The Document

João Morais

November 27, 2019

Abstract

This will be the document where I write everything I know about what I learn during my studies. I hope it serves me in the future as a "go to" document when I need to remind something.

Introduction

This document will focus on definitions, understanding concepts and transmitting intuitions/perspectives.

Each chapter will have the same structure. First the definitions and logic connections between everything with references to derivations and demonstrations that will be in attach at the end of the document. Second an overview of the derivations and how formula connect together to make a quick & easy way of finding the connections for hurry times. And third, simply one or two pages with the most important formulas from that chapter.

The first chapter will be about antennas. The second about microwaves. And the ones after that will consist of applications such as Microwave Links/ Hertzian Beams, Satellites and Radar. Maybe I'll join in a chapter about Probabilistic/Statistic Detection and Estimation of signals and one about some fundamentals of communications.

Please be aware that this is a document always at work. I'll write right here when I think a certain chapter is closed, otherwise it might be target of modifications.

Finally, there's a resource folder that I'll be refferencing frequently. It has many of my notes hand written, books, slides. Things I thought to be interesting or important. Probably all books mentioned in any section will be there.

This folder can be accessed through my Shared Drive Link

Enjoy

Contents

1 Resources & Hand Written Notes

This will be a list with all resources that you can find in the Resource folder.

Signal Processing Document

2 Linear Algebra

Before anything else, I strongly recommend watching a series of short videos that will give you a very graphical notion of Linear Algebra. That series of videos is "Essence of linear algebra" by 3Blue1Brown.

Here will be presented a concise analysis of Linear Algebra topics ranging from:

- various topics from one of the best resources to learn linear algebra, the MIT professor **Gilber Strang**'s lectures
- a summary of the above course can be found in here by Sho Nakagome ("A neuroengineer researching Brain Computer Interface (BCI)");
- geometric interpretations on a basic matrix, from here ;
- orthogonality, from MIT here ;
- important information for Signal Processing and various other applications.

2.1 What is a matrix?

We know what a matrix looks like, but what it is exactly?

Well, it can be seen as a group of vectors, as an equation and probably as many other things I haven't figured out yet.

2.1.1 A system of Equations

Let's start as seeing it as means of simplifying the process of solving a set of equations. Imagine that we have the following equations:

$$\begin{cases} 3x + 5y = 11 \\ x + 4y = 6 \end{cases}$$

These can be written in matrix from like this:

$$\begin{bmatrix} 1 & 2 \\ 3 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 11 \\ 6 \end{bmatrix}$$

And solvable through the Row Echelon Form (REF):

$$\left[\begin{array}{cc|c} 3 & 5 & 11 \\ 1 & 4 & 6 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 4 & 6 \\ 0 & -7 & -7 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 4 & 6 \\ 0 & 7 & 7 \end{array}\right] \tag{1}$$

Is important to note that to obtain the REF, only elementary equation operations can be used. These are the sum/subtraction of two equations, the multiplication of a equation by a scalar different from 0 and change the place of the equations. In the particular case of (1) the operations that took place were: row1 - 3*row2, switch rows and multiply the last row by -1.

From the our new form, is trivial to go back to the equation formula and directly attribute values to x and y. However, a few calculations are still required. The absolute best way is using the Reduced Row Echelon Form (RREF). This consists of using the rows with less elements to "cut out" elements from the other rows, leading to a very simplified matrix (see (2))

$$\left[\begin{array}{cc|c} 1 & 4 & 6 \\ 0 & 7 & 7 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 4 & 6 \\ 0 & 1 & 1 \end{array}\right] = \left[\begin{array}{cc|c} 1 & 0 & 2 \\ 0 & 1 & 1 \end{array}\right] \tag{2}$$

Note that this RREF is unique and characterised by 1's in the diagonal and in stairs, i.e the only non-zero entry in that column starting with the leftmost column. In this form, all results can be obtained immediately.

However, putting a matrix in this form isn't always possible. For instance, is impossible to do so if two rows are **linearly dependent** meaning that is possible to multiply a constant to one to obtain the other - seeing rows as equations, linear dependency can be thought of an equation that adds absolutely no other constraint to the solution set or, in other words, does the exact same as one equation that already exists thus can be discarded hence ending with more unknowns than equations.

Before further analysing the concept of linear dependency and all that derives from that, is pertinent to have yet another look at the matrix. A geometric interpretation is often useful.

2.1.2 Vectors in Space

Exactly, have you thought your matrix could be vectors in space?

Well, maybe this decomposition helps to thing about it:

$$\begin{bmatrix} 3 & 5 \\ 1 & 4 \end{bmatrix} \begin{bmatrix} x \\ y \end{bmatrix} = \begin{bmatrix} 11 \\ 6 \end{bmatrix}$$

$$\downarrow$$

$$x \begin{pmatrix} 3 \\ 1 \end{pmatrix} + y \begin{pmatrix} 5 \\ 4 \end{pmatrix} = \begin{pmatrix} 11 \\ 6 \end{pmatrix}$$

Thus we can imagine the vectors (5,4) and (3,1) is space. And one solution to obtain the vector (11, 6) is by multiplying 2 by the first vector and 1 by the second.

Note that we broke the matrix in half and gave it a meaning: "What do you have to multiply to this vector and to this vector to obtain the vector you want?" Where the vectors were the columns of the matrix. And the solution were the x and the y that were going to scale those vectors.

One can also think, despite being a bit more complicated, that what we want is to write (11, 6) in the basis of those two vectors (3,1) and (5,4). This is how the matrix represents a transformation. We are transforming the "normal" into the space defined by those two vectors. Further note that that space is given by the column vectors of the matrix! This will be important in the future.

Let's have a look at basis and spaces.

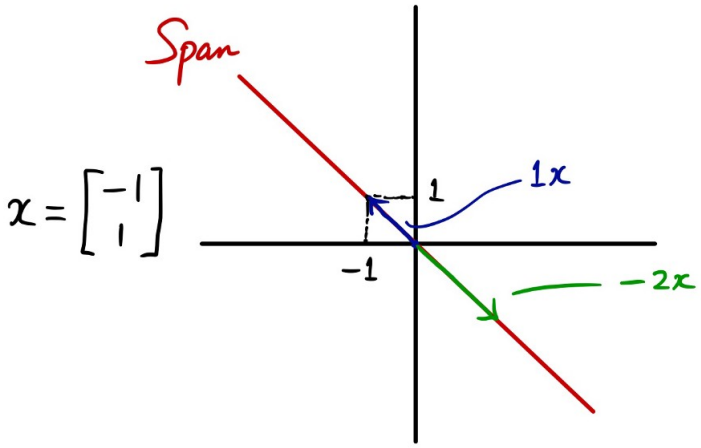
2.2 Basis, Spaces and Subspaces

A space is nothing more than a set of vectors that can be obtained from a linear combination from vectors in the basis.

If we want to take a subset of a space that is also a space we call that a subspace.

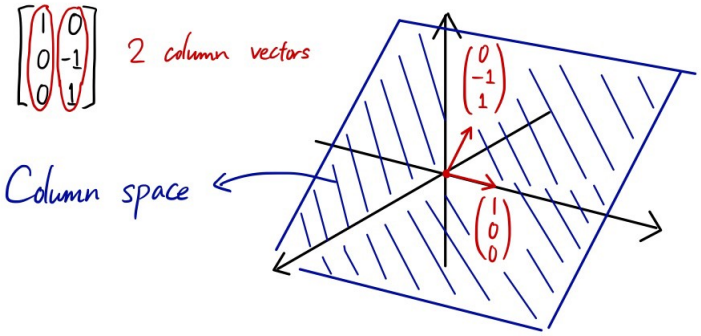
About basis: these are the vectors which when linearly combined generate a certain space. Note that one space can have infinite many basis. Actually, any set of N linearly independent vectors from a space with dimension N is a basis of that space. A set of vectors is only considered a basis if it spans a space (all linear combinations of its vectors create a space) and the vectors in it are linearly independent.

A good image to describe what "span" is:



2.2.1 Column Space or Range

The range of A (ran(A)) is the space spanned by the columns of A.



As expected, the row space is nothing more than the exact same thing for the rows.

2.2.2 Row Nullspace or Kernel

The kernel of A (ker(A)) is the space created by the vectors that when A is applied to them the result is 0.

Mathematically, the nullspace of a is nothing more than the vectors x that lead to $Ax = 0$.

Note that that thinking from before as "what vectors should be multiplied to get a certain solution" will now have a problem: there can be more than one answer to the null result. This only happens if there is a problem with the transformation. If the transformation is well done (if the matrix has full rank), then the column subspace will be the full space and anything multiplied to the matrix will have one and only one solution. If the problem/matrix is ill-conditioned (the rank is less than the dimension of the vectors of the columns), then there can be more than one solution to a give problem.

————From an answer in math.stackexchange.com: "Let's suppose that the matrix A represents a physical system. As an example, let's assume our system is a rocket, and A is a matrix representing the directions we can go based on our thrusters. So what do the null space and the column space represent?

Well let's suppose we have a direction that we're interested in. Is it in our column space? If so, then we can move in that direction. The column space is the set of directions that we can achieve based on our thrusters. Let's suppose that we have three thrusters equally spaced around our rocket. If they're all perfectly functional then we can move in any direction. In this case our column space is the entire range. But what happens when a thruster breaks? Now we've only got two thrusters. Our linear system will have changed (the matrix A will be different), and our column space will be reduced.

What's the null space? The null space are the set of thruster instructions that completely waste fuel. They're the set of instructions where our thrusters will thrust, but the direction will not be changed at all."

If you are someone with some linear algebra background you might have spotted some kind of connection between the null space and the column space. So it is worth to talk now about the rank and the relation these spaces.

2.3 Rank and Spaces relationships

The rank of the matrix is nothing more than the number of pivots in the Reduced Row Echelon Form.

Seeing a matrix as a set of equations with N variables, a full rank matrix has rank N meaning that is possible to discover the N variables without ambiguity. If the matrix is rank deficient or ill-ranked, that means we have too many unknowns for our equations.

Is common sense that if a matrix has more columns than rows, it won't have full rank as this is exactly the same as saying it will have more unknowns than equations.

In terms of the relations between the spaces: was it evident so far that the less precise is the matrix (the less rank it has compared to full rank), the more it will nullify vectors, in other words, take away their identity and the more vectors would the space that gets to zero have.

Therefore, the sum of the dimensions of the column span and nullsubspace should equal the rank that the matrix needs to be full rank, i.e the number of unknowns or simply the number of columns.

(there are a few more subspaces and we still need to explain the question that is asked every time.)

3 Antennas

Permeabilidade (magnética) do vácuo

$$\mu_0 = 4\pi \times 10^{-7} \ \text{H.m}^{-1}$$

Permitividade (eléctrica) do vácuo

$$\varepsilon_0 = \frac{10^{-9}}{36\pi} \ \text{F.m}^{-1}$$

Velocidade da luz no vácuo

$$c_0 = \frac{1}{\sqrt{\mu_0 \, \varepsilon_0}} = 3 \times 10^8 \ \text{m.s}^{-1}$$

The electric permittivity is what relates the Electric Displacement Vector (D) and the Electric Field (E). For a deeper meaning one has to look into the constitutive relations as well. Likewise, the magnetic permeability relates the Magnetic Induction Field (B) with the Magnetic Field (H).

As equações de Maxwell, as relações constitutivas e a força de Lorentz

$$\mathbf{F} = q(\mathbf{E} + \mathbf{v} \times \mathbf{B})$$

mostram que os campos fundamentais (básicos) são E e B . Excepto para o vácuo, D e H dependem de um modelo do meio.

$$\mathbf{D} = \varepsilon \mathbf{E} = \varepsilon_0 \mathbf{E} + \mathbf{P} \qquad \mathbf{P} = \varepsilon_0 \chi_e \mathbf{E} \qquad \varepsilon = \varepsilon_0 \varepsilon_r \qquad \varepsilon_r = 1 + \chi_e$$

$$\mathbf{B} = \mu \mathbf{H} = \mu_0 (\mathbf{H} + \mathbf{M}) \qquad \mathbf{M} = \chi_m \mathbf{H} \qquad \mu = \mu_0 \mu_r \qquad \mu_r = 1 + \chi_m$$

P – (vector) Polarização eléctrica — χ_e – susceptibilidade eléctrica

M – (vector) Magnetização — χ_m – susceptibilidade magnética

The Electric Polarization and Magnetization vectors show how the material changes when to being subjected to a Electric/Magnetic field. This change can be displacement of charges according due to the application of the Electric field or similar in relation to the magnetic induction field. The higher the permittivity, the most charges move with the application of the field. The higher the permeability, the higher the internal magnetic field.

4 Radio Wave Propagation

It is absolutely pivotal some nomenclature in order to understand each other.

- Vector $\mathbf{E}$ ou $\underset{\sim}{\mathrm{E}}$
- Versor $\hat{\mathbf{x}}$ ou $\underset{\sim}{\hat{\mathrm{x}}}$
- Tensor $\underset{\approx}{\mathrm{E}}$
- Amplitude Complexa $\overline{\mathrm{E}}$ ($\overline{\mathbf{E}}$ ou $\underset{\sim}{\overline{\mathrm{E}}}$)
- Produto interno $\mathbf{A} \cdot \mathbf{B}$
- Produto externo $\mathbf{A} \times \mathbf{B}$
- Operadores Diferenciais (*Nabla* $\nabla = \hat{\mathbf{x}}\frac{\partial}{\partial x} + \hat{\mathbf{y}}\frac{\partial}{\partial y} + \hat{\mathbf{z}}\frac{\partial}{\partial z}$)
  - Gradiente $\nabla \mathrm{A}$
  - Divergência $\nabla \cdot \mathbf{A}$
  - Rotacional $\nabla \times \mathbf{A}$
  - Laplaciano $\nabla^2 \mathbf{A}$

Mainly, everything that is bold is a vector and everything that has a bar on top is a complex amplitude. With these definitions out of the way, we can start:

A wave propagating across the z direction has the following (mathematical) shape:

$$e(z,t) = E_o cos(wt - kz)$$

We can only talk about waves when the field disturbance changes in time and propagates in space, thus the dependence with a spatial coordinate and time. Note further that we start off with a very simple wave: is sinusoidal, doesn't have attenuation, only propagates in one direction and only oscillates in one direction.

To write a (slightly) more general formula for a wave, while taking advantage of the complex notation that assumes already a sinusoidal wave - note that the real part of a exponential is a cossine -, we get:

$$e(z,t) = Re\{Ee^{jwt}\}, E = E_o e^{-jkz}$$
$$e(r,t) = Ecos(wt - \boldsymbol{k} \cdot \boldsymbol{r})$$

Something important to keep in mind:

In free space, electric and magnetic fields are mutually orthogonal and orthogonal to the propagation direction.

Note, however, that this is not true for propagation in matter that is anisotropic or when the waves are contained in a waveguide such like a metallic waveguide where there are TE/TM modes. But we won't consider those cases, at least for now.

Also, the vector $\boldsymbol{k} = k_x\hat{x} + k_y\hat{y} + k_z\hat{z}$ represents the direction of propagation because each of its components will contribute a phase to the oscillation of the wave.

Further note that the above wave equation is for a plane wave. Fortunately in the far-field all waves are plane waves or can be obtained from them therefore that expression will pop-up fairly often. However, most waves are spherical waves that are no more than plane waves that decay with the radius to the source, therefore the surfaces of equal amplitude are spheres.

$$\frac{A}{r}e^{j(wt-kr)}$$

The more general formula for the complex amplitude is:

$$\bar{\mathbf{E}}(r)=\mathbf{E_0}\,e^{-j\mathbf{k}\cdot\mathbf{r}}=(E_{0x}\hat{\mathbf{x}}+E_{0y}\hat{\mathbf{y}}+E_{0z}\hat{\mathbf{z}})e^{-j(k_x x+k_y y+k_z z)}$$

$$k^2-\omega^2\mu_0\,\varepsilon_0=0 \qquad \textbf{Equação de dispersão}$$

$$\mathbf{k}=k\hat{\mathbf{n}} \qquad k=\sqrt{k_x^2+k_y^2+k_z^2}=\frac{\omega}{c_0}$$

$$\nabla=\frac{\partial}{\partial x}\hat{\mathbf{x}}+\frac{\partial}{\partial y}\hat{\mathbf{y}}+\frac{\partial}{\partial z}\hat{\mathbf{z}}=-j\mathbf{k}=-jk\hat{\mathbf{n}}$$

The last part is important because the expression for the free space impedance Z_0 comes directly from the Maxwell equations and it is easier to solve them in the frequency domain (with complex amplitudes/phasors).

$$\begin{cases}\nabla\times\mathbf{H}=\mathbf{J}+\dfrac{\partial\mathbf{D}}{\partial t} & \text{Corrente de Deslocamento} \quad \text{Lei de Ampere}\\ \nabla\times\mathbf{E}=-\dfrac{\partial\mathbf{B}}{\partial t} & \text{Lei de Faraday}\\ \nabla\cdot\mathbf{D}=\rho & \text{Lei de Gauss}\\ \nabla\cdot\mathbf{B}=0\end{cases}$$

E – Campo Eléctrico [V.m^{-1}]

H – Campo Magnético [A.m^{-1}]

D – Deslocamento Eléctrico [C.m^{-2}]

B – Indução Magnética [T = Wb.m^{-2}]

J – Densidade de Corrente [A.m^{-2}]

ρ – Densidade de Carga [C.m^{-3}]

Resulting in the expression below, from where we can prove the previous statement of orthogonality between fields and direction of propagation.

$$\begin{cases}-jk\hat{\mathbf{n}}\times\bar{\mathbf{H}}=j\omega\varepsilon_0\bar{\mathbf{E}}\\ -jk\hat{\mathbf{n}}\times\bar{\mathbf{E}}=-j\omega\mu_0\bar{\mathbf{H}}\\ -jk\hat{\mathbf{n}}\cdot\bar{\mathbf{D}}=0\\ -jk\hat{\mathbf{n}}\cdot\bar{\mathbf{B}}=0\end{cases} \Rightarrow \begin{cases}\hat{\mathbf{n}}\perp\bar{\mathbf{E}}\perp\bar{\mathbf{H}}\\ \bar{\mathbf{E}}=-Z_0(\hat{\mathbf{n}}\times\bar{\mathbf{H}}) \quad Z_0=\sqrt{\dfrac{\mu_0}{\varepsilon_0}}=120\pi\ \text{Ohm}\\ \bar{\mathbf{H}}=\dfrac{1}{Z_0}(\hat{\mathbf{n}}\times\bar{\mathbf{E}})\end{cases}$$

Then many expressions come from the Maxwell equations when we consider losses. Because the actual Dispersion Equation is the following:

$$\gamma^2-j\omega\mu(\sigma+j\omega\varepsilon)=0 \qquad \textbf{Equação de dispersão}$$

$$jk=\gamma=\alpha+j\beta=\sqrt{j\omega\mu\,(\sigma+j\omega\varepsilon)}=j\omega\sqrt{\mu\varepsilon}\sqrt{1-j\frac{\sigma}{\omega\varepsilon}}$$

$$Z=\sqrt{\frac{j\omega\mu}{\sigma+j\omega\varepsilon}}=\sqrt{\frac{\mu}{\varepsilon(1-j\frac{\sigma}{\omega\varepsilon})}}=\sqrt{\frac{\mu(1+j\frac{\sigma}{\omega\varepsilon})}{\varepsilon\left[1+\left(\frac{\sigma}{\omega\varepsilon}\right)^2\right]}}$$

Tangente do ângulo de perdas
$\tan\theta=\sigma/\omega\varepsilon$

Note that the amplitudes of the field for each component can NOT be a positive number. They can be a complex number in the sense that they will influence the polarization of the field. We'll see that in the next section!

In summary:

$$Z=\sqrt{\frac{j\omega\mu}{\sigma+j\omega\varepsilon}}$$

Resumo

$$\beta=\frac{2\pi}{\lambda}$$

$$\gamma=\alpha+j\beta=\sqrt{j\omega\mu\,(\sigma+j\omega\varepsilon)}$$

$$Z_0=\sqrt{\frac{\mu_0}{\varepsilon_0}}=120\pi\ \Omega$$

$$c_0=\frac{1}{\sqrt{\mu_0\,\varepsilon_0}}=3\times10^8\ \mathrm{m\cdot s^{-1}}$$

Meio	Z [Ω]	α [Np.m^{-1}]	β [rad.m^{-1}]
Dieléctrico Perfeito $\sigma=0$	$\sqrt{\frac{\mu}{\varepsilon}}=Z_0\sqrt{\frac{\mu_r}{\varepsilon_r}}$	0	$\frac{\omega}{c_0}\sqrt{\mu_r\,\varepsilon_r}$
Bom Dieléctrico $\tan\theta=\frac{\sigma}{\omega\varepsilon}<<1$	$\sqrt{\frac{\mu}{\varepsilon}}(1+j\frac{\sigma}{2\omega\varepsilon})$	$\frac{\sigma}{2}\sqrt{\frac{\mu}{\varepsilon}}$	
Bom Condutor $\tan\theta=\frac{\sigma}{\omega\varepsilon}>>1$	$\sqrt{\frac{\omega\mu}{2\sigma}}(1+j)$	$\sqrt{\frac{\omega\mu\sigma}{2}}=\frac{1}{\delta}$	

4.1 Polarization

The polarization is the shape the field oscillation describes while looking at the wave from propagation direction with the wave going away from us . An easier way of finding if it is rotating to the left or to the right, we can use the right-hand, with the thumb along the propagation. If it is rotating along the way our (right-)hand closes, then it is a right circular polarisation.

The reference for horizontal polarisation is the earth.

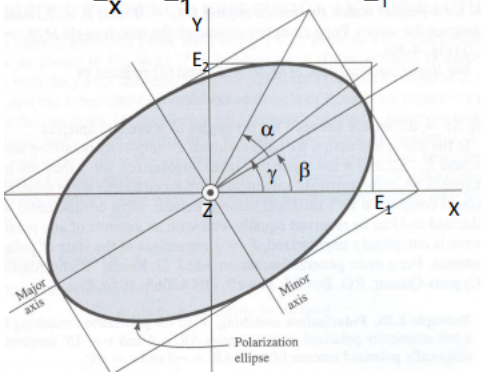
This derivation of The general Elliptic polarization shows why plane waves propagating freely can have its polarization described generally by an ellipse. Also shows that certain specific parameters of that ellipse can lead to certain more familiar polarizations, such as linear and circular.

$$\left(\frac{E_x}{E_1}\right)^2+\left(\frac{E_y}{E_2}\right)^2-2\left(\frac{E_x}{E_1}\right)\left(\frac{E_y}{E_2}\right)\cos\Phi=\mathrm{sen}^2\Phi$$

Equação da curva (elipse) de polarização

$$p=\frac{\overline{E}_y}{\overline{E}_x}=\frac{E_2}{E_1}\,e^{j\Phi}\qquad \tan\gamma=\frac{E_2}{E_1}$$

p é a razão de polarização



Estado de Polarização (α,β) ou (γ,Φ)

$$\cos 2\gamma=\cos 2\alpha\,\cos 2\beta$$

$$\tan\Phi=\frac{\tan 2\alpha}{\mathrm{sen}\,2\beta}$$

$$\mathrm{sen}\,2\alpha=\mathrm{sen}\,2\gamma\,\mathrm{sen}\,\Phi$$

$$\tan 2\beta=\tan 2\gamma\,\cos\Phi$$

4.2 Reflection

All reflections' chapter is according to Snell Laws.

A few bullet points to take away from the first class on reflection are:

- a dry soil behaves as a metal for a very small angle with the ground, typical in very long distances. "Behaving as a metal" means that the reflection coefficient will be -1, for either polarisation;
- Brewster's Angle is of incidence for which the parallel component of the field is passes completely to the other side of the surface. The reflected wave does a 90° angle with the refracted wave and only has H

field. Note that this only happens for parallel polarisation. In perpendicular is the H field that will be suppressed on reflection. The perpendicular component is never fully absorbed.

- Horizontal polarisation always has bigger reflection coefficients therefore is worst to use because it will cause more variability on the arrival. Other word for incidences with very short angles with the ground is "grazing incidences".
- Dependences with frequency increase with dielectric conductivity. Else the losses will simply be too small.

One thing that needs to be introduced are the Fresnel coefficients:

$$\Gamma_H = \frac{\sin\psi - \sqrt{n^2 - \cos^2\psi}}{\sin\psi + \sqrt{n^2 - \cos^2\psi}}$$

$$\Gamma_V = \frac{n^2\sin\psi - \sqrt{n^2 - \cos^2\psi}}{n^2\sin\psi + \sqrt{n^2 - \cos^2\psi}}$$

$$n^2 = \frac{\varepsilon}{\varepsilon_0}\left(1 - j\frac{\sigma}{\omega\varepsilon}\right)$$

Note further that the last bullet point mentioned the tangent of the loss angle.

$$\tan\delta = \frac{\sigma}{\omega\epsilon}$$

Note that if the conductivity is too small, the whole expression will be controlled by that and the frequencies would have to be very small as well to make a difference. Since we don't use those frequencies in Radio Propagation, we may say that only for a sufficiently high conductivity, the frequency dependence increases.

4.3 Spherical Earth

Surprise: the Earth is not flat, it's spherical. And this curvature needs to be accounted specially because even the direct ray may be influenced by this curvature.

5 Mobile Communications: Cellular & Radio Access Networks

Currently, 70 to 2600 MHz is the used band mainly due to propagation characteristics - a notion that we'll reinforce is that the higher the frequency, the higher the attenuation - and due to the size of the antennas we can achieve. They are human scale, from 3m to 10cm.

- In the VHF/UHF band, propagation is characterised by being:
  - almost independent of polarisation and soil electromagnetic properties;
  - essentially done via direct and reflected rays;
  - influenced by the presence of obstacles;
  - almost insensitive to refraction by atmosphere;
  - basically limited by the radio-horizon;
  - basically independent of rain, gases and others.

5.1 3GPP Specifications

These specifications are completely open! Are standards, everyone needs to know what is the standard. You just have to know where to find it.

First, find the Technical Specification Groups (TSGs): 3GPP website Specifications Groups .

What is useful for a 5G thesis is probably the RAN part. The physical layer group is TSG-RAN Working Group 1 (WG1): TSG-RAN WG1 .

In their list of specifications, if one looks closely, the following can be found:

TS 38.201	NR; Physical layer; General description
TS 38.202	NR; Services provided by the physical layer
TS 38.211	NR; Physical channels and modulation
TS 38.212	NR; Multiplexing and channel coding
TS 38.213	NR; Physical layer procedures for control
TS 38.214	NR; Physical layer procedures for data
TS 38.215	NR; Physical layer measurements
TR 38.802	Study on new radio access technology Physical layer aspects
TR 38.812	Study on Non-Orthogonal Multiple Access (NOMA) for NR

Clicking on one of them gets us to a FTP (File Transfer Protocol) page it's just needed to click in the "show all versions of this specification" and choosing the last one. Then is done! You have the study/standard about that matter!

The rest comes with experience ;)

Some free experience:

- How releases work: 3GPP Releases
- How specification numbering works: 3GPP Spec Numbering (Look that Series 38 is for Radio Technology beyond LTE, which is basically all 5G (image above))
- FAQs: 3GPP FAQs
-

6 Telecommunication Networks - Transport Networks

From the courses Internet Networks and Services (RSI in portuguese) and Telecommunication Networks (RTel in pt) I've had an insight about how the whole network is multiplexed into optic fibers and many other interesting topics such as the triple play services and a bunch of protocols that are used in today's world to make everything communicate with everything. Therefore, I propose to write a sum up of the slides and bibliography of RSI and RTel in this section. I'll mainly give importance to RTel since it is what I'm studying at the moment, but I hope to go through the slides of RSI as well.

1. Introduction
   (2 lessons)
2. Fundamentals of networks
   (7 lessons)
3. Ethernet and data centre networks
   (5 lessons)
4. SDH transport networks
   (4 lessons)
5. Optical transport networks
   (4 lessons)
6. Access networks
   (3 lessons)

1. The Internet
2. Quality of Service on the Internet
3. IP Network Models
4. Next Generation Networks
5. The Telephony Network
6. Technologies for data transport
7. MPLS - Multi-Protocol Label Switching

6.1 Introduction

Definition *Telecommunications* : is the transmission of information at a distance through the use of electromagnetic signals.

Definition *Telecom. Network* : collection of nodes and links with the purpose of interchanging these signals in order to have an information flow.

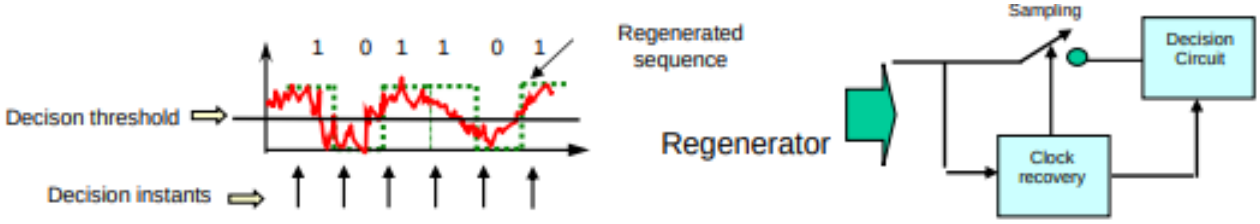
These Telecommunication Networks can be public, owned by Telecom. / Network Operators that use that network to provide services to the general public, or can be private, used by a company to connect infrastructures. Many of these private networks also rely on leased links by public networks.

There are mainly 3 layers in a network: the backbone or core, the metropolitan and the access layer. As expected, the access layer collects the traffic, connections to homes, offices, everywhere the internet is required. The metropolitan area connects different parts of the city that use that network, typically with a ring (made out of optic fiber). The core is the most extensive layer, with a mesh of nodes and very extensive links that connect cities of the whole world. Hundreds or thousands of km's is not atypical.

What makes possible to communicate with everyone connected to the internet is that the networks of both operators are also connected.

As a public service, the public networks must provide fidelity (transmit the information without loss of changes) and reliability (less than 3 minutes down per year).

Nowadays, most of transmission is digital. A series of pulses is transmitted through a channel with attenuation, dispersion, interference from other signals and noise. Therefore what reaches the other side is considerably different and has to be estimated what the original input was.



As having a dedicated physical infrastructure for each service would be far to expensive and messy, the big majority of services share the same channel, the optic fiber. It is easily shared because of the available bandwidth

in it. Thus, the signals are multiplexed at the entrance, using different wavelengths (**Wavelength-Division Multiplexing (WDM)**) and de-multiplexed at the other end, to follow each one to their device that is requiring the service.

Note that WDM is exactly like **Frequency Division Multiplexing (FDM)** but in the optical domain. Technically they are exactly the same as changing the wavelength is nothing more than changing the transmit frequency.

A single mode optical fiber can reach throughputs of 10 Terabits per second.

Remember that there are many ways of scheduling frames in a multiplexer. With time slots or doing it statistically are two ways. Also, there are 2 types of switching: packet switching and circuit switching. Circuit is when a channel is constantly reserved for a certain application even if it is not being used. Packet switching allows a much better share of the resources. Packet switching principle is based on sending packets whenever there's a packet to transmit and use all the resources to do so as fast as possible. Therefore, the "speed" of the internet depends a lot on the amount of people that are accessing it.

Regarding the physical infrastructures for the transmission of data, those go from satellites, well the open space in general as microwave links are also a thing, twisted pairs, optical fibers and even a few more that are less common.

Finally, a look at the tendencies is pertinent. The traffic is increasing constantly, at a rate of 30% a year, therefore the network must be upgraded as the time passes as well, or else it won't be able to handle the traffic of the future. Not only are the links being upgraded since now we have fiber to the home, terminating really in our router, but each node must be upgraded as well to cope with the traffic resulting in new switches, larger datacentres, ect... However, all of this must be standardised to guarantee compatibility between countries, operators, manufacturers and users and to ensure minimum quality of service for all users. This standardisation is done by the International Telecommunication Union (ITU) that has two main sectors of interest: the -T sector regarding telecommunications in general and the -R sector for radiocommunications that is more focused in point-to-point, mobile, satellite links, ect... Additionally, ETSI, ISO, OSI, ANSI, IEEE are some of the main organisations that standardise technologies. IEEE is the best :)

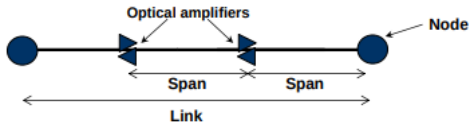
6.2 Networks Fundamentals

A network is composed of nodes and links and can be represented by graphs. However, a clear distinction between networks and graphs has been made in class: a network is a graph with a few more numbers that represent various network parameters. These parameters will be talked later, but can be delay of a link, distance, ...

The physical topology concerns the physical connections that are in place while the logical topology for a certain case concerns the actual flow of information. Even though every computer is connected in a network, maybe the information always flows from one to the others and the graph that represents that has much less links.

A link can be unidirectional or bidirectional. If the link is unidirectional, sometimes is referred to as an arc. $e_1 = (v_1, v_2)$ is the representation of a link, and the order of the nodes matter if it's an arc.

In optical fiber networks, or other networks that require amplifiers, the space between amplifiers (distance the signal has to go attenuating) is called a span.



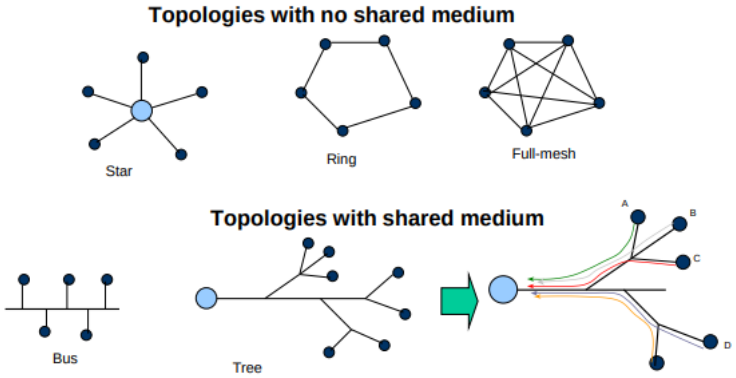
In a graph there's N number of Vertices(v_i), and L number of Edges(e_j). And the degree of the vertex is the number of edges it has. It's called the order of the graph, it's number of vertices, and the size of the graph it's number of edges.

Directed graphs only have unidirectional links, while undirected graphs only have bidirectional links.

The reason to make the distinction between directed and undirected graphs (unidirectional and bidirectional edges): in case of optical fiber, which is what connects most of long distance networking, is required to use more than one fiber. Because and optical emitter can't receive as well (at least in the same fiber). Also, if amplifiers are required, note that they are directed as well.

A path can be represented by a set of links, starting at some node. Source and sink are the names for the first and last vertex of that path.

6.2.1 Network Topologies



Bus, Ring and Star are the main physical topologies. Tree as well.

A tree is simply a graph with no cycles.

6.2.2 Network representative Matrices

A graph can be represented with an **Adjacency matrix (A)** , with $a_{ij} = 1$ if there's a direction from the vertices i to j.

The average node degree is given by the average of each node's degree which won't be more than the sum of all links, times 2 divided by the number of nodes. Times 2 because each link contributes for the degree twice, once at each end.

$$\delta_i = \sum_{j=1}^{N} a_{ij}$$

The average node degree is given by

$$<\delta> = \frac{1}{N}\sum_{i=1}^{N}\delta_i = \frac{2L}{N}$$

The Network diameter (D_R) is the maximum number of links between nodes through the shortest path between them. D_R is the longest of the shortest paths between every node.

Every link can have an associated cost (a function of distance, delay, reliability, actual cost, or other parameters) and a capacity (u_e denotes the capacity of node e).

The link capacity can be measured in any traffic unit that is appropriate for the problem. In packet networks: bit/s; in SDH networks: STM-N; in OTN networks: OTU-k; in WDM networks: number of wavelengths

Despite similar to an Adjacency Matrix, the **Demand matrix (D)** is slightly different. $d_{ij} = 1$ if the traffic flows from the vertex i to the vertex j.

Note that the diagonal of this matrix should be empty, or else it would mean that a certain node would receive information from himself, which makes no sense.

The mean number of demands is the number of demands divided by the number of nodes.

$$<d> = \frac{1}{N}\sum_{i=1}^{N}\sum_{j=1}^{N} d_{ij}$$

The number of unidirectional demands (edges) in a case of full mesh logical topology is $D_1 = N(N-1)$. N nodes x the other N-1 nodes. Note that One other way of seeing it is that the D matrix is $N \times N$ but we need to take N away due to the empty diagonal. $D_2 = \frac{D1}{2}$ is the number of bidirectional demands, which is when only the top triangle of the D matrix is considered. This is usual because with bidirectional links the D matrix will always be symmetric.

Another interesting matrix is the **Traffic matrix (T)** and it's used to denote traffic intensities. It only has entries different from 0 in the exact same places the Demand matrix has. It's used for static traffic designs.

In transport networks the traffic units is the type of client signals: Ex: E3 (34 Mb/s), STM-1 (155.51 Mb/s), GbE (1 Gb/s), 10 GbE (10 Gb/s) ,etc. This traffic units must be converted to traffic units appropriate to be used in network design. These traffic units must be VC-n in SDH networks and in OTN networks ODU-k signals.

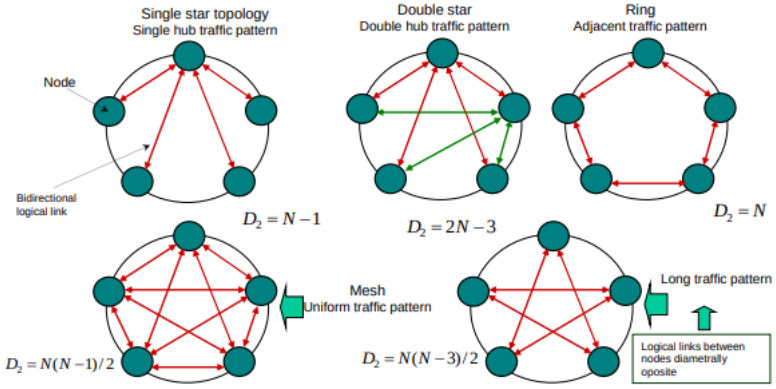
Considering now a Dynamic Traffic case, a few concepts arise:

- Average intensity of data flow between two nodes;
- Traffic bursts are time intervals where the flux of data is considerably higher than the average rate;
- Peak rate is the maximum instantaneous intensity.

Aggregation level or multiplexing level reduces the traffic *burstiness* because there less often flows get fully used and is more likely that the information can flow at a constant pace instead of in bursts.

Note one important distinction between Logical and Physical Topologies that we haven't done yet is that there can be many logical topologies on one physical topology. This can happen in the following way:

- **In a ring network there are diferent ways how the traffic can flow in a network leading to different network topologies (single star, double star, ring, mesh, etc.). *N*: number of nodes; D_2: number of bidirectional (two way) logical links.**



Routing is how a packet travels in a network. Therefore, routing ends up being the map between logical and physical topologies.

In optical networks the path between two nodes is usually called "lightpath".

We can also define a **Cost Matrix (C)** where each element c_{ij} represents the costs between nodes i and j.

The path can be performed manually (static routing - Demand matrix is time invariable) or dynamically, through routing algorithms (dynamic routing - Traffic matrix is time dependent, with constant arrival and termination of new demands).

Additionally, if the a given traffic demand(connection) is able to use more than one route, it is called a multipath routing process, else it is a mono-path process. Because there are usually many paths connecting two nodes, some metrics are taken into account when choosing which to follow:

- 1) Minimizing the network cost;
- 2) Minimizing the traffic in the most loaded link;
- 3) Minimizing the number of hops (number of links in the path);
- 4) Minimizing the path distance;
- 5) Maximizing the protection capacity, etc.

Most of the routing strategies incorporate some sort of shortest path algorithm to determine which path minimizes a particular metric, which can be for example 1), or 3), or 4). Note that the same algorithm used to 4) can be applied to 3) making all link distances identical.

From the physical topology, described by a graph G(V,E) and the traffic matrix T, describing all the traffic demands to be routed, one can perform shortest path algorithms such as Dijkstra's algorithm.

Order the demands according to a certain sorting strategy:

- Shortest-first: The demands with the lowest number of nodes in its path come first in the list;
- Longest- first: The demands with the highest number of nodes in its path come first in the list;
- Largest-first: The demands with the highest number of traffic units come first in the list;
- Random ordering: the demands are not known initially.

Route de demands according to the orderings. To break a tie choose the path that minimizes the load in the most loaded link.

Dijkstra's Algorithm

Consider a generic node *i* in a network with *N* nodes from where one wants to determine the shortest path to all the other nodes in the network. Lets l_{ij} be length (cost) of the link between node *i* and node *j* and d_{ij} the length of the shortest path between node *i* and *j*.

Algorithm:

1) Start with the source node i in the permanent list of nodes, i.e. $S=\{i\}$; all other nodes are put in the tentative list labeled S'. Set $d_{ii}=0$ and $d_{ij}=\infty \quad \forall j \neq i$.
2) For all the adjacent nodes to i set $d_{ij} \leftarrow l_{ij} \quad \forall j$ adjacent to i .
3) Identify the adjacent node j (not in the current list S) with the minimum value of d_{ij} (permanent node), add it to the list S ($S=S\cup\{j\}$) and remove it from ($S'=S'\setminus\{j\}$). If S' is empty stop.
4) Consider the list of neighboring nodes of the intermediate node j (but not consider nodes already in S) to check for improvement in the minimum distance path by setting $d_{ik} \leftarrow \min(d_{ik}, d_{ij}+l_{ij})$. Go to Step 3.

Now is pertinent to introduce yet another matrix, the **Hop Matrix (H)** where each element h_{ij} denotes the minimum number of hops from node i to node j.

The average number of hops per demand is nothing more than the sum of the hops of all demands divided by the number of demands. The number of demands was set as the unidirectional links between two nodes. Therefore, if 2 nodes share information between themselves (don't need to have a physical link, a logical one is enough) then there's a demand.

Therefore, the average number of hops can be computed:

$$< h >= \frac{1}{D}\sum_{i=1}^{N-1}\sum_{j=i+1}^{N} h_{ij}$$

Note that coherence is key here. If the amount of demands are the bidirectional demands, then the number of hops considered should only be the top half of the hop matrix. **If links are bidirectional, then there will always be a symmetry in these matrices** and we should compute the average with amount that mean the same thing.

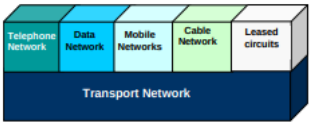
As means of simplifying this calculation, because hops and demands can be dynamic, a way of having a notion on the order of magnitude and get a fairly good approximation, when the number of nodes N is $4 \leq N \leq 100$ and the average node degree $< \delta >$ is $2.5 \leq < \delta > \leq 5$, is by computing the semi-empirical relation:

$$< h > \cong 1.12\sqrt{\frac{N}{< \delta >}}$$

6.2.3 Layers

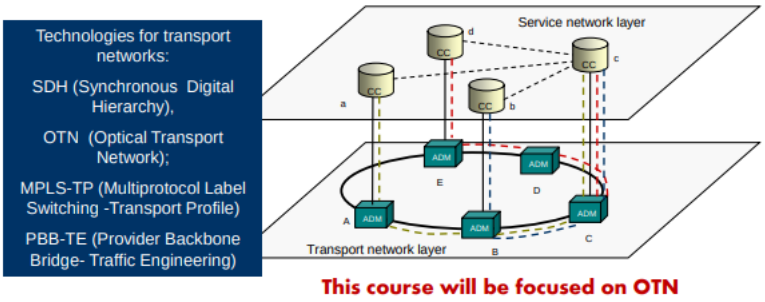
Typically, there's a layered structure in the network. The layer above acts as a client of the layer beneath and each layer appears a black box that supplies a service to the layer above.

The service layer is the one closer to us.



Add/Drop Multiplexing(ADM) are multiplexers controlled by **Control Centres (CC)** that decide what to add and what to drop form the fibre. Note that these don't manage the network, they just use it.

Nowadays, apart from local networks, everything is connected with fibre. Therefore, is pertinent to mention 4 key technologies for transport networks:



This course will be focused on OTN

SDH is also used in Hertzian links, MPLS is Multiprotocol Label Switching and is very useful for routing through different technologies and to choose more carefully the paths. As said above, OTN will be our transport technology.

OTN has become the new standard for a while now and has a few differences compared with SDH. The most important of which is distinction between fixed frame size and fixed rates. SDH has a fixed rate while OTN can increase it's rate to match the client's and this is very important for scalability and being more future proof

A very important distinction between the transport and the service layers is that the representation in the service layer has nodes connected with **logical topologies** while the transport layer has nodes connected with **physical topologies** .

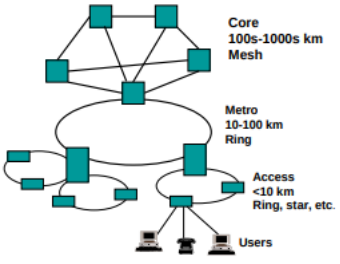
OTN	SONET/SDH
Asynchronous mapping of payloads	Synchronous mapping of payloads
Timing distribution NOT required	Requires right timing distribution across networks
Designed to operate on multiple wavelengths (DWDM)	Designed to operate on multiple wavelengths
Scales to 100Gb/s (and beyond)	Scales to a maximum of 40Gb/s
Performs single-stage multiplexing	Performs multi-stage multiplexing
Uses a fixed frame size and increases frame rate to match client rates	Uses a fixed frame rate for a given line rate and increases frame size (or uses concatenation of multiple frames) as client size increases
FEC sized for error correction to correct 16 blocks per frame	Not applicable (no standardized FEC)

The network management systems sends configurations through the **Data Communication Channel (DCC)** . Moreover, not all parts of the network are the same!

In the core and metro networks the physical topology is normally determined by the transport layer.

The access network uses a large variety of technologies and topologies and is responsible for the largest fraction of the operator´s investment on a telecom network.

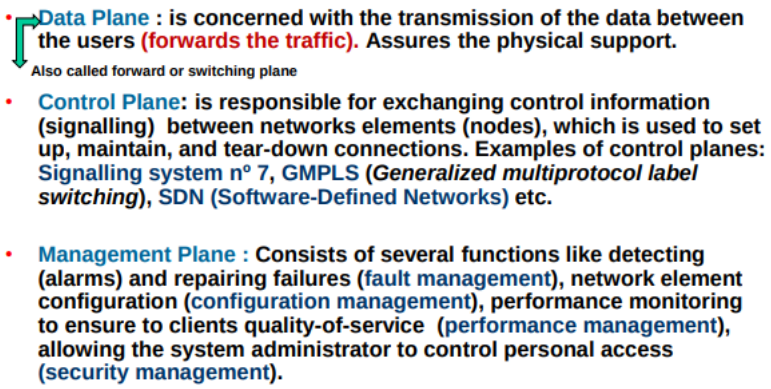
Transmission technologies in an access network: twisted pair (copper), coaxial cable, optical fibre, wireless solutions (FWA).



One important distinction in terms of how things are connected physically is evident comparing the access networks with the other parts of the network. Access Networks, as opposed to what happens to the rest of the network, uses twisted pair or one optic fiber. This is because of the cost versus the bandwidth a fiber offers.

One fiber can carry several times the traffic of one user, therefore can be used for multiple users. Moreover, very often the final leg is done with twisted pair due to cost reasons.

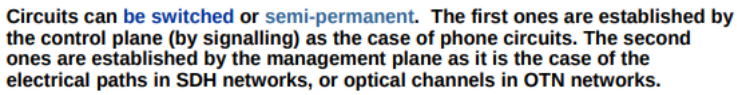
In a more abstracted way, one can identify three planes:



As you already know, there are circuit switched or packet switched networks.

An important note: Packet switching can be connection oriented (MPLS) or simply connectionless (pure IP packets, best effort)

Circuit Switched require circuit establishment and tear-down at the beginning and at the end, respectively. However, a distinction is made between physically *switchable* circuits and semi-permanent circuits. The first ones, a physical circuit is easily switched to connect one end to the other, while the second type regards circuits that are more static, that are much more difficult to switch. The semi-permanent must be switched by the administrators in order to attribute these circuits to a user for a somewhat long period of time (not just one transmission).



The biggest difference is the amount of time each one uses de circuit for. Switched circuits is around minutes, Semi-permanent are months.

Then, of course, there are packet switched networks that allow a much better share and efficient use of resources.

The key notion to have is that all services use the transport network! This is the highway of data! Nowadays it is impossible to have dedicated physical connections for each service. And this section of *Networks - Transport Networks* is based on that!

Telephone Networks

Telephone Networks before used circuit-switching. Note that the only ones that use circuit-switching are the landlines! Nowadays, our mobile communications are able to replace this fixed circuits lines

Local exchanges (Access) - are small switching centres that serve a small area.

Transit exchanges(metro, core) occur in **Primary trunk exchanges** is used to transfer the traffic and to interconnect several circuits.

They usually have a physical connection between them, but not one of their own. It wouldn't be viable to have dedicated circuits. It's here that the transport networks comes.

Different components have different topologies. The network is a hierarchy, having different topologies in different parts of the network.

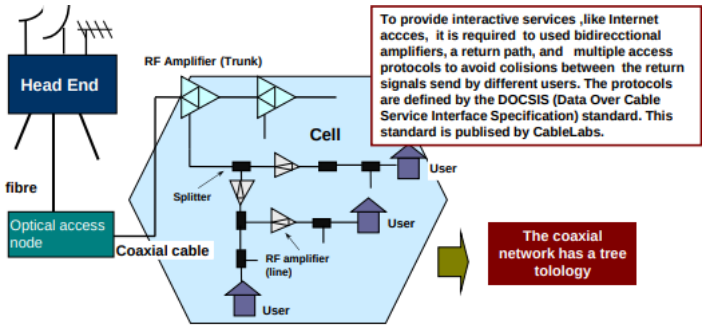
Digital Circuit-switching meaning that the data being switched is of digital nature.

Hybrid Fibre-Cable Network

There's an head end that distributes all the channels for all the users.

The network is called hybrid because it uses both fibre and coaxial cable.

Excellent example of a tree network. Like all the networks with share mediums, it required multiple access protocols to allow everyone to use the medium (FDMA and TDMA are 2 of the multiple access technologies).



The transport network is represented by the optical fibre that connects the head head to the optical access node. Like the cellular communications, one optical access node covers a certain area, with coaxial cable.

IP Networks

There are 2 ways of sending IP packets:

- Without Connection, it is necessary to have a buffer to reassemble all the IP packets. The packets are simply send to the network in a best effort way.

- With Connection: when certain QoS is required, then MPLS is used, so that certain resources can be reserved and the QoS met. MPLS establishes a virtual circuit before starting transmitting the packets. MPLS acts between layer 2 and layer 3. It places a label between the layer 2 and layer 3 labels.

There are 2 types of MPLS routers: - LER - Label Edge Router are the ones that put the labels and take them out; - LSR - Label Switching Router simply take labels out.

MPLS works by pre-establishing a path for information flow. Then, at each hop, according to the label id's available in each LSR the label arrives with one number and it leaves with other, based on the MPLS routing table in each MPLS capable router. This table entries are created based on the MPLS protocol and pre-path find procedures.

This is how a MPLS table looks like:

	Input port	Input label	Ouput port	Output label
FEC 1	1	15	2	10
FEC 2	2	25	3	20

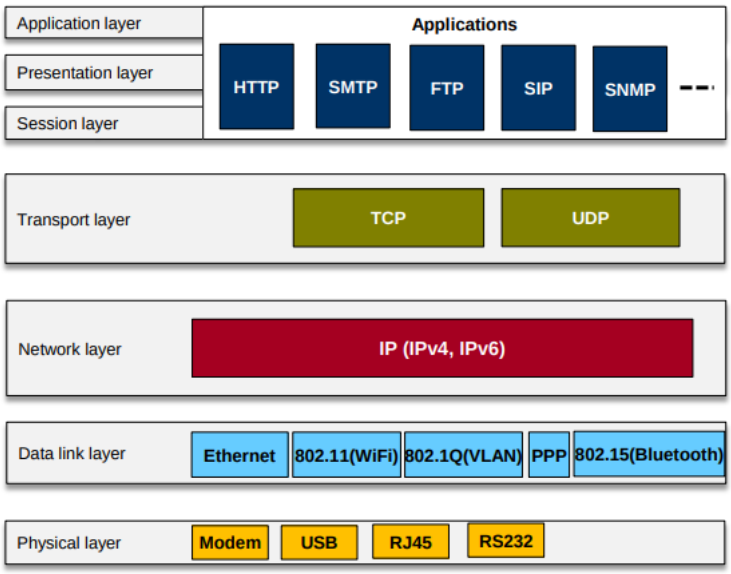
Generalized MPLS (GMPLS) is the *de facto* (practice that exists practically even though it is not formalized by law) of the control plane of the Wavelength Switched Optical Network (WSON).

Label Switching allows Traffic Engineering:

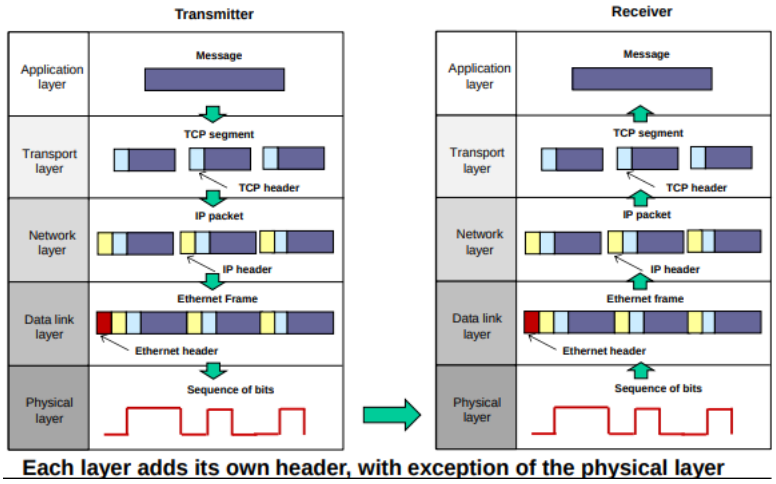
Traffic Engineering (TE) deals with a set of procedures required to optimize the performance of telecommunications networks and use network resources in an efficient way. It permits, for example, to route the traffic in order to avoid congestion and to maximize the transported traffic.

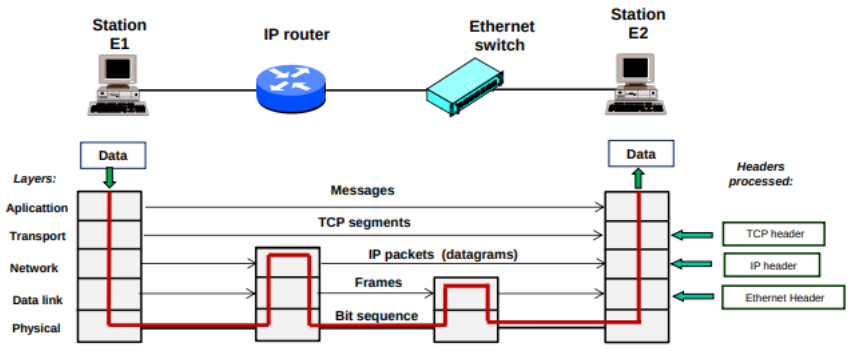
6.2.4 Layered Model Overview

Open Systems Interconnection Model:



Transport layer connects applications. Network layer connects machines with different IPs. Data link layer connects two machines with their interfaces' MAC addresses. Physical layer is what is responsible for putting the data into a cable or into the air to perform the actual transmission. For instances, spectrum, modulation and coding techniques and intervals of transmission.





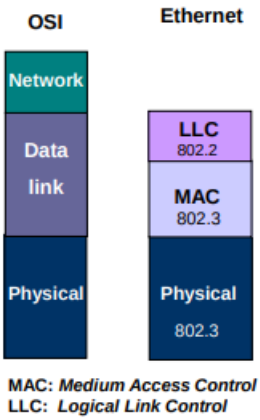
6.3 Ethernet Networks

Most of today's network traffic is generated from Ethernet interfaces. Made from twisted copper pair, through slightly more advance techniques and protocols is still possible to get quite a good throughput with a low cost cable.

Uses CSMA/CD Carrier-sense multiple access with collision detection because it can sense if there's anyone using the bus at all times. WiFi for instances CSMA/CA is used since we may be causing a collision we can't detect.

The Ethernet protocol 802.3 is used in all parts of the transport layer, not only at the access level and specially not only in LAN (Local Area Networks). Note that the Ethernet protocol was standardised by IEEE and the standard specifies not only the physical specifications but also Data Link Layer frame formats. WiFi 802.11 is another protocol that includes physical, like transmit powers and modulations, and Data Link specifications

A rule of thumb is to use optical cables for distances above 100 metres.

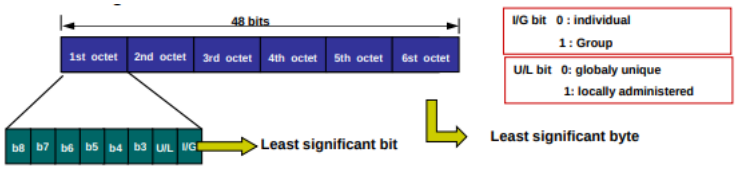


- The LLC (Logical Link Control) sub-layer is responsible for the flow and error control between the nodes.
- The MAC sub-layer is responsible for the media access control, addressing, error detection, frame delimitation, by organizing the bit sequences into frames.
- The physical layer deals with the bit transmission and reception, with the electrical, optical and mechanical properties of the interfaces, with the type of connectors used, etc.

The Data Link layer is responsible for frame processing, and error detection through the calculation of a CRC (Cyclic Redundancy Check). If a frame has errors, it is discarded. Most of error correction is done higher up in the stack. TCP guarantees error free transmissions when it says they were successful.

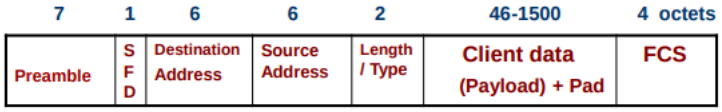
MAC stands for Media Access Control. Thus, the MAC addresses are addresses to access the physical media. CSMA protocols are placed in this layer.

There are 2 bits in the MAC address, the two least significant ones form the first octet.



U/L refers to the uniqueness of the interface. The MAC address can be changed becoming a local one. I/G refers to one interface only or a group that has that MAC address that received always the same thing.

Write *ifconfig* or *ip a* in linux (or *ipconfig* in windows) is possible to see this MAC address.



Preamble: sequence of 7 octets (0101....) permits the recovery of the signal clock in the receiver, when it operates in burst mode.
SFD (*Start of Frame Delimiter*): Pattern of 8 bits (10101011) that indicates the beginning of the frame.
The destination and source address are fields with 6 octets.
Length/type: sequence of 2 octets to indicate the length of the data field (≤ 1500) or the type of frame (≥1536) (ex: data frames (IPV4, IPV6, MPLS, etc) , 802.1Q, 802.1ad, control frame (faults, flow, etc.))
FCS (*Frame Check Sequence*): Uses a four-octet CRC code calculated over al the the octets apart from the preamble and SDF fields.

Because there are only 1500 bytes to address (to count) with the Length/Type field, then there will be quite a few empty bits in this field. The rest of the bits are used to denote the type of the frame:

If the value in the field lies from 0 to 1500 it indicates the length of the Date field . If the value ranges from 1536 to 65 535 it indicates the nature of the Data field. Examples of field types:

0x0800: IPv4; 0x8600: IPv6; 0x8808: Ethernet flow control; 0x8847: MPLS unicast; 0x8848: MPLS multicast; 0x8100: IEEE 802.1Q; 0x88A8: IEEE 88A8.

The purpose of the Preamble is to achieve clock synchrony.

6.3.1 Multiple Access

TDMA, FDMA (WDMA), CDMA all have the same principle: divide signals in a given domain.

TDMA divides signals in the time domain: different transmissions occupy different time slots.

FDMA divides in frequencies, WDMA divides in wavelengths. The only difference is that one is more used in wireless applications and the other with fibres.

OFDMA starts combine the previous ideas: divides in the time and in the frequency domains, therefore is able to attribute resource blocks with much better efficiency. A use of OFDMA is in 4G and 5G where a physical resource block is nothing more than a set of sub-carriers used for a given time period, usually called one TTI (Transmission Time Interval).

CDMA is used in 3G and with GPS, so that different satellite can transmit all at the same time.

Then there's another category of multiple access techniques like CSMA. Namely CD and CA, collision detection and collision avoidance.

Collision Detection is when a collision is possible to be detected and after being detected certain steps are taken to minimize the risk of happening again.

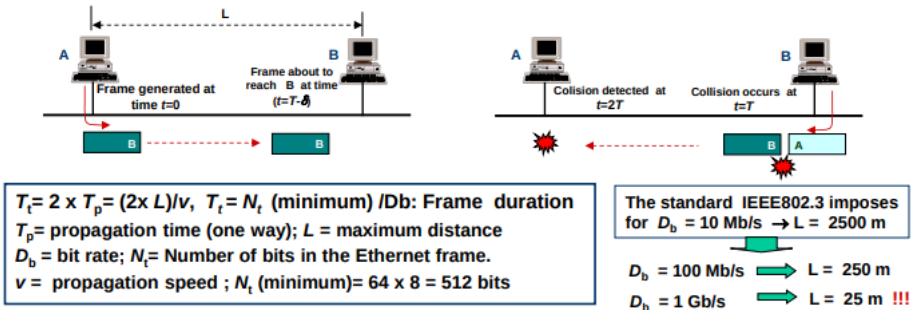
Collision Avoidance is when the collision can't be detected and should at all costs be avoided.

A curiosity: he speed of light inside an optical fibre, considering an average glass refractive index of 1.5, is:

$$v = \frac{c}{n} = 2 \times 10^8 m/s$$

In cabled Ethernet, CSMA/CD is used. Disregarding all other phenomena that limit the debit - dispersion that leads to distortion, attenuation, etc. . . - the use of CSMA/CD is limited by the distance that the information can travel before one discovers that a collisions as occurred. In essence, if the maximum debit is 10 Mbit/s and the maximum ethernet frame is around 1500 bytes long, it will take $\frac{1500*8bits}{10\times10^6 bits/s} = 0.0012s$.

The electrical signals propagate at speed of light - don't confuse with the speed of light in the fibre, in the fiber the light is propagating in glass, here there's a wave propagating in TEM mode along a twisted pair or coaxial cable. Therefore, in order to notice a collision, the signal must be able to go all the way to the other end and comeback BEFORE the sender is ready to send another frame. Mathematically: $T_{max-frame} \geq 2T_{propagation}$. This is because, if there is a collision at the other end, the sending machine must receive that collision before finishing the transmission, otherwise it would keep sending frames without knowing about the collision.



Since $T_{propagation}$ is proportional to the length of the connection, and the time of transmission is inversely proportional to the bit rate, then the bigger the rate, the smaller the maximum distance between machines can be.

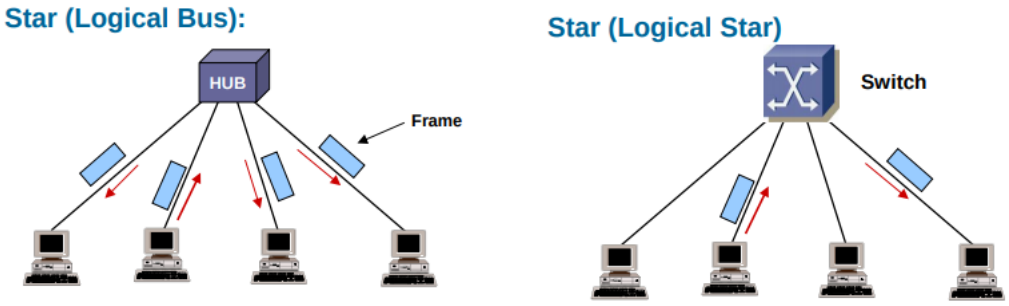
That is also why CSMA/CD is not used when the rates get too high!

Type	Bit Rate	Mode	Topology	CSMA/CD	Medium
Ethernet	10 Mb/s	Half-duplex	Bus	Yes	Coaxial cable
Fast-Ethernet	100 Mb/s	Half e full duplex	Star	Yes	Copper + Fiber
Gigabit-Ethernet	1 Gb/s	Half e full duplex	Star	Yes	Copper+ Fibre
10 Gigabit Ethernet	10 Gb/s	Full duplex	Star	No	Copper+ Fibre
100 Gigabit Ethernet	100 Gbit/s	Full duplex	Star	No	Fibre

Half-duplex → CSMA/CD (Carrier Sense Multiple Access/Collision Detection)
Full-duplex → Switched Ethernet

Note that this is only necessary if a medium is shared! If it is dedicated, it is not necessary.

A switch is a layer 2 - the Link Layer - equipment that commutes Ethernet frames.



A Hub only has physical layer, propagates to all interfaces everything. A Switch has layer 2 and Mac Address Tables. A Router has layer 3 capabilities and is able to route IP packets.

Switches with higher capacities are higher up in the grid, closer to the core, because they have to handle many more traffic flows.

Some functions of switches:

- Forwarding - Simply check the destination, it knows to which port that destination is connected to and send the frame in that port destination is
- Broadcasting - Sending to all destinations, this happens when it doesn't know where to send it and needs to find out.
- Filtering -

In order to avoid problems such as exponential spreading of frames, a **Spanning Tree Protocol** at a logical level is used. Closed loops go away with this protocol since a tree is created and all notes still are connected since it is a spanning tree. BPDU - Bridge Protocol Data Units are the frames sent to establishing this tree.

Ethernet frames don't have time to live. But in BPDU's have TTL

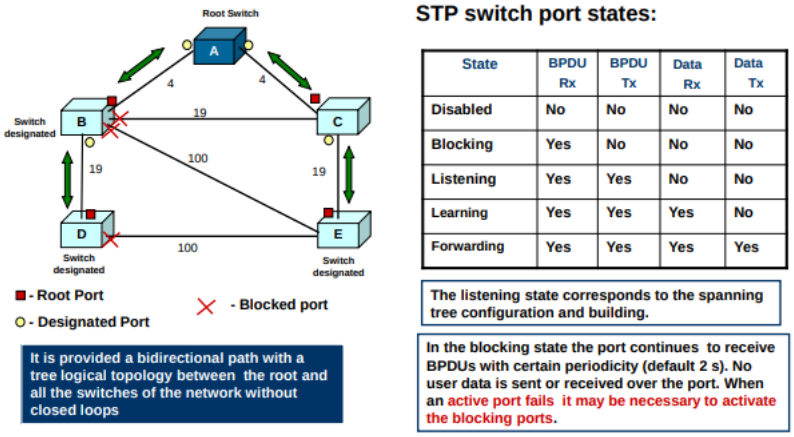
This Spanning Tree Protocol can be implemented in the following way:

1. Root Switch election - typically the one that has the smallest MAC in the network because computing the one that has the smallest distance to every other node is too troublesome. Therefore, after receiving a BPDU with an ID lower than the current root, it replaces his belief of root and propagates that information.
2. Convergence to Spanning tree starting at the Root - The root propagates BPDU's with a cost to root (if the root is sending, the cost is 0). The nodes that receive do the same: **(root id, my id, cost to root)** . - If the received packet tells a cost to root that is smaller than before, that port is designated as the root port.

   - If the received packet tells a cost that is bigger, than that port will be blocked: not packets should be sent that way!

   - The ports that are not blocked and that will connect the lower nodes to the root will be called **designated ports** . These are the ports that node can send things to.

This is an example of the status of each



STP switch port states:

State	BPDU Rx	BPDU Tx	Data Rx	Data Tx
Disabled	No	No	No	No
Blocking	Yes	No	No	No
Listening	Yes	Yes	No	No
Learning	Yes	Yes	Yes	No
Forwarding	Yes	Yes	Yes	Yes

The listening state corresponds to the spanning tree configuration and building.

In the blocking state the port continues to receive BPDUs with certain periodicity (default 2 s). No user data is sent or received over the port. When an active port fails it may be necessary to activate the blocking ports.

How is the switch able to build the Source Address Table (SAT)

It can be filled by hand, by the network administrator. Or it can learn the MAC that communication to each port.

How packets Travel in the network We write a message in the phone or in the laptop and press send in the application (layer). First, there is some compression and encryption done by the application layer right away. Then in the application code, there is a section that is defining TCP or UDP sockets - very likely TCP -

How to build minimum spanning trees? They are needed for the spanning tree algorithms like the Link Layer LAN minimum spanning tree algorithm.

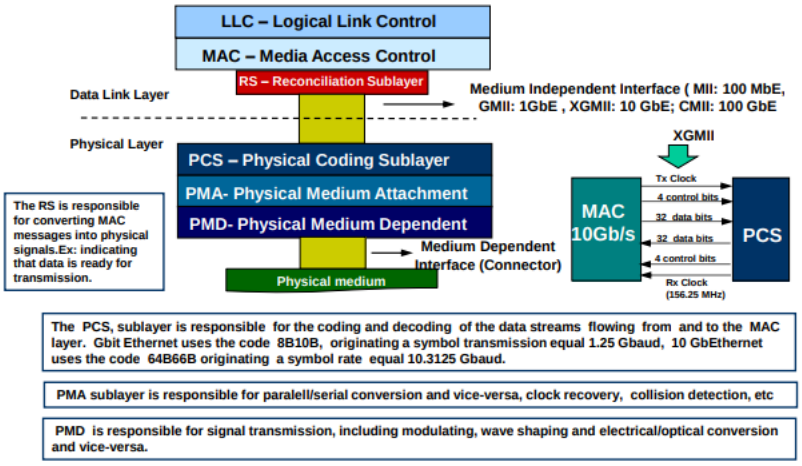
With the Kruskal's Algorithm:

1. order the edges by their cost, minimum first.
2. keep adding edges to a the spanning tree, as long as the edge doesn't create a loop (connects two nodes already connected in another way).
3. stop when the tree has **n - 1** edges.

6.3.2 Physical Layer of the Ethernet

The Ethernet protocol includes physical specifications.

Between the link layer and the physical layer (the layer right before the physical medium) there's a **medium independent interface (MII)** . In the picture below, depending on the debits being transmitted, this interface has several names. It has different names because it is a parallel interface that depends on the debit.



The first sublayer of the physical layer is coding. It will had signals to the bits. Each bit will

The second sublayer is responsible for the parallel/series conversion.

The last sublayer must generate the signal into the medium being optic fibre, twister pair, etc...

In essence:

10GMII (10-*Gigabit Media Independent Interface*) : standardized interface between the MAC layer and the physical layer. Permits that the MAC layer can work with different implementations of the physical layer.

PCS (*Physical Coding Sublayer*): It is responsive for coding/decoding the information originated/destined to the MAC layer. There are many types of codes that can be implemented : 64B66B, 8B/10B, PAM-5, etc. O PAM-5 is a multilevel code used in the 1000-Base T standard.

PMA (*Physical Medium Attachment*): It is responsible for the serial/parallel conversion and vice-versa. The clock synchronization is also carried out by this sub-layer

PMD (*Physical Medium Dependent*): It is responsible for the signal transmission. There are different PMD devices according with the medium.

MDI (*Medium Dependent Interface*): Indicates the type of connector used.

Code - mBnB

Firstly, the code. Required for robustness of transmission and to send symbols at lower frequencies than the actual bitrate.

m input, n output bits

By switching between Mode 1 and Mode 2, the number of 0's is identical to the number of 1's, which is important to keep the stochastic balance between 1's and 0's - important for detection.

3B4B Code

Input bits	Mode 1	Mode 2
000	0010	1101
001	0011	0011
010	0101	0101
011	0110	0110
100	1001	1001
101	1010	1010
110	1100	1100
111	1011	0100

After the coding, a 1Gbit/s bit rate is converted into a $\frac{n}{m} * Rate$. For 3B4B, it will be necessary to transmit 4 bits in the time it took to transmit 3. Before, the rate per bit would be $\frac{1Gbit/s}{3}$. And this needs to be multiplied by 4 to have the 4 bits going at the same time. Therefore, 1.(3) Gbaud (symbols per second) is the new rate.

The ethernet transmission is done in baseband, no modulation.

Format used in the interfaces: [Value1] Base [Value 2]
Value1 = Bit rate for the transmission : Ex: 100 ⟹ 100 Mb/s
Base = *Baseband Mode*: Base band transmission (no modulation)
Value 2: Type of cable (T: *Twisted Pair*, F, X, R: Optical Fibre)

Examples:

10 Mbit/s Ethernet

10BaseT: Uses UTP twisted pairs of category 3 ou 5; max distance = 100 m
10Base F: Uses multimodal optical fibre

100 Mbit/s Ethernet (PCS: 4B5B)

100BaseT: Uses UTP twisted pairs of category 5; max distance = 100 m
100Base FX: Uses multimode optical fibre (62.5 µm); max distance = 2000 m

Gigabit Ethernet (PCS: 8B10B)

1000BaseT: Uses UTP twisted pairs cat. 5e; max distance = 100 m
1000Base SX: Uses multimode optical fibre (62.5 µm); max distance = 275 m
1000Base Fx; Uses single mode optical fibre; max distance = 5000 m

For the correct names Ethernet phy layer .

Avoiding Crosstalk

Electromagnetic interference becomes a limiting problem in very high frequencies. To reduce these interferences, a metal sheet foil can be used. A more effective solution is really to use a shield Therefore, shielding is necessary for twisted pairs.

The necessary bandwidth to transmit 10Gbaud/s.

1 baud can have several bits of significance, therefore there is a strong leverage in the coding that is used.

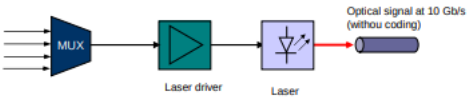
COMPLEEEEEEEEEEEEEEEEEEEEEEEEEEEEEEETE THIIIIIIIIIIIIIIIIIS

Category 7 (CAT7) : Bandwidth equal to 600 MHz((@ 100m). Supports bit rates up to 10Gbit/s (10 GbE), using a cable with 4 STP pairs.

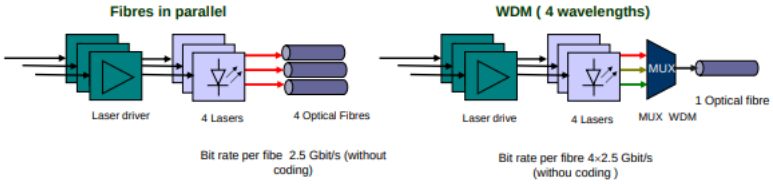
In order to multiplex everything into the physical medium, there are two ways of doing it, normally the second one is used.

- **The serial solution is based on using a single optical channel at 10 Gbit/s.**

The solution LAN PHY uses a bit rate of 10 Gb/s. In the solution WAN-PHY the coded signal (64/66B) must be compatible with the STM-64 (9.953 Gb/s). To make a STM-64 frame one uses an additional sublayer betweem the PCS and PMA designated as WIN (WAN interface sub-layer).



- **In the parallel solution there are multiple channels that can be implemented, using different optical fibres or different wavelengths (WDM).**



The different interfaces for 10 GbE are the following:

Interface	Medium	PCS	Distance	Source
10GBase-SR	Multimode fibre (1 pair)	64B/66B	100-300 m	VCSEL, ou FP laser (850 nm)
10GBase-LR	Monomode fibre 1 pair)	64B/66B	10 km	VCSEL, FP (1310 nm)
10GBase-ER	Monomode fibre I (1 pair)	64B/66B	40 km	Laser DFB (1510 nm)
10GBase-CX4	UTP twisted pair CAT 5 (4 pairs)	8B/10B (PAM)	15 m	Standard 10GBASE-X uses the code 8B/10B
10GBase-T	UTP CAT 6A or STP CAT 7 (4 pairs)	64B/66B (10-PAM)	100 m	4 pairs × 833Mbaud × 3 bits/Baud = 10Gb/s

Gigabit + Ethernet

When the speeds are too big, either the distances need to be smaller or fibres must be used. Single mode avoids dispersion and the pulse is able to keep is form for longer.

6.3.3 Virtual LAN

The time it would take to implement the Spanning Tree Protocol in very big networks makes it infeasible. Therefore, the same physical grid should be separated in many logical grids and the spanning tree only covers one VLAN.

VLANs also allows for traffic separation in triple play services, allow for cloud computing and storage separation. Is possible to separate departments, buildings, etc...

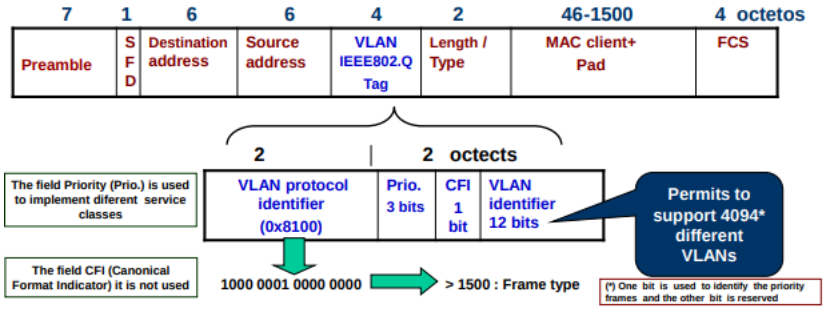
Additionally, it improves the network security and privacy since the traffic inside a VLAN doesn't need to come outside. And facilitates network management since the administrator is able to organize user groups independently of the physical topology of the network. It is the network administrator that defines the limits/boundaries of the VLAN.

The communication between VLANs is done at Layer 3.

There is 3 types of VLANs implementations:

- Layer 1 VLAN (port-based VLAN): the frames don't need to be tagged and the switch distinguish VLANs based on the ports it come from.
- Layer 2 VLAN (MAC Address-based VLAN): the frames need to have VLAN specific tag.
- Layer 3 VLAN (Network Address-based VLAN): The information of VLAN includes IP addresses, requires routers.

Here is the tag:



Note the first field of the tag is bigger than 1500. This happens because the switch is expecting to see a Length/Type field. This way, the switch reads it as a type that is a VLAN type and it knows how to treat the frame - that can be a BPDU or any other kind of frame - based on that identifier.

There is 2 octects to identify the VLANs (4094 because 2 are reserved). At some point, it isn't enough. What is done is multiple tagging: **simply by adding another tag, a VLAN inside a VLAN is created** . However, VLAN IEEE802.1Q is not very clear on how to treat such encapsulations, thus IEEE802.1ad Provider Bridges is used. The tag is called a Service Provider Tag and requires a Provider Bridges aware switch.

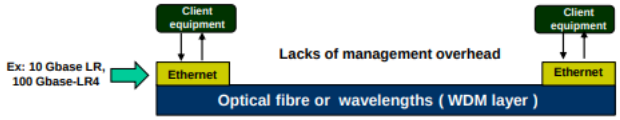
A switch can be aware to simple untagged frames, can be aware of VLANs and can be aware of Provider Bridges.

However, with these 2 tags, tables with 16 Million entrances is already too big... **There must be another way of scale the network properly and way of connecting PB switches at long distances.**

Solution 1- Ethernet doesn't have regenerators, therefore the ethernet frame can be place inside a transport network frame. Using SDH or OTN there's regeneration and there can be links with more than 40 km (the maximum length without regeneration).

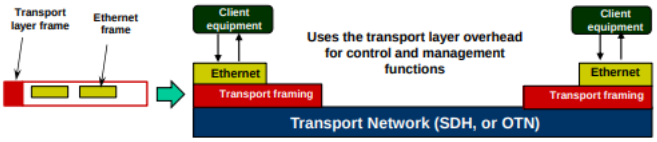
- **Direct Ethernet Transport**

  Puts native Ethernet frames directly on optical fibres or wavelengths (if one uses WDM) using line codes such as 64B66B for 10 GbE (distances < 100km)



- **Ethernet over Transport Network (Ethernet over SDH, Ethernet over OTN, etc.)**

  The Ethernet frames are encapsulated in other transport frames such as SDH, or OTN frames (distances: hundreds or thousands of kms).



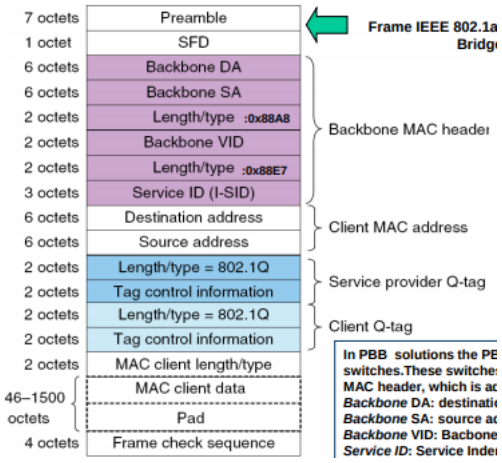
Carrier Ethernet is the idea of using Ethernet in all of the network.

PB switches are barely capable of handling Metro, can't be used in the core. PBB switches are the ones used in the core and they only process their own headers.

However, this is quite far from Ethernet Networks and requires many more functions to compete with the transport network

The standard of the transport network is recovering links after a failure in less than 50 ms. OAM functions and much more is needed to be implemented in Ethernet protocols in order to provide reliable traffic.

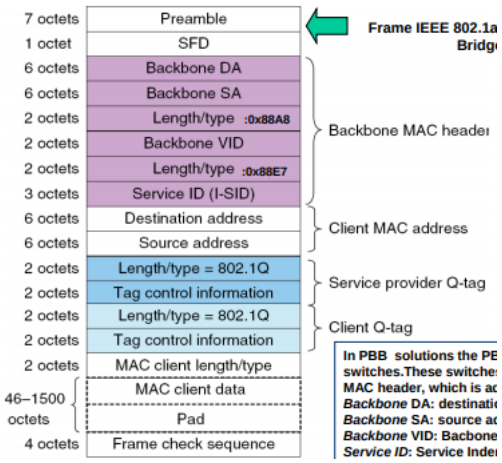
But, there's a huge Ethernet frame...

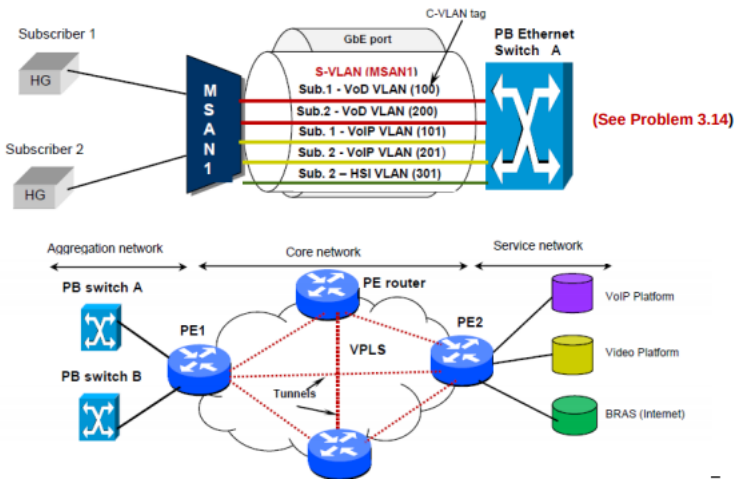


And they then it still uses spanning tree protocols? Non deterministic networks in the core? This would lead to big delays every time there's a change in a link

It is necessary to establish hard links through the management plane.

PBB-TE is a variation of PBB that doesn't use the spanning tree protocol.





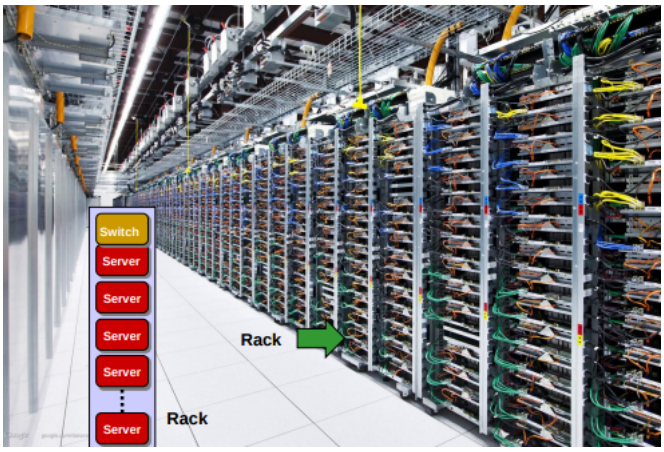
PBB-TE is a technology that can be used in the transport network. It uses only Ethernet but has a lot of overhead to

6.3.4 Data Centres

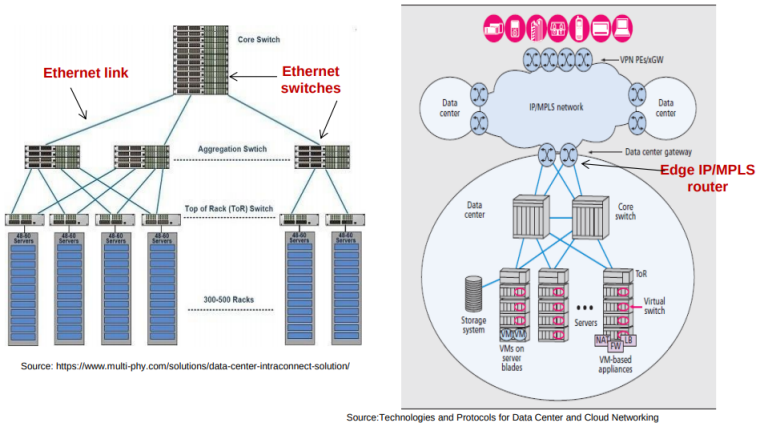
Big Ethernet Networks where an organization stores, manages and disseminates data. 10 and 100 GbE is used. VLAN aware, PB and PBB are possible to be found in Data Centres. Then there are Servers and Routers (IP and MPLS).

The connection between datacentres is called inter-datacentre connection and sometimes the data-rate needs to be so big that

There a switch on top of each rack to connect all of the servers.

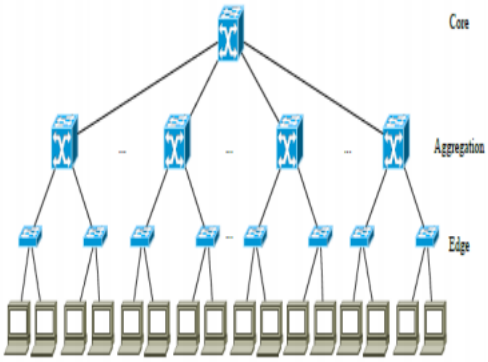


Top of Rack (ToR) switch, are connected to aggregation switches connected to just a couple of central switches and these are connected to Edge IP/MPLS routers that connect the IP/MPLS network that is on top of the global transport network.



For instances, to use a cloud based service a virtual machine must be created in one of those servers.

A tree physical topology despite being easy to implement provides no fault tolerance, leads to congestion (no load balancing) and doesn't scale properly since the maximum data rate the central node can handle will limit the size of the tree.



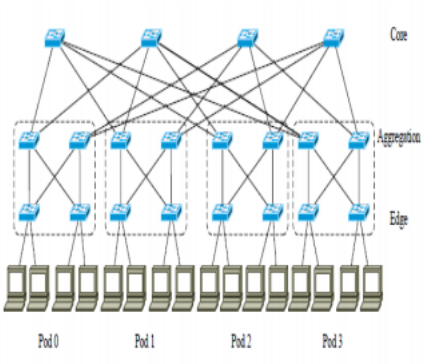
Instead, a Fat Tree is used.

Identical bandwith at any bisection

Each layer has the same agregated bandwith

Advantages:
- **Redundancy**

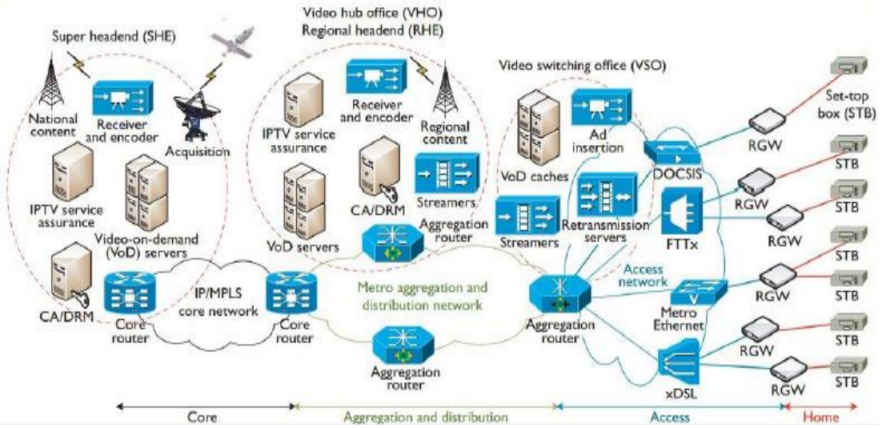
Disadvantages:
- **Wiring complexity in large networks**
- Problems of scalability



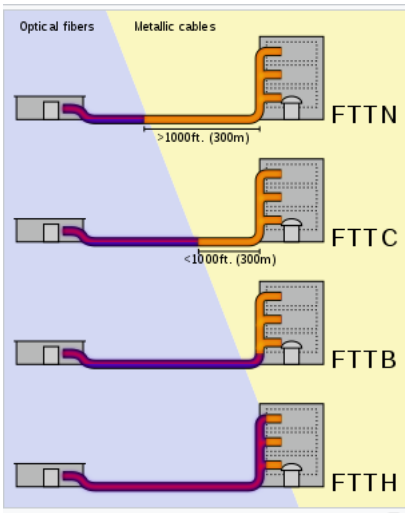
Note that:

- Each ToR switch is connected to 2 aggregation switches;
- Each aggregation switch should be connected to more than one core/central switch to achieve redundancy and flexible load balancing;
- It usually comes 40Gbps from the rack but the switch only outputs 20Gbps, this is quite normal since the traffic is bursty. Overprovisioning would be to have the same output as the input.

A general overview of IPTV network:



DOCSIS Modem is used for HFC, there's FTTx (Fibre to the x, where x can Node, Curb, Building or Home, depending on how close the fibre is terminated) and xDSL (x Digital Subscriber Line).



A schematic illustrating how FTTX (Node, Curb, Building, Home) architectures vary with regard to the

The final is node in our homes in the set-top box (STB). Contains a TV-tuner input, display output for the television. Also contains, external source signal to receive the data from a cable, satellite or even other ways.

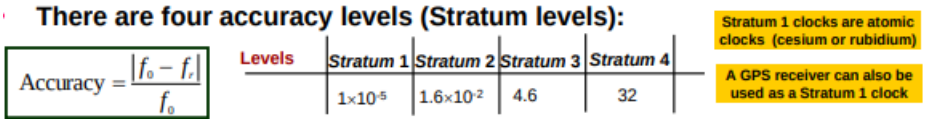
6.4 SDH - Synchronous Digital Hierarchy

SDH is not used very much, OTN is the current standard. MPLS-TP is yet another way of doing transport, using MPLS with Transport Profile.

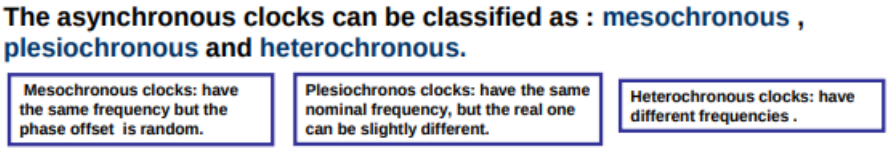
To have deterministic links, the Management plane has to establish them.

There are multiple ways of doing multiplexing. In SDH (and in OTN), TDM is used.

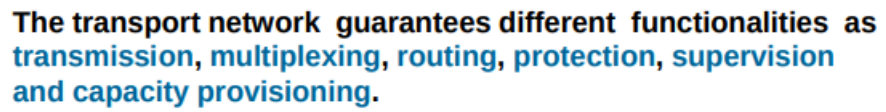
To transmit our signal in SDH using TDM, it will have a slight overhead with the Start Frame Delimiter (SDM). There are many flavours of TDM. However, in a deterministic network, the timeslots are assigned in a fixed way. Therefore, even if one of the input channels doesn't have information to transmit, there will be a slot assigned to it and will be empty.



Two clocks are synchronous if they operate with the same frequency and a constant phase offset ($\triangle\phi(t) = const.$). Moreover, they can be classified with other words regarding the similarities between frequencies and phases.

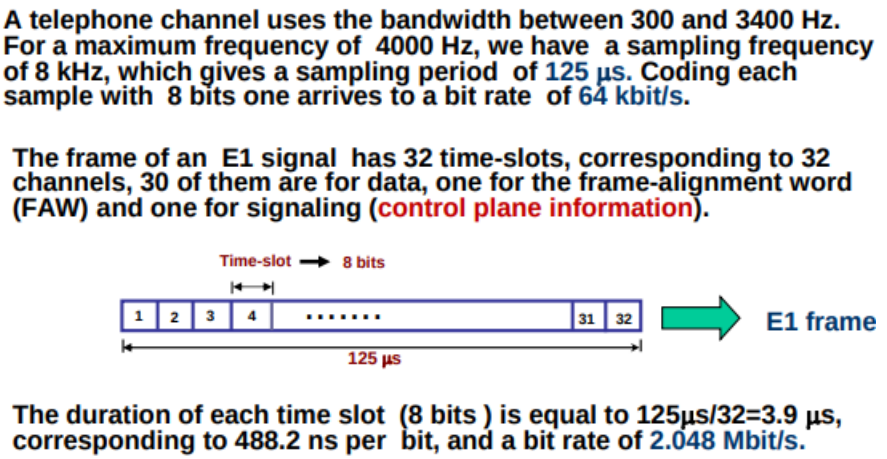


Regarding the accuracy of the frequency of the real clock (f_r), expressing in Parts Per Million (ppm):



A curiosity: nowadays is possible to find atomic clocks for less than 5k euros. Some as low as 1500 euros.

Because the sampling rate should be at least twice the maximum frequency of the signal to sample it properly (according to the Nyquist Theorem). Therefore:

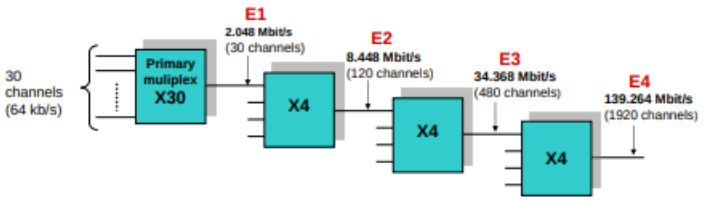


Recalling the meaning of Synchronous and Plesiochronous, now is possible to understand what are the Digital Hierarchies based on these types of synchrony. In the first (SDH), both ends are synchronized with a central clock while the second (PDH) doesn't require this synchronization.

In the PDH network, the frequencies are similar and depending on the multiplexing factor, they can drift a certain amount. If the amount of signals being multiplexed grows considerably,

The signals of the various hierarchical levels are denote with E*i* (*i*=1,2,3,4).

The first level of the hierarchy carries 30 telephone channels(64 kbit/s), while the next levels are obtained by multiplying the previous levels by four.



The clocks of the European Hierarchy require the following accuracy:

Hierarchy	E1	E2	E3	E4
Accuracy	50 ppm	30 ppm	20 ppm	15 ppm

For some reasons, the American and the Japanese networks follow slightly different standards, having different amounts of multiplexed channels.

Note also

But these PDH still use TDM? Maybe the drift in frequency is small enough to still allow it.

In SDH, the network operator has to provide the synchronization clock, the **PRC - Primary Reference Clock** .

Note that a synchronization network is required to keep all elements with the same clock. Examples of synchronous networks:

- SDH
- TDT - Terrestrial Digital Television
- GPON - Gigabit Passive Optical Networks is Synchronous as well. (OTN is not)
- Mobile Network
- PSTN

Regarding SDN then, one of the first proposals was called SONET (Synchronous Optical Network). SDN was later defined by ITU-T as an international standard compatible with SONET. TDM frames were used. Some more nomenclature:

The SDH basic signal is named *Synchronous Transport Module* (STM). The SONET basic signal is named *Synchronous Transport Signal* (STS) in the electrical domain and *Optical Carrier* (OC) in the optical domain.

SONET (Optical)	SONET (Electrical)	SDH	Bit Rate (Mb/s)
OC-1	STS-1	-----	51.840
OC-3	STS-3	STM-1	155.520
OC-12	STS-12	STM-4	622.080
OC-48	STS-48	STM-16	2488.320
OC-192	STS-192	STM-64	9953.280
OC-768	STS-768	STM-256	39813.120

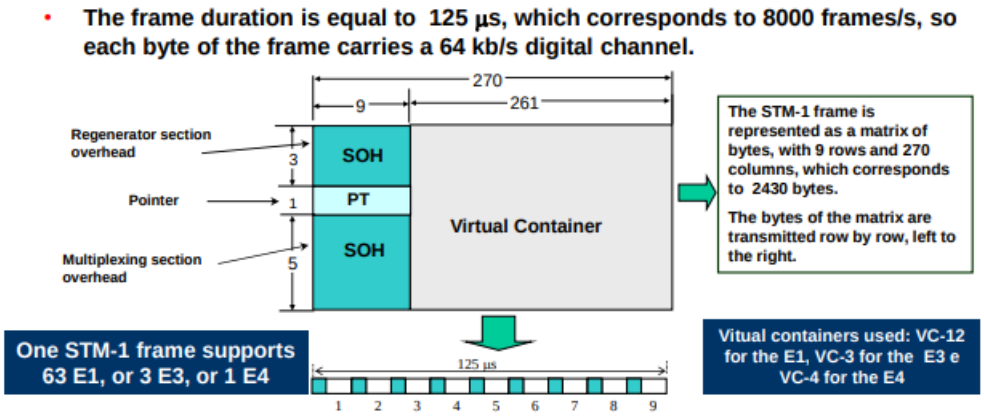
Note : SDH is compatible with PDH E-n signals. Since SDH is able to carry more information in each TDM frame, it can carry PDH hierarchies in its frames.

A SHD basic frame (STM-1) contains 3 blocks:

- SOH - Section OverHead for regenerations and multiplexing

- PT - Pointer to the Virtual Container start.
- VC - Virtual Container with the payload and the

Since this basic frame is sent every 125 micro seconds, 8000 frames per second is the frame rate. Note again that 155 MB/s is the bitrate.



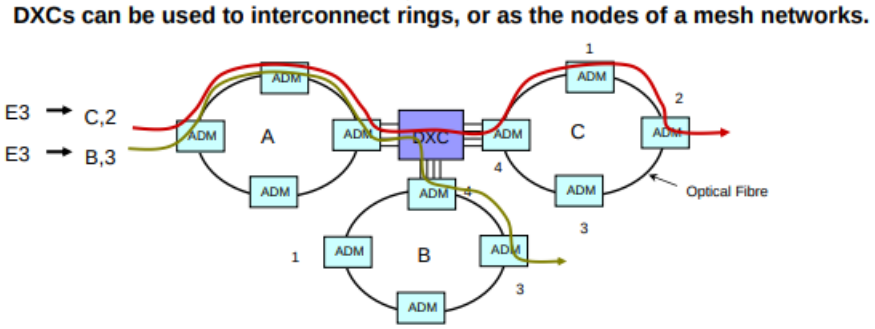
Some equipment that is necessary in this kinds of networks:

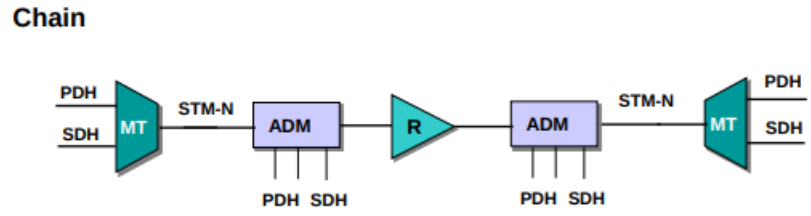
-An ADM (Add and Drop Multiplexer) selects one of the containers in the TDM frame to drop and one of its own to add, in case it is expecting containers or has containers to send, respectively;

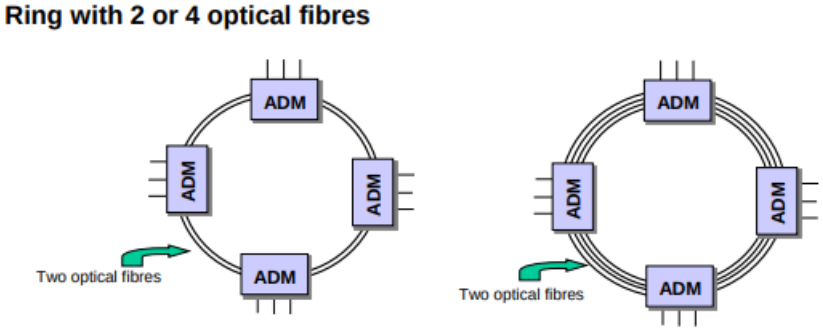
- A regenerator;

- A Terminal Multiplexer (TM) that multiplexes any amount of signals at the entrance to one signal at the output.

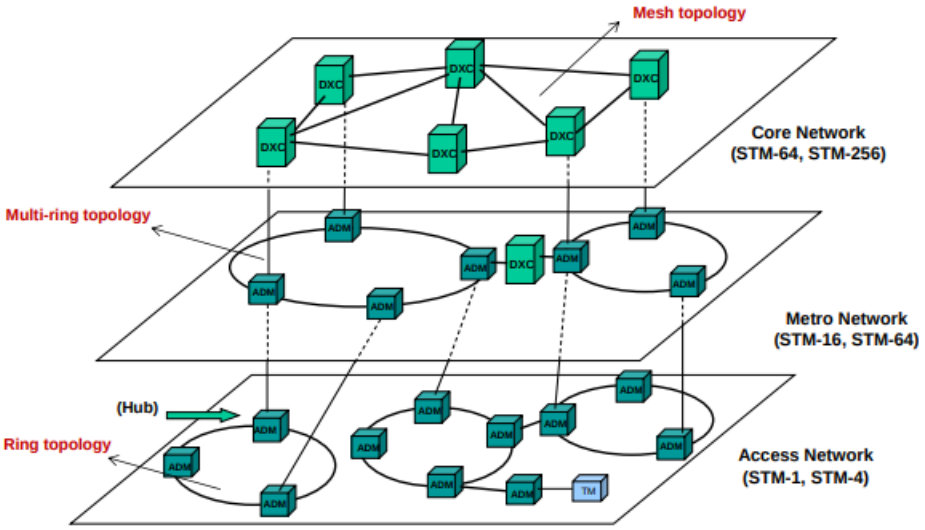
- A Digital Cross-Connect (DXC) is able to connect any input to any output, and is controlled by the management plane to establish the previously mentioned semi-permanent paths.





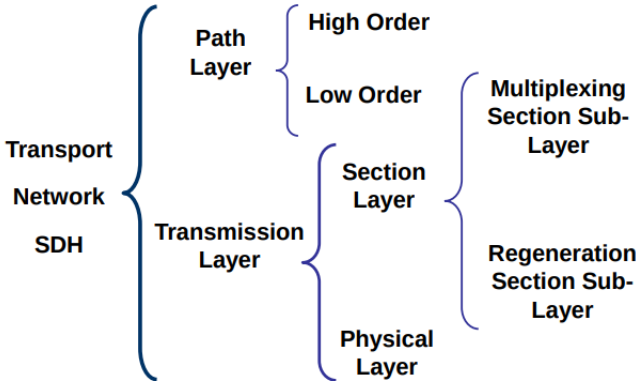


These equipments are used in accordance to the densification we have in the network. Closer to the core there are more DXCs, closer to the access there's more ADMs.



When we mention "containers" we are referring to Ethernet frames or other hierarchies of digital flows.

The transport Network SDH based can be divided in 2 parts, the Path and the Transmission.



This layers have important functionalities. Path layer functionalities are related with the connections, the paths established. High and Low order .....

These functions are why it is worth it to have extra overhead in all layers:

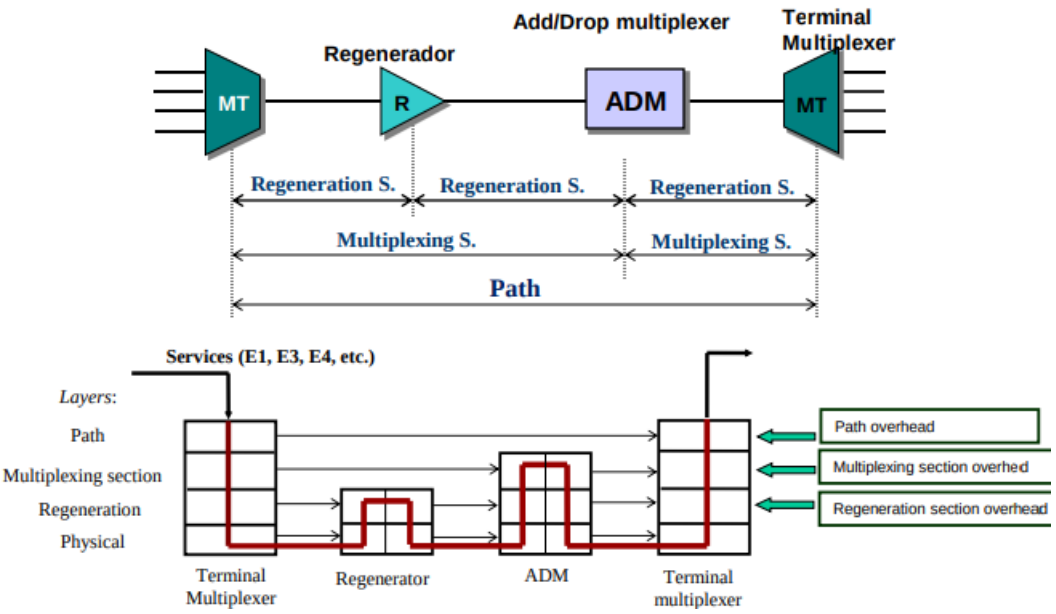
Path :
Connection integrity verification, defines the type of traffic transported in the path, error monitoring, protection switching.

Multiplexing section:
Clock synchronization, protection switching, error monitoring, communication with the network management system.

Regeneration section:
Frame alignment signal, error monitoring, communication with the network management system.

Physical:
Optical pulses shape, power level, wavelength, receiver sensitivity, etc.

Note that these layers are similar to the OSI model layers, each of them has an overhead related to its *modus operandi* , exactly like there are IP headers and MAC headers:

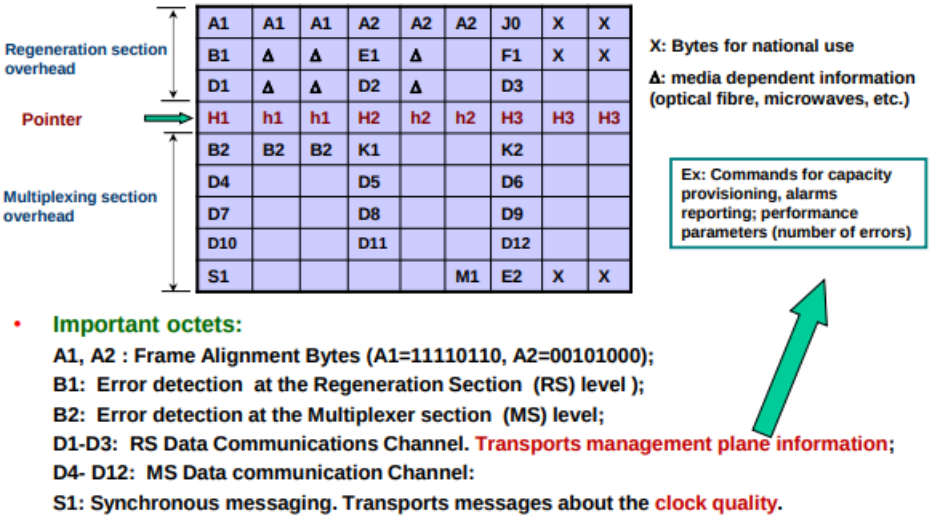


Note the scopes in a digital transmission! All of them do regeneration, but not all of them multiplex. In the multiplexing layer its where the clock synchronization comes in, note that despite being able to carry Plesiochronous DH frames, it this is a synchronous hierarchy, so it needs a PRC (primary reference clock).

To step back a bit:

1. An SDH basic frame (STM-1) contains section overhead (SOH), PT (pointer) to the beginning of the virtual container with the payload data and then the VC.
2. **One STM-1 frame supports 63 E1 frames, 3 E3 or 1 E4**
3. SOH has a regeneration section and a multiplexing section overheads.
4. A VC is a container with a path overhead: "The STM-1 frame is represented as a matrix of bytes, with 9 rows and 270 columns, which corresponds to 2430 bytes. The bytes of the matrix are transmitted row by row, left to the right." - That is why there's a 9 segment separation over the transmission interval, because the frame has 9 rows and each row is transmitted

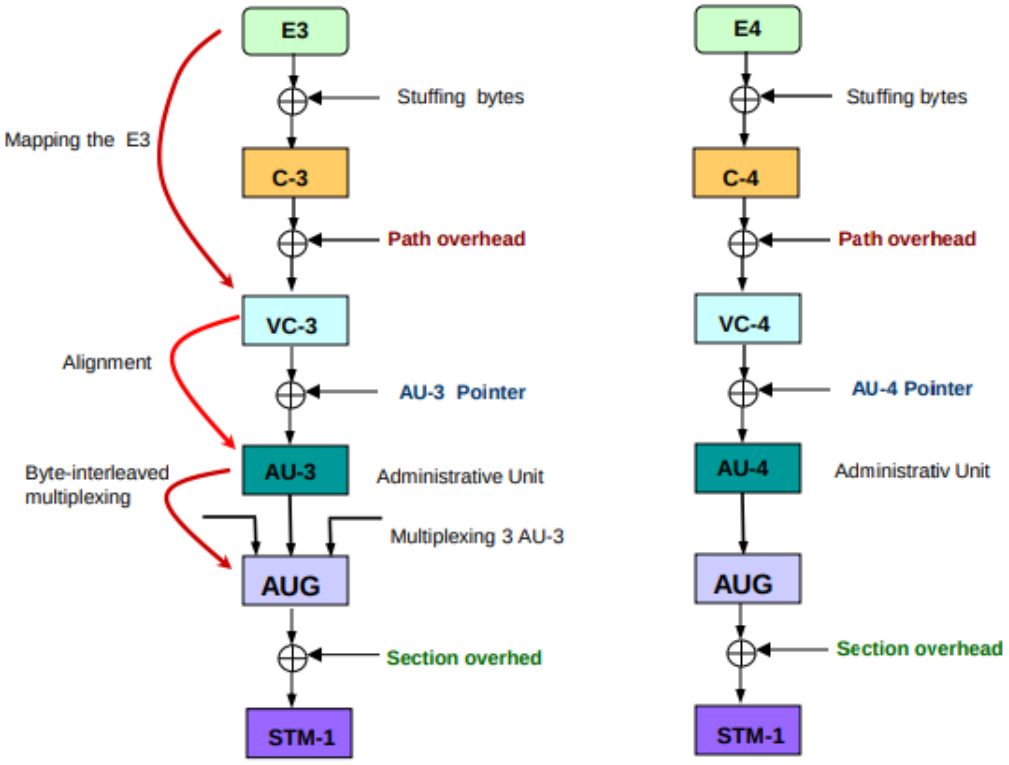
5. In a VC there may be several containers + path-overhead
6. There it takes 125 microseconds to transmit a complete frame, therefore 8000 frames can be transmitted per second.
7. One payload byte corresponds to 64 kb/s because 8000 of those bytes will be sent per second. Therefore, to know the bitrate of a certain portion of the frame we just need to count the number of bytes and multiply it by 64kbps.
8. The pointer part of the section points to all container beginnings in the virtual container?
9. The Frame Alignment Signal sinalizes the beginning of the frame. Despite
10. The SOH format is:



Note that there are 3 bytes in a frame to transmit Management plane information regarding the Regeneration. D4-12 are the correspondent for Multiplexing. B's are for error corrections for both those planes.

11. There are multiple Virtual containers sizes, all 9 rows long:
    - VC-4 - 261 columns long (1 path overhead, 260 C4 payload)
    - VC-3 - 86 columns long (1 path overhead, 85 C3 payload)

    Note: In a Virtual Container is possible to carry 1 VC-4 or 3 VC-3s
12. A VC-4 with a pointer is called an AU-4. A VC-3 with a pointer is called an AU-3. A frame may have 1 AU-4 or 3 AU-3.
13. The VC-4 can fluctuate inside the AU-4, it doesn't need to start in the beginning. The same of VC-3. If it doesn't start in the beginning, it can simply continue to other frames.
14. Overall:



AUG aggregates all AUs. If AU-4 is used, then AUG is AU-4, else, it includes more than one AU.

Management plane uses a DCC (Data Communication Channel) which has bytes in the frame overhead, more specifically, D1 to D12 bytes.

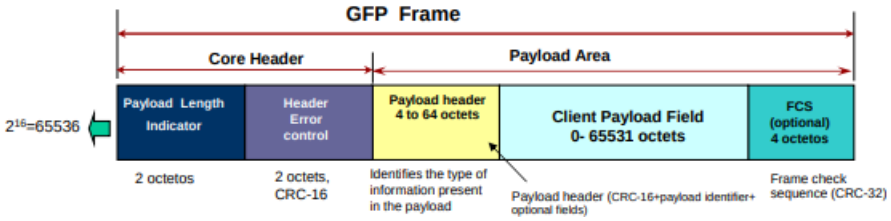
Control plane uses B types of bytes.

However, S type of bytes are of the responsibility of the data plane since synchronism is nothing more than expecting bits at certain times.

GFP

Generic Framic Procedure (GFP) is an ITU-T standard on how to map packet traffic like Ethernet and SAN (see ahead), which is of bursty nature, into SDH and OTN with a constant bit rate .

GFP client frames:



Other type of GFP frames are GFP control frames, where only the header is sent.

Storage Area Networks

A very redundant server and disk access network, stores information for a big quantity of data. Each server is connected to more than one switch and each switch aggregates more than 2 disk/storage arrays.

One of the most common SAN protocols are Fibre Channels that range from 1 to 128 Gbit per second.

Virtual Concatenation

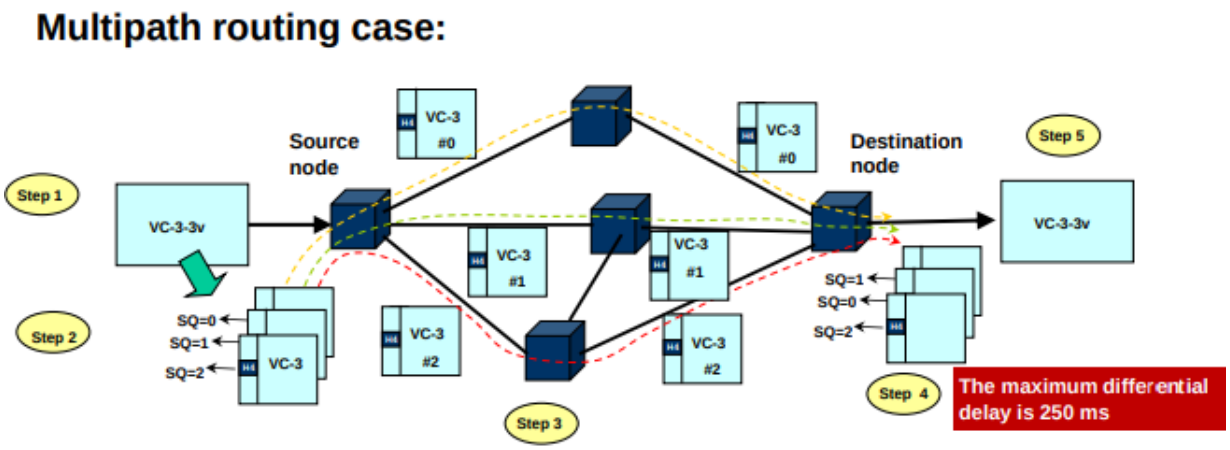
How is SDH able to cope with 1GbE? It uses virtual concatenation: creates **X** output flows with STM-1 frames. This is called inverse multiplexing.

Containers	Type	Available capacity (Mb/s)
VC-11-Xv	Low order	X x 1.600 (X=1,..,64)
VC-12-Xv	Low order	X x 2.176 (X=1,..,64)
VC-3-Xv	High order	X x 48.384 (X=1,..,256)
VC-4-Xv	High order	X x 149.76 (X=1,..,256)

Therefore, with $X = 256$, is possible to achieve $256x150Mbit/s = 38.4Gbit/s$.

This technique can also be used to do balancing in the network since it implements multipath routing on the physical level since the X SDH flows may not do the same path.

There's a field called H4 that has the sequence number of these containers. 8 bits are able to index 256 containers, that is why X is limiter to 256.



Step 1: The source node maps the traffic to be transported in to local memory to form a continuous SDH signal.

Step 2: This is then allocated into the different virtual containers that belong to the same VCG, which are identified by the sequence indicator SQ. This indicator is inside of the path overhead of each virtual container.

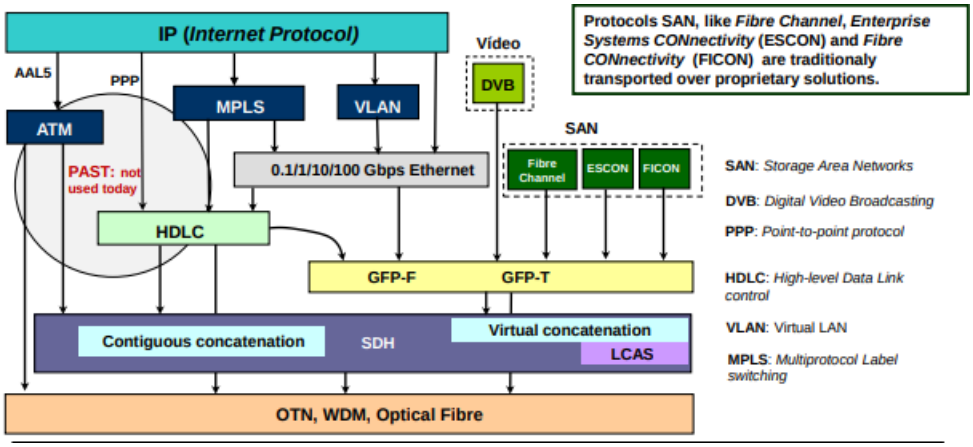
Step 3: The different virtual containers are then transported individually across the SDH network , possibly following different paths, which leads to different propagation times (differential delay).

Step 4: The individual virtual containers are received in buffered memory at the destination node in order to compensate the differential delay.

Step 5: The containers are realigned and then combined back in order to form the initial flow.

IP over SDH / OTN / WDM

Note that in order to use SDH, a synchronous MPLS is one of the important ones. But be aware that MPLS, SDH and OTN are in different planes!



In summary:

- MPLS, VLANs and directly IP use Ethernet which needs GFP to convert this non constant bit rate traffic to the constant bit rate of SDH;
- SAN and DVB are also supported by GFP (GFP-F and GFP-T) As a curiosity:
    - *Framed GFP* (GFP-F) is optimized for bandwidth efficiency at the expense of latency. It encapsulates complete Ethernet (or other types of) frames with a GFP header.
    - *Transparent GFP* (GFP-T) is used for low latency transport of block-coded client signals such as Gigabit Ethernet, Fibre Channel, ESCON, FiCON, and Digital Video Broadcast (DVB). In this mode, small groups of 8B/10B symbols are transmitted rather than waiting for a complete frame of data.

- DVB transmits without compression? FHD = 1920 x 1080 = 2073600 QHD = 2560 x 1440 = 3686400 UHD = 3840 x 2160 = 8294400 (4 times as much as FHD) If we have 24 bit color depth, 8 bits per color, we would uncompressed bit rates above 1Gbit/s for FHD at 30 fps. With compression we get around 10 Mbit/s.

  DVB includes Satellite, Cable, Microwave Links portabilities.

Generalization of Network Element

MultiService Provisioning Platform

- *Framed GFP* (GFP-F) is optimized for bandwidth efficiency at the expense of latency. It encapsulates complete Ethernet (or other types of) frames with a GFP header.
- *Transparent GFP* (GFP-T) is used for low latency transport of block-coded client signals such as Gigabit Ethernet, Fibre Channel, ESCON, FiCON, and Digital Video Broadcast (DVB). In this mode, small groups of 8B/10B symbols are transmitted rather than waiting for a complete frame of data.

7 Artificial Intelligence

7.1 Basic Problems & Nomenclature

There are some goal states and one initial state. The objective is to find the goal state that is closer (with the shortest path to the root/ with the least search cost). Is given as the solution the steps necessary to perform that path.

To keep the formulation as general as possible, abstractions are required, keeping in mind that they will need a correspondence when applied to a real problem.

Example of the application to a problem: a vacuum cleaner that needs to clean every square. In this case, only 2 squares are presented, no localization sensors are present, only "rubbish" sensors that tell if the current square is dirty or not.

Therefore:

- States : $< r, d_1, d_2 >$ where r is the robot position, d_1 and d_2 are binary, representing the existence of dirt in each of the rooms.
- Operators/Actions : L (go left), R (go right) or S (suck dirt)
- Goal Test : d_1 = False and d_2 = False. $((d_1 \text{ nor } d_2) == 1)$
- Initial State : $< r, d_1, d_2 >=< 1, T, T >$ is at square 1, and squares 1 and 2 are dirty
- Step Cost : the description of each action cost. In this case, 1 for each one.

8-puzzle

States: specify in a 3x3 matrix the location of the 8 numbered tiles plus the blank one

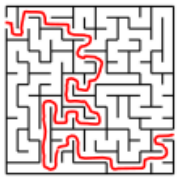
Initial state: any puzzle configuration

Operators: move "blank" to the left, right, up and down

Goal test: checks whether the state matches the goal configuration

Path cost: each step costs 1 (so path cost = number of steps)

Mazes



States: maze configuration plus current location

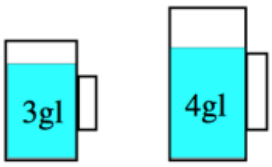
Initial state: any maze location

Operators: move to an adjacent and linked position

Goal test: current location = maze exit

Path cost: number of steps

Water jugs



States: amount of water in each jug (e.g., <1,4>)

Initial state: <0,0>

Operators: fill J4; fill J3;
empty J4; empty J3;
pour water from J4 into J3 until J3 is full or J4 is empty;
pour water from J3 into J4 until J4 is full or J3 is empty;

Goal test: <0,2>

Path cost: amount of water used

Hanoi towers

States: location of the disks in the three poles

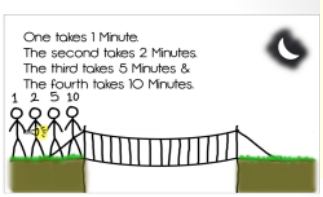
Initial state: all disks correctly arrange in the left pole

Operators: move a free disk to another pole, which is empty or all disks in it are bigger

Goal test: all disks correctly arrange in the right pole

Path cost: number of disk moves

4 Men and the Bridge



States: location of the men and the flashlight

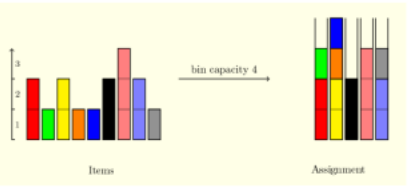
Initial state: all on one side

Operators: move one or two men, with the flashlight, to the other side

Goal test: all on the other side

Path cost: number of minutes needed

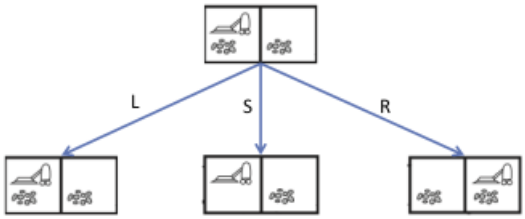
Bin packing problem



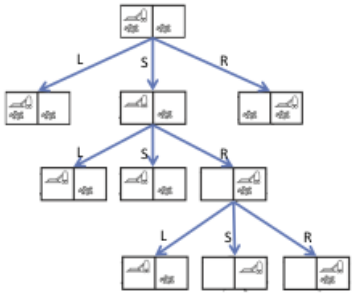
Bin packing

As you can see, every problem is very well defined in terms of the start, the end, the possible moves and what makes a solution better than other. Therefore, the conditions to put the computer thinking how to get from the possible steps to the best solution are assembled. One way the computer should be able to get to the solution is by exploring every move combination and check the state it ended up with.

Given the initial state, use the operators to generate successor states.



Then a choice of which is more "profitable" or more likely to lead to a goal state needs to be made and this cycle continues until the arrival at a goal state.



In this case, the solution would be $\{S, R, S\}$.

However, before diving directly in algorithms, is important to note the characteristics of the Environment and of the Agent. Based on these we'll be able to perform much better choosing the search algorithm that is best for us.

7.2 Environment

Observability: how much can the agent know about the environment.

An environment is fully observable if the agent can see everything. Or partial observable: - part of the state is occulted and you simply can't

Deterministic: the effect of the actions are predictable If I move one queen from one place to the other, I can predict the effects, what is attacking what, etc... Instead of Deterministic, it can be stochastic, where the outcomes are functions of probability functions. Therefore, you can't be 100% sure of the outcome. Therefore, you can use probability to make choices...

Neither the environment nor the agent performance change while the agent is deliberating. –¿ Static.

In a dynamic environment, the agent performance can change with time.

Semi-dynamic means that the world is static, but the performance is changing. ex: turn game where the game doesn't change while you are thinking but the more you take, the less point you get.

Continuous or discrete

Sequential or episodic. Episodic means that one episode doesn't influence the next one. It's very related with causality. In episodic environments, there's no influence in consequent problems.

E.g. one game of chess is sequential, but different games are episodic.

The outcomes for all actions are given. –¿ Known environment.

In a case where the less time you spend thinking before answering,

Knowing these is very important because it allows us to best choose the shelf of methods from which we take our algorithms from. Some are best from some things and some can only be used in certain situations as well.

Internally to the agent, you have a way of representing of the world. (Internal representation)

Example when you don't have an internal representation: random vacuum cleaning robots don't know anything about the environment, they just rotate randomly when they see an object.

- Fully observable vs. partially observable
  (partially observable because of noise, inaccurate or faulty sensors, or hidden parts)
- Single agent vs. multiagent (cooperative vs. competitive)
  (in multiagent environments, communication may be a key issue)
- Deterministic vs. stochastic
  (an environment is uncertain if it is not fully observable or not deterministic)
  (nondeterministic environment is when actions have different possible outcomes but no probabilities associated)
- Episodic vs. sequential
- Static vs. dynamic
  (if the environment doesn't change with the time but agent's performance does, it is **semidynamic**)
- Discrete vs. continuous (applies to states, time, percepts, and actions)
- Know vs. unknown
  (depends on the agent's knowledge about how the world evolves – the "laws of physics" of the environment)

	observable	deterministic	episodic	static	discrete
Chess	Y	Y	N	Y	Y
Poker	N	N	N	Y	Y
Taxi	N	N	N	N	N
Image analysis	Y	Y	Y	Semi	N

7.3 Agent

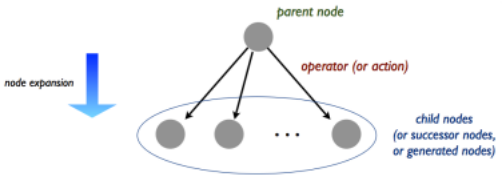
Model-based: typically, when you have partial observability.

Goal-based: when you aim for a goal. You have states and actions and can search for the goal.

Utility-based: are very similar to goal-based but are a bit more, not all goals are the same. There a preference between goals. Utility Theory handles how to translate preferences to numbers.

7.4 Search Problems

In essence, is necessary some **Search Terminology** to refer to certain things:



Successor function: given a node, returns the set of child nodes

Open list (or frontier or fringe): set of nodes not yet expanded

Closed (explored) list: set of nodes already expanded

Leaf node: a node without successors

State space can be:

- Tree-based – no repeated nodes in search
- Graph-based – directed cycle graph

General algorithm shape, starting with $open\_list = \{initial_node\}$, iterate over:

1. Select a node from the open_list;
2. Check if it's a goal node (in case it satisfies the goal test). If yes, return solution by backing up to the root;
3. If not, remote it from the open_list, expand it with the successor function and insert the nodes that come from there to the open_list.

Different selection criteria leads to a variety of search methods.

If it's tree based, then no nodes will be repeated and it's not necessary to have a list of the already visited nodes. In a graph however, is needed to have a list of these, or else a cycle is possible.

To evaluate the algorithm, many parameters may be enumerated:

Completeness: guarantee that a solution is found if there is one

Optimality: the solution found minimizes path cost over all possible solutions

Time complexity: how long does it take to find a solution (usually measured in terms of the number of nodes generated)

Space complexity: how much memory is needed to find a solution (usually measured in terms of the maximum number of nodes stored in memory)

Branching factor (b): maximum number of successors of any node

Depth (d) of the shallowest goal node

m = maximum length of any path in state space (may be infinite)

g(n): the cost of going from the root node to node n (path cost function)

Mentioning types of search strategies:

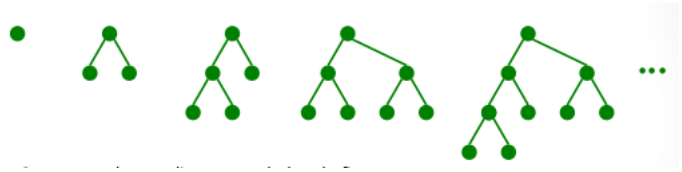
Uninformed (or blind) search: does not use additional (domain-dependent) information about states beyond that provided in the problem definition

Informed (or heuristic) search: uses problem-specific knowledge about the domain to "guide" the search towards more promising paths

7.5 Uniformed Search Strategies

A list of **Uniformed search strategies** we'll have a deeper look to:

- **Breadth-first Search** - Select earliest expanded node first - uses a FIFO queue (First In, First out). This leads to opening every node at the safe depth first before moving the deeper nodes.



Because order of depth is followed and every node checked, this search strategy is Complete and Optimal (if the path cost increases - or at least doesn't decreases - with depth). Being **d** the depth of the solution and **b** the branching factor(max number of successors of a node.) then in the worst case, the total number of nodes generated is: $1 + b + b^2 + b^3 + \cdots + b^d$

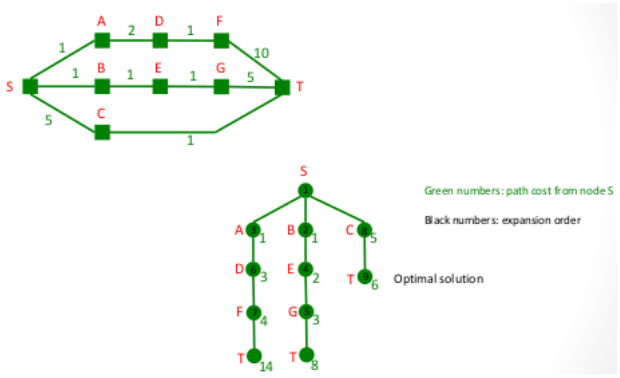
Time Complexity - O(b^d) Space Complexity - O(sum no of nodes) = O(b^d)

Depth	Nodes	Time	Memory
2	110	.11 milliseconds	107 kilobytes
4	11,110	11 milliseconds	10.6 megabytes
6	10^6	1.1 seconds	1 gigabyte
8	10^8	2 minutes	103 gigabytes
10	10^{10}	3 hours	10 terabytes
12	10^{12}	13 days	1 petabyte
14	10^{14}	3.5 years	99 petabytes
16	10^{16}	350 years	10 exabytes

Figure 3.13 Time and memory requirements for breadth-first search. The numbers shown assume branching factor $b = 10$; 1 million nodes/second; 1000 bytes/node.

Note that it makes a difference if you test the node before or after expanding it. If it's tested before, there's no need of expanding the node. If the test is made only after the expansion, the complexities grows to O(b^{d+1}).

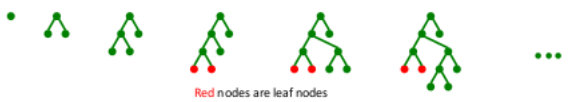
- **Uniform-cost Search** - expands the node that has the smallest cost from the root to it.



Note that each action generates one possible state from the state where the robot was previously.

This search strategy is only complete and optimal if the step costs are strictly positive. Else, it can give several steps that cost nothing to places far away from the solution.

- **Depth-first Search** - Exactly as the name suggests, goes until the deepest node first, and only then looks at the other nodes at the first level. Therefore, it can be very inefficient on a large or infinite tree. Open the last node added to the list (LIFO- Last In, First Out).



- **Backtrack Search** - a variation of depth-first, but expands one node at a time, only stores in memory that only node and the expansion is made by modifying the node, while backtrack is nothing more than undoing the modification..

  It's not complete nor it is optimal... but saves a lot of memory.

- **Depth-limited search** - another variation of depth-first where limiting the search tree to a depth L, is possible to contain the inefficiency. It's complete if the depth of the solution if smaller than L but still not optimal.

  Time complexity: O(b^L) Space complexity: O($b \times L$)

- **Iterative deepening depth-first search** - A variation of the previous one. In this one, the idea will be to run depth-limited search for an increasing L. Run for L=1, L=2, ...

This way, it is complete and it's optimal (if the path cost is a non-decreasing function of depth).

Time complexity: $O(b^d)$ Space complexity: $O(b \times d)$

Node expansion:

Breadth-first

$$1+b+b^2+b^3+\cdots+b^{d-1}+b^d$$

Iterative deepening

$$(d+1)1+(d)b+(d-1)b^2+(d-2)b^3+\cdots+(2)b^{d-1}+(1)b^d$$

Example: b = 10 and d = 5

breadth-first = 111 111

iterative deepening = 123 456 (+ 11%)

- **Bidirectional Search** - Search both from initial node and from the gold node. Note however that can only be used when a goal node is known and when the parent nodes can be computed given its child (through the sets of available actions). It's complete (if breath-first in both directions) and Optimal, if the step costs are equal.

  Time and space complexity: $O(b^{d/2})$

Example: $b = 10$ and $d = 5$

$b^d = 1\,111\,111$

$2\,b^{d/2} = 2\,222$

A summary of the above analysis:

Criterion	Breadth-first	Uniform-cost	Depth-first	Depth-limited	Iterative deepening	bidirectional
Complete?	✓	✓[1]	×	×	✓	✓[3]
Optimal?	✓[2]	✓	×	×	✓[2]	✓[2,3]
Time	$O(b^d)$	$O(b^{1+\lfloor C^*/\varepsilon \rfloor})$	$O(b^m)$	$O(b^L)$	$O(b^d)$	$O(b^{d/2})$
Space	$O(b^d)$	$O(b^{1+\lfloor C^*/\varepsilon \rfloor})$	$O(b.m)$	$O(b.L)$	$O(b.d)$	$O(b^{d/2})$

[1] for strictly positive step costs
[2] for path costs a non-decreasing function of depth
[3] for breadth-first in both directions

A **General Search Algorithm** can be formulated in the following way:

function General-search (***problem***, ***strategy***) **returns** a solution or failure
insert the root node into the open list
(the root node contains the initial state of problem)
loop do
if there are no candidate nodes for expansion **then return** failure
choose a node for expansion according to strategy (using strategy function)
if the node contains a goal state (using goal checking function) **then**
return the corresponding solution
else
for each operator in the list of operators (or successor function)
create a child node (for the new child state)
update child node path cost (using g-function)
add the resulting node to the open list [*unless… see graph search versions*]
end

domain (problem) independent

problem argument should include at least:

- initial state (using a specific state representation)
- successor function: new state = succ (current state, operator)
- path cost function (g-function)

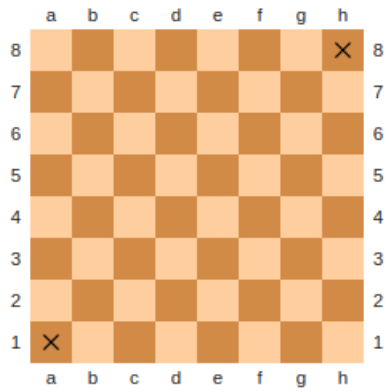
domain (problem) dependent

strategy argument (e.g., FIFO, LIFO, priority queue (by path cost), etc.)

algorithm selection

Finally, there are problem dependent and problem independent details. The search strategies are problem independent, but how we define the actions, states, ect… is problem dependent. And the way we see and think about the problem can be highly related to the way we represent it.

An incredibly well formulated example is the Mutilated Chess Board problem. If we take just the squares of the corners of a chess board, can we still fill the whole board with dominos (each domino takes 2 squares). It becomes slightly harder! If you keep removing squares it gets increasingly harder to figure out by head.



However, if you represent the chess board as the remaining black and white pieces and notice that a domino piece must always cover one black and one white squares, then by taking those two corners then 30 black and 32 white squares will be remaining making it impossible to fill with dominos. As a matter of fact, accordingly with Gomory's Theorem is also possible to say that if 2 square of opposite colours are removed then is always possible to fill the board with dominos! More in: Mutilated chessboard problem

The bottom line is that the representation (problem dependent details) matters.

With this is mind, consider the following example:

Initial state with: bedroom(3), living room(2), kitchen(3), hall(2), truck(0)

a) State: $< pos, n_{bedroom}, n_{living}, n_{kitchen}, n_{hall}, n_{truck} >$

b) Initial State: $< truck, 3, 2, 3, 2, 0 >$

c) m for move, p for push $< m_N, m_S, m_E, m_W, p_N, p_W, p_E, p_W >$. However, is only possible to push in directions where there are rooms (as for walking) and when there are boxes in the current room we are located in.

d) A goal condition could be $n_{truck} = 10$. Another could be the sum of all rooms to be 0. But the first one is more elegant and also seems more general... We don't want to throw boxes out of the window.

Sometimes, just finding the solution is enough. Some times, there's a best solution.

The heuristic (the only difference between informed and uninformed search) is a function of a state that gives us the appreciation we have for that state - how good that state is. If we want the best solution, we must have the heuristic function is key. If you just want a solution, that function can be much simpler or even nonexistent.

7.6 Informed Search Strategies

A problem-solving agent is a goal-based agent that acts on the environment, leading him to go through a series of states in order to achieve the desired goal.

In this course, we only study the single-state problems , meaning that continuous, dynamic, non-deterministic or partially-observable problems are out of scope.

Single-state problem

- **observable (at least the initial state)**
- **deterministic**
- **static**
- **discrete**

Multiple-state problem

- **partially observable (initial state not observable)**
- **deterministic**
- **static**
- **discrete**

The difference between Informed and Uniformed is the heuristic function. In informed search, we know something about the problem that allows us to choose better paths while doing uninformed search the algorithm would simply cover all paths.

As a reminder, the Uniform search is nothing more than opening the one closest to the root first and discarding all repeated nodes with a longer path to them.

The **evaluation function** $f(n)$ that outputs the value accordingly to which the nodes will be expanded first or not include as a component the **heuristic function** $h(n)$, in case of informed search. One good example of an heuristic is the straight line distance. The closer something is in a straight line, probably the closer it is in general.

From now on, the only difference is in the evaluation function!

- Greedy best-first search: uses $f(n) = h(n)$
- A* Search: uses $f(n) = g(n) + h(n)$ where $g(n)$ is the **path cost** function. Exactly identical to Uniform-cost-search but has the additional contribution of $h(n)$

Depending on what we are doing our search on, graphs or trees, there are different requirements on the heuristic function to achieve optimality. If graphs, then **consistency** is required for optimality. If the search is on trees, then **admissibility** is enough for optimality.

But what is Admissibility and Consistency?

Admissibility is never overestimating the cost to reach the goal. For instances, the straight line distance is an optimistic distance and presents a good heuristic.

Consistency is exactly that. The heuristic in n must be smaller than the cost necessary to reach n' from n plus the heuristic in the previous node. In essence, the difference between heuristics in adjacent nodes can't me bigger than the cost of going from one to the other.

$$h(n) \leq c(n, a, n') + h(n') \; .$$

Some light about heuristics:

- h_1 = the number of misplaced tiles. For Figure 3.28, all of the eight tiles are out of position, so the start state would have $h_1 = 8$. h_1 is an admissible heuristic because it is clear that any tile that is out of place must be moved at least once.
- h_2 = the sum of the distances of the tiles from their goal positions. Because tiles cannot move along diagonals, the distance we will count is the sum of the horizontal and vertical distances. This is sometimes called the **city block distance** or **Manhattan distance**. h_2 is also admissible because all any move can do is move one tile one step closer to the goal. Tiles 1 to 8 in the start state give a Manhattan distance of

$$h_2 = 3 + 1 + 2 + 2 + 2 + 3 + 3 + 2 = 18 \; .$$

As expected, neither of these overestimates the true solution cost, which is 26.

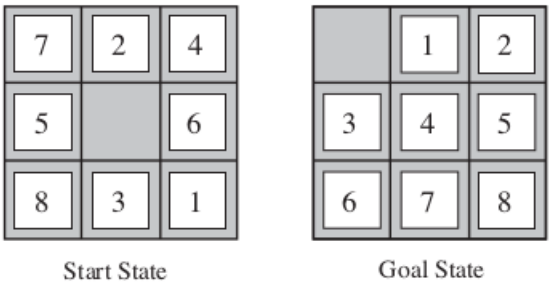


Figure 3.28 A typical instance of the 8-puzzle. The solution is 26 steps long.

Now, obviously, one of the heuristics will give results closer to the reality and the other will underestimate more severely. **Does this difference affects the quality of the heuristic?**

The quality of the heuristic can be assessed by the **effective branching factor** b^* . The **branching factor** of a tree is the maximum number of successors a node has. It is a useful measure to quantify the maximum number of nodes it will be necessary to expand. The effective branching factor is the necessary branching factor a uniform tree would have to have to expand as many nodes as were expanded in our tree.

Therefore, because A* does a better job (due to the heuristic), the solution will be a smaller branching factor than the uniform tree.

$$N + 1 = 1 + b^* + (b^*)^2 + \cdots + (b^*)^d$$

If $h_2(n) \geq h_1(n)$ then h_2 dominates h_1 and domination translates directly into efficiency! Because A* using h_2 will never expand more nodes than using h_1.

This is true because all nodes with an evaluation smaller than the cost of the solution will be evaluated. Therefore, by making $h(n)$ as big as possible, less nodes will be in that set!

Therefore it is very important to find an heuristic that estimates the distance to the solution as close to the real distance as possible.

One way of coming up with good heuristics is to solve the relaxed problem perfectly: a problem where there are no constraints, no movement constraints, no nothing...

Then that apply that heuristic to the constraint model because it will certainly underestimate but may be close to the actual value.

A tile can move from square A to square B if
A is horizontally or vertically adjacent to B **and** B is blank,

we can generate three relaxed problems by removing one or both of the conditions:

(a) A tile can move from square A to square B if A is adjacent to B.
(b) A tile can move from square A to square B if B is blank.
(c) A tile can move from square A to square B.

Considering only one of them at the time:

- From a), there's the Manhattan distance.
- From b) the Gasching's heuristic.
- From c), the misplaced tiles heuristic because the actual distance would be one movement.

If we have many heuristics, it may be hard to find a clear best heuristic. Therefore, one may simply use the maximum of them at each node as a new heuristic and that would be the best one and one that is still admissible.

$$h(n) = \max\{h_1(n), \ldots, h_m(n)\}$$

A curiosity, a program called ABSOLVER (Prieditis, 1993) was able to create relaxed problems from problem definitions and did output very useful results for Rubik's Cube and for 8-puzzle.

One other way of learning heuristics is from experience. By solving that kind of problem a few times, we start to know when we are getting closer and that can be coded into an heuristic.

7.7 The best of formal and natural languages - First Order Logic

The Chapter 8 of the book of the course is a wonderful read. Here is the key content:

One can define objects, relations (among objects) and functions (relations with an unique output for the given input).

- Objects: people, houses, numbers, theories, Ronald McDonald, colors, baseball games, wars, centuries ...
- Relations: these can be unary relations or **properties** such as red, round, bogus, prime, multistoried ..., or more general n-ary relations such as brother of, bigger than, inside, part of, has color, occurred after, owns, comes between, ...
- Functions: father of, best friend, third inning of, one more than, beginning of ...

The **domain** is the set of objects. Symbols come in three kinds: **constant symbols** , these stand for objects, **predicate symbols** that stand for relations and **function symbols** which stand for functions. Symbols will begin wtith uppercase letters.

$$
\begin{array}{rcl}
\mathit{Sentence} & \rightarrow & \mathit{AtomicSentence} \mid \mathit{ComplexSentence} \\
\mathit{AtomicSentence} & \rightarrow & \mathit{Predicate} \mid \mathit{Predicate}(\mathit{Term}, \ldots) \mid \mathit{Term} = \mathit{Term} \\
\mathit{ComplexSentence} & \rightarrow & (\ \mathit{Sentence}\ ) \mid [\ \mathit{Sentence}\ ] \\
& \mid & \neg\ \mathit{Sentence} \\
& \mid & \mathit{Sentence} \wedge \mathit{Sentence} \\
& \mid & \mathit{Sentence} \vee \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Rightarrow \mathit{Sentence} \\
& \mid & \mathit{Sentence} \Leftrightarrow \mathit{Sentence} \\
& \mid & \mathit{Quantifier}\ \mathit{Variable}, \ldots\ \mathit{Sentence} \\
\\
\mathit{Term} & \rightarrow & \mathit{Function}(\mathit{Term}, \ldots) \\
& \mid & \mathit{Constant} \\
& \mid & \mathit{Variable} \\
\\
\mathit{Quantifier} & \rightarrow & \forall \mid \exists \\
\mathit{Constant} & \rightarrow & A \mid X_1 \mid \mathit{John} \mid \cdots \\
\mathit{Variable} & \rightarrow & a \mid x \mid s \mid \cdots \\
\mathit{Predicate} & \rightarrow & \mathit{True} \mid \mathit{False} \mid \mathit{After} \mid \mathit{Loves} \mid \mathit{Raining} \mid \cdots \\
\mathit{Function} & \rightarrow & \mathit{Mother} \mid \mathit{LeftLeg} \mid \cdots \\
\textsc{Operator Precedence} & : & \neg, =, \wedge, \vee, \Rightarrow, \Leftrightarrow
\end{array}
$$

Figure 8.3 The syntax of first-order logic with equality, specified in Backus–Naur form (see page 1060 if you are not familiar with this notation). Operator precedences are specified, from highest to lowest. The precedence of quantifiers is such that a quantifier holds over everything to the right of it.

8 Machine Learning - Supervised Learning

Supervised learning concerns the problems where the objective is to predict something based on previous data. The counterpart Unsupervised Learning tries to find patterns in unlabelled data.

More generally, the dataset for supervised learning problems consists on a feature vector **x** and a output vector **y** as opposed to unsupervised learning where everything is features / data.

There are two main types of problems, regression and classification. The only difference between them is the expected output: regression aims to predict continuous outcomes while classification regards separating inputs in classes, thus a discrete output.

Some tools can be used to solve both problems, like Neural Networks. We'll have a look which tools are best for which problems.

8.1 Regression Problems - Least Squares

These are the two most commons types of problems. Probably every supervised learning problem can be *classified* as one of these.

Regression is when the output should be continuous, classification when the output should be in discrete classes.

About the first, a measure to minimize is the difference between our prediction to the value we want to achieve. The Sum of the Square Errors (SSE) is very standard cost function to minimize.

The function that is required to minimize is loss/cost/risk function. Nomenclature wise is a problem... Therefore, the following letters/terms can be will be used interchangibly: L (Loss Function) or J (more used when the weights or coefficients are θ) or R (Risk Function):

$$R = \frac{1}{2m} \sum_{i=1}^{m} \left(y^{(i)} - \hat{y}^{(i)} \right)^2$$

Where $\hat{y}$ is our prediction or estimation of the true value of y and **m** is the number of training samples we have. Therefore $y^(i)$ constitutes the outcome of sample i.

One prediction can be made with:

$$\hat{y} = \beta_0 + \beta_1 x_1 + \cdots + \beta_n x_n$$

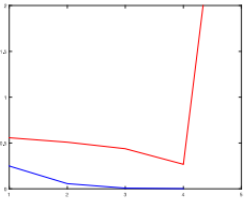
Where **n** is the number of features we are using. Note that features and data points are different things. We can have data regarding only one measure but take the square and the cube of the measure multiplied to different coefficients in order to try to better estimate the function.

β's are usually used in regressions while in neural networks letters like θ or w are more common.

One intuition that is important to have is that **the more we increase the features, the more likely it is that we end up overfitting our training set and loosing generalization capabilities for the actual test data** . This is why it is important to divide all the available data in sets, the model will be trained to guess the training samples, not samples that it never saw before.

- One set for training and one set for testing the prediction capabilities;
- One training set, one validation set and one test set.

The last option is meant to validate the model. The model is trained with the training set, then some parameters are tuned with the validation set, namely the number of polynomial features, regularization term and step size related parameters (like momentum or adaptive step size), and the actual performance is testing in the test set. This way, we avoid optimistic measures of performance by not testing in data used for training.



Average loss in the training set (blue) and in an independent set (red), as a function of polynomial degree.

The result wouldn't be very different if done with the number of iterations or number of features. In particular, it is called doing an "early stop" when the iteration that minimises the loss in the test set is early in the minimization process. It is useful when the model starts overfitting the data.

8.1.1 How to calculate the coefficientes

After having the coefficients, given any other set of data points we can already give predictions on the output.

Note that we are trying to minimize the cost/risk/loss function. The actual cost function would have to be some sort of prediction because it's impossible to know exactly how much the actual outcome will be, despite knowing exactly what the outcome of the model will be to a certain feature vector. As opposed to the empirical risk function where the training outcome and the training predicted result are used, therefore being able to calculate the difference between each estimation and the supposed outcome. To compute the real risk, the expected value of the SSE is necessary. Also, there are continuous results so an integral is required:

$$R = E[y - \hat{y}] = \int_{-\infty}^{+\infty} L(y, x)\phi(y)dxdy$$

Because a potential function to give us a measure on how frequent certain values are is not known, the only way is to approximate the actual error empirically, using the model with some test data. This will degenerate in the actual cost function presented before. L here is meant to denote the loss of one sample which is nothing more than the squared error.

One thing that won't happen in all problems is having an analytical and optimum solution for them. Actually, minimizing the SSE is a kind of problem is called the **Least Squares** . This kind of problem is very usually used in optimisation and often a closed solution is possible.

In this case, since we are searching for the function's minimum, the functions partial derivatives need to be zero in order to have a critical point (maximum, minimum or saddle point).

In this case,

$$\frac{\delta R}{\delta \beta_0} = -2\sum_{i=1}^{n}(y^{(i)} - \beta_0 - \beta_1 x^{(i)}) = 0$$
$$\frac{\delta R}{\delta \beta_1} = -2\sum_{i=1}^{n}(y^{(i)} - \beta_0 - \beta_1 x^{(i)})x^{(i)} = 0$$

Is possible to simplify further these equations, putting the β's in evidence and separating sums, arriving at:

$$\begin{bmatrix} \sum_{i=1}^{n} 1 & \sum_{i=1}^{n} x^{(i)} \\ \sum_{i=1}^{n} x^{(i)} & \sum_{i=1}^{n} x^{(i)2} \end{bmatrix} \begin{bmatrix} \hat{\beta}_0 \\ \hat{\beta}_1 \end{bmatrix} = \begin{bmatrix} \sum_{i=1}^{n} y^{(i)} \\ \sum_{i=1}^{n} y^{(i)} x^{(i)} \end{bmatrix}$$

Is possible to invert these equations and get the expressions for the coefficients. However there's one important factor to have in mind. Are we aiming at a minimum, maximum or something different like a saddle point? The Hessian Matrix will tell us.

8.1.2 Extrema Conditions and Hessian Matrix

Videos: 87 - Warm up to the second partial derivative test to... 89 - Second partial derivative test intuition

$$H = \begin{bmatrix} \frac{\partial^2 SSE}{\partial \beta_0^2} & \frac{\partial^2 SSE}{\partial \beta_0 \partial \beta_1} \\ \frac{\partial^2 SSE}{\partial \beta_1 \partial \beta_0} & \frac{\partial^2 SSE}{\partial \beta_1^2} \end{bmatrix} = 2 \begin{bmatrix} \sum_{i=1}^{n} 1 & \sum_{i=1}^{n} x^{(i)} \\ \sum_{i=1}^{n} x^{(i)} & \sum_{i=1}^{n} x^{(i)2} \end{bmatrix}$$

In one dimension, to find an extrema is necessary to equalise the first derivative to zero, and the second derivative must be positive - in case of a minimum - or negative - in case of a maximum. If the second derivative is zero at the critical point, then there's an inflexion point. A similar analysis must be done in $(n+1)$-D where n is the number of features used in the regression.

Guaranteeing that the first derivative is zero, which in N dimensions is correspondent to guaranteeing that the gradient is zero in that point, is the first step. Setting $\nabla f = 0$ means that, in that point, the function is not increasing or decreasing in any of the N directions. So the first derivative makes sense.

However, when we go to the second derivative, the meanings get a bit more complicated.

In 2D, if the second derivative was 0, it was certainly a saddle point, if it was > 0 or < 0 it was, respectively, a local minimum or maximum.

The conditions we should impose in 3D is to have a positive (for finding a minimum) or negative (for finding a maximum) definite Hessian Matrix. While it is possible to attribute a meaning to second derivatives in order to just one variable, being nothing more than the concavity in those 2 directions, why do the cross derivatives play a role as well? And, why do they mean really?

Well, first the explanation on why it is needed: there are functions that across multiple dimensions still show that it is an extrema but then there's an inflexion along directions that are not along the axis. So, checking the axis is not enough. Why checking the cross partial derivatives makes it enough?

The Second Derivative Test

$$f_{xx}(x_o, y_o) f_{yy}(x_o, y_o) - f_{xy}(x_o, y_o)^2 \gtrless 0$$

If it is greater than 0, we have a maximum or a minimum and have to check the value of f_{xx} or f_{yy} to be sure. If it is less than 0, we have a saddle point. If it equals 0, then we don't know if it is a saddle point, but it is not a min or max therefore, at least for now, we certainly don't care.

Cross or Mixed partial derivatives can be switched? Yes if the function is C^2. (Boring to prove theorem called: Schwarz' Theorem)

Therefore, we just need to compute one of the cross derivatives.

Also this works because the second derivative test is nothing more than the determinant of the Hessian matrix. The determinant is the product of every eigenvalue of that matrix, therefore it can only be positive if they are both positive or both negative, in which cases there is, respectively, a minimum or a maximum.

But why do eigenvalues tell us this? Because they tell us how the eigenvectors are scaled! And the eigenvectors of such matrix will be the greatest and the least curvatures. Therefore, they either have the same signal / are scaled the same amount, or

Some other links that helped with this:

- Differential Geometry
- Criterior for critical points - Maximum, Minimum or Saddle?
- David Butler - Facts about Eigenvalues

The two main properties of eigenvalues that allow us to quickly calculate them from the Hessian matrix (specially if it is 2x2) are:

$$\mathrm{tr}(A) = \sum_{i=1}^{n} \lambda_i$$

$$\det(A) = \prod_{i=1}^{n} \lambda_i$$

Because there are only 2 variables, there can't be very big changes across more than 2 main directions, so it is possible to quantify the main directions which will be the eigenvectors directions.

The eigenvalues of the Hessian Matrix are also called principal curvatures and the eigenvectors the principal directions.

8.1.3 Analytical Expression for the Coefficientes

From the equation presented in the end of 8.1.1, we can re-write the SSE and the normal equations in the following way.

$$\text{cost function:} \quad SSE(\beta) = \|y - X\beta\|^2$$

$$\text{normal equations:} \quad (X^T X)\hat{\beta} = X^T y$$

And arrive at the analytical expression through the simple inversion of the normal equations. Another way of reaching the analytical expression is deriving the cost function.

$$\text{parameter estimates:} \quad \hat{\beta} = (X^T X)^{-1} X^T y$$

Cost function

$$\begin{aligned} SSE &= \|y - X\beta\|^2 = (y - X\beta)^T (y - X\beta) \\ &= y^T y - 2y^T X\beta + \beta^T X^T X\beta. \end{aligned}$$

Computing the gradient and making it equal to zero

$$\nabla_\beta SSE = -2X^T y + 2X^T X\beta = 0,$$

Note however that:

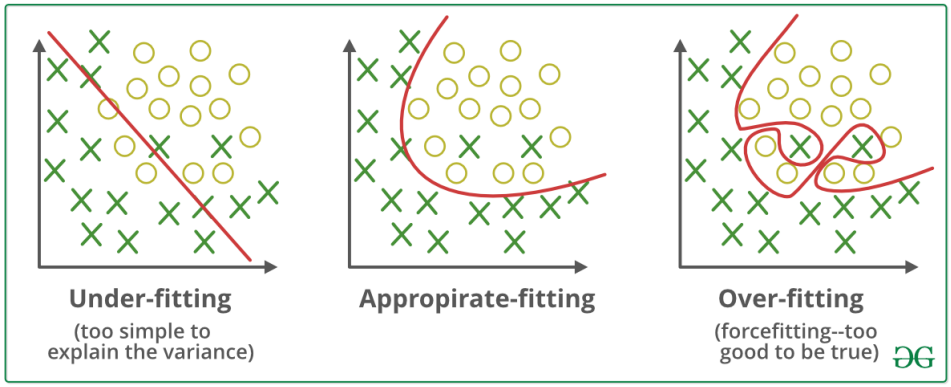
The inverse of matrix $X^T X$ may not exist due to two main reasons:

- small amount of data *e.g.*, number of data points smaller than the number of features.
- redundant features (linearly dependent) *e.g.*, duplicated features.

A final remark on multiple outputs: in case our feature vector serves to estimate more than one output, we can simply use it separately for each output!

8.1.4 Regularization

This is the method of taking importance away from the minimization of the errors between the training set supposed outcomes and the actual model outcomes for those samples. If we don't take importance away, the model may become too good at predicting training examples and may forget that it should predict a tendency and generalize well for the test data.



Performing a regularization consists on nothing more than adding a new parameter to the cost function, in order to shift away the focus of minimizing the SSE.

There are generally two terms that can be added. One with the **norm of the coefficients squared** and the other is the with the module of the coefficients squared.

For the norm squared, if the regularization is applied to a regression - **which is not a necessity since regularization can even be applied to Neural Networks** - it's called Ridge Regression:

$$\hat{\beta}_{\text{ridge}} = \arg\min_{\beta} \|y - X\beta\|^2 + \lambda\|\beta\|^2$$

$$\hat{\beta}_{\text{ridge}} = (X^T X + \lambda I)^{-1} X^T y$$

Two key things to note:

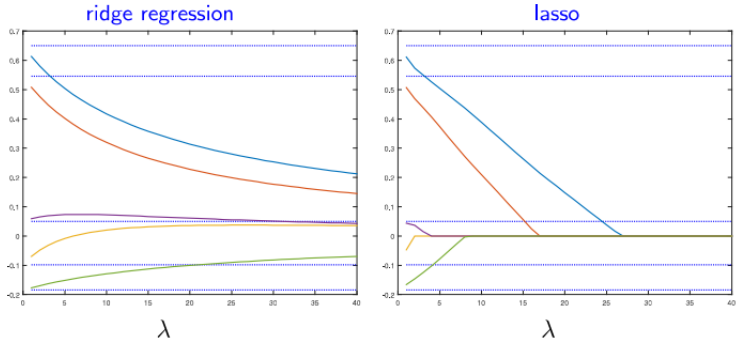
- Note that β_0 is usually not included as the data should be normalised already for much better results. Normalised data means data that has zero mean in every feature and outcome.
- If $(X^T X)$ is singular, the least squares estimate is not unique. Regularization will help finding an estimate even then because $(X^T X + \lambda I)$ is always non-singular.

For the simple norm of the coefficients, when applied to a regression problem it is called the Lasso Regression:

$$\beta_{\text{lasso}} = \arg\min_{\beta} \|y - X\beta\|_2^2 + \lambda\|\beta\|_1$$

$$||\beta||_1 = \sum_{j=1}^{n} |\beta_j|$$

The key difference between the two is that Ridge aims to minimize the norm of all of them while Lasso aims to minimize the sum of module of each of them. Therefore, Ridge it is much likely to pull closer to zero the biggest ones as those are the ones that matter the most for the Euclidean norm, while Lasso will try to pull each of them to 0, there's a direct dependency between a coefficients and the cost function. This is also why the Lasso Regression is called to do feature selection: because if the SSE doesn't depend on the coefficients, the regularization term with the sum of the norms will put that coefficients to zero very quickly.



Again, recall that the data should be centered - have zero mean - and that after calculating the model we need to de-centre it to obtain the real predictions!

How should we proceed if the training data $\mathcal{T} = \{(x^{(1)}, y^{(1)}), \ldots, (x^{(n)}, y^{(n)})\}$ are not centered?

1. pre-processing: $x'^{(i)} = x^{(i)} - \bar{x}$, $y'^{(i)} = y^{(i)} - \bar{y}$ ($\bar{x}, \bar{y}$ average values computed in the training set);
2. estimate linear model without intercept: estimate model $y' = x'^T \beta'$, with $\beta' \in \mathbb{R}^p$, using the pre-processed data $\mathcal{T} = \{(x'^{(1)}, y'^{(1)}), \ldots, (x'^{(n)}, y'^{(n)})\}$ and regularization;
3. invert pre-processing: $\hat{\beta} = [\hat{\beta}_0 \ \hat{\beta}'^T]^T$ where $\hat{\beta}_0 = \bar{y} - \bar{x}^T \hat{\beta}'$;

8.1.5 Optimization problems - Gradient Descent and Newton's Method

The gradient descent is probably the most know method to approximate a functions minimum. By changing the direction of the step we have the gradient ascent.

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta J(\theta^{(t)})$$

The above expression works because the gradient points the direction of the maximum growth of the function. Therefore, taking a step in the opposite direction will lead to the minimum.

Momentum and Adaptive StepSize

Momentum $0 \leq \alpha \leq 1$:

$$\Delta \mathbf{x}^{(n+1)} = \alpha \Delta \mathbf{x}^{(n)} - \eta \nabla f[\mathbf{x}^{(n)}]$$
$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta \mathbf{x}^{(n+1)}$$

or, alternatively, by

$$\Delta \mathbf{x}^{(n+1)} = \alpha \Delta \mathbf{x}^{(n)} - (1-\alpha)\eta \nabla f[\mathbf{x}^{(n)}]$$
$$\mathbf{x}^{(n+1)} = \mathbf{x}^{(n)} + \Delta \mathbf{x}^{(n+1)}$$

We'll use this second version.

Note that we want to pick the $\mathbf{x}$ that brings the function to a minimum. Therefore, the "exponent" will simply refer to the iteration number, not the sample like in the previous section. If α is closer to 1 the memory of the previous increment is more taken into account, meaning that the increment will change only slightly. The closer the parameter gets to 0, the closer we get to the normal situation. This is called the momentum term because it gives the convergence some inercia, the behaviour of momentum. By changing the increment slowly, it may converge faster and have less abrupt changes.

Adaptive Step size, with typical values: $u = 1.2, d = 0.8$:

$$\theta_i^{(t+1)} = \theta_i^{(t)} - \eta_i^{(t)} \frac{\partial J}{\partial \theta_i}\left(\theta^{(t)}\right).$$

Step size update

$$\eta_i^{(t)} = \begin{cases} u\eta_i^{(t-1)} & \text{if } \frac{\partial J}{\partial \theta_i}\left(\theta^{(t)}\right) \cdot \frac{\partial J}{\partial \theta_i}\left(\theta^{(t-1)}\right) > 0 \\ d\eta_i^{(t-1)} & \text{otherwise} \end{cases}.$$

Let f be the function where the objective minimum lies, the Divergence criterium is given by:

$$f(x^{(n+1)}) > f(x^{(n)})$$

Where the threshold is found in the equality.

Newton's method

Given by:

$$\theta^{(t+1)} = \theta^{(t)} - \left[H(\theta^{(t)})\right]^{-1} \nabla J_\theta(\theta^{(t)})$$

Where the gradient and the Hessian Matrix are given by:

$$\nabla_\theta J = \begin{bmatrix} \frac{\partial J}{\partial \theta_1} \\ \frac{\partial J}{\partial \theta_2} \\ \vdots \\ \frac{\partial J}{\partial \theta_d} \end{bmatrix} \qquad H = \begin{bmatrix} \frac{\partial^2 J}{\partial \theta_1^2} & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_2} & \cdots & \frac{\partial^2 J}{\partial \theta_1 \partial \theta_d} \\ \frac{\partial^2 J}{\partial \theta_2 \partial \theta_1} & \frac{\partial^2 J}{\partial \theta_2^2} & \cdots & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_d} \\ \vdots & \vdots & \cdots & \vdots \\ \frac{\partial^2 J}{\partial \theta_d \partial \theta_1} & \cdots & \frac{\partial^2 J}{\partial \theta_2 \partial \theta_2} & \frac{\partial^2 J}{\partial \theta_d^2} \end{bmatrix}$$

The Newton's method converges insanely fast! But requires the inversion of the Hessian matrix which can be a serious problem...

Newton's Method - Intuition and Demonstration

This is a very very good proof!

8.1.6 How to optimise hyperparameters

There will always be parameters to optimise in order to obtain the best model possible. It was said before that 3 sets should be selected: training, validation and testing. And that it was in the validation set that all hyperparameter tunning should be done. The simplest way is calculate the model with all combinations of the parameters possible and see which one performs better in the validation set.

There can be one other problem: little data. If the data is too little, dividing it into sets can start to give biased results to the accuracy.

One way of calculating the accuracy with more ... accuracy ... is splitting the data in k folds and rotate:

Data: k folds $\mathcal{T}_k$.
for k=1, ..., K **do**
  $f = \text{train}(\mathcal{T} \setminus \mathcal{T}_k)$;
  $P_k = \text{perform}(f, \mathcal{T}_k)$;
end
$P = \bar{P}_k$

Finally, if both things need to be done at the same time, then: 1- k folds need to be made. 2- for each fold, all values of the hyperparameters need to be used for training and tested in the test set. Note: it will be used for training T except (T_i and T_j) that are, respectively, the test set and the validation set. So the hyperparameters testing will be done with T_i. When the all combinations are done, the hyperparameters are selected for the bet one and the actual model is trained with T except T_i depending on the fold considered. 3- use the performances of each fold to get the best average of performance.

(Note that this is not very used...)

Data: k folds $\mathcal{T}_k$.
for $i = 1, \ldots, K$ **do**
  for *all values of* ξ **do**
    for $j \neq i$ **do**
      $f = \text{train}(\mathcal{T} \setminus (\mathcal{T}_i \cup \mathcal{T}_j), \xi)$;
      $P(\xi)_j = \text{perform}(f, \mathcal{T}_j)$;
    end
    $P(\xi) = P(\bar{\xi})_j$
  end
  $\hat{\xi}_i = \arg\min_\xi P(\xi)$;
  $f = \text{train}(\mathcal{T} \setminus \mathcal{T}_i, \hat{\xi}_i)$;
  $P_i = \text{perform}(f, \mathcal{T}_i)$;
end
$P = \bar{P}_i$

8.2 Neural Networks

8.2.1 Formalisation

On the surface, a NN is nothing more than a set of weights connecting a set of neurons. This is represented in Figure 4.

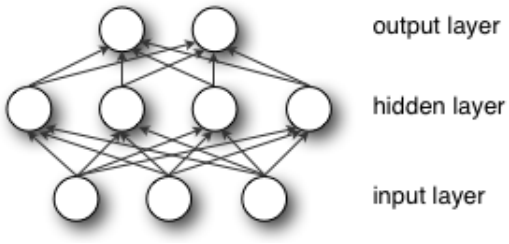


Figure 4: Overview of a Neural Network

This particular architecture is called a Multilayer Perceptron (MLP), the standard NN. In other architectures not all layers are required to be fully connected e.i every neuron from the previous layer is connected to all neurons from the next layer, however, for simplicity, let's restrict this formalisation to MLPs.

Let $w_{ij}^{(l)}$ be the weight that connects the output of the j-th neuron of layer $l-1$ to the i-th neuron of layer l. If we call the output of a neuron j-th of layer l, $z_j^{(l)}$, and the input of the i-th neuron of layer l $s_i^{(l)}$, one may write Equation (3).

$$s_i^{(l)} = w_{i0}^{(l)} + \sum_{j=1}^{N^{(l)}} w_{ij}^{(l)} z_j^{(l-1)} \tag{3}$$

With $N^{(l)}$ being the number of neurons of layer l. Also, the input of the first layer is the input of the network, i.e $z^{(0)} = \mathbf{x}$.

Note further that is possible to relate the input of a neuron to its output through that neuron's activation function $g(x)$. Equation (4) shows this relation, with L the number of layers in the MLP.

$$z_i^{(l)} = g\left(s_i^{(l)}\right) \quad , i = 1, ..., N^{(l)} \quad , l = 1, ..., L \tag{4}$$

Activation functions of neurons may vary across layers and they simply relate the input with the output.

There are many kinds of activation functions, each with its advantages and disadvantages - refer to actFunctions for a more in-depth analysis. In this work only two are used, Rectified Linear Unit (ReLu) and Softmax, which are represented in the set of Equations (5).

$$\begin{cases} \text{ReLu}\left(s_i^{(l)}\right) = \max\left(0, s_i^{(l)}\right) \\ \text{Softmax}\left(s_i^{(l)}\right) = \frac{\exp\left(s_i^{(l)}\right)}{\sum_{j=1}^{N^{(l)}} \exp(s_j)} \end{cases} \tag{5}$$

So far we've see how to get the input to the output - forward propagation is the technical term - but wasn't explained yet how to adjust the weights such that the networks starts behaving like expected. It is done with backpropagation.

Backpropagation is an algorithm that consists of calculating the effect that each weight has on the output and adjust that weight accordingly to that relation and accordingly to how wrong the output is. Backpropagation can be done with Gradient Descent methods and all their associated optimization techniques. The simple version of backpropagation with the classic gradient descent is presented in Equation (6) where η denotes the step size.

$$w_{ij}^{(l)} = w_{ij}^{(l)} - \eta \bigtriangledown_{ij}^{(l)} \quad , \text{with} \quad \bigtriangledown_{ij}^{(l)} = \frac{\delta J}{\delta w_{ij}^{(l)}} \tag{6}$$

The complete expression for the partial derivatives of every weight is in Equation (7).

$$\bigtriangledown_{ij}^{(l)} = \delta_i^{(l)} z_j^{(l-1)} \quad , \tag{7}$$

$$\text{with} \begin{cases} \delta_i^{(l)} = g\left(s_i^{(l)}\right) \sum_{k=1}^{N^{(l+1)}} \delta_k^{(l+1)} w_{ki}^{(l+1)} & , \text{for } l = 1, ..., L-1 \\ \delta_i^{(L)} = g\left(s_i^{(L)}\right) \frac{\delta J}{\delta z_i^{(L)}} & , \text{ otherwise, i.e for } l = L) \end{cases}$$

Note that $z_i^{(L)} = \hat{y}_i$. Additionally, J is the cost/loss function, the function that tells us how far from the correct result the output is. For classification problems, a good cost function usually is Cross Entropy, Equation (8). However, bear in mind that modifying the cost function to one that is more frequently used in Regression problems one can easily use the NN in regression problems.

$$J(y, \hat{y}) = -\frac{1}{N} \sum_{i=1}^{N} \sum_{k=1}^{K} \mathbb{1}_{y_i = y_k} \log\left(\hat{y}_i\right) \tag{8}$$

Where N is the number of samples and $\mathbb{1}_{y_i=y_k}$ denotes the indicator function that only is 1 when the supposed output is y_k. In other words, the inner sum should always have just one term corresponding to the logarithm of exit of the neural network that should be 1 for that class. This is because due to the common use of Softmax activation function in the last layer, the outputs will be percentages of certainty. And the cost function should be the logarithm of that the certainty of that class only.

More specifically, if we want to categorise images of digits, our neural networks will have 10 outputs, one for each class/digit. When a sample that has the number 3 written on it is propagated until the end, the loss of that computation should be the logarithm of the probability in the 3^{rd} exit, because a perfect NN would output a 1 in the 3^{rd} exit and 0 in the others. In fact, $\log(1)$ is 0 loss and $-\log(0)$ is infinite (positive) loss.

Further optimizations

In order to achieve an efficient implementation, the previous equations can be written in a vectorized way and the forward and backward propagation will be reduced to matrix multiplications. Additionally, for a low level analysis it becomes relevant to keep track of all matrices dimensions, thus they are the following:

- $z^{(l)}$ is $\left(1 + N^{(l)}\right) \times 1$ and $z^{(l)} =$
- $s^{(l)}$ is $N^{(l)} \times 1$ and $s^{(l)} = W^{(l)} z^{(l-1)}$;
- $W^{(l)}$ is the weights matrix and is $N^{(l)} \times \left(1 + N^{(l-1)}\right)$ which should make sense when looking for the above formula and that along its rows are the weights multiplied to the previous layer plus one for the bias unit;
- $\bigtriangledown^{(l)} = \delta^{(l)} \ldots$

Note that feed forward of all samples at once is possible through the correct definition of X matrix and the correct changes.

Some history

Data: k folds $\mathcal{T}_k$.
for k=1, ..., K **do**
  $f = \text{train}(\mathcal{T} \setminus \mathcal{T}_k)$;
  $P_k = \text{perform}(f, \mathcal{T}_k)$;
end
$P = \bar{P}_k$

Pros
It can be proved that the Rosenblatt algorithm solves any binary problem in a finite number of iterations, provided the training data can be separated by a hyperplane in feature space.

However, there's a big problem with only being able to distinguish data that can be separate with an hyperplane: very often the data doesn't behave that way.

How should we choose the number of layers?
Cybenko (1989) proved that a multilayer perceptron with 1 hidden layer is an universal approximator of any continuous function defined on a compact subset of $\mathbb{R}^p$. This is a useful theorem but it does not explain how many units are needed nor how should the weights be chosen.

- Common practice shows that it is often better to use more layers since the network can synthesize a wider variety of nonlinear functions with less units.
- It also shows that deeper networks (with more layers) are more difficult to train.

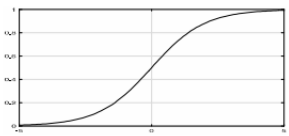
One can make a more complex analysis of the situation...

Activation Functions and Architecture

Some activation functions:

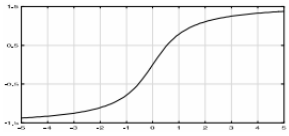
sigmoid: logistic function

$$g(s) = \frac{1}{1 + e^{-s}}$$



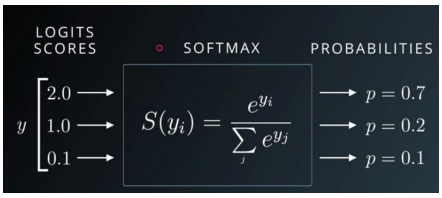
sigmoid: arctangent function

$$g(s) = \arctan s$$



Two other are Rectifier Linear Unit and the softmax:

$$f(x) = \begin{cases} x & \text{if } x > 0, \\ ax & \text{otherwise.} \end{cases}$$



In the all layers with the exception of the last one, ReLu or the logistic function work very well. The last layer must have a function that returns results between 0 and 1, therefore only SoftMax and sigmoid functions like the logistic function would work properly.

Training methods:

The gradient vector includes the contribution of all the training patterns. The weight update using all the training patterns in each iteration is called the batch mode.

$$\Delta w_{ij} = -\eta \frac{\partial \mathcal{R}}{\partial w_{ij}} = -\frac{\eta}{n} \sum_k \frac{\partial L(y^{(k)}, \hat{y}^{(k)})}{\partial w_{ij}} = -\frac{\eta}{n} \sum_k \frac{\partial L^k}{\partial w_{ij}} .$$

Another alternative consists of using one training pattern k, only, and updating the weights with that information. This is called the on-line mode

$$\Delta w_{ij} = -\eta \frac{\partial L^k}{\partial w_{ij}} .$$

A third hypothesis consists of updating the NN weights using a small subset of training patterns. This is known as mini-batch mode.

One very interesting fact is that the image features are not selected by anyone. The network itself crafts its features throught backpropagating the changes required to get the images right. This is true for all Neural Networks.

Also, in image recognition for instance, but in deep neural networks in general, the last layers usually are fully connected.

8.2.2 Neural Networks - BackPropagation

Some of the most useful websites to check while trying to demonstrate the backpropagation algorithm:

- all backpropagation derivatives
- Error (deltas) derivation for backpropagation in neural networks
- 3Blue1Brown Neural Networks - Specially the last one, on backpropagation.

The fastest way is the following:

Hand written demonstration on Drive, link in Intro - the first page.

Note that multiple training modes are possible. The normal one is on-line, where the increment to the weight is nothing more than the step multiplied by the respective partial derivative - Equation (9). Then one can do the batch-mode where all the samples are considered for the weights update - Equation (10). Finally, there's the mini-batch mode that doesn't use all the training samples.

$$w_{ij}^{(l)} = w_{ij}^{(l)} + \triangle w_{ij}^{(l)}$$

$$\triangle w_{ij}^{(l)} = -\eta \frac{\delta C^k}{\delta w_{ij}^{(l)}} \tag{9}$$

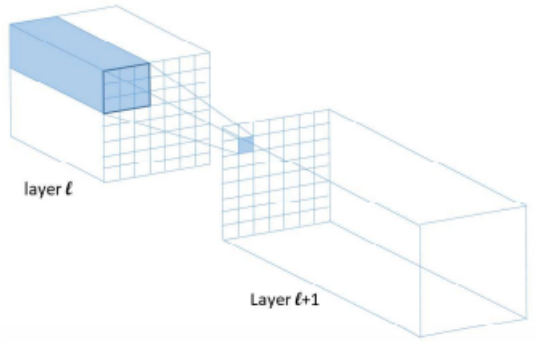
$$\triangle w_{ij}^{(l)} = -\frac{\eta}{m} \sum_{k=1}^{m} \frac{\delta C^k}{\delta w_{ij}^{(l)}} \tag{10}$$

8.2.3 Neural Networks - Convolutional

Convolutional Neural Networks are extremely useful for applications like image recognition. They take a width x height x image depth 3D array and convolve it with a kernel, generating an 2D array.

Because many elements in an image are translation invariant, considering patches of the image is one of the best ways of acquiring features.

Convolutional neural networks allow us to use less inputs to our neural networks. We map a region of a picture to just one pixel/input. We tend to use many different kernels - a set of weights to multiply to each of pixel in the set we chose - to perform this step.



At the end of the convolution, an activation function (like ReLu) is applied.

Note that this convolutional layer can be applied to a 3D array to reduce it to 2D. Moreover, many different kernels can be convolved with the section of the 3D array creating several layers. If there are 20 kernels, we'll have 20 2D arrays, therefore a new 3D array that is all the pixels in the several vicinities, weighted with different kernels.

3D input: $z_{ijk}^{\ell-1}$ $\ell - 1$ - number of input layer

3D kernel: h_{ijk}^{ℓ}

2D output:

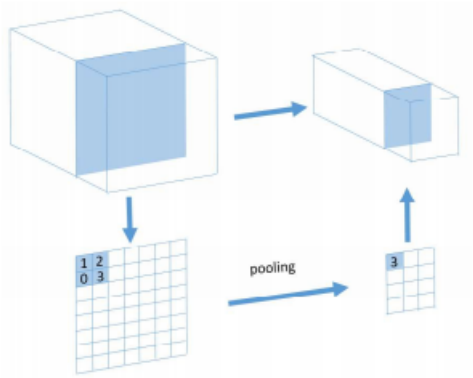
$$s_{ij}^{\ell} = \sum_{p} \sum_{q} \sum_{r} h_{pqr}^{\ell} z_{i+p,j+q,0+r}^{\ell-1}$$

$$z_{ij}^{\ell} = g(s_{ij}^{\ell})$$

Each filter produces a 2D output known as a feature map. Stacking the feature maps produced by multiple filters leads to a 3D array.

Pooling

2D to 2D but with smaller dimensions. Downsize the feature array by choosing what we consider to be the most important values. For instance, for each 4x4 square of pixels, we choose the maximum of them (the highest value in grey scale, for instance). Therefore, the end result will have 16x less pixels/inputs.



Overall, the tendency is the kernel to be smaller, but the network to have many many layers.

8.2.4 Kernels

Why kernels are important? They facilitate a lot the mapping of features to higher dimensions!

Why kernels?

They will become specially important in support vector machines when the data is not separable in certain dimensions but if we increase the dimensions is possible to separate the data by an hyperplane.

8.2.5 Classification Problems

When the required output is a discrete class. Instead of regressing something, the objective is to separate data into classes. Through the learning of what features each class has, be able to classify new data accordingly.

A Bayes Estimator is a Maximum Likelihood estimator, a perfect estimator. However, it requires information that we usually don't have. Estimators like the Naive Bayes are useful because they enable simplifications if one is willing to accept the assumptions they entail.

We'll start with the maximum likelihood estimator and then particularize to Naive Bayes.

Let's assume K classes and y_k being each one. We want to choose y_k such that $k = \underset{k}{\text{argmax}} p(y_k|\mathbf{x})$. This is a maximum likelihood estimator because we want to choose the class that is more likely given the data we received.

From the Bayes Theorem, one can rewrite that probability as $p(y_k|\mathbf{x}) = \frac{p(\mathbf{x}|y_k)p(y_k)}{p(\mathbf{x})}$. Given the denominator as a scaling factor and, normally, an equal probability of each class, one can rewrite the maximum likelihood estimator as:

$$y_k : k = \underset{k}{\text{argmax}}\ p(\mathbf{x}|y_k)$$

Note that the rigorous shape of the above probability is $p(x_1, x_2, ..., x_n|y_k)$, where n is the number of features in our feature vector $\mathbf{x}$.

So far, some considerations have been made but no approximations. The Naive Bayes estimator simplifies the estimation at a cost: assuming the features are independent. If the features are independent of each other, the joint probability distribution degenerates in the multiplication of the marginal PDFs. Likewise, the joint conditional probability density function degenerates in the multiplication of the marginal probability density functions. Mathematically:

$$p(x_1, x_2, ..., x_n|y_k) = \prod_{i=1}^{n} p(x_i|y_k)$$

Further, one should note that due to the logarithmic function being monotone, maximizing a function or the its logarithmic has the same maximizing argument. This is a way of transforming the above multiplication into $\sum_{i=1}^{n} \log(p(x_i|y_k))$.

One particular case where the Naive Bayes Estimator performs decently is language recognition. Not because the letters in the *ngrams* considered are independent - that is certainly not the case, e.g 'ã' is much more likely to be followed by a 'o' in portuguese - but because the assumption of independence throughout the languages has somewhat of the same effect, not influencing the estimation too much. Therefore, assuming independence between characters in a case where there are so many characters and so many *ngrams* combinations to derive our estimator from, doesn't return that bad results and is quite a good application of Naive Bayes, a very simple estimator.

8.3 Support Vector Machines

8.3.1 Linear Classifiers

We call linear methods the classifiers whose decision boundaries are linear, hyperplanes.

A way of classifying multiple classes is to assign to class i, a discriminant $f_i(x) = \left[1x^{\mathrm{T}}\right]\beta_i$.

The purpose of this function is to be 1 when the class of the input is class i, and be 0 when the class of the input is not y_i.

Therefore, decisions are made with $\hat{y}_i : i = \underset{i}{\operatorname{argmax}} f_i(x)$.

Logistic Regression is a Generalized Linear Model (GLM) which can perform prediction and inference while linear Perceptrons can only achieve predictions, and in this case will perform similarly as to logistic regression. Statistical Modelling versus Machine Learning in practice.

In logistic regression, since each output represents a probability, the maximisation of the correct probability is the aim. Therefore gradient ascent (the same apart from a signal) is used.

8.3.2 SVM's

This is a great tutorial on the vector math part of SVM.

Vectors of SVMs

This is a very very good and complete tutorial about SVM: Fletcher: SVM explained

SVMs can only separate between two classes, then is necessary to do strategies like 1 vs All to separate more classes. This is our n sample dataset with p features per sample.

$$\mathcal{D} = \{(\mathbf{x}_i, y_i) \mid \mathbf{x}_i \in \mathbb{R}^p,\ y_i \in \{-1, 1\}\}_{i=1}^{n}$$

From Fletcher's:

In order to use an SVM to solve a linearly separable, binary classification problem we need to:

- Create $\mathbf{H}$, where $H_{ij} = y_i y_j \mathbf{x}_i \cdot \mathbf{x}_j$.
- Find $\boldsymbol{\alpha}$ so that

$$\sum_{i=1}^{L} \alpha_i - \frac{1}{2} \boldsymbol{\alpha}^T \mathbf{H} \boldsymbol{\alpha}$$

  is maximized, subject to the constraints

$$\alpha_i \geq 0 \quad \forall_i \text{ and } \sum_{i=1}^{L} \alpha_i y_i = 0.$$

  This is done using a QP solver.

- Calculate $\mathbf{w} = \sum_{i=1}^{L} \alpha_i y_i \mathbf{x}_i$.
- Determine the set of Support Vectors S by finding the indices such that $\alpha_i > 0$.
- Calculate $b = \frac{1}{N_s} \sum_{s \in S} (y_s - \sum_{m \in S} \alpha_m y_m \mathbf{x}_m \cdot \mathbf{x}_s)$.
- Each new point $\mathbf{x}'$ is classified by evaluating $y' = sgn(\mathbf{w} \cdot \mathbf{x}' + b)$.

We'll now have a look to some of the whys. Note that this is for a Linearly Separable Binary Classification.

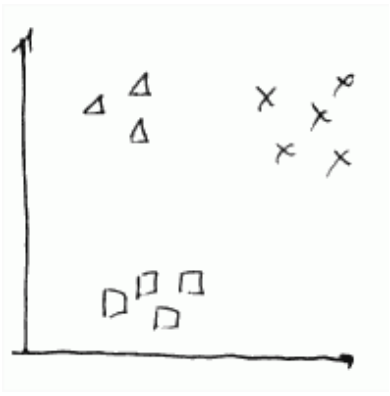
- How about non-Binary? One vs All.
- What if it is not Linearly Separable? Kernels.
- What else, Linear Regression? Yes.
- But the SVM always separates with an hyperplane right, always linearly? Unless a Non-linear SVM is created.

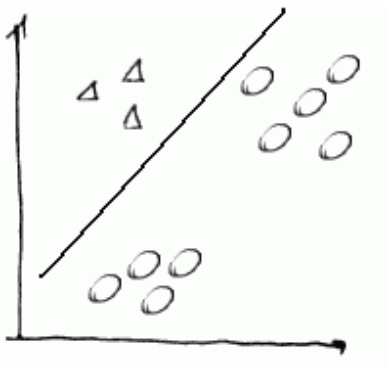
8.3.3 One vs All approach

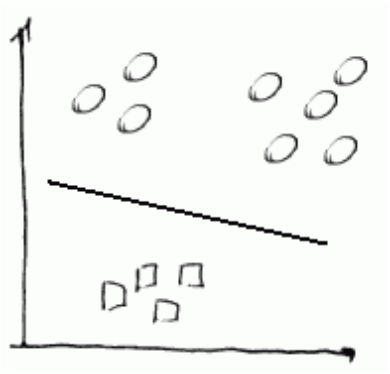
Applies to SMVs and all classification problems that can only distinguish between 2 classes.

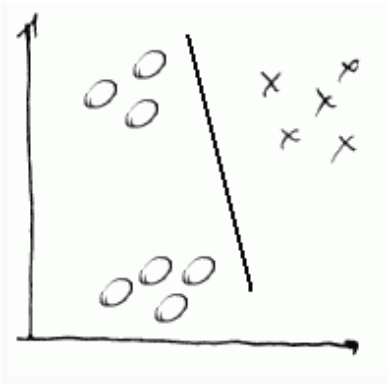
It consists of training K classifiers that will make decisions between the K classes of option. Each classifier will consider the labels of that class against the "other" label that will be all the other samples.

The class with higher certainty is the correct one.









8.3.4 Formulation of SVMs

All the edge cases explained above will be constructed above this, usually being a simple modification.

We want to define a plane in an n-dimension space that divides our data in its classes.

Also, the best plane will be the one that has the highest margin of error, thus choosing correctly with higher probability.

One can define a plane in $\mathbb{R}^n$ like:

$$\vec{n} \cdot (\vec{r} - \vec{r_o}) = 0$$

$\vec{n}$ is the vector normal to the plane, $\vec{r}$ is a random point and $\vec{r_o}$ is the vector that has the point $x_o = (x_1, x_2, \ldots, x_n)$.

For instances, in 3D, calling now the normal to the plane $\vec{w} = (a, b, c)$ and $\vec{x} = (x, y, z)$ and representing vectors in bold:

$$\begin{aligned} & \vec{n} \cdot (\vec{r} - \vec{r_o}) = 0 \\ \Leftrightarrow \; & a(x - x_o) + b(y - y_o) + c(z - z_o) = 0 \\ \Leftrightarrow \; & ax + by + cz + d = 0 \\ \Leftrightarrow \; & \mathbf{w} \cdot \mathbf{x} + b = 0 \end{aligned}$$

To compute the distance of a point to the hyperplane, we simply have to calculate the norm of the projection to the unitary normal vector.

$$dist = \mathbf{x} \cdot \frac{\mathbf{w}}{||\mathbf{w}||}$$

Note now that if we want the distance of the plane to the origin, calling $\mathbf{v}$ to our vector such that $\mathbf{v} = (x_1 - x_o, y_1 - y_o, z_1 - z_o)$, where P_o belongs to the plane and P_1 is the point in question, and $\mathbf{n}$ to our unitary normal just so:

$$\begin{aligned} dist &= \left| v \cdot \frac{\mathbf{w}}{||\mathbf{w}||} \right| \\ dist &= \left| (x_1 - x_o, y_1 - y_o, z_1 - z_o) \cdot \frac{\mathbf{w}}{||\mathbf{w}||} \right| \\ dist &= \frac{|A(x_1 - x_o) + B(y_1 - y_o) + C(z_1 - z_o)|}{\sqrt{A^2 + B^2 + C^2}} \end{aligned}$$

Note now that we need the point in the plane that is closer to the origin to calculate this distance. In fact is possible to find that point by intersecting with the plane a line directed with the normal to the place from the origin. This point however is not required for this analysis. Also, note that if P_1 is the origin, and P_o is

the nearest point to the origin belonging to the plane, because it belongs to the plane $\mathbf{w} \cdot \mathbf{x_o} + b = 0$, so the expression degenerates in:

$$dist = \frac{b}{||\mathbf{w}||}$$

More generally:

1. Is possible to calculate the distance of any point to the plane if we choose any point in the plane and get a vector from that point to the point the distance is required. This is because the plane is not defined uniquely by the orthogonal vector.
2. A sign can be given to the distance $\frac{b}{||\mathbf{w}||}$. If it is below the plane, the distance is negative. If above it is positive. Before only the absolute value was considered, however if we consider w to always be the normal pointing up, we can choose this convention;

This is where everything gets interesting:

This is called Support Vector Machines because they use Support vectors that are the points that are closer together. The amount of support vectors required will depend on the dimensions, in 2 dimensions, 2 points are enough to define a plane of equal distance to both. In 3 dimensions 2 points are not enough because there's a line of points of equal distance and infinite planes can pass through that line. Therefore, 3 points are required. N dimensions, n points to define the problem. Actually, these points will be used to define $\mathbf{w}$. The value of b will be set with the margin between the hyperplanes!

Two hyperplanes can be defined as:

$$\mathbf{w} \cdot \mathbf{x} + b = \pm\delta$$

The actual value of δ won't matter because it won't be more than a factor of scale.

The mid way between the two hyperplanes will be the boundary. And the decision will be based on what side of the boundary the point lies in.

$$x \cdot w + b > 0 \Rightarrow \hat{y} = +1$$
$$x \cdot w + b < 0 \Rightarrow \hat{y} = -1$$

$$\hat{y} = \text{sign}(x \cdot w + b)$$

We want to maximise the distance between the two planes in order to have the boundary as far from the support vector as possible.

Also, may be it is a good idea to take more than the n closest points because those might generate a plane that doesn't divide the data properly. But this can be taken into account later. First, what is the margin between the two described planes and how to maximise it?

Note that since the hyperplanes are parallel, the constant will be the only thing moving them along the normal direction and the distance between them can certainly be deduced directly from that constant.

The constants of both planes are $b \pm \delta$. To calculate the distance between them, is doing $\frac{|D_1 - D_2|}{||w||}$. This formula can be explained by the calculation of the difference of both planes' distance to the origin. Note now that the b will be $b - \delta$.

Therefore, we get:

$$d = \frac{2\delta}{||w||}$$

And to maximize this distance is necessary to minimize $||w||$, with the a constraint per training sample:

Constrains: training data must obey

$$\left.\begin{array}{ll} x^{(i)} \cdot w + b \geq +1 & \text{for } y^{(i)} = +1 \\ x^{(i)} \cdot w + b \leq -1 & \text{for } y^{(i)} = -1 \end{array}\right\} \Rightarrow y^{(i)}(x^{(i)} \cdot w + b) - 1 \geq 0, \ \forall i$$

Constraint optimization is a problem for the Lagrange multipliers. See Section 10.4.2 to see how to apply the method carefully.

The first application gets us to the primary Lagrangean function:

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i [y^{(i)}(x^{(i)} \cdot w + b) - 1] \,,$$

$$L_P = \frac{1}{2}\|w\|^2 - \sum_{i=1}^{n} \alpha_i y^{(i)}(x^{(i)} \cdot w + b) + \sum_{i=1}^{n} \alpha_i \,,$$

$\alpha_i \geq 0$ are Lagrange multipliers.

Then, by doing the gradient and replacing it in the above expression:

Optimization

$$\frac{\partial L_P}{\partial w} = 0 \Rightarrow w = \sum_{i=1}^{n} \alpha_i y^{(i)} x^{(i)} \,, \qquad \frac{\partial L_P}{\partial b} = 0 \Rightarrow \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \,.$$

Replacing these variables, we obtain the dual formulation,

$$L_D = \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \sum_{i,j=1}^{n} \alpha_i y^{(i)} (x^{(i)} \cdot x^{(j)}) y^{(j)} \alpha_j \,, \quad s.t. \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \,,$$

Note that this is a problem dual from the first one. We are still solving the same optimization problem but with this substitution, the final expression to optimize depends only on the dot inner product of $x^{(i)}$ and $x^{(j)}$ which will be very important for the **kernel trick** . Also, with the formulation below (after some simplifications) it's possible to use a Converx Quadratic Programming (QP) Solver . See the end of Section 10.4.2 for more info on this Duality.

$$\max_{\alpha} \sum_{i=1}^{n} \alpha_i - \frac{1}{2} \alpha^T H \alpha \,, \quad s.t. \quad \alpha_i \geq 0 \;\; \forall \, i \,, \quad \sum_{i=1}^{n} \alpha_i y^{(i)} = 0 \,,$$

where $\alpha = [\alpha_1 \; \ldots \; \alpha_n]^T$ and $H_{ij} = y^{(i)}(x^{(i)} \cdot x^{(j)}) y^{(j)}$.

This is a convex quadratic programming (QP) problem that can be solved by standard QP algorithms and provides all α_i.

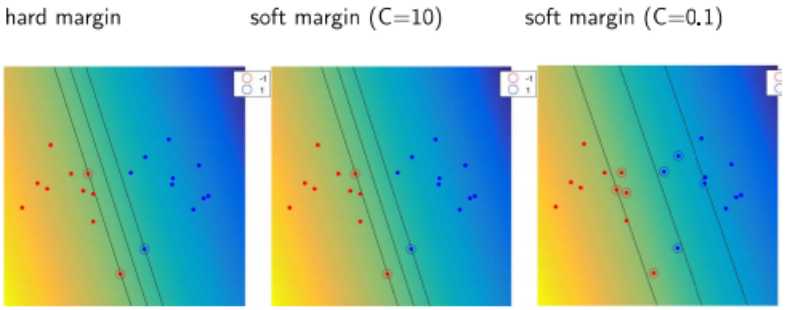
From α's we may obtain:

1. support vectors (S): all $x^{(s)}$ such that $\alpha_s > 0$,
2. normal vector: $w = \sum_{s \in S} \alpha_s y^{(s)} x^{(s)}$
3. offset: $b = \frac{1}{N_s} \sum_{s \in S} \left[y^{(s)} - \sum_{m \in S} \alpha_m y^{(m)} (x^{(m)} \cdot x^{(s)}) \right]$
4. Classification of data: $f(x) = sign(x \cdot w + b)$

In conclusion, with $f(x) = xcdotw + b$, the SVM provides more than a decision, a score, a certainty associated with that decision.

8.3.5 Soft Margin - Slack Variables

This was performed for an hard margin in linear separable data. If the data is not linear separable, either some **Kernel Trick** is performed or some slack is added to account for variables in the other side of the boundary. Slack variables are the penalties that will be added to points in the wrong side of the boundary and thieir sum should be minimised, i.e $C \sum_{i=1}^{n} \epsilon_i$, where C is the scale of the penalty.



8.3.6 Non-Linear SVM & The Kernel Trick

In higher dimensions the data may be separable, but in the current one it may not be.

There are mappings/transformations to increase the feature space dimensionality. With higher dimensions and more complex features that are the weird combinations of the current ones, is possible that there's a separation.

$$\tilde{\mathbf{x}} = \phi(\mathbf{x}) = \begin{bmatrix} \phi_1(\mathbf{x}) \\ \phi_2(\mathbf{x}) \\ \vdots \\ \phi_p(\mathbf{x}) \end{bmatrix}$$

The kernel function $\phi(\mathbf{x})$ will map n features to p, where (usually at least) $p \geq n$.

The biggest problem is that this high dimensionality may drive the problem unfeasible since that each sample will have much more information. And the process of mapping all samples, apply the SVM and them map the boundary back to the first feature space is quite time consuming.

Instead of that, since the dual formulation just requires the inner product of features, is possible to define a kernel function that is the product of the two mappings and compute the inner products in the low dimension input space instead of really having to climb the dimensionality ladder.

$$k(x^{(i)}, x^{(j)}) = \phi(x^{(i)}) \cdot \phi(x^{(j)})$$

The typical kernels:

linear: $k(x^{(i)}, x^{(j)}) = x^{(i)T} x^{(j)}$

rbf: $k(x^{(i)}, x^{(j)}) = e^{-\frac{1}{2\sigma^2}\|x^{(i)} - x^{(j)}\|^2}$

polynomial: $k(x^{(i)}, x^{(j)}) = (x^{(i)T} x^{(j)} + a)^b$

The linear kernel is the one adopted in linear SVM.

We note that some kernels depend on hyperparameters that have to be specified or learned during the training phase *e.g.*, typically by *ad hoc* procedures or by cross validation.

9 Machine Learning - Unsupervised Learning

9.1 Reinforcement Learning and Decision Making

An agent to make the wisest decision in its situation has to have knowledge on what state he is in, how the environment will evolve, how it will be like if he performs a certain action (taking into account stochastic environments) and a utility function to know how much that state contributes to its happiness.

10 Math

10.1 Taylor Series

Incredibly important for simplifications, approximations and to have a notion of the shape of the function.

The definition is as follows: A real or complex-valued function $f(x)$ that is infinitely differentiable at a real or complex number a can be written as a power series the following way

$$f(x) = \sum_{n=0}^{\infty} \frac{f^{(n)}(a)}{n!}(x-a)^n = \\ = f(a) + \frac{f'(a)}{1!}(x-a) + \frac{f''(a)}{2!}(x-a)^2 + \frac{f'''(a)}{3!}(x-a)^3 + \cdots$$

Some utilities of this formula:

- The approximation $(1+x)^n \approx 1 - x^n$, when x is much smaller than 1. (Binomial Series)
- When approximating $y(x) = cos(2x) + \epsilon$ with a polynomial using least squares, the coefficients fo that polynomial will tend for the coefficients of Taylor's polynomial. Many MacLaurin Series (Taylor Series at the origin) can be found Here in the Wiki .

10.2 Erlang Models B and C

These are statistical models to predict the necessary resources to satisfy a certain amount of traffic with the constraint of not serving at most a certain percentage of the incoming traffic at a given time.

Instead of thinking like:

> "" *Hey, it's simple arithmetic! We get 3 200 calls a day. That's 400 calls an hour. Each call lasts three minutes, so each person can handle 20 calls and hour. So we'll need 20 incoming lines and 20 people to answer the phones.*"

Good models were created because of this! This is completely wrong if the calls don't arrive evenly distributed across the day, which is very likely. There are busy hours where the traffic is much bigger than in any other time of the day.

A.K. Erlang derived nice formulas for us to use. Note that the terms "call" and "resource" will be used interchangeable because these models can be used for a immense variety of applications besides call-centres. For instance, to figure how many printers to buy so that people in the office can print decently.

One last note: these formulas were derived for "infinite sources" of incoming traffic. However, they work very very well for cases when there are about 10x more sources than trunks. For real world applications that tends to be the case.

The concepts:

- **Erlang** : Dimensionless unit. Regards the continuous use of a resource. Normally is referred to hours, therefore 90 minutes of traffic are regarded as 1.5 erlangs.
- **Grade of Service** : Probability of all servers/trunks being busy when a resource is requested for a given amount of traffic and for a certain amount of trunks.

Erlang B

Use if a call is really blocked, i.e if the person trying to use the resource doesn't wait to use it when all the trunks are busy at that time. Given 2 from Traffic , Grade of Service or Number of Trunks one may calculate the other.

Tables can be found here.

Erlang C

Use if a blocked call is delayed instead of the "come back later" that happens in the previous model. This one needs as well the average duration of each call .

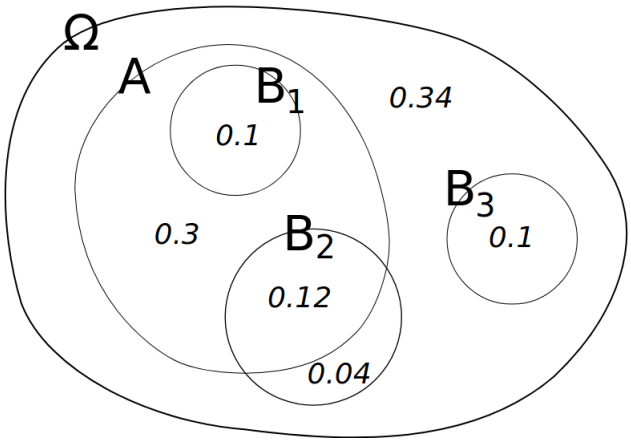
10.3 Probabilities

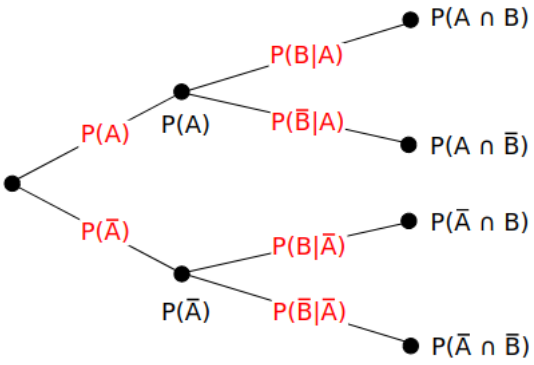
10.3.1 Conditional Probability

Probability of A happening given that B happened before. The formula:

$$P(A \mid B) = \frac{P(A \cap B)}{P(B)}$$

Some intuitive images:





10.3.2 Bayes Theorem

This an important topic. Is appears just about everywhere!

What is the probability of something, given that something happened.

What we'll see very very often is that this is associated with decision boundaries and thresholds. Depending on what we saw, the likeliness of what we thing was transmitted or the class we thing our data may belong, may change.

This is Bayes' Theorem in its discrete form:

$$P(A|B) = \frac{P(B|A)P(A)}{P(B)}$$

$$P(y|X) = \frac{P(X|y)P(y)}{P(X)}$$

In just about every case, we want to choose:

- the symbol that was most likely transmitted given that we received a certain signal (the maximum likelihood criteria, which Bayes Theorem definitely is, may degenerate in a shortest distance criteria in phase and quadrature plots - remember QAM and PSK)
- the class that a certain sample most likely fits into is the class that most likely would generate that kind of sample.

Therefore

10.4 Optimization

10.4.1 Unconstraint Optimization

Let $f : \Omega \to \mathbb{R}$ be a continuously twice differentiable function at $\mathbf{x}^*$.

If $\mathbf{x}^*$ satisfies $\nabla f(\mathbf{x}^*) = 0$ and $\nabla^2 f(\mathbf{x}^*)$ is positive definite then $\mathbf{x}^*$ is a local minimum.

Refer to Section 8.1.2 for details of the Hessian Matrix.

10.4.2 Constraint Optimization - Lagrange Multipliers

The Lagrange Multipliers is a mathematical optimization method, allows the finding of local maxima or minima of a function, subject to constraints.

Perfect Explanation why it works:

Quora - Intuition on Langrange Multipliers

And a video showing exactly how it's done:

Khan Academy - Lagrange Multipliers Example

How to use:

Let f be the optimizing function and g the constraint function. They are both scalar functions, i.e $\mathbb{R}^n \to \mathbb{R}$. For example purpose, let is consider $n = 2$.

$$f(x, y) = x^2 y$$
$$g(x, y) = x^2 + y^2 = c$$

The Lagrange Multipliers Method for Constraint Optimization tells us that:

$$\triangle f = \lambda \triangle g \tag{11}$$

Thus, is common to formalize the Lagrangean function:

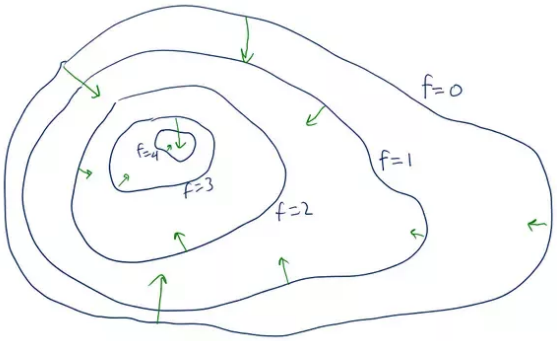
$$L(x, y, \lambda) = f(x, y) - \lambda g(x, y)$$

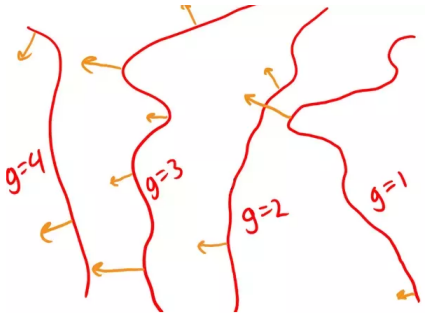
To get the optimized point, we just have to solve the following system of $n + 1$ equations for the n dimensions and the lagrange multiplier:

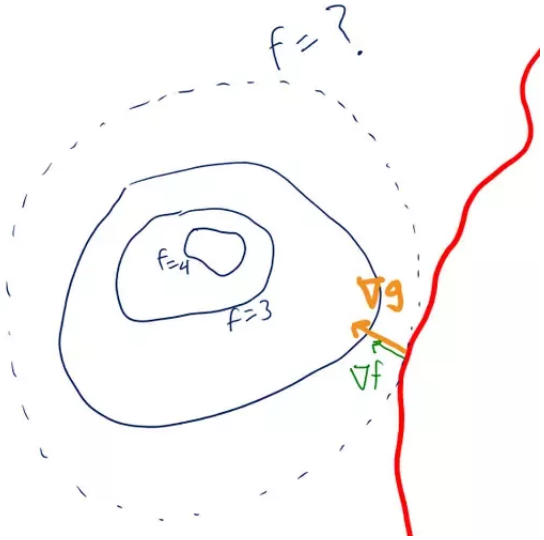
$$\begin{cases} \triangle L(x, y, \lambda) = 0 \\ g(x, y) = c \end{cases}$$

Curiosity notes:

- it doesn't matter if $g(x, y) = c$ is the constraint of if $g(x, y) = x^2 + y^2 - c = 0$ is the constraint. The constraint should be the last equation and the Lagrangean doesn't need that constant in there as it will disappear for our purposes
- The core of this methods is equation (11). This is true because in the point we want to obtain, there's tangency between the constraint and the optimizing function's contour lines







- If there are many constraints, they are summed in the Lagrangean, there will be an extra multiplier for each one and each one will appear in the final system of equations. Note that for each Lagrange Multiplier there should be one constraint equation to get that multiplier value.
- Yet another generalization can be made when there are constraints that are not $= c$ but are $\geq c$. Refer to page 7 on Langrage Duality of Andrew Ng stanford Lectures notes on SVMs .
- Still regarding Lagrange Duality, consists of transformating the general Lagrangean into something else. In particular, Quadratic Programming is done is the SVM chapter.

10.4.3 Constraint Optimization - Quadratic Programming

11 English

Unbelievable how this got to this point, right? I know, I know... But,indeed English is important.

11.1 Commas

Always the same thought "should I put a comma here?" followed by the exercise rereading the sentence multiple times until one makes up his mind. Fluency requires well founded rules and this is what we are here for.

To do after: check this website, it seems to have cool rules Grammar Commas

11.1.1 Commas on adverbs like *therefore,* *however,* and *indeed*

When the adverb is essential to the meaning of the clause, or if no pause is intended or desired, commas are not needed.

Two sentences showing when commas aren't needed:

- If you cheat and are therefore disqualified, you may also risk losing your scholarship.
- That was indeed the outcome of the study. (this one goes both ways really, how'd you like it to sound?)

Two other showing when they fit well:

- A truly efficient gasoline-powered engine remains, however, a pipe dream.
- Indeed, not one test subject accurately predicted the amount of soup in the bowl.

11.1.2 Commas to separate Adjectives - Only if they are parallel

Consider the following sentence "The event is part of a catchy, public health message about the importance of emergency preparedness." Catchy and "public health" are not coordinate adjectives. The point is not that the message is catchy and public health; it's that the public health message is catchy. Therefore, no comma is necessary: "The event is part of a catchy public health message about the importance of emergency preparedness."

If, by contrast, the sentence read, for example, "The event is part of a catchy, quirky message about the importance of emergency preparedness," note that because catchy and quirky are parallel — they are coordinate adjectives — a comma should separate them.

11.2 British English vs American English

This probably was the main reason for the birth of this section. I never know when to use certain letters in both spelling ways. So this part will cover firstly that issue. The following website was helpful in many ways and I'll cite others as the list gets extensive.

Wikipedia - American and British English spelling differences

Moreover, I intend to organise this collection in many sub-sections, each presenting a ground truth or a general rule that allows one to replace the doubts with a short recall and reasoning exercise, which tends to be quicker and feel better than searching the thing in Google.

11.2.1 -ise, -ize (-isation, -ization)

This is the worst. But the pain will be over after a few paragraphs.

Generalise is British, I guess. That's how far I've gone so far.

11.3 Irony vs Sarcasm

Who the hell knows the difference? Probably no one. But I'll attempt to establish a solid base for argumentation on this topic with enough examples.

The following definitions are cited from the Sarcasm Wiki Page :

1. Definition:

   "Sarcasm is "a sharp, bitter, or cutting expression or remark; a bitter gibe or taunt". Sarcasm may employ ambivalence, although is not necessarily ironic. Most noticeable in spoken word, sarcasm is mainly distinguished by the inflection with which it is spoken and is largely context-dependent."

2. Usage:

   "In sarcasm, ridicule or mockery is used harshly, often crudely and contemptuously, for destructive purposes. It may be used in an indirect manner, and have the form of irony, as in *What a fine musician you turned out to be!* , *It's like you're a whole different person now...,* and *Oh... Well then thanks for all the first aid over the years!* or it may be used in the form of a direct statement, *You couldn't play one piece correctly if you had two assistants.* The distinctive quality of sarcasm is present in the spoken word and manifested chiefly by vocal inflection ..."

3. Derek Bousfield writes:

   "The use of strategies which, on the surface appear to be appropriate to the situation, but are meant to be taken as meaning the opposite in terms of face management. (. . .) sarcasm is an insincere form of politeness which is used to offend one's interlocutor."

4. John Haiman writes:

   "There is an extremely close connection between sarcasm and irony, and literary theorists in particular often treat sarcasm as simply the crudest and least interesting form of irony."

   Also, he adds:

   "First, situations may be ironic, but only people can be sarcastic. Second, people may be unintentionally ironic, but sarcasm requires intention. What is essential to sarcasm is that it is overt irony intentionally used by the speaker as a form of verbal aggression."

5. Henry Watson Fowler writes:

   "Sarcasm does not necessarily involve irony. But irony, or the use of expressions conveying different things according as they are interpreted, is so often made the vehicle of sarcasm ... The essence of sarcasm is the intention of giving pain by (ironical or other) bitter words."

From (2) and (3) we can tell right away that **Irony and Sarcasm are not two forms of saying sentences, they have inherently different meanings. The sentence "is this sarcasm or irony" should change to "is this comment sarcastic or just somewhat ironic?"**

Furthermore, **Sarcasm is made to offend or attack. Attacks can be made in a playful way, but is an attack nonetheless** .

From the Irony Wiki Page :

One can say right away that Irony is much broader than Sarcasm, the wiki page is much longer.

1. Definition:

   "Irony (from Ancient Greek, meaning 'dissimulation, feigned ignorance'), in its broadest sense, is a rhetorical device, literary technique, or event in which what appears, on the surface, to be the case, differs radically from what is actually the case.

   Irony can be categorized into different types, including: verbal irony, dramatic irony, and situational irony. Verbal, dramatic, and situational irony are often used for emphasis in the assertion of a truth. The ironic form of simile, used in sarcasm, and some forms of litotes can emphasize one's meaning by the deliberate use of language which states the opposite of the truth, denies the contrary of the truth, or drastically and obviously understates a factual connection."

By the way, *simile* is:

And *litotes* is:

2.

11.3.1 Fun Facts

In late August 2016, North Korea banned sarcasm against the government. It was reported government gave the warnings in mass meetings across the country. Subsequent media reports suggest that North Korea banned sarcasm altogether.

11.4 How to Study English

From the course Spoken English for Technologists 2 in TUDelft, some tricks arise. One of them is writing everything that someone says in a video, for instance, a TEDx talk.

12 MATLAB

12.1 Plots

First things first: to create a vector of equally spaced elements is best to use **linspace** . It guarantees the same amount of points in each array more easily.

```
t = linspace(-Ts, Ts, 200);
plot(t, 2*t);
xlabel('x axis, duh');
ylabel('y axis, hud');
title('guess?');
grid minor; %sometimes..
```

Use **meshgrid** and **surf**ace to plot functions with multiple variables:

```
[xx, yy] = meshgrid(x,y);
surf(xx,yy, 2 * xx, 'EdgeColor', 'none');
```

Meshgrid is necessary because matlab doesn't know how to take do value iteration! The **surf** function DOES know that should take one value from x, run for all values of y and get the (x,y) value for each iteration from z. THEREFORE, z must be an length(x) by length(y) matrix! This is why the mesh grid command is used. It allows for this matrix creation just in the same way as one creates values for a 2D plot.

What **surf** will do is: check the sizes of the x and y vectors to see if it should iterate or just choose the right values. If x and y are row vectors, it will have to manually calculate all the possible combinations. If they are matrices (like the ones returned by meshgrid), it is able to just choose the appropriate combinations: take each element of each vector and plot it in space - notice what is returned by meshgrid is exactly the same dimensions as the Z vector will be.

To conclude: is possible to give x and y or X and Y as inputs for the **surf** function of MATLAB, however, Z must be a matrix! And that will only happen with meshgrid, otherwise it will be a vector without all combinations of values in x an y.

12.1.1 Contour

In portuguese "contorno", returns the levels curves of the function.

contour(X1, X2, Z, 10.^(-20:3:0)') will plot the Z points of the coordinates X1, X2, on the levels 10.^[-20 -17 -14 -11 -8 -5 -2]. Note that if the points X1 and X2 are not provided, there is no "set" where to which is possible to compute the level curves, so Matlab chooses one and it may not fit your previous graph.

12.1.2 Extra Stuff for Graphs

Check all properties of a graph here. Some of them:

Subplots - allow several figures in the same window. Writing subplot(1,2,1) divides a window in a grid of 1x2 and will access the first element of it. Increasing the final number changes the division of the grid selected which is where the subsequent plots are going to end up on.

xlabel, ylabel - labels for the Graphs

xlim, ylim - limits for the Graphs

set current figure position (and size) - set(gcf,'position',[x0, y0, width, height]); x0 is measure from the left of the screen. y0 is measure from the bottom of the screen. width is along x, height along y.

imagesc - Image with scaled colours. imagesc(A); colorbar; shows quite fast the values that the matrix A has. A very useful tool to debug and to quickly have an insight on the matrix contents and patterns.

LineWidth : a number for the thickness of the line

Dashed line : '–' makes the line dashed.

DisplayName : To specify a specific legend

legend is possible to position the legend according to the cardinal points, "North", etc... Use legend('Location', 'North').It is also possible to put it in a desired location with legend('Position', [.25 .25 .25 .25]).

'HandleVisibility' 'on' by default. This doesn't show a legend for a certain graph, useful for when we need to plot thresholds or something that we don't want to have a legend about.

12.2 Functions

To make a function, create a new file and write in the first line:

```
function [o1,o2,o3] = nameOfTheFunction(i1,i2,i3)
    %func code here
end
```

Where the o's are the outputs and the i's are inputs.

Is also possible to pass a function as an argument! For instance, if that function is what is suppose to be used inside another function.

integral(@log, x) is passing the built-in function *log* as an argument.

12.3 Set and Matlab Objects

Since you've probably done something with Java or something with python oriented in that sense, you probably know what an object is. In Matlab, there are many object as well. Instead of changing its properties in the arguments of the instantiation of that object, one may create that object and then use the set method.

The following two pieces of code do exactly the same thing regarding legends.

```
    h = legend;
    rect = [0.25, 0.25, .25, .25];
    set(h, 'Position', rect);

    legend('Position', rect);
```

The rectangle should be [x0, y0, width, height], in percentages, given that 1 is the size of the figure.

check

Actually, I have the sensation that in this case, "Position" is nothing more than an internal variable.

12.4 Save images

```
fig = figure;
plot(t,x);
pause(0.2); %may be necessary so that the figures ends plotting, before it starts saving
print(fig,'-dpng','img_filename'); %as png
```

Or use the following for printing the current figure:

```
plot(t,x);
print('myfig.png', '-dpng');
```

OR EVEN imagining it was necessary to open a file to plot that picture. File named 'ola.bin' (binary file), for instance. If the image is going to be included in some report, using .eps is a good idea since it allows "infinite zoom" because it describes the picture by vectors, instead of saving the information of each pixel of compressed.

The code below will save as .eps (Encapsulated PostScript) file.

```
plot(t,x, 'r');
hold on;
plot(t, x_window, 'b');
print([file(1:end-4) '-hamming'], '-depsc')
```

Check more formats in print page of matlab.

12.5 Opening stuff

In CSV format:

```
file = csvread('file.csv');
```

But better is to open as a matrix, in a big variety of formats:

```
%Excel
A = readmatrix('File.xlsx','Sheet', 3,'Range','A1:AX5000');
A = readmatrix('file.xlsx','Sheet', "name of sheet3",'Range','A1:AX5000');
%More formats coming when I use it for them
```

12.6 Max and Min

Max function is incredibly useful. Returns the maximum of a vector and the place it appears in! Of course, the **min** flavour exists too.

```
[maximum, index] = max(1:10)
```

You can even specify the dimension on which you want the maximum. So, if you want the maximum of each row, max(A, 2) will do a column vector with as many maximums as there are rows, thus a column vector.

12.7 Other useful tools

- It is possible to see **EVERY** command written in matlab. Just write 'commandhistory' or press the above arrow. The commands there are OLD!
- **Load** , **save** , **exist** , **return** to stop execution and **clear** one variable:

```
try
    B = load("A.mat");
    B = B.A; %to get the matrix out of the structure
catch
    if ~exist('Traces.xlsx', 'file')
        return
    end
    %create A code here;
    clear aux;
    save('A.mat', 'A');
end
```

- **squeeze** to take away not needed dimensions. For instance in a 1x1x5000 vector, the following code will result in a 5000x1 vector.

```
squeeze(R(prb,1,:))
```

- **find** to find an element in an array! Simply returns a vector with the indexes of where that element appears.

```
find(a == 0)
find(a < 5)
find(a == b) %return all i that a(i) == b(i), i.e all elements that are in both vectors
```

"The relational operators (>, <, >=, <=, ==, ~=) impose conditions on the array, and you can apply multiple conditions by connecting them with the logical operators and, or, and not, respectively denoted by the symbols &, |, and ~." From the Matlab Docs .

Additionally: "[row,col] = find(X) returns the row and column subscripts of each non-zero element in array X " And this explains the above uses of this function.

- A cool way of **copying an array size** :

```
a = zeros(size(b)) %copies size of b to the use we want to give a
```

- To select random elements of an array: permute the indexes and select the first n elements of that permutation.

```
m = 100; n = 10;
a = 1:m;
randIndexes = randperm(m);
b = a(randIndexes(1:n));
```

- Cell to Logical to Double:

```
B = double(cell2mat(A));
```

Copy

The `cell2mat` call converts it from a `cell` to a `logical` array, and `double` converts it to a double array.

12.8 Label data in Scatter plots

By far the best way of getting a decent result is to use an already written function. Adam Danz published here a function that does this perfectly for you.

From the examples it becomes very evident the use. The function requires at least the coordinates of every point and the labels to put in every data point.

```
labelpoints(x, y, 'Color', [1 0.5 0.5], 'FontSize', 12);
```

12.9 Create Gif from plots

It can't get easier than this. Chad Greene wrote a miraculous script and published it here .

Before any plot, write:

```
gif('mygiffilename.gif');
gif(gif_name,'DelayTime',0.2,'LoopCount',1,'frame',gcf);
```

Then, to insert a frame simply write **gif** :

```
for i = 1:10
    plot(x,y);
    gif;
end

web('mygiffilename.gif'); %will open on matlab web.
```

That is it. It can't get simpler. To view the gif with controls you can use a video player.

12.10 Write table to Excel

A matrix looks like a table but will look like a line if you use the writematrix. Instead, use **writetable** and convert the matrix to a table before!

```
table = array2table(squeeze(avg_rates_perTest(1,:,:)))

for scheduler = 1:length(schedulers_for_testing)
    writetable(table, ...
               num2str(scheduler) +"-avg_rates_per_test"+ ".xlsx");
end
```

Additionally, is possible to add the correct names to the columns, but they have to have possible variable names... So there are some limitations: just letters, numbers and underscores and has to start with a letter.

```
table = array2table(array, 'VariableNames', {'first_name', 'second', 'etc'});
```

13 LaTeX

In this section, basic easy to search commands will be presented.

As a very honourable mention as a perfectly clear LaTeXtutorial, please visit latex-tutorial.com/tutorials/ . I couldn't like that website more. It has the symbols, the tools and the tutorials in the top bar.

Although I prefer much more to use tablesgenerator.com/ a table generator, the symbols and tutorial sections are simply perfect.

13.1 Symbols that you never remember

First of all, the best place to search is Detexify . You can draw a symbol and the latex correspondence appears right away. For other common options, here a list:

- $\sim$ **Tilde** : $ \sim $ (uses: textcomp)
- \ **Backslash** : \textbackslash
- $\ll$ and $\gg$: \ll and \gg
- $\widetilde{\mathbf{h}}$: \widetilde{h}
- $\circledast$: \circledast
- $\otimes$: \otimes

13.2 Important Packages

Import packages with:

```
\usepackage{packagename}
```

Some important package to keep around:

- verbatim: allows for "\comments" which are very useful.
- todo or todonotes: allows for "\todo" and that is the best way of having reminders in your code.
- bm: allows for bold math "$\mathbf{H}$" = $\mathbf{H}$

13.3 Margins

To change the margins of the document this is simply the best place to go. The example sets you up with a standard wide paper size. Overleaf - Margins

13.4 Code listings

[language=Tex]

Use package:

```
\usepackage{listings}
```

To input a complete file of code, use the expression below.
In this case, the file is assumed to be right outside the report folder. Note also that the code should not have non unicode characters like º or else LaTeXwon't do a proper job, even if they are in strings.

```
\lstinputlisting{../gen_data1.m}
```

To paste code in a certain language do:
(the "no numbers" serves to suppress the numbers on the left, to be easier to copy)

Create a list of all code listings with:

```
\lstlistoflistings
```

13.5 Images side by side

From this tex.stackexchange question :

- 2 images: Side by side with one legend for each and one for both

```
\begin{figure}
    \centering
    \begin{subfigure}{.5\textwidth}
      \centering
      \includegraphics[width=.4\linewidth]{image1}
      \caption{A subfigure}
      \label{fig:sub1}
    \end{subfigure}%
    \begin{subfigure}{.5\textwidth}
      \centering
      \includegraphics[width=.4\linewidth]{image1}
      \caption{A subfigure}
      \label{fig:sub2}
    \end{subfigure}
    \caption{A figure with two subfigures}
    \label{fig:test}
\end{figure}
```

- 2 images: Side by side with one legend for each:

```
\begin{figure}
    \centering
    \begin{minipage}{.5\textwidth}
      \centering
      \includegraphics[width=.4\linewidth]{image1}
      \captionof{figure}{A figure}
      \label{fig:test1}
    \end{minipage}%
    \begin{minipage}{.5\textwidth}
      \centering
      \includegraphics[width=.4\linewidth]{image1}
      \captionof{figure}{Another figure}
      \label{fig:test2}
    \end{minipage}
    \end{figure}
```

- 4 images: 2x2 config. Note that is possible to replicate the reasoning for this code and make any number of figures side by side.

```
\begin{figure}[ht]
    \label{ fig7}
    \begin{minipage}[b]{0.5\linewidth}
      \centering
      \includegraphics[width=.5\linewidth]{example-image-a}
      \caption{Initial condition}
      \vspace{4ex}
    \end{minipage}%%
    \begin{minipage}[b]{0.5\linewidth}
      \centering
      \includegraphics[width=.5\linewidth]{example-image-b}
      \caption{Rupture}
      \vspace{4ex}
    \end{minipage}
    \begin{minipage}[b]{0.5\linewidth}
      \centering
      \includegraphics[width=.5\linewidth]{example-image-c}
      \caption{DFT, Initial condition}
      \vspace{4ex}
    \end{minipage}%%
    \begin{minipage}[b]{0.5\linewidth}
      \centering
      \includegraphics[width=.5\linewidth]{example-image}
      \caption{DFT, rupture}
      \vspace{4ex}
    \end{minipage}
  \end{figure}
```

13.6 Math - All of it

As before, a list of useful commands. First the code, then the result.

- System of equations:

```
\begin{equation*}
    \begin{cases}
       a = b \\
       c = 0
    \end{cases}
```

```
    \end{equation*}
\end{itemize}
```

$$\begin{cases} a = b \\ c = 0 \end{cases}$$

- Matrixes: short story

```
\[
\begin{bmatrix}
    x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\
    x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\
    \vdots & \vdots & \vdots & \ddots & \vdots \\
    x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn}
\end{bmatrix}
\]
```

$$\begin{bmatrix} x_{11} & x_{12} & x_{13} & \dots & x_{1n} \\ x_{21} & x_{22} & x_{23} & \dots & x_{2n} \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ x_{d1} & x_{d2} & x_{d3} & \dots & x_{dn} \end{bmatrix}$$

- Matrixes long story: from matrix environment stackexcange question, " amsmath: it defines 6 types of matrix environments: matrix (without any delimiter), pmatrix (delimiters: ( )), bmatrix ([ ]), Bmatrix ( ), vmatrix (— —), Vmatrix (—— ——)"

$$\begin{pmatrix} x' \\ y' \end{pmatrix} = \begin{bmatrix} \cos\theta & -\sin\theta \\ \sin\theta & \cos\theta \end{bmatrix} \begin{pmatrix} x \\ y \end{pmatrix}$$

```
\[
\begin{pmatrix}x'\\y'\end{pmatrix}=
\begin{bmatrix}
    \cos\theta & -\sin\theta\\
    \sin\theta & \cos\theta
\end{bmatrix}
\begin{pmatrix}x \\y \end{pmatrix}
\]
```

- If the matrix is coefficient extended, then some smart stuff needs to be used. Use the command below to implement the workaround, written by Stefan Kottwitz

```
\makeatletter
\renewcommand*\env@matrix[1][*\c@MaxMatrixCols c]{%
\hskip -\arraycolsep
\let\@ifnextchar\new@ifnextchar
\array{#1}}
\makeatother
```

After that you can use:

```
\begin{equation}
    \begin{bmatrix}[cccc|c]
     1 & 0 & 3 & -1 & 0 \\
     0 & 1 & 1 & -1 & 0 \\
     0 & 0 & 0 & 0 & 0 \\
  \end{bmatrix}
\end{equation}
```

$$\left[\begin{array}{cccc|c} 1 & 0 & 3 & -1 & 0 \\ 0 & 1 & 1 & -1 & 0 \\ 0 & 0 & 0 & 0 & 0 \end{array}\right]$$

- **A good practice** (depending on the context) would be to only number referenced equations. For this, all equations should have a short label like "eq:eqlabel", be referenced with \eqref and you need to use this to enable that automatic behaviour of only display names if referenced:

```
\usepackage{mathtools}
\mathtoolsset{showonlyrefs}
```

- Funky stuff. Argmin and a way of doing the norm. If a norm that fits automatically is required, then Google it and put it here.

```
\begin{equation}
    \hat{\bb{S}} = \operatorname*{argmin}_\bb{S} \left\Vert\bb{X} - \bb{A}\bb{S}\right\Vert^2_F
\end{equation}
```

$$\hat{\mathbf{S}} = \underset{\mathbf{S}}{\operatorname{argmin}} \|\mathbf{X} - \mathbf{AS}\|_F^2$$

- But this one is quicker..

```
\underset{x}{\operatorname{min}}
```

$\underset{x}{\min}$

- **Vectors** $\vec{o}$

```
\usepackage{ dsfont }
\vec{o}
```

- To center equations, use:

```
\begin{gather}
    x = 3+4+5+6+7+9\\
    \Leftrightarrow x = \sum_{i=3}^9 i
\end{gather}
```

Result:

$$x = 3 + 4 + 5 + 6 + 7 + 9$$
$$\Leftrightarrow x = \sum_{i=3}^{9} i$$

- Is also possible to have gathered and aligned environments inside equations:

```
\begin{equation}
    \begin{aligned}
        3(a-x) = 3.5x + a - 1        \\
        3a - 3x = 3.5x + a -1        \\
        a = \frac{13}{4}x - \frac{1}{2}
    \end{aligned}
\end{equation}
```

$$\begin{aligned} 3(a-x) &= 3.5x + a - 1 \\ 3a - 3x &= 3.5x + a - 1 \\ a &= \frac{13}{4}x - \frac{1}{2} \end{aligned}$$

- **Tilde**

```
$\tilde{x}$
```

Gives: $\tilde{x}$

13.7 Multicolumns

13.7.1 Multicolumns in Text

```
\setlength{\columnseprule}{1pt}
\def\columnseprulecolor{\color{blue}}

\begin{document}

\begin{multicols}{3}
[
All human things are subject to decay. And when fate summons, Monarchs must obey.
]

Hello, here is some text without a meaning. This text should show what
a printed text will look like at this place.

If you read this text, you will get no information. Really? Is there
no information? Is there.

\columnbreak
```

```
18
19  This will be in a new column, here is some text without a meaning. This text
20  should show what a printed text will look like at this place.
21
22  If you read this text, you will get no information. Really? Is there
23  no information? Is there...
24
25
26  A blind text like this give you information about the selected font, how the letters are written and an impression of the look.
        This text should contain all letters of the alphabet and it should be written in the original language. There is no need for
         special content, but the length of words should match the language.
27  \end{multicols}
```

Results in:

All human things are subject to decay. And when fate summons, Monarchs must obey.

Hello, here is some text without a meaning. This text should show what a printed text will look like at this place.

If you read this text, you will get no information. Really? Is there no information? Is there.

This will be in a new column, here is some text without a meaning. This text should show what a printed text will look like at this place.

If you read this text, you will get no information. Really? Is there no information? Is there...

A blind text like this give you information about the selected font, how the letters are written and an impression of the look. This text should contain all letters of the alphabet and it should be written in the original language. There is no need for special content, but the length of words should match the language.

13.7.2 Multicolumns in lists

```
1   \usepackage{multicol}
2   \begin{multicols}{2}
3       \begin{itemize}
4           \item item 1
5           \item item 2
6           \item item 3
7           \item item 4
8           \item item 5
9           \item item 6
10      \end{itemize}
11  \end{multicols}
```

Results in:

- item 1
- item 2
- item 3
- item 4
- item 5
- item 6

13.8 Itemize, Enumerate and Lists

I just found this website that has a great tutorial about these.

The main and most interesting thing I wanted to show here is a thing that is also present in the link with further explanations: is possible to change the bullet point style very easily:

```
1   \begin{itemize}
2       \item here is one item
3       \item[--] here is another
4       \item[$\ast$] here another
5       \item[$-$] Here another and I can continue!
6   \end{itemize}
```

Just by putting the character inside the rectangular parentheses is possible to create a list with that character as a bullet. The result of the above code:

- here is one item

– here is another

∗ here another

– Here another and I can continue!

Another way, better in terms of verbose per list, is this one:

```
%requires \usepackage{enumitem}
\begin{itemize}[label=$\ast$]
    \item ok, now they are the same
    \item see?
\end{itemize}
```

* ok, now they are the same
* see?

13.9 How to insert images from files outside the report file

Don't know... Tell me if you find out. I just know from inside the folder.

13.10 Good Tables with that diagonal line

Make a table with the tablegenerator.com and then use this:

```
%uses \usepackage{diagbox}

\begin{table}[h]
    \small
    \begin{tabular}{c|c|c|c}\hline
        \backslashbox{Room}{Date} & Monday & Tuesday & Wednesday \\ \hline
        room1 & & & \\ \hline
        room2 & & & \\
    \end{tabular}
\end{table}
```

Room \ Date	Monday	Tuesday	Wednesday
room1			
room2			

13.11 Useful little things

13.11.1 Tables

- **Centre tables' captions** , works for figures and tables

```
\usepackage[center]{caption}
```

- **Set space between columns**

```
\setlength{\tabcolsep}{2.8pt}
```

- ...

13.11.2 Horizontal lines in a page

Like this:

"An important quote here should be placed."

---

or even like this

Explain what "I am hungry" means in Portuguese/Spanish if translated literally, i.e if translated to "Eu sou esfomeado"/"Yo soy hambriento".

I used, respectively these two pieces of code:

```
    \hrule
    ``An important quote here should be placed.''
    \hrule

    %needs the definition before.
    \def\Vhrulefill{\leavevmode\leaders\hrule height 0.7ex depth \dimexpr0.4pt-0.7ex\hfill\kern0pt}

    \noindent\Vhrulefill \ \bb{or even like this} \Vhrulefill

    Explain what "hello" means.

    \noindent \Vhrulefill
```

13.11.3 Others

- \appendix - used to start an appendix. Right after that, put \section and a label inside the section and then the appendix will be referenced with a letter.
- This is a HUGE one for the use of LaTeXand VS code. You can type "\be equa" letting the autocomplete do the work and then TAB your way into the scope! It doesn't get faster than that.

13.12 How to Debug LaTeX

1. Isolate the Error. If you can comment all the code where you are sure the error is not it, do so. If you have many included parts, including only the one with the error is quite a good measure as well.
2. Open the Log. ".tex:" and line number will tell you the error EXACTLY. What usually happens is that the error you see is the lack of references or something because as a consequence to the problem, the .aux files generated a problem as well. **The actual error description and the precise place where to find it is above! You might as well start in the beginning of the log.** There also is a log file where you can Ctrl + F.

14 Python

14.1 Important Concepts

Python is a language with many characteristics, ones more obvious and easy to understand, like in-fix notation, others are more complex, like Dynamic Typing and Automatic Memory Allocation.

Moreover, many programming principles like anonymous functions, map, zip are important and should be addressed in order to achieve flexible and overall good programming skills.

Beforehand, check the Python Notebook and the slides from Rodrigo Ventura in the "Additional Material" Folder. They are very complete and give a proper insight on how everything works.

Tuples are faster than lists. So, if the idea is doing a very big iteration where there's no memory problem, converting to tuple before probably helps.

However, regarding speed, always use built-in functions and libraries! Almost always what is slowing the program down is a huge for loop... Libraries are compiled already and are very optimised, therefore by using them you are probably running very optimised C code, which couldn't be faster than it is.

14.1.1 Lambda and Anonymous functions

Lambda is the keyword used to make an anonymous function. The following 2 pieces of code do exactly the same.

```
    def my_key(x):
    return x[0]

    l.sort(key=my_key)


    OR

    l.sort(key = lambda x: x[1])
```

And a short example is:

```
    -----short example on how cool python is-----
    def make_multiplier(factor):
    return lambda x: factor*x

    f = make_multiplier(2)
    -----------------------------------------------
```

Therefore, lambda is nothing more than defining a function, without giving it a name. It helps keeping the code simple specially when the function is just going to be used once.

14.1.2 __main__

This can be perfectly understood in the following stackoverflow post .

In a nutshell, when a script is executed python assigns many names to certain variables like __main__. Is the script is executed in the terminal, then main will be the name of that script (without the .py). The *if __name__ == '__main__':* aims to distinguish between situation where the script was imported, so that the functions it contains can be used elsewhere (in this case the main name won't be the name of the script which is stored in __name__) and the situation where the script is directly called in the terminal. If the script is directly called, for it to do something, something has to execute and that's usually what goes inside that if. When the script is imported, most of the times only the definitions matter and the calls will be made in the script that is calling that one.

Therefore, is a good tool in case you want to make a script importable but also callable.

14.2 Some useful tools

To check if a variable points to a certain data type:

```
    isinstance(var, [list, tuple, int])
```

To define functions with optional arguments and call them in the incorrect order:

```
1
2       def draw_point(x, y, color='red', thickness=2):
3           print('x =', x, 'y =', y, 'color=', color, 'thickness=', thickness)
4
5       x = 1
6       y = 2
7       draw_point(x,y,'blue', 5)
8
9       draw_point(x,y,thickness=2, color='blue')
```

14.2.1 Unpacking Argument Lists

Unpacking Argument Lists

A very useful trick to pass many arguments at once.

14.3 Anaconda

Basically, it is the Python distribution. Because Python is so big, with so many packages and Python development is becoming quite big, a program to install things was created.

Go to the official Anaconda website and download the installation script for Python 3 and for Linux: anaconda.com/distribution

Install Linuxize - How to Install Anaconda on Ubuntu 18.04

14.3.1 Package Manager

User's Guide for Package Manager

Has instructions on how to install non-conda packages and many other useful things. It allowed the installation of streamlit!

14.3.2 Broken Jupyter

Tornado 6.0 breaks jupyter notebooks. Is required to uninstall it through pip and/or pip3 and through anaconda! Downgrade to 4.5.3. (This subsubsection will probably be removed in the future when it's fixed.)

14.3.3 Other

(base) Problem Ask Ubundu (base) in terminal - then close and open the terminal to take effect.

14.4 Pandas

Check this link :)

14.5 Jupyter Notebooks

Along with Anaconda comes a full installation of the most recent python version and the Jupyter Notebooks, which are awesome to write python.

Here are some very useful shortcuts to tame that beast:

There are 2 modes of shortcutting:

- The Command Mode (when border of the cell is blue). Press ESC to access this mode.
- The Edit Mode (when border of the cell is green). Press ENTER to access this mode.

The Edit Mode is clearly superior:

- Ctrl + Enter to run cell
- Shift + Enter to run cell and get directly to the next
- Ctrl + D to delete a whole line
- Ctrl + arrows jumps words (usual)
- Ctrl + Backspace deletes whole word (usual)

The Command Mode can be usefull sometimes, especially when adding cells is needed, but remember that the cell has to have a blue border:

- A - insert cell before
- B - insert cell after
- DD - delete current cell
- Z - undo cell deletion
- M - markdown input type (Check: Netbook MarkDown)

14.6 Spyder

Jupyter Notebooks is very fun for prototyping, however, to hard debug and develop some complex stuff, Spyder is the tool.

Install Anaconda. It is the absolute best way of getting everything up to date and running smoothly with the least complications. Check here for the complete tutorial on how to do so (very easy).

After installing Anaconda, check the Official Spyder Releases github page and install the most recent one. **Also remember never to run pip install spyder or similar or it can completely break Anaconda installation** .

Some useful shortcuts in Spyder:

- Add # % % to separate cells.
- Ctrl + Enter - Run cell
- Alt + Ctrl + Enter - Debug cell
- Ctrl + Shift + I - Go to Console
- Ctrl + Shift + E - Go to Editor

Note: spyder debugger is fucking shit. You can't get comfortably inside functions. VS Code does the job fairly well for the time being.

14.7 Keras - A powerful API for TensorFlow

Keras is simply the way of doing neural networks. It can't get easier than that! TensorFlow was created by Google and PyTorch, it's rival, was created by Facebook. Not only is Google a better company in terms of the use of NN to do things, but this website also agrees that TensorFlow 2.0 is a complete game changer since it also declares keras as it's main API, therefore making it very very easy to do complex computations.

Overall, two things are fundamental to have success with these tools:

1. To understand the functions and what all options mean, is necessary to have many mathematical expression in the head
2. Have a very good control of the API - Keras. This section will take care of this step, trying, as best as possible, to explain the first step as well.

However, learning how to use the api can take a while. Following are the main parts of a NN.

14.7.1 Basic Flow - Image Classification Example

Imports, data load, normalization, on mean and on stddev(not included in the example), create model, add layers, check summary, configure callback(s) and the optimizer, compile the model to apply the loss function and the optimizer, fit the model, predict with the model, plot training and validation loss curves, plot confusion matrices and accuracy scores opposing the predicted data with the actual test data outcomes.

```
import seaborn as sns
import numpy as np
import matplotlib.pyplot as plt
import tensorflow.keras as keras
from keras import utils, layers, models, callbacks, optimizers
from tensorflow.keras.layers import Conv2D, MaxPooling2D, Flatten, Dense
from sklearn.metrics import confusion_matrix, accuracy_score

x = np.load("mnist_train_data.npy")
y = np.load("mnist_train_labels.npy")

x_test = np.load("mnist_test_data.npy")
y_test = np.load("mnist_test_labels.npy")

print(x.shape)
print(y.shape)

plt.imshow(np.squeeze(x[5]))

#normalizing
x_norm = x / 255
x_test_norm = x_test / 255

#y to categorical
labels = keras.utils.to_categorical(y, num_classes=None)

model = keras.models.Sequential()

model.add(keras.layers.Flatten(input_shape = (28,28,1)))
#in case the samples are one dimensional, Input layer is enough.

#after the first layer it is not necessary to specify the input size
model.add(Dense(64, activation='relu'))
model.add(Dense(128, activation='relu'))
model.add(Dense(10, activation='softmax'))

#Expected: Layer 1 weights: (784+1) * 64 = 50240
#         Layer 2 weights: (64 +1) * 128 = 8320
#         Layer 3 weights: (128+1) * 10 = 1290
#                            Total  = 59850
model.summary()

"""
Example of CNN:
model2.add(Conv2D(16,kernel_size=(3,3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2,2)))

model2.add(Conv2D(32,kernel_size=(3,3), activation='relu'))
model2.add(MaxPooling2D(pool_size=(2,2)))

model2.add(Flatten(input_shape = (5,5,32)))
model2.add(Dense(64, activation='relu'))
model2.add(Dense(10, activation='softmax'))
"""



stopper = callbacks.EarlyStopping(patience=15, restore_best_weights=True)
#restore_best_weights - restore weights from best epoch

# min_delta may me zero because the epochs after the best only count if there's improvement

Adam = optimizers.Adam(lr=0.01, clipnorm = 1)

model.compile(loss='categorical_crossentropy', optimizer=Adam)

hist1 = \
    model.fit(x = x_norm, y=labels, batch_size=300, epochs=400,\
        verbose = 0, callbacks = [stopper], validation_split=0.3)

y_pred_labels = model.predict(x_test_norm)

y_pred = np.argmax(y_pred_labels, axis=1)

plt.plot(hist1.history['loss'])
plt.plot(hist1.history['val_loss'])
plt.title('Model loss')
plt.ylabel('Loss')
plt.xlabel('Epoch')
plt.legend(['Train', 'Test'], loc='upper left')


```

```
sns.heatmap(matrix,annot=True,cbar=True)
plt.ylabel('True Label')
plt.xlabel('Predicted Label')
plt.title('Confusion Matrix')

print('Accuracy:', accuracy_score(y_test, y_pred, normalize=True)* 100, '%')
```

14.7.2 Sequential Model

There are Sequential and Functional Models. Functional is when the mess it too big. Very likely not be needed since Sequential can do much more that I know how to do, including image analysis, convolutional and recurrent NN. To learn the basic steps of it: Keras-Getting started with the Sequential Model And to learn the detailed methods of it and their functions: The Sequential Model - all steps

14.7.3 An optimizer

An optimizer is one of the two arguments required for compiling a Keras model - optimizes the casual gradient descent algorithm to achieve a faster convergence, i.e to find the minimum faster. We've seen a few methods on adaptive step size and momentums.

```
    model.add(Dense(32, input_dim=784))
    model.add(Dense(32, input_shape=(784,)))
```

They are equivalent and both simply mean: Add a fully connected layer with 32 outputs and 784 one dimensional inputs.

14.8 Plotting

Very similar to MATLAB, but different.

It all depends on all complex you want to go.

Way 1:

```
    plt.subplot(1,2,1) #n_rows, n_cols, index
    plt.plot(x,y) #or sns.heatmap...
    plt.xlabel("hey")
    plt.ylabel("no")
    plt.title("yello")
```

Way 2:

```
    fig, (ax1, ax2) = plt.subplots(1, 2, figsize=(15,5))
    fig.suptitle('Confusion Matrices comparison')

    sns.heatmap(matrix,annot=True,cbar=False, ax = ax1) %#ax1.plot(x, y)
    ax2.set(xlabel='True Label', ylabel='Predicted Label')
    ax1.set_title('With EarlyStopping')

    sns.heatmap(matrix2,annot=True,cbar=False, ax = ax2)
    ax2.set_title('Without EarlyStopping')
    ax2.set(xlabel='True Label', ylabel='Predicted Label')
```

Way 3 - a great midterm between the other two:

```
    f = plt.figure(figsize=(10,3))

    ax1 = plt.subplot(1,2,1)
    sns.heatmap(matrix,annot=True,cbar=False, ax = ax1)

    ax2 = plt.subplot(1,2,2)
    sns.heatmap(matrix2,annot=True,cbar=False, ax = ax2)
    plt.title("helo")
```

In case is necessary to have plots of different sizes

14.9 Artificial Intelligence: A Modern Approach - Search Configuration

Clone the 3 repositories below.

Aima Code

Aima Data

ipythonblocks GitHub

Then put the ipythonblocks.py and the data folder inside the aima-python folder. That way the Jupyter notebooks should work

14.10 From Python 2 to Python 3

In Linux, if 2to3 is installed, one may simply run:

```
2to3 -w -n file.py
```

This will write the file translated into python 3 in the same file(-w) and without creating a backup(-n).

14.11 Good Practices for Python Code

This section is irrelevant if you only code alone for yourself. In case someone else will see your code, then you should write it as everyone else likes it. You'll probably agree with most if not all of these guidelines.

Style Guide for Python Code

Some of the ones I tend to violate the most:

- Variable names with underscore (underline)

15 Linux

15.1 Linux Essentials

15.1.1 Pipe

List installed packages

```
sudo apt list --installed | grep -i apache
```

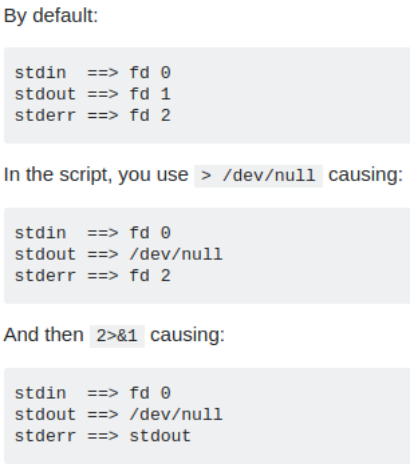
grep -i filters without concerning about the case.

15.1.2 grep

15.2 Redirect with ¿

One can redirect the stdout and the stderr to other files with that little “bigger than ” sign.

```
xdg-open ~/Pictures/1.png > /dev/null #will redirect the stdout to null.
xdg-open ~/Pictures/1.png &> /dev/null #will redirect everything to null.
xdg-open ~/Pictures/1.png > /dev/null 2>&1 #will perform the above and redirect stderr to stdout.
```



15.3 Shortcuts or Link [ln]

ln -s /Docs/importantFolder . will create a shortcut in the current folder for the importantFolder.

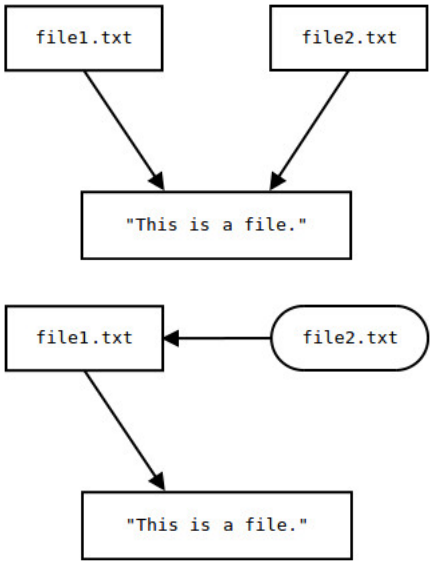
However, there’s much more to links that what meets the eye!

A link is an entry in your file system which connects a file name to the actual bytes of data on the disk.

The previous Redirect ¿ command does this exactly. Actually, doing ln test1.txt test2.txt will link them both and they become the same file with different names ¿¿ allows for line appending! Try to append a line to one and check the other! echo ”hey” ¿¿ test1.txt cat test2.txt

We can delete test1.txt but test2.txt still exists, therefore the data is not forgotten. If there isn’t any link to the information, then it becomes useless and is marked as available - to be writable over eventually.

What is the difference between an hard link and a symbolic link (the -s option in the command)?



The main difference is that the symbolic link point to where a link is, not to the exact location. Like a pointer to pointer. If we delete the actual hard link to where it points, it can no longer access the file.

This should change, even if slightly, the perspective on Linux filesystem. Without the link, nothing can "hold" in memory, it is simply lost and the space is marked as available.

Also, it can save quite a bit of verbose while changing directories in the terminal because a symbolic link will get you to the same location.

What a great site this one proved to be: Computer Hope uln

15.4 How to build from source

1. Be sure you have everything needed:

```
sudo apt-get install build-essential
```

2. Download source. May come in various *.tar* formats. The flags **xvf** mean e**x**tract to a **f**ile with **v**erbose. When not specified, the file will have the same name. Then flag **z** for *.gz* or flag **j** for *.bz2*. If the extension is *xz* then no flag is needed because GNU tar recognizes the format by default. The code should be one of the following:

```
tar -xzvf file.tar.gz
tar -xjvf file.tar.bz2
tar -xvf file.tar.xz
```

   Note: sudo apt install xz-utils if tar gives error.

3. Check dependencies, i.e if you have everything necessary to build that package. It may require libraries specific to that package. Do the following code or check the README or INSTALL files.

```
./configure
```

4. Now various paths are possible. First of all, read the README and/or the INSTALL to look for instructions.

15.5 How to change Permissions and Ownership

Sadly, this happens very often. Anytime something like "permission denied" is presented in the terminal or similar but anywhere else, it is because permissions and ownerships are not set the right way. If they were, the owner of the file would be able to do what he wants with it.

First, what the permissions mean?

- **read** restricts or allows viewing the directories contents, i.e. ls command

- **write** restricts or allows creating new files or deleting files in the directory. (Caution: write access for a directory allows deleting of files in the directory even if the user does not have write permissions for the file!)
- **execute** restricts or allows changing into the directory, i.e. cd command

4 is read. 2 is write. 1 is executing. Therefore 6 is r+w, 7 is r+w+x-.

Additionally, running **ls -l** one gets something like **-rw-r–r–**. Each 3 refer to Owner, Group or Other. In this order.

Is possible to do the command in 2 different ways. The quickest is just using the numbers. One number per group. Like this:

```
chmod 754 filename
```

This way, the owner can rwx, group rw and other people/users can only read the file. Note: 770 is possible to not even allow the files to be discovered by other users.

Now we know how to alter what certain groups can do to a file. However, if the files doesn't belong to the correct group, those permissions changes may not do much regards the "permission denied" problem. Therefore, how to change ownership?

With **ch**ange **own**er duh!

```
sudo chown owner_name:group_name filename
```

A little small note: **whoami** gives the user name. Alternatevely, **users** and **groups** give the users with initiated sessions and the groups the current user belongs to.

To finalize, why running **./script.sh** and running **bash script.sh** require different permissions?
The first one needs executing permissions and the second just needs read permissions. This is because the first we are saying to execute the shell script. In the latter it is said to 1º load bash, then give "script.sh" as an argument to it. Therefore, bash just needs to read what is written in the script.sh file.

Running scripts as ./script.sh is a more general way as it also works with ./script.py as long as the script has a *hashbang* (#!)inside to tell the program loader what should be used.

15.6 Formatting a partition as exFAT

exFat is readable in every computer. In linux is necessary to install some utilities:

```
sudo apt-get install exfat-fuse exfat-utils
```

First: HOW TO MANAGE PARTITIONS?

Ans: install gparted, it does a good job.

Second: HOW TO FORMAT AS exFAT?

Ans: gparted can't do this... supposing the devices has only one partition (all partitions can be deleted in gparted to achieve this) and is called mmcblk0 in the devices directory.

To list all connected storage devices

```
sudo fdisk -l
```

To wipe the filesystem (almost the same as gparted does):

```
sudo wipefs /dev/mmcblk0
```

To create only one partition. Note: in the fdisk utility is necessary to confirm several intermediate steps, because of that "all enters" means to press enter until other instruction in the normal utility prompt is asked.

```
sudo fdisk /dev/mmcblk0
n
"all enters"
t
7
w
"all enters"
```

I am pretty sure until lhere it could be done in gparted as well. But now comes the important part! After creating a partition, when doing sudo fdisk -l, there is a partition in the device we want to format. This is the partition we will format as exFAT. That partition in this case is called mmcblk0p1 (makes sense right?). Do the following to format is as exFAT:

```
sudo mkfs.exfat -n hardDisk /dev/mmcblk0p1
```

hardDisk is nothing more than the name you want to call the partition, doesn't matter.

You can check everything went fine with:

```
sudo fsck.exfat /dev/mmcblk0p1
```

Congratz, done :)

15.7 Install Custom ROM with Linux

The custom ROM is installed in the android device, as I hope you remember...

The major difference in the processes between Windows and Linux is in the initial part. After the initial part concerning installing TWRP in the phone, the rest is trivial. In fact, I've just noticed that in Windows is necessary to write several commands as is required in Linux, so by covering the part in Linux the might be some transferability to Windows.

The steps necessary (general, independent of Linux or Windows):

1. Get developer options
2. Install adb and fastboot utilities
3. Backup (optional)
4. Allow OEM (Original Equipment Manufacturer) bootloader unlocking
5. Boot into fastboot
6. Unlock bootloader
7. Install TWRP (Team Win Recovery Project)
8. Put ROMs, Gapps and Magisk in an SD card and clean & dirty flash as much as you like.

The difference between the two OS's is on how to perform the step 2 and step 6 may be marginally different also. Regarding step 2, Windows has an executable that lets you do everything at once while Linux has to be more paced and do everything separately. Regarding step 6, one is done in the terminal (wanna guess which?) and the other has a somewhat nice gui.

Maybe it wouldn't be such a bad idea do use a windows virtual machine for this actually... But oh well.

NOTE : If the phone can't transfer files to the pc, don't even start this! On linux MTP must be enabled and installed. Enabling can be done by simply pressing the pop up on the phone to MTP(Media Transfer Protocol) and installing on Linux:

```
sudo apt install mtp-tools libmtp mtpfs
```

ADB (Android Debugging Bridge) is the utility that allows a connection with the phone with a considerable degree of permissions, for debugging purposes.

After this, in order:

1 Enable Developer Options

Go to settings, system, tap 7 times in the build number of the phone.

2 Install ADB and fastboot utilities

In Linux, this is the hardest step! A list of the steps:

1. Go here to download the latest abd tool.

2. If the above doesn't download properly, is possible to get it from the apt repository from android-tools-adb. If it works, do one more thing to be installed system wide: change path to this adb tool by adding to .bashrc the following:
   export PATH=${PATH}:/home/joao/plataform-tools/adb
   , considering the extracted file was placed in my top user folder.
3. Run: sudo apt install android-tools-fastboot.
4. Now adb is "installed" and fastboot too.. If you want to make a FULL backup of your phone - worth it because TWRP will delete everything - do the following: **suicide** .

Obviously kidding, but this is uncertain territory. Now is necessary to be sure everything is 100%. The following set of procedures did work for me after trying TOO many things:

1. don't turn on usb debugging still and don't connect the phone yet to the pc: DELETE .android from home!
2. revoke all usb debugging authorizations just below that option (on phone)
3. kill adb server with adb kill-server
4. connect phone to pc, enable MTP on phone and then enable usb debbuging
5. start adb server on pc: adb start-server
6. with "adb devices" the output should now be "number device" instead of "number unauthorized"

3 Backup (optional)

If you still are with me, then we might be able to backup things now! The following will allow a complete backup to be placed in the Documents folder: (confirmation on phone required and a password for encryption can be used)

```
adb backup -apk -shared -all -f ~/Documents/backup.ab
```

To get everything back, the following can be done to restore the backup: (again confirmation needed and password requested if used before)

```
adb restore ~/Documents/backup.ab
```

4 Enable OEM Bootloader unlocking

Go to general settings again, developer options and activate: OEM unlocking and Advanced reboot.

5 Boot into fastboot

Reboot into bootloader/fastboot. Setting the advance reboot option makes this much easier. The other way would require booting into recovery and from recovery to fastboot.

6 Unlock Bootloader

Check if the fastboot is working:

```
sudo fastboot devices
```

If one shows up, cool.

Next, unlock bootloader with:

```
sudo fastboot oem unlock
```

Confirm on phone and then should be cool. Takes a while to boot back to the OS, but then all is fine.

7 Install TWRP

After booting to the OS, boot back to fastboot/bootloader with the restart method in number 5. It will be necessary to enable the previous options again.

When in fastboot, check if everything is fine

```
sudo fastboot oem device-info
```

On the line saying "Device unlocked" is should be "true". The other lines don't matter.

To install TWRP in the recovery partition, so that it runs when we try to recover, do:

```
sudo fastboot flash recovery recovery.img
```

8 Install custom ROMs, Gapps, Magisk and others

Now, put the ROMs you like in the SD card, Magisk and Gapps.

Some clarification on the above:

- ROMs can be downloaded by searching 'name_of_phone ROM
- Gapps is necessary with the given ROM doesn't come with google apps, like the playstore. I don't know if it happens always, but is very frequent. Can be downloaded here: opengapps.org. Note that is necessary to know which processor architecture your smartphone uses. Oneplus X is ARM.
- Magisk is the BEST way to rooting your phone and have full access to it inside the ROM. Can be downloaded here: magiskmanager.org and go to install magisk for non-rooted phones, which will give you a zip from github.

Some clarification on the differences between clean and dirty flash.
Clean flash is FORMATTING COMPLETELY EVERYTHING in TWRP, by doing Wipe-¿ format data and only after that installing something. This is only needed when flashing incompatible ROMs, which are all ROMs that are not direct patches from the previous one. ROMs for different versions of android (8/Oreo to 9/Pie) are not compatible and from 8 to 8.1 I have might doubts. Dirty flash is just erasing some cache and then flashing/installing. It can be done with simple programs that don't change all the things like installing a new ROM does.

Having an Oneplus X, a great developer for this phone is YumeMichi and he/she posts everything here: YumeMichi Repo.

15.8 Android Studio with Linux

The above section is related with this one because ADB is necessary to run apps in the phone quickly.

In practice, if "adb devices" detects a good "device", then is all good. this section is also related with the above because if you want to test on an older phone, let's say more than 3 years old, it may not be possible due to API constrains. Therefore, what I needed to do was installing a very stable custom ROM Omni, by YumeMichi, that can be found the repository mentioned above.

To get everything ready to be able to plug and unplug the phone from the pc and still be recognized by Android Studio is was necessary to disconnect the phone, put MTP has a default mode, revoke USB debugging authorizations, delete .android, open android studio (the current session can't have already tried to connect to the phone once or else it will lead to a adb loop), "adb kill-server", then connecting the phone WITHOUT usb debugging enabled, then enable usb debugging and wait a bit. During about 10 seconds wait, there can be a popup on the phone which represents great news. Mark as "always trust". Check "adb devices" and the devices should be "device" and not "unauthorized" or anything.

From here, android studio will almost certainly work.

From here on, I'll write EVERYTHING I learn about Android Studio. It is hard to structure the amount of random information I'll gather so my strategy will be not doing so. I'll just add titles with the objective task that I learned to perform and some related information.

Change text size

By adding:

```
android:textSize="25sp"
```

to activity_main.xml, near the part of "android:text="Hello World!" ", was possible to change the text size.

Additionally, by messing with some things on the design tab, layout window, was possible to adjust the text window to the text.

To end: WHAT'S THE DIFFERENCE BETWEEN SP AND DP?
One means scale-independent pixels and the other density-independent pixels. The density of pixels a screen has is know as dpi/ppi (dots per inch/ pixels per inch). Now the choice is: is better to disregard the various format rations of screens of their density, what changes more? The form factor changes a lot more and a lot more abruptly. So it influences the text much more and we should use sp for text almost every time. Also, the scale can change with the user given scale and be messed up. Using sp prevents that. (a better look at this topic wouldn't be a bad idea...): DP change from phone to phone, regarding the aspect ratio. SP shouldn't...right?

Java background

I've created a document with the essentials, very bare material of course, of Java. Additionally, I'll write in another google document the things I learn from experienced Java and Android programmers.

Git with Android Studio

This is one of the best things about Android Studio: you can go to VCS (Version Control System) and check your previous commits, commit and push, resolve merges, ect...)

15.9 Downloading videos from all over the web

The Linux way, with the terminal.

For this, a tool called youtube.dl.

```
pip install --upgradable youtube_dl
```

There are a TON of supported websites that can be consulted here.

Some usage examples:

- To downlaod description, metadata, annotations, subtitles and thumbnail

  ```
  youtube-dl --write-description --write-info-json --write-annotations --write-sub --write-thumbnail url
  ```

- for the best single file video and audio quality

  ```
  youtube-dl -f best url
  ```

  [language=bash, numbers = none]

- NOTE: the best quality may be really good... do you want 4k if you display if only FHD (1920x1080)? take the "best" out and choose the quality by number and just put it instead of the "best".

This topic came up because self english training required transcribing videos. From Ted is difficult to have the exact transcripts downloaded automatically, but they can be easily copied.

However, a good video player is key for this so that it becomes much easier to jump to different parts of the video.

15.10 MPV - The best video player

Download as usual, these will be some advices to work with it.

Check Manual for a complete and clear description of all commands.

For sound adjustments use 9 and 0, to decrease and increase the volume.

For previous and next frame, use , and . , respectively.

For controlling the speed use:

or {}. Backspace to revert the speed back to normal.

Ctrl + Shift + Left or Right to align subtitles in time.

To use subtitles is just needed to use the GUI.

All the rest is at the provided link.

15.11 Keybindings - Keyboard and Mouse

For the mouse: Configure mouse buttons to custom sets of commands - Use to detect only mouse button presses:

```
xev | grep -A2 ButtonPress
```

To detect anything, simply run 'xev' in the terminal.

Then do as the examples show, 'keydown Control_L', 'keyup', 'key z' for a short press. THEN: 'xbindkeys -p' to apply the .xbindkeysrc conf file.

And the Fn keys can't be used... Explanation here .

BEFORE : trying the same with the keyboard, like mapping the arrows to the middle of the keyboard, check Section 15.15.2 (clickable) .

For the keyboard, the answer is using xmodmap if what you really need it is chaning functionalities instead of adding them. If you need to add them, xbindkeys is the way to go. But I urge you, read 15.15.2 before doing anything, I've invested quite a bit of my life, speacially while I was the most busy, exploring no way out and very unefficient paths.

15.12 Linux Image Editor

If you want something like paint, don't bother. Use pixlr.com/editor/ .

15.13 Linux Video Editor & Instagram

Yes, one may use Instagram. If one does, one may need some tools. To resize photographs or videos, something that is needed in case one needs to upload multiple files at once, the kapwing resize tool does that perfectly.

Additionally, the general Kapwing Tool is very usefull for video and image editing. Quick and easy really.

15.14 Other Linux related stuff

- More than enough about bash here
- To download files from links: wget –user-agent="Mozilla" link
- To convert .ppt to .pdf with libreoffice installed:

```
libreoffice --headless --invisible --convert-to pdf *.ppt
```

- Matlab Continuous Distributions
- MATLAB Ubuntu fixNecessary for help displaying and no errors on terminal
- To search stuff on the terminal by name:
  find directory -name "filename", e.g: find . -name "error.png"
  Or add the following for files: find directory -type d -name "foldername"
- sudo rm /var/crash/* for removing error reports every time they appear, or else they will continue to appear every time the computer boots
- xdg-open . to open the file explorer in the terminal location. xdg-open helo.mp4 is will open with the default program the specified file.
- To create a File shortcut for Thunar, the beautiful file manager that XUBUNTU uses, visit this;
- **sudo service network-manager restart** To restart the network manager because it stops unexpectedly sometimes.

15.15 Linux Life Lessons

15.15.1 Wine and PlayOnLinux - Project: Kindle to PDF

Sometimes Linux is not ready for somethings. Or somethings are not ready for Linux. In either case, the things should be used in the correct place.

Lesson: Wine and PlayOnLinux are fun and are becoming increasingly less of a cancer, but for now... they still a very substantial cancer. So, if something is meant to be done in Windows, for the next 2 years, at least, do so.

It took me 10 minutes in windows to achieve what I couldn't after 4 hours in different days in Linux. Don't waste your life in stupid things like this.

The solution for achieve kindle to PDF convertion is below. Basicly, uses a older version of kindle to pc to download a file that is protected with a breakable cypher and is just needed to load a plug-in in a e-book reader called Calibre. Full instructrions are below and work flawlessly in windows. In Linux they don't: the plug-in loaded into Calibre is meant to run on windows, so it requires far more stuff and has too many windows dependencies, so it would take too much time to even have the possibility of making it work. Completly not worth the try.

Do this in windows and get the pdf after that.

Then you can use stuff like this to have a nice print out of it.

```
1    pdfnup --nup 2x1 --suffix test file.pdf
```

15.15.2 Keyboard keybindings

Know what you are looking for .

Simplify the problem.

Ask in Ubuntu Forums.

After more than a morning on this, the major difficulty with mapping keyboard keys is that most of them already have a mapping that completely interferes what any other function you want to give them! For instance, **Alt** is used when your mouse is f*cked and you need press menus in windows. **Control** is practically reserved for every application. Not all keys thought, but if you want to map the arrows to JKLI for instance, Ctrl is not a very good key to use because everytime you press an arrow it will count the Ctrl as a modifier of the arrow and jump a whole word, for instance. **Shift** similar. **Alt Gr** seemed to be the answer, but for some reason, it prints f*cked up characters when you try to use it... Combinations of these are out of question.

Then what other modifiers do you have that aren't being used, are close to your hands when you write and you never use them? **Caps Lock** ? Haven't you been paying attention to what I said: You have to remove the function that keypress has, or else it will still be identifier as a caps Lock.

Bottom line: if you can find a way of changing the event that a certain keypress sends, you can make it seem like that key is being pressed... For instance: when you press ctrl, you make it seem like you are presing Insert (that no one f*cking uses), then you can make combinations with Ctrl. **HOWEVER** you completely miss the ability of pressing control!

What you really want to do is changing the response to a set of key presses. If your keyboard is already taken, and most of those things don't serve any purpose, you want to change what happens when you press those things! **Alt Gr** is the perfect example! Do you know that when you press that cripple cousin of Alt that is Alt Gr, you end up with a modifier letter? Moreover, if you press Shift + AltGr you get yet another variation! **Here is the opportunity!**

You will change what happens when you press those combinations! Use **xmodmap** for this!

For knowing what you need to change, look this what for the key. The "h H h H" is just something that is uniquely in the line of the h key. Do this for every key you want to change.

```
1    xmodmap -pke | grep "h H h H"
```

From here the image below. You want to change that line because every key means what happens when it is pressed.

```
$ xmodmap -e "keycode <keycode> = <key1> <key2> <key3> <key4> <key5> <key6>"
```

```
<key1> ...              <Key>
<key2> ...        <Shift-Key>
<key3> ...          <Alt-Key>
<key4> ...    <Shift-Alt-Key>
<key5> ...        <AltGr-Key> (at least by Unity)
<key6> ... <Shift-AltGr-Key> (at least by Unity)
```

Step 2) Map the keys to your desire:

```
$ ##                        1 2 3        4         5      6
$ xmodmap -e "keycode 43 = h H NoSymbol NoSymbol Left   NoSymbol"
$ xmodmap -e "keycode 44 = j J NoSymbol NoSymbol Down   NoSymbol"
$ xmodmap -e "keycode 45 = k K NoSymbol NoSymbol Up     NoSymbol"
$ xmodmap -e "keycode 46 = l L NoSymbol NoSymbol Right  NoSymbol"
$ ##                        1          2          3         4         5
$ xmodmap -e "keycode 47 = odiaeresis Odiaeresis NoSymbol NoSymbol Home"
$ xmodmap -e "keycode 48 = adiaeresis Adiaeresis NoSymbol NoSymbol End"
```

The exact mappings can be different. I did something like:

```
joao@joaoPC:~$ xmodmap -e "keycode 44 = j J j J Left dead_hook dead_horn"
joao@joaoPC:~$ xmodmap -e "keycode 45 = k K k K Down ampersand kra"
joao@joaoPC:~$ xmodmap -e "keycode 46 = l L l L Right Lstroke lstroke"
joao@joaoPC:~$ xmodmap -e "keycode 31 = i I i I Up idotless rightarrow"
joao@joaoPC:~$ xmodmap -e "keycode 30 = u U u U Home downarrow uparrow"
joao@joaoPC:~$ xmodmap -e "keycode 32 = o O o O End oslash Oslash"
```

So, with AltGr JKLI I have the arrows and have the Home and End in U and O, respectively. And this way is not necessary to change any shortcuts anywhere, will work everywhere and is the fastest way possible of doing it.

The lesson to take away is that if something somewhat simple is already becoming a mess, you are probably asking the wrong questions and there is a much shorter and more pleasant path that you are not finding. Take a step back, look and think about the problem verbally... Ask a question in the Ubuntu forums because they sugest a load of different answers that most certainly will have your question!

One last thing: if you want to make everything "reboot proof", put everything into a script and run it at the beginning.

16 Database work - SQL

16.1 SQL commands

The basic syntax is:

SELECT {Column name } FROM {Table name} WHERE {condition}

From here, there are a lot of variety that can be added to be possible lot of flexibility.

There are also other types of commands, that I haven't used so I won't talk here. Search :)

16.2 Browsing Tool with Filters

To use filters while using DB browsing tool installed on linux from sqlitebrowser.org with:

```
1    sudo apt install sqlitebrowser
```

Consult their wiki on that, it can be found here .

Additionally, is possible to export the result of filters to a csv by selecting the table with the mouse, pasting it in the side window and selecting "Export", not forgetting to add the .csv extension to the file name.

One last important feature: is possible to import and merge tables: if there is a similar database with a table with the same name (and structure) is possible to open that database with the same program, press on the table and Export it to csv. Then when opening the database where we want to import a table or various rows to, we select import from the File menu and select the csv. Note that the columns names in that csv may be in the first row. If this happens, we must check the square box that says "columns in the first row" so that the table can be read appropriately. Then it will ask if we want to merge and the previous table with the same name as the imported table will have the new rows from the imported table. If we want to do some filtering, for instance, *if we just wanted to import certain rows* we can now filter the table, or we could've filtered the table that was imported before importing it.

17 Visual Studio Code: The Environment for Development

This whole document was created with LaTeX on Visual Studio Code.

I ran commands for installing the necessary LaTeX stuff, these can be seen below.

I installed vscode with the software center from the .deb package downloaded through their website and then installed 2 extensions. The first extensions is Latex Workshop from the extensions market inside vscode. The second I can't remember which one was or even if it did something...

```
sudo add-apt-repository ppa:jonathonf/texlive
sudo apt update && sudo apt install texlive-full
```

Some useful things to know:

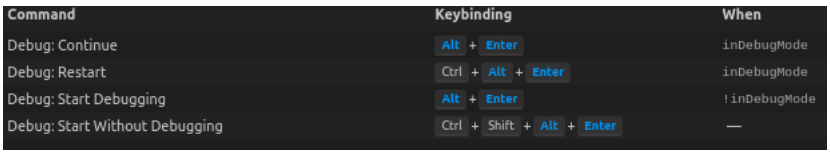
- F1 $->$ opens the command pallet. From there, type what you need :P
- Ctrl + S $->$ saves file. Sometimes compiles if everything is well defines(Latex for instances)
- Alt + Z $->$ word wrap
- Ctrl + . $->$ Compiles and allows for choosing the rules beforehand.
- Ctrl + Alt + V $->$ open on a window on the right the document that is being edited on the main window;
- Ctrl + Alt + H $->$ go to the same place on the LaTeXdocument as where the source is;
- You can commit and push and all that git stuff with vscode;
- Ctrl + Alt + A $->$ After selecting a word, adds it to user dictionary, therefore is not considered an error anymore.
- Ctrl + T + Ctrl + A $->$ Toggle Activity Side-bar (shows files, ect...)
- Ctrl + K + Ctrl + S $->$ Show all shortcuts (They can be edited but close and open between changes, they are not written right away - a bug)
- Ctrl + P $->$ Allows action taking!
  - :10 $->$ takes you to like 10
  - ? $->$ help on what you can do! probably much more than I know.
- Ctrl + Shift + P $->$ is the same as doing ¿ in the Action box, allows running commands.
- Ctrl + 1 $->$ go to the first windows when they are side by side. This is useful for Python Interactive Console or when editing side by side documents.
- Ctrl + K + Ctrl + ≥ 0 $->$ Fold all level ≥ 0 depth sections. \section is depth 0. \is depth 1, \subsubsection is depth 2 and subsequent blocks inside that will be the next depths. Similarly, with code: no indentation is depth 0, 1 indentation is depth 1, etc...
- Ctrl + K + Ctrl + J $->$ unfold all sections.
- Ctrl + K + Ctrl + L $->$ toggle fold in current section.
- Ctrl + , $->$ for settings
- Ctrl + Backspace $->$ delete full word

17.1 Using VS Code as an Environment for Debugging Python

Install the correct python package, the one from Microsoft and that should be enough

Add a ruler for character limit: Ctrl + Shift + P, write "setting json", add to the final line "editor.rulers": [80]

- F5 $->$ debugs the file
- Ctrl + F5 $->$ runs the file

- F9 − > toggle breakpoint
- Shift + F9 − > Go to Next breakpoint
- Ctrl + F9 − > Disable all breakpoints
- Alt + F9 − > Remove all breakpoints
- Ctrl + Shift + F9 − > Inline Breakpoint (only useful if it's a line with many commands)
- Ctrl + Alt + D − > Open Debug Mode (needs to be configure View: Show Debug)
- Ctrl + Shift + Y − > Debug Console
- 

18 GitHub

So far, what a pain. A globally used tool that takes so much to learn. Here's the short guide.

Create or clone

It depends who starts it. If you start it, you have to create it. If someone else already created it, then clone.

In the creation part, it can be done in the terminal, with some kind of GUI/in an IDE (Visual Studio Code and Android Studio do this very nicely). Or it can be done on a Git website which is has been necessary for me, due to terminal problems.

Create on Website. Works all the times, then clone it.

How Git works

Other important files

They are: gitignore, README, license...

GitIgnore is made to make git ignore certain files in the directory, so that those aren't included in the repository. Examples of those files are the .log files generated by compilations, build auxiliary files, etc...

About README, there's a whole website about this practice HERE. Normally, markdown is used to do this and that is a fairly easy language that can be fully consulted in HERE.

Another very good guide done on GitHub is Cheatsheet.

Regarding the license, it should be included if the code is to be used by others. A good example of a classical license is the MIT License. The website Choose a License explains everything perfectly.

18.1 Basics

merge this subsection with the above things.

18.1.1 Start a repository

Git global setup

```
git config --global user.name "João Morais"
git config --global user.email "joao.morais.jumper@hotmail.com"
```

Create a new repository

```
git clone https://gitlab.com/jmoraispk/test.git
cd test
touch README.md
git add README.md
git commit -m "add README"
git push -u origin master
```

Push an existing folder

```
cd existing_folder
git init
git remote add origin https://gitlab.com/jmoraispk/test.git
git add .
git commit -m "Initial commit"
git push -u origin master
```

Push an existing Git repository

```
cd existing_repo
git remote rename origin old-origin
git remote add origin https://gitlab.com/jmoraispk/test.git
git push -u origin --all
git push -u origin --tags
```

18.1.2 Pull

When you want to pull, two cases may happen:

Either there are no changes in you local repository to the last remote repository and you can simply pull and the changes in the remote repository are implemented in your local one. In this case, simply **pull**

Or, in case you made changes to your local files, a merges must occur! So do, **git stash** to put your changes safely in a stash, then **git pull** to overwrite your local repo. Then **git stash pop** to take you local changes out of the bag and attempt a merge. Usually, it goes well. If it doesn't check the merge section!

18.1.3 Merge

...

18.1.4 SSH key

"You won't be able to pull or push project code via SSH until you add an SSH key to your profile" Is it even necessary? It seems to work fairly well through https.

18.1.5 Delete a repository

It is hard on purpose!

To cut local updates is just to delete the ".git" file.

To delete the remote repo as well (usually not needed), it required to open GitLab/GitHub page, open project repository, settings, general, advanced and delete project.

19 Interesting stuff and People

19.1 ArcXiv

It's pronounce "archive" because the X is read as a χ (Chi, the greek letter).

arxiv.org has more than 1.5 million papers, including the paper published by **Grigori Perelman** , the man who cracked the first millenium problem rejecting the one million euros price and won a Fields medal rejecting the medal as well because he didn't want fame:

"After 10 hours of attempted persuasion over two days, Ball gave up. Two weeks later, Perelman summed up the conversation as follows: "He proposed to me three alternatives: accept and come; accept and don't come, and we will send you the medal later; third, I don't accept the prize. From the very beginning, I told him I have chosen the third one ... [the prize] was completely irrelevant for me. Everybody understood that if the proof is correct, then no other recognition is needed."

Overall, a place to read freely about what is and what once was the state of the art in science.

19.2 The writings of IST president

He writes well. Very well. After hearing him once talk for less than 5 minutes I did see why he was president.

Some of his writings in Public, one of the most well known portuguese newspapers, can be found in publico.pt/autor/arlindo-ol
.

Just in case he gets retired or something, his name is **Arlindo Oliveira** .

19.3 YIFY/YST release group

A release group so famous for their quality content, speed of delivery and that due to their websites taken down somewhat simultaneously.

Most of movie related piracy would still have their names attached. In quite a few cases, they don't have anything to do with it anymore.

In conclusion, search [YSF][YIFY] when you are searching a movie, it will be found much much faster.

Likewise, searching for yify subtitles will give great results very fast as well.

A good way of organizing movies would be: movie, jpeg with credits for release (yify jpeg), subtitle for yify and torrent source. Then ship all this to the drive.

About formats, in essence BLU ¿ WEB ¿ everyhing else. You can read everything about them in the Wikipedia: Pirated movie release types.

19.4 Interesting links

- https://www.sciencedirect.com/browse/journals-and-books a website with many scientific articles

20 Books

Every book I find interesting, reasons why I want to read it and what I thought after reading it.

20.1 Emotional Inteligence - Daniel Goleman

20.2 The Digital Mind - Arlindo Oliveira

20.3 Inteligência Artificial - Arlindo Oliveira

20.4 12 Rules for Life: An Antidote to Chaos - Jordan Peterson

20.5 Maps of Meaning - Jordan Peterson

20.6 Enlightenment Now: The Case for Reason, Science, Humanism, and Progress - Steven Pinker

20.7 The Better Angels of Our Nature: Why Violence Has Declined - Steven Pinker

20.8 The Beginning of Infinite - David Deutsch

20.9 How We Know What Isn't So - Thomas Gilovic

Another similar to this one is "Thinking, Fast and Slow" by Daniel Kahneman and both of them are important scientists that evaluate to what extent we (humans) don't make the logical decisions every time and what other factors influence ours decision making. We are probably not making the most rational decision because of those factors...

21 A few lessons

21.1 Be professional & make up your mind

Make your decisions. Sometimes in life you have to know what you want to do to make effective choices! Don't let the time pass, don't be a fucking passive cunt that only watches a mess unroll in front of you. Be there, be conscious of the decisions you are making and the impact they'll have and make them anyway because if you don't, you'll just get a random result and you'll probably piss-off and disrespect the people around you that are dependent on your decisions.

21.2 Insure properly

Insurance is the best way of risk transferring. Ensure until the money you would get back is preferable to the contents of the package. The money you are spending in the first place is probably completely irrelevant to make sure you get out of the situation winning.

This lesson came from insuring the package for 100 euro and paying for the shipping 33.5 euro while I could have insure it for 500 euro paying only 36. Of course the package got lost and I totally regret because it had important things that costed much more than that and that I valued much much more than that. Also I am in a phase I would like to buy a 500 euro headphones for motor cortex stimulation during workouts/trainings and I just sold a huge amount of hours of work in notes in TU Delft for 100 euro. Looking back, 500 euro wouldn't even be enough.

Be fucking sure you value your time properly.

21.3 Read

...