Types of Learning

Learning (Training)

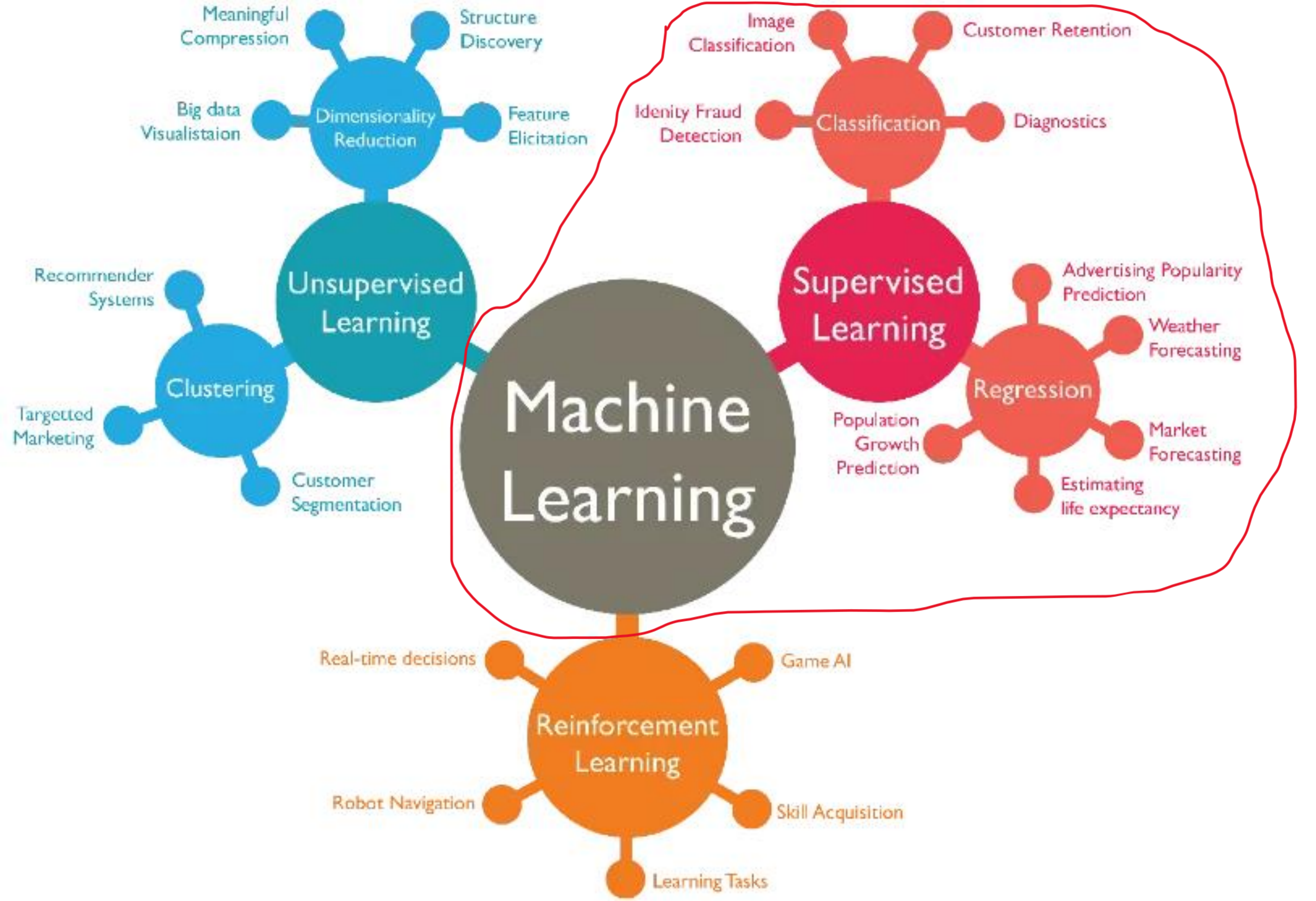
- The ability of a system to **change its parameters** – internal states to evolve, to perform its tasks better.
- Process in which a structure is progressively modified (optimized) to provide better performance in its environment.

Types of learning (in general)

- **Supervised learning** (**labeled data**)
- Unsupervised learning (unlabeled data)
- Semi-supervised learning (small amount of labeled data with a large amount of unlabeled data)
- Reinforcement learning
- Transfer learning

Learning support

- sets of training data
- training algorithms



https://deeplylearning.fr/cours-theoriques-deep-learning/les-differents-types-dapprentissage/

Difficulties in the learning process

- Large number of variables
- Complex evaluation functions, nonlinear, variable in time and in the space of the input parameters

Law of sufficiency

[Eberhart, R., Shi, Y., Computional Intelligence, Concepts to implementations, Elsevier Inc., 2007, ISBN: 978-1-55860-759-0, 467 pp.]

- If a solution is:
  - good enough (meets specifications)
  - fast enough
  - cheap enough

then it is sufficient!

Learning (Training)

- Supervised learning
- Unsupervised learning

Supervised learning is done using a ground truth; there is prior knowledge of what the **output values** for the corresponding inputs **should be**.

The goal of supervised learning is **to learn an application** that, given a sample of data and desired outputs, best approximates the relationship between input and output observable in the data.

Unsupervised learning, on the other hand, does not have labeled outputs, so its goal is to infer the natural structure present within a set of data points.

Supervised learning

- A type of machine learning where **a model is trained on a labeled dataset**.

- In this process, the model **learns to make predictions or decisions** based on input data (features) that is **paired with** the correct output (labels or targets).

- The goal is for the model to generalize from the training data so that it can **accurately predict outcomes for new, unseen data**.

Supervised learning

Key components of supervised learning

1. **Training Data**: **A collection of data** where each example consists of input features and the corresponding correct output (label). The data is labeled, meaning both inputs and outputs are known.
   - **Features (Input Data)**: Variables used to make predictions,
     - characteristics of an object like size, shape, color, age, etc.
   - **Labels (Output Data)**: True values or categories one want to predict
     - the category of an object like "cat" or "dog.“
2. **Model**: A **machine learning algorithm** (decision trees, neural networks, support vector machines) that learns the relationship between the features and the labels.
   - The model is trained on the labeled data and then used to predict the labels for new, unseen inputs.

Supervised learning

Key components of supervised learning

3. **Training Process**: The model adjusts its internal parameters (weights in a neural network) based on the training data by minimizing the difference between its predicted output and the actual labels.
   - optimization techniques like gradient descent; training epochs.

4. **Objective**: To learn a mapping from inputs to outputs
   - when given new inputs, the model can accurately predict the correct outputs.
     - In classification tasks, the output is a category (like “spam” or “not spam” for emails).
     - In regression tasks, the output is a continuous value (like predicting house prices).

5. **Evaluation**: After training, the model is evaluated on a **separate test dataset** that the model hasn’t seen during training.
   - Assess how well it generalizes to new data.

Supervised learning

Types of supervised learning:

- **Classification**: The output variable is categorical. The task is to predict the **class label** of the input.
  - Example: Classifying emails as "spam" or "not spam.“

- **Regression**: The output variable is continuous. The task is to predict a **real number** based on input features.
  - Example: Predicting the price of a house based on its size, location, and other features.

Supervised learning

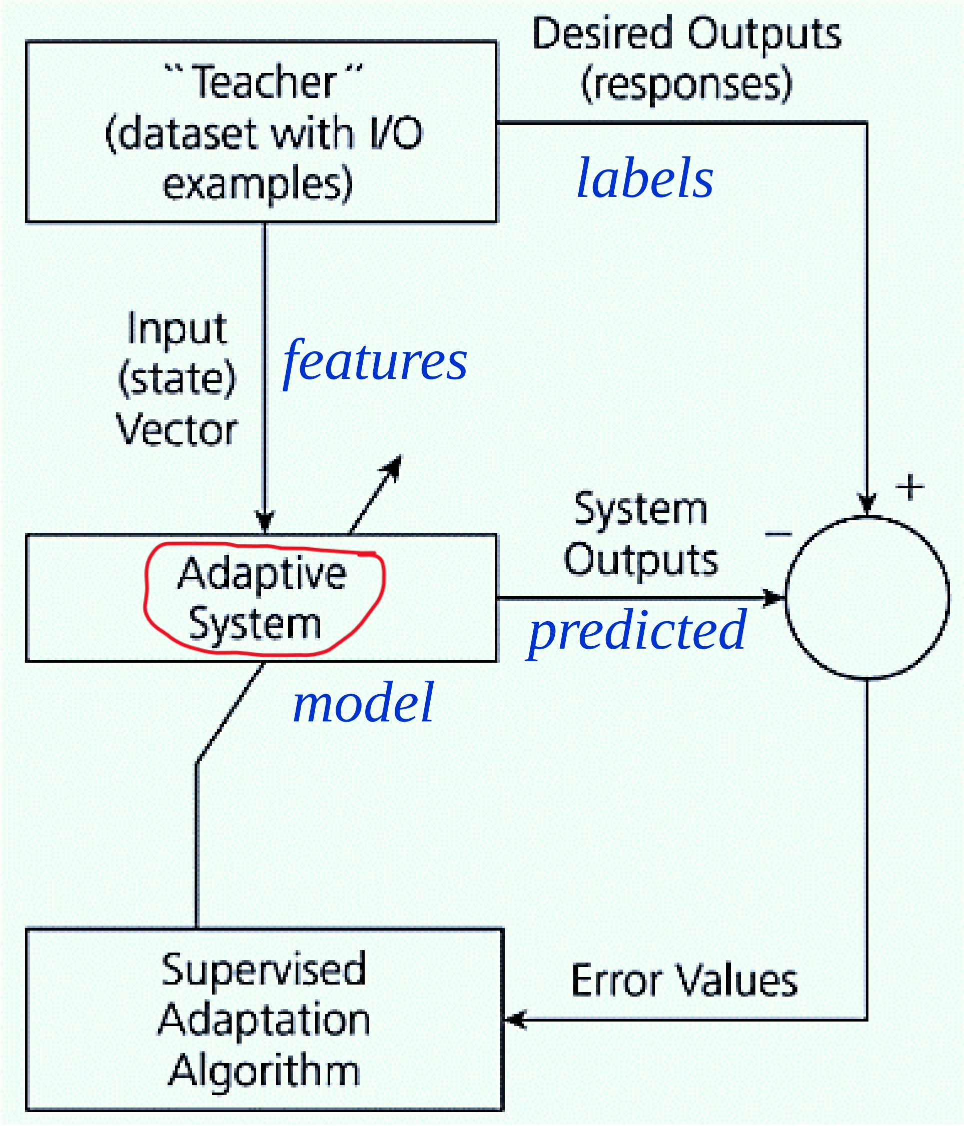
Example:

If we are building a model to classify whether an email is spam or not:

- **Input Features**: Email content, subject line, sender address, etc.
- **Labels**: "Spam" or "Not Spam."

- The model is trained on many (a lot of) labeled examples, then it learns to classify future emails as spam or not spam.

Supervised learning



- "Teacher": I/O datasets
- Learning is usually carried out one step (epoch) at a time;
- System performance metric is often inversely proportional to the sum of errors;
- Used for function approximation (fitting), pattern recognition, prediction.
- Example: backpropagation algorithm – used to train neural networks

[Eberhart, R., Shi, Y., Computational Intelligence, Concepts to implementations, Elsevier Inc., 2007, ISBN: 978-1-55860-759-0, 467 pp.]

Supervised Learning

Input(x)	Output (y)	Application	
Home features	Price	Real Estate	Standard NN
Ad, user info	Click on ad? (0/1)	Online Advertising	
Image	Object (1,…,1000)	Photo tagging	CNN
Audio	Text transcript	Speech recognition	RNN
English	Chinese	Machine translation	
Image, Radar info	Position of other cars	Autonomous driving	Custom/ Hybrid

Neural Networks and Deep Learning, https://www.coursera.org/learn/neural-networks-deep-learning/lecture/2c38r/supervised-learning-with-neural-networks

Unsupervised learning

- A type of machine learning where the model is **trained on unlabeled data**.

- In contrast to supervised learning, the dataset contains:
  - input data (features)
  - **no associated output labels**.

- The model tries to learn patterns, structures, or relationships in the data without being given explicit guidance on what the correct outputs should be.

Unsupervised learning

Key components of unsupervised learning:

1. **Input data**: The dataset contains **only input features**, and no labels or target outcomes are provided. The model must find hidden patterns or groupings based solely on the input data.
2. **Learning process**: The model explores the data to identify **inherent structures or relationships**. Since there are no labels to guide the learning, it focuses on finding underlying patterns, groups (clusters), or the distribution of the data.
3. **Objective**: The main objective of unsupervised learning is to **discover hidden structures** in the data. This could mean grouping similar items together (clustering) or finding patterns and relationships between data points (association).

Unsupervised learning

Types of unsupervised learning:

1. **Clustering**:
   - **Group similar data points together** based on their characteristics.
     - Segmenting customers into groups based on their purchasing behavior. You don't know the categories in advance; the algorithm figures out how to cluster similar customers.

2. **Dimensionality reduction**:
   - This technique **reduces the number of features** (dimensions) in the dataset while retaining as much information as possible.
     - Simplify data, especially when there are many input variables, making it easier to visualize or analyze.
       - Principal Component Analysis (PCA) transforms a high-dimensional dataset into a lower-dimensional space while preserving key relationships.

Unsupervised learning

Types of unsupervised learning:

3. Anomaly detection:

- Identifies **unusual data points** that do not conform to the general pattern in the dataset.
  - Detecting fraud in credit card transactions by identifying outliers or anomalies in the transaction history.

4. Association:

- Finds relationships between variables in a large dataset, discovering **which items frequently co-occur**.
  - In a supermarket, association rules may reveal that customers who buy bread are likely to buy butter as well. This is the basis for market basket analysis.

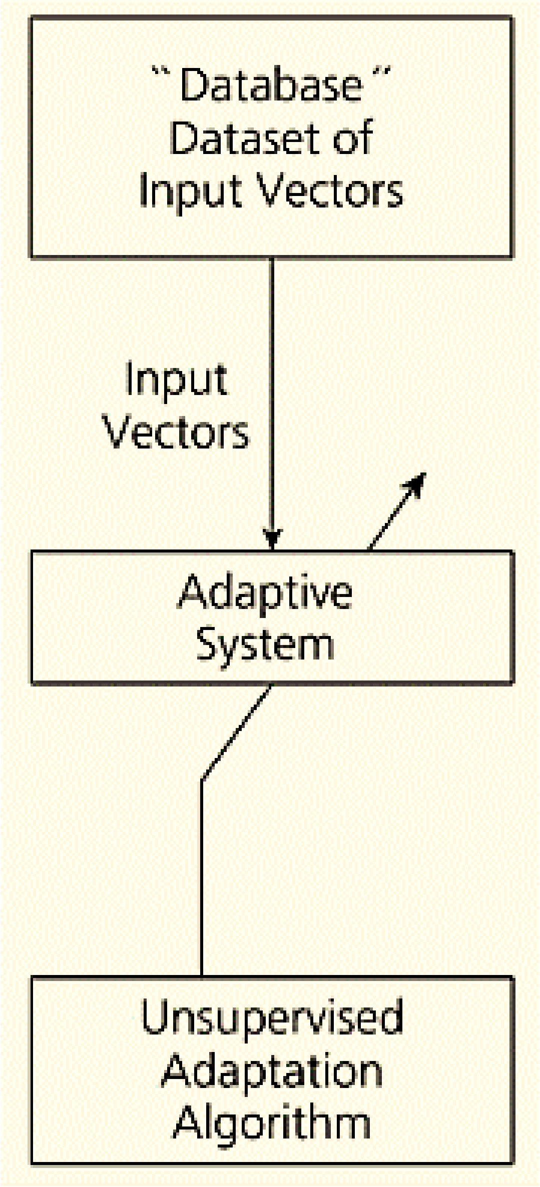
Unsupervised learning

Example:

Consider the scenario of customer segmentation for marketing purposes:

- You have data about customer behavior (purchase history, demographics, etc.), but no labels indicating customer types.
- The algorithm groups customers into clusters based on similarities in their behavior, such as high-spenders, occasional buyers, etc.
- You can then tailor your marketing strategy to these different customer segments.

Unsupervised learning



- Generally, for clustering problems, or "competitive learning"
- Grouping similar input vectors into subsets - clusters (classes)
- The elements in a cluster have a high degree of "natural association" between them
- The clusters are "relatively distinct" from one another
- Three aspects:
    - number of clusters
    - cluster centers
    - distribution of data points (pattern) to clusters

Reinforcement Learning

- A type of machine learning where an **agent** learns to **make decisions by interacting with an environment** in order to **maximize some notion of cumulative reward**.

- Unlike supervised learning, where the model learns from labeled data, reinforcement learning focuses on **learning from the consequences of actions, through trial and error**.

An agent interacts with an environment by taking actions, learning from the rewards (or penalties) it receives, and continuously improving its decision-making policy to maximize cumulative rewards over time. Through trial and error, the agent learns which actions lead to the best outcomes.

Reinforcement Learning

Key components of reinforcement learning:

1. **Agent**: The learner or decision-maker that interacts with the environment to **learn the best actions**.
   - Could be a robot, a software program, or an AI.
2. **Environment**: The external system with which the agent interacts. It **provides feedback** to the agent based on its actions.
   - In the case of a self-driving car, the environment would be the road, other cars, obstacles, and traffic signals.
3. **State**: A **specific situation** in the environment that the agent observes.
   - Could be the current location of a robot, or the position of a piece on a chessboard.

Reinforcement Learning

Key Components of Reinforcement Learning:

4. **Action**: The **choices** the agent can make to interact with the environment.
   - Could be discrete actions (e.g., "move left" or "move right") or continuous actions (e.g., "accelerate by a certain amount").

5. **Reward**: A signal the agent receives after taking an action in a particular state. The reward is a measure of **how good or bad the action was**, given the state.
   - The agent's goal: maximize the cumulative reward over time.

6. **Policy**: A **strategy or rule** that the agent follows to decide which actions to take based on the current state. It defines the agent’s behavior at any given time.
   - The policy can be deterministic (specific action for each state) or stochastic (actions chosen based on probabilities).

Reinforcement Learning

Key Components of Reinforcement Learning:

7. **Value Function:** This estimates the **expected cumulative reward for each state, given a particular policy**. It tells the agent how good it is to be in a certain state based on future rewards.
   - The agent uses this to make decisions that will lead to better outcomes in the long run.

8. **Q-Value (Action-Value Function):** This estimates the **expected cumulative reward for taking a certain action in a particular state and then following a given policy**.
   - It helps the agent choose the best action in each state.

Reinforcement Learning

How reinforcement learning works:

1. **Interaction with the environment**:
   - The agent starts in a particular state and takes an action based on its current policy.
   - The environment reacts to this action by transitioning to a new state and providing the agent with a reward (positive, negative, or neutral) based on how good or bad the action was.
2. **Learning through feedback**:
   - The agent uses the reward as feedback to adjust its policy. If the action led to a positive reward, the agent will be more likely to repeat that action in similar states. If the reward was negative, the agent will try to avoid that action in the future.
3. **Maximizing cumulative reward**:
   - The agent's goal is not just to maximize the immediate reward but to maximize the **cumulative reward** over the long term. It learns to balance short-term gains with long-term benefits (exploration vs. exploitation).
4. **Trial and error**:
   - The agent learns the best actions through trial and error. It explores different actions and gradually improves its decision-making by refining its policy based on the rewards it receives.

Reinforcement Learning

Exploration vs. Exploitation:

- **Exploration**: The agent tries new actions to discover which ones yield better rewards. It may involve taking risks to learn more about the environment.

- **Exploitation**: The agent leverages the knowledge it already has, choosing actions that are known to maximize rewards.

  - A key challenge in reinforcement learning is balancing exploration (trying new actions) with exploitation (using known actions that give high rewards).

Reinforcement Learning

Example of reinforcement Learning:

Imagine training a robot to navigate a maze.

- The robot is the **agent**, the maze is the **environment**, and at any point in time, the robot is in a specific **state** (location in the maze).
- The robot can take **actions** like "move left," "move right," "move forward," or "move backward."
- After each action, the robot receives a **reward**: +10 for reaching the exit, -1 for bumping into a wall, or 0 for moving forward without hitting anything.
- Through repeated exploration of the maze, the robot will learn the **policy** (strategy) that helps it find the exit while minimizing the number of negative rewards (like bumping into walls).
- Over time, the robot will know the best actions to take in each state of the maze.

Reinforcement Learning

Types of reinforcement learning:

1. **Model-free reinforcement learning**:
   - The agent learns purely from experience, without trying to understand the underlying model of the environment.
   - Common algorithms include Q-Learning and Deep Q-Networks (DQN).
2. **Model-based reinforcement learning**:
   - The agent tries to build a model of the environment to predict the outcomes of actions and learn faster.
   - The agent uses this model to simulate future states and make better decisions.

Reinforcement Learning

Applications of reinforcement learning:

- **Robotics:** Training robots to perform tasks like walking, picking objects, or navigating environments.
- **Game AI**: AI that can play video games like chess, Go, or even complex games like Dota or StarCraft, often outperforming humans.
- **Autonomous vehicles**: Helps cars navigate roads, avoid obstacles, and make driving decisions.
- **Recommendation systems**: Recommending content (movies, music, etc.) based on user interaction to maximize engagement.
- **Finance**: Automated trading systems that learn strategies to maximize profits in complex, dynamic markets.

Transfer Learning

- A machine learning technique where **a model developed for one task is adapted or reused for a different, often related, task**.

- This method leverages the knowledge gained from solving one problem and applies it to a new, but similar, problem.

- Instead of training a model from scratch, transfer learning allows for the **reuse of a pre-trained model**
  - saves time
  - saves computational resources
  - often leads to better performance, especially when data for the new task is limited.

Transfer Learning

How transfer learning works:

1. **Pre-trained model**: A model is first trained on a **large dataset** for a specific task learning a wide range of patterns and features from the data.
   - A convolutional neural network (CNN) trained on ImageNet, (dataset with millions of labeled images in thousands of categories), to perform image classification.
2. **Transfer of knowledge**: The pre-trained model's learned features, typically in the earlier layers, **are reused** because they capture general patterns. These layers are often left unchanged or "frozen."
   - In image recognition, the lower layers of a CNN might learn basic features (edges, textures, shapes). These general features are useful across many tasks, even those different from the original task.
3. **Fine-tuning for the new task**: The **final layers** of the pre-trained model **are modified or retrained to adapt to the new task**. In most cases, only the later layers of the network (responsible for task-specific features) are fine-tuned.
   - If the pre-trained model was initially trained on general images, it can be fine-tuned for a specific task like identifying cancerous cells in medical images. The final classification layers will be adapted to the new classes relevant to the target task.

Transfer Learning

Why transfer learning is useful

- **Reduced data requirement**: Transfer learning is especially useful when there is **limited data for the new task**. Since the model has already learned general patterns from a large dataset, it doesn't need as much new data to perform well.
  - If you have a small dataset of medical images, using a model pre-trained on a large general dataset can still result in good performance with minimal fine-tuning.

- **Faster training**: Because the pre-trained model has already learned a lot of useful features, training on the new task is much faster. You **avoid starting the learning process from scratch**, reducing the number of training epochs needed.

- **Better generalization**: Models that are pre-trained on large, diverse datasets often generalize better to new tasks. By leveraging the broad knowledge from the source task, they can recognize features that help with the target task, even when data is scarce.

Transfer Learning

Common use cases:

1. **Image Classification**: Models pre-trained on datasets like **ImageNet** can be fine-tuned to classify objects in specialized datasets, such as medical or satellite images.
2. **Natural Language Processing (NLP)**: Pre-trained language models like **BERT** or **GPT** are commonly used for tasks like sentiment analysis, text classification, or translation. The pre-trained models understand language structure, which can be fine-tuned for specific tasks with small amounts of task-specific text data.
3. **Speech Recognition**: Models pre-trained on large datasets of audio files can be fine-tuned to recognize specific dialects, languages, or audio domains.

Transfer Learning

Example of transfer learning:

- **Source task**: A deep learning model is trained on **ImageNet**, learning to recognize thousands of general object categories (dogs, cars, buildings)
- **Target task**: You want to build a model to detect specific types of flowers from a small dataset.
- **Process**:
  - Load the pre-trained model (such as ResNet or VGG) that was trained on ImageNet.
  - Freeze the earlier layers (which capture general image features like edges, shapes).
  - Replace the final classification layer with a new one specific to your flower categories (e.g., Rose, Tulip).
  - Fine-tune the model on the flower dataset.

This process allows the model to quickly adapt to the new task without requiring massive amounts of training data.

Transfer Learning

Common transfer learning architectures:

- **Convolutional Neural Networks (CNNs):** Pre-trained on large datasets like ImageNet, commonly used for image classification tasks.
- **Transformer Models**: Pre-trained models like **BERT**, **GPT**, or **T5** are used in various NLP tasks, ranging from text generation to question answering.
- **Recurrent Neural Networks (RNNs):** Used in sequential tasks like speech recognition, where pre-trained models can be transferred to new domains with limited training data.

Transfer Learning

Advantages

1. **Efficiency**: Less training time and computational power are required because you're not starting from scratch.
2. **Better performance**: It often results in higher accuracy, especially when the target task has limited labeled data.
3. **Generalization**: The model may generalize better to new tasks by leveraging knowledge from the original task.

Limitations

1. **Task relevance**: The source and target tasks need to be somewhat related. Transfer learning may not work well if the tasks are too different (e.g., using a model trained on images to predict stock prices).
2. **Overfitting**: If the pre-trained model is not fine-tuned properly, it might overfit to the new task, particularly if the target dataset is too small.