Methods

Murray Shanahan

Overview

- Classical symbolic AI
- Behaviour-based AI
- AI through machine learning

Classical symbolic AI

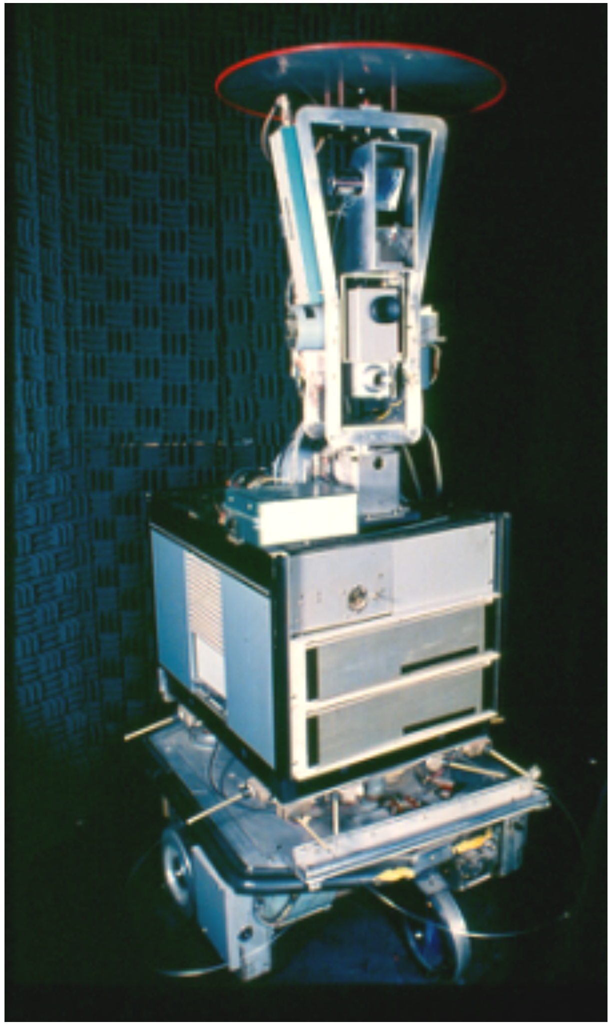
Classical AI

- Based on the manipulation of *symbolic representations* with a language-like structure (sets of sentences or propositions)
- In its purest form, these symbolic representations are sets of sentences in formal logic
  - Representations are composed of symbols, such as the constant *Block09* and the predicate *On(x,y)*
  - They have a *combinatorial syntax*. Symbols can be combined thanks to their predicate-argument structure, and logical connectives such as conjunction ($\wedge$), disjunction ($\vee$), and implication ($\rightarrow$). For example:

    $$On(Block03,Block02) \wedge On(Block07,Block08)$$
  - They have a *denotational semantics*. Constants stand for “things” in the world, such as the 9^{th} block in a scene, and predicates stand for relations in the world, such as *x* being on top of *y*

Case Study: Shakey

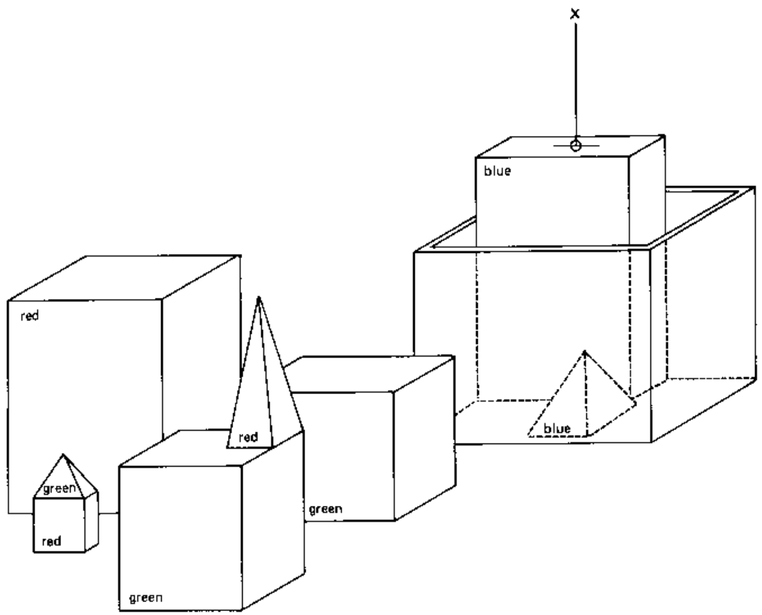
- Shakey (Nilsson, *et al*.) was a robot that inhabited a simple Blocks World type environment
- Working from camera images, it built and maintained a model of its environment in a predicate calculus style formalism
- It responded to commands to achieve given goal states by constructing a plan, using symbolic reasoning, to achieve that state
- If the world changed during execution of the plan, it tried to replan and carry on



Case Study: SHRDLU

- SHRDLU (Winograd) was a disembodied (ie: not robotic) system that carried a dialogue with a user about a virtual "microworld" – the Blocks World, comprising a set of cuboids and pyramids on a flat surface, plus a robot arm
- The system parsed the user's English sentences and turned them into a predicate calculus like notation
- It then used logical deduction to work out how to answer the user's questions or carry out their commands (modifying the Blocks World accordingly)
- The current state of the Blocks World was represented internally in the same predicate calculus style formalism used for translating English sentences. The "physics" of the Blocks World was represented as a set of rules in the same notation

SHRDLU in Action



A SHRDLU screenshot

```
Person: PICK UP A BIG RED BLOCK.
Computer: OK.
Person: GRASP THE PYRAMID.
Computer: I DON'T UNDERSTAND WHICH
PYRAMID YOU MEAN.
Person (changing their mind): FIND A
BLOCK WHICH IS TALLER THAN THE ONE YOU
ARE HOLDING AND PUT IT INTO THE BOX.
Computer: BY "IT", I ASSUME YOU MEAN THE
BLOCK WHICH IS TALLER THAN THE ONE I AM
HOLDING.
Computer: OK.
Person: WHAT DOES THE BOX CONTAIN?
Computer: THE BLUE PYRAMID AND THE BLUE
BLOCK.
Person: WHAT IS THE PYRAMID SUPPORTED BY?
Computer: THE BOX.
```

A fragment of SHRDLU dialogue

The Role of Planning

- Both Shakey and SHRDLU, in common with modern examples of classical AI, made extensive use of *planning*, which we will study later in the course
- Planning is the task of finding a sequence of actions to achieve a given goal. Shakey used planning to find a sequence of movements, rotations and pushes to get itself and the objects in its (real) environment into desired configurations. Similarly, SHRDLU used planning to find a sequence of pickup and putdown actions to get its (virtual) Blocks World into a requested configuration
- Planning involves *search* and is therefore computationally intensive. In any non-trivial domain, planning is NP-complete (even in the Blocks World)

The Role of Representation

- Another feature of both Shakey and SHRDLU that they share with modern examples of classical AI is their reliance on symbolic *representations*, specifically *propositional* representations
- A propositional representation is a set of *propositions*, that is to say sentences that can be *true*, either in the robot's actual environment (real or virtual) or in some hypothesised world
- A representation comprising true propositions can be said to be *correct*. Similarly, truth-preserving operations on propositional representations, such as deductive reasoning, can be said to be *correct*
- Planning can be given a deductive account, as we'll see later
- So a classical AI system is intelligent because its actions are the outcome of *correct reasoning with correct representations*

Agent-based Systems

- Classical symbolic AI evolved into the agent-based approach
- An *agent* is a “complete” system that exists an environment that it *senses* and within which it *acts* autonomously. The environment in question can be virtual (eg: the Internet)
- Examples include web crawlers, auction agents, shopping agents
- A large and complex system with many heterogenous, distributed components can sometimes be usefully organised as a collection of agents that interact with each other (a *multi-agent system*)
- A major issue then is how the agents should interact with each other. Researchers have developed agent communication languages to facilitate this, along with protocols for interaction

Probability and Statistics

- Classical symbolic AI tended to use “crisp” representations, where propositions are etiher true or false
- But recurring theme in contemporary AI is the need to reason under uncertainty
  - *Probabilistic robotics* (probabilistic models from noisy sensor data with noisy motor outputs)
  - *Statistical machine learning* (eg: machine translation)
  - *Reinforcement learning* (maximising *expected* reward)
- The mathematics of Bayesian probability and decision theory is applicable here

Behaviour-based AI

A Critique of Classical AI

- Classical AI was the dominant paradigm from the 1960s to the late 1980s
- It *is* a beautiful idea. By appealing to logic-based representations, we can explain the success of a system's behaviour in terms of the *correctness* of its representations and reasoning processes
- But in the late 1980s and early 1990s, a number of authors, notably roboticist Rodney Brooks, launched a devastating critique
- He drew attention to several issues
  - Fragmentation of AI as a discipline
  - Top-down methodology
  - Disregard for embodiment
  - The representational bottleneck

Brooks' s Critique (1)

- Fragmentation of AI as a discipline
  - The field of AI had become broken up into many specialist sub-disciplines – vision, natural language processing, theorem proving, knowledge representation, robotics, learning
  - No-one was worrying how all the bits would fit back together again to make a whole agent
  - Solution: Build complete agents with perception, action, and learning
- Top-down methodology
  - Researchers were trying to build systems that exhibited high-level behaviour, but which performed poorly and only in specialised domains (such as chess, theorem proving, or parsing)
  - No-one seemed concerned that the methods would not scale up or generalise
  - Solution: Be more biologically inspired. Follow evolutionary order

Top-down versus Bottom-up

High-level cognition
(planning, reasoning, language)

Top-down
(classical AI)

Argument against:
"By neglecting the low-level foundations of intelligence, it is blind to many important issues especially relating to embodiment"

Bottom-up
(biologically-inspired AI)

Argument against:
"It never gets off ground-level. No-one has demonstrated how high-level cognition emerges from low-level behaviours"

Low-level motor control
(locomotion, insect-like behaviour)

The Symbol Grounding Problem

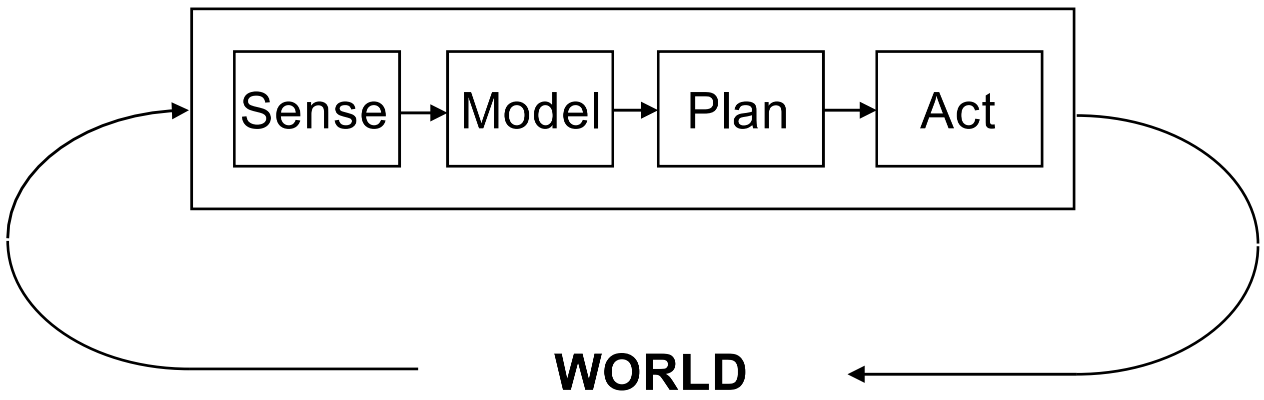
- The *symbol grounding problem* is another difficulty for the top-down methodology of classical AI
- Classical AI relies on the ability to give a denotational semantics (meaning) to the symbols used in its representations: *Block9* stands for the 9th block in the scene, for example
- But how can this meaning be made *intrinsic* to the system, and not *parasitic* on meanings in the head of the system's designer? In other words, how can symbols be meaningful *for* the system itself, in the way they are for we humans?
- The answer cannot rely solely on defining the meaning of one symbol in terms of another (itself meaningless) symbol. This process must bottom out somewhere, such as in a robot's sensorimotor interactions with the physical world

Brooks's Critique (2)

- Disregard for embodiment
  - Most classical AI programs were disembodied question-and-answer systems, as opposed to embodied systems like robots (or animals).
  - But in Nature, cognition helps an organism to preserve its integrity, maintain its wellbeing, and reproduce. Its essence is to support embodied interaction with a fast-moving, incompletely known world of complex, spatially-located objects
  - So to try to understand cognition in a disembodied setting is misguided
  - Solution: Build robots not expert systems

Brooks' s Critique (3)

- The representational bottleneck
  - "Embodied classical AI systems like Shakey don't scale up because they are tied to a 'sense-model-plan-act' control architecture"
  - The "model" part builds a representation of the ongoing situation in the world, which is only viable in static, engineered environments. Moreover, maintaining an accurate model of the world is difficult, and planning is computationally intensive
  - Solution: Get rid of representation and planning!



Behaviour-Based Robotics

- Eliminating representation is a highly controversial step. Many researchers who were sympathetic with the rest of Brooks's critique felt that he had "thrown out the baby with the bathwater" by rejecting representation
- (Eventually Brooks himself became less radical)
- Nevertheless, a whole new research area was spawned that adopted representation-free alternatives to classical AI's sense-model-plan-act architecture
- Indeed, rich and complex behaviour is possible without recourse to internal representation, and Brooks demonstrated this using *behaviour-based* control architectures, which have had a lasting influence on robot design and AI

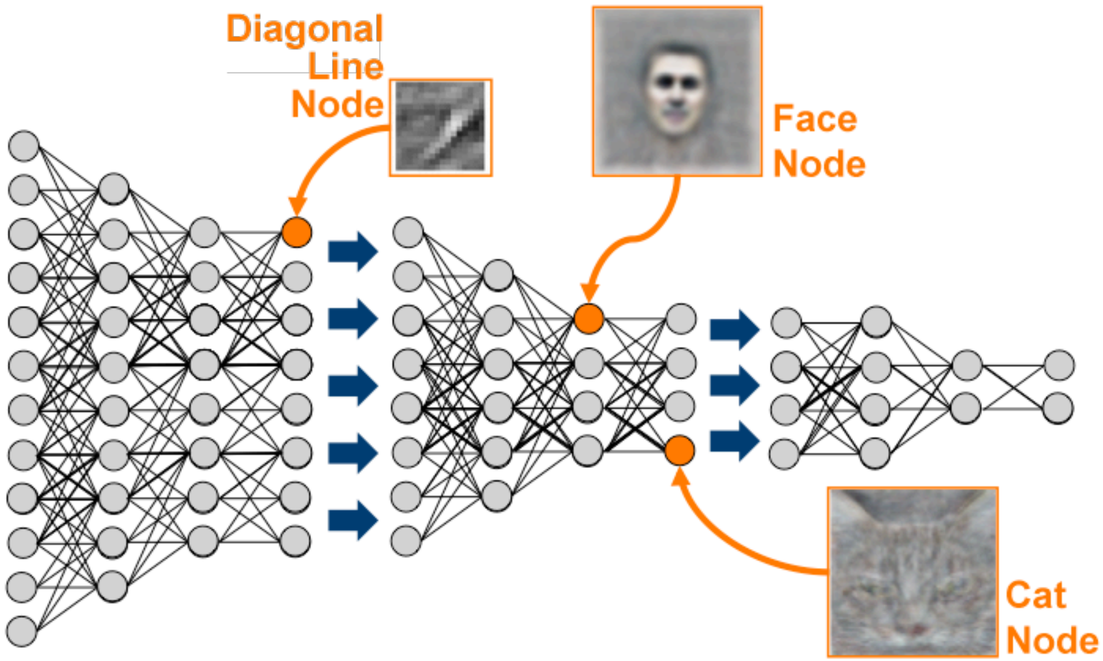
AI through machine learning

Learning

- An intelligent system must *adapt* to its environment, which is why machine learning is a central topic in AI
- Training a neural net is one form of machine learning. Other prominent techniques include
  - *Reinforcement learning* – learning a behaviour on the basis of positive and negative feedback
  - *Inductive logic programming* – learning a logic-based representation of a category
- Machine learning tasks can be either
  - supervised – in which case a set of training data is supplied that is already structured (into, say, positive instances and negative instances of a category to be learned) – and
  - unsupervised – in which finding structure in the data is part of the problem

Deep Learning (1)

- Deep learning involves a hierarchy of layers of neurons that learn increasingly abstract features
- Here's an example from the lab of Andrew Ng. It learns features in a visual dataset

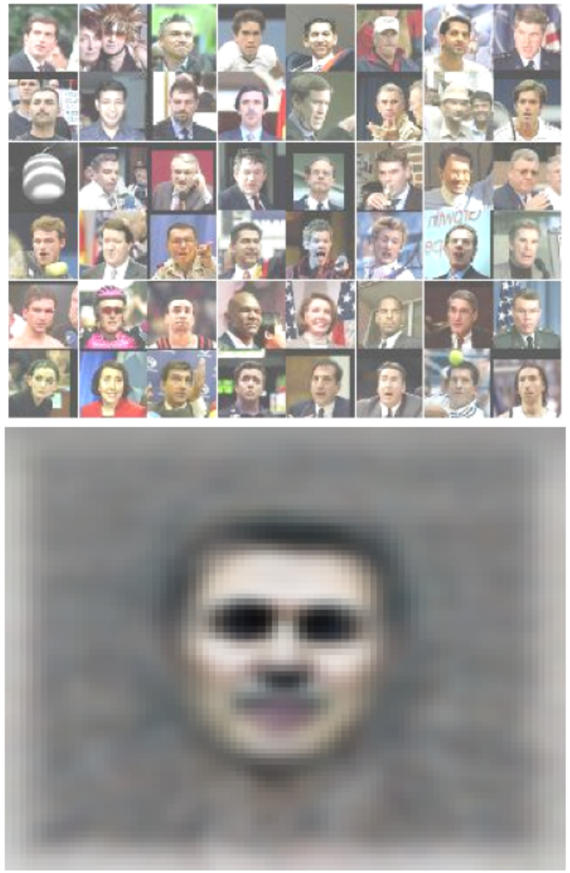


- The system is presented with tens of thousands of images, including faces and distractors

Architecture for deep learning
(Le, *et al.*, ICML 2012
image: http://theanalyticsstore.com)

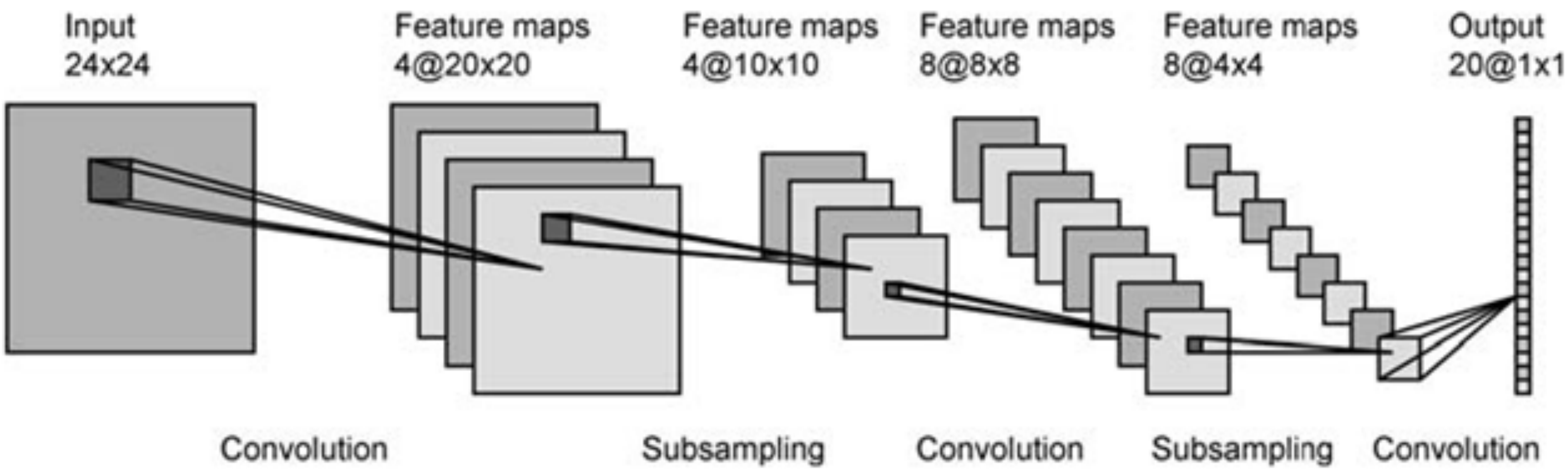
Deep Learning (2)

- This network is an *autoencoder*, whose job is to reconstruct the original data, but to reduce its dimensionality
- In other words, the intermediate layers are a *compressed*, distributed representation of the input image
- Learning here is an *optimisation* problem: maximise the network's ability to compress the data with minimal loss
- Networks are trained by adjusting the weights of the connections using *back propagation*



What excites the best neuron
(Le, *et al.*, ICML 2012)

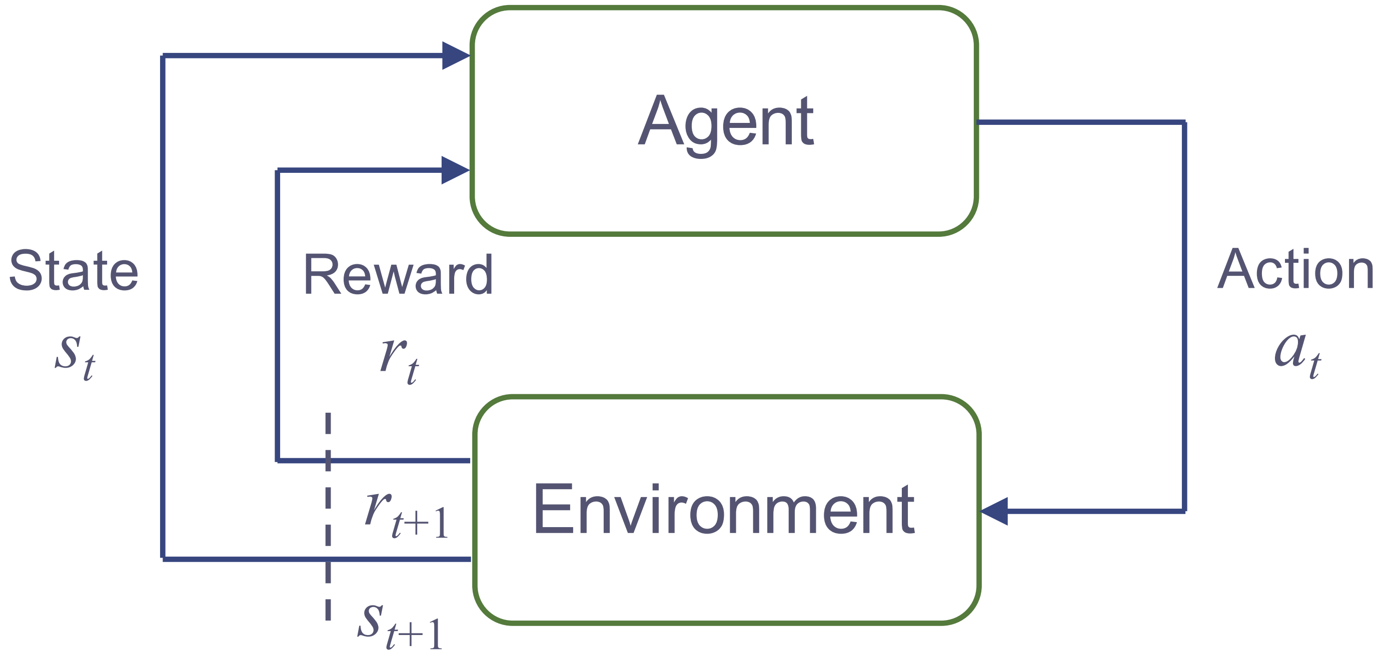
Convolutional Networks



CNN architecture
(Haykin, 2008)

- A convolutional neural network (CNN) alternates convolution and sub-sampling (or max-pooling) layers
- Convolutional layers pick out features, using shared weights
- Sub-sampling layers isolate max (most prominent) cell

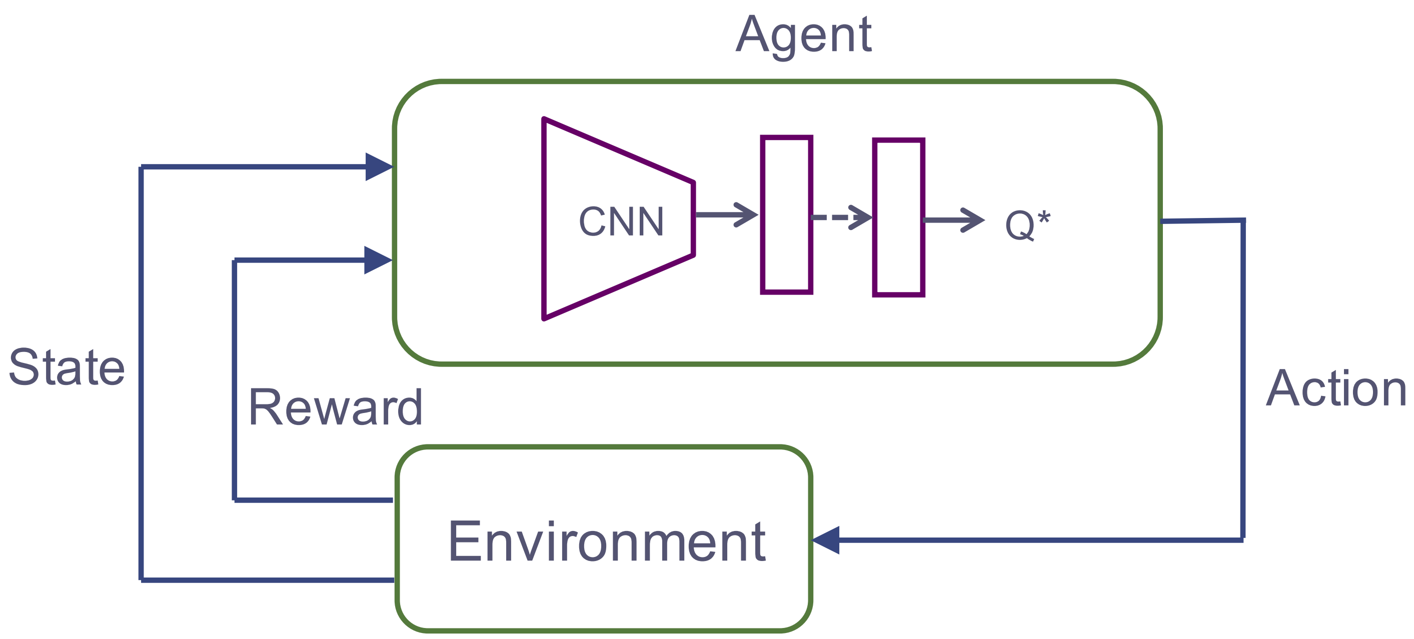
Reinforcement Learning



Sutton & Barto, 1998

- A *policy* is a mapping from states to actions
- Use trial-and-error to estimate expected future reward given action and state
- Find a policy that maximises expected future reward

Deep Reinforcement Learning



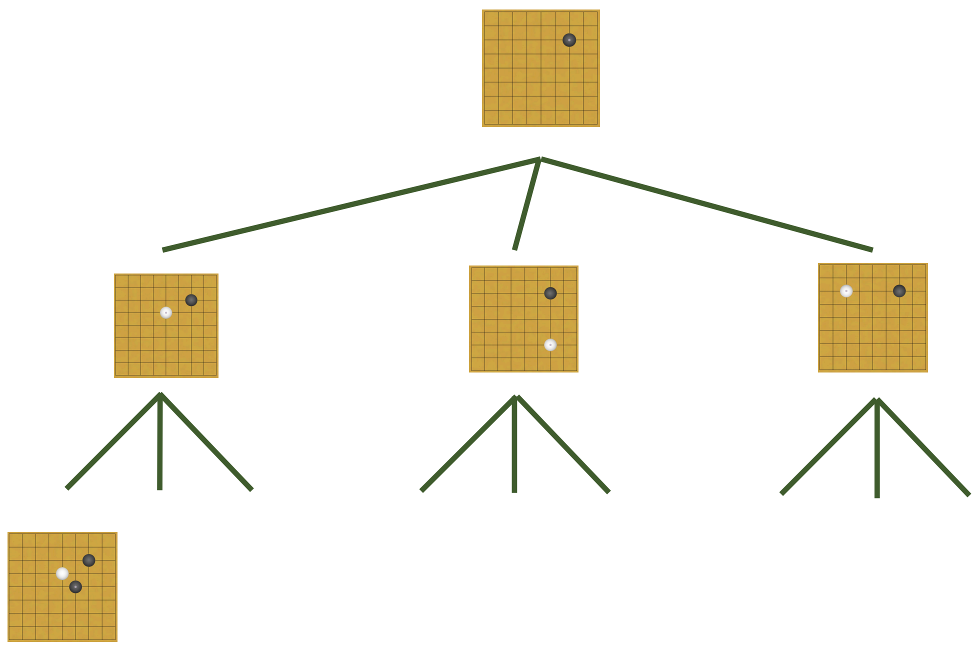
- Combines deep learning with reinforcement learning
- In effect “discovers” state space while doing trial-and-error learning

DeepMind's DQN

- Google DeepMind's DQN attains superhuman capabilities in a suite of Atari video games
- Learns each game by trial-and-error, given only raw pixels and score
- Uses a deep reinforcement learning architecture

How Does AlphaGo Work?
Classical AI (search)

- Like IBM's DeepBlue, AlphaGo searches a tree of possibilities
- It uses *Monte Carlo tree search*, which explores random branches
- But it exploits neural networks to *prune* the tree and narrow down the set of board positions and moves it looks at

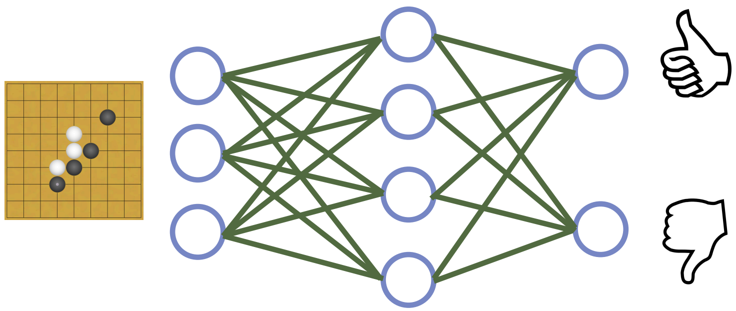


Searching the tree of possibilities

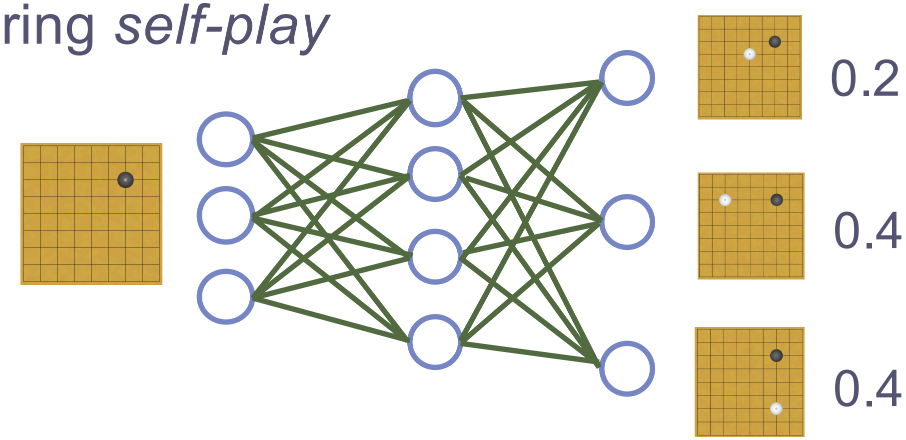
How Does AlphaGo Work?

Neural network AI

- Uses *deep learning*
  - Trained on a large database of expert games
  - Learns to evaluate board positions
  - And to determine likely moves
- Uses *reinforcement learning* during *self-play*



A neural network that
evaluates board positions



A neural network that
determines likely moves

Reading

Pfeifer, R. & Scheier, C. (2001). *Understanding Intelligence*, MIT Press. *Presents the embodied, biologically-inspired alternative to classical AI*

LeCun, Y., Bengio, Y. & Hinton, G. (2015) Deep learning. *Nature*, 521(7553):436–444. *A good overview of deep learning by three pioneers of the field*

Hutter, M. (2005). *Universal Artificial Intelligence: Sequential Decisions based on Algorithmic Probability*. Springer. *This book is expensive and very mathematical, but take a look at* http://www.hutter1.net/ai/uaibook.htm