Introductory remarks for RL seminar

6/9/19

The aim of this series of seminars is to understand AlphaGo, which is an important advance not only in the field of artificial intelligence but in science more broadly. To understand AlphaGo we need to understand its three components:

- Monte-Carlo tree search (Lectures 7, 8)
- Deep learning (Lectures 0, 4, 5, 6)
- Reinforcement learning (Lectures 2, 6, 8, 9)

Why is AlphaGo important?

That depends who you are. It is not clear AlphaGo is important in pure mathematics, except perhaps for logic (see the interview with Szegedy linked on the seminar webpage). However, for applied mathematics, physics and other natural sciences it seems obviously important, for the same reason that calculus and the study of polynomial approximation has been important for the last several centuries.

Theorem (Stone-Weierstrass) If $X \subseteq \mathbb{R}^n$ is compact, then polynomial functions are dense in $Cts(X, \mathbb{R})$. — (compact-open topology, i.e. sup-metric)

- Banach + Picard: solve ODEs by iteration of a contraction mapping to find a fixed point, gives polynomial approximations.
- This is actually useful in many cases!

But there are functions in nature for which this approach completely fails:

Example Consider the state space of the game Go on a 19×19 board:

$$S = \{\text{empty, white, black}\}^{\{1,\dots,19\}\times\{1,\dots,19\}} \qquad |S| = 3^{361}$$

$$S_{\text{legal}} \subseteq S \qquad |S_{\text{legal}}| \sim 2.082 \times 10^{170}$$

Let $V^*: S_{\text{legal}} \to \mathbb{R}$ denote the optimal value function, given rewards +1 for win, −1 for a loss, and a discount factor $\gamma < 1$ (i.e. the expected value of time discounted reward for a "perfect" player). (that this makes sense will be justified in Lecture 2), in a game of Go.

To find V^* by fixed point iteration $V_0, V_1, \dots$ would need to store

$$\begin{aligned} 2\times 10^{170} \times 16 \text{ bits} &= 4\times 10^{170} \text{ bytes} \\ &= 4\times 10^{152}\times 10^{18} \text{ bytes} \\ &= 4\times 10^{152} \text{ exabytes} \end{aligned}$$

(16 bit real number)

In 2011 world disk storage amounted to 295 exabytes. So maybe you don't want to do that.

We can extend V^* to a continuous function (say by linear interpolation)

$$[0,1]^{3\cdot 361} \cong \left([0,1]^3\right)^{\{1,\dots,19\}\times\{1,\dots,19\}} \longrightarrow \mathbb{R}$$

$[0,1]^3$: $(1,0,0)$ blank, $(0,1,0)$ white, $(0,0,1)$ black

Although this optimal value function exists as a mathematical object, it was far from clear that there existed any reasonable algorithm for producing a sequence of simple functions $(f_n)_{n\geq 0}$ converging to V^*

$$f_n \longrightarrow V^*$$

what kind of function is this — what kind of process is this?

The work of the research group at DeepMind led by David Silver shows that deep neural networks and stochastic gradient descent + RL + Monte-Carlo tree search + self-play suffices.

Theorem If $X \subseteq \mathbb{R}^n$ is compact then feedforward ReLU neural networks of width $n+1$ are dense in $Cts(X, \mathbb{R})$.