COSC343/AIML402

MODULE: MACHINE LEARNING

LECTURE 10: LEARNING

Lech Szymanski

Hand-built vs. learning agent

Two different approaches to building an AI agent.

- Designing the whole agent by hand
- Designing an AI agent which can learn from the sensory data it receives

The agent function

Percepts/Input to Actions/Output

$$\mathcal{P} \mapsto \mathcal{A}$$

$$y = f(x)$$

$$x_1 \in \{\overset{0}{\text{A}}, \overset{1}{\text{B}}\}$$

$$x_2 \in \{\overset{0}{\text{Clean}}, \overset{1}{\text{Dirty}}\}$$

$$y_1 \in \{\overset{-1}{\text{Left}}, \overset{0}{\text{No-op}}, \overset{1}{\text{Right}}\}$$

$$y_2 \in \{\overset{0}{\text{No-op}}, \overset{1}{\text{Suck}}\}$$

$\mathcal{P}$		$\mathcal{A}$	
Room sensor	**Dirty sensor**	**Drive control**	**Suck control**
0	0	1	0
0	1	0	1
1	0	-1	0
1	1	0	1

The agent function as a hypothesis

True function,

$$y = f(\mathbf{x})$$

is unknown.
So, we make a **hypothesis** function

$$\hat{y} = h(\mathbf{x}, \mathbf{w})$$

(Hypothesis will often be referred to as the **model**)

$$x_i \in \{0, \ldots, 255\}$$
$$y_1 \in \{\overset{0}{\text{No face}}, \overset{1}{\text{Face}}\}$$

$\mathbf{x}$			$\mathbf{y}$
x_1		x_{8256}	y_1
73	...	22	1
161	...	30	0
22	...	14	0
211	...	126	1

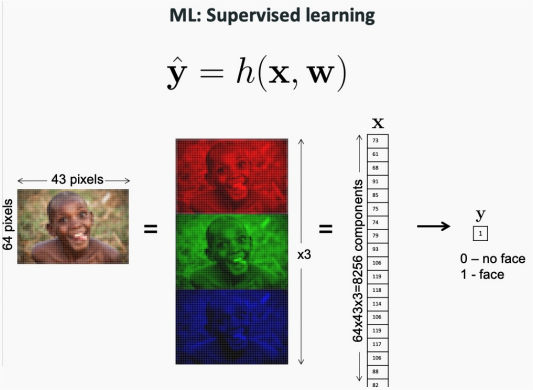
Supervised learning

In **supervised learning**, the agent learns a function from inputs to outputs:

- A sample of labelled data (i.e. inputs and corresponding outputs of the "supposed" true agent function) for a given task is given.
- A parameterised model mapping inputs to outputs is chosen.
- The learning algorithm works out the parameter values of the model that produce the "correct" output for given input.

Supervised learning example

Given a set of labelled images learn the how to classify them.



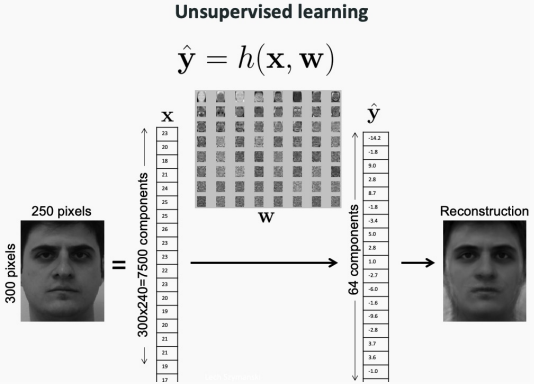
Unsupervised learning

In **unsupervised learning**, the learning algorithm receives a set of training data, and has to work out what regularities it contains:

- A sample of unlabelled data is given (i.e. examples of inputs, but not target outputs).
- A parameterised model is chosen.
- The learning algorithm works out the the parameter values of the model that "organises"/"categorises" data according to some chosen criteria.

Unsupervised learning example

Say we have a set of data, where there is a lot of redundant information; unsupervised algorithm can spot the most relevant and common patterns to retain as the compressed representation.



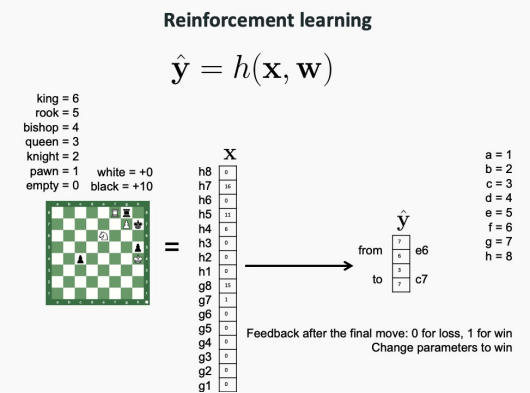
Reinforcement learning

In **reinforcement learning**, the agent receives data and generates actions in response:

- An agent is placed in an environment, which provides a reinforcement signal, a positive or negative reward depending on the effects of agent's actions
- A parameterised model mapping percepts to actions is chosen.
- The agent generates sequences of actions recording their effects through percepts and any received reward(s).
- The learning algorithm works out the the parameter values of the model that generate actions that maximise the rewards for a given situation (inferred from the percepts).

Reinforcement learning example

An agent learning how to play chess needs to perform a sequence of actions (chess moves) in order to bring the game to a close and then, depending whether it won or lost, receives a positive or negative rewards, from which it needs to deduce ‘good’ and ’bad’ actions.



Classification vs. regression tasks

From the modelling point of view, there are two important distinctions of the learning tasks:

- **classification** – those where the model outputs a discrete value from a finite set of choices (related to decision making)
  For example: Given images of two faces – are the individuals related or not?
- **regression** – those that involve the model to output a continuous (real number) value (related to approximation)
  For example: Given images of two faces – what is the genetic distance between two individuals?

Performance metric

In machine learning, we evaluate the performance of a model with a **metric**, which gives:

- the average "distance" of the model's output from the target value, such **mean squared error** (typically used in regression)
- the fraction of the model outputs that are wrong (don't match the target outputs) referred to as the **classification error** (typically used in classification)
  - Sometimes expressed as **accuracy**, the fraction of examples labelled correctly.

$$\text{accuracy} = 1 - \text{classification error}$$

Confusion matrix

Overall accuracy/classification error often hides vital information on the distribution of the errors between classes. A **confusion matrix** provides more information based on grouping of actual labels and the predicted labels. For example, in **binary classification**, where one class can be thought as *positives* and the other as *negatives*, the confusion matrix shows the counts of true/false positives/negatives, from which various measurements can be derived:

True positives (TP)	False negatives (FN)
False positives (FP)	True negatives (TN)

Total number of positives (labels of one type) in the data is TP+FN.
Total number of negatives (labels of second type) in the data is FP+TN.
Total number of positive predictions by the model is TP+FP.
Total number of negative predictions by the model is FN+TN.

Supervised learning: induction

The basic principle behind supervised learning is **induction**. We can define an inductive learning procedure as follows:

- Assume there is some "true", unknown function, f which takes input and returns "true" output $\mathbf{y} = f(\mathbf{x})$
- An indirect evidence of the "true" function is data sample of pairs $(\mathbf{x}, \mathbf{y})$
- Inductive learning process:
  1. Create a hypothesis function $\hat{\mathbf{y}} = h(\mathbf{x}, \mathbf{w})$ with random state $\mathbf{w}$
  2. Compute $\hat{\mathbf{y}} = h(\mathbf{x}, \mathbf{w})$ for current $\mathbf{w}$
  3. If $\hat{\mathbf{y}}$ not close to $\mathbf{y}$, modify $\mathbf{w}$ and go back to step 2.
- Hopefully $h(\mathbf{x}, \mathbf{w}) \approx f(\mathbf{x})$

We want a hypothesis function which **generalises** well to unseen examples of f.

Generalisation

"All generalisations are false, including this one."

Mark Twain

"Essentially, all models are wrong, but some are useful."

George Box

Useful models are the one that perform well on unseen data, i.e. **they generalise**

Consistency and simplicity

A **consistent** hypothesis* is one which agrees with all the training examples.

- It is possible for a consistent model to not perform well on new (previously unseen) data – such model is said to have been **overtrained**, and it **overfits** the data.

There are typically many consistent hypotheses for a given training set. How to choose between them?

- **Occam's razor** tells us to prefer the simplest hypothesis. Simpler solutions tend to generalise better to new examples.

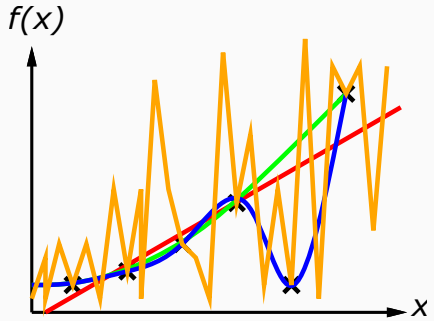
There is typically a trade off between consistency and simplicity:

- We can often get a much simpler hypothesis if we're allowed to ignore a few training examples.

*Don't confuse with consistent heuristic.

Example: Generalisation and curve-fitting

Question: What's the function?



Noise

It would be nice if the training data was perfect...

...but in the real world nothing is perfect - observations are noisy and labels might be inaccurate.

Machine learning algorithms must make useful inferences in the imperfect world.

Training and testing

- Data available for training is usually split in two sets
  - **Training** – data used to train the model
  - **Testing** – data used to verify the performance of the model
- Consistency, good performance on training data, is not necessarily an indication of good generalisation – your hypothesis might be overfitting the data.
- Good performance on test data is a better indicator of generalisation... but only as good as your test set.

Lab 5: Spam filtering

Objectives:

- To review the concepts from Bayesian reasoning
- To implement a Naive Bayes' Classifier
- To apply the Naive Bayes' Classifier to the problem of spam filtering
- To get into the habit of training and testing machine learning models on separated train and test sets

Study guide

- Understand how input/output table relates to a function
- What is a parameterised hypothesis?
- Know the three types of learning feedback (and be able to give examples): supervised, unsupervised and reinforcement learning.
- Understand the inductive supervised learning process.
- Terms and definitions: hypothesis consistency, generalisation, overtraining, overfitting, Occam's razor, discriminative vs. generative models, classification vs. regression, loss function, classification error/accuracy, confusion matrix.
- Understand the need for separating training and test sets.

Reading for this lecture

AIMA Chapter 18, Sections 1–2

What's next?

Decision Trees

Read AIMA Chapter 19, Section 3