Flexible Shaping

Kai A Krueger and Peter Dayan

Gatsby Computational Neuroscience Unit, UCL

Cognitive flexibility, the ability to acquire, adapt, combine and recombine behaviors appropriate to ever-changing tasks, is a hallmark of intelligent behavior. In mammals, flexibility is likely to depend crucially on mechanisms and representations within the prefrontal cortex (PFC), and on the PFC’s extensive connections with structures such as the basal ganglia (BG) and hippocampus. To study flexibility, it is necessary at the very least to present collections of related tasks; unfortunately, most experimental, and almost all computational, approaches have hitherto focused on learning single underlying tasks, albeit with subtly changing contingencies. In this work, we consider a foundational form of flexibility – the way that separate behavioral components can be acquired through shaping and then combined to solve an overall task.

We study the 12-AX task, which was proposed and then modelled by O’Reilly et al[2] as a rich test bed for analysing PFC-PFC and PFC-BG interactions. Subjects are presented sequentially with letters or the digits ‘1’ or ‘2’. If the most recent digit they have seen was a ’1’, they have to provide a non-default response only to the sequence segment ‘AX’; if it was a ’2’, then they must react only to ‘BY’. Storing ’1’ or ’2’ is an outer working memory loop defining a cognitive context; storing either ‘A’ or ‘B’ is an inner loop. O’Reilly et al modeled a complex reinforcement-based learning process for this task, and showed that it out-performed a standard architecture for learning to use working memory in tasks. However, in both cases, the networks had to learn the full task monolithically, in one fell swoop. Instead, we considered the consequences of shaping, by training individual subcomponents separately, and learning their combination. We performed our shaping in an LSTM[1] network.

We first confirmed that it is substantially easier to learn to combine partial competences than to learn from scratch. However, if we include the time taken to learn the components the overall benefit is a little less evident. Nevertheless, shaping leads to more abstract representations with better generalization. For instance, if training involved restricted numbers of inner loops per outer loop, then the shaped network, with its more abstract inner-loop storage, generalized more proficiently to larger numbers. Shaping also helps learning when there are so many inner loops that the credit assignment path for the outer loop is very extended.

Shaping is only one simple aspect of flexibility. However, it already poses issues such as neural modularization, and also even more evanescent structure in ongoing behaviors, that we are starting to address.

Acknowledgments
Thanks to Randy O’Reilly for discussions and software. Support from the Gatsby Charitable Foundation.

References