Lecture 6 – Reinforcement Learning I: Action Selection

Alexander Mathis, Ashesh Dhawale

March 24th, 2015, due date March 31st, 2015

1 Background: Reinforcement learning

Connectionist learning models can be subdivided into three categories depending on the type of feedback they use to update their synaptic weights.
Unsupervised learning models such as Hebb's rule do not receive any feedback. Instead they learn to represent statistical features of the training data.
Supervised learning models such as perceptrons or graded neurons learn through training examples, where the distance between the expected and actual output is available to the model.
Reinforcement learning models (RL) receive simple, one dimensional, often binary feedback in the form of a reward when they produce the right output. Because of the simple nature of the feedback being received, RL models learn via trial and error. The goal of RL learning is to maximize the reward received over trials or over time.

1.1 The Rescorla-Wagner model: Learning from reward prediction errors

Consider a model neuron that estimates expected reward (r_{est}) as the linear sum of its inputs or stimuli ($\boldsymbol{u}$) multiplied by synaptic weights ($\boldsymbol{w}$).

$$r_{est} = \boldsymbol{w} \cdot \boldsymbol{u} \tag{1}$$

Here the weights $\boldsymbol{w}$ represent the *reward value* of the corresponding stimuli $\boldsymbol{u}$.
A delta rule can be used to update the weights $\boldsymbol{w}$, in order to minimize the difference between the actual reward (r) and expected reward (r_{est}) by gradient descent.

$$\Delta \boldsymbol{w} = \eta \, \delta \, \boldsymbol{u} \tag{2}$$

$$\text{where } \delta = r - r_{est} \tag{3}$$

where δ is the *reward prediction error*, and η is the learning rate. Equation 2 is similar to the delta-rule for updating weights of graded neurons, albeit in an RL framework.

Despite its simplicity, the Rescorla-Wagner model is able to account for a wide range of phenomena observed in classical (Pavlovian) conditioning experiments (which you can explore further in Assignment 2.1).

1.2 Action selection

Besides estimating the reward value of stimuli, reinforcement learning models are also used in the context of operant or instrumental conditioning, where animals learn to perform specific actions in order to obtain rewards. In these models, the goal of an agent is to maximize the reward received over trials (or time) via optimal selection of rewarded actions.

1.2.1 n-armed bandit task and action values

Let us consider a simple task where an agent has to choose between n possible actions, each of which has a specific probability (called its true *action value*) of yielding a reward. This is known as an *n-armed* bandit task. Typically the true action value is simply the reward yield obtained from performing the action, averaged over trials $\langle r_i \rangle_t$.

The agent has estimates of the true action values (denoted by Q_i). If the true action values are static and equal to the estimated action values ($Q_i = \langle r_i \rangle$), the optimal strategy is trivial - pick the action with the maximum action value. This is called a *greedy* strategy. However, at the start of the simulation, or if reward probabilities are non-stationary, the greedy strategy is bound to fail due to its failure to reliably estimate or track changing action values of non-selected arms.

1.2.2 Softmax action selection

In a changing world, a strategy that balances *exploration* and *exploitation* would be far more successful, on average, than the greedy algorithm. The *softmax* decision rule is one way to control the relative levels of exploration and exploitation.

$$P_i(t) = \frac{exp(Q_i(t)/\tau)}{\sum_{j=1}^{n} exp(Q_j(t)/\tau)} \tag{4}$$

Here, $Q_i(t)$ is the action value of, and $P_i(t)$ is the probability of choosing action i on trial t. This softmax rule is derived from the Boltzmann/Gibb's distribution, and defines a sigmoid (in the case of a two-armed bandit) whose slope is controlled by a positive 'temperature' parameter τ.
τ sets the balance between exploration versus exploitation. A low temperature ($\tau \to 0$) will bias action selection towards a greedy exploitative strategy, whereas a high temperature value ($\tau \to \infty$) would tend towards making every action equiprobable (i.e. pure exploration).

1.2.3 Action value estimation

Similar to strategy we used to update stimulus reward values (equation 2), we can update action value estimates when we choose an action i on trial t.

$$Q_i(t+1) = Q_i(t) + \eta\delta \tag{5}$$

$$\text{where } \delta = r_i(t) - Q_i(t) \tag{6}$$

This delta update rule results in action value estimates eventually converging towards $\langle r_i \rangle$.

2 Assignment

2.1 Rescorla-Wagner rule and classical conditioning

Write a program to simulate the Rescorla-Wagner (R-W) model (equations 1 and 2). Set $\eta = 0.05$.

1. Pair a stimulus with reward ($s_1 \to r$) on every trial, for 100 trials. Set $r = 1$. Plot the evolution of the reward value (w_1) of this stimulus against trial number.
   After the association between stimulus and reward has been learned, present the stimulus alone for 100 more trials ($s_1 \to 0$). This phenomenon is called extinction. Verify that the stimulus reward value (w_1) decreases over extinction trials.
2. Repeat the above classical conditioning procedure, but this time pair the stimulus with reward on just 50% of the first 100 trials ($s_1 \to r$; $s_1 \to 0$). How does the resulting conditioning and extinction curve compare with the case of 100% stimulus-reward co-incidence?
3. Use the R-W rule to model the following classical conditioning paradigms. Check what happens to w_1 and w_2 at the end of Block 2. Let each block be 100 trials long.

Procedure	Block 1	Block 2
Forward Blocking	$s_1 \to r$	$s_1 + s_2 \to r$
Backward Blocking	$s_1 + s_2 \to r$	$s_1 \to r$
Inhibitory Conditioning		$s_1 \to r$; $s_1 + s_2 \to 0$
Secondary Conditioning	$s_1 \to r$	$s_2 \to s_1$

   Compare the outcome of your simulations to the observed experimental outcomes of these conditioning paradigms. Which results match the predictions of the R-W model? Which do not? Why?

2.2 Action selection

Consider the following three-armed bandit problem. It is shopping period and you are a student choosing between the three tutorials on offer. Every day, the tutorials run concurrently, so you can only visit one tutorial at a time and receive (or not) unitary intellectual fulfillment $r \in \{0, 1\}$.
If tutorial i yields rewards with a probability of p_i, let $p_1 = 1/4$, $p_2 = 1/2$, $p_3 = 3/4$. Implement an action selection model using the softmax rule (equation 4), in conjunction with the delta rule to update action values (equation 5). Set learning rate $\eta = 0.01, 0.1, 0.5$ and softmax temperature $\tau = 1, 1/10, 1/100$. Initialize all action values $Q_i = 0$.

1. Under conditions of static reward probabilities p_i, find the η and τ values that maximize the cumulative reward received over a long run.
2. Now consider the case of non-stationary rewards. Let p_i get randomly swapped between tutorials every 250 classes. Over a long run, which combination of η and τ yields in the maximum cumulative reward?