

## **Apprentissage collaboratif dans les réseaux de capteurs (sans fil)**

Paul Honeine

Professeur, Laboratoire LITIS  
Université de Rouen Normandie

Journée "Réseaux de capteurs, phénotypage et modélisation"  
ModStatSAP - 14/12/2022

## Outline

- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus
- 3 Analyse en composantes principales dans les réseaux de capteurs
- 4 Apprentissage statistique dans les réseaux de capteurs
- 5 Épilogue

# Outline

- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus
- 3 Analyse en composantes principales dans les réseaux de capteurs
- 4 Apprentissage statistique dans les réseaux de capteurs
- 5 Épilogue

## Les réseaux de capteurs sans fil

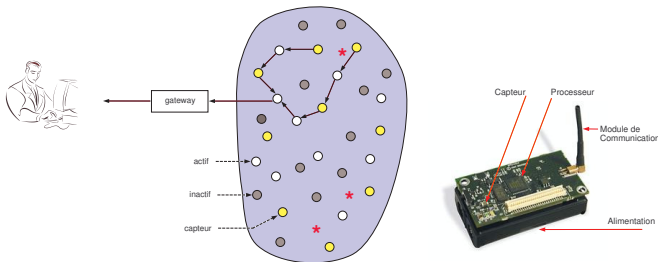
## Les réseaux de capteurs sans fil

Smart dust !

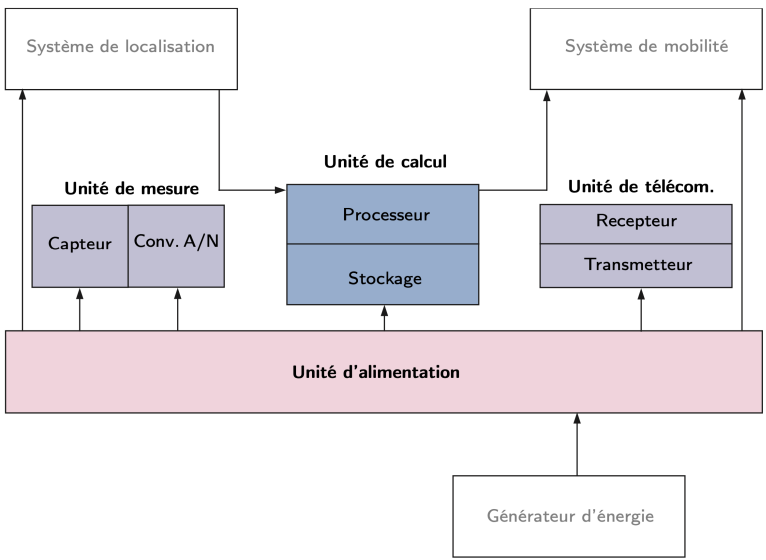
### Definition

Un réseau de capteurs sans fil est un réseau sans fil constitué de modules autonomes, répartis dans la région scrutée, utilisant des capteurs pour surveiller des conditions physiques ou environnementales, comme par exemple la température, le son, la vibration, la pression, le mouvement, ou encore la pollution, dans différents endroits.

"One of the 10 technologies that will change the world", MIT Technology Review, 2003.



# Le capteur sans-fil, le capteur intelligent



# Les réseaux de capteurs sans fil sont là !



LWIM III  
(UCLA, 1996)



WeC  
(Berkeley, 1999)



Rene  
(Berkeley, 2000)



Dot  
(Xbow, 2001)



Mica2  
(Xbow, 2002)



Telos  
(Moteiv, 2004)



Tmote  
(Moteiv, 2005)



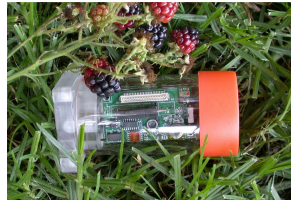
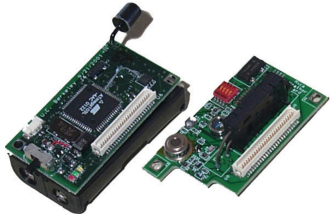
pParticle  
(Particle Comp., 2006)

...

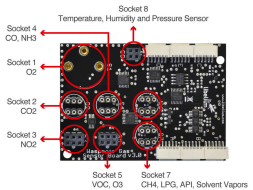
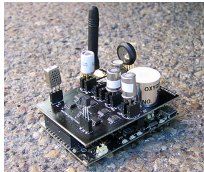
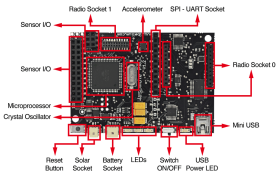
# Exemples de capteurs intelligents : extensibles

Des modules extensibles ...

- Mica Weather Board pour la surveillance environnementale avec le boîtier acrylique



- Libelium Wasp mote, avec son Gases Sensor Board





# Les réseaux de capteurs sans fil sont là ! Des capteurs sur étagère



Intégration des capteurs



... pour gaz



... pour liquide



... pour l'agriculture

**Applications for air monitoring:**

- Noise maps: Monitor in real time the acoustic levels in the streets of a city
  - Air quality: Detect the level of gases and particulates in the air
  - Waste management: Measure the garbage levels in bins to optimize the trash collection routes
- Such as:
- City pollution: CO, NO, NO<sub>2</sub>, O<sub>3</sub>, SO<sub>2</sub>, Particle Matter - Dust
  - Air Quality Index calculation: SO<sub>2</sub>, NO<sub>2</sub>, Particle Matter - Dust, CO, O<sub>3</sub>, NH<sub>3</sub>
  - Emissions from farms and hatcheries: CH<sub>4</sub>, H<sub>2</sub>S, NH<sub>3</sub>
  - Greenhouse management: CO<sub>2</sub>, CH<sub>4</sub>, Humidity
  - Control of chemical and industrial processes: CH<sub>4</sub>, SO<sub>2</sub>, CO<sub>2</sub>
  - Indoor air quality: CO<sub>2</sub>, CO, Particle Matter - Dust, O<sub>3</sub>
  - Forest fires: CO, CO<sub>2</sub>

**Applications for water monitoring:**

- Industrial and public sewage treatment plants
  - Wastewater management (nitrification and de-nitrification)
  - Surface water monitoring
  - Fish farming, aquaculture
  - Drinking water monitoring
  - Process engineering plants
- Such as:
- Potable water monitoring: pH, ORP, Dissolved Oxygen (DO), Nitrates, Phosphates
  - Chemical leakage detection in rivers: Extreme pH values signal chemical spills, Dissolved Oxygen (DO)
  - Swimming pool remote measurement: pH, Oxidation-Reduction Potential (ORP)
  - Pollution levels in the sea: Temperature, Conductivity (Salinity), pH, Dissolved Oxygen (DO) and Nitrates
- Including water ions monitoring:
- Drinking water quality control: Calcium (Ca<sup>2+</sup>), Iodide (I<sup>-</sup>), Chloride (Cl<sup>-</sup>), Nitrate (NO<sub>3</sub><sup>-</sup>), Magnesium (Mg<sup>2+</sup>), Sodium (Na<sup>+</sup>), pH
  - Agriculture water monitoring: Calcium (Ca<sup>2+</sup>), Nitrate (NO<sub>3</sub><sup>-</sup>), Magnesium (Mg<sup>2+</sup>), Sodium (Na<sup>+</sup>), Potassium (K<sup>+</sup>), Ammonium (NH<sub>4</sub><sup>+</sup>), pH
  - Swimming pools: Bromide (Br<sup>-</sup>), Chloride (Cl<sup>-</sup>), Fluoride (F<sup>-</sup>), pH
  - Waste water treatment: Cupric (Cu<sup>2+</sup>), Silver (Ag<sup>+</sup>), Lithium (Li<sup>+</sup>), Nitrite (NO<sub>2</sub><sup>-</sup>), Perchlorate (ClO<sub>4</sub><sup>-</sup>), pH

**Applications for agriculture:**

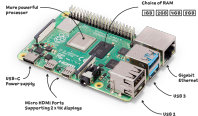
- Precision Agriculture: Leaf temperature, fruit diameter
  - Irrigation systems: Soil moisture, leaf wetness
  - Greenhouses: Solar radiation, humidity, temperature
  - Weather stations: Anemometer, wind vane, pluviometer
- Examples of used sensors:
- Air Temperature, Humidity and Pressure
  - Soil Temperature / Moisture
  - Leaf Wetness
  - Atmospheric Pressure
  - Solar Radiation - PAR
  - Ultraviolet Radiation - UV
  - Trunk Diameter
  - Stem Diameter
  - Fruit Diameter
  - Anemometer
  - Wind Vane

# Les réseaux de capteurs sans fil sont là !

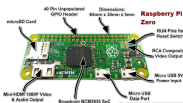
## Une évolution remarquable



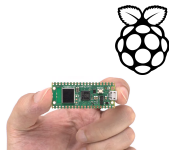
Raspberry Pi (2012)



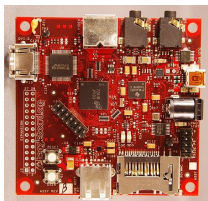
Raspberry Pi 4 (2019, 35€)



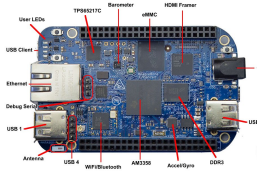
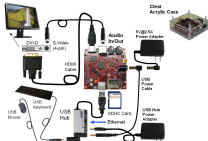
Raspberry Pi Zero (2015-2021, 15€)



Raspberry Pi Pico (2021, 4€)



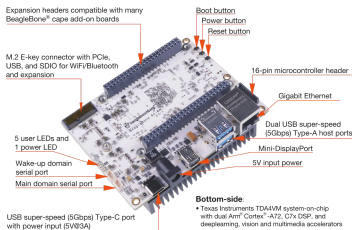
BeagleBoard (TI, 2008)



BeagleBone Black (TI, 2013)



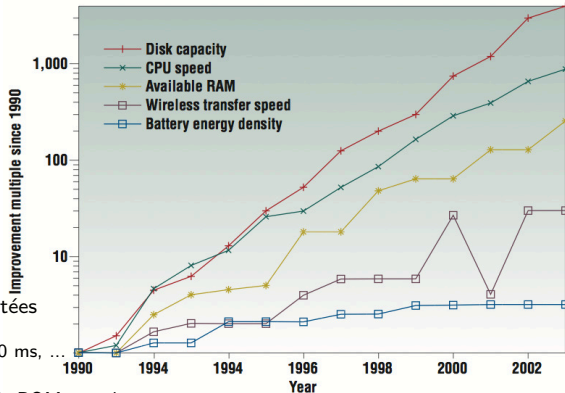
PocketBeagle (TI, 2017, 50€)



BeagleBone AI-64 (TI, 2019, 160€)



## Défis dans les réseaux de capteurs sans fil



### ① Capacités calculatoires limitées

- Faible vitesse (8 MHz)
- FFT de 512 points en 450 ms, ...

### ② Mémoire limitée

- 10 KB de RAM, 60 KB de ROM pour le programme
- Une bonne partie étant réservée au système d'exploitation

### ③ Possibilité d'un stockage extensible

- Certains modèles prennent en charge des mémoires flash MicroSD (ex. 2 Go)
- Mais coûteux en accès : 13 ms en lecture/écriture par block de 512 octets ;  $\approx 25$  mA

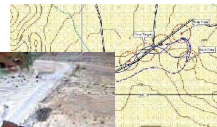
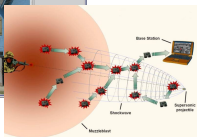
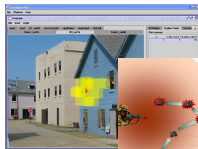
### ④ Communication radio à faible puissance : "ZigBee"

- IEEE 802.15.4. Meilleures performances : 100 Kbits/s (transmission à un seul nœud, pas d'interférence, courte portée)
- Environ 50 m de portée, et très peu fiable

## Domaines d'application : 1 - Applications militaires

### 1 - Applications militaires

- Surveillance des forces militaires, équipements et munitions
- Surveillance des champs de bataille, exploration du terrain, suivi de cibles
- Détection d'attaques nucléaires, radiologiques, biologiques et chimiques (NRBC)



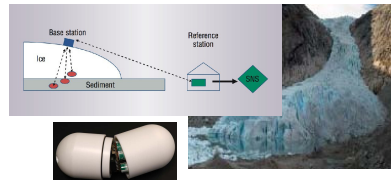
# Domaines d'application : 2 - Surveillance environnementale

## 2- Surveillance de l'environnement

- Agriculture de précision, modélisation de l'habitat, suivi de la faune
- Avertissement d'inondation et de feux de forêt
- Surveillance des calottes glaciaires et des glaciers



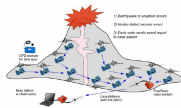
Détection de feux de forêt



Surveillance des glaciers



Agriculture de précision



Surveillance des volcans



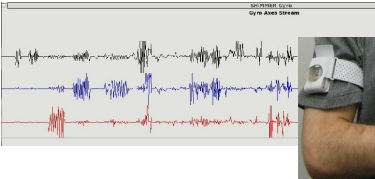
ZebraNet



# Domaines d'application : 3 - e-Santé

## 3- e-Santé

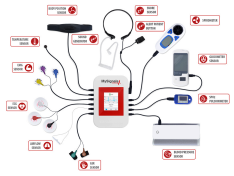
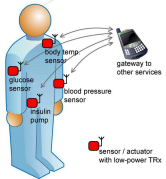
- Surveillance des données physiologiques
- Suivi des patients et des médecins
- Détection de chute ou d'inconscience
- *Body area network* et textiles intelligents



Évaluation de maladies neuromotrices

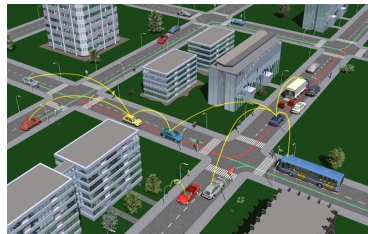
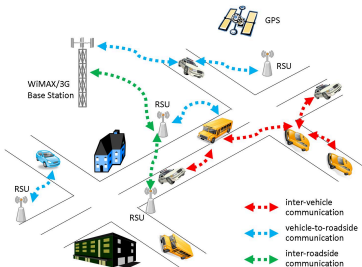
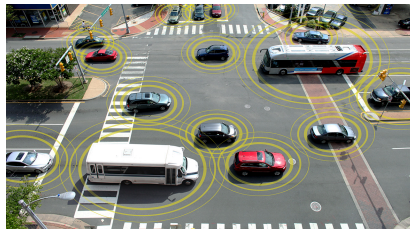
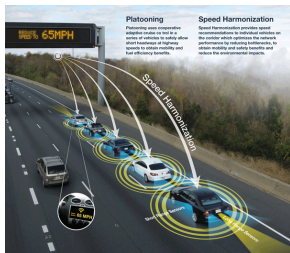


Soins médicaux d'urgence



# Domaines d'application : 4 - Réseaux véhiculaires

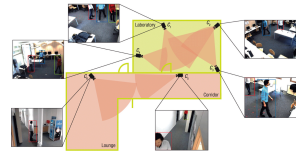
## 4- Réseaux véhiculaires (V2V, V2I, ... V2X)



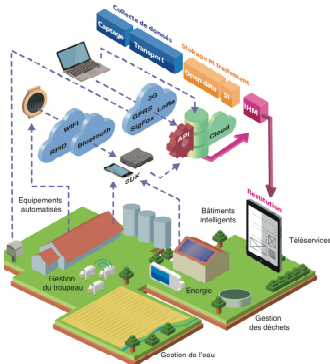
## Domaines d'application : Et plein d'autres ...

### Et plein d'autres applications...

- Applications industrielles, transport et logistique
- Surveillance de structures civiles
- Sécurité et surveillance avec des réseaux de caméras
- Smart grid et systèmes de contrôle d'énergie (Linky)
- Smart cities, dont les fermes intelligentes !!
- ... et l'Internet des objets connectés

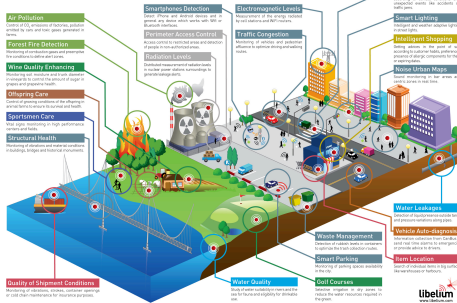


Réseau de caméras intelligentes



L'accès aux données pour la Recherche et l'Innovation en Agriculture (Livre Blanc, ACTA)

### Libelium Smart World





## Enjeux et défis dans les réseaux de capteurs sans fil

Les principaux verrous scientifiques sont

- Limite de l'énergie qu'ils peuvent stocker ou "récolter"
- Topologie dynamique
- Contrôle et routage dans le réseau
- Auto-organisation et localisation
- Hétérogénéité des nœuds
- Déploiement à large échelle
- Traitement collaboratif de l'information
- Capacité à faire face à la défaillance de nœuds
- Capacité à résister à des conditions environnementales sévères
- Miniaturisation et faible coût
- ...

### Conclusion

Les nœuds sont de plus en plus petits et moins chers, tandis que la batterie semble être le goulot d'étranglement.

## Outline

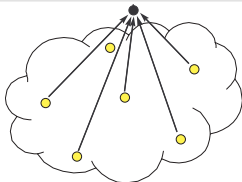
- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus**
- 3 Analyse en composantes principales dans les réseaux de capteurs
- 4 Apprentissage statistique dans les réseaux de capteurs
- 5 Épilogue

## Collaboration : exemple du consensus

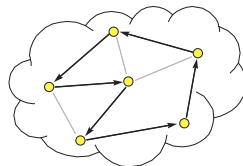
## Collaboration

## Réseaux de capteurs

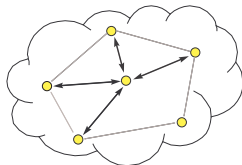
Les réseaux de capteurs sont conçus pour effectuer trois tâches principales : mesurer une grandeur physique, transmettre les informations à des capteurs proches (ou à un récepteur central), et effectuer une estimation centralisée ou distribuée.



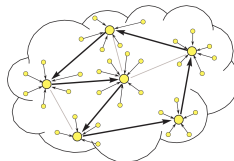
Centralisé



Collaboration selon un chemin



Collaboration par diffusion



Collaboration hybride

# Estimation centralisée dans les réseaux de capteurs

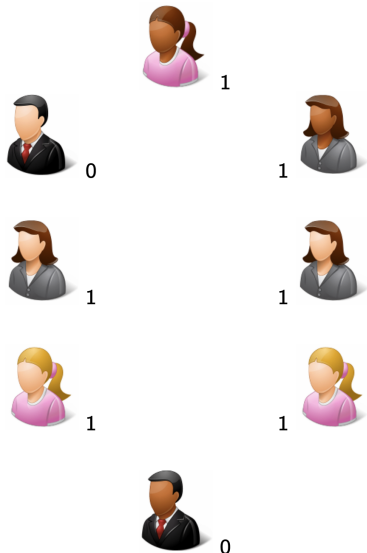
## Estimation de la moyenne

**Exemple d'un problème d'estimation :** Quelle est la proportion de femmes dans la salle ?

**Approche centralisée**

$$\bar{x} = \frac{\#femmes}{\#population} = 0.75$$

$$x_i = \begin{cases} 1 & \text{si femme ;} \\ 0 & \text{sinon} \end{cases}$$

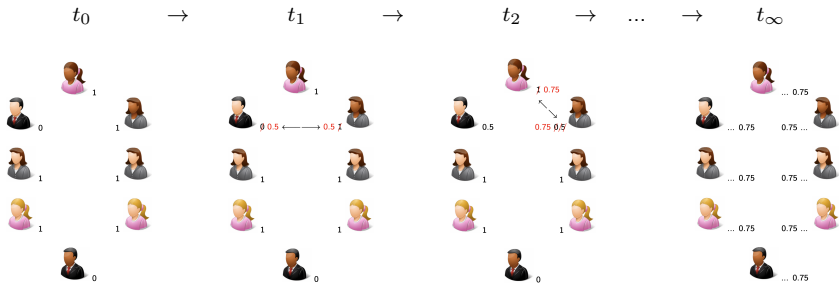


# Estimation décentralisée dans les réseaux de capteurs

## Estimation de la moyenne

**Approche décentralisée**

Interactions locales (par paires ici), jusqu'au consensus (= se mettre d'accord sur  $\bar{x}$ )



## Estimation dans les réseaux de capteurs

### Conclusion

#### Résumé :

- Calcul parallèle et distribué, sans centralisation : traitement en réseau
- Estimation et convergence
- Au delà du décompte : poids, âge, ...
- Au delà de l'estimation de la moyenne : tous les moments, les statistiques, ...

#### Réseaux de capteurs synchronisés ou asynchrones :

- Les algorithmes de consensus nécessitent une synchronisation
- Nous avons souvent des réseaux asynchrones : algorithmes de *broadcast gossip*<sup>1</sup>
  - Les capteurs sont en mode diffusion/écoute : tous les capteurs écoutent, selon une portée ; de temps en temps, un nœud se réveille pour diffuser des informations.

### Penser localement !

Interactions locales simples, au lieu d'un fonctionnement globalement optimal.

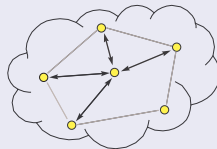
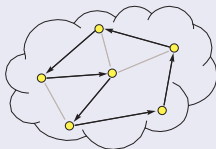
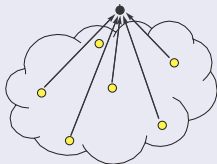
---

<sup>1</sup>*Gossip* = bavardage

## Conclusion

### Modes de traitement de l'information dans les réseaux de capteurs

#### modes de coopération inter-capteurs



- Le mode centralisé n'est pas recommandé, puisqu'il nécessite trop d'énergie<sup>a</sup>, en<sup>b</sup>  $\mathcal{O}(nm^{1+1/d})$ , par opposition à  $\mathcal{O}(n\epsilon^{-2})$  pour les modes de coopération.
- En modes de coopération inter-capteurs, les capteurs profitent de l'information disponible de leurs voisins, afin de mettre-à-jour le modèle.

<sup>a</sup>L'énergie est essentiellement consommée par la communication, et nettement moins pour le traitement (CPU).

<sup>b</sup> $n$ =nombre de capteurs,  $m$ =nombre de mesures par capteur.

### Réseaux de capteurs (sans fil)

- Capteur **intelligent**  $\iff$  capable à **apprendre** à partir de mesures
- Il s'agit d'un problème d'apprentissage statistique, d'estimation spatio-temporelle



## Outline

- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus
- 3 Analyse en composantes principales dans les réseaux de capteurs**
- 4 Apprentissage statistique dans les réseaux de capteurs
- 5 Épilogue

# Analyse en composantes principales dans les réseaux de capteurs

## Analyse en composantes principales (ACP)

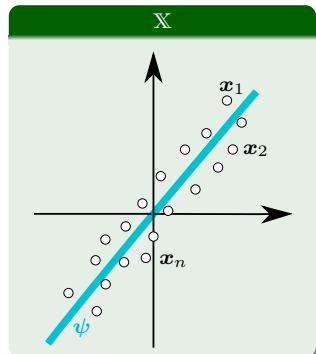
L'ACP détermine les axes principaux qui rendent compte du maximum de variance des données

**Algorithme :**

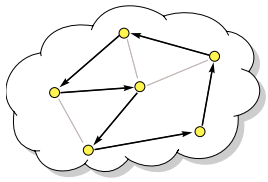
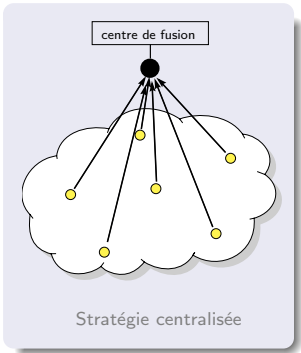
- $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{R}^d$
- Soit la matrice  $\mathbf{C} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j \mathbf{x}_j^\top$
- Chaque axe  $\psi$  est donné par  $\lambda \psi = \mathbf{C} \psi$

**Difficultés :**

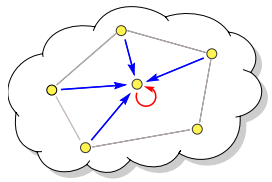
- Estimation de la matrice  $\mathbf{C}$
- Estimation de vecteurs propres  $\psi$



# ACP dans les réseaux

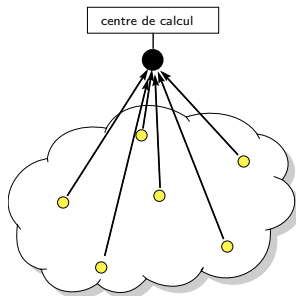


Stratégie en ligne



Stratégie coopérative (diffusion)

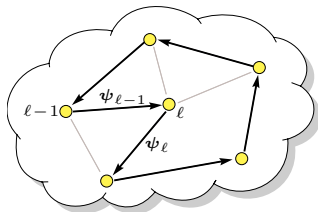
## ACP – Stratégie centralisée –

**Traitement centralisé :**

- 1 Transmettre les échantillons  $\mathbf{x}_1, \dots, \mathbf{x}_n \in \mathbb{R}^d$   
 $\rightsquigarrow$  coût de transmission en  $\mathcal{O}(nd)$  sur une distance en  $\mathcal{O}(1)$
- 2 Estimation de la matrice  $\mathbf{C} = \frac{1}{n} \sum_{j=1}^n \mathbf{x}_j \mathbf{x}_j^\top$ , de taille  $d \times d$
- 3 Estimation de ses vecteurs propres  $\psi^*$   $\rightsquigarrow$  complexité de calcul en  $\mathcal{O}(d^3)$

Les modes de coopération inter-nœuds permettent de réduire la complexité.

## ACP – Stratégie en ligne –



**Stratégie en ligne :** Mise à jour adaptative à chaque nœud  $\ell$ , en utilisant  $\mathbf{x}_\ell$  :

$$\psi_\ell = \psi_{\ell-1} + \eta_\ell (\mathbf{x}_\ell y_\ell - y_\ell^2 \psi_{\ell-1})$$

où  $y_\ell = \psi_{\ell-1}^\top \mathbf{x}_\ell$  et  $\eta_\ell$  est le pas de convergence.

Lien avec l'erreur quadratique instantanée de reconstruction :  $\min_{\psi} \|\mathbf{x}_\ell - y_\ell \psi\|^2$

### Theorem (Cette stratégie converge vers le premier axe principal)

À l'état d'équilibre  $\psi$ , on a  $\mathbf{x}_\ell y_\ell = y_\ell^2 \psi$ , c'est à dire  $\mathbf{x}_\ell \mathbf{x}_\ell^\top \psi = \psi^\top \mathbf{x}_\ell \mathbf{x}_\ell^\top \psi$ . Il suffit simplement de faire la moyenne sur tous les échantillons pour obtenir  $\mathbf{C}\psi = \psi^\top \mathbf{C}\psi$ , avec la valeur propre  $\psi^\top \mathbf{C}\psi$  correspondant à  $y^2$  que l'on souhaite maximiser.

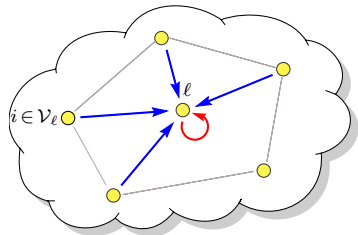
↔ coût de transmission  $\mathcal{O}(n)$  sur une distance en  $\mathcal{O}\left(\frac{1}{\sqrt{n}}\right)$

E. Oja, "Simplified neuron model as a principal component analyzer," *Journal of Mathematical Biology*, 1982.

## ACP – Stratégies de coopération –

## Stratégies de coopération :

- Chaque nœud  $\ell$  a accès à ses voisins  $\mathcal{V}_\ell$ .
- Plusieurs stratégies de diffusion proposées :
  - agrégation-puis-adaptation ;
  - adaptation-puis-agrégation ;
  - consensus / gossip ;
  - ...

Stratégie **agrégation-puis-adaptation** :

$$\psi_{\ell,*} = \sum_{i \in \mathcal{V}_\ell} a_{li} \psi_{i,t-1}$$

$$\psi_{\ell,t} = \psi_{\ell,*} + \eta_\ell (\mathbf{x}_\ell y_\ell - y_\ell^2 \psi_{\ell,*})$$

Stratégie **adaptation-puis-agrégation** :

$$\psi_{\ell,*} = \psi_{\ell,t-1} + \eta_\ell (\mathbf{x}_\ell y_\ell - y_\ell^2 \psi_{\ell,t-1})$$

$$\psi_{\ell,t} = \sum_{i \in \mathcal{V}_\ell} a_{li} \psi_{i,*}$$

Stratégie consensus/gossip :

$$\psi_{\ell,t} = \sum_{i \in \mathcal{V}_\ell} a_{li} \psi_{i,t-1} + \eta_\ell (\mathbf{x}_\ell y_\ell - y_\ell^2 \psi_{\ell,t-1})$$

## Modèles non linéaires

## Modèles non linéaires

- Transformer les observations avec une fonction non linéaire  $\phi(\cdot)$ , vers un espace  $\mathbb{H}$

## ACP non linéaires :

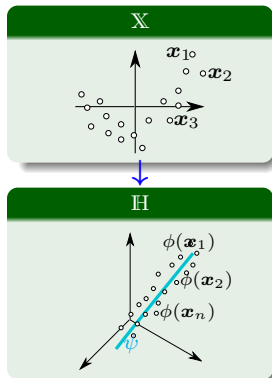
En transformant les données, l'ACP est appliquée dans  $\mathbb{H}$ .  
Les axes sont alors non linéaires dans l'espace  $\mathbb{X}$ .

## Algorithme :

- $\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n \in \mathbb{X}$ ,  
noyau  $\kappa(\mathbf{x}_i, \mathbf{x}_j) = \langle \phi(\mathbf{x}_i), \phi(\mathbf{x}_j) \rangle_{\mathbb{H}}$
- Soient  $[\mathbf{K}]_{i,j} = \kappa(\mathbf{x}_i, \mathbf{x}_j)$  et  $\lambda \boldsymbol{\alpha} = \frac{1}{n} \mathbf{K} \boldsymbol{\alpha}$
- Les axes sont définis par

$$\psi = \sum_{i=1}^n \alpha_i \phi(\mathbf{x}_i)$$

$$\text{avec } \psi(\mathbf{x}) = \langle \phi(\cdot), \phi(\mathbf{x}_i) \rangle_{\mathbb{H}} = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \mathbf{x}).$$



## Solution :

- P. Honeine, "Online kernel principal component analysis: a reduced-order model". IEEE Transactions on Pattern Analysis and Machine Intelligence, 2012
- N. Ghadban, P. Honeine, F. Mourad-Chehade, C. Francis, and J. Farah. "In-network principal component analysis and diffusion strategies". International Journal of Wireless Information Networks, 2016.

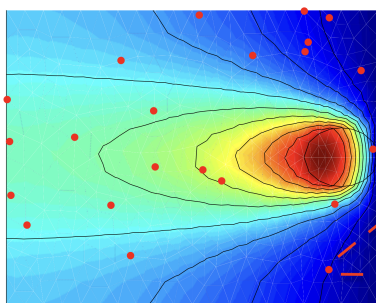


# Outline

- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus
- 3 Analyse en composantes principales dans les réseaux de capteurs
- 4 Apprentissage statistique dans les réseaux de capteurs**
- 5 Épilogue

## Apprentissage statistique dans les réseaux de capteurs

## Apprentissage dans les réseaux : inférer une diffusion



On souhaite estimer une fonction  $\psi(\cdot)$  de la position  $x$  qui modélise le phénomène physique, en se basant sur des données de mesure de position.

La surveillance de phénomènes complexes, par exemple la diffusion, exige que l'on réponde aux questions suivantes :

- la non-linéarité du système
- l'estimation en ligne des paramètres du modèle
- la rareté des données
- stratégie de traitement en réseau

## Apprentissage dans les réseaux

Approches paramétriques / non paramétriques :

- Les approches paramétriques supposent que l'on dispose de suffisamment de **données** ou de **connaissances a priori** pour spécifier un modèle statistique  
→ Estimation des paramètres
- Les méthodes non-paramétriques ont connu un grand succès dans les stratégies centralisées.  
→ **Apprentissage statistique**

Notre objectif est d'étudier des méthodes non paramétriques robustes pour **l'inférence distribuée**.

Le modèle que nous considérons est le suivant :

- $n$  capteurs répartis dans l'espace, situés à  $\{\mathbf{x}_1, \mathbf{x}_2, \dots, \mathbf{x}_n\}$ .
- Chaque capteur  $i$  mesure  $y_i$ , la mesure locale du champ
- La topologie du réseau de capteurs est spécifiée par un graphe.

Objectif : Trouver une estimation globale du champ, c'est-à-dire aux emplacements non occupés par un capteur :

$$y_i = \psi(\mathbf{x}_i) + \varepsilon_i$$

## Apprentissage dans les réseaux

## Problème d'optimisation (centralisé)

$$\psi^*(\cdot) = \operatorname{argmin}_{\psi \in \mathbb{H}} \frac{1}{n} \sum_{i=1}^n \mathcal{C}(\psi(\mathbf{x}_i), y_i) + \eta \mathcal{R}(\psi)$$

- Terme attache aux données :  $\mathcal{C}(\psi(\mathbf{x}_i), y_i) = (\psi(\mathbf{x}_i) - y_i)^2$  ou  $|\psi(\mathbf{x}_i) - y_i|$  ou ...
- Terme de régularisation :  $\mathcal{R}(\psi) = \|\psi\|_{\mathbb{H}}^2$  ou parcimonie ou contrainte physique ...

## Apprentissage dans les réseaux

## Problème d'optimisation (centralisé)

$$\psi^*(\cdot) = \operatorname{argmin}_{\psi \in \mathbb{H}} \frac{1}{n} \sum_{i=1}^n \mathcal{C}(\psi(\mathbf{x}_i), y_i) + \eta \mathcal{R}(\psi)$$

- Terme attache aux données :  $\mathcal{C}(\psi(\mathbf{x}_i), y_i) = (\psi(\mathbf{x}_i) - y_i)^2$  ou  $|\psi(\mathbf{x}_i) - y_i|$  ou ...
- Terme de régularisation :  $\mathcal{R}(\psi) = \|\psi\|_{\mathbb{H}}^2$  ou parcimonie ou contrainte physique ...

## Théorème de Représentation

La solution est de la forme :  $\psi^*(\cdot) = \sum_{i=1}^n \alpha_i \phi_{\mathbf{x}_i}(\cdot)$

Preuves :

- 1971** Kimeldorf, G., Wahba, G. Some results on Tchebycheffian spline functions. *Journal of Mathematical Analysis and Applications* 33, 82–95.
- 2001** Schölkopf, B., Herbrich, R., Smola, A.J. A generalized representer theorem, in: *Proc. 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, Springer-Verlag, London, UK. pp. 416–426.
- 2006** Belkin, M., Niyogi, P., Sindhvani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7, 2399–2434.
- 2009** Argyriou, A., Micchelli, C.A., Pontil, M. When is there a representer theorem? Vector versus matrix regularizers. *The Journal of Machine Learning Research* 10, 2507–2529.
- 2019** Unser, M. A representer theorem for deep neural networks. *Journal of Machine Learning Research* 20, 1–30.
- 2021** Schlegel, K. When is there a representer theorem? *Advances in Computational Mathematics* 47, 54.

## Apprentissage dans les réseaux

## Problème d'optimisation

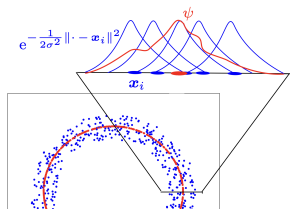
$$\psi^* = \operatorname{argmin}_{\psi \in \mathbf{H}} \frac{1}{n} \sum_{i=1}^n \mathcal{C}(\psi(\mathbf{x}_i), y_i) + \eta \mathcal{R}(\psi)$$

## Théorème de Représentation

La solution est :  $\psi^*(\cdot) = \sum_{i=1}^n \alpha_i \phi_{\mathbf{x}_i}(\cdot)$

Exemple : noyau gaussien  $\phi_{\mathbf{x}_i}(\cdot) = \kappa(\mathbf{x}_i, \cdot)$  :

$$\psi^*(\cdot) = \sum_{i=1}^n \alpha_i \exp^{-\frac{1}{2\sigma^2} \|\cdot - \mathbf{x}_i\|^2}$$



Preuves :

- 1971 Kimeldorf, G., Wahba, G. Some results on Tchebycheffian spline functions. *Journal of Math. Analysis and Applications* 33, 82–95.
- 2001 Schölkopf, B., Herbrich, R., Smola, A.J. A generalized representer theorem, in: *Proc. 14th Annual Conference on Computational Learning Theory and 5th European Conference on Computational Learning Theory*, Springer-Verlag, London, UK, pp. 416–426.
- 2006 Belkin, M., Niyogi, P., Sindhwani, V. Manifold regularization: A geometric framework for learning from labeled and unlabeled examples. *Journal of Machine Learning Research* 7, 2399–2434.
- 2009 Argyriou, A., Micchelli, C.A., Pontil, M. When is there a representer theorem? Vector versus matrix regularizers. *The Journal of Machine Learning Research* 10, 2507–2529.
- 2019 Unser, M. A representer theorem for deep neural networks. *Journal of Machine Learning Research* 20, 1–30.
- 2021 Schlegel, K. When is there a representer theorem? *Advances in Computational Mathematics* 47, 54.

## Le problème de l'apprentissage en ligne

## Modèle optimal

$$\psi(\cdot) = \sum_{i=1}^n \alpha_i \kappa(\mathbf{x}_i, \cdot)$$

Non adapté aux réseaux de capteurs  
car  $n \rightarrow \infty$  !!

Modèle réduit à l'instant  $t$  ( $m \ll n$ )

$$\psi_t(\cdot) = \sum_{j=1}^m \alpha_{t,j} \kappa(\mathbf{x}_{\omega_j}, \cdot)$$

Dictionnaire  $\mathcal{D} = \{\mathbf{x}_{\omega_1}, \mathbf{x}_{\omega_2}, \dots, \mathbf{x}_{\omega_m}\}$   
avec  $m \ll n$

**Apprentissage en ligne :**

A chaque instant  $t$ , le processus est constitué de deux étapes :

- Mise à jour du dictionnaire, si nécessaire
- Mise à jour des paramètres  $\alpha_{t,j}$



## Algorithmes adaptatifs non linéaires

Soit  $\boldsymbol{\alpha}_t = [\alpha_{t,1} \ \alpha_{t,2} \ \dots \ \alpha_{t,m}]^\top$ , à estimer à partir de  $\boldsymbol{\alpha}_{t-1}$ .

$$\min_{\boldsymbol{\psi} \in \mathcal{H}} \sum_{i=1}^t \theta^{t-i} (y_i - \psi(\mathbf{x}_i))^2 + \eta \theta^t \|\boldsymbol{\psi}\|_{\mathcal{H}}^2 \Rightarrow \boldsymbol{\alpha}_t = (\mathbf{H}_t^\top \boldsymbol{\Theta}_t \mathbf{H}_t + \eta \theta^t \mathbf{K}_t)^{-1} \mathbf{H}_t^\top \boldsymbol{\Theta}_t \mathbf{y}_t$$

## Algorithme des moindres carrés récursifs – à la RLS –

$$\begin{aligned} \mathbf{h}_t^\top &= [\kappa(\mathbf{x}_t, \mathbf{x}_{\omega_1}) \ \kappa(\mathbf{x}_t, \mathbf{x}_{\omega_2}) \ \dots \ \kappa(\mathbf{x}_t, \mathbf{x}_{\omega_m})] \\ \mathbf{P}_t &= \theta^{-1} \left( \mathbf{P}_{t-1} - \frac{\mathbf{P}_{t-1} \mathbf{h}_t \mathbf{h}_t^\top \mathbf{P}_{t-1}}{\theta + \mathbf{h}_t^\top \mathbf{P}_{t-1} \mathbf{h}_t} \right), \text{ avec } \mathbf{P}_1 = \eta \mathbf{K}_1 \\ \boldsymbol{\alpha}_t &= \boldsymbol{\alpha}_{t-1} + \mathbf{P}_t \mathbf{h}_t (y_t - \mathbf{h}_t^\top \boldsymbol{\alpha}_{t-1}) \end{aligned}$$

## Algorithme de projection affine – à la APA –

- Principe de fluctuation minimale
- Contrainte sur l'erreur *a posteriori* nulle des  $q$  dernières données

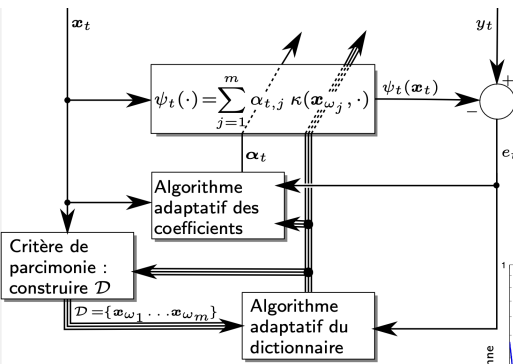
$$\begin{aligned} \min_{\boldsymbol{\alpha}_t} \quad & \|\boldsymbol{\alpha}_t - \boldsymbol{\alpha}_{t-1}\|^2 \\ \text{sous contrainte} \quad & \mathbf{y}_t = \mathbf{H}_t \boldsymbol{\alpha}_t \end{aligned}$$

$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + \rho \mathbf{H}_t^\top (\mathbf{H}_t \mathbf{H}_t^\top + \eta \mathbf{I})^{-1} (\mathbf{d}_t - \mathbf{H}_t \boldsymbol{\alpha}_{t-1})$$

## Algorithme de gradient stochastique – à la LMS –

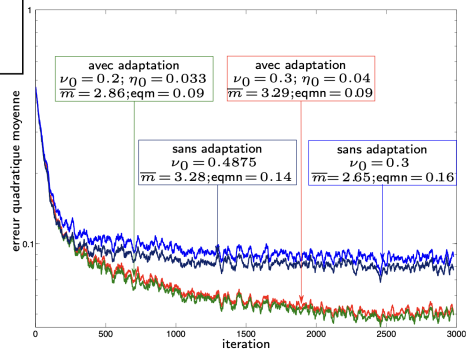
$$\boldsymbol{\alpha}_t = \boldsymbol{\alpha}_{t-1} + \frac{\rho}{\|\mathbf{h}_t\|^2 + \eta} (\mathbf{d}_t - \mathbf{h}_t^\top \boldsymbol{\alpha}_{t-1}) \mathbf{h}_t$$

# Algorithmes adaptatifs non linéaires avec adaptation du dictionnaire

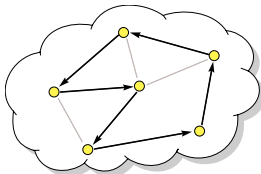


Exemple :

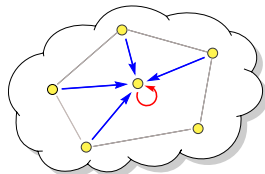
$$\begin{cases} y_t = v_t^2 \\ v_t = 1.1 \exp(-|v_{t-1}|) + x_t \\ x_t \sim \mathcal{N}(0, \frac{1}{4}) \\ v_0 = 0.5 \end{cases}$$



## Conclusion : apprentissage collaboratif dans les réseaux



Stratégie non coopérative

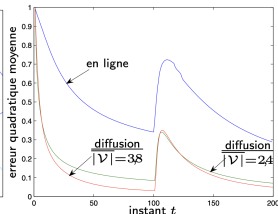
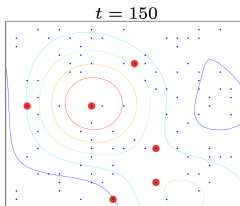
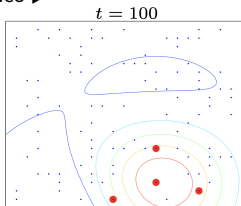


Stratégie coopérative (diffusion)

### Récentes contributions en collaboration

- Algorithmes décentralisés de type filtrage adaptatif non linéaire
- Algorithmes de coopération avec stratégies de diffusion et de consensus
- Adaptation des nœuds pour le suivi de l'évolution du champ estimé

Video ►



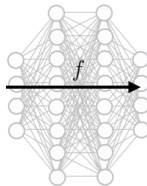
## Outline

- 1 Les réseaux de capteurs sans fil
- 2 Collaboration : exemple du consensus
- 3 Analyse en composantes principales dans les réseaux de capteurs
- 4 Apprentissage statistique dans les réseaux de capteurs
- 5 **Épilogue**

## Conclusion

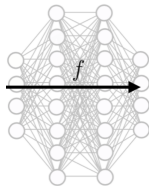
## Qu'en est-il du *On-device Deep Learning*, *TinyML*, ... ?

### Apprentissage par réseaux de neurones profonds



## Qu'en est-il du *On-device Deep Learning*, *TinyML*, ... ?

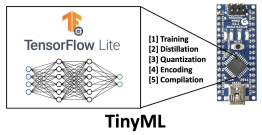
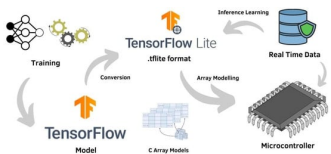
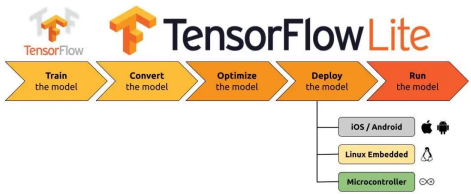
### Apprentissage par réseaux de neurones profonds



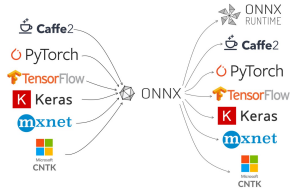
### Des réseaux profonds dans de petits dispositifs (i.e., *edge computing*, *on-device*, ...)

- 2016 F. Iandola, et al. "**SqueezeNet**: AlexNet-level accuracy with 50x fewer parameters and < 0.5 MB model size." arXiv:1602.07360. 2016.
- 2017 A. Howard, et al. "**MobileNets**: Efficient convolutional neural networks for mobile vision applications." arXiv:1704.04861. 2017.
- 2018 X. Zhang, X. Zhou, M. Lin and J. Sun, "**Shufflenet**: An extremely efficient convolutional neural network for mobile devices." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- 2018 I. Freeman, L. Roesse-Koerner and A. Kummert, "**EffNet**: An Efficient Structure for Convolutional Neural Networks," 25th IEEE International Conference on Image Processing (ICIP), 2018.
- 2018 M. Sandler, A. Howard, M. Zhu, A. Zhmoginov and L. C. Chen. "**MobileNetV2**: Inverted residuals and linear bottlenecks." Proceedings of the IEEE conference on computer vision and pattern recognition. 2018.
- 2019 M. Tan, and L. Quoc. "**EfficientNet**: Rethinking model scaling for convolutional neural networks." International conference on machine learning. PMLR, 2019.
- 2019 M. Tan, et al. "**MnasNet**: Platform-aware neural architecture search for mobile." Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
- 2019 A. Howard, et al. "Searching for **MobileNetV3**." Proceedings of the IEEE/CVF international conference on computer vision. 2019.
- 2020 X. Chu, Z. Bo, and X. Ruijun "Searching beyond **MobileNetV3**." IEEE International Conference on Acoustics, Speech and Signal Processing, 2020.
- 2020 K. Han, Y. Wang, Q. Tian, J. Guo, C. Xu and C. Xu. "**GhostNet**: More features from cheap operations." Proceedings of the IEEE/CVF conference on computer vision and pattern recognition. 2020.
- 2021 B. Koonce, "**MobileNetV3**." Convolutional Neural Networks with Swift for Tensorflow. Apress, Berkeley, 2021.
- 2022 Y. Chen, et al. "**Mobile-former**: Bridging MobileNet and transformer." Proc. of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2022.

# Qu'en est-il du On-device Deep Learning, TinyML, ... ?



Avec possibilité d'interopérabilité par ONNX<sup>a</sup> :



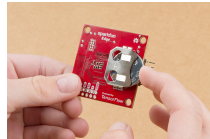
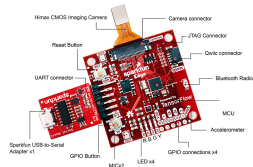
<sup>a</sup> ONNX : Open Neural Network eXchange



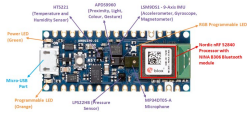
# Qu'en est-il du *On-device Deep Learning, TinyML, ... ?*

## TensorFlow Lite for Microcontrollers

- SparkFun Edge board (Powered by TensorFlow)



- Arduino Nano (33 BLE Sense - Bluetooth Low Energy)



- Sony Spresence, ESP-EYE, OpenMV, ...



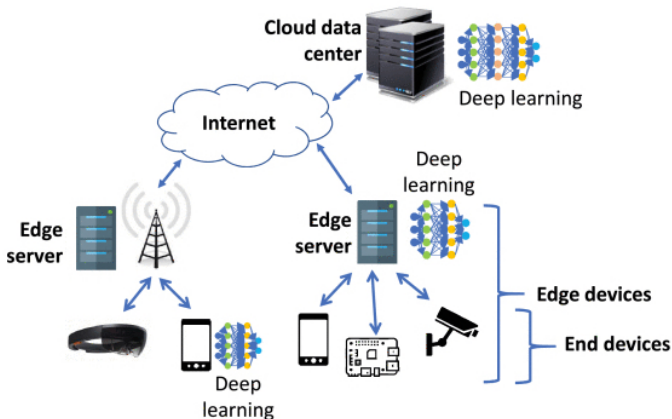
Powered by TensorFlow



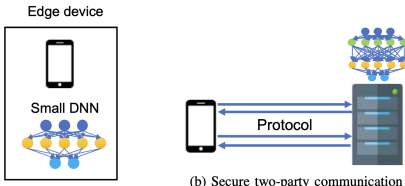
## Qu'en est-il du Deep Learning ?

### Apprentissage par réseaux de neurones profonds

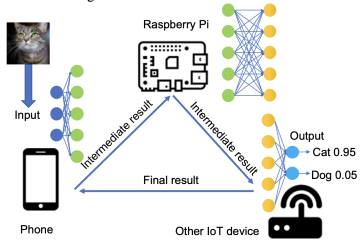
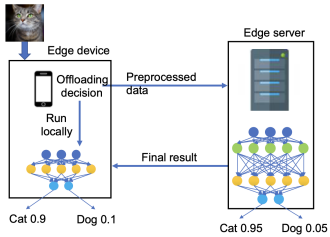
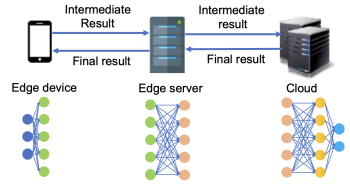
- Le paradigme “centralisé/distribué” est dit “cloud/edge”
- Tout ce qui est vieux est à nouveau nouveau !



# Un problème ouvert : Cloud vs. Edge computing



(a) On-device computation



J. Chen and X. Ran, "Deep Learning With Edge Computing: A Review," in Proceedings of the IEEE, vol. 107, no. 8, pp. 1655-1674, 2019

**Merci pour votre attention**