Guest Post (Part I): Demystifying Deep Reinforcement Learning

2015-12-22

#neon

Author Bio Image

Tambet Matiisen

Two years ago, a small company in London called DeepMind uploaded their pioneering paper "Playing Atari with Deep Reinforcement Learning" to Arxiv. In this paper they demonstrated how a computer learned to play Atari 2600 video games by observing just the screen pixels and receiving a reward when the game score increased. The result was remarkable, because the games and the goals in every game were very different and designed to be challenging for humans. The same model architecture, without any change, was used to learn seven different games, and in three of them the algorithm performed even better than a human!

It has been hailed since then as the first step towards general artificial intelligence – an AI that can survive in a variety of environments, instead of being confined to strict realms such as playing chess. No wonder DeepMind was immediately bought by Google and has been on the forefront of deep learning research ever since. In February 2015 their paper "Human-level control through deep reinforcement learning" was featured on the cover of Nature, one of the most prestigious journals in science. In this paper they applied the same model to 49 different games and achieved superhuman performance in half of them.

Still, while deep models for supervised and unsupervised learning have seen widespread adoption in the community, deep reinforcement learning has remained a bit of a mystery. In this blog post I will be trying to demystify this technique and understand the rationale behind it. The intended audience is someone who already has background in machine learning and possibly in neural networks, but hasn't had time to delve into reinforcement learning yet.

The roadmap ahead:

1. **What are the main challenges in reinforcement learning?** We will cover the credit assignment problem and the exploration-exploitation dilemma here.
2. **How to formalize reinforcement learning in mathematical terms?** We will define Markov Decision Process and use it for reasoning about reinforcement learning.
3. **How do we form long-term strategies?** We define "discounted future reward", that forms the main basis for the algorithms in the next sections.
4. **How can we estimate or approximate the future reward?** Simple table-based Q-learning algorithm is defined and explained here.
5. **What if our state space is too big?** Here we see how Q-table can be replaced with a (deep) neural network.
6. **What do we need to make it actually work?** Experience replay technique will be discussed here, that stabilizes the learning with neural networks.
7. **Are we done yet?** Finally we will consider some simple solutions to the exploration-exploitation problem.

Reinforcement Learning

Consider the game Breakout. In this game you control a paddle at the bottom of the screen and have to bounce the ball back to clear all the bricks in the upper half of the screen. Each time you hit a brick, it disappears and your score increases – you get a reward.

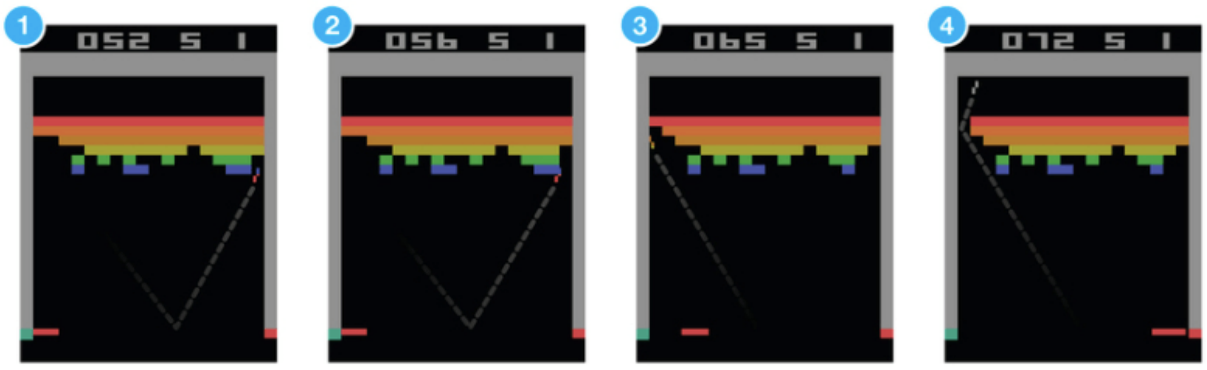
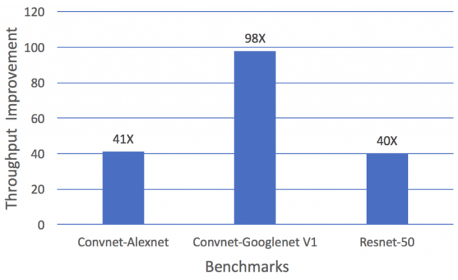


Figure 1: Atari Breakout game. Image credit: DeepMind.

Suppose you want to teach a neural network to play this game. Input to your network would be screen images, and output would be three actions: left, right or fire (to launch the ball). It would make sense to treat it as a classification problem – for each game screen you have to decide, whether you should move left, right or press fire. Sounds straightforward? Sure, but then you need training examples, and a lots of them. Of course you could go and record game sessions using expert players, but that's not really how we learn. We don't need somebody to tell us a million times which move to choose at each screen. We just need occasional feedback that we did the right thing and can then figure out everything else ourselves.

Related Blog Posts

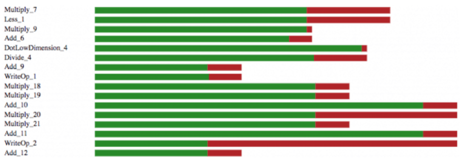


neon™ 2.0: Optimized for Intel® Architectures

neon™ is a deep learning framework created by Nervana Systems with industry leading performance on GPUs thanks to its custom assembly kernels and optimized algorithms. After Nervana joined Intel, we have been working together to bring superior performance to CPU platforms as well. Today, after the result of a great collaboration between the teams, we...

Read more ›

#neon

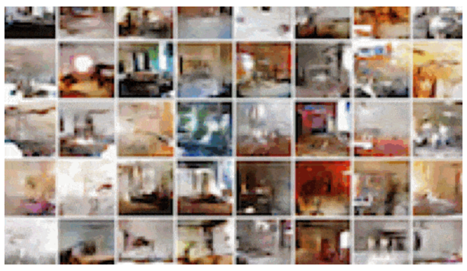


Intel® Nervana™ Graph Beta

We are building the Intel Nervana Graph project to be the LLVM for deep learning, and today we are excited to announce a beta release of our work we previously announced in a technical preview. We see the Intel Nervana Graph project as the beginning of an ecosystem of optimization passes, hardware backends and frontend...

Read more ›

#Intel Nervana Graph #neon



Training Generative Adversarial Networks in Flexpoint

Training Generative Adversarial Networks in Flexpoint With the recent flood of breakthrough products using deep learning for image classification, speech recognition and text understanding, it's easy to think deep learning is just about supervised learning. But supervised learning requires labels, which most of the world's data does not have. Instead, unsupervised learning, extracting insights from...

Read more ›

#neon

Load More Posts

Inspired by this Blog?

Keep tabs on all the latest news with our monthly newsletter.

SUBSCRIBE