AUGUST 7

Kenji Doya

"Learning algorithms and the brain architecture; Bayesian inference and mental simulation"

(Report by Mike Butler & Farshid Jafarpour)

Report on Kenji Doya's Talk at ISSA

Mike Butler, Farshid Jafarpour

On Friday, August 7, Professor Kenji Doya of the Okinawa Institute of Science and Technology delivered a talk on learning algorithms and brain architecture. The talk began with a basic overview of three basic categories of machine learning algorithms; namely, supervised, unsupervised and reinforcement learning. Professor Doya then demonstrated how these learning algorithms could effectively model certain brain processes. This report will have a similar structure. First we will give a very brief overview of the three types of learning algorithms, then we will discuss one particular insight that came up in our discussion of Prof. Doya's talk. Specifically, we were interested in the possibility that studying the effect of various parameters in Q-learning (a specific kind of reinforcement learning algorithm) might provide us with a model for understanding how the brain's computational function can be affected by certain chemical processes that correlate with different moods or emotional states.

The first type of learning algorithm that Prof. Doya discussed was supervised learning. Suppose you want to teach a machine to play a simple computer game where it has to control a cannon with an adjustable angle. The goal of the game is to hit a fixed target. A supervised learning algorithm learns the relationship between a set of given angles and targets and can predict the required angle to hit a new target not in the given set. In other words, a user provides a set of input output pairs to a computer program where the input is the target and the output is the angle of the cannon required for hitting the target. Then, given a new target, the program predicts the necessary angle and provides it as an output.

Classification problems can also be solved with a supervised learning algorithm. Imagine for instance that you want your phone to classify an image that you take with the phone's camera, say, tell you what sort of dog you are looking at. A supervised learning algorithm could do this provided it has enough stored input output pairs where the input is an image of a dog and the output is the type of dog. Because there is enough data available, the algorithm can learn the relevant features of the image for classifying it. In this way it is supervised because it is given a large set of correct predictions and asked to make a similar prediction. It is as though, with each pair, you are telling the program, "this is a picture of German shepherd, this is a picture of a poodle, etc" then, based on what you have told it, you are asking it to make a similar prediction.

In contrast to this, unsupervised learning algorithms are only given inputs, then asked to find the structure behind the data provided, for instance, to identify clusters of data points on a plane. There are several sorts of unsupervised learning algorithms that can be used for different tasks. Bottom up clustering, for instance, works by assuming each data point is a cluster then grouping together the closest clusters. It then repeats the operation again to increase the size of each cluster. Its final output is thus a map of your data, which begins with exactly as many clusters as there are data points. The next level of clusters would have half as many clusters. This process continues at each level until you have one, all encompassing cluster. This differs from, say, K-means clustering where you set in advance the number K of clusters. The program then clusters the data in K groups in a way that minimizes the sum of square distances of each point from the mean value of the cluster it belongs to. K-means clustering also has a probabilistic version called mixture of Gaussians. In this method, the program learns the distribution of its training set by approximating it as a sum of K Gaussians centered at the mean value of the clusters, then given a new data point, the program can estimate the probability of the given point belonging to each of the K clusters. While there are several other types of unsupervised learning algorithms, the basic idea common to all of them is that they only receive straight inputs, rather than input/output pairs. In this way it is unsupervised because it is simply presented with an object and asked to find a pattern on its own, rather than being told in advance the relationship between a set of objects and actions.

The third form of machine learning algorithm Prof. Doya discussed was reinforcement learning. Unlike supervised and unsupervised learning, reinforcement learning works by providing a series of state/action/reward triplets instead of a series input/output pairs or straight inputs. The goal of the algorithm is to identify an action state relationship that maximizes the total reward. Of the reinforcement learning algorithms which professor Doya discussed, the one we found most interesting was Q-learning. This method works by approximating a Q value, which is the system's expected future reward formulated as a function of the current state and the action the system could take (Q-value). One objective in reinforcement learning tasks is to achieve the desired outcome sooner rather than later. The way Q-learning systems achieve this is by, "discounting" or reducing the value of the future reward the more distant it is from the present action in calculating the Q-value. The strength of this discount is determined by a parameter, gamma. At each state, the algorithm selects an action with the probability proportional to the Boltzmann factor of its Q value at an inverse temperature or exploration parameter beta. In other words, actions are selected probabilistically such that the high Q-value actions are more probable. After each action is taken, and the reward is determined, the reward is used to update the value of Q of the current state/action pair through an equation obtained from the definition of Q at a rate proportional to some constant parameter alpha. Over the course of training, the algorithm learns the value of Q with more accuracy, and as a result, the actions taken by the algorithm correspond to the actions that maximize the total future-discounted reward.

Remember that the parameters alpha, beta, and gamma correspond to learning rate, exploration rate, and future discount rate respectively, and the performance and the behavior of the algorithm is highly dependent on the choice of these parameters. The process of tuning these parameters is called metalearning. In the brain, neuromodulators such as Acetylcholine, Noradrenaline, and Serotonin play analogues roles to these parameters in the way that they affect our behavior. If we conceive of emotions as chemical variations in the brain which affect cognitive function, such as rate of reaction to stimuli, learning rate, level of risk aversion, etc. then it seems as though emotions play the same function as these parameters in Q-learning. This offers a promising model for understanding the relationship between emotion and cognition.