Deep learning framework for Artificial General Intelligence

Shimon Komarovsky and Jack Haddad*

***Abstract*—**

I. INTRODUCTION

ML as the technical implementation of AI is not fully exploited as it could and should be. "Learning" in ML is actually an identification process based on sophistical memorizing, without involving any thinking or imagination. Although comparing to a lot of control methods, this modeling is highly dynamic as it changes by each input, but in the big picture, when looking at the full potential of intelligence, this is a passive model, since it has a fixed structure (such as DNN or any other network) and there're no higher brain features/capabilities involved in it, as thinking, imagination, etc.

We suggest in this paper, a general methodology how to tackle this issue, in order to develop further the AI field, to a more meaningful phase in the history.

Most guidelines are situated in a continuum between mandatory to suggestion, but it is hard to define which is which.

Not only that "Learning" as defined in ML is simply an identification process based on sophistical but limited memorizing, without involving any thinking or imagination. But reinforcement learning is also type of a low/narrow intelligence, which exists also in animals, such as in Pablo's dog experiment, where learning were taught by positive (reward) and negative (punishment) feedbacks.

ML is still a part of AI for a good reason - it replaces the need for the user to know and be familiar with the problem extensively [1], and gives a general method that can find good solutions from example training. Also, this is kind of a higher type of programming - where no explicit coding is done.

However one must remember that studies showed that for a successful ML we need to know the system, also ML is a very task-specific strategy. Also ML is a static structure/model imitation of the brain, while there is the effect of dynamical processes, such as construction/destruction of neurons. Moreover, it is not fair to compare performances of ML to a regular control-based method designed by a human [1] or to the use of statistical methods [2], after studying the system. The reason is that ML know the problem only from a limited set of examples, that frequently do not represent all situations, but it also lack the comprehensive vision as a human has. Hence, this is another reason for the necessity with conclusion-generating AI.

Also the curse of dimensionality (COD) says that as the input vector of information is larger, then we require exponentially more examples to keep the same performance level. That's whyBut downsizing dimensions methods because of a small number of available examples result also in poorer performance.

But this is because of the wrong attitude towards AI. If we'd relax the performance or replace it totally with other criteria to judge AI, then we'll exchange the desire to fit some NN model to the data (by enormous amount of examples) and thus fixing this model to represent this specific data and nothing else, with enabling freedom for the AI to generalize and process (by less amount of examples).

....(taken from proposal) There are such NNs which use back-propagation for their weight adaptation, e.g. for traffic congestion prediction [3], or to replace traffic simulator [4] by training on simulation parameters (as input) and their outcomes (as outputs). [4] also demonstrates that NN is best for performance if its structure is appropriate to the data it is trained upon. Not too much neurons nor too little.

Remark: This optimal number of neurons and layers might be because there are the most appropriate features to describe the trained data. Any other structure may result in some kind of a spreading over some features that are not really representative, of the data.

... Many DNN's are developed for a very specific tasks in relation to visual information. Such features for example as a change in light, different angles, recognition of boundaries of objects in an image. And all that only on static photos, before we consider to add movement and video type of change in time. The amount of work and theory developed around computer vision area is huge, but I don't consider this as an effective method to use real intelligence. Since we doing it all again as we did with simple model algorithms: programming very big code to handle for example a task of computer recognizing space or a robotic arm movement task, to take into account all possible scenarios.

Or the fact that most of the thinking and planning occurs in the human part taking, and in the machine end all is left is an execution.

Shimon Komarovsky and Jack Haddad are with the Technion Sustainable Mobility and Robust Transportation (T-SMART) Laboratory, Technion–Israel Institute of Technology. * Corresponding author: jh@technion.ac.il

But this is not a progress. We should learn the basic real intelligent learning as it occurs in human, without these complex programming, the way we distinguish and recognize objects in a changing environment. And we have to build an AI such that it isn't expected to be stable or perfect right away, immediately at first execution. This is a wrong attitude towards actual intelligence.

On the contrary, we have to give it the time to develop, just as an infant baby and a child do, and not demand from it to give always "correct" answers, but instead - as a human does - allow it to make a mistakes based on partial understanding and learn from it not as a new input in DNN, but as another step in a more general and mature step of development. As a more general point of view on the data he encountered during its lifetime.

In other words, as the comparison between serial thinking verse parallel one [5] show, that in serial one we prefer judgment and quick selection or fast results with emphasis on certainty, in parallel however it is fluent and non-judgmental in favor of fluency and multiple solutions and options together with their probabilities, i.e. allowing and welcoming uncertainty instead of fighting it. The same is here, model-based methods and control are mostly designed for fast good results, to prove effectiveness. But human intelligence shows, that it takes years for an infant to gather linguistic capabilities and fine motor sensing. It takes a many months, in which the baby is mumbles or pronounce poorly and have a gross motor skills (i.e. in movement and drawing). This observation support the idea that the more AGI is general, to deal with as variant knowledge as possible, the more efforts it take to adapt and learn this knowledge. And the opposite is true also - the more the AGI is specific, like current control methods, than the more it fits to be effective in special data and faster in results.

This is actually the difference between efficiency and effectiveness [6], where efficiency concentrates on the best exploitation of available resources, while effectiveness is about performance measure of how well the goal is achieved. Consequently, the AI must be efficient more than effective, since we're less interested in some specific desired outcomes, but rather a good thinking machine which can be validated only on the long run. Same idea expressed within the comparison between serial and parallel thinking [7], where using parallel thinking means a more general, comprehensive and planning, including the freedom and the space to explore, taking intelligent risks and long trial-and-error. In contrast, we have demand for perfection and immediate success in serial thinking. In other words, a broader view to see how one can be more effective in the bigger picture rather than in some specific and narrow tasks.

We shouldn't forget that we mimic the next best thing. Just like we imitated many processes in nature such as GA, flying, and many more - what a man engineered frequently had a poorer quality. Because usually these versions were only a simplified copies of the real thing. So how can we expect so much more from an imitation of human brain activity? With so little knowledge about it yet, and yet expect it to perform better than the real thing? Better than human.

Artificial intelligence (AI) is a very wide term with multiple applications and refers to the simulation of a human brain function by machines. AI is classified into two parts, general AI and Narrow AI. General AI refers to making machines intelligent in a wide array of activities that involve thinking and reasoning. Also referred as high-level AI, true AI, or AGI.

Narrow AI, on the other hand, involves the use of artificial intelligence for a very specific task. Machine learning (ML) [8] is a subset of the Narrow AI that focuses on a narrow range of activities, and is the only one currently implemented and used. ML is the ability of computer system to improve itself from experience without the need for any explicit programming. It can do so by three types of learning: supervised, unsupervised and reinforcement learning (RL).

Data science practical application of ML to analyze data and make predictions. However it is expressed by a human engineering effort, extracting relevant insights from data, which involves mostly statistics. Though there are studies try to automatize this effort [9].

AI as a proven method to deal with big-data, is a good method to use to tackle the large-scale traffic control problem. Also in the AI field the use of NNs (specially DNNs [10]) is also great to deal with non-linearity.

The author suggests a more general type of solution referred as AGI (Artificial *(*General) Intelligence), using NNs and not other statistical AI methods, such as logistic regression, support vector machines (SVM), decision-trees, decision forests, kernel machines, and Bayesian classifiers and more [11]. This is because these are all human-made mathematical inventions, while NNs are the closest thing we have to an intelligence - an imitation of a human brain.
[9]

Though there were purely symbolic systems, and there are also a combination of them with NNs, such as [12], [13], still this is only the middle of the way to AGI. In order to make flexible AGI, a full NNs system should be designed, without any limiting structures within it.

A. MOTIVATION

There are two types of motivation for this paper:

1) General: If we regard human mind as deep neural networks (DNNs), then people's consciousness is at the higher levels of this DNN, hence they are good at generalizing and simplifying things. Thus we use it for planning such as modeling and control.

However, we have generalized assumptions upon what we model and how to solve problems. We do not account for the fine details of a specific data set we deal with. So the optimal algorithms we use are good only for very limited cases, otherwise they are inefficient. So we build general solution that search "blindly" for solution, i.e. without considering the specific data we solve. On the other hand we can always "sit" and learn ourselves the data we deal with, but it will take plenty of time and efforts, but eventually we will get

much better tailored solution for this specific data. Either way, we are limited in our capacity.

Example: you can plan model and control strategy on traffic network, based on your general assumptions about them. But you can also plan it for a given network, observe it and learn it, and then plan.

Hence we have two choices here (see Fig. 1). We can either plan specifically for some task, or we can think more generally - how some AI can understand the data/system for multiple use/tasks. AI that actually learn the data and plan for us.

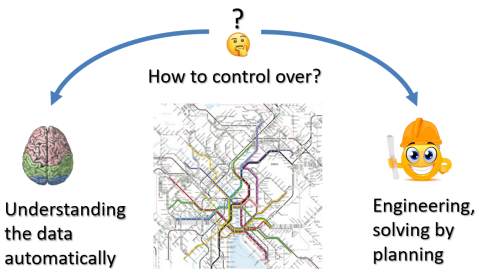


Fig. 1. How to control over some system?

2) Specific: This sort of things are the essence of DNNs or more generally of Machine Learning (ML), where we learn the data. But the aim of this research is multi-tasking that is not defined in advance, by some particular tasks. I.e. the goal is to end up with a system that learn the data and can operate on it in a variety of ways, efficiently. Today most ML models designed for single task, or multi-tasking with specific tasks. Any new task or type of data require retrain the model. The novelty here is about the ability to generalize not only for unseen data, but also over unseen tasks. More about this in III-.1.

Most important message of this research is that the goal is not to search best performance model, and compare it to other models. The goal is to try to understand what mechanism organize data in a usable way, no matter its primary quality. To build a slow machine but more intelligent worth it.

Full intelligence comprises of modeling the world (understanding the data), then after it operate in it, this is the control part. This paper investigates the modeling, as the encoder of a AE system, where the decoder is the controller which executes what was decided in the model part of the system fused with the user's will.

II. LITERATURE REVIEW

A. Similar ideas...

In robotics field can be found similar ideas to the general DLM proposed here. Different tasks are accomplished by a command→execution models, but mainly robotic arm movement [14]–[17].

One study [14] proposes a framework by combining composable instructions with visuomotor control for multi-task problems. Perception information of the environment is encoded by variational AE (VAE) into a small latent space. The embedded perception information and composable instructions are combined as state-action pairs by the LSTM module to guide robotic motion based on different intentions. The robot could move and manipulate objects for various purposes by actions: grasping, pushing, pouring, opening, picking, placing and so on. Here a one-hot vector is used to represent instructions, which means only one item of the vector is one and others are all zeros. Prospection is used not only for predicting the next action but also for predicting a sequence of future actions at the same time. Only the nearest future action is executed, and other future actions are only used to guide the robot to pay more attention to the global features (like in MPC).

Similar study [16] uses VAE-GAN (Generative Adversarial Network) instead of VAE as the perception module.

Next study [15] developed a sampling-based robotic planner (for arm moving tasks) that can be trained to understand sequences of natural language commands. They take a sentence, breaks it down into pieces, these pieces are translated into small networks, which trained to understand single concepts, which are recombined back together, thus can uncover and represent the meaning of whole sentences. This model is interpretable, i.e. allow researchers to pinpoint issues and to improve.

Other study [17] developed an algorithm for learning stable hand grasps of unknown objects based on an object shape classification and on the extraction of some associated geometric features. Though no commands present here, but planning do occur as in previous studies, which demonstrate the fact that instructions can be implicit and embedded inside a given input, indistinguishable from it.

B. Prior knowledge in DNNs

As shown from the above studies [18]–[22], CNN and RNN present better prediction results in accuracy and stability compared to other methods, such as support-vector-machines and random forest. However, this might be due to being tailored to the specific problem, which makes them best due to specificity and particular planning. Apparently, CNN and RNN are examples of simplification of the general NN, since they are more intuitive and understandable, with the price of being task-specific [18].

However, these special structures have less variables and contain more prior knowledge, compared to fully-connected (FC) layers of regular (vanilla) NN. Hence we can see the transform from FC to CNN or to RNN as localization, where the network structure designed specifically to the data it handles. Another example we will see later when we will introduce graph input, which will require an appropriate graph NN model to handle it.

Prior knowledge can appear in many forms: in the general type of the structure (e.g. different DLMs), in the hyper-parameters, in the regularization method (e.g. dropout, constraints, etc), in the decision about sharing or grouping [23] or separating features/variables and more. E.g. CNN uses sharing parameters. Most studies perform 3D convolutional kernels on an RGB input, thus treating this data as grouped

information of each pixel. Other studies may treat them separately, where each of the RGB channels do not merge with others throughout the whole DLM. Similar is when we set apart the depth channel from RGB in a RGB-D input images. Or when we set apart traffic data from weather data [24], or roads from stations [25] and fuse them only later (how later is also prior knowledge to be decided upon). We can also separate tasks to groups [25].

However all these structures can be too restricted or best perform for a narrow data variations. Hence many studies try different hyper-parameters to get better performance. A more comprehensive topic where such more flexible models are studied called Network Architecture Search, e.g. AutoML [26], Neuroevolution, hypernetwork, Meta-learning (learning over learning) [27] and more, which are all searching or adapting a given architecture, to better fit the data.

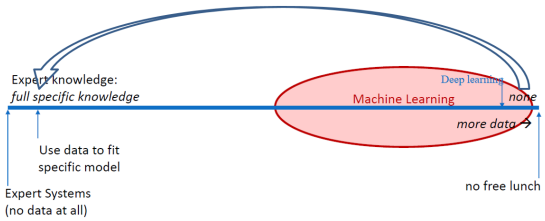
Nowadays, ML theory assumes best performance is in DL, see Fig. 2(a). Especially for large data set [28], see Fig. 2(b), where the main effect is for big data, since then there is a difference between the models. But in small data the different sizes of NNs are all behave similarlyapproximately the same.

However, in our humble opinion, unlike the place where AI supposed to be to the most right part of the scale, where we almost totally dependent on data (minimal prior knowledge), it is actually the opposite - we need full knowledge for designing AI (lots of assumptions), which do not depend on data at all, but is flexible and adaptive, and able to tune gradually and gently according to it. Whose more important here? It is not the data, but rather the assumptions and lots of planning of AI. Similarly to planning any system, e.g. control system, here we make it general as possible that account for as most cases as possible.

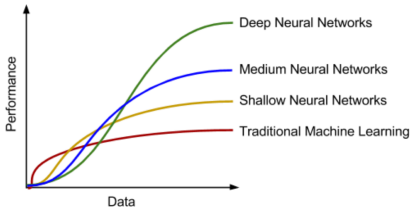
Same idea in [29]: *"learning ability does not come for free, and is far from automatic. It relies on very specific assumptions that are mostly encoded in the design of the NN architecture itself - here denoted as structural priors."*.

[29] also mention the Occam's razor approach, that a more complex model with more parameters, will be able to explain a wider range of data, however a simpler model will necessarily assign higher probability to the narrow range in which the data of interest lies. I.e. we should plan the AI to fit the data (all its variants) and not beyond it, since it will ruin its performance.

[29] suggests sparsifying the NN (e.g. removing connections in CNN, or grouping/sharing parameters) - results with less parameters and more prior knowledge and efficiency, similar to the effect of hand feature engineering. It also removes redundancy.



(a) No free Lunch for highly-designed AI.



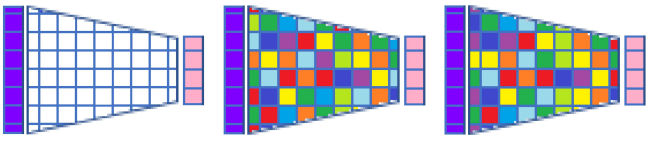
(b) Performance verse the amountdimensionality/scale of data, using different methods, taken from [28].

Fig. 2. Comparison between AI methods to comprehend the environment.

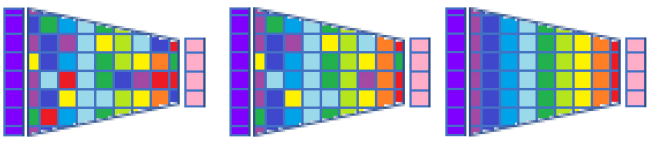
There is an issue to define the problem of AI. Unlike regular classification problem well defined in DNNs, it is difficult to know what is the overall purpose of AI. We believe for now, that it is to better predict or/and organize data. However prediction by itself is only the test for how organized data is. So perhaps prediction is not the goal of the supervisor, but rather only to its tool (inner or secondary objective) to estimate how well the organization is (strive to low entropy).

See for example in Fig. 3, a sketch diagram, illustrating how supervised learning in NN, where the data given is input and output. The NN is structured hierarchely, i.e. it is features of features, etc. So starting from random dis-ordered weights, we gradually use inputs and outputs as magnets, for diffusion or rearranging the weights in hierarchical way, where the most input-related features will be closest to the input and the same for the output.

Note that when training on random labels, then the task is actually merely memorization [30], since there is no consistency in the output data. Hence it also has no generalization.



(a) Input→NN→output (b) Start (c)



(d) (e) (f) Finish.

Fig. 3. Evolution of the parameters in supervised learning in NN (from right to left).

C. Cognitive Architecture

Finally, two studies [31], [32] tried to implement AI based on distributed *cognitive* architectures [33]. The first study [31] is based on Global Workspace Theory (GWT), where the local controllers usually perform a purely reactive behavior, and compete each other in order to define which of them is experiencing the most critical traffic situation. This architecture contains agents imitating brain functions, such as different types of memory and sensor, and implemented using codelets. The codelet is a sequence of machine instructions that can be represented by: NNs, fuzzy systems, evolutionary computation, rule-based systems, Bayesian networks and more. There arewhich contains agents imitating four types of brain functions, which implements four types of memories and sensorscodelet: (i) sensory codelettype, which holds positions and velocities of vehicles; (ii) behavioral codelettype, which determines the light of the traffic signal in a particular intersection; (iii) consciousness codelettype, which represents the working memory and interacts with other codeletsmemories/sensorsbrain functions; and (iv) motor codelettype, which executes the chosen signal phase for each intersection. Two regimes were tested in the study: (i) Parallel Reactive - where each intersection behavior codelet decides its phase based solely on the information it receives from its respective sensory codelet, and (ii) Artificial Consciousness - where intersections receive broadcast about the situation of the critical intersection in the network, and then decide based on this information its own next phase.

In the second study [32], an AI was implemented on a single intersection using a Multipurpose Enhanced Cognitive Architecture (MECA) [12], which was adapted for TSCP. The design consists of Cognitive manager, a special kind of agent managing a set of physical objects made available at internet, providing information about themselves and receive commands. E.g. car and an intersection agents. MECA is composed out of two independent systems that communicate with each other: (i) a fast reactive system whichthat holds the input sensors and output actuators, which is suited for automatic normal situations, and (ii) a motivational system which is goal-oriented and suited for unexpected situations.

Both papers show improvements in the overall mean travel time of vehicles in the network, compared to fixed timing plan controller, under different conditions. It is thanks to automatic reactive and the conscious elements, which together imply an intelligent behavior. The papers use simple cases of small scale networks, e.g. an isolated signalized intersection and a corridor of 4 intersections. Hence, such results that concentrate on specific performance index and study simple casestherefore they provide only a proof-of-concept. Moreover, real AI should be tested on various conditions, and with different objectives, for a more comprehensive analyzes. The presented architectures in [31], [32] are rule-based design, while future research should strive to develop architecture based on DL or biological models.

III. METHODOLOGY

1) Hierarchical principles: An important notion repeating itself in this research is hierarchy. It appears in different forms:

Fitting of model with the data it represents. No-free-lunch theorem [34] says that no learner can succeed on all learnable tasks: every learner has tasks on which it fails while other learners succeed. Therefore it must have some prior knowledge: either on distribution of the data or some finite hypotheses set.

It means that the information a person learns so succesfully have to be pre-adapted to him in advance, to his/her learning system. If there were no fit between the data and the learner, then most chances that effective learning would not occur.

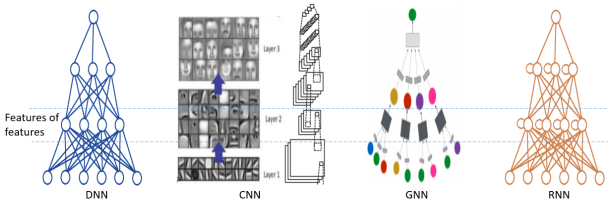
DNN is structure that fit the data it models, the data itself is hierarchical in essence, and what it does is organize the data in its natural structure, hierarchical features structure. Hence we can exploit this structure to solve problems (by search for solution) directly and not blindly as it performed in heuristically types of search breadth-first-search (BFS), depth-first search (DFS) and so on. It is due to the fact that all of these methods are random, logic-based and combinatorial (trial-and-error). Hence search and optimization tools that randomly try to optimize some goal are inefficient when working blindly on unorganized data. While the brain do the opposite, it solves problems directly, after organizing data. So solution is not optimal generally, but it is the best for the current data organization. E.g., when a person thinks of a several solutions for a problem, it is not via blind search, but associative direct processing, no permutations involved at all. We have a survival type of processing; it cannot depend on time-consuming search in some space for solutions. Therefore, we need to use this type of thinking in AI.

Hence, if we combine the idea from II-B and here we come to the following conclusion: just as we optimized blindly not considering data in the past, nowadays we highly-dependent on data but blind to the hidden assumptions about the models that learn it. Hence the role now is to combine these faults together: we plan the AI considering as many scenarios as possible as a function of all possible input data.

In Fig. III-.1(a) we can see visual comparison of the different NNs we encountered until now. The common to all structures that they are all hierarchical. And in Fig. III-.1(b) we see some basic comparison via some features, from [35].

Hierarchical parameter tuning. Just as we claim that NN structure is the best fit for the data it receives, and this is the prior knowledge, the same goes for the fact that our reality is ever-changing, always dynamic, only on different scales. There are stuff that change fast, and there are stuff that change slow. Same idea we propose to be in the hyper-parameters we tune in DNNs. Some should change frequently (e.g. weights) and some should change very slow (e.g. hyper-parameters). Many examples encourage this idea: Spall's SPSA [36], Network Architecture Search strategies (see MyDNN presentation).

Hierarchical learning. We learn by blind imitation via



(a) Visual comparison.

Component	Entities	Relations	Rel. inductive bias	Invariance
Fully connected	Units	All-to-all	Weak	-
Convolutional	Grid elements	Local	Locality	Spatial translation
Recurrent	Timesteps	Sequential	Sequentiality	Time translation
Graph network	Nodes	Edges	Arbitrary	Node, edge permutations

(b) Charachteristics comparison from [35].

Fig. 4. Comparison of different hierarchical structures in DL

supervised learning (input←desired output), but we propose to change it to gradual type of learning, where things are built one on top of the other.

”hierarchical learning”, meaning instructions or commands are learn in a pyramid way, as a child, we first teach basic instructions, then composite instructions and so on. So that it will enable us in the future create highly supervising system, that can have multiple object functions at once, regulating the changes in it and react accordingly. For instance instead of an operator tells the machine ”clear road 34 for presidential platoon”, it can teach it to supervise the system, like automatic pilot, and clear roads when presidential tweet occur in the text input.

This hierarchical learning composed of two types of ”stuff” to learn: objects and actions. So 1st teach basic elements/objects such as road, intersection, roundabout, pedestrian. Then continue to composite objects and when all objects learnt - actions can be learnt also.

It can learn using attention mechanism as guidance: We input text=”intersection” and in visual input a graph/image with attention on some intersection, then we train it again on intersection with attention on other intersections. Similarly we can teach it instances of objects, such as ”23th intersection” with attention on this specific intersection.

Similarly actions, e.g. ”from-to”, we show it a path from x to y, then we train it with different paths from same x to same y. And can do it for different x’s and y’s. Also we can start from learning specific objects with specific NN components, then learning new stuff by adding more components on-top of them and freezing the 1st components if needed. E.g. learn about objects such as roads and intersections and other elements of transportation network including text and other resources, and then add new NN to learn now operations over these objects. Also, this system is reversible, i.e. we do not need to learn from component to new component only, we can train actions after objects they operate upon and recall that we forgot to teach some object. So we can retrain only the relevant component, and then return where we were previously. Here [37] for example a gradual learning is proposed from a simple level to a complex level, either manually (expert guided) or automatically (scoring each sample by its training loss). However, this loss is highly dependent on the models and their hyper-parameters. Hence a different learning take place: from fewer categories or output tasks (local) to more categories (global).

A. Learning sequence

All above proposed architectures and more to come, will be using different learning input sequences to the AGI. eventually number of architectures multiplied by the number of learning sequences - all will be evaluated and compared, using some performance measures, at some points in time.

The basic sequence of learning, is starting from simple to complex. Starting from objects, than operations related to them. Finally using either some communication system to command the AGI, or using an external motivational system to dictate high-hierarchy goals.

Also, for adequate control, that considers the transportation laws and other constraints, we might start teaching the AGI where the executing part (actuators) is detached/removed from it, until the period that it understands and can percept this data.

This issue introduces the importance of different types of AGI’s mental experiences, such as learning, learning and acting/exploring, communicating or querying from the human operator (to perform specific tasks, such as controlling a network or solving a problem) and more.

Scalability is a part of hierarchical gradual AGI teaching. We start from simple (defining objects, i.e. junction, pedestrian, vehicle) and gradually add more complexity, where the full network is advance stage in AGI progress.

Also we can learn from baby’s development: in first months he’s gaining a lot information and do not use motoric system to explore. He sleeps alot since the orgsnizing or optimizing (such as backtracking) needed alot to start make sense, meaning that in waking there’s no processing involved but rather some type of storing for future arrangement. The more time passes the less sleep he actually needs. And at some point he gained the insight that he can use its motorics, and at some point also for exploring.

This can also derive a distinction between the left hemisphere which responsible for solving problems not analytically, similar to switching to other activities or meditation or sleeping, as mentioned, and the right hemisphere which responsible for conscious thinking. From the observation above, it seems that thinking do not mix with organizing data, but rather manipulating existing data, in its current order, from memory. So there are 2 processes: thinking which uses the current structure of knowledge, and self-organizing which is a background, unconscious process (since its internal restructuring cannot be viewed/interpreted logically as serial organized sequence of thoughts).

Important note: what exists in a lot of planned AI, such as specific functions and structures, are limiting in essence, so it is better to find an alternative to implement these ideas by teaching the AGI whose structure is as flexible and as general as possible. E.g. activity level of a drive in MECA system with parameters of urgency threshold [12].

Add more...

Many principles will be evolved in the development of this AGI. Most principles are come from philosophical and brain sciences (top to bottom effect). And from the practical side - principles that come from engineering, computer science, in particular data science and ML (bottom to top effect).

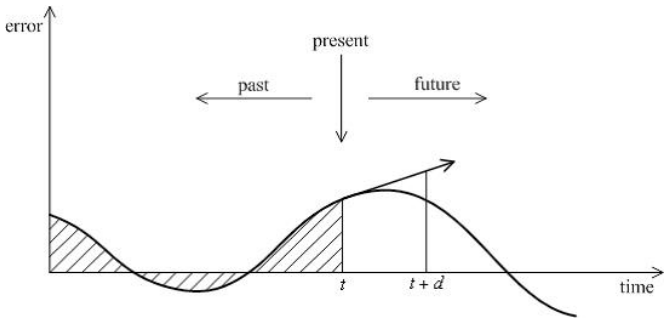


Fig. 5. PID error as a function of time, taken from [40].

B. Principles from Humanities (high order)

It is important to note, that most of the psychological and educational research in the field of intelligence, is actually about the external end of the inner and real intelligence. Intelligence test (e.g. IQ) and logical test are all about testing and analyzing the final edge of intelligence. Therefore, one has to consider this as the most external tool to conclude about how the inner intelligence operate. It's problematic in couple of things: 1st it might be very far to conclude, 2nd it's not only external but also highly developed in advance stage of it, and we have to extrapolate backward in time to figure how it has to start and develop.

1) Human plan effect: Human is the one that plans the AGI. Also, the desire is to have a logical output from AGI, and a human-like AGI, in-order to have a good communication and interpretation. All these creates naturally a limited AGI in advance [38]. Meaning we aim to tune the AGI to be fit to humans, so it must act as planned (so it is biased for humans). This limitation must be considered in the development.

2) Order verse disorder: Although we assume nothing about the data, we do assume that the data is patterned, meaning ordered or organizable. In other words [39], nested and hierarchical just as the brain.

Hence the data human learn is such, and the learning is actually organizing it. The architecture of the brain is designed to find order, and cannot function in chaotic data. May be because we need meaning and purpose from the data we encounter, which requires order.

This idea expressed also in the important aspect of the order of teaching the AGI. As in humans, it has to be from the simple to complex. Order also appears in the relation forms between data units: causality (cause-effect), hierarchy and more.

3) Convergent verse Divergent thinking: Convergent thinking is the process of gathering many ideas and process them into some implementation, while divergent thinking is the opposite - we have a problem, and now we start creating as many ideas as possible.

This idea can be related metaphorically also to the PID controller, as illustrated in Fig. 5.

Where integrated operation expressed using integral is converging using the past, and differential operation expressed using derivative predicts the future. Similarly, analysis verse synthesis, or feed-forward NN vs backtracking of feedback.

But PID is a good analogy for the full AGI system, which includes control also. Meaning having both sensors (input) and actuators (output), where the control is state-feedback control, similarly to MPC and IC, in the sense that it depends on the present and the future. The present expressed by the low-level fast/immediate control, while the future expressed in the full perception of reality not only in that instance but for some period/sequence, thus enabling also planning and prediction in the higher levels of hierarchy.

But additionally, as in AI methods, such as NNs or RL, it is dependent also on the past (experience).

Similarly to state-feedback controllers, such as MPC, IC, LQR, etc, here the state describes the traffic network that the AGI observe. The model is AGI's neural network, which decide upon the control. Also illustrated in Fig. 6.

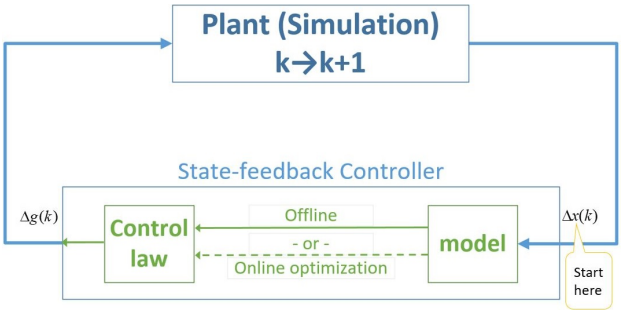


Fig. 6. State-feedback Controllers implementation on a plant.

4) NN black-box issue: Every part of an algorithm is interpretable, and we can easily understand what the algorithm does, but NN, which has also inputs and outputs like the algorithm is not interpretable.

It is important to separate the inner operation of the brain with its external manifestation by logic, language and thus explanation. Explanation/logic is the output of the NN, hence NNs today are used merely as a tool for control or function approximation, while missing their full potential or their original function as AGI (whose logic reached only via

communication and not via its internal structure).

The NN function is a black box to the designer. You may have an intuitive sense of how this function operates and the hidden features that this network has identified, but you do not know the reason behind the value of any given weight or bias. You also do not know what these features represent. So if the function does not meet a specification or if the operating environment changes, you will not know how to adjust the policy to address that problem.

Unlike NN, control system is explainable therefore manageable. You can also isolate a specific controller or loop and focus your analysis there, if an issue occur. You also cannot be sure it generalizes correctly unless you test it in different conditions, unlike control system which were planned after human generalization for as many as possible scenarios. In control system unlike NN, there are many types of formal verification/analysis: do not have to test to make sure a signal will always be nonnegative if the absolute value operation of that signal is performed in the software. Also: calculating robustness and stability factors like gain and phase margins.

Causality: understanding the effects of actions (sometimes called interventions, treatments). See regression for example: Understanding correlations is not enough to understand the effects of actions. Thats why NN isnt enough, sequential pattern recognition are necessary. Most applications of data science are predictive, not causal. They map inputs to outputs, but they do not consider how the world would look like under different courses of action (whether the diagnosis would change if we operated on the retina) which is called Counterfactual inference. Causality can be seen in algorithms and classical control planning (see 1 above). All of which eventually necessary for decision-making process, since causality allow test different scenarios of a given problem (using imagination), to choose the best set of actions. The purpose of NNs is not the logic of some process, but a representation of data, see (Deep Learning For Sequential Pattern Recognition/Pooyan Safari): The deep learning philosophy is fed by the assumption that the truth is not a solid monolithic thing, but rather a distributed concept flows along and among variety of things, phenomena, categories or here units and layers, which might all be integrated to superficially form a solid entity, event, being. One (human or machine) might not realize the whole, or even sometimes a part of it, without trying to perceive the overall structure and interactions between these single and monadic elements. Sometimes these individual elements have no meaning , each within itself, to us, however these are the individuals that form a complex, a whole. Its even more: The job of a physicist or an applied mathematician is often to come up with a functional description of a phenomenon from first principles so that its possible to estimate the unknown parameters from measurements and get an accurate model of the world. Deep neural networks, at the other end, are families of functions that can approximate a wide range of input/output relationships without necessarily requiring one to come up with an explanatory model of a phenomenon. In a way, youre renouncing an explanation in exchange for the possibility of tackling increasingly complicated problems. In another way, you sometimes lack the ability, information, or computational resources to build an explicit model of what youre presented with, so data-driven methods are your only way forward. (Deep Learning with PyTorch Essential Excerpts Eli Stevens and Luca Antiga). I come up with the understanding that logics and therefore explainability are both the output of thinking, while the thinking itself is unexplainable in essence, it only have a base, NN, where data can be self-organized, for logic to be founded on top of it. Since it first has to have organize model of the world. Also, using DNNs for tasks is not explainable since its not built for that. The explainable part will be if we combine textual NN. The NN is actually a system of nested conditional operators, so it is not so unexplainable.

In Fig. 7 there is a scheme representing how current action is decided. On what it depends.

PID is based on closed-loop error correction control, where I=integral correcting based on accumulating error from the past, P=proportional is correcting proportionally to the current error, and D=differential correcting by the expected change depending on the slope direction.

Similarly position→velocity→acceleration are the changes in the previous measure within this chain. I.e., the differential future expectation, while the opposite direction is the source and calculated via integration (of the past).

MPC, IC [41] are control methods based on a known model, and they calculate control for future steps, finally implementing the first move/control. Each time depending on the current state.

MECA [12] depends on past till present, and it also introduces randomality. It has two units, 1st one is reactive, depending on the past learning, while the 2nd one is goal-oriented, non dependent on the past, and performs according to some higher objective(s).

MECA is actually represent the right hemisphere of the brain, which is not dependent on previous learning, and search for new insight for a given situation.

DNN, DBN and RL are learn either from past examples or from past experimentation (RL), but they all act at each moment as state-feedback, meaning DNN/DBN are give output specifically by the given input, similarly RL has specific action by specific current state.

RNN however, introduce the ability to account for not only past learning but also past dependence for current reaction. Which is important for evaluation of the forming situation before our eyes.

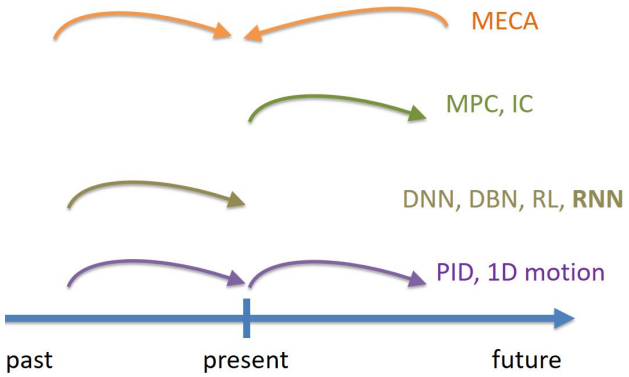


Fig. 7. How current action is decided, in different models.

5) Stimulated AGI (high objective): In humans, the brain is operate non-stop. When we sleep, close our eyes or meditate, or even by unaware physiological processes. But even cognitive operations work at all times. Meaning we have to excite the AGI, in times when its development from the input channels or from the self-organizing processes is terminated. It is due to the fact that any energy-dependent process in nature eventually dissipates, as it does here also. This effect occurs in adulthood when a person feels he has no significant cognitive development as he sensed in the earlier years. In relation to stability verse balance, the meaning is not to fixate in the middle of some range (e.g. between convergence and divergence), but to stimulate going to some edge, but keep balancing it using the contrary edge.

Other examples for running about the edges is in [5], where planning occurs in the middle, between fixation or concurrent data, and the dynamic/possible/random. Same idea goes for free-will as the mid of determinism and stochastic.

It also means that the system has another input beside the usual input channels. It has also some external control from a human user, either a direct control or a programmed one, that stimulate the AGI to keep the development process going.

It reminds the performance index or objective function from optimal control field. In reference to that term, the AGI also has to have such general objective, that manage the AGI. Just as anything in the world have a purpose and therefore a design, same is here. It can be either implicit or explicit. Philosophically it can be expressed also as some permanent deficiency, drive humans to fulfill and propagate.

As in NNs, where learning defined as the problem of improving some measure of performance when executing some task, through some type of training experience, similarly in AGI will be some internal performance index that strive to better the knowledge assimilation and organizing it in an efficient way. Additionally, an external goal to perform some human dictated tasks will be inserted directly into AGI or indirectly by commanding using communication.

As a part of this higher goal supervisor job, will be also handling attention [42].

6) Context generalization: Just as in vision, where we receive sensory information, but save it as a sequence in-order to identify some known pattern, similarly it is done with textual information, where we put each word in a context of a sentence, and than in the context of the last set of sentences, and finally in the context of the current speech and mood. Same can be done with any type of time-evolving information, such as the transportation network state [43], finally generalizing the situation in a hierarchical pyramid fashion [39].

The idea [39] is that at each instant, the AGI recognize the general state it is, after the gain of time-serial information, which can than as a feedback go down the hierarchy to control also how to collect the information more efficiently.

This idea can be implemented also on predictive behavior, for different time scales (from immediate response until long-range). But more important, hierarchical model can be also for operators on data that creates and updates our knowledge base. Starting from low operators to high operators, that can function as the organizers and the managers of AGI's structure or as called in psychology as meta-cognition.

7) Efficiency: AGI have to be efficient in the meaning that it has to stand in many tests. It must not be optimized only for some specific task, but rather have a multi-objective openness [6]. Optimization do not organize the data (but rather act upon some unknown data), therefore changing objectives to optimize have to perform new calculations from the beginning. But information organization take this case into account, to be efficient for many possible goals, such as efficient memory for fast retrieval, searching for best solution, searching for fast solution and more.

8) Growth Potential: This is important property of AGI, that distinguish it from other AI methods such as rule-based AI or static structures such as NN. It is the very important task of AGI designer to plan AGI such that it can evolve and develop further independently, with the anticipation of it to reach some stages of growth.

The problem with MECA for instance, as an example for a complex cognitive structure or a rule-based structure, is that if it is designed with lot of components than it is prone to further growth in complexity if new situations occur to deal with [36]. At some point it might become so complicated that it may loose its reliability/credibility.

9) Reasonable AGI: There problem of using CNN with LSTM with attention over an image to generate text caption describing an image [44], is that it is not intuitive. Meaning, this is simplified method to bypass the actual complexity needed for this task. Even if it works, this is not the way it can scale the AGI to develop further. Since text description is actually high-cognitive operation, that requires the knowledge of words, linguistic grammar, and more-over it has nothing to do with attention over an image beside the single attention on the object(s) described.

Similarly, supervised NN can be described as sticking a pin in a middle of a big NN delivering a label, that by back-propagating influence the hierarchy of properties needed to be distributed between this label and the inputs. Not only that, but number of outputs represent classifications or clusters, which represent some concepts. But in reality there are huge amount of concepts, so real intelligence supposingly

using NN should have huge number of outputs. I claim that most of the concepts suppose to be represented in the features spread in the network, rather being at the edge of the network.

Also in reality humans have no labeling at all. Same goes for language processing using documents.

These methods are working backwards: they start from highly complex data, such as language (even worse - an abstract terminology) and highly complex visionary scenery, and try to process it, with the deliberate intention that this data only need to be fitted, not understood.

It suppose to spread naturally from sensors to conscious and then to memory and so on..

In summary, in order for any AGI would work, it have to make sense.

ADD THIS: It seems we don't use supervised learning at all and reinforcement learning is just a primitive method of a more general one. At some point the baby start to use its motoric actuators (speech, muscles) but only after organizing the memory to grasp the skill for these in order to let it out. And when the actuators start to work, it can be added somehow as the input influencing the result we see.

The problem with most ai research either visual or textual that they work backwards. They always start from high-complexity data and try to reach it. Instead of simple to complex method as it should be.

Babies cry often if awaken (meaning interrupted sleep) perhaps because sleep mechanisms are more critical for them comparing to adults. Or perhaps just as any deficit such as food or any other need have to be satisfied fully and immediately.

ADD THIS:

כמו שכל ספירה ימנית משפיעה על השמאלית, החל מהראש (חכמה על בינה) כך גם חסד שמסמל את האור והיום לעומת דין המסמל את הלילה והדין שאדם נותן בשנתו. ביום לאדם יש פעילות ספיגה וקליטה של כל מה שארע ובלילה יש פעילות אחרת כשאזורי הקליטה מכובים. אז נכנס לפעולה פעילות הפוכה (שעם קליטה לא יכולה לעבוד בו זמנית כנראה כי אולי הן סותרות). פעילות בעיקר של סדר וניקיון של היום שעבר. כעת במקום כיוון ספיגה זה עובד לכיוון ההפוך. אולי פריקה לעומת טעינה. כנראה יש זיכרון אפיזודי של אירועים שמתמלא במהלך היום וזקוק לפירוק. לכן אסור לאדם לא לישון. כי זה לא מאפשר לא לנתח ולארגן אירועים שכבר הצטברו אצלו. הוא חייב תהליך שינה כדי לפנות מקום לזיכרון היומי הזה. כלומר מחזור יום-לילה בעצם סוג של מאתחל את המח ליום חדש, אלא הפעם פעילות הסידור של הלילה יצרה אישהו סדר טוב יותר של האירועים. סדר שמשפיע על כל סוגי הזיכרון שיש לטווח הארוך. גם אירועים גם הצהרתי וגם חושי. לכן כל יום יש התקדמות.

Fig. 8.

Just like backpropagating and feed-forward can't act simultaneously, similarly wakening absorbing data and process it cannot occur together with reordering the system itself. Otherwise it makes the brain inconsistent. During the day you act as you are. And you upgrade the system when its non working.

Should explain the difference between DNN and CNN for example object image recognition verse actual high-complexity abstract object recognition, where we look at a scene and more than simple recognize bodies in it but also recognize their behavior and functionality. A meaningful interpretation and understanding (show the picture and add it source). Than i explain the stick in CNN OR DNN unlike DBN that allows the wakening propagation move forward to all other parts of the brain such as memories and etc.

C. Principles from Exact sciences (low order)

1) Evolutionary Algorithms: Ant colony, GA and even Reinforcement Learning (RL) are examples for applying randomization together with keeping/storing the best results gained so far [45]. Ant colony use random paths to find optimal solution, but preferring the most walked paths so far. GA is a bit less random in the sense that for a chosen population the new members in it are only a random mutation of the best members so far, but not a totally random members. RL on the other side starting as most stochastic trial-and-error exploration, but as the time passes there is less exploration and more adapting best rewarded actions.
All these methods represent a "calculated risk". Meaning a mid point between total randomness and total accuracy, e.g algorithmic.
One suggestion may be yielded from this to DNN back-tracking optimization algorithm: not to choose the steepest descent, but rather some fraction of it, in-order to leave some room for future learning and reduce over-fit to the data. Or to use regularization method that constraining the optimization problem (as if pushing the brakes on optimization) and reduces over-fit this way.

In any case, we want a fast updating weights method, very fast for huge NNs. Unlike precise GD/SGD, slow convergence but low number of examples N, we will use fast convergence and high N.

IV. Proposed AGI model

A general sketch of the first draft describing the AGI model/structure is shown in Fig. 9.

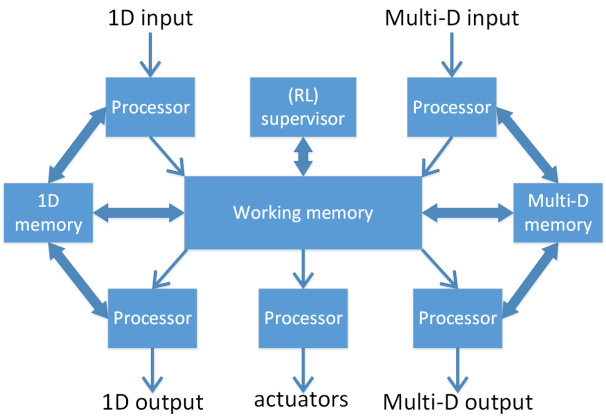


Fig. 9. AGI possible architecture.

is based on global workspace theory (GWT) [31], [46], [47], only here there is no competition among independent agents in the system, but rather centralized control with different elements with specific functioning.

But I don't believe in dichotomous structures such as in Fig. **??** and in Fig. 9, since they are limited in nature (as explained in Section **??**). I rather believe in continuum that appears in fuzzy logic and in Section III-B.5. Nevertheless, we need to take into account different function components

in these architectures and interactive-ties between them, but also remember that the boundaries between them are not so well defined and are changing, see Fig. 10.

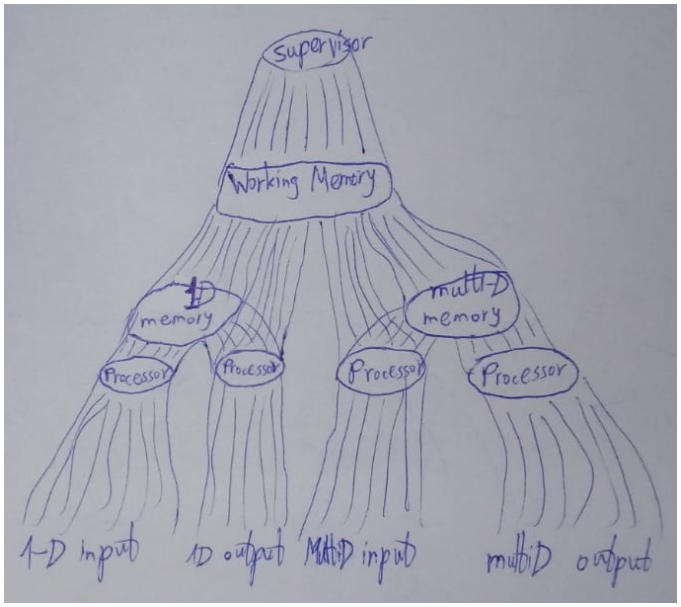


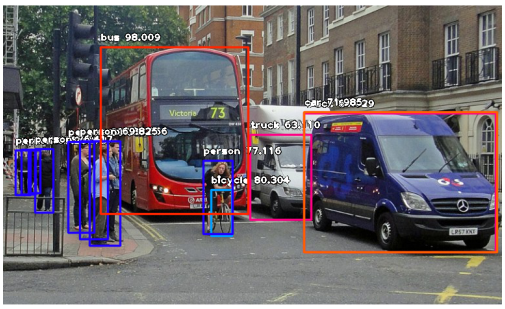
Fig. 10. AGI possible architecture in 3d hierarchical pyramid form.

There is sequential processing of 1D and multi-D data separately, for data identification. Eventually using it in the working memory. And when event for output occurs, such as in case some idea or thought emerged, then the output can either go through the 1D channel, as humans describe their inner thoughts to the outer world verbally, or through a multi-D channel, which is an extension of human structure (if it is possible at all), and can be regarded as "screening imagination". Because if we have 2 types of inputs, why shouldn't there be outputs of these types either? And of-course 1D information have some common memory for both input and output and the working memory (such as language), and likewise for the multi-D. The double arrows mean the acquisition and update of this storage.

Output communication is also an issue. Should it be monitoring thoughts, or wait for meaningful output, or maybe even AGI will have some sort of free-will to choose when to interact and with what.

Notice, that this particular AGI based upon Stimulus-Response behavioral theory [48], that claims that we cannot observe the mind itself, directly, but rather communicate with it. This assumption is of-course the worst-case scenario. It is also called intelligent behavior, and expressed by humans in many scientific fields, such as engineering, math, spiritualism, computer sciences, education, psychology and more.

Two types of data are needed to perform inference, since we use meaningful words (1-dimensional input) to construct world representation, seen by the multi-dimensional input. Also, it is not only identification of objects, but also their meaning and predicted behavior, as illustrated in comparison in Fig. 11



(a) AI identifying.

(b) AGI interpretating.

Fig. 11. Comparison between AI and AGI comprehending the environment.

Notice that we have an asymmetrical component of control in the AGI: the actuators (in our case traffic signals). Humans have touch senses and motor control very interconnected and interdependent [49], as can be perceived from third Newton's law, i.e. when one push/pull something (control action) he immediately sense it, as a feedback to confirm its prediction. However, in our specific case there is no sense for traffic signals, they are controlled intrinsically (ultimate control).

The purpose of having two types of channels are for distinguishing the outer and inner world that the intelligence interacts with. The outer world is objective, and the details in it can be agreed upon all AGIs, through the visual multi-dimensional channel. But each AGI is individual in the sense that it builds its own special and unique inner representation. But furthermore, humans (as should be followed by AGI) base their inner representation on meaning, which expressed by words or any symbolic language. This is why meaning can be transfered to humans not by the objective world, but rather only by other meaning-based beings using the meaning-based language, which in our case is 1D sequential type of data. Illustrated in Fig. 12.

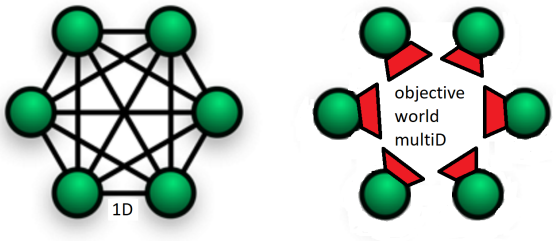


Fig. 12. Objective (right) verse Subjective inner representation (left).

This is the reason why there are no labeled outputs in DNN in reality, other than the one that represent meaning, which is given by the 1D communication with other similar beings.

Also, the equivalence of generating conclusions is actually finding some order in the data or pattern recognition, but in a more general sense rather than specific narrow-type pattern. And all this is by using 1D symbolic pattern/model construction.

Notice, that labeled DNNs are actually a bypass of the actual intelligence development. Since an infant starts from unsupervised learning, and only after a long time period can make connection between the objects he seen and their meaning (either vocal or symbolic).

In DNNs it is known that network's layers are order from 1st layer as most specific to the last layer as most general. This means that one way to deal with fix structure and the problem of generalization of hypothesizes (where small data requires simpler model and bigger data require a more complex model), is by enlarging the network by adding more neurons in the layers or adding more layers. Which makes the structure dynamic, and apt for development as in humans.
Also we'll use artificial cognition papers [12], [31], [33], [47], [50]–[53], for suggestion of different cognition architectures and models.

More about AGI in literature: [33], [54]–[59].

Note: If we want to develop not only AGI model, but also control, meaning add to the system actuators as additional output, then Fig. 9 has to be updated by taking them into account.

Fig. 13

Fig. 14

Try to implement hierarchical principles III-.1. Especially an idea I was developing very early in my study. The idea is generally come from the fact that regular NNs have specific outputs. But as human learn more and more, he add more layers/abstraction or a new NNs branching from the old ones. So we need to think how to expend a given network. Also if we combine this with the idea of hierarchical learning, where we teach the DLM a specific task, then move on to a more complex task based on previous task(s), thus adding more layers for that reason. Then I suggest why don't we do this automatically, as I assume our brain does? I.e. we can compare our fully-guided hierarchical learning to some automatic hierarchical learning, in which some supervisor

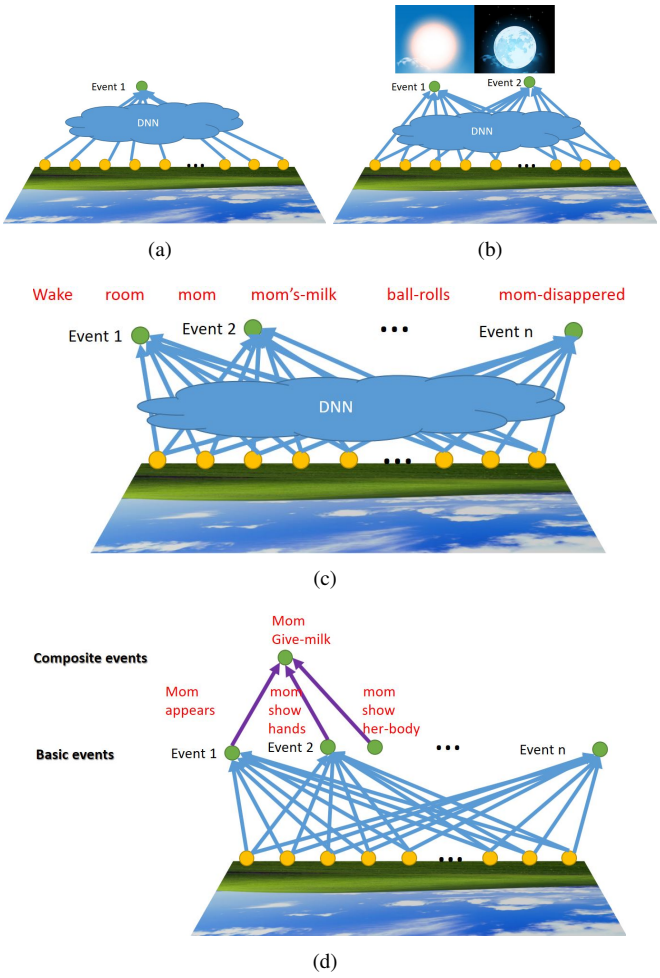
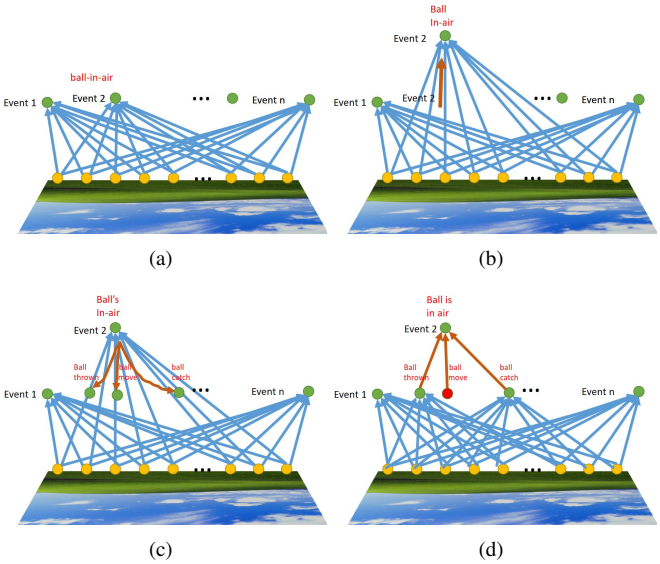


Fig. 13. Comparison between AI and AGI comprehending the environment.

program knows how to change the NN in dependence of the data it receives. Perhaps it is done by memorizing all the "un-attended" data that couldn't been handled in the awaken period, while we use only the proven-to-be-effectively-predictive NN in the awaken periods. Implementation of it can utilize hierarchical parameter tuning and some Network Architecture Search technique.
Guided teaching may be a problem, since we might be wrong in the correct order of teaching. E.g. we teach it very early to detect accidents in the network. We might realize later that for good detection it must be taught concepts as intersection and roads, also different modes of normal situations, and more. Hence the development of automatic learning is crucial in the long run.

This idea is also resembles the biological model of Jeff, where we pick the simplest data we can predict well, then we learn to predict more abstract data only after the more fundamental data has been established.

We can also use the model-fit-data principle in the decision of how to expend the NN. I.e. just as CNN best fit for visual feature extraction, and RNN use recurrent connection since the sequenced data is recurrent in nature, and GNN use adjacency of neighbors since the data is behaves this way,



(a) (b) (c) (d)

Fig. 14. Comparison between AI and AGI comprehending the environment.

we can check different NNs to best fit the extension of the present NN.

V. CONCLUSION

REFERENCES

[1] B. Abdulhai, R. Pringle, and G. J. Karakoulas, "Reinforcement learning for true adaptive traffic signal control," *Journal of Transportation Engineering*, vol. 129, no. 3, pp. 278–285, 2003.

[2] M. G. Karlaftis and E. I. Vlahogianni, "Statistical methods versus neural networks in transportation research: Differences, similarities and some insights," *Transportation Research Part C: Emerging Technologies*, vol. 19, no. 3, pp. 387–399, 2011.

[3] A. Ata, M. Khan, S. Abbas, G. Ahmad, and A. Fatima, "Modelling smart road traffic congestion control system using machine learning techniques," *Neural Network World*, vol. 29, no. 2, pp. 99–110, 2019.

[4] P. Gora and M. Bardoński, "Training neural networks to approximate traffic simulation outcomes," in *2017 5th IEEE International Conference on Models and Technologies for Intelligent Transportation Systems (MT-ITS)*. IEEE, 2017, pp. 889–894.

[5] E. De Bono, *Parallel thinking*. Random House, 2016.

[6] [Online]. Available: https://simania.co.il/bookdetails.php?item_id=145306

[7] E. De, *Parallel thinking: From Socratic thinking to de Bono thinking*. Penguin Books, 1995.

[8] M. I. Jordan and T. M. Mitchell, "Machine learning: Trends, perspectives, and prospects," *Science*, vol. 349, no. 6245, pp. 255–260, 2015.

[9] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, p. 452, 2015.

[10] Y. Bengio *et al.*, "Learning deep architectures for ai," *Foundations and trends® in Machine Learning*, vol. 2, no. 1, pp. 1–127, 2009.

[11] T. Hastie, R. Tibshirani, J. Friedman, and J. Franklin, "The elements of statistical learning: data mining, inference and prediction," *The Mathematical Intelligencer*, vol. 27, no. 2, pp. 83–85, 2005.

[12] R. Gudwin, A. Paraense, S. M. de Paula, E. Fróes, W. Gibaut, E. Castro, V. Figueiredo, and K. Raizer, "The multipurpose enhanced cognitive architecture (meca)," *Biologically Inspired Cognitive Architectures*, vol. 22, pp. 20–34, 2017.

[13] G. Granlund, "A cognitive vision architecture integrating neural networks with symbolic processing," *Künstliche Intelligenz*, vol. 2, pp. 18–24, 2005.

[14] Q. Shao, J. Hu, W. Wang, Y. Fang, M. Han, J. Qi, and J. Ma, "Composable instructions and prospection guided visuomotor control for robotic manipulation," *International Journal of Computational Intelligence Systems*, vol. 12, no. 2, pp. 1221–1231, 2019.

[15] Y.-L. Kuo, B. Katz, and A. Barbu, "Deep compositional robotic planners that follow natural language commands," *arXiv preprint arXiv:2002.05201*, 2020.

[16] R. Rahmatizadeh, P. Abolghasemi, L. Bölöni, and S. Levine, "Vision-based multi-task manipulation for inexpensive robots using end-to-end learning from demonstration," in *2018 IEEE International Conference on Robotics and Automation (ICRA)*. IEEE, 2018, pp. 3758–3765.

[17] F. Ficuciello, A. Migliozzi, G. Laudante, P. Falco, and B. Siciliano, "Vision-based grasp learning of an anthropomorphic hand-arm system in a synergy-based control framework," *Science Robotics*, vol. 4, no. 26, 2019.

[18] Y. Wu, H. Tan, L. Qin, B. Ran, and Z. Jiang, "A hybrid deep learning based traffic flow prediction method and its understanding," *Transportation Research Part C: Emerging Technologies*, vol. 90, pp. 166–180, 2018.

[19] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, "Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks," *Sensors*, vol. 17, no. 7, p. 1501, 2017.

[20] X. Ma, Z. Tao, Y. Wang, H. Yu, and Y. Wang, "Long short-term memory neural network for traffic speed prediction using remote microwave sensor data," *Transportation Research Part C: Emerging Technologies*, vol. 54, pp. 187–197, 2015.

[21] Y. Tian, K. Zhang, J. Li, X. Lin, and B. Yang, "Lstm-based traffic flow prediction with missing data," *Neurocomputing*, vol. 318, pp. 297–305, 2018.

[22] M. Aqib, R. Mehmood, A. Alzahrani, I. Katib, A. Albeshri, and S. M. Altowaijri, "Smarter traffic prediction using big data, in-memory computing, deep learning and gpus," *Sensors*, vol. 19, no. 9, p. 2206, 2019.

[23] M. Schlichtkrull, T. N. Kipf, P. Bloem, R. Van Den Berg, I. Titov, and M. Welling, "Modeling relational data with graph convolutional networks," in *European Semantic Web Conference*. Springer, 2018, pp. 593–607.

[24] A. Koesdwiady, R. Soua, and F. Karray, "Improving traffic flow prediction with weather information in connected cars: A deep learning approach," *IEEE Transactions on Vehicular Technology*, vol. 65, no. 12, pp. 9508–9517, 2016.

[25] W. Huang, G. Song, H. Hong, and K. Xie, "Deep architecture for traffic flow prediction: deep belief networks with multitask learning," *IEEE Transactions on Intelligent Transportation Systems*, vol. 15, no. 5, pp. 2191–2201, 2014.

[26] AutoML. Automl. [Online]. Available: https://towardsdatascience.com/how-to-apply-continual-learning-to-your-machine-learning-models-4754adcd7f7f

[27] Z. Li, F. Zhou, F. Chen, and H. Li, "Meta-sgd: Learning to learn quickly for few-shot learning," *arXiv preprint arXiv:1707.09835*, 2017.

[28] R. Chalapathy and S. Chawla, "Deep learning for anomaly detection: A survey," *arXiv preprint arXiv:1901.03407*, 2019.

[29] Y. A. Ioannou, "Structural priors in deep neural networks," Ph.D. dissertation, University of Cambridge, 2018.

[30] C. Zheng, X. Fan, C. Wang, and J. Qi, "Gman: A graph multi-attention network for traffic prediction," in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 34, no. 01, 2020, pp. 1234–1241.

[31] A. L. O. Paraense, K. Raizer, and R. R. Gudwin, "A machine consciousness approach to urban traffic control," *Biologically Inspired Cognitive Architectures*, vol. 15, pp. 61–73, 2016.

[32] R. Gudwin, A. Paraense, S. M. de Paula, E. Fróes, W. Gibaut, E. Castro, V. Figueiredo, and K. Raizer, "An urban traffic controller using the meca cognitive architecture," *Biologically inspired cognitive architectures*, vol. 26, pp. 41–54, 2018.

[33] A. Lieto, M. Bhatt, A. Oltramari, and D. Vernon, "The role of cognitive architectures in general artificial intelligence," 2018.

[34] S. Shalev-Shwartz and S. Ben-David, *Understanding Machine Learning: From Theory to Algorithms*. Cambridge University Press, 2014.

[35] P. W. Battaglia, J. B. Hamrick, V. Bapst, A. Sanchez-Gonzalez, V. Zambaldi, M. Malinowski, A. Tacchetti, D. Raposo, A. Santoro, R. Faulkner *et al.*, "Relational inductive biases, deep learning, and graph networks," *arXiv preprint arXiv:1806.01261*, 2018.

[36] J. C. Spall and D. C. Chin, "Traffic-responsive signal timing for system-wide traffic control," *Transportation Research Part C: Emerging Technologies*, vol. 5, no. 3-4, pp. 153–163, 1997.

[37] H. Cheng, D. Lian, B. Deng, S. Gao, T. Tan, and Y. Geng, "Local to global learning: Gradually adding classes for training deep neural networks," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 4748–4756.

[38] F.-Y. Wang, X. Wang, L. Li, and L. Li, "Steps toward parallel intelligence," *IEEE/CAA Journal of Automatica Sinica*, vol. 3, no. 4, pp. 345–348, 2016.

[39] J. Hawkins and S. Blakeslee, *On intelligence: How a new understanding of the brain will lead to the creation of truly intelligent machines*. Macmillan, 2007.

[40] [Online]. Available: http://sysbook.sztaki.hu/sysbook6.php

[41] S. Komarovsky and J. Haddad, "Robust interpolating traffic signal control for uncertain road networks," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 3656–3661.

[42] Z. Yang, D. Yang, C. Dyer, X. He, A. Smola, and E. Hovy, "Hierarchical attention networks for document classification," in *Proceedings of the 2016 conference of the North American chapter of the association for computational linguistics: human language technologies*, 2016, pp. 1480–1489.

[43] M. Bielli, G. Ambrosino, M. Boero, and M. Mastretta, "Artificial intelligence techniques for urban traffic control," *Transportation Research Part A: General*, vol. 25, no. 5, pp. 319–325, 1991.

[44] K. Xu, J. Ba, R. Kiros, K. Cho, A. Courville, R. Salakhudinov, R. Zemel, and Y. Bengio, "Show, attend and tell: Neural image caption generation with visual attention," in *International conference on machine learning*, 2015, pp. 2048–2057.

[45] P. Lucic and D. Teodorovic, "Transportation modeling: an artificial life approach," in *14th IEEE International Conference on Tools with Artificial Intelligence, 2002.(ICTAI 2002). Proceedings.* IEEE, 2002, pp. 216–223.

[46] R. C. M. da Silva and R. R. Gudwin, "An introductory experiment with a conscious-based autonomous vehicle," in *4th Workshop in Applied Robotics and Automation*, 2010.

[47] J. A. Reggia, "The rise of machine consciousness: Studying consciousness with computational models," *Neural Networks*, vol. 44, pp. 112–131, 2013.

[48] J. B. Watson and P. Meazzini, *John B. Watson*. Il mulino, 1977.

[49] B. J. Baars and N. M. Gage, *Cognition, brain, and consciousness: Introduction to cognitive neuroscience*. Academic Press, 2010.

[50] K. Raizer, A. L. Paraense, and R. R. Gudwin, "A cognitive architecture with incremental levels of machine consciousness inspired by cognitive neuroscience," *International Journal of Machine Consciousness*, vol. 4, no. 02, pp. 335–352, 2012.

[51] D. Hassabis, D. Kumaran, C. Summerfield, and M. Botvinick, "Neuroscience-inspired artificial intelligence," *Neuron*, vol. 95, no. 2, pp. 245–258, 2017.

[52] R. R. Gudwin, "Evaluating intelligence: A computational semiotics perspective," in *Smc 2000 conference proceedings. 2000 ieee international conference on systems, man and cybernetics.'cybernetics evolving to systems, humans, organizations, and their complex interactions'(cat. no. 0*, vol. 3. IEEE, 2000, pp. 2080–2085.

[53] S. Ribeiro, A. Loula, I. de Araújo, R. Gudwin, and J. Queiroz, "Symbols are not uniquely human," *Biosystems*, vol. 90, no. 1, pp. 263–272, 2007.

[54] B. Khayut, L. Fabri, and M. Avikhana, "Modeling of intelligent system thinking in complex adaptive systems," *Procedia Computer Science*, vol. 36, pp. 93–100, 2014.

[55] A. Cortese, B. De Martino, and M. Kawato, "The neural and cognitive architecture for learning from a small sample," *Current Opinion in Neurobiology*, vol. 55, pp. 133–141, 2019.

[56] P. R. Lewis, A. Chandra, S. Parsons, E. Robinson, K. Glette, R. Bahsoon, J. Torresen, and X. Yao, "A survey of self-awareness and its application in computing systems," in *2011 Fifth IEEE Conference on Self-Adaptive and Self-Organizing Systems Workshops*. IEEE, 2011, pp. 102–107.

[57] J. M. Chein and W. Schneider, "The brains learning and control architecture," *Current Directions in Psychological Science*, vol. 21, no. 2, pp. 78–84, 2012.

[58] D. S. Levine, *Introduction to neural and cognitive modeling*. Routledge, 2018.

[59] W. Duch, R. J. Oentaryo, and M. Pasquier, "Cognitive architectures: Where do we go from here?" in *Agi*, vol. 171, 2008, pp. 122–136.