Réseaux de convolution sur graphes et application aux données spatio-temporelles

Stage L3

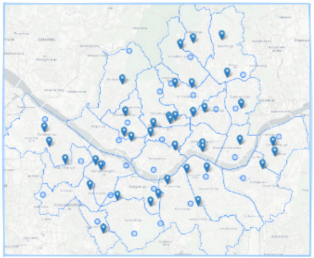
Amer Essakine et Quentin Boiret

Encadré par : Argyris Kalogeratos et Xavier Cassagnou

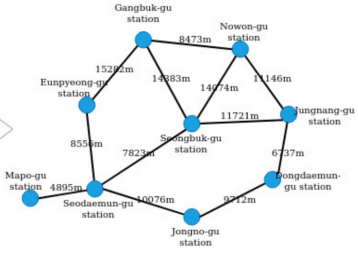
CENTRE BOREL LI

école normale supérieure paris—saclay

Données spatio-temporelles



Seoul city map with air pollution monitoring stations

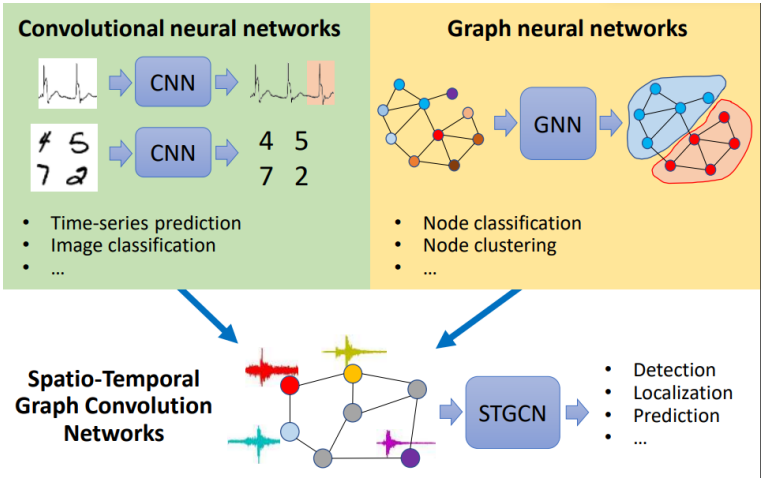


A Graph structure from monitoring stations

1

[1] Van-Duc Le,Tien-Cuong Bui,Sang-Kyun Cha,*Spatiotemporal Graph Convolutional Recurrent Neural Network Model for Citywide Air Pollution Forecasting*

Réseaux de convolution sur graphes



Progression du stage

Suivi des progrès et réalisations :

- ✓ Machine Learning
- ✓ Apprentissage Profond
- ✓ Réseaux de convolutions
- Graph Neural Networks
- Données spatio-temporelles

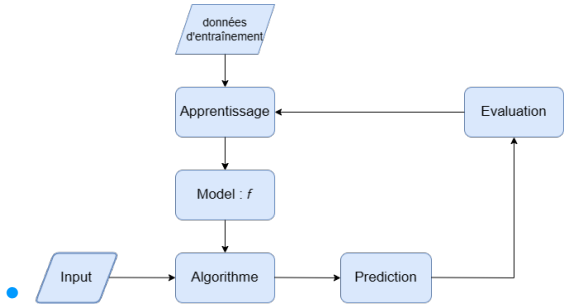


Figure: Machine Learning

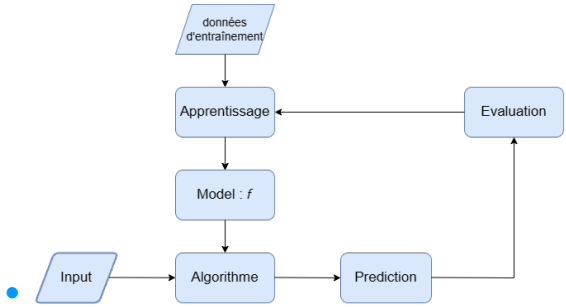
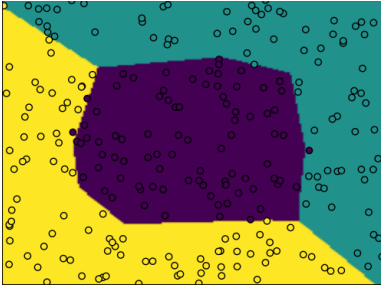


Figure: Machine Learning

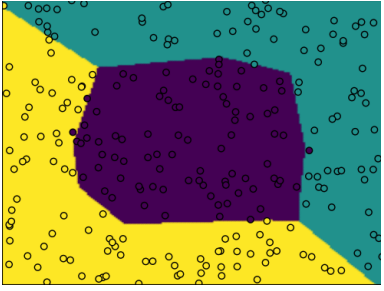
- **Mathématiquement**, nous considérons un ensemble d'entrées $x_1, \ldots, x_n$ et les attributs correspondants $y_1, \ldots, y_n$. L'objectif est de déterminer la distribution de la probabilité conditionnelle des points (x_i) étant donné qu'ils sont associés aux attributs (y_i).

Classification et Regression

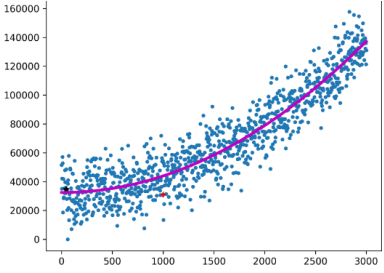


- Classification d'images
- Reconnaissance vocale
- Identification de la langue

Classification et Regression



- Classification d'images
- Reconnaissance vocale
- Identification de la langue



- Prévision de séries temporelles
- Prédiction du prix des maisons
- Prédiction de la consommation d'énergie

Principe de fonctionnement

Dans un algorithme d'apprentissage, nous recherchons une fonction f telle que $f(x) = y$. **Le principe de fonctionnement est le suivant :**

Principe de fonctionnement

Dans un algorithme d'apprentissage, nous recherchons une fonction f telle que $f(x) = y$. **Le principe de fonctionnement est le suivant :**

- Nous définissons un modèle paramétré $f(.; w)$.
- Nous cherchons la valeur optimale w^* telle que, pour n'importe quelle entrée x, $f(x)$ se rapproche au mieux de l'attribut y.

Principe de fonctionnement

Dans un algorithme d'apprentissage, nous recherchons une fonction f telle que $f(x) = y$. **Le principe de fonctionnement est le suivant :**

- Nous définissons un modèle paramétré $f(.; w)$.
- Nous cherchons la valeur optimale w^* telle que, pour n'importe quelle entrée x, $f(x)$ se rapproche au mieux de l'attribut y.

On définit une **fonction de perte** pour formaliser à quel degré $f(.; w)$ est une bonne approximation des exemples:

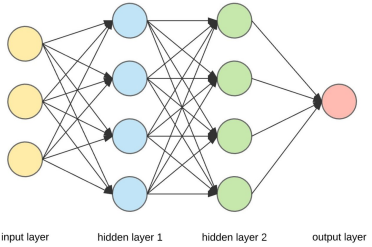
- Erreur quadratique moyenne (MSE) : $J(w) = \frac{1}{N} \sum_{n=1}^{N} (y_n - f(x_n; w))^2$.
- Entropie croisée catégorielle : $J(w) = \frac{1}{n} \sum_{n=1}^{N} \sum_{c=1}^{C} y_c \log(f(x_n; w)$.

Apprentissage profond

Dans un réseau d'apprentissage profond, on peut écrire $f = f_1 \circ \ldots \circ f_n$, où chaque f_n est appelée une couche.

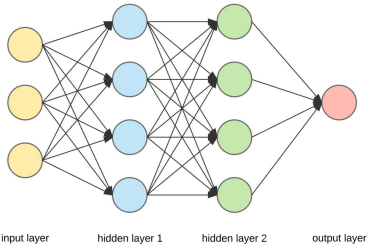
Apprentissage profond

Dans un réseau d'apprentissage profond, on peut écrire $f = f_1 \circ \ldots \circ f_n$, où chaque f_n est appelée une couche.



Apprentissage profond

Dans un réseau d'apprentissage profond, on peut écrire $f = f_1 \circ \ldots \circ f_n$, où chaque f_n est appelée une couche.



Un exemple des couches utilisées est la couche linéaire, où $f(x) = W \cdot x + b$ et W ainsi que b sont des paramètres à optimiser.

Couches

En général, f_i est la combinaison d'une transformation linéaire et d'une fonction appelée fonction d'activation.

Couches

En général, f_i est la combinaison d'une transformation linéaire et d'une fonction appelée fonction d'activation.

- **Output layer**
  - sigmoid : $\sigma(x) = \frac{1}{1+\exp(x)}$, a valeurs dans $[0, 1]$
  - softmax : fonction a valeurs dans R^n, $softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$

Couches

En général, f_i est la combinaison d'une transformation linéaire et d'une fonction appelée fonction d'activation.

- **Output layer**
  - sigmoid : $\sigma(x) = \frac{1}{1+\exp(x)}$, a valeurs dans $[0, 1]$
  - softmax : fonction a valeurs dans R^n, $softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$
- **Hidden layers**
  - Tangente hyperbolique
  - Relu : fonction presque linéaire, $Relu(x) = \mathbb{1}_{x>0}$

Couches

En général, f_i est la combinaison d'une transformation linéaire et d'une fonction appelée fonction d'activation.

- **Output layer**
  - sigmoid : $\sigma(x) = \frac{1}{1+\exp(x)}$, a valeurs dans $[0, 1]$
  - softmax : fonction a valeurs dans R^n, $softmax(x)_i = \frac{\exp(x_i)}{\sum_{j=1}^{n} \exp(x_j)}$
- **Hidden layers**
  - Tangente hyperbolique
  - Relu : fonction presque linéaire, $Relu(x) = \mathbb{1}_{x>0}$

Conception d'architecture

- Profondeur : nombre des couches
- Nombre de connexions entre les couches.

Capacite,Overfitting,Underfittng

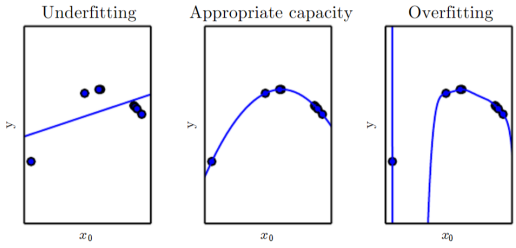


Figure: Overfitting

Capacite,Overfitting,Underfittng

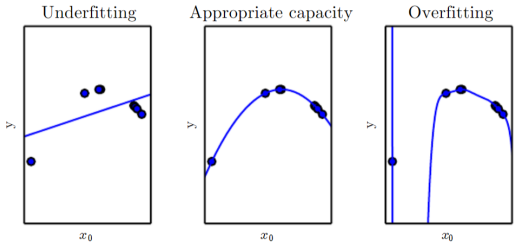
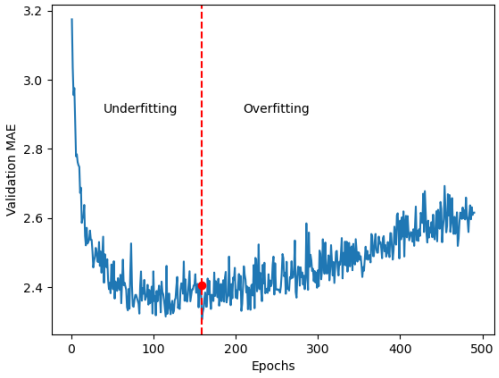


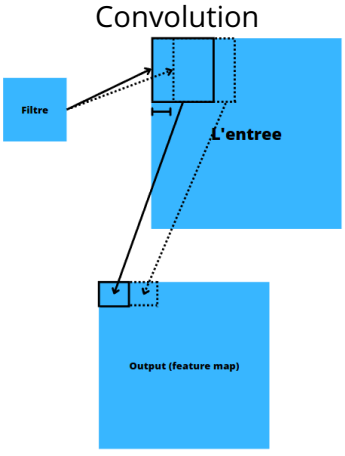
Figure: Overfitting

Nous entraînons un ensemble de données d'entraînement constitue de 404 échantillons pour prédire les prix des logements à Boston.

Résultats

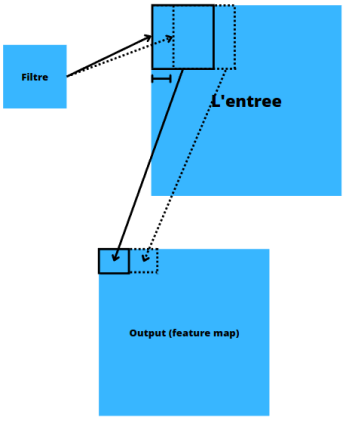

Underfitting
Overfitting
Validation MAE
Epochs
3.2
3.0
2.8
2.6
2.4
0
100
200
300
400
500

Réseaux de convolutions

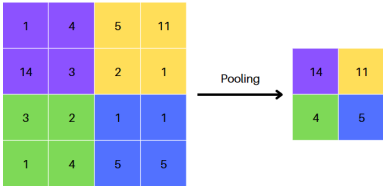


Réseaux de convolutions

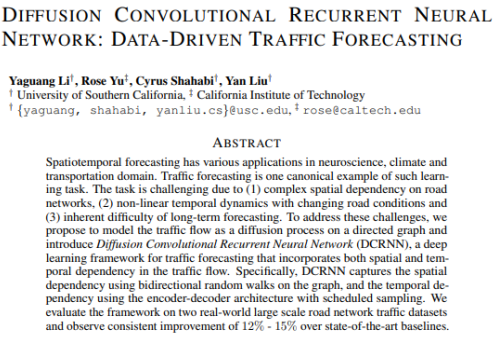
Convolution



Pooling



Objectives



DIFFUSION CONVOLUTIONAL RECURRENT NEURAL NETWORK: DATA-DRIVEN TRAFFIC FORECASTING

Yaguang Li[†], **Rose Yu**[‡], **Cyrus Shahabi**[†], **Yan Liu**[†]
[†] University of Southern California, [‡] California Institute of Technology
[†] {yaguang, shahabi, yanliu.cs}@usc.edu, [‡] rose@caltech.edu

ABSTRACT

Spatiotemporal forecasting has various applications in neuroscience, climate and transportation domain. Traffic forecasting is one canonical example of such learning task. The task is challenging due to (1) complex spatial dependency on road networks, (2) non-linear temporal dynamics with changing road conditions and (3) inherent difficulty of long-term forecasting. To address these challenges, we propose to model the traffic flow as a diffusion process on a directed graph and introduce *Diffusion Convolutional Recurrent Neural Network* (DCRNN), a deep learning framework for traffic forecasting that incorporates both spatial and temporal dependency in the traffic flow. Specifically, DCRNN captures the spatial dependency using bidirectional random walks on the graph, and the temporal dependency using the encoder-decoder architecture with scheduled sampling. We evaluate the framework on two real-world large scale road network traffic datasets and observe consistent improvement of 12% - 15% over state-of-the-art baselines.

- Tester les performances de l'algorithme de l'article.
- Essayer d'améliorer l'algorithme.