



YIN AND YANG OF DECISION MAKING

TWO-TIER HYBRID APPROACH



Mykola Rabchevskiy

Jul 25



There are two fundamentally different decision-making approaches possible in an AGI implementation:

(A) The decision is made based on the current situation (context) by applying an explicitly or implicitly specified rule. If it is possible to have different goals, the rules include goal dependency.

Learning, in this case, is the formation/modification of the corresponding rules.

This category includes, in particular, neural networks, systems based on reward learning, and systems based on rules explicitly formulated by specialists (expert systems).

The essential aspect is that *past experience is not explicitly used in decision making*; the accumulated experience is first *transformed into rules* (explicit or implicit), which are then used to produce a decision.

When the rules are not formed by experts, as a rule, a statistically significant number of relevant cases is required for each combination of potentially possible situations with the current goal/intention/desire. Accordingly, as the complexity of the environment

© 2021 Mykola Rabchevskiy. See [privacy](#), [terms](#) and [information collection notice](#)



Publish on Substack

AGI engineering is on [Substack](#) – the place for independent writing

(forming a set of rules) and their expansion/modification, and the complexity (or even impossibility) of implementing permanent training.

The number of required rules in such systems can be drastically reduced by forming generalized descriptions of situations (ontology, classification of set of possible situations), but this requires a subsystem for identifying subsets of situations that produce the exact result for a specific goal/intention.

(B) Systems based on the analysis of forecasts of the consequences of certain decisions, as described in Architecture. **Predictions are made based on explicitly stored past experience** and the current situation. The choice of a solution comes down to choosing one of the possible, which is most consistent with current goals/intentions/needs/desires.

The *amount of required accumulated information about the past experience does not depend on the number of possible goals/intentions* (up to the continuum), making it possible to add new goals/targets/desires at any time without restarting the learning process.

This approach requires *storing past experience* suitable for forecasting and *relatively large computational resources for making forecasts*. The "compressing" of past experience, based on the identification of recurring situations (DIY pattern mining), as in case (A), can significantly reduce both the amount of stored information and the description of the current situation. Learning in this approach is *naturally permanent* since it boils down to replenishing the accumulated experience.

"Transplantation" of knowledge is reduced to extending the system experience with the experience accumulated by the "sibling" system and is technologically implemented without problems.

The engineering approach to choosing a variant of the decision-making method consists of a suitable combination of the two techniques.

A possible variant of such a hybrid variant uses as the primary approach (B) with the choice of a decision based on explicitly constructed forecasts. The child subsystem of fast decision-making uses a set of explicitly formulated rules that can be easily modified.

If the situation is well known and, therefore, the low-level component of type (A) has a decision rule, the decision is made at the lower "fast" level. If there is no such rule, the decision is made at the top level by a "slow" component of type (B).

The rules in the lower-level subsystem use as input *not a description of the current situation and goals/intentions/desires, but the sequence of previous decisions/actions*. If a

particular following action is regularly performed after such a sequence of steps, it is selected to be completed. The lower-level subsystem tries to "guess" the next action based on the sequence of previous actions. If there is no suitable option or more than one option, control is transferred to the upper-level system.

To form the lower-level "quick response" subsystem rules, the search for patterns in sequences is described in [DIY pattern mining](#) can be used. The most challenging element of this approach is the rules for canceling the rules for fast "reflective" decision-making when performing of which led to an unexpected result. The trade-off between the rules of adding and excluding quick response patterns can significantly affect the system's behavior as a whole.

Such combination realizes what a person calls the development of the “automatism” of performing actions.

The hybrid two-tiered approach *naturally provides the ability for permanent learning*.

This two-tier architecture improves the overall efficiency of the AGI system by delegating control over routine operations to a lower-level subsystem that consumes little time and computational resources for decision making.



Subscribe

← Previous

Next →



Write a comment...

Ready for more?

Subscribe