

# MACHINE LEARNING POUR LES ONDES GRAVITATIONNELLES

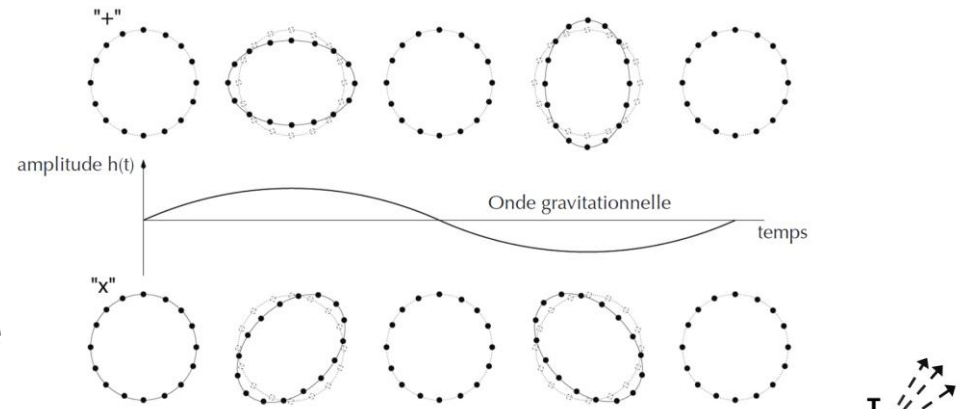
POUR LA DÉTECTION EN TEMPS RÉEL

# Ondes gravitationnelles

**Propagation** dans le vide: similaire aux ondes EM

$$\square \bar{h}_{\mu\nu} = 0 \iff (\nabla^2 - \frac{\partial^2}{\partial t^2})h_{\mu\nu} = 0$$

**Origine:** masses accélérées, moment quadripolaire

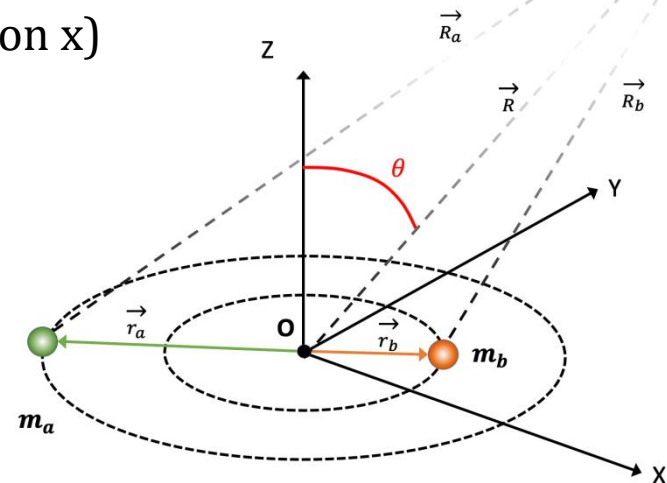


$$\bar{h}_{\mu\nu}(\mathbf{x}, t) = -\frac{4G}{c^4} \int_{source} \frac{T_{\mu\nu}(\mathbf{x}', t - \frac{|\mathbf{x}-\mathbf{x}'|}{c})}{|\mathbf{x}-\mathbf{x}'|} \quad \bar{h}_{ij}(t) = \frac{2G}{rc^4} \frac{d^2 I_{ij}(t - R/c)}{dt^2} \quad I_{ij}(t) = \int_{source} dx x_i x_j T_{00}(t, \mathbf{x})$$

**Système binaire:** Calcul dans le cadre de l'EM (polarisation x)

$$h(t) = \frac{\eta(GM)^{5/3} \omega^{2/3}(t)}{4Rc^4} \cos 2\Phi(t)$$

$$\Phi(t) = \int_{t_0}^t \omega(u) du = -\left(\frac{2}{5}\right)^{5/8} \left(\frac{t_c - t}{t_{SC}}\right)^{5/8} + \Phi_c$$



# Matched filter

**Forme des données:**  $s(t) = h(t) + n(t)$

**PSD:**

$$S_n(f) = \lim_{T \rightarrow \infty} \frac{\mathbb{E}[|\tilde{n}_T(f)|^2]}{2T}$$

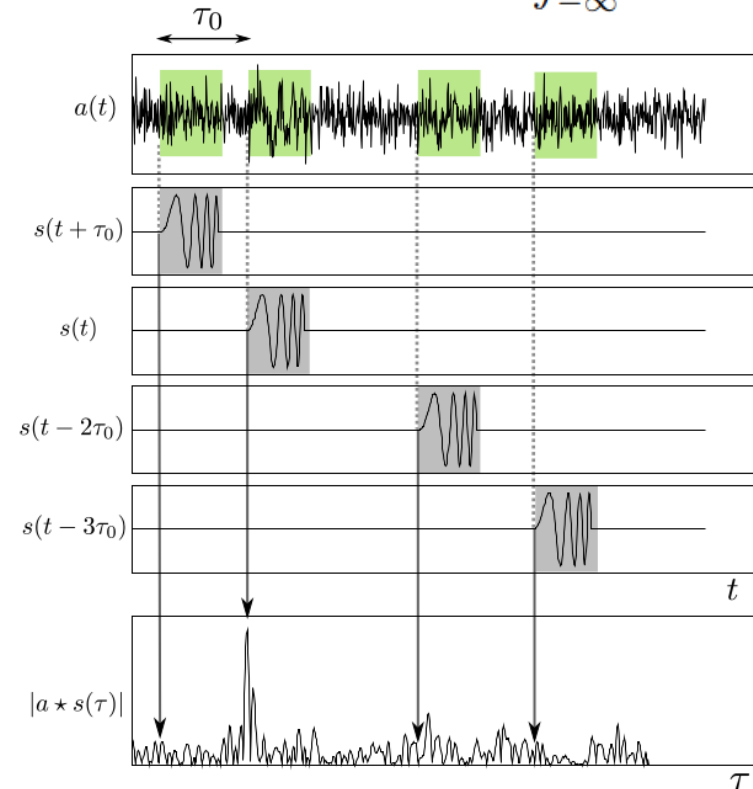
**Filtre adapté:**  $\tilde{g}(f) = 2 \frac{\tilde{h}^*(f)}{S_n(f)}$

**Sortie du filtre:**  $x(t) = \int_{-\infty}^{+\infty} \tilde{s}(f) \tilde{g}(f) e^{+2i\pi f t} df$

**Normalisation des templates:**

$$\rho_{opt}^2 = \langle x(t)x^*(t) \rangle = 2 \int_{-\infty}^{+\infty} \frac{|\tilde{h}(f)|^2}{S_n(f)} df$$

$$\rho(t) = \frac{|x(t)|}{\rho_{opt}}$$



# Machine learning: Regression

## Principe de l'entraînement supervisé

Equation régression linéaire:  $\hat{y} = \mathbf{X}\mathbf{w} + b$

Erreur quadratique moyenne à minimiser

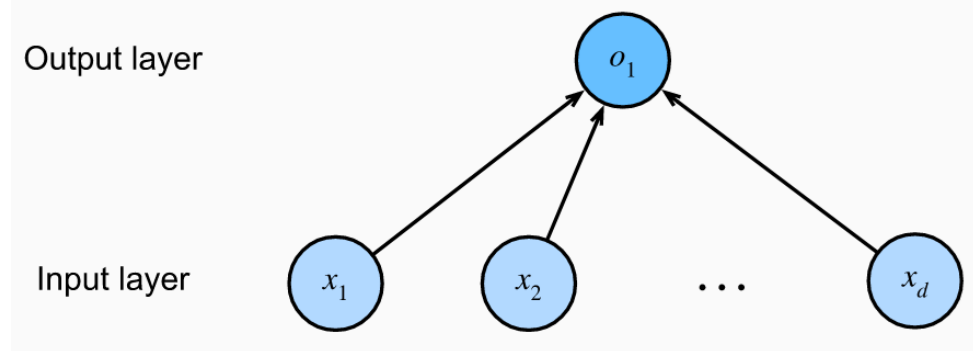
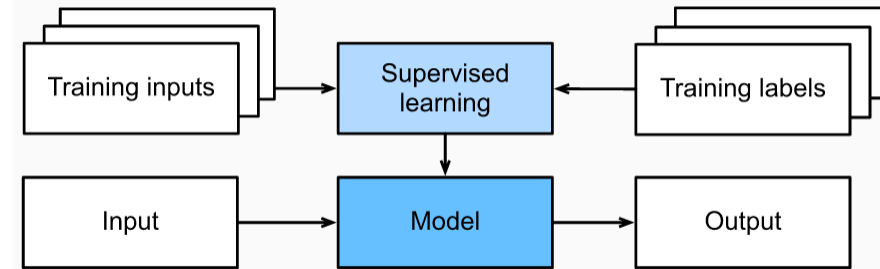
$$L(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n l^{(i)}(\mathbf{w}, b) = \frac{1}{n} \sum_{i=1}^n \frac{1}{2} (\hat{y}^{(i)} - y^{(i)})^2$$

Solution analytique: équations normales

$$\mathbf{w}^* = (\mathbf{X}^\top \mathbf{X})^{-1} \mathbf{X}^\top \mathbf{y}$$

Descente du gradient stochastique

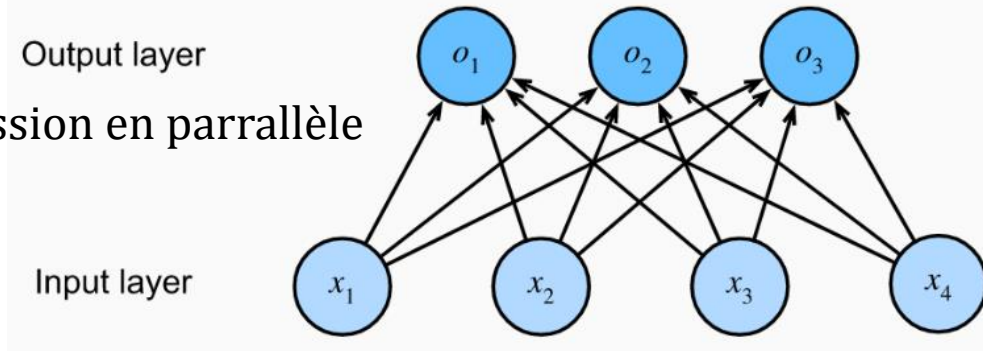
$$(\mathbf{w}, b) \leftarrow (\mathbf{w}, b) - \frac{\eta}{|\mathcal{B}_k|} \sum_{i \in \mathcal{B}_k} \partial_{(\mathbf{w}, b)} l^{(i)}(\mathbf{w}, b)$$



# Machine learning: Classification

**Modèle pour la classification:** plusieurs régression en parallèle

$$\hat{\mathbf{O}} = \mathbf{XW} + \mathbf{b}, \text{ et } \hat{\mathbf{Y}} = \text{softmax}(\hat{\mathbf{O}})$$



**Opération de softmax:** définition de la probabilité d'appartenance

$$\hat{y} = \text{softmax}(\hat{\mathbf{o}}) \text{ avec } \hat{y}_k = \frac{\exp(o_k)}{\sum_i \exp(o_i)}$$

**Maximisation de la vraisemblance**

$$P(\mathbf{Y}|\mathbf{X}) = \prod_{i=1}^n P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)})$$

En pratique minimisation du log négatif  $P(\mathbf{y}|\mathbf{x})$  définit comme **l'entropie croisée**

$$-\log P(\mathbf{Y}|\mathbf{X}) = -\sum_{i=1}^n \log P(\mathbf{y}^{(i)}|\mathbf{x}^{(i)}) = -\sum_{i=1}^n \sum_{j=1}^s y_j^{(i)} \log \hat{y}_j^{(i)}$$



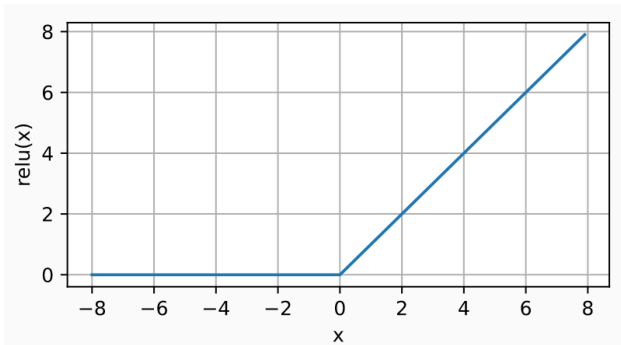
# Machine learning: Multicouches

**Modèle multicouches** ajout de couches intermédiaires:

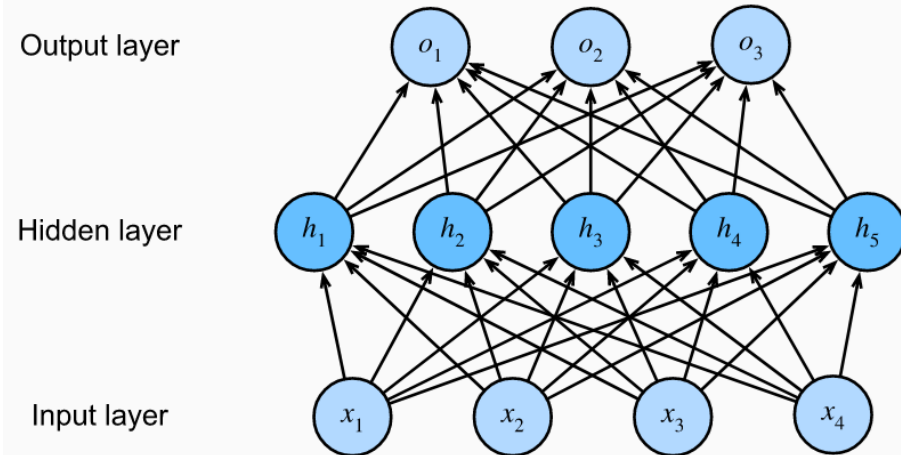
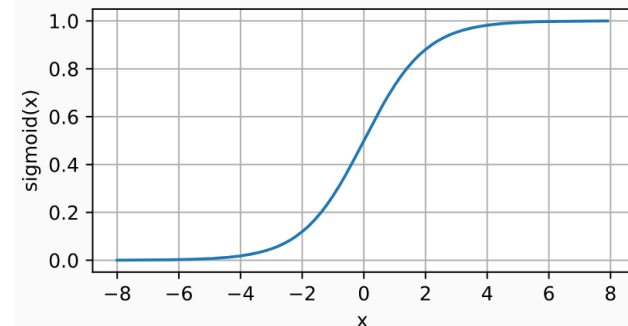
$$\hat{\mathbf{H}} = \sigma(\mathbf{XW}^{(1)} + \mathbf{b}^{(1)})$$
$$\hat{\mathbf{O}} = \hat{\mathbf{H}}\mathbf{W}^{(2)} + \mathbf{b}^{(2)}$$

Nécessité d'ajout **de fonctions d'activations non linéaires**:

$$\text{ReLU}(x) = \max(0, x)$$



$$\text{sigmoid}(x) = \frac{1}{1 + \exp(-x)}$$



# Machine learning: Convolution

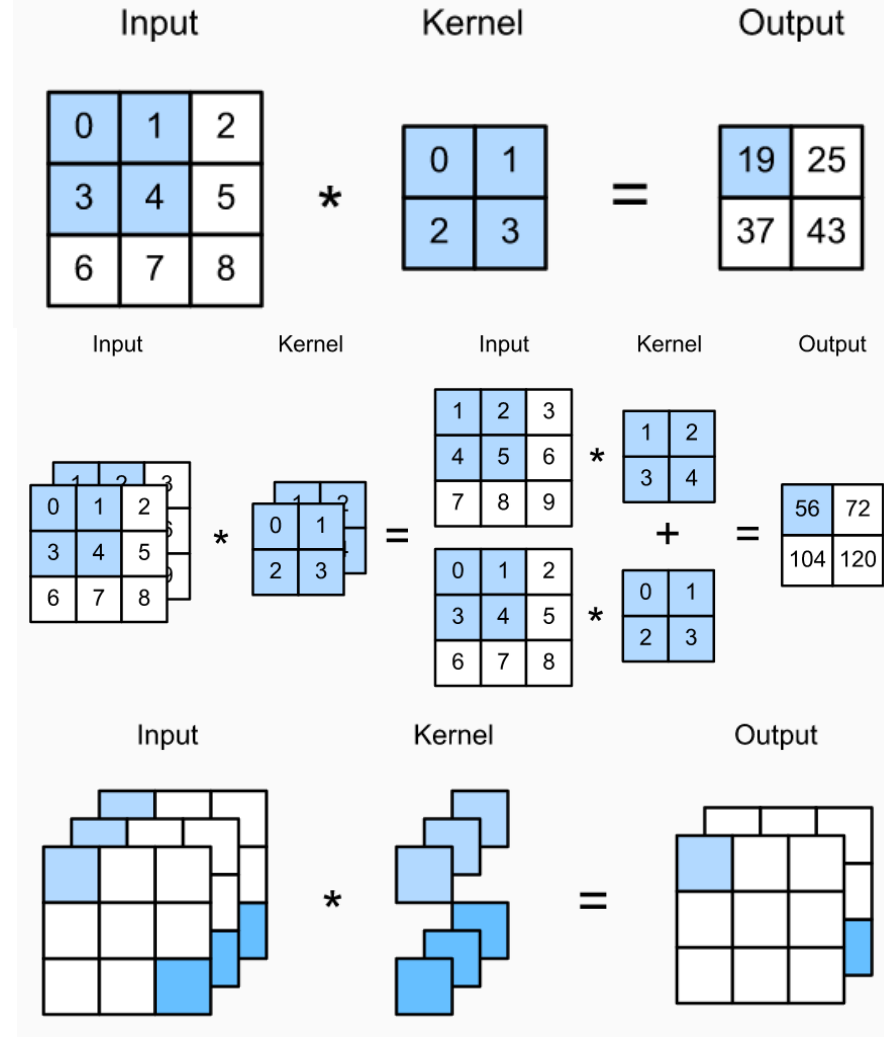
**Principe de la convolution:** Prise en compte de la corrélation entre les données (temporelle et/ou spatiale)

→ réduire le nombre de paramètres

$$[\mathbf{H}]_{k,l,s} = \sum_{a=-\Delta}^{\Delta} \sum_{b=-\Delta}^{\Delta} \sum_i [\mathbf{V}]_{a,b,i,s} [\mathbf{X}]_{k+a,l+b,i}$$

**Pixels à plusieurs entrées:** choix du nombre de kernel et de la taille

**Plusieurs sorties:** Comme dans les réseaux multicouches mais en dimension supérieure



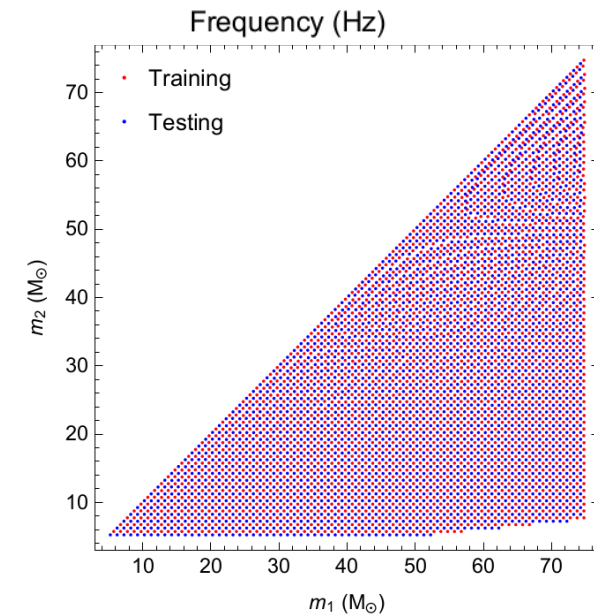
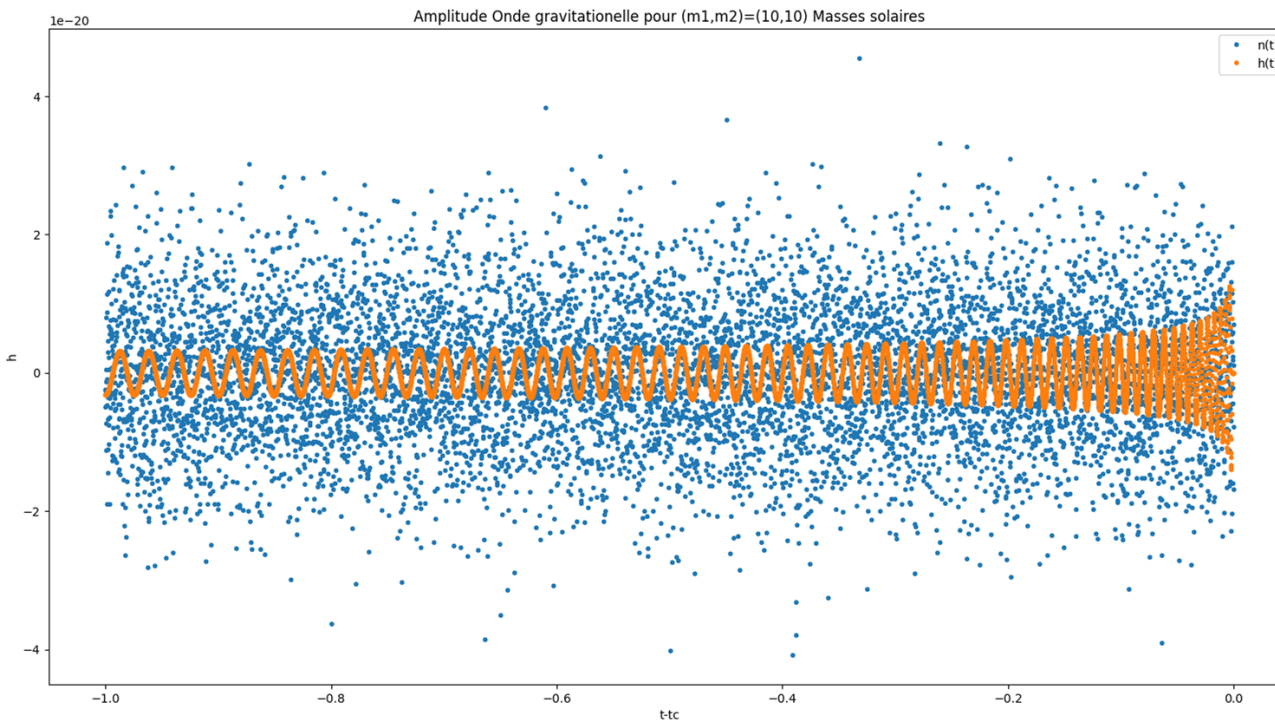
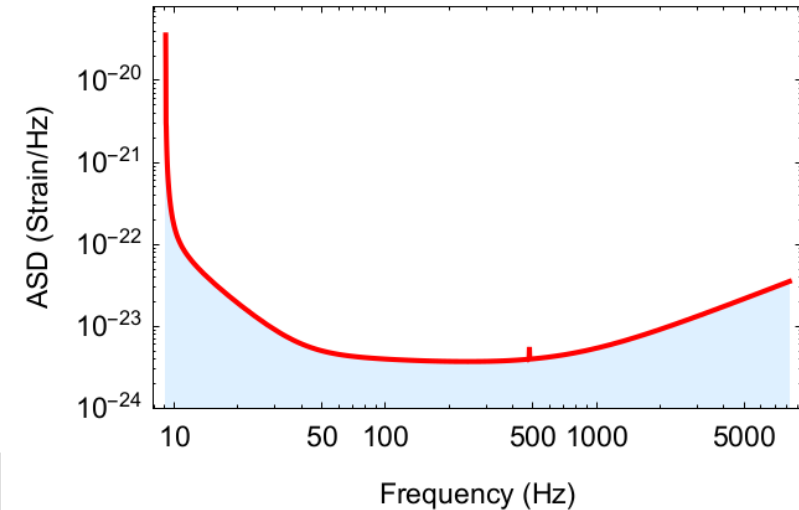
# Génération des données

**Génération des templates:** 10 à 50 Msolaire :

1. Formalisme Newtonien
2. EOB Numérical Relativity

**Génération du bruit:**

1. PSD plate (bruit blanc gaussien)
2. PSD Zero Detuned High Power sensitivity





# Réseau et entraînement

## Entrée:

Matrice où chaque ligne représente un échantillon

- Moitié de la matrice contenant du bruit seul
- Autre Moitié bruit + signaux (10 à 50 Msolaire)



## Réseau mis en place:

- 3 couches de convolution
- 1 couche linéaire dense (fonction d'activation relu)
- Une couche de sortie avec 2 neurones et un softmax



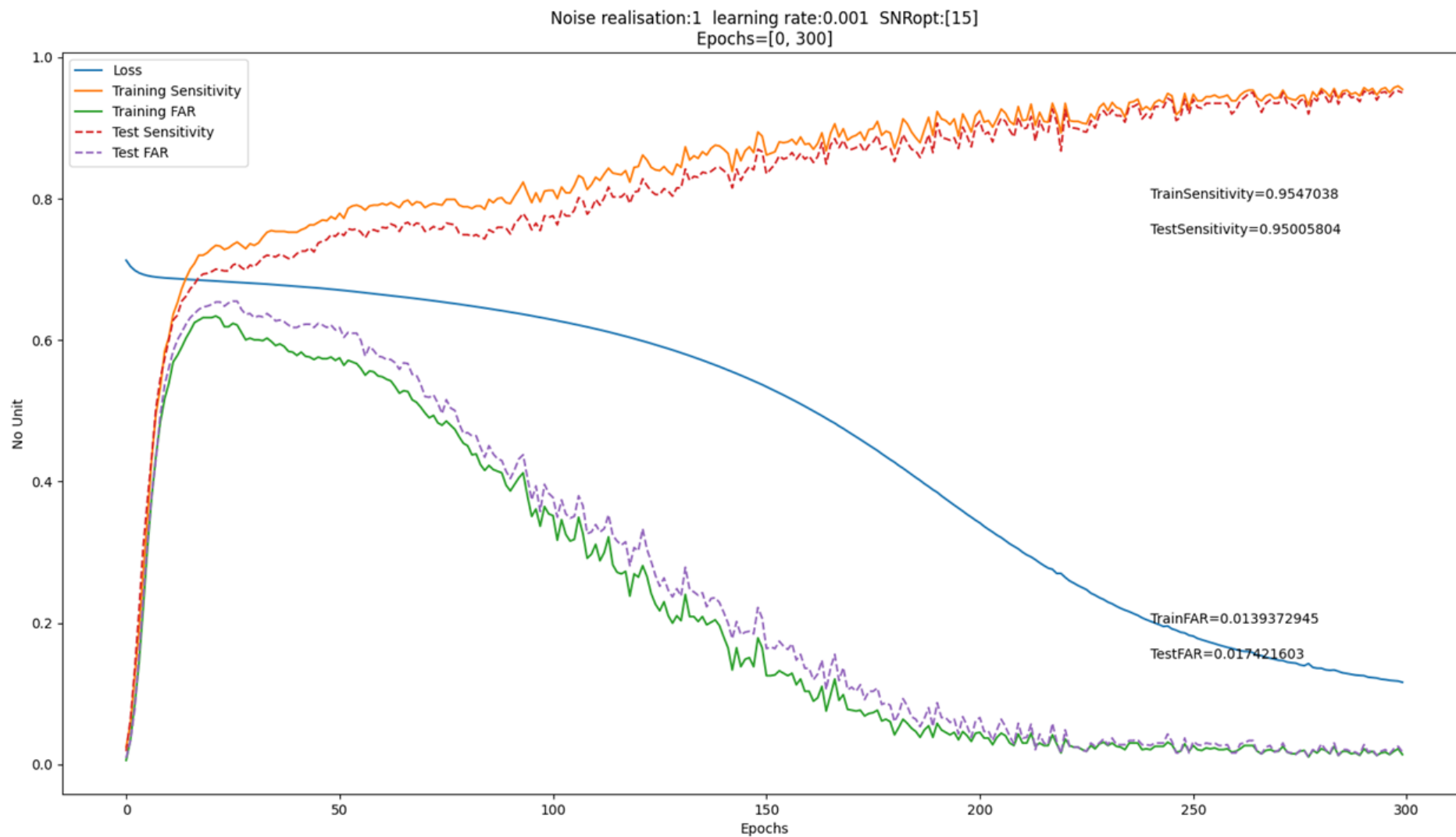
## Entraînement:

- Descente du gradient stochastique
- Entropie croisée
- Deux types d'entraînement à SNR optimal fixe ou en réduisant le SNR au fur et à mesure

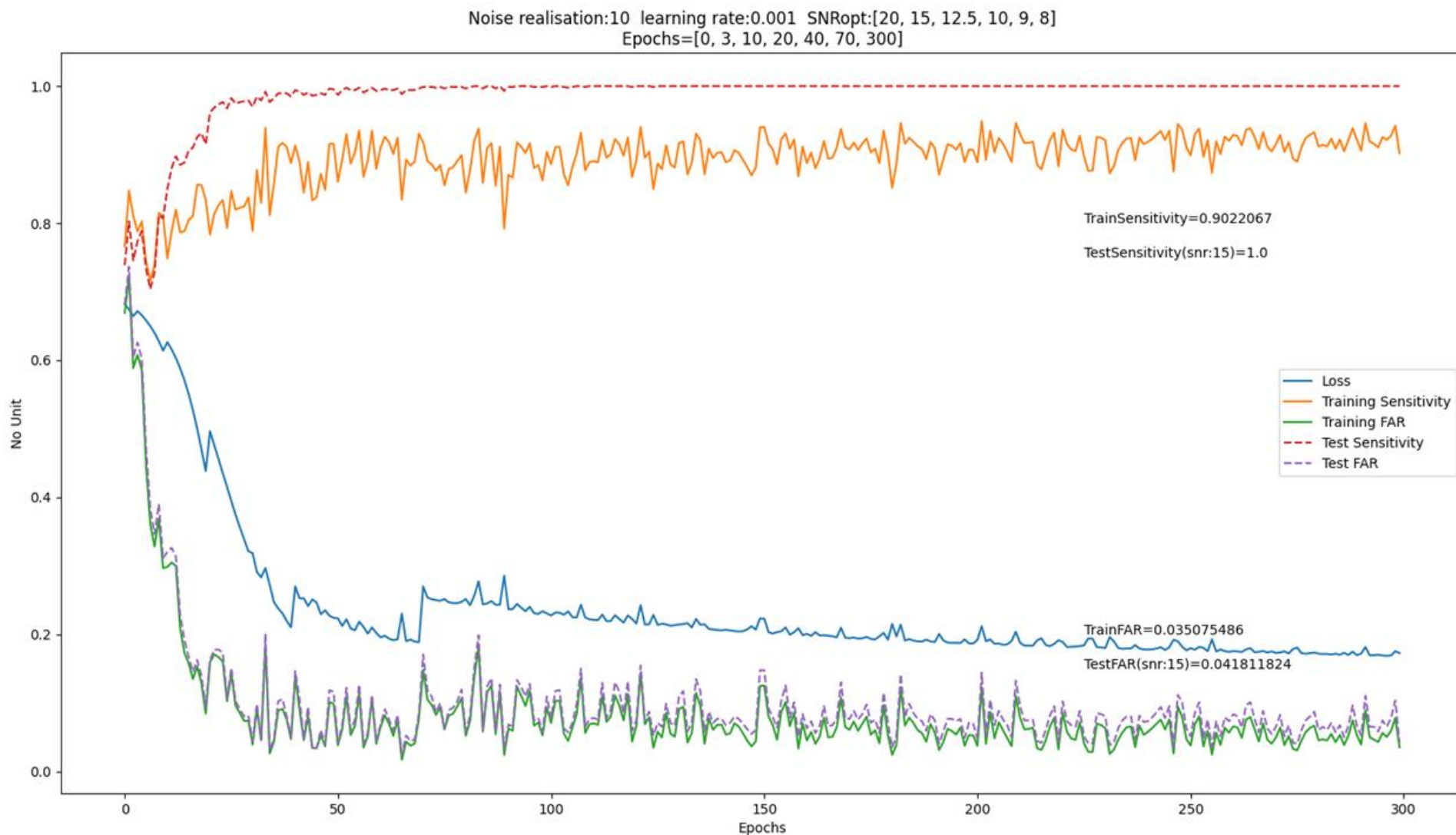
## Sortie:

Probabilité d'appartenance à la classe "bruit" et "bruit + signal"

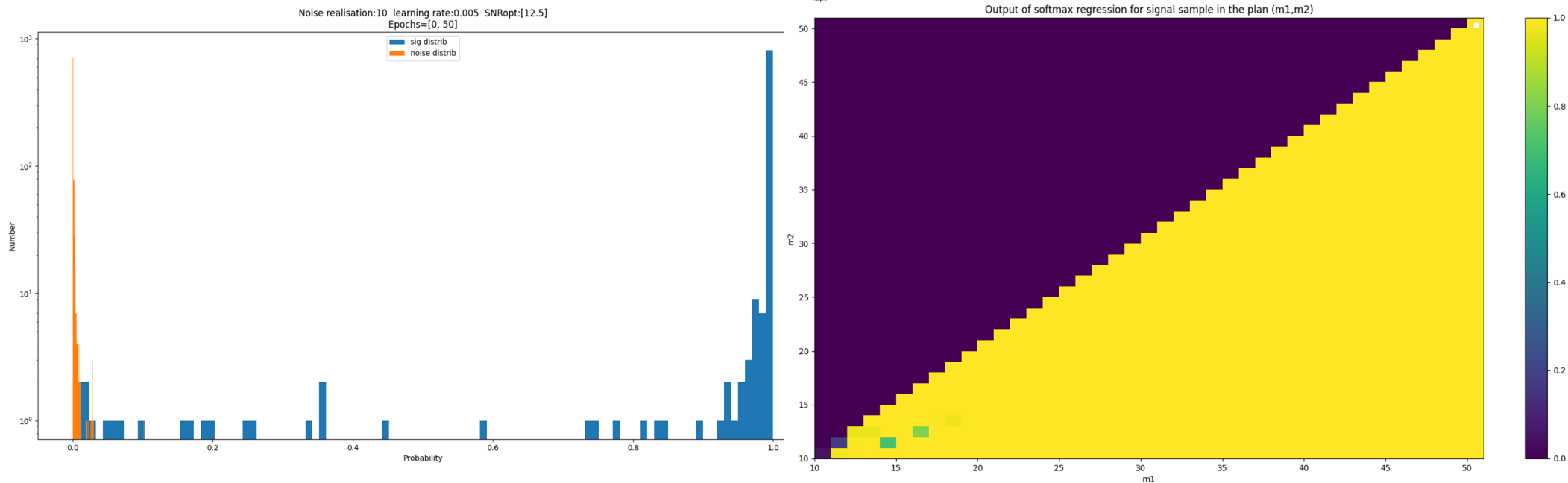
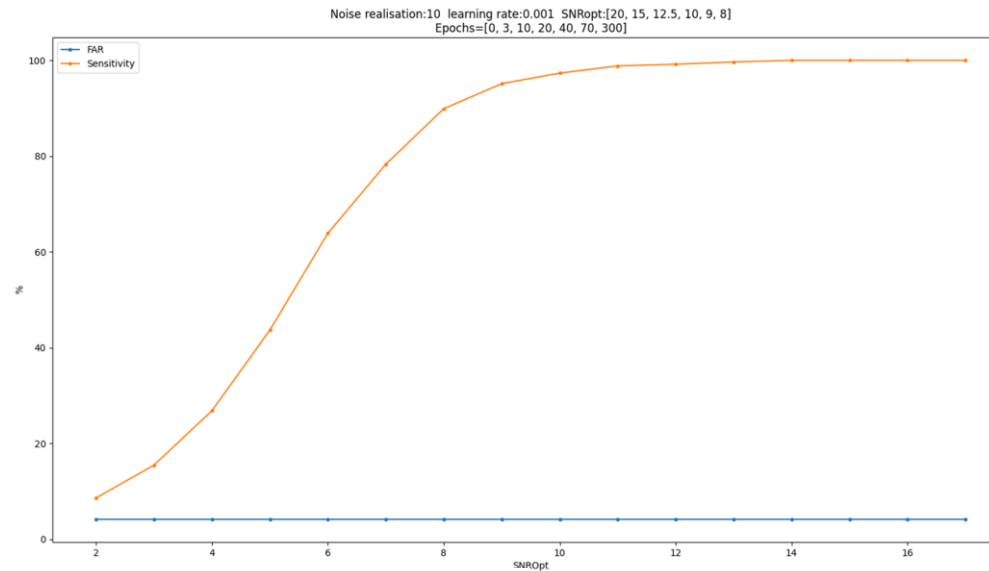
# Premiers résultats: SNR fixe



# Premiers résultats: SNR décroissant



# Premiers résultats: Efficacité



# Perspectives

- Prise en compte du **bruit coloré** de LIGO/Virgo à travers la PSD
- Amélioration des signaux avec une simulation **utilisant la NR**
- Extension sur des **signaux plus long**, de faibles masses



MERCI!