Imperial College
London

AI Basics

Zoe Landgraf

1. Introduction

First introduced at the 1956 Dartmouth College workshop organised by John McCarthy [1], Artificial Intelligence (AI) is now a well-known term and within the last decade has proven increasingly useful across a wide range of fields including healthcare and health research. But what exactly is AI? Described as ‘the thinking machine’ in 1956 [1], the definition we adopt here is that AI is a system giving a practical solution to a problem that currently needs human intelligence to solve.

AI encompasses a variety of subfields such as Machine Learning, Automated Reasoning (logical deduction), Robotics (adjusting movement to dynamic environments), Visual Computation (or Machine Vision), Natural Language Processing (NLP) with machine translation and text generation [2]. There are of course many overlaps between subfields, such as NLP or Visual Perception algorithms using machine learning, or both used in robotic systems.

In the last decade, Machine Learning has received the most attention and investment for several reasons:

- ready availability of large data volumes from increasingly digitalised sectors (entertainment, finance, logistics, healthcare), needed to train intelligent (AI) algorithms
- affordable hardware able to handle the huge calculation loads involved in learning from data.

which lead to..

- demonstrable performance in delivering solutions that rival the performance of human specialists
- successes in delivering algorithms and programmes that human experts could never produce, or could only produce with a lot of difficulty, time and expenditure

AI in use

AI/ML algorithms are slowly making their way into our lives, often going unnoticed by users when deployed to automate simple every day tasks. When you unlock your iPhone with Apple’s latest Face ID technology, it is an AI which detects your face – a Neural Network trained on billions of images to detect faces at varying lighting conditions and angles [3]. Your e-mail inbox is kept spam free using machine learning and most of Google’s products now use machine learning. YouTube uses it to identify inappropriate content, Google Photos to make picture suggestions for friends, and Gmail for sentence completion.

At the other end of the spectrum, AI has been the major or sole component in programmes solving problems beyond advanced human abilities. As well beating world champions at Chess and GO, AI has dramatically accelerated progress in some areas of basic science. For over 50 years the challenge of predicting proteins’ structure from their sequence has been tackled manually, and with conventional and AI computing supported by large and well curated databases of known protein structures. In 2020 the AlphaFold system (by Google DeepMind) achieved such a large advance in performance – with scores above 90% on tasks where other systems and human-computer teams were scoring below 70% - that the challenge was described as “solved”.

What is an AI/ML algorithm?

Algorithms provide a set of instructions to execute a task and existed long before computers came into existence. The first evidence of an algorithm dates from 2500 BC in form of a division algorithm. Computer algorithms give instructions to a computer to process data and produce a certain output.

In AI/ML algorithms, the instructions defining how to process input data to give an output are not defined by a human but learnt from data. These input-output relationships are functions transforming input x into an output y, shown as y = f(x). AI/ML becomes especially useful when these functions are very complex, often with high dimensionality (i.e., many separate parameters that can be quantified), which make them too complex to estimate with traditional methods. In essence, AI/ML methods are very complex function approximators and can be seen as a type of applied statistics.

2. Learning paradigms – the ways machines can learn

Whether based on Neural Nets or other AI approaches, the ways in which AI is developed and evaluated depends on how the system *experiences* the data or environment they learn from. The learning methods can be loosely divided into three categories: **Supervised Learning, Unsupervised Learning** and **Reinforcement Learning**.

In ***Supervised Learning***, each example in the data consists of an input datapoint and the correct label for that input, and from this, algorithms are developed that can correctly label any new input the same way. When training a model to find tumours from CT scans in a supervised manner, an input and label pair would be CT scans without annotation/delineation of the tumour and the same image labelled, with the tumour delineated by a medical professional.

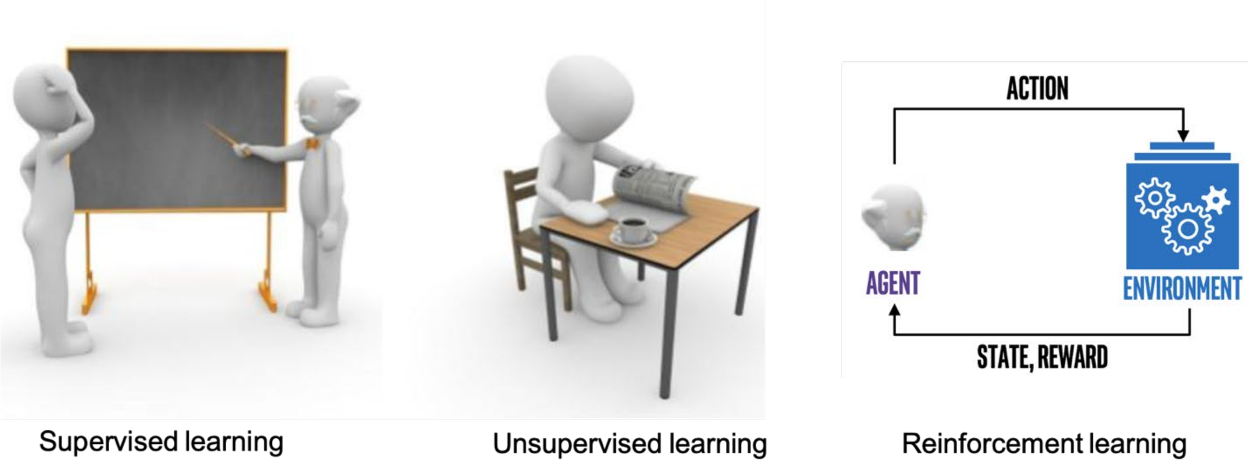

Figure 1: Learning paradigms. Learning can be broadly divided into Supervised, Unsupervised and Reinforcement Learning [4], [5].

During training the correct labels are compared against the system's output to calculate the *prediction error* – and cycle after cycle, an optimiser updates the model parameters so that the network improves its prediction accuracy.

Supervised learning is the most straightforward way to train an AI/ML model, but it requires a lot of **labelled data**, often of the order of 10 000 examples or more. This can be very costly and time consuming to generate. A growing number of publicly available labelled datasets for supervised learning in a variety of general domains are addressing this (e.g., *Wikipedia Links Data* for Natural Language Processing or *ImageNet* for image classification tasks and *Kaggle,* which provides access to over 50 000 public datasets). In healthcare, there are a growing number of accessible de-identified data sets in areas such as mammography or intensive care medicine. The value of the algorithms developed depends on the quality of this training data.

It is important to distinguish between labels reflecting categorical labels (e.g., names) and continuous labels (e.g., temperature, stock prices). Depending on the label type the prediction task is referred to as *classification* (e.g., recognising a person from a photo) or *regression* (e.g., predicting tomorrow's temperature from todays' temperature).

In ***Unsupervised Learning*** the data contains no labels and the model's task is to distil meaningful patterns or features from the data. Most unsupervised methods are either based on *clustering*, - where the data is grouped based on a similarity measure - or on *sample specificity analysis*, whereby every datapoint is treated as a separate class and the algorithm learns a class-to-class similarity [6]. Unsupervised learning can be particularly useful when similarity matters - Google News uses unsupervised algorithms to categorize articles on the same story [7]. In health research it has often been used to provide new insights in personalised medicine, identifying groups of similar patients with similar responses to treatment from very complex combinations of genomics, biomarkers and medical histories.

Application areas

Image processing *(robotic vision)* involves the AI/ML model extracting information from image data – coloured pixels obtained from a camera, for example - and learning underlying features to describe or classify the input data or to use it to make predictions. Image processing is applied in fields including autonomous driving, virtual reality and medical imaging, and tasks include:

Image classification: given an image, the model assigns a label for entire image. e.g., when presented with the scan of a liver, the model would predict 'liver'.

Image segmentation: given an image, the model assigns labels for every pixel in the image, assigning each pixel to a class. e.g., the model finds each pixel which is part of a tumour.

Object detection: the model finds the location of each object in the image, usually predicting the centre of the object (e.g. an infarct) and its dimensions.

Natural Language Processing (NLP) builds models which understand natural language and are able to reason about it. The applications are manifold and include:

Machine Translation: given an input sentence in one language, NLP models translate into another language. This requires an underlying understanding of the grammar of both languages.

Text Generation: based on an input, e.g., an image, a word or a sentence, the model generates text. This is used for image captioning or chat-bots, as well as sentence completion in e-mails and story generation.

Speech Recognition: based on spoken language (soundwaves), the model can execute a command and/or generate a response. Voice assisting applications such as Siri and Alexa are based on Natural Language Processing.

Time-Series Prediction involves predicting the future from past sequential data. Examples include predicting the electricity demand for a city or recognising normal and abnormal EEG or ECG signals.

Dynamic systems modelling tackles more complex models, for systems whose state changes over time – such as climate, disease progression, and the financial market [44]. Machine Learning techniques can not only model dynamical systems but can also *discover* underlying parameters and causal relationships. This can allow AI/ML models to be incorporated into **Digital Twins**, which are real-time highly specific computer simulations of physical entities, first used by NASA for spacecraft simulation. Digital Twins incorporating machine-learned elements are now applied across different industries, for example simulating production processes in manufacturing [44], and in 2020 Babylon Health introduced an AI-powered Patient Digital Twin, to allows users to track and predict risk factors in their own health [45].

Unsupervised learning has the advantage of not requiring labelled training data but designing unsupervised algorithms for deep learning is usually much harder than with supervised algorithms.

Intermediate solutions can be very effective: in ***Self-supervised*** **(or *Semi-supervised or Weakly supervised) learning*,** the dataset is used in other ways – the algorithm uses the labelled data to help learn additional features from the unlabelled data [8]. Many Self-supervised methods learn features based on data transformations, by learning how to recognise that two datapoints which underwent different transformations (such as images that are distorted or viewed from different angles) are the same, which is very important in real world applications. Recently, **contrastive learning** has gained popularity among self-supervised methods. In this method, the model learns how to best differentiate between positive and negative examples in the data [9].

In ***Reinforcement Learning***, the model is trained to make a sequence of decisions to achieve a goal. The model interacts with a environment (such as game or a simulated physics engine) and only receives feedback at the end of the predicted sequence of steps. While feedback (i.e., supervision) is provided at the end in form of a *reward*, no supervision is given during the intermediate steps – so that learning can find indirect or even creative ways of achieving the end goal. Reinforcement learning was behind the success of AlphaGo [10]) in beating the world champion at Go. In healthcare, RL methods are being applied for dynamic treatment regimes in chronic disease or critical care, and in automated medical diagnosis [11]. In healthcare, unlike simulated environments or games, an RL algorithm cannot exploit vast numbers of possible scenarios where it learns how to intervene for the best possible outcome - learning by trial and error with real patients would not be ethical. Instead, RL in healthcare usually uses historical observational data (large data sets of care given and outcomes) [12]. This is referred to as "off-policy" learning – i.e., the model's choices are not acted upon.

Deep Learning is one of the most active research fields within Machine Learning. Although not formally restricted to Neural Networks (see section 3), Deep Learning commonly refers to AI/ML processes which use Neural Networks of three or more layers. Deep learning requires the development of end-to-end solutions for complex tasks using machine-learning processes, rather that limiting machine learning to specific tasks withing a framework set by experts. It offers scope to find new and better solutions but is harder to understand or test.

Other terms

Domain adaptation (or more broadly **transfer** learning) refers to methods that can translate learned relationships to a related but different task. For example, if an algorithm has learned to recognise people from their photographs, transfer learning methods can allow this capability to be used to recognise people from cartoon sketches with only a small amount of adaptation (without having to retrain from scratch on cartoon sketches). A related method is **multi-task learning**, whereby an AI/ML algorithm is trained to solve many different tasks. This allows it to learn very general parameters which are useful for multiple tasks and generalise more easily to new tasks. ***Meta-Learning*** refers to algorithms which *learn* *how to learn*, such as **few-shot** and **one-shot learning, which** address the challenge of learning to learn from a very limited set of examples.

Bayesian Deep Learning*:* most machine learning is based on Frequentist statistics where one set of model parameters is learned and used to make predictions. In Bayesian statistics the uncertainty about the parameters is considered, and Bayesian methods offer advantages when the available training data is limited [13].

The quality of training data: AI/ML models are very sensitive to the properties of the data they are trained with. It is important that training data is *representative* i.e., free of *bias* and *correctly* labelled.

A simple and common form of *bias* arises through class imbalance e.g., the dataset of CT scans contains many more images without a tumour than with tumours. Without the proper adjustments during training, an AI/ML model trained on this dataset will be biased towards predicting 'no tumour'. Gender or ethnic bias can also arise : for example, if a disease is diagnosed later and less effectively in disadvantaged groups of people, AI can wrongly learn that with the same lifestyle and genetics, there is a lower risk for the disadvantaged groups.

Data can be *unrepresentative* if it reflects only a subset of the population, if it comes from a different geographical region or health-care system, or if it is out of date. AI trained on data from early in the COVID-19 pandemic in 2020 might not be useful in 2021, if vaccination status, treatments available, and virus strains change the relationship between patients' medical status and their outcomes.

Synthetic data: Collecting real data is very time consuming and often not possible (e.g., obtaining data for patients with a very rare disease). Furthermore, real health data is often held in decentralised locations requiring either complicated decentralised training methods (*federated ML)* or data transfer which incurs privacy and security risks*.*

Instead, AI/ML models can be trained on synthetic data generated by a computer algorithm, offering a near infinite amount of data to train on and allowing experts to control the dataset statistics for bias-free data.

For example, the Simulacrum [46] contains fully synthetic data (not anonymised data) created from UK cancer data sets (Public Health England's National Cancer Registration and Analysis Service). The data creation algorithms retain the characteristics and relationships found in the original data and models trained on the synthetic data can be applied to, and perform well on, real data [46].

Federated Learning (FL) is a decentralised machine learning setting where multiple agents – which could be large organisations holding regional health data, or individual mobile devices - train a model collaboratively. Although it calls for interdisciplinary efforts in areas such as distributed optimisation, differential privacy and cryptography [47], FL provides key advantages in data privacy. Keeping the data locally mitigates many of the security and privacy risks involved in assembling large centralised sets of training data.

A variety of federated learning models are possible – for example, each location can share locally trained models with a hub which integrates the learnt elements, or locally trained models can be swapped between centres and given further training, through several cycles.

A recent international collaboration released a highly accurate AI/ML model trained using FL, and demonstrated that FL can achieve performance that matches central training, while greatly enhancing privacy, and allowing more organisations to use their data for large scale AI projects. According to the main author: "To keep patient data safe, it should never leave the clinic where it is collected" [43].

3. From Decision Trees to Neural Networks – An overview of AI Models

While there are several ways in which AI/ML models can be grouped, here we will separate them into Classical methods and Neural Networks.

Classical Methods : among the simplest AI/ML methods is ***K-Nearest Neighbours (KNN***), a supervised method, whereby the class or continuous value of a datapoint is estimated based on the average of the class or value of K matching datapoints from the training set. This method is versatile and can be applied for regression and classification tasks but becomes slow for a large dataset.

Clustering techniques group datapoints together in 'natural' groups [14] according to a similarity measure (e.g., their proximity), without the help of explicit labels.

Among the most common clustering algorithms is ***K-means***, which forms clusters by minimising the variance within each cluster. This requires the number of clusters to be known in advance, but alternatives such as ***Mean Shift*** find the right number of clusters as part of the process[i].

Support Vector Machines **(SVM)** offer more advanced ways of grouping or classifying datapoints. While clusters that can be separated by a line, curve or 2-D plane are easy to imagine, it is also possible to define a 'hyperplane' in 3, 4 or more dimensions that can separate the groupings in the data better. The SVM algorithm calculates a hyperplane that maximises the separation distance between the data points in each class (the support vectors) and the hyperplane [15]. The complex non-linear functions that produce these for the training data are then used to classify new datapoints. SVMs are commonly used for supervised learning but have been extended to unsupervised clustering approaches [16]. They can be applied for classification as well as regression tasks.

Classification and regression: Classification places data in categories and offers discrete values – for example defining a region in a CT scan as tumour or not tumour or scoring disease severity. Regression estimates continuous values – for example estimating probability of developing diabetes within a certain period.

Decision Trees classify datapoints by building a branching tree in which each 'node' (branching point) represents a decision point – such as separating the cases with BMI <20 from those with BMI>20. For machine learning, different parameters and threshold values are assessed for each node until a solution is found that sorts inputs into 'leaf' nodes which provide the classification desired.
Decision trees are best suited for classification tasks [2]. The depth of the tree (i.e., the number of branch and leaf nodes) is an important hyperparameter to choose, to get the right balance between generality and precision. Trees are robust to missing input data but prone to overfitting. Decision trees are inherently easy to analyse and explain and their extension into Random Forests (see below) increases their applicability to complex tasks in real-world settings.

Random Forests consist of multiple decision trees, which together form a model which generates more accurate and stable predictions than any individual trees. Multiple, decorrelated decision trees each generate answers for each input, and either the average or mode is used. During training, diversity among the decision trees within the 'forest' is ensured by *bagging* – each individual tree in the forest uses a random subsample from the data during training [17]. Random forests are primarily classification models but have been used for regression tasks [18], and proved very useful in medical imaging, where they were the first ML-method to outperform the previous atlas-based approaches using image registration for organ segmentation and detection. Their uses include Detection and Localisation, Image-based prediction and Semantic Segmentation, where Random Forests made their largest impact [18]. Despite the recent trend towards deep learning models, Random Forests are still one of the most powerful AI/ML methods for real-world applications. They are easy to implement and train and compared to deep neural networks, they are more interpretable.

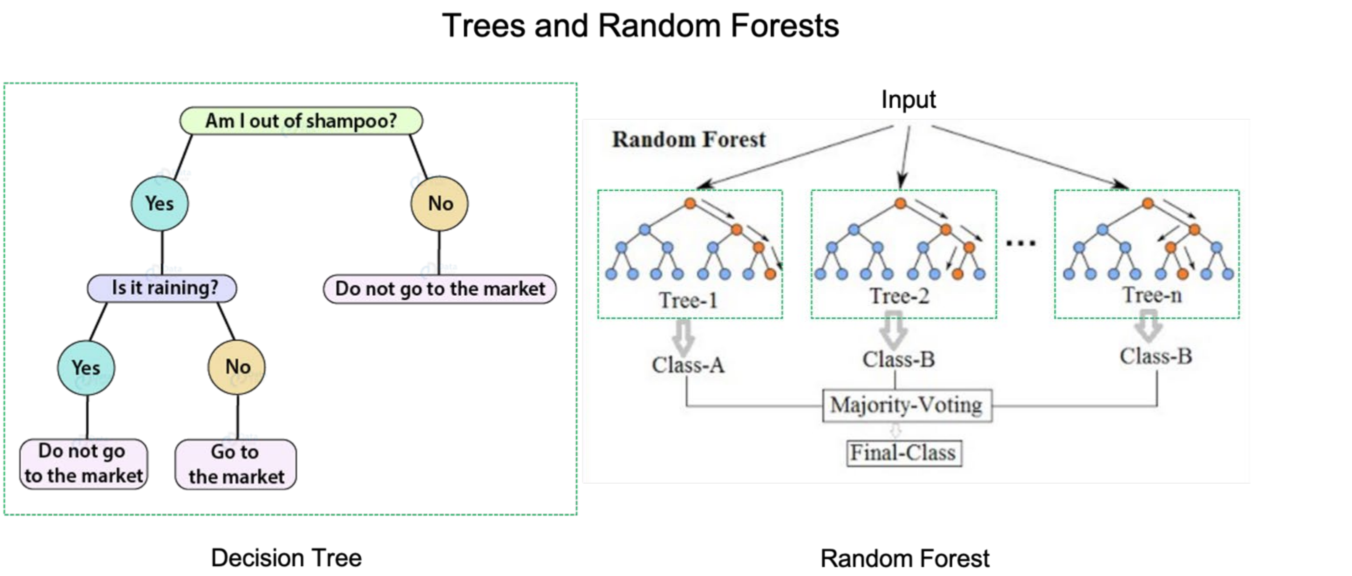

Figure 2: **Left**: An example of a Decision Tree [19] **Right**: An illustration of a Random Forest [20]

Graphical Models are probabilistic models for representing relationships and complex interdependencies (e.g., gene variant X causes illness only if gene Y and lifestyle Z are also present). Although some graphical models are very similar in structure to decision trees, graphs provide an extension to trees: their *causal relationships* can be directed, undirected (bi-directional) and even cyclic (see Figure 3). Graphs therefor have more representational power compared to trees. A graph is typically a network of 'nodes', representing variables, connected by 'edges' representing possible relationships between them. These features can be partly specified by experts, with machine learning finding the model that best represents the data. Graphical models can be very useful where *causation and explainability* are important, and variants include *Naïve Bayes classifiers*, *Markov Random Fields*, restricted *Boltzmann Machines*, *Factor Graphs* and *Belief Networks*.

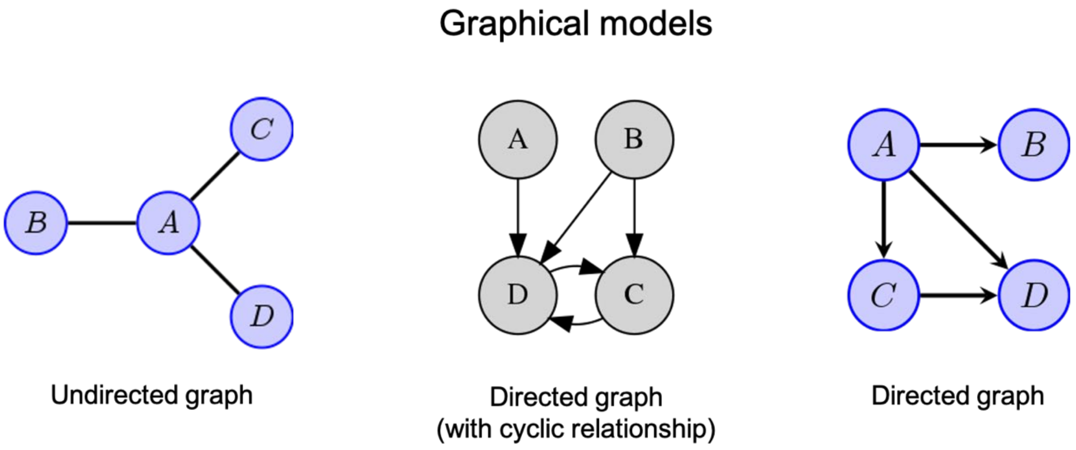

Figure 3: Graphical models with undirected relationships, directed and cyclic relationships [21].

Causal analysis and causality in Machine Learning

Training AI/ML models can be viewed a form of statistical analysis: aiming to infer parameters of a distribution by sampling from that distribution (the samples are examples from the dataset) to the able to predict future quantities. All of this depends on the assumption that conditions remain constant. Causal analysis goes one step further – it aims to learn how one factor affects another and make predictions under conditions that change [48]. Going beyond learning a distribution from data, causal analysis aims to answer questions such as *Why?* And *What If?* and understand which feature or input to change in order to achieve a specific result.

Conventional Machine Learning methods are built for pattern recognition and correlational analyses but are insufficient for causal analysis. Classical AI/ML models such as decision trees and Bayesian networks offer some insight into cause and effect: the final output is directly related to the features (thresholds) at the individual tree nodes. Neural Networks however, are often referred to as **'black box'** algorithms, as they are much less interpretable, and the absence of causality is widely seen as a key limitation of deep learning models. In Machine Learning, inferring causality remains an open question but active research field.

Causality is directly related to the central challenges in AI: robustness, generalizability, explainability, trust, bias and fairness [52], [51], and could be particularly valuable in increasing trust and usability in medical applications of AI. Causal understanding could improve handling novel situations as it makes it easier to repurpose acquired knowledge to new domains [50]. According to the authors of the recent publication 'Towards Causal Representation Learning', "Generalizing (…) requires learning not mere statistical associations between variables, but an underlying causal model".

Neural Networks

(Artificial) Neural Networks (ANNs) are AI/ML models is inspired by the patterns of neuronal connections in brains. Their structure consists of stacked layers of 'neurons' which perform a simple mathematical operation (such as addition or multiplication). Each neuron receives input from a subset of (or all) neurons in the previous layer and sends its output to the next layer. The connections between neurons are weighted – this means that the output to the next layer can be strong for one receiving neuron, and weak for another). Raw data enters the first layer and is processed through the intermediate layers to reach a final output layer. Although inspired by neuronal connections, the actual resemblance of artificial neural networks to the brain is quite limited. More biologically faithful models exist, such as Spiking Neural Networks, but their performance still falls short of their artificial counterparts.

Multi-Layer-Perceptron The simplest type of a neural network consists of a sequence of layers in which each neuron is connected to all neurons of the subsequent layer. The weighted linear connections are augmented by simple non-linear transformations, such as the Rectified Linear Unit (ReLU) which leaves positive inputs to each neuron unchanged and replaces all negative input values with zeros. MLPs are still widely used, mainly as sub-components of more complicated networks, and are often referred to as *Fully Connected Layers.*

To give an idea of their potential power in modelling complex functions, an MLP that was 50 layers deep, and 40 neurons 'wide' in each layer, would have 1600 weighted connections between the 40 neurons in one layer and those in the next, or 78,400 (1600 x 49) connections in total. Each connection contains multiple parameters which can be learnt and given that each parameter holds a continuous value, the number of possible configurations becomes very large. For many applications, however, the more advanced models such as CNNs and RNNs are more widely used, which often have many millions of parameters such as the natural language transformer model BERT (340 million) [22].

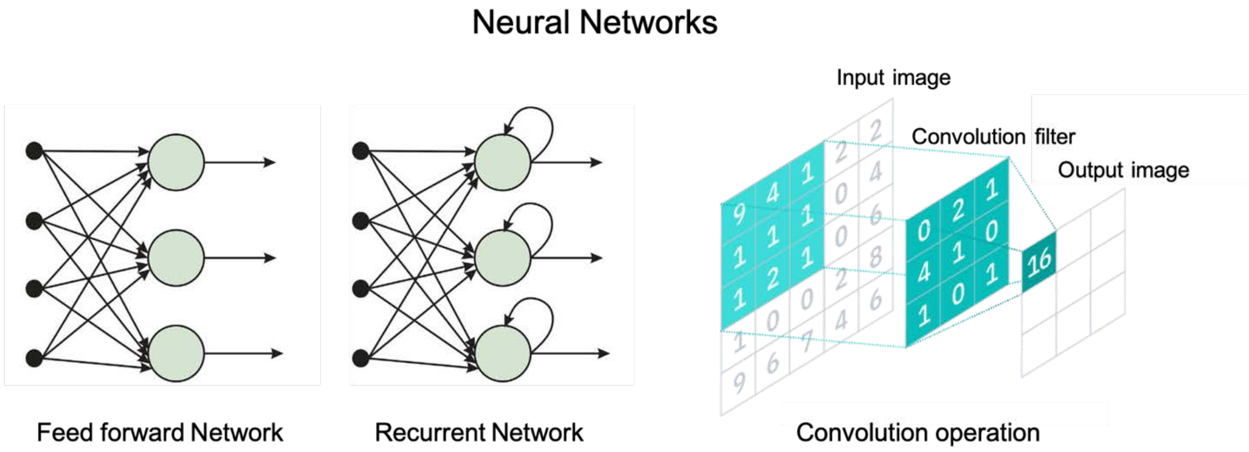

Figure 4: Schematic of a Feed Forward Network (Multi-Layer Perceptron), Recurrent Network and the convolutional layer of a CNN [23], [24].

Convolutional Neural Networks *(CNNs)* have proven extremely successful at image-based tasks such as image classification or segmentation. Their advantage over MLPs is that they can recognise objects regardless of where they are located within an image. This is achieved by learning processing inputs through "convolutions" and learning 'filters' (also called kernels). A convolution is a mathematical function which combines the values of a local collection of pixels from the image. Filters become sensitive to features such as vertical or diagonal edges, contrast changes etc., which can recognise a pattern or a shape wherever it occurs. The filters of CNNs are the weighted connections of ANNs - during training, the system learns which filters are most useful in classifying the images from the dataset.
CNNs are now surpassing human performance on may image recognition tasks [25]. The most common *image perception* tasks solved by CNNs include:

- *Image Classification* – assigning a label to an image.
- *Semantic Segmentation* - labelling every pixel of an image (e.g., as "blood" or "heart")
- *Object Detection* which finds the location of one or multiple objects in an image and

- *Instance Segmentation*, whereby all the pixels of an object are found.

In medical imaging they are used to classify or segment CT and MRI scans and X-rays (as 2-D images, as a series of slices, or as a true 3-D image) and have proven very effective at detecting tumours from X-rays or CT scans, or delineating organs [26].

Recurrent Neural Networks *(RNNs)* are an extension of MLP models for sequential data, such as language, where interpretation takes account of the order of things in a sequence. In RNNs, each layer in the neural network doesn't only receive the input from the previous layer but also information from its own previous state (see Figure), and this information from a previous time step provides a sort of very short-term memory of the sequence of data. The applications of RNNs are in time-series analysis and forecasting (e.g., predicting seizures based on EEGs, predicting demand on hospital services) and Natural Language Processing (NLP) for speech recognition, text generation, or translation.

Like CNNs, the RNN architecture has undergone many iterations of improvement of which the most successful ones are the Long-Short Term Memory (LSTM) and the Gated Recurrent Unit (GRU), both of give more advanced algorithms for retaining and processing the short-term memory in the Recurrent model.

Other techniques

Deep Ensembles Even with the same architecture and data, each Neural Network training involves a stochastic optimisation process and can result in a slightly different function mapping - a different set of learned connection weights. To improve overall prediction quality, deep ensemble architectures train multiple networks for the same task and deliver predictions based on combined results. The approach is similar in nature to how random forests leverage multiple decision trees.

Twin Networks These models link two or more identical subnetworks with exactly the same connection weights and use them to predict a similarity score between two or multiple inputs. They are used in areas such as ID checks based on face recognition and are useful in cases of limited training data as they can learn differentiating features from fewer examples and have shown good results for one-shot learning tasks [27].

Generative Adversarial Networks *(GAN),* have a separate learning paradigm and a different network architecture. Two networks are pitted against each other, with one trying to create ever more sophisticated synthetic inputs while the other to steadily its algorithms to distinguish fakes correctly. Their best-known use is in image generation, where adversarial training allows accelerated development of highly realistic images which are useful in training other networks and in simulations, but also have led to the so-called **Deep Fake** videos.

Attention-based models and Transformers*.* The Attention mechanism was first proposed in 2016 for machine translation [28] to allow a network to process an entire block of language without a pre-set sequence (as in RNNs) and instead learn which parts of the input to focus on. Attention is a vector of the importance weights learned during training. In 2017 researchers at Google produced the *Transformer* model, a NN model for translation based on attention, which outperformed all RNN methods. It was followed by BERT, a transformer-based model for NLP pre-trained on all of Wikipedia, and by OpenAI's GPT2 and GPT3 (Generative Pre-trained Transformer). Outside of NLP, Transformer architecture with learned attention has also found its way into the Computer Vision community (Visual Transformers (*ViT, 2020)* [29]) and the attention-based processing also has potential to improve explainability in some uses.

Graph Neural Networks (GNNs) are graph-based learning models consisting of nodes and edges, which don't process information in a *feedforward* or *recursive* manner, but instead passing messages between neighbouring nodes along edges. Training GNNs affects how and when messages are transferred along the edges and the output is collected from the updated nodes of the graph. In structure, GNNs resemble graphical models, but are embedded into a deep learning framework. Their node and edge structure is generally static, but very advanced techniques have explored dynamic graphs [30].

4. AI case-study: building an end-to-end AI system

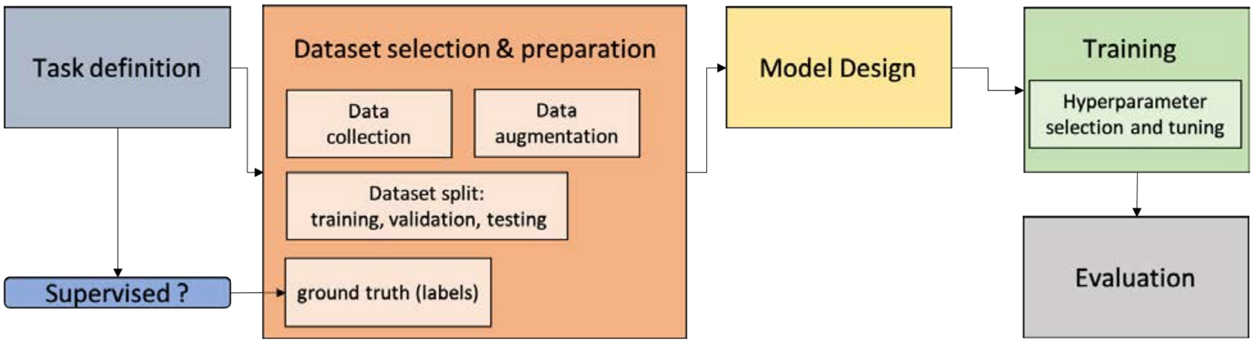
Task definition
Supervised ?
Dataset selection & preparation
Data collection
Data augmentation
Dataset split: training, validation, testing
ground truth (labels)
Model Design
Training
Hyperparameter selection and tuning
Evaluation

Figure 5: The core steps to building an end-to-end AI system.

Building an end-to-end AI/ML system requires many design steps and choices, and the design choices vary hugely between application areas. In this section we give just one illustration, in medical imaging, drawing on the breast cancer detection system presented by DeepMind in 2019 [31]

As a first step, the task has to be defined. *Defining the task* to solve involves translating the problem (detecting breast cancer from mammograms) into an AI/ML task: what should the algorithm predict?

- From a clinical perspective the questions will include: who will use it and how, with what types of input images? will it provide a preliminary screen, or a confirmatory second opinion? what levels of sensitivity and specificity are needed to add value to existing practice? is it more important to minimise false-negatives or false-positives?
- From an AI perspective, it has to be decided whether the task should be formulated as a regression or classification task, and if classification, how many classes to predict. This decision is usually done in parallel with deciding on the learning method, since for supervised training, whether the task is set as a regression or classification problem often depends on the ground

truth data that is available. DeepMinds' cancer detection task is formulated as a regression task, predicting a score between 0 and 1 for the cancer risk score.

Deciding on the dataset is crucial for the *generalisability* and *performance* of the AI/ML algorithm, as well as its *fairness.* A good dataset has to be representative of the domain and free of bias (no under- or overrepresentation of certain classes). For training their model on breast cancer detection, DeepMind chose two large clinically representative datasets from the cancer screening programs of the UK and the US. The UK dataset consists of a total of 10350 lesion detections in 9611 images from 4947 women.

Obtaining ground truth When training in a supervised way, the data has to be annotated with the correct labels, so the AI/ML model can learn from examples. In medical imaging applications, ground truth data is often obtained from human experts. The mammograms of the DeepMind dataset were annotated by OPTIMAM [ii]as well as US-board-certified mammographers. Another portion of their dataset was obtained from the annotated CBIS-DDSM public dataset. Combining multiple sources for data not only increases the dataset size but also adds variety to the training data.

Training, Validation and Test datasets After a labelled image dataset is collected, it is usually split into training, validation and testing sets. Usual split sizes are 70%, 15% and 15% respectively. However, for medical product development some additional and fully independent test data is often required to demonstrate that the trained AI/ML model will work reliably across different domains. The DeepMind team tested their model trained on the UK dataset on US data, and vice-versa.

Data augmentation In most cases, in addition to collecting a large dataset, a common step is further enrich the training dataset by data augmentation. This involves manipulating existing data in ways that add slight variations, e.g., rotations and warping for images, or overlaying noise patterns. This increases the variety of data the network can learn from and should give a more robust finished product. The transformations applied by the DeepMind team included *elastic deformation*, *shearing*, *rescaling*, *translation* and *flipping* of the mammograms.

AI/ML Model design Model design consists of selecting the correct type of model for the task and input structure (e.g., a CNN for imaging and a RNN or a Transformer model for time-series prediction or natural language processing). In challenging real-world tasks, it is often necessary to combine different AI/ML models for different subtasks of the prediction.

The model built by DeepMind consists of an *Ensemble* of three AI/ML models whose prediction scores are averaged for the final evaluation. Since the inputs are images, all three are based on CNN architectures. The first model, *the Lesion model* consists of two sequential sub-modules, a CNN-based detection model which detects suspicious regions - regions of interest (ROI) - in the image and a classification model which assigns a cancer risk score to each of these regions. The second model, *the Breast model*, generates a per-breast cancer score using a CNN feature extractor on each mammogram and combining the extracted features for each breast. The third model, *the Case model,* combines all mammogram features into a case cancer score.

Training Once the dataset and AI/ML model are set up, the training hyperparameters have to be chosen. Mainly, these parameters are the optimizer, the batch or mini-batch size (dataset subsample selected for one training iteration), the learning rate and the learning rate scheduler, which sets how the learning rate reduces to smaller increments as training progresses. Hyperparameters are often chosen using trial-and-error or empirical knowledge and experience from training previous models. To train their Ensemble, Deep Mind used the Adam optimizer for all three models and trained using mini-batch sizes of 4, 16 and 2 for the Lesion, Breast and Case models respectively. The learning rates were initialised at 0.0015, 0.0001 and 0.001.

Evaluation Finally, a model's performance has to be evaluated. Evaluation is performed on one or more test datasets, using the selected evaluation metric. For evaluation on the test dataset, the model parameters which performs best on the validation dataset are selected. To evaluate their models' performance on breast cancer classification, the DeepMind team computed the Sensitivity, Specificity and the AUC score, common metrics for binary classification. They also provide an insight into the number of false positive and false negative rate of the model prediction, an important factor in medical diagnosis. In addition, they provided a specific study to evaluate their models' ability to generalise to new domains – they evaluated their model on the US test dataset after only training on only the UK dataset.

At the time of writing, this mammography system is now in further clinical evaluation, supported by an NHSX AI in Health and Care Award (2021).

5. Deeper inside machine learning - how does learning actually work?

"All the impressive achievements of deep learning amount to just curve fitting" (Judea Pearl) [32]

We have discussed the high-level elements in AI machine learning - a **dataset**, a **model** and the **task** to solve – and new need to consider two more technical, internal components – the **optimisation algorithm**, and **cost function** (performance measure). The optimisation algorithm defines how the model is progressively improved to perfectly fit the dataset; the cost function gives a measure of how far the model is from perfect performance.

Optimisation is at the heart of machine learning – most ML algorithms can be formulated as an optimisation challenge with the goal of finding the lowest possible value of a calculated cost function (or performance measure) [33]. Mathematically, optimization functions are usually divided into first-order, higher-order (2nd derivative and above) and derivative-free methods [33], and while higher-order optimization methods converge on the answer faster [33], they are not easy to use with deep learning methods. Deep learning methods therefore use first-order optimisation methods, based on *gradient descent*.

For a mathematical function – e.g., a function that defines how multiple inputs should combine to give a disease risk score - G*radient Descent* finds the minimum (the best form) of the function by 1) computing the first-order gradient (the slope) and 2) *stepping in the negative direction of this gradient*. How far the algorithm steps into the gradient direction in each calculation cycle, is set by the **learning rate** – large steps are faster but can miss the optimal values.

General concepts

Underfitting, overfitting and generalisation
The goal of training an AI/ML model is to be able to make predictions for new, unseen data. The ability of a model trained on one set of data (the **training data**) to make good predictions on new data (the **test data**) is **generalisation**. Models which don't generalise well but performed well on the training data are said to have **overfitted** – their parameters fitted specific features of the training examples which are not generally useful. If the model can't make good predictions on the training data either, it **underfitted** the data.

Training, validation and test set – a common practice is to work from a single data set but split it into training, validation and test sets (usually at a 70%, 15% and 15% ratio). The examples from the training dataset are the only ones the AI/ML sees during training. The validation set is used to give an estimate of the models' performance on unseen data during training. This is used to help tune the model parameters and decide when the model has reached maximum performance and can stop training. Final evaluation of the AI/ML model is done on the test set. Validation and test sets have to be as the validation set is indirectly involved in the training process.

Cost functions and Evaluation metrics Cost functions (also called loss functions) are used during training to compute the gap (or error) between the AI/ML model's performance and the desired performance and drive model training; evaluation metrics are used to measure the trained AI/ML models' performance. Although a cost function can sometimes serve as an evaluation metric, these are mostly different.

Data augmentation AI/ML algorithms are extremely data hungry and since there is rarely enough data to train them, methods often rely on data augmentation methods which modify existing data in order to obtain new examples. In the case of images, this includes flipping, warping as well as rotating the original image to obtain a modified version of it, which can be used as an additional example in the training set.

In neural network (NN) training the optimisation processes will be adjusting the weights of the vast number of connections between neurons deep inside the layers of the network. To compute the gradient and the step-change needed for a particular weight in a particular connection in the model, a long chain of partial differentials is computed, commonly referred to as **back-propagation.** Intuitively, each weight (or other parameter) in the neural network model is improved in each cycle by propagating the prediction error backwards through the layers until reaching the parameter where a change will improve performance.

In theory, the cost function might be calculated using the whole training set for each step change in the cycle, however, when a NN contains tens of thousands of parameters, and is trained on data sets with tens of thousands of data points, this becomes unfeasible. In practice, AI/ML methods therefore use **stochastic gradient descent (SGD)**, where a subsample from the data is used in each training iteration – a **batch** or **mini-batch** of the data.

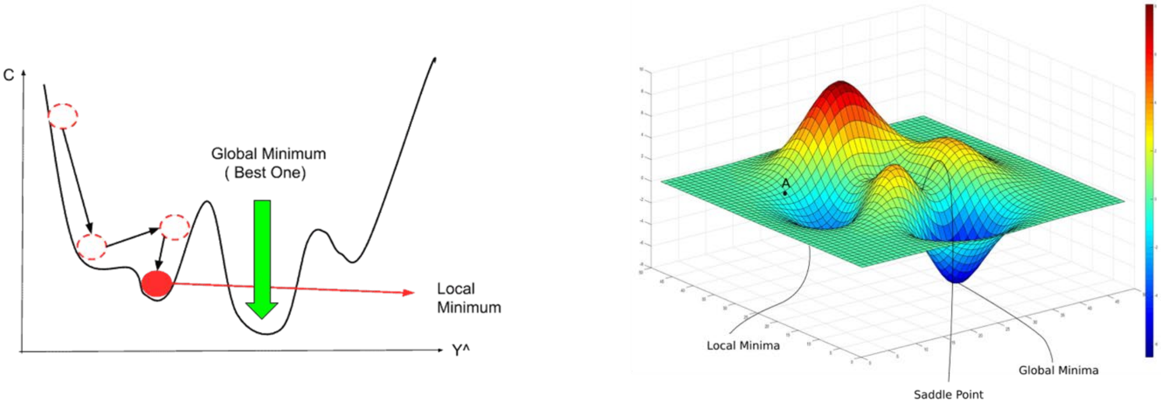

Figure 6: **Left**: A non-convex function with local and global minimum in 1D. **Right**: The equivalent in 2D [34], [35].

Before the cycles of training begin, several hyperparameters have to be selected, among which are the learning rate as well as how the learning rate changes over time – the learning rate schedule or scheduler. Common learning rate values are of the order of 0.0001 and are scheduled to decay to smaller values as training progresses. The AI/ML model is then trained for a pre-set number of **epochs**, which is the number of times the model sees the entire training set. Remember that only a subsample (a batch) of the data is used at each optimisation step.

Convergence Most functions we want to approximate in AI/ML methods are non-convex, meaning they have multiple **local minima** - positions where the function is at its optimal (lowest) point in a local region (see Figure 2), but which are not the lowest point of the function. Sometimes non-convex functions don't even have a **global minimum** (i.e. there is no single best solution). A major challenge is therefore finding the best minimum and not getting stuck in local, non-optimal minima. Several updated optimisation methods have been proposed, which improve convergence, such as *AdaGrad* [36], *AdaDelta* [37], *RMSprop* [38] and the *Adaptive Moment Estimation (Adam)* [39] – this was the optimizer used by DeepMind (see section 4 for the case study). These methods are based, among others, on **momentum**, which accumulates an exponentially decaying moving average of past gradients [13].

6. Software and hardware for building and deploying AI

AI/ML models are created within computer programs and have separate characteristics in their training phase – with frequent adjustment of hyperparameters and extraordinarily heavy computation workloads - and implementation phase – where algorithms are usually locked, and computation demands lower.

AI/ML models can be built and trained in any programming language, but *python* is the most popular for research and development. Users are supported by large number of open-source libraries and ML Frameworks for model development, training and data pre- and post-processing. The most prominent recent (2015, 2016) frameworks are *Tensorflow* (Google Brain) and Facebook's (FAIR) framework, *Pytorch.* There are also now cross-language frameworks (such as Apache MXNet (2017) which supports twelve languages) and tools for interoperability between frameworks, such as the '*Open Neural Networx exchange' (ONNX)*, a joint project of Microsoft, Facebook and AWS.

The computational demands of model optimisation can be very high. Training (optimisation) is usually run using GPUs (Graphics Processing Units) which have a structure better suited to processing large volumes of data and calculation in parallel, compared to the CPUs that drive most computers. A few exceptionally complex AI models have needed over 10,000 hours of GPU time (such as AlphaFold which was trained for several weeks using computational power of 100-200 GPUs) [40], but most current healthcare projects are orders of magnitude smaller [31].

Once a model is trained, the computational workload of applying it to (e.g.) process an image is much lower, since the backward pass (parameter tuning) is omitted. The hardware requirements to run such a model are storage space of Megabytes or a couple of Gigabytes at most, as well as the memory requirements to make a prediction from data. Recent progress has allowed for trained (locked) object detection and image segmentation AI models to be deployed on mobile phones (MobileNetv2 [41]). The implementation model may also be switched to different programming language, to improve speed, stability or quality. Some new frameworks such as TensorFlow Extended (TFX) are intended as production-grade platforms that can be used at every stage from data acquisition to model training and analysis to large-scale deployment.

AI models are inevitably deployed as components within much larger software systems. These will have processes covering, for example: input sources and interfaces; data collection and verification; feature extraction; non-ML data analysis; processing management; user interfaces and controls; reporting; monitoring and error management. Often, there will be several AI models within a single system, for example as a processing pipeline for images, or as an ensemble of models running in parallel. Performance and safety will depend on the quality of the whole system, and programming and testing can be hard if the functioning of the AI model is not well understood. The system challenges can include:

- Ensuring data input ranges and formats are always within the expected (trained range), especially as apparently trivial variations can have large effects
- Outputs from AI models that entangle or combine variables in ways that affect logic elsewhere, or create feedback loops that are hard to detect
- Designing test protocols that fully probe all areas of the AI model.

Based on approaches used in conventional software development and testing, “MLOps” is a growing body of methods and good practice in product development, and training programmes, that cover both experimental AI/ML models and their translation into a production system. It typically covers eight stages: *data collection, data processing, feature engineering, data labelling, model design, model training and optimization, endpoint deployment,* and *endpoint monitoring.* [42].

Bibliography

[1] C. Smith, *The History of Aritificial Intelligence,* University of Washington, 2006.

[2] T. Mitchell, Machine Learning, 1997.

[3] M. Tilman, "What is Apple Face ID and how does it work?," Pocket Lint, [Online]. Available: https://www.pocket-lint.com/phones/news/apple/what-is-apple-face-id-and-how-does-it-work. [Accessed 19 08 2021].

[4] S. Arora, "Supervised, Unsupervised, Reinforcement Learning," Aitude, [Online]. Available: https://www.aitude.com/supervised-vs-unsupervised-vs-reinforcement/. [Accessed 19 08 2021].

[5] D. Lee, "A brief introduction to Reinforcement Learning," Medium, [Online]. Available: https://medium.com/theory-practice-business/reinforcement-learning-part-1-a-brief-introduction. [Accessed 19 08 2021].

[6] J. Huang, *Unsupervised Deep Learning by Neighbourhood Discovery,* ArXiv, 2019.

[7] T. Upstill, "The new Google News: AI meets human intelligence," 08 05 2018. [Online]. Available: https://blog.google/products/news/new-google-news-ai-meets-human-intelligence/. [Accessed 19 08 2021].

[8] O. Six, "The ultimate guide to AI in radiology," Quantib, 2020. [Online]. Available: https://www.quantib.com/the-ultimate-guide-to-ai-in-radiology.

[9] T. Chen, "A Simple Framework for Contrastive Learning of Visual Representations," in *ICML*, 2020.

[10] D. Silver, "Mastering the game of Go without human knowledge," *Nature,* vol. 550, p. 354–359, 2017.

[11] C. Yu, *Reinforcement Learning in Healthcare: A Survey,* ArXiv, 2020.

[12] I. Godfried, "A review of recent reinforcement learning applications to healthcare" towardsdatascience, 10 12 2018. [Online]. Available: https://towardsdatascience.com/a-review-of-recent-reinforcment-learning-applications-to-healthcare. [Accessed 19 08 2021].

[13] I. Goodfellow, Deep Learning, MIT Press, 2016.

[14] A. H. Witten, Data Mining: Practical Machine Learning Tools and Techniques, 2016.

[15] R. G, An introduction to kernel-based learning algorithms., IEEE Trans Neural Netw., 2001.

[16] S. Winters-Hilt, "SVM clustering," *BMC BioInformatic,* 2007.

[17] T. Yiu, "Understanding Random Forest" towardsdatascience, 12 June 2019. [Online]. Available: https://towardsdatascience.com/understanding-random-forest. [Accessed 19 08 2021].

[18] S. K. Zhou, Handbook of Medical Image Computing and Computer Assisted Intervention Chapter 19, 2020.

[19] "Decision Trees" DataFlair, [Online]. Available: https://data-flair.training/blogs/r-decision-trees/. [Accessed 19 08 2021].

[20] W. Koersen, "Random Forests," [Online]. Available: https://williamkoehrsen.medium.com/random-forest-simple-explanation. [Accessed 19 08 2021].

[21] "Graphical_model," [Online]. Available: https://en.wikipedia.org/wiki/Graphical_model. [Accessed 19 08 2021].

[22] "Pretrained models" huggingface, [Online]. Available: https://huggingface.co/transformers/pretrained_models [Accessed 19 08 2021].

[23] A. Eliasi, *The role of AI in capital structure to enhance corporate funding strategies,* 2020.

[24] "Convolutional Neural Networks" IBM, [Online]. Available: https://www.ibm.com/cloud/learn/convolutional-neural-networks.

[25] Selvikvåg Lundervold, "An overview of deep learning in medical imaging focusing on MRI" *Zeitschrift fuer Medizinische Physik,* vol. 29, no. 4, 2019.

[26] H. Yu, "Convolutional neural networks for medical image analysis: State-of-the-art, comparisons, improvement and perspectives," *NeuroComputing,* vol. 444, 2021.

[27] G. Koch, Siamese Neural Networks for One-shot Image Recognition, ICML, 2015.

[28] D. Bahdanau, Neural machine translation by jointly learning to align and translate, ICLR, 2015.

[29] A. Dosovitskiy, "An Image is Worth 16x16 Words: Transformers for Image Recognition at Scale," in *ICLR*, 2021.

[30] "Deep Learning on Dynamic Graphs," 29 07 2020. [Online]. Available: https://towardsdatascience.com/deep-learning-on-dynamic-graphs. [Accessed 19 08 2021].

[31] S. McKinney, "International evaluation of an AI system for breast cancer screening," *Nature,* vol. 577, 2019.

[32] "To Build Truly Intelligent Machines, Teach Them Cause and Effect," [Online]. Available: https://www.quantamagazine.org/to-build-truly-intelligent-machines-teach-them-cause-and-effect. [Accessed 19 08 2021].

[33] S. Sun, *A Survey of Optimization Methods from a Machine Learning Perspective,* 2019.

[34] "Stochastic Gradient Descent," [Online]. Available: stochastic-gradient-descent-a-super-easy-complete-guide/. [Accessed 19 08 2021].

[35] A. Kathuria, "Intro to optimization in deep learning: Gradient Descent," [Online]. Available: https://blog.paperspace.com/intro-to-optimization-in-deep-learning-gradient-descent/. [Accessed 19 08 2021].

[36] J. Duchi, "Adaptive Subgradient Methods for Online Learning and Stochastic Optimization" *Journal of Machine Learning Research,* 2011.

[37] M. D. Zeiler, *ADADELTA: An Adaptive Learning Rate Method,* ArXiv, 2012.

[38] S. Ruder, *An overview of gradient descent optimization algorithms∗,* ArXiv, 2017.

[39] J. B. Diederik P. Kingma, "Adam: A Method for Stochastic Optimization," in *ICLR*, San Diego, 2015.

[40] W. D. Heaven, "MIT Technology Review" [Online]. Available: https://www.technologyreview.com/deepmind-protein-folding-ai-solved-biology-science-drugs-disease/. [Accessed 19 08 2021].

[41] "MobileNetV2: The Next Generation of On-Device Computer Vision Networks" GoogleAiBlog, [Online]. Available: https://ai.googleblog.com/mobilenetv2-next-generation-of-on. [Accessed 19 08 2021].

[42] "MLOps" Wikipedia, [Online]. Available: https://en.wikipedia.org/wiki/MLOps. [Accessed 19 08 2021].

[43] G. Kaissis, End-to-end privacy preserving deep learning on multi-institutional medical imaging, Nature, 2021.

[44] P. Rajendra, "Modeling of dynamical systems through deep learning," *BioPhysical Reviews,* vol. 12, 2020.

[45] I. Lunden, "Babylon Health is building an integrated, AI-based health app to serve a city of 300K in England," Tech Crunch, 23 01 2020. [Online]. Available: https://techcrunch.com/babylon-health-is-building-an-integrated-ai-based-health-app. [Accessed 19 08 2021].

[46] "Simulacrum," [Online]. Available: https://simulacrum.healthdatainsight.org.uk/. [Accessed 19 08 2021].

[47] P. Kairouz, *Advances and Open Problems in Federated Learning,* ArXiv, 2021.

[48] J. Pearl, "An Introduction to Causal Inference," *Int J Biostat,* 2010.

[49] D. B. Rubin, "Estimating Causal Effects of Treatments in Randomized and Nonrandomized Studies," *Journal of Educational Psychology,* 1974.

[50] B. Dickson, "Why machine learning struggles with causality," 15 03 2021. [Online]. Available: https://bdtechtalks.com/2021/03/15/machine-learning-causality/. [Accessed 19 08 2021].

[51] "Causal Bayesian Networks: A flexible tool to enable fairer machine learning" 03 10 2019. [Online]. Available: https://deepmind.com/blog/article/Causal_Bayesian_Networks. [Accessed 19 08 2021].

[52] "Causality and Machine Learning," [Online]. Available: https://www.microsoft.com/en-us/research/group/causal-inference/. [Accessed 19 08 2021].

[53] J. Brownlee, "10 Clustering Algorithms With Python" machinelearningmastery, [Online]. Available: https://machinelearningmastery.com/clustering-algorithms-with-python/. [Accessed 19 08 2021].

[54] "OPTIMAM MAMMOGRAPHY IMAGING" [Online]. Available: https://medphys.royalsurrey.nhs.uk/omidb/. [Accessed 21 08 2021].

Notes

[i] Other popular clustering techniques are *DBSCAN*, *BIRCH*, *OPTICS* and *Spectral Clustering* [53].

[ii] Large curated and centralised database of mammograms [54]