**Training Machine Learning Models [ready]**

The standard method to train machine learning models to classify images is to split a dataset into batches and use these batches to repeatedly optimise the parameters of a model via backpropagation. This is known as Offline Training.

**[Use Laser]** As Figure 1 shows, Offline Training is capable of producing high-quality results. The chart shows the classification accuracy of a ResNet-18 model trained for 100 epochs on the CIFAR-10 dataset where it attains an average classification accuracy of over 77%.

However, Offline Training is limited by the requirement to have the dataset available in its entirety at training time. If it is infeasible to do so, or the data is real-time then Offline Training is unsuitable due to a phenomenon known as Catastrophic Forgetting.

**Catastrophic Forgetting [review]**

Catastrophic Forgetting is defined as the complete and sudden loss of previously acquired knowledge in a neural network.

It is caused by training an already trained model on unseen data. The model overwrites the weights and biases it has learnt previously causing it to forget the knowledge it has gained. The ability of a neural network to learn continuously is often overlooked and, at a high-level, is caused by the lack of context that existing weights represent acquired knowledge.

**Example of Catastrophic Forgetting [ready]**

In this example, I have split the CIFAR-10 dataset into 5 disjoint tasks each containing 2 classes without replacement. A ResNet-18 model is then trained sequentially on the tasks for 20 epochs each. **[Use Laser]** as you can see by the charts, the model forgets about the previous tasks and overwrites the knowledge.

The final graph shows the loss peaking as each new task is introduced and decreasing as the model is trained.

**Continual Learning [ready]**

Continual Learning is the study of techniques to reduce the effect of Catastrophic Forgetting while trying to match the performance of Offline Training. Elastic Weight Consolidation, a technique that penalises changes to weights that are important to previous tasks, was one of the first techniques that gave rise to the field.

The literature since then has mainly focused on alternatives to Offline Training, but there has been a recent interest into how the architecture of a model may contribute to forgetting. The main difference from Offline Training is the relaxation of the requirement for the whole dataset to be present at training time.

**Part I: Literature Evaluation [ready]**

Initially, the focus of my project will be on implementing and comparing techniques from the literature. This is important because of ongoing issues in the literature which were highlighted by the authors of a technique known as GDumb. They argued that many techniques in the literature were not applicable to the real-world because of assumptions that their authors make. As such, it is difficult to fairly compare recent state of the art techniques.

This will start with implementation of the baselines: Offline Training and Finetuning, which is similar to the catastrophic forgetting example shown earlier and represents the lower bound. The goal is to match Offline Trainings results.

After this, I will implement historical techniques such as Elastic Weight Consolidation as discussed previously. And following this, I will implement the current state of the art such as GDumb, a method that stores a subset of data and trains a classifier at inference time, Rainbow, another method that samples data that is representative of its own class and discriminative against others, and Mnemonics which is a similar sampling technique. Following this, I will also implement cutting edge techniques that present new ideas such as Learning to Prompt, a technique that learns small inputs to be prepended to the input prior to classification, and meta-learning techniques that learn how to preserve knowledge during training.

Finally, to evaluate and compare these methods I will conduct fair experiments and use metrics such as overall accuracy, time taken to train, memory consumption, and continual learning specific metrics such as average forgetting which measures the decrease in performance between tasks where applicable.

**Part II: Novel Experimentation [ready]**

Following this, I will focus on experimenting with gaps in the literature with the aim to contribute something novel to the field. This will be influenced by the cutting-edge techniques as well as potentially experimenting with the architecture of the model. This will involve explaining and understanding the theoretical underpinning, implementing these ideas, and comparing them to existing techniques using the same metrics previously mentioned.

Some potential avenues for this part are exploring improvements to sampling techniques, experimenting feature extraction methods, and investigating dynamic network structures.

**Applications of Continual Learning [ready]**

Continual Learning offers real-world benefits. Training, and retraining, models is costly in terms of energy usage, computational resources, and time. If Continual Learning research is able to propose techniques that are capable of preserving knowledge in existing models this has the potential to significantly reduce the cost of updating models.

Furthermore, there would be major benefits for real-time applications as well. For example, incoming data could be used to update a model on the fly in near real-time to improve the output of a model incrementally. This could give adopters of Continual Learning an edge over competitors while also realising the benefits of cost reductions at the same time.

It is also important to note that Image Classification is the main domain used for research, but the techniques can also be applied to other domains such as Natural Language Processing.

**Setup for Continual Learning**

**[slim this down]**

The setup of the Continual Learning is vitally important. This has been a key issue in the literature as highlighted by the authors of a technique known as GDumb. They are argue that many techniques in the literature have too simplistic setups to be applicable to the real-world.

There are a few key factors affecting the difficulty of the problem:

* Online vs Offline: Online assumes that the data is streamed continuously and not fixed unlike Offline Training
* Disjoint vs Overlapping: Disjoint task formulations partitions the problem into tasks with distinct classes without overlap. Non-disjoint is more real-world applicable as seen in Figures 2 and 3.
* Class Incremental vs Task Incremental: Task Incremental uses the disjoint task formulation but importantly the model is told which task a sample belongs to at classification time greatly simplifying the problem. Class Incremental can use either disjoint or overlapping and is more realistic
* Imposing limits on the resource consumption of the techniques is also important to ensure real-world applicability