Annex A To
AWS for the Land Domain

EXTENDED TECHNICAL BACKGROUND

Artificial Intelligence

AI is a catchall phrase to describe any method by which a man-made object does something smart and acts in order to achieve some goal in the world. In broad terms, AI for robotics can be divided into subsets such as Heuristic Methods, Machine Learning, Natural Language Processing, Reinforcement Learning and Probabilistic Methods.

Heuristic Methods

Heuristic or Rule-based methods are common approaches used to achieve autonomy in robots. The algorithm and its parameters are determined by design based on principles of physics and maths. They are engineered by humans to meet the requirements and specifications that are required by the system. Whilst determinism for known input is shown, most heuristic methods can manifest emergent behaviour, due to the interaction of multiple heuristics or new environments providing unforeseen situations. They are used for control functions, behaviour generation and navigation[1] to name a few applications.

Machine Learning (ML)

Machine Learning covers a large number of algorithms whose parameters are determined by data. There are two broad families based on parameter optimisation:

1. Supervised. Desired outputs for a given input are provided to an optimiser which aims to minimise the error between the outputs of the algorithm and the example data.
2. Unsupervised. The algorithm deals with unlabeled data sets and tries to group or cluster the data so that Like Data is close and Unlike Data is distant.

Deep Convolutional Neural Nets (DCNN or Deep Learning) are mentioned specifically due to the utility in achieving state of the art in many ML tasks. It is important to note that most of the algorithms were developed in the 1980s. Their current resurgence and performance is due to availability of extremely large quantities of labeled data, availability of generally programmable parallel processors. Which makes them accurate and fast to execute compared to other methods.

They are also a universal function approximator in that they can be used to approximate any general function[2], provided enough training examples are provided that accurately describe the distribution of the function inputs to outputs. Outside that training distribution, their

[1] Arkin, *Behaviour Based Robotics*.
[2] Nielsen, "Neural Networks and Deep Learning."

performance is undefined. Typically when training, the learned model is evaluated on a set of inputs not in the training distribution, this is used to evaluate how the model can be expected to perform on new data. They are routinely applied across AI in areas such as Image Processing, Reinforcement Learning, Natural Language Processing. But as a general function approximator they can or have been applied nearly everywhere sufficient labeled data is available.

DCNN can be sensitive to high frequency pertubations in the input signal, so called Adversarial Attacks [3]. Such changes have been shown to cause the DCNN to fail. For example, in an image classification task, it could cause misclassification of the image. Early Adversarial Attacks required knowledge of network architecture as well as the parameters, however there are some black box attacks, which require no specific knowledge. It has been shown that a suitable remedy to innoculate a network against adversarial attacks includes filtering images, and training the network in an adversarial environment.

The methods above are optimisations, where the objective is to try to minimise the error against some metric. There is another method that instead uses a game between a generator and a discriminator to learn the distribution of objects from data. The Discriminator tries to learn to differentiate between true and fake images, minimising the loss from errors in that task. The Generator generates fake images from random noise such that it tries to fool the Discriminator, for an inverse objective. These are the Generative Adversarial Networks (GANs)[4]. They provide a new way of learning a distribution and the concept of a game between the Generator and Discriminator is an architecture that is yet to be fully exploited[5].

Reinforcement Learning (RL)

Reinforcement Learning aims to learn from experience in order to maximise the future reward for a selection of possible actions, given the current state[6]. They can be formulated into many different architectures which work on related concepts[7]. A key part is to try and anticipate the reward for a given situation and action (the Q function). This is usually achieved by measuring the state, applying actions according to the best option given by the existing Q function and measuring the rewards returned. There is a period between sets of these trials in which the Q function estimator is optimised to improve its prediction of the reward. The updated Q function is then applied in another epoch of experiences.

Current RL methods require many training epochs because rewards do not occur after every action, they may only occur after several hundred actions, e.g. when the agent wins, loses or scores a point. This makes them impractical to implement on real robots as the number of examples required exceeds usual physical reliability limits. Most implementations that target

[3] Haohui, “Adversarial Attacks in Machine Learning and How to Defend Against Them.”
[4] Goodfellow et al., “Generative Adversarial Networks.”
[5] Rocca, “Understanding Generative Adversarial Networks (GANs).”
[6] Lambert, “Convergence of Reinforcement Learning Algorithms.”
[7] Lambert, “Gists of Recent Deep RL Algorithms.”

robots do so by simulation, speeding up the time taken to converge, and eliminating reliability issues. As the simulated environments do not replicate the real world environment accurately, most RL policies need to handle the simulation limitations. Usually the cheapest method is by randomising most of the parameters of the simulation, so that when confronted by the real world, its differences to the simulated environment appears as additional noise to the policies it has learned.

Using simulation greatly reduces the number of real-world tuning examples required and, if the limitations of the simulation have been addressed, have produced good results in domains such as robot grasping[8] and robot control[9].

Probabilistic Methods

Probabilistic methods aim to estimate parameters from data not as point estimates with certain values, but as an underlying distribution of possible values, observed at various times as measurements by a sensor. Probabilistic methods estimate both the expected value of the parameter (mean) and the uncertainty in the parameter value (variance). As new information becomes available, it is used to update the estimate for the parameter[10].

These methods are powerful in that they can not only estimate what is known, but also give a measure for how well it is known . This quality of knowledge is given by the variance in the distribution. They do rely on the model being explicitly designed for inference which can make explaining the outputs of the model easier, and can improve the predictability of the outputs. However as these models scale up, dimensionality makes them harder to interpret and compute. However as the uncertainty is explicitly stated, often a simpler model, with uncertainty measured will provide just as usuable a model, as one that is more complicated.

A key benefit is that the models can contain parameters which are important but not directly observed, so-called hidden variables. Other parameters which are observed will have some relationship to the hidden variable as specified by the model. As observations are made, likely states for the hidden variable can be estimated that are consistent with both the observation and the model. A military example might be the estimation of an enemy's location, mission and intent, based on the observation of receiving fire by a particular weapon system in a particular direction, given an enemy situational template, and other prior knowledge.

Probabilistic methods are widely used for robotic state estimation, sensor fusion and map building[11], target tracking to name but a few applications, but is also widely used across

[8] James et al., "Sim-to-Real via Sim-to-Sim."
[9] Arkin, *Behaviour Based Robotics*, 310–20.
[10] Lambert, *A Student's Guide to Bayesian Statistics*.
[11] Thrun, Burgard, and Fox, *Probabilistic Robotics*.

science, politics, economics and social science[12]. These methods are computationally intensive but not easily parallelizable which can make inference computationally intense.

Natural Language Processing

NLP is the processing and factorisation of spoken or written language by a machine. Typically approaches aim to find relationships of words that go together [13]or to predict the next word given a string of words[14]. There are various challenges in processing key terms in natural language for relationships in statements, understanding what that language actually implies and processing in or parsing it into commands that a robot can respond to[15] [16]. Additionally NLP architectures based on deep learning are much more expensive to compute, due to the requirement to keep context and memory, making them hard to deploy to low power environments[17].

More Info

The ICRC has released a thoroughly excellent article[18] on the technical aspects of the application of AI and Robotics to AWS. Addressing whether the current state of the art in AI and Machine Learning is sufficiently robust enough to deploy to AWS. They draw on commercial experiences in the development of self-driving cars, and the ability of the current technology to be certified as being predictable and reliable.

[12] Lee and Wagenmakers, *Bayesian Cognative Modelling*.
[13] "A Beginner's Guide to Word2Vec and Neural Word Embeddings."
[14] Sutskever, Vinyals, and Le, "Sequence to Sequence Learning with Neural Networks."
[15] Tangiuchi et al., "Survey on Frontiers of Language and Robotics."
[16] Matuszek et al., "Learning to Parse Natural Language Commands to a Robot Control System."
[17] Devlin et al., "BERT."
[18] Davison, "Autonomy, Artificial Intelligence and Robotics: Technical Aspects of Human Control."