Cognitive Neuroscience II

Prof. Dr. Andreas Wendemuth

Lehrstuhl Kognitive Systeme

Institut für Elektronik, Signalverarbeitung und Kommunikationstechnik

Fakultät für Elektrotechnik und Informationstechnik

Otto-von-Guericke Universität Magdeburg

http://iesk.et.uni-magdeburg.de/ko/

Lecture 13

Dynamics of Temporal Difference Learning – an Analytical Calculation

Classical Conditioning

- Classical: Reinforcers delivered independently of actions taken by the animal

- Stimulus u
- Expected reward r, R
- Weight w
- Predicted reward v

Temporal Difference Learning

- Total trial time T
- Predicting Future Reward (only *after* stimulus onset!) $R(t) =< \sum_{\tau=0}^{T-t} r(t+\tau) >$
- Stimuli u over a range of time are weighted: (Sutton and Barto 1990)

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

Rule derivation (Dayan)

- Error function:

$$< R(t) - v(t) >^2 = < \sum_{\tau=0}^{T-t} r(t+\tau) - \sum_{\tau=0}^{t} w(\tau) u(t-\tau) >^2$$

- Stochastic gradient

$$\frac{\partial < R(t) - v(t) >^2}{\partial w(\alpha)} = < \sum_{\tau=0}^{T-t} r(t+\tau) - \sum_{\tau=0}^{t} w(\tau) u(t-\tau) > * u(t-\alpha)$$

- Rule $\Delta \mathbf{w}(\tau) = \varepsilon \delta(t) \mathbf{u}(t-\tau)$; $\delta(t) = < \sum_{\tau=0}^{T-t} r(t+\tau) > - v(t)$

Introducing temporal difference

- Have $\delta(t) = < \sum_{\tau=0}^{T-t} r(t+\tau) > - v(t)$ where

$$< \sum_{\tau=0}^{T-t} r(t+\tau) > = r(t) + < \sum_{\tau=0}^{T-(t+1)} r((t+1)+\tau) > = r(t) + v(t+1)$$

  i.e. prediction is used in formula again.

- Hence prediction error $\delta(t) = r(t) + v(t+1) - v(t)$ where $\Delta v(t) = v(t+1) - v(t)$ is called the *temporal difference term*.

- Allows to predict future rewards.

Full analytical treatment

- We want to compute, for all trials n and for any time of trial t, the predicted reward $v^n(t)$.
- The (single) stimulus u is given at t_u, the (extended) reward r(t) is presented at times t_r,min ... t_r,max.
- I.e. we will end up with a formula describing the following effects of temporal difference learning:

Effects of temporal difference learning

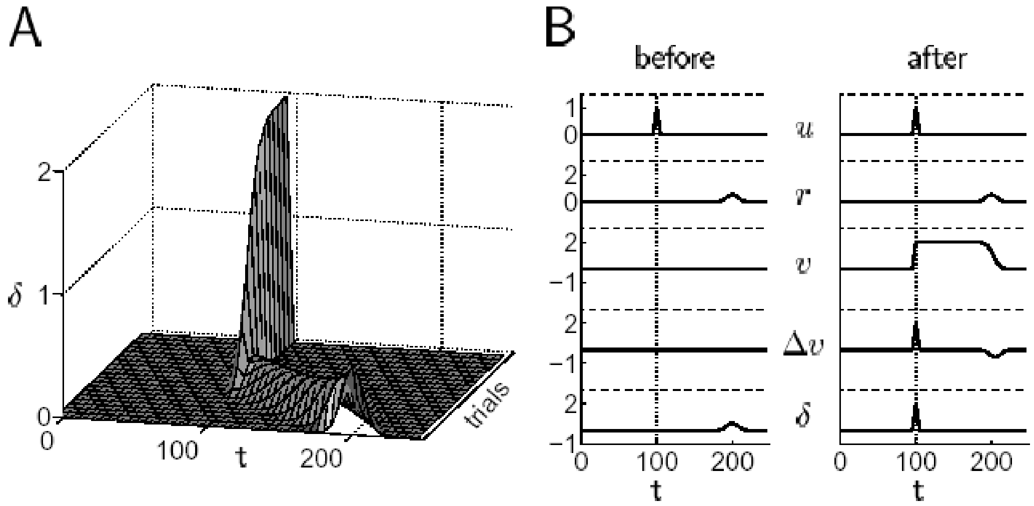

Figure 9.2: Learning to predict a reward. A) The surface plot shows the prediction error $\delta(t)$ as a function of time within a trial, across trials. In the early trials, the peak error occurs at the time of the reward ($t = 200$), while in later trials it occurs at the time of the stimulus ($t = 100$). (B) The rows show the stimulus $u(t)$, the reward $r(t)$, the prediction $v(t)$, the temporal difference between predictions $\Delta v(t-1) = v(t) - v(t-1)$, and the full temporal difference error $\delta(t-1) = r(t-1) + \Delta v(t-1)$. The reward is presented over a short interval, and the prediction v sums the total reward. The left column shows the behavior before training, and the right column after training. $\Delta v(t-1)$ and $\delta(t-1)$ are plotted instead of $\Delta v(t)$ and $\delta(t)$ because the latter quantities cannot be computed until time $t+1$ when $v(t+1)$ is available.

Analytical treatment (1)

- Have for single stimulus $u(t) = u\partial(t, t_u)$:

$$v^{n+1}(t) = \sum_{\tau=0}^{t} w^{n+1}(\tau) u(t-\tau) \overset{t>t_u}{=\!=} u w^{n+1}(t-t_u)$$

and $\Delta w(k) = \varepsilon\delta(t) u(t-k) = \varepsilon u \delta(k+t_u)$. Hence

$$v^{n+1}(t) \overset{t>t_u}{=\!=} u w^{n+1}(t-t_u) = u w^n(t-t_u) + u \Delta w^n(t-t_u) =$$

$$v^n(t) + \varepsilon u^2 \delta^n(t) = v^n(t) + \varepsilon u^2 \left[r(t) + v^n(t+1) - v^n(t) \right]$$

- This is a recursive (in the trials n) relation in $v^n(t)$, for all times of trial t. It shows: $v(t)=0$ for $t<t_u$ and for $t > t_{r,max}$

Analytical (2)

- If this is to converge, we must have $\delta^n(t) \to 0$ and hence $v^{n+1}(t) \to v^n(t)$. This is o.k.

- $\delta^n(t) \to 0$ leads to $v(t+1) = v(t) - r(t)$, as seen in the rule derivation, satisfies $v(t) = \sum_{\tau=0}^{T-t} r(t+\tau)$
- However, that is just the required final state and says nothing about the *dynamics* and whether it actually *converges* to this state. We will therefore analytically derive the full dynamics now.

Analytical (3)

- Write $v^{n+1}(t) \overset{t>t_u}{=\!=} (1-\varepsilon u^2) v^n(t) + \varepsilon u^2 \left[v^n(t+1) + r(t) \right]$ or

$$\begin{pmatrix} v^{n+1}(t_u) \\ v^{n+1}(t_{u+1}) \\ \vdots \\ \\ v^{n+1}(t_{r,\max}) \end{pmatrix} = \begin{pmatrix} 1-\varepsilon u^2 & \varepsilon u^2 & & & \\ & 1-\varepsilon u^2 & \varepsilon u^2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \varepsilon u^2 \\ & & & & 1-\varepsilon u^2 \end{pmatrix} \begin{pmatrix} v^n(t_u) \\ v^n(t_{u+1}) \\ \vdots \\ \\ v^n(t_{r,\max}) \end{pmatrix} + \varepsilon u^2 \begin{pmatrix} 0 \\ \vdots \\ r(t_{r,\min}) \\ \vdots \\ r(t_{r,\max}) \end{pmatrix}$$

- In matrix notation, with component index t:

$\mathbf{v}^{n+1} = \mathbf{A} * \mathbf{v}^n + \varepsilon u^2 \mathbf{b}$ and with $\mathbf{v}^0 = \mathbf{0}$, one has

$$\mathbf{v}^{N+1} = \varepsilon u^2 \sum_{n=0}^{N} \mathbf{A}^n * \mathbf{b}$$

Analytical (4)

- The sum can be calculated (geometric series, **E** is the unit matrix):

$$v^{N+1} = \varepsilon u^2 \sum_{n=0}^{N} A^n * b = \varepsilon u^2 (E - A)^{-1} (E - A^{N+1}) * b$$

- This gives the *full dynamics*! $\mathbf{A}^{N+1}$ has to be calculated, it depends on u and ε.
- I.e. we know $v^N(t)$ for any trial N and for any time of trial t analytically.

Convergence

- If $A^N \xrightarrow{N\to\infty} 0$ (which is the case), the sum converges, and

$$v^N = \varepsilon u^2 (E - A)^{-1} (E - A^{N-1}) * b \xrightarrow{N\to\infty} \varepsilon u^2 (E - A)^{-1} b$$

- Insert **A** and **b**: convergence to $R(t) = \left\langle \sum_{\tau=0}^{T-t} r(t+\tau) \right\rangle$:

$$\begin{pmatrix} v^\infty(t_u) \\ v^\infty(t_{u+1}) \\ \vdots \\ \\ v^\infty(t_{r,max}) \end{pmatrix} = \begin{pmatrix} 1 & -1 & & & \\ & 1 & -1 & & \\ & & \ddots & \ddots & \\ & & & \ddots & -1 \\ & & & & 1 \end{pmatrix}^{-1} \begin{pmatrix} 0 \\ \vdots \\ r(t_{r,min}) \\ \vdots \\ r(t_{r,max}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & \cdots & 1 \\ & 1 & 1 & \cdots & 1 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 1 \\ & & & & 1 \end{pmatrix} \begin{pmatrix} 0 \\ \vdots \\ r(t_{r,min}) \\ \vdots \\ r(t_{r,max}) \end{pmatrix}$$

Analytical (5)

- The result for trial n+1 is, in full detail:

$$\begin{pmatrix} v^{n+1}(t_u) \\ v^{n+1}(t_{u+1}) \\ \vdots \\ \\ v^{n+1}(t_{r,max}) \end{pmatrix} = \begin{pmatrix} 1 & 1 & \cdots & \cdots & 1 \\ & 1 & 1 & \cdots & 1 \\ & & \ddots & \ddots & \vdots \\ & & & \ddots & 1 \\ & & & & 1 \end{pmatrix} *$$

$$\left[\begin{pmatrix} 1 & & & & \\ & 1 & & & \\ & & \ddots & & \\ & & & \ddots & \\ & & & & 1 \end{pmatrix} - \begin{pmatrix} 1-\varepsilon u^2 & \varepsilon u^2 & & & \\ & 1-\varepsilon u^2 & \varepsilon u^2 & & \\ & & \ddots & \ddots & \\ & & & \ddots & \varepsilon u^2 \\ & & & & 1-\varepsilon u^2 \end{pmatrix}^n \right] * \begin{pmatrix} 0 \\ \vdots \\ r(t_{r,min}) \\ \vdots \\ r(t_{r,max}) \end{pmatrix}$$

Conclusion

- We have started with a trio:

  1. Cost function: $v^N(t) \rightarrow (?)\ R(t)$,
  2. Production (Prediction) rule: Sutton and Barto,
  3. Learning (update) rule: temporal difference.

- We have integrated Production and Learning into a recursive formula $\mathbf{v}^{n+1} = \mathbf{A} * \mathbf{v}^{n} + \varepsilon u^{2}\mathbf{b}$
- From this, we have obtained a single (!) closed formula $v^N(t, \varepsilon, u, r(t))$, as compared to the former trio
- We have shown convergence $v^N(t, \varepsilon, u, r(t)) \rightarrow R(t)$