

# Des réseaux de neurones sous toutes leurs formes

Pr. Hélène Paugam-Moisy

Université des Antilles - DMI - LAMIA

Séminaire LAMIA - 13 octobre 2016



# Qui je suis et d'où je viens...

## Postes occupés précédemment

1978 - 1989	professeur de Mathématiques Lycées et collèges	Île-de-France et Lyonnais
1989 - 1992	Agrégé Préparateur, Informatique ENS Lyon	LIP (URA CNRS 1398)
1992 - 1998	Maître de Conférences, CNU 27 ENS Lyon	LIP (URA CNRS 1398)
1998 - 2016	Professeur d'Informatique, CNU 27 Université Lyon 2	[1998-2000] labo. ERIC [2000-2006] ISC (UMR CNRS 5015) [2007-2016] LIRIS (UMR CNRS 5205)

## Détachements divers

1996 - 1998	délégation CR CNRS	LIP (ENS Lyon)
2004 - 2006	délégation DR CNRS	ISC + séjour 3 mois IDIAP (Suisse)
2008 - 2012	délégation DR INRIA	LRI (Université Paris-Sud, Orsay)

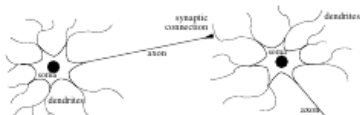
## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
    - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*

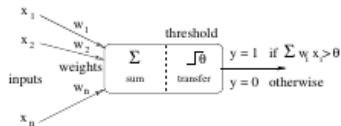


# Les modèles de neurones artificiels

Le modèle historique : un **neurone à seuil** [McCulloch et Pitts, 1943]



Elementary scheme of biological neurons



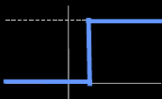
First mathematical model of artificial neuron

Les entrées sont binaires et la sortie est binaire :  $y = f(\sum_{i=1}^n w_i x_i - \theta)$   
où  $f$  est la fonction signe ou bien la fonction de Heaviside.

Premières applications : synthèse des **fonctions booléennes**.

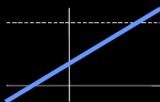
Limitation aux fonctions linéairement séparables  $\Rightarrow$  XOR impossible.

# Différents types de neurones



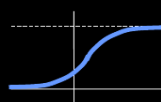
neurone à seuil  
(signe ; Heaviside)

$$f = H$$



neurone linéaire  
(fx identité)

$$f = I$$



neurone sigmoïde  
(exp ; tanh)

$$f = \sigma$$

$$y = f \left( \sum_{i=1}^n w_i x_i - \theta \right)$$

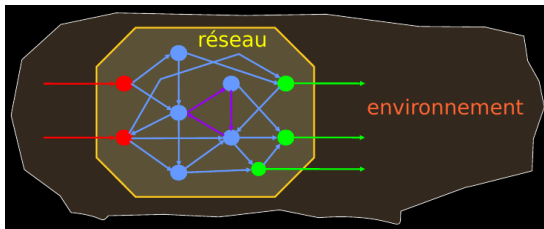
calculs basés sur  
le produit scalaire

Dans certains modèles (cartes de Kohonen, réseaux RBF) :  
calculs basés sur la distance  $y = g (|| X - W ||)$



## Réseau de neurones artificiels : architecture et dynamique

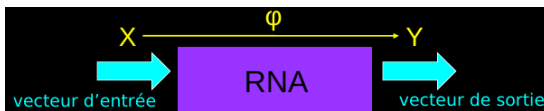
Un réseau de neurones artificiels est un ensemble de neurones formels, connectés entre eux selon une certaine **architecture** / **topologie**.



Les activités se propagent de neurone en neurone selon une certaine **dynamique** : feedforward, récurrente, etc. . . selon qu'il y a ou non des circuits dans le graphe des connexions.

## Réseau de neurones artificiels (RNA) : apprentissage

On peut aussi voir un réseau de neurones comme une “**boîte noire**” qui permet de **modéliser** une fonction (inconnue !) entre un espace d'entrée et un espace de sortie.



Modifier la fonction correspond à modifier les poids des connexions du réseau : cela se fait par **apprentissage** à partir d'exemples.

L'apprentissage est dit **supervisé** si l'on utilise les sorties désirées (i.e. pour chaque  $\vec{x}$  on connaît  $\vec{y}$ ), non supervisé sinon (on parle alors de *clustering*).

## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - **Petit historique**
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*





## Petite histoire du connexionnisme



## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - **Parallélisme**
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Parallélisation des RNA

- L'espion parallèle : optimisation des hyperparamètres [ML&NN, 1990] [ICANN'91] [J.Wiley, 1993]
- Optimisation des temps d'apprentissage, adaptée à plusieurs modèles connexionnistes : MLP, RBF, HME (algorithme EM) [IJCNN'92] [CONPAR'92] [IFIP'94] [IPPS'95] [EuroPar'96] [PDPTA'98] [Parallel Computing, 2002]

Réseaux de transputers, machines MIMD, usage de PVM et de MPI.  
Modélisation des temps de calcul et de communication, mesure de speedup



# Parallélisation des RNA

- L'espion parallèle : optimisation des hyperparamètres [ML&NN, 1990] [ICANN'91] [J.Wiley, 1993]
- Optimisation des temps d'apprentissage, adaptée à plusieurs modèles connexionnistes : MLP, RBF, HME (algorithme EM) [IJCNN'92] [CONPAR'92] [IFIP'94] [IPPS'95] [EuroPar'96] [PDPTA'98] [Parallel Computing, 2002]

Réseaux de transputers, machines MIMD, usage de PVM et de MPI.  
Modélisation des temps de calcul et de communication, mesure de speedup

## Apports :

- “Penser” l'apprentissage des RNA en parallèle
- Granularité : efficacité maximum au niveau “module” [MIT Press, 1995]

## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - **Approches théoriques**
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Puissance et capacité des RNA multicouches

- Capacité : bornes sur les nombres de neurones [LATIN'92] [IJCNN'93] [J.Complexity, 1994]
- Puissance de calcul : une ou deux couches cachées [NIPS'96] [AI&Math.'98] [AMAI, 1998] [CAp'99]

**Études théoriques** : calculs combinatoires, analytiques, géométriques, espaces booléens (hypercubes), programmation linéaire, graphes (treillis)



# Puissance et capacité des RNA multicouches

- Capacité : bornes sur les nombres de neurones [LATIN'92] [IJCNN'93] [J.Complexity, 1994]
- Puissance de calcul : une ou deux couches cachées [NIPS'96] [AI&Math.'98] [AMAI, 1998] [CAp'99]

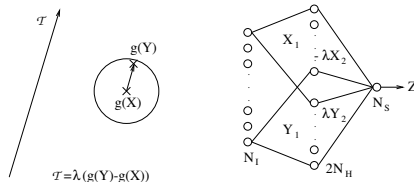
**Études théoriques** : calculs combinatoires, analytiques, géométriques, espaces booléens (hypercubes), programmation linéaire, graphes (treillis)

## Apports :

À contre-courant d'une idée reçue des années 1990 : deux couches valent mieux qu'une  $\Rightarrow$  l'idée sera reprise au niveau international à partir de 2006  
les **deep networks** (à suivre ...)

# Complexité de l'apprentissage

Jacobian Neural Network (JNN) : apprentissage exact par un algorithme en temps polynomial [NIPS'96] [ICANN'98] [Neurocomputing, 1999]



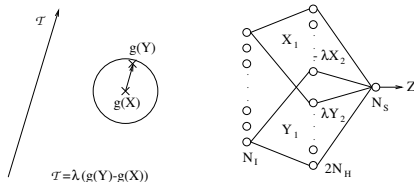
Approche analytique basée sur le **théorème d'inversion locale**.





# Complexité de l'apprentissage

Jacobian Neural Network (JNN) : apprentissage exact par un algorithme en temps polynomial [NIPS'96] [ICANN'98] [Neurocomputing, 1999]



Approche analytique basée sur le **théorème d'inversion locale**.

**Algorithme randomisé** pour l'apprentissage exact, régularisation.

**Apport :**

Résultat de complexité, à contre-courant d'une idée reçue (NP-complet).

## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - **Apprentissage statistique et SVM**
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Théorie de l'apprentissage de Vapnik

Vapnik a revisité la notion d'**induction** en démontrant que le risque (*d'erreur*) dépendait à la fois du **risque empirique**, que l'on peut mesurer sur les exemples appris, et d'un **risque structurel** dépendant de la famille de modèles choisie :

$$R(\alpha) \leq R_{emp}(\alpha) + \Phi(VC_{dim}(\text{modèle}))$$



# Théorie de l'apprentissage de Vapnik

Vapnik a revisité la notion d'**induction** en démontrant que le risque (*d'erreur*) dépendait à la fois du **risque empirique**, que l'on peut mesurer sur les exemples appris, et d'un **risque structurel** dépendant de la famille de modèles choisie :

$$R(\alpha) \leq R_{emp}(\alpha) + \Phi(VC_{dim}(\text{modèle}))$$

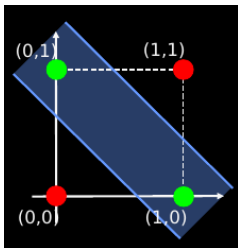
En repartant du Perceptron, Vapnik propose alors les **Support Vector Machines** (SVM), un nouveau modèle connexionniste plus propice à contrôler ce risque. Les SVM sont basées sur deux “astuces” :

- le plongement du problème dans un espace de plus grande dimension, par le biais d'une **fonction noyau** ;
- une séparation de **marge optimale**, dans cet espace virtuel.

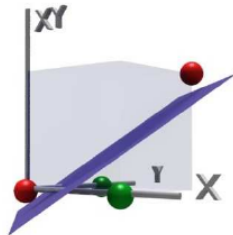
# Les SVM en images (1)

## Plongement dans un espace de plus grande dimension

Fonction de changement d'espace :  $\phi(x, y) = (x, y, xy)$



En dimension 2, le XOR n'est pas linéairement séparable.

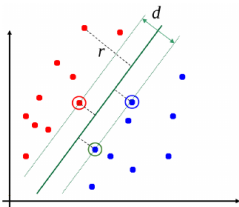


Plongé en dimension 3, le XOR **devient** linéairement séparable.

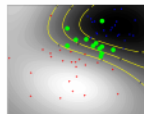
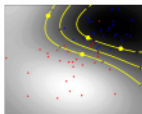
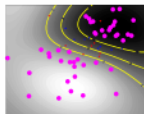
La **fonction noyau** est :  $K(\vec{x}_i, \vec{x}_j) = (1 + \vec{x}_i \cdot \vec{x}_j)^2 = \phi(\vec{x}_i) \cdot \phi(\vec{x}_j)$

# Les SVM en images (2)

## Maximisation de la marge



Parmi tous les hyperplans séparateurs, un seul **maximise la marge** : il s'appuie sur les “**support vectors**”.



données inutiles  
bien classées

$$I_0, \alpha_i = 0 \\ y_i(f(\mathbf{x}_i) + b) > 1$$

données importantes  
support

$$I_\alpha, 0 < \alpha_i < C \\ y_i(f(\mathbf{x}_i) + b) = 1$$

données suspectes

$$I_C, \alpha_i = C \\ y_i(f(\mathbf{x}_i) + b) < 1$$

# Support Vector Machines (SVM)

- Synthèse, en français, du livre de Vapnik (1995) "The nature of statistical learning theory" [[chapitre](#), [Hermès 1999](#)]
- Discrimination multiclasse, SVM multiclassés [[ICANN'98](#)] [[ICANN'99](#)] [[IJCNN'2000](#)] [[Neurocomputing, 2004](#)]

Apprentissage statistique, théorie des bornes, convergence uniforme



# Support Vector Machines (SVM)

- Synthèse, en français, du livre de Vapnik (1995) “The nature of statistical learning theory” [[chapitre](#), [Hermès 1999](#)]
- Discrimination multiclasse, SVM multiclassés [[ICANN'98](#)] [[ICANN'99](#)] [[IJCNN'2000](#)] [[Neurocomputing, 2004](#)]

Apprentissage statistique, théorie des bornes, convergence uniforme

## Apports :

Aide pédagogique pour les cours sur l'apprentissage statistique et les SVM.

Approche globale des problèmes de classification multiclassés.





## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - **Modélisations Cognitives**
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Un exemple de modélisation cognitive

## Une mémoire associative multimodale

### *Aspects cognitifs*

- Validation d'une architecture fonctionnelle (psycho.) [ICCNS'2000]  
[IC-AI'2000]
- Évocation d'images mentales, causalité, effet McGurk [IJCNN'2001]  
[ESANN'2002]



# Un exemple de modélisation cognitive

## Une mémoire associative multimodale

### *Aspects cognitifs*

- Validation d'une architecture fonctionnelle (psycho.) [ICCNS'2000] [IC-AI'2000]
- Évocation d'images mentales, causalité, effet McGurk [IJCNN'2001] [ESANN'2002]

### *Aspects informatiques*

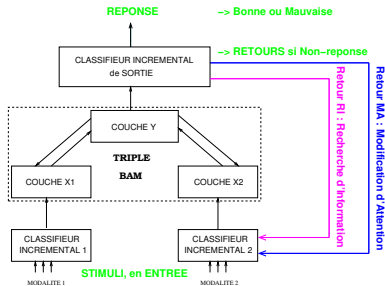
- Modèle connexionniste modulaire, apprentissage local [CAp'2000] [IJCNN'2005]
- Fusion de données, traitement des données incomplètes [IC-AI'2000] [CAp'2000]





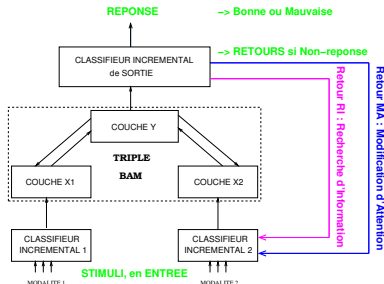
# Informatique : modèle connexionniste modulaire

## Modèle connexionniste modulaire



# Informatique : modèle connexionniste modulaire

## Modèle connexionniste modulaire

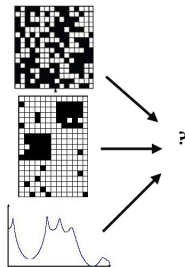


- N modules d'entrée, 1 par stimulus perceptif : ex. son, image, odeur
- 1 mémoire associative multimodale : intègre les informations sensorielles ← *modèle BAM modifié* ⇒ *BAM multiple*
- 1 module de sortie : identifie l'objet perçu ← *classifieur à prototypes*

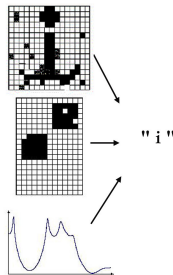
# Évocation d'images mentales

## Un des phénomènes reproduits par le modèle

Le processus d'identification commence avec deux prototypes, visuel et tactile, codant pour un "i", mais un prototype manquant en vision.



**Avant** réverbération

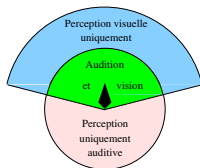


**Après** réverbération

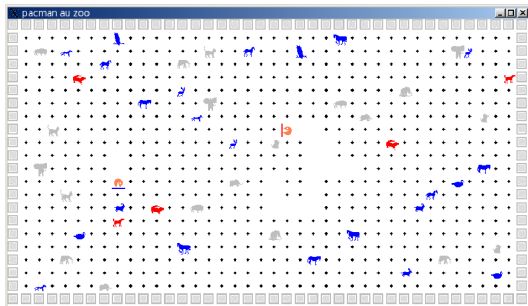
Le prototype visuel de "i" est reconstruit par la dynamique de réverbération de la BAM multiple et l'objet est correctement identifié en sortie du modèle.

# Environnement robotique virtuel

Le modèle anime un **robot virtuel**, doué de **vision** et d'**audition**, qui se déplace dans un zoo virtuel peuplé de proies et de prédateurs.



[IJCNN'2001]



proies (en **bleu**), prédateurs (en **rouge**) et animaux neutres (en gris).



# Ajout d'une dimension temporelle (1)

*Solution 1* ← **parallélisme**

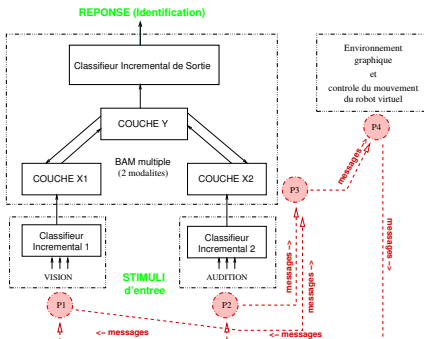
- Traitement parallèle des modules cognitifs et communications non-bloquantes → simulation d'un amorçage intermodal [PDCS'2003]
- Simulation de mécanismes attentionnels [PDCN'2005]



# Ajout d'une dimension temporelle (1)

*Solution 1* ← **parallélisme**

- Traitement parallèle des modules cognitifs et communications non-bloquantes → simulation d'un amorçage intermodal [PDCS'2003]
- Simulation de mécanismes attentionnels [PDCN'2005]



Modules répartis sur 4 processeurs,  
communications bloquantes ou non.

## Ajout d'une dimension temporelle (2)

*Solution 2* ← **spiking neurons**

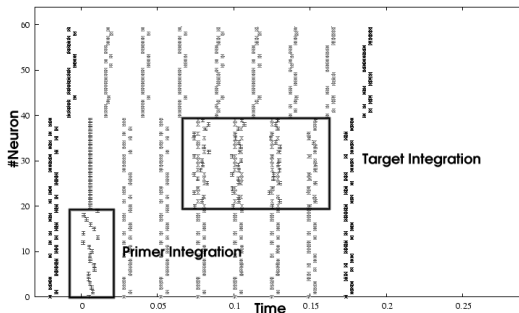
- Réseau de *spiking neurons* pour rendre dynamique le module associatif  
→ simulation d'un amorçage intermodal [NCI'2004]
- Simulation de mécanismes attentionnels [PDCN'2005]



# Ajout d'une dimension temporelle (2)

Solution 2  $\leftarrow$  **spiking neurons**

- Réseau de *spiking neurons* pour rendre dynamique le module associatif  
→ simulation d'un amorçage intermodal [NCI'2004]
- Simulation de mécanismes attentionnels [PDCN'2005]



La cible, qui n'est pas  
congruente à l'amorce,  
est intégrée progressivement.

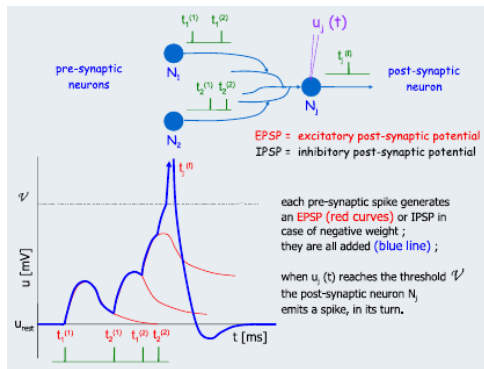
## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Notion de neurone temporel : *spiking neuron*

Les biologistes ont observé que le **temps**, tout autant que la **fréquence** d'émission des *spikes*, jouait un rôle essentiel dans les processus cognitifs.



L'instant précis d'émission d'un *spike* joue un rôle fondamental dans le codage de l'information.

potentiel d'action

= pulse

= spike

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]

$$C \frac{du}{dt} = -g_{Na} m^3 h (u - E_{Na}) - g_K n^4 (u - E_K) - g_L (u - E_L) + I(t)$$

$$\tau_n \frac{dn}{dt} = -[n - n_0(u)]$$

$$\tau_m \frac{dm}{dt} = -[m - m_0(u)]$$

$$\tau_h \frac{dh}{dt} = -[h - h_0(u)]$$

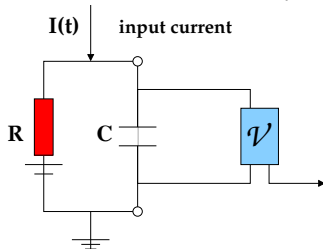


# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]



$$C \frac{du}{dt} = -\frac{1}{R}u(t) + I(t)$$

spike timing  $t^{(f)}$  defined by  $u(t^{(f)}) = \vartheta$  with  $u'(t^{(f)}) > 0$

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]
- Gerstner's **Spike Response Model** (SRM, SRM<sub>0</sub>) [ $\approx$ 1998]

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]
- Gerstner's **Spike Response Model** (SRM, SRM<sub>0</sub>) [≈1998]

$$u_j(t) = \sum_{t_j^{(f)} \in \mathcal{F}_j} \eta_j(t - t_j^{(f)}) + \sum_{i \in \Gamma_j} \sum_{t_i^{(f)} \in \mathcal{F}_i} w_{ij} \epsilon_{ij}(t - t_i^{(f)}) + \underbrace{\int_0^\infty \kappa_j(r) I(t - r) dr}_{\text{if external input current}}$$

or simpler model SRM<sub>0</sub> (very close to traditional formula)

$$u_j(t) = \sum_{i \in \Gamma_j} w_{ij} \epsilon(t - t_i^{(f)} - \delta_{ij}^{ax})$$

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]
- Gerstner's **Spike Response Model** (SRM, SRM<sub>0</sub>) [ $\approx$ 1998]
- **Izhikevich**'s neuron model [2003]

# Les modèles de neurones temporels

- Model of **Hodgkin-Huxley** (HH model) [since 1952]
- **Integrate-and-Fire** model (IF, LIF, QIF...) [since Lapicque, 1907!]
- Gerstner's **Spike Response Model** (SRM, SRM<sub>0</sub>) [ $\approx$ 1998]
- **Izhikevich**'s neuron model [2003]

$$\frac{du}{dt} = 0.04u(t)^2 + 5u(t) + 140 - w(t) + I(t)$$

$$\frac{dw}{dt} = a(bu(t) - w(t))$$

# Simulateur pour de grands réseaux dynamiques

*Parallélisme + Spikes  $\Rightarrow$  modélisation des aspects dynamiques*

DAMNED, a Distributed and Multithreaded Neural Event-Driven simulation framework

**thèse de l'UAG** : Anthony Mouraud, avec Didier Puzenat et HPM - 2009

Un simulateur parallèle et “multithreadé”, dédié à la programmation événementielle de **grands réseaux d'interactions dynamiques**, en particulier des *Spiking Neuron Networks* (SNN).



# Simulateur pour de grands réseaux dynamiques

*Parallélisme + Spikes  $\Rightarrow$  modélisation des aspects dynamiques*

DAMNED, a Distributed and Multithreaded Neural Event-Driven simulation framework

**thèse de l'UAG** : Anthony Mouraud, avec Didier Puzenat et HPM - 2009

Un simulateur parallèle et “multithreadé”, dédié à la programmation événementielle de **grands réseaux d'interactions dynamiques**, en particulier des *Spiking Neuron Networks* (SNN).

## *Développement et utilisation de DAMNED*

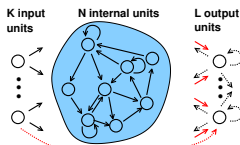
- L'architecture parallèle du simulateur [PDCN'2006]
- SNN modulaire pour étudier le mécanisme des saccades oculaires [NeuroComp'2006] [ICANN'2010]



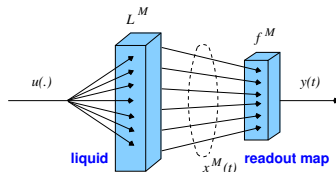
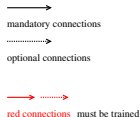
# Un nouveau paradigme : *Reservoir Computing*

Deux architectures similaires de réseaux de neurones (SNN) ont été proposées :

- **Echo State Network** - H. Jaeger (2001) Report TR-GMD-148
- **Liquid State Machine** - W. Maass, T. Natschläger, H. Markram (2002)  
Neural Computation



**Echo State Network**



**Liquid State Machine**

La dynamique, influencée par les entrées, évolue librement dans le **réservoir** ; on apprend à en extraire de l'information : **readout**.

## Avantages et inconvénients

Avantages : possibilité de traiter des données temporelles

Performances améliorées dans de très nombreux domaines applicatifs :

**Speech** *Isolated word recognition using a Liquid State Machine*

**Speech** *Automatic speech recognition with neural spike trains*

**Vision** *Rapid visual processing using spike asynchrony*

**Vision** *Saliency extraction with a distributed SNN [ESANN'06]*

**Control** *Echo State Networks used for motor control*

**Robotics** *From wheels to wings with evolutionary spiking neurons*



## Avantages et inconvénients

Avantages : possibilité de traiter des données temporelles

Performances améliorées dans de très nombreux domaines applicatifs :

**Speech** *Isolated word recognition using a Liquid State Machine*

**Speech** *Automatic speech recognition with neural spike trains*

**Vision** *Rapid visual processing using spike asynchrony*

**Vision** *Saliency extraction with a distributed SNN [ESANN'06]*

**Control** *Echo State Networks used for motor control*

**Robotics** *From wheels to wings with evolutionary spiking neurons*

Inconvénients : la notion d'apprentissage doit être redéfinie

On sait émuler les modèles de RNA classiques avec des SNN mais, si l'on veut **exploiter la dimension temporelle**, il faut repenser l'**apprentissage**.

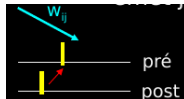
- Apprentissage local et non supervisé dans le réservoir (ex. STDP)
- Une autre forme d'apprentissage est nécessaire si l'on veut **superviser**

# Apprentissage par STDP : plasticité synaptique

Depuis les expériences biologiques de Bi and Poo (2001), la **STDP** est appliquée comme méthode d'apprentissage locale et non supervisée.



si  $t_{post} > t_{pre}$  alors  $\Delta W > 0$   
**causalité**



si  $t_{post} < t_{pre}$  alors  $\Delta W < 0$   
**(inutilité)**

Formule d'application multiplicative (pour éviter la saturation des poids) :

$$\begin{aligned} &\text{if } \Delta T \geq 0 \text{ potentiate the weight:} \\ &w_{ij} \leftarrow w_{ij} + \alpha \times (w_{max} - w_{ij}) \times \Delta W \\ &\text{if } \Delta T \leq 0 \text{ depreciate the weight:} \\ &w_{ij} \leftarrow w_{ij} + \alpha \times (w_{ij} - w_{min}) \times \Delta W \end{aligned}$$

# Spiking Neuron Networks et “Reservoir Computing”

## *Algorithmes d'apprentissage multi-échelles*

- Plasticité synaptique et **algorithme évolutionnaire** [IJCNN'2005]
- Plasticité synaptique et critère à **marge temporelle** [ESANN'2007]  
[Neurocomputing, 2008]



# Spiking Neuron Networks et “Reservoir Computing”

## *Algorithmes d'apprentissage multi-échelles*

- Plasticité synaptique et **algorithme évolutionnaire** [IJCNN'2005]
- Plasticité synaptique et critère à **marge temporelle** [ESANN'2007]  
[Neurocomputing, 2008]

## *Codage et complexité*

- Algorithmes de détection des groupes polychrones [ICANN'2009]
- Étude des différentes formes de codage neuronal [J. Physio, 2010]
- Une métrique neuronale à l'usage de la classification [ESANN'2012]

## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - **Modélisation d'un système complexe**
  - Réseaux de neurones profonds : *deep learning*



# Modélisation d'un système complexe

Dans le cadre d'un projet européen sur la robotique en essaim, on a étudié l'émergence d'une organisation dans une colonie de fourmis allant chercher de la nourriture : l'**aspect temporel** est essentiel.

À partir d'une compétition entre deux **neurones temporels**, pour la prise de décision de chaque fourmi, on voit **émerger** des groupes qui travaillent en alternance, au niveau de la fourmilière.





# Modélisation d'un système complexe

Dans le cadre d'un projet européen sur la robotique en essaim, on a étudié l'émergence d'une organisation dans une colonie de fourmis allant chercher de la nourriture : l'**aspect temporel** est essentiel.

À partir d'une compétition entre deux **neurones temporels**, pour la prise de décision de chaque fourmi, on voit **émerger** des groupes qui travaillent en alternance, au niveau de la fourmilière.

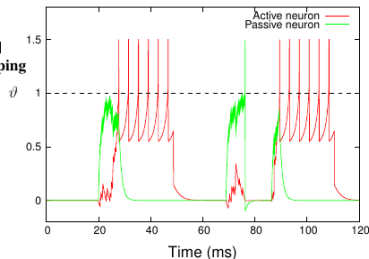
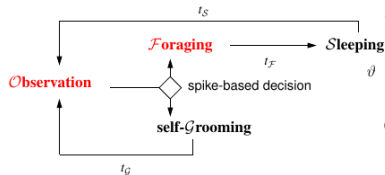
## *Émergence de synchronisations dans un système complexe*

- Émergence d'une organisation dans les colonies de fourmis [NIPS'2010]
- Vers une application à la robotique autonome [ECAL'2011]



## Au niveau microscopique

Une compétition temporelle entre deux neurones contrôle la prise de décision de chaque fourmi.



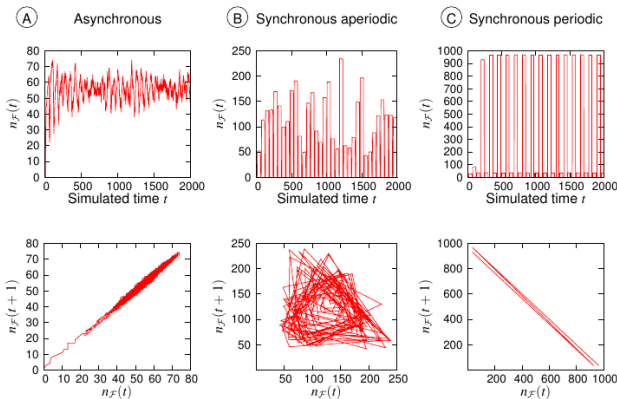
Le système est déterministe.

La durée d'observation dépend des événements perçus.

Les fourmis n'ont pas besoin de savoir compter.

# Au niveau macroscopique

On observe l'émergence d'une synchronisation au niveau de la fourmilière : selon les paramètres du modèle, trois formes d'activité sont observables.

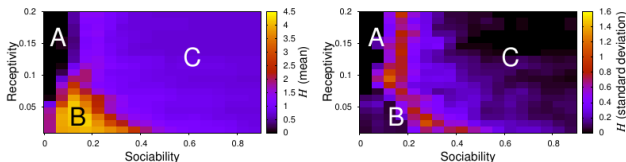


# Diagramme de phase de l'émergence de synchronie

En noir : fourmis individualistes  $\Rightarrow$  absence d'organisation

En bleu : fourmis très sociables  $\Rightarrow$  organisation en deux équipes.

Entropy-based indicator:  $H = - \sum_{k=1}^K \frac{n_k}{\sum_m n_m} \log \left( \frac{n_k}{\sum_m n_m} \right)$



Order parameters govern emergence of shape of activity:

- A Asynchronous,  $H = 0$
- B Synchronous aperiodic,  $H > \log 2$
- C Periodic synchronous,  $H \sim \log 2$

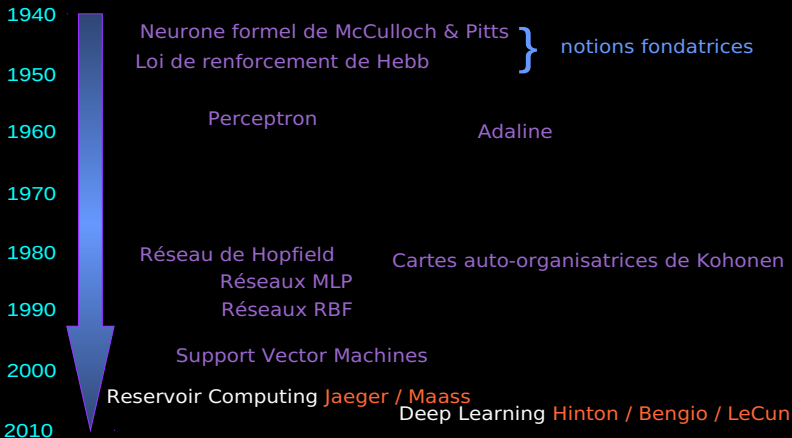
## Plan de l'exposé

- 1 Réseaux de neurones artificiels (RNA)
  - Quelques définitions
  - Petit historique
- 2 Travaux à l'ENS Lyon
  - Parallélisme
  - Approches théoriques
- 3 Travaux à l'Université Lyon 2
  - Apprentissage statistique et SVM
  - Modélisations Cognitives
  - Réseaux de neurones temporels : *spiking neurons*
- 4 Travaux à l'INRIA Saclay
  - Modélisation d'un système complexe
  - Réseaux de neurones profonds : *deep learning*



# Un autre nouveau paradigme : *deep learning*

## Nouveaux paradigmes pour les réseaux de neurones



# Apprentissage dans un réseau profond

On apprend à **reconstruire chaque couche successivement**, avec des RBM ou des auto-associateurs, puis on opère une étape de *fine-tuning*, i.e. une rétro-propagation globale et supervisée sur l'ensemble des couches ainsi empilées.

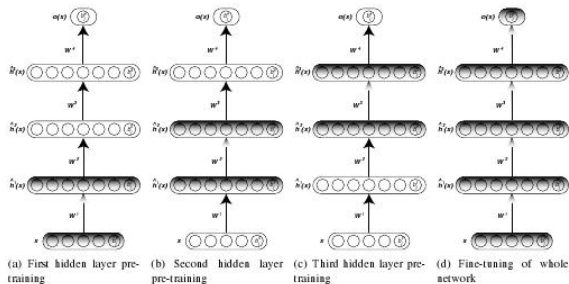


Figure 2: Unsupervised greedy layer-wise training procedure.

*Pour plus de détails, voir le séminaire du LAMIA du 5 janvier 2016.*

# Réseaux de neurones profonds et “Deep Learning”

## ... Les réseaux multicouches : le retour !

- Apprentissage couche par couche, dans les réseaux profonds [RFIA'2010] [ECAI'2010] [tutoriel ESANN'2011]
- “Sparsité” : nouvelle fonction d'apprentissage [Snowbird'2012]
- “Sparsité” : apprentissage de caractéristiques éparses [chapter in “Growing Adaptive Machines”, Springer, 2014]





# Réseaux de neurones profonds et “Deep Learning”

## ... Les réseaux multicouches : le retour !

- Apprentissage couche par couche, dans les réseaux profonds [RFIA'2010] [ECAI'2010] [tutoriel ESANN'2011]
- “Sparsité” : nouvelle fonction d'apprentissage [Snowbird'2012]
- “Sparsité” : apprentissage de caractéristiques éparses [chapter in “Growing Adaptive Machines”, Springer, 2014]

### Apports :

Apprentissage plus performant, complexité réduite de l'architecture.  
Apprentissage de grandes masses de données, de grande dimension.  
Possibilité d'inclure dans l'apprentissage des données non-étiquetées.

# C'est fini !



*Je vous remercie pour votre attention.*

*Je suis prête à répondre à toutes vos questions.*



## Chapitres de livres, tutoriels dans des conférences

- Multiprocessor simulation of neural networks  
[Handbook of Brain Theory and Neural Networks, MIT Press, 1995]
- Théorie de l'apprentissage de Vapnik et SVM  
[Apprentissage automatique, Hermès, 1999]
- Neural networks for computational neuroscience  
[tutoriel ESANN'2008]
- An introduction to Deep Learning  
[tutoriel ESANN'2011]
- Computing with Spiking Neuron Networks  
[Handbook of Natural Computing, Springer, 2012]
- De nouveaux paradigmes pour les réseaux de neurones : [Sémin. LAMIA, janv. 2016]  
“Deep Learning” et “Reservoir Computing” [tutoriel CAp'2012]
- Learning sparse features with an auto-associator  
[Growing Adaptive Machines, Springer, 2014]