Prévision climatique à base de réseaux récurrents.

PROBLÈME

- Aujourd’hui la modélisation et la simulation de l’évolution des variables climatique est un enjeu important.
- C’est un des piliers de la prévision météo et l’étude de risques météorologiques.
- Cependant les méthodes de simulation courantes ont quelques défauts de précisions.
- Ainsi les prévisions météo et d’incident météorologique ne sont pas infaillibles et présentes souvent des erreurs.

OBJECTIF

- Notre objectif est de prédire l'évolution temporelle aussi précisément que possible de variables climatique.
- Comme étude de cas nous avons sélectionner la variable de pression moyenne au niveau de la mer.
- Nous avons aussi pris la variable de température à 2 mètres du sol.
- Nous avons exploré l'utilisation de machine learning pour prédire l'évolution temporelle de ces deux variables.

POURQUOI LA PRESSION?

- Les données de pression moyenne au niveau de la mer sont relativement fiables.
- L'évolution de cette variable est moins chaotique que pour d'autres variables climatiques comme la précipitation par exemple.
- C'est une variable importante dans la simulation du climat et est prise en compte pour l'étude d'incident météorologiques.
- Une amélioration de la précision des simulations actuelles est souhaitée.

POURQUOI LA TEMPÉRATURE?

- Dans les modèles climatiques, la température est fortement liée à la pression.
- Utiliser les données de température devrait contribuer à la prédiction de la pression et vice versa.
- C’est une variable avec un cycle journalier et permet de tester si on peut prédire ce genre de variable.

POURQUOI LE MACHINE LEARNING

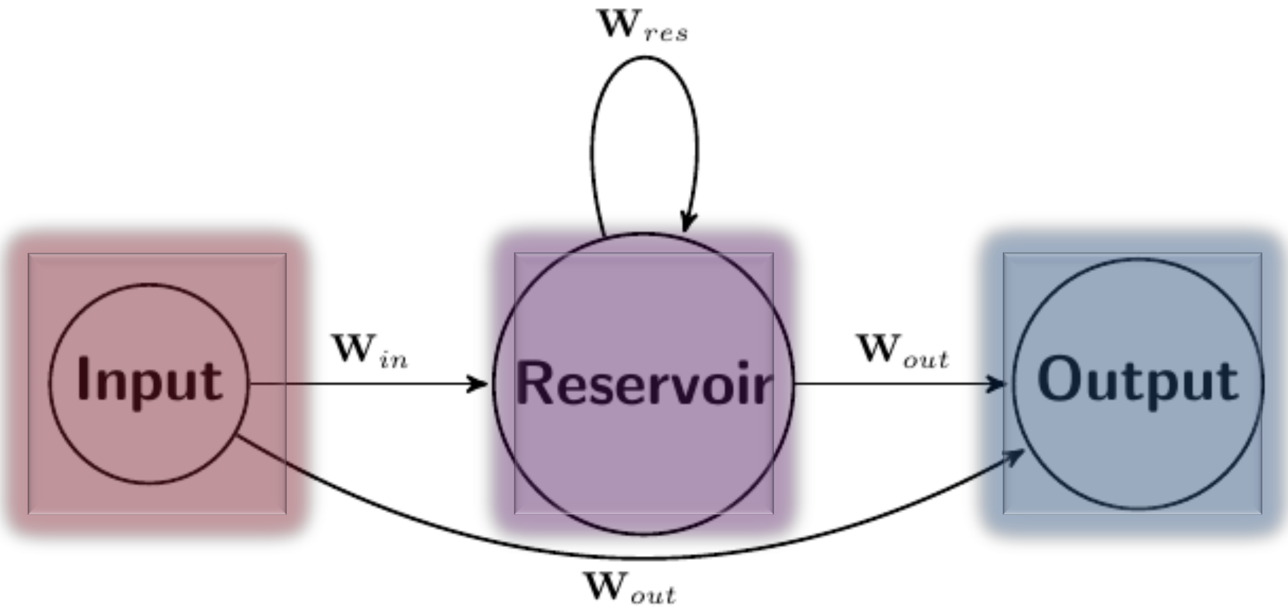
- Les processeurs sur le marché évoluent dans une optique d'optimisation des calculs fais dans des méthodes de machine learning.
- Le machine learning a montré des résultats intéressants dans l'apprentissage de l'évolution de variables temporelles chaotiques.
- On dispose de beaucoup de données sur le climat et on continue d'en récolter.
- Ainsi le machine learning pourrait être une nouvelle approche intéressante comme alternative aux simulateurs.
- Le machine learning pourrait aussi compléter et améliorer des méthodes existante.

QUELLE MÉTHODE?

- Des articles ont montré qu'une la méthode de machine learning appelée Echo state networks pouvait être fructueuse pour la prédiction de séries temporelles chaotiques.
- En 2004 un article de Jaegar montre des résultats pour la prédiction de séries temporelles chaotiques de petites dimensions.
- En 2017 un article de Pathak fait de même pour des séries de variables spatio-temporelles avec quelques centaines de dimensions.

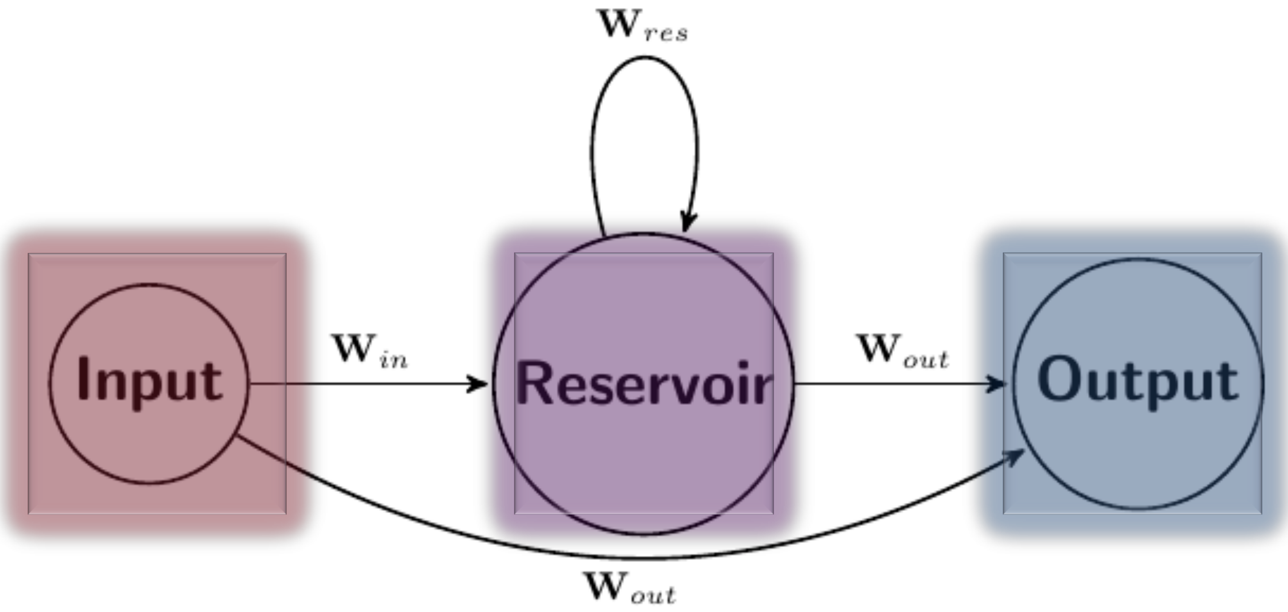
ENTRAINEMENT

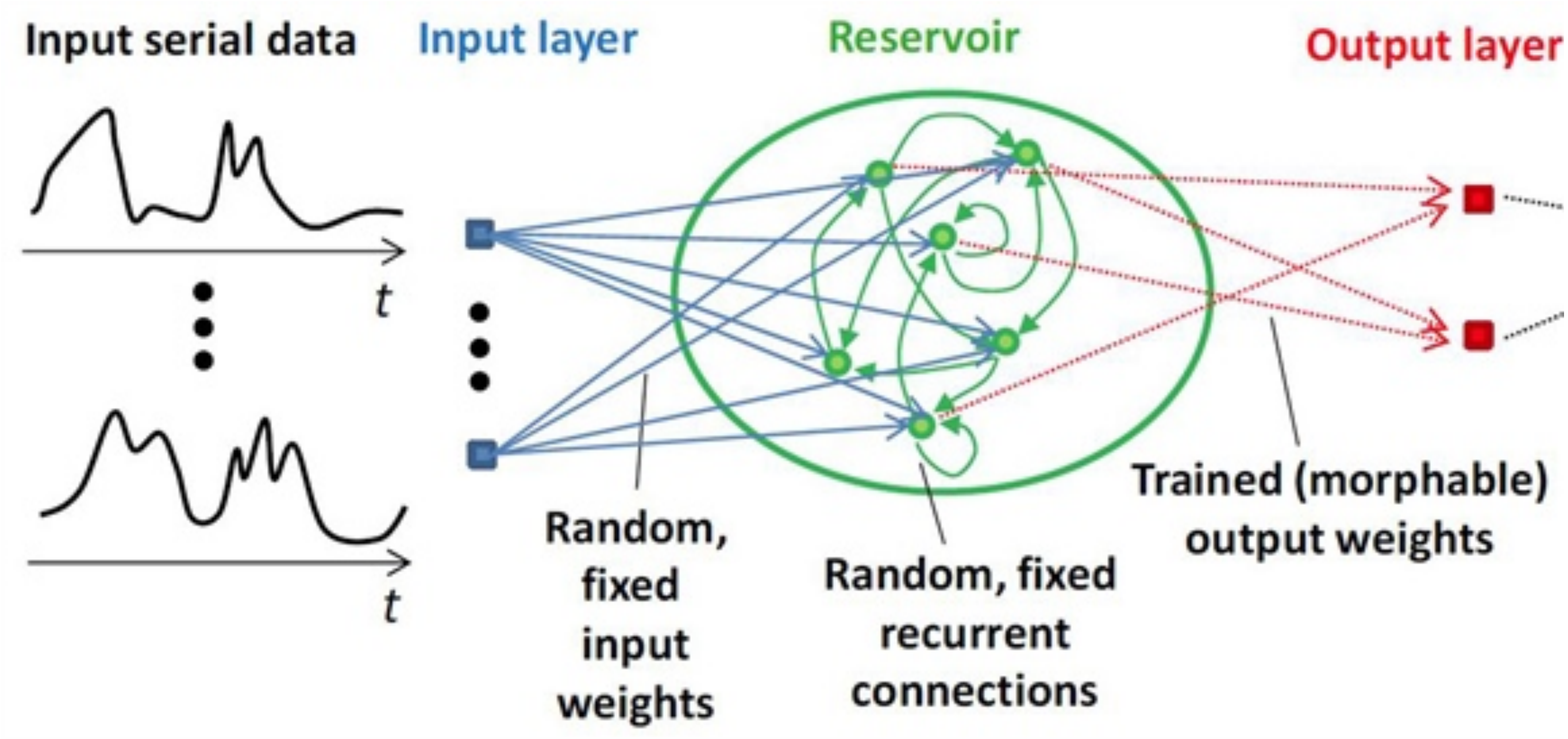
- 1: Le réseau parcours les données d'entraînement.
- 2: A chaque pas de temps le réseau met à jour son état en function de la donnée observée.
- 3: Il mémorise son état à ce temps ainsi que la valeur de la donnée au pas de temps suivant.
- 4: A la fin de l'entrainement le reseau estime une relation entre la donnée au temps t, son état au temps t et la donnée au temps t+1.



PREDICTION

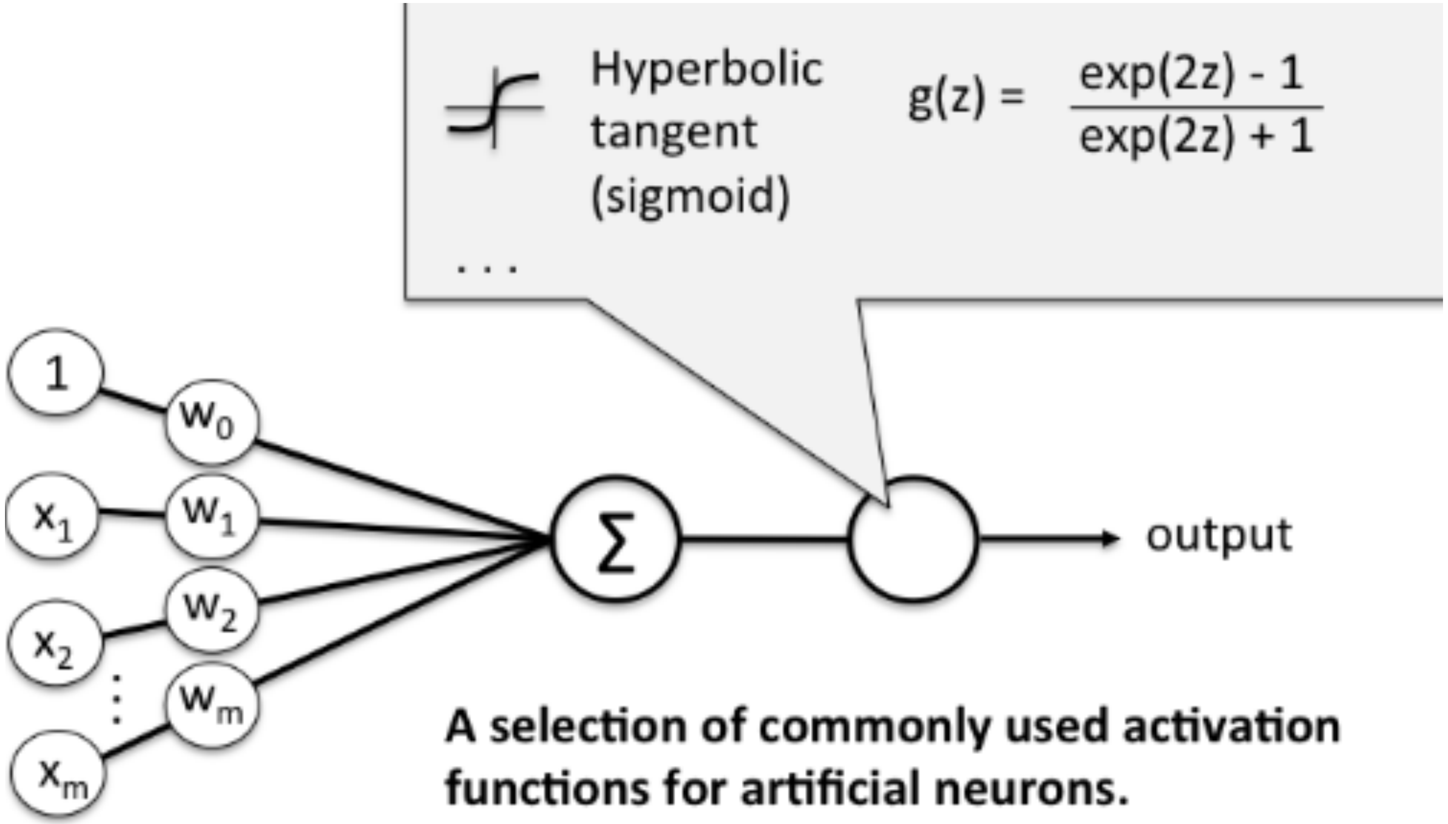
- 1: La donnée initiale est transmise au reservoir.
- 2: L’état des neurones du reservoir est mis à jour.
- 3: On decode l’état du reservoir pour obtenir la valeur de la donnée à l’instant suivant.
- 4: Cette valeur et ensuite mise en input et on reprend à l’étape 1.




Input serial data
Input layer
Reservoir
Output layer
t
t
Random, fixed input weights
Random, fixed recurrent connections
Trained (morphable) output weights

FONCTIONNEMENT D'UN NEURONE

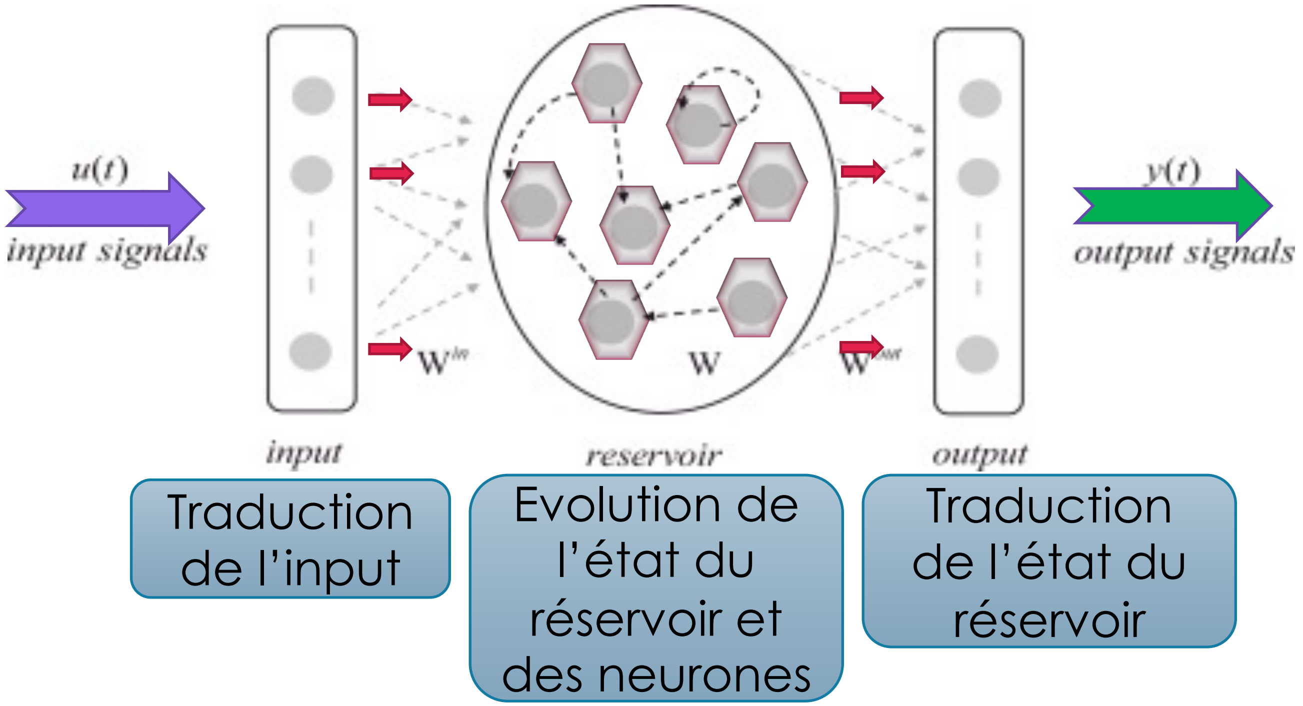
- Les x sont les variables en entrée du neurone.
- Les w sont les poids associés à chaque variable.
- Sigma est la somme pondérée des variables d'entrée.
- On applique la function tanh à cette somme pour obtenir le nouvel état du neurone.
- Les x sont les données d'entrées et les états des neurones connecté au neurone recepteur.



A selection of commonly used activation functions for artificial neurons.

RÉSEAU

- 1: La couche d’entrée transforme le signal d’entré en signal neuronal.
- 2: Le signal est transmis au reservoir et continue d’évoluer au sein du reservoir.
- 3: La couche de sortie decode l’état du reservoir et renvoie l’output.

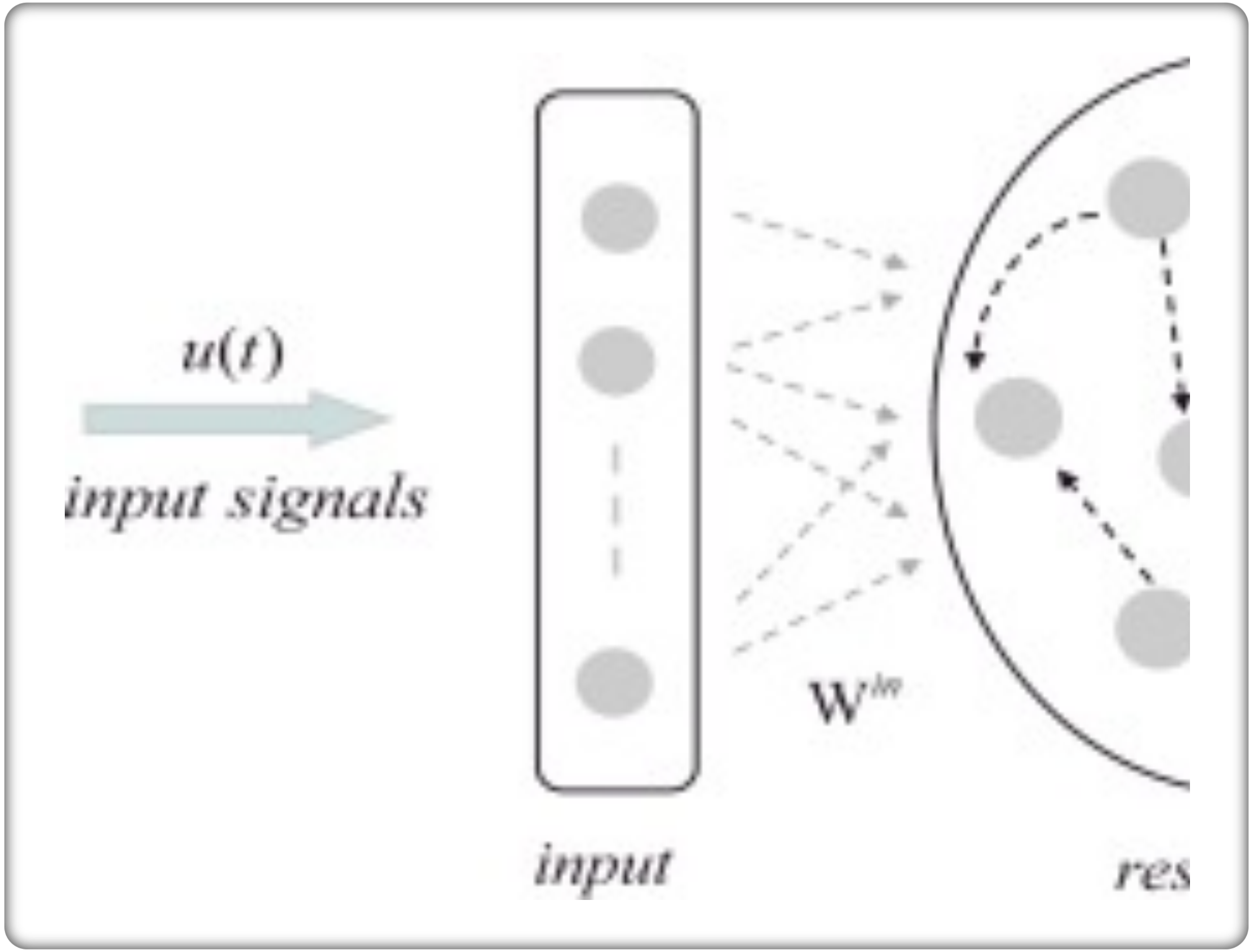


INPUT

- La couche d'entrée a pour but de traduire le signale d'entrée en signal neuronal.
- Les poids de la couche sont généralement fixés aléatoirement dans un intervalle [-**a**, **a**].
- Le choix du réel **a** est déterminant dans la précision du réseau.
- C'est une opération linéaire.
- Matrice des poids **W_in**:

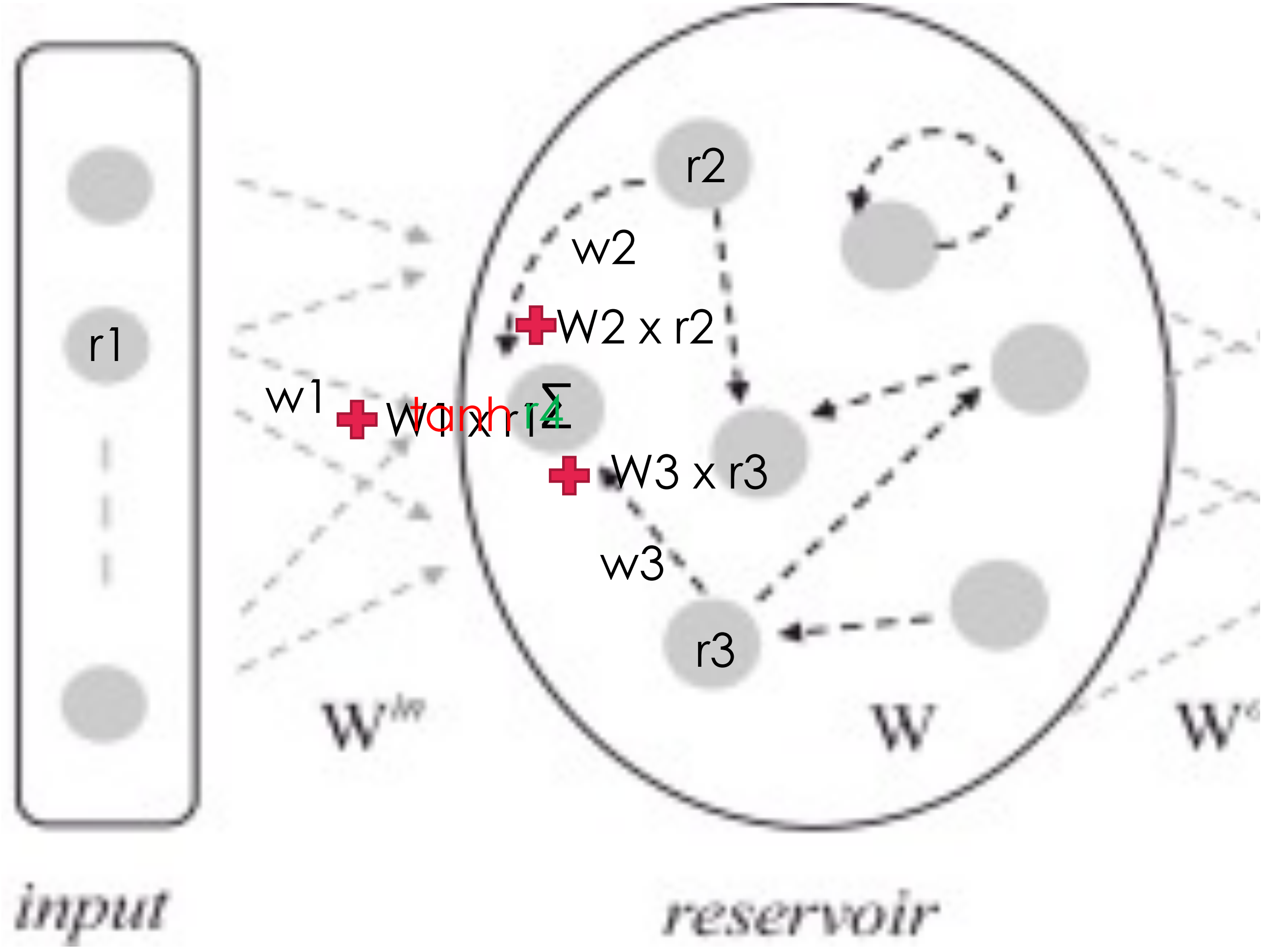
Dimension: [**n_reservoir** * **n_inputs**]

Valeurs: Aléatoirs entre [-**a**, **a**]



RESERVOIR

- Chaque neurone du réservoir possède un état.
- Chaque neurone est relié à un neurone de la couche de sortie.
- Chaque neurone possède aussi des connections entrantes avec des neurones du réservoir.
- L'état de chaque neurone évolue en fonction des états précédent et de l'input qu'il reçoit par la couche d'entrée.
- On applique une fonction d'activation, souvent tanh à l'état résultant.
- Le rôle de la fonction tanh est d'apporter des composantes non linéaire au calcul des états.



RESERVOIR (SUITE)

- Le réservoir peut être représenté par sa matrice d'adjacence notée **A**.
- Cette matrice représente les liaisons entre les neurones d'où le nom matrice d'adjacence.
- Matrice d'adjacence **A**:

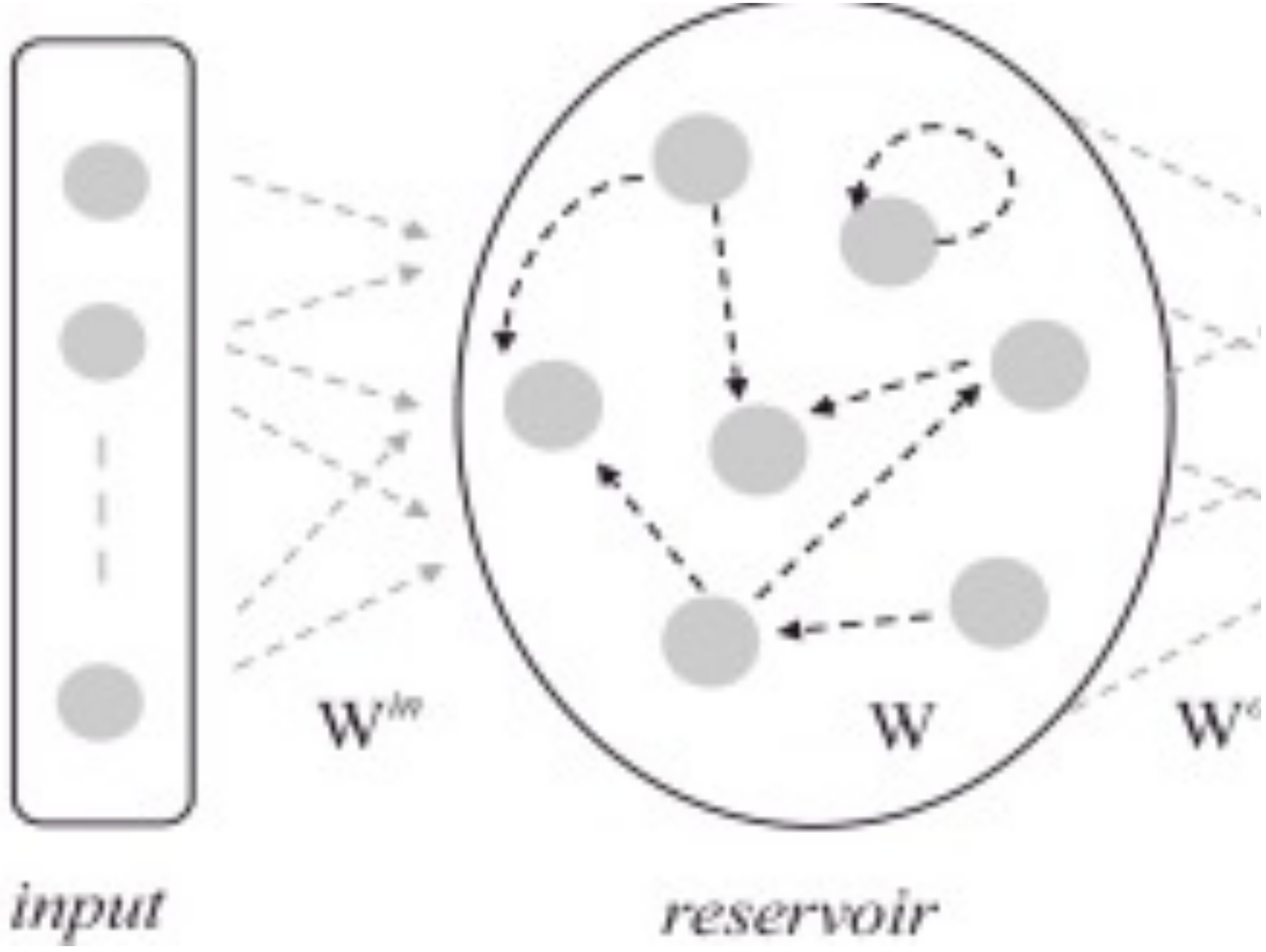
Dimensions: [n_reservoir * n_reservoir]

Le coefficient **aij** représente le poids de la liaison du neurone i vers le neurone j.

Si **aij** est nul c'est qu'il n'y a pas de connexion du neurone i vers le neurone j.

- Les coefficients de A sont fixés aléatoirement en fixant certains paramètres.

1. Le nombre moyen de liaison par neurone.
2. La plus grande valeur propre de la matrice.



FORMULATION DE L'ÉVOLUTION DE L'ÉTAT DU RÉSERVOIR

$$\mathbf{r}(t+\Delta t) = \tanh[\mathbf{A}\mathbf{r}(t) + \mathbf{W}_{in}\mathbf{u}(t)]$$

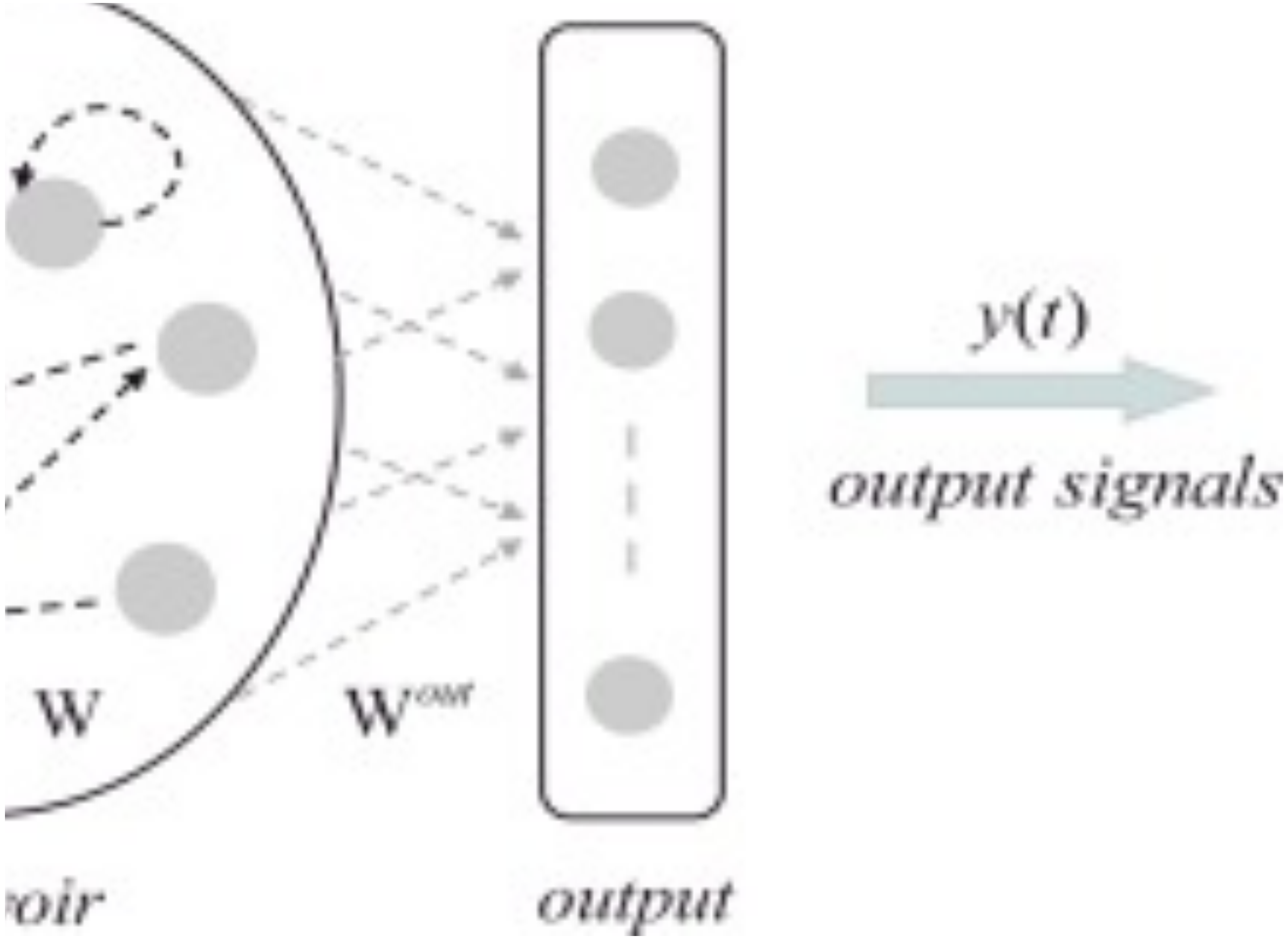
Arguments:

- A: Matrice d'adjacence générée aléatoirement. Détermine les poids du graphe des neurones du réservoir.
- r(t): Vecteur des états des neurones au temps t. Représente l'état de chaque neurone.
- W_{in}: Matrix weighting the effects of the inputs on the state.
- u(t): the input at time t.

OUTPUT

- Les valeurs prédites de pression en sortie est obtenue à partir des valeurs initiales de pression et de l'état du réservoir.
- On retrouve la valeur prédite par regression linéaire.
- Autrement dit on cherche une matrice W_out tell que
- **W_out * r(t)** = **Pression(t)**
- En posant **f(r)=W_out*r**
- Et **Yi=Pression(t+1)**, **Xi=Pression(t)**
- On trouve W_out par regression Ridge:

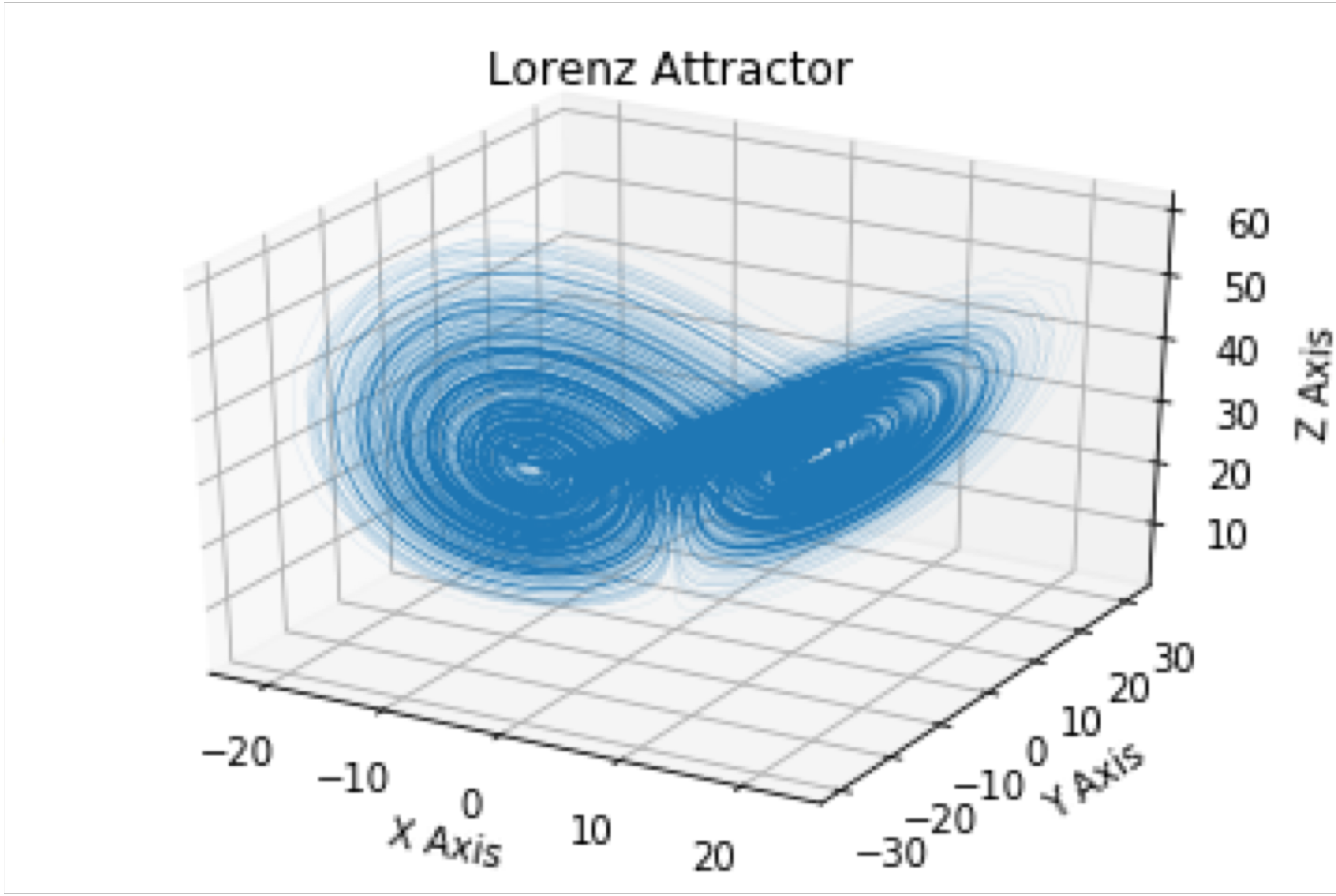
$$\min_{f \in H} \sum_{i=1}^{n} |Y_i - f(X_i)|^2 + \lambda \|f\|_H^2$$



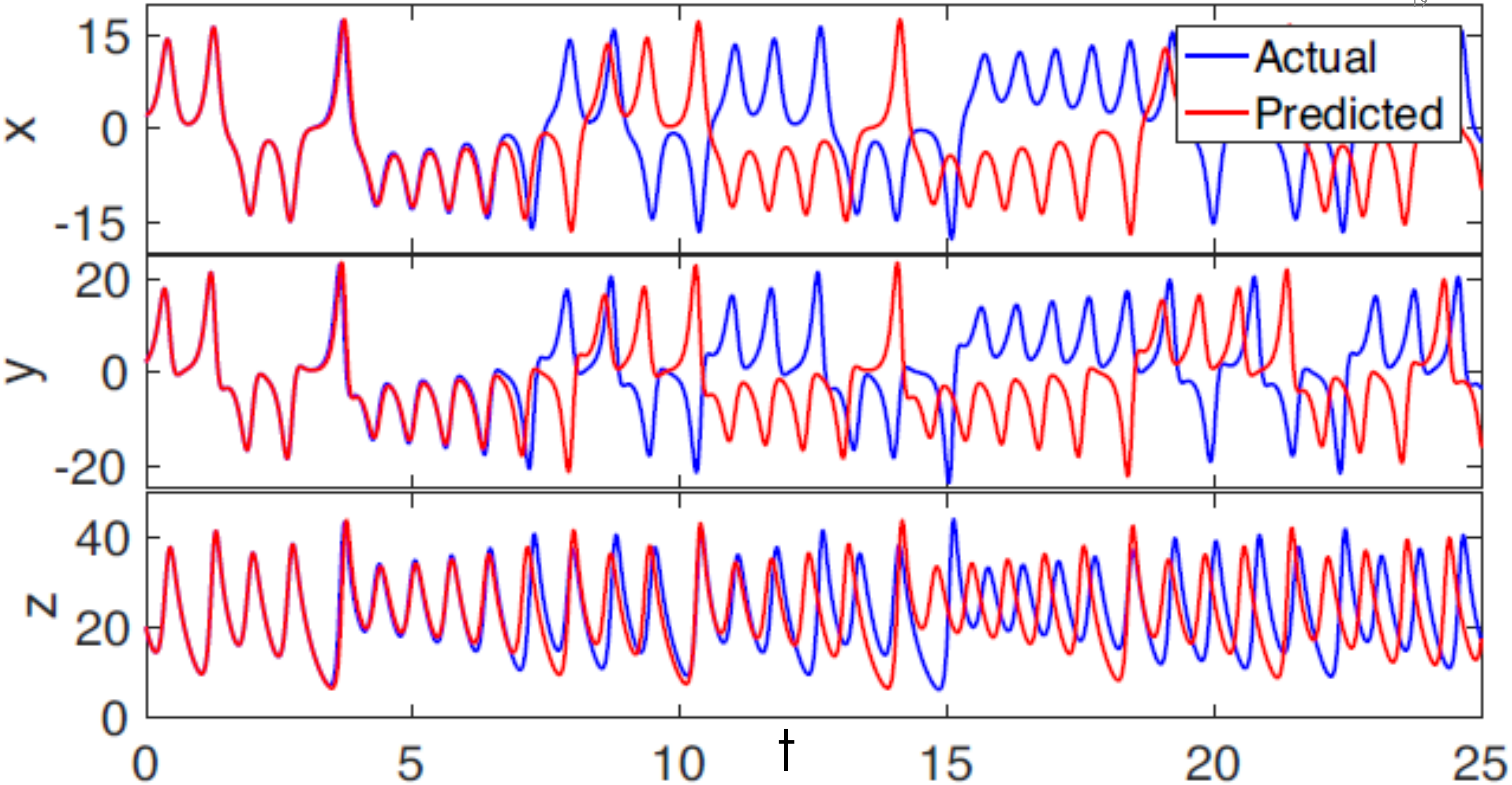
$$\min_{f} |Y_i - f(X_i)|^2$$

EXAMPLE: LORENTZ ATTRACTOR

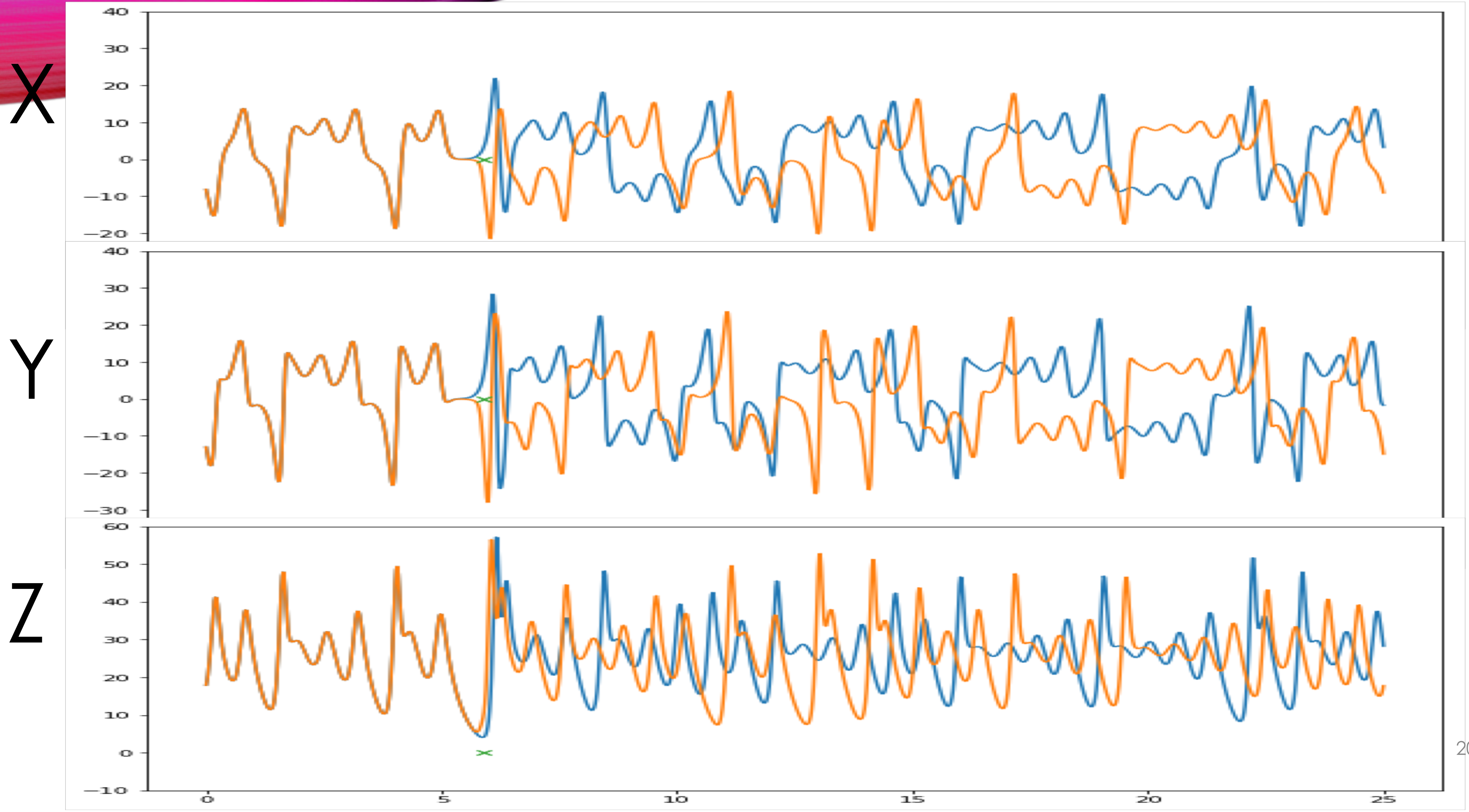
$$\dot{x} = 10(y - x),$$
$$\dot{y} = x(28 - z) - y$$
$$\dot{z} = xy - 8z/3.$$



RESULTS OBTAINED IN PATHAK ET AL 2017.

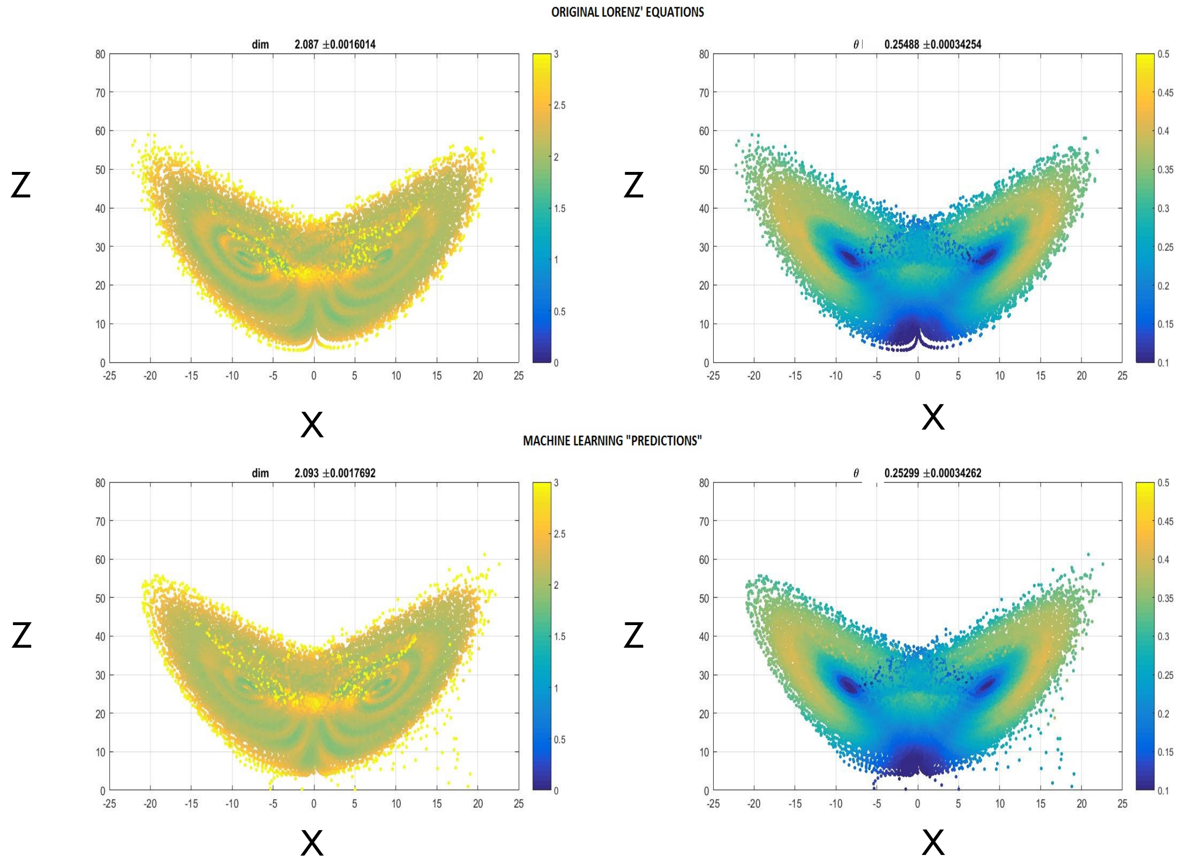


RÉSULTATS REPRODUITS



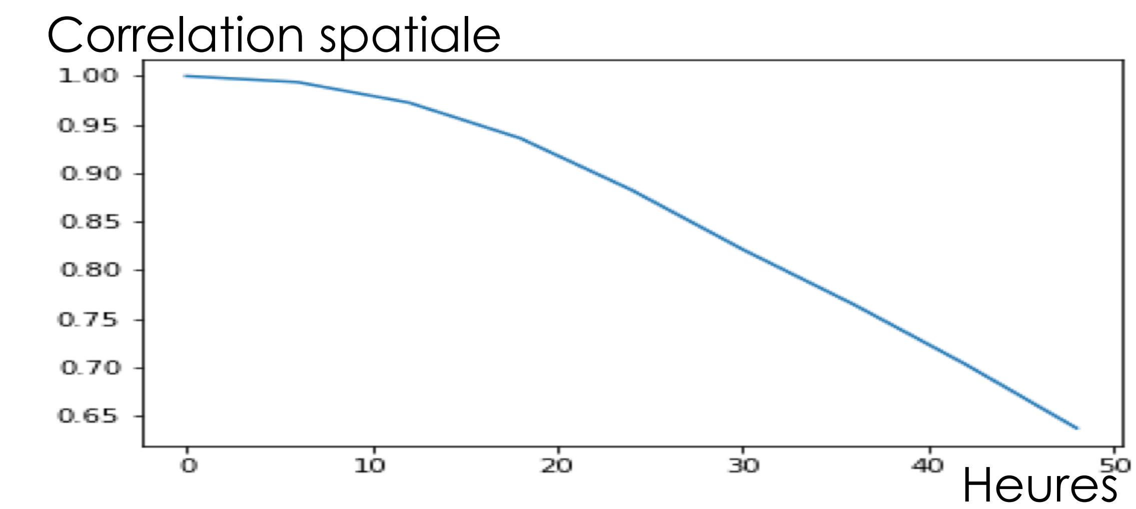
COMPORTEMENT À LONG TERME

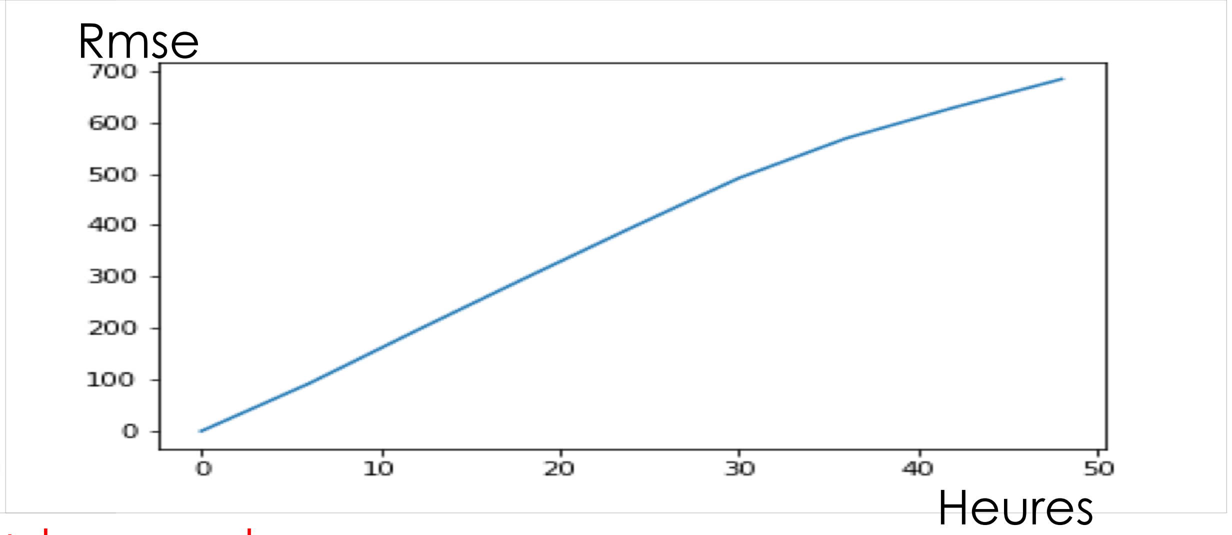
- On laisse boucler le reseau de neurone longtemps après qu'il ai quitté la trajectoire réelle.
- On voit que la trajectoire est cohérente avec celle observée pour les variables x et z.
- On peut se demander si on peut simuler l'evolution des propriétés climatiques à long terme grâce au machine learning.



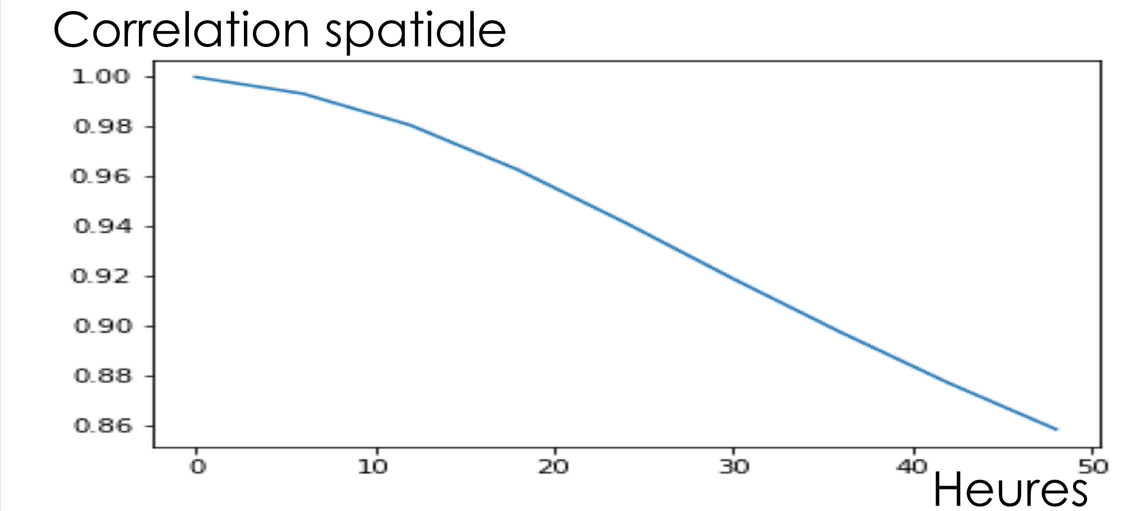
SCORE PRESSION

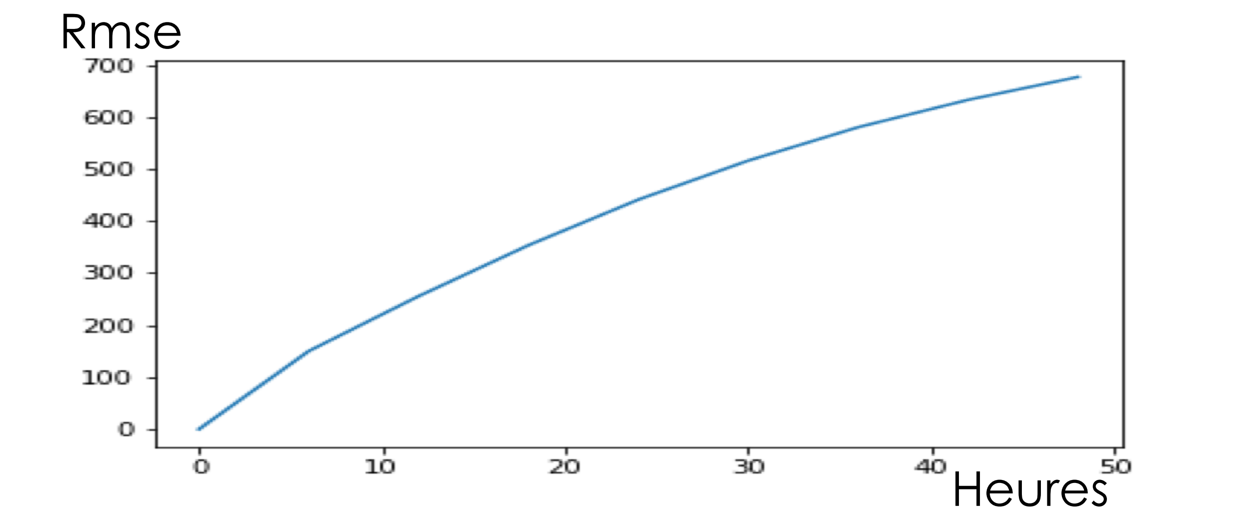
Pression seulement





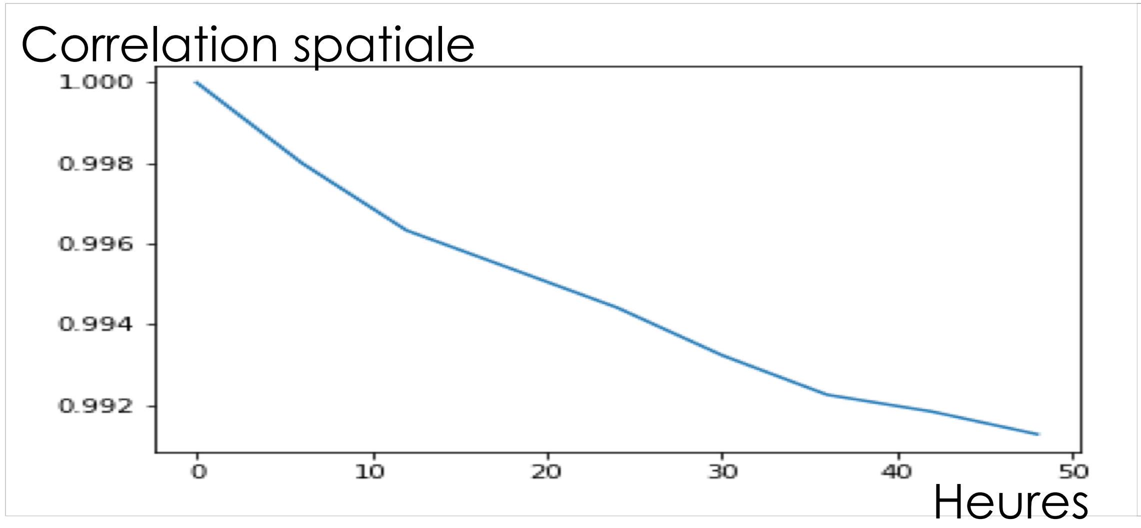
Pression + temperature

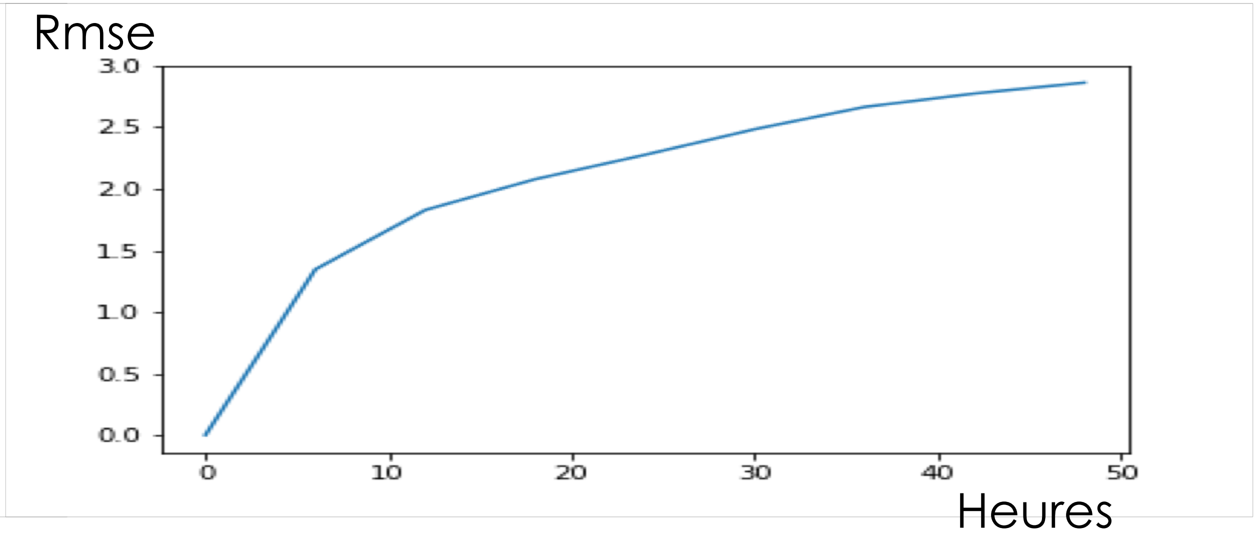




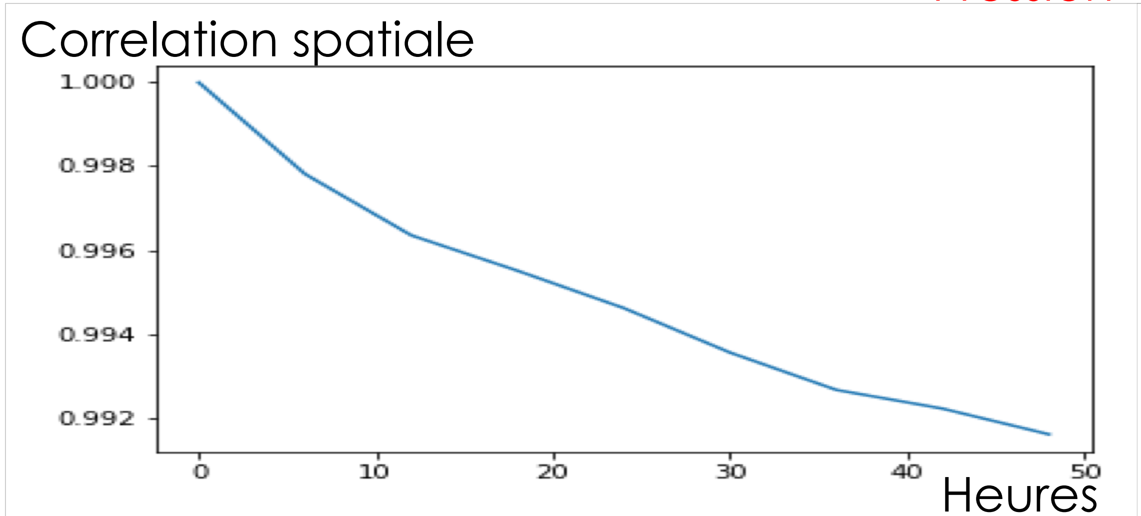
SCORE TEMPERATURE

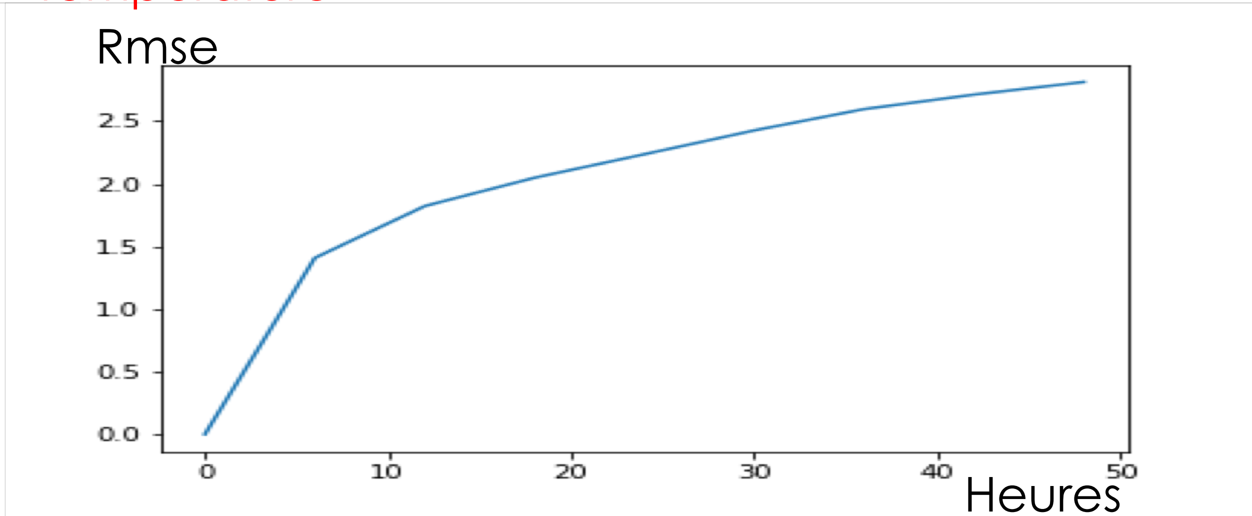
Température seulement





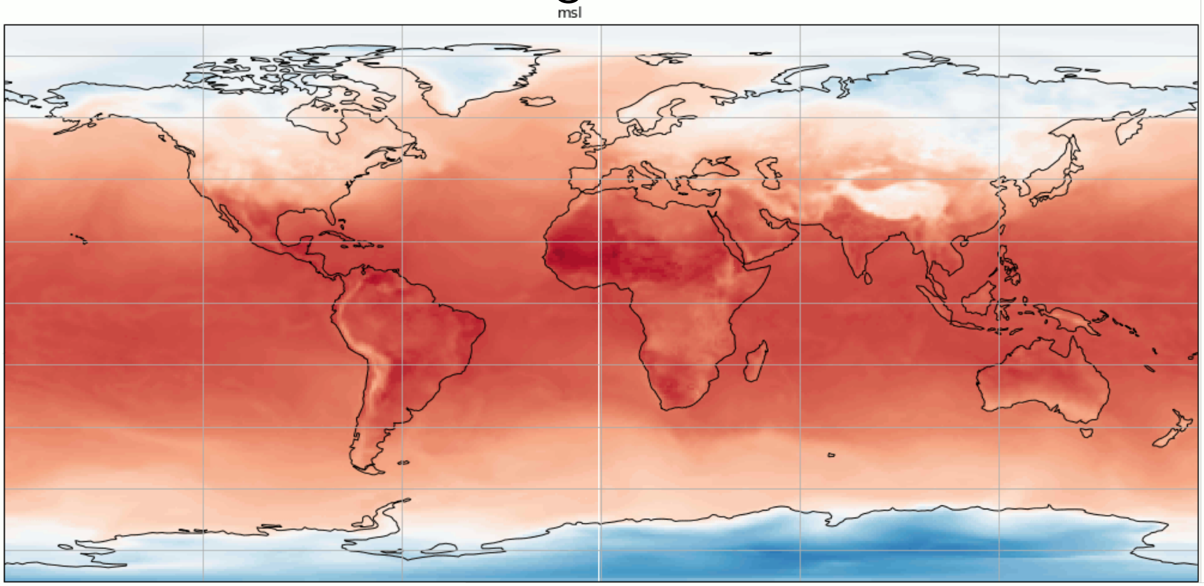
Pression + temperature



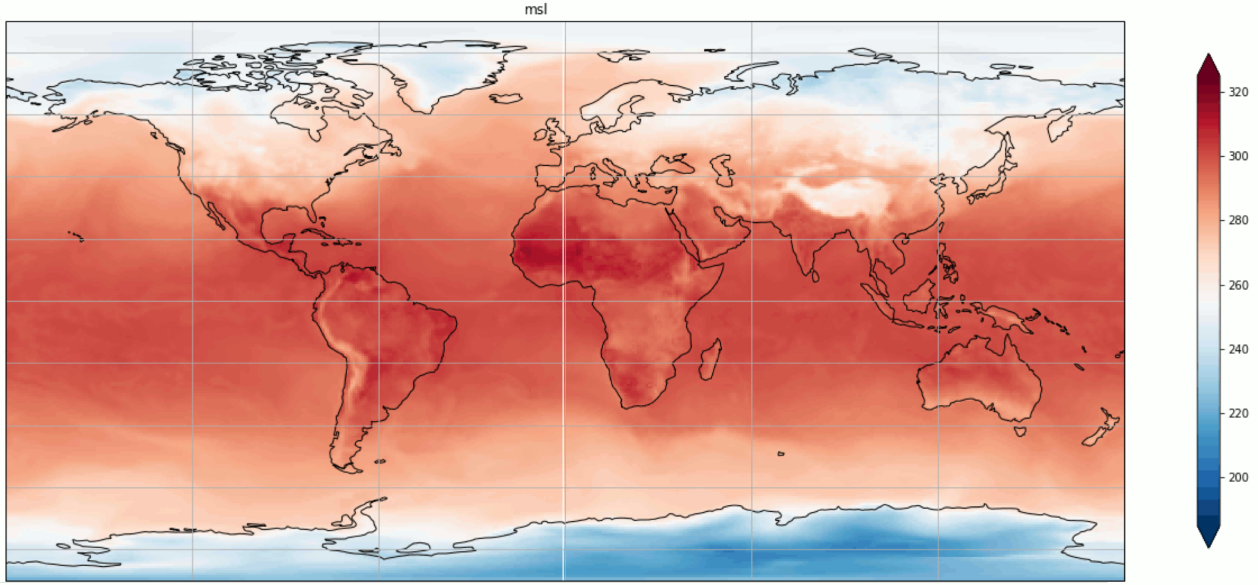


EVOLUTION DE LA TEMPERATURE

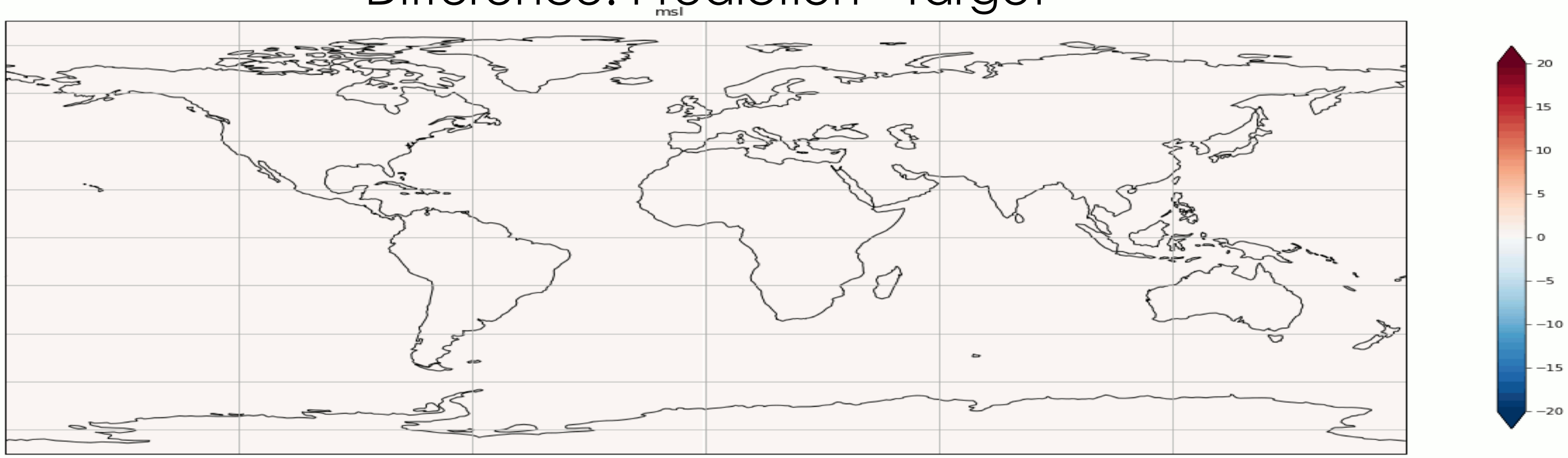
Target



Prediction



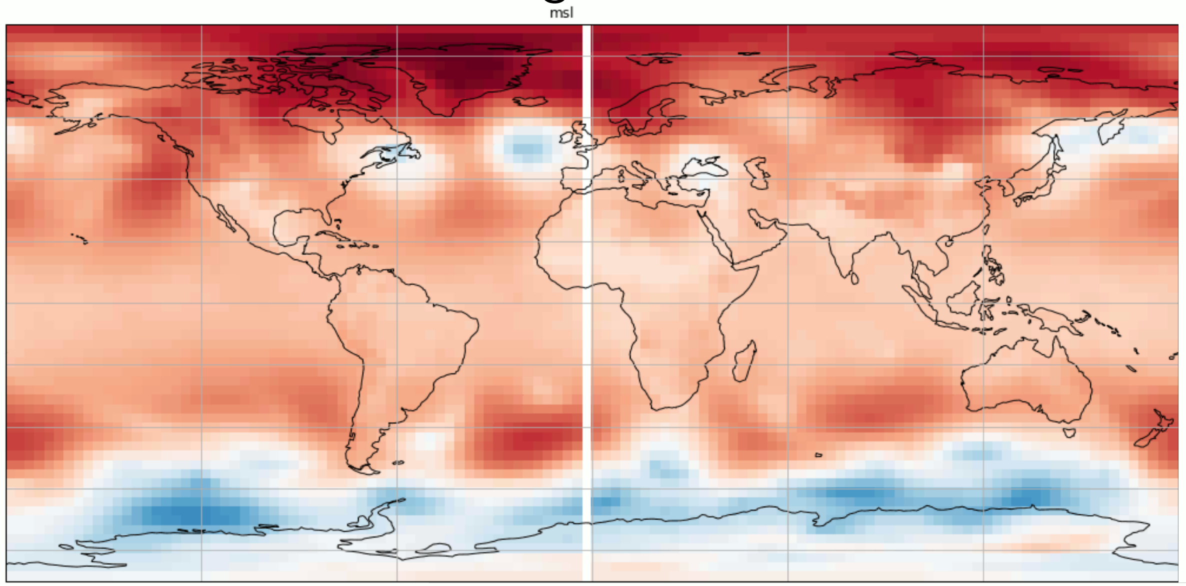
Difference: Prediction - Target



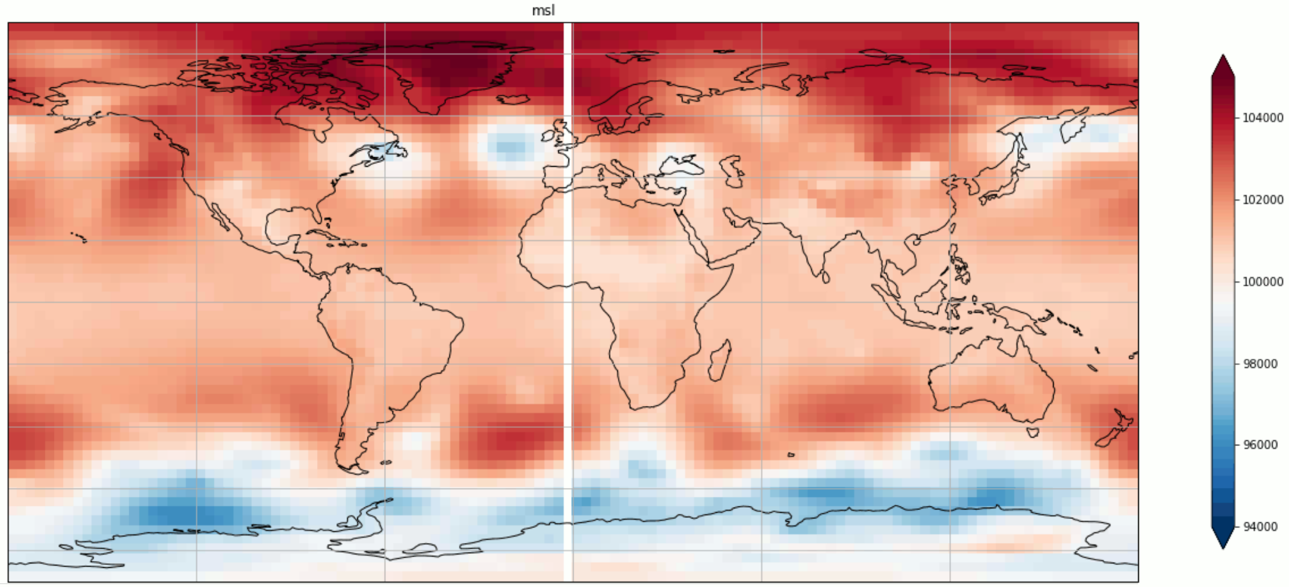
Unité: Kelvins
Durée: 5 jours
Données: Température

EVOLUTION DE LA PRESSION

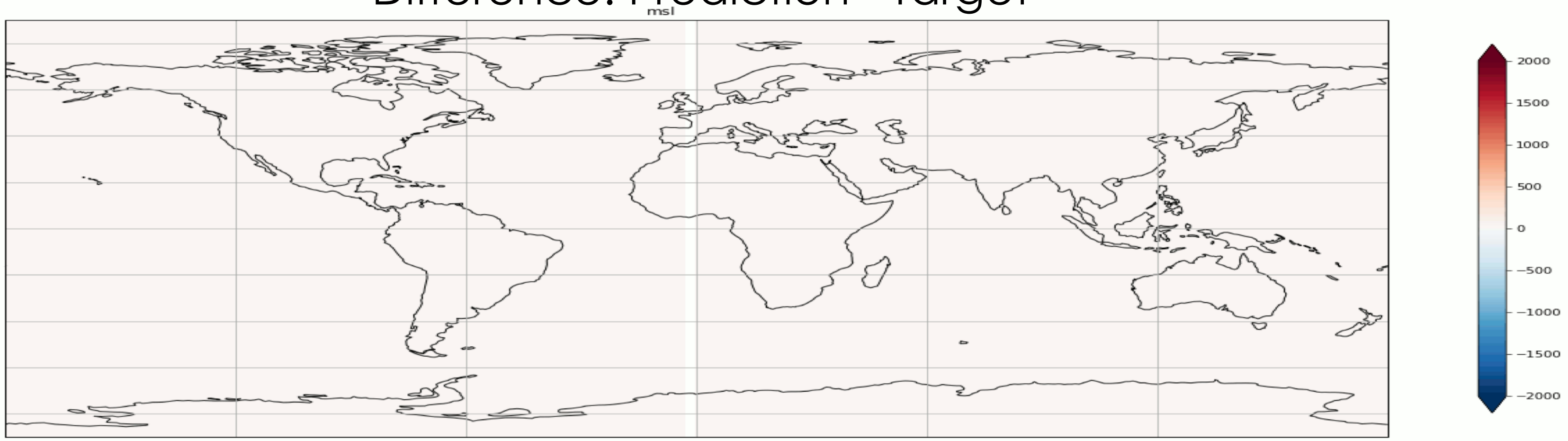
Target



Prediction



Difference: Prediction - Target



Unité: Pascals
Durée: 48h
Données: Pression