INSTITUT NATIONAL POLYTECHNIQUE DE GRENOBLE

DEA en Sciences Cognitives

MONOPLAN: Un réseau constructiviste avec la règle *Minimerror*

Juan Manuel Torres Moreno

Jury:
Bernard Amy
Jeanny Hérault
Mirta B. Gordon

CEA/Département de Recherche Fondamentale sur la Matière Condensée
Centre d'Etudes Nucléaires de Grenoble
FRANCE ***30 Juin, 1994***

PLAN DE LA EXPOSE

- **MOTIVATION**
- **LA REGLE MINIMERROR**
- **L'ALGORITHME MONOPLAN**
- **SIMULATIONS NUMERIQUES**
- **CONCLUSION**

HUMAN FAIL ON XOR PATTERN CLASSIFICATION PROBLEMS

S.J. Thorpe, K.O. Regan et A. Pouget
(Institut des Neurosciences, Univ. Pierre et Marie Curie / Lab. de Psychologie Expérimentale. Paris 1989)

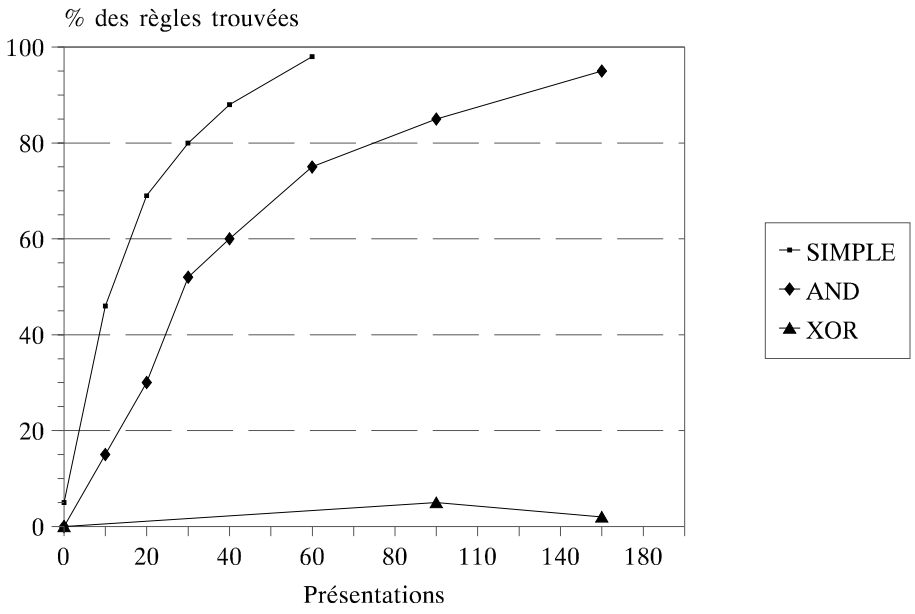
EXPERIENCE.

Configuration de 4x4 carrés illuminés présentés à une certaine vitesse.
On a mis 3 types des règles:

SIMPLE	1 carré est "ON"
ET (AND)	1 pair de carrés sont "ON"
OU EXCLUSIF (XOR)	Si un ou l'autre carré sont "ON" mais non tous deux.

BUT. Trouver la règle correspondante à la configuration

RESULTATS.



I. MOTIVATION

LA PROBABILITE DE REALISER FONCTIONS BINAIRES COMME FONCTION DU NOMBRE DE COUCHES

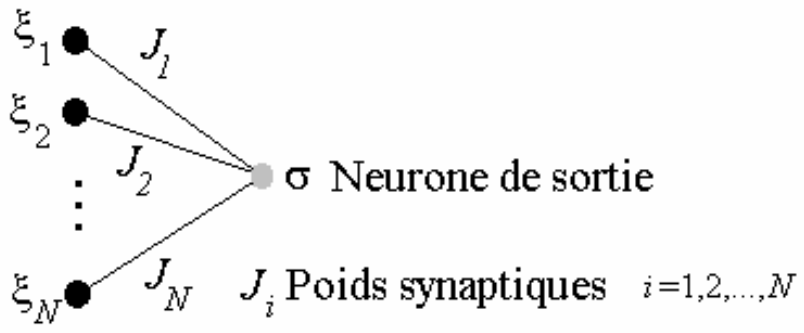
Gordon et Peretto, 1990

4 ou 5 couches sont suffisantes pour que la probabilité de réaliser n'importe quelle fonction binaire à 2 entrées, soit équiprobable.

- Dans les systèmes naturels, on pense que l'**ARCHITECTURE COGNITIVE** est déterminée en partie **GENETIQUEMENT** et en partie par l'**APPRENTISSAGE**.

- Dans les systèmes artificiels neuronaux, l'architecture et les efficacités synaptiques devraient être déterminées par un **PROCESSUS D'APPRENTISSAGE** afin d'obtenir un comportement désiré face à un problème posé.

LE PERCEPTRON



N Neurones d'entrée

FONCTIONNEMENT

• Un neurone *i* qui se trouve dans un état ξ_i il agit sur la sortie avec une intensité proportionnelle à son état.
• La somme des influences creéé un **CHAMP** à la sortie:

$$h = \sum_{i=1}^{N} J_i \xi_i$$

• Si le champ dépasse un seuil d'activation $\Theta = -J_0$, la sortie devient active, dans le cas contraire elle est inactive:

$$\sigma = \begin{cases} +1 & \text{si } h > \Theta \\ -1 & \text{si } h < \Theta \end{cases}$$

1. L'ENSEMBLE D'APPRENTISSAGE

Constitué de ***P*** couples:

$$\{ \text{exemple, sortie} \} = \{ \xi^{\mu}, \tau^{\mu} \}$$

Chaque exemple sera à ***N*** entrées.

Le type d'apprentissage à etudier sera:

- **SUPERVISE:** On connaît la réponse que le réseau doit donner aux exemples.
- **BINAIRE:** Etats des neurones d'entrée sont ± 1.
- **EXHAUSTIF:** Si $P = 2^N$
- **NON-EXHAUSTIF:** Si $P = \alpha N < 2^N$

2. FONCTIONS BOOLEENNES

L'ensemble B de toutes les fonctions booléennes à N entrées contient N_B éléments:

$$N_B = 2^{2^N}$$

Les fonctions booléennes **LINEAIREMENTE SEPARABLES (LS)** appartiennent à un sous ensemble réduite de B, avec N_L éléments d'environ:

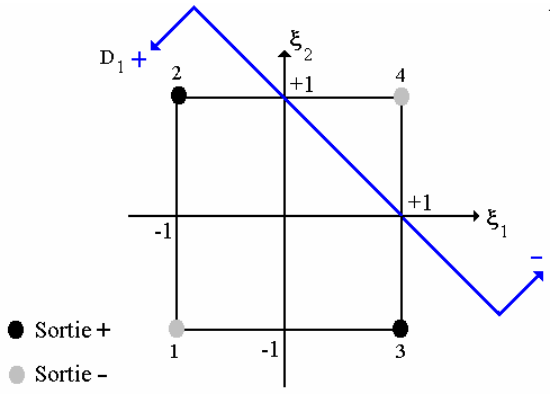
$$N_L \cong 2\,\frac{2^{N^2}}{N!}$$

Par exemple, pour $N = 4$:

- $N_B = 65536$
- $130 < N_L < 170$

Le perceptron peut apprendre seulement des fonctions **LS**

- Le OU EXCLUSIF (*XOR*) n'admet pas une séparation linéaire par un plan quel que soit son orientation.



u	ξ1	ξ2	τ
1	– 1	– 1	1
2	– 1	1	– 1
3	1	– 1	– 1
4	1	1	1

OU EXLUSIF

3. LA STABILITE

On définit la **STABILITE** γ^{μ} d'un exemple μ comme:

$$\gamma^{\mu} = \tau^{\mu} \frac{\vec{J}}{\|\vec{J}\|} \cdot \vec{\xi}^{\mu}$$

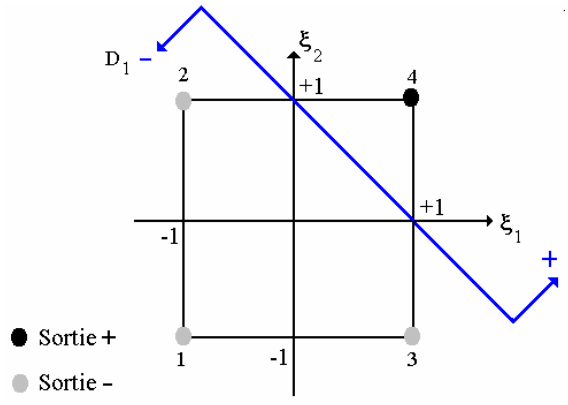
avec:

$$\|\vec{J}\| = \sqrt{\sum_{i=0}^{N} J_i^2} \quad = \text{constante}$$

- La **STABILITE** *est la distance de l'exemple* μ *à l'hyperplan séparateur:*

  $\gamma^{\mu} > 0$ si l'exemple est bien appris
  $\gamma^{\mu} < 0$ autrement.

- Une grande stabilité positive assure une certaine robustesse de la réponse du neurone.



μ	ξ1	ξ2	τμ
1	– 1	– 1	– 1
2	– 1	1	– 1
3	1	– 1	– 1
4	1	1	1

ET (AND)

REGLES D'APPRENTISSAGE TYPE HEBBIAN

Il y a règles d'apprentissage du type Hebbian:

- **LA REGLE DE WIDROW-HOFF**
- **LA REGLE DU PERCEPTRON**

Problèmes:

- **LES ALGORITHMES NE CONVERGENT QUE SI UNE SEPARATION LINEAIRE EST POSSIBLE.**

- **S'IL EXISTE PLUSIEURS SOLUTIONS LA *QUALITE* DE LA SOLUTION TROUVEE N'EST PAS OPTIMALE.**

PROBLEME

- ***Quelle règle d'apprentissage?***
- ***Quelle architecture?***
- ***Combien d'unités par couche...?***

II. MINIMERROR

1. L'ERREUR MOYENNE

La règle d'apprentissage ***MINIMERROR*** utilise des considérations de physique statistique pour évaluer l'erreur moyenne.

Elle repose sur deux idées fondamentales:

1. MINIMISER LE NOMBRE DES FAUTES.

2. FAIRE QUE LES STABILITES POSITIVES SOIENT LES PLUS GRANDES POSSIBLES.

MINIMERROR travaille avec la minimisation de la fonction coût:

$$\langle n \rangle = \sum_{\mu=1}^{P} V\left(\gamma^{\mu}, \beta\right)$$

avec:

$$V\left(\gamma^{\mu}, \beta\right) = \frac{1}{2}\left(1 - \tanh \frac{\beta \gamma^{\mu}}{2}\right)$$

β est un taux de bruit.

MINIMERROR consiste à minimiser $\langle n \rangle$ à une température finie $T=1/\beta$

2. LE PROBLEME DE LA PARITE A *N* ENTREES

La généralisation à *N* entrées du *XOR* s'appelle la **PARITE: *N*-PARITE**

$$\textbf{\textit{N}-PARITE} = \begin{cases} 1 & \text{Si le nombre de neurones dans l'état 1 à l'entrée est pair.} \\ -1 & \text{Autrement.} \end{cases}$$

La *N*-Parité est une fonction intéressante:

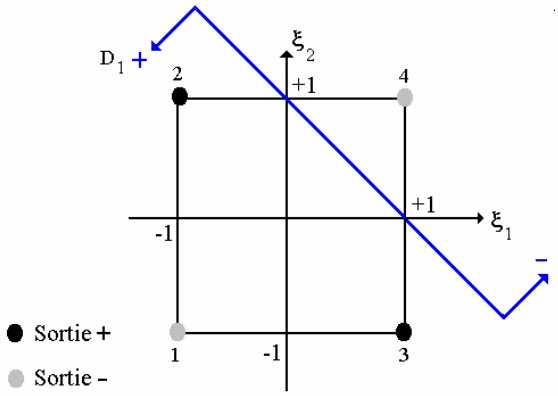
Un perceptron ne peut pas résoudre le problème, mais avec un bon algorithme d'apprentissage on peut réaliser le **NOMBRE MINIMUM DE FAUTES**.

μ	ξ1	ξ2	ξ3	τ
1	−1	−1	−1	1
2	−1	−1	1	−1
3	−1	1	−1	−1
4	−1	1	1	1
5	1	−1	−1	−1
6	1	−1	1	1
7	1	1	−1	1
8	1	1	1	−1

La 3-Parité

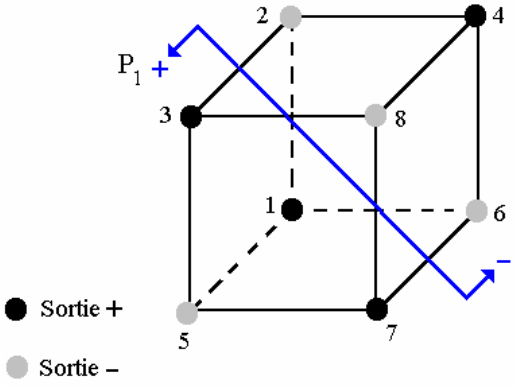
LE NOMBRE MINIMUM DE FAUTES

Mais, quel est le **NOMBRE MINIMUM DE FAUTES** (NMF) pour la *N*- Parité?



Le XOR

NMF = 1 Faute



La 3-Parité

NMF = 2 Fautes...

	k=0	1	2	3	4	5	6	7	8	9	10	
N	-	+	-	+	-	-	+	-	+	-	+	**Fautes**
2	1	2	1									**1**
3	1	3	3	1								**2**
4	1	4	6	4	1							**5**
5	1	5	10	10	5	1						**10**
6	1	6	15	20	15	6	1					**22**
7	1	7	21	35	35	21	7	1				**44**
8	1	8	28	56	70	56	28	8	1			**93**
9	1	9	36	84	126	126	84	36	9	1		**186**
10	1	10	45	120	210	252	210	120	45	10	1	**386**

NMF pour la *N*-Parité

Soit ν_k la distribution du nombre des sommets à sortie -1 ou +1 de façon alternée, séparés par des hyperplans successifs.

Le tableau représente le triangle de Pascal.
La distribution des sommets ν_k est donnée par les coefficients du binôme:

$$\nu_k = \binom{N}{k} = \frac{N!}{k!\ (N-k)!}$$

ANALYSE

• Pour *N* impair, le NMF est égal à deux fois le nombre des fautes commises par un perceptron à *N*-1 entrées.

• Pour *N* pair, grâce à la symétrie du problème on doit comptabiliser seulement les erreurs commises par les premiers *N*/2 hyperplans.

$$f(N)=\begin{cases} f(N=2p)=\sum_{i=1}^{p}\binom{2p}{2p+i-1} \\ f(N=2p+1) \;=\; 2f(2p) \end{cases} \qquad p = 1,2,3,...,N$$

SIMULATION NUMERIQUE

```
Fichier des entrées : 3-PARITE
Neurones (N)        :     3
Patterns (P)        :     8
-------------------------------- NEURONE CACHE 1
Erreur =   2
J:{ -0.11  1.16 -1.16 -1.14 }
-----------------------------------------------


Fichier des entrées : 5-PARITE
Neurones (N)        :     5
Patterns (P)        :    32
-------------------------------- NEURONE CACHE 1
Erreur =  10
J:{ -0.02 -1.01 -1.01 -1.01 -1.01 -1.01 }
-----------------------------------------------
```

III. MONOPLAN

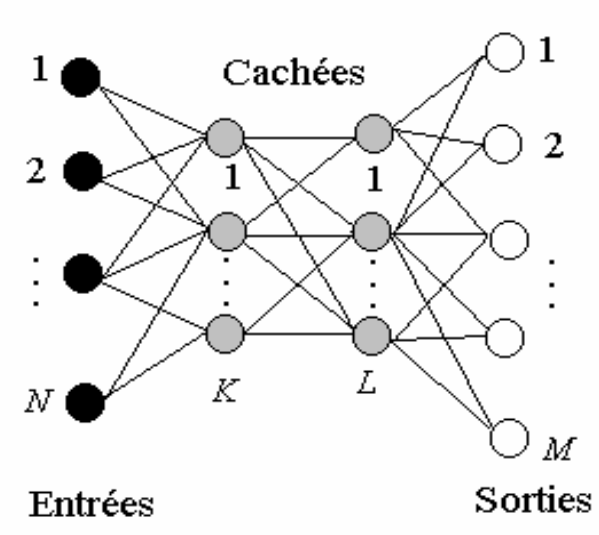
1. LES RESEAUX MULTICOUCHES

- Résoudre des problèmes complexes (non-LS)
- Chaque unité est un perceptron simple
- Unités connectées formant des couches cachées

1. *Quelle est l'architecture la plus performante du réseau?*

   - **ARCHITECTURE FIXE**
   - **ARCHITECTURE CONSTRUCTIVISTE**

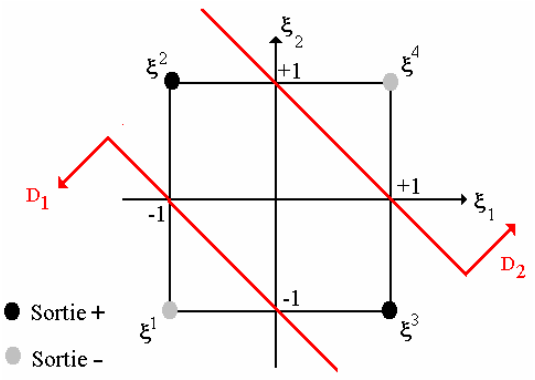
2. *Combien d'unités cachées doit-on mettre dans chaque couche?*



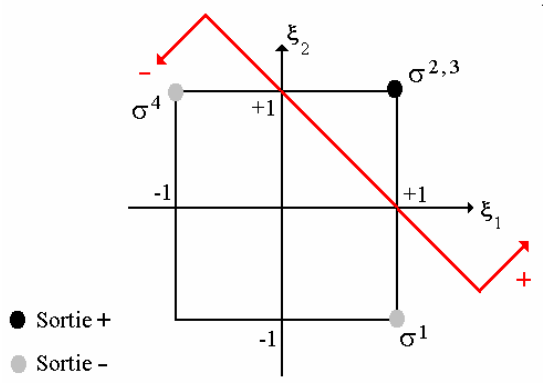
2. LES REPRESENTATIONS INTERNES

Les **REPRESENTATIONS INTERNES** (***RI***) sont l'ensemble d'états des neurones cachés.

Il y a une *RI* σ^{μ} pour chaque exemple ξ^{μ}



Solution pour le *XOR*



RI σ^{μ}

μ	ξ1	ξ2	τμ
1	−1	−1	−1
2	−1	1	1
3	1	−1	1
4	1	1	−1

Entrées ξ^{μ}

σ1	σ2
1	−1
1	1
1	1
−1	1

σ^{μ}

3. ARCHITECTURE FIXE: LA RETRO-PROPAGATION DE L'ERREUR (RP)

• L'apprentissage:
Minimiser l'erreur quadratique commise sur l'ensemble d'apprentissage par une descente en gradient:

Si l'erreur est inférieure a une limite choisie, on arrête l'apprentissage.

• PROBLEMES DE LA RP...

1. Le nombre de couches cachées et d'unités par couche sont déterminés *a priori* et modifies par **essai et erreur**!

2. Des neurones analogiques doivent toujours être utilisés même pour des problèmes qui nécessitaient uniquement des neurones binaires.

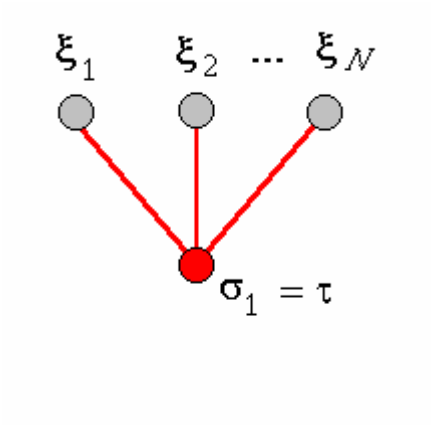
4. ARCHITECTURES CONSTRUCTIVISTES

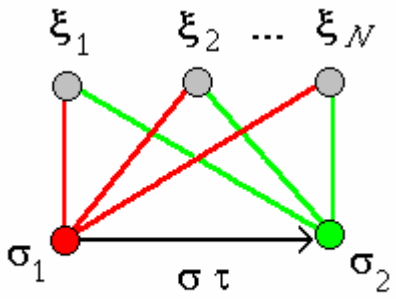
- ***THE TILING ALGORITHM*** (Mézard et Nadal, 1989)
- ***THE UPSTART ALGORITHM*** (Frean, 1990)
- ***THE OFFSET ALGORITHM*** (Martinez et Estève, 1992)
- ***MONOPLAN*** (Peretto et Gordon, 1992)

5. L'ALGORITHME *MONOPLAN*

COUCHE CACHEE.

- Enseigner au premier neurone l'ensemble d'apprentissage $\{\xi^\mu, \tau^\mu\}$ avec ***MINIMERROR***
- Calculer l'état σ_1^μ
- S'il y a des erreurs, on ajoute le deuxième neurone mais pour lui, on enseigne la projection du neurone avant $\sigma_1^\mu \tau^\mu$
- Ajouter neurones tandis qu'il y a des erreurs, avec la projection $\sigma_{NH-1}^\mu \tau_{NH-1}^\mu$ pour l'unité *NH*.

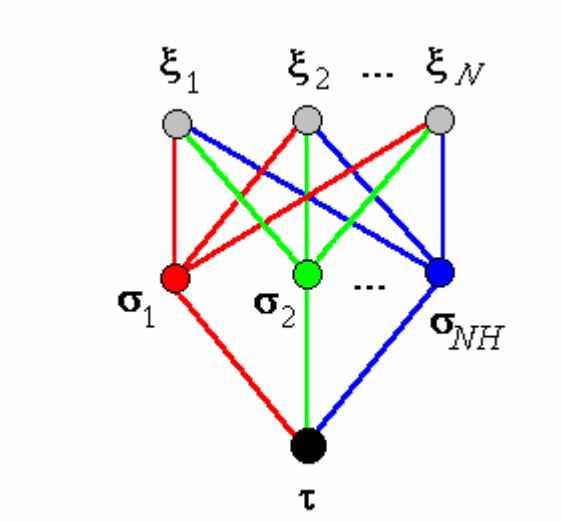




SORTIE.

- Enseigner l'ensemble {Représentations Internes, sortie} $\{\sigma^{\mu}, \tau^{\mu}\}$ avec ***MINIMERROR***
- Calculer son état σ
- S'il n'y a plus des fautes la procédure termine, sinon ajouter l'unité *NH*+1
- Itérer avec la projection $\sigma\ \tau^{\mu}$ jusqu'à la convergence.

ENTREES → *REP. INTERNES* → SORTIE



MONOPLAN construit un réseau simple: à une couche cachée

IV. SIMULATIONS

1. APPRENTISSAGE EXHAUSTIF DE LA *N*-PARITE

Tests de la *N*-Parité (*N*≤10).

On a trouvé toujours la solution optimale avec *NH*=*N* unités cachées.

		Couche cachée								
i	Biais	Synapses (couche d'entrée à l'unité i)								
1	-1.04	-1.10	0.52	1.00	1.03	-1.03	-1.07	-1.02	-1.00	-1.07	-1.00
2	1.44	-0.93	0.88	0.92	0.92	-0.96	-0.93	-0.97	-1.06	-0.93	-0.93
3	2.45	0.68	-0.69	-0.73	-0.71	0.68	0.72	0.74	0.73	0.71	0.68
4	2.47	-0.70	0.72	0.69	0.70	-0.70	-0.71	-0.69	-0.71	-0.68	-0.69
5	2.87	-0.54	0.54	0.51	0.53	-0.52	-0.52	-0.54	-0.50	-0.54	-0.52
6	2.84	0.49	-0.50	-0.58	-0.53	0.54	0.54	0.57	0.56	0.54	0.55
7	3.03	0.39	-0.37	-0.46	-0.45	0.41	0.42	0.43	0.47	0.42	0.45
8	3.06	-0.45	0.46	0.33	0.39	-0.42	-0.43	-0.40	-0.37	-0.43	-0.39
9	3.12	0.23	-0.17	-0.62	-0.40	0.26	0.19	0.31	0.49	0.25	0.39
10	3.12	-0.49	0.63	0.17	0.22	-0.28	-0.42	-0.33	-0.22	-0.38	-0.15

	Unité de sortie										
Biais	Synapses (couche cachée à la sortie)										
1.00	1.00	-1.00	1.00	1.00	-1.00	-1.00	1.00	1.00	-1.00	-1.00

La 10-Parité

DEGENERESCENCE DES RI

- A chaque exemple de l'entrée on associe une ***RI***.
- Plusieurs exemples peuvent être associés à la même ***RI.***

On aura pour *P* exemples, *P** ***RI*** différentes. *P** < *P.*

On doit travailler avec toutes ou sans les ***RI*** répétés?...

DEUX REPRESENTATIONS INTERNES IDENTIQUES
SONT INCAPABLES DE PRODUIRE DEUX SORTIES DIFFERENTES

Donc, on a décidé:

- Eliminer les RI répétés ⇒
  Trouver la sortie la plus stable possible.

AVEC OU SANS TOUTES LES REPRESENTATIONS INTERNES?

Solution pour la 6-Parité avec toutes les ***RI***.

	Couche cachée						
i	Biais	Synapses (couche d'entrée à l'unité i)					
1	1.00	-1.00	1.00	-1.00	-1.00	1.00	-1.00
2	1.42	0.91	-0.91	0.91	0.91	-0.91	0.91
3	2.17	-0.62	0.62	-0.62	-0.62	0.62	-0.62
4	2.17	0.62	-0.62	0.62	0.62	-0.62	0.62
5	2.38	0.47	0.47	-0.47	-0.47	0.47	-0.47
6	2.38	0.47	-0.47	0.47	0.47	-0.47	0.47

Unité de sortie						
Biais	Synapses (couche cachée à la sortie)					
-0.73	**1.16**	**1.16**	**-1.07**	**-1.07**	**0.86**	**0.86**

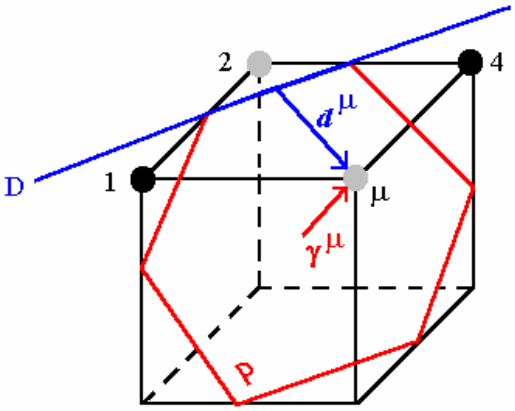
Solution sans ***RI*** répétés (la couche cachée a les mêmes poids).

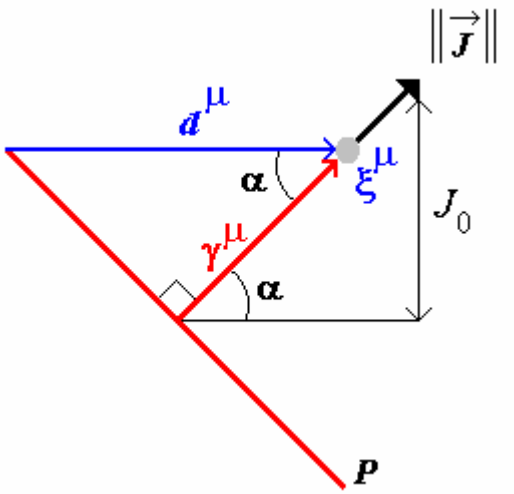
Unité de sortie						
Biais	Synapses (couche cachée à la sortie)					
-1.00	**1.00**	**1.00**	**-1.00**	**-1.00**	**1.00**	**1.00**

LA STABILITE DANS LE MEME ESPACE DES ENTREES

Pour un problème à 2 entrées, la **STABILITE** γ^{μ} mesure la distance du plan P à l'exemple μ dans un espace **TRIDIMENSIONNEL.**

La distance efficace qu'on *voudrait maximiser* est la **DISTANCE** d^{μ} (de la droite *D*) mesuré sur un espace **BIDIMENSIONNEL.**

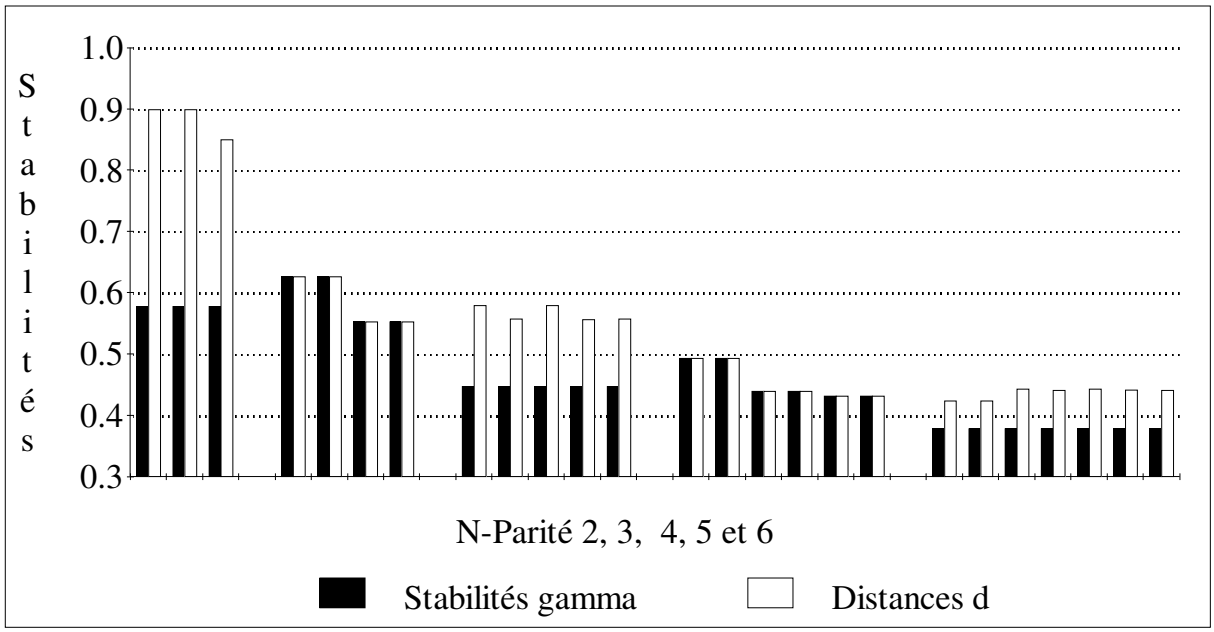




γ^{μ} vs. d^{μ}

$$d^{\mu} = \frac{\gamma^{\mu}}{\sqrt{1 - \frac{J_0^2}{\|\vec{J}\|^2}}}$$

Résultats de la *N*-Parité ($N \leq 6$) à la sortie.



$\gamma\mu$ vs $d\mu$

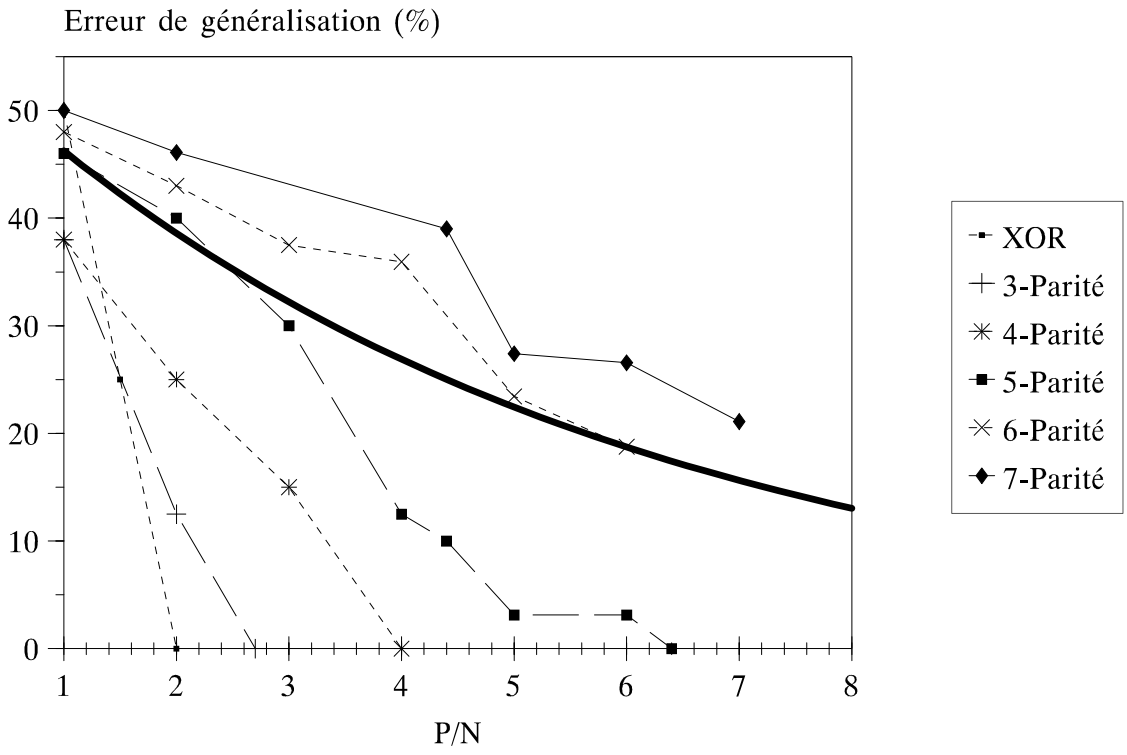
2. APPRENTISSAGE NON EXHAUSTIF DE LA *N*-PARITE: LA GENERALISATION

Si l'ensemble d'apprentissage a moins de 2^N exemples (NON-EXAHUSTIF), on peut se poser la question:

Quelle est la performance du réseau face à des exemples qui n'a vu jamais?...

La généralisation de la *N*-Parité ($N \leq 7$).
Pourcentage de réponses erronées par rapport à $\alpha = P/N$

Généralisation pour la *N*-Parité



V. CONCLUSION

1. L'architecture final est simple: une seule couche cachée

2. Les tests sur la *N*-Parité sont très satisfaisants: on trouve la solution de réseau minimal de plus grande stabilité et NMF.

   La minimisation avec d^μ rendre plus grandes les stabilités qu'avec γ^μ dans le *N*-Parité. Mais de plus en plus $d^\mu \rightarrow \gamma^\mu$. A la limite quand $N \rightarrow \infty$, $d^\mu \equiv \gamma^\mu$.

PERSPECTIVES

1. Tests sur l'altération aléatoire des sorties:
i). Apprendre une fonction f LS à *N* entrées.
ii). Pour $k=1,...,N$
f ' $\leftarrow$ f avec *k* bits changés
Essayer d'apprendre f ' avec *NH* unités
Vérifier que $NH \leq k+1$.

2. Utilisation de momentum α, pour la descente en gradient.

3. Tests sur l'altération des synapses.

4. Parallèliser les algorithmes.

Si l'Homme est "Neuronal", le Neurone, lui, est très certainement inhumain.