Goal Oriented Machine Learning: An Introduction for Middle and High School Students (U.S. Grades 7-12)

Institution
M.I.T.

Lab
Media Lab

Research Groups
Society of Mind
Commonsense Computing
Neuroengineering & Neuromedia
Responsive Environments

Professors
Marvin Minsky
Ed Boyden
Joe Paradiso

Current Students
Bo Morgan

Research Projects
NeuralMoM
Funk2
BrainViz
LifeNet
Roboverse

Introductory Reference
Goal Oriented Machine Learning
Computational Reflective Thinking

Software
Downloads

Goal Oriented Machine Learning

Machine learning is a subfield of the field of Artificial Intelligence, which is a subfield of both Computer Science and Cognitive Science.

Parts of a Machine Learning Computer Program

So what are different parts of computer programs that can be replaced by learning algorithms? Well, since there are many different types of computer programs, let's try to make a list of the different types of parts of each of a sample list of different types of computer programs that currently exist in the world.

Example Application of Learning	Learning Algorithm Used
Music Recommendation System	Eigenvector and other Linear Techniques
Spam Filtering	Probabilistic Bayesian Network Classifiers
4-Wheel Drive Antilock Braking System	Feedforward Backpropagation Neural Networks
Polyworld: Artificial Life Evolution	Genetic (and other Search) Algorithms
Simple Animal Psychology (Reward/Punishment)	Reinforcement Learning Algorithms
Credit Assignment in Plan Failure	Causal Reflection Algorithms
OpenMind Commonsense	Being Told
Emotion Machine One (EM1)	Analogy

What is a Goal?

Every part of a computer program that can be learned automatically has two parts: (1) static, and (2) dynamic. The part of the program that is programmed by the programmer and is *static* is called the learning *algorithm* or *process*, whereas the part of the program that is *dynamic* is the part that adapts to the run-time experience of the program is the *representation*. A goal is a state in your representation space that you would like your computer program to obtain. For example, if there is a part in your computer program that you would like to be handled automatically, then that part can be learned by a learning algorithm. How does a learning algorithm get closer to the goal performance? A learning algorithm has experiences that it learns from. The more experience that a learning algorithm has, the better or closer that algorithm comes to accomplishing the goal. There are many different types of goals and there are many different types of learning algorithms that handle each of these different types of goals! Basically, for every different part of a computer program that a person can write, there is a different type of learning algorithm that can replace that human designer with a learning algorithm that learns from enough experience.

Static (Programmed) Algorithm	Dynamic (Learned) Representation	Goal Representation
Eigenvector and other Linear Techniques	Normalization (Whitening) Matrices	None *(Information Compression)*
Probabilistic Bayesian Network Classifiers	Conditional Probability Distributions	None *(Predictive Accuracy)*

Feedforward Backpropagation Neural Networks	Numerical Weights for Each Synapse	Supervised Training Samples
Genetic (and other Search) Algorithms	Recombinant Data (Strings)	Fitness Function (and Heuristics)
Reinforcement Learning Algorithms	Reward Model and World Model	Active Reward and Punishment
Causal Reflection Algorithms	World Model and Planning Model	Declarative Structures and Failure Types
Being Told	None	None
Analogy	None	None
Experimentation	None	None

When should we use Machine Learning?

Why wouldn't we use a learning algorithm in order to replace all parts of a computer program so that people don't have to design any of the parts? Well, each part of the computer program that is a learning algorithm, must have enough experience to learn from so that it can learn to function as well as the same part of the program that could have been designed by a human programmer. In other words, for every part of a computer program that is a learning algorithm, there must be a lot of experience in order to train that part of the program. Also, sometimes learning from experience is not a good idea because the goal that is to be accomplished is very important or dangerous. For example, if the algorithm is learning to control a nuclear power plant cooling system, a learning algorithm that learns under what conditions the power plant melts-down would not be a good part of the program to be learned from experience. Instead, important or dangerous parts of computer programs should not be learned by a learning algorithm that must learn from experience.

λ