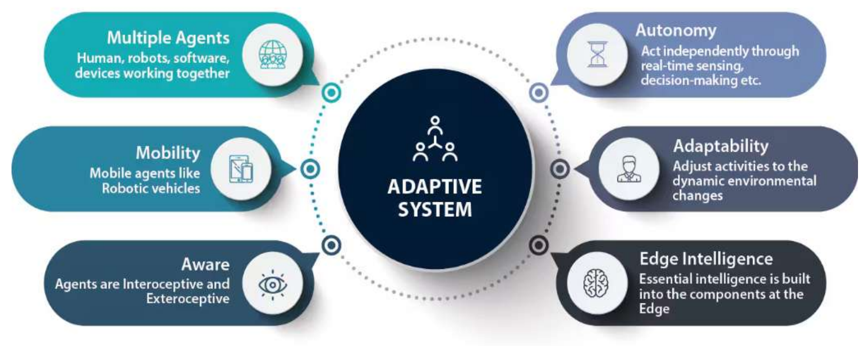
CN7050 – Intelligent Systems

Week 3: Agent Adaptability with Machine Learning

Dr Azhar Mahmood

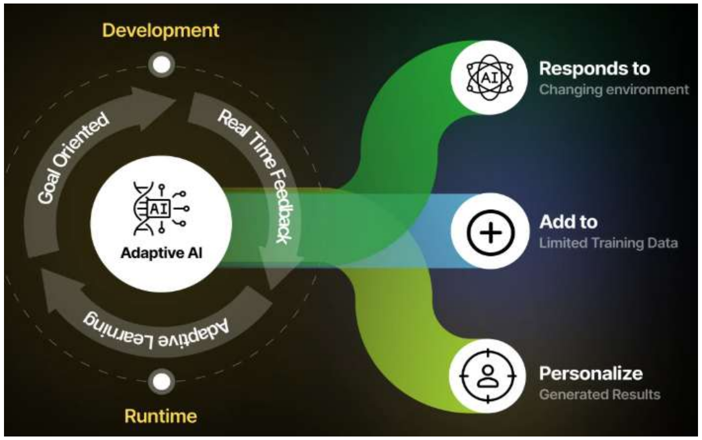
Office:EB.1.85

Email: a.mahmood3@uel.ac.uk



Adaptive AI Agents

Feature	Adaptive AI	Traditional AI
Maintenance	Requires ongoing monitoring and maintenance to ensure the system continues learning and adapting to new information.	Requires less maintenance, as the system does not change or adapt once it has been deployed.
Human Interaction	Required	Optional or not required
Learning	Online learning or continuous learning involves feeding data to the system and updating its model in real-time.	Batch learning, where the system is trained on a fixed dataset and then deployed, with no further updates to its model.
Performance	Improves over time	Fixed or degrades over time
Real-World Application	Ideal for dynamic and rapidly changing environments where the system must evolve and continuously provide value.	Suitable for environments with stable and well-defined conditions, where the system's performance does not need to change over time.
Adaptability	High, able to adapt to new information and changing conditions.	Low, limited to the performance capabilities determined by the fixed training dataset.
Scalability	High	Low
Implementation	Dynamic and flexible	Static and inflexible
Definition	AI systems that can adapt and improve their performance over time through continuous learning.	AI systems that are trained on a fixed dataset and do not adapt to new information or changing conditions.



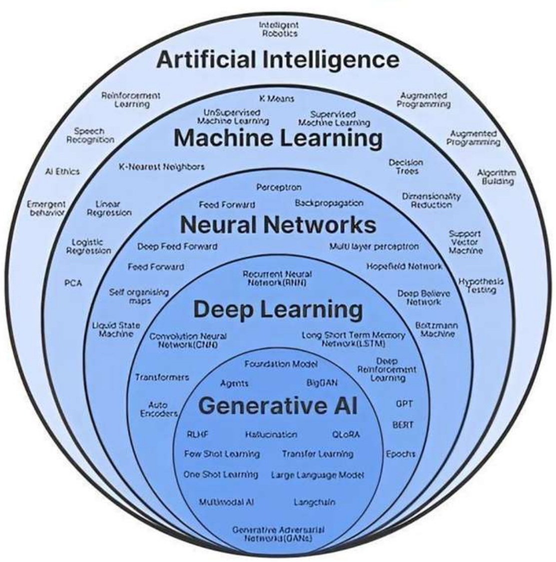
Why Agent Adaptability?

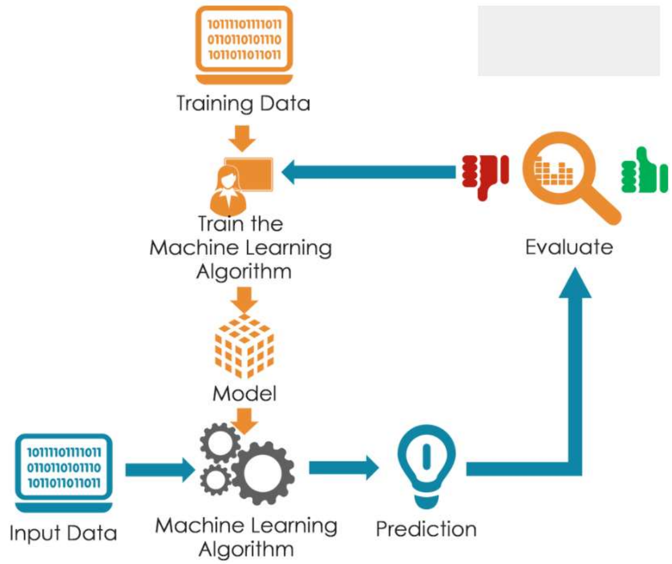
- Ability of an AI agent to **adjust its behavior** when environment or task changes
- **Key feature** for intelligent, real-world applications
  - A robot vacuum cleaner adapting to new furniture layout
  - Recommender systems adapting to changing user preferences

- Real-world environments are **dynamic**
- Agents must handle **unexpected changes**
- Ensures flexibility, robustness, and usefulness in daily life

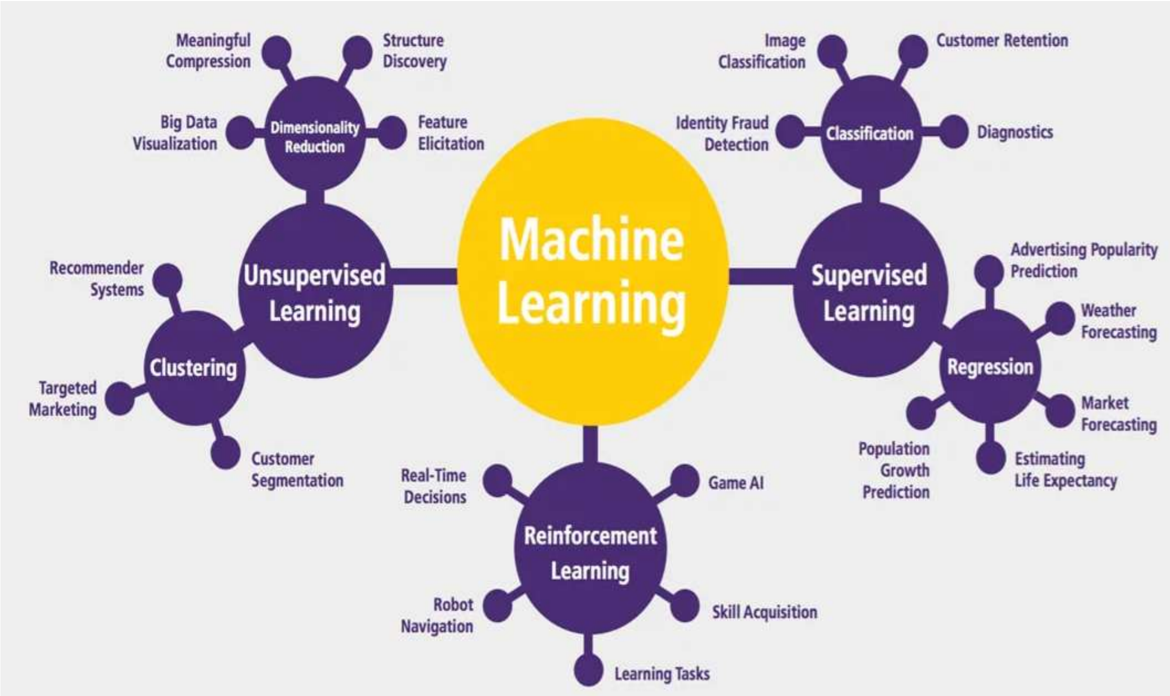
Why Machine Learning?

Machine learning (ML) is crucial for developing **agents that learn and act autonomously**, with techniques like supervised learning, unsupervised learning, and especially **reinforcement learning** being used to enable agents **to perceive, decide, and act.**





Applications of Machine Learning



https://swisscognitive.ch/2021/03/18/applications-of-machine-learning/ https://medium.com/@HunterKempf/2024-trends-in-ai-and-machine-learning-58f45a2007ca

Agent Learning from Data

Data is recorded from some real-world phenomenon.

What might we want to do with that data?

Agent Learning from Data? Questions

Q1. How can we **extract knowledge** from data to help humans make decisions?

Q2. How can we **automate decisions** from data?

Q3. How can we **adapt systems** dynamically to enable better user experiences?

Write code to explicitly do the above tasks

Write code to make the computer **learn how to do the tasks**

Machine Learning Framework

$$\boldsymbol{y} = \boldsymbol{f}(\boldsymbol{x})$$

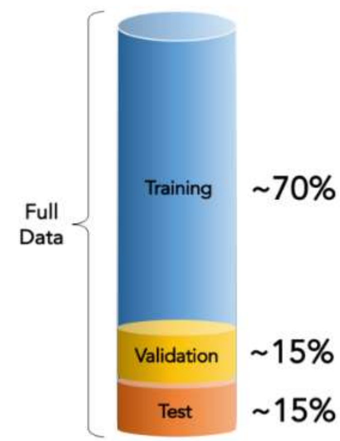
Output

Prediction function

Data feature

Input Data (x) or the feature is composed of:

- **Training set**
- **Validation set**
- **Test set**

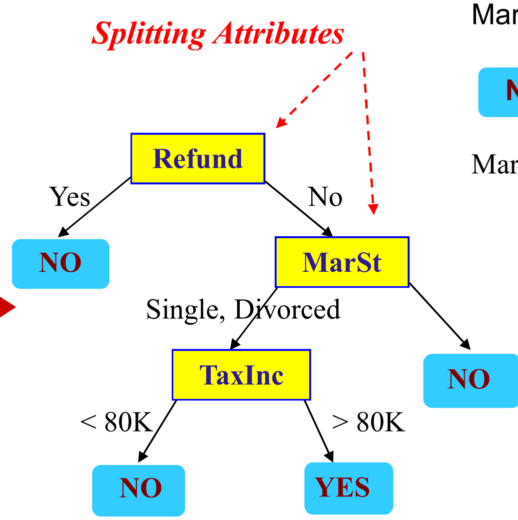


Example of a ID3 & CART

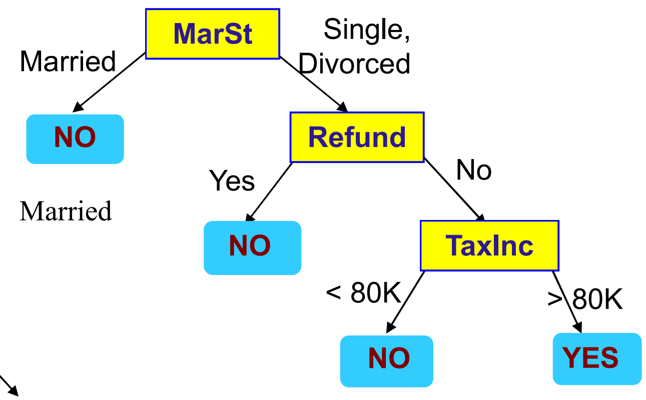
categorical *categorical* *continuous* *class*

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Model: ID3 & CART



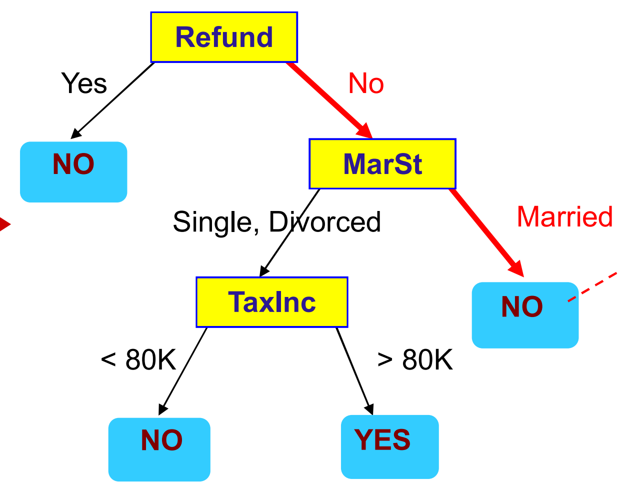
There could be more than one tree that fits the same data!

Apply Model to Test Data

categorical *categorical* *continuous* *class*

Tid	Refund	Marital Status	Taxable Income	Cheat
1	Yes	Single	125K	No
2	No	Married	100K	No
3	No	Single	70K	No
4	Yes	Married	120K	No
5	No	Divorced	95K	Yes
6	No	Married	60K	No
7	Yes	Divorced	220K	No
8	No	Single	85K	Yes
9	No	Married	75K	No
10	No	Single	90K	Yes

Training Data



Test Data

Refund	Marital Status	Taxable Income	Cheat
No	Married	80K	?

Assign Cheat to “No”

Bayesian Classification

- **Supervised, Statistical** Learning Method.
- Can solve problems involving both **categorical and continuous valued attributes**.

Total probability Theorem: $P(B) = \sum_{i=1}^{M} P(B|A_i)P(A_i)$

Bayes' Theorem: $$P(H|\mathbf{X}) = \frac{P(\mathbf{X}|H)P(H)}{P(\mathbf{X})} = P(\mathbf{X}|H) \times P(H) / P(\mathbf{X})$$

- Let X be a data sample ("*evidence*"): class label is unknown
- Let H be a *hypothesis* that X belongs to class C
- Classification is to determine **P(H|X),** (*posteriori probability):* the probability that the hypothesis holds given the observed data sample X
- P(H) (*prior probability*): the initial probability
  - E.g., X will buy sports car, regardless of age, income, …
- P(X): probability that sample data is observed
- P(X|H) (likelihood): the probability of observing the sample X, given that the hypothesis holds
  - E.g., Given that X will buy sports car, the prob. that X is 31..40, medium income

Bayesian Classification: Why?

- **Performance**: A *simple Bayesian classifier*, *naïve Bayesian classifier*, has comparable performance with ID3, CART and *neural network classifiers.*
- **Incremental**: Each training example can **incrementally increase/decrease the probability** that a hypothesis is correct AND **prior knowledge** can be combined with observed data.
- **Standard**: Even when Bayesian methods are **computationally intractable**, they can provide a **standard of optimal decision** making against which other methods can be measured.
- Informally, this can be viewed as

Posteriori = Likelihood x Prior/Evidence

Naïve Bayes Classifier: An Example

X = (age <= 30 , income = medium, student = yes, credit_rating = fair)

- $P(C_i)$: P(buys_computer = "yes") = 9/14 = **0.643**
  P(buys_computer = "no") = 5/14= **0.357**
- Compute $P(X|C_i)$ for each class
  P(age = "<=30" | buys_computer = "yes") = 2/9 = 0.222
  P(age = "<= 30" | buys_computer = "no") = 3/5 = 0.6
  P(income = "medium" | buys_computer = "yes") = 4/9 = 0.444
  P(income = "medium" | buys_computer = "no") = 2/5 = 0.4
  P(student = "yes" | buys_computer = "yes) = 6/9 = 0.667
  P(student = "yes" | buys_computer = "no") = 1/5 = 0.2
  P(credit_rating = "fair" | buys_computer = "yes") = 6/9 = 0.667
  P(credit_rating = "fair" | buys_computer = "no") = 2/5 = 0.4
- **X = (age <= 30 , income = medium, student = yes, credit_rating = fair)**

$P(X|C_i)$: P(X|buys_computer = "yes") = 0.222 x 0.444 x 0.667 x 0.667 = **0.044**
P(X|buys_computer = "no") = 0.6 x 0.4 x 0.2 x 0.4 = **0.019**

$P(X|C_i)*P(C_i)$: P(X|buys_computer = "yes")* P(buys_computer = "yes")=**0.044 *0.643 = 0.028**
P(X|buys_computer = "no") * P(buys_computer = "no") = **0.019 *0.357=0.007**

P(X|buys_computer = "yes") = 0.028/(0.028+0.007) = 0.80 = 80 %
P(X|buys_computer = "no") = 0.007/(0.028+0.007) = 0 .20 = 20 %

Therefore, X belongs to class ("buys_computer = yes")

Age	Income	Stud	Cred Rating	Buy Com
<=30	high	no	fair	no
<=30	high	no	excellent	no
31…40	high	no	fair	yes
>40	medium	no	fair	yes
>40	low	yes	fair	yes
>40	low	yes	excellent	no
31…40	low	yes	excellent	yes
<=30	medium	no	fair	no
<=30	low	yes	fair	yes
>40	medium	yes	fair	yes
<=30	medium	yes	excellent	yes
31…40	medium	no	excellent	yes
31…40	high	yes	fair	yes
>40	medium	no	excellent	no

Naïve Bayes - Text Classification Example

Text (Document)	Category
a great Game	Sports
The election was over	Non Sports
very clean match	Sports
I clean but forgettable game	Sports
It was a close election	Non Sports

"A VERY CLOSE GAME"

Find Category
Sports ?
Non Sports ?

Naïve Bayes- Text Classification

- **P(Category="Sports")=3/5**
- **P(Category="Non Sports")=2/5**

- Total Words in Sports Category : **11**
- Total Words in non Sports Category : **9**
- Total Unique Words: **14**

- **P(Sports/"A very close game") =?**
- **P(Non Sports/"A very close game") =?**

X= **" a very close game"**

P(x)= P(a) x P(very) x P(close) x P(game)

P(x/Sports)= P(a/Sports) x P(very/Sports) x P(close/Sports) x P(game/Sports)

P(x/Non Sports)= P(a/Non Sports) x P(very/Non Sports) x P(close/Non Sports) x P(game/Non Sports)

Text (Document)	Category
A great game	**Sports**
The election was over	**Non Sports**
Very clean match	**Sports**
I clean but forgettable Game	**Sports**
It was a close election	**Non Sports**

Laplace Smoothing

In statistics, **additive smoothing**, also called **Laplace smoothing** (not to be confused with Laplacian smoothing as used in image processing), or **Lidstone smoothing**, is a technique used to smooth categorical data.

$$\hat{\theta}_i = \frac{x_i + \alpha}{N + \alpha d} \qquad (i = 1, \ldots, d).$$

$\alpha > 0$ is a smoothing parameter

d: no of unique words or distinct words

N: Total number of words

P(word) = word count + 1/ Total no of words + no of unique words //here *α=1*

Naïve Bayes- Text Classification

Word	P(word/Sports)	P(word/Non-Sports)
a	1+1/11+14 =0.080	1+1/9+14 = 0.086
very	1+1/11+14 = 0.080	0+1/9+14 = 0.043
Close	0+1/11+14 = 0.040	1+1/9+14 = 0.086
game	2+1/11+14 = 0.120	0+1/9+14 = 0.043

P(x/Sports)= 0.080 x 0.080 x 0.040 x 0.120 =0.0000307

P(x/Non Sports)= 0.086 x 0.043 x 0.086 x 0.043 =0.0000136

P(x\Category= Sports)= 0.0000307 x 0.6 = 0.00001842

P(x\Category= Non Sports)= 0.0000136 x 0.4= 0.00000544

Text (Document)	Category
A great Game	Sports
The Election was over	Non Sports
Very clean Match	Sports
I clean but forgettable Game	Sports
It was a close Election	Non Sports

Total Words in Sports Category : 11

Total Words in non Sports Category : 9

Total Unique Words: 14

“ A Very Close Game” = SPORTS

ML based Agents

Algorithm Type	Examples	Common Agent / Problem Types	
Linear Models	Linear Regression, Logistic Regression	- **Predictive agents** (sales forecasting, pricing models) - **Risk assessment** (credit scoring)	**Supervised Learning**
Tree-Based Models	Tree-Based Models	**- Recommendation systems** **- Fraud detection agents** **- Customer churn prediction**	
k-Nearest Neighbors	KNN	- **Recommendation agents** (content or product) - **Anomaly detection** (simple cases)	
Neural Networks	Feedforward, CNN, RNN, LSTM	- **Computer vision agents** (object detection, image tagging) - **NLP agents** (Chabot's, translators) - **Time-series forecasting** (stock, weather)	
Clustering	K-Means, DBSCAN, Hierarchical Clustering	**- Customer segmentation agents** **- Market analysis** **- Anomaly detection agents**	**Un-Supervised Learning**
Association Rule Learning	Apriori, FP-Growth	**- Recommendation engines** **- Market basket analysis agents**	
Graph-Based Learning	Graph Neural Networks (GNNs)	**- Social network analysis** **- Recommendation or link prediction agents**	**Semi- Supervised Learning**

Reinforcement & DL based Agents

Algorithm Type	Examples	Common Agent / Problem Types	
Value-Based	Q-Learning, Deep Q-Network (DQN)	- **Game-playing agents** (AlphaGo) - **Navigation agents** (robots, drones)	**Reinforcement Learning**
Policy-Based	REINFORCE, PPO, A3C	- Trading bots - **Dynamic pricing agents**	
Model-Based	Dyna-Q, World Models	- Simulation-based control - **Autonomous driving and robotics**	
Convolutional Neural Networks (CNNs)		- **Vision agents** (object detection, medical imaging)	**Deep Learning Variants**
Recurrent Neural Networks (RNNs, LSTMs, GRUs)		- **Sequential prediction** (speech recognition, text generation)	
Transformers	BERT, GPT, Vision Transformers	- **Conversational AI agents** - Language understanding - Image captioning	
Generative Models	GANs, VAEs, Diffusion Models	- Image generation - **Content creation agents** - Simulation and design optimization	

Summary:Mapping Algorithm to Agent Type

Agent Type	Likely Algorithm(s)	Problem Domain
Chatbot / Conversational Agent	Transformers (e.g., GPT, BERT)	NLP, text generation
Fraud Detection Agent	Random Forest, Isolation Forest, Autoencoder	Classification / Anomaly detection
Stock Trading Agent	Reinforcement Learning (PPO, DQN), LSTM	Time-series prediction & control
Recommendation Agent	kNN, Matrix Factorization, Neural CF	Collaborative filtering
Vision Agent (e.g., for drones, security)	CNN, YOLO, ResNet	Image detection & recognition
Predictive Maintenance Agent	LSTM, Random Forest, Autoencoder	Predictive analytics & anomaly detection
Game AI Agent	Deep Q-Network, PPO	Sequential decision-making

Limitations of ML: In context of Agents

- **Data Dependency:** Agents that must handle diverse user inputs or real-world environments, collecting **enough representative data** is difficult and expensive.
- **Lack of Generalization:** Agents may **struggle** to adapt to **novel contexts** or **unexpected user behavior** without retraining.
- **Limited Understanding & Reasoning:** Agents can mimic understanding (e.g., generating text or making predictions) but lack **true comprehension**, **logical reasoning,** or common sense.
- **Explainability and Transparency:** Many ML models are **black boxes** this makes it **hard to debug, justify**, **or trust** an agent's actions especially in sensitive applications (e.g., healthcare or finance).
- **Difficulty in Continuous Learning:** Most of ML agents are **static after training,** building **adaptive agents** that improve safely and reliably over time is still a research challenge.
- **Safety and Control:** Autonomous agents can behave **unpredictably** in dynamic environments. Ensuring **safe exploration, alignment with human values, and controllability** is an ongoing challenge.
- **High Computational and Resource Costs:** ML-based agents, especially those using deep learning or reinforcement
- **….**

A Real case of ML Limitation

sensors

Submit to this Journal

Review for this Journal

Propose a Special Issue

Article Menu

Subscribe SciFeed

Recommended Articles

Related Info Links

More by Authors Links

Article Views

Citations

Order Article Reprints

Open Access Article

A Context-Aware Accurate Wellness Determination (CAAWD) Model for Elderly People Using Lazy Associative Classification

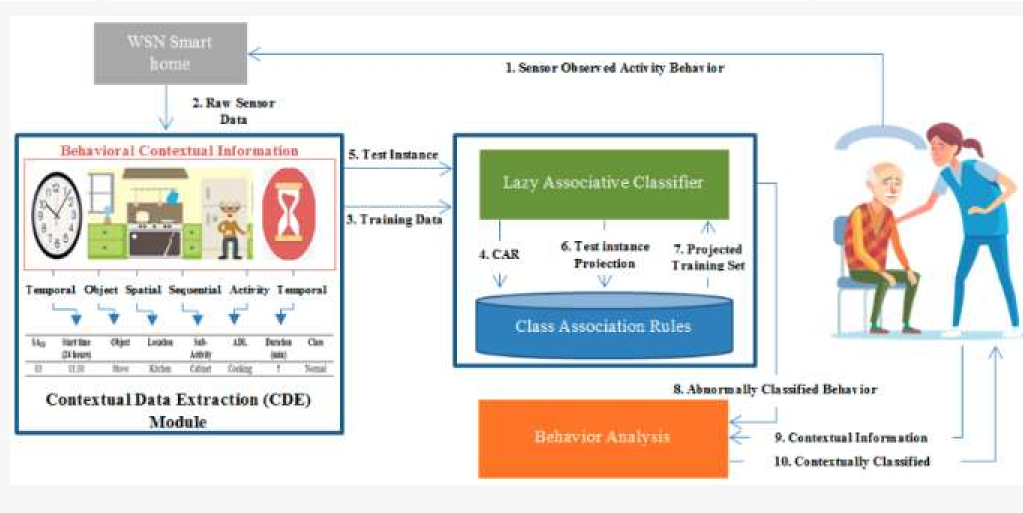
by Farhan Sabir Ujager and Azhar Mahmood *

Faculty of Computing and Engineering Sciences, Shaheed Zulfikar Ali Bhutto Institute of Science and Technology (SZABIST), Islamabad 44000, Pakistan

* Author to whom correspondence should be addressed.

Table 7. Behavior anomaly examples.

Behavior Routine Anomalies	Possible Reasons
Sleeping longer than usual	Medication, fatigue, hypertension, dizziness
Higher frequency of toileting	Stomach upset, higher diabetes level
Zero duration of leaving	Feeling unwell, skipping daily walk, fatigue



https://www.mdpi.com/1424-8220/19/7/1613#:~:text=The%20frequent%20behavioral%20patterns%20are,generate%20behavior%2Dfocused%20classification%20rules.

Approaches to Agent Adaptability

1. **Meta-Learning (Learning to Learn):**
   - Learns how to learn new tasks quickly

     A robot trained on many tasks adapts to a new target in just a few tries
2. **Continual Learning:**
   - Learns new tasks over time without forgetting old ones

     A Chabot learns about sports but still remembers weather questions
3. **Domain Adaptation:**
   - Works across different environments (train in one, adapt to another)

     A self-driving car trained in simulation adapts to real-world weather
4. **Transfer Learning:**
   - Reuses past knowledge for new related tasks

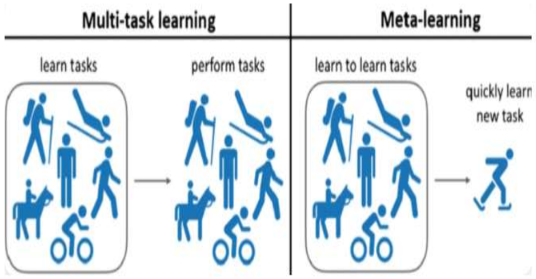
     A vision model trained on images is adapted to medical scans

Meta-Learning

Meta-learning, also known as "**learning how to learn**," is a cutting-edge approach in machine learning that focuses on algorithms that learn from their experiences and adapt to new data more effectively.

Instead of learning one fixed task, it learns *the process* of learning itself, so it can **quickly adapt to *new tasks* with very little data**.

- In **robotics**, Meta-learning helps robots **learn motor skills and adapt to new tasks quickly**, with minimal human intervention.
- In **natural language processing**, Meta-learning **enhances** text classification, sentiment analysis, and language modeling by connecting **prior knowledge with new tasks**.
- In **computer vision**, Meta-learning enables object recognition, image synthesis, and style transfer by **learning from diverse datasets and tasks.**

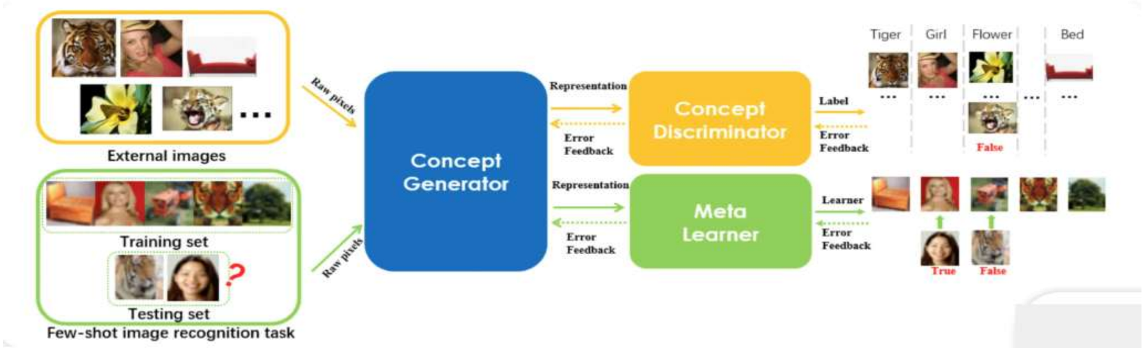


Role of Meta-Learning

- In meta-learning, a **concept generator** is a module, often a deep neural network, that **extracts high-level representations** or "concepts" from individual data instances.

- A **concept discriminator** is a component of a system that works alongside a concept generator **to perform meta-learning** in a "concept space“

- **Meta learner** is trained on meta datasets where it learns the **meta-knowledge and meta-strategy**, and during meta-testing, the knowledge gained is evaluated on new and unseen tasks.



Personalized Learning Assistant

- **Personalized Learning Assistant**

Imagine an AI tutor that helps different students.

Student A learns best with visuals.

Student B learns best with text.

Student C prefers examples.

A traditional model would need lots of data per student to adapt.

But a **meta-learning tutor** would **learn from past students** *how to adjust its teaching style quickly* after just a few interactions with a new student.

Continual Learning

Continual learning aims to allow the model to effectively **learn new concepts** while ensuring it **does not forget already acquired information**.

- **Class incremental continual learning**

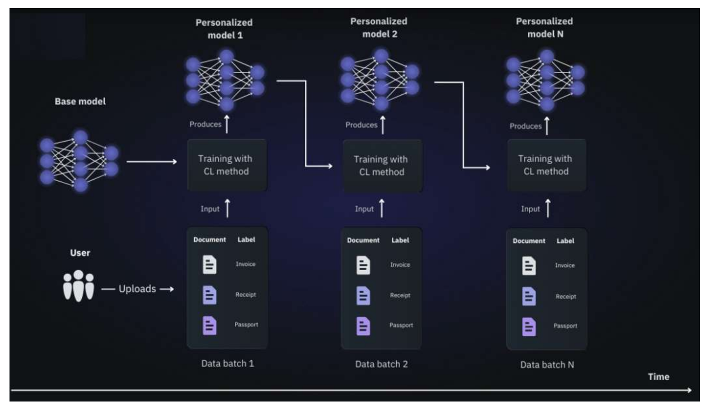
Class Incremental (CI) continual learning is a scenario in which the number of classes in a classification task is not fixed but can increase over time.

- **Domain incremental continual learning**

Model to extract data from invoices, and users upload invoices with a different layout, then we can say that the input data distribution has changed.

- **Task incremental continual learning**

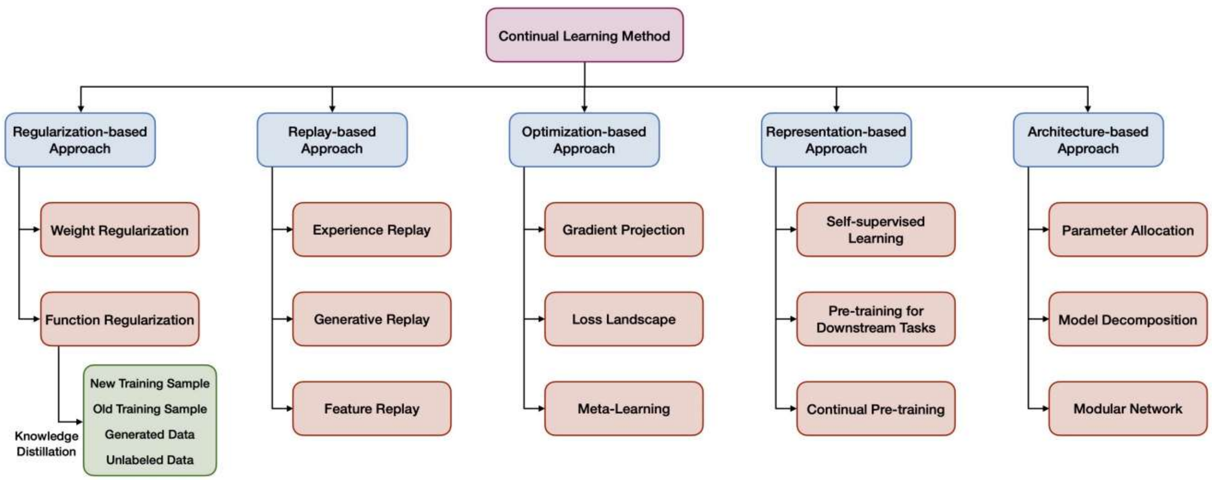
Instead of having separate models for each task, one model is trained to solve them all. The difficulty in the continual learning setting is that data for each task arrives at a different time, and the number of tasks might not be known beforehand, requiring the model's architecture to expand over time.



Model personalization via continual learning in a document classification process. The user uploads documents to the system, and the model is retrained after each batch, creating a new personalized model.

https://neptune.ai/blog/continual-learning-methods-and-application

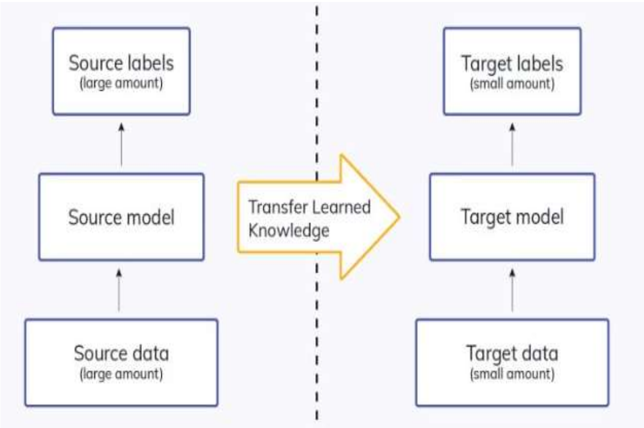
Continual Learning



Regularization: During incremental training, it makes the **model learn new data without forgetting the past**, they use techniques like knowledge distillation.
Replay-based: **Natural process** of learning such as child-centered educational method.
Representation-based : Focuses on building models that can **learn a sequence of tasks without forgetting previous ones** by creating flexible and robust feature representations.
Optimization-based: Aim to **control how model parameters are updated** during learning new tasks and preserve prior knowledge.
Architecture-based: Designs the neural network's structure **(layers, neurons, or subnetworks)** in a way that helps it remember past tasks while learning new ones.

Transfer Learning

The **reuse of a previously learned model** on a new problem is known as transfer learning. It uses knowledge **learn from one model to perform a new task**.

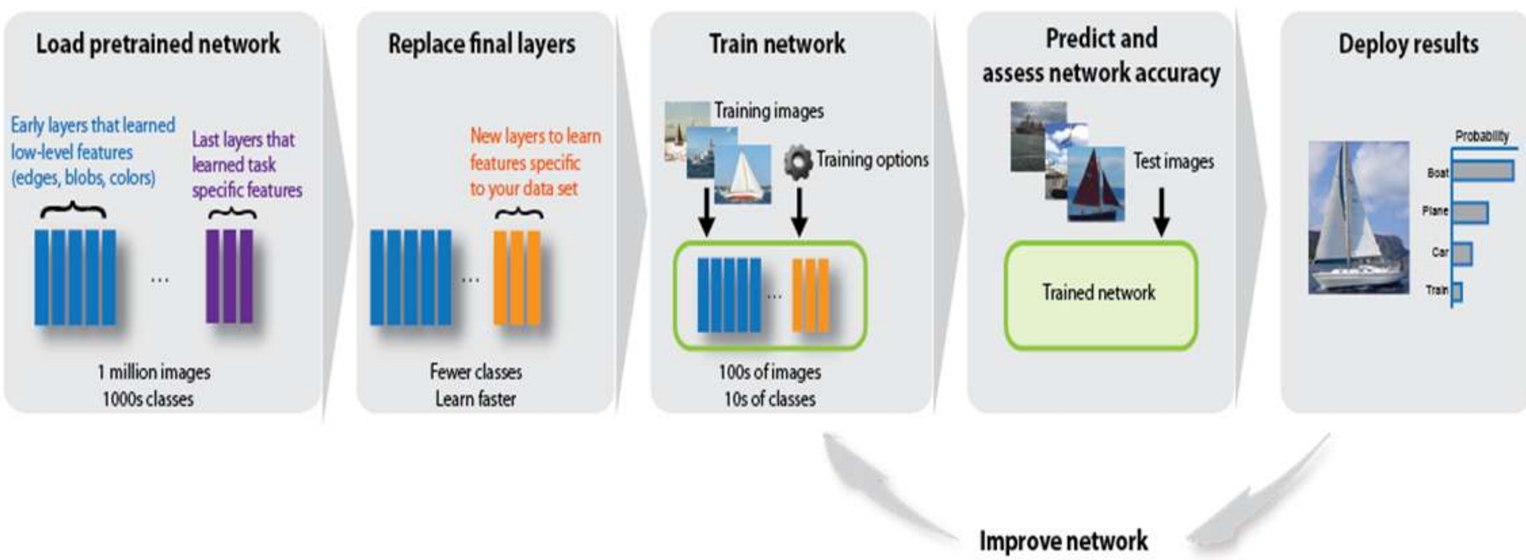


Motivation

1. Lots of data, time, resources needed to train and tune a deep learning from scratch
2. An ImageNet deep neural net can take weeks to train and fine-tune from scratch.
3. Unless you have 256 GPUs, possible to achieve in 1 hour
4. Transfer Learning, cheaper, faster way of adapting a neural network by exploiting their generalization properties

Transfer Learning Steps

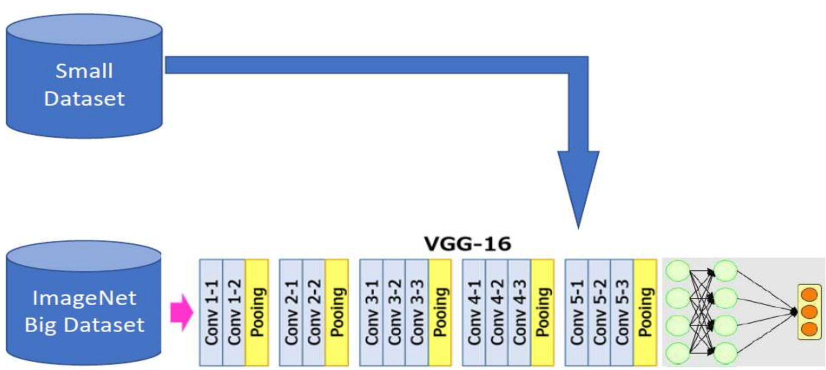
Reuse Pre-trained Network



Transfer Learning - VGG-16 Model

Pretrained Model on ImageNet: The **VGG-16** model is first trained on a large dataset (ImageNet), which contains millions of images across thousands of categories.
- 1.2 M images in 1K categories

Fine-Tuning: The small dataset fine-tunes the higher-level layers while keeping lower-level features learned from ImageNet.



Challenges & Future in Adaptability

1. **Catastrophic Forgetting:** Losing old knowledge when learning new tasks
2. **Safety:** Making sure adaptation doesn't cause harmful actions
3. **Data Efficiency:** Adapting with limited data
4. **Generalization:** Performing well in unseen situations

Future:

- Safer adaptation in real-world environments
- More efficient algorithms needing less data
- Combining multiple adaptability methods
- Wider applications in healthcare, robotics, and education

Summary

- Agent adaptability, key feature for intelligent systems
- Helps agents handle changes in environment or tasks
- ML types and capabilities
- Examples of ML agents and limitations
- Development of ML based agents
- Limitations of ML for agent adaptability
- Uses techniques like meta-learning, continual learning, and transfer learning
- **Still an active research area with exciting opportunities…**