General AI Challenge
Round 1: Results and Evaluation
Gradual Learning – Learning Like a Human

Autumn 2017

GoodAI team, 30.09.2017
info@goodai.com
www.GoodAI.com
www.general-ai-challenge.org
Table of contents

1. Introduction 3

2. Evaluation curriculum 3
3. Quantitative results 3
   3.1 Number of tasks solved 4
   3.2 Number of steps used 6
   3.3 Graduality ratio 9

4. Qualitative results 11

5. Human performance on the public curriculum 12
   5.1 Results 12

6. Lessons learned 13

7. Next steps 13

8. Appendix A: Additional test results 14
   8.1 Tests on the evaluation curriculum with relaxed conditions 14
   8.2 Further analysis of the number of steps taken by the agents 15

9. Appendix B: Part of the evaluation curriculum 16
General AI Challenge – Round 1 Evaluation

1. Introduction

The goal of Round 1 of the General AI Challenge was to design and create a gradually learning agent that will pass a curriculum of learning tasks. The participants of the challenge could choose to submit only the design (idea) or also create an implementation (agent). All submissions were evaluated by a jury for the best idea and were eligible for the qualitative prize. The submissions that included an implementation were also evaluated for best performance and competed for the quantitative prize.

We are fully aware that the challenge is difficult, so we were excited to receive 13 submissions! Eight of them included an agent and competed in both categories, quantitative and qualitative. The remaining five submissions competed for the “best idea” prize only. Among the full submissions, there were two Windows-based solutions and six Linux-based solutions.

We studied all of the submissions and tested the quantitative submissions according to rules set down in the specification document. We tested the agents on a nonpublic evaluation curriculum, which was based on the public one. To learn more about the submissions, we also tested the agents on the public training curriculum described in Appendix A of the specification document.

2. Evaluation curriculum

The evaluation curriculum was created by modifying and extending the public training curriculum. There were two types of changes. First, individual changes to the content of the tasks (this involved removing and adding certain tasks). Second, global changes designed to encourage general solutions and prevent hard-coding based on the training curriculum; for instance, a scrambler (see below).

The first four tasks of the curriculum remained the same, only the “identity mapping” task became the first since it is the simplest one (solved by repeating the input). In the next few tasks, we introduced some changes (like different separators) to prevent hard-coding. These changes were then consistently applied in the following tasks. Towards the end of the curriculum, there were more changes to the content of the tasks and some completely new tasks added.

To further encourage general solutions we employed a scrambler on top of the other modifications. The scrambler remapped all used characters to a broader subset of ASCII printable characters. For example, a space could be represented by the letter F and semicolon could be represented by the _ (underscore) symbol. This mapping was fixed during the whole run of the environment.

The first 10 tasks of the evaluation curriculum are described in Appendix B.

3. Quantitative results

First we tested how far the agents could progress through the evaluation curriculum. We performed several variants of this test. Then we did additional tests for the most capable agents. We were interested in how many timesteps the agents needed to solve each task and if we could detect the use of gradual learning.

Two of the submitted agents didn’t work. We managed to fix one of them so we were left with seven agents to test. All of the solutions opted for CPU-bound evaluation HW.
3.1 Number of tasks solved

Unfortunately, none of the submitted solutions approached one quarter of the tasks in the evaluation curriculum.

The scrambler turned out to be a significant obstacle for most agents, so we also tested them with the scrambler disabled.

And, out of curiosity, we also tested the agents on the public training curriculum. Interestingly, one agent solved all of it, and two of the agents did not solve even a single task.
The best results were achieved by Agent A which handled the scrambler very well. Nevertheless, this is only a basic capability assessment; it does not tell, for example, whether the agent makes any use of gradual learning.

Agent B solved the highest number of tasks (nine) with the scrambler disabled. It was able to handle the scrambler at least partially (four tasks). Agent C was able to handle the scrambler (four tasks) and performed relatively well on unscrambled tasks (seven). It was the only agent able to solve all tasks in the public training curriculum. Agent D looked promising on unscrambled tasks (eight) but struggled with the scrambler.

Agent E solved one evaluation task (both with and without scrambler) but no training task. Agent F solved one training task and no evaluation tasks. Agent G did not solve any task in the standard setting. It was able to solve 20 tasks of the training curriculum starting from task 5.3. However, it was not able to solve any of three randomly chosen tasks from the part of the evaluation curriculum that corresponded to the 20 tasks of the training curriculum.

We ran additional tests on part of the evaluation curriculum with relaxed conditions for advancing the agent to the next task. You can find the results in Appendix A.
The following table summarizes the results along with additional information such as submission size and target operating system.

<table>
<thead>
<tr>
<th>Name</th>
<th># Solved tasks(^1) (with scrambler)</th>
<th># Solved tasks (no scrambler)</th>
<th># Solved tasks (training)</th>
<th>Submission size(^2) (compressed) [MB]</th>
<th>Target OS</th>
</tr>
</thead>
<tbody>
<tr>
<td>Agent A</td>
<td>7</td>
<td>7</td>
<td>26</td>
<td>80.7</td>
<td>Linux</td>
</tr>
<tr>
<td>Agent B</td>
<td>4</td>
<td>9</td>
<td>24</td>
<td>79.6</td>
<td>Linux</td>
</tr>
<tr>
<td>Agent C</td>
<td>4</td>
<td>7</td>
<td>all <strong>46</strong></td>
<td>0.1</td>
<td>Windows</td>
</tr>
<tr>
<td>Agent D</td>
<td>2</td>
<td>8</td>
<td>6</td>
<td>430.1</td>
<td>Linux</td>
</tr>
<tr>
<td>Agent E</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>80.6</td>
<td>Linux</td>
</tr>
<tr>
<td>Agent F</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>302.0</td>
<td>Linux</td>
</tr>
<tr>
<td>Agent G</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0.1</td>
<td>Windows</td>
</tr>
</tbody>
</table>

Note \(^1\): Winners of quantitative prize were supposed to solve the whole evaluation curriculum (with scrambler) and demonstrate gradual learning ability on additional tests. Since no agent solved all 43 tasks, we did not award the quantitative prize.

Note \(^2\): Size of the learner compressed by zip (or gzip). The two Windows-based submissions (C and G) are zipped .NET binaries which are very small. For Linux-based submission these are sizes of gzipped Docker images (empty image based on Python-slim has around 80 MB).

3.2 Number of steps used

There was a clear difference between the four most capable agents (A to D) and the rest of them. We wanted to compare performance of the agents on individual tasks which they were able to solve. Therefore, we selected only the first four agents because there are too few tasks to test on for the other agents. In addition, we disabled the scrambler for these tests to be able to test on more solved tasks. This and the following sections contain test results that evaluate only the four selected agents.

We wanted to measure number of timesteps used to solve individual tasks. We were not sure how much variation would be in the results, so we ran the test multiple times to find out.

We ran the agents five times on the first seven tasks of the evaluation curriculum (these tasks could be solved by all of them). It was a continuous run through the curriculum. We estimated mean number of timesteps needed by each agent for each task with 90% confidence intervals.

The results were split into three charts to allow for different ranges of the Y axis. The top of the bars shows the upper bound of the confidence intervals, the boundary between the yellow section and the blue section of the bars shows the average value (from five continuous runs), and the boundary between the blue section and the gray section shows the lower bound.
The following chart shows median number of steps on each task (based on five continuous runs). Please note that the variation presented in the previous charts is not shown here. You can see that the total count is dominated by the number of steps used on task 3, so we moved it to the end and selected light gray for it to be able to compare the results both with and without the task.
Agents A and B typically use relatively small number of steps with small variation between the runs (with the exception of Agent B on task 3). Agent C performs slightly worse than A and B on most tasks and much worse on task 5. It also exhibits large variation on tasks 5 to 7. Agent D uses the greatest number of steps in general with one exception of task 5 where it is just behind the leaders. Its results also have the largest variation. More detailed analysis of used timesteps can be found in Appendix A.

We would like to point out that all of the agents stopped progressing on the evaluation curriculum relatively quickly (within minutes or tens of minutes). Focusing on a good incremental acquisition of skills therefore currently seems more important than minimizing the number of steps the agent needs for learning.

The analysis of the number of steps makes greater sense when used for assessing the “graduality” of the agents.

3.3 Graduality ratio

One of the main goals of the challenge was to get working examples of agents that can acquire skills in a gradual manner and use learned skills to learn new skills. Therefore, we tested the solutions for use of gradual learning by comparing the performance (the number of timesteps used) in two scenarios:

1. Standard curriculum walkthrough: The agent retains knowledge from previous tasks when solving the next task. (It is referred to as “continuous run” in the text above.)
2. Tabula-rasa agent: the agent was wiped clean (restarted) before solving each of the tasks. Thus it could not reuse any past knowledge. (We also refer to it “individual run”.)

We combined the resulting two numbers into a **graduality ratio** calculated as the number of timesteps used in the continuous run, divided by the number of steps used in the individual run on the same task. But we did a bit more than just that because we wanted to estimate 90% confidence intervals of the graduality ratios.

First, we selected three tasks with the lowest uncertainty (the narrowest confidence interval) in the number of timesteps used. We observed relatively low variation in tasks 5 to 7 (respective tasks from the evaluation curriculum are denoted as 5.1.1, 5.1.2, and 5.2.1 in the appendix B), which are also the more interesting tasks. We wish we had more tasks to compare but the agents didn’t get that far in the curriculum. We performed five continuous runs and five individual runs on these three tasks for each agent.

We used all combinations of the two sets of measurements to get 25 graduality ratios (assuming the runs are independent). We calculated median, 5th percentile, and 95th percentile to estimate the graduality ratio and 90% confidence intervals.

The following chart shows the results. Note that the highest column (Agent D, T6) is truncated, the upper bound was 13.0.
None of the agents showed graduality ratio consistently above or below 1. The confidence intervals are quite narrow for agents A and B and very wide for Agent C on all tasks and for Agent D on two of three tasks. The interval for Agent D, task 5 is the only one with the whole confidence interval below 1.0.

In the chart above, median graduality ratio is represented by the boundary of the yellow and violet sections of the bars. It’s a bit hard to read, so the next chart zooms-in on it and adds labels with the actual median graduality ratio (values below 1 are good, lower is better).
The one confidence interval (Agent D, T5) that is whole below 1.0 is not sufficient for a definitive conclusion. With 90% confidence intervals, we can still expect one of ten intervals to be below the 1.0 mark even if the agent would not employ any gradual learning.

In conclusion, we did not prove or disprove use of gradual learning in any of the agents.

4. Qualitative results

The Qualitative prize is to be awarded for the idea, concept or design that shows the most promise for scalable gradual learning.

All 13 submissions were eligible for the qualitative prize. First, to avoid bias, part of our team analyzed the white papers, without looking at agents' performance. The GoodAI team shortlisted six of the submissions. Authors of four best scoring papers happened to submit agents, which also performed better than the rest in objective tests.

The shortlisted solutions were passed on to the final jury made up of:

- Pavel Kordik (Czech Technical University)
- Alison Lowndes (NVIDIA)
- Tomas Mikolov (Facebook Research)
- Roman Yampolskiy (University of Louisville)
- Members of the GoodAI team

In addition to all the information that the participants made available in their solution, including their papers, the jury took into account the agents' performance on the evaluation curriculum, and on additional tests for gradual learning (see summary of evaluation for details of the tests). Each member of the jury voted for three best solutions (by awarding points from 0 to 3).

Four solutions with highest number of points emerged. The jury concluded that the presented solutions are closely comparable, and more work is required to demonstrate that the authors are on track to robust gradual learning mechanisms.

The jury selected no winner, but to encourage further work on gradual learning and to reward the participants for their considerable efforts, they decided to split the 2nd prize ($7000) among the four finalists.

The recipients of the joint award are (in alphabetical order):

- Dan Barry (Agent D)
- Andrés del Campo Novales (Agent C)
- Andreas Ipp (Agent A)
- Susumu Katayama (Agent B)

Although all the judges agreed that the solutions are closely comparable, and each of them have their own notable aspects, the solution by Andreas Ipp was the one that received most points. On top of the split monetary prize, the jury awards Andreas Ipp with the special GPU prize from our Challenge partner NVIDIA – a GEFORCE GTX 1080 graphics card.
5. Human performance on the public curriculum

Before the challenge started, we wanted to make sure the challenge tasks are actually solvable by a human solver. That is why we tested the public training curriculum on a human solver.

We asked a colleague from the GoodAI team who was not involved in the design of the challenge tasks, and thus had no knowledge about them, to be the test subject. He then tried to solve the whole curriculum, one task after another, just as an artificial agent would.

We measured the number of instances the human solver needed for solving the tasks. We did not measure the number of timesteps, as rigorous time measurements were not the target of the tests.

5.1 Results

There were 46 tasks. The human solver took approximately 13.5 hours to solve them. The statistics of his performance are shown in the charts below.

In contrast to an artificial agent, the human solver was required to collect fewer successful instances in a row. This was mainly due to the fact that the human solver was unlikely to solve any of the tasks by brute force or guessing. Moreover, the human solver would tire quickly if he were required to keep solving a task for e.g. an hour (to collect the necessary successes) when the solution was obvious to him after first few minutes.

We would like to stress out that the intent of the human performance testing was to verify that the solutions of the tasks could be found, rather than to collect data for 1:1 comparison with AI performance.
6. Lessons learned

The following are several takeaways from the current round which we intend to use as inputs for improving the future round(s).

- Since nobody was able to solve the Gradual Learning round, we are going to dedicate one of the upcoming rounds of the General AI Challenge to solving gradual learning.
- We want to keep the participants excited and motivated during the whole run of the challenge round. We are considering a leaderboard with automated evaluation for this purpose.
- Iterative evaluation: some submissions took the whole 24 hours of the evaluation CPU time for trying to solve the first task. This was essentially a waste of resources. We are considering including stages in the challenge evaluation so that only agents that reach a threshold task within the given time limit will be allowed to run for the full 24 hours.
- Simplified interface: we found out that the 256-value UTF-8 interface caused too many (unnecessary) technical difficulties for the participants. We will therefore limit the data on the interface to printable ASCII characters only.
- More time for solving a task instance: some of the earlier tasks switched the instances earlier than participants of the challenge would like. Since learning from limited data was not the point of the challenge (although faster learners are preferred by the evaluation), we'll let the instances run for longer.
- We are considering performing another round of human performance tests, this time with scrambler to remove many of the human biases from the tasks.
- In future rounds of the Challenge that will involve programming, in order to be as objective and transparent as possible, we will go with quantitative evaluation only (objective tests). Evaluating for the qualitative (aka “best idea”) prize was a great experience for us and the wider jury, but at the same time, to judge technical proposals based on relatively limited information available proved to be more challenging than we expected.

7. Next steps

In 2018, we plan to launch another round of the General AI Challenge focused on gradual learning. For this reason, we are publishing only those tasks from the evaluation curriculum, that the AI agents were able to solve in Round 1. The rest of the tasks will stay secret and will be reused in the upcoming gradual learning round – part II.

In the meantime, in November 2017, we will launch Round 2 of the Challenge: AI Race Avoidance. Since the goal of the Challenge is to help solve all pieces of the puzzle that will lead to beneficial future with general AI, we are interested in societal implications of AI as much as in technical milestones.

We invite you to keep following the General AI Challenge and participate in future rounds!
8. Appendix A: Additional test results

8.1 Tests on the evaluation curriculum with relaxed conditions

We wanted to make sure that the test results do not depend too much on the particular setting of the evaluation environment. Therefore we ran additional tests on part of the evaluation curriculum that started from the fifth task. We also relaxed conditions for marking the tasks successfully solved and advancing the agent to the next task.

There were the following changes compared to the previous evaluation tests:

1. Skipped first four tasks of the evaluation curriculum which arguably have a bit different nature.
2. One task instance is presented almost indefinitely (1000× more timesteps than before).
3. A single solved instance is enough to advance to the next task. (There is much greater chance of solving a task just by accident but it still remains relatively small.)
4. Number of required consecutive rewards lowered to 10 (same as it was in the training curriculum).
5. The scrambler was disabled.

The chart below shows the results (the “Five-plus” bars). For comparison, the chart shows the baseline for this test, the number of tasks solved beyond the task number 4 (with no scrambler).

We were glad to see that the results were very similar to the ones with more strict conditions suggesting the results are not overly sensitive to these conditions.

Only Agent C benefited from these relaxed conditions. It solved two more tasks compared to the baseline. None of the agents benefited from skipping the first four tasks.
8.2 Further analysis of the number of steps taken by the agents

The following chart shows the number of steps relative to average number of steps used by the best-performing four agents. It allows to compare agents with the same weight put on each task. It also shows median of these seven relative measures (each of which is based on median of 5 continuous runs).

Some observations from the results:

- Agents A and B typically use relatively small number of steps with small variation between the runs, most notably on tasks 5, 6, 7.
- Agent C performs well on tasks 1 to 4, it is the worst among the finalists on task 5, and somewhere in the middle on tasks 6 and 7. The results show large variation on tasks 5 to 7.
- Agent D uses the greatest number of steps in general with one exception of task 5 where it is just behind the leaders. It is comparable to Agent B on task 3 and to Agent C on task 7. In general it experiences the largest variation.
- Task 3 (searching for a mapping) falls out of the general pattern. Agent B uses the most steps on this task, slightly more than Agent D. Agents A and C are the leaders in this task.
- It is difficult to draw definitive conclusions with so wide confidence intervals. The lower bounds of the intervals show comparable performance between all four finalists on four of the tasks (1, 4, 6, and 7).
9. Appendix B: Part of the evaluation curriculum

In 2018, we plan to launch another round of the General AI Challenge focused on gradual learning. For this reason, we are publishing only those tasks from the evaluation curriculum, that the AI agents were able to reach in Round 1. The rest of the tasks will stay secret and will be reused in the upcoming gradual learning round – part II. The evaluation tasks that we publish here are, on the other hand, likely to be replaced with different tasks in the next gradual learning round.

None of the agents solved the last task listed here, 5.3.2, but this task was at least seen by one of the agents.

1-4 Introductory tasks

The introductory tasks are the same as in the public challenge, with the exception that a different subset of the first 128 ASCII characters is presented to the agent. Also, the fourth task from the public curriculum (identity mapping) was moved to become the first task of the evaluation curriculum.

Note that, because of this ordering change, in the test results above, task 1 corresponds to the 4th task of the public training curriculum (described in the section “A.2.4 Learning to copy input to output” of Appendix A in the specification of the first round). Tasks 2, 3, 4 then correspond to the 1st, 2nd, and 3rd task of the public training curriculum.

5 Learning that feedback is delivered on the input, not just in the reward signal

The following five tasks were either solved or at least seen by the evaluated agents. None of the agents solved task 5.3.2, so it is the last task we are making public from the evaluation curriculum.

In the tests results above, tasks 5.1.1, 5.1.2, and 5.2.1 are referred to as tasks 5, 6, 7 respectively.

5.1.1 One byte for input, output and feedback (restricted alphabet)

This task is the same as in the public rounds. The only exception is that it uses a different subset of the alphabet because of the random mapping that is applied on the environment.

```
Input : 0917091709091
Output: 53248 7 1 9
Reward: ------ + - + -
```

5.1.2 One byte for input, output and feedback (whole Alphabet used)

Same as the previous task, with the exception that any character from the Alphabet may appear in the input. A single instance, however, will draw from a limited subset of characters (10 for input, 10 for feedback).

```
Input : AfG7*?Af*?Af*?
Output: f 7 f ?
Reward: --- + - + +
```
5.2.1 Feedback/next input separator (1)

Same as “one byte for input, output and feedback”, but the **feedback and the next input** are separated by a special symbol ‘;’. In the following example, ‘A’ represents input and ‘f’ represents feedback. ‘G’ then represents next input.

<table>
<thead>
<tr>
<th>Input</th>
<th>Af;G7;<em>?;Af;</em>?;Af;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>2 8 f * f</td>
</tr>
<tr>
<td>Reward</td>
<td>- - - + - +</td>
</tr>
</tbody>
</table>

5.2.2 Feedback/next input separator (2)

Same as previous task, but the **feedback and the next input** are separated by two special symbols ‘;’.

<table>
<thead>
<tr>
<th>Input</th>
<th>00;;11;;00;;11;;00;;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>2 8 1 1 0</td>
</tr>
<tr>
<td>Reward</td>
<td>- - - + - +</td>
</tr>
</tbody>
</table>

5.3.1 Input/feedback separator (1)

Same as “feedback/next input separator”, but the **input and feedback** (and output) are separated by a special symbol ‘:’.

<table>
<thead>
<tr>
<th>Input</th>
<th>0:0;;1:1;;0:0;;1:1;;0:0;;0:0;;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>2 8 1 1 0</td>
</tr>
<tr>
<td>Reward</td>
<td>- - - + - +</td>
</tr>
</tbody>
</table>

5.3.2 Input/feedback separator (2)

Same as previous task, but the **input and feedback** (and output) are separated by two special symbols ‘:’.

<table>
<thead>
<tr>
<th>Input</th>
<th>0::0;;1::1;;0::0;;1::1;;0::0;;0::0;;</th>
</tr>
</thead>
<tbody>
<tr>
<td>Output</td>
<td>2 8 1 1 0</td>
</tr>
<tr>
<td>Reward</td>
<td>- - - + - +</td>
</tr>
</tbody>
</table>