Is Artificial Intelligence Always Intelligent?

Akira Imada

Brest State Technical University
Moskowskaja 267, Brest 224017 Republic of Belarus
akira@bstu.by

Summary. We have lots of successful reports claiming an establishment of an artificial intelligence by exploiting one of the state-of-the-art machine learning techniques. However, we doubt some of these claims more or less. If an artificial agent has an intelligence in a real sense, it may sometimes enjoy a different option that still results in a solution of the problem given, while some of the agents reported showed an excellent performance but always the identical one. In order to take it a consideration on this issue, we observe behaviors of an agent, specifically controlled by a neural network here, in a very simple environment of a two dimensional grid-world without any obstacle in it. This is not a report of success but a speculation of so-far claimed artificial intelligences. Then finally we propose a benchmark – two dimensional jeep problem – for a serious challenge.

1 Introduction

There have been lots of reports which claimed a success of establishing an artificial intelligence with so-called a *machine-learning technique.* Sometimes, however, we wonder what is intelligence. Assume, for example, we are in a foreign country where we are not so conversant in its native language, and assume we ask, "Pardon?" or something like that, to show we have failed to understand what they were telling us, then intelligent people might try to change the expression with using easier words so that we understand this time, while others would repeat the same expression perhaps with a little bigger voice. In this paper, we take a look at a possibility of, specifically here, artificial neural networks as a case study. Then the question is, *"Can an artificial neural network create such a flexible response like a human intelligence?"*

Regardless of structure of such neural networks, say, multi layer perceptron or recurrent neural network, or regardless of model of each neuron, say, McCulloch-Pitts model or spiking neuron, if we determine the weight configuration of those connection between neurons in a deterministic way, such as Back Propagation or whatever else, the resultant behavior of an agent controlled by such neural network will be deterministic too. However well it might adjust to an environment, or however successfully it might result in an optimized behavior, they simply repeat the action when it is to perform the action multiple times. When, on the other hand, we human go for a walk every morning, for example, we sometimes try a different course without any specific reason even if the environment is not different from the yesterday's.

We start our contemplation with a simple experiment. An agent controlled by a neural network explores a two dimensional grid-world in which the agent must spend one unit of energy to move one cell to the next. The size of the grid-world is infinitely large. The goal of the agent is maximize the entire distance till it eventually reaches before consuming all its energy given at the start and without a recharge after the departure.

It would not be of our interest when we determined a weight configuration of the neural network which control the agent in a deterministic way. Though in a real world application, it might be very useful when an agent solve the problem in exactly the same way every time, our interest here is if an agent can choose a different solution among others, if any, that also fulfills the same goal.

The above mentioned task of exploring a two-dimensional grid-world gives us such an environment. Assuming the agent is given q units of energy, the number of paths that has the maximum distance from the starting point when measured in Hamming distance is:

$$4 \sum_{r=0}^{q} (\, {}_{q}C_{r} \,) \, . \tag{1}$$

Quite a large number if q is sufficiently large. Then our task is: *"How we find the recurrent network which makes an agent take a different routes from those many options at each time of trials?"*

Floreano et al. [1] once proposed such a method in which they trained a physical micro robot controlled in a physical small world also using a recurrent neural network to control the robot by evolving the learning rules which are rules divided the Hebbian learning rule into four separate rules, based on the concept proposed by Wilshow et al. [2]. Later, Stanley et al. [3] elaborated the algorithm by evolving a single general learning rule for both excitatory and inhibitory connections by combining the properties of Floreano et al's rules, instead of dividing Hebbian learning into separate rules.

In this paper, we simply search for a rule by which each of the connections determine the degree to how it follows the Hebbian rule, in order to check if this scheme works in our simplified situation, that is, *"Can the agent after* *training enjoy a different route, which is one of the existing whole solutions, whenever they start the action from scratch?"* Thus, starting with a random weight configuration anew at every start of trials, agent applies the obtained learning rule at every step from one cell to the next.

An important difference of our experiment from those by Floreano et al. and Stanley et al. is that we train the agent in a grid-world where we have no obstacle in it. Such an environment was once explored by Stolle et al. [4] in the context of their reinforcement learning. Our concern here is, if the agent can take a different but still intelligent action at every trial, and furthermore, if such an agent who have learned the environment with obstacles can avoid them that it has never learned in its training phase.

Then we finally propose a more sophisticated benchmark as a challenge by extending those experiments mentioned above. The agent is allowed to store a part of its energy at any location for a future usage, and also allowed to return to the base, which locates in the center of the grid-world, to recharge its energy and start the base again.

2 Methods

As a preliminary experiment, we train agents in a very simple environment – no obstacle at all. The goal of the agent is to go as far as possible starting from the center of the infinitely large grid-world consuming energy to move. There are many such options of routes. The question is, *"Whether or not the agent who learned how to explore along a route with the maximum distance can change its route to the different one that is still maximum at the next trial."* Let us now summarize the task.

Task 1. *Starting from the center of the two-dimensional grid-world with q unit of energy, an agent must travel as far Manhattan distance as possible from the starting point spending one unit of energy to move from one cell to the next.*

2.1 A recurrent neural network approach

We control the agent, in this paper, by a fully connected recurrent neural network made up of $N(> 6)$ neurons each of which takes the state of either 1 or -1. An instantaneous state of a neuron is updated asynchronously as:

$$s_i\,(t+1) = 2\ sgn\left(\sum_{j=1}^{N} w_{ij} s_j(t)\right) - 1 \qquad (2)$$

where w_{ij} is the weight value of the connection from neuron j to neuron i, and $s_i(t)$ is a state of the neuron i at time t. The two out of those N neurons work as outputs so that the agent moves one step to north, south, east, or west when the two states are (-1, -1), (-1, 1), (1, -1), or (1, 1), respectively. The other specific four neurons work as sensors. That is, when the agent sees an obstacle at time t in its forward direction within p cells, the state of the neuron becomes 1 at time $t+1$, otherwise it takes -1. All the $(N-6)$ other neurons work as intermediate hidden neurons.

A preliminary experiment – searching for weights for solution

Before going further, let's see if such a recurrent neural network can anyhow challenge the task, we tried to assign a random configuration of w_{ij} and observed the action of the agent with those w_{ij}'s remaining fixed. It was quite easy to find such weight configurations just by a random search, that is, all we found when we repeated the procedure was that not a few agents fulfilled the solution. So the task is easy. However, as we easily imagine, those successful agents will take exactly the same route again when it makes another trial. This is not of our interest here.

Searching for rules to be applied during exploration

Starting with a random configuration of w_{ij}^0 we renew the configuration every time step of the agent's movement as:

$$w_{ij}^{t+1} = w_{ij}^t + (-1)^{x[i]x[j]}\eta_{ij}(1 - |w_{ij}^t|) \tag{3}$$

where η_{ij} is the degree to how much the change affects each of the connections. We might call a configuration of η_{ij} a rule in a sense. That is, we strengthen connections between correlated neurons while weaken the connections between non-correlated neurons more or less. What we should do now is how we determine the rule which will give the best performance to the agent.

Note that since every trial of the agent starts with a random weight configuration anew, we can expect a different behavior in each of multiple trials.

3 Results and Discussion

The task we have explored so far is so easy that neural network we use to control an agent need not to be trained in the environment given, but instead, a random search is enough to find such a neural network that solves the task. What we are concerning here is twofold. One is, *"Can an agent always enjoy different route?"* while the other is, *"Can an agent adapt to a slight change newly added to the environment?"*

3.1 Can an agent always enjoy different route.

In Fig. 1, we show two examples of the routes taken by the agent who was controlled by the recurrent neural network made up of 25 neurons, and not with the fixed connection weights but with those adapting the environment each time step. What we see in the figure is that the agent takes a little different route at another trial with the identical adapting rule but with a different initial random weight configuration. In this experiment agents started with 30 units of energy, and the former reached the point of 30 Hamming distance from the starting point while the latter's being 22.

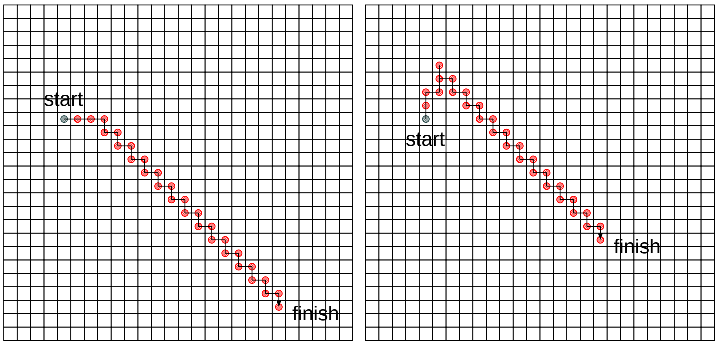

Fig. 1. Two examples of the routes taken by the agent who was controlled by a recurrent neural network not of the fixed connection but of those adapting the environment each time steps. A successful results (Left), and another trial with the identical adapting rule but a different random weight configuration at the start (Right). The former reached the point of the maximum Hamming distance 30 from the start, while the latter's being 22, both by starting with 30 units of energy.

These results, however, have not satisfied our expectation. Despite of the fact that we observed many trials of an agent who applied the identical rule but a different random weight configuration at each start, the agent chooses from just a few slightly different routes than the previous successful trial. This was not what we had expected. Although we have infinitely large number of routes which have the maximum possible Hamming distance, agents neglected most of them.

3.2 Can an agent adapt to a slight environmental change?

Then what about the next issue? We searched for a rule which enabled the agent to avoid a preset obstacle. Then we somehow enlarged the size of the obstacle to observe what would be the behavior of the same agent. An example of this experiment is shown in Fig. 2.

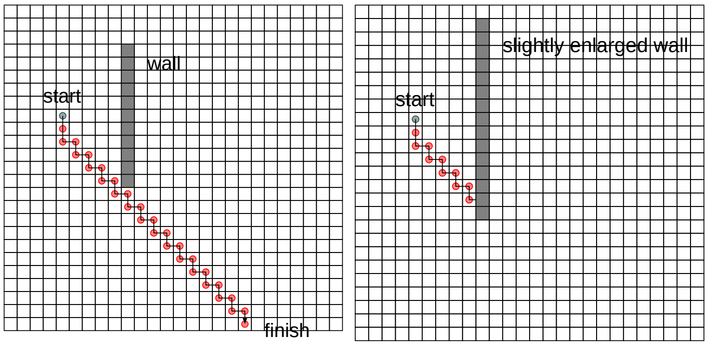

Fig. 2. A successful route avoiding an obstacle (Left), and what happened if we enlarge the size of the obstacle while applying the identical adapting rule (Right).

As shown in the figure, the agent failed to adjust to a slightly different new environment. We repeatedly observed this experiment, however mostly when the agent came across the newly found obstacle failed to avoid it.

The study is still on going, and we hope we will obtain more satisfactory results by elaborating the method. Then what about a more complicated task? Still can we expect a success?

4 A Benchmark as Challenge

The goal of the experiments above was not solving the task per se, but just to claim that artificial intelligence by neural network is not necessarily intelligent in the sense that it simply repeat the successful past action. Actually, designing an artificial intelligence in a real sense is not so easy even to solve such a simple task, if not at all. Then what about with a much more complicated task? Now we propose a more sophisticated benchmark here extending the task we have explored above.

This is an extension of so called *Jeep Problem.*[1] The original version is an exploration problem of a jeep in a desert of one-dimensional, as they call it. A jeep starts the base with fuels full in its tank. The jeep has containers to put a part of its fuels at any place of the desert to use it later during its exploration. The jeep is allowed to return to the base n times to refill its tank. The task of the jeep is to maximize the penetration into the dessert.

[1] According to Rote et al. [5], the problem first appeared as *"a camel carrying grain in a desert"* as the 52nd problem in the *"Propositions ad acuendos inventes"* (in Latin) attributed to Alcuin of York (around in B.C. 732–804).

We already know the solution. We now assume the capacity of the tank is 30 units of fuel, and $n = 3$ for the sake of simplicity of description. See, for example, the WWW page of *Wolfram MathWorld* [6] for more in detail.

> *(i) Start the base with 30 units of fuel; (ii) Go forward 10 distances, put 10 units, and then go back to the base with the remaining 10 units of fuels, and refill 30 units again; (iii) Go forward 10 spending 10 units, and get 10 units there. Again the tank is full; (iv) Go forward 6 further, spending 6 units and put there 8 units; (v) Go back to the base spending all remaining 16 units; (vi) With 30 units again, go forward 16 , spending 16 units, and get 8 there; (vii) Go forward further until spending now remaining 22 units of fuel. Thus eventually reaches the point which is 38 unit distance apart from the base.*

So, the solution is unique in this original one-dimensional version. We extended this problem to two-dimensional desert so that the jeep can enjoy different options of diversity every time when it is to try another trial. As far as we know, we have not such extension so far. The task is:

Task 2 – a challenge. *Starting from the base located at the center of an infinitely large two-dimensional grid-world, an agent must travel as far as possible from the base. The agent must spend one unit of fuel to move from one cell to the next in the grid-world. The agent starts the base with q units of fuels. The agent carries empty containers with which the agent puts part of its fuels to any cell during the exploration for a future usage. The agent is allowed to go back to the base n times to refill q units of fuels again. The goal is to reach the farthest possible distance from the base.*

5 Concluding Remarks

We have taken a speculation about how an artificial intelligence should be like, through simple experiments in which an agent controlled by a neural network explores a two-dimensional grid-world. Although the task itself and the method to solve it described in this paper might look trivial like an undergraduate student project, the essential idea is crucial if we are to claim an establishment of an artificial intelligence, say, by a neural network. However the task might be sophisticated and however performance of the agent to solve it might be elegant, such as those published so far, the behavior should not be deterministic if we are to claim the behavior is an intelligent one. Our experiment in a simple world with a simple task suggests an establishment of such an artificial intelligence will not so easy, if not at all though.

The task of the agent in this paper is to maximize the Manhattan distance to the point where the agent eventually reaches before consuming all its energy given at the start. In this experiment, we have many options to attain this goal and we expected a resultant intelligent agent can enjoy different route taking from these options at every trial. Furthermore we made an agent learn how to avoid an obstacle and then observed if the agent also would be able to adjust an additional obstacle that it had not learned before. We have not obtained satisfactory result in both of these experiments so far.

Finally we have proposed a more challenging problem as a benchmark, that is, a two-dimensional extension of Jeep Problem. We hope this evoke stimulating discussions on how an artificial intelligence should look like.

References

1. Floreano, D., and Urzelai, J. (2000) “Evolutionary robots with online self-organization and behavioral fitness.” Neural Networks 13: 431–434.
2. Willshaw, D., and Dayan, P. (1990). “Optimal plasticity from matrix memories: What goes up must come down.” Neural Computation 2: 85–93.
3. Stanley, K. O., Bryant, B. D., and Miikkulainen, R. (2003) “Evolving adaptive neural networks with and without adaptive synapses.” Proceedings of the 2003 IEEE Congress on Evolutionary Computation 4: 2557–2564
4. Stolle, M., and Precup, D. (2002) “Learning options in reinforcement learning.” Proceedings of the International Symposium on Abstraction, Reformulation and Approximation, Lecture Notes in Computer Science 2371: 212–223.
5. Rote, G., and Zhang, G. (1996) “Optimal logistics for expeditions – The jeep problem with complete refilling.” Projektbereigh: Diskrete Optimierung, Bericht Nr. 71.
6. http://mathworld.wolfram.com/JeepProblem.html.