MOD 202 Excercise 7

Ciprian Bangu

May 13 2024

1 Temporal-difference learning with discounting

The temporal difference learning that takes into account future discounting is a modificiation of the standard temporal difference learning rule. Let the value of a state be $V(s_t)$, the estimated value of a state be $\hat{V}(s_t)$, the reward at a state be $r(s_t)$, the discounting factor be γ and the learning rate be ϵ. The value of any state is given by the sum of all future possible discounted rewards from that state. i.e.,

$$\begin{aligned} V(s_t) &= r(s_t) + \gamma r(s_{t+1}) + \gamma^2 r(s_{t+2}) + \dots \\ &= r(s_t) + \gamma V(s_{t+1}) \end{aligned} \tag{1}$$

Recalling the delta learning rule, the value of a state is updated by the difference between the estimated value of the state, and the actual value of the state, i.e.,

$$\delta = V(s_t) - \hat{V}(s_t) \tag{2}$$

However, this presents an issue, since the agent does not know $V(s_t)$, as this is exactly what they are trying to learn. Thus, the agent needs to give an approximate value which they can iteratively improve.

As noted before, the true value of the a state is the reward at that state plus the the discounted value of all future rewards. Thus, the agent can approximate $V(s_t)$ by setting

$$V(s) \approx r(s_t) + \gamma \hat{V}(s_{t+1}) \tag{3}$$

That is, the value of a state is approximately the reward recieved at that state (which the agent is aware of), plus the agent's estimate of the value of the next state (which the agent is also aware of).

Thus, the delta term is given by:

$$\begin{aligned} \delta &= V(s_t) - \hat{V}(s_t) \\ &= r(s_t) + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t) && \text{from Eq. (3)} \end{aligned}$$

The entire learning rule, then, is given by:

$$V(s_t) \to V(s_t) + \epsilon(r(s_t) + \gamma \hat{V}(s_{t+1}) - \hat{V}(s_t)) \tag{4}$$

2 Models for the value function

a) The value function $V(\mathbf{u})$ is given by $V(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$
Since $V(u = (1, 0)) = \alpha$, then:

$$\begin{aligned} V(u = (1,0)) &= \mathbf{w} \cdot (1,0) \\ &= w_1 * 1 + w_2 * 0 \\ &= w_1 = \alpha \end{aligned}$$

Similarily, since $V(u = (0, 1)) = \beta$, then:

$$\begin{aligned} V(u = (0,1)) &= \mathbf{w} \cdot (0,1) \\ &= w_1 * 0 + w_2 * 1 \\ &= w_2 = \beta \end{aligned}$$

Supposing that the agent runs into a state where both stimuli are presented, i.e., $u = (1, 1)$, then the value of that state is given by:

$$\begin{aligned} V(u = (1,1)) &= \mathbf{w} \cdot (1,1) \\ &= w_1 * 1 + w_2 * 1 \\ &= w_1 + w_2 = \alpha + \beta \end{aligned}$$

This type of generalization implies that the value of a state is the sum of the weights of the individual stimuli. That is, the value of a state scales linearly with the number of stimuli present.

A real world scenario where this generalization makes sense is given options to choose from in the supermarket. Taking meat for example, each type of meat can either be present or absent in the supermarket. Moreover, the agent may have certain preferences for some types of meats over others, and thus assign different weights to the different types of meats. The value of the state, i.e., how good the supermarket is in terms of their meat selection, is then the sum of the weights of the individual meats present in the supermarket.

A scenario in which this is not a good generalization is one in which the presence of additional stimuli impact the value of the state in a non-linear way. For example, the agent may assign some intrinsic value to gasoline, and some intrinsic value to a car. But, when both stimuli are present (gasoline and car) the value of the this state will likely be greater that the sum of the intrinsic values the agent assigned, since having both adds a functional (non-stimulus value creation)element to the state that is not present when they are presented in isolation - the agent can drive somewhere.

b) If the value function is given by $V(\mathbf{u}) = \mathbf{w} \cdot \mathbf{u}$, we can derive the following temporal difference learning rule for the parameters $\mathbf{w}$:

First, we know that the value of a state is given by the sum of the reward at that state plus the value of all future rewards, i.e.,

$$V(s_t) = r(s_t) + r(s_{t+1}) + r(s_{t+2}) + \dots \tag{5}$$

$$v(t) = r(t) + r(t+1) + r(t+2) + \dots \quad \text{rewriting in terms of time} \tag{6}$$

$$v(t) = \sum_{\tau} r(t+\tau) \quad \text{summing over all future rewards} \tag{7}$$

But, from above, we know that we can estimate the sum of the value function as a sum of the product of weights and states, so we can rewrite (7) as:

$$v(t) = \sum_{\tau=0}^{T} w(\tau)u(t-\tau) \tag{8}$$

where T is the total number of time steps, $w(\tau)$ is the weight at time τ, and $u(t-\tau)$ is the stimulus history.

Thus, we can apply gradient descent to a similar delta learning rule as before, where updating the weights depends on the difference between the actual value of the state and the estimated value of the state, i.e.,

$$\delta(t) = R(t) - v(t) \tag{9}$$

Where $R(t)$ is the experienced total future reward at time t.

$R(t)$ can the be rewriten as the sum of all future rewards, i.e.,

$$\begin{aligned} R(t) &= r(t) + r(t+1) + r(t+2) + \dots \\ &= r(t) + R(t+1) \end{aligned} \tag{10}$$

And, $R(t+1)$ can be approximated by the value function, i.e., by $v(t+1)$ (from Eq. 8).

Thus, Equation 11 can be rewritten as:

$$\delta(t) \approx r(t) + v(t+1) - v(t) \tag{11}$$

Finally, the gradient of the learning rule is simply the $\frac{\partial V(\mathbf{u}_t)}{\partial \mathbf{w}_t} = u(t-\tau)$ (from Eq. 8).

Thus, the full learning rule is given by:

$$w(\tau) \to w(\tau) + \epsilon(r(t) + v(t+1) - v(t))u(t-\tau) \tag{12}$$

On the other hand, if the value function given by $V(\mathbf{u}) = f(\mathbf{w}\cdot\mathbf{u})$, where $f(\cdot)$ is some non-linear function, the learning rule would be different. Specifically, we would have to apply a gradient descent algorithm to the non-linear function. (Note that, technically, the delta-learning rule is just a special case of gradient descent, i.e., where $\frac{\partial V(\mathbf{u})}{\partial \mathbf{w}} = \mathbf{u}$)

In this case, the gradient for the learning rule would be given by:

$$\begin{aligned}\frac{\partial V(\mathbf{u})}{\partial \mathbf{w}} &= \frac{\partial f(w \cdot u)}{\partial w} \\ &= f'(\mathbf{w} \cdot \mathbf{u})\mathbf{u} && \text{from the chain rule}\end{aligned}$$

Let's take as our known non-linear function the sigmoid function, denoted σ. We know that $\sigma'(x) = \sigma(x)(1 - \sigma(x))$. Thus, the learning rule would be given by:

$$\begin{aligned}w(\tau) &\to w(\tau) + \epsilon\delta(t)\sigma(\mathbf{w}_t \cdot \mathbf{u}_t)(1 - \sigma(\mathbf{w}_t \cdot \mathbf{u}_t))\mathbf{u}_t \\ &\to w(\tau) + \epsilon(r(t) + v(t+1) - v(t))\sigma(\mathbf{w}_t \cdot \mathbf{u}_t)(1 - \sigma(\mathbf{w}_t \cdot \mathbf{u}_t))\mathbf{u}_t \\ &\to w(\tau) + \epsilon(r(t) + v(t+1) - v(t))v(t)(1 - v(t))\mathbf{u}_t && \text{since } v(t) = \sigma(\mathbf{w}_t \cdot \mathbf{u}_t) \\ &\to w(\tau) + \epsilon(r(t) + v(t+1) - v(t))v(t)(1 - v(t))u(t - \tau) && \text{replacing } \mathbf{u}\end{aligned}$$

Thus, we have that the learning rule for a sigmoid value function is:

$$w(\tau) \to w(\tau) + \epsilon(r(t) + v(t+1) - v(t))v(t)(1 - v(t))u(t - \tau) \tag{13}$$

where $v(t) = \sigma(\mathbf{w} \cdot \mathbf{u})$.