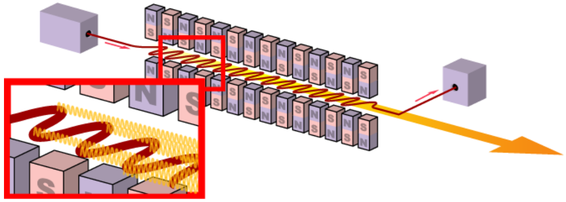
Hybrid AI for Interstellar Missions

Prof Alex Ellery
Centre for Self-Replication Research (CESER)
Carleton University, Ottawa, CANADA

Preamble

- I do not discuss human astronauts for interstellar flight – robotic missions only
- Interstellar flight will require autonomous operations (humans asleep anyway)
- **Lubin approach** – extreme redundancy scattershot approach with minimal cost per nanospacecraft to permit millions
- This makes sense for today…
- We assume a targetted approach….

- There are **3 types of interstellar mission**:
  - (i) **flyby**, e.g. Breakthrough Starshot → BIS Daedalus
  - (ii) **in-situ exploration of extrasolar system** to search for life – deceleration stage
    e.g. Forward-type laser-propelled sail
  - (iiia) **biological ETI encounter**
  - (iiib) **artificial ETI encounter**



- ALL involve 50-100 year transit phase
- We assume interstellar spacecraft that encapsulate significant engineering investment with extensive payload instrumentation, i.e. high cost, e.g. **Daedalus-type starship**

Be Careful – It's Dangerous Out There

- I assume that Kuiper belt/Oort cloud **icy bodies** of solar/extrasolar system are diffusely distributed and are not collision hazards
- High particle flux from ISM gas and dust - mitigated through an **eroding shield**
- Radiation – high energy galactic cosmic rays – **onboard electronics** based on vacuum tube technology
- **Random stuff** – long-duration components exhibit a bathtub failure rate distribution over time
  - infant mortality and senility flank background finite probability of failure ~constant
- Spacecraft can fail – software workarounds can reconfigure mission, EXOSAT (1983)
  - hardware failures require redundant systems, Galileo Jupiter mission (1989)
- ∃ modern fault diagnosis methods, e.g. EKF, PCA, ANFIS
- 50-100 year interstellar transit introduces the problem of **AVAILABILITY**
- 100 year starship study advocated multiple redundancy, high reliability components and evolutionary hardware (FPGA programmed by GA)
- There are diminishing returns to redundancy: $A = \frac{MTBF}{MTBF+MTTR+MTFS}$
  - MTBF→∞ is a measure of reliability (traditional approach)
  - *MTTR→0 is a measure of maintainability* (onboard servicing)
  - *MTFS→0 is a measure of logistic supply* (onboard manufacture)

Bare Necessities for Interstellar Flight

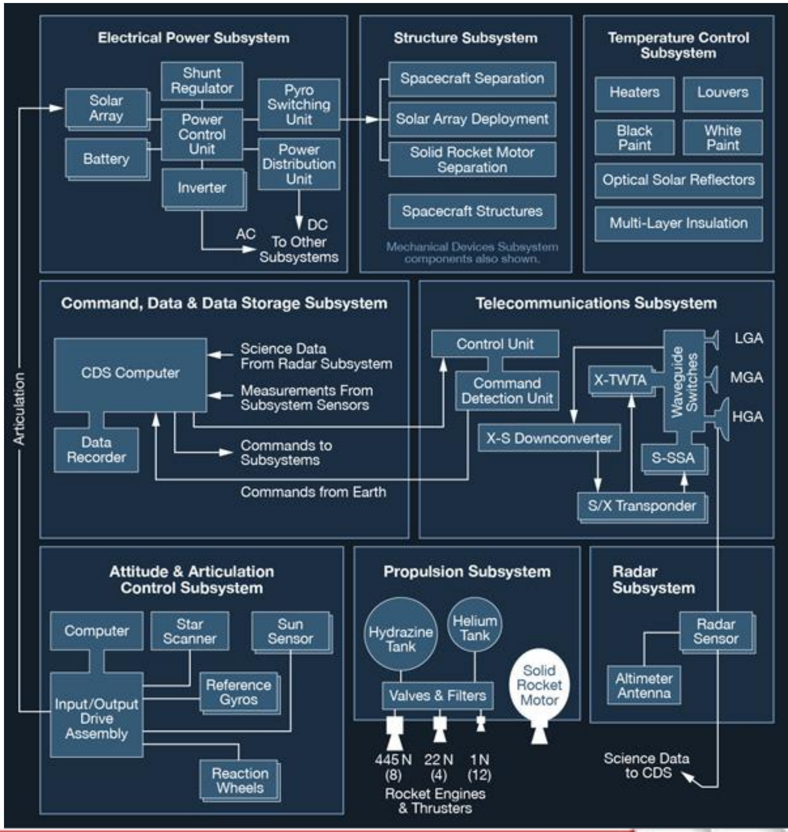
- BIS Daedalus starship was supported by several **Wardens** – freeflying servicing “fixit” robots similar to robotic freeflying servicer concepts

- **Lessons from space servicing**:
  - (i) standard interfacing of modules
    – mechanical/electrical/optical/thermal
  - (ii) robotic handling is challenging
- Modularity simplifies robotic handling
- It assumes that ∃ stockroom of pre-manufactured modules for replacement

- This is not realistic for a starship
- Corollary: **Full availability requires that starship is designed for self-repair**
- We need full **self-manufacturing facility onboard** to replace ANY component on-demand from a limited set of feedstock
- Even though a structured environment, **onboard manufacture will require some onboard intelligence** to plan manufacturing schedules including conflict resolution, Bayesian interpretation of data in relation to hypotheses, pattern recognition, etc

Minimalist Demandite Concept

- It is **CRUCIAL** to minimise the range and amount of feedstock required to maintain and repair the starship
- Our **DEMANDITE** concept maps <u>functional material requirements</u> with a fixed set of feedstock resources
- There 7 basic subsystems onboard a spacecraft:
  - Propulsion system
  - Attitude/orbit control including sensors/actuators
  - Structure & mechanisms
  - Thermal control
  - Power
  - Onboard computing including sensor nets
  - Communications (microwave or optical)

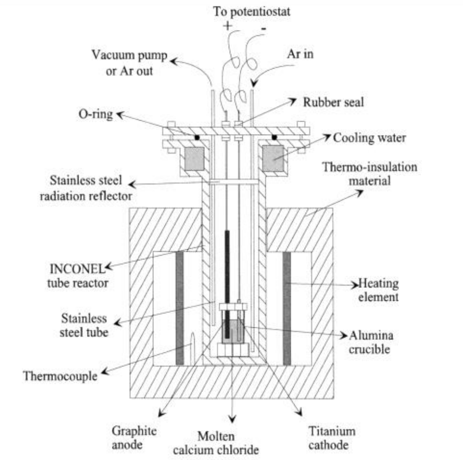


- **~10 basic materials** can supply full functionality for all the subsystems of a generic robotic spacecraft
- **Onboard FabLab** must manufacture feedstock into replacement components **<u>including itself</u>**

Carleton UNIVERSITY

Multifunctional Alumina

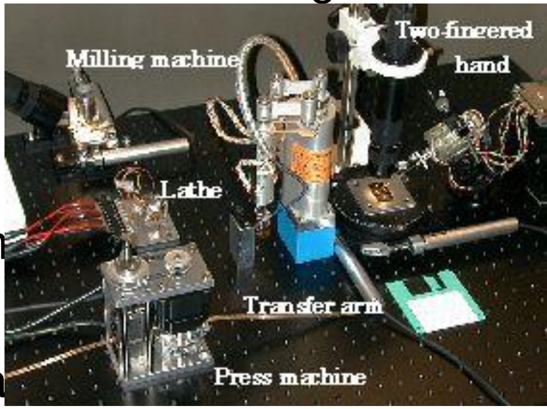
- Consider feedstock of **alumina** (Al_2O_3):
- Physical properties are second only to diamond:
  - **Refractory** and chemically inert
  - **High hardness** for tooling applications (including drilling)
  - **Additive to composites**, e.g. **cermet** (Al_2O_3 in Ni binder) for high strength and high temperature tolerance
- Alumina may be reduced directly to >99% Al metal through **FFC molten salt electrolysis**
- **Aluminium is a multifunctional metal** – its versatility suggests that it should be a **high priority target**:
  - good thermal conductor at low temperature below 520^{o}C
  - good thermal radiator for thermal louvres/radiators
  - good electrical conductor (pylon-mounted electrical cables)
  - lightweight structural metal ideal for spacecraft structures
  - Al powder combusted with Fe_2O_3 acts as thermite weld
  - alloyed with Si, silumin is wear resistant for tooling



Functionality (mass fraction)	Lunar-Derived Material
Tensile structures (25%)	Wrought iron Aluminium
Compressive structures (+50%)	Cast iron Regolith + binder
Elastic structures (trace)	Steel springs/flexures Silicone elastomers
Hard structures (3%)	Alumina
Thermal conductor straps (1%)	Fernico (e.g. kovar) Nickel Aluminum
Thermal radiators (3%)	Aluminium
Thermal insulation (3%)	Glass (SiO_2 fibre) Ceramics such as SiO_2
High thermal tolerance (4%)	Tungsten Alumina
Electrical conduction wire (7%)	Aluminium Fernico (e.g. kovar) Nickel
Electrical insulation (1%)	Glass fibre Ceramics (SiO_2, $Al_2O_3$and TiO_2) Silicone plastics Silicon steel for motors
Active electronics devices (vacuum tubes) (12%)	Kovar Nickel Tungsten Fused silica glass
Magnetic materials for actuators (5%)	Ferrite Silicon steel Permalloy
Sensory transducers (5%)	Resistance wire Quartz Selenium
Optical structures (11%)	Polished nickel/aluminium Fused silica glass lenses
Lubricants (trace)	Silicone oils Water
Power system (20%)	Fresnel lens + thermionic conversion Flywheels
Combustible fuels (+250%)	Oxygen Hydrogen

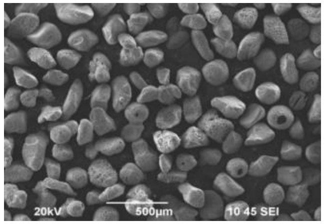
Manufacturing & Assembly

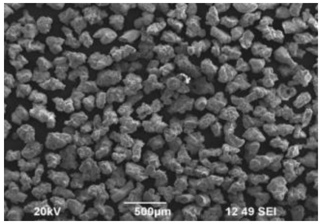
- Manufacturing converts **feedstock into parts and components**
- There are many manufacturing machines for fashioning objects – casting, moulding, lathe, milling station, drill press, bending press, laser cutter, electric discharge machining, etc (e.g. FabLab)
- There are many joining operations including parts assembly, friction stir welding, wire winder, etc
- Almost all such processes may be integrated into a cartesian robot configuration – the 3 or 5 DOF **CNC machine**
- **Subtractive manufacturing methods waste ~90% materia**
- Additive manufacturing (3D printing) builds parts and components layer-by-layer without wastage
- 3D printing is a versatile manufacturing method for constructing 3D structures
- There are many 3D printing technologies – FDM, SLA, LOM, SLS/M, EBAM, BJ, etc
- All 3D printers = **cartesian robots**

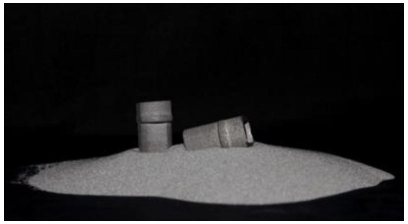


3D Printing

- **EBAM** is **electron gun** (high voltage **vacuum tube**) but restricted to metals, e.g. NRC wire-fed EBAM – Al alloy printing
- FFC molten salt electrolysis yields >**99% metal alloy sponge** that can be crushed into powder for powder metallurgy or 3D printing – **SLS printing has been demonstrated** with Ti test parts





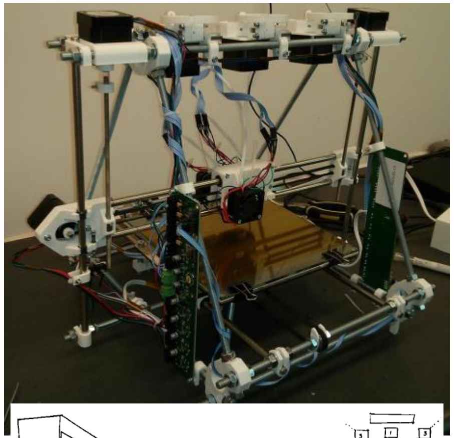


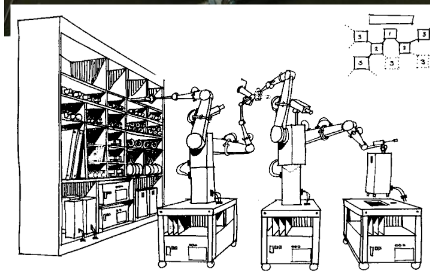
TiO_2 powder → Ti powder → 3D printed Ti parts

- We are building rigid multi-material 3D printer to print in metals and plastics (potentially ceramic)

- **Selective solar sintering** uses Fresnel lenses to generate thermal energy
- 3D printing by **Fresnel lens** for melting Al alloy powder on powderbed
- We have deposited molten Al wire tracks onto silicone plastic insulation
- **Problem:** coupling Fresnel lens focus into fibre optic cable
- ➢ integrate milling head (CNC machine) - integrate assembling wrist
- ➢ steels/silicone-derived ceramics ($SiO_xC_y + (1\text{-}x+2y)O_2 \rightarrow SiO_2 + yCO_2$)

3D Printing = Universal Construction Mechanism





- **RepRap** FDM 3D printer can print many of its own **plastic parts**
- Full self-replication requires 3D printing:

  (i) structural metal bars and components (SLS/M or EBAM)

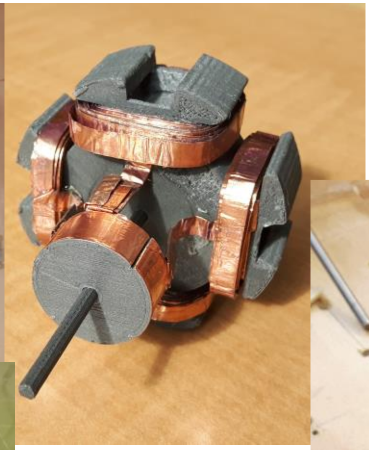
  (ii) **electric motor drives**

  (ii) **electronics** boards

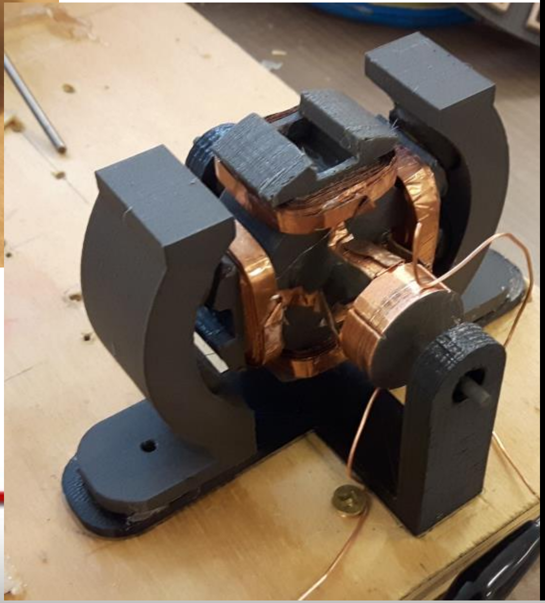
  (iv) **computer hardware/software**
- **Universal constructor is a kinematic machine** that can manufacture any other machine *including a copy of itself*
- We adapt UC to unstructured environments through a suite of **kinematic machines**
- **All machines of production are kinematic machines**
- 3D printer suite constitutes a **Universal Constructor** as a generalized kinematic machine that can construct any other kinematic machine
- **Kinematic machines** are specific kinematic configurations of **electric motor systems**
- **From 3D printed electric motors, sensors and control electronics, *omnia sequitur*…**
- If we can 3D print motor systems, we can build any manufacturing machine onboard on-demand

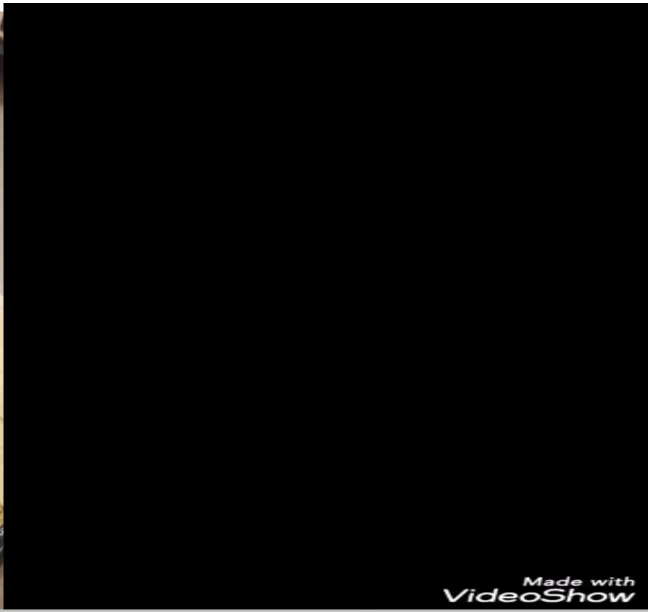
Fully 3D Printed Motor

- 3D printed rotor (ProtoPasta)
- 3D printed permanent stator magnet (Oak Ridge National Laboratory)
- LOM-style copper tape wiring/commutator wound around rotor
- 3D printed shaft + bearings
- **It is a small step from a self-repairing to self-replicating starship**

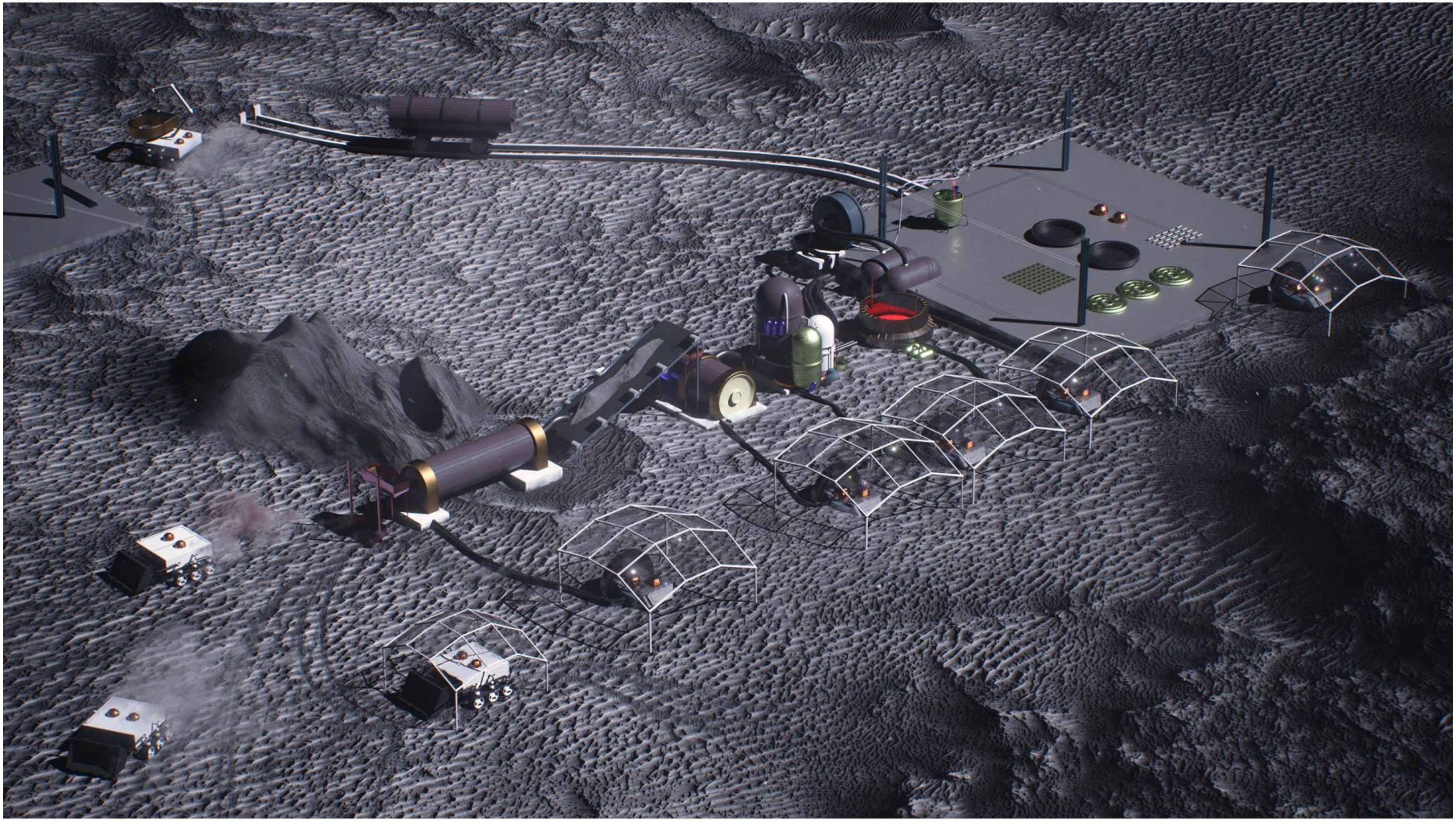


Kinematic machines need sophisticated planning and control

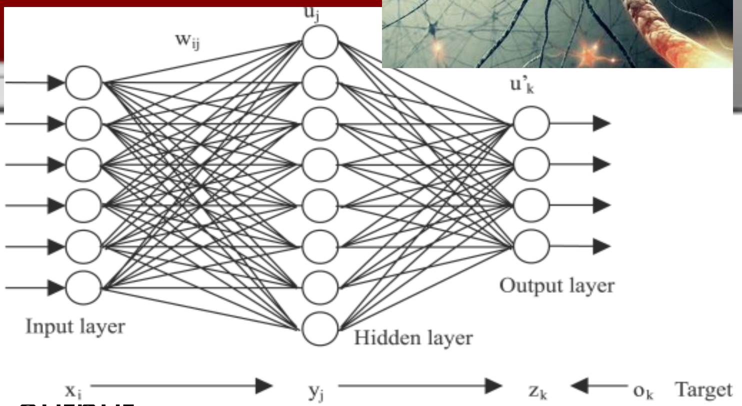




Lunar Industrial Architecture

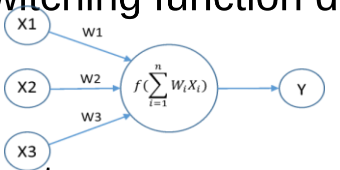


Artificial Neural Networks



- Stone-Weierstrauss theorem states that MLP can approximate any nonlinear function – it is a multidimensional curve-fitting algorithm similar to a polynomial function
- MLP comprises a minimum of 3 layers – input – hidden -output
- Neural networks implement weighted switching function determined by threshold:

$$y_i(x(t)) = f\left(\sum_{j=1}^{n} w_{ij} x_j(t) + w_i\right)$$

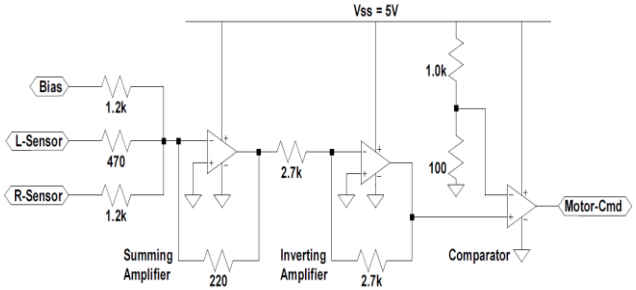


- If $w_{ij} > w_i$, neuron fires; otherwise, no output
- Activation function f() is a nonlinear squashing function:

  (i) sigmoidal function is typical

  (ii) signum function in McCulloch-Pitts neuron yields discrete outputs (for logic gates)

  (iii) Gaussian function to models neural tuning curves (e.g. RBF)



- Neural networks are pattern recognition machines
- Neural networks are opaque to analysis – information is distributed across its connection weights

Neural Network Learning

- There are many neural network learning rules
- **MLP backpropagation supervised learning rule** is the commonest:

$$w_{ij}(t+1) = w_{ij}(\mathrm{t}) - \eta \frac{\partial E(t)}{\partial w_{ij}} + \alpha \Delta w_{ij}(t) \text{ where } E = \frac{1}{2}\sum_{i=1}^{n}\left(y^d - y\right)^2 \text{=error function}$$

2nd term is least square steepest descent algorithm, 3rd term is momentum to smooth descent

- Alternatively, weight may be estimated by Kalman filter (BP is degenerate form of KF):

$$\hat{w}(t+1) = \hat{w}(t) + K(t)[y^d(t) - h(\hat{x}(t))]$$

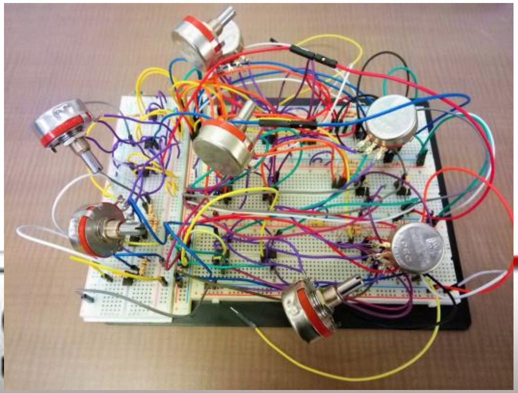
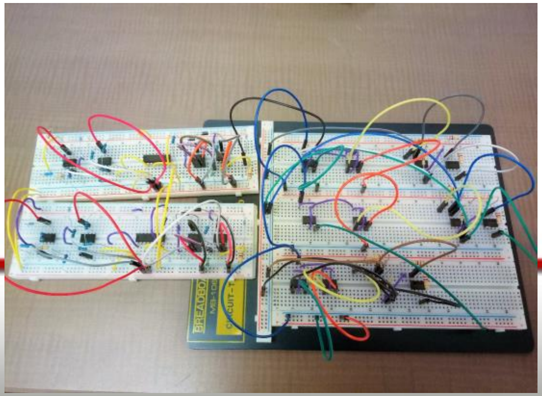
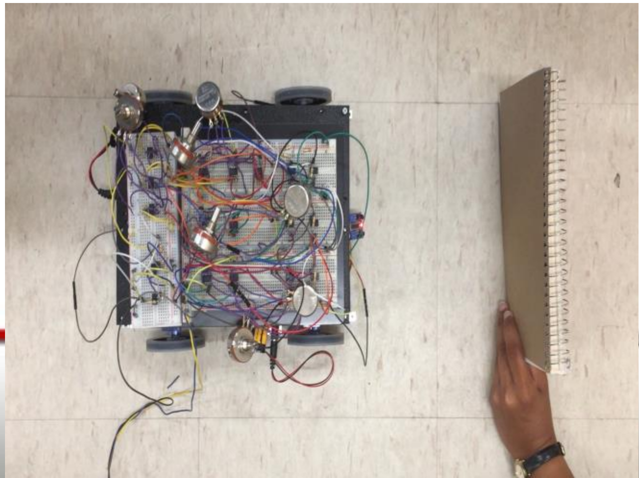
where $[y^d(t) - h(\hat{x}(t))]$ = error between estimated and measured output

$K(t) = P(t)H(t)[(1/\eta)I + H(t)^T P(t)H(t)]^{-1}$ = Kalman gain

$\eta = [H(t)P(t)H(t)^T + R(t)]^{-1}P(t)$ = **adaptive learning rate**

H(t) = observation model, P(t) = state covariance, R(t)=observation covariance

- We have implemented backpropagation in analogue circuitry

Reinforcement Learning

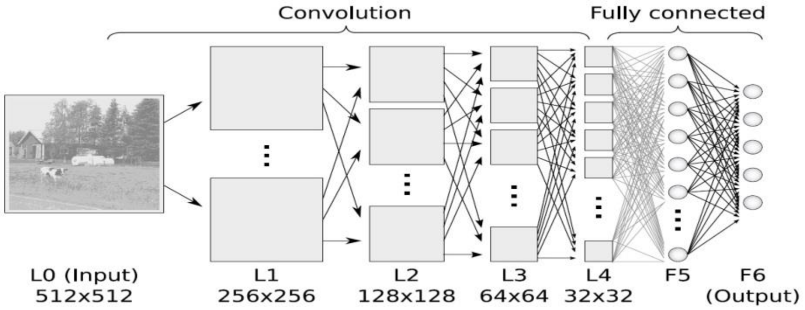
- **Reinforcement learning (RL)** offers continuous learning over time
- **RL** may be non-associative (habituation/sensitization) or associative (conditioning) – it **involves reward/punishment signals from the environment**
- In **classical conditioning** (e.g. Pavlov's dogs), an **innate US-UR reflex** couple of US (food) with UR (salivation) may become associated through training with a learned **conditioned CS-CR reflex** – CS (bell) is predictive of US such that it invokes CR (salivation) that mimics UR even if US does not occur
- Classical conditioning is described by a generalized (unsupervised) Hebbian learning rule (**Rescorla-Wagner theory of classical conditioning**):

  $\Delta w_i = \eta \varepsilon_i (\lambda - y)$ where η=US-dependent learning rate, ε=CS-dependent learning rate, λ=external reinforcement scalar, $y = f(\Sigma_i w_i x_{CS(i)})$=**associative CS-CR strength**
- There are more sophisticated RL algorithms
- **Temporal difference (TD) algorithm predicts future rewards** r(t):

  $r(t) = \lambda(t+1) + \gamma \Sigma w_i(t) x(t+1) - y(t)$ where γ=discount rate
- **Watkins Q-learning algorithm** is simpler than TD(λ) because it is model-free
- Reinforcement learning of deep neural networks in self-play was the **basis of AlphaGoZero**

Deep Learning

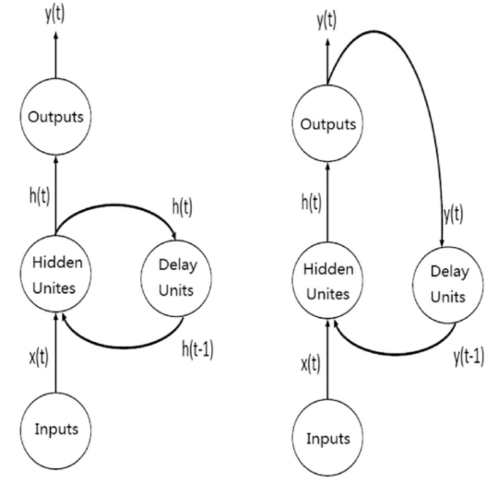
- **Deep learning systems are multiple layered ANN** – typically unsupervised frontend with supervised backend
- **Unsupervised frontend** is trained to perform data clustering on unlabelled input data – each layer is trained sequentially from output of previous layer
- **Supervised backend** is trained with unsupervised front end using labelled data using back propagation
- For example, **convolutional neural network** (CNN) for image processing – linear filter (translation invariance) – max-pooled (downsampling) - classification



- CNN require huge datasets for training by GPU
- Despite high success rates, CNNs are vulnerable to humanly-obvious misclassification errors
- **No significant CNN performance improvement in 25 years**
- Humans can learn pattern recognition in a few images – CNN requires enormous training datasets yet make fundamental errors
- C elegans worm possesses only 380 neurons yet is capable of adaptive behaviour

Recurrent Neural Networks

- **Recurrent neural networks exhibit feedback connections** and are Turing-complete
- Feedback enables temporal sequences to be stored, i.e. **memory**
- Linguistic sequences subject to neural processing, using **Elman network, e.g. predict word sequences**
- ChatGPT is extension of this
- Elman and Jordan recurrent networks have limited sequence memory



Elman Jordan

- **Long short-term memory (LSTM) is RNN** with specialised memory neurons to store long linguistic sequences
- RNN trained using backpropagation-through-time (BPTT) or real-time recurrent learning (RTRL) algorithms
- **Transformer neural network is RNN that processes whole sentences** rather than word-by-word
- **RNN showed that linguistic symbol sequences may be predicted using neural networks**

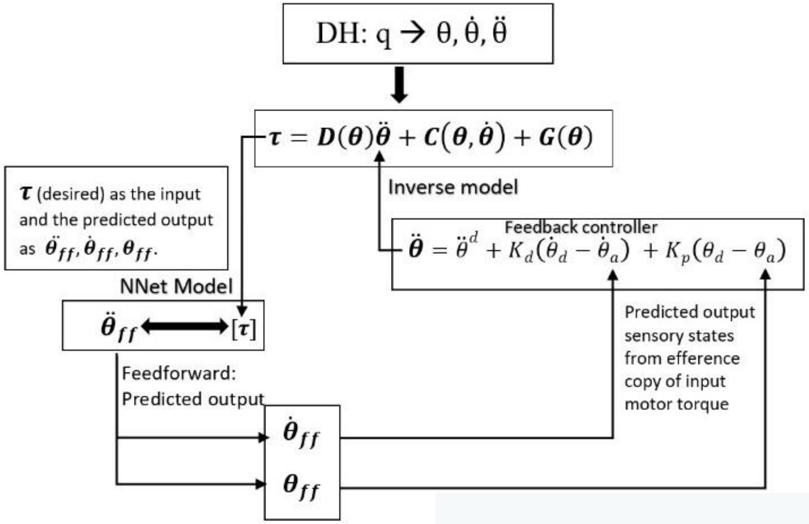
Large Language Models

- **Large language models are statistical models of language** that encapsulate joint probability function of word sequences – conditional probability of next word in a sequence given the prior words: $p(w_T) = \prod_{t=1}^{T} p(w_t|w_{t-T})$
- Statistical dependence between words increases with their proximity, $p(w_t|w_{t-T}) \approx p(w_t|w_{t-n+1})$ for n contexts
- Chat GPT is a transformer RNN model that processes sentences to maximise log-likelihood of word sequences $(w_1,\ldots,w_T)$
- **GPT-3 (2020) neural network** for language processing comprises **175 B parameters approximating number of neurons of the human brain**
- MT-NLG neural network has **530 B parameters but with only marginal increase in performance** against a standard text processing benchmark
- GPT-3 was trained using 1.3 GWh electrical energy - GPT-4 was trained using 25-30 GWh
- GPT is a text processor (a grander version of ELIZA chatbot) that generates plausible text – I queried ChatGPT-3 "How much energy is required to manufacture liquid oxygen and liquid hydrogen rocket propellant?" – its response was that such propellant and oxidiser yields high specific impulse performance, etc – even further qualification yielded the same result
- It has **no understanding of reality**….It is a deadend for AI

Piaget – motor learning (self) must precede symbol learning (environment)

Transfer Learning

- Transfer learning applies learned models from a prior domain to a new domain – it is a form of analogous learning (meta-learning)
- It is assumed that probability distribution of target domain data is similar to that of the source domain $p(y_s|x_s) = p(y_t|x_t)$
- We used **transfer learning methods apply a forward model of a manipulator trained on Earth to same in space** (modelling human cerebellum function)



Forward model: $\ddot{\theta} = D(\theta)^{-1}(\tau - C(\theta,\dot{\theta}) - G(\theta))$

Integrated to predict $\dot{\theta}$ and θ from τ

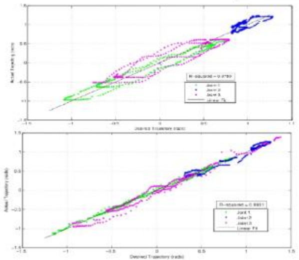
TL from $p = \prod_{i=1}^{n} R_i l_i$

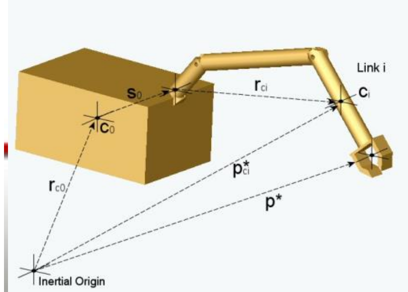
to $p* = C + \prod_{i=1}^{n} R_i \lambda_i$

where $\lambda_i = \frac{1}{m_T}\sum_{j=0}^{i}(m_j l_i - m_i r_i)$

Earth/space kinematics/dynamics possess the same algorithmic form, ANN cannot transfer learn between domains because i/o pairs cannot extract underlying algorithms

ANN cannot generalise beyond their training

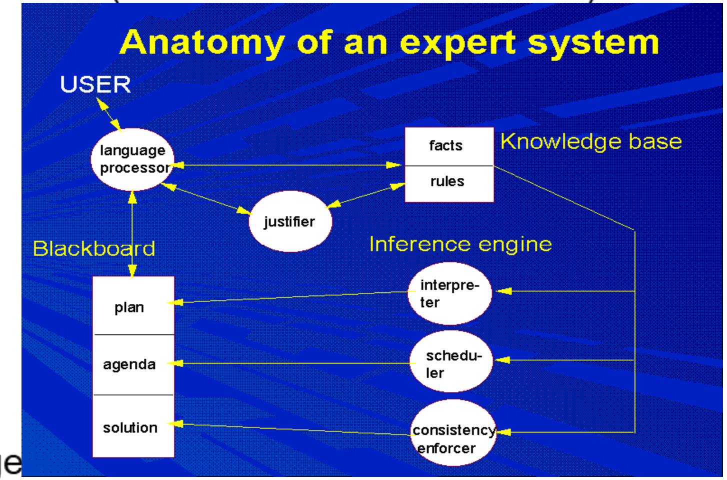




Symbol Manipulation

- Reasoning is an essential property of AI
- Symbol manipulation (GOFAI) is based on logical rules of inference operating on internal model of the world encoded as symbols
- Logic is the formalism that simulates events of the external world as world models
- First order logic is based on single subject predicates with quantifiers (universal ∀ and existential ∃) expanding its expressiveness
- FOL statement has the form $\forall x, y, z[give(x, y, z) \rightarrow own(y, z)]$

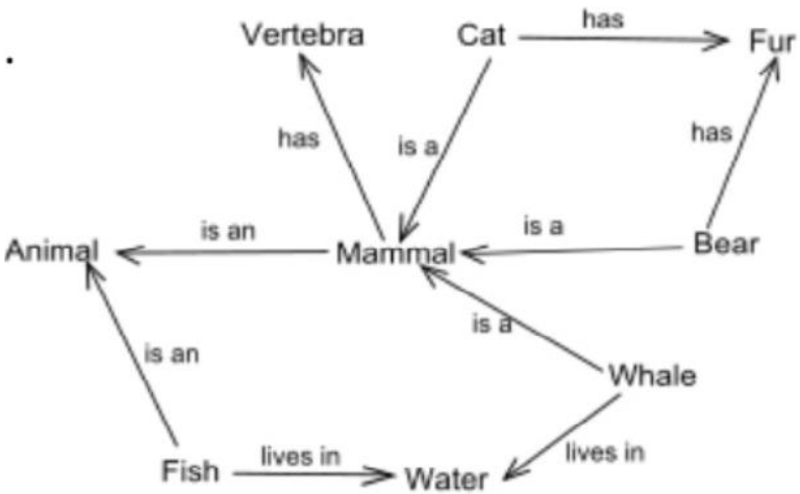
  where x=giver, y=recipient, z=object
- Horn clause logic has a single consequent (e.g. Prolog)
- Expert system comprises knowledge base of production rules of form: “IF (conditions) THEN (action)”
- SOAR is a cognitive model of symbol processing that implements subgoaling for problem-solving
- CYC with 25 M rules attempted to encapsulate “tacit” knowledge



- Large expert systems suffer from large computational footprint, consistency maintenance and brittleness
- Non-monotonic logics – modal logic (possibility), temporal logic (time windows), situational calculus (local restrictions), etc - weaken theorem proving validity of classical logic – RAX (DS1) and ASE (EO1) were temporal logic implementations

Semantic Networks

- Semantic networks are symbolic systems that organize knowledge into taxonomic trees that define relationships between concepts
- They are hierarchically structured categorisations through “is-a” links
- Properties are inherited for compact representation
- They permit exceptions to inherited properties
- Modular semantic networks may be sparsely connected
- Spreading activation between concepts provides a measure of their semantic relatedness



- Searle’s Chinese Room argument asserts that syntactic manipulation alone cannot impart semantic content
- This is so if symbols are not grounded in the real world (e.g. ChatGPT)
- Symbol grounding imparts semantic properties to symbols suggesting that sensors and actuators that directly interact with the real world is essential

Neurosymbolic Processing

- There is little dispute that human intelligence implements **symbol processing on a neural substrate** (there is neurophysiological as well as cognitive evidence)
- Neurosymbolic approaches **map symbol representations such as logical inferences into neural networks** – neural network learns new data – new logical rules are extracted
- There are three major approaches:
- (i) **logical rules mapped into the i/o function of neural networks**, e.g. KBANN
- (ii) weighted logical rules mapped to energy function-based neural networks, e.g. restricted Boltzmann machine
- (iii) logical rules embedded into tensor representation (vectors), e.g. logic tensor networks
- **Recursive auto-associative memory (RAAM) can represent compositional structure of symbolic tree representations as tensors**
- Fuzzy neural networks (FNN) map fuzzy rules into neural networks
- Fuzzy if-then rules are characterized by membership functions [0,1] quantifying degree of linguistic accuracy – they are universal approximators
- ANFIS is a 5-layer feedforward FNN – input layer – fuzzification layer – AND rule layer – weight normalization layer – defuzzification layer – output inferencing layer

Knowledge-Based Artificial Neural Network (KBANN)

- **Knowledge-based artificial neural networks (KBANN) inserts symbolic Prolog rules into the weights of a neural network as prior knowledge** (Towell & Shavlik)
- **Rules-to-network translator involves several steps**:

  (i) disjunctions are expressed as multiple rules to create AND/OR tree mapped into the neural network;

  (ii) neural weight is determined by the number of positive antecedents;

  (iii) add hidden neurons to accommodate more complex rules;

  (iv) add input neurons to extend input range;

  (v) add small random weight connections for full connectivity;
- Neural network can be trained using BP to accommodate new datasets
- There are three approaches to **extraction of new rules** (incl non-monotonic rules with prior pruning) but decomposition such as **M-of-N algorithm is suited to KBANN – certainty factors may be computed from squashed connection weights**
- KBANN is particularly promising because **pre-programming shapes its future learning within boundaries** – this ensures its future behaviour

Bayesian Approach

- Bayes rule defines posterior probability of cause H based on data E:

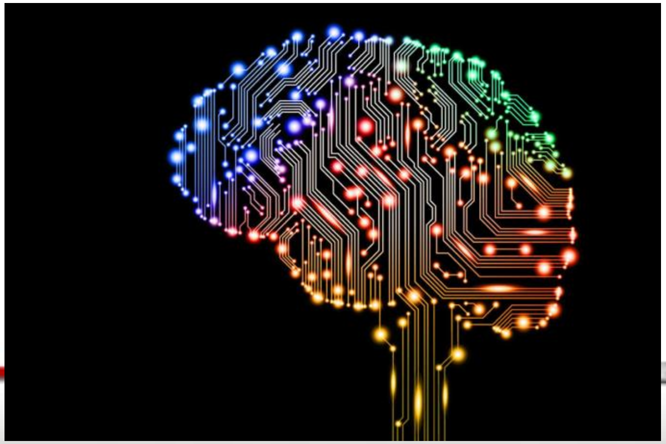
$$p(H|E) = \frac{p(H \cap E)}{p(E)} = \frac{p(E|H)p(H)}{p(E)}$$

p(H)=prior

p(E|H)=likelihood function

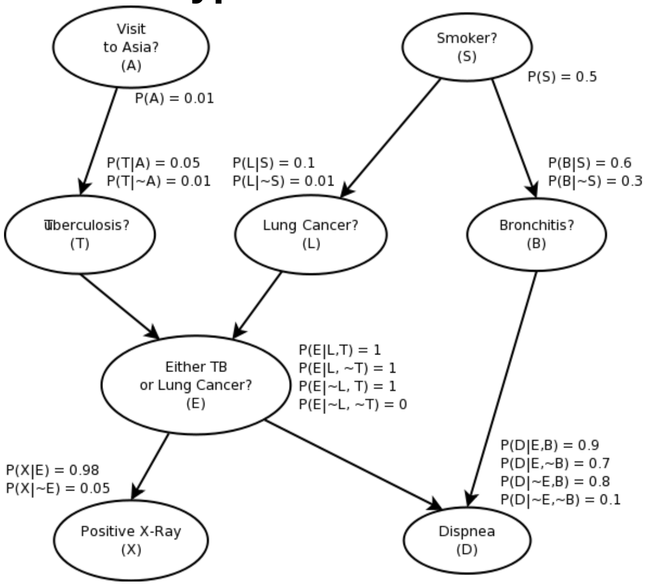
p(E)=p(E|H)+p(E|~H)=probability of evidence regardless of cause

- Bayesian brain hypothesis posits that the brain is a Bayesian inferencing machine that mediates between top-down (prior) and bottom-up (evidence) processing
- Only prediction error (difference between predicted and actual input) is propagated neurally – predicted input is generated by feedforward models
- Bayesian network learning may be implemented through Rescorla-Wagner associative learning of causal relations between objects and events in the environment



Bayesian Networks

- **Bayesian networks (BN) encode conditional probabilistic rules** to represent dependencies (association) between symbols
- **Prolog rules can form Bayesian network** as structured expert system with inheritance links
- **For large BN, Bayes rule is intractable** - approximation through expectation maximization or Markov chain Monte Carlo algorithms (MCMC Gibbs sampling is most common)
- Bayes rule may be deployed to determine **causal relations between hypotheses and evidence – likelihood ratio LR=p(E|H)/p(E|~H)**:
  (i) if LR>1, evidence E increases p(H)
  (ii) if LR<1, evidence E decreases p(H)
  (iii) if LR=1, evidence is irrelevant to p(H)
- **Example**: Appeal (2007) of Regina v Barry George conviction of murder of Jill Dando was overturned on the basis that gunpowder residue evidence had LR~1 because p(E|G)=p(E|~G) and only p(E|~G) was presented at original trial.
- Prosecutor assumed fallacy that p(H|E)=p(E|H)



Bayesian Neural Networks

- Bayesian neural networks (BNN) incorporate a Bayesian treatment to neural networks – they do not integrate Bayesian networks with neural networks
- BNN are stochastic neural networks that introduce an additional term to the error function to accommodate weight uncertainty

$$E = \alpha E_w + \beta E_d$$

where E_d=mean square error in outputs, E_w=mean square error in weights

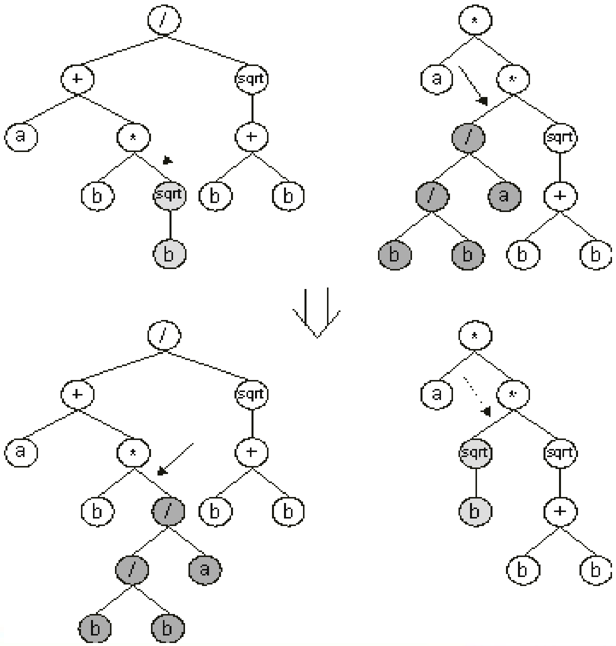
- This prevents overfitting (poor generalisation) in network learning by imposing prior (desired) weights
- Prior is assumed a Gaussian function of the weight normalized across all the weights of the network
- Optimal weights maximise the posterior probability distribution

Markov Logic Networks

- BN may be represented as Markov logic networks (MLN)
- **MLN comprise a set of F first order logic (FOL) rules subject to probabilistic weights (f_i,w_i)** where f_i=FOL formula, $w_i = \frac{p(H|E)}{1-p(H|E)}$ =weight representing joint probability distribution over possible worlds (degree of certainty)
- Weights permit contradictory formulae – if w_i=1, we have pure knowledge base
- MLN inferencing (**weight update**) **involves computing ratio** $\mathrm{p(H|E)} = \frac{p(H,E)}{p(E)}$ using approximate MCMC solutions such as Gibbs sampling
- MLN **structure is learned through inductive logic programming**
- Neural MLN represents rules at potential functions

Genetic Learning

- **Job-shop scheduling** of multiple manufacturing jobs across multiple machines is NP-hard
- Multi-agent coordination may be implemented through genetic algorithms (GA)
- **GA simulates biological evolution** through random search with direction imposed by a fitness function
- Solutions are represented as a population of binary strings (e.g. machine code) subject to mutation and crossover
- **Genetic program** implements high-level computer algorithms (e.g. Prolog) represented as hierarchical decision trees
- Genetic operations include crossover – swapping subtrees
- Programs are subject a fitness function evolved from generation to generation
- **Learning classifier systems** (LCS) have condition-action (if-then) rule format with weights representing fitness
- Bucket-brigade learning algorithm is based on reinforcement learning to allocate credit and update weights



Discussion

- Multiple problems with AI
- (i) ANN neurons are simple switches that have highly diminished capabilities compared with biological neurons - though neural networks based on spiking neurons have been implemented, they are for small networks.
- (ii) A biological neuron has a response time of ~1 ms and human cognitive reaction speed ~100 ms – this suggests that cognitive tasks require <100 neuronal steps….
- Symbol manipulation assumes the human inferencing is logical – there is much evidence that it is not (e.g. Wason selection rule)
- Any symbolic program will be error-prone - average released software has 11 bugs/1000 lines of code (Space Agencies reduce this to 0.11 bugs/1000 lines of code through extensive V&V methods)
- Until recently, it was assumed that human level intelligence would be achieved through scaling to permit brute force computation….
- We have achieved brute force computation – human-level intelligence has not been achieved

Conclusions

- We have several tools but they are deficient….
  - We can employ learning classifier systems to learn if-then rules
  - Bayesian versions of Markov Logic Networks require structuring of these learned rules
  - Symbolic rules must mapped into a structured neural network
  - Probabilistic representations may be into the weights of neural networks
  - How to represent symbols in switching neurons is not clear…though RAAM can incorporate symbol trees, they cannot be manipulated
  - LSTM backend provides sequential processing capability
- We are missing a deep understanding of the mechanism of intelligence….
- It is plausible that AI might be good enough for flyby missions
- It is unclear if class (ii) robotic astrobiological investigations of extrasolar planet encounters are achievable which require scientific hypothesis generation and testing under unknown environments
- Encounters with biological ETI cannot be supported - current AI does not emulate human value reasoning especially human risk-aversion (loss outweighs equivalent gain of same magnitude) - such non-Bayesian asymmetry has sound evolutionary rationale (Darwin awards)
- AI-to-AI encounters will be explored at the next Interstellar Symposium…