***Chapter 1 – Introduction***

* Machine Learning is the use and development of computer systems that can learn and adapt without following explicit instructions, through algorithms and statistical models.
* A learning problem is mainly composed by a task, a performance measure and data (experience).
* 3 classes of ML problems are Supervised Learning, Unsupervised Learning, Reinforcement Learning.
* Supervised Learning focuses on finding a learn function h:X -> Y that best approximates f: X-> Y, given a dataset.

It has Classification and Linear Regression problems.

* Reinforcement Learning focuses in finding the optimal (near optimal) policy given training data in the form of state-action-reward sequences.

***Chapter 2 – Evaluation***

* Performance Evaluation in Classification are based mainly on accuracy and error rate.
* Overfitting occurs when an algorithm fits too closely or even exactly to its training data, resulting in a model that can’t make accurate predictions or conclusions from any data other than the training data. Low error rates and a high variance are good indicators of overfitting.
* Underfitting occurs when the model has not trained for enough time, or the input variables are not significant enough to determine a meaningful relationship between the input and output variables.

***Chapter 3 – Probability***

* See slides (just a recall)

***Chapter 4 – LinearClass***

* See slides (just some models)

***Chapter 5 – Linear Regression***

* Linear regression hypothesis are linear funcions -> h(x; w) = w0 + w1\*x1 + … + wm\*xm

Where wi are weights and xi are parameters (inputs from Dataset).

* Loss function: Least Squared Error ->

Immagine che contiene testo, Carattere, nero, tipografia

Descrizione generata automaticamente

* Optimal condition is reached when, starting from Normal Equation, the partial derivative of the equation wrt weights wi is 0

Immagine che contiene testo, Carattere, schermata

Descrizione generata automaticamente

* Gradient Descent:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Stochastic Gradient Descent:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

* Characteristics:

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

***Chapter 6 – Neural Networks***

* It represents a function approximator using a parametric model.
* Suitable when there is a need to associate a vector to another vector.
* Feedforward Networks lets the info go from input to output without loops, through a composition of elementary functions in an acyclic graph.
* NN are composed by node layers: input layer, hidden layers (which elaborate the data focusing on acquiring more details or discarding some details for more general aspects) and output layers.
* It uses a linear regression model to predict the output (with weights and biases at each layer).
* Activation functions allow the network to learn and model complex, non-linear relationships between inputs and outputs. Certain activation functions, like ReLU (Rectified Linear Unit), can help the network converge faster during training compared to other functions by mitigating issues like vanishing gradients.

ReLU (**Rectified Linear Unit)**: Outputs input directly if it’s positive; otherwise, it outputs zero. It’s popular due to its simplicity and effectiveness in practice.

ReLU(x)=max(0, x).

* Backpropagation composed by Forward step (simply apply linear regression + activation function until you reach the last layer, output), Loss Function MSE, Backward step which is gradient computation.
* Techniques to possibly improve NN performances: regularization, normalization, augmentation, early stopping, dropout, etc.

***Chapter 7 – Reinforcement Learning***

* Markov Decision Problem (MDP): M = (S, A, delta, r), in which next transition depends on current state and action, not on history current state is completely known.
* Reinforcement Learning (RL) is a type of machine learning where an agent learns to make decisions by interacting with an environment to maximize cumulative rewards.
* **Agent**: The learner or decision-maker that interacts with the environment.
* **Environment**: Everything that the agent interacts with. The environment provides feedback to the agent based on the actions it takes.
* **State**: A representation of the current situation or context that the agent is in within the environment.
* **Action**: Choices made by the agent that affect the state of the environment.
* **Reward**: Feedback from the environment in response to the agent’s actions. It can be positive (reward) or negative (penalty).
* **Policy**: A strategy or mapping from states to actions that the agent follows to decide what action to take in each state.
* **Value Function**: A function that estimates the expected cumulative reward (or future rewards) that can be obtained from a given state or state-action pair.
* **Q-Function**: A specific type of value function that estimates the expected cumulative reward for taking a particular action in each state and then following a policy.
* **Exploration vs. Exploitation**: The trade-off between exploring new actions to discover their effects (exploration) and using known actions that give high rewards (exploitation).
* **Model-Free Methods**: These methods learn directly from interactions with the environment without a model of the environment's dynamics.

**Q-Learning**: An off-policy algorithm that learns the value of actions taken in states and updates its Q-values based on the Bellman equation.

Immagine che contiene testo, schermata, Carattere

Descrizione generata automaticamente

Also valid for deterministic and alfa can be set to a value [0, 1).