Beyond Supervised Learning

Dr Benjamin Evans

UNIVERSITY OF SUSSEX

Recap Decision Trees and Random Forests

https://www.pollev.com/bdevans

Are decision trees more expressive than linear classifiers?

Yes

0%

No

0%

It depends

0%

For classification with decision trees, what can we use to measure how effective a node split is?

The false positive rate of the resulting nodes	0%
The efficiency of the split criteria	0%
The purity of the resulting nodes	0%
The number of features used to make the decision	0%
None of the above	0%

In Machine Learning, an ensemble is...

A single model which we run multiple times to aggregate predictions — 0%

A collection of different models where we choose the best prediction — 0%

A collection of different models over which we aggregate predictions — 0%

A collection of different models where we average the model parameters — 0%

None of the above — 0%

How do we train different decision trees within a random forest?

Select different subsets of the data	0%
Select different subsets of the features to split on	0%
All (either) of the above	0%
None of the above	0%

Course Summary

- Probability
  - Bayes' Theorem
  - Probability Density Estimation
- Linear Regression
- Model Selection
  - Sources of Error: Bias and Variance
  - Regularisation
  - Cross-validation
- Data Pre-processing
  - Normalisation
  - Imputing
  - Feature selection / engineering
  - Dimensionality Reduction
- Clustering
- Neural Networks
  - The Perceptron & Gradient Descent
  - Multi-layer perceptron & Backpropagation
  - Deep Learning and Convolutional Neural Networks
- Logistic Regression
- Support Vector Machines
- Decision Trees
  - Ensemble methods & Random Forests
- Beyond Supervised Learning

When poll is active, respond at **pollev.com/bdevans**

Text **BDEVANS** to **07480 781235** once to join

If we group Machine Learning algorithms on the basis of whether they need labelled data or not, they would be:

- Classifiers vs. Regressors
- Unsupervised vs. Supervised
- Regularised vs. Overfit
- Shallow vs. Deep
- None of the above

Beyond Supervised Learning: Outline

- What's beyond Supervised Learning?
  - Why do we care?
- Semi-supervised Learning
  - Weakly-supervised Learning
  - Self-supervised Learning
- Reinforcement Learning

Learning Outcomes

- Understand the principles and benefits of:
  - Semi-supervised Learning
  - Reinforcement Learning

- Be able to propose some simple approaches for semi-supervised learning.

Intelligence as cake!

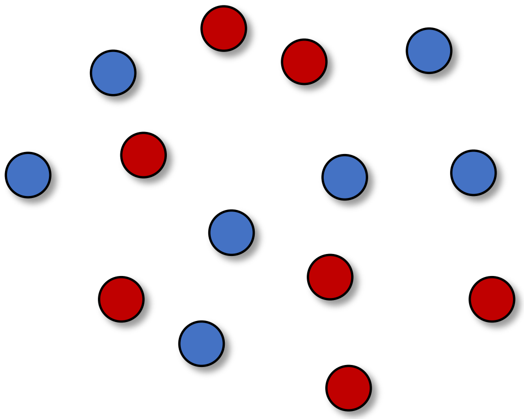
“If intelligence is a cake, the bulk of the cake is unsupervised learning, the icing on the cake is supervised learning, and the cherry on the cake is reinforcement learning (RL).”

--- Yann LeCun

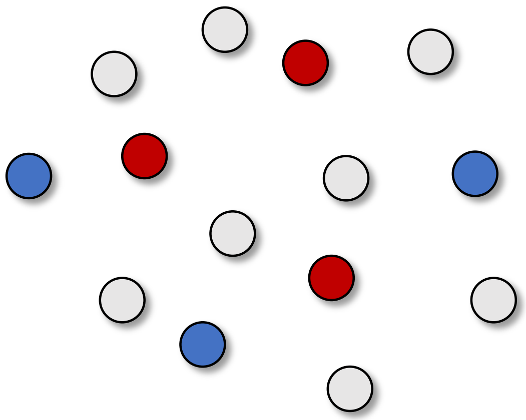
Why do we care?

- Many of the ML models we have looked at are trained through Supervised Learning.
- This is still commonly used e.g. for DCNN image classification models.
- However, manually labelling data (especially audio/video) is very laborious, time-consuming and can be error-prone!
- Case Study: ImageNet
  - Undergrads collecting images at $10/hr would take 90 years to complete!
  - Even using Mechanical Turk it took 2 ½ years to complete (3.2M images)!
  - Now over **14M** images in 21,841 subcategories!

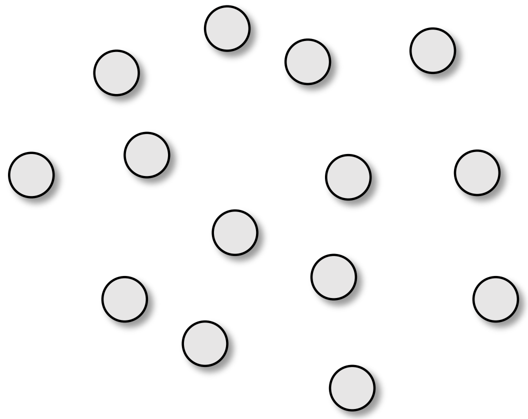
Recap on Types of Learning



Supervised Learning



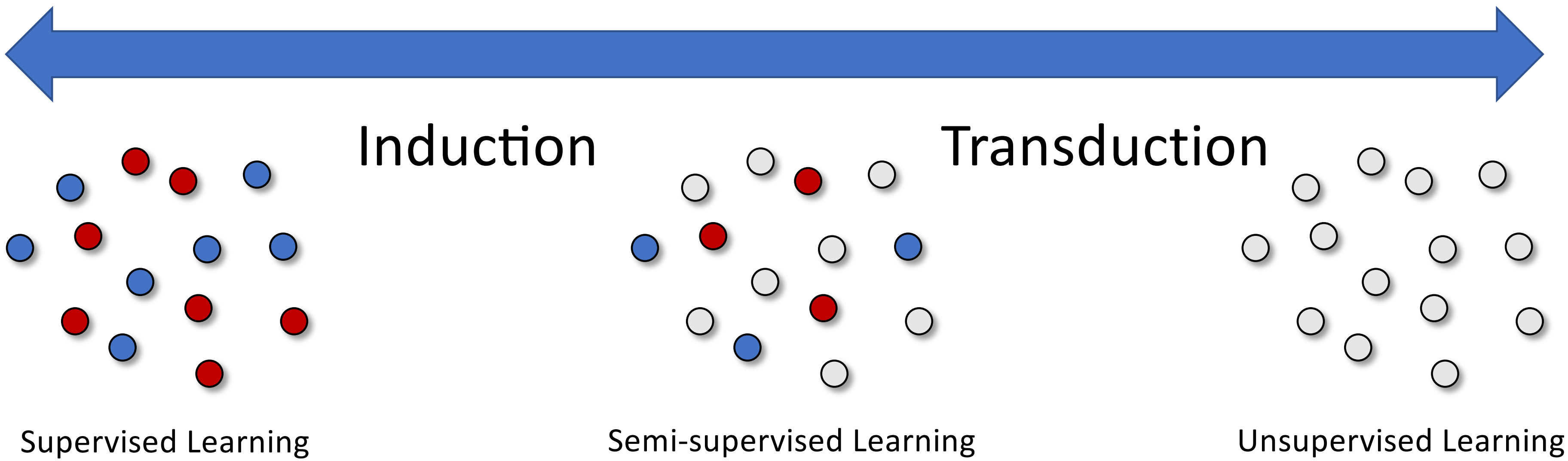
Semi-supervised Learning



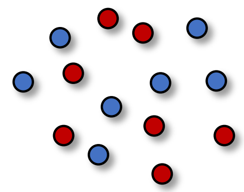
Unsupervised Learning

Continuum of learning modes

- Transductive: reasoning from training data to test data e.g. similarity
- Inductive: infer general rules from training data to apply to test data



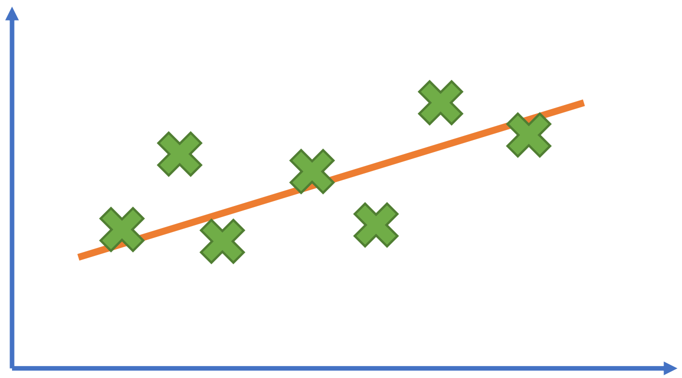
Supervised Learning



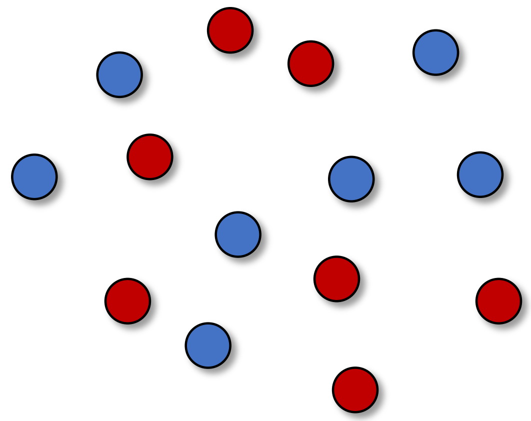
$$f : X \longrightarrow Y$$

- **Example Task 1**: Classify images containing cats or dogs.
  - What do we want to achieve?
  - What data do we require?
  - What data issue might limit the performance?

- **Example Task 2**: Regress salary offer for a job candidate.
  - What do we want to achieve?
  - What data do we require?
  - What data issue might limit the performance?



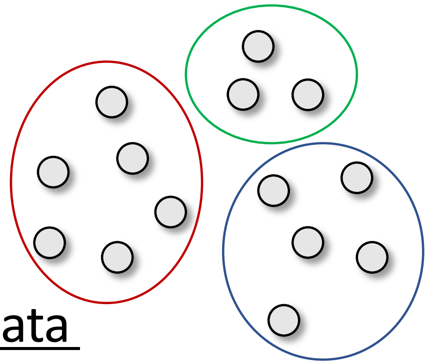
Supervised Learning



$$f : X \longrightarrow Y$$

- What do we want to achieve?
  - Accurate predictions on *unseen* test examples
- What data do we require?
  - Input data and **labels** for all instances
- What data issue might limit the performance?
  - Lack of labelled examples
- **More data would improve performance but we don’t always have labels for every instance…**

Unsupervised Learning



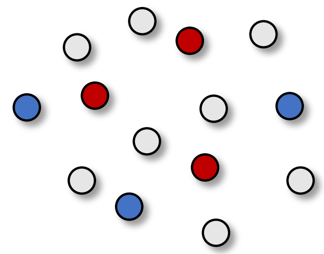
- Unsupervised (sometimes called self-supervised) algorithms learn data representations from unlabelled examples.

- These representations provide an alternative description of the data.

- This is one of the big goals in machine learning!
  - There is a big conference dedicated to this --- International Conference on Learning Representations (ICLR).

- Aims to discover the structure of the data. Applications include:
  - Clustering
  - Dimensionality reduction
  - Anomaly/outlier detection

Semi-supervised Learning



- Although labels may be hard to acquire, generally there are plenty of unlabelled examples e.g. images of cats & dogs scraped off websites.
- What can we always do without labels?
- What kind of learning is always possible?

- One approach → Combine clustering with classification algorithms.

PCA with Classification

- Example Task 1: Classify images containing cats or dogs.
  - Apply PCA on a large set of images.
  - Train a linear classifier on the principle components using a small set of labelled data.
  - The unlabelled data contributes to the PCA, which helps to understand what variations exist in the dataset.
  - This should lead to better performance of the classifier.

Clustering with Classification

- Use a clustering algorithm e.g. *k*-means to identify clusters in the unlabelled data.
- Manually label (or find) representative samples for each cluster (those closest to the centroids).
- Train a classifier on these *k* most representative examples.

- Alternatively, apply the same labels to other members of each cluster.

Clustering with Regression

- Example Task 2: Regress salary offer for a job candidate.
  - Apply *k*-means clustering on all candidates' CVs.
  - Assign salaries to cluster centres based on the labelled data.
  - Predict salary for test-point based on inverse-distance to cluster centres.
  - The unlabelled data helps to understand clusters of similar CVs.

Self Training

- Take a supervised learning classifier that implements prediction probabilities and train it on labelled data.
- `While iteration < max_iterations or len(unlabelled_samples) == 0:`
  - Predict labels for the unlabelled samples.
  - Select a subset of the predicted labels according to e.g. a prediction probability threshold or the *k* highest prediction probabilities.
  - Add the chosen subset of these (pseudo-)labels to the labelled training dataset and retrain.

Label Propagation

- Construct a similarity graph over all inputs.
- Apply the same labels to all other members of each cluster.

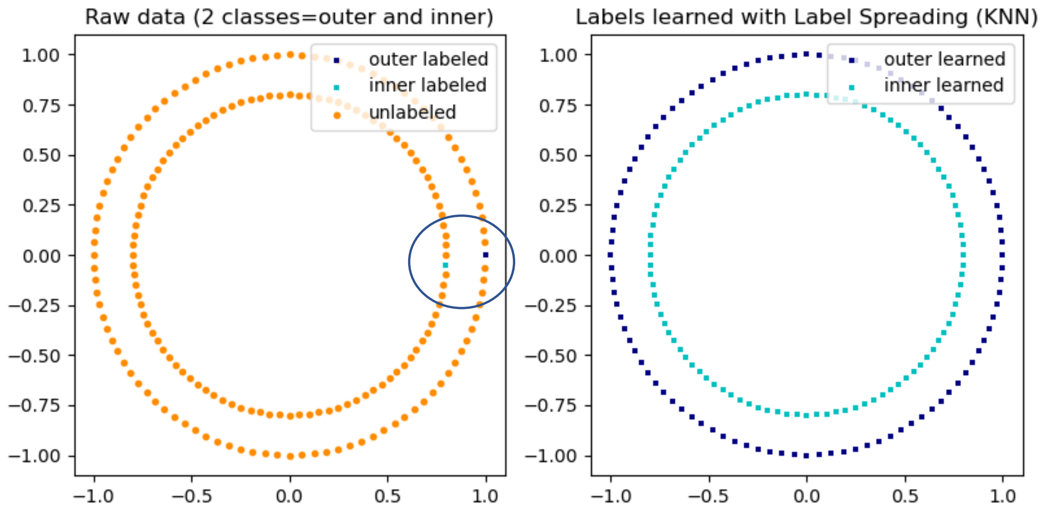


Illustration from scikit-learn documentation: https://scikit-learn.org/stable/modules/semi_supervised.html

Semi-supervised Learning Assumptions

- **Continuity**: Points close together in feature space are likely to be the same class.

- **Cluster**: Points in the same cluster are likely to share the same label.

- **Manifold**: The data lie on a low-dimensional manifold in the feature space.

Semi-supervised Learning Applications

- **Internet Content Classification**: vast amount of data, making it practically impossible to obtain labels for it all.

- **Audio/Video Analysis**: similarly vast amount of data, very intensive task to label.

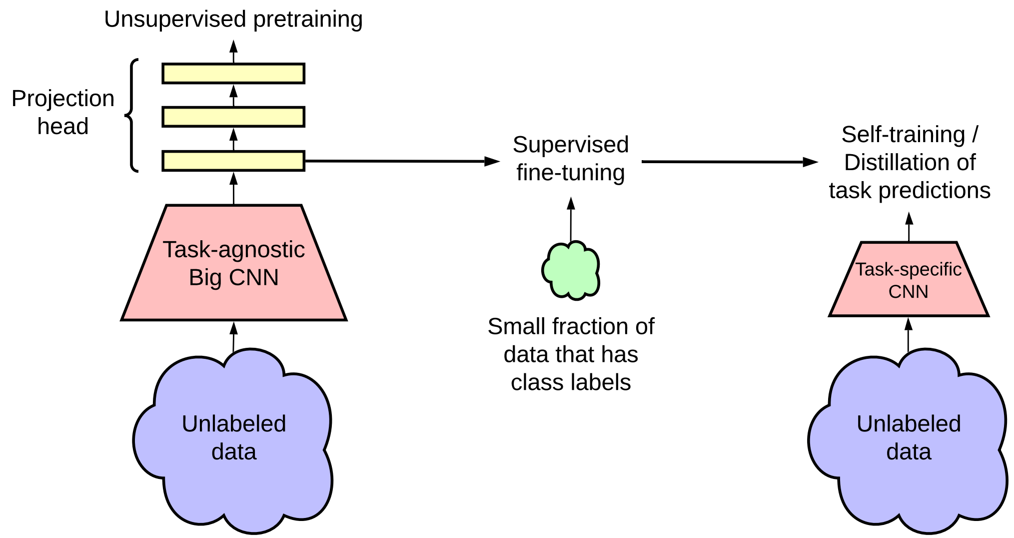
- **Protein Sequence Classification**: Related to one of the hardest problems in science!

Self-supervised Learning

- Interpolate data e.g.:
  - When given a partial sentence, fill in the missing words
  - Reconstruct missing parts of an image
- Learn about the underlying properties of the data through the many unlabelled examples

Self-supervised Learning Example

- Use augmentation to bootstrap unsupervised pre-training.
- Minimise the loss function between augmentations so that the representations are similar regardless of transformations.
- Find a more robust representation of the data using the unlabelled examples.
- Similar approach: autoencoders.



Big Self-Supervised Models are Strong Semi-Supervised Learners. Chen et al. NeurIPS 2020

Self-supervised Learning Example

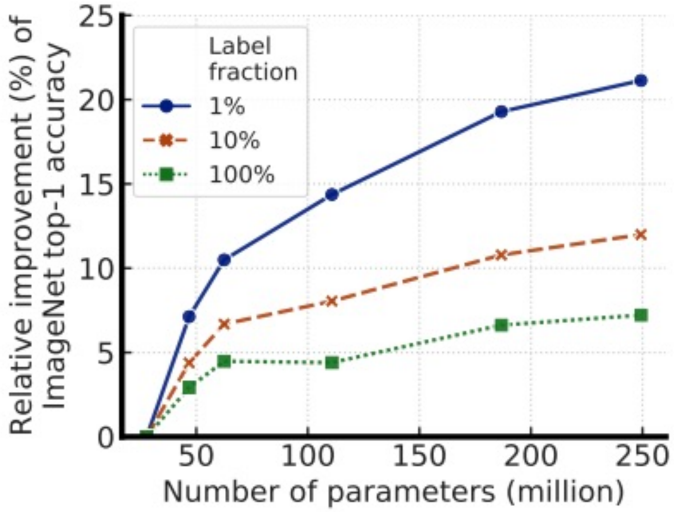


Figure 1: Bigger models yield larger gains when fine-tuning with fewer labeled examples.

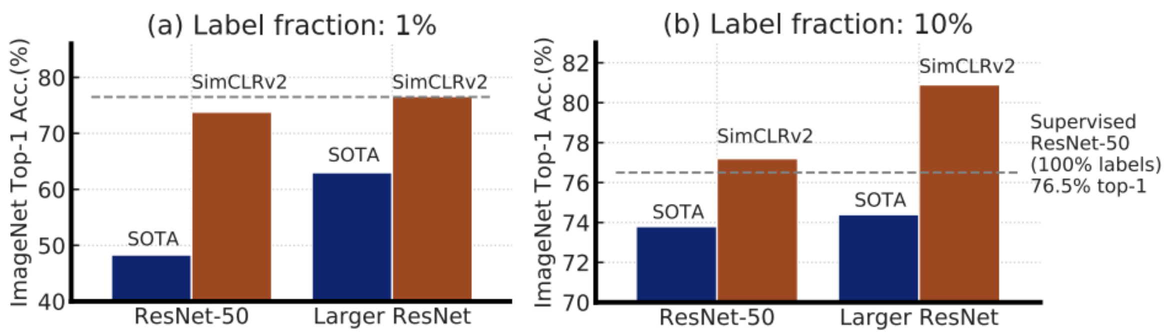


Figure 2: Top-1 accuracy of previous state-of-the-art (SOTA) methods [1, 2] and our method (SimCLRv2) on ImageNet using only 1% or 10% of the labels. Dashed line denotes fully supervised ResNet-50 trained with 100% of labels. Full comparisons in Table 3.

Big Self-Supervised Models are Strong Semi-Supervised Learners. Chen et al. NeurIPS 2020

Weakly Supervised Learning

- Can refer to supervised learning with noisy or imprecise labels.
- "I have some useful information but not exactly what I need."
- Could relate to training with confidence levels rather than hard class labels.
- Could be a way of extracting more specific labels or other information from input data…

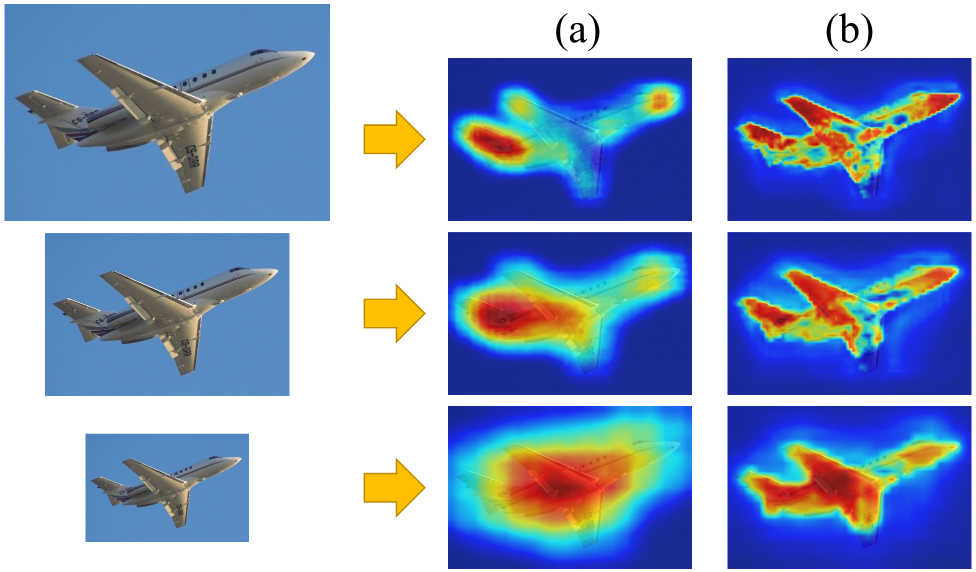


Figure 1. Comparisons of CAMs generated by input images with different scales. (a) Conventional CAMs. (b) CAMs predicted by our SEAM, which are more consistent over rescaling.

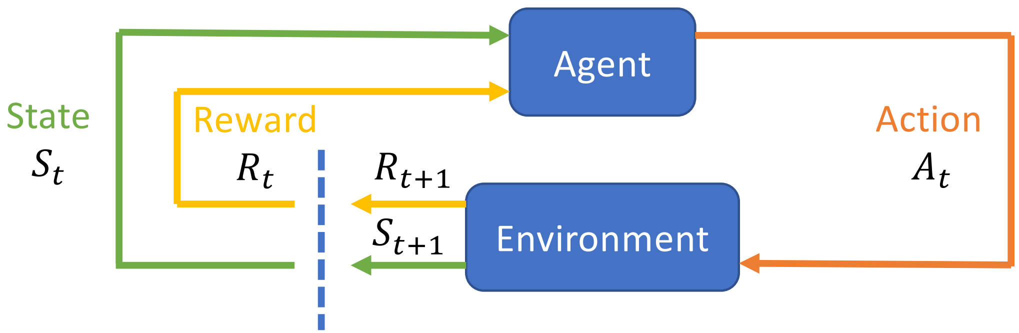
Self-supervised Equivariant Attention Mechanism for Weakly Supervised Semantic Segmentation. Wang et al. CVPR 2020

Weakly Supervised Learning

- Alternative refers to data where only a subset have labels and human annotation is required.
- Algorithm finds the subset of unlabelled data which minimises manual labelling cost while enhancing model performance.
- Two criteria to select the subset of data for manual labelling:
    - Informativeness
    - Representativeness

Reinforcement Learning

- Goal: Learn to maximise a **reward** (or minimise negative rewards).
- An agent explores the **environment** / feature space by taking **actions** (randomly at first) which alter the **state** and receives a reward.
- Uses gradients to adjust its **policy** network, $\pi(s)$, (which maps states to actions) to increase **value** (expected reward).



$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i(s_i, a_i)$$

γ: discount factor; $0 < \gamma < 1$

Types of Reinforcement Learning

$$R_t = \sum_{i=t}^{\infty} \gamma^i r_i(s_i, a_i)$$

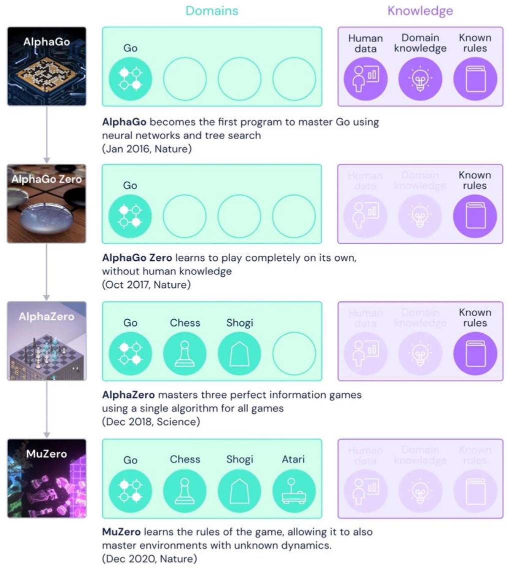
- Markov Decision Process
  - A finite set of states and actions
  - Probabilities and rewards associated with each transition between states
  - Discount factor to weight importance between immediate and future rewards
  - Memoryless: the future is independent of the past given the present
- Q-Learning: Action-Value Methods
  - Q function := E[R|s,a]
  - Policy := select action to maximise Q function
- Policy Learning: mapping states to actions $a = \pi(s)$
  - DNNs (DQNs) successfully applied to learn the policy e.g. for Atari games

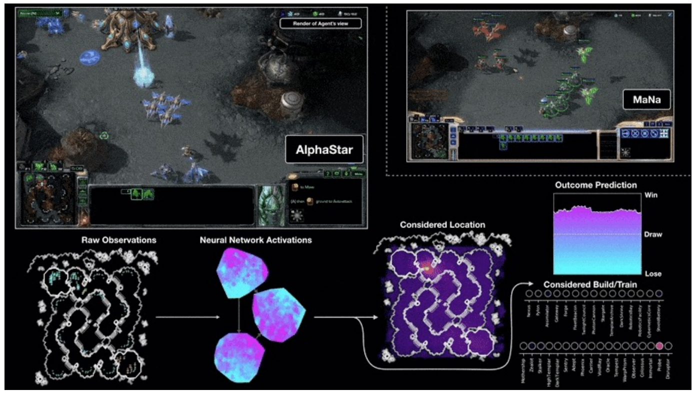
Characteristics

- Closed-loop problems
- Searches through trial & error (or using a *model* of the environment)
- Useful for when there is no known label or when we wish to exceed human performance → go ***beyond supervised learning***! e.g. games
- Issues with delayed reward
  - Which action(s) in a (long) sequence caused the reward (or punishment)?
  - Credit assignment problem
  - May need action shaping to help the model get rewards

Applications

- Robotics
  - Navigation
  - Controlling a production line to optimise yield/cost/quality
- Game Play
  - Chess
  - Go (AlphaGo, AlphaZero, MuZero)
  - StarCraft II (AlphaStar)





Animated figure from DeepMind's Blog
https://www.deepmind.com/blog/alphastar-mastering-the-real-time-strategy-game-starcraft-ii

Summary

- Semi-supervised Learning is most commonly solved by learning an improved data representation e.g.:
  - Weakly-supervised Learning.
  - Self-supervised Learning
- This is important given how laborious it can be to label data manually.
- Reinforcement Learning uses a reward signal rather than labels and adjusts a policy (which maps states to actions) to maximise the expected value of the reward.

End of Fundamentals of Machine Learning!

- I hope you enjoyed the course :)
- Thank you for your attention
- Please complete the Module Evaluation Questionnaire (links on next slide)
- Good luck with your assignments!

Module Evaluation Survey

- Your tutors want to hear about your experiences of your modules this semester: your feedback will help identify what's been working well and where changes might be useful for the future.
- This short survey is anonymous and should only take around 3 minutes.
- Your responses are really valued and we appreciate your time completing this.
- [School Of Engineering & Informatics Module Evaluation Survey]