

Compendium of Neuro-Symbolic Artificial Intelligence

Pascal Hitzler^a, Md Kamruzzaman Sarker^b, and Aaron Eberhart^a

^a*Kansas State University*

^b*Kansas State University*

Contents

| | |
|---|----------|
| 1. Autodidactic and Coachable Neural Architectures | 1 |
| Loizos Michael | |

Chapter 1

Autodidactic and Coachable Neural Architectures

Loizos Michael, Open University of Cyprus & CYENS Center of Excellence, Cyprus

The prediction made by a learned model is rarely the end outcome of interest to a given agent. In most real-life scenarios, a certain policy is applied on the model's prediction and on some relevant context to reach a decision. It is the (possibly temporally distant) effects of this decision that bring value to the agent. Moreover, it is those effects, and not the model's prediction, that need to be evaluated as far as the agent's satisfaction is concerned. The formalization of such scenarios naturally raises certain questions: How should a learned model be integrated with a policy to reach decisions? How should the learned model be trained and evaluated in the presence of such a policy? How is the training affected in terms of the type of access that one has on the policy? How can the policy be represented and updated in a way that is cognitively compatible with a human, so that it offers an explainable layer of reasoning on top of the learned model?

This chapter offers a high-level overview of past work on the integration of modular reasoning with autodidactic learning and with user-driven coaching, as it applies on neural-symbolic architectures that combine sequentially a neural module with an arbitrary symbolically represented (and possibly non-differentiable) policy. In this context, the chapter offers responses to the questions above when the policy can be reasoned with only in a deductive manner, or in a deductive and an abductive manner. It further discusses how the policy can be learned / updated in an elaboration-tolerant and cognitively-light manner through machine coaching, and highlights the connections of the dialectical coaching process with the central role that argumentation plays in human reasoning.

Keywords:

autodidactic learning, introspective forecasting, neural-symbolic compositionality, never-ending rule discovery, machine coaching, simultaneous learning and prediction, explainability, contestability, abduction, argumentation, fourth AI revolution, XIXO principle

1.1. Introduction

A model learned through supervised learning is evaluated in terms of its predictive ability on some held-out set of labeled data. Such an evaluation makes two critical assumptions:

(A1) that the learned model's predictions are the outcomes of interest to some agent; and (A2) that the target predictions are independent of the learned model and the agent, and are available up front. Both assumptions are demonstrably false in certain domains.

Consider, for example, the scenario of a doctor using a model to predict the onset of a certain disease in patients. Even if such a model could be trained to predict accurately, this evaluation metric is far from measuring what is important to the doctor and the patients, who seek to delay or prevent the onset of the disease; and not to simply anticipate it, as they would if assumption (A1) were true. In fact, the agents in this scenario could be seen as wishing, or actively trying, to falsify the predictions of the learned model. Consequently, the doctor might consider using, instead, a model that predicts the survivability of patients given a medical intervention (e.g., the administration of a particular drug). However, these interventions are not, necessarily, independent of the doctor, who might choose them based on past interactions with the model and other personal experiences. Thus, the model cannot be trained up front, as it would if assumption (A2) were true.

Indeed, in several domains where learned models are used, they can be best viewed not as end-to-end solutions, but as modules of a larger, and conceivably much more involved, architecture, within which they interact with (by receiving inputs from and/or producing outputs for) other modules. An agent utilizing such an architecture is, understandably, interested in the global performance of the architecture, and not necessarily in the local performance of any of its constituent modules. In some cases, in fact, the agent itself could be considered as being a module of the architecture, within which it interprets and acts upon the outputs of, and provides inputs to, other modules.

Such larger architectures can resolve the tension between using either neural or symbolic representations, between focusing either on learning or on reasoning, and between operating in an autonomous or in a human-centric manner, with different modules taking a different stance on these presumed dichotomies. One module could support the perception of unstructured data through a massive-scale neural-based learning process, while another module could offer a cognitive layer of human-machine interaction through native symbolic representation and reasoning. Ultimately, the modules could jointly make the concepts in learned models more aligned with human concepts [1, 2], and contribute towards the oncoming Fourth AI Revolution [3] of highly collaborative AI systems that interact as peers with, rather than work for, human users, towards a shared goal.

In this chapter, we shall focus on neural-symbolic architectures comprising a neural module feeding into a symbolic module (which could possibly feed, in turn, into other modules). We shall seek to answer, in particular, how the neural module can be trained and evaluated depending on the access one has on the symbolic module, and how the symbolic module can be trained and evaluated given its reliance on the neural module.

1.2. From Predictive Accuracy to User Satisfaction

Continuing with our medical example, we shall consider an AI system that receives the profile of a patient as input, and produces an outcome of medical relevance. Accordingly, we shall assume the availability of data that pair patient profiles with such outcomes.

1.2.1. Learning is Not the End

One could consider building the AI system by using an end-to-end neural-only architecture. This approach would fail to utilize, however, the considerable and communicable medical knowledge, such as that not all patient profiles are valid, or that certain heuristics (e.g., ranges in medical indices) can already offer an indication of the sought outcome.

An alternative approach would be to build the AI system by using a neural-symbolic architecture, where the knowledge or heuristics can be encoded in the symbolic module, and the role of the neural module would be reduced to perceiving certain salient features in the patient's profile. Setting aside, for now, the question of how the neural module would be trained and/or evaluated (given that the available data do not provide direct supervision signals on what the neural module seeks to predict), it is evident that the neural predictions are no longer the point of focus in this AI system. Accordingly, the local performance of the neural module alone (or the symbolic module alone) does not necessarily align with the global performance of the AI system as a whole.

It is possible, for instance, that the neural module misperceives the fat built-up in the liver of patients based on their ultrasound scans, yet the AI system still yields useful outcomes, either because the outcomes are orthogonal to the misperceived information, or because the symbolic module misuses the misperceived information so that "two wrongs make a right". Neural predictions in a neural-symbolic architecture are a means, and not an end in themselves, and are subject to interpretation by the symbolic module.

We shall refer to the symbolic module content as a "policy", and consider what roles it could play. One role follows from our medical example: *(P1)* A policy can combine, or abstract away from, low-level features that correspond to the neural outputs to higher-level concepts that are not directly observable in the neural inputs, yet they exhibit a sufficiently-strong correlation with the inputs, as statistically encoded in the training data.

A second role is also referenced in our medical example: *(P2)* A policy can constrain, or block from further consideration, neural outputs that represent incoherent configurations. Thus, even in cases where the neural outputs are directly of interest, and even assuming that the neural module is trained on data that provide direct supervision signals, the policy can still be applied *ex post* to improve the neural module's performance.

There is another, perhaps less explored in the neural-symbolic literature, role for a policy: *(P3)* A policy can impute information that is obscured in the neural inputs. Unlike a policy's first role, which can be likened to the imputation of certain unobservable (in the neural inputs) information, here we refer to the imputation of information that is observable in principle, but so happens to be obscured in a particular neural input. In our medical example, the presence of a condition could be obscured in a given ultrasound scan due to fat built-up in the liver, but could be perceived in other ultrasound scans.

Finally, there is a role for a policy in architectures where the symbolic module's output is not the (only) outcome of direct interest: *(P4)* A policy can suggest an action that is warranted based on the neural outputs, and offer it to other modules to predict the effects of that action, which are also outcomes of interest. In our medical example, the policy could suggest a medical intervention, rather than evaluate the patient's health status, and other modules could then make a prognosis for a patient following that intervention.

Policies to *(P1)* abstract, *(P2)* constrain, *(P3)* impute, and *(P4)* suggest can be mixed and combined within an AI system, by having, for instance, the neural outputs first constrained for coherence, then imputed to construct a fuller situational awareness, and then

abstracted away to create a mental model, based on which suggestions for actions can be finally made in an informed manner. Combined or in isolation, the type of the policies used in a neural-symbolic architecture has little effect on how the architecture's modules interact during deployment. In all cases, the neural module receives an input and computes its output (using any process that is applicable for the type of neural network used), the neural output is mapped in a pre-determined fashion into symbolic concepts, and the symbolic module receives those as input and computes its output (possibly feeding that into any subsequent modules in the architecture). However, the policy type greatly affects the training of the modules, as covered in the subsequent sections of this chapter.

1.2.2. Cognitive Explainability

Policies ultimately reflect structure in the AI system's environment or in the mind of the user interacting with the AI system. Imputation and constraining policies, on the one hand, are fit for the former type of structure, in that they capture world knowledge that is independent of any user interacting with the AI system, and use it to create a fuller and coherent awareness of the perceived situation. Abstraction and suggestion policies, on the other hand, are fit for the latter type of structure, in that they capture user-specific knowledge, appealing even to unobservable user-specific concepts, and use it to create a mental model of the perceived situation and decide meaningfully on how to act.

Regardless of the source of knowledge or structure, a policy aims to offer a cognitive explanation layer on top of the neural module, so that the AI system as a whole can be understandable to its user. Especially when the user is a human, cognitive explanations are meant to be expressible in a form aligned with human cognition, going insofar as to use concepts familiar only to the AI system's particular human user. Evidence from Cognitive Psychology suggests that argumentation, which is inherent in human reasoning, can play the role of this lingua franca between AI systems and human users [4, 5].

A primary feature of argumentation is that one may represent competing reasons in support or against a position, with conflicts being resolved during deployment while considering the context. In our medical example, for instance, a constraining policy may include a constraint that the height of an adult patient cannot be less than 1.47m, and a further counter-constraint that the height of an adult patient suffering from dwarfism can be less than 1.47m. If the neural module perceives a short-statured adult, then the neural output will be blocked by the first constraint. However, if the neural module also perceives dwarfism, then the blocking becomes inapplicable, since the argument put forward by the second constraint will defeat the argument put forward by the first one.

A second commonality of policy learning across the two sources of knowledge or structure is the modular nature of the policies. A policy is not a direct mapping from its input to its output. Rather, the policy's constituent parts, which will shall refer to as "rules", may form chains that build on earlier rules. In our medical example, for instance, an abstraction policy may include the rule that the presence of cough and high temperature implies the presence of a respiratory infection, and the rule that the presence of a respiratory infection and breathing difficulty implies the presence of pneumonia. If the neural module perceives the low-level concepts of cough, high temperature, and breathing difficulty, then the policy will abstract to the high-level concept of pneumonia.

All policy rules are subject to being learned, but their learning cannot proceed independently from one another. To guarantee their reliability, each rule should be trained in

a context similar to the one in which it will be deployed. In particular, rules that might be applied on the implications of other rules, and not directly on the neural outputs, need to be trained on the implications of those rules as well. This training necessitates an iterative process of simultaneous learning and prediction [6]. Consequently, and to maintain a computationally efficient training, the lengths of the chains need to be kept small.

If a concept is thought of as a node in a graph, and a rule is thought of as a hyper-edge connecting some premise nodes to a conclusion node, then an argument is a directed acyclic sub-graph with one root node, whose concept is the position being supported by the argument. An argument is triggered in a neural output if all its leaf nodes are perceived, while multiple arguments can be triggered simultaneously. A triggered argument can attack another triggered argument, either directly by supporting a conflicting position, or indirectly by supporting a position incompatible with the conclusion of a rule in the second argument. In our medical example, the argument in support of the presence of pneumonia can be attacked on its intermediate conclusion of the presence of a respiratory infection by the argument that the presence of cough and high temperature implies recent intense exercise, which is a competing diagnosis for what was perceived.

Direct attacks between arguments are symmetric, and in the absence of additional information, neither of the conflicting positions can be accepted, which leads to a dilemma that can be resolved only externally. Alternatively, the policy could identify conditions under which a certain argument has priority over, and is stronger than, another argument, so that in case of a direct attack, the lurking dilemma is resolved in favor of the stronger argument. Such priorities between arguments need to be learned along with the policy.

1.3. Neural Training with Restricted Policy Access

We shall consider, first, the case of training the neural module given restricted access to an arbitrary fixed policy. We shall assume, in particular, that the policy can be accessed only through a deduction method. The deduction method receives an input and yields an output entailed by the policy, based on whatever semantics the symbolic module adopts.

1.3.1. Imputation Policies

In the context of this section, we shall first explore a policy of type (*P3*) scenario, where a neural module perceives certain features in a neural input, and the symbolic module's policy imputes information that, despite not being perceived directly, is likely to be true in the underlying reality that is partially depicted in the neural output. In our medical example, the neural module could perceive the presence of jaundice, but fail to perceive directly any liver disease, and the policy could subsequently impute the presence of hepatitis. The outcome of interest here is a fuller perception of the underlying reality [7, 8, 9], and not its partial appearance that happens to be encoded in the neural output.

Assuming access to data of patient profiles associated with full descriptions of the corresponding underlying realities would be untenable. Accordingly, we shall assume only partial descriptions, and ask how we can usefully train an AI system on these data.

The problem here is that of autodidactic learning [10, 11], where we seek to learn in a supervised manner, but using only supervision signals that happen to be available. Consequently, the neural module has outputs corresponding to all observable features,

even if some are occasionally obscured in the data. To accommodate the fact that, in certain cases, the presence, or value, of some feature might not be predictable, we allow the neural module to abstain by predicting a special “abstention” value for that feature.

It should be noted that the lack of information on a feature during training should not be treated as a supervision signal to reinforce the prediction of the “abstention” value. This point follows from the semantics of the data that we have available. If the data were labeled according to what is present in a patient profile, then we could meaningfully treat the obscureness of a feature as just another ordinary label, and we would expect the neural module to predict this label during deployment. However, this is not the case. The data are labeled with some selection of information on what holds in the underlying reality. The mere lack of information on a feature is not an indication that it is not present in the underlying reality (and hence that it should not be predicted), but rather denotes true ignorance on its value. Therefore, the neural module should not be reinforced to make a certain prediction for a feature that is unlabeled in a given data point.

In our medical example, a neural input representing the facial image of a patient with yellow eyes and an ultrasound scan showing scarred liver tissue could be labeled with an indication of liver cirrhosis in one data point, but not in another. The lack of a label on liver cirrhosis in the second data point should not be used as a training signal. Relatedly, the abstention of the neural module in either data point should be treated uniformly, and it should not be taken to be appropriate for the second data point but not for the first one.

In training the neural module as above, we guarantee that the neural module is allowed to make a prediction even on features that are obscured in a particular neural input if there is supporting evidence from other features, or to abstain if (and only if) this is not the case. This semantics for the neural output accommodates cleanly the use of the policy for imputing a neural abstention [11], since an abstention is meant to capture (as it should) the absence of a prediction, and not the prediction of an absence. Intriguingly, even with the ability to abstain, the learning problem can still become an intractable one in adversarial settings, by obscuring even a minimal amount of information [12].

From our analysis above, it should be clear that the training of the neural module is not affected by the specific imputation policy, but only by the semantics of the process of imputation. More specifically, the trained neural module does not end up encoding any aspect of the specific policy. This dissociation should be viewed only favorably, as it allows the neural module to be trained in isolation, and then paired with any policy.

Nonetheless, one might meaningfully ask whether the policy could be encoded into the neural module through an alternative form of training. The answer is affirmative, by using the policy to impute the data before training the neural module. We shall not elaborate on this any further, as it would give rise to a form of neural-symbolic integration that differs from the sequential architectures that we are considering in this chapter.

1.3.2. Suggestion Policies

In the context of this section again, we shall now explore a policy of type (*P4*) scenario, where a neural module makes a prediction about the future, and the symbolic module’s policy maps that prediction into a suggested action to be taken. In our medical example, the neural module could predict the likelihood of liver cirrhosis, and the policy could then suggest a medical intervention in line with the magnitude of that likelihood. Notably, the outcome of direct interest here is neither the neural module’s prediction, nor the symbolic module’s suggestion. Rather, it is the potential to delay the onset of liver cirrhosis.

Accordingly, we shall assume that we have historical data of patient profiles associated with their eventual development of liver cirrhosis. How can we usefully train an AI system on these data, while coping, in particular, with the seeming paradox of having a symbolic module that suggests actions that would work against the neural predictions?

The problem here is that of introspective forecasting [13, 14], where we seek to make a forecast in a manner that anticipates the ramifications of acting upon it, and remains reliable nonetheless. Before we delve into how this could be done, let us first assume that we managed to train a neural module whose predictions lead a policy to suggest actions that give rise to situations that evolve over time to fulfill those same predictions. Is this satisfactory, or have we simply built an AI system that makes self-fulfilling prophecies?

It should be evident that making self-fulfilling prophecies is not sufficient, since the outcome of direct interest in our medical example, for instance, is not to predict (in a self-fulfilling fashion or otherwise) whether the patient will develop liver cirrhosis, but to delay the onset of the disease. However, having the ability to make self-fulfilling prophecies takes us a step further than simply making predictions that we know might be invalidated simply by acting upon them. One can now compare multiple self-fulfilling prophecies, each associated with a different policy, and decide to act on that policy whose suggested action has the best prognosis. One of the considered policies should be the empty one, suggesting no action to be taken regardless of the prediction, ensuring, thus, the AI system's conformance with the "primum non nocere" principle [15], by contrasting the benefit and harm of any potential intervention against a non-intervention baseline.

How, then, can we train a neural module to make such self-fulfilling prophecies? Training on the available historical data is arguably inappropriate, for a rather fundamental reason. Those data represent an environment in which the given policy was not necessarily applied, whereas the AI system being built will be deployed in an environment in which the given policy will be guaranteed to be applied, as a direct consequence of the very deployment of the AI system itself. What we would need, then, is training data from this second environment, despite our inability to time travel (!) to a future where the AI system is deployed, and gather data to bring to the present and train the AI system.

Fortunately, appropriate training data can be generated through a trial-and-error process (if one accepts the inevitable ethical consequences [15]). For each given neural input, generate a random prediction, apply the policy to get a suggested action, act upon that action, and note whether the environment evolves in a manner that coincides with the random prediction. Keep only those data points (comprising the neural input and the random prediction) for which the random prediction happened to materialize, and train the neural module on those alone using standard supervised learning. It can be formally shown [13] that the AI system with a neural module so trained and a symbolic module that encodes the considered policy, will end up making self-fulfilling predictions.

1.3.3. Unrestricted Access

The restricted access to the two types of policy that were considered in this section was adequate because, in one way or another, the policies were used to draw only forward inferences on the effects, implications, or ramifications of the neural predictions, for which a method of deduction sufficed. One may wonder whether having more information on the policies might have yielded a different analysis. Let us assume, in fact, that we could have full or unrestricted access to the policies. Would that make any difference?

In the case of imputation policies, we have already pointed out that a policy is not used at all when training the neural module, not even through the deduction method. The policy is invoked only to draw inferences when deploying the AI system. Thus, having more access than what is available through the deduction method would be inconsequential. (Admittedly, in attempting to encode the policy into the neural module, as we had briefly entertained, full access to the policy could be useful, as it could allow the neural module to be engineered to encode the policy directly, instead of being trained to do so.)

Things are different in the case of suggestion policies, where a policy is principally involved in the generation of the neural module's training data. Having full access to the policy could presumably lead to a more direct generation process compared to the discussed trial-and-error process. This, however, would not be enough, as the generation process also involves an additional module that computes the effects of the policy's suggested action. Although in special cases this module could learn to encode the effects of actions in a symbolic form [16], in practice, this module would most likely be an opaque predictor of the effects of actions, or it could even be the physical passage of time to perceive those effects. Pragmatically, then, full access to the policy would be immaterial.

In conclusion, access to a deduction method is, in a certain sense, maximally sufficient for training a neural module in the presence of a symbolic module with a policy of type (*P3*) or (*P4*), in that having more access would not yield any obvious improvements.

1.4. Neural Training with Enhanced Policy Access

As in the previous section, we shall consider the case of training the neural module given an arbitrary fixed policy, but we shall, now, assume enhanced access, and, in particular, that the policy can be accessed through both an abduction and a deduction method. The abduction method receives an output and yields the set of inputs on which the policy would entail the output, based on whatever semantics the symbolic module adopts [17].

1.4.1. Abstraction Policies

In the context of this section, we shall first explore a policy of type (*P1*) scenario, where a neural module perceives certain features in a neural input, and the symbolic module's policy abstracts them away into higher-level concepts. In our medical example, the neural module could perceive the presence of high blood pressure and an increased cholesterol level, and the policy could assess the risk of a heart attack. The outcomes of interest here are the unobservable high-level concepts, and not the perceived low-level features.

Accordingly, we shall assume that we have data of patient profiles associated with an assessed risk of a heart attack. How can we usefully train an AI system on these data, considering that they do not provide direct supervision signals for the neural module?

It is instructive to start by thinking of what one would do had the symbolic module been replaced with a second neural module. In doing so, we would end up with a larger neural module consisting of the original neural module feeding into the second neural module. With regard to this larger neural module, then, the available data are in a form that one would expect for carrying out regular supervised learning: they comprise the given neural inputs and the expected neural outputs. Training this larger neural module, and by extension its component that corresponds to the original neural module, could, therefore, be carried out by simply using the standard backpropagation algorithm.

The module substitution process that we have described above is not an entirely fictive one. Depending on their syntax and semantics, certain policies might be translatable into a form amenable to backpropagation. However, this requires first, a restrictive assumption that the policy syntax and semantics are differentiable, and second, knowledge of the actual policy so that it can be translated into the appropriate form. These requirements clash with the context of this section, where the policy is arbitrary, and where the access that we have is only through the use of the abduction and deduction methods.

The problem here is that of neural-symbolic compositionality [18], where we seek to train the neural module in a supervised manner, but using supervision signals coming only indirectly through an arbitrary symbolic module. What is required, then, is a process to compute these supervision signals, and a way to utilize them for neural training.

To understand what type of indirect supervision signals would be useful during neural training, let us consider the AI system’s expected behavior during deployment. Given a data point’s neural input, we would expect the neural module to produce a neural output that, when interpreted as concepts, would be given as input to the symbolic module’s policy, which, in turn, would yield the data point’s label. Reversing the process, we can compute an input to the policy that would lead it to yield a given label, by using the policy’s abduction method. This immediately provides a solution to the first task, whereby we reduce each possible label in the training data into a set of inputs for the symbolic module, and by extension, to a set of outputs for the neural module. Computationally, this reduction process needs to be carried out only once for each distinct label, and its result can be simply applied to each training data point that is associated with that label.

After addressing the first task, we end up with training data that associate each neural input with a set of expected neural outputs, rather than a single one. In our medical example, the neural outputs could be that the patient exhibits either high blood pressure and cramping, or heart palpitations and increased cholesterol level, or nosebleeds and fat built-up in the liver. Each conjunction of concepts corresponds to one neural output.

To utilize the set of labels for training the neural module, we do not use a regular loss function for supervised learning, but instead we use the semantic loss function [18], which acknowledges that any member of the given set would be acceptable as a neural output. By using this loss function with the standard backpropagation algorithm, we can then train the neural module, and address, in this manner, the second task as well.

The neural module is trained, thus, to perceive the low-level features that the policy expects as input to map to the unobservable high-level concepts. While only the abduction method is used for training, the deduction method is still required for deployment.

1.4.2. Constraining Policies

In the context of this section again, we shall now explore a policy of type (*P2*) scenario, where a neural module perceives certain features in a neural input, and the symbolic module’s policy determines whether the perceived features are collectively coherent. In our medical example, the neural module could perceive the presence of low systolic blood pressure and high diastolic blood pressure, and the policy could check to ensure that the former value is larger than the latter value. The outcome of interest here is the neural output, but only conditioned on its conformance to the constraints of the policy.

There are two ways to approach this scenario. The first one would be as a special case of the scenario for abstraction policies. In this case, we would assume that we have

unlabeled data of patient profiles, an extreme case of not having direct supervision signals. Supervision would come only through whatever indirect supervision signals come from the policy, along the lines of what was discussed earlier. In the second approach we shall, instead, assume that each patient profile is associated with a regular supervised learning label, and ask how we can usefully train an AI system on these data given that the policy might be offering additional constraints that need to be taken into account.

What role does the policy play in an otherwise supervised learning setting? As usual, we expect the AI system to generalize over and above the training data. One view of a constraining policy, then, is as a means to reduce the likelihood that certain undesirable generalizations will materialize. The policy can contribute by generating additional supervision signals through abduction, which can be utilized by combining the use of a semantic loss function along with the use of a regular loss function for supervised learning.

In a manner similar to the case of imputation policies, we are not aiming for the policy to be encoded in the neural module. Rather, the constraining policy will be available also during deployment, to block any undesirable neural outputs whose elimination could not be guaranteed simply through training. In case the neural module produces a ranked list of potential neural outputs, the blocking of the top-ranked candidates could lead to the acceptance of lower-ranked candidates. Especially in this case, the deployment-time application of the policy could serve also as a post-hoc filter for neural modules that were trained through regular supervised learning alone, and are oblivious to the policy.

1.4.3. Unenhanced Access

The enhanced access to the two types of policy that were considered in this section sufficed to translate the high-level supervision signals, including the coherence requirement, into low-level supervision signals that were directly usable for training the neural module. Although having access to the abduction and deduction methods only is, already, considerably less demanding than having full access to the policy, one may still wonder whether the former access is also necessary, or whether it can be restricted further.

The abduction method could be replaced by repeated trial-and-error invocations of the deduction method, towards identifying a sample of all the neural outputs that the abduction method would produce. It seems plausible that having access to such a sample only could still provide useful supervision signals, and that a good trade-off could be established between the gain in efficiency and the increase in noise coming from smaller samples. This trial-and-error approach would also allow the sampling of neural outputs from possibly diverse distributions, which offers an interesting space for exploration.

We shall not speculate on other forms of more restricted access, but we shall conjecture that abduction and deduction, or approximations thereof, would seem necessary for neural training in the presence of a symbolic module with a policy of type (*P1*) or (*P2*).

1.5. From Pre-Determined to Learnable Policies

In the two preceding sections, we have focused on the case of training the neural module given an arbitrary fixed policy. We shall, now, turn our attention to how the policy can itself be learned, first considering the base case of an arbitrary fixed neural module, and then considering how this is affected in the presence of a learnable neural module.

1.5.1. Autodidactic Learning

We start by considering policies that capture structure in the environment, and we shall initially focus on imputation policies, and return to constraining policies later on.

The problem here is that of never-ending rule discovery [19], where we seek to learn rules or arguments that are statistically supported by the training data. In the base case of learning a policy given an arbitrary fixed neural module, we have neural outputs that partially depict some underlying reality. Upon receiving a neural output, we randomly create a new hypothesis rule with a premise of small size, such that the rule’s premise and conclusion contain concepts perceived in the neural output. When the premise of a rule is perceived in a subsequent neural output, we check the rule’s conclusion. If it is also perceived, we promote our confidence in the rule, whereas if a conflicting concept is perceived, we demote our confidence in the rule. Whenever, and for as long as, a certain threshold of confidence is exceeded, we characterize a hypothesis rule as being active.

At this point, the interaction of learning and reasoning kicks in. Any subsequently received neural output is first imputed by using the active rules. Specifically, the imputation is done by considering all the arguments that result from the active rules, and resolving the attacks between them by using the order in which the rules became active as capturing the strength of the resulting arguments. Following the imputation step, we continue as before, with the only difference being that when the premises of rules are considered, they are checked on whether they are either perceived or imputed, ensuring that rules are trained in the same context on which they will eventually be deployed.

In terms of rule demotion, this occurs only if a concept in conflict with a rule’s conclusion is perceived but not supported by any argument, capturing in this manner the semantics of the priorities, where stronger arguments protect the learning of weaker ones by explaining away some of the latter’s counter-examples. Intuitively, the learning process ends up first identifying rules that have fewer counter-examples, which typically become part of stronger arguments as they correspond to exceptions of the weaker arguments.

Turning to the general case where the neural module is itself learnable, we recall from an earlier section that the neural module can be trained up front, and in a manner that is oblivious to the particular imputation policy. After the neural training, we proceed to learn the policy as in the base case. Notably, this stratification in learning is in line with the interaction of learning and reasoning that we have advocated [6], since the policy is trained and deployed always in the same context of the outputs of the neural module.

Shifting our focus to constraining policies, one may be tempted to approach them as a special form of imputation policies, comprising special (one-rule) arguments that support either a contradiction (capturing constraints) or a tautology (capturing counter-constraints). Treating the training data as if they are all labeled by a tautology, one may seek to use the learning process for imputation policies also on constraining policies.

The major issue with this treatment relates to the difference on how imputation and constraining policies interact with the neural module. As discussed, imputation policies seek to complete what is already perceived in the neural output. In particular, anything perceived cannot be questioned, and effectively acts as an exogenous argument that attacks and defeats any contradictory policy argument. It is this exact reason that makes it meaningful to train the neural module first and then learn the imputation policy. Constraining policies, on the other hand, seek to filter and block what is already perceived in the neural output. In particular, everything perceived is subject to further scrutiny, and

effectively acts as an exogenous argument that is attacked and defeated by any contradictory policy argument. Consequently, and in line with the interaction of learning and reasoning that we have advocated [6], the constraining policy needs to be learned first, and then deployed to offer supervision signals for the training of the neural module.

We recall that the training data available for an AI system that employs a constraining policy come with regular supervised learning labels for the neural module. These labels for the neural module correspond to inputs of the symbolic module, which can be used to learn the policy. Note that the training data so derived for the symbolic module are not unlabeled, but are rather all labeled as belonging to the tautology class. Thus, the learning setting at hand is one of learning a policy from positive examples only, and this can be done by using any standard learning algorithm for such a setting. Once the policy is learned, the neural module can then be trained in the manner discussed earlier.

1.5.2. User-Driven Coaching

We continue by considering policies that capture structure in the mind of the user, and we shall focus on abstraction policies, with an analogous handling for suggestion policies.

The problem here is that of machine coaching [20], where we seek to learn rules or arguments by means of a dialectical interaction with the user. In the base case of learning a policy given an arbitrary fixed neural module, we have neural outputs that perceive features in the neural inputs. Upon receiving a neural output, the current policy is applied to compute some high-level concept. The user inspects the arguments that support the policy output, and may object by providing additional arguments that either complete the ones in the policy (e.g., in case certain conclusions that were expected by the user were not drawn by the policy), or attack and defeat them (e.g., in case certain conclusions that were not expected by the user were drawn by the policy). The policy is revised through the addition of the provided stronger arguments, and a new round commences.

The interaction of learning and reasoning is built into the bilateral explainability between the user and the AI system, in line with the XIXO principle that only by providing good eXplanations In the AI system, the user may expect to get good eXplanations Out [3, 21]. The revision of the policy happens only as a reaction to, and in the context of, the currently triggered arguments. Since the user-provided arguments dispute the existing policy arguments, the attacks between conflicting arguments are resolved in favor of those that were provided more recently, with their addition in the policy effectively making the policy a ranked list of arguments with ever-increasing strengths [22].

Notably, the priorities of arguments are reversed compared to those in imputation policies, whose induction-based learning semantics make it meaningful for arguments learned earlier to be stronger, while the interaction-based semantics for acquiring abstraction policies make it meaningful for arguments acquired earlier to be weaker.

The reversal holds also with regard to the strengths of exogenous arguments, which can be thought of as being the first arguments learned or acquired. Exogenous arguments in imputation policies come from the neural outputs, and are, accordingly, considered stronger than all policy arguments. Exogenous arguments in abstraction policies come from labels that the user may provide (possibly in an online fashion) for the training data. While the user may initially evaluate a policy output against those labels, the arguments provided by the policy in support of its output might convince the user to revise that initial evaluation. The exogenous arguments in abstraction policies are, accordingly, con-

sidered weaker than all policy arguments. The coaching process is fully aligned with this perspective, since the user never objects directly to a policy's output (which would correspond to invoking a weak exogenous argument), but rather to the policy's arguments in support of that output (which corresponds to invoking a stronger endogenous argument).

Turning to the general case where the neural module is itself learnable, we recall from earlier that the neural module should be trained by using the policy to get indirect supervision signals. This, however, leads to a circular dependence, since the policy itself is acquired by using the neural outputs. A natural solution is to iterate between coaching rounds, to improve the policy given the current neural module, and training rounds, to adapt the neural module to the current policy [1]. Some preparation of the neural module and/or the policy through transfer learning might be necessary to bootstrap the process.

1.6. Conclusion

The oncoming Fourth AI Revolution [3] promises AI systems that will balance their autonomy against the requirement to collaborate closely with humans. Their design necessitates, therefore, architectures that can accommodate and integrate their uncanny ability to sift through massive amounts of data and identify patterns, with the expectation to operate in a manner compatible with human cognition. This cognitive compatibility encompasses the ability to engage dialectically with humans, to exchange arguments in support of positions, to adapt their operation in a developmental manner, and, ultimately and consequently, to align their conceptual models with those of humans [2]. Computational argumentation has a key role to play as a formal foundation for such human-centric AI systems [23]. The desire for explainability and contestability [24] by design, not in terms of what position an AI system should support functionally, but in terms of how to support it structurally, according to legal, ethical, and social considerations [1, 25], follows then naturally. This chapter presented some guidelines, grounded on theoretical and empirical work, on how to move towards these goals over a neural-symbolic substrate.

Acknowledgements

This work was supported by funding from the EU's Horizon 2020 Research and Innovation Programme under grant agreements no. 739578 and no. 823783, and from the Government of the Republic of Cyprus through the Deputy Ministry of Research, Innovation, and Digital Policy.

References

- [1] Michael L. Coachable AI. In: ERCIM News No. 132, Special Theme: Cognitive AI & Cobots; 2023.
- [2] Michael L. Concept Alignment through Machine Coaching. In: Proceedings of the Cognitive Artificial Intelligence Meeting of The Royal Society; London, U.K.; 2022.
- [3] Michael L. Explainability and the Fourth AI Revolution. In: Carayannis EG, Grigoroudis E, editors. Handbook of Research on Artificial Intelligence, Innovation and Entrepreneurship. Edward Elgar Publishing; 2023.
- [4] Kakas A, Michael L. Cognitive Systems: Argument and Cognition. IEEE Intelligent Informatics Bulletin. 2016;17(1):14–20.

- [5] Dietz E, Kakas A, Michael L. Argumentation: A Calculus for Human-Centric AI. *Frontiers in Artificial Intelligence*. 2022;5.
- [6] Michael L. Simultaneous Learning and Prediction. In: *Proceedings of the 14th International Conference on Principles of Knowledge Representation and Reasoning (KR'14)*; Vienna, Austria; 2014.
- [7] Michael L, Valiant L. A First Experimental Demonstration of Massive Knowledge Infusion. In: *Proceedings of the 11th International Conference on Principles of Knowledge Representation and Reasoning (KR'08)*; Sydney, Australia; 2008.
- [8] Michael L. Reading Between the Lines. In: *Proceedings of the 21st International Joint Conference on Artificial Intelligence (IJCAI'09)*; Pasadena, California, U.S.A.; 2009.
- [9] Michael L. Machines with WebSense. In: *Proceedings of the 11th International Symposium on Logical Formalizations of Commonsense Reasoning (Commonsense'13)*; Ayia Napa, Cyprus; 2013.
- [10] Michael L. Autodidactic Learning and Reasoning [dissertation]. Cambridge, Massachusetts, U.S.A.: School of Engineering and Applied Sciences, Harvard University; 2008.
- [11] Michael L. Partial Observability and Learnability. *Artificial Intelligence (AIJ)*. 2010;174(11):639–669.
- [12] Michael L. Missing Information Impediments to Learnability. In: *Proceedings of the 24th Annual Conference on Learning Theory (COLT'11)*; Budapest, Hungary; 2011.
- [13] Michael L. Introspective Forecasting. In: *Proceedings of the 24th International Joint Conference on Artificial Intelligence (IJCAI'15)*; Buenos Aires, Argentina; 2015.
- [14] Michael L. The Disembodied Predictor Stance. *Pattern Recognition Letters (PRL)*. 2015;64(C):21–29.
- [15] Michael L. “Primum Non Nocere” for Personalized Education. In: *Proceedings of the NIPS 2012 Workshop on Personalizing Education with Machine Learning (PEwML'12)*; Lake Tahoe, California, U.S.A.; 2012.
- [16] Michael L. Causal Learnability. In: *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI'11)*; Barcelona, Catalonia, Spain; 2011.
- [17] Kakas A, Michael L. Abduction and Argumentation for Explainable Machine Learning: A Position Survey; 2020. arXiv:2010.12896.
- [18] Tsamoura E, Hospedales T, Michael L. Neural-Symbolic Integration: A Compositional Perspective. In: *Proceedings of the 35th AAAI Conference on Artificial Intelligence (AAAI'21)*; virtual; 2021.
- [19] Michael L. Cognitive Reasoning and Learning Mechanisms. In: *Proceedings of the BICA 2016 International Workshop on Artificial Intelligence and Cognition (AIC'16)*; New York City, New York, U.S.A.; 2016.
- [20] Michael L. Machine Coaching. In: *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence (XAI'19)*; S.A.R. Macau, P.R. China; 2019.
- [21] Markos VT, Thoma M, Michael L. Machine Coaching with Proxy Coaches. In: *Proceedings of the COMMA 2022 International Workshop on Argumentation and Machine Learning (ArgML'22)*; Cardiff, Wales, U.K.; 2022.
- [22] Markos VT, Michael L. Prudens: An Argumentation-Based Language for Cognitive Assistants. In: *Proceedings of the 6th International Joint Conference on Rules and Reasoning (RuleML+RR'22)*; virtual / Berlin, Germany; 2022.
- [23] Dietz E, Kakas A, Michael L, editors. Research Topic on Computational Argumentation: A Foundation for Human-Centric AI. *Frontiers in Artificial Intelligence*; 2023.
- [24] Tubella AA, Theodorou A, Dignum V, et al. Contestable Black Boxes. In: *Proceedings of the 4th International Joint Conference on Rules and Reasoning (RuleML+RR'20)*; Oslo, Norway; 2020.
- [25] Michael L. Machine Ethics through Machine Coaching. In: *Proceedings of the 2nd Workshop on Implementing Machine Ethics (IME'20)*; virtual; 2020.