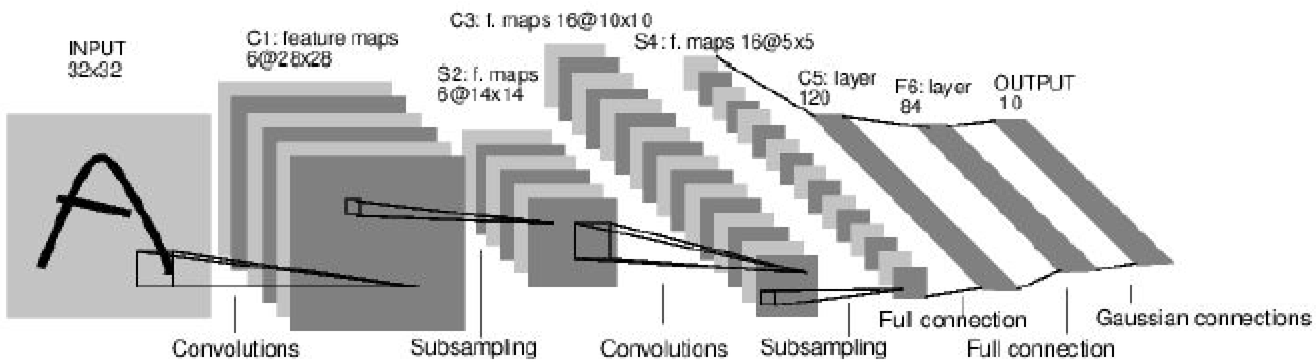
# Réseaux de neurones récurrents

# Rappels

- Réseaux convolutionnels : traiter la dimension spatiale
  - Hypothèse : des pixels voisins représentent des choses similaires
  - Convolution : Connexion locale des pixels (voisinage) pour détecter des objets plus gros (lignes/courbes)




- Réseaux récurrents : Traiter la dimension temporelle ?

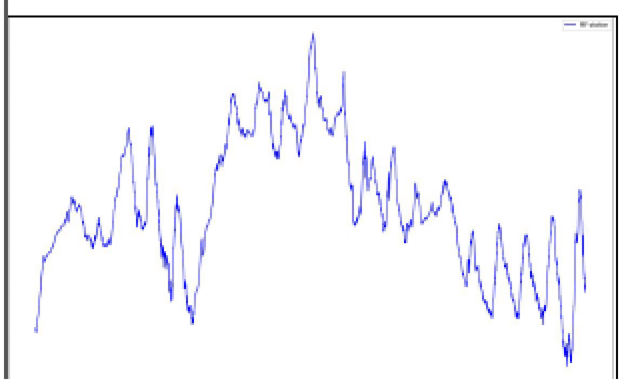
# Série Temporelles

Selon Wikipédia :
*Les séries temporelles sont considérées à tort comme étant une branche exclusive de l'économétrie. Cette dernière est une discipline qui est relativement jeune alors que les séries temporelles ont été utilisées bien avant par exemple en astronomie (1906) et en météorologie (1968).*

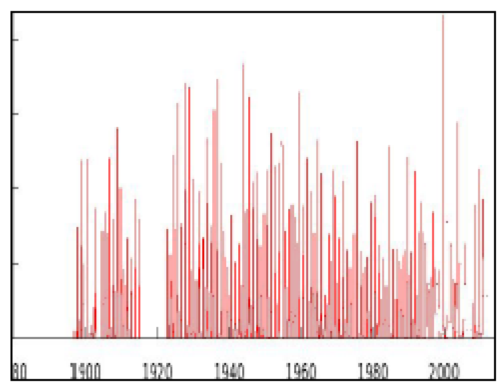
- L'objet des séries temporelles est l'étude des variables au cours du temps

Exemples :

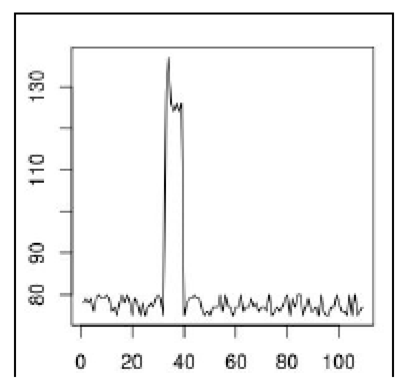
Température



Précipitation



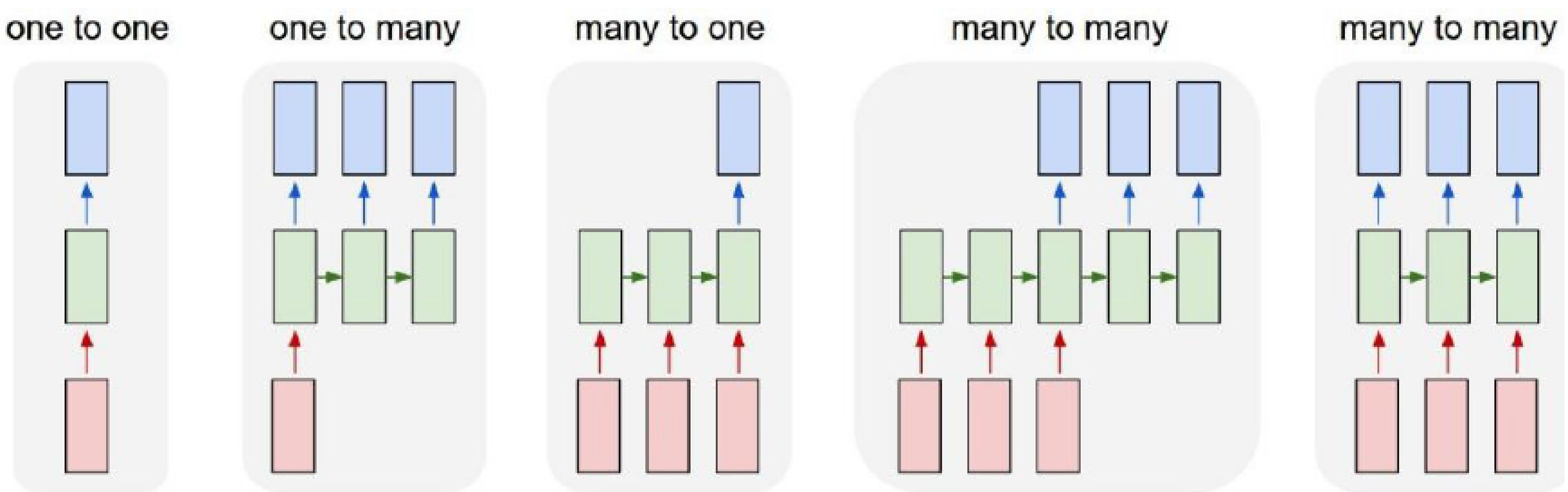

?



## Apprentissage avec des séries temporelles

- Classification
  - Identifier les épisodes pluvieux/non pluvieux
  - Identifier si une personne fait du sport, dort, etc ...
  - Identifier si une station météo est défaillante.
- Regression
  - Prédire la température maximale de la journée
  - Prédire la quantité de pluie attendue
  - Corriger la température mesurée

# Les différents types de RNN




Réseaux de neurones **standards**

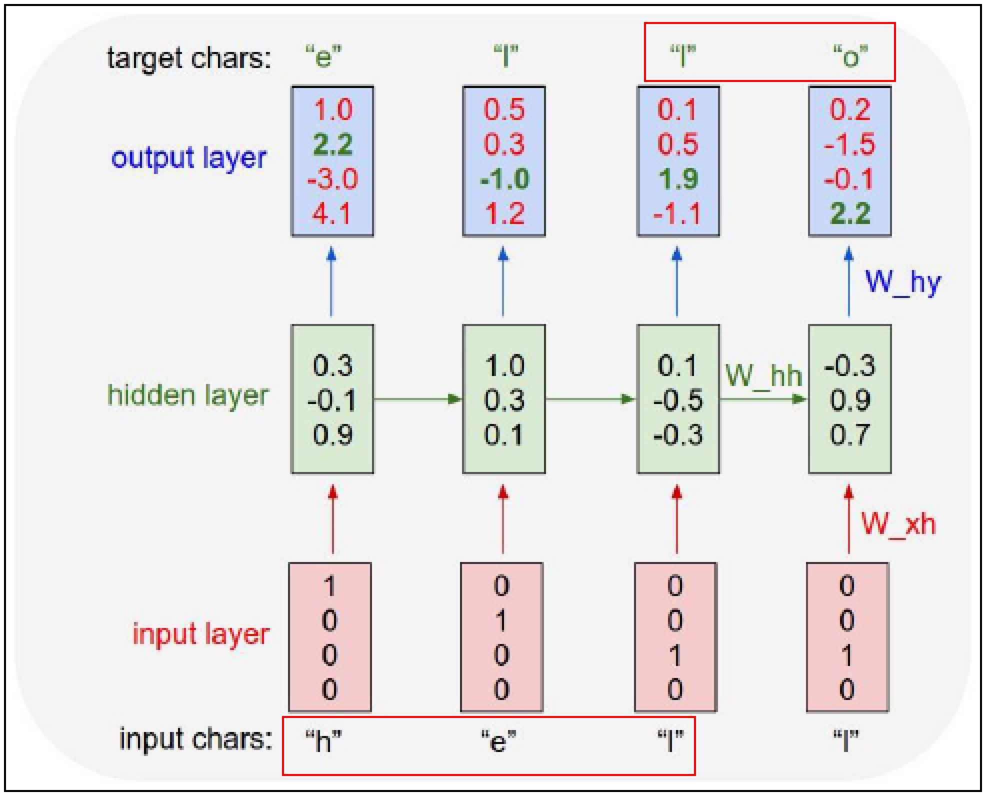
Résumer une **image en texte**

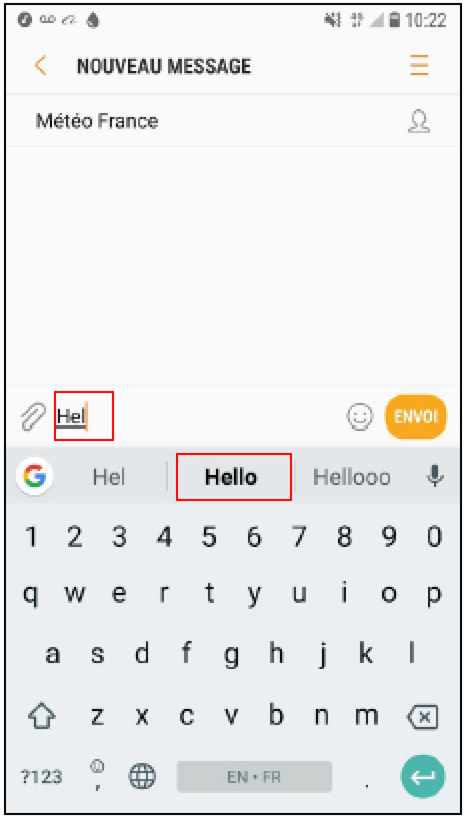
Transformer un **texte en image**

- Prédiction de lettres
- **Prédiction** de températures

**Correction** de Température

# Exemple : prédiciton de lettre

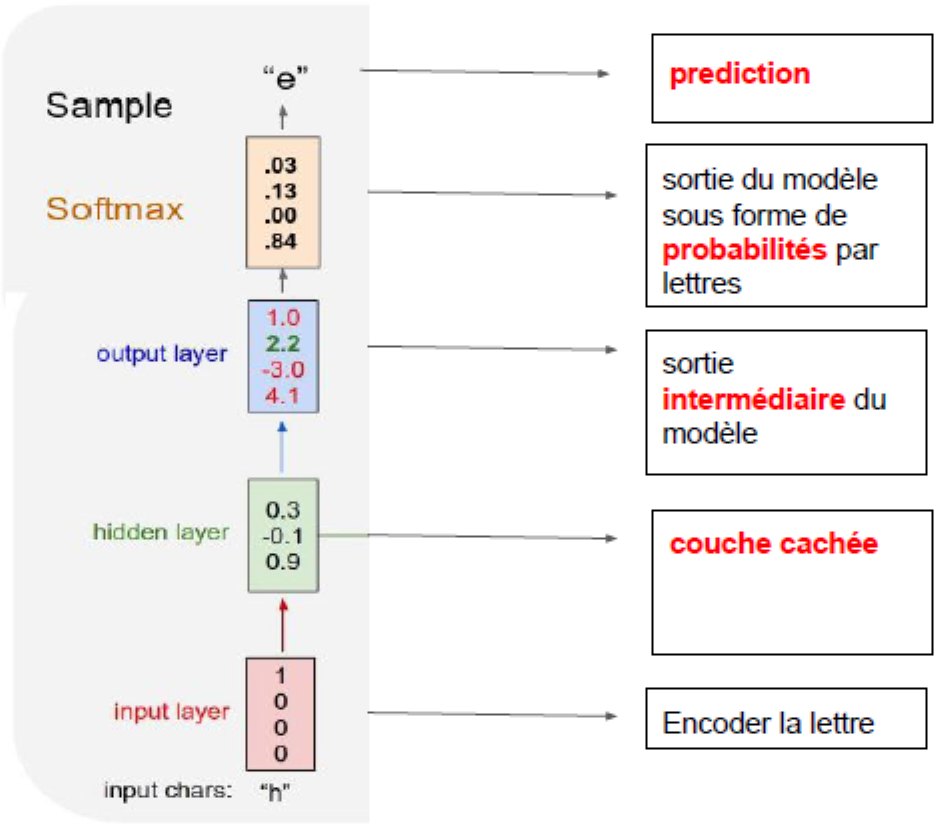


# Exemple : prédiction de lettre

Première période : Réseau de neurone classique

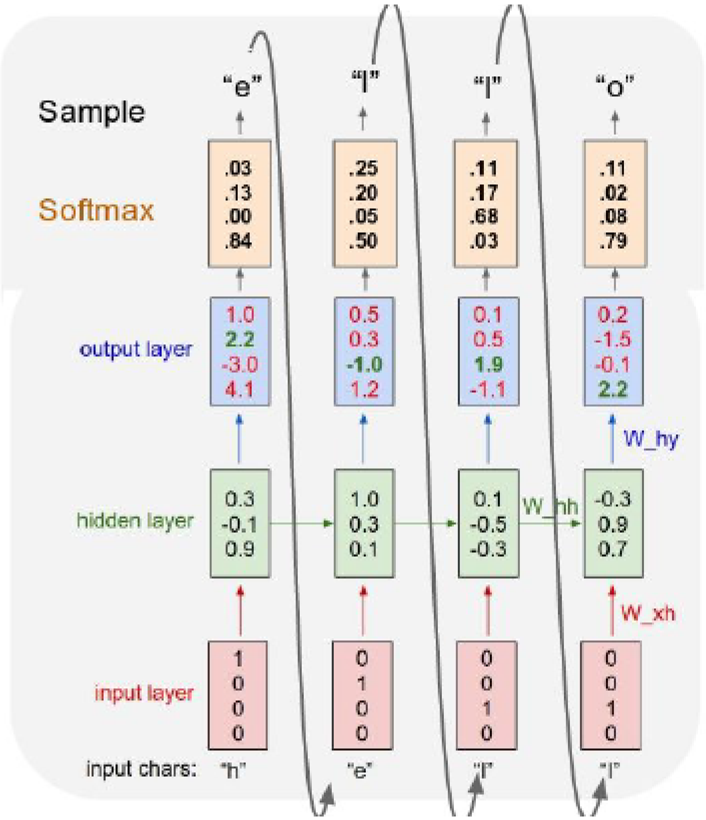
- La couche cachée est appélée "cellule"
- Il y a une sortie par période



# Exemple : prédiction de lettres

Périodes suivantes - réseau de neurone presque classique :

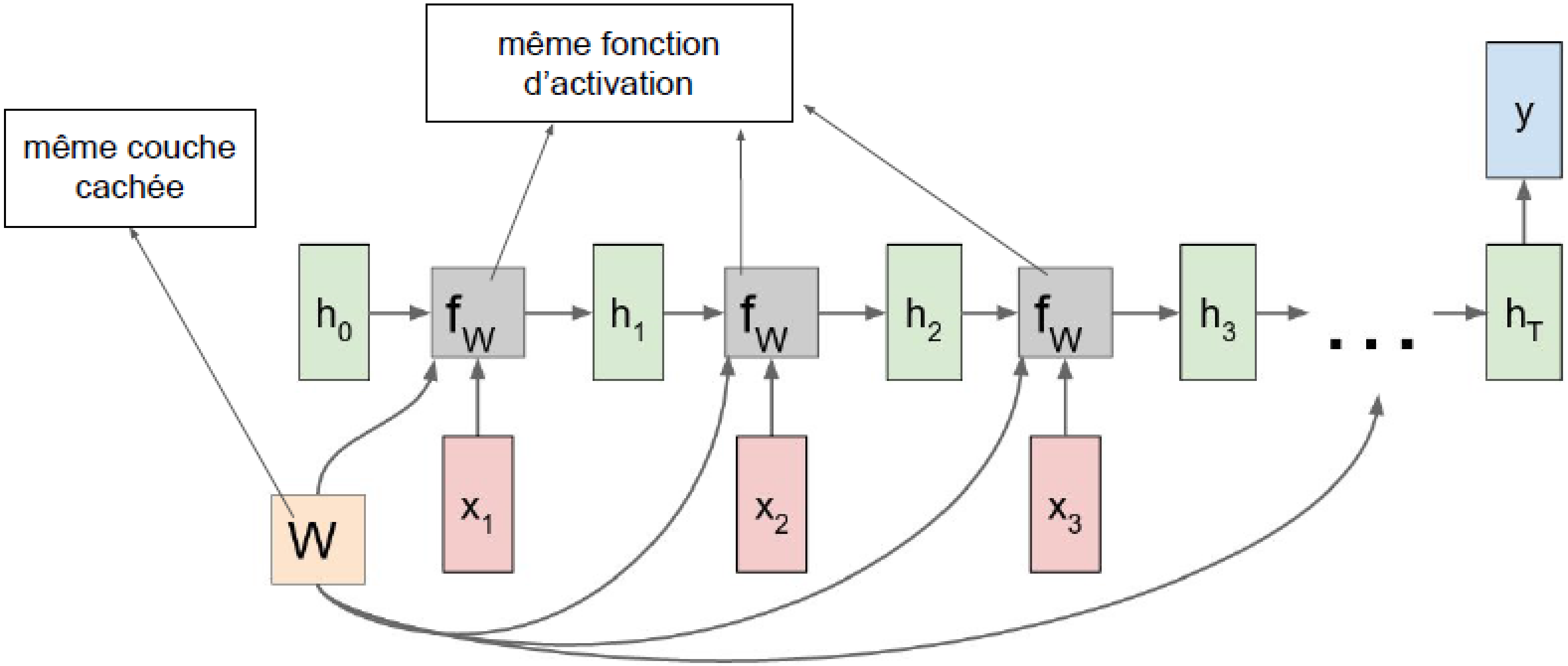
- la couche intermédiaire a une entrée supplémentaire : la sortie intermédiaire du réseau précédent



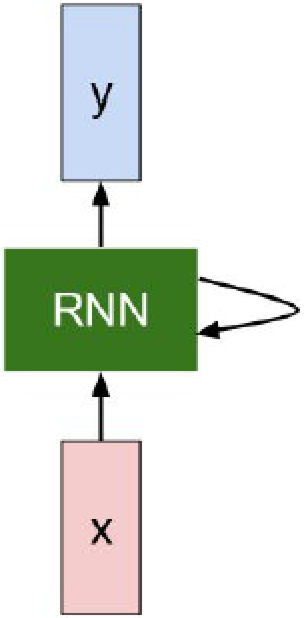

1. utiliser la prédiction au temps précédent
2. utiliser la couche intermédiaire du temps précédent

La **tâche** de prédiction de lettre est **similaire** à chaque instant. Pourquoi ne pas utiliser le même réseaux ?

# Représentation en graphe



## Représentation enroulée




$$h_t = f_W(h_{t-1}, x_t)$$

$$h_t = \tanh(W_{hh}h_{t-1} + W_{xh}x_t)$$

$$y_t = W_{hy}h_t$$

## Autres exemples : Générer du Shakespeare

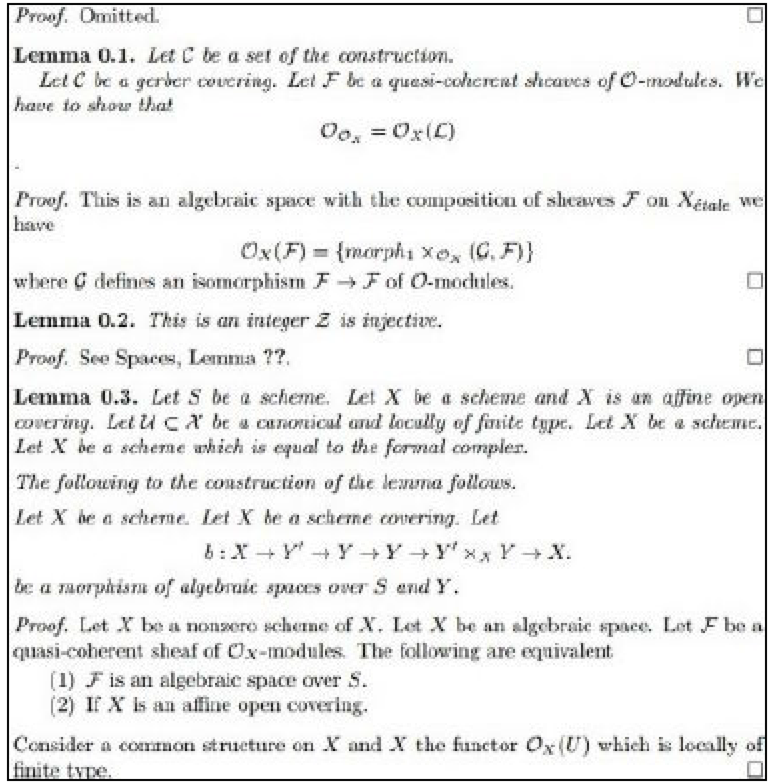
```
VIOLA:
Why, Salisbury must find his flesh and thought
That which I am not aps, not a man and in fire,
To show the reining of the raven and the wars
To grace my hand reproach within, and not a fair are hand,
That Caesar and my goodly father's world;
When I was heaven of presence and our fleets,
We spare with hours, but cut thy council I am great,
Murdered and by thy master's ready there
My power to give thee but so much as hell:
Some service in the noble bondman here,
Would show him to her wine.

KING LEAR:
O, if you were a feeble sight, the courtesy of your law,
Your sight and several breath, will wear the gods
With his heads, and my hands are wonder'd at the deeds,
So drop upon your lordship's head, and your opinion
Shall be against your honour.
```

Comprend le concept de paragraphe, majuscules, saut de lignes ...

- Entraînement : tous les textes de Shakepeare
- Puis on donne au réseau une première lettre, et il prédit la suite

# Autres exemples : générer du code LaTex



*Proof.* Omitted. □

**Lemma 0.1.** *Let $\mathcal{C}$ be a set of the construction.*

*Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that*

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

*Proof.* This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

**Lemma 0.2.** *This is an integer $\mathcal{Z}$ is injective.*

*Proof.* See Spaces, Lemma ??. □

**Lemma 0.3.** *Let $S$ be a scheme. Let $X$ be a scheme and $X$ is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let $X$ be a scheme. Let $X$ be a scheme which is equal to the formal complex.*

*The following to the construction of the lemma follows.*

*Let $X$ be a scheme. Let $X$ be a scheme covering. Let*

$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

*be a morphism of algebraic spaces over $S$ and $Y$.*

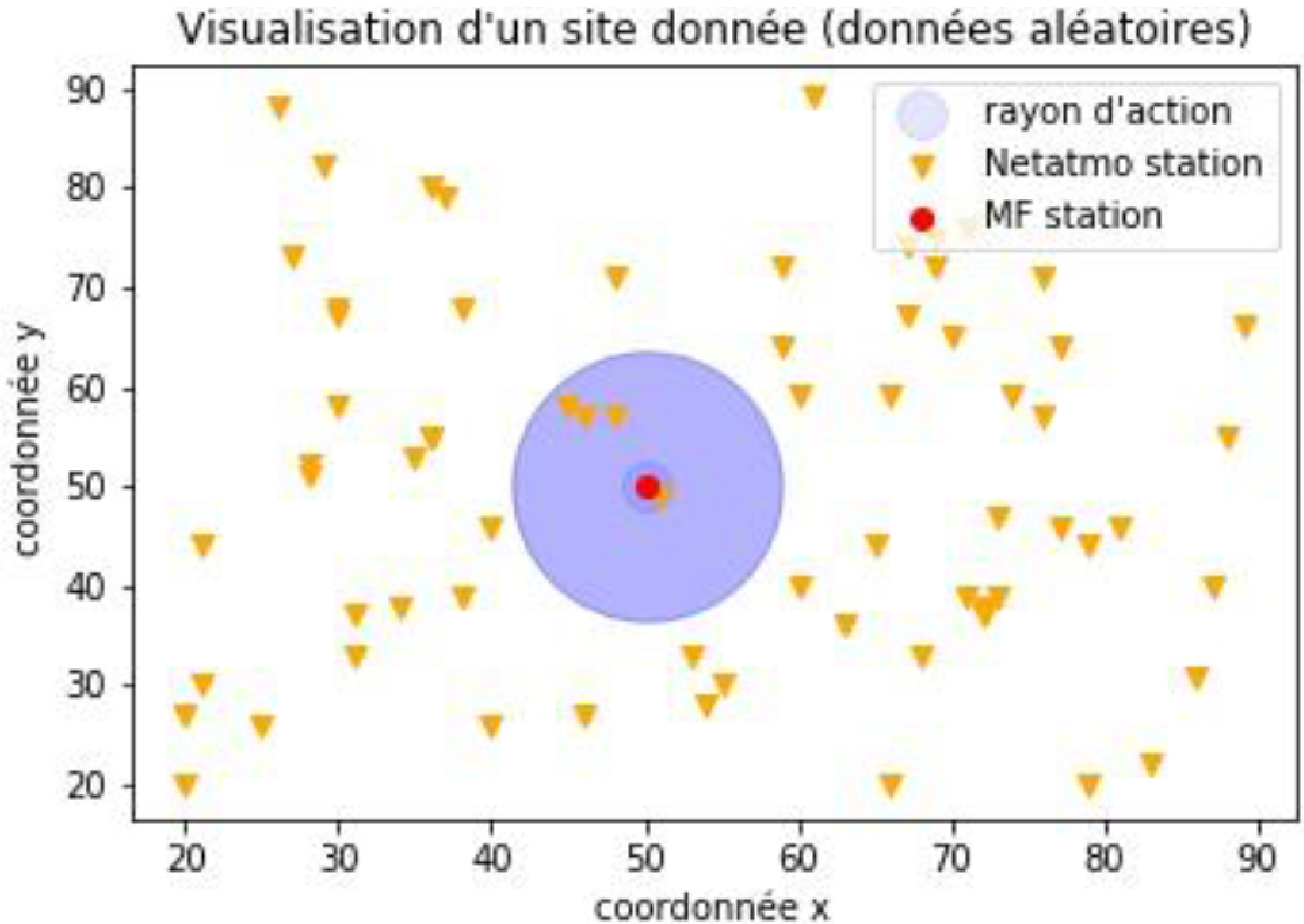
*Proof.* Let $X$ be a nonzero scheme of $X$. Let $X$ be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over $S$.
(2) If $X$ is an affine open covering.

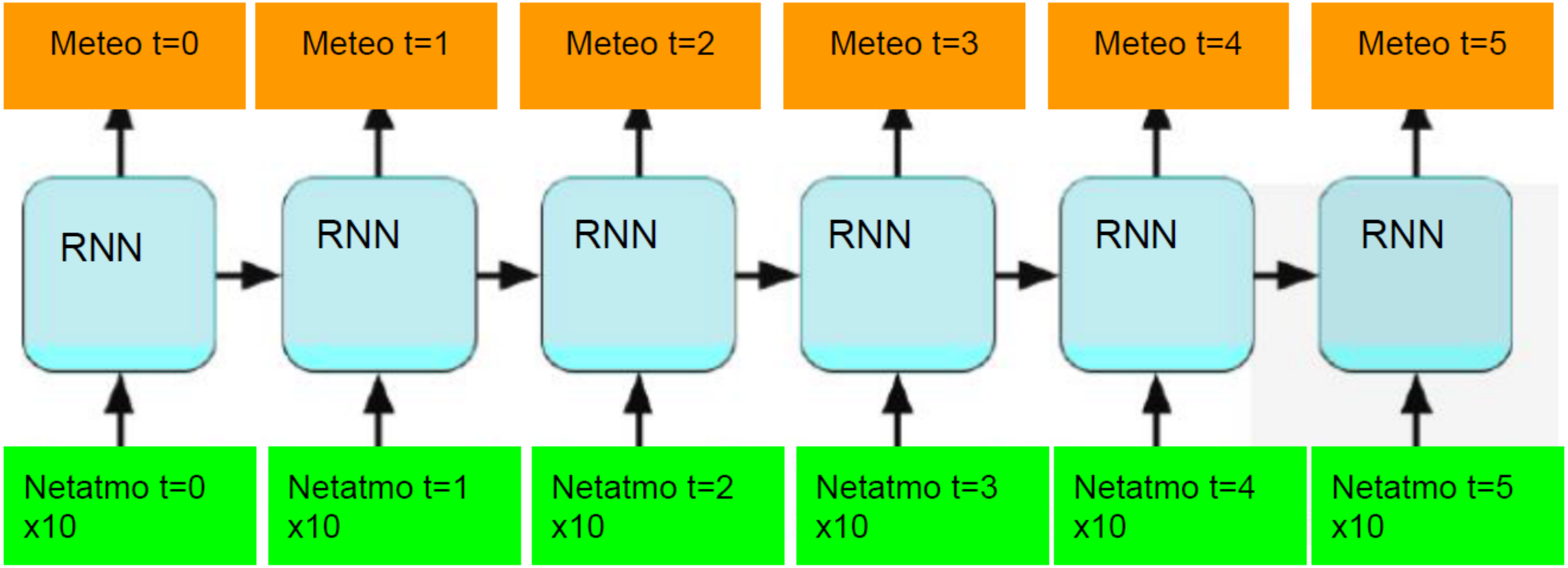
Consider a common structure on $X$ and $X$ the functor $\mathcal{O}_X(U)$ which is locally of finite type. □

- Le réseau tente de démontrer les lemmes
- Il reproduit la structure d'un cours de maths
- Mais ça ne veut rien dire !

# Étude de cas NetAtmo

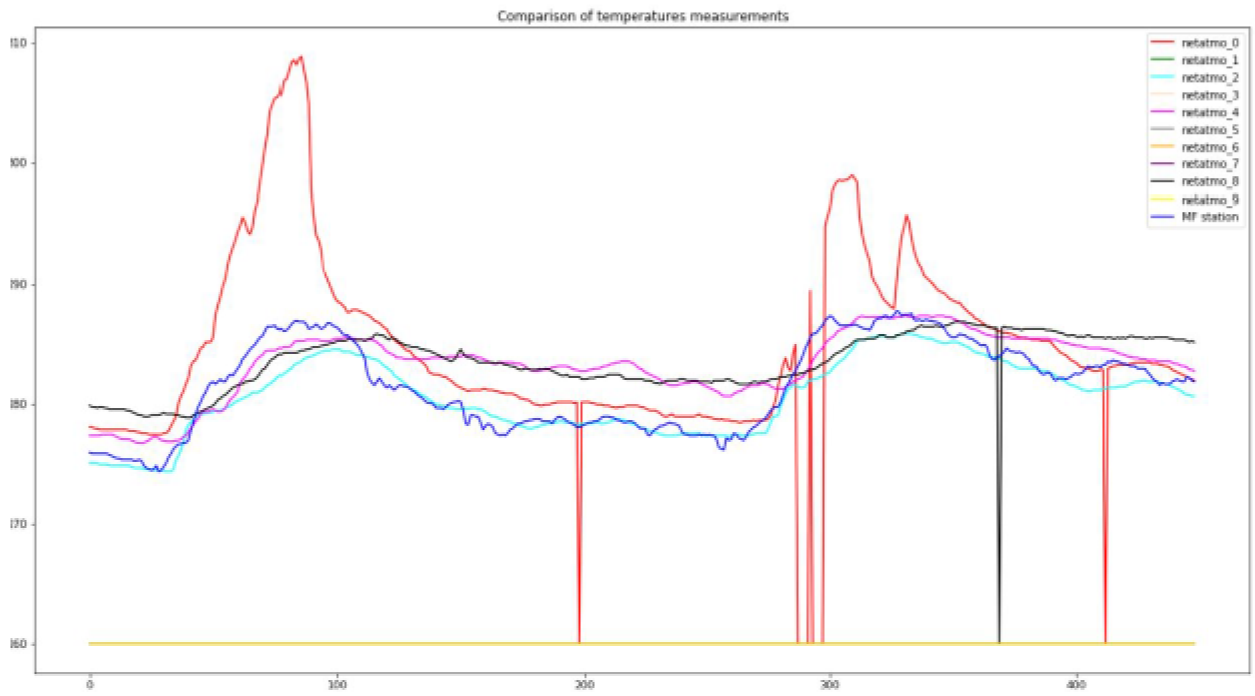


# Étude de cas NetAtmo



# Étude de cas NetAtmo

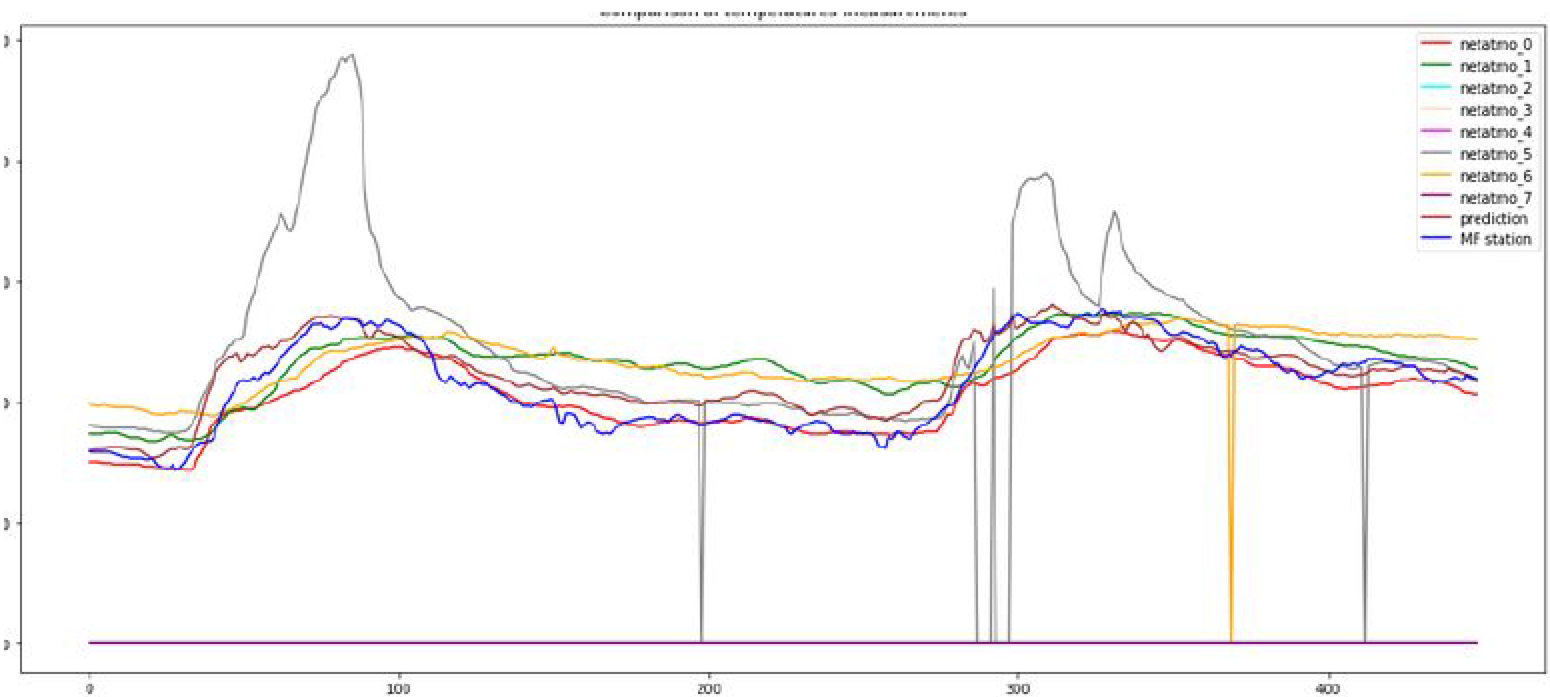
**Objectif : deviner la courbe bleue (Météo-France) à partir des autres courbes (NetAtmo)**



# Étude de cas NetAtmo

Fréjus Janvier 30h

Erreur moyenne : 1.67°C




- La prédiciton est robuste aux pics anormaux de températures des stations NetAtmo (grises)

# Bonus : code Keras

```
model=Sequential()
nb_netatmos=10
model.add(SimpleRNN(1,return_sequences=False,input_shape=(None,nb_netatmos)))
model.summary()
```

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
simple_rnn_5 (SimpleRNN)     (None, 1)                 12
=================================================================
Total params: 12
Trainable params: 12
Non-trainable params: 0
```

- le paramètre return_sequences permet de spécifier si on veut uniquement la sortie de la dernière période (False) ou toute la séquence y compris périodes intermédiaires (True).