TASK 22

Supervised Learning:

Supervised learning is a machine learning approach where the algorithm is trained on a labeled dataset. In supervised learning, the input data is paired with corresponding target labels or outputs. The goal is to learn a mapping between the input data and the desired output, such that the algorithm can make accurate predictions or classifications on unseen data.

During training, the algorithm learns from the labeled examples by adjusting its internal parameters to minimize the discrepancy between its predictions and the true labels. The algorithm generalizes from the training data to make predictions on new, unseen data. Examples of supervised learning algorithms include linear regression, decision trees, support vector machines, and neural networks.

Unsupervised Learning:

Unsupervised learning is a machine learning approach where the algorithm learns patterns or structures in unlabeled data. Unlike supervised learning, unsupervised learning does not have access to target labels or outputs during training. Instead, it aims to discover hidden patterns, groupings, or relationships within the data.

The main objective of unsupervised learning is to find inherent structures or distributions in the data without explicit guidance. Common unsupervised learning algorithms include clustering algorithms like k-means clustering and hierarchical clustering, dimensionality reduction techniques like principal component analysis (PCA) and t-SNE, and generative models like Gaussian mixture models (GMM) and autoencoders.

Self-Supervised Learning:

Self-supervised learning is a variation of unsupervised learning where the algorithm learns from the data itself, creating its own supervision signals or labels. Instead of relying on human-labeled data, self-supervised learning uses techniques to generate surrogate labels from the input data.

Typically, self-supervised learning involves defining a pretext task, where the algorithm is trained to predict certain parts of the input data that have been intentionally removed or modified. By learning to reconstruct or predict missing parts of the data, the algorithm implicitly learns useful representations or features. These learned representations can then be transferred to downstream tasks, such as classification or regression.

Self-supervised learning has gained attention in recent years as it allows leveraging large amounts of unlabeled data, which are often more abundant than labeled data. It has shown promising results in various domains, including natural language processing and computer vision.

Reinforcement Learning:

Reinforcement learning is a learning paradigm where an agent learns to make sequential decisions in an environment to maximize a reward signal. Unlike supervised learning, reinforcement learning does not have access to labeled examples but relies on an interactive feedback loop.

In reinforcement learning, the agent interacts with an environment and takes actions based on its current state. The environment provides feedback in the form of rewards or penalties, which the agent uses to update its strategy or policy. The goal is to learn an optimal policy that maximizes the cumulative reward over time.

Reinforcement learning often involves concepts like exploration and exploitation, where the agent explores the environment to discover optimal strategies and balances it with exploiting the already learned knowledge. Reinforcement learning has been successfully applied to various domains, including robotics, game playing, recommendation systems, and autonomous vehicles.

Overall, the choice between supervised, unsupervised, self-supervised, or reinforcement learning depends on the nature of the data, the availability of labeled examples, and the specific problem at hand. Each learning paradigm offers its own advantages and is suited for different types of tasks and scenarios.

Classification and regression glossary

Sample or input: One data point that goes into your model.

Prediction or output: What comes out of your model.

Target: The truth. What your model should ideally have predicted, according to an external source of data.

Prediction error or loss value: A measure of the distance between your model's prediction and the target.

Classes: A set of possible labels to choose from in a classification problem. For example, when classifying cat and dog pictures, “dog” and “cat” are the two classes.

Label: A specific instance of a class annotation in a classification problem. For instance, if picture #1234 is annotated as containing the class “dog,” then “dog” is a label of picture #1234.

Ground-truth or annotations: All targets for a dataset, typically collected by humans.

Binary classification: A classification task where each input sample should be categorized into two exclusive categories.

Multiclass classification: A classification task where each input sample should be categorized into more than two categories: for instance, classifying handwritten digits.

Multilabel classification: A classification task where each input sample can be assigned multiple labels. For instance, a given image may contain both a cat and a dog and should be annotated both with the “cat” label and the “dog” label. The number of labels per image is usually variable.

Scalar regression: A task where the target is a continuous scalar value. Predicting house prices is a good example: the different target prices form a continuous space.

Vector regression: A task where the target is a set of continuous values: for example, a continuous vector. If you’re doing regression against multiple values (such as the coordinates of a bounding box in an image), then you’re doing vector regression.

Mini-batch or batch: A small set of samples (typically between 8 and 128) that are processed simultaneously by the model. The number of samples is often a power of 2, to facilitate memory allocation on GPU. When training, a mini-batch is used to compute a single gradient-descent update applied to the weights of the model.