The Design of Scruff: A Framework for AI Based on Probabilistic Programming

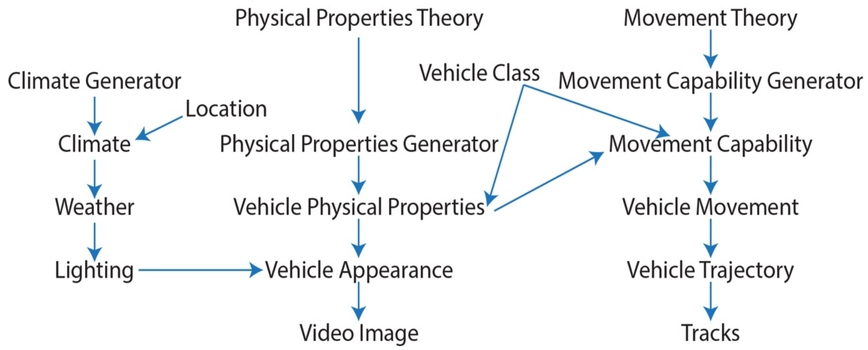
Avi Pfeffer, Jarred Barber, McCoy R. Becker, Joe Campolongo, Sanja Cvijic, Joe Gorman, Michael Harradon, Kenneth Lu
Charles River Analytics
Contact: apfeffer@cra.com

charles river analytics

Scruff is a framework for building AI systems that sense, reason, and learn in their environment using probabilistic programming, founded on the principles of predictive coding.

Alpha release expected in Fall 2020

Hierarchies of Hypotheses



Each layer is generated by the layer above using a probabilistic program
A discovery at one layer might require an explanation at the layer above

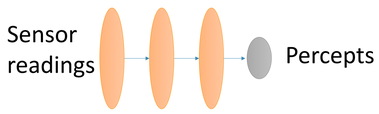
- Observing that a Vehicle Movement is sideways requires a new hypothesis about its Movement Capability
- Encountering vehicles whose Movement Capability includes hopping requires a new Movement Capability Generator that can generate vehicles that hop

Surprise is localized to a particular layer

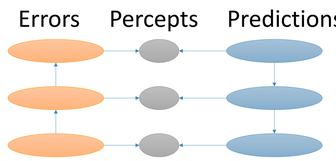
- Sideways motion is a surprise at the Movement Capability level, which is explained by a Movement Capability Generator
- Hopping is a surprise at the Movement Capability Generator level, which is explained by the general Movement Theory

Scruff for Predictive Coding

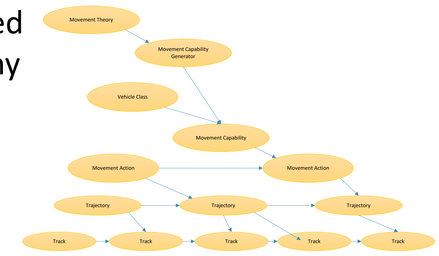
Traditional view: Brain encodes sensory stimuli as they occur; perception flows bottom up from stimuli



Predictive coding (Rao & Ballard, Hohwy, Friston, Clark): Beliefs about the world generate predictions about sensory signals; sensory cortex encodes prediction error; perception results from combination of prediction and error

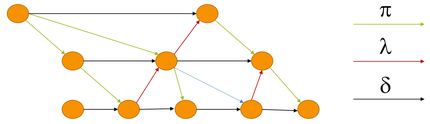


Predictive coding is hierarchical: predictions are propagated down the hierarchy; errors are propagated up the hierarchy



Scruff implements a Bayesian interpretation of predictive coding: Predictions are priors, errors are likelihoods, and percepts are posteriors

Scruff uses asynchronous belief propagation to manage the propagation of priors and likelihoods as the AI system interacts with its environment



Scruff for Building AI Systems

Scruff is a middleware framework for composing the models and algorithms needed to build AI systems

- Stochastic functions (sfuncs) describe conditional probability distributions
  - May be generative, but do not have to be
  - Current sfuncs include standard discrete CPDs, some continuous CPDs, expanders, mixtures, network sfuncs for composition, and expanders that recursively generate network sfuncs depending on an input
- Operators perform computations on sfuncs
  - E.g., generate a sample, compute a π message
- Capabilities describe an sfuncs ability to support an operator, along with metadata such as accuracy and performance characteristics
- Variables are associated with models; instantiating a variable at a particular point in time creates an sfunc derived from its model
- Networks consists of variables and dependencies; the structure of a network can be recursive and change over time
- Algorithms are applied to networks to perform functions such as filtering and online learning by invoking operations on sfuncs for instantiated variables
  - Current algorithms include ABP, particile filtering, lazy structured factored inference for infinitely recursive models