Lecture 1

Definition of Machine Learning

A Computer Program is said to learn from experience, E, with respect to some class of task, T, and performance, P if it's performance as tasks in T improves, as measured by P with experience E

Tasks, T

Classification

- Construct a function, $f : \R^n \rightarrow { 1,...,k }$, s.t. if an object with features $x \in \R^n $ belongs to class , y, then $f(x) = y$
- Alternatively, Construct a function which given features returns the probability of each class

```
print("Hello, World!")
```

Regression

- Predict a numerical value given some inputs, i.e. a function: $ f : \R^n \rightarrow \R$
- e.g prediction of car value /£ give milage/ miles

Transcription

- Produce Text from unstructured data
- egs.
  - Optical Character Recognition (OCR)
  - Speech recognition

Machine Translation

- Translation from a source language to a target language

Synthesis & Sampling

- Generation of new examples, similar to those in the training data
- Useful in applications where content is expensive to manually produce

Performance Measure, P

- Usually specific to the task, *T* being carried out by the system.
- Accuracy is the proportion of examples for which the model produce the correct output. Equivalent to the **Error Rate**. Often refer to the error rate as **"0-1 loss"**.
- Performance measure must be calculated using **unseen** data to avoid over-fitting.
- For tasks such as density estimation, 0-1 loss doesn't make sense as a performance measure.

Experience, E

- ML often describes *"Nature"* as an unknown probability distribution, *D* over some space e.g. $\R^d$.
- Our "experience" of nature samples from this distribution:
  - i.e $(X_1,...,X_n) \sim D$
- The experience is also sometimes called our *"Dataset"*

Supervised Learning

- The distribution, D is over some set $X \times Y$ where:
  - X is a set of features (e.g. pictures)
  - Y is a set of classes (e.g. cats, dogs)
- A *"teacher"* gives the algorithm labelled examples
  - e.g. a sequence of samples from the distribution which includes elements from both X and Y
    - $((x_1,y_1),...,(x_n,y_n)) \sim D$
- The goal of the algorithm is to predict the class, y, given only the features, x. i.e. to learn/ approximate a conditional probability dataset.

Unsupervised Learning

- Probability distribution D over set X
- We observe some dataset
  - $(x_1, ... , x_n) \sim D$
- The goal of the algorithm is to learn something about the distribution.

Re-enforcement learning

- The algorithm interacts with an environment through a sequence of actions
- each action is rewarded or penalised