Fondamentaux théoriques du machine learning

Organization

Repository of the course : `https://github.com/nlehir/FTML`.
This repo will contain all material related to the course.

- lecture slides
- practical sessions guidelines and solutions, exercises
- projects descriptions
- etc

Organisation of the course :

- 8×2 hours lectures.
- 16×2 hours practical sessions.

General information

- Lectures : you can ask questions during the lectures, which can be interactive.
- Some textbooks are listed in references.pdf
- You can reach me by email / discord if there is a specific notion that you are not comfortable with (in which case we might spend more time on it).
- Before most practical sessions and lectures I will send some exercises to prepare and we will correct them during the session.

Overview of lecture 1

Introduction to AI and ML

Learning paradigms

- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Examples

What makes ML a hard problem ?

- Overfitting
- Curse of dimensionality
- Optimization

Definitions

We will start by the a short general discussion on **AI and ML**.
Quick remarks :

- Machine learning (ML) is **easier** to define than AI. Several definitions of AI exists, and the term can be used in different contexts. However the practical usage of the term "ML" has been shifting a bit for the last few years.
- These topics are evolving at a fast pace, it is hard to be up to date on all their aspects.

Example 1 : digit classification

Figure – MNIST database [LeCun and Cortes, 2010]

Example 2 : robots

- Boston Dynamics robots (video)
- https://www.youtube.com/watch?v=LikxFZZO2sk
- https://www.youtube.com/watch?v=g0TaYhjpOfo
- https://www.youtube.com/watch?v=tF4DML7FIWk&t=57s
- https://www.youtube.com/watch?v=EezdinoG4mk

Example 3 : go player

Figure – Go game, won by AlphaGo in 2017 [Silver et al., 2016]

Example 4 : audio engineering helper

Figure – Semi-automatic compressor : `https://www.sonible.com/smartcomp2/`

Example 5 : coffee machine

Figure – Coffee machine (https ://www.stockresto.com/fr/machine-a-cafe/83-machine-a-cafe-conti-cc100-2-groupes.html)

Introduction examples

- - Boston Dynamics robot
  - MINST classification
  - AlphaGo
  - AI compressor
  - Coffee Machine
- All of them perform different tasks but could be qualified as "AI" (or are branded as AI) ?

Definition

- Example definition of AI : "the theory and techniques aiming at emulating intelligence".
- Hence defining AI requires definining "intelligence".

Turing test

- It is reasonable to say that the problem of AI was born at the same time as that of **Computer Science**. One of the founders of Computer Science is Alan Turing (1912-1954).
- In 1950, with the article **"Computing machinery and intelligence"** [Turing, 2009], Turing introduces the **Turing test**.
- One of the forms of a Turing Test is a game in which a computer tries to behave like a human by answering questions.

Turing, A.M. (1950). Computing machinery and intelligence. Mind, 59, 433-460.

COMPUTING MACHINERY AND INTELLIGENCE

By A. M. Turing

1. The Imitation Game

I propose to consider the question, "Can machines think?" This should begin with definitions of the meaning of the terms "machine" and "think." The definitions might be framed so as to reflect so far as possible the normal use of the words, but this attitude is dangerous, If the meaning of the words "machine" and "think" are to be found by examining how they are commonly used it is difficult to escape the conclusion that the meaning and the answer to the question, "Can machines think?" is to be sought in a statistical survey such as a Gallup poll. But this is absurd. Instead of attempting such a definition I shall replace the question by another, which is closely related to it and is expressed in relatively unambiguous words.

The new form of the problem can be described in terms of a game which we call the 'imitation game." It is played with three people, a man (A), a woman (B), and an interrogator (C) who may be of either sex. The interrogator stays in

Turing test

- Importantly, Turing stresses that the terms "think", and "machines" are far from being **unambiguous**.
- Many (most ?) questions regarding our own "intelligence" and "consciousness" as humans remain open today.
- In that respect, it is tricky to give a indisputable definition of AI.

Moravec Paradox

- It is harder to emulate or simulate simple sensorimotor capacities than level abstract reasoning.
- ie : the sensorimotor system of an ant is harder for us to emulate than beating the world champion at chess (Deep Blue, 1997).

Strong AI, Weak AI

- There exists a distinction between **Weak AI**, and a **hypothetical** AI called **Strong AI**.
- **Weak AI** (also known as **narrow AI**) designed and trained for a particular task (e.g. Siri).
- **Strong AI** (also known as Artificial General Intelligence AGI) : able to find a solution, faced with an **unfamiliar task**. This is slightly ill-defined.
  However other people probably see the current state of LLMs as a form of AGI. But again, this depends on how these concepts are defined.

Conscious reasoning

Broadly speaking, the first phases of research in AI were more dedicated to emulating the **conscious brain.**

- logic programming (Prolog)
- expert systems

History : AI Winter (1970's)

- In 1974 fundings for AI reseach started being cut by goverments because of unsuccessful projects.
- there were actually several so-called "AI winters".

Famous AI : Deep Blue (1997)

- Beat the world chess champion Gary Kasparov in 1997.
-

Figure – Deep Blue

- Despite its name, Deep Blue is **not related** to Deep learning !

Decision trees

One of the underlying mechanisms of Deep Blue were **Decision Trees** / minimax algorithm.

Limitations of the method

- Let p be the **branching factor** of the tree. (Here, the average number of children at each node)
- Let d be the **depth** of the tree.

Limitations of the method

Exercice 1 : **Computation time**

- Let p be the **branching factor** of the tree of actions. (Here, the average number of children at each node)
- Let d be the **depth** of the tree.
- What is the order of magnitude of the number of nodes in the tree ?

Limitations of the method

- Let p be the **branching factor** of the tree of actions. (Here, the average number of children at each node)
- Let d be the **depth** of the tree.
- What is the order of magnitude of the number of nodes in the tree ?
- So in order to run the minimax algorithm needs to perform $\mathcal{O}(p^d)$ evaluations.

Exercise

- In the **Othello game**, the average number of actions in each state is around 8.
- We assume that the evaluation for one leaf node takes 1×10^{-6} seconds.

-

Exercise

Exercice 2 : **Computation time**

- In the **Othello game**, the average number of actions in each state is around 8.
- We assume that the evaluation for one leaf node takes 1×10^{-6} seconds.
- How long would be the search of the minimax if we look 10 actions ahead ?

Exercise

- In the **Othello game**, the average number of actions in each state is around 8.
- This duration is too long to be used.

- In AlphaGo, the algorithm used was really different (Deep Reinforcement Learning.)

2000's and 2010's

- Around the 2000's, there has been a shift in the focus of AI.
- Machine learning and data science have become prominent research and industrial topics.
- We could argue that emulating the **unconscious brain** is more of a focus today.

2000's and 2010's

- Around the 2000's, there has been a shift in the focus of AI.
- Machine learning and data science have become prominent research and industrial topics.
- We could argue that emulating the **unconscious brain** is more of a focus today.
- An important contemporary and future trend in research might be to connect research on emulating the conscious brain and research emulating the unconscious brain.

Machine learning : proposition of definitions

- (proposed definition of) **learning** : "Modification of a behavior, based on a life experiment"
- **Machine learning** : "artificial intelligence which can learn and model some phenomena without being explicitely programmed".
- In Machine learning, some parameters are learned in an **semi-automatic way**, based on a **dataset** in order to solve a problem or to optimize a solution.

Classical programming vs ML

- Classical program : predict the total amount of money spent, based on the number of fruits bought and the price of each individual fruit (a summation is enough)
- ML program : predict which fruit a person buys, based on a database of customers and on some information about this person (e.g. buying log, age, profession).

Use cases of ML

ML is useful for problems :

- that we cannot solve direcly thanks to an explicit representation (such as the amount of money spent in the previous first example).
- that we could solve explicitely, but the computationnal ressources would be too heavy (e.g. molecular dynamics), so we **approximate** a solution with ML.

Sometimes, ML solutions to a problem are effective, but the underlying mechanism is not perfectly understood.

Denomination

- The name "machine learning" is rather deceitful !
- It is no more machine driven than any algorithm or coffee machine.
- The denomination "statistical learning" is used in some contexts.

Machine learning

Machine learning has received attention and funding because it has reached state-of-the-art efficiency on several problems such as :

- computer vision
- spam classification
- machine translation
- speech recognition
- self-driving cars

Deep learning is involved in several of these solutions.

Example 1 : ImageNet

- A database of images (more than 15M), hand-annotated in order to indicate what objects are present in the image. More than 20000 categories of images.
- Contest : ImageNet Large Scale Visual Recognition Challenge.
- The best classification (top-5 error rate) went from 25% in 2011 to $\simeq$ 15.3% in 2012.
- The technology used was deep learning, exploiting GPUs (AlexNet).

`https://paperswithcode.com/sota/image-classification-on-imagenet`

Example 2 : AlphaGo

- In 2015 : beats a professionnal player. In 2017 : beats the world champion.
- Uses several technologies : among them **Deep reinforcement learning.**
- Improvements : AlphaGo Zero, trained without a database of played games. In 2017, AlphaZero beats AlphaGo Zero after 3 days of learning.

Example 3 : AlphaFold

- Goal : to predict the spatial configuration of proteins, from their DNA sequence.
- Achieves a breakthgough performance on the CASP challenge :
  - 2018 : more than 50% GDT (Global distance test), whereas it was $\leq 40\%$ before then.
  - 2020 : $92,4\%$ GDT. At a $\geq 90\%$ score, the method is considered competitive with experimental methods..
- `https://alphafold.ebi.ac.uk/`
- Also based on Deep learning.

Deep learning $\subset$ Machine Learning $\subset$ AI

- Machine learning is not all about deep learning.
- In this course, we will not focus much on deep learning, but rather on more elementary / fundamental tools. But some parts of the course will be dedicated to neural networks.
- Deep learning is better suited for a dedicated course that allows experimentation with the network architectures.
- However, the notions that we will study are definitley useful to master in a deep learning project.
- Globally speaking, ML has a strong **experimental** component in industrial applications.

AI and Machine Learning

In the four first examples, according to you which one are NOT a Machine Learning system ?

AI and Machine Learning

- Alpha Go : Machine Learning (Reinforcement Learning)
- MNIST : Machine Learning (Supervised Learning)
- Coffee Machine : no Machine Learning (Automation)
- Boston Dynamics : no Machine Learning (Robotics)
- **edit :** while the robots on the video have no Machine Learning, Boston Dynamics seems to now show interest in the topic. ML technologies might start being implemented in their robots, especially for artificial vision.

ML revolution

- Technical progress in the computing and storage capacities
- Increase in amount of **available data**. According to IBM, 10^{18} bytes are created each day.
- Progress in algorithmic methods to analyze the data.

ML revolution II

- The speed of evolution has increased again with LLMs (first Chat GPT release : 2022)
- Spectrum of activities between maths and computer science : the jobs linked to AI and ML evolve. We could argue that the balance is more shifted towards data engineering / devops / mlops / computer science in general than 20 years ago.
- In this course, we will not focus on LLMs, or large models in general, but rather on **elementary** aspects of ML (also : I am not an LLM expert). We could almost call the course "classical ML" ?

Introduction to AI and ML

Learning paradigms
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Examples

What makes ML a hard problem ?
- Overfitting
- Curse of dimensionality
- Optimization

Supervised learning

Predict an output from an input.

- discrete output : classification
- continuous output : regression

We learn from a dataset D_n of n **labeled examples** ;

$$D_n = \{(x_i, y_i), i \in [1, n]\} \tag{1}$$

Supervised Learning : example of regression

- x : age and height of a person. Here x, is a **vector** containing 2 **features**.
- y : record on a 100 meters track

Regression : raw inputs

```
nv (node)
20 2.928290359468087800e+01 1.819015686789989559e+02
19 2.804812241859411515e+01 1.808859323205479654e+02
18 1.812029840482162513e+01 1.614667413796299229e+02
17 2.565190120950552100e+01 1.844006827739839309e+02
16 3.054629307666820637e+01 1.739981941657430866e+02
15 2.600748351639588662e+01 1.811637151224612410e+02
14 1.507418892029618895e+01 1.868432490529098970e+02
13 3.042766519165544281e+01 1.789727990680546270e+02
12 3.658212742350856672e+01 1.785726961288677046e+02
11 2.353156221187539643e+01 1.780926290043428253e+02
10 2.830054102263598281e+01 1.618048901441496241e+02
 9 2.327509936340274166e+01 1.908102611218433537e+02
 8 3.044525330771638849e+01 1.736836766109009886e+02
 7 2.488014451964265206e+01 1.794912044737207282e+02
 6 2.226526635755104166e+01 1.788433969242797446e+02
 5 1.556357798027287487e+01 1.909257627614161379e+02
 4 1.233905436641630615e+01 1.769331920900993964e+02
 3 2.430793641280866879e+01 1.712111558087560752e+02
 2 3.881928396681708193e+01 1.794596597181927962e+02
 1 1.982792868143741671e+01 1.845118060871293721e+02
21 3.118524101800569781e+01 1.636248014144912872e+02
 1 2.005698608291917395e+01 1.803034695205161881e+02
 2 2.700274799779473511e+01 1.792502475568811064e+02
 3 2.506227027010324448e+01 1.762586026486706032e+02
 4 1.907277569274998186e+01 1.826660104477087145e+02
 5 2.528781623067084183e+01 1.883758877752532612e+02
 6 3.232888827510984697e+01 1.767863031548815229e+02
 7 2.209651833311454183e+01 1.746617964376699206e+02
 8 3.087017061896954573e+01 1.780176276478124180e+02
 9 2.220465169371339087e+01 1.927466639363435377e+02
10 2.347947811151988162e+01 1.889400651201518144e+02
11 2.185924634187936633e+01 1.729108385149218350e+02
12 1.808735582805482878e+01 1.719105999524763604e+02
13 1.844754334541114105e+01 1.694675852134845400e+02
14 2.817410966313331855e+01 1.715017351672584027e+02
15 2.523219998240179862e+01 1.798536540705599407e+02
16 3.109896511888699422e+01 1.824544193833422696e+02
17 3.185907680322553404e+01 1.759604448214206229e+02
18 2.969165277770446210e+01 1.875971336737983961e+02
19 2.787730255078729158e+01 1.757878208874049051e+02
20 2.591225877027716606e+01 1.601041967482410087e+02
21 2.498572495922823578e+01 1.721740687961011247e+02
22 2.290736984087772043e+01 1.749157042606373409e+02
23 3.372473127822453876e+01 1.906407728744829058e+02
24 2.607195861993280062e+01 1.727834180024526347e+02
25 2.600320546710192460e+01 2.021277060723097918e+02
26 2.491918989168637566e+01 1.768863549264243886e+02
27 3.377987890024456163e+01 1.726478999441613098e+02
28 1.650983933678821103e+01 1.782204896060512169e+02
29 1.908717367205185766e+01 1.817607063871927551e+02
30 2.629780828398855164e+01 1.801754036532959162e+02
NORMAL  master  github/code/examples/X.txt
:NERDTreeToggle
```

Input data.

Regression : raw ouptuts

```
1.618446082230513028e+01
1.186648764508175802e+01
1.464778565654695086e+01
1.634498113765004490e+01
1.745913980223237516e+01
1.166095693440647096e+01
1.520870379572047604e+01
1.406742858858115319e+01
1.288786198880906753e+01
1.455705638818248104e+01
1.659472779771285644e+01
1.545656976520241521e+01
1.400667994864794963e+01
1.316859968706612349e+01
1.850569481499744739e+01
1.522353555851539397e+01
1.357865817589575563e+01
1.575478790439513865e+01
1.151989857308148757e+01
1.366059581731965622e+01
1.474667733562292504e+01
1.420670108820832667e+01
1.695499948932924639e+01
1.351075264826661382e+01
1.421153509268070181e+01
1.554265426599405409e+01
1.490759453066251261e+01
1.607137159898928758e+01
1.435965829372504032e+01
1.493929537374976135e+01
1.286585663123390688e+01
1.653170786752944110e+01
1.524065698484768028e+01
1.421366775637391555e+01
1.431635706484540194e+01
1.216972043271089277e+01
1.333268127294045513e+01
1.628442909496436641e+01
1.491744545525917331e+01
1.392289095842998847e+01
1.237963466793929257e+01
1.454636430040251938e+01
1.591321928754086912e+01
1.517261623645199720e+01
1.417429098481692940e+01
1.430763524632289752e+01
1.426651768644284068e+01
1.485261785696788195e+01
1.417034608280185104e+01
9.800000000000000711e+00
1.413277600956676672e+01
NORMAL  master  github/code/examples/time.txt
:NERDTreeToggle
```

Output data.

Precision

Exercice 3 :

What order of magnitude of precision do we expect to obtain with this example ?

Precision

Exercice 3 :

What order of magnitude of precision do we expect to obtain with this example ?

In a machine learning problem, there is (almost) always a statistical error associated with a prediction. Our goal is to minimize it !

Random variables

In a machine learning problem, there is (almost) always a statistical error associated with a prediction. Our goal is to minimize it ! Datasets should be seen as **sampled from a random variable**. Hence, **any quantity derived from the dataset** should be analyzed as a random variable as well.

Histograms

Histograms

histogram of height
120
100
80
60
40
20
0
140
150
160
170
180
190
200
210
height (cm)

Histograms

histogram of 100m time
100
80
60
40
20
0
10
12
14
16
18
20
22
time (s)

We assume that there is a relationship between the input variable X and the output variable Y, given by a mapping f. $Y = f(X)$.
Importantly, f might contains some **randomness**.
Equivalently, we assume the existence of a **joint distribution** ρ, such that $(X, Y) \sim \rho$.

- **Objective :** find a good **estimation** $\tilde{f}$, of f.
- $\tilde{f}$ maps an input to an output (most often deterministically). For the input x_i, we predict $\tilde{f}(x_i)$.
- In order to measure the quality of $\tilde{f}$, we use **loss functions**.

Notations

Usually, in this course

- as uppercase letter will often denote a random variable, e.g : X
- a lowercase letter will be a sample x_i or a specific value x of the variable.

Do not hesitate to ask for a clarification when a notation is unclear.

- The loss function should be a measure of the **discrepancy** between our prediction $\tilde{f}(x_i)$ and the label y_i.
- For an individual sample, **a** discrepancy is the squared loss

$$(\tilde{f}(x_i) - y_i)^2 \tag{2}$$

- Taking into account the whole dataset, the mean squared error (MSE) **loss function** writes :

$$MSE(\tilde{f}) = \frac{1}{n}\sum_{i=1}^{n}(\tilde{f}(x_i) - y_i)^2 \tag{3}$$

- Other loss functions are possible, like the absolute loss :

$$AL(\tilde{f}) = \frac{1}{n}\sum_{i=1}^{n}|\tilde{f}(x_i) - y_i| \tag{4}$$

Loss function

- The loss function is a **real number** measuring the relevance of a **collection** of parameters ($\tilde{f}$ is defined by these parameters.)
- The number of parameters depends on the situation, and varies between 1 (e.g. for a simple linear model) and millions (e.g. for some deep neural networks).

Important question

To what subset of functions does $\tilde{f}$ belong ?

Notion of optimization

We will often look for functions $\tilde{f}$ such that $L(\tilde{f})$ is a small as possible (empirical risk minimization).

Unsupervised Learning

- From a number of samples x_i, we want to retrieve information on their structure : **modelisation**.
-
  - density estimation
  - clustering
  - dimensionality reduction

Density estimation

- density estimation : anaylze the distribution of the data.

- Question : how do we constrain the distribution to fit the data ? (parametric or non parametric models)
- Loss functions are also used, although not as intuitive to define.

Clustering

Figure – K-means clustering

Dimensionality reduction

Figure – t-SNE projection of the MNIST dataset.

Reinforcement Learning

- A **more general paradigm** that describes an **agent** in a **world**.
- The standard formalization was the one proposed by Richard Sutton [Sutton and Barto, 2018].
- At each time, the world is in a state s. An agent perfoms an **action** a according to a **policy** π. When performing an action, the agent receives an **reward** r.

Reinforcement Learning applications

- Example : some Chessplayers, AlphaGo, automatic vacuum cleaner, self-driving cars (trajectory optimization, parking, etc.)
- healthcare : `https://arxiv.org/pdf/1908.08796.pdf`
- `https://deepmind.com/blog/article/alphago-zero-starting-scratch`

Reinforcement Learning

- At each time, the world is in a state s. An agent perfoms an **action** a according to a **policy** π. When performing an action, the agent receives an **reward** r.
- The agent wants to learn an **optimal policy**, meaning the policy that maximises its reward (or rather the **aggregation** of rewards).

This paradigm has many variants

- State s, action a, policy π, reward r.
- Is the policy **deterministic** ? Is it **stochastic** ?
- Does the agent have a **model** of its environment ?
- How many steps ahead should the agent look ?
- All these conditions change the way the problem should be addressed and solved. The **Bellman equations** rule the computation of the optimal policy.

Example problem

Exercice 4 : **Exploration and exploitation**

Should an agent explore its environment more or exploit what it has learnt so far ?

- These paradigms can be mixed. For instance,
  - unsupervised learning can be used in a supervised learning problem (semi-supervised learning)
  - unsupervised learning and supervised learning can be used in a reinforcement learning problem
  - **self-supervised learning** creates labels from the dataset itsself.
- Given a problem and a dataset, formalizing a problem as a supervised learning or unsupervised learning task is a **choice**.
- Often, obtaining labels for all samples is a hard, long, or even impossible process : e.g. labelling millions of images. Importantly, the dataset collection, preparation or preprocessing is at least as important as the algorithms that are run on it !

Example I

Predict the winning team of an NBA game at half-time.

- Dataset : 15 years of games (comments, text) : approximately 17000 games.
- The dataset is preprocessed to have as an input a time-series : each time contains the score **and** 10 technical features (rebounds, etc.). So for each time the dimension is 11. Each game is a matrix of size 1440×11, reorganized as a line vector.
- Output : Receiving team wins or looses (classification)
- Evaluation metric : classification error ("0-1" loss).

Example II

Predict the quantity of oil in a rock.

- Input : tomographic image of a rock.
- Output : material of the rock, average presence of residual oil in the rock (regression).

Example III

Detect issues in wind farms.

- Input : sensors on the wind turbine (wind direction, air temperature, electric tension, rotation speed, component temperature, etc.) as a time series. Each step represents 10 minutes (several years).
- Output : Power generated by the turbine (regression)
- Evaluation metric : MAE (mean absolute error).

Introduction to AI and ML

Learning paradigms
- Supervised learning
- Unsupervised learning
- Reinforcement learning
- Examples

What makes ML a hard problem ?
- Overfitting
- Curse of dimensionality
- Optimization

Polynomial regression

Polynomial regression

Exercice 5 : Is there a polynom that interpolates the data ?

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Polynomial regression

Overfitting

If there was no noise, there would be no overfitting !

No noise / no hidden variable : no overfitting

Importantly, the amount of overfitting also depends on the number of available samples :

Number of samples

Importantly, the amount of overfitting also depends on the number of available samples :

Curse of dimensionality

Two numbers are important in machine learning :

- n : number of samples
- d : dimension (number of features) of a unique sample

Both can be large and prohibitive for some algorithms.

Curse of dimensionality

Two numbers are important in machine learning :

- n : number of samples
- d : dimension (number of features) of a unique sample

Both can be large and prohibitive for some algorithms.
Exercice 6 : **Example :** what is the algorithmic complexity of the inversion of an invertible matrix $A \in \mathbb{R}^{d,d}$?

Curse of dimensionality

- n is large when the dataset has many samples.
- d is large if each sample has many features :
  - image
  - DNA sequence
  - text
  - audio/video file

Local averaging

The curse of dimensionality is the reason why **local averaging** does not work in large dimensions.

Local averaging

Local averaging

Local averaging

Local averaging

In d dimensions

Optimization

We often have to optimize functions that are **hard** to optimize.

- non convexity
- non linearity
- high dimension
- instability of estimators

Convex function

Non-convex function

The state of ML

ML is an active research area. One of the biggest effort is trying to understand the performance of deep learning.

- they are non convex
- but we experimentally manage to optimize them so that they have a state-of-the-art generalization error

In the case of neural networks with 1 hidden layer, the convergence of gradient descent to a global minimizer of the objective function has been proven in 2018 in a specific case (large number of hidden neurons), using tools from optimal transport theory ([Chizat and Bach, 2018]).

The double descent phenomenon

Figure – [Belkin et al, 2019]

References I

Chizat, L. and Bach, F. (2018).
On the global convergence of gradient descent for over-parameterized models using optimal transport.

LeCun, Y. and Cortes, C. (2010).
{MNIST} handwritten digit database.

Silver, D., Huang, A., Maddison, C. J., Guez, A., Sifre, L., van den Driessche, G., Schrittwieser, J., Antonoglou, I., Panneershelvam, V., Lanctot, M., Dieleman, S., Grewe, D., Nham, J., Kalchbrenner, N., Sutskever, I., Lillicrap, T., Leach, M., Kavukcuoglu, K., Graepel, T., and Hassabis, D. (2016).
Mastering the Game of {Go} with Deep Neural Networks and Tree Search.
Nature, 529(7587) :484–489.

References II

Sutton, R. S. and Barto, A. G. (2018).
Reinforcement Learning : An Introduction.
The MIT Press, second edition.

Turing, A. M. (2009).
Computing machinery and intelligence.
In *Parsing the Turing Test : Philosophical and Methodological Issues in the Quest for the Thinking Computer*.