Model Up-Keep with Continual Learning

Kishansingh Rajput

Jefferson Lab, Newport News, VA, USA

1

Jefferson Lab

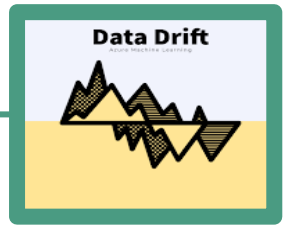
U.S. DEPARTMENT OF ENERGY | Office of Science

JSA

Outline

01

Introduction and Motivation



02

Data Drifts

03

Continual Learning Pipelines

04

CL in Neural Network; Challenges

05

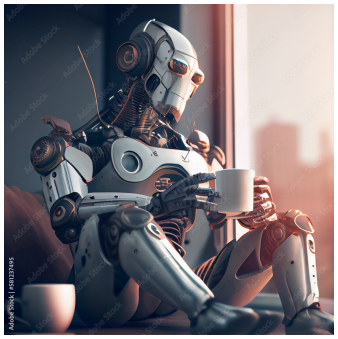
Summary

Hands on example!

Introduction

ML Models and "the hypothesis"

- Machine Learning (ML) models are trained on historical data or simulations
- "the hypothesis": Training done! Everything I will see from now on will be within my training data distribution ☺
- The world is evolving…
- ML Models fail on unseen data distribution



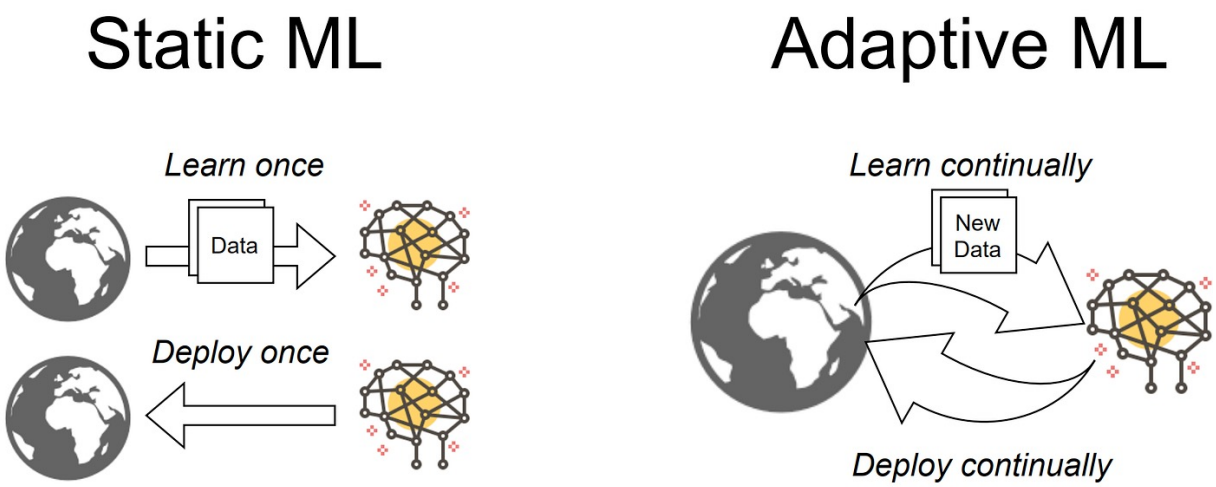
AI Today: Impressive but Narrow

AI is impressive

But still narrow

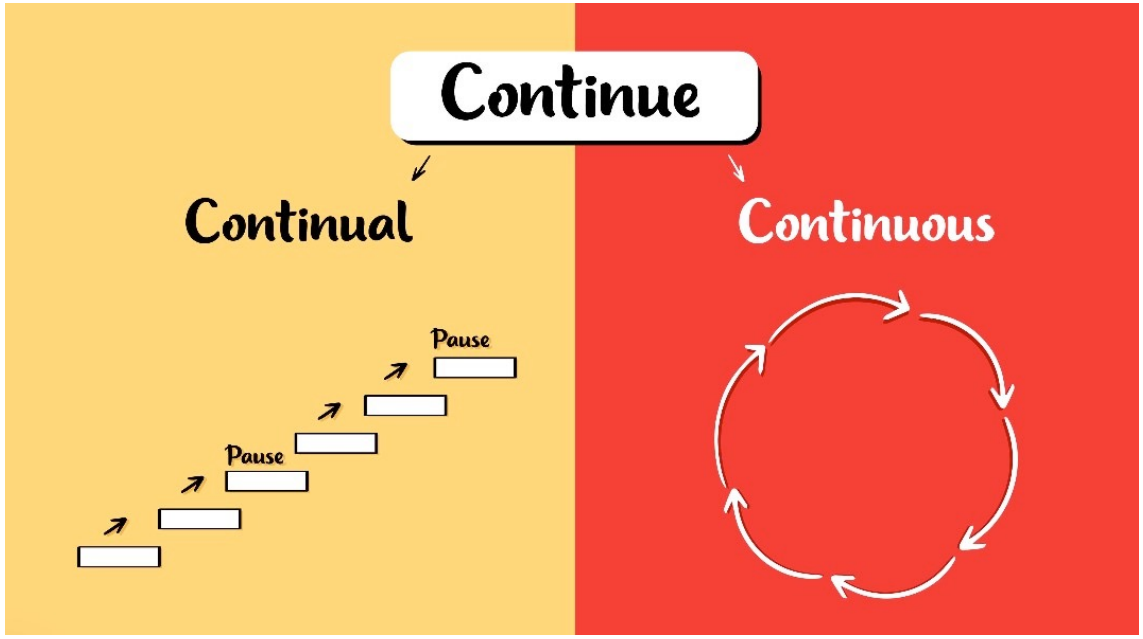
What is Continual Learning?

- Ability to continually acquire new knowledge and adapt to changing data over time
- Traditional ML → Static training data → Static Model
- Continual Learning → learn from and apply to evolving data streams
- Incremental learning or Lifelong learning



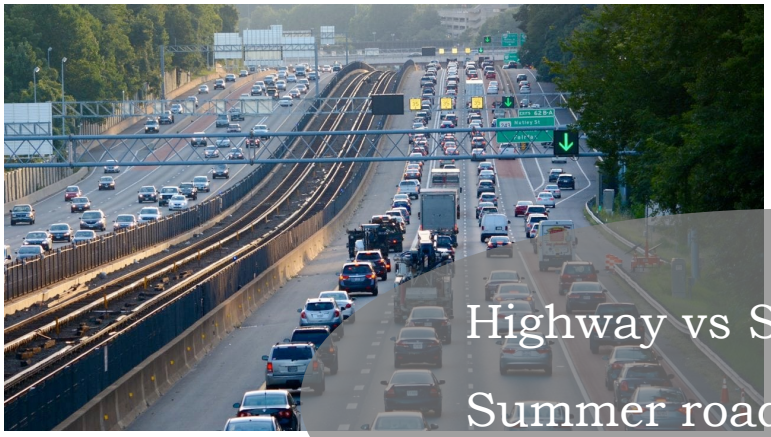
Continual vs Continuous Learning

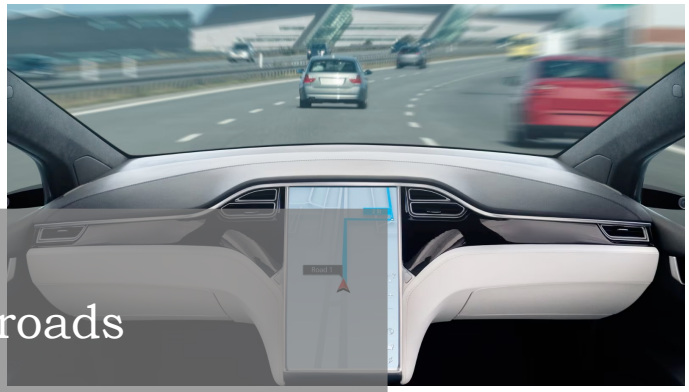
- "Continual" refers to something happening at regular intervals (with breaks in-between)
- "Continuous" refers to something happening continuously without interruption



Practical Application of Continual Learning

- Self driving cars adapting to different environments





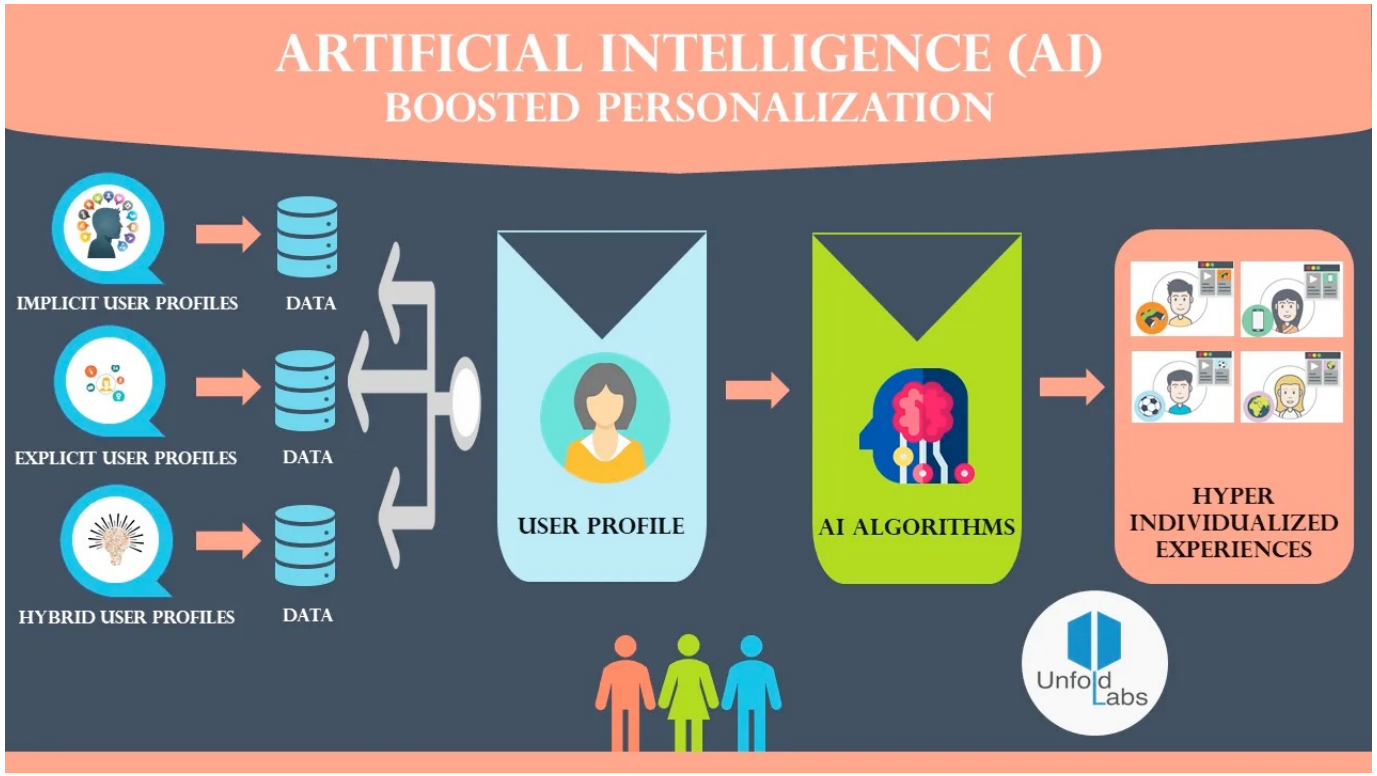
Highway vs Street roads

Summer roads vs snow

Different road construction

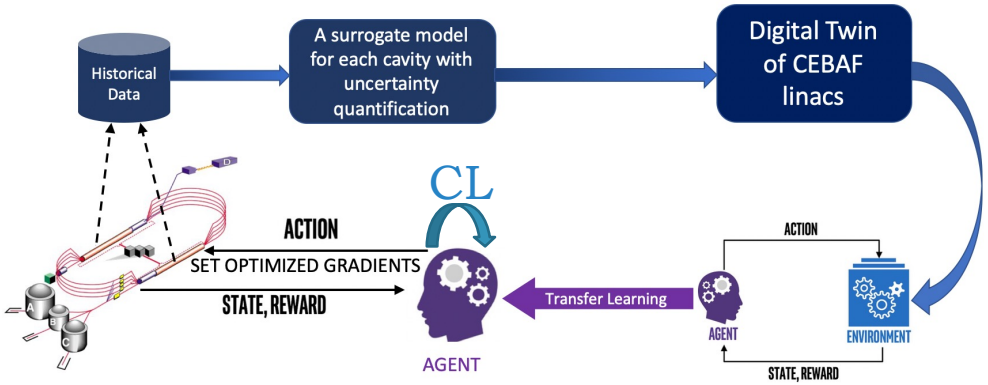
Practical Application of Continual Learning

- Language models adapting to different users / languages



Practical Application of Continual Learning

- ML models tuning particle accelerators adapting to system aging, equipment replacements, etc.

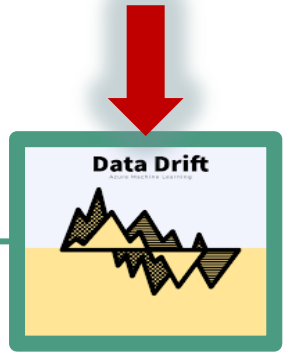


Adapting to changing conditions
- environmental conditions
- Equipment degradation
- and others…

Outline

01

Introduction and Motivation



02

Data Drifts

03

Continual Learning Pipelines and its components

04

CL in Neural Network; Challenges

05

Summary

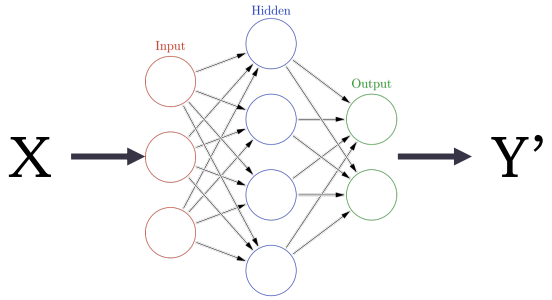
Drifts

- Changes or evolution in the input data or **concept**

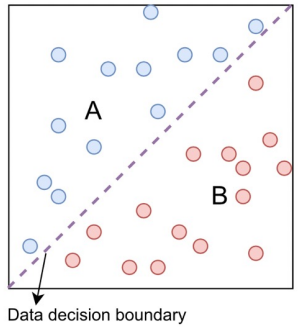
Input	Target
X_1	Y_1
X_2	Y_2
.	.
X_n	Y_n

$$Y = F(X)$$

Concept



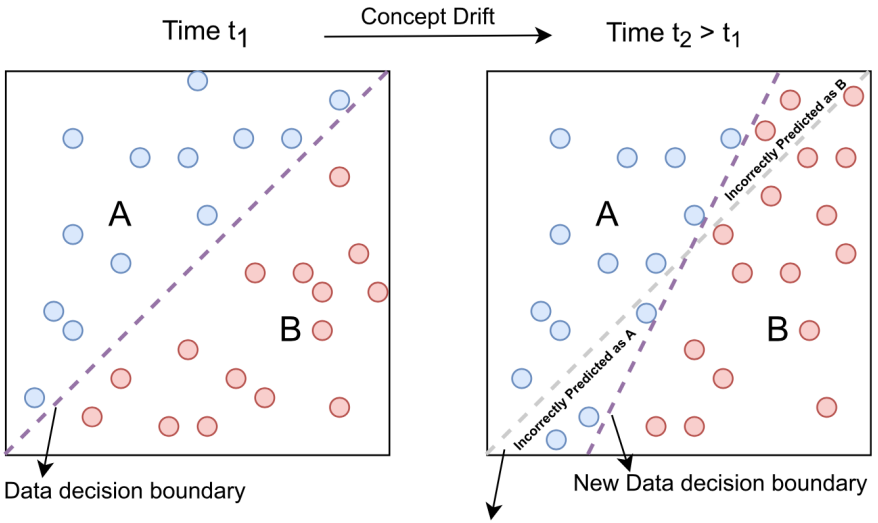
Concept in Classification

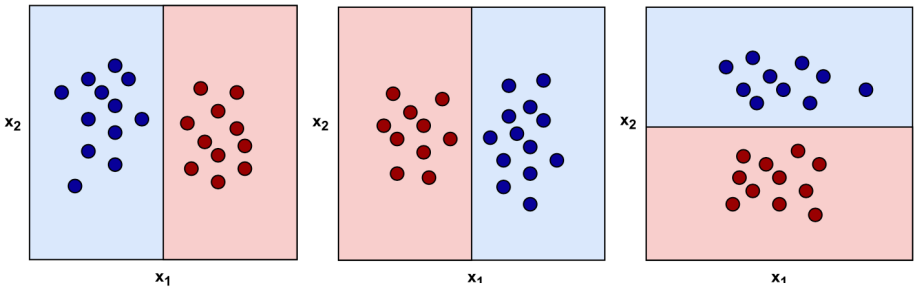


Data Drifts		Concept Drifts
• Drift in the input data X • Out of training distribution data can degrade performance	Vs	• Learnt concept changes • Model is unaware of new concept

Concept Drift

Concept change over time

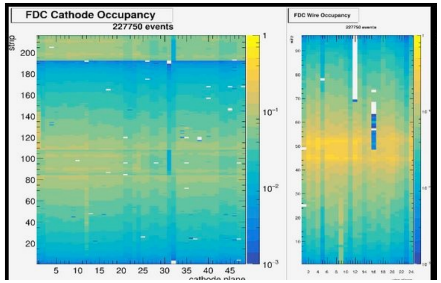


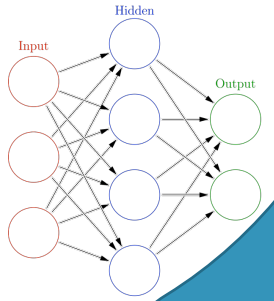


(a) Original data (b) Virtual Drift (c) Real Drift

Source: https://www.sciencedirect.com/science/article/pii/S0957417422019522?via%3Dihub

FDC:

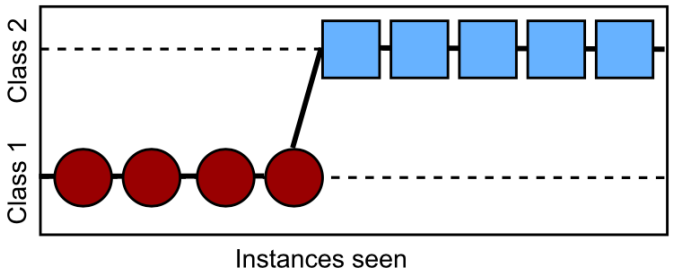




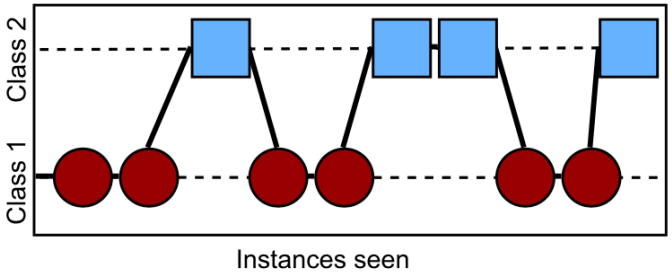
Anomaly!
Anomaly!

Well, I know those strips are not working, consider it normal!

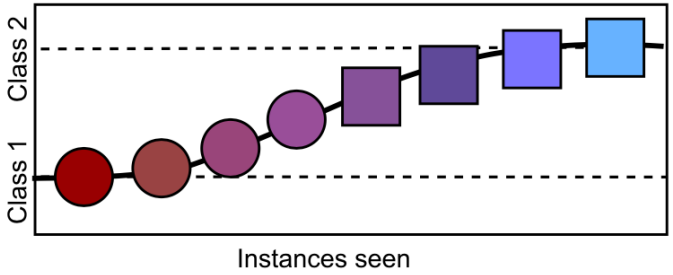
Types of Drifts



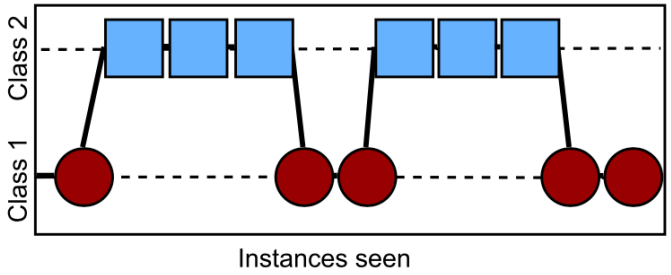
(a) Abrupt



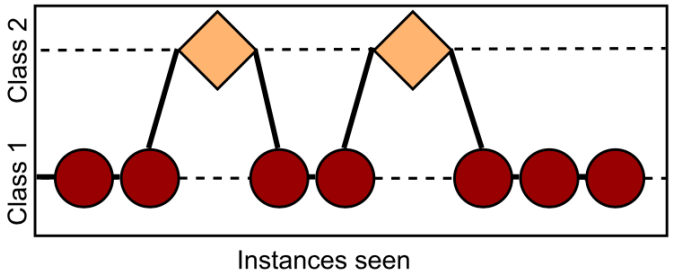
(b) Gradual



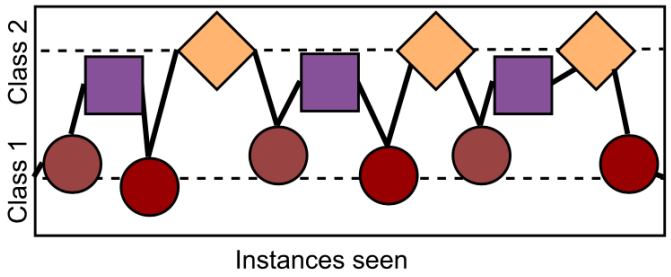
(c) Incremental



(d) Recurring



(e) Blips



(f) Noise

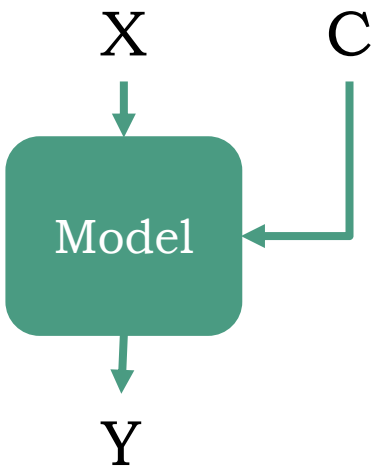
Distribution/Data Drifts

- Effect of measurable parameters
- Usually abrupt changes but can be gradual as well
- Periodicity can also be seen
- Example: Beam Tuning changing beam current waveforms
- Running at different energies
- Others…

- Effect of non-measurable parameters
- Usually gradual but can be abrupt in some cases
- Periodicity is common
- Example: drift due to machine aging in accelerators
- Equipment degradation
- Replacement of parts
- Others…

Conditional Models

- Conditional models to include effect of measurable parameters on input data
- Additional input to the model that represent current condition
- For example: Anomaly prediction using beam current data → conditional input = Beam configuration parameters that changes beam current data distribution

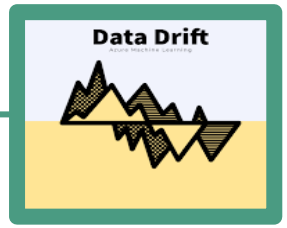


$$Y = F(X|C)$$

Outline

01

Introduction and Motivation



02

Data Drifts

03

Continual Learning Pipelines and its components

04

CL in Neural Network; Challenges

05

Summary

Data Science Pipeline

Development

- Real or synthetic
- Quality
- Dimensionality
- Format
- Density
- Size

- Data cleaning
- Data restructuring
- Correlations
- Dynamics
- Visualization

- Classification
- Regression
- Clustering
- Feature extraction

- Cross-validation
- HPO
- ROC
- Error metrics

- Deploy the model

Data Science Pipeline

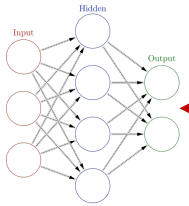
What is being done!

Sensors

Data

Input Data

Data Source | Data Preparation | ML Approach | Training and Testing | Deployment



ML Model

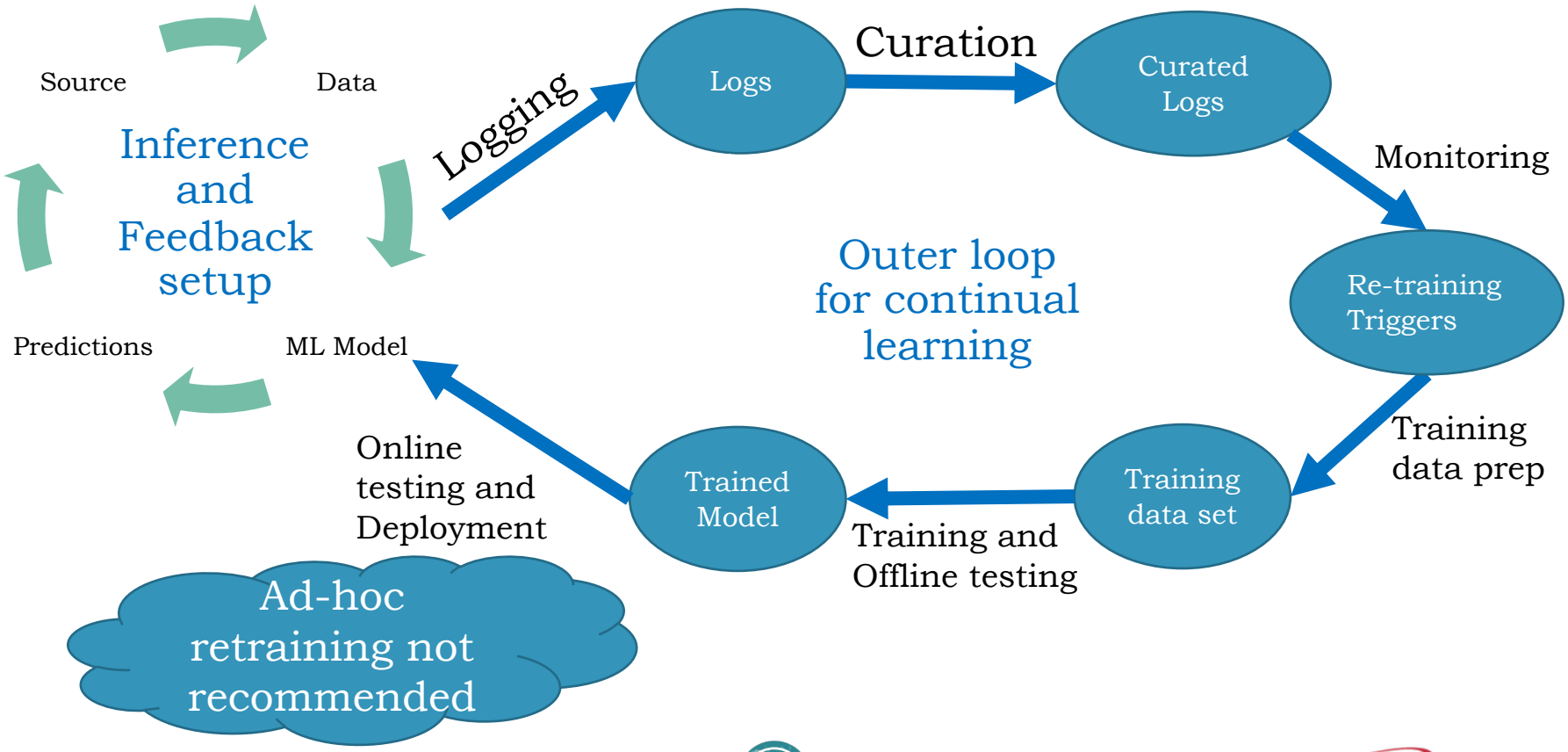
Periodic re-deployment

Predictions

Prediction
(w/ uncertainty)

Continual Learning Pipeline

The outer loop (Retraining Strategy)



Periodic re-training

A Baseline

What to Log?

- All the data being inferred on
- All the predictions
- All the evaluation metrics relevant
- Any user feedback

Limitations

More data than capacity

Profile: Statistical profile of the data to decide what to log

Sample: Sample and store

Periodic re-training

A Baseline

Curation

- Maximum amount of data for the capacity (storage, processing and Labeling)
- Sample uniformly/relevant sampling strategy
- Label the data if applicable
- Outlier checks on the data and cleaning results

Limitations

Rare events/edge cases

Labeling is expensive

Periodic re-training

A Baseline

Monitoring and Trigger

- Periodic re-training
- Monitoring to understand model performance over time

Limitations

Re-training unnecessarily

Sudden drifts will not always coincide with periodic re-training

Periodic re-training

A Baseline

Training data preparation

- Rolling window over logged curated dataset
- Prioritized replay
- Advanced data selection techniques

Limitations

Cost of not including rare events or edge cases

Periodic re-training

A Baseline

Training and offline testing

- Model re-training using previous state
- Advance re-training strategies (Transfer Learning, freeze part(s) of the model etc.)
- Evaluate model on test data set (with relevant metrics)
- Use a threshold or manually check the predictive performance

Challenges

Periodic re-training may not be in sync with sudden drifts

Periodic re-training

A Baseline

Online testing and deployment

- Test the model online with new data
- Window size for testing online; threshold on evaluation metrics
- Replace the old model

Custom re-training strategies

- Continual Learning is the least understood part of the ML Lifecycle
- No standards / best practices yet!
- Custom re-training strategies are almost always required in the science use cases
- Example: Particle Accelerator applications; custom monitoring and triggers are required

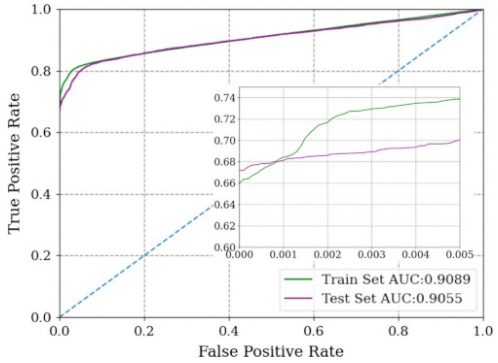
Basic questions when developing custom strategies…

- When should the model be trained?
  - What metrics to monitor for triggers?
  - What data drifts should be monitored?
- What data should be used for training?
  - Only new recent data or mix with old data?
  - What's a good balance?
- What should be re-trained?
  - Simply re-train the same model (use previous state or not)?
  - More intrusive: Re-run the whole pipeline (includes HPO/NAS)

Monitoring

What to monitor?

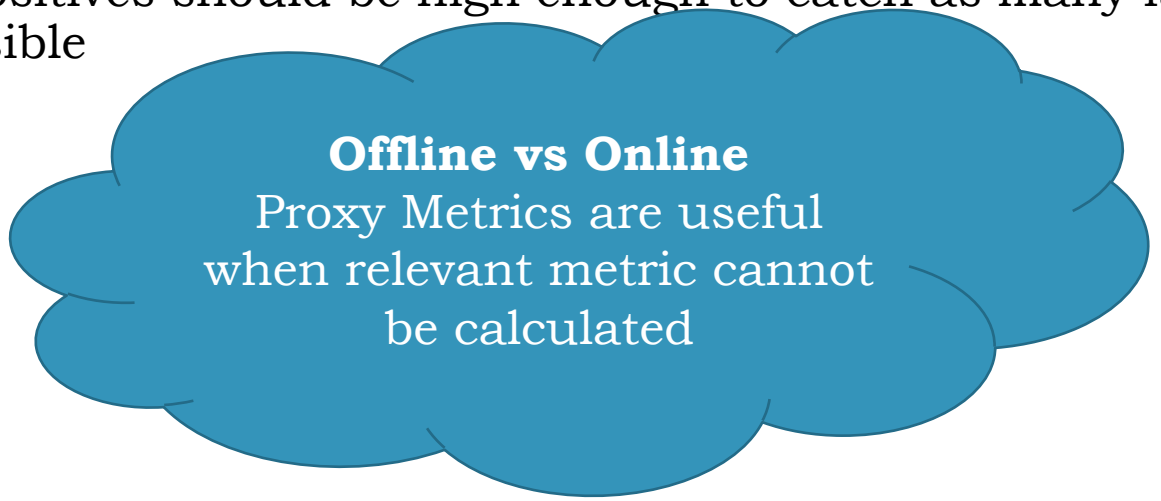
- Model performance metrics and **outcome**, example: False Positive Rate for accelerator anomaly prediction
- Data quality checks, example: schema checks, noise level…
- Distribution Drifts, example: Drift in the input data



- Model outcome and system response, example: How did the system react when certain actions were taken
- Proxy Metrics

Model Performance Metrics based triggers

- Performance metrics will depend on underlying problem and outcome of wrong/right predictions
- Accelerator anomaly prediction: Abort beam when model predicts an incoming fault
  - False positives should not increase downtime
  - True positives should be high enough to catch as many faults as possible



Data Quality Checks

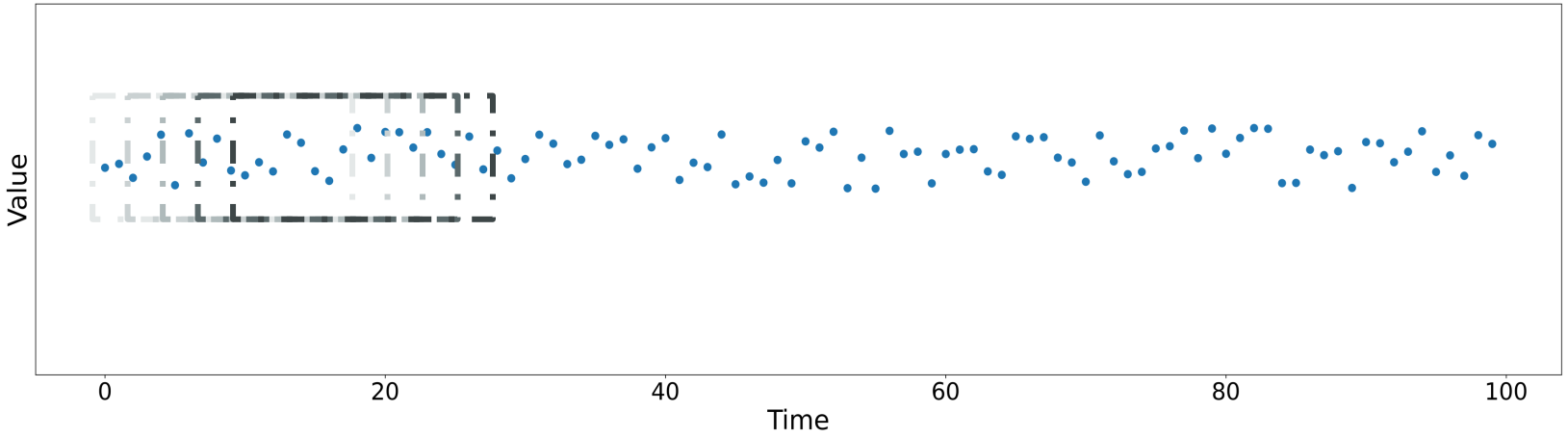
- Accuracy: is the data accurate representation of what's happening underneath?
- Completeness: is the incoming data sample complete? Might break the model if incomplete or missing values!
- Consistency: is the data format consistent?
- Validity: is data sample valid?
- Noise level: is noise level above/below threshold?
- Integrity: Can this sample be added to the pool of collected dataset so far?

An analysis of 15 years of outages in ML pipelines by google:
https://www.usenix.org/sites/default/files/conference/protected-files/opml20_talks_70_slides_papasian.pdf

Distribution Drifts

How to detect distribution drifts?

- Sliding window distance measure
- Each sliding window data subset is compared with reference window

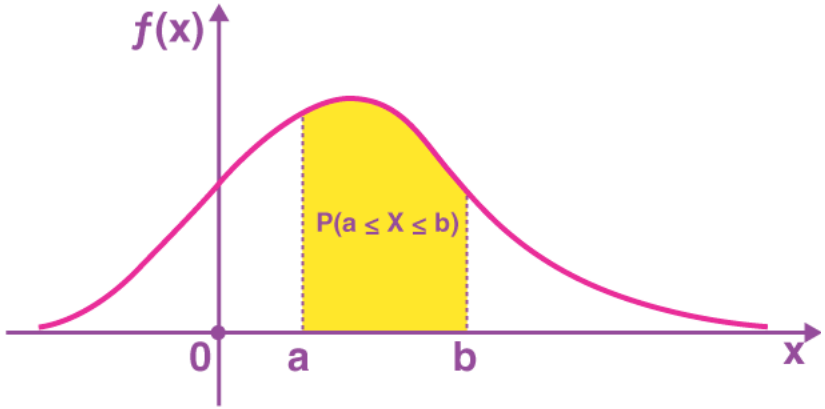


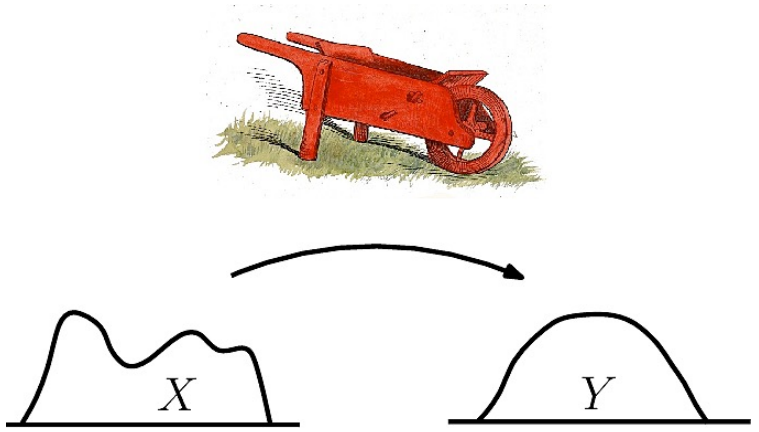
Key questions: How to pick appropriate window size.
Should reference be dynamic or static?

ADaptive WINdowing 2 (ADAWIN2) : https://epubs.siam.org/doi/10.1137/1.9781611972771.42

Distribution Drift: 1D data

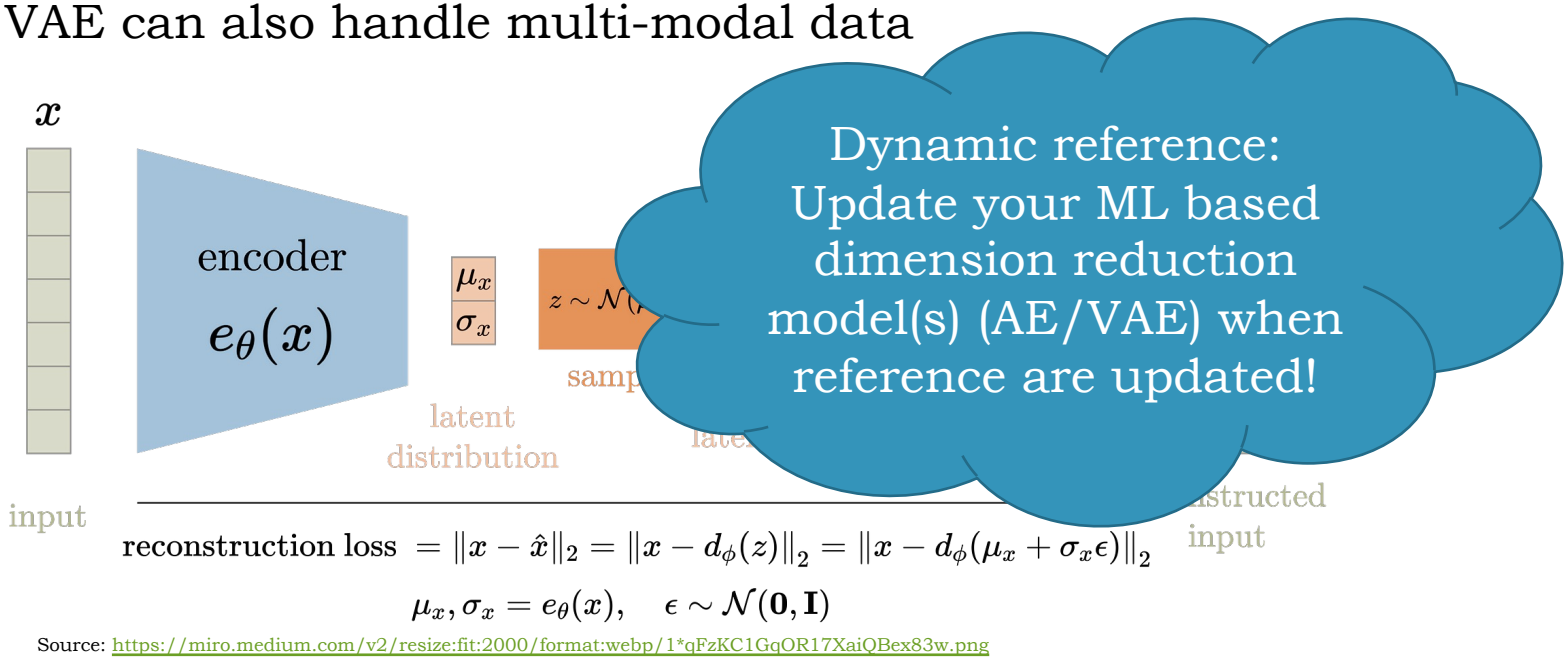
- Does your data follow certain Probability Density?
  - Yes: KL-Divergence, Kolmogorov-Smirnov test etc.
- Earth Mover distance
- Other metrics for distance between 1D time series





Distribution Drift: Multi dimensional data

- Reduce the data into a smaller embedded space using ML/non-ML methods
- Auto-Encoder (AE) or Variational-AE (VAE); reconstruction error
- AE/VAE can also handle multi-modal data

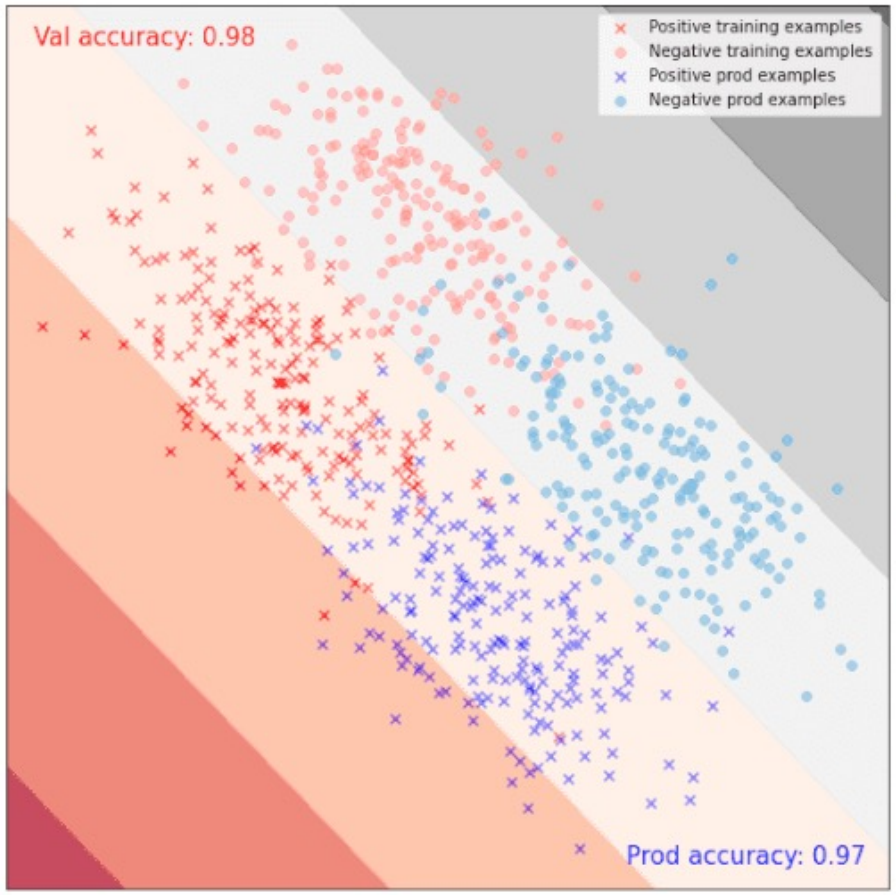


Source: https://miro.medium.com/v2/resize:fit:2000/format:webp/1*qFzKC1GqOR17XaiQBex83w.png

Model Performance vs Distribution Drifts

Which is better?

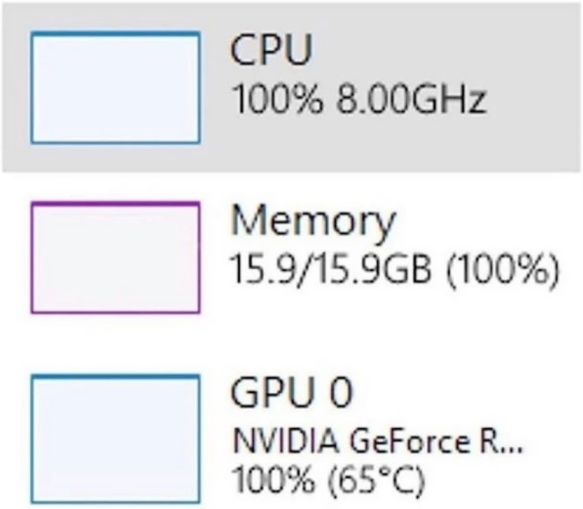
- It depends on underlying problem
- In general, model performance metric is a winner
- Distribution shift might not always degrade your model performance
- Model re-training is expensive



Picture Source:

Metrics for debugging Production Models

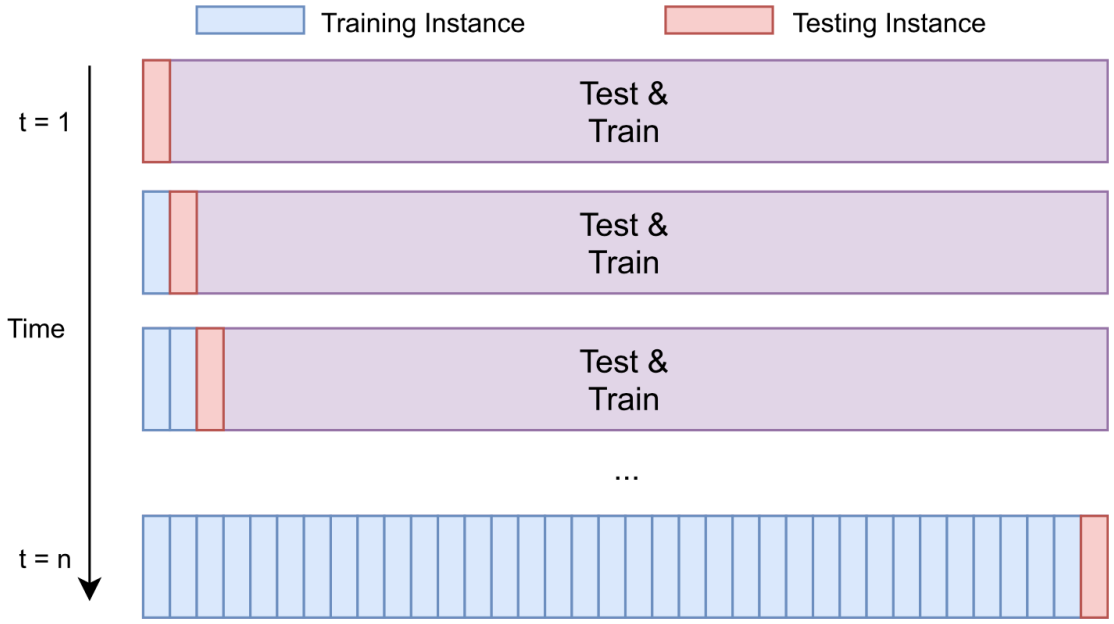
- Monitor basic System metrics, example, CPU and GPU utilizations, memory consumption
- Inference time



Low/high GPU, memory utilization than expected can indicate problems with data/model

Online Learning; Continuous Learning

- Each data instance is used first to predict
- Once the ground truth is known, prediction error is computed
- Finally, the model can be updated, using that instance for training



Data selection for re-training

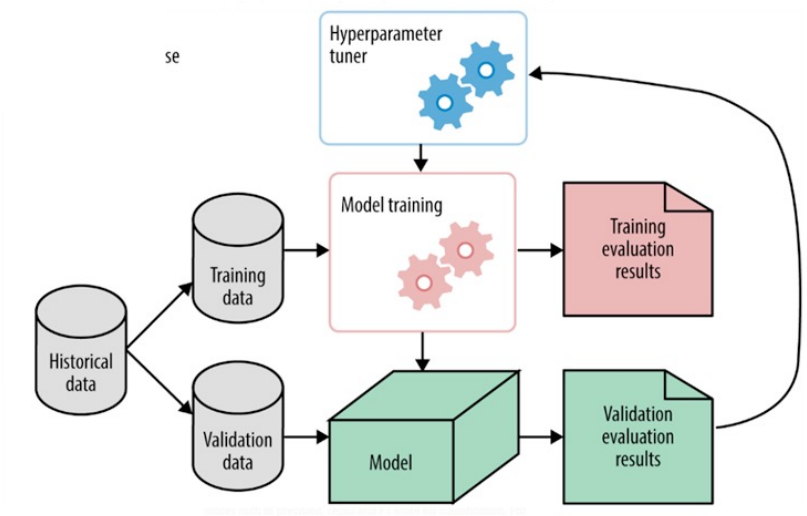
- S1: window
- S2: Sliding window - overlap with old data
- S3: Priority based

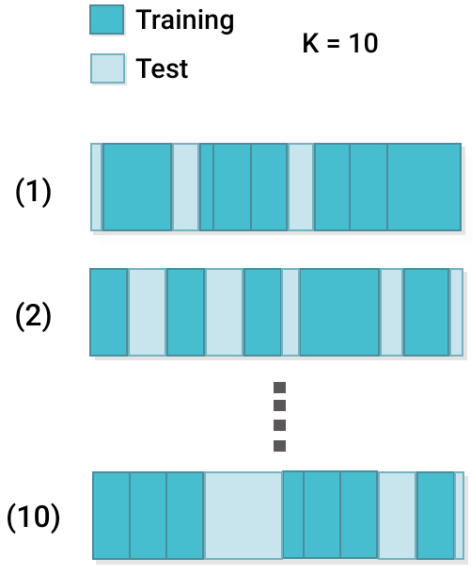
Data Preparation for re-training

- Sample randomly
  - Not all the data is relevant
  - Miss rare events / edge cases
- Balanced sampling
  - From many configuration / states
  - Balance the labels
  - Balance types even for binary classification
- User driven; feedback based
  - Operators marking most interesting types of faults missed
- Based on model performance; priority replay
- Based on model uncertainty
- Active learning, and others strategies…

Offline Evaluation

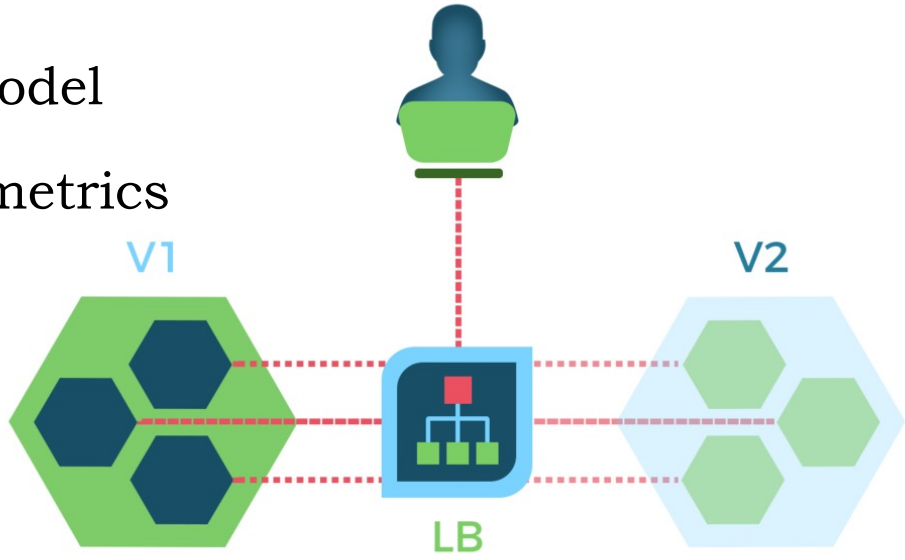
- Compare with previous version of the model
- All the relevant metrics
- If old data distribution can become relevant again; make sure the model works on old data





Online Evaluation

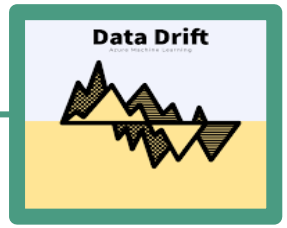
- Run in shadow mode
- Verification and validation; compatibility
- Collect references for system metrics monitoring
- Produce a report card
- Compare with the existing model
- Threshold on most relevant metrics
- Roll-back if needed



Outline

01

Introduction and Motivation

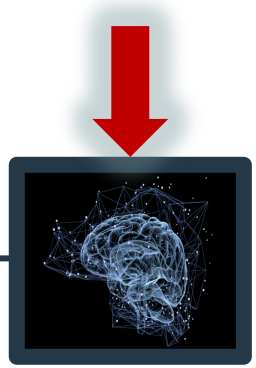


02

Data Drifts

03

Continual Learning Pipelines and its components



04

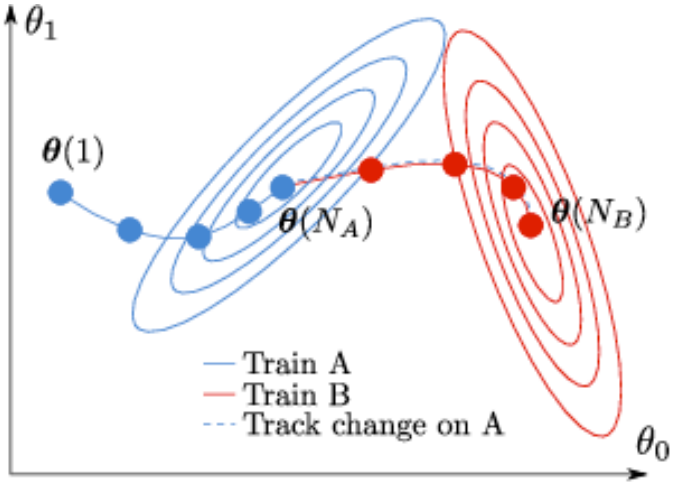
CL in Neural Network; Challenges

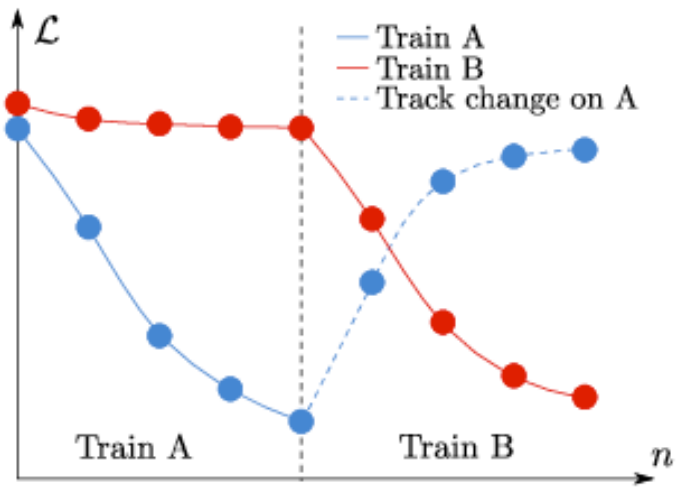
05

Summary

Catastrophic Forgetting

…the process of learning a new set of patterns suddenly and completely erased a network's knowledge of what it had already learned." (French, 1999)



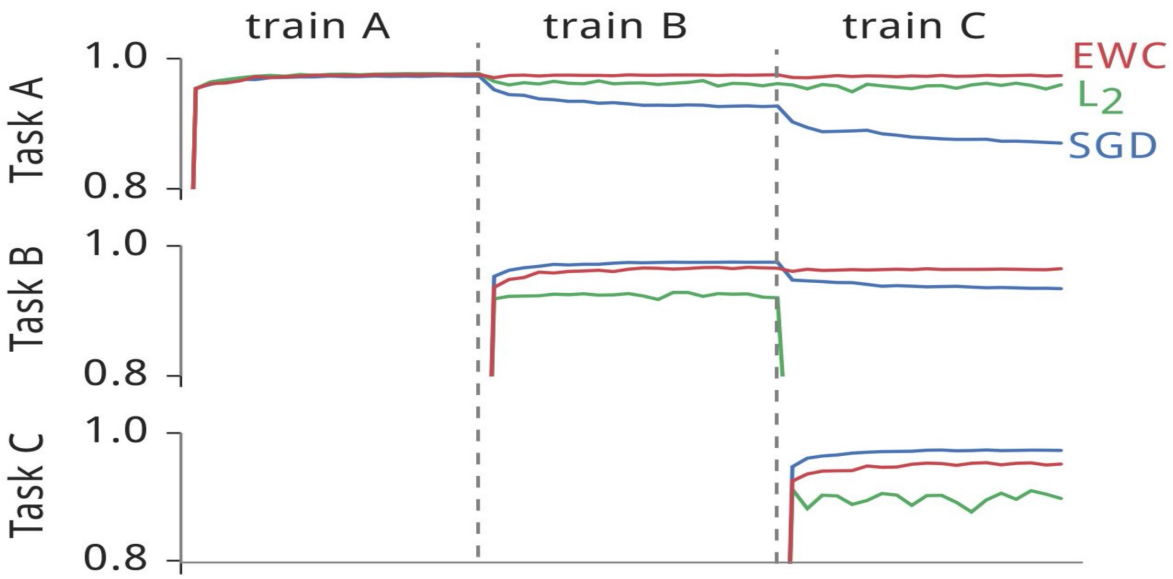


Picture source: https://arxiv.org/abs/1906.02568

Stability vs Plasticity

Plasticity ⇔ ability to adapt to a new task

Stability ⇔ ability to retain the learned skills on the old tasks

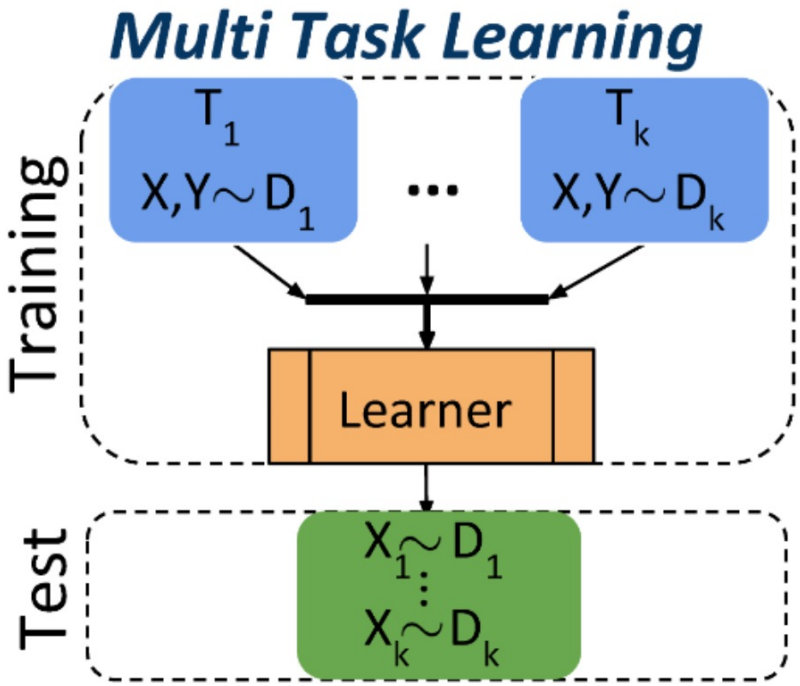


SGD shows catastrophic forgetting

L2 is too stable/rigid

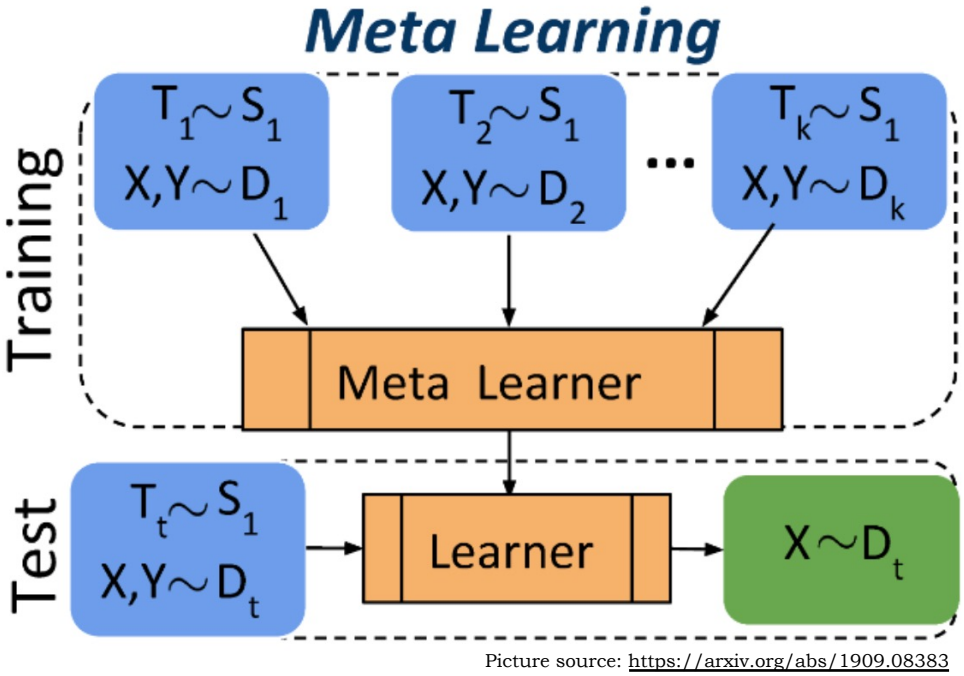
Elastic Weight Consolidation (ECW) provides a reasonable balance between stability and plasticity

Multi-task Learning



- Multiple tasks / datasets learnt at the same time
- All the relevant task datasets are available at the same time
- Does not care about continuously adapt to new tasks/datasets
- Common scenario in Continual Learning settings

Meta Learning



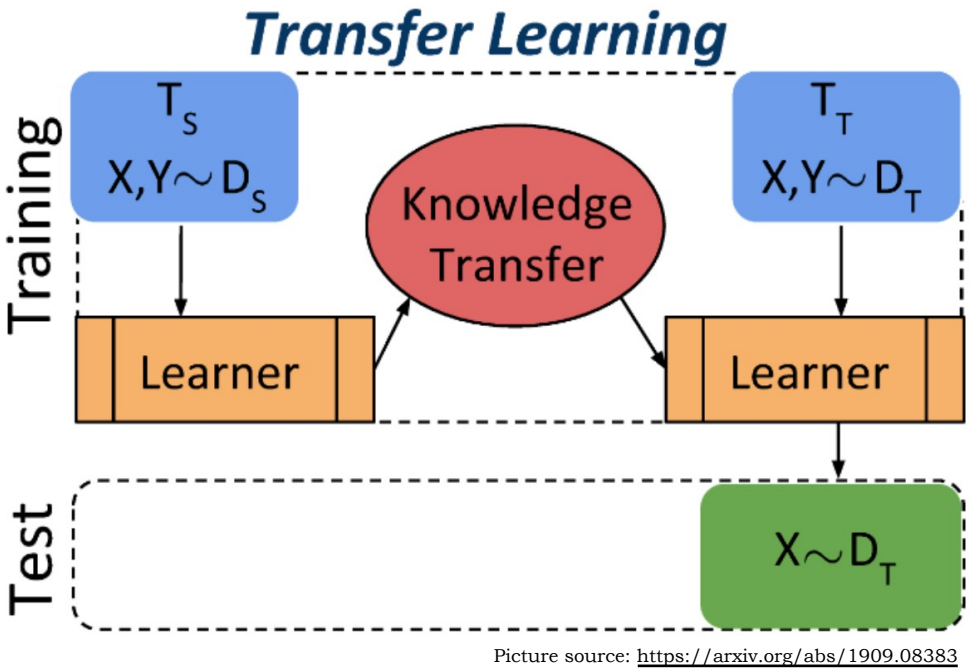
Picture source: https://arxiv.org/abs/1909.08383

Given a large number of tasks, quickly adapt to a new task

Offline training: set of training tasks available at the same time.

Can be used as re-training strategy

Transfer Learning



Picture source: https://arxiv.org/abs/1909.08383

Help learning the target task using model trained on the source task

No continuous adaptation after learning the target task

Performance on the source task(s) is not taken into account

Can be used as re-training strategy

Three Scenarios in Continual Learning

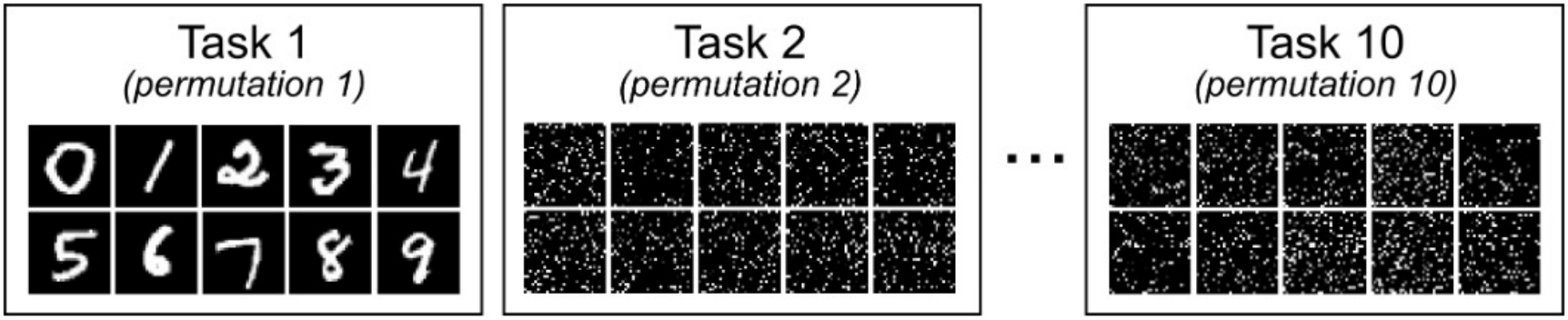


Figure 2: Schematic of permuted MNIST task protocol.

Table 3: Permuted MNIST according to each scenario.

Task-IL	Given permutation *X*, which digit?
Domain-IL	With permutation unknown, which digit?
Class-IL	Which digit *and* which permutation?

Source: https://arxiv.org/pdf/1904.07734.pdf

Progressive Neural Networks

- Neural Networks can saturate
- Add capacity to the neural network while learning new tasks / data distributions

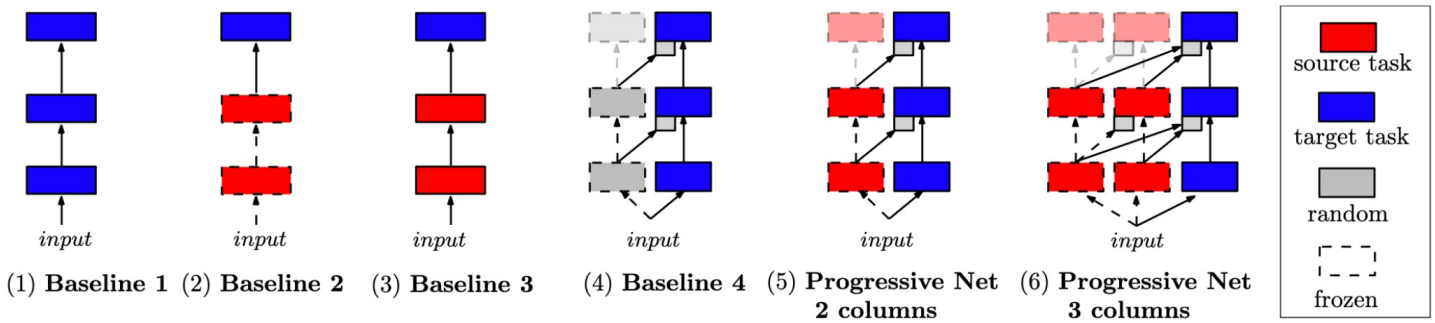
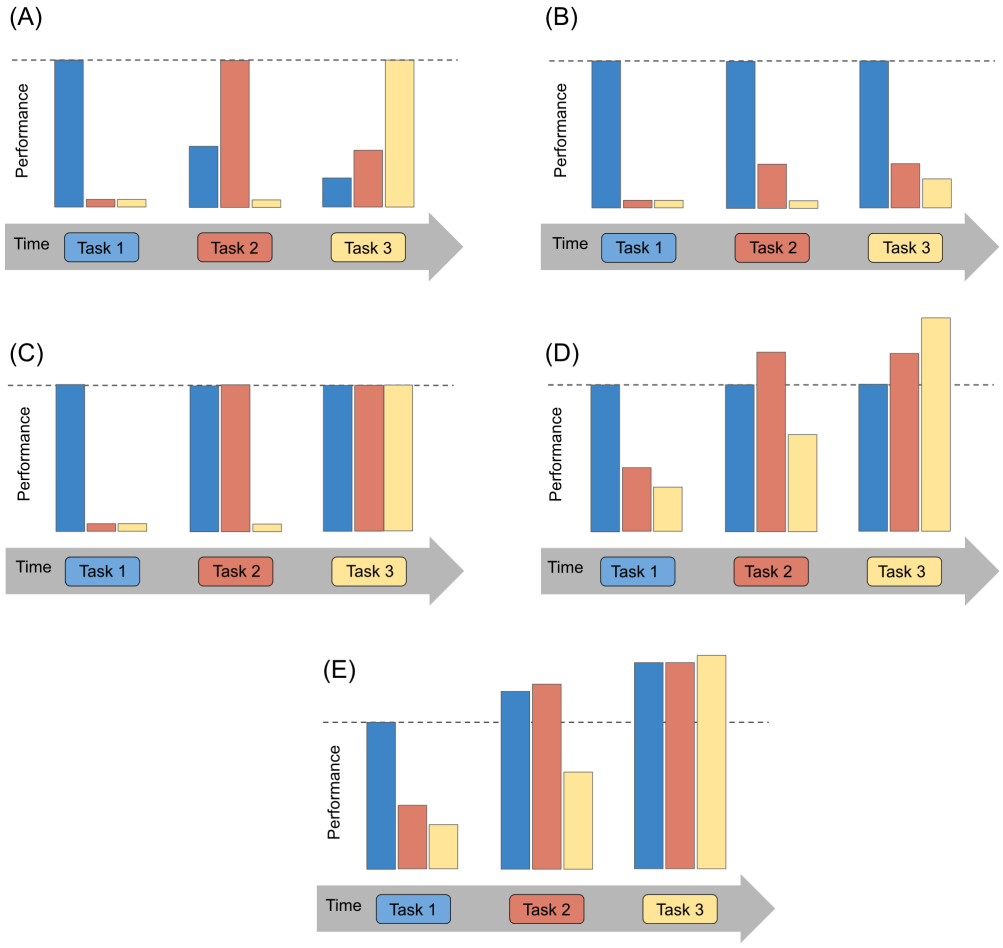


Figure 3: Illustration of different baselines and architectures. *Baseline 1* is a single column trained on the target task; *baseline 2* is a single column, pretrained on a source task and finetuned on the target task (output layer only); *baseline 3* is the same as baseline 2 but the whole model is finetuned; and *baseline 4* is a 2 column progressive architecture, with previous column(s) initialized randomly and frozen.

Source: Progressive Neural Networks

Possible Scenarios in Continual Learning

Tasks here can be considered as data distributions



Not desired

Good

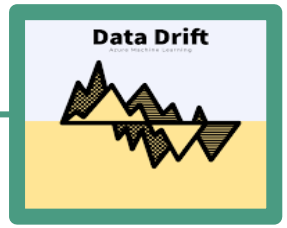
Ideal

Ideal scenario is to get both forward and backward transfer

Outline

01

Introduction and Motivation



02

Data Drifts

03

Continual Learning Pipelines and its components

04

CL in Neural Network; Challenges

05

Summary

Summary

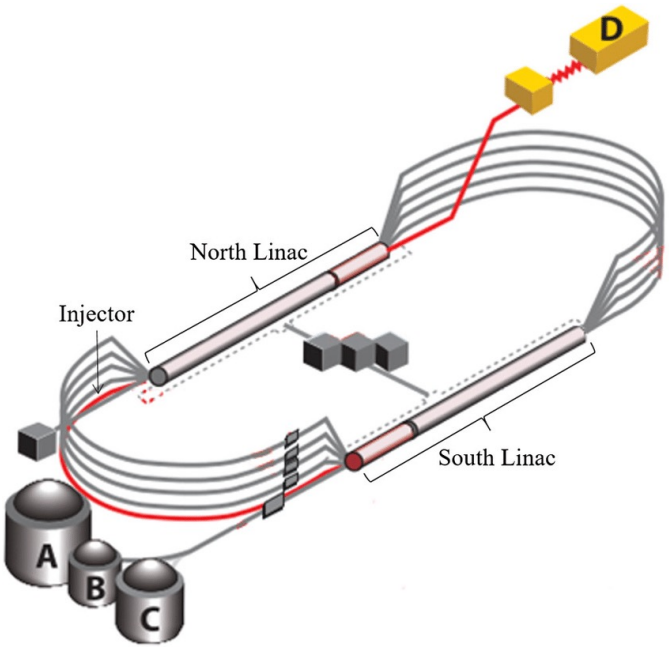
- Traditional ML models fail when data distribution or concept drifts
- Periodic re-training model might work in some cases
- Monitoring and re-tuning the model is better to keep it alive
- Monitoring model performance drift is usually better than monitoring data drifts
- Conditional models can be used to handle data drifts due to known measurable parameters
- Continual Learning lifecycle involves logging, curation, monitoring, re-training, evaluation, and deployment
- Continual Learning evaluation metrics takes into account performance on both past, and present data distributions
- Re-training neural networks introduce catastrophic forgetting problem; stability vs Plasticity dilemma
- Three scenarios in continual learning are Task incremental learning (IL), domain-IL, and class-IL
- Forward and backward transfer is desired in continual learning

Good practices

- Version control everything!
- Metric definitions and values, offline and online testing
- Performance thresholds
- Data curation logic
- Datasets used for training and evaluation
- ML-Flow might be useful

Continual Learning Application

- Particle Accelerators
  - Anomaly prediction
  - Optimization
  - Tuning and Controls
  - Field Emission detection
  - Data Quality Analysis
    - Drift due to
      - Configuration/tuning parameters
      - Environmental changes
      - Equipment aging
      - Replacement of equipment



References

[1] Grossberg. S. (1982) Studies of Mind and Brain: Neural Principles of Learning, Perception, Development, Cognition, and Motor Control.

[2] Carpenter, G. and Grossberg, S. (1987) ART 2: Self-organization of stable category recognition codes for analog input patterns.

[3] Matthias De Lange, Rahaf Aljundi, Marc Masana, Sarah Parisot, Xu Jia, Ales Leonardis, Gregory Slabaugh, Tinne Tuytelaars, A continual learning survey: Defying forgetting in classification tasks

[4] Raia Hadsell, Dushyant Rao, Andrei A. Rusu, Razvan Pascanu, Embracing Change: Continual Learning in Deep Neural Networks

[5] Andrei A. Rusu, Neil C. Rabinowitz, Guillaume Desjardins, Hubert Soyer, James Kirkpatrick, Koray Kavukcuoglu, Razvan Pascanu, Raia Hadsell, Progressive Neural Networks

[6] Yen-Chang Hsu, Yen-Cheng Liu, Anita Ramasamy, Zsolt Kira, Re-evaluating Continual Learning Scenarios: A Categorization and Case for Strong Baselines

[7] James Kirkpatricka, Razvan Pascanua, Neil Rabinowitz, Overcoming catastrophic forgetting in neural networks

[8] Kishansingh Rajput et. al. Robust Errant Beam Prognostics with Conditional Modeling for Particle Accelerators

[9] Felix Wiewel, Bin Yang, Localizing Catastrophic Forgetting in Neural Networks

[10] Liyuan Wang, Xingxing Zhang, Hang Su, Jun Zhu, A Comprehensive Survey of Continual Learning: Theory, Method and Application

[11] Gido M. van de Ven & Andreas S. Tolias, Three scenarios for continual learning

Hands on example on google colab at:
https://colab.research.google.com/drive/1H9nXa_dxqgvaOLlAlr0LLS8AmA0iUI0-?usp=sharing

Requires basic understanding of python, pytorch, and machine learning