

Learning is the ability to improve one's behavior based on experience.

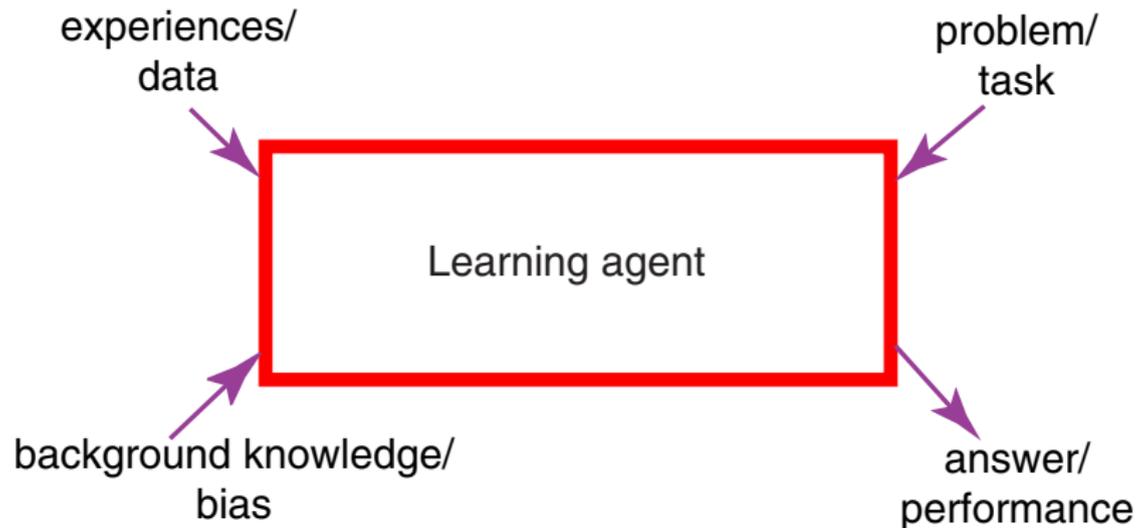
- The range of behaviors is expanded: the agent can do more.
- The accuracy on tasks is improved: the agent can do things better.
- The speed is improved: the agent can do things faster.

# Components of a learning problem

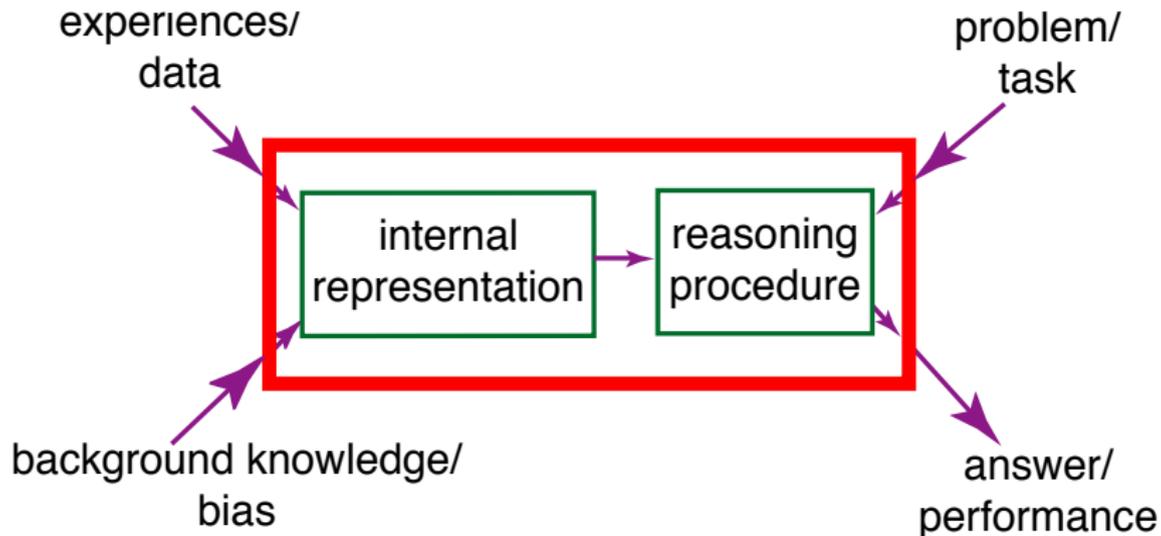
The following components are part of any learning problem:

- **task** The behavior or task that's being improved.  
For example: classification, acting in an environment
- **data** The experiences that are being used to improve performance in the task.
- **measure of improvement** How can the improvement be measured?  
For example: increasing accuracy in prediction, new skills that were not present initially, improved speed.

# Learning task



# Learning architecture



# Choosing a representation

- The richer the representation, the more useful it is for subsequent problem solving.
- The richer the representation, the more difficult it is to learn.

# Common Learning Tasks

- **Supervised classification** Given a set of pre-classified training examples, classify a new instance.
- **Unsupervised learning** Find natural classes for examples.
- **Reinforcement learning** Determine what to do based on rewards and punishments.
- **Analytic learning** Reason faster using experience.
- **Inductive logic programming** Build richer models in terms of logic programs.
- **Statistical relational learning** learning relational representations that also deal with uncertainty.

# Example Classification Data

## Training Examples:

	Action	Author	Thread	Length	Where
e1	skips	known	new	long	home
e2	reads	unknown	new	short	work
e3	skips	unknown	old	long	work
e4	skips	known	old	long	home
e5	reads	known	new	short	home
e6	skips	known	old	long	work

## New Examples:

e7	???	known	new	short	work
e8	???	unknown	new	short	work

We want to classify new examples on feature *Action* based on the examples' *Author*, *Thread*, *Length*, and *Where*.

Learning tasks can be characterized by the feedback given to the learner.

- **Supervised learning** What has to be learned is specified for each example.
- **Unsupervised learning** No classifications are given; the learner has to discover categories and regularities in the data.
- **Reinforcement learning** Feedback occurs after a sequence of actions.

# Measuring Success

- The measure of success is not how well the agent performs on the training examples, but how well the agent performs for new examples.
- Consider two agents:
  - ▶  $P$  claims the negative examples seen are the only negative examples. Every other instance is positive.
  - ▶  $N$  claims the positive examples seen are the only positive examples. Every other instance is negative.
- Both agents correctly classify every training example, but disagree on every other example.

- The tendency to prefer one hypothesis over another is called a **bias**.
- Saying a hypothesis is better than  $N$ 's or  $P$ 's hypothesis isn't something that's obtained from the data.
- To have any inductive process make predictions on unseen data, you need a bias.
- What constitutes a good bias is an empirical question about which biases work best in practice.

# Learning as search

- Given a representation and a bias, the problem of learning can be reduced to one of search.
- Learning is search through the space of possible representations looking for the representation or representations that best fits the data, given the bias.
- These search spaces are typically prohibitively large for systematic search. E.g., use **gradient descent**.
- A learning algorithm is made of a search space, an evaluation function, and a search method.

- Data isn't perfect:
  - ▶ some of the attributes are assigned the wrong value
  - ▶ the attributes given are inadequate to predict the classification
  - ▶ there are examples with missing attributes
- **overfitting** occurs when a distinction appears in the data, but doesn't appear in the unseen examples. This occurs because of random correlations in the training set.

# Characterizations of Learning

- Find the best representation given the data.
- Delineate the class of consistent representations given the data.
- Find a probability distribution of the representations given the data.