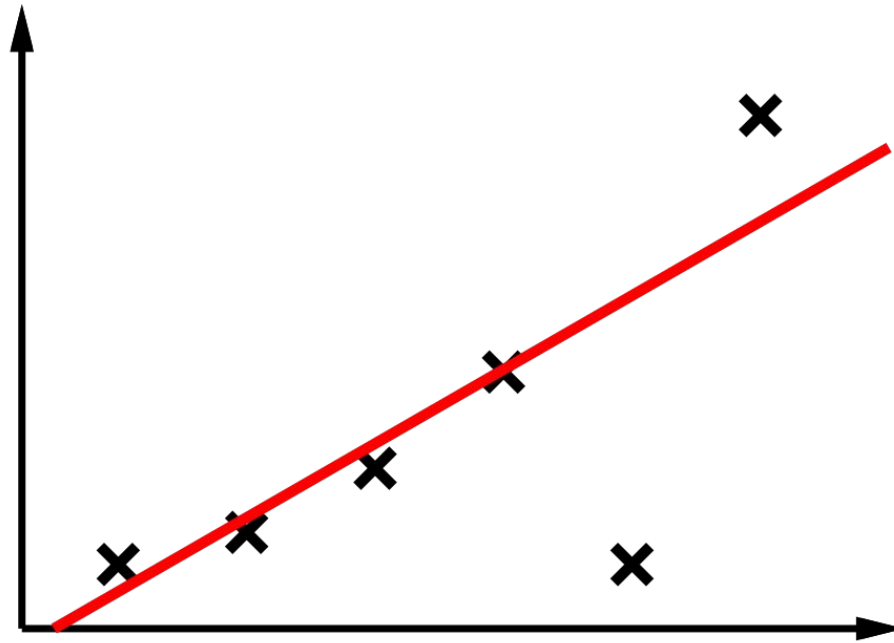
f(x)=?

# Example: curve fitting

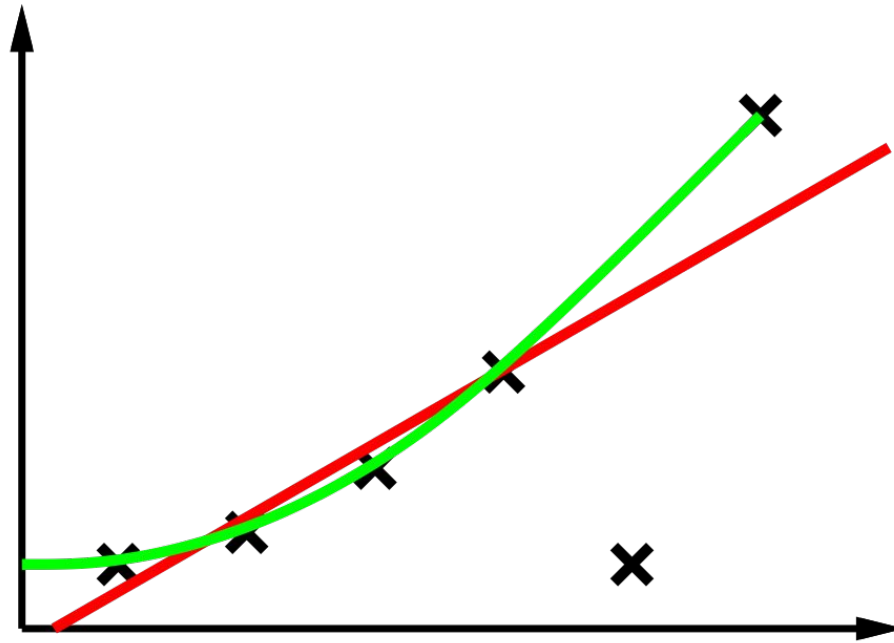




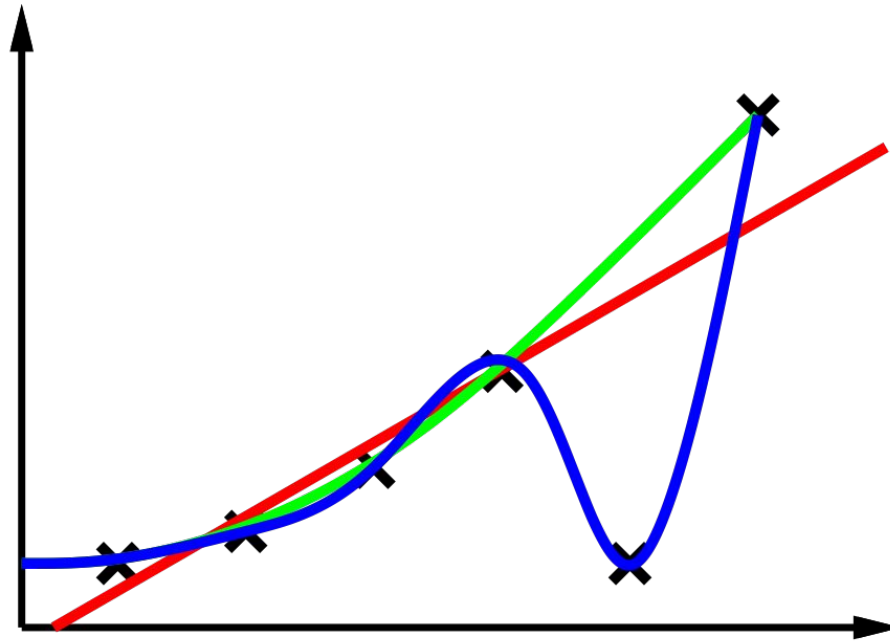
# Example: curve fitting



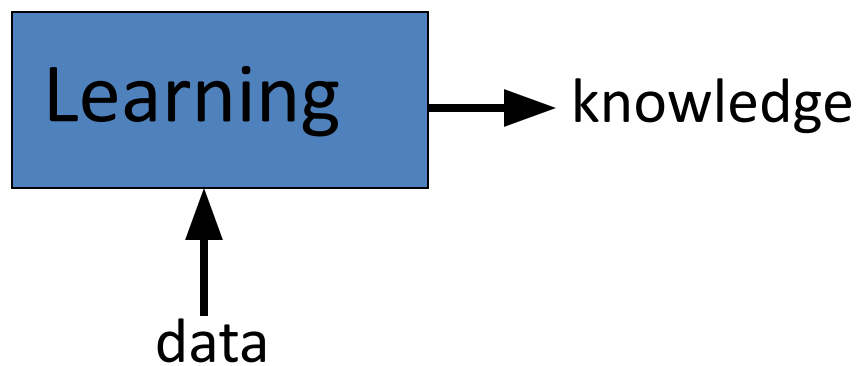
# Example: curve fitting



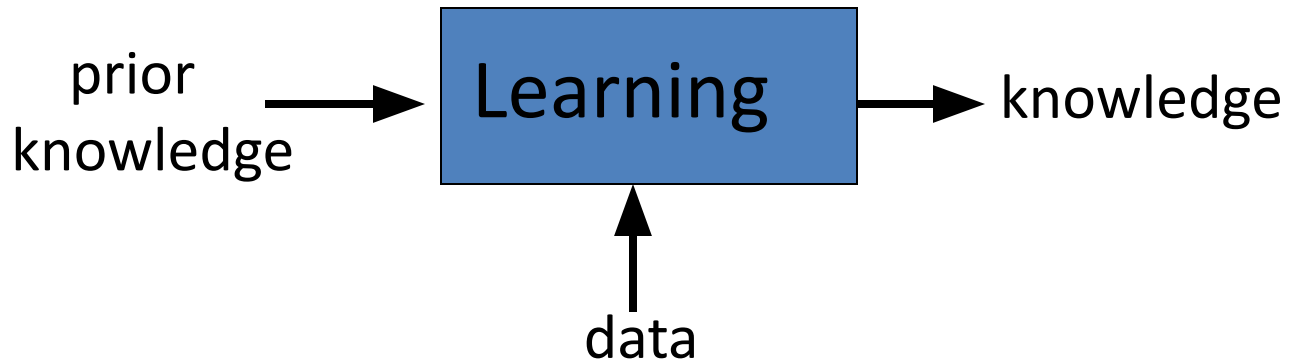
# Example: curve fitting



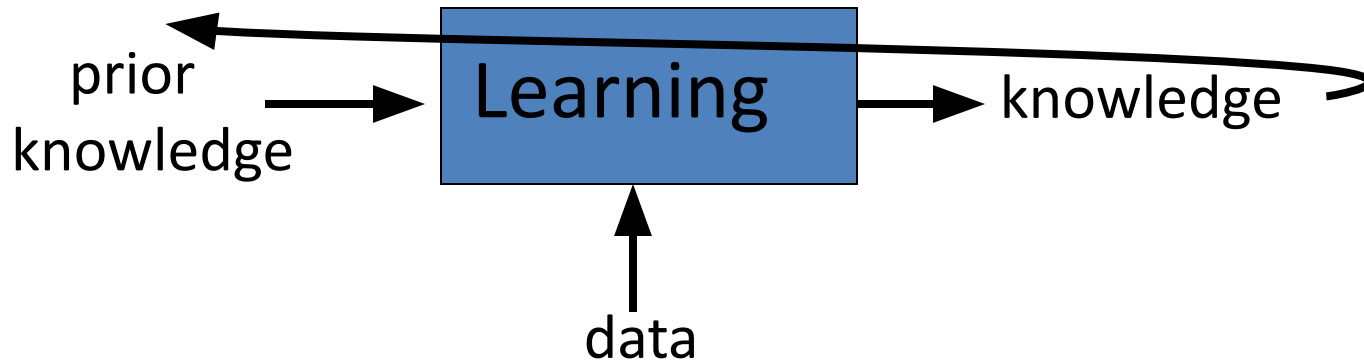
# Learning



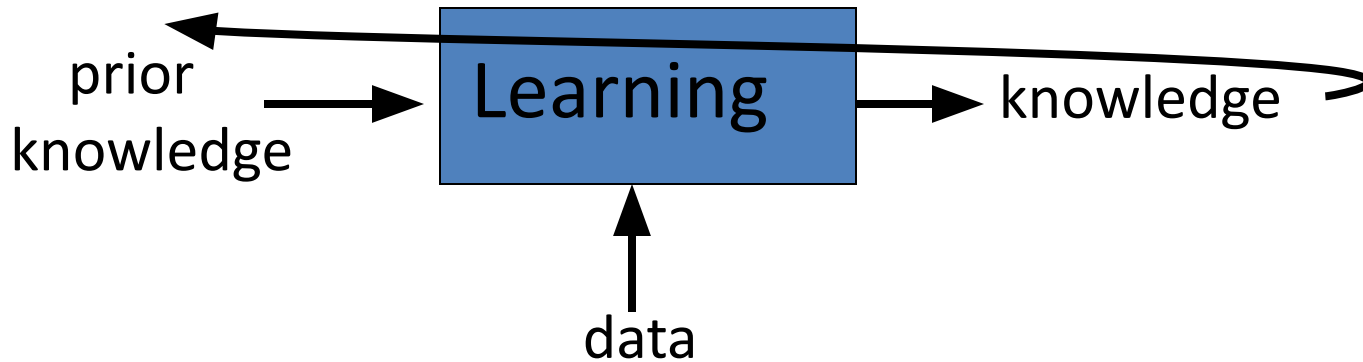
# Learning



# Learning



# Learning



Crucial open problem: weak intermediate forms of knowledge that support future generalizations