Fundamental Tools behind the Recent Breakthroughs in Artificial Intelligence

Keith L. Downing

The Norwegian University of Science and Technology (NTNU)
Trondheim, Norway
keithd@idi.ntnu.no

November 5, 2018

...with some Technical Details

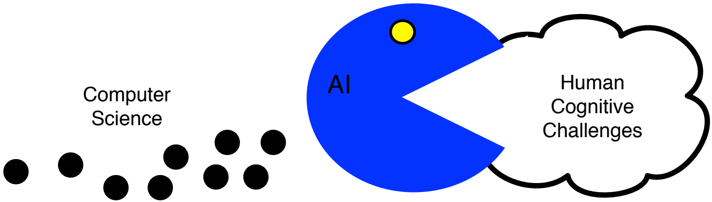
$$\frac{\partial Z}{\partial V} = \frac{\partial f(S)}{\partial V} = z_1(1-z_1) \times \frac{\partial (Y \bullet W)}{\partial V}$$

$$= z_1(1-z_1) \times \begin{pmatrix} x_1 y_1(1-y_1)w_{11} & x_1 y_2(1-y_2)w_{21} \\ x_2 y_1(1-y_1)w_{11} & x_2 y_2(1-y_2)w_{21} \\ x_3 y_1(1-y_1)w_{11} & x_3 y_2(1-y_2)w_{21} \end{pmatrix}$$

$$= \begin{pmatrix} z_1(1-z_1)x_1 y_1(1-y_1)w_{11} & z_1(1-z_1)x_1 y_2(1-y_2)w_{21} \\ z_1(1-z_1)x_2 y_1(1-y_1)w_{11} & z_1(1-z_1)x_2 y_2(1-y_2)w_{21} \\ z_1(1-z_1)x_3 y_1(1-y_1)w_{11} & z_1(1-z_1)x_3 y_2(1-y_2)w_{21} \end{pmatrix}$$

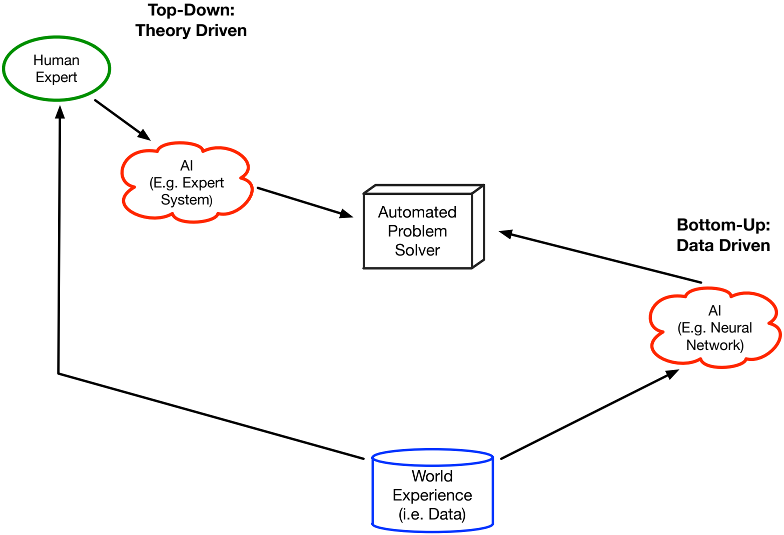
$$= \begin{pmatrix} \frac{\partial Z}{\partial v_{11}} & \frac{\partial Z}{\partial v_{12}} \\ \frac{\partial Z}{\partial v_{21}} & \frac{\partial Z}{\partial v_{22}} \\ \frac{\partial Z}{\partial v_{31}} & \frac{\partial Z}{\partial v_{32}} \end{pmatrix}$$

Artificial Intelligence



- *AI is the study of how to make computers do things at which, at the moment, people are better* ... Elaine Rich
- Cognitive processes become demystified when shown to be algorithmic.
- Many agree that what makes AI systems special are automated a) **learning**, and b) **creativity**

Top-Down -vs- Bottom-Up AI



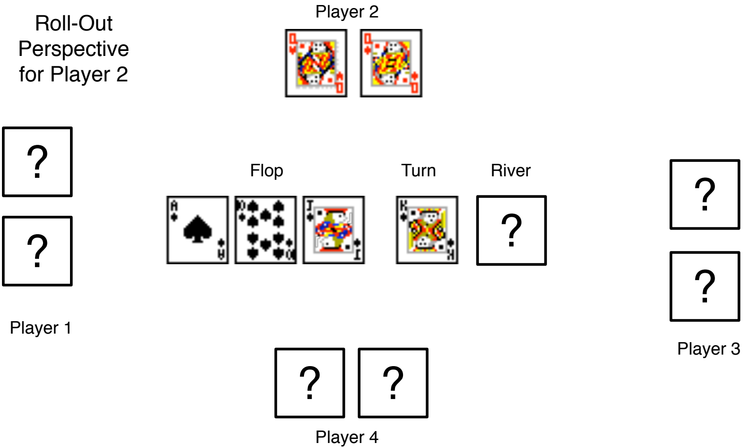
Theory-Driven Poker: Expert Systems

1. Play poker, interview experts, read strategy books, etc.
2. Then produce a large set of situation-action rules:

IF your hole cards are a pair of face cards, AND the third such face card appears in the flop, THEN slow play the hand (i.e. do not raise, only call).

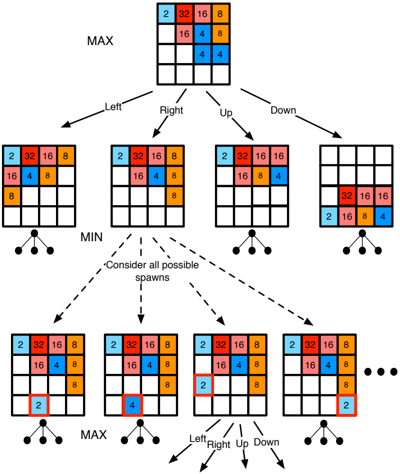
IF it is the final ('river') round of betting, AND an opponent raises before you, AND the pot odds are favorable THEN call the raise.

Data Driven Poker: Rollouts



Do this **thousands** of times to assess winning odds for player 2.

Search: The Core of AI



- Traditionally use Minimax for Adversarial Games
- At lowest (leaf) nodes, **heuristic evaluation** required.

Representation: Key to Effective AI Search

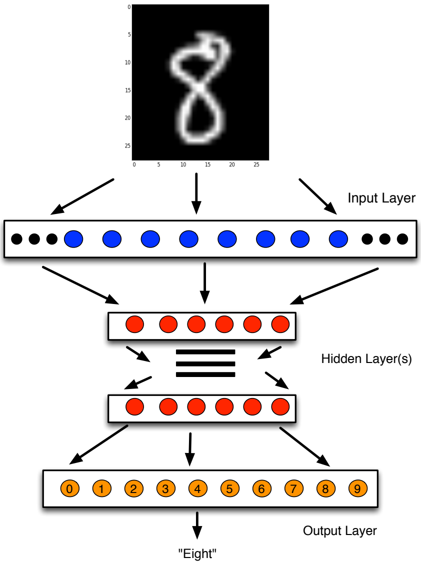
Pivotal Representational Tradeoff

1. Crafted by human engineers? Hard work to design, but the system's decision-making is often understandable.
2. Discovered by the AI system? Relatively easy to facilitate, but a headache to interpret results, particularly with non-symbolic systems (e.g. neural nets).

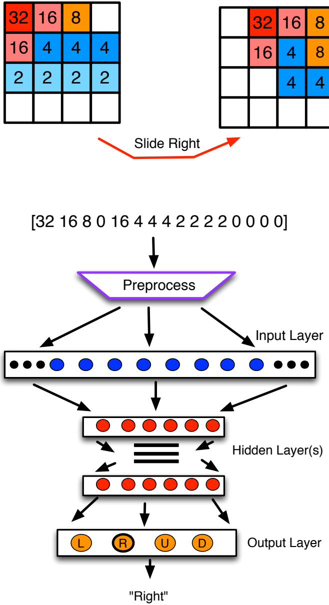
Recent Trends:

- Focus on AI-detected features, but
- Screaming need to explain workings of the black box.

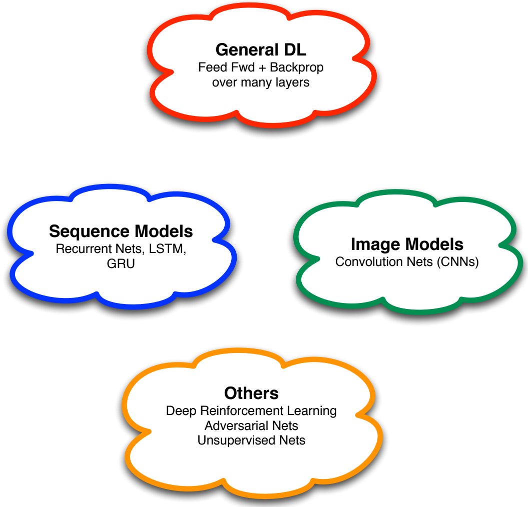
Deep Learning for Classification



Deep Learning for Action Policies

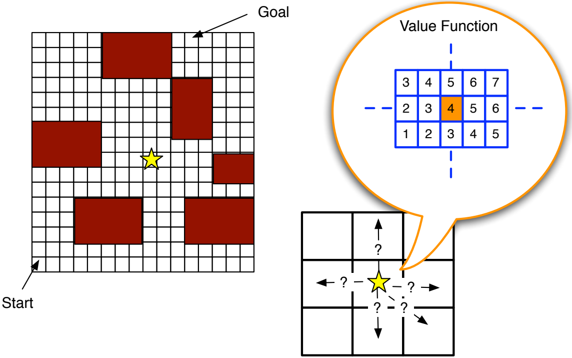


The Universe of Deep Learning



Andrew Ng (Youtube Video on DL)

Reinforcement Learning

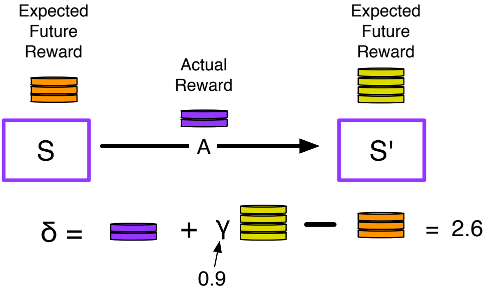


- Trial-and-error learning; Learning by doing.
- Intermittent feedback, often only at end of run/game.
- Goal: Learn two key functions: Value (Critic) and Policy (Actor).
- Value: game state(s) $\longrightarrow$ value (expected payoff from s to game end).
- Policy: state $\longrightarrow$ best action (yields highest-valued successor state).

Temporal Difference (TD) Learning

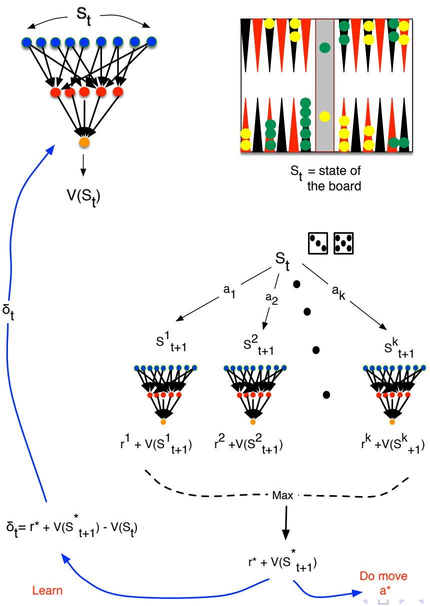
$$V(s) \leftarrow V(s) + \alpha[r + \gamma V(s') - V(s)]$$

- α = learning rate, γ = discount factor,
- r = reward after moving from state s to s'
- $\delta = r + \gamma V(s') - V(s)$ = TD Error = Level of **Surprise**



Elegant: Bootstrapping the value function via successor-state values.

TD-Gammon: Tesauro (1995)

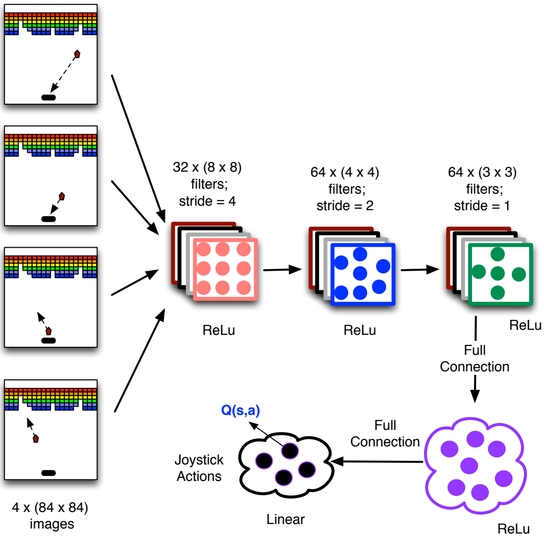


Sources of Recent AI Success

Enabling Techniques and Technologies

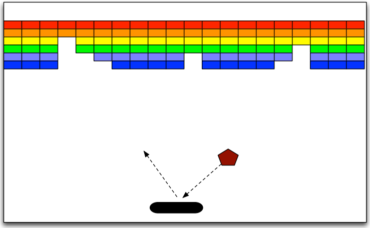
1. Backpropagation - the system learns important features without (much) human bias. Old method gradually improved over 40 years.
2. GPUs - handling **deep** nets with huge numbers of input neurons, millions of training cases, and millions of rollouts.
3. Convolution - finding meaningful patterns in complex images (such as Go boards).
4. Reinforcement Learning - beyond standard Supervised Learning in NNs...but still using backprop!
5. RL + NNs: The neural net maps states to values, but without requiring a lookup table for all 3^{361} states.
6. Monte Carlo Tree Search - extends Minimax search to games that are deterministic but evade good heuristic evaluation (of leaf nodes).
7. Google's Machine Park: 48 CPU's + 8 GPU's; 1201 CPU's + 176 GPU's for distributed AlphaGo.

Deep Convolution Network for Playing Atari Games



Human-level control through deep reinforcement learning, Mnih et. al.,

Deep Convolution Nets + RL$\rightarrow$ Atari Mastery



- Raw pixel images with minimal preprocessing used as input.
- 29 of 49 games (Breakout, Gopher, Space Invaders, etc.) yielded human-level performance or higher.
- Artificial General Intelligence (AGI): Same NN and training regime used for all 49 games.

Learning: Backprop using TD Error

Temporal Differencing

$$\delta = r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)$$

- $Q(s,a;\theta_i)$ = activation of output neuron for action a when given input state s and using parameters (i.e. weights) θ_i.
- s' = the next game state achieved when performing action a in state s.
- θ_i^- = network parameters from a slightly earlier phase of training.
- γ = discount factor (0.99).
- $r + \gamma \max_{a'} Q(s', a'; \theta_i^-)$ = target value (label) for supervised learning.
- δ^2 = Error term for supervised learning (via backprop).

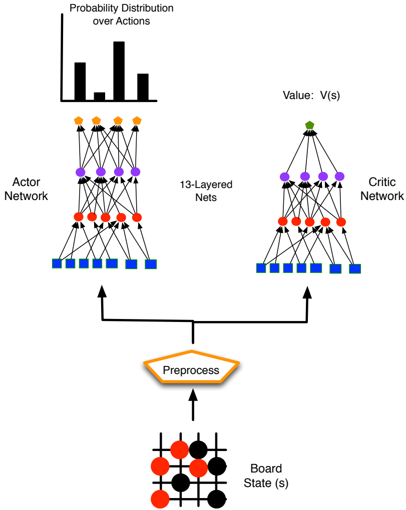
Very similar to TD-Gammon, but:

- Larger, deeper network, with convolution.
- Neural network computes action-value, Q(s,a), not state-value, V(s).
- Experience replay: constant retraining on old s-a pairs.

Keys to Atari-NN Success

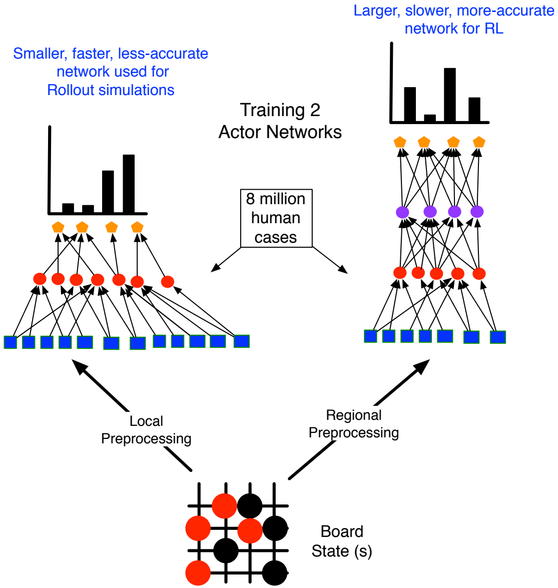
1. Experience Replay - Past (s,a) pairs are saved and random mini-batches of them are rerun by interleaving them with the current situations. This reduces various training biases by removing correlations between consecutive cases.
2. Convolution - detects and reuses subpatterns. Useful in many of Atari's homogeneous environments.
3. Target-generating Network (defined by θ_i^-) - Separating current model θ_i from the earlier model θ_i^- helps stabilize Q-Learning.
4. Clipping δ to the range [-1,1] also added stability.
5. Image preprocessing - reduces a 210 x 160 RGB image to 84 x 84 luminance values.
6. 4 images per input - provides enough context around each snapshot, e.g. to discern puck trajectory.

AlphaGo: Deep Nets + RL + MC Search



Mastering the Game of Go with deep neural networks and tree search, Silver et. al., Nature, 2016.

Supervised Learning from Expert Moves

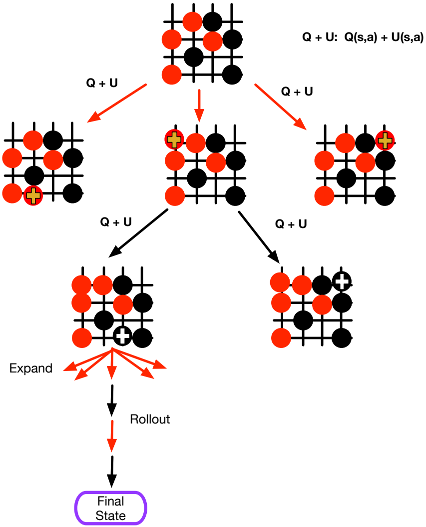


Monte Carlo Tree Search (MCTS) for Go

- Most AI Go players use MCTS.
- Originally designed for stochastic games: backgammon, poker and scrabble.
- Why use it for a deterministic game? The search space is so large, and most evaluation functions (i.e. heuristics) are so bad, that you need to try many alternate futures to assess the current state.
- AlphaGo combines MCTS with RL actors and critics produced by Deep Learning.

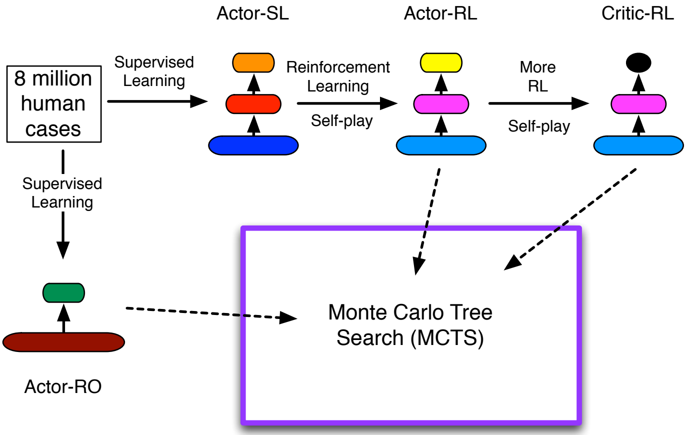
Monte-Carlo tree search and rapid action value estimation in computer Go, Gelly and Silver, AI Journal, 2011.

Monte Carlo Tree Search (MCTS)



Adversarial (like Minimax), but lower average branching factor, and heuristics are optional

AlphaGo = SL + RL + Search



Summarizing Roles of the 4 Neural Networks

1. Actor-SL: policy network based on human games
   - Starting point for creation of the Actor-RL policy network.
   - Source of prior probabilities, $P(s,a) = p(a \mid s)$, when expanding a leaf node in MCTS.
2. Actor-RL: policy network based on self-play
   - Basis for the value network, Critic-RL
   - Using Actor-SL instead of Actor-RL for producing $P(s,a)$ gives better results.
3. Critic-RL: value network based on Actor-RL and self-play
   - Evaluates leaf nodes in MCTS, which, in turn, leads to updated evaluations of $Q(s,a)$ on (s,a) branches; and $Q(s,a)$ values affect routes chosen during repeated tree search.
4. Actor-RO: faster (less intelligent) policy network based on human games (and local preprocessing).
   - Governs action-selection during the rollout phase of MCTS.

Results

1. October 2015: AlphaGo beats European champion , Fan Hui, 5 - 0.
2. March 2016: AlphaGo beats a Grand Master (ranked 2nd in the world), Lee Sedol, 4-1.
    - AlphaGo makes some moves that experts find very unusual, but prove to be brilliant.
    - I guess I lost the game because I wasn't able to find any weaknesses...Lee Sedol.
    - Here in Korea, back in the U.S. and all over the world, there is much more sadness and introspection than I expected. Andrew Okun, President American Go Association.
    - When you watch really great Go players play, it is like a thing of beauty. So I'm very excited that we've been able to instill that level of beauty inside a computer. Sergery Brin, Google.
3. May 2017: AlphaGo beats world champion Ke Jie, 3-0...with the version *Alpha Go Master*, which won 60 online games -vs- other Go masters.

Differences that Make a Difference

- Atari NN - The board is 84 x 84 pixels.
- Go - The board is 19 x 19 spots, with 3 possible values (black, white, open).

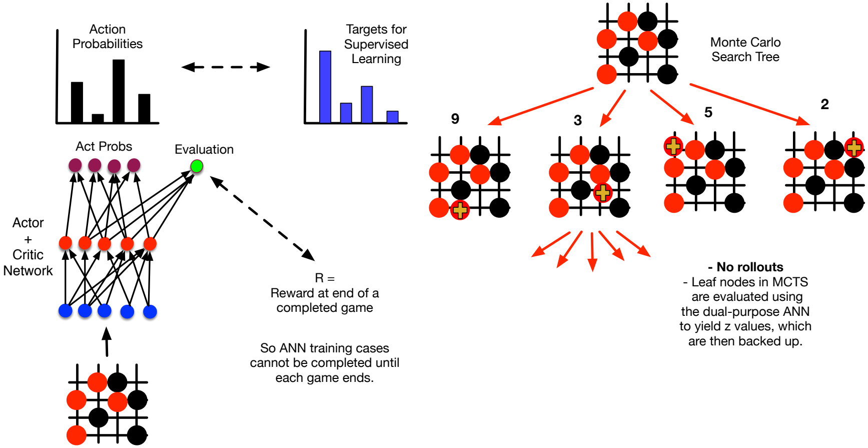
1. Why is Go much harder for a neural network to learn than Atari?
2. Why did the Go board have to be semantically preprocesed (i.e. interpreted w.r.t. rules of Go), while the original Atari board was only superficially/syntactically preprocessed (i.e., pixel aggregation and filtering)?

AlphaGo Zero

- Learns entirely from self-play. No supervised training on expert games.
- No significant preprocessing of boards.
- Combines actor and critic into one neural network.
- MCTS $\rightarrow$ supervised learning of actor + critic.
- No rollouts during MCTS expansion and traversal.
- Begins as a horrible Go player, but after 3 days surpasses version of AlphaGo that beat Lee Sedol.
- After 40 days of training... Beats AlphaGo 100 - 0.
- A huge victory for bottom-up AI.
- One more step toward AGI: Basic approach also used for other games, e.g. chess.

Mastering the game of Go without human knowledge. Silver et. al., 2017

AlphaGo Zero Learning



Alpha Zero

- Extending AlphaGo Zero to other games.
- Still no expert knowledge. All learning from self-play.
- Beat Stockfish, world's (former) best chess bot: 28 wins - 0 losses - 72 ties..after 9 hours of training.**
- Beat Elmo, world's (former) best Shogi (Japanese Chess) bot: 90-8-2... after 9 hours of training.
- Uses much less computing power than AlphaGo (44 TPU's -vs- 4)

**Some chess experts claim the tests were unfair.

Mastering Chess and Shogi by self-play with a general Reinforcement Learning algorithm. Silver et. al., 2017.

Google Deepmind extends their DL and DRL beyond games.

- Climate Control via DL- 40% reduction in cooling costs for Google's server park.
- Medical Diagnosis using DL
  - Expert-level classification of retinal-disease severity from OCT scans.
  - Early detection of breast cancer from AI-interpreted mammograms
- Navigation via Deep RL - showed emergence of grid cells $\rightarrow$ empirical support for neuroscientific theories.

+ Early Detection of Cerebral Palsy from DL-based motion analysis (Norwegian Open AI Lab)

Humans Need not Apply

- No need for human expertise (at least in games). AI learns by itself.
- No need for human-labeled data. RL systems generate their own labeled datasets as they explore the world of possibilities (e.g. play the game).
- Unbiased by humans $\rightarrow$ extremely creative solutions!
- Humans are no longer the undisputed masters of pattern recognition.

Key insights by Nicholas Carr

- The Shallows (2011)

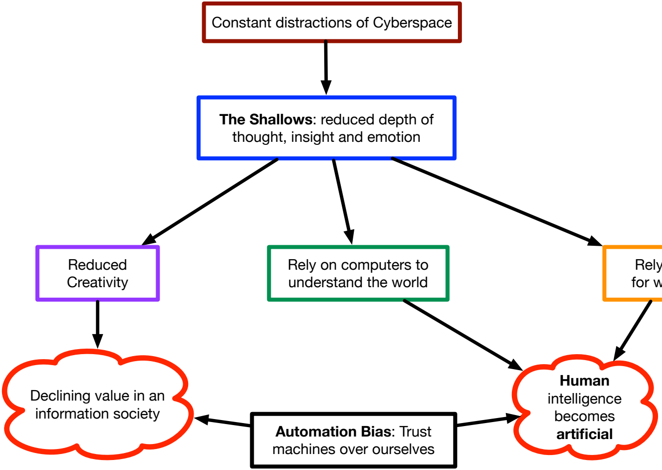
  Dumbing-down of a tech-dependent humanity

- The Glass Cage (2014)

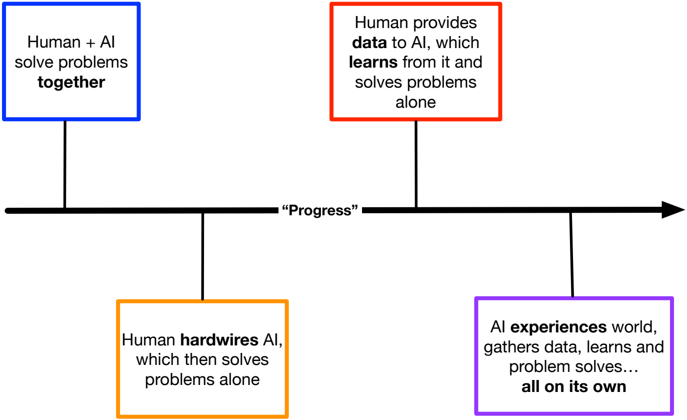
  *Reclaim our tools as instruments of ourselves, as instruments of **experience** rather than just means of **production***

AI systems should help us learn, not just do the job themselves.

The Plight of the Infovore



The Future of Human-AI Interaction



- For humanity's sake, move left!
- AI needs to work with humans...and help **us** get smarter.