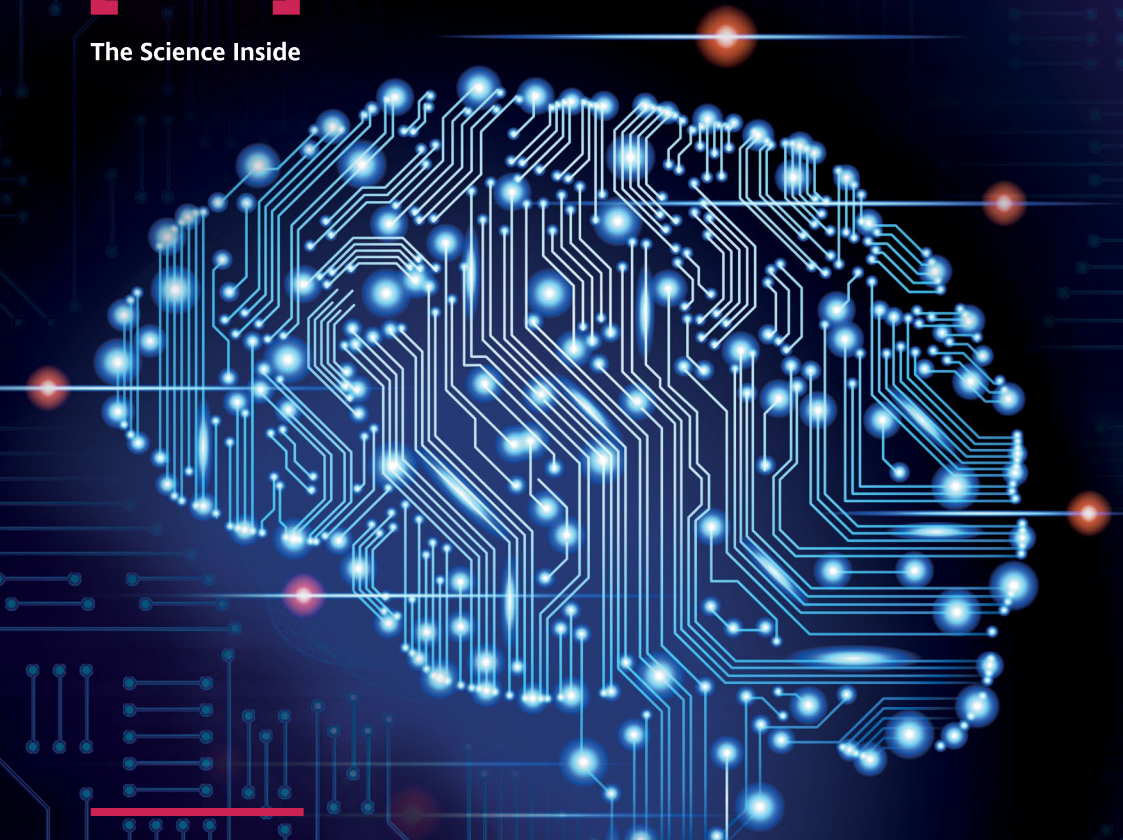
[dstl]

The Science Inside



Defence Science and Technology Laboratory

Machine Learning with Limited Data

Future of AI for Defence Project
Autonomy Programme

Ministry of Defence

Defence Science and Technology Laboratory

Machine Learning with Limited Data

Future of AI for Defence Project Autonomy Programme

Foreword

Machine learning offers Defence significant benefits, such as increasing the amount of data that can be analysed and reducing the load on human analysts. State-of-the-art approaches for machine learning, such as deep learning, can achieve high accuracy on a wide range of problems if sufficient training data is available.

Whilst there are a large number of problems in Defence which could benefit from machine learning, for many of these, state-of-the-art machine learning models cannot be applied due to a lack of training data or because of the expense and time required to label sufficient examples.

Machine learning is a data driven approach, with deep learning models usually requiring thousands of examples per class. In a limited data problem, there is not the large amount of data in the problem of interest needed to train a model, so we must use limited data approaches which generally utilise large labelled datasets with similar distributions or human knowledge to allow it to adapt rapidly to a small dataset. Whilst limited data machine learning approaches all improve over classical machine learning approaches for problems with a small amount of data, the best way to improve machine learning in limited data scenarios is to collect and label more data. In Defence there are a large number of problems where collection of further data is not possible or cost effective.

This handbook looks to provide a guide to the landscape of machine learning techniques for limited data problems in 2020. It is designed to inform and guide machine learning practitioners on the limited data machine learning approaches currently possible. This captures some of the knowledge gained from Dstl research into low-shot learning carried out by the Future of AI for Defence project, part of the Autonomy programme.

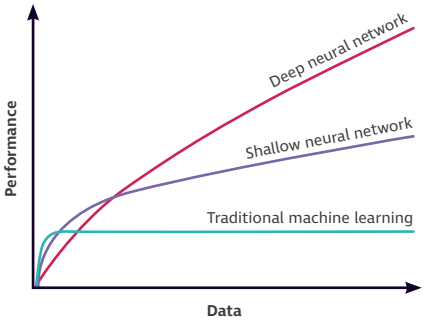
DSTL/PUB126061

Contents

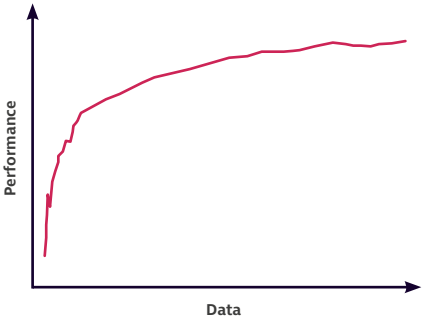
How much data is required?

As a data driven approach, machine learning is heavily reliant on the availability of training data. The amount of labelled data required to train a machine learning model depends on the dataset and machine learning model, with complex datasets and deeper models typically requiring more data for training.

Deep neural networks are currently the state-of-the-art machine learning approach for most problems. Deep neural networks allow far more generalisation than shallow neural networks and traditional machine learning approaches, and therefore achieve significantly better accuracy. The key challenge with applying deep learning to a problem is the large amount of data required to train deep learning models.



A comparison of the performance of traditional machine learning and neural networks when training data is limited.



An example of the performance of a deep neural network when training data is limited.

When the amount of data is limited, machine learning models tend to overfit. This is when the model is over optimised to the training data. Overfitting will mean that the model will not be able to generalise to unseen examples, resulting in poor performance.

Heuristically the training of deep neural networks requires thousands of examples per class. When there are problems with tens or hundreds of examples per class, we should consider limited data learning approaches.

The learning with limited data approaches considered in this guide look to use the power of deep neural networks on problems where there is insufficient data to train them classically.

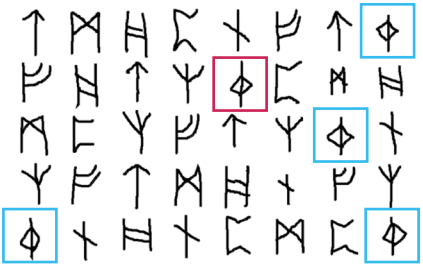
Human ability to learn with limited data

Most humans are able to learn new objects when given only a couple of examples. There are a number of different ways that humans approach learning with limited data, such as using prior experience, comparing known and unlabelled examples and the ability to internalise semantic descriptions of the object of interest.

The learning process of babies is of particular interest for machine learning as initially a machine learning model has no prior experience to exploit. If we can start to train machine models in a similar manner to how babies learn, machine learning models may be able to get closer to the human ability at learning with limited data.

Many of the machine learning approaches for limited data problems are inspired by the human approach to these tasks. Data driven approaches look to use prior experience for low-shot problems, or learn how to learn tasks.

Approaches to machine learning with limited data also look to leverage the human ability at recognising objects with only a few examples, through human labelling of examples or semantically describing objects to aid classification.



A limited data problem from the Anglo Saxon alphabet (Lake et al. 2015). When given a single example of an unseen character (red), most humans are able to find all other examples (blue).

6 Lessons from Babies (Smith et al. 2005):

1. Babies' experience of the world is profoundly multimodal.
2. Babies develop incrementally, and they are not smart at the start.
3. Babies live in a physical world, full of rich regularities that organise perception, action, and ultimately thought.
4. Babies explore – they move and act in highly variable and playful ways that are not goal-oriented and are seemingly random.
5. Babies act and learn in a social world in which more mature partners guide learning and add supporting structures to that learning.
6. Babies learn a language, a shared communicative system that is symbolic.

Limited data problems

Machine learning with limited data can be required in all types of problem and dataset. Most of the academic research on learning with limited data has centred on image classification problems but many of these techniques should apply directly to any machine learning problem or dataset.

Some examples of problems and datasets that machine learning with limited data have been applied to are:

Types of problem

Classification
Determining the class of the example.

Regression
Approximating the value of a function.

Object detection
Localising and classifying objects.

Event detection
Detecting and classifying events in temporal data for example time series, audio or video.

Segmentation
Partitioning the pixels of an image into multiple segments representing classes.

Reinforcement learning
Teaching an agent to take the best actions to achieve a goal.

Machine learning problems in datasets

Imagery
Object detection, segmentation and classification of overhead or natural images.

Time series
Classification and event detection in single and multi-dimensional time series.

Text
Natural language processing, generation and question answering.

Video
Object detection and event detection.

Audio
Event detection and classification.

Structured data
Regression or classification for tabular data.

Selecting the approach

	Small amount of data	Large amount of data
Mostly unlabelled	**Small amount of data, mostly unlabelled** ■ Zero-shot learning ■ Manual labelling See **Page 006**	**Large amount of data, mostly unlabelled** ■ Semi-supervised learning ■ Active learning ■ Unsupervised learning ■ Self-supervised learning See **Page 020**
Mostly labelled	**Small amount of data, mostly labelled** ■ Traditional machine learning ■ Meta-learning ■ Knowledge reasoning See **Page 008**	**Large amount of data, mostly labelled** ■ Deep learning

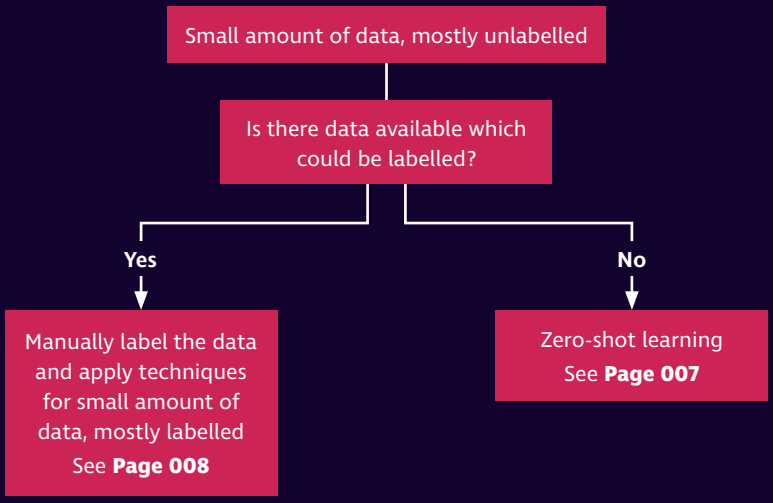
Model refinement

See **Page 025**

Small amount of data, mostly unlabelled

For some Defence problems we may not have any labelled examples of the event or object of interest, but we may have unlabelled examples or information from other sources, such as text or semantic descriptions. This section considers approaches which leverage semantic descriptions of the event or object of interest to train the model.

Whilst these approaches can be powerful for some problems, if there is unlabelled data available one would expect the best performance by manually labelling the data and considering the approaches for a small amount of data which is mostly labelled.



Zero-shot learning

TRL: 2

Zero-shot learning looks to apply deep learning techniques to problems where there is no training data available. Instead zero-shot learning uses descriptions of concepts to train the model.

Zero-shot learning approaches

User-defined attributes – Attributes defined by human experts or concept ontology.

Relative attributes – Measuring attributes relative to other examples.

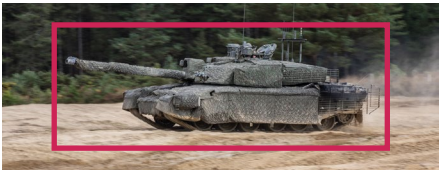
Data-driven attributes – Augmenting user-defined approaches by learning further attributes from data.

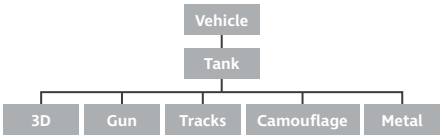
Concept ontology – Combining embeddings of data and ontologies, such as WordNet, and leveraging the ontology structure to classify.

Semantic word embedding – Utilising a semantic word vector space learned from linguistic knowledge bases.

Most zero-shot learning approaches take an embedding approach, where the data is embedded into a vector space alongside an embedding of the attribute information. Examples are then classified using the mutual embeddings.

Datatypes – Zero-shot learning is generally applied to classification in image or video datasets. Recently a few applications of zero-shot learning for semantic segmentation and object detection have been published.





An image of a tank and corresponding example description.

Implementations – Many of the recent zero-shot learning models have code available on GitHub.

Performance – State-of-the-art zero-shot learning approaches have achieved over 80% accuracy on the Animals with Attributes dataset (Lampert et al. 2009) and over 60% accuracy on the SUN scene understanding dataset (Xiao 2010). These are both standard research datasets.

Challenges:

- Generating a suitably labelled training set. The labelling required is significantly more detailed than is available in most research datasets.

Requirements:

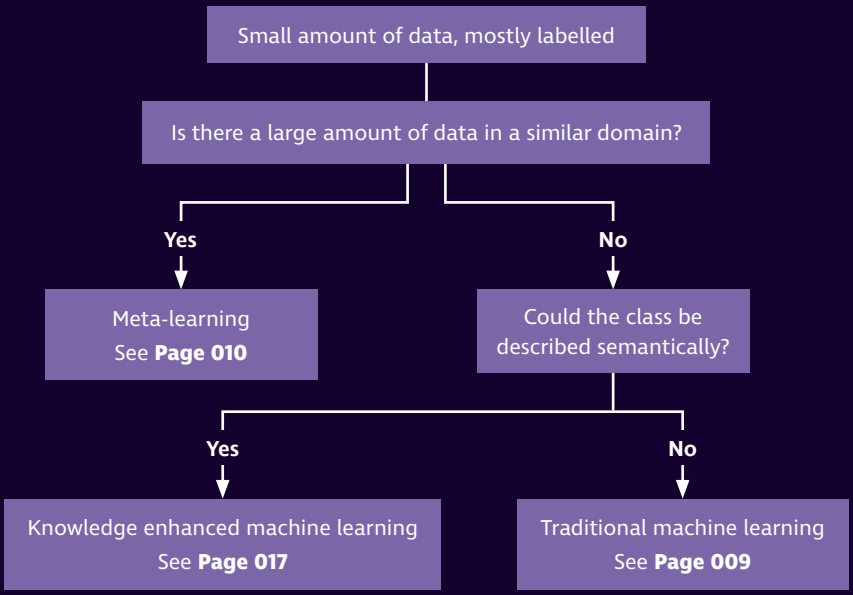
- A large dataset with relevant attributes labelled for base training.

Small amount of data, mostly labelled

Defence is typically interested in rare events, whether that is to identify unusual or unexpected objects, or events that have not previously occurred. In these scenarios we may only have a small number of examples of events or objects of interest, but a large amount of data to find other examples of these events or objects in. Computers are very effective at searching large amounts of data, so if we can apply machine learning to problems with only a few examples, we can significantly reduce the load on analysts.

A number of approaches have been developed to train machine learning models on small amounts of labelled data. In the absence of sufficient data to train classical machine learning models in the problem of interest, these models look to utilise prior experience learned from similar datasets with large amounts of labelled data, or use human knowledge to reduce the training required.

Collecting and labelling more data, if possible, is the best way to improve the accuracy achieved by machine learning on limited data problems.



Traditional machine learning

TRL: 6

Whilst deep learning approaches are the state-of-the-art machine learning models, they require large amounts of data to be trained. Shallower neural networks and traditional machine learning require less data to train and can outperform deep neural networks in limited data scenarios.

Traditional machine learning approaches

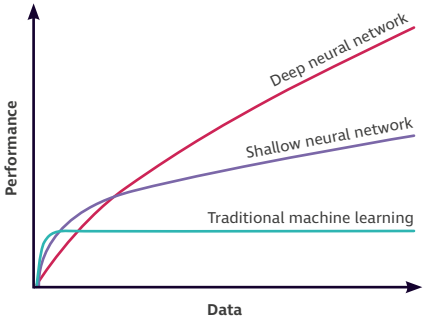
Support Vector Machines – Learn a hyperplane to separate the space into two-regions for two-way classification. Kernels can be used to map the space in order to apply SVMs to problems which are not linearly-separable.

Decision Trees – Learning the structure and rules of a decision tree to classify examples. Ensembles of decision trees, known as random forests, are often used to reduce the overfitting of decision trees.

Multi-Layer Perceptrons – An artificial neural network approach consisting of multiple layers of perceptrons. A perceptron is a model of a neuron, which outputs a linear combination of the inputs. MLPs are then trained via backpropagation.

Datatypes – Traditional machine learning methods generally take a vector input, so for the complex data structures of interest to Defence, such as imagery, feature extraction is generally required.

Implementations – The Scikit-learn python library implements most traditional machine learning approaches.



A comparison of the performance of traditional machine learning and neural networks when training data is limited.

Performance – For simple problems, traditional machine learning approaches can achieve high accuracy. The training of traditional machine learning approaches still needs a number of examples per class, typically hundreds.

Challenges:

- Suitable pre-processing to effectively train these models on complex datasets.

Requirements:

- Hundreds of examples per class.

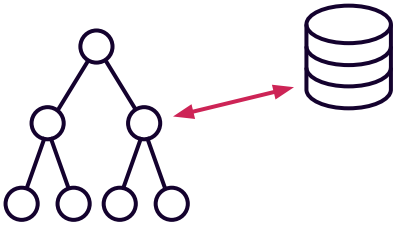
Meta-learning

TRL: 1-3

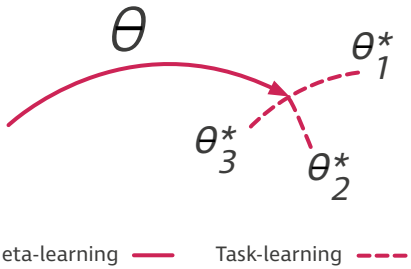
The goal of meta-learning is to learn how a machine learning model learns and leverage this to train the model more quickly. A meta-learner is trained on a number of similar learning tasks. This allows it to learn the optimal way to learn an unseen task. For limited data problems, meta-learning can be used to learn how to train a task when given only a few examples, rapidly adapt to a new environment and generalise to unseen tasks.

Meta-learning is a data driven approach, so it requires a large number of tasks from a similar dataset for meta-training. Once the meta-model is trained, it can be applied to learn an unseen task with limited data. Meta-learning also supports life-long learning so the model can be improving as it is being used.

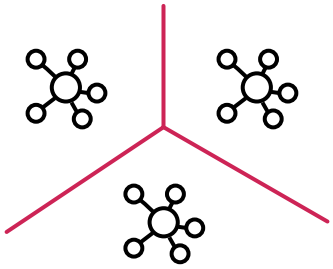
Types of Meta-learning



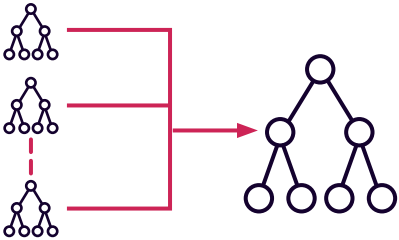
Learning from memory
Augmenting a neural network with an external memory source to remember examples seen previously.



Learning as optimisation
Expressing meta-learning as an optimisation problem, either using a second network to predict parameters or learning an initialisation for the neural network.



Learning to compare
Leveraging similar features in the examples, these approaches embed examples into a vector space and then use a metric to classify.



Learning by task inference
Using prior knowledge of the structure of naturally occurring tasks to improve model performance.

Images from icon8.com

Few-shot learning

TRL: 3

Few-shot learning is an approach to classification when there are a small number of examples per class. Few-shot learning considers N-way k-shot problems, in which we have N classes and each class has k labelled examples.

Few-shot learning uses meta-learning across a range of classification tasks to adapt rapidly to a new task. The meta-learning model is trained on a large set of N-way k-shot tasks from a similar dataset.

Learning to compare:

- Matching Networks (Vinyals et al. 2016).
- Prototypical Networks (Snell et al. 2017).
- Relation Networks (Sung et al. 2018)

Learning as optimisation:

- MAML (Finn et al. 2017).
- Meta Networks (Munkhdalai and Yu 2017).
- SNAIL (Mishra et al. 2018).

A 5-way 4-shot problem.

Datatypes – The approaches to few-shot learning from literature focus on image classification, with only a few examples considering other datasets, such as time-series. Many of the few-shot learning approaches for image classification could be adapted for other datasets by replacing the embedding neural network, or are model agnostic so could be applied to any model.

Implementations – Most recent papers on few-shot learning have made their code available on GitHub. These implementations are typically specific to the datasets tested in the paper so require adapting to new datasets. Dstl have modified some of these models to apply to Defence and more general datasets.

Performance – On the miniImageNet dataset (Vinyals et al. 2016) the current best performing models achieving around 80% accuracy on 5-way 5-shot tasks and 60% accuracy on 5-way 1-shot tasks.

Challenges:

- Few-shot learning requires the task to be structured as a N-way k-shot problem.
- Hyper-parameter optimisation can be difficult.

Requirements:

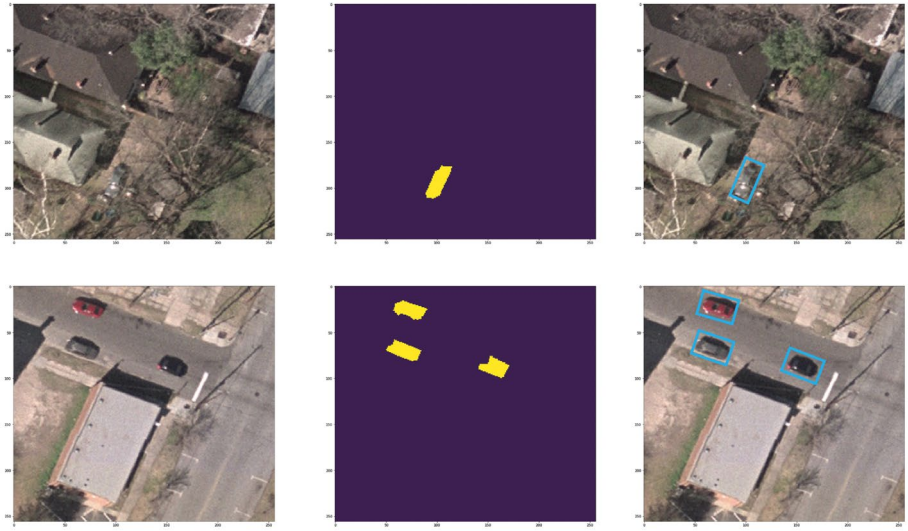
- A large dataset from a similar distribution.
- Access to high performance compute to train the meta-learner.

Few-shot segmentation and object detection

TRL: 2

Many of the approaches to few-shot learning and meta-learning can also be applied to image segmentation and object detection problems.

In few-shot segmentation we look to classify the pixels of the image into N classes where we have k examples for each of the N classes. In few-shot object detection we look to detect N classes of object using k examples per class. These problem structures mirror the N-way k-shot problems used for few-shot classification.

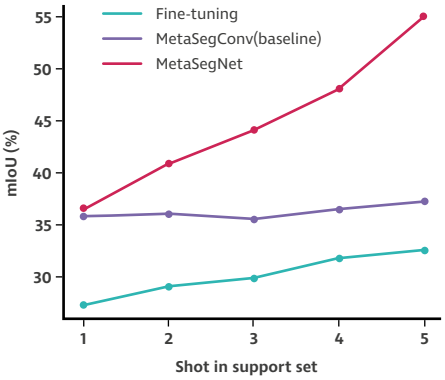


An example of segmentation and object detection on overhead imagery (Tanner et al 2009).

Datatypes – Few-shot segmentation and object detection have currently only been applied to research datasets such as PASCAL VOC (Everingham et al. 2010) and FSS-1000 (Li et al. 2020).

Implementations – Some of these approaches have code available on GitHub. At Dstl we have implemented some of the base meta-learning techniques using standard segmentation and object detection models.

Performance – State-of-the-art few-shot segmentation and object detection models demonstrate a significant improvement in mean IOU (Intersection over Union) over fine-tuning approaches.



The improvement in mean IOU compared to fine-tuning (MetaSegNet from Tian et al 2020).

Challenges:

- Few-shot segmentation and object detection are immature and have only been applied to relatively simple research datasets.
- Segmentation models are complex which makes training meta-learners difficult and computationally expensive.
- Hyper-parameter optimisation can be difficult.

Requirements:

- A large dataset of similar problems.
- Access to high performance compute to train the meta-learner.

Meta-reinforcement learning

TRL: 2

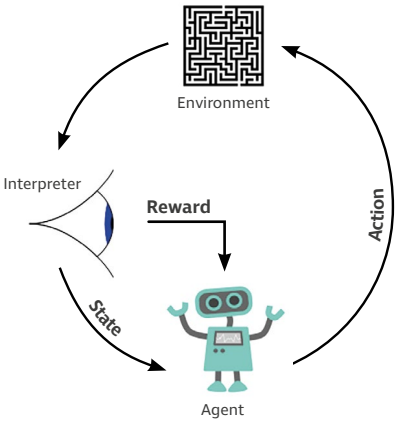
In reinforcement learning, limited data problems occur when a model is required to adapt quickly to a new scenario or reinforcement learning is used to train a real-world agent, such as a robot, and extensive training is not possible. Meta-learning is able to train reinforcement learning models to adapt rapidly to new environments.

A common approach to meta-reinforcement learning is learning an initial state of the agent, such that it can be updated in a few training runs. This approach was introduced in the MAML model (Finn et al. 2017) and there has been a large number of papers developing meta-reinforcement learning recently.

Meta-reinforcement learning can also help to train real-world agents through simulation, such that they can better handle the uncertainty present in the real-world. This has recently been applied to train a robot hand to solve a Rubik's cube through simulation (Akkaya et al. 2019).

Datatypes – Meta-reinforcement learning can be applied to any reinforcement learning environment where there are a large number of similar tasks to the target task, or where the agent needs to adapt quickly to the environment.

Implementations – Many of the recent meta-reinforcement learning models have code available on GitHub.



The reinforcement learning process.

Performance – Meta-reinforcement learning has been able to learn research tasks in a few simulated runs where classical reinforcement learning approaches require thousands of runs.

Challenges:

- Meta-reinforcement learning has only been applied to bounded challenges, such as research datasets and simple robots.

Requirements:

- A task where the goal can be randomly generated.
- Access to high performance compute to train the meta-learner.

Knowledge enhanced machine learning

TRL: 1-4

Machine learning is heavily dependent on the availability of data. Knowledge enhanced machine learning looks to use the innate human ability to learn with limited data and reason over the data available. These approaches therefore look to combine human knowledge, expressed in a structured manner, with the available training data to produce a machine learning model which can be trained with limited data.

Ontological approaches are an effective way of representing and reasoning over knowledge, but are not able to learn new objects. On the other hand machine learning has proved effective at learning new objects, but requires a large amount of training data and does not allow explainability and the injection of human knowledge to improve prediction. The approaches to knowledge enhanced machine learning look to bridge these two fields, by allowing human knowledge to be represented and utilised by the machine learning model and reducing the amount of training data required.

Zero-shot learning (considered in section "Small amount of data, mostly unlabelled") uses some of the knowledge enhanced embedding approaches to train a classifier using only a semantic description. In this section we instead look at augmenting the training data with expert knowledge.

Approaches

- **Symbolic models** – Fusing a representation of contextual knowledge into machine learning algorithms to improve algorithm performance.
- **Hierarchical learning** – Using a taxonomy representing the relationship between objects and higher level classes.
- **Zero-shot learning** – Leveraging descriptions of objects in terms of known attributes when there are no examples available for training. This was considered in the section on problems with a small amount of data, mostly unlabelled.

Symbolic models

TRL: 1

As humans, we use our understanding of the world to help us perform limited data learning tasks. This could be identifying known features of the class of interest, or decomposing it into common parts.

Some zero-shot learning approaches use attributes to classify unseen classes, given only a description in terms of these attributes. Symbolic models look to exploit similar approaches when there is some training data available.

Symbolic model machine learning algorithms use some representation of knowledge, generally human curated, such as ontology, database or contextual information. One can consider a traditional machine learning algorithm as learning a representation of the features in the classes, this approach looks to reduce the amount of data by introducing a human constructed representation.

Datatypes – Symbolic models could in theory be applied to any machine learning model. In most research to date, these models have been applied to image classification problems. The knowledge representation could be a common ontology (e.g. WordNet (Miller 1998)), a database of object specifications (e.g. Janes.com) or a custom representation curated for the task by a human operator.

An example of a decomposition of a Foxhound vehicle into its constituent parts.

Implementations – Symbolic models for machine learning are a very low TRL technique so there are only a few implementations available on GitHub.

Challenges:

- This is a very low TRL technique with significant research required before symbolic model approaches can be effectively applied to real-world problems.

Requirements:

- A representation of knowledge, generated by the operator or from a common ontology or taxonomy, that can effectively describe the classes.
- Training data in a similar domain to train the model to use the representation.

Hierarchical learning

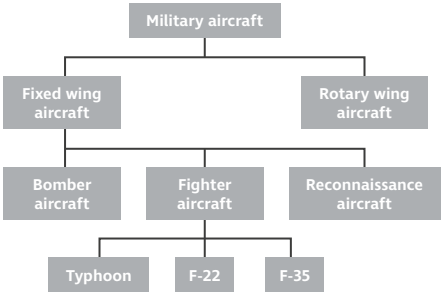
TRL: 2-4

As humans we categorise objects in hierarchical taxonomies. For example a tank is a type of vehicle, and a Challenger 2 is a type of tank. For limited data problems, we can exploit this hierarchical taxonomy to train classifiers for the higher level classes, which have more data available.

Hierarchical learning also gives a way to manage the uncertainty inherent in limited data problems, by being able to express confidence in higher level classes, while having less confidence in the lower level classes. For example with hierarchical learning we can express that an object is a tank, but the model is uncertain about what model of tank.

In fact this hierarchy is a simplification of the way humans understand taxonomies and poly-hierarchy gives a more accurate representation. For example a tank is a military vehicle, but is also a tracked vehicle and an armoured vehicle. These classes are distinct, but overlapping so cannot be represented in a standard hierarchy. Ontology is a common way of expressing semantic poly-hierarchies which could be exploited by hierarchical learning.

Datatypes – Hierarchical learning is typically applied to classification problems, with most examples using imagery datasets.



A hierarchy for military aircraft.

Implementations – Hierarchical learning can be easily implemented for the simple hierarchy by training different models for each layer in the taxonomy. The poly-hierarchical and ontological approaches to learning are less developed and more of an academic topic. There are some implementations on GitHub accompanying recent papers on the topic.

Challenges:

- Hierarchical learning requires the training of many different models for different levels of the hierarchy.

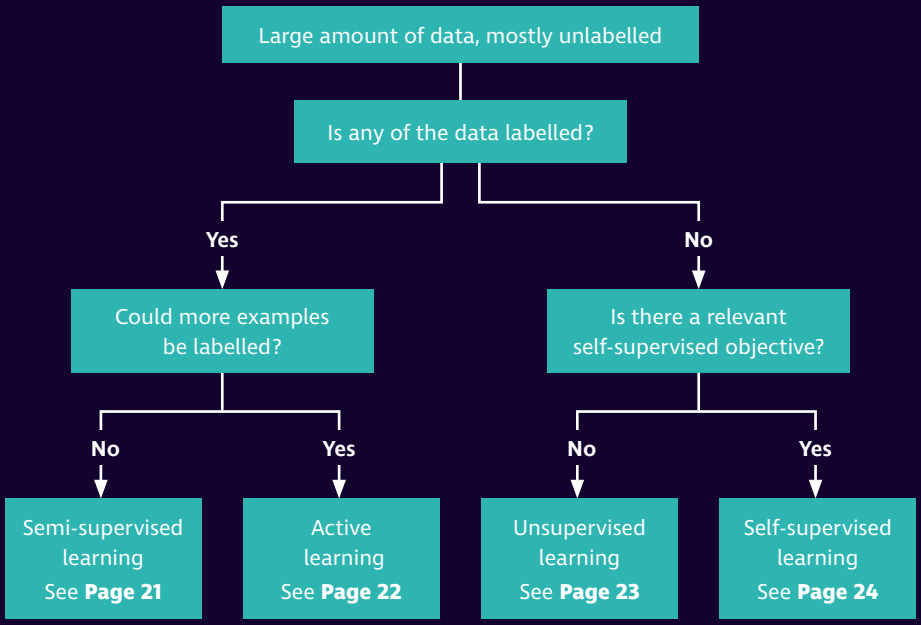
Requirements:

- A (poly-)hierarchical taxonomy to describe the objects of interest.
- Data for other classes within the taxonomy.

Large amount of data, mostly unlabelled

Defence has access to large amounts of data, but most of this does not have suitable labelling for applying machine learning. To be able to apply machine learning to this data, human labelling is required. Human labelling of data is a time consuming and expensive process.

To apply machine learning to such problems, inherent features in the data can be used to classify the examples through semi, self and unsupervised learning. The human labelling of data can also be made more efficient by selecting the most informative examples through active learning.



Semi-supervised learning

TRL: 5

Semi-supervised learning looks to utilise both labelled and unlabelled data to train machine learning models. For semi-supervised learning to be effective, the data must satisfy one of the following assumptions:

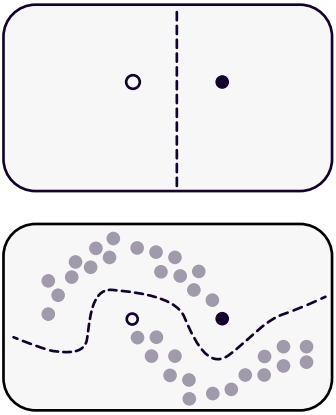
- Data points that are close to each other are more likely to share a label.
- The data forms clusters and points in the same cluster are more likely to share a label.
- The data points lie approximately on a manifold of much lower dimension than the input space.

If one of these assumptions are satisfied, a semi-supervised algorithm can assign a label to unlabelled examples using their proximity to known labelled examples. This produces extra labelled data for the machine learning model to be trained on.

Datatypes – Semi-supervised learning can be applied to any datatype, as long as there is a suitable embedding into a vector space which satisfies one of the assumptions above.

Implementations – The Scikit-learn python library implements an approach to label propagation for semi-supervised learning. For more complex approaches, many of the semi-supervised papers in literature have accompanying code available on GitHub.

Performance – Although semi-supervised learning approaches will not perform as well as if the whole dataset was human labelled, semi-supervised approaches can achieve an accuracy close to this performance with a fraction of the labelled data. Semi-supervised will not always improve the performance over using just the labelled examples, such as when neither of the assumptions above hold.



The influence of unlabelled examples in training.

Challenges:

- Semi-supervised learning will not perform as well as if all data points were labelled.

Requirements:

- The data must satisfy one of the assumptions listed above.

Active learning

TRL: 3

Active learning looks to reduce the amount of data required for human labelling. An active learning model uses a query strategy to select examples for a human operator to label, with the aim of selecting the most informative examples.

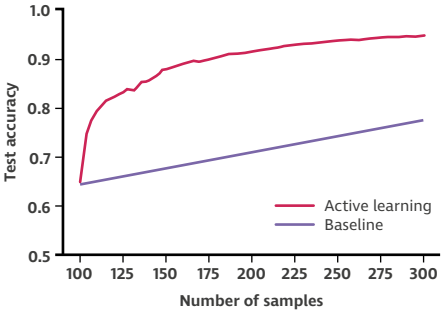
Types of query strategy

- **Uncertainty**
- **Query-by-committee**
- **Expected model change**
- **Expected error or variance minimisation**
- **Information gain**
- **Meta-learning**

Given a small number of seed labelled examples, a machine learning model is trained initially on these examples. The query strategy is then used to select unlabelled examples for human labelling. Once these have been labelled, the machine learning model is updated using the new labelled examples and this process is repeated to train the model.

Datatypes – Active learning is a training strategy rather than a model, so can be applied to any machine learning model. Active learning has been applied to classification, detection, segmentation and regression problems.

Implementations – There are a number of open source python libraries (ModAl, Libact, AliPy) implementing active learning frameworks with common query strategies. Many recent developments are not implemented in these libraries but have code available on GitHub.



The improvement in accuracy that active learning can bring over arbitrary data labelling.

Performance – Active learning has been shown to improve the rate of training for a number of academic datasets. Active learning has also been applied effectively to remote sensing problems.

Challenges:

- For deep models or complex datasets, hyper-parameter refinement is required for active learning to outperform arbitrary labelling.

Requirements:

- A human operator to label the selected examples for the algorithm.

Unsupervised learning

TRL: 5

Unsupervised learning is a machine learning approach which looks to learn patterns in data without labels. The unsupervised learning models therefore have to learn the structure of the input data based on features. Often the characteristic features of a dataset are not obvious from the raw data, and therefore performing feature extraction on the input data can significantly improve the performance of some unsupervised learning approaches.

Unsupervised dimension reduction and clustering on the MNIST dataset (LeCun et al. 1998).

Unsupervised learning approaches

Clustering – Clustering techniques look to group data points based on features in the data.

Anomaly detection – Anomaly detection techniques look to identify whether a data point lies in the same distribution as the existing examples.

Neural network – Neural network approaches look to embed the data in a lower dimensional space, whilst retaining the characteristic properties. Manifold learning is one of the main unsupervised neural network approaches.

Latent variables – These are techniques for reducing complexity of the data by learning latent variables. This includes Principal Component Analysis and the Method of Moments.

Datatypes – Unsupervised learning can be applied to any dataset. Some unsupervised learning approaches, such as clustering, benefit from pre-processing the data with a feature extractor.

Implementations – The Scikit-learn python library implements many unsupervised learning algorithms.

Performance – Unsupervised learning approaches perform significantly worse than supervised and semi-supervised approaches because the algorithm does not know the goal it is being tested against in unsupervised learning.

Challenges:

- Unsupervised learning approaches perform significantly worse than supervised learning approaches.

Requirements:

- Feature extractor for data.
- Human interaction with the algorithms to ensure the outputs are relevant to the problem of interest.

Self-supervised learning

TRL: 1

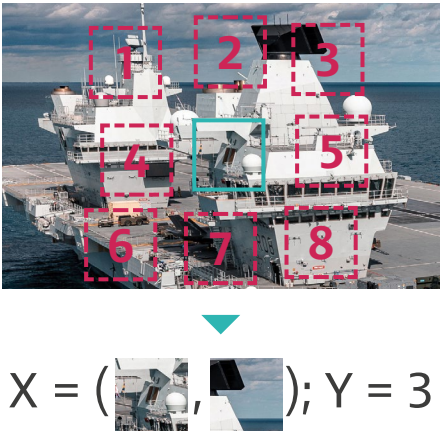
Self-supervised learning is a type of unsupervised learning where the data is autonomously labelled to provide the supervision. Self-supervised learning uses learning objectives in a special form to predict only a subset of information using the rest.

By training with the self-supervised learning objective, the model will learn a good representation of the dataset which can be used for other tasks. The self-supervised task therefore doesn't need to be related to the task we want to train against.

Self-supervised tasks

- Relative position in images (Doersch et al. 2015).
- Image patch prediction (Pathak et al 2016).
- Pixel colour from monochrome input (Zhang et al. 2016).
- Video frame sequence ordering (Misra et al. 2016).
- Audio-Visual co-supervision (Relja and Zisserman, 2018).
- Prediction of next word in a sentence (Zhenzhong Lan, et al 2019).

Datatypes – Self-supervised learning has mainly been applied to image, video and text datasets. It could be applied to any dataset given a suitable self-supervised learning objective.



A relative positional self-supervised learning objective (Doersch et al. 2015).

Implementations – Self-supervised learning is a very low TRL technique and therefore there are only a few implementations available on GitHub.

Challenges:

- Self-supervised learning is a very low TRL technique so has not been applied to real-world datasets or problems.

Requirements:

- A large amount of unlabelled data.
- A relevant self-supervised learning objective.

Model refinement

Whilst the approaches to learning with limited data presented in this guide improve on classical approaches to machine learning, these models cannot achieve the accuracy of a deep neural network trained with a large amount of data. There are a number of model refinement techniques which could improve the accuracy and allow effective deployment of limited data machine learning models. Whilst all of these techniques are not specific to machine learning with limited data, they are likely to be particularly powerful and relevant to limited data problems.

Approaches

- **Transfer learning** – Using datasets with a similar distribution to initially train the dataset, or pretrained models, then fine tune on the dataset of interest.
- **Uncertainty** – Modelling the uncertainty in the problem to know when the model can be trusted.
- **Context injection** – Using contextual information to improve the performance of machine learning models.
- **Data augmentation** – Generating more data through transformations on the existing dataset.
- **Generated data** – Generating more data for use in training.

Transfer learning

TRL: 3-6

When training machine learning models, part of the training effort is to learn how to extract features from the data. For problems on similar datasets, the feature extraction part of the neural network will be very similar. Transfer learning looks to reduce the amount of training data required to learn a task, by reusing the feature extraction layers learned on other datasets. Transfer learning can be used in conjunction with the approaches to learning with limited data to boost the performance of these models.

One approach to transfer learning is to take a deep learning model trained on a large dataset, with a similar distribution to the problem of interest and fine tune or replace and train the final few layers of the neural network on the new problem. It is generally understood that the first layers of a neural network are performing general feature extraction and the final layers are problem specific. The number of layers to fine-tune or retrain is generally chosen by heuristics and guesswork. There has been an effort to quantify the generality of layers for transfer learning (including Yosinski et al. 2014).

One type of transfer learning which has been well studied is that of Domain Adaptation. Domain adaptation refers to transfer learning when the sample and label spaces are unchanged, but the examples come from different probability distributions. In Defence, we may be able to train a vehicle classifier using imagery from a ground based camera, and use domain adaptation to transfer this classifier to overhead imagery.

Performance – Transfer learning has been demonstrated to be an effective approach to improving performance of machine learning models on limited data problems when there is a large dataset available which is similar to the task problem. For some problems, transfer learning can outperform limited data machine learning approaches, such as few-shot learning.

Implementations – Transfer learning can be easily applied in most machine learning libraries. A number of pre-trained models have been released to allow transfer learning such as Google's Big Transfer model (Kolesnikov et al. 2019).

Uncertainty

TRL: 3

In machine learning there are many sources of uncertainty, but classical machine learning does not model the uncertainty in the model. The uncertainty in machine learning can be split into two types:

Aleatoric uncertainty – The inherent variation in a physical system or environment. Otherwise known as statistical or stochastic uncertainty.

Epistemic uncertainty – The uncertainty due to a lack of knowledge of the system or environment.

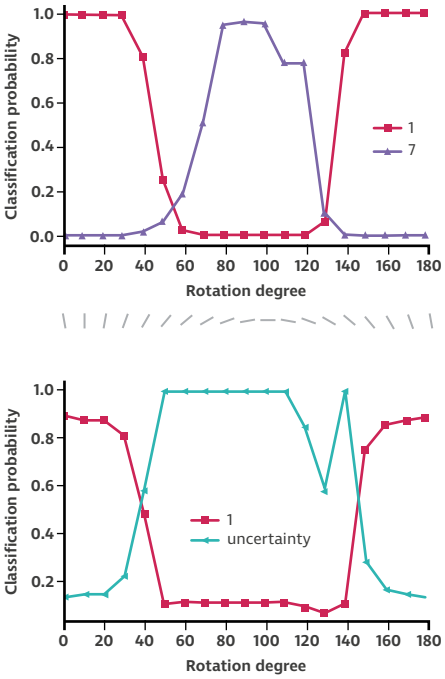
For limited data problems, the epistemic uncertainty will be high. To be able to effectively apply limited data machine learning approaches to real-world problems, approaches to handle the uncertainty need to be considered.

Confidence estimation – A softmax layer will turn the output of the neural network into a probability distribution. This gives a representation of the uncertainty of the prediction.

Dirichlet loss – Learning an uncertainty measure through a loss function based on the Dirichlet distribution (Sensoy et al 2018).

Bayesian machine learning – Bayesian neural networks take distributions for the weights, rather than values used in standard neural networks. This allows the neural network to represent and learn the uncertainty in prediction. Bayesian neural networks are trained using Bayes theorem.

Dropout as Bayesian approximation – Training Bayesian neural networks is difficult and deep networks can become intractable. Instead Bayesian neural networks can be approximated using dropout (Gal 2016).



Prediction of a neural network as a digit 1 is rotated, and the Dirichelt loss uncertainty (BAE, 2019).

Context injection

TRL: 3

In many machine learning problems, there is metadata or contextual information available. With typical machine learning models, this context is thrown away, but incorporating this metadata or context into the machine learning model can improve the accuracy of the model on the problem of interest. Context injection approaches look to augment typical machine learning models with the contextual information.

Examples of contextual information

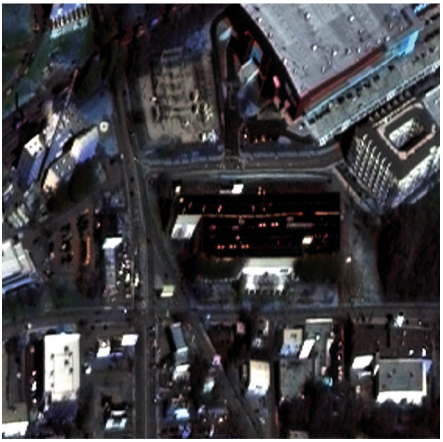
Temporal information – Time of day, Season.

Spatial information – Region, Latitude and Longitude.

Collection information – Nadir angle.

There are a number of different ways context can be included in a machine learning model. Including context in a machine learning model can be trained from scratch, with end-to-end approaches, or can augment an already trained model by combining the prediction and context distributions to make a final prediction.

Datatypes – Context injection has largely been applied to imagery data for classification or detection problems. At Dstl we have done research to demonstrate that context injection can also be effectively applied to audio classification.



An example of off-nadir imagery from the SpaceNet-4 dataset (SpaceNet 2018).

Implementations – There are a number of context injection models available on GitHub from recent papers and successful contestants in the SpaceNet 4 challenge.

Performance – Context injection has been demonstrated to improve the performance on a number of problems. For overhead imagery, a model including the nadir angle as context achieved the best performance on the SpaceNet 4 challenge. We have demonstrated up to 7x improvement in classification accuracy by including context in an audio classification problem.

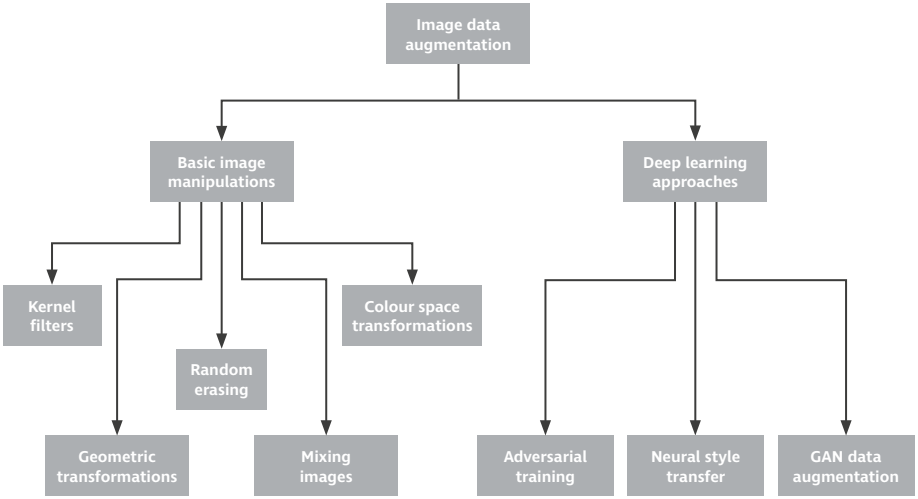
Data augmentation

TRL: 2-6

Data augmentation is a technique widely employed in machine learning, not just for limited data problems, to increase the variation in the dataset. Data augmentation exploits invariances in the dataset to generate new examples from the available dataset.

For larger datasets, simple transformations such as reflection, rotation and translation are effective. There are also more complex approaches to data augmentation such as Generative Adversarial Networks and Style Transfer Networks. In the limited data scenario, data augmentation needs to be applied more carefully in order to avoid reinforcing the biases in the limited amount of data.

Implementations – There are a number of common implementations of simple transformations. The standard machine learning libraries like TensorFlow and PyTorch have simple augmentations built in, and a wider range of transformations are available in python libraries such as ImgAug. For the deep learning approaches, many of the models have GitHub repositories containing code for the approaches.



A taxonomy of data augmentation approaches from (Shorten and Khoshgoftaar, 2019).

Generated data

TRL: 3

When training data is limited, data augmentation may not be sufficient to effectively train machine learning models. In particular, data augmentation often cannot generate the variation required to represent the full task distribution and therefore cause overfitting of the model. Generating similar data can increase the variation in the training data to ensure more of the task distribution can be represented.

Simulated data – Simulating data in artificial environments, such as game engines, can be an effective way to generate more data for training. For simulated imagery, game engines such as Unity and Unreal can be used to generate realistic imagery.

Generative neural networks – Generative Adversarial Networks (GANs) have become an effective way to generate data based off existing datasets. GANs are trained adversarially to generate more realistic data and, depending on structure, can be seeded by existing examples or random noise to generate new data. There are many variants of GANs which have been developed for different applications, many of these have code available on GitHub. Generative neural network approaches require a large amount of training data, but there have been some GAN approaches developed for data generation when data is limited, such as DAGAN (Antoniou et al. 2017).

A GAN generated face.

There are also non-GAN approaches to data generation for limited data problems, such as the Delta-Encoder (Schwartz et al. 2018), which learns to apply translations in populous classes and apply these translations to under-represented classes.

Technology readiness levels

TRL	Description
TRL 1	Basic principles observed and reported.
TRL 2	Technology concept and/or application formulated.
TRL 3	Analytical and experimental critical function and/or characteristic proof-of-concept.
TRL 4	Technology component and/or basic technology subsystem validation in a laboratory environment.
TRL 5	Technology component and/or basic technology subsystem validation in a relevant environment.
TRL 6	Technology system/subsystem model or prototype demonstration in a relevant environment.
TRL 7	Technology system prototype demonstration in an operational environment.
TRL 8	Actual Technology system completed and qualified through test and demonstration.
TRL 9	Actual Technology system qualified through successful mission operations.

References

- Akkaya, Ilge, et al. "Solving Rubik's Cube with a Robot Hand." *arXiv preprint arXiv:1910.07113* (2019).
- Antoniou, Antreas, Amos Storkey, and Harrison Edwards. "Data augmentation generative adversarial networks." *arXiv preprint arXiv:1711.04340* (2017).
- Arandjelovic, Relja, and Andrew Zisserman. "Objects that sound." *Proceedings of the European Conference on Computer Vision (ECCV)*. 2018.
- Doersch, Carl, Abhinav Gupta, and Alexei A. Efros. "Unsupervised visual representation learning by context prediction." Proceedings of the IEEE International Conference on Computer Vision. 2015.
- Everingham, Mark, et al. "The pascal visual object classes (voc) challenge." *International journal of computer vision* 88.2 (2010): 303-338.
- Finn, Chelsea, Pieter Abbeel, and Sergey Levine. "Model-agnostic meta-learning for fast adaptation of deep networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.
- Gal, Yarin, and Zoubin Ghahramani. "Dropout as a bayesian approximation: Representing model uncertainty in deep learning." *International conference on machine learning*. 2016.
- Lake, Brenden M., Ruslan Salakhutdinov, and Joshua B. Tenenbaum. "Human-level concept learning through probabilistic program induction." *Science* 350.6266 (2015): 1332-1338.
- Lampert, Christoph H., Hannes Nickisch, and Stefan Harmeling. "Learning to detect unseen object classes by between-class attribute transfer." *2009 IEEE Conference on Computer Vision and Pattern Recognition*. IEEE, 2009.
- LeCun, Yann, et al. "Gradient-based learning applied to document recognition." *Proceedings of the IEEE* 86.11 (1998): 2278-2324.
- Li, Xiang, et al. "Fss-1000: A 1000-class dataset for few-shot segmentation." *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 2020.
- Miller, George A. *WordNet: An electronic lexical database*. MIT press, 1998.
- Mishra, Nikhil, et al. "A simple neural attentive meta-learner." *arXiv preprint arXiv: 1707.03141* (2017).
- Misra, Ishan, C. Lawrence Zitnick, and Martial Hebert. "Shuffle and learn: unsupervised learning using temporal order verification." *European Conference on Computer Vision*. Springer, Cham, 2016.
- Munkhdalai, Tsendsuren, and Hong Yu. "Meta networks." *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, 2017.

- Kolesnikov, Alexander, et al. "Big transfer (BiT): General visual representation learning." *arXiv preprint arXiv:1912.11370* (2019).
- Pathak, Deepak, et al. "Context encoders: Feature learning by inpainting." *Proceedings of the IEEE conference on computer vision and pattern recognition*. 2016.
- Schwartz, Eli, et al. "Delta-encoder: an effective sample synthesis method for few-shot object recognition." *Advances in Neural Information Processing Systems*. 2018.
- Sensoy, Murat, Lance Kaplan, and Melih Kandemir. "Evidential deep learning to quantify classification uncertainty." *Advances in Neural Information Processing Systems*. 2018.
- Shorten, Connor, and Taghi M. Khoshgoftaar. "A survey on image data augmentation for deep learning." *Journal of Big Data* 6.1 (2019): 60.
- Smith, Linda, and Michael Gasser. "The development of embodied cognition: Six lessons from babies." *Artificial life* 11.1-2 (2005): 13-29.
- Snell, Jake, Kevin Swersky, and Richard Zemel. "Prototypical networks for few-shot learning." *Advances in neural information processing systems*. 2017.
- SpaceNet on Amazon Web Services (AWS). "Datasets." The SpaceNet Catalog. Last modified April 30, 2018. Accessed on [August 20, 2020]. https://spacenetchallenge.github.io/datasets/datasetHomePage.html.
- Sung, Flood, et al. "Learning to compare: Relation network for few-shot learning." *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*. 2018.
- Tanner, Franklin, et al. "Overhead imagery research data set—An annotated data library & tools to aid in the development of computer vision algorithms." *2009 IEEE Applied Imagery Pattern Recognition Workshop (AIPR 2009)*. IEEE, 2009.
- Tian, Pinzhuo, et al. "Differentiable Meta-Learning Model for Few-Shot Semantic Segmentation." AAAI. 2020.
- Vinyals, Oriol, et al. "Matching networks for one shot learning." *Advances in neural information processing systems*. 2016.
- Yosinski, Jason, et al. "How transferable are features in deep neural networks?" *Advances in neural information processing systems*. 2014.
- Zhang, Richard, Phillip Isola, and Alexei A. Efros. "Colorful image colorization." *European conference on computer vision*. Springer, Cham, 2016.
- Lan, Zhenzhong, et al. "Albert: A lite bert for self-supervised learning of language representations." *arXiv preprint arXiv:1909.11942* (2019).
- Xiao, Jianxiong, et al. "Sun database: Large-scale scene recognition from abbey to zoo." *2010 IEEE computer society conference on computer vision and pattern recognition*. IEEE, 2010.

Further information

The approaches to learning with limited data presented in this guide are a part of ongoing research carried out by the Future of AI for Defence project at Dstl. For the approaches presented we are carrying out extensive research to understand the current state-of-the-art, develop the theory to benefit Defence and demonstrate how these approaches for learning with limited data could be applied to Defence problems.

For further information on these approaches to learning with limited data, contact:

E lowshot_learning@dstl.gov.uk

[dstl]
The Science Inside

T +44 (0) 1980 950000
E centralenquiries@dstl.gov.uk

For more information about Dstl's work,
visit **www.gov.uk/dstl**

DSTL/PUB126061