Carnegie Mellon University Africa
Certificate I: Understanding AI and Machine Learning in Africa

Course AIML01: Artificial Intelligence – Past, Present, and Future

Module 4: Future Challenges
Lecture 2: Self-programming and Self-learning Systems

Welcome to Lecture 2 of Module 4 in which we will study the challenge of creating computer systems that can program themselves and learn by themselves.

In this lecture, we will introduce the quest for self-programming and self-learning computer systems. We will look at self-programming based on deep learning and natural language processing (NLP). We will explain why self-learning is important in developmental robotics. We will also look at the way that reinforcement learning and self-supervised learning are involved in self-learning. We will finish up by summarizing what we have covered and identifying the articles that you should read to consolidate what you have learned.

We have three learning objectives, so that, after studying the material covered in this lecture, you should be able to do the following.

1. Explain the difference between a self-programming system and a self-learning system.

2. Provide examples of self-programming and self-learning systems.

3. Identify some of the techniques that are employed by self-programming and self-learning systems.

Slide 1 Welcome to Lecture 2 of Module 4 in which we will study the challenge of creating computer systems that can program themselves and learn by themselves.

Slide 2 The quest for the automatic generation of computer programs,
also known as program synthesis or self-programming machines,

has been one of the main challenges of AI since the outset.

Since the first symbolic (GOFAI) approaches to AI,

such as Newell's and Simon's General Problem Solver, or GPS,

the goal was to develop AI systems that can use general-purpose knowledge

to generate new text,
to solve mathematical and practical problems,
and to create new computer programs.

GPS is considered the first useful AI program. It was intended to solve many different problems, rather than one specific problem, using the same reasoning mechanism for each problem.

It eventually evolved into the Soar cognitive architecture.

Slide 3 More recently, with the advent of machine learning approaches, AI has started to put emphasis on self-learning systems that can learn with no or minimal human supervision.

We say more about this approach later in this lecture.

Slide 4 This self-programming system challenge has recently received a significant boost through deep learning methods for NLP.

For example, DeepCode is a code generator that uses a neural network to predict the properties of a program that can produce the outputs given specific inputs.

This neural network is used to augment search-based techniques to solve the inductive program synthesis problem for simple types of programs typical of programming competitions.

Slide 5 SketchAdapt is a system that learns, without direct supervision, when to rely dynamically on pattern recognition and when to perform symbolic search for explicit reasoning (Hewitt and Tenenbaum 2020).

This mimics the human ability to dynamically incorporate pattern recognition from examples and reasoning to solve programming problems from examples or a specification expressed in natural language.

Slide 6 GPT-3, the third generation General Pre-trained Transformer (Brown et al. 2020) from OpenAI, has potential for automatic program synthesis

We met GPT-3 briefly in Lecture AIML01-02-02.

It's an advanced large-scale neural network

It was trained on trillions of words and has some 175 billion machine learning parameters,

GPT-3 can generate new text without the need of further training or task-specific fine tuning of its parameters.

Slide 7 This has been evaluated with few-shot demonstrations, i.e., via text interaction with the model giving the task description and one or few examples.

It can also perform several tasks that require on-the-fly reasoning, such as unscrambling words, using a novel word in a sentence, or performing arithmetic.

Slide 8 As we noted in Lecture AIML-02-02, GPT-3 can produce samples of news articles which human evaluators have difficulty distinguishing from articles written by humans.

In addition, GPT-3 has been used for generating programs, such as for the code to create the Google home page.

Slide 9 Regarding the challenge of creating a self-learning machine, the first attempts to design AI systems and robots that autonomously learn without supervision by humans have recently been realised in developmental robotics

This area of robotics takes inspiration from child development to design robots that go through stages of developmental for the incremental acquisition of sensorimotor and cognitive skills (Cangelosi & Schlesinger 2015).

Slide 10 Developmental robots use intrinsic motivation mechanisms

Implemented with reinforcement learning, for example

to allow them to initiate and manage self-learning via curiosity-driven mechanisms for open-ended, cumulative acquisition of skills.

Slide 11 Another example of self-learning AI is the AlphaGo Zero system which we met in Lecture AIML01-03-04

in which artificial agents play the game Go against each other,

bootstrapping their final learning capabilities.

This allowed the AlphaGo Zero system to acquire skills that far outperformed the skills of the best human players and previous versions of AlphaGo.

Slide 12 Going on to achieve 60 straight wins in time-control games against top international players in 2017.

Slide 13 Subsequently, in AlphaGo Zero, even better performance was achieved based purely on reinforcement learning without any prior supervised training.

Apart from its formidable performance, what is significant about AlphaGo is that it uncovered several innovative strategies that greatly surprised expert players, demonstrating the potential for AI to augment human abilities and exceed human performance.

Let's listen again to a five-minute video describing the development of the several incarnations of AlphaGo.

Slide 14 Returning to Yann LeCun’s cake analogy for machine learning which we met in Lecture AIML01-02-03,

LeCun revised his cake analogy in 2019, extending the concept of unsupervised learning, using the terms **self-supervised** and **predictive learning** to describe the situation where the data provides the supervision.

Slide 15 Here, the agent generates its own labels and teaching input, e.g., with autoencoders or Word2vec unsupervised learning methods, so that the system predicts the output from partial, incomplete, or self-generated input.

This suggests that AI will increasingly be based on predictive, self-learning methodologies.

To summarize:

1. Self-programming systems focus on automatic generation of computer programs.
2. Self-programming systems also focus on using general purpose knowledge and reasoning to solve many different problems.
3. Today, self-programming systems are levering recent advances in natural language processing (NLP).
4. Self-learning systems can learn with little or no supervision from humans.
5. Artificial curiosity drives self-learning in developmental robotics.
6. Self-learning systems can also use reinforcement and self-supervised learning.

Here is some recommended reading.

Peng T. (2019) LeCun Cake Analogy 2.0. https://medium.com/syncedreview/yann-lecun-cake-analogy-2-0-a361da560dae

Here are the references cited to support the main points in what we covered today.

Balog, M., Gaunt, A. L., Brockschmidt, M., Nowozin, S., & Tarlow, D. (2016). DeepCoder: Learning to write programs. arXiv preprint arXiv:1611.01989. https://www.researchgate.net/publication/309738607_DeepCoder_Learning_to_Write_Programs

Brown TB, Mann B, Ryder N, Subbiah M, Kaplan J, Dhariwal P, Neelakantan A, Shyam P, Sastry G, Askell A, Agarwal S, Herbert-Voss A, Krueger G, Henighan T, Child R, Ramesh A, Ziegler DM, Wu J, Winter C, Hesse C, Chen M, Sigler E, Litwin M, Gray S, Chess B, Clark J, Berner C, McCandlish S, Radford A, Sutskever I, Amodei D (2020) Language models are few-shot learners. arXiv:2005.14165. https://arxiv.org/abs/2005.14165

Bussler, F. (2020) Will the latest AI kill coding? https://towardsdatascience.com/will-gpt-3-kill-coding-630e4518c04d

Cangelosi, A., & Schlesinger, M. (2015). Developmental Robotics: From Babies to Robots. MIT press. https://mitpress.mit.edu/books/developmental-robotics

Heaven, W. D. (2020). OpenAI’s new language generator GPT-3 is shockingly good—and completely mindless. MIT Technology Review. https://www.technologyreview.com/2020/07/20/1005454/openai-machine-learning-language-generator-gpt-3-nlp/

Nye, M., Hewitt, L., Tenenbaum, J., Solar-Lezama, A. (2019). Learning to Infer Program Sketches, Proceedings of the 36th International Conference on Machine Learning, PMLR 97:4861-4870, 2019. http://proceedings.mlr.press/v97/nye19a.html

Newell, A., Shaw, J.C., Simon, H.A. (1959). Report on a general problem-solving program. Proceedings of the International Conference on Information Processing. pp. 256–264. http://bitsavers.informatik.uni-stuttgart.de/pdf/rand/ipl/P-1584_Report_On_A_General_Problem-Solving_Program_Feb59.pdf

Silver, D., Schrittwieser, J., Simonyan, K., Antonoglou, I., Huang, A., Guez, A., Hubert, T., Baker, L., Lai, M., Bolton, A., et al. (2017). Mastering the game of Go without human knowledge. Nature, 550(7676):354. https://www.researchgate.net/publication/320473480_Mastering_the_game_of_Go_without_human_knowledge