Overview of Machine Learning

Click on a question number to see how your answers were marked and, where available, full solutions.

Question Number	Score		
Question 1	1	/	1
Question 2	1	/	1
Question 3	1	/	1
Question 4	1	/	1
Question 5	1	/	1
Question 6	1	/	1
Question 7	1	/	1
Total	**7**	**/**	**7 (100%)**

The pass rate for the questions in the tutorial is 50%, if you score less than this you might want to revisit the questions you had difficulty with and read some of the resources pertaining to that topic.

Thank you for using this tool, in order to improve the system please complete the questionnare linked below:

https://forms.ncl.ac.uk/view.php?id=6719176 (https://forms.ncl.ac.uk/view.php?id=6719176)

Performance Summary

Exam Name:	Overview of Machine Learning
Session ID:	12230462814
Exam Start:	Thu Dec 10 2020 11:56:45
Exam Stop:	Thu Dec 10 2020 11:57:22
Time Spent:	0:00:36

Question 1

What is Machine Learning?

The field of machine learning has gained a lot of attention within the past few years due to the uptake of these approaches within consumer products, as well as the rise of big data.

Machine learning algorithms are utililsed for a number of tasks including:

- voice assistant on smartphones
- online product recommendation
- detecting credit card fraud
- spam filtering for email
- detecting and diagnosing diseases

What is machine learning and how does it work?

Some definitions from the resources below:

> "Instead of requiring humans to manually derive rules and build models from analyzing large amounts of data, machine learning offers a more efficient alternative for capturing the knowledge in data to gradually improve performance of predictive models and make data driven decisions." *(1)*

> "A machine learning algorithm, also called a model, is a mathematical expression that represents data in the context of a problem...The aim is to go from data to insight." *(2)*

> "Machine learning is a category of an algorithm that allows software applications to become more accurate in predicting outcomes without being explicitly programmed." *(3)*

> "Machine learning is an application of AI that provides systems the ability to automatically learn and improve from experience without being explicitly programmed." *(4)*

As you can see from the definitions above, machine learning enables computers to perform complex tasks by learning from data, instead of using pre-programmed rules.

What is a task?

Learning is not the task, learning is our means of requiring the ability to perform the task.

In the following sections we will look at the different types of machine learning including supervised and unsupervised learning.

Resources:

(1) S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

(2) https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9 (https://towardsdatascience.com/10-machine-learning-methods-that-every-data-scientist-should-know-3cc96e0eeee9)

(3) https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08 (https://towardsdatascience.com/introduction-to-machine-learning-for-beginners-eed6024fdb08)

(4) https://expertsystem.com/machine-learning-definition/ (https://expertsystem.com/machine-learning-definition/)

I.Goodfellow, Y. Bengio, A. Courville (2016) Deep Learning. Massachusetts: MIT Press. Online link: http://www.deeplearningbook.org/ (http://www.deeplearningbook.org/)

A machine learning algorithm can also be called a model?

- (●) True
- ( ) False

Expected answer:
- (●) True
- ( ) False

✔

✔ Your answer is correct. You were awarded **1** mark.

You scored **1** mark for this part.

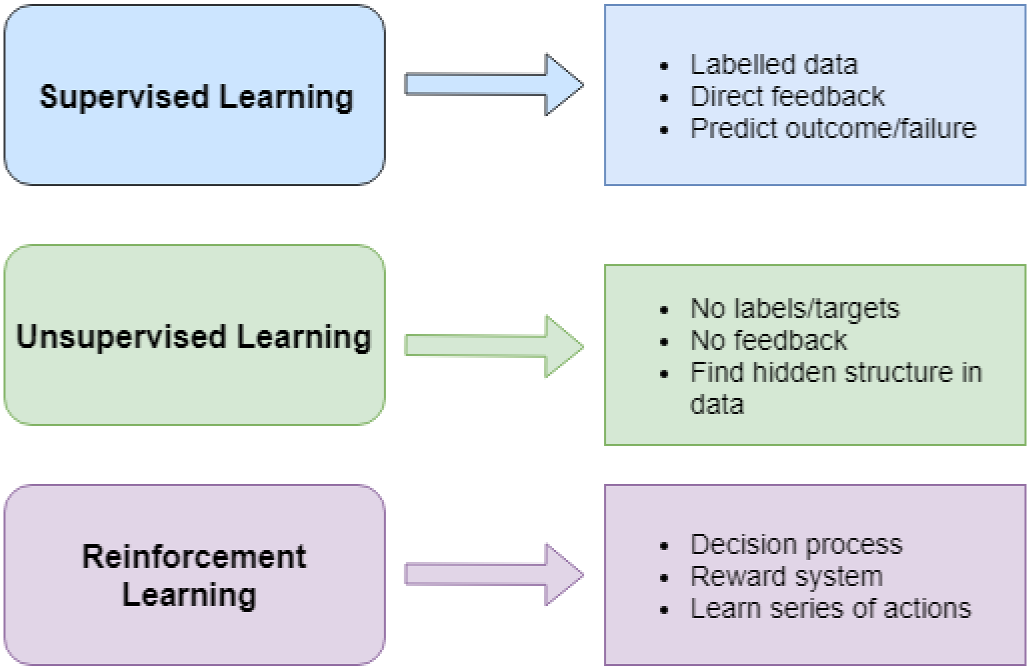
Score: 1/1 ✔

Question 2

Learning Algorithms

There are a number of different types of machine learning- **supervised learning**, **unsupervised learning, reinforcement learning and evolutionary algorithms.**

In this tutorial we will look at the differences between these types to develop an idea of which practical problems are appropriate for each learning type (to keep it straightforward evolutionary algorithms will be covered in the Machine Learning Algorithms section).



Supervised Learning

Supervised learning algorithms learn a model from labelled training data which allows predictions to be made on unseen or future data. For example: email spam filtering, a collection of emails are correctly labelled as either spam or not-spam and the model is used to predict whether a new email belongs to either of the categories.

Examples of supervised learning algorithms:

- Linear Regression
- Naive Bayes Classifier
- Decision Trees
- Support Vector Machine (SVM)

Unsupervised Learning

Unsupervised learning deals with unlabelled data or unstructured data, using an unsupervised learning algorithm we are able to explore the structure of our data to extract useful information without the guidance of a known outcome variable (supervised learning) or reward function (reinforcement learning). Example: marketing teams may use unsupervised learning to discover customer groups based on their interests to develop more personalised marketing materials.

Examples of unsupervised learning algorithms:

- Principle Component Analysis (PCA)
- K-Means
- Autoencoders
- Self-Organising Map
- Expectation-Maximization Algorithm (EM)

Reinforcement Learning

A reinforcement learning algorithm aims to develop a system (referred to as an agent) that improves its performance based on interactions with the environment. The information about the current state of the environment also includes a reward signal, this feedback is not a label (like supervised learning) but a measure of how well the action was measured by the reward function. Through this interaction with the environment, the agent can then use reinforcement learning to learn a series of actions that maximises the reward via a trial and error approach or deliberative planning. Example: a chess engine, where the agent decides a series of moves depending on the current state of the board (the environment) and the reward would be defined as win or lose at the end of the game.

Examples of reinforcement learning algorithms:

- State-Action-Reward-State-Action (SARSA)
- Deep Q Network (DQN)
- Q-Learning

Each of the types of learning will be explained in further detail in the upcoming tutorials.

Resources used:

S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

I.Goodfellow, Y. Bengio, A. Courville (2016) Deep Learning. Massachusetts: MIT Press. Online link: http://www.deeplearningbook.org/ (http://www.deeplearningbook.org/)

You have a YouTube channel and you want to find out more information such as similarities between your subscribers, which type of machine learning would you use?

- ○ Supervised Learning
- ◉ Unsupervised Learning
- ○ Reinforcement Learning

Expected answer:

- ○ Supervised Learning
- ◉ Unsupervised Learning
- ○ Reinforcement Learning

You chose a correct answer. You were awarded **1** mark.

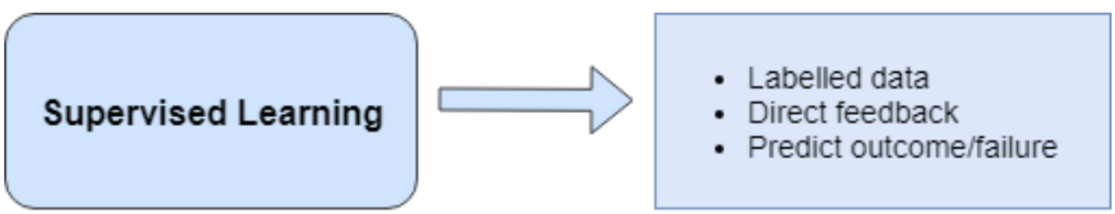
You scored **1** mark for this part.

Score: 1/1

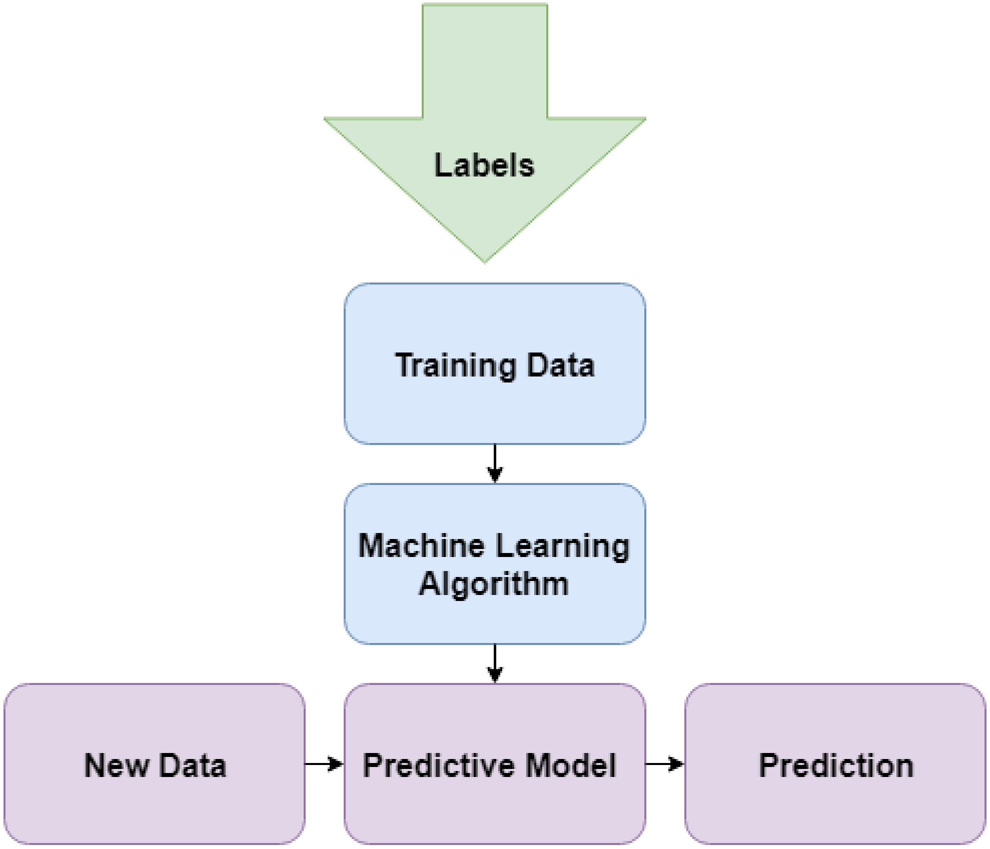
Question 3

Supervised Learning

Supervised learning algorithms learn a model from labelled training data which allows predictions to be made on unseen or future data.



The diagram below shows how a supervised machine learning algorithm is trained using labelled data, which then allows us to make predictions on new, previously unseen data.



We are now going to look at two subcategories of supervised learning - *classification* and *regression.*

Classification

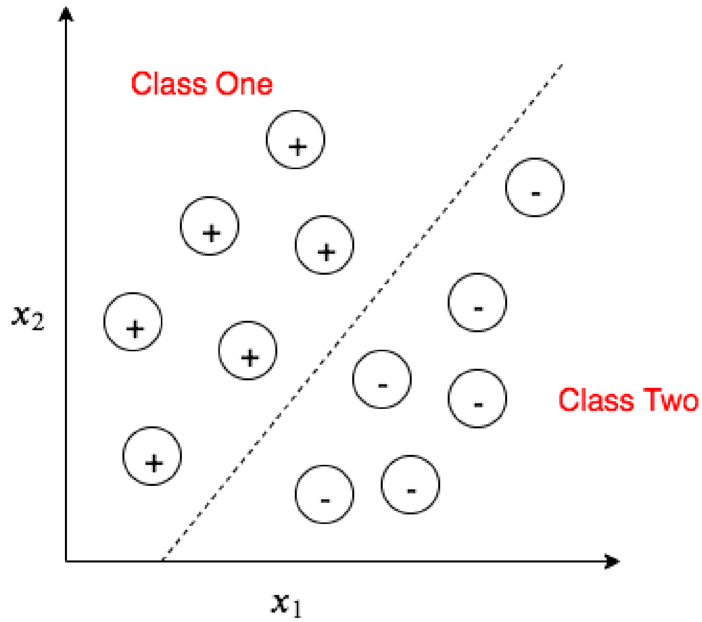
The goal of classification is to predict the class label of new instances, based on past observations.

Common Machine Learning Terminology:

A type of category in a classification problem is called a *class*. Data points are called *samples*. The class associated with a specific sample is called a *label*.

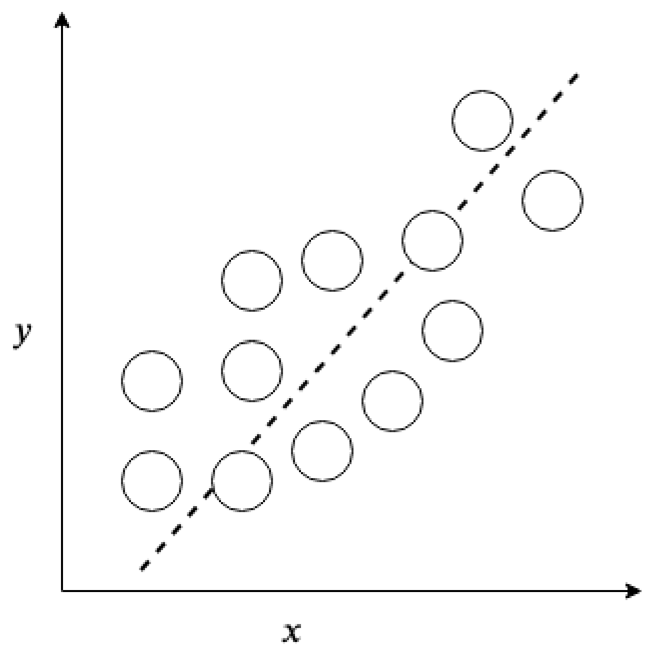
The email spam filtering example in the *Learning Algorithms* section is an example of a binary classification task. The algorithm in this example learns a set of rules to distinguish between the two classes (spam and not-spam).

The figure below demonstrates the concept of binary classification given 12 training examples- 6 training samples are labeled as a positive class (plus signs) and 6 training samples are labeled as negative class (minus sign). Our dataset is two-dimensional meaning that each sample has two values associated with it: x_1 and x_2. We can use a supervised learning algorithm to learn a rule: the decision boundary (represented as a dashed line) that can separate those two classes and classify new data into either of the two categories based on its x_1 and x_2 values.



Regression

Regression analysis is the prediction of continuous outcomes. We are given a number of **predictor** variables (also called explanatory variables) and a continuous **response** variable (also called the outcome or target) and we try to find a relationship between those variables that allows us to predict an outcome.



The figure above demonstrates linear regression. Given a predictor variable x and a response variable y, we fit a straight line to the data which minimises the distance between the sample points and the fitted line- *many of you may have learnt how to fit a line to a set of points whilst at school and the principle here is exactly the same*. We can now use the intercept and slope learned from the data to predict the outcome of new data.

Resources (information from):

S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

F. Chollet (2018) Deep Learining with Python. New York: Manning Publications Co.

I.Goodfellow, Y. Bengio, A. Courville (2016) Deep Learning. Massachusetts: MIT Press. Online link: http://www.deeplearningbook.org/ (http://www.deeplearningbook.org/)

You are asked to predict the height of a growing plant x days after germination. You are given a dataset of 30 measurements of plant age and height. Would you use classification or regression to solve this?

- [ ] Classification
- [x] Regression

Expected answer:

- [ ] Classification
- [x] Regression

✔

✔ You chose a correct answer. You were awarded **1** mark.

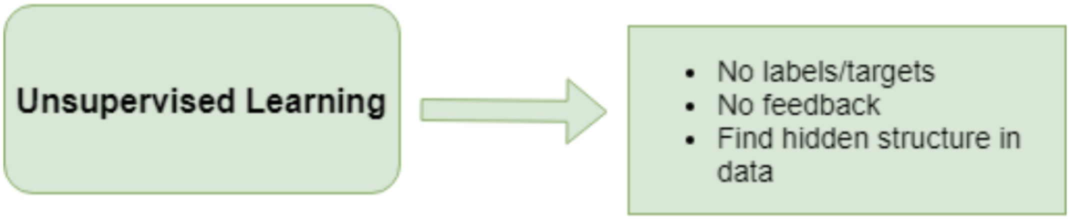
You scored **1** mark for this part.

Score: 1/1 ✔

Question 4

Unsupervised Learning

Unsupervised learning deals with unlabelled data or unstructured data (because the data is unlabelled the learning algorithm has to identify each data point based on its properties/characteristics), using an unsupervised learning algorithm we are able to explore the structure of our data to extract useful information without the guidance of a known outcome variable (supervised learning) or reward function (reinforcement learning).

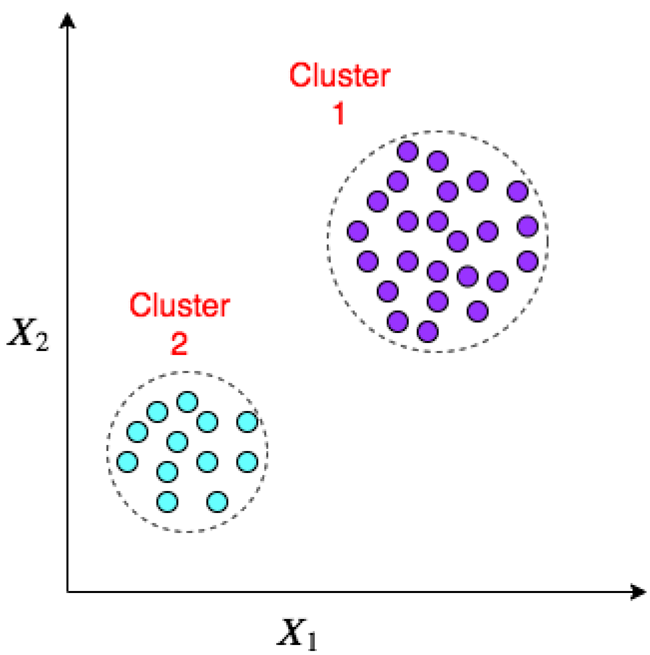


We are now going to look at two different types of unsupervised learning- *clustering* and *dimensionality reduction.*

Clustering

Clustering can be thought of as a form of unsupervised classification, it is a technique for structuring data and exposing meaningful relationships (which of our data points seem to naturally go together).

Clustering is used as an exploratory method for data analysis which creates subgroups (clusters) from our data without any prior knowledge or information about the group memberships. Each cluster created by this technique defines a group of objects that share an amount of similarity but which are more dissimilar to other objects in the other clusters.



The figure above demonstrates how clustering can be applied to organise unlabelled data into distinct groups, based upon feature similarity.

Dimensionality reduction

Dimensionality reduction is often used in feature preprocessing to remove noise from the data as well as reduce the data into a smaller dimension, whilst attempting to keep most of the relevant information.

One of the issues to be aware of is the *curse of dimensionality*, this occurs when we have a large number of features. The more features we have, the greater number of samples we need to have all feature value combinations well represented in our sample.

The model becomes more complex when the number of features increase leading to a greater chance of overfitting (*we will cover overfitting in more detail in later tutorials*).

The most popular techniques for dimensionality reduction are based on linear transformations such as PCA (Principal Component Analysis).

Resources:

S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e (https://towardsdatascience.com/dimensionality-reduction-for-machine-learning-80a46c2ebb7e)

https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/ (https://www.analyticsvidhya.com/blog/2016/11/an-introduction-to-clustering-and-different-methods-of-clustering/)

https://www.techopedia.com/definition/33696/unlabeled-data (https://www.techopedia.com/definition/33696/unlabeled-data)

Which type of unsupervised learning would you use to group your data based upon similarity?

- (•) Clustering
- ( ) Dimensionality Reduction

Expected answer:

- (•) Clustering
- ( ) Dimensionality Reduction

✔

✔ You chose a correct answer. You were awarded **1** mark.

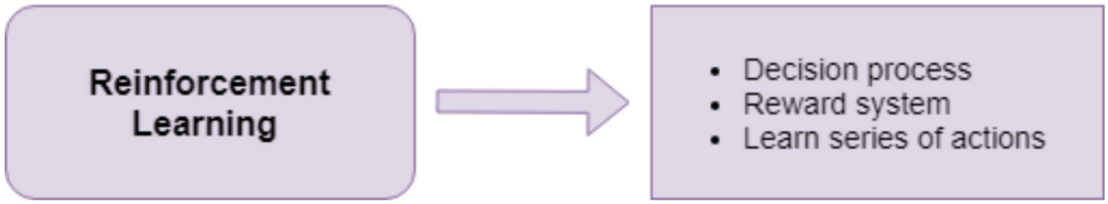
You scored **1** mark for this part.

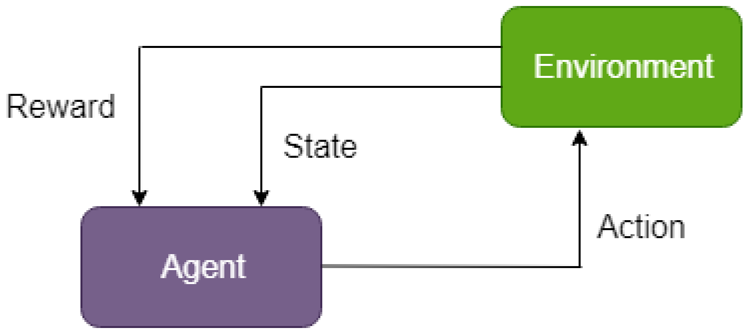
Score: 1/1 ✔

Question 5

Reinforcement Learning

A reinforcement learning algorithm aims to develop a system (referred to as an agent) that improves its performance based on interactions with the environment.





In general, the agent in a reinforcement learning algorithm attempts to maximise the reward through a series of interactions with the environment, each state can be associated with a positive or negative reward.

Reward is the concept which describes feedback from the environment. Reinforcement learning is corncerned with both the short term and long term reward which is learned through many trials and errors and interactions with the environment.

The *state* refers to the current situation, for example in a game of chess, the state would be the position of all the pieces on the board.

The *action* refers to the possibilities of what the agent can do in each state. There are usually a fixed number of actions an agent can take.

The agent learns from experience, it amasses training examples through trial and error as it attempts the task with the end goal of maximising the long-term reward. Its aims to find the best possible behaviour or path for a specific situation.

In a reinforcement learning algorithm the *input* is an initial state from which the model is to start, there are many possible *outputs* due to the variety of possible solutions. *Training* is based upon the input as the model returns a state and it is down to the user to determine whether to reward the model based on its output. The best solution is based on the maximum reward.

Resources:

S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

https://www.geeksforgeeks.org/what-is-reinforcement-learning/ (https://www.geeksforgeeks.org/what-is-reinforcement-learning/)

https://medium.com/machine-learning-for-humans/reinforcement-learning-6eacf258b265 (https://medium.com/machine-learning-for-humans/reinforcement-learning-6eacf258b265)

https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287 (https://towardsdatascience.com/introduction-to-various-reinforcement-learning-algorithms-i-q-learning-sarsa-dqn-ddpg-72a5e0cb6287)

https://www.oreilly.com/radar/reinforcement-learning-explained/ (https://www.oreilly.com/radar/reinforcement-learning-explained/)

A reinforcement learning algorithm improves its performance through interaction with the environment

---

- (•) True
- ( ) False

Expected answer:
- (•) True
- ( ) False

✔

✔ Your answer is correct. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1 ✔

Question 6

Overfitting and Underfitting

Supervised learning can be simplified as the process of approximating a target function (f) that maps input variables (X) to an output variable (Y).

$$Y = f(X)$$

One important consideration when learning the target function from the training data is how well the model *generalizes* to unseen data.

Generalization: how well the concepts learned by the machine learning model apply to examples previously unseen to the model whilst it was learning.

The main goal of your machine learning model is to generalize well from the training data to any data from the problem domain, this will allow us to make predictions on data the model has previously not seen.

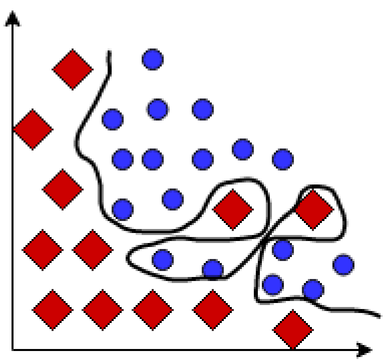
Two of the main issues you are likely to encounter with your machine learning models are *overfitting* and *underfitting.*

Overfitting

A machine learning model which captures the patterns in the training data well but fails to generalize to unseen data is said to be overfitting.

Overfitting can be caused by high variance (too many parameters) and low bias that can lead to a model which is too complex given the underlying data.

The diagram below shows that the line covers all of the points, however this also includes noise and outliers and can lead to a poor result due to the complexity.



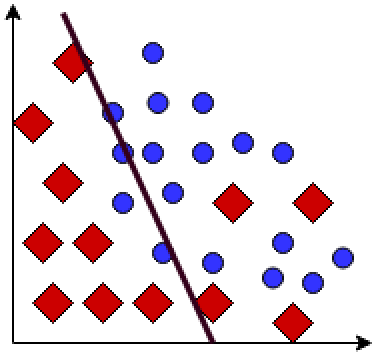
Overfitting

Underfitting

A machine learning model which is not complex enough to capture the pattern in the training data and therefore suffers from low performance on unseen data is said to be underfitting.

Underfitting can be caused by low variance and high bias.

As we can see from the diagram below, we can predict that the line does not cover all of the points. Models like this tend to lead to underfitting.



Underfitting

Bias / Variance Trade Off

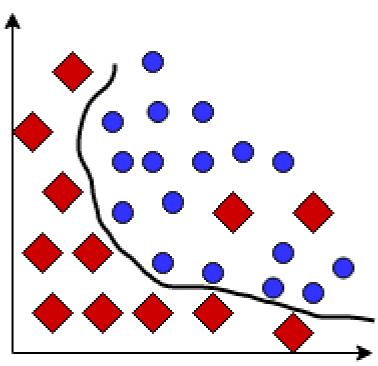
One way to try and avoid encountering overfitting or underfitting is to take into account the bias/variance tradeoff.

Bias: the bias is the difference between the average prediction made by the model and the correct value we are trying to predict.

Variance: a model with high variance tends to pay a lot of attention to training data and does not generalize well to the data which it hasn't seen before.

To build a good machine learning model we need to find a compromise between bias and variance so that it minimises the total error.

As you can see from the diagram below, the predicted line covers the majority of the points and compromises between bias and variance.



Appropriate Fit

Resources:

S. Raschka , V. Mirjalili (2017) Python Machine Learning. Birmingham: Packt Publishing

https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/ (https://machinelearningmastery.com/overfitting-and-underfitting-with-machine-learning-algorithms/)

https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229 (https://towardsdatascience.com/understanding-the-bias-variance-tradeoff-165e6942b229)

https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-machine-learning-and-how-to-deal-with-it-6803a989c76 (https://medium.com/greyatom/what-is-underfitting-and-overfitting-in-

machine-learning-and-how-to-deal-with-it-6803a989c76)

Your machine learning model is performing well on the training data however it is failing to generalize to your new unseen dataset, is your model overfitting or underfitting?

- (selected) Overfitting
- Underfitting

Expected answer:

- (selected) Overfitting
- Underfitting

✔

✔ You chose a correct answer. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1 ✔

Question 7

The Machine Learning Workflow

The aim of this page is to give you a universal plan that can help you try and solve any machine learning problem. It is based on the universal workflow of machine learning by F. Chollet (2018).

Some of the terms you encounter in this tutorial may be new, and only briefly explained here, however they will be covered in greater detail in future tutorials.

The main idea of this tutorial is to give you an overview of the process of creating a machine learning model.

1. Defining the problem and data

- First you should ask yourself the following questions: What will your input data be? What are you trying to predict?
- What type of problem are you trying to solve? Is it classification? Regression? Or clustering? Reinforcement learning? - Identifying your problem type will guide your choice of model

architecture.

You cannot progress to the next stage until you know what data you're using and what your inputs and outputs are. It is important to recognise the hypotheses (assumptions) you are working under at this stage:

- You're hypothesising that your outputs can be predicted given your inputs.
- You're hypothesising that your available data is sufficiently informative to learn the relationship between the inputs and outputs.

These hypotheses will be either validated or invalidated when you have a working model. It is worth noting that not all problems can be solved.

2. Choosing a measure of success

You must define what you mean by success- is it accuracy? Precision? Recall? Customer-retention rate?

This success metric will guide your choice of loss function (this is what your model will optimize) - This will be covered in the Loss Function tutorial.

To give you an idea of types of success metrics and their uses, *accuracy* is a common metric when you have a classification problem where every class is equally likely. For class imbalanced problems, *precision* and *recall* can be used.

3. Deciding an evaluation protocol

Once you have established the success measure you are aiming for, you must think about how you'll measure your current progress.

To give you an idea, here are three common evaluation protocols:

- A hold-out validation set- this is a good idea if you have plenty of data
- K-fold cross validation - this is a good choice if your data is too sparse for a hold-out validation set
- Iterated K-fold validation - this is good for performing high accuracy model evaluation when there is little data

4. Preparing your data

Now you know what you are training on, what you are optimizing for and how to evaluate your approach you need to format your data so that if can be fed into your machine learning model.

Refer to the Data Preprocessing tutorial for specifics.

5. Developing a model that does better than a baseline

You are now ready to begin training your model.

At this stage the goal is to develop a small model that can beat a simple baseline. This step is important to achieve statistical power- if you can't beat a random baseline after trying multiple suitable architectures it might be that you cannot solve your problem based on your current data.

Once you have beat the baseline you need to make three decisions to build your first working model:

- Last-layer activation- this creates useful constraints on the network's output
- Loss function- this should match the type of problem you're trying to solve
- Optimization- what optimizer will you use?

6. Scaling up your model

Once your model has statistical power you need to question whether your model is sufficiently powerful- does it have enough layers and parameters to properly model the problem?

The ideal model sits between underfitting and overfitting.

To determine how big your model needs to be, you must develop it so that it overfits. This can be achieved by:

- Adding layers
- Making the layers bigger
- Training for more epochs

Check the training and validation loss as well as the training and validation values for any metrics which are important. When you see that the performance on the validation data degrades you have achieved overfitting.

You now need to start regularizing and tuning the model to try and achieve a model that neither underfits nor overfits.

7. Regularizing your model and tuning your hyperparameters

This step is the most time consuming- you will repeatedly modify your model, train it, evaluate on your validation data (not the test data yet) and repeat until the model is as good as you can get.

Things you can try include:

- Adding dropout
- Try different architectures by adding or removing layers

- Add L1 and/or L2 regularization
- Try different hyperparameters (e.g. number of units per layer) to find the optimal configuiration
- Iterate on feature engineering: add new features or remove features which don't seem to be informative

Once you feel you have developed a satisfactory model configuration, you should train your model on all of the available data (training and validation) and evaluate it on the test set.

Overview of Terms:

Loss Function - a loss function maps decisions to their associated costs. It is a method for evaluating how well the specific algorithm models the data, if the prediction deviates too far from the actual results, the loss function will be very large. We want to try and minimise the loss function.

Activation Function- an activation function is very important for an artificial neural network (ANN), their main purpose is to convert the input signal of a node in the ANN to an output signal, which can then be used as input in the next layer.

Optimization - this refers to the process of altering a model to get the best possible performance in the training data. An *optimizer* updates the model in response to the output of the loss function.

Resources:

F. Chollet (2018) Deep Learning with Python. New York: Manning

Further resources if you want to read up on loss functions and activation functions:

https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8 (https://towardsdatascience.com/deep-learning-which-loss-and-activation-functions-should-i-use-ac02f1c56aa8)

https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23 (https://towardsdatascience.com/common-loss-functions-in-machine-learning-46af0ffc4d23)

https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/ (https://www.analyticsvidhya.com/blog/2019/08/detailed-guide-7-loss-functions-machine-learning-python-code/)

https://towardsdatascience.com/demystifying-optimizations-for-machine-learning-c6c6405d3eea (https://towardsdatascience.com/demystifying-optimizations-for-machine-learning-c6c6405d3eea)

https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f (https://towardsdatascience.com/activation-functions-and-its-types-which-is-better-a9a5310cc8f)

https://algorithmia.com/blog/introduction-to-optimizers (https://algorithmia.com/blog/introduction-to-optimizers)

What do you need to incur to determine how big your model should be?

- [x] Overfitting
- [ ] Underfitting

Expected answer:

- [x] Overfitting
- [ ] Underfitting

You chose a correct answer. You were awarded **1** mark.

You scored **1** mark for this part.

Score: 1/1