LTCI Data Science seminar

Steps, concepts and issues involved in providing learning guarantees in the Deep Learning scenario

Rodrigo Fernandes de Mello

Invited Professor at Télécom ParisTech

Associate Professor at Universidade de São Paulo, ICMC, Brazil

http://www.icmc.usp.br/~mello

mello@icmc.usp.br

November 29th, 2018

Basics

- The Statistical Learning Theory proves learning bounds for Supervised Learning Algorithms
  - That should also include the DL scenario

- That is based on the concept of Generalization

$$G = |R_{\text{emp}}(f) - R(f)|$$

Basics

- SLT relies on the Law of Large Numbers:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i - E(\xi)\right| > \epsilon\right) \leq 2\exp(-2n\epsilon^2)$$

- Assumptions:
  - The random variable must produce values in range [0,1]
  - The function must be independent from data
  - The joint probability distribution P(X x Y) must be static
  - Examples must be sampled in an i.i.d. manner

Basics

- So, from the Law of Large Numbers:

$$P\left(\left|\frac{1}{n}\sum_{i=1}^{n}\xi_i - E(\xi)\right| > \epsilon\right) \leq 2\exp(-2n\epsilon^2)$$

- The following can be ensured for a function independent from data:

$$P(|R_{\text{emp}}(f) - R(f)| > \epsilon) \leq 2\exp(-2n\epsilon^2)$$

- In which:

$$R_{\text{emp}}(f) = \frac{1}{n}\sum_{i=1}^{n}\ell(X_i, Y_i, f(X_i))$$

$$R(f) = E(\ell(X, Y, f(X))$$

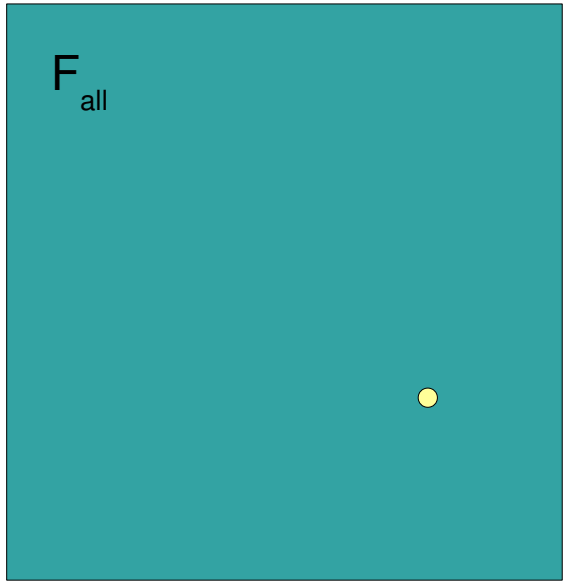
Basics

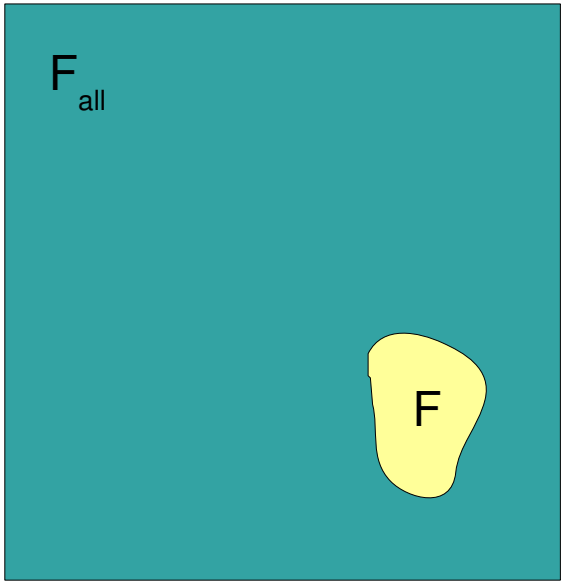
- So we could ensure that for every function inside a bias:

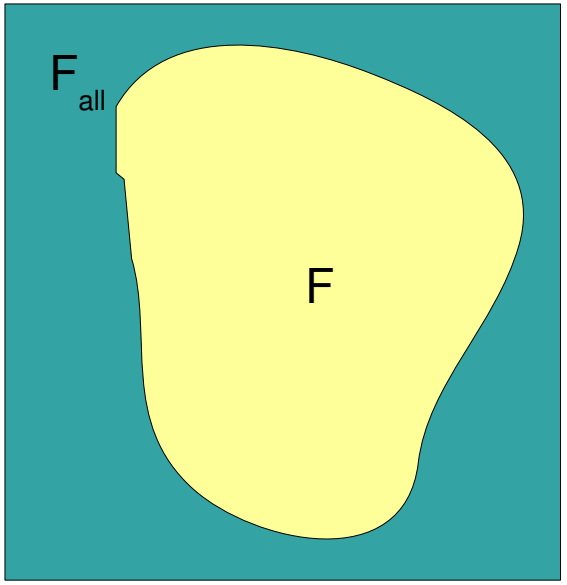
$$P(|R_{\text{emp}}(f) - R(f)| > \epsilon) \leq 2\exp(-2n\epsilon^2)$$

- Then:

$$P\Big(|R(f_1) - R_{\text{emp}}(f_1)| > \varepsilon \text{ or } |R(f_2) - R_{\text{emp}}(f_2)| > \varepsilon \text{ or } \ldots \text{ or } |R(f_m) - R_{\text{emp}}(f_m)| > \varepsilon \Big)$$







Basics

- So we could ensure that for every function inside a bias:

$$P(|R_{\text{emp}}(f) - R(f)| > \epsilon) \leq 2\exp(-2n\epsilon^2)$$

- Then:

$$P\Big(|R(f_1) - R_{\text{emp}}(f_1)| > \varepsilon \text{ or } |R(f_2) - R_{\text{emp}}(f_2)| > \varepsilon \text{ or } \ldots \text{ or } |R(f_m) - R_{\text{emp}}(f_m)| > \varepsilon \Big)$$

- What is bounded as follows:

$$P\Big(|R(f_1) - R_{\text{emp}}(f_1)| > \varepsilon \text{ or } |R(f_2) - R_{\text{emp}}(f_2)| > \varepsilon \text{ or } \ldots \text{ or } |R(f_m) - R_{\text{emp}}(f_m)| > \varepsilon \Big)$$

$$\leq \sum_{i=1}^{m} P(|R(f_i) - R_{\text{emp}}(f_i)| > \varepsilon)$$

Basics

- So, for all functions inside the space of admissible functions (a.k.a. bias), we will have:

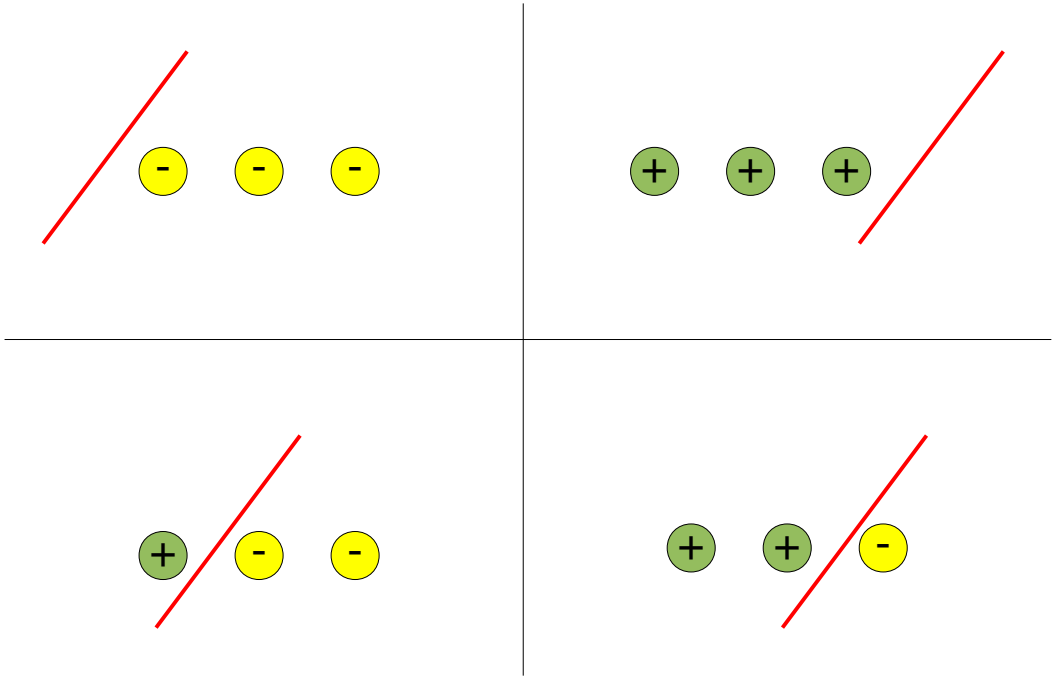
$$\sum_{i=1}^{m} P(|R(f_i) - R_{\text{emp}}(f_i)| > \varepsilon) \leq 2m \exp(-2n\varepsilon^2)$$

- Term m is not a constant but a function in terms o the sample size n

Computing the Shattering Coefficient

- For example, consider 3 points in a two-dimensional plane as follows:

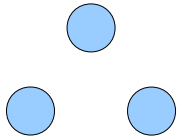
- Suppose linear functions are used to form classifiers:



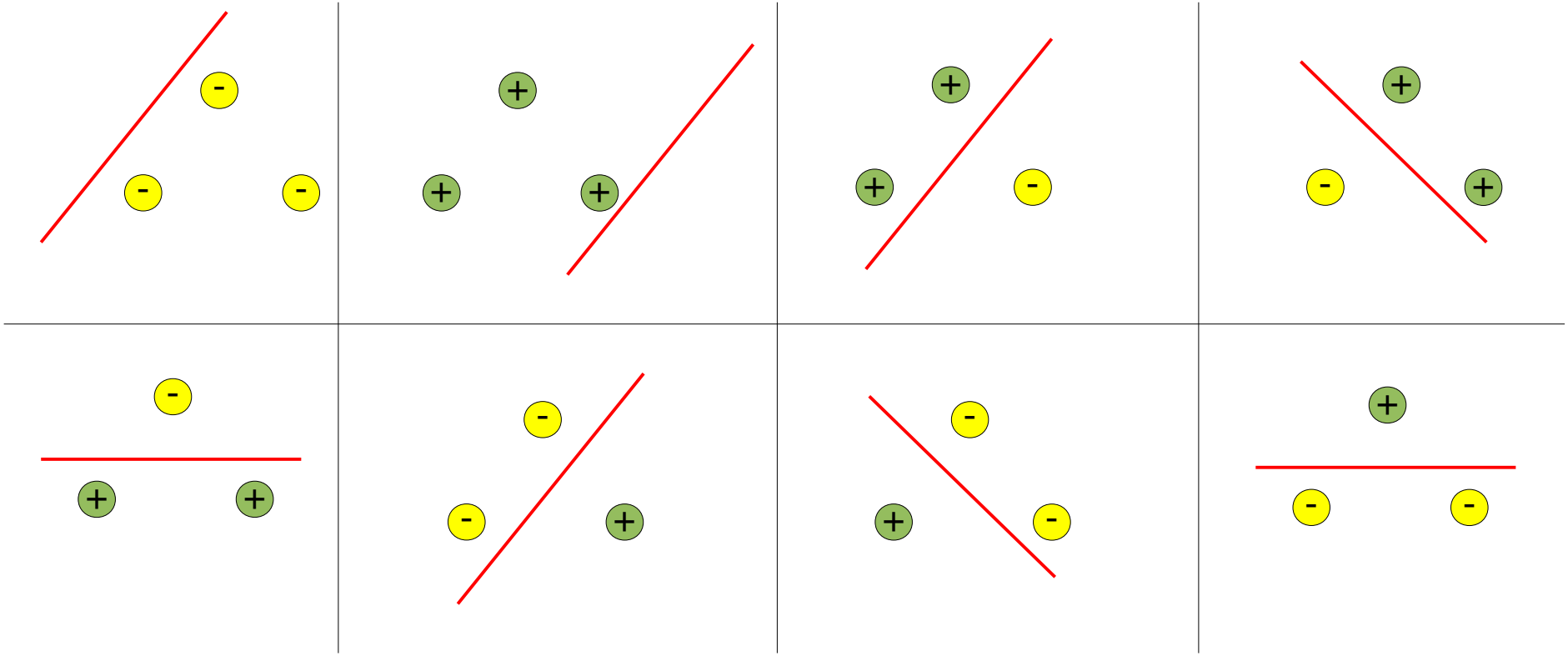
- We could shatter this sample in 4 different ways
  - But is there any other 3-point sample that we could shatter in more ways?

Computing the Shattering Coefficient

- Suppose we have the 3 points in different setting (still in R^2):



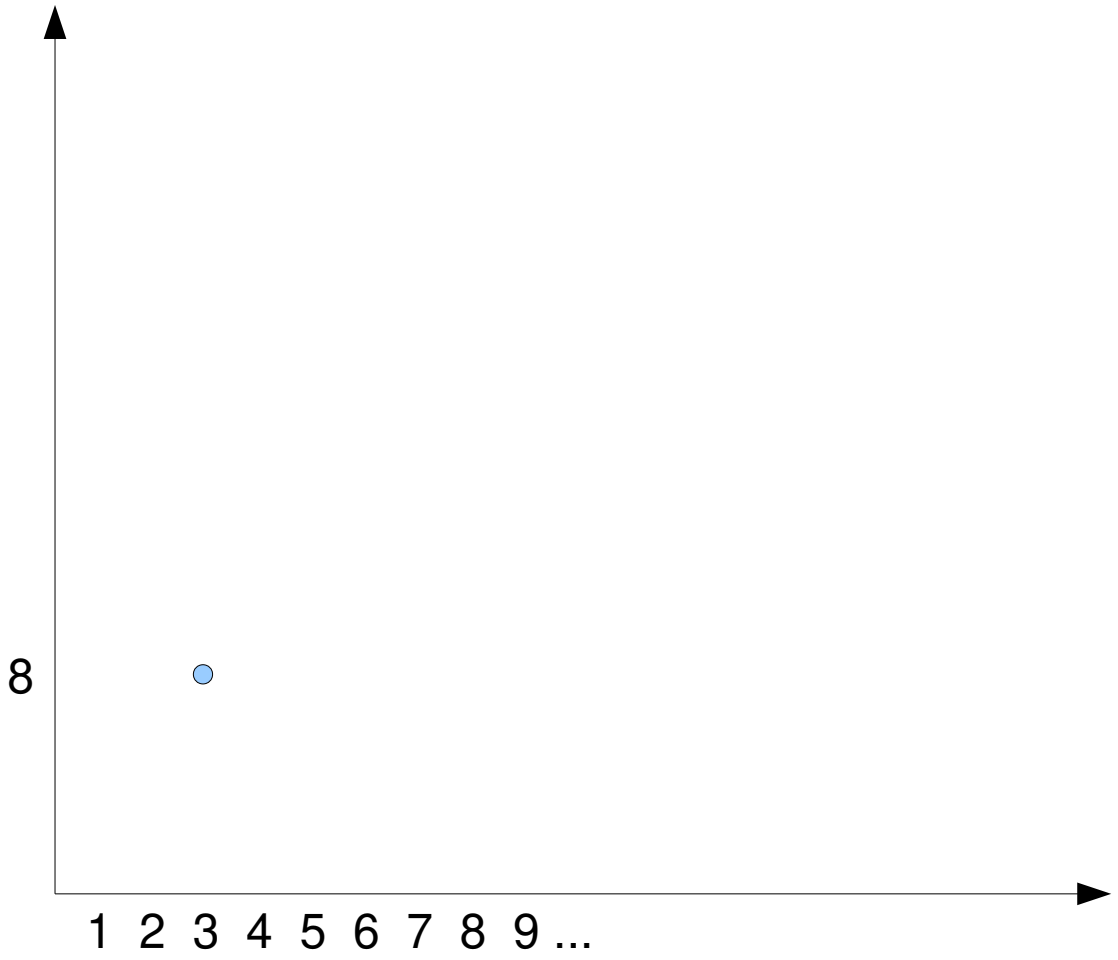
- Again consider F contains all linear functions:



- Observe F was capable of shattering this sample in all 2^n possible ways, what take us to the fact that F has a VC dimension at least equal to 3
  - Because there is at least one sample with 3 instances that can be shattered in all possible ways

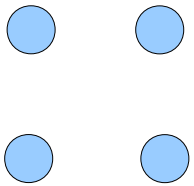
Computing the Shattering Coefficient

- In that sense, we conclude that for R^2:

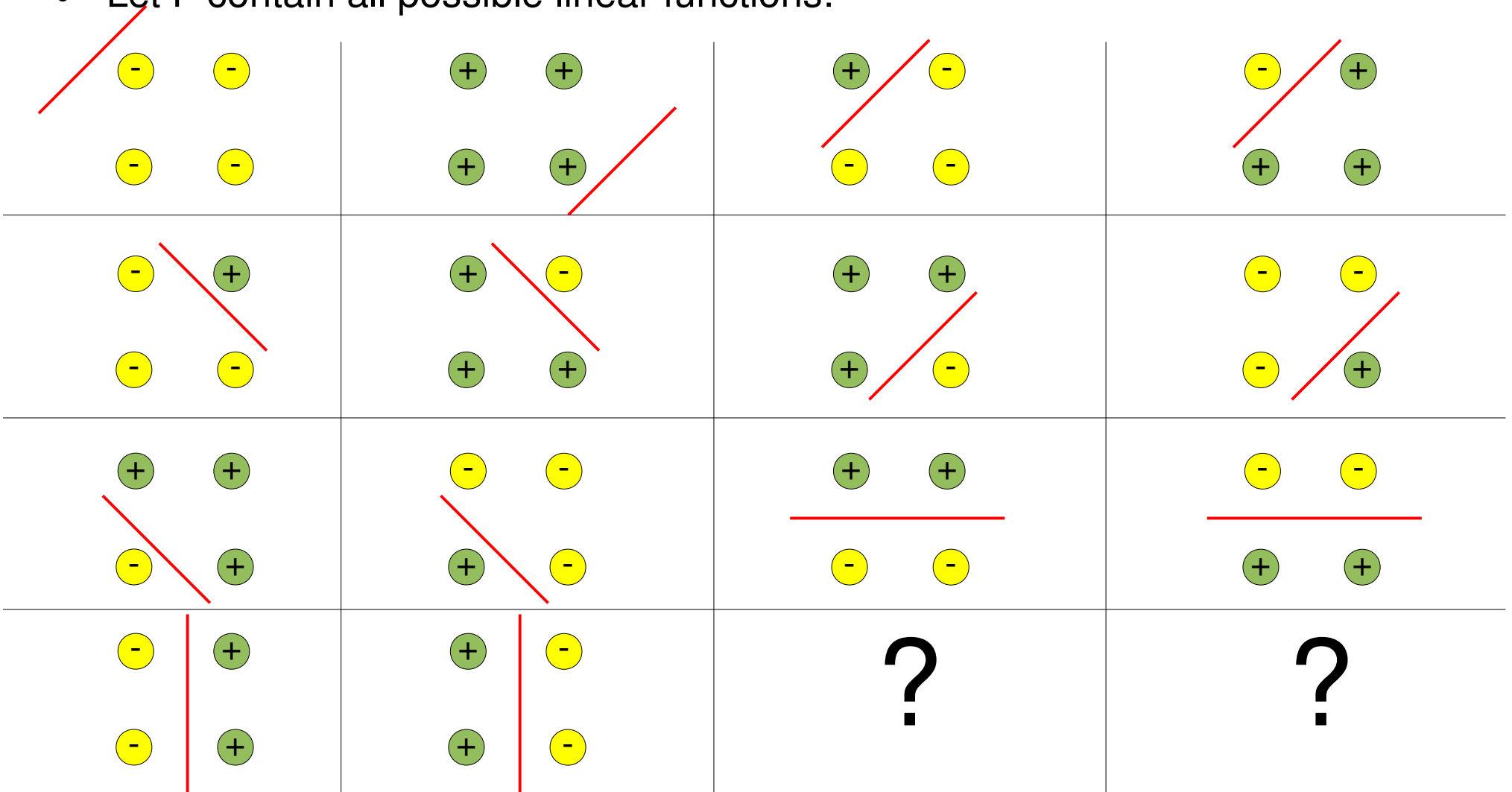


Computing the Shattering Coefficient

- Let's still consider in R^2:

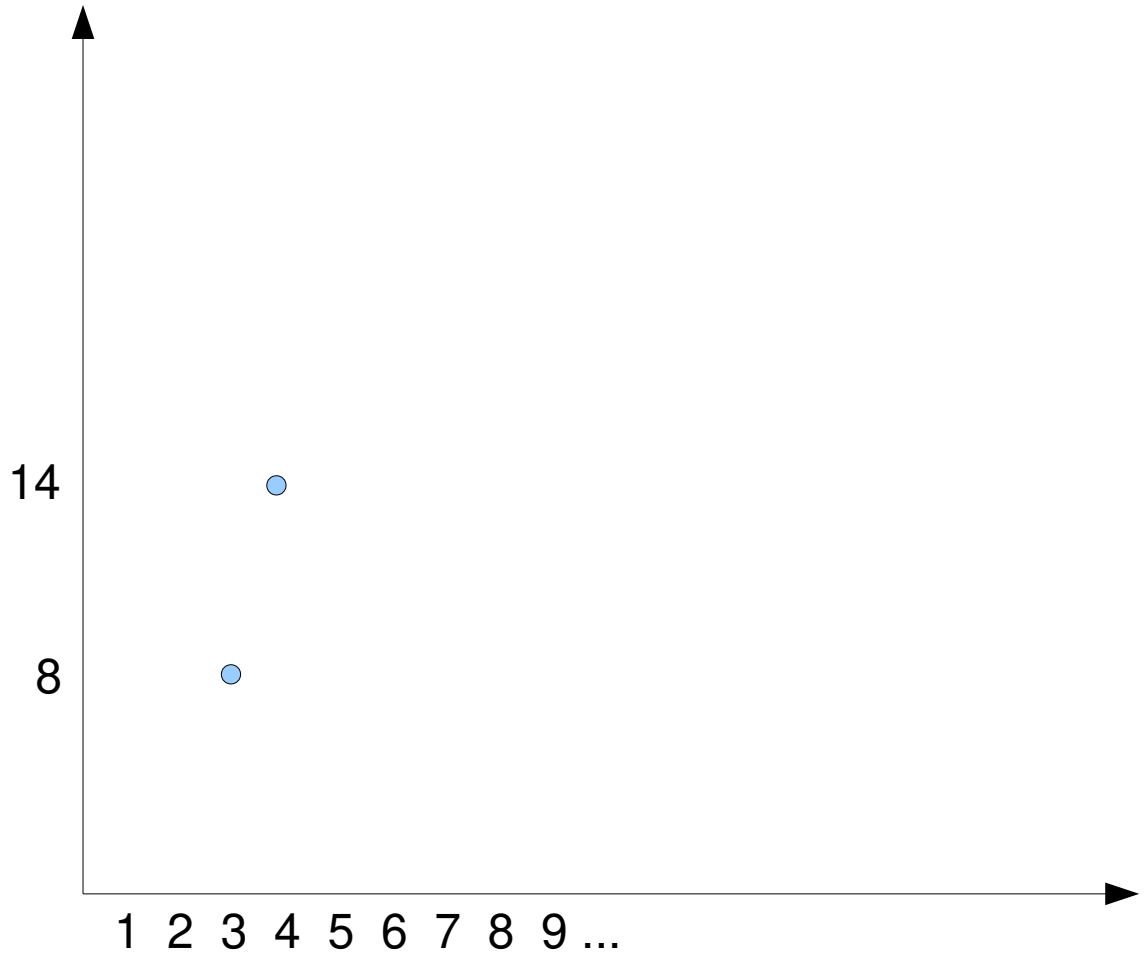


- Let F contain all possible linear functions:



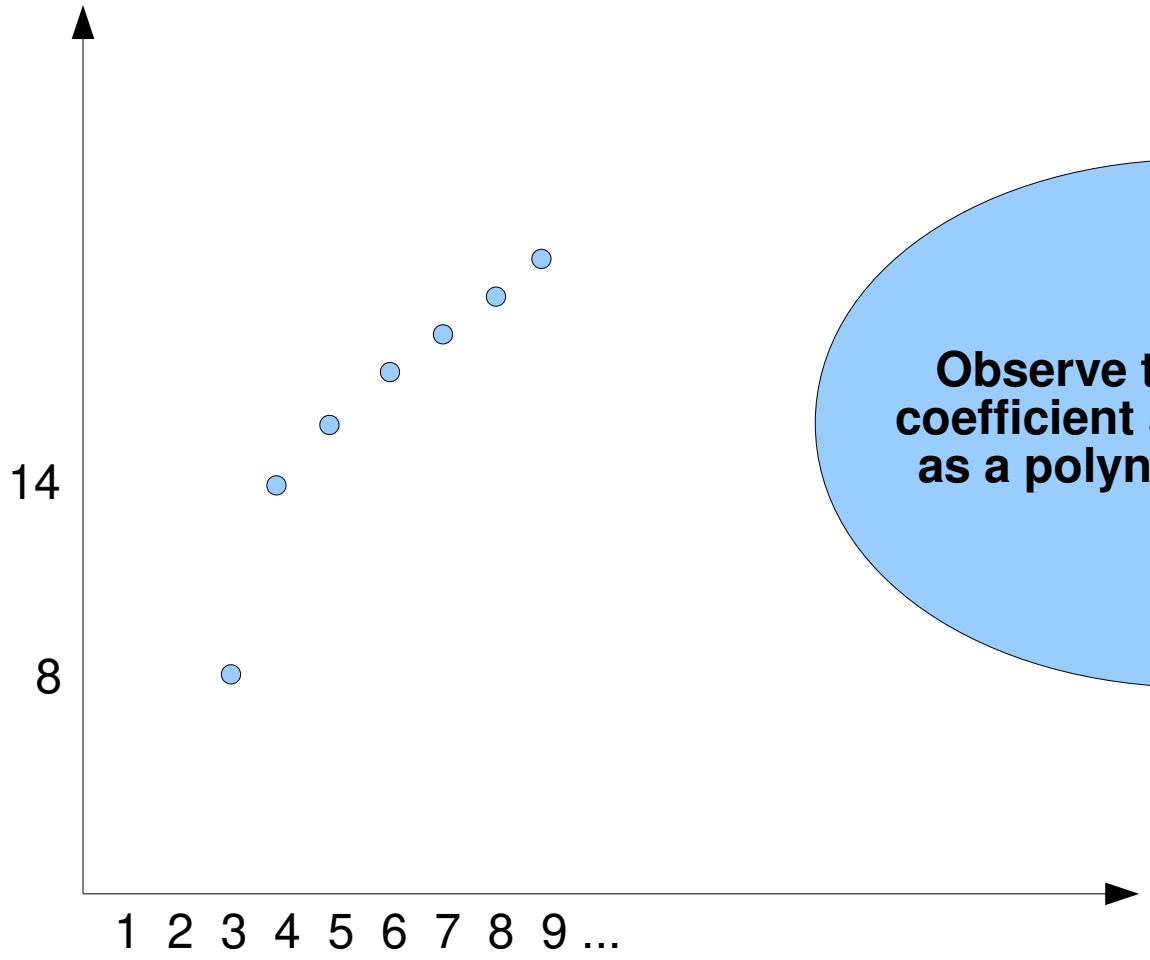
Computing the Shattering Coefficient

- In that sense, we conclude that for R^2:



Computing the Shattering Coefficient

- In that sense, we conclude that for R^2:



Observe the Shattering coefficient starts behaving as a polynomial function

Computing the Shattering Coefficient

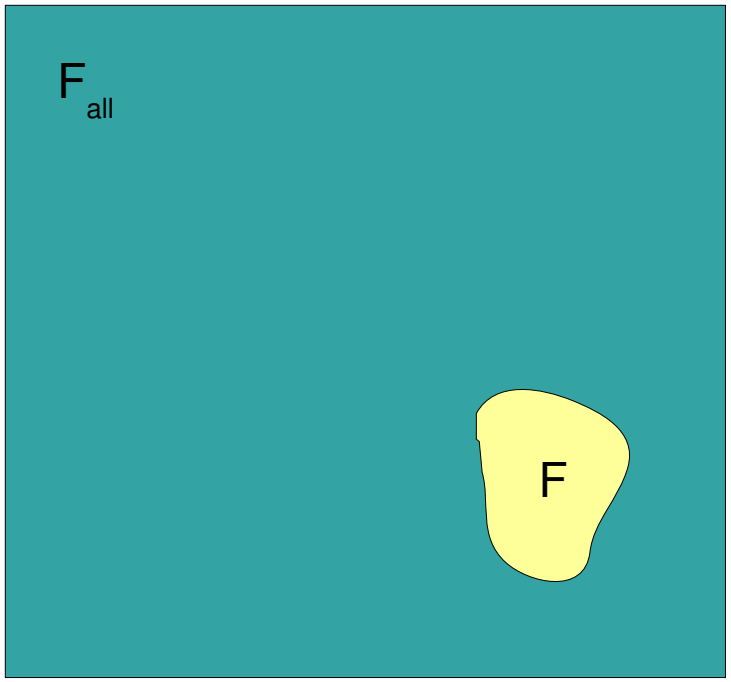
- In fact, **Learning** is only ensured if m(n) grows polynomially:

$$\sum_{i=1}^{m} P(|R(f_i) - R_{\text{emp}}(f_i)| > \varepsilon) \leq 2m \exp(-2n\varepsilon^2)$$

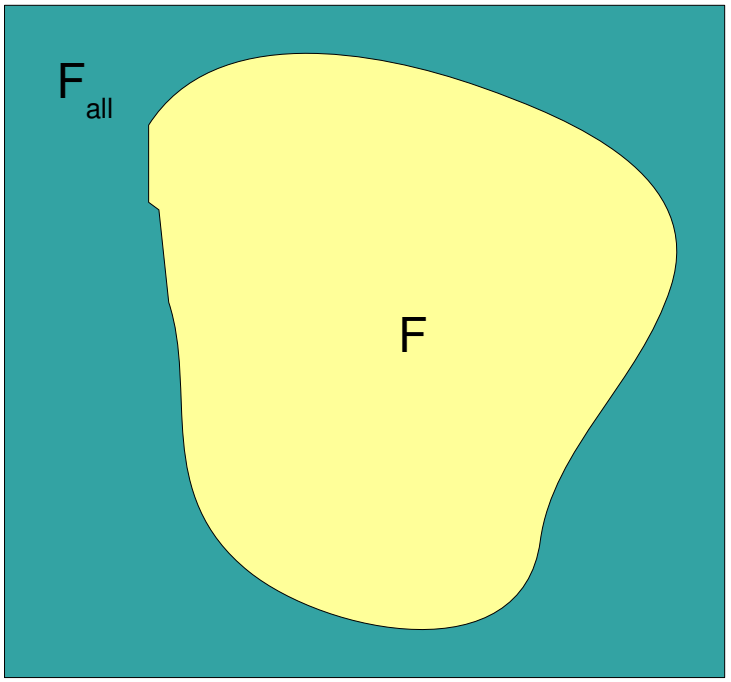
- Let us open the formulation and see what happens:
  - If it is polynomial
  - If it is exponential

Computing the Shattering Coefficient

- In this sense:



Polynomial Shattering coefficient



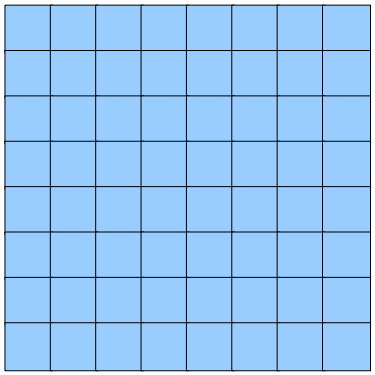
Exponential Shattering coefficient

What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm

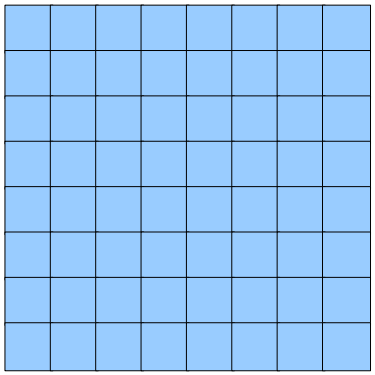
What about DL?

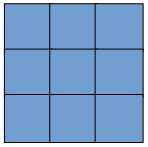
- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



What about DL?

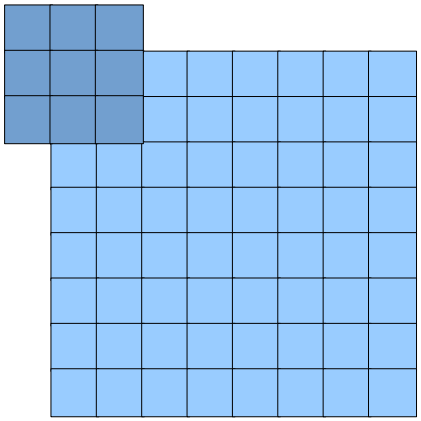
- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm





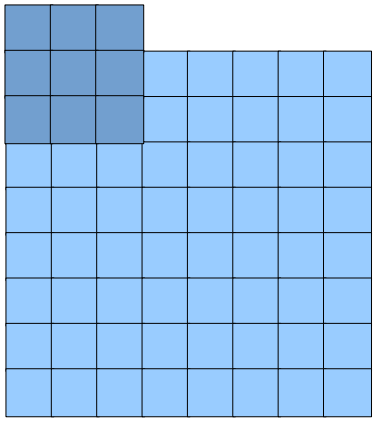
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



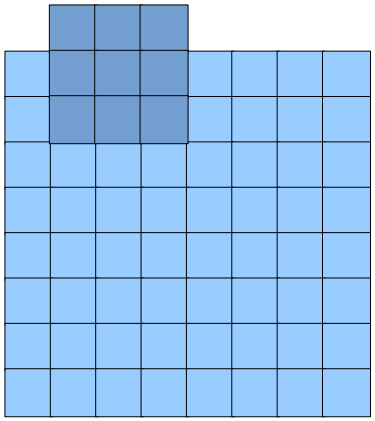
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



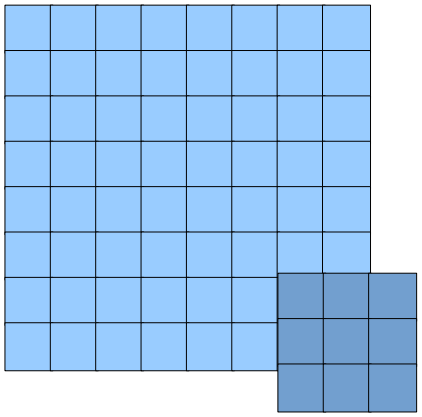
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



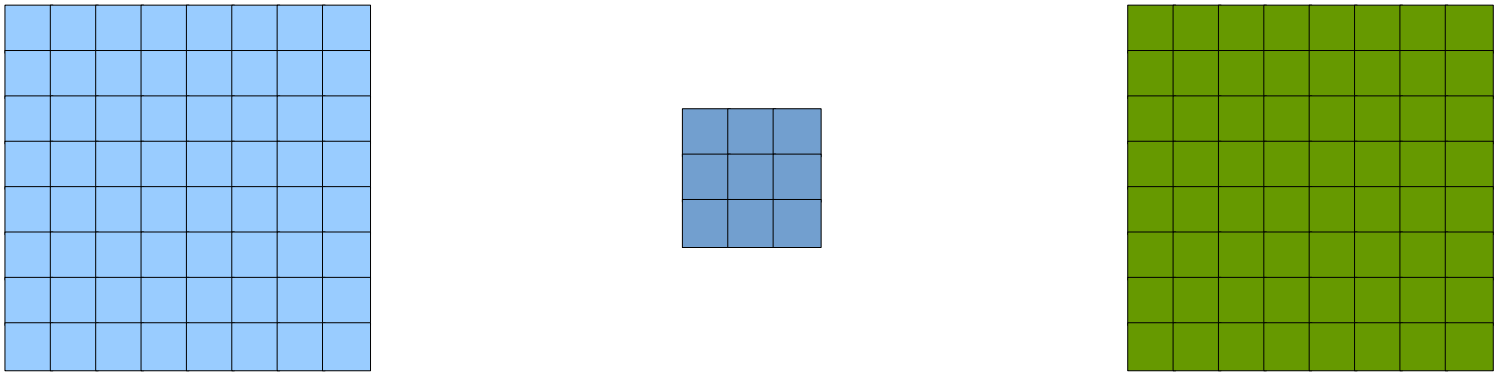
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



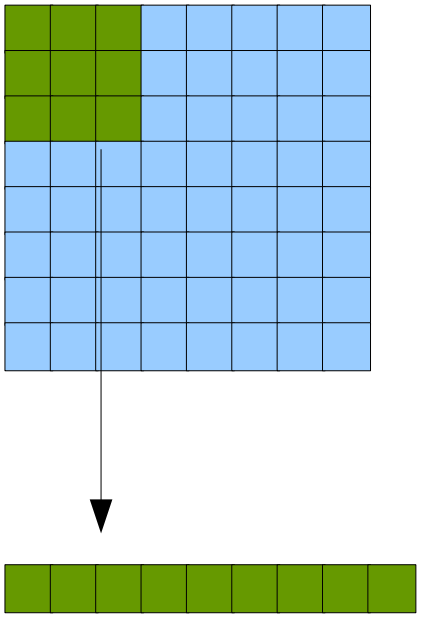
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



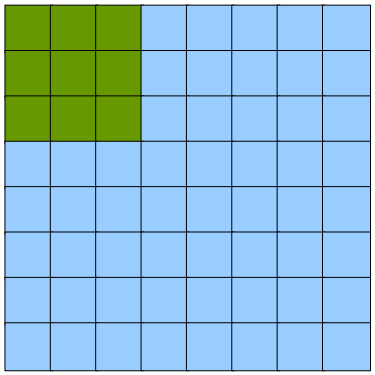
What about DL?

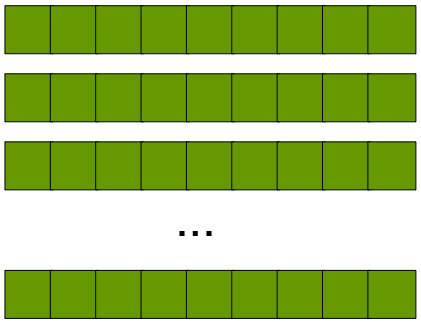
- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



What about DL?

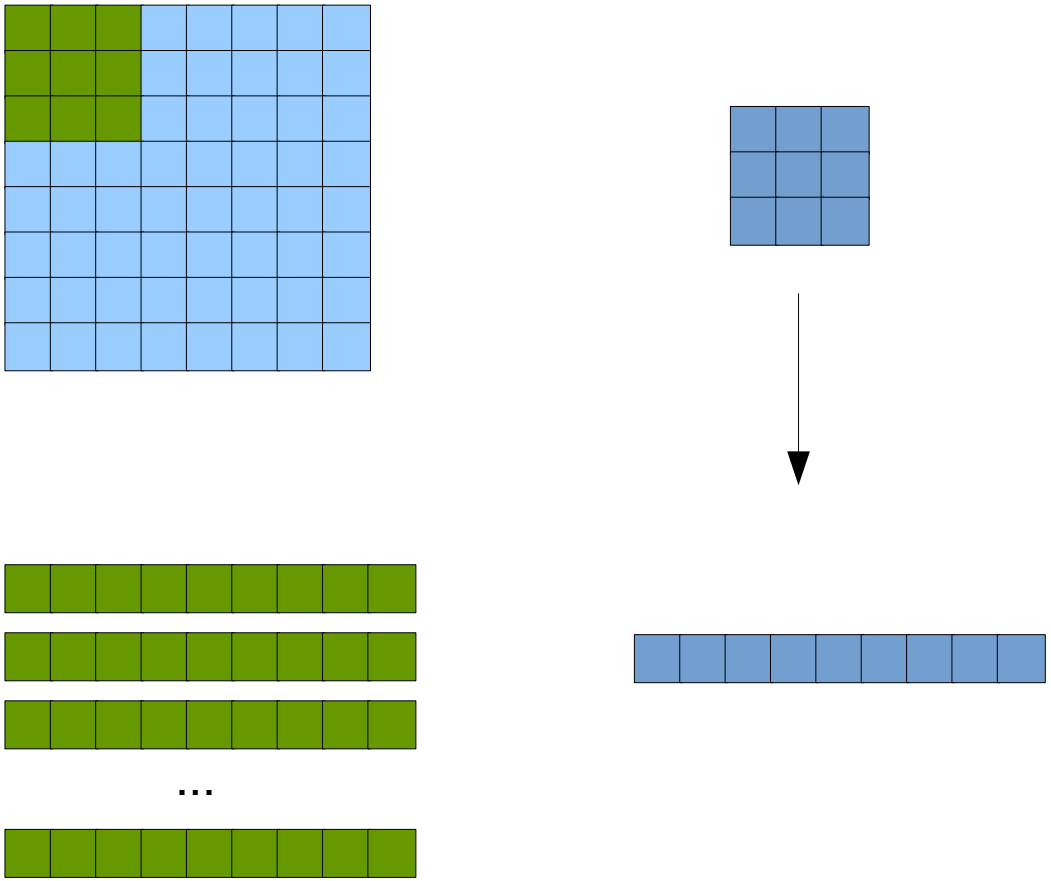
- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm





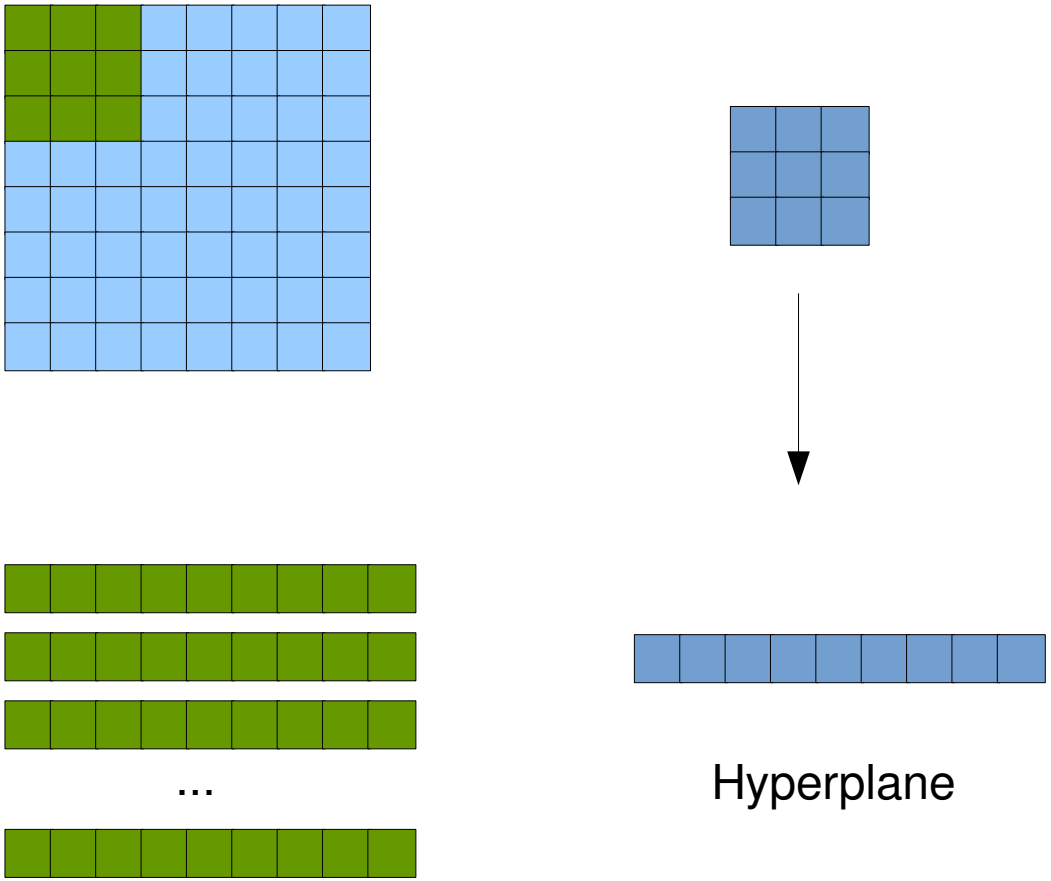
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



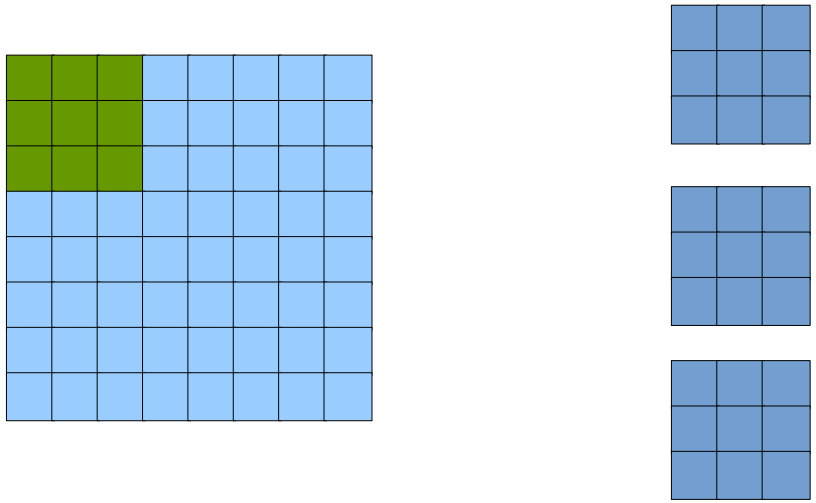
What about DL?

- We will discuss about Convolutional Neural Networks (CNNs) as a relevant example of DL algorithm



What about DL?

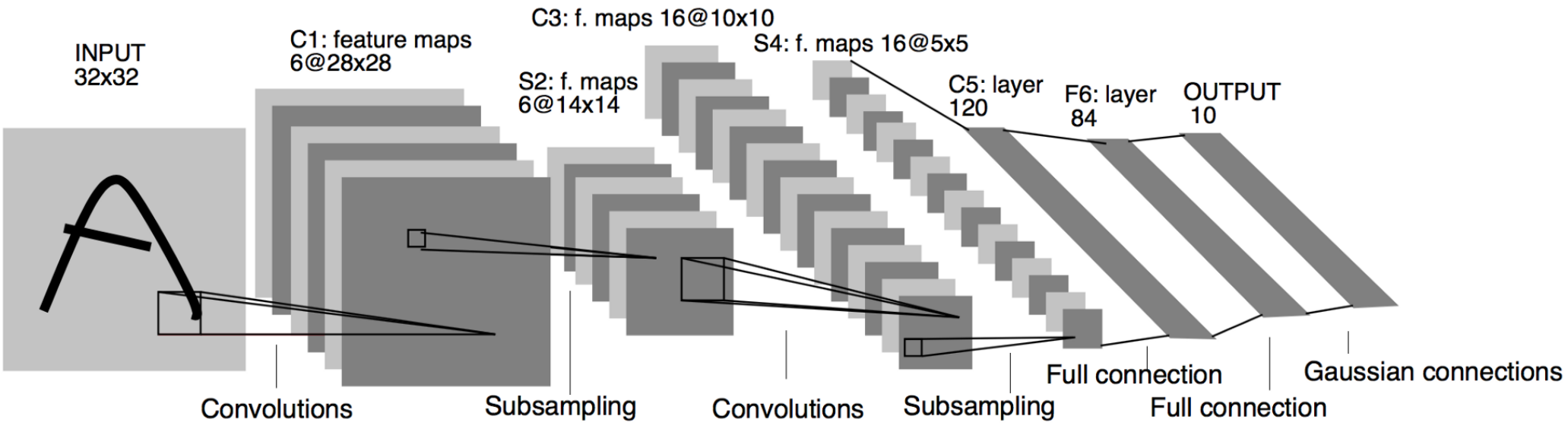
- Now we see a convolutional mask defines the **input space** on which hyperplanes are built up
  - So the Shattering coefficient could be computed on top of that scenario



3 hyperplanes in R^9

What about DL?

- What happens with an architecture such as LeNet-5?



Some Conclusions

- Is it fair to compute the Shattering Coefficient like that for complex models?

Some Conclusions

- Is it fair to compute the Shattering Coefficient like that for complex models?

- Mello, R. F., Ponti, M. A., Ferreira, C.H.G.. Computing the Shattering Coefficient of Supervised Learning Algorithms, https://arxiv.org/abs/1805.02627
- Mello, R. F., Ferreira, M. D., Ponti, M. A.. Providing theoretical learning guarantees to Deep Learning Networks, https://arxiv.org/abs/1711.10292
- Ferreira, M.D., Corrêa, D. C., Nonato, L. G., Mello, R. F.. Designing architectures of convolutional neural networks to solve practical problems. Expert Syst. Appl. 94: 205-217 (2018)

References

- Mello, R. F. and Ponti, M. A.. Machine Learning: A Practical Approach on the Statistical Learning Theory, Springer, 2018
- Vapnik, V., The Nature of Statistical Learning Theory, Springer, 2011
- Schölkopf, B., Smola, A. J., Learning With Kernels: Support Vector Machines, Regularization, Optimization, and Beyond, MIT, 2002