

Introduction aux réseaux de Neurones :

De la biologie...
...aux sciences pour l'ingénieur

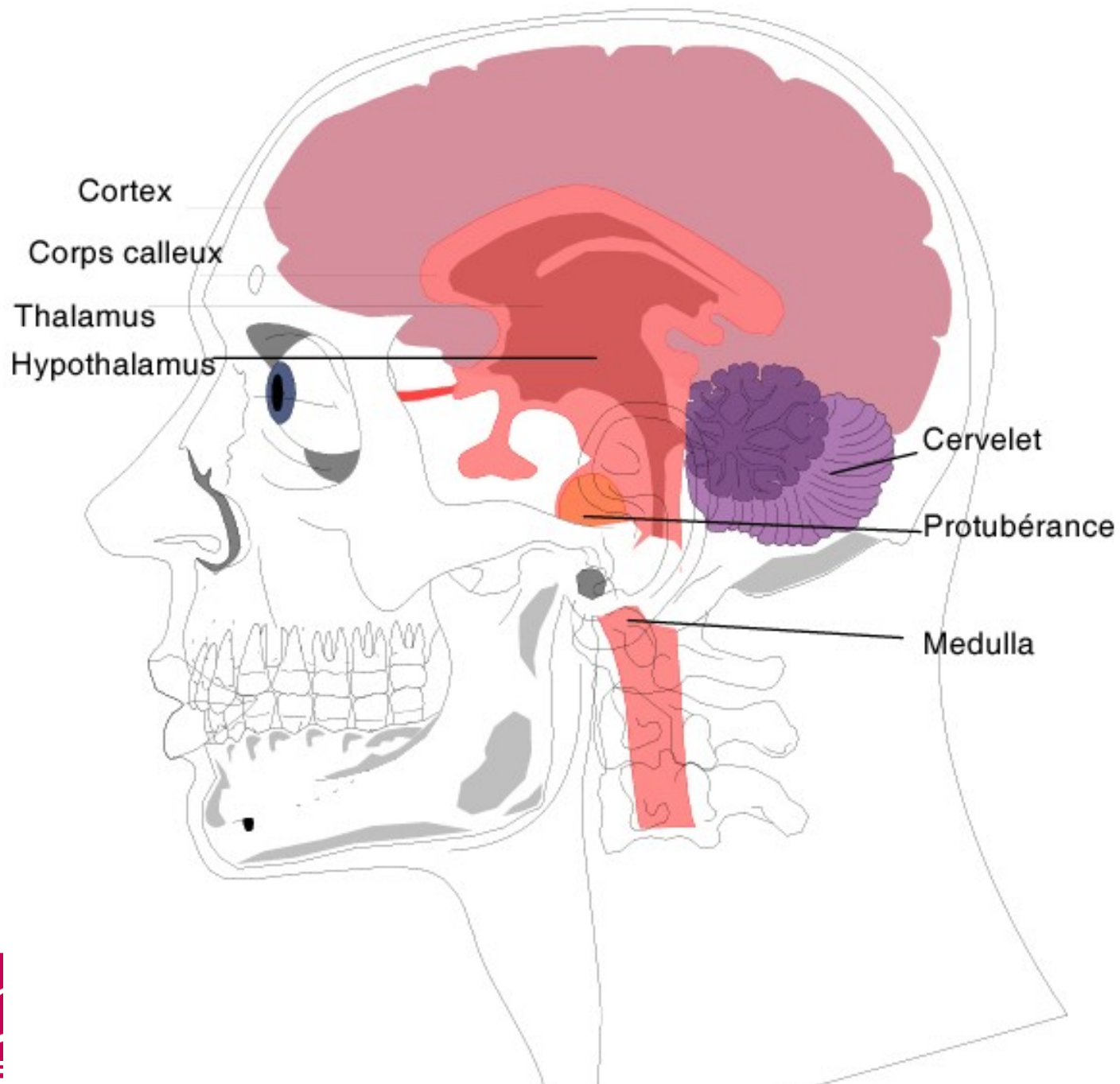
Introduction

- Objectif
 - Découvrir les approches d'intelligence artificielles d'inspiration biologique
 - Réseaux de neurones
 - Architectures comportementales
 - Application au contrôle robotique

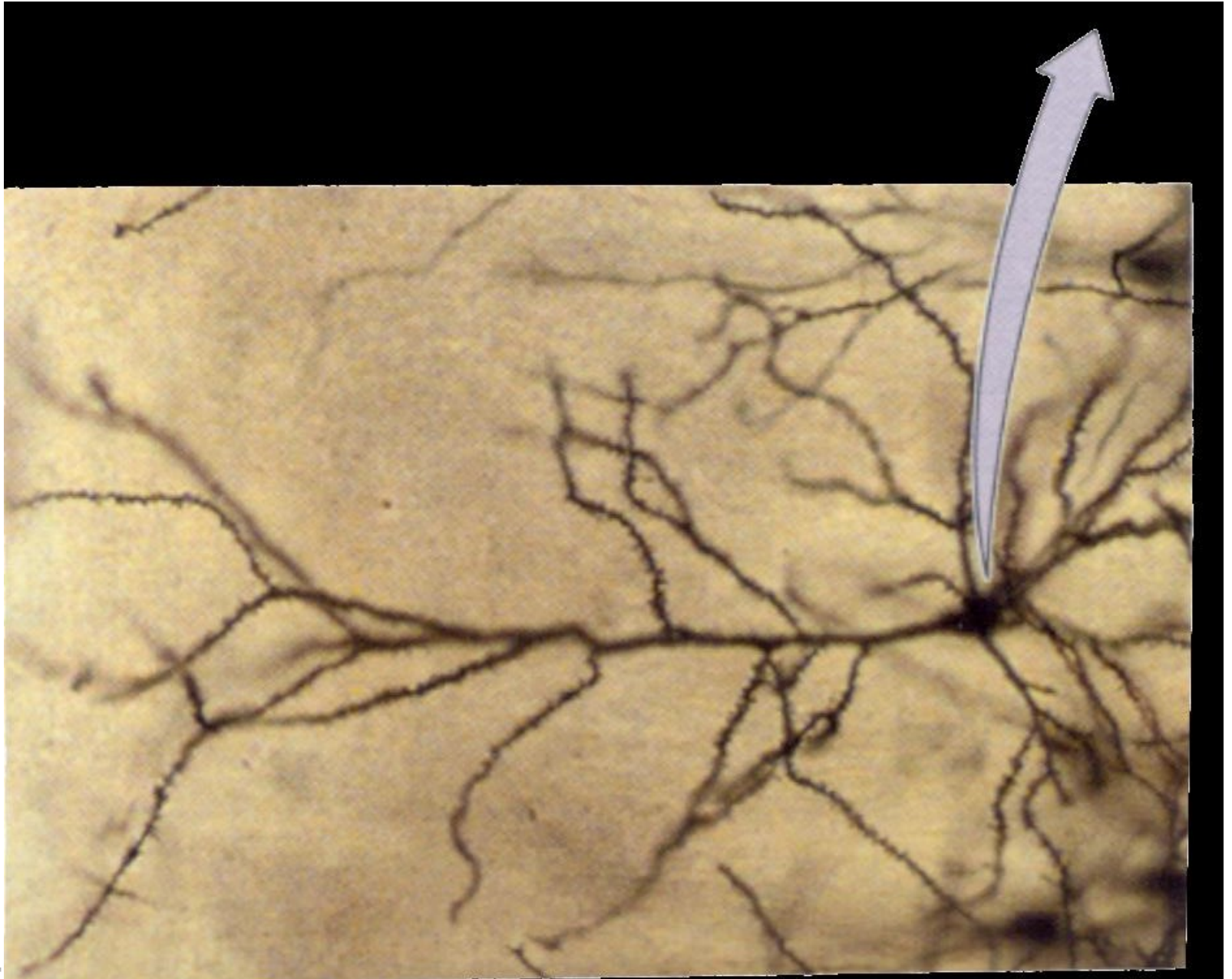
Introduction

- Cerveau humain
 - 10 à 100 Milliards de neurones
- Architecture très différente de celle d'un ordinateur

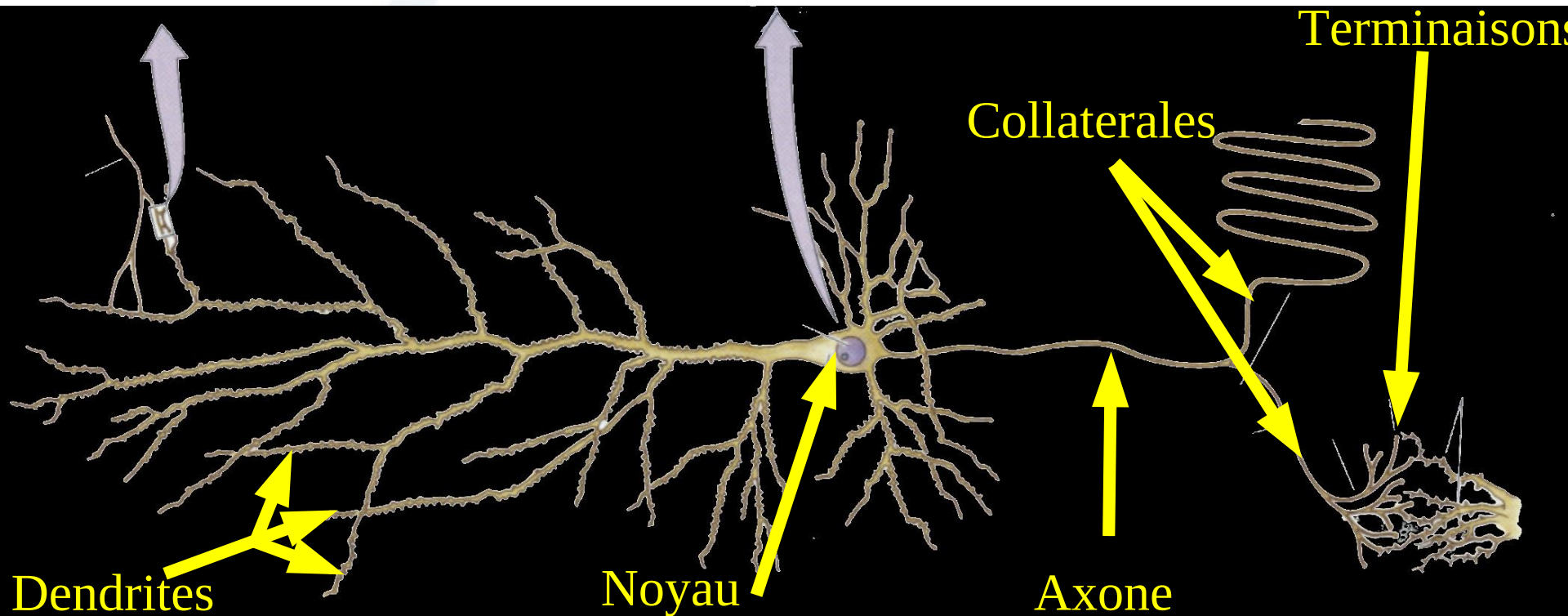
?



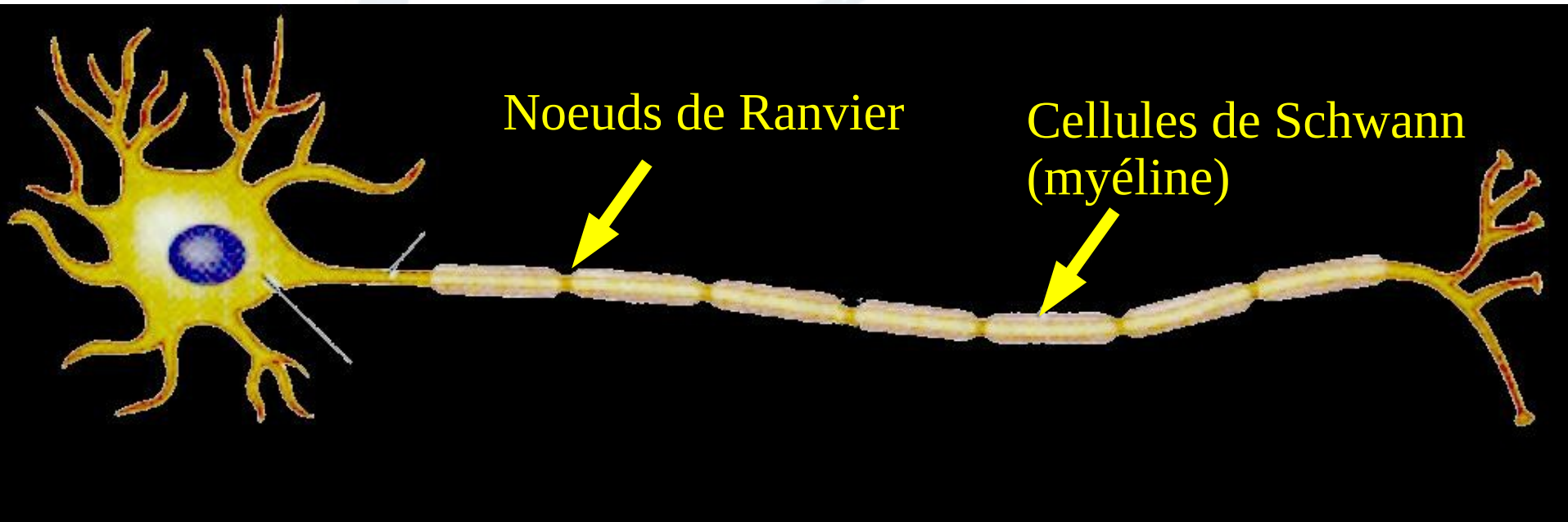
Le neurone biologique

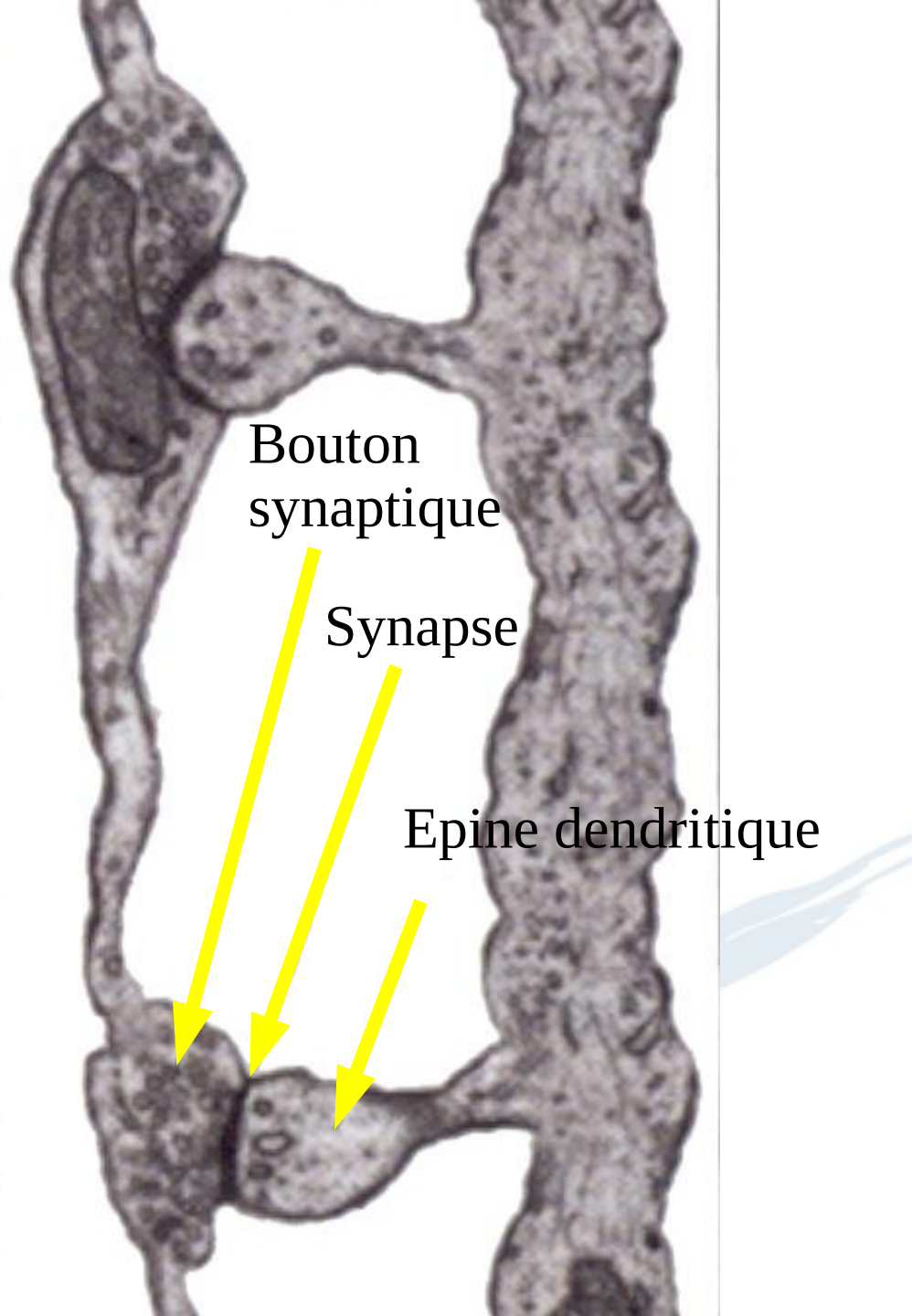


Le neurone biologique



Neurone biologique





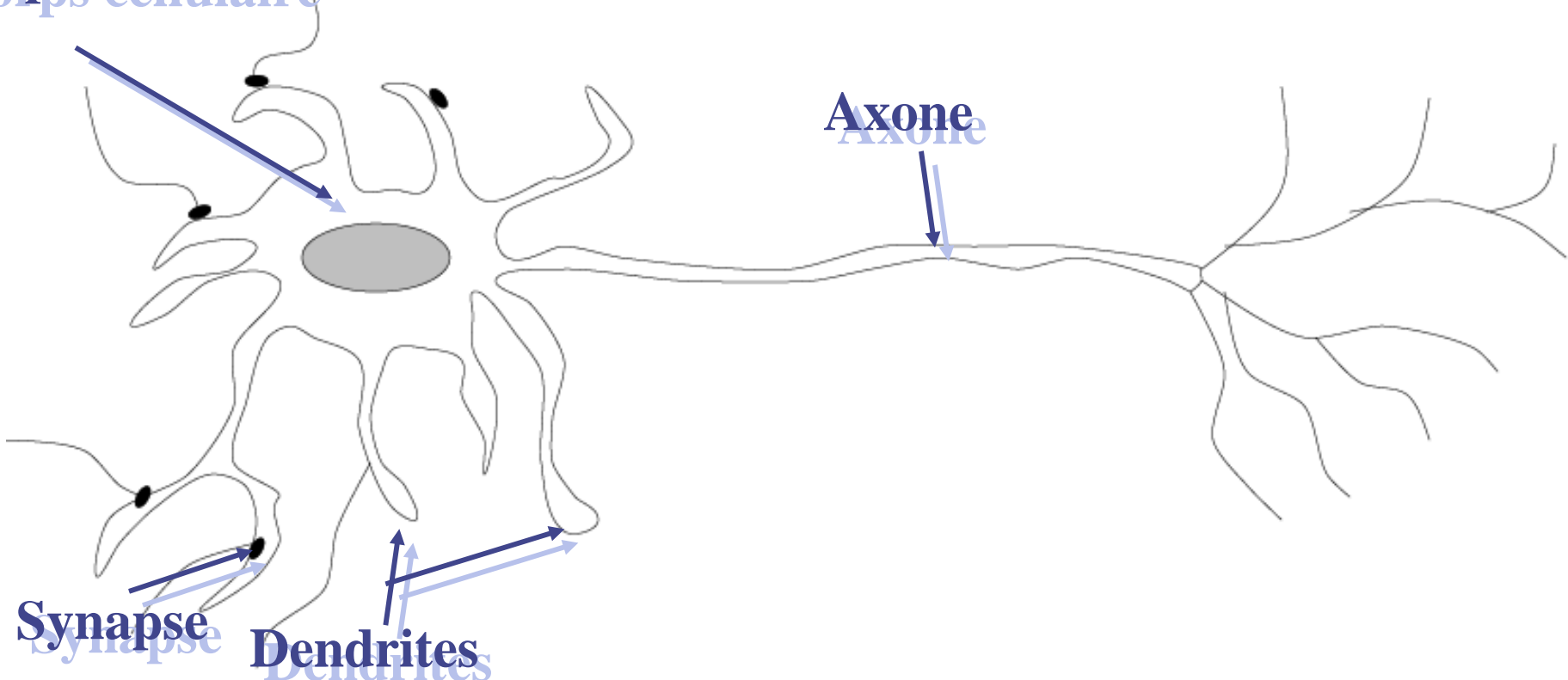
Données neurobiologiques

Corps cellulaire

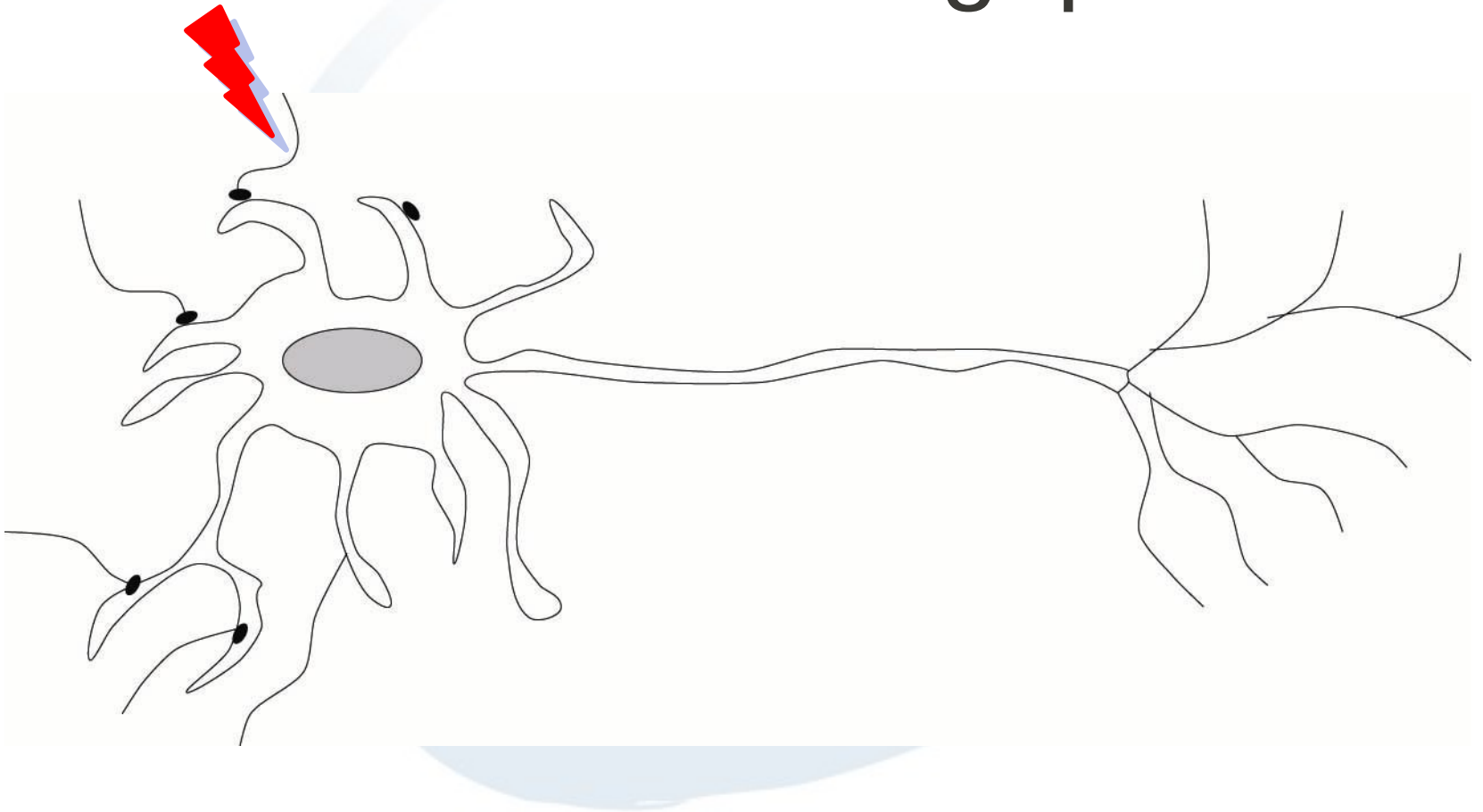
Axone

Synapse

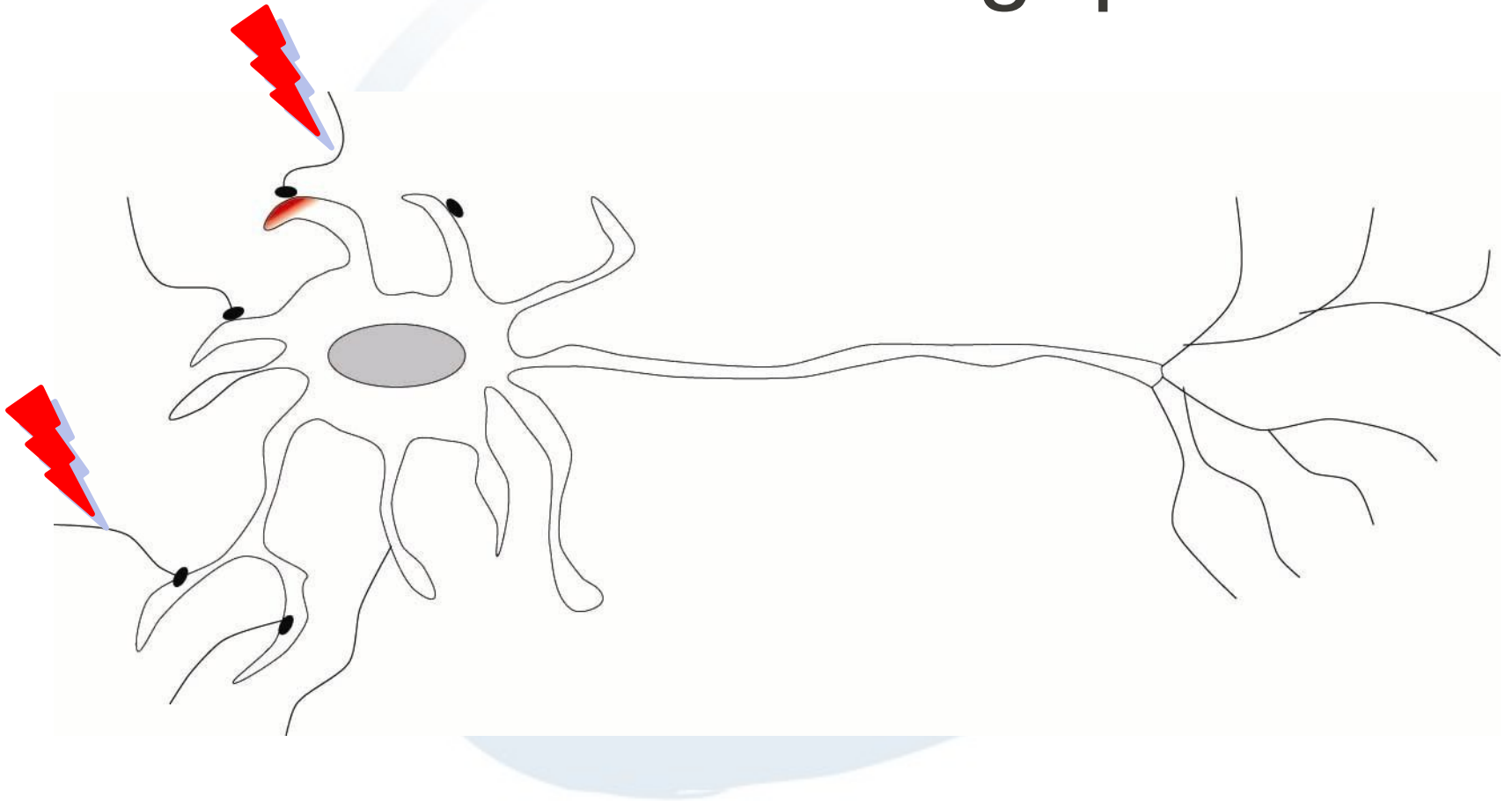
Dendrites



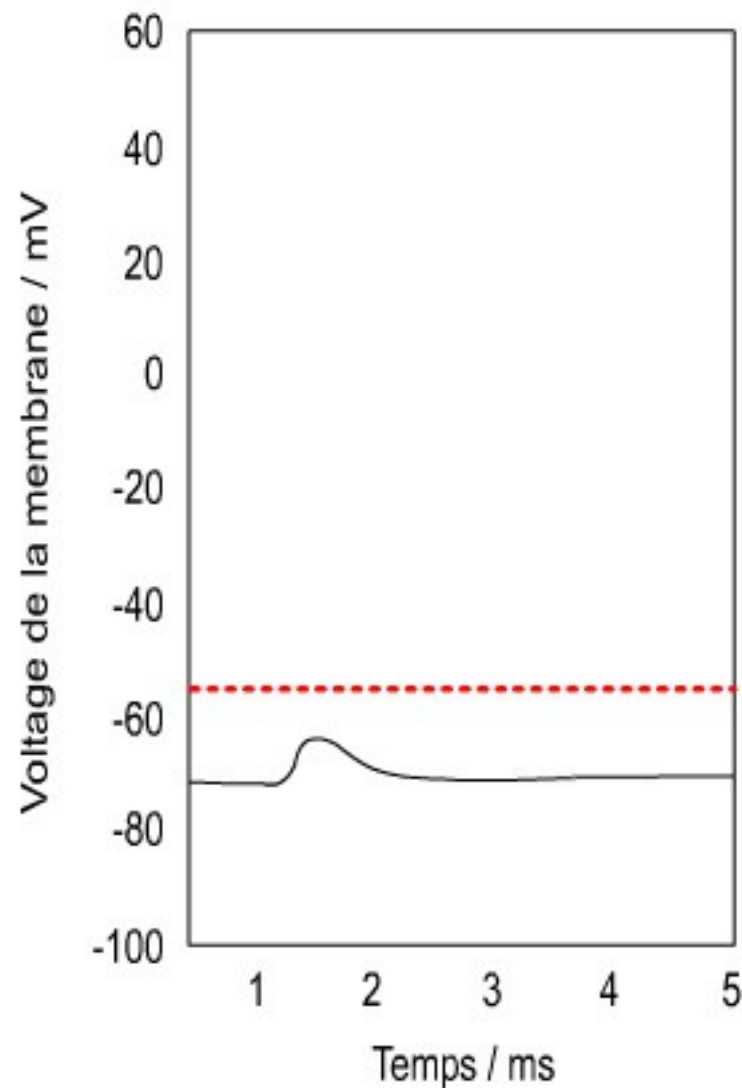
Données neurobiologiques



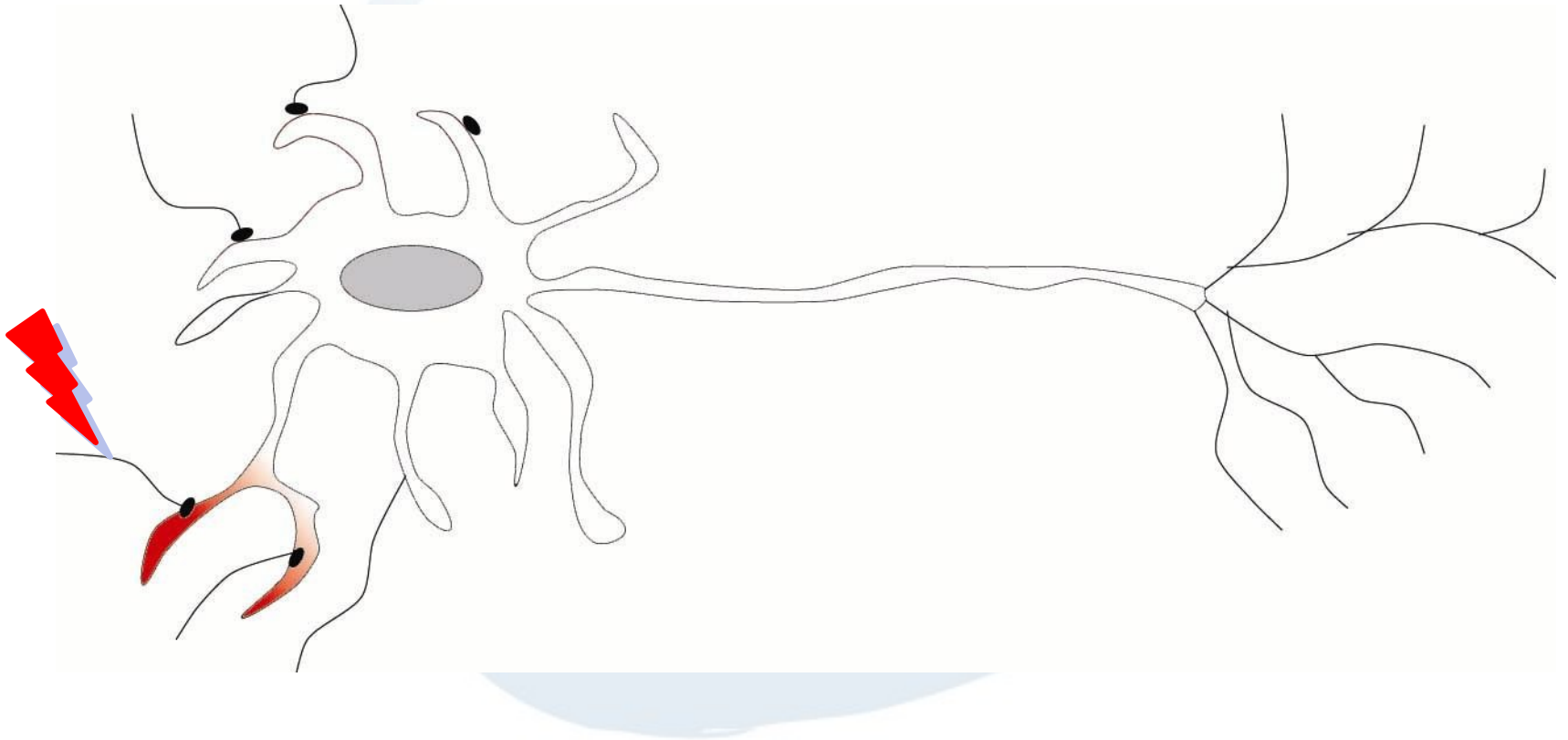
Données neurobiologiques



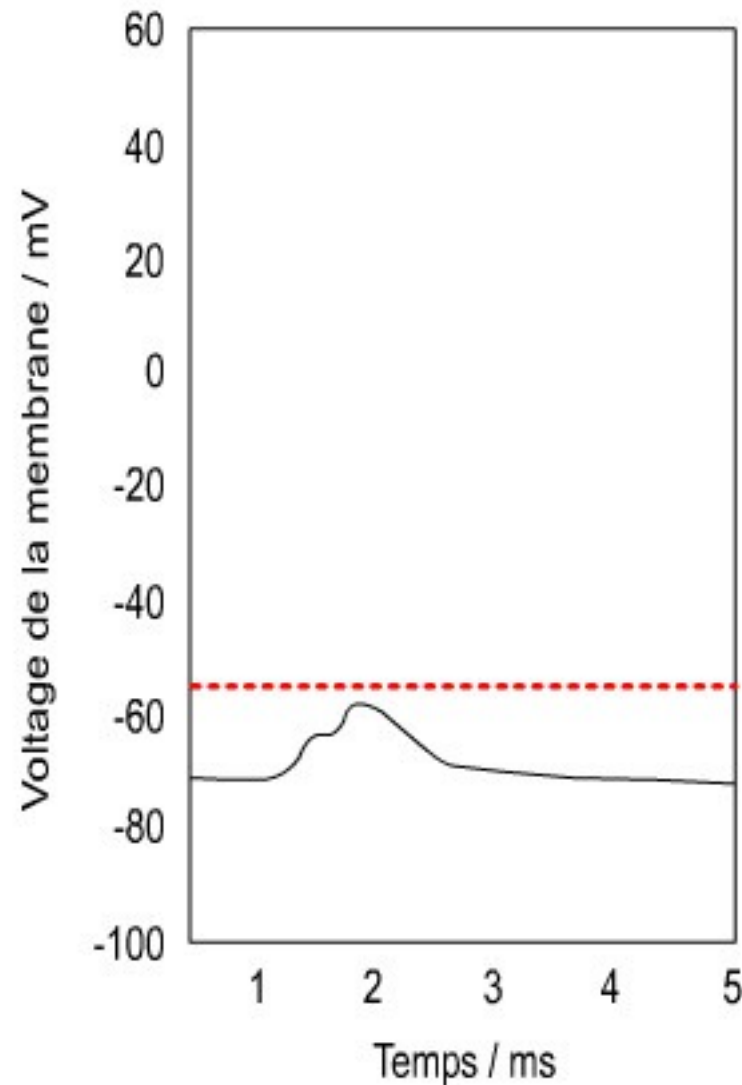
Données neurobiologiques



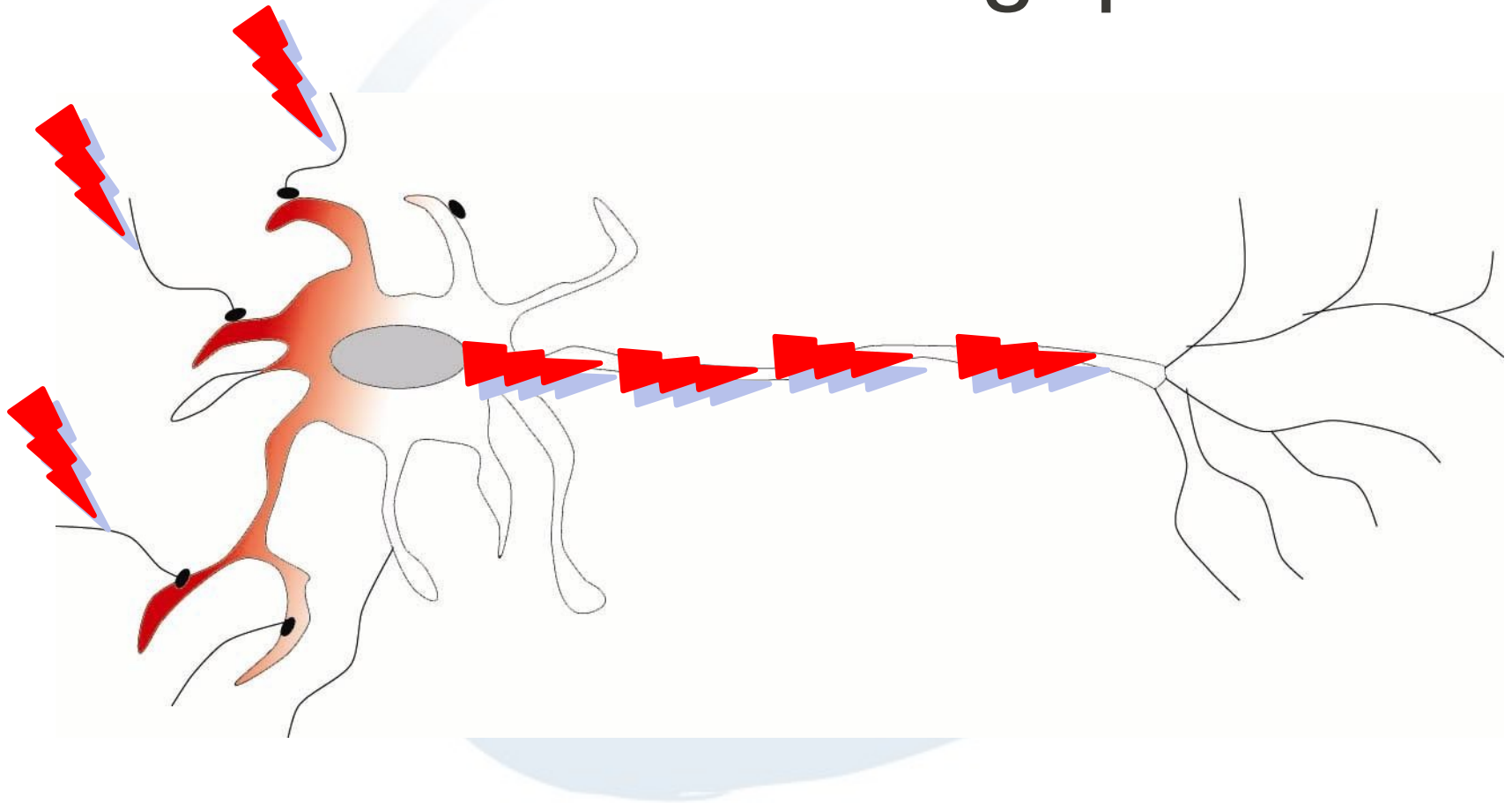
Données neurobiologiques



Données neurobiologiques



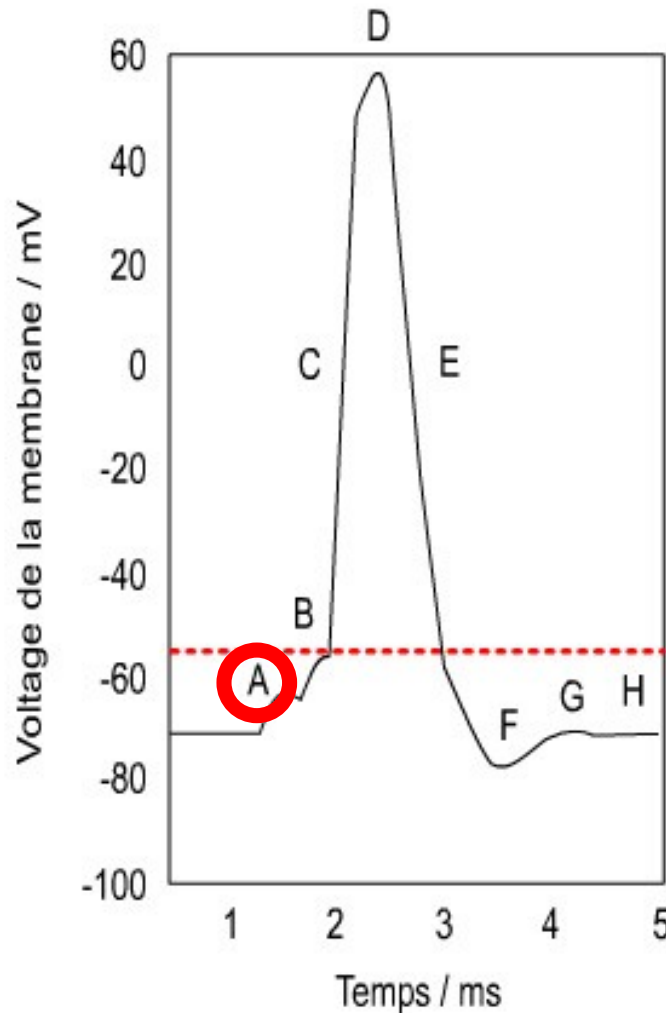
Données neurobiologiques



Physiologie du neurone

- Potentiel du neurone

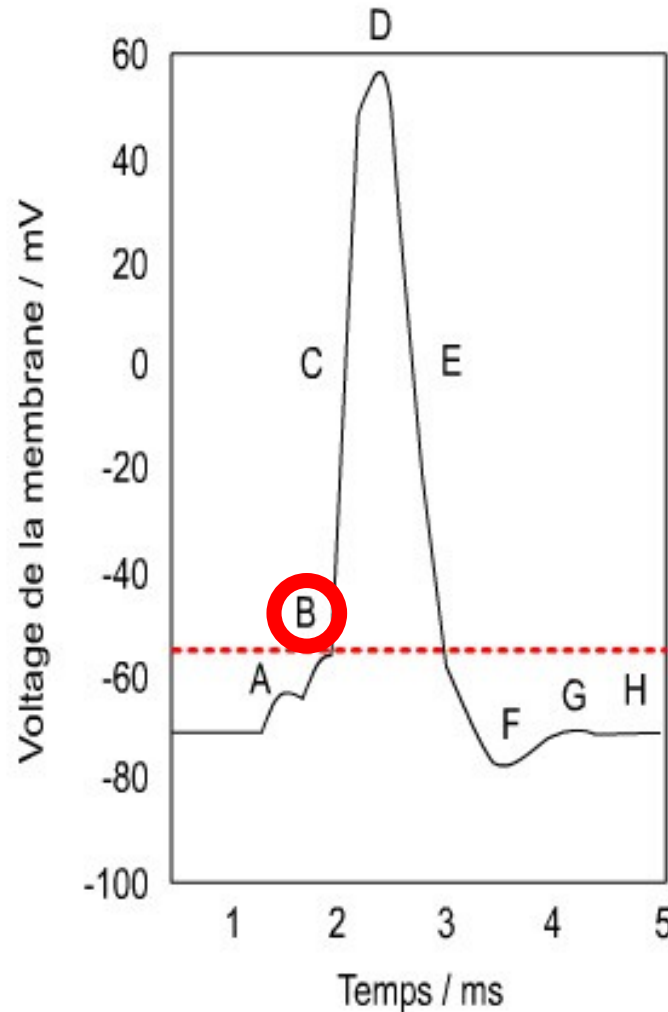
A) potentiel récepteur qui dépolarise le neurone au-delà du seuil.



Physiologie du neurone

- Potentiel du neurone

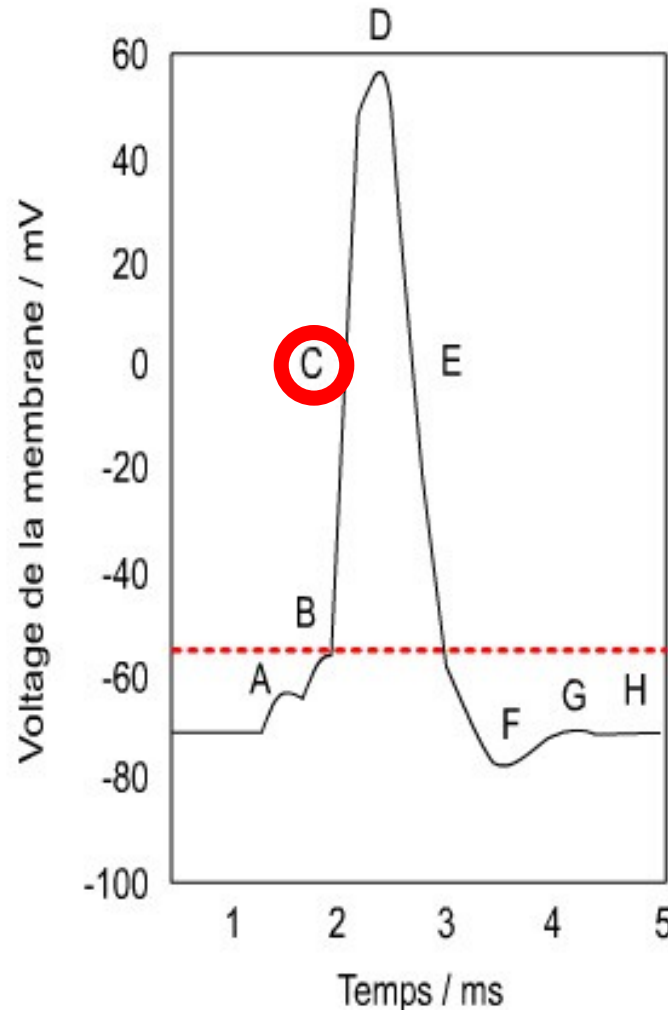
B) les canaux sodiques sensibles au voltage s'ouvrent.



Physiologie du neurone

- Potentiel du neurone

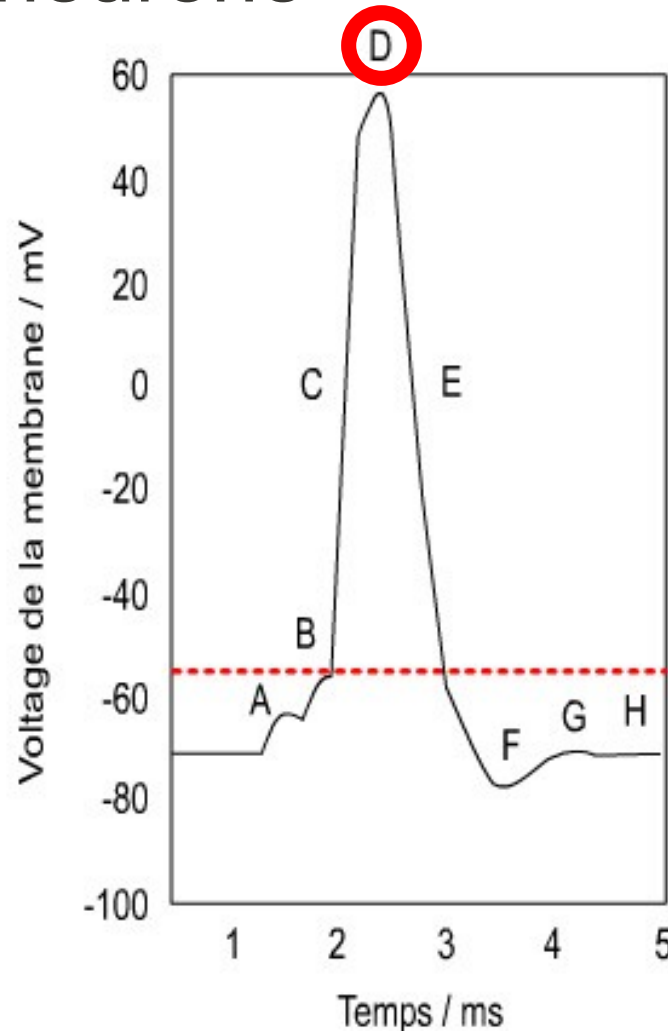
C) entrée massive d'ion sodium qui dépolarisent davantage le neurone.



Physiologie du neurone

- Potentiel du neurone

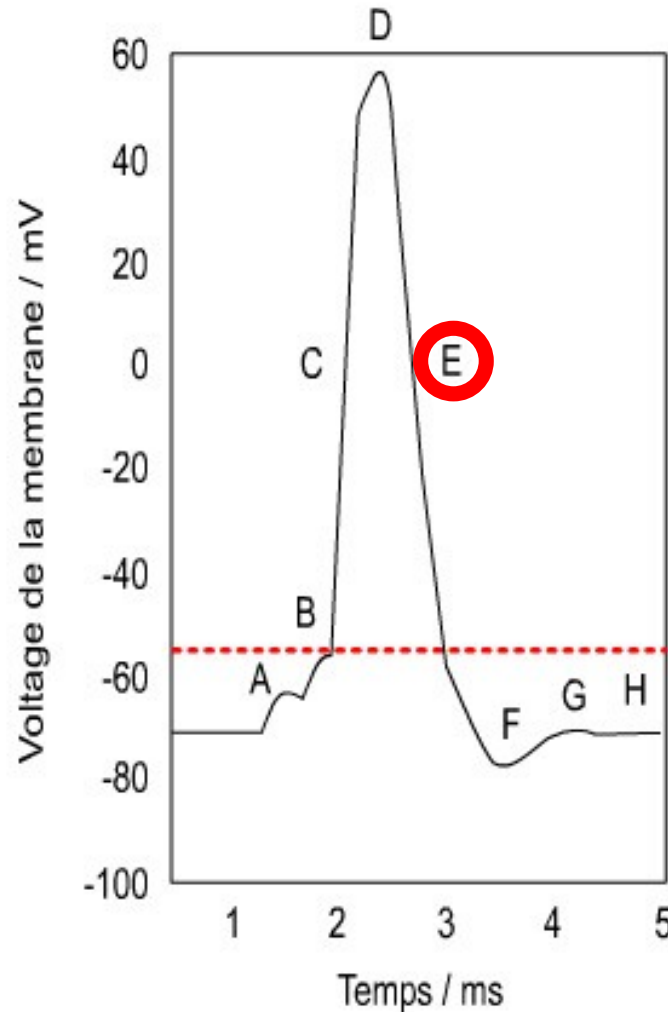
D) fermeture des canaux sodiques et ouverture des canaux potassiques.



Physiologie du neurone

- Potentiel du neurone

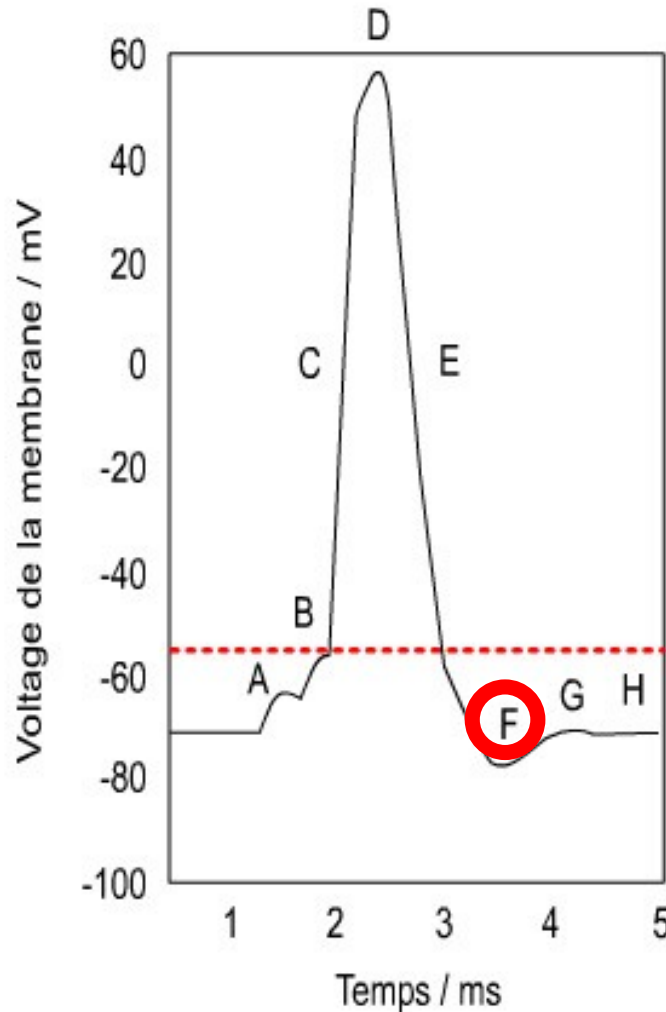
E) les ions potassium chargés positivement sortent du neurone.



Physiologie du neurone

- Potentiel du neurone

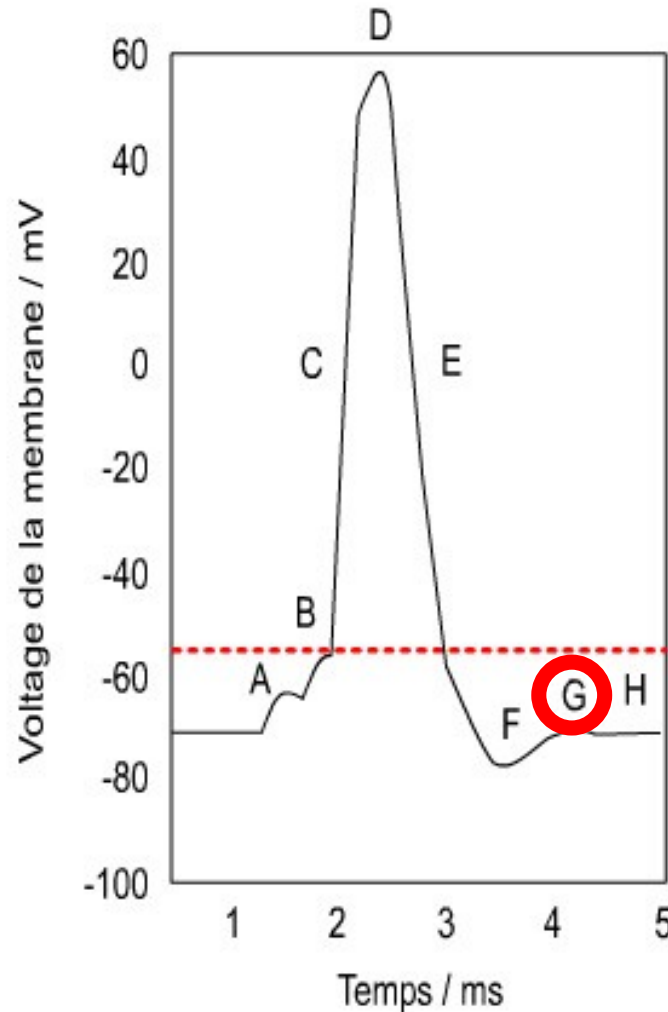
F) période réfractaire.



Physiologie du neurone

- Potentiel du neurone

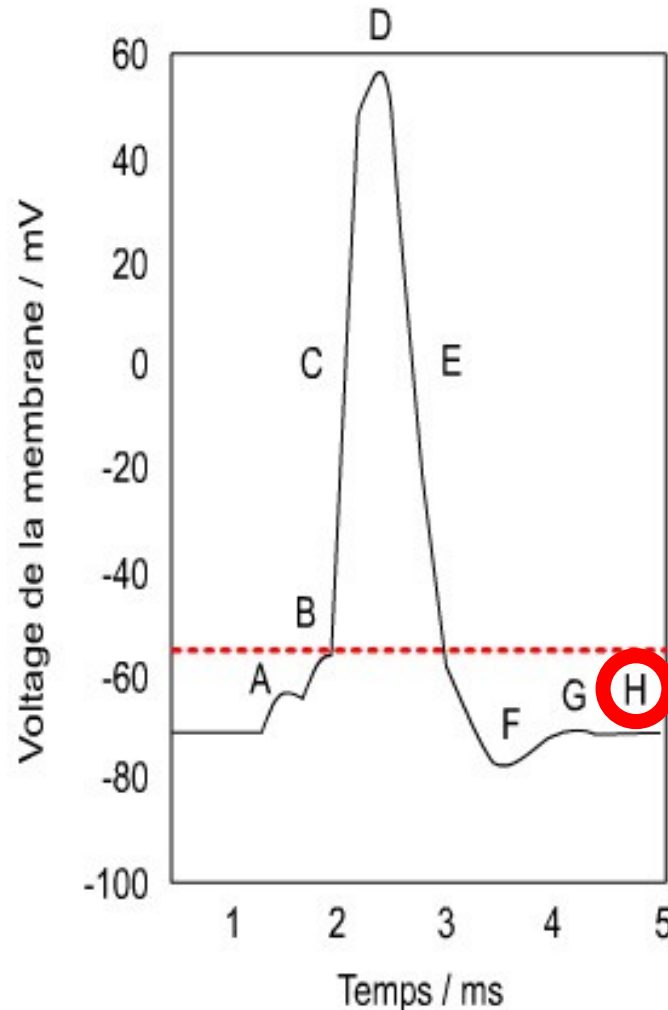
G) le potentiel de repos est restauré.



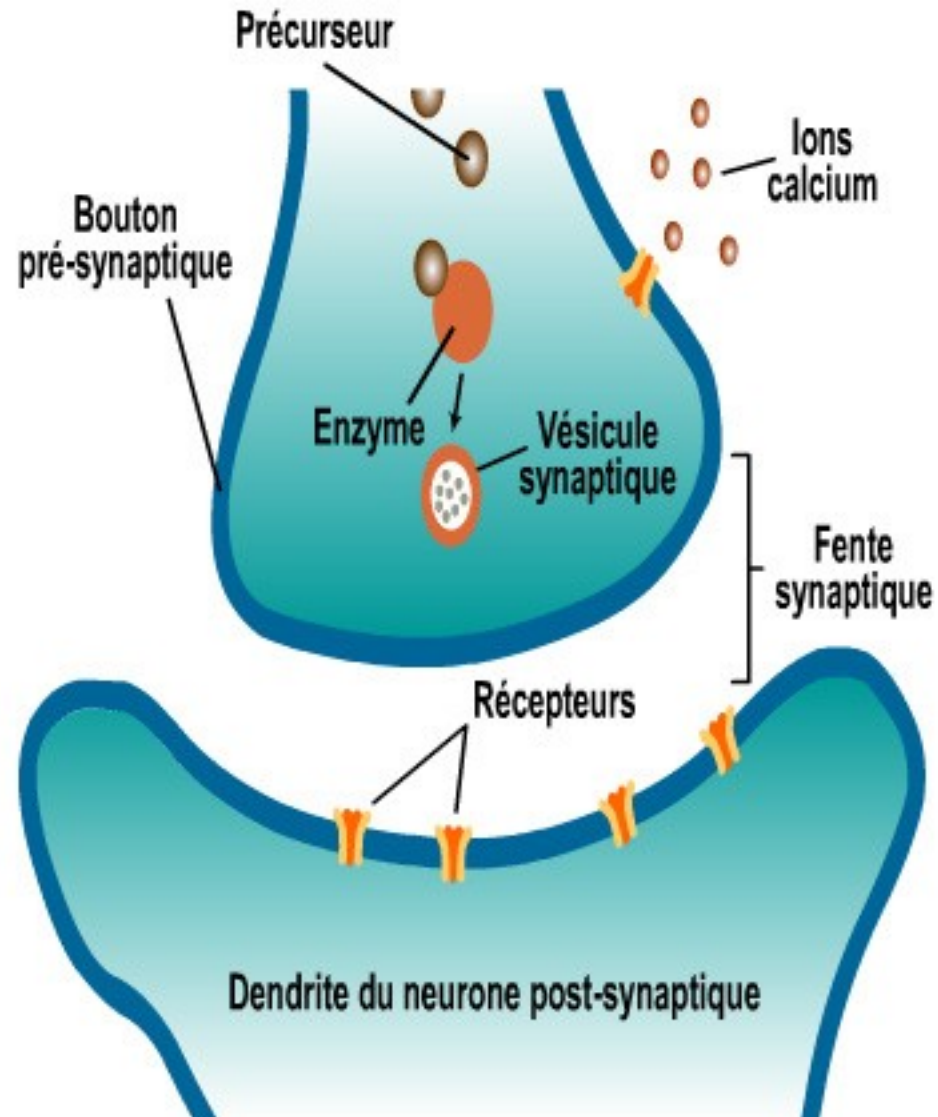
Physiologie du neurone

- Potentiel du neurone

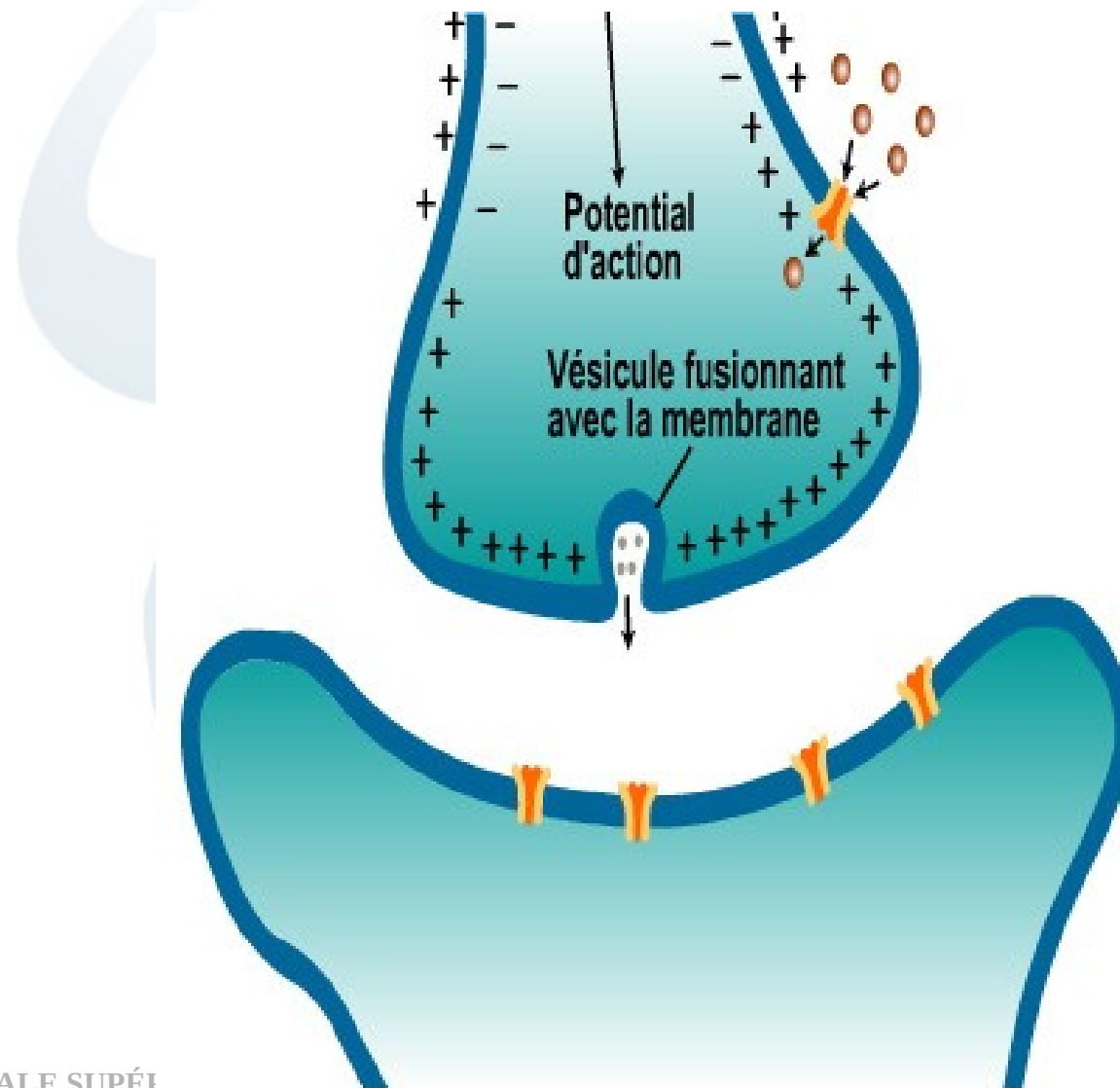
H) le sodium est pompé vers l'extérieur et le potassium vers l'intérieur, restaurant ainsi la distribution originale des ions.



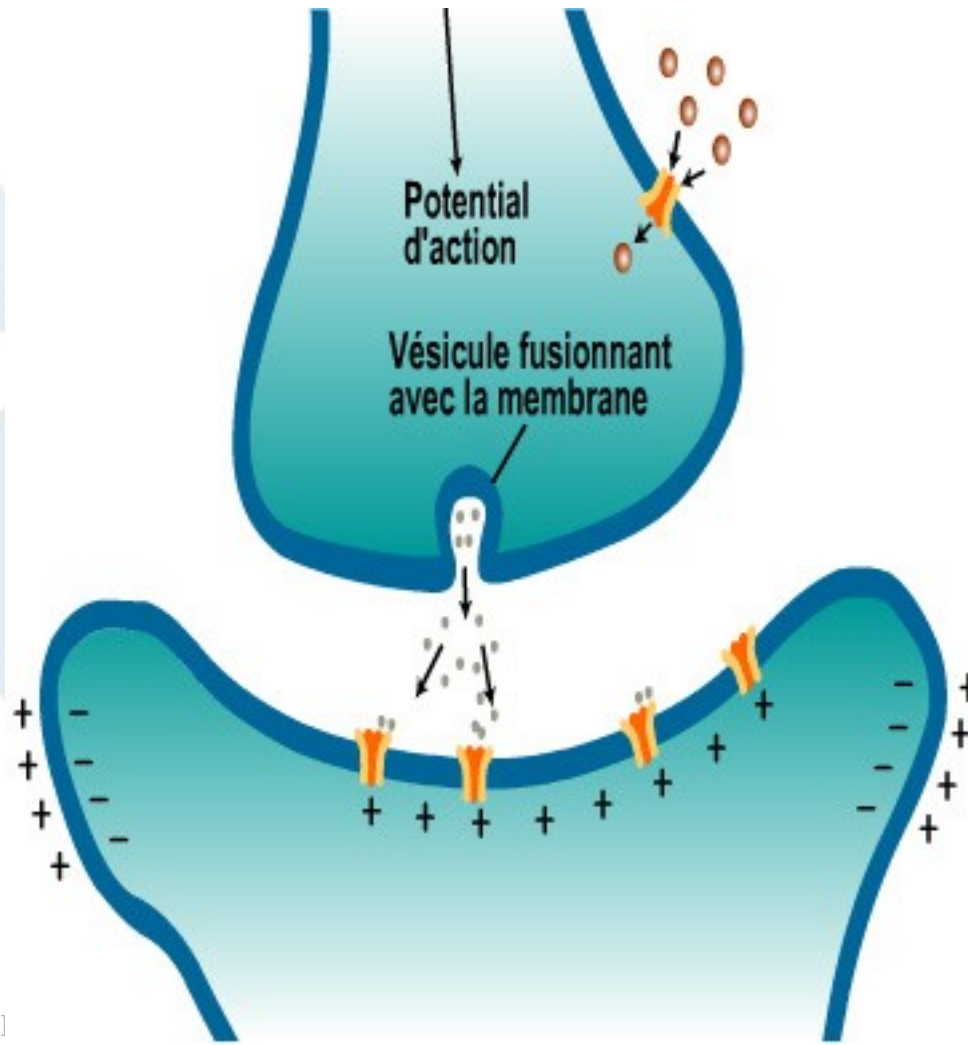
Détail de la synapse



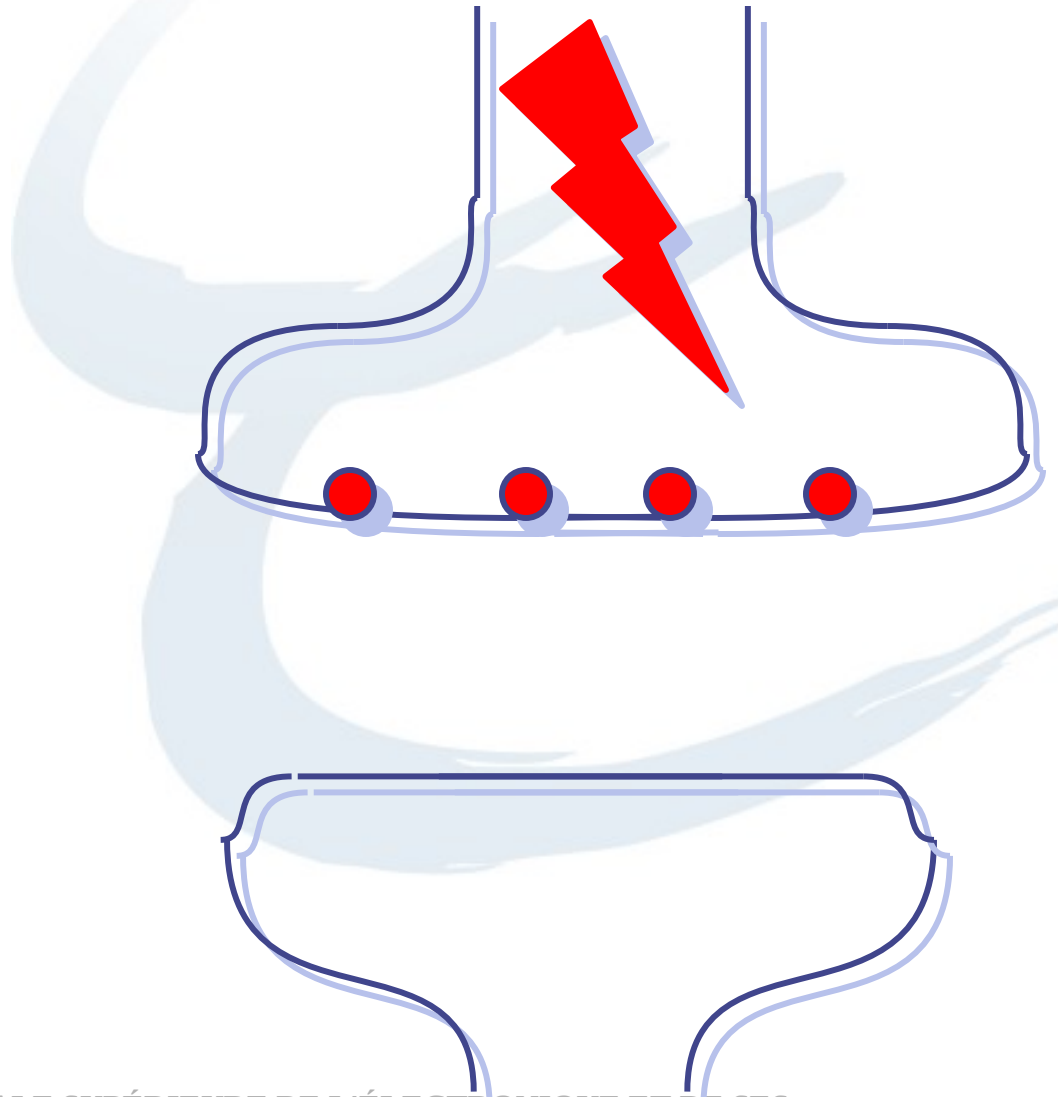
Détail de la synapse



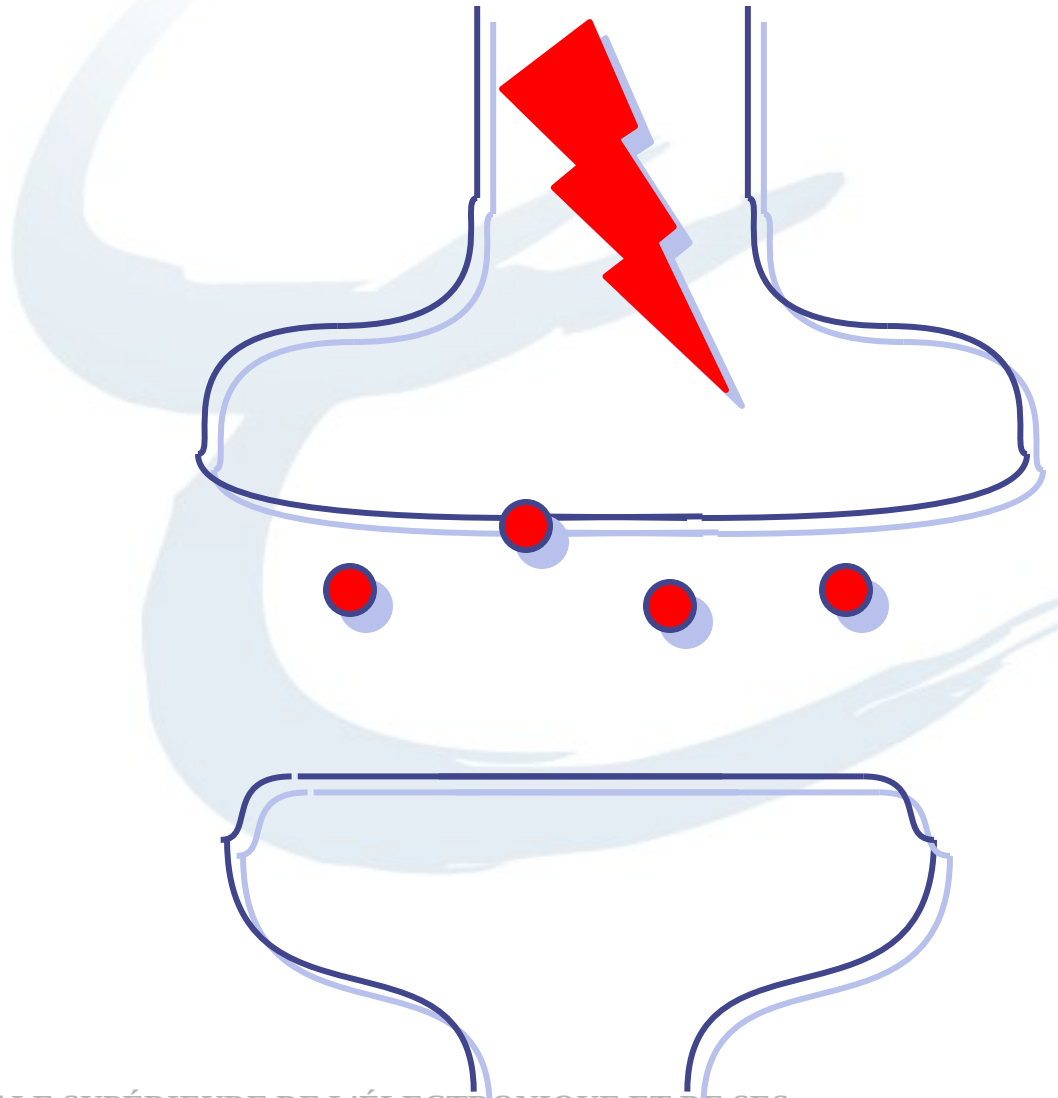
Détail de la synapse



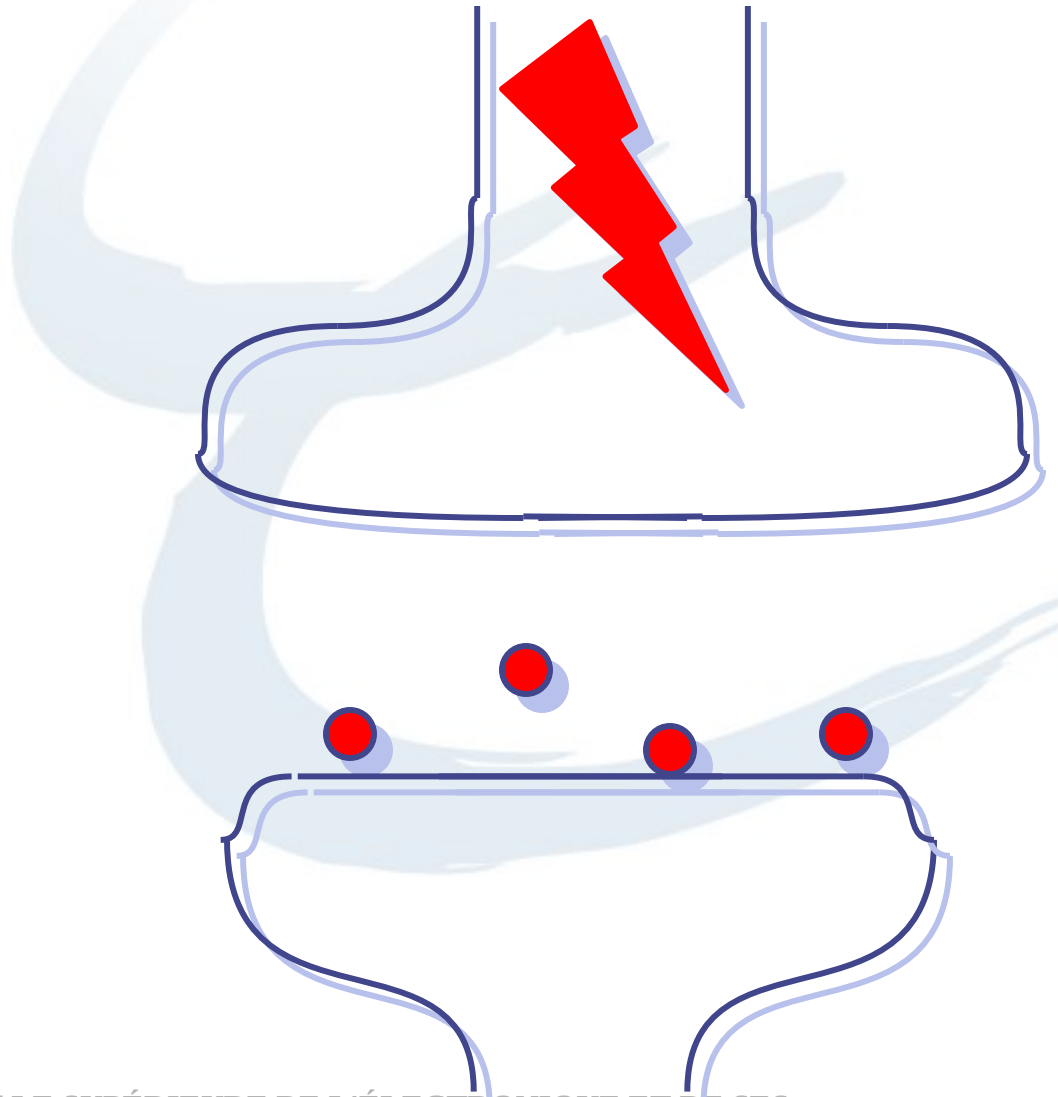
Détail de la synapse



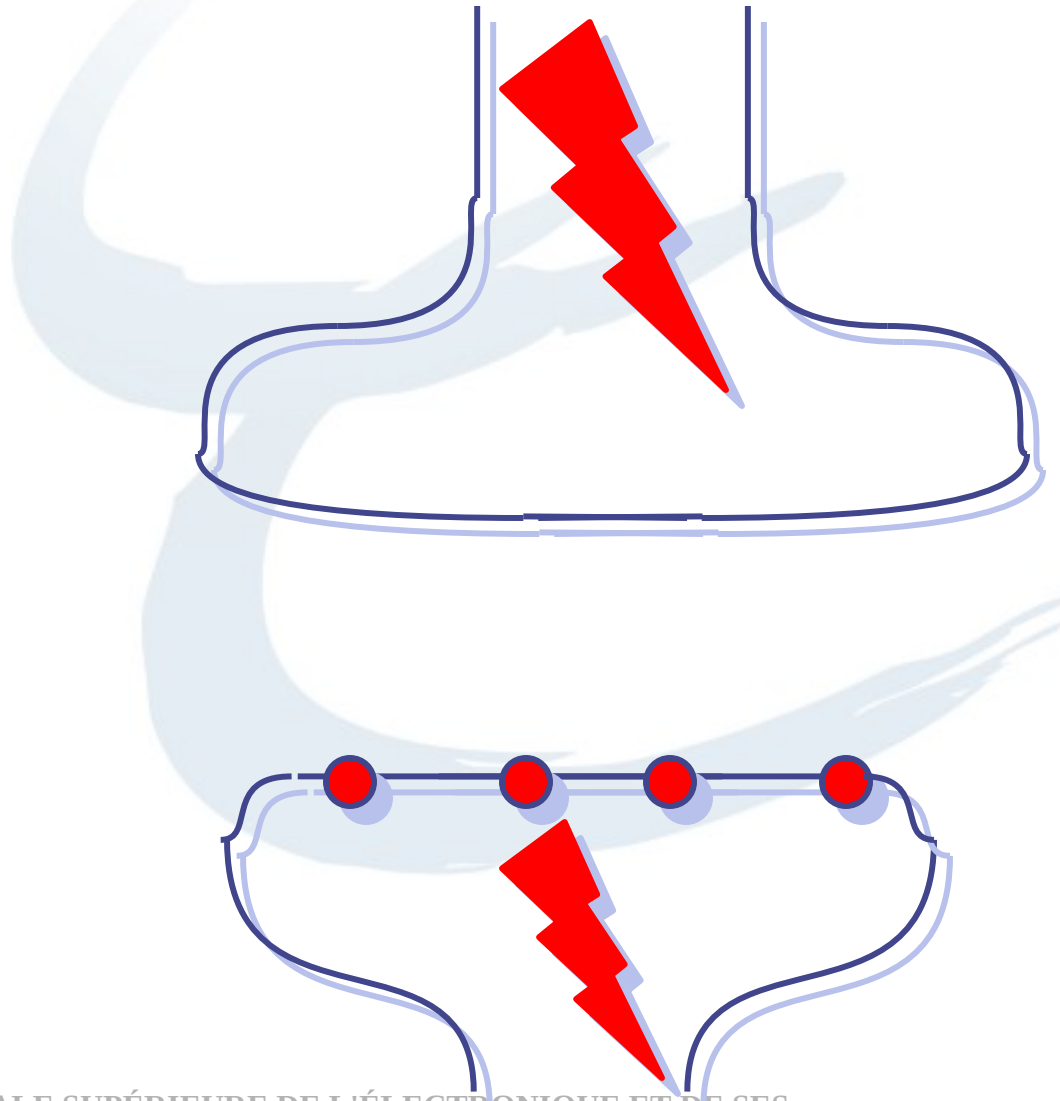
Détail de la synapse



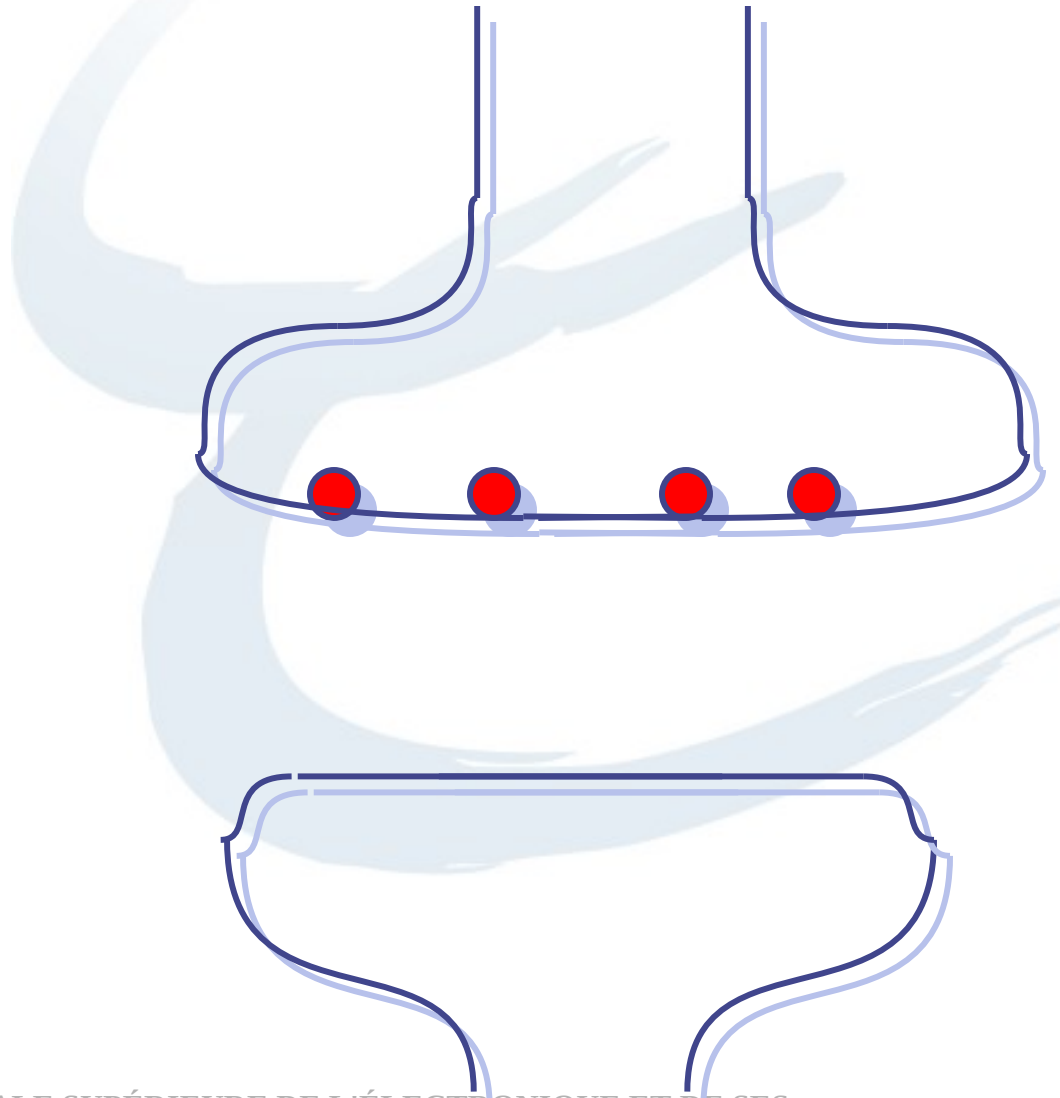
Détail de la synapse



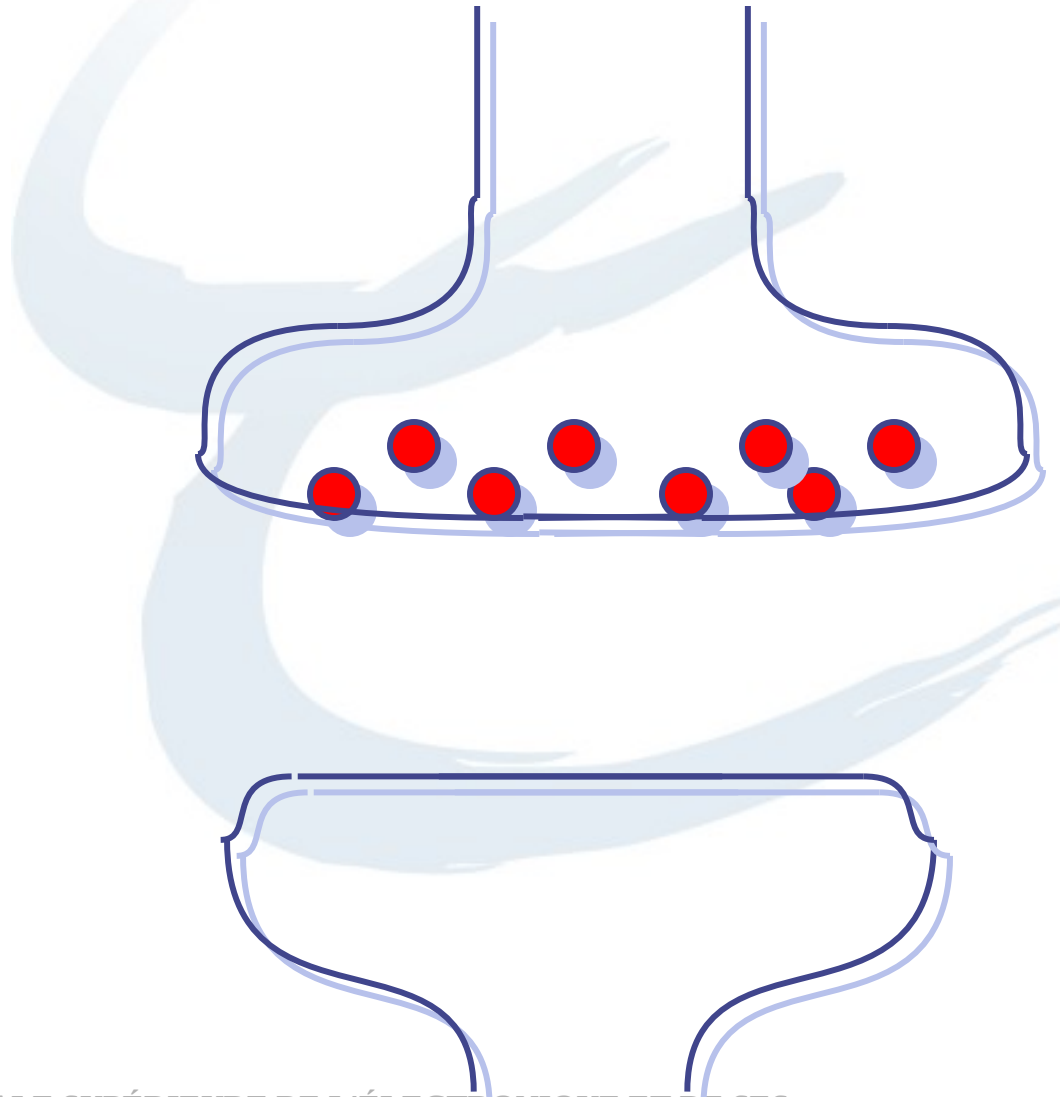
Détail de la synapse



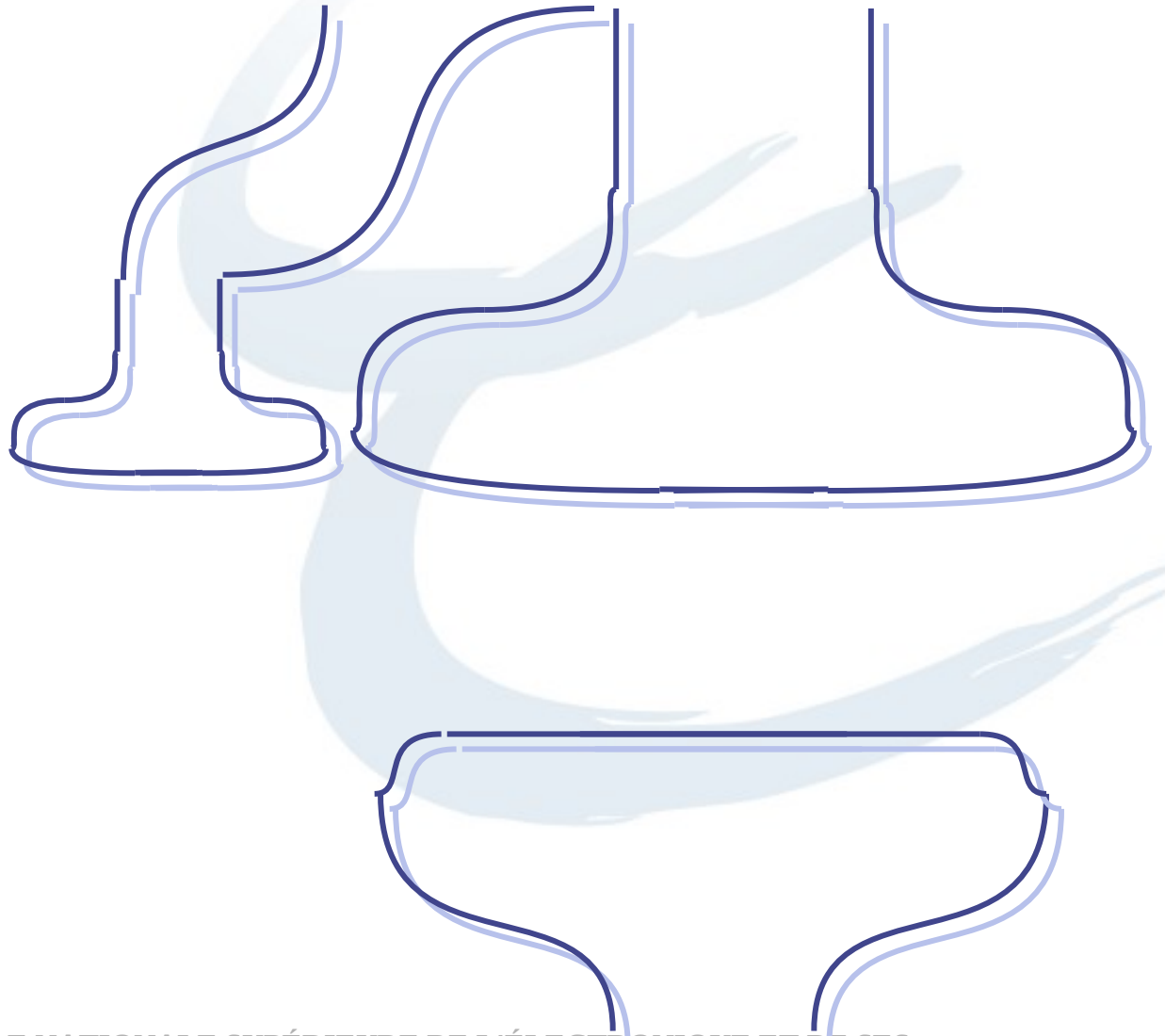
Apprentissage



Apprentissage



Apprentissage

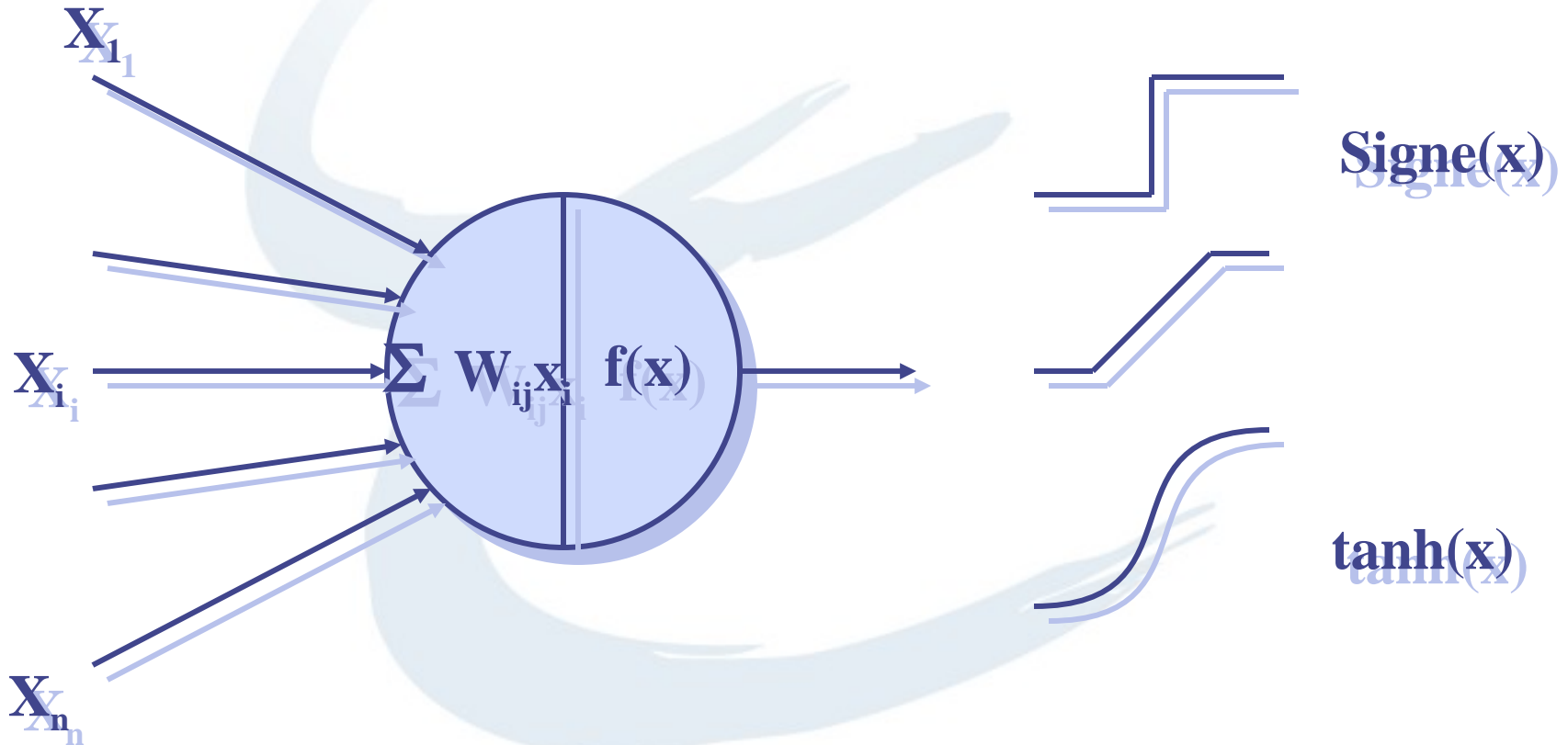


Apprentissage: psychologie

- Règle de Hebb
 - *Quand une cellule A excite par son axone une cellule B et que, de manière répétée et persistante, elle participe à la genèse d'une impulsion dans B, un processus de croissance ou un changement métabolique a lieu dans l'une ou dans les deux cellules, de telle sorte que l'efficacité de A à déclencher une impulsion dans B est, parmi les autres cellules qui ont cet effet, accrue*

Hebb 1949

Modèle du neurone formel



McCulloch & Pitts 1943

Modèle du neurone formel

- \mathbf{x}_i : entrées du neurone (observations)
- \mathbf{W}_i : poids du neurone (pondérations)
- $\sum \mathbf{W}_{ij}\mathbf{x}_i$: combinaison linéaire des entrées
- $f()$: fonction de mise en forme

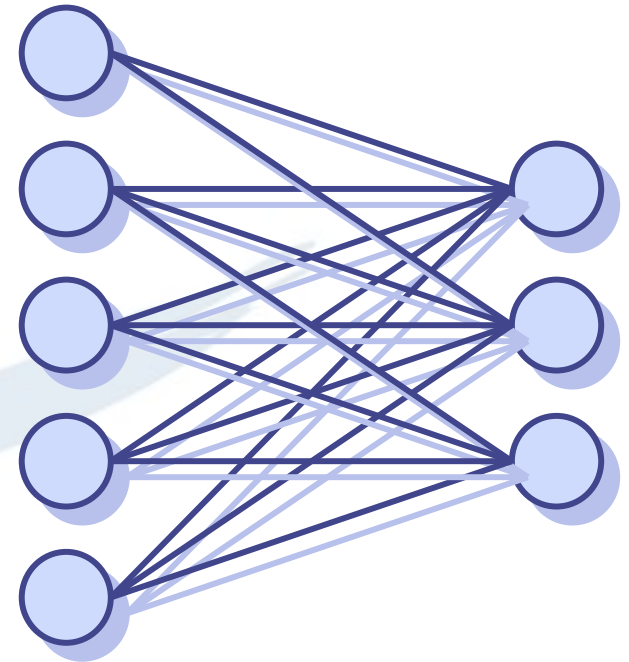
On optimise les \mathbf{W}_i de manière à ce que $f(\sum \mathbf{W}_{ij}\mathbf{x}_i)$ donne le résultat attendu

Fonction d'activation

- Échelon :
 - Si la somme pondérée des entrées dépasse un seuil, activation du neurone
- Linéaire seuillée :
 - Activation proportionnelle à la somme pondérée des entrées, seuillée entre +/- max (flou)
- Sigmoidale :
 - Activation non linéaire, dynamique rapide

Réseau ?

- Un réseau de neurone est un ensemble de neurones
 - Plusieurs sorties : chacune correspondant à une combinaison différente



Apprentissage du réseau

- Algorithme itératif

1. Initialisation le réseau avec des poids initiaux
2. Présentation au réseau d'un exemple
3. Calcul de la valeur de sortie
4. Mise à jour des poids en fonction de des entrées et de la valeur de sortie
5. Retour en 2.

Modèle de la règle de Hebb

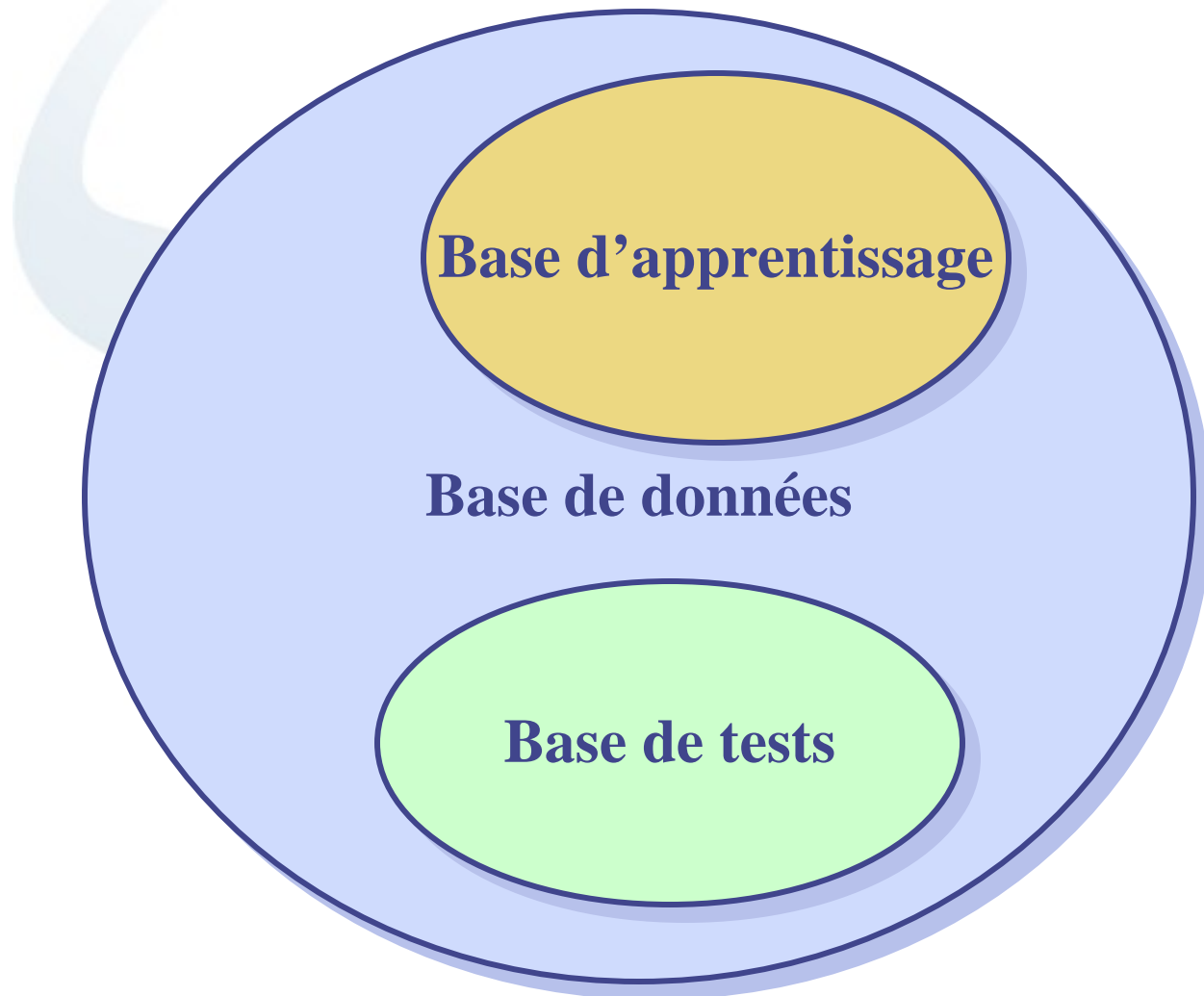
Taux d'apprentissage

$$\Delta W_{ij} = \epsilon \cdot X_i \cdot Y_j$$

Modèle de la règle de Hebb

- Augmentation des poids proportionnellement au produit de l'entrée avec la sortie
 - Si activation : augmentation du poids

Réseaux supervisés



Réseaux supervisés

1. On coupe l'ensemble d'apprentissage en 2 (train et val)
 - On entraîne sur train tant que l'erreur baisse sur val
2. On évalue les performances du réseau sur l'ensemble de test
3. On peut éventuellement répéter l'étape 1. un certain nombre de fois afin de garder le meilleur réseau (l'erreur la plus faible sur val)



Exemples de réseaux

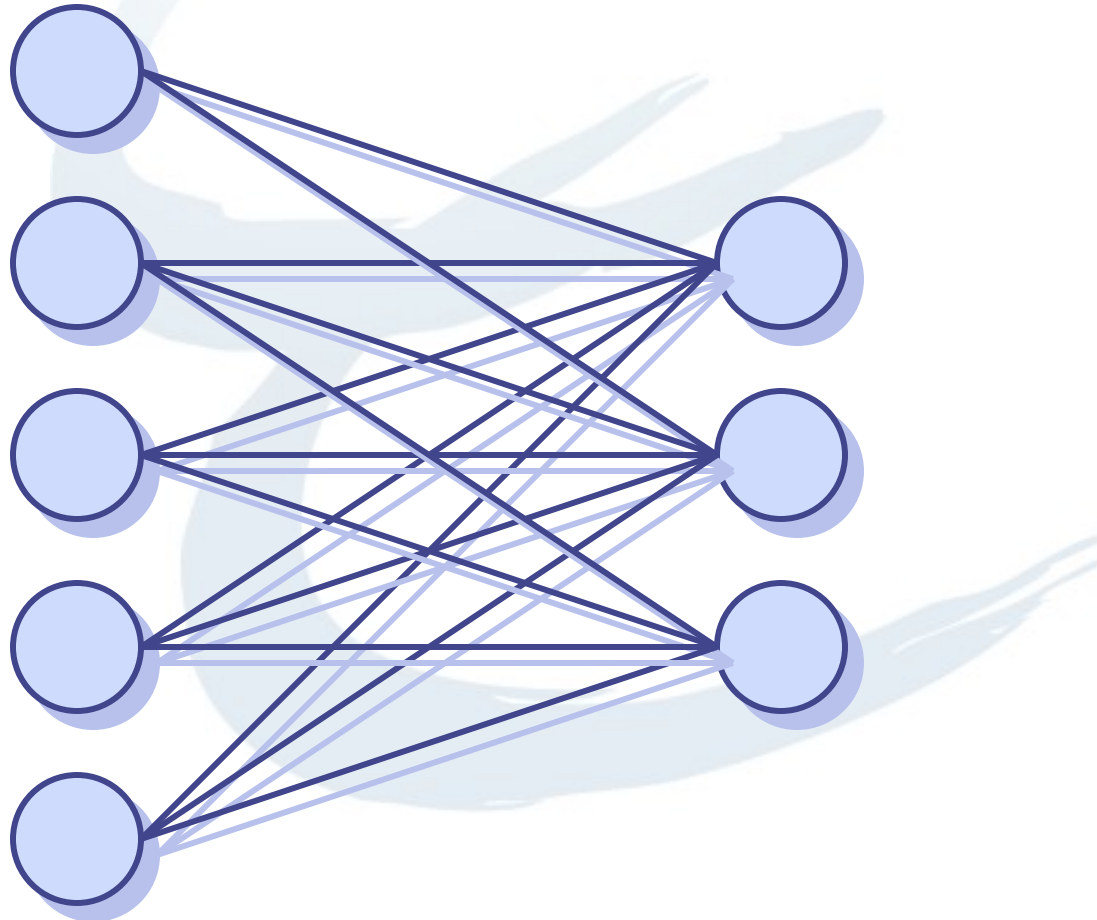


Mémoires associatives

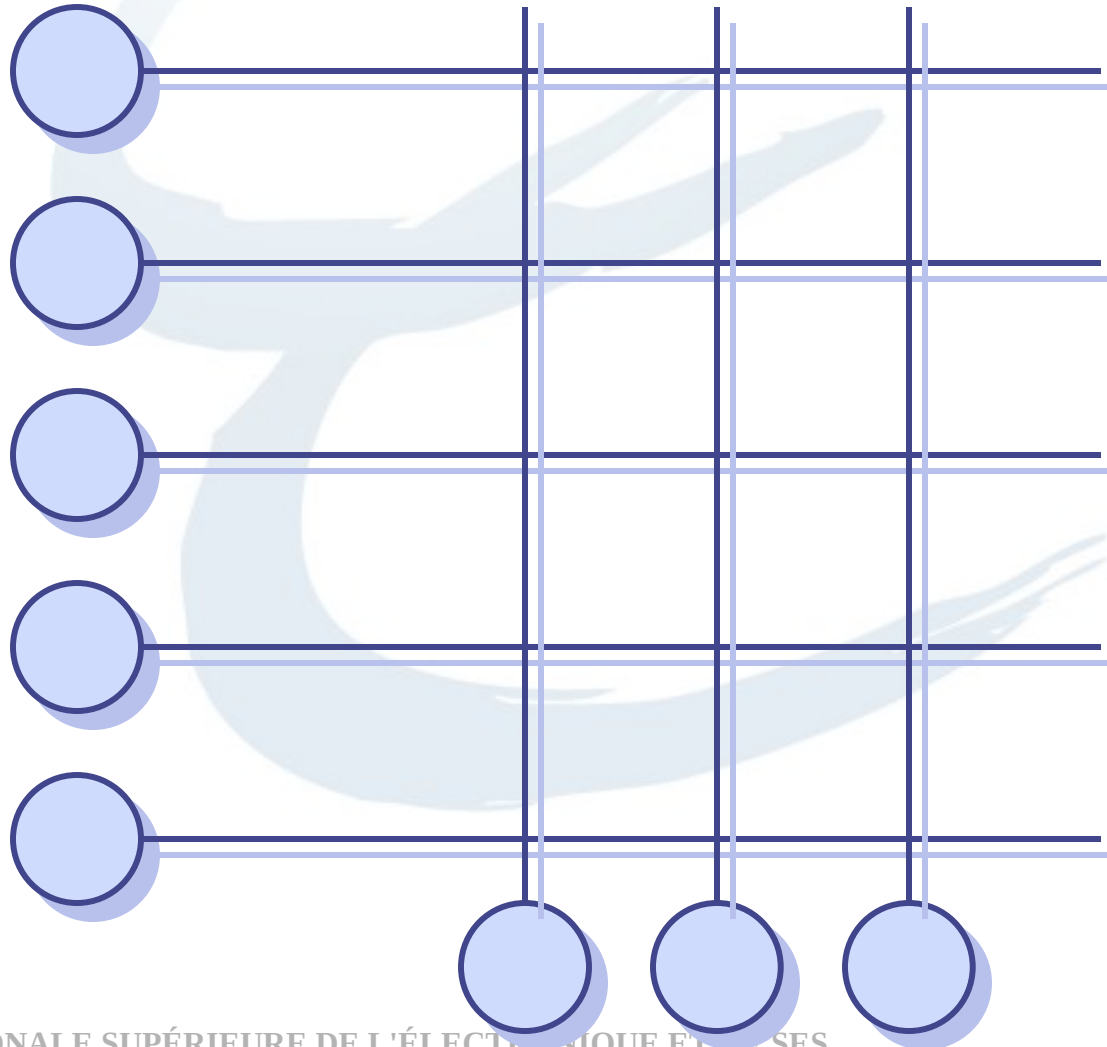
Les mémoires associatives

- Architecture
 - Réseaux entièrement interconnectés
- Apprentissage
 - Règle de Hebb
- Vecteur d'entrée de taille N
- Vecteur de sortie de taille P
 - Apprentissage hétéroassociatif
 - Si $P = N$, apprentissage autoassociatif

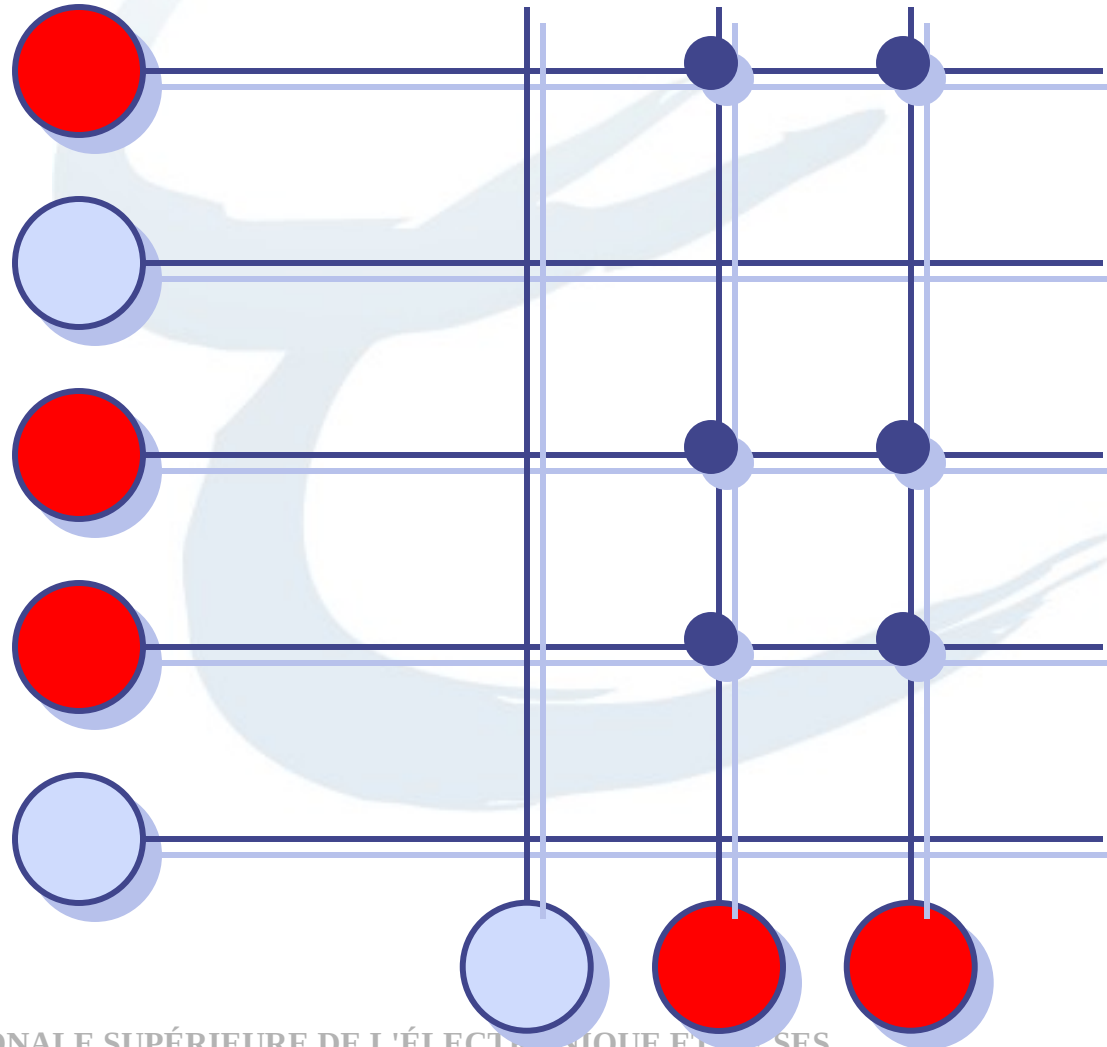
Les mémoires associatives



Les mémoires associatives



Les mémoires associatives



Les mémoires associatives

- Apprentissage
 - $W = \sum Y.X^T$
 - W est la matrice de poids, Y vecteur à apprendre, X vecteur de rappel
- Rappel
 - $Y' = W.X$
 - Si X dans la base d'apprentissage alors
 - $Y' = Y$

Mémoires associatives

- Exercices
 - Apprendre l'association
 - $[1\ 0\ 0\ 1]$ et $[0\ 1\ 0]$
 - $[0\ 1\ 1\ 0]$ et $[1\ 0\ 0]$
 - Faire le rappel de $[1\ 0\ 0\ 0]$
 - Mettre $[1\ 1\ 0\ 0]$ en entrée

Les mémoires associatives

- Limitations
 - Les vecteurs doivent être **orthogonaux**
 - Nombre d'apprentissages limité
- Applications
 - Complétion de motifs

Les mémoires associatives





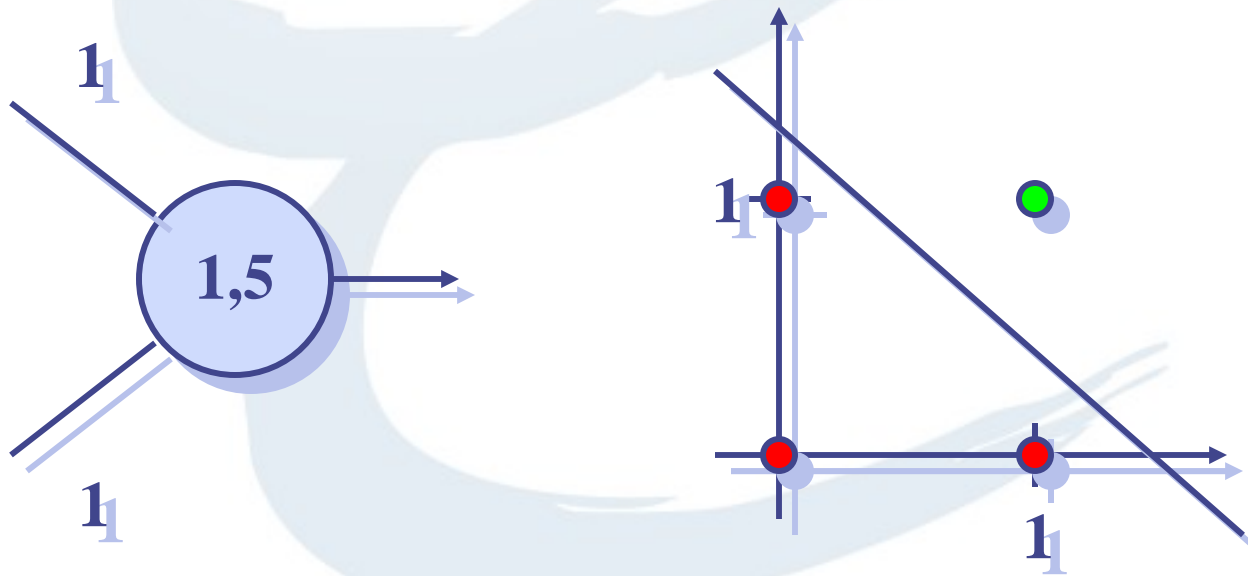
Perceptron

Le Perceptron

- Architecture
 - Neurones formels
 - Sortie seuillée
- Règle d'apprentissage
 - $\Delta W = (Y_d - Y) \cdot X_i$
- Convergence
 - Séparation en hyperplans

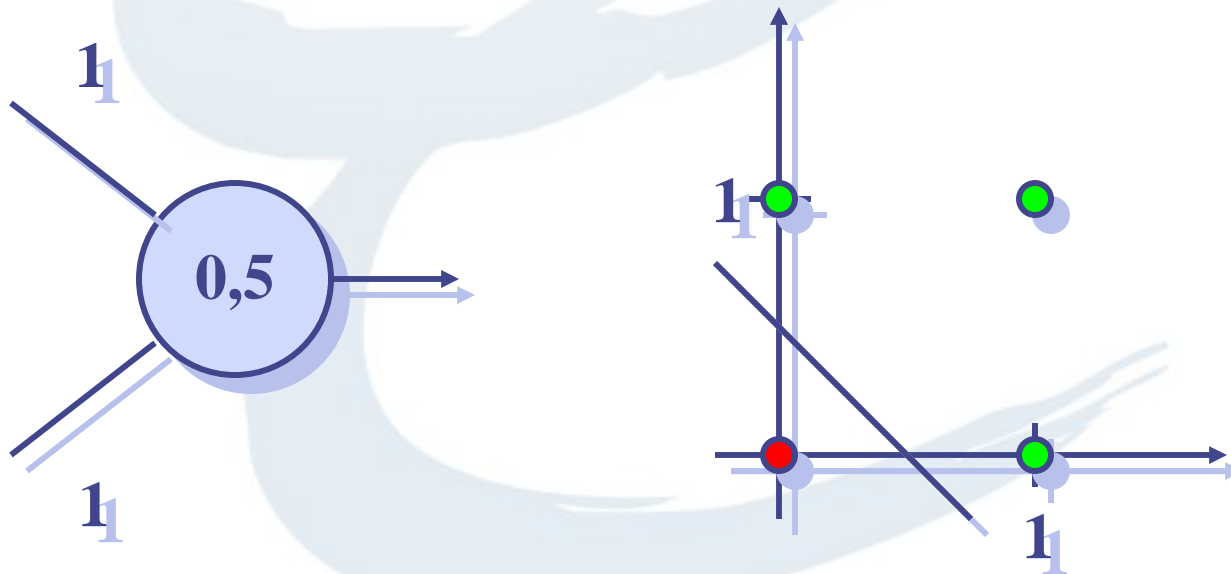
Le Perceptron : exemples

- Le ET logique



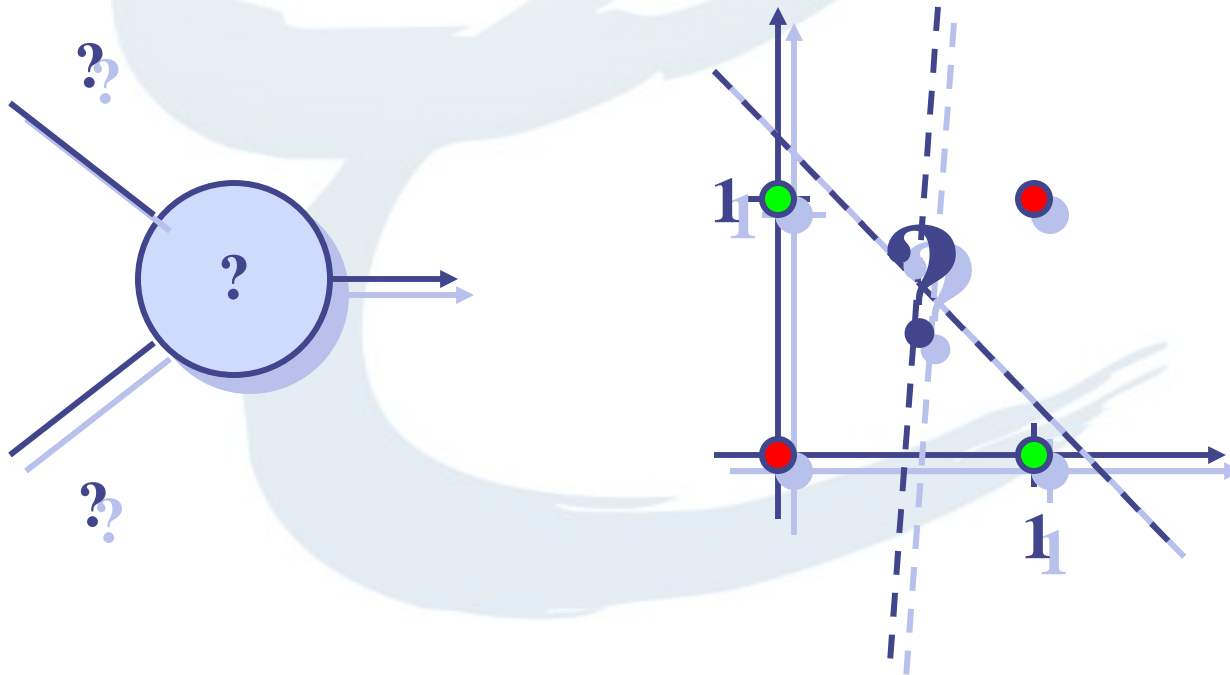
Le Perceptron : exemples

- Le OU logique



Le Perceptron : exemples

- Le XOR (ou exclusif)

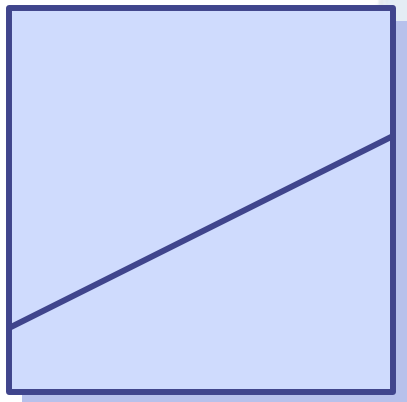


Le Perceptron : Limitations

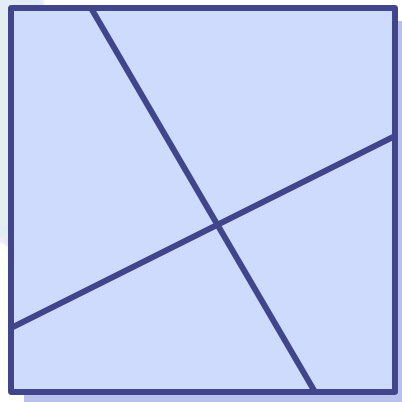
- **Problèmes linéairement séparables**
- **Latitude**
 - Augmenter le nombre de couches

Le Perceptron : Limitations

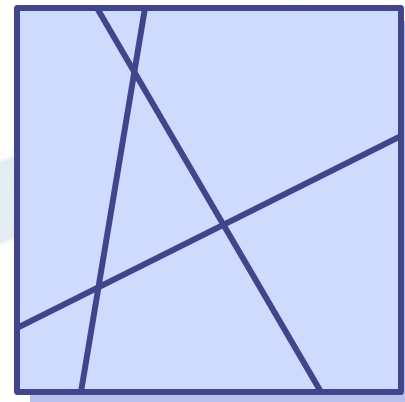
- **Problèmes linéairement séparables**
- **Latitute**
 - Augmenter le nombre de couches



1 couche



2 couches



3 couches

...

Le Perceptron : Exercice

- Résoudre le problème du XOR
- Utiliser 2 couches de neurones

Le Perceptron multi-couche

- Apprentissage
 - Règle de rétropropagation du gradient
- Critère
 - On veut minimiser l'erreur quadratique de classification en sortie du réseau
 - $E = \frac{1}{2} \cdot (Y_d - Y)^2$
- Comment doit-on corriger les poids ?

Le Perceptron multi-couche

- Cas à 1 couche
 - $dE/dW_{ij} = d/dW_{ij} \cdot 1/2 \cdot (Y_d - Y)^2$
 - Avec $Y = f(\sum W_{ij} \cdot X_i)$
- $dE/dW_{ij} = -dY/dW_{ij} = (Y_d - Y) \cdot X_i \cdot f'(\sum W_{ij} \cdot X_i)$
- D'où la règle $\Delta W_{ij} = \alpha \cdot (Y_d - Y) \cdot X_i$
 - Règle du LMS (Widrow et Hoff)

Le Perceptron multi-couche

- Cas à n couches
 - On peut montrer de la même manière

$$\Delta w_{kh}^j = -\alpha \cdot \delta_k^j \cdot y_h^{j-1}$$

$$\delta_k^j = \begin{cases} y_k^j - y_{d_k} & \text{pour la dernière couche} \\ \left[\sum_{i \in \text{couche } j+1} \delta_i^{j+1} \cdot w_{ik}^{j+1} \right] \cdot \sigma'(p_k^j) & \text{pour les autres couches} \end{cases}$$

Le Perceptron multi-couche

- Propriétés
 - 2 couches cachées
 - Approximateur universel
- Pratiquement:
 - Lissage de l'apprentissage
 - Rétropropagation avec momentum

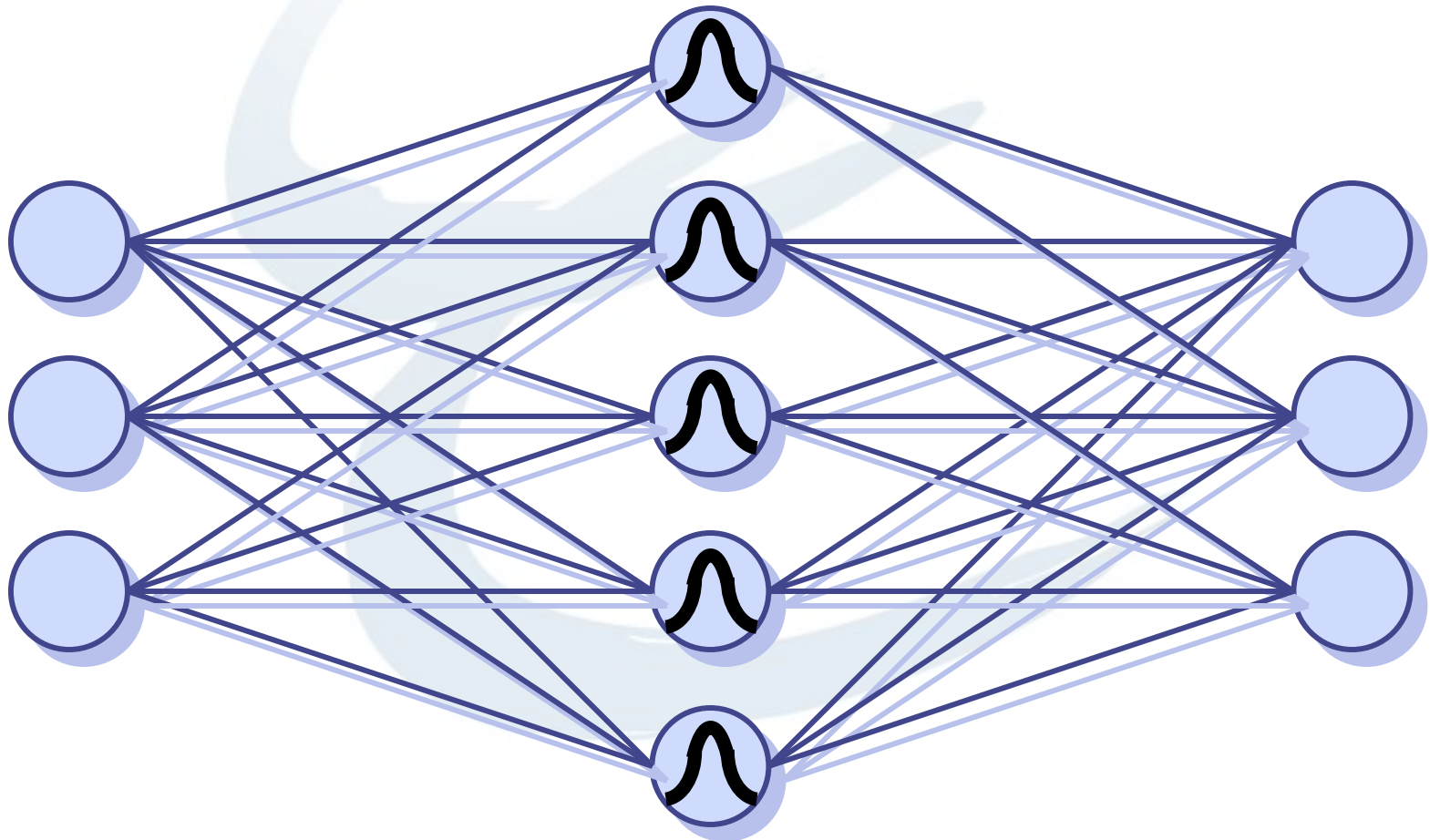
Réseaux de Fonctions à Bases Radiales

- Définition
 - C'est un MLP dont la fonction de sortie des neurones de la couche d'entrée est une RBF (triangle, gaussien...) multiplié par la probabilité a priori de la classe
- Intérêt
 - Estimer la densité de probabilité des données

Réseaux de Fonctions à Bases Radiales

- Apprentissage
 - Chaque noyau est centré sur une donnée d'apprentissage
 - L'apprentissage est un LMS en sortie de couche 1
- Problème
 - Taille des noyaux ?
 - Si trop de données dans la base d'apprentissage ?
- Solution
 - Plus proches voisins
 - Auto-organisation (Kohonen...)

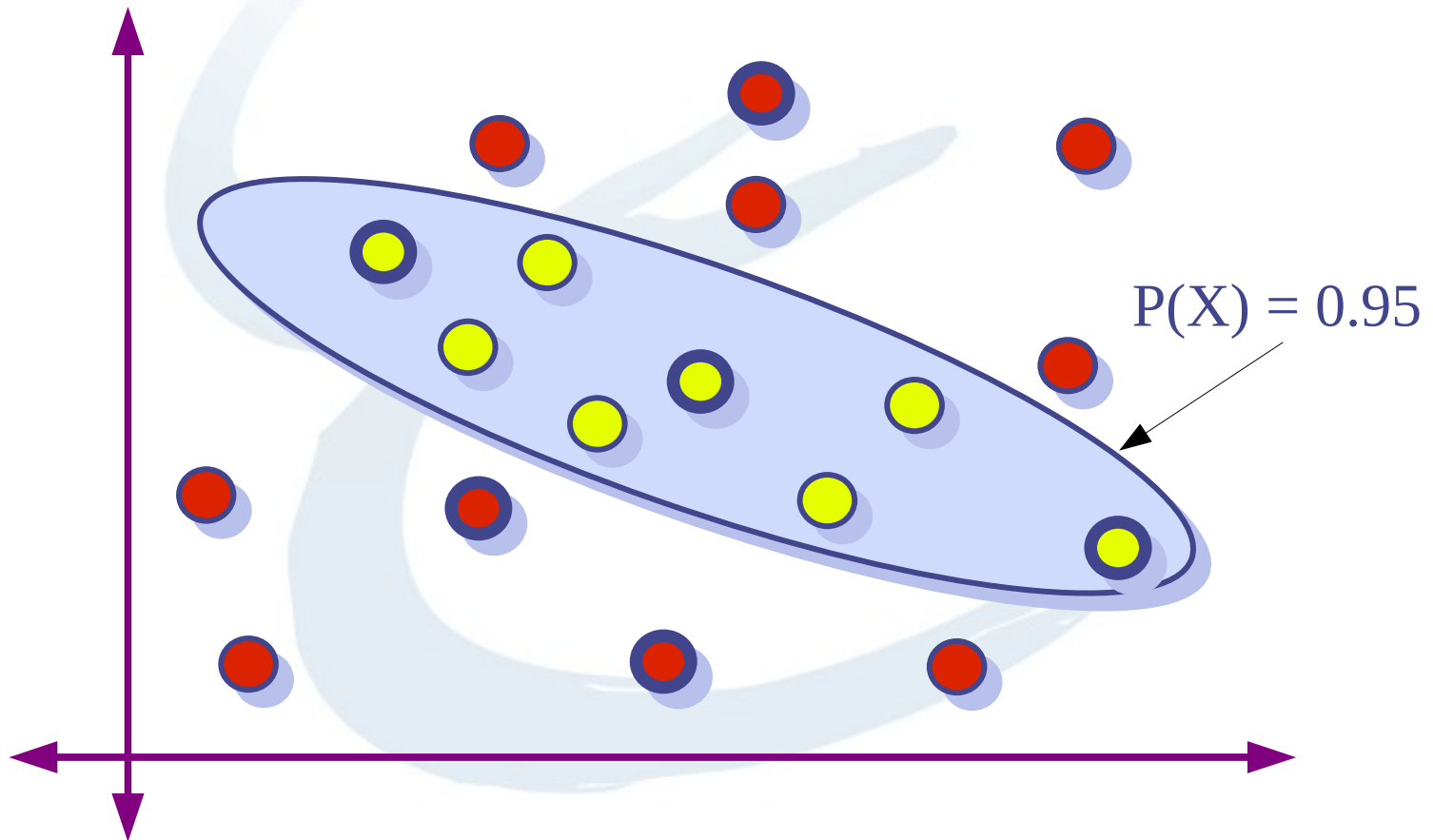
Réseaux de Fonctions à Bases Radiales



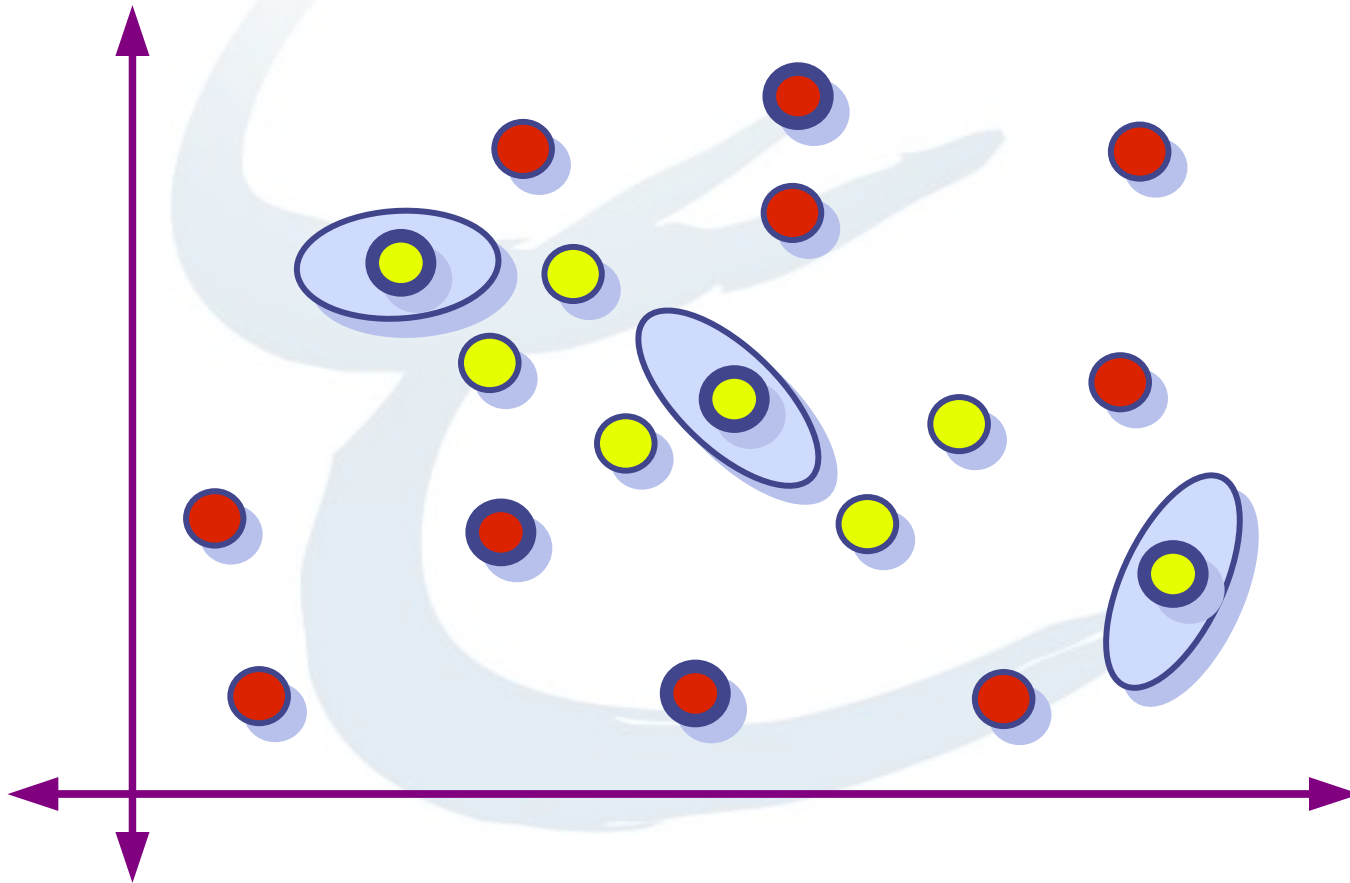
Réseaux de Fonctions à Bases Radiales

- Un réseaux RBF utilisé dans le cadre du contrôle est très comparable avec un contrôleur flou
 - Les entrées RBF sont semblables aux entrées floutées
 - Le réseaux agit sur les sorties en calculant une mixture non-linéaires des entrées
- Différences
 - Il n'y a pas « d'expert » mais le réseau apprend les règles de production

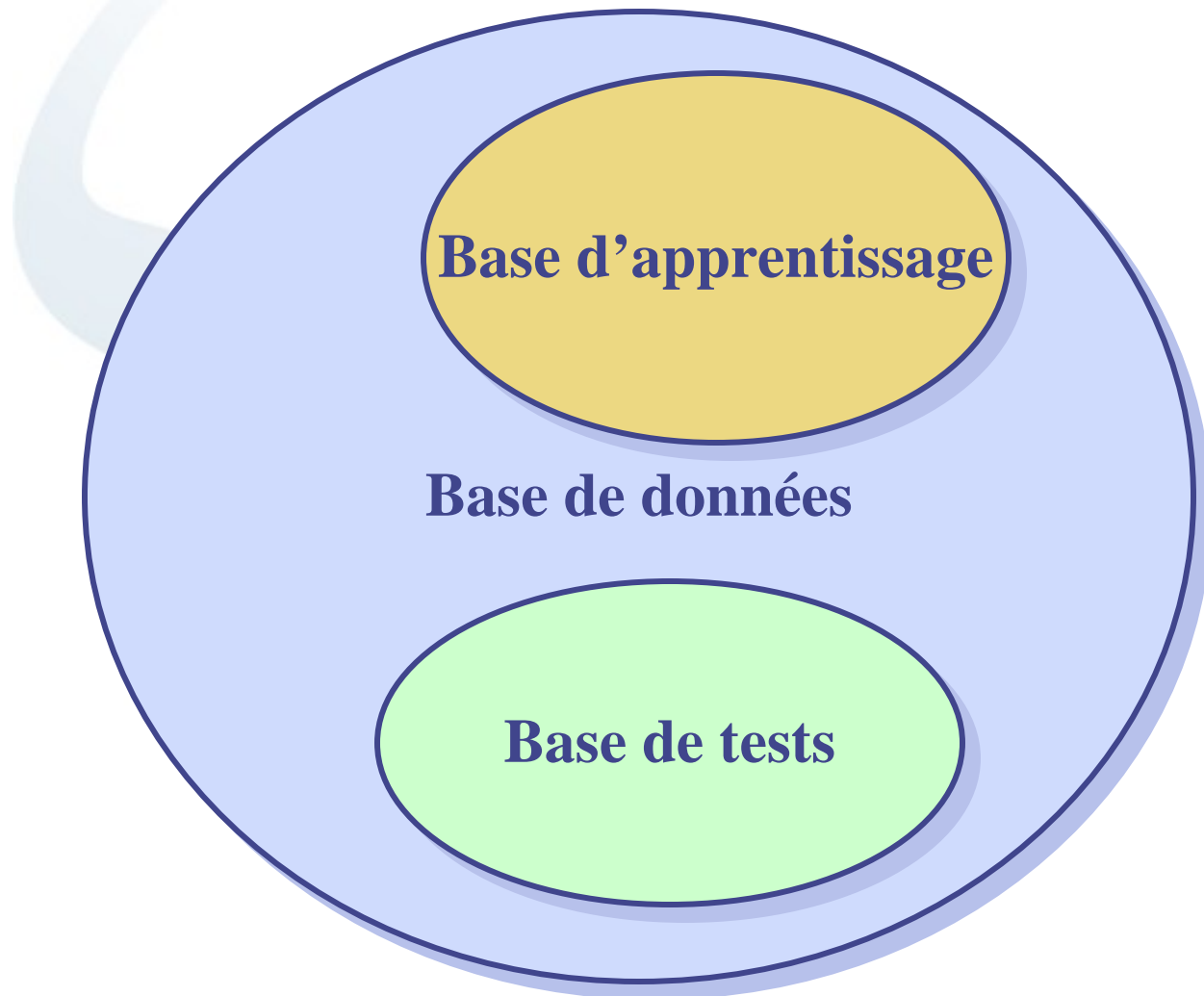
Réseaux de Fonctions à Bases Radiales



Sur-apprentissage



Réseaux supervisés



Réseaux supervisés

1. On coupe l'ensemble d'apprentissage en 2 (train et val)

- On entraîne sur train tant que l'erreur baisse sur val

Si l'erreur sur val remonte => sur-apprentissage

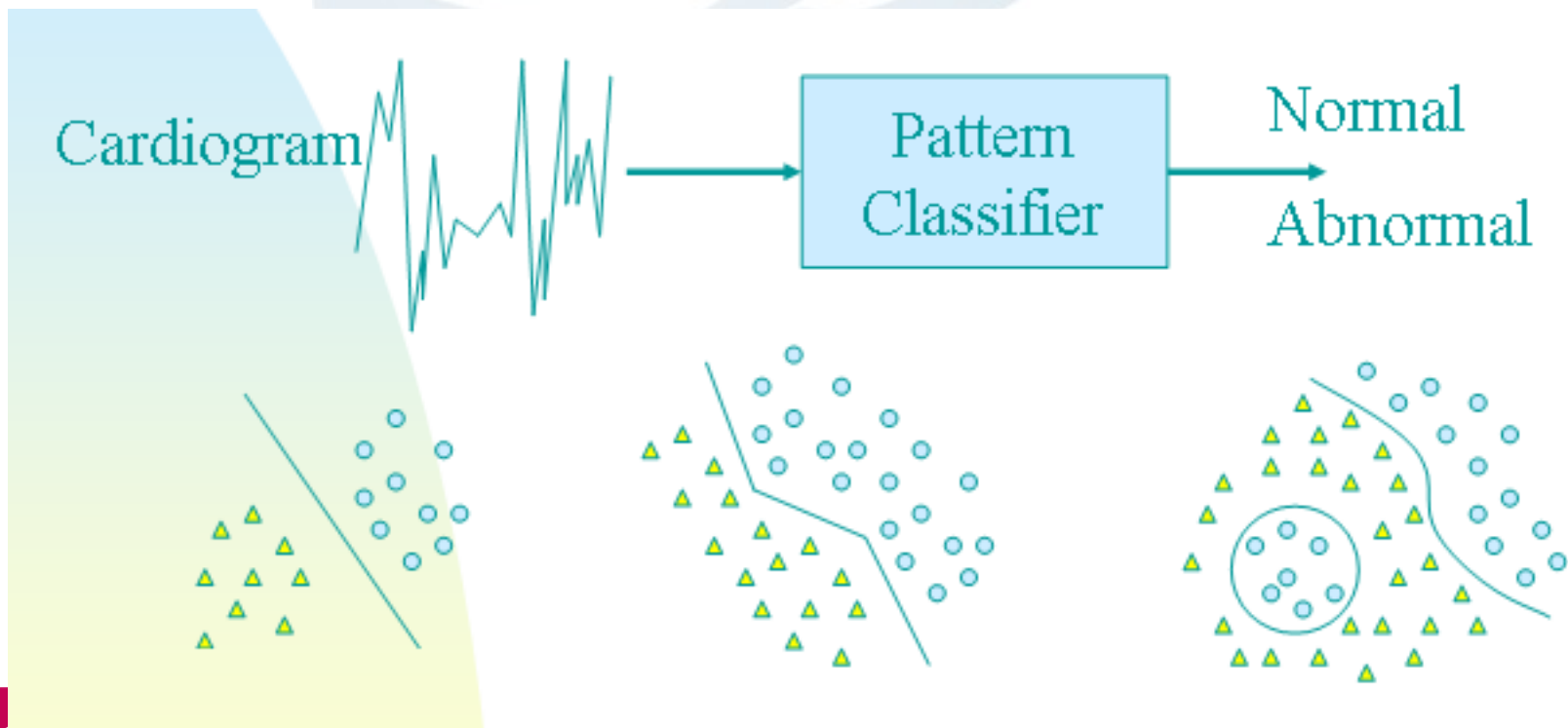
1. On évalue les performances du réseau sur l'ensemble de test

Réseaux de neurones

Applications
les différents domaines
détail sur le contrôle

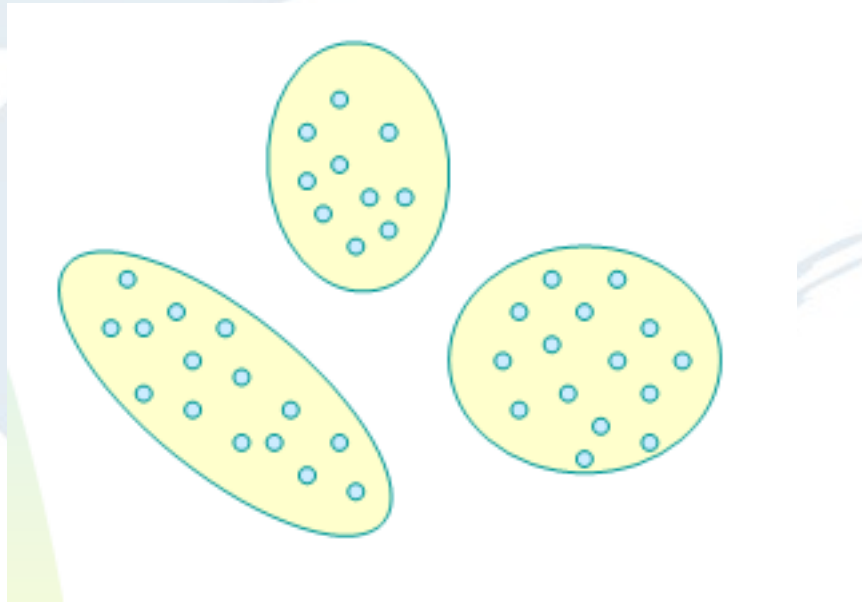
Problèmes traités

- Classification de signaux



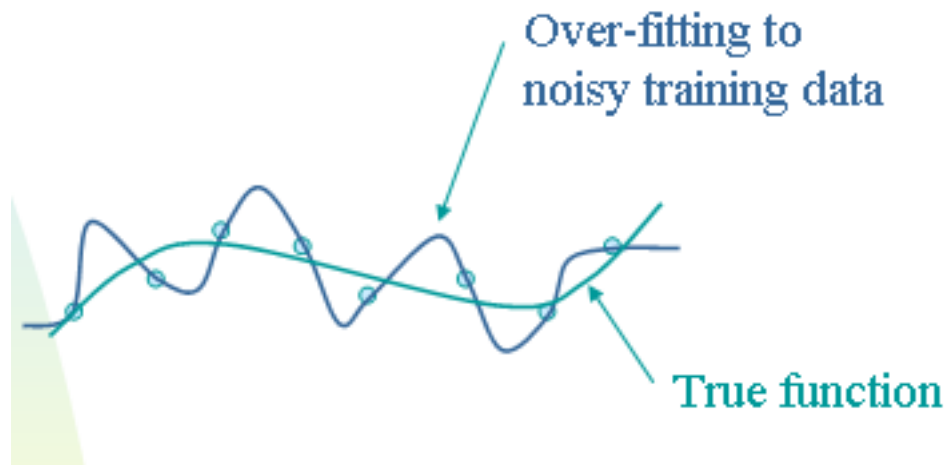
Problèmes traités

- Clustering, catégorisation



Problèmes traités

- Approximation de fonctions



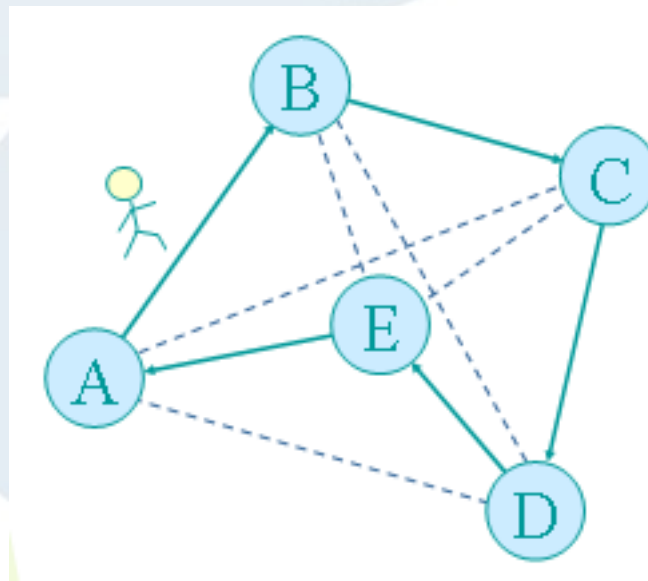
Problèmes traités

- Prédiction/Prévision



Problèmes traités

- Optimisation



Domaines d'applications

- **MATHEMATIQUES ET CALCUL**
 - itérations discrètes
 - réseaux systoliques
 - calcul parallèle
 - calculabilité sur réseaux et algorithmique
 - optimisation combinatoire
 - approximation de fonction
 - modèles probabilistes
 - systèmes dynamiques
 - codage
 - prédiction

Domaines d'applications

- **TRAITEMENT DE DONNEES**
 - classification
 - Auto-organisation et catégorisation
 - décision, diagnostic
 - linguistique, traitement de la parole
 - séparation de sources en traitement du signal
 - affectation de ressources, routage
 - fusion de données
 - décision spatiale multi-critères

Domaines d'applications

- **APPRENTISSAGE ET SYSTEMES AUTO-ADAPTATIFS**
 - apprentissage dans les réseaux algorithmique
 - génétique et évolutionniste
 - apprentissage dans les modèles physiques
 - mémoires associatives
 - apprentissage de séquences temporelles

Domaines d'applications

- **RECONNAISSANCE DE FORMES ET VISION**
 - traitement d'images
 - invariance de formes
 - reconnaissance de caractères et d'écritures manuscrites
 - imagerie médicale

Domaines d'applications

- **COGNITION ARTIFICIELLE**
 - modélisation des processus cognitifs
 - intelligence artificielle
 - systèmes experts
 - systèmes hybrides
 - intelligence artificielle distribuée, systèmes multi-agents
 - extraction de connaissance
 - réseaux sémantiques
 - interfaces homme-machine
 - robotique mobile
 - robotique autonome et robotique sous-marine

Domaines d'applications

- **SIMULATION DE PHENOMENES PHYSIQUES**
 - mécanique des fluides, gaz sur réseaux
 - phénomènes de combustion et d'explosion
 - processus de réaction/diffusion
 - étude des milieux désordonnés, verres de spins

Domaines d'applications

- **PHENOMENES D'INTERACTION, DE PROPAGATION ET DE CROISSANCE**
 - processus d'émergence
 - épidémiologie, développement, immunologie
 - morphogenèse et croissance
 - percolation
 - interaction entre gènes

Domaines d'applications

- **NEUROSCIENCES**
 - neurosciences théoriques
 - modélisation en neurophysiologie
 - modélisation en biologie
 - réseaux neuromimétiques
 - simulation du fonctionnement de systèmes naturels

Domaines d'applications

- **MODELISATION EN PSYCHOLOGIE ET EN SOCIOLOGIE**
 - étude de la mémoire et de l'apprentissage
 - apprentissage et reconnaissance en didactique
 - comportements collectifs

Domaines d'applications

- **TECHNOLOGIE INFORMATIQUES**
 - machines parallèles
 - machines réseaux
 - ASIC neuronaux
 - microcircuits dédiés
 - traitements analogiques
 - procédés optiques

Domaines d'applications

- **INDUSTRIE ET AUTRES**
 - identification et conduite de procédés
 - commande et contrôle
 - sidérurgie, économie, finance, droit, imprimerie, cartographie, téléphonie

Applications privilégiées

- Les traitements non linéaires
- L'approximation de fonctions à valeurs continues
- L'identification de lois non analytiques
- L'identification et le contrôle de procédés
- Le contrôle flou (cf. logique floue)
- L'apprentissage, en particulier dans le domaine de l'apprentissage non supervisé
- l'optimisation dynamique contextuelle

Applications privilégiées

- **La reconnaissance de formes**
 - En particulier la reconnaissance de caractères manuscrits où ils sont parmi les meilleurs outils la séparation de sources
 - En 1984, les réseaux neuronaux ont apporté la première solution à ce problème

Exemple : Classification acoustique des phénomènes de cavitation

- **Contexte de résolution du problème :**
 - les hypothèses de statistique classique ne s'appliquent pas.
 - les relations entre les données sont fortement non-linéaires
 - on ne dispose pas de modèle théorique satisfaisant.
 - on a besoin d'adaptativité parce que de nouvelles données peuvent arriver.
 - il existe un expert capable de traiter le problème, ici une personne ayant appris à reconnaître le type de cavitation à l'oreille.

⇒ **Utilisation d'un MLP avec rétropropagation**

Classification acoustique des phénomènes de cavitation

- Résultats :
 - le taux de réponse correcte atteint 95,4 % sur des signaux non appris par le réseau mais correspondant à des conditions similaires aux signaux appris.
 - A titre de comparaison, une personne ayant eu une bonne écoute préalable de ces signaux à atteint 93 % de réponses correctes.
 - Méthode : totalement expérimentale
 - plusieurs types de pré-traitement des données ont été essayés et couplés avec des réseaux, puis l'on a choisi le réseau permettant d'obtenir la meilleure généralisation

Exemple :

reconnaissance de codes postaux

- Principe
 - Pour chaque objet postal, on doit
 - soit identifier le code
 - soit indiquer qu'elle ne peut pas l'identifier, et donc faire appel à un opérateur humain
 - **!** il est plus onéreux de rectifier une erreur de tri commise par une machine que de faire lire un code postal par un opérateur !
 - le critère de performance utilisé est le suivant :
 - pour un taux d'erreur maximum fixé (par exemple 1 %) sur les codes identifiés, quelle est la fraction du courrier qui devra être traitée par un opérateur ?

Exemple : reconnaissance de codes postaux

- Motivations
 - Intérêt économique
 - Dès 1990 : disponibilité de BdD

Reconnaissance de codes postaux

- Types de données

65473 60198 68544
70065 70117 19032^{ZIP} 96720
27260 61820 19559
74136 19137 63101
20878 60521 38002
48640-2398 20907 14868

Reconnaissance de codes postaux

- Nécessité de pré-traitements
 - Segmentation des données
 - détection de contours
 - rehaussement de contraste
 - la *normalisation* (pour que tous les traitements portent sur des chiffres de même taille)
- Une fois les pré-traitements réalisés
 - Le classifieur peut être simple

Reconnaissance de codes postaux

- Architectures
 - « *LeNet* » [LeCun,1991]

10 Neurones de sortie

Connectivité complète

30 Neurones cachés

Connectivité complète

12x16

Neurones cachés

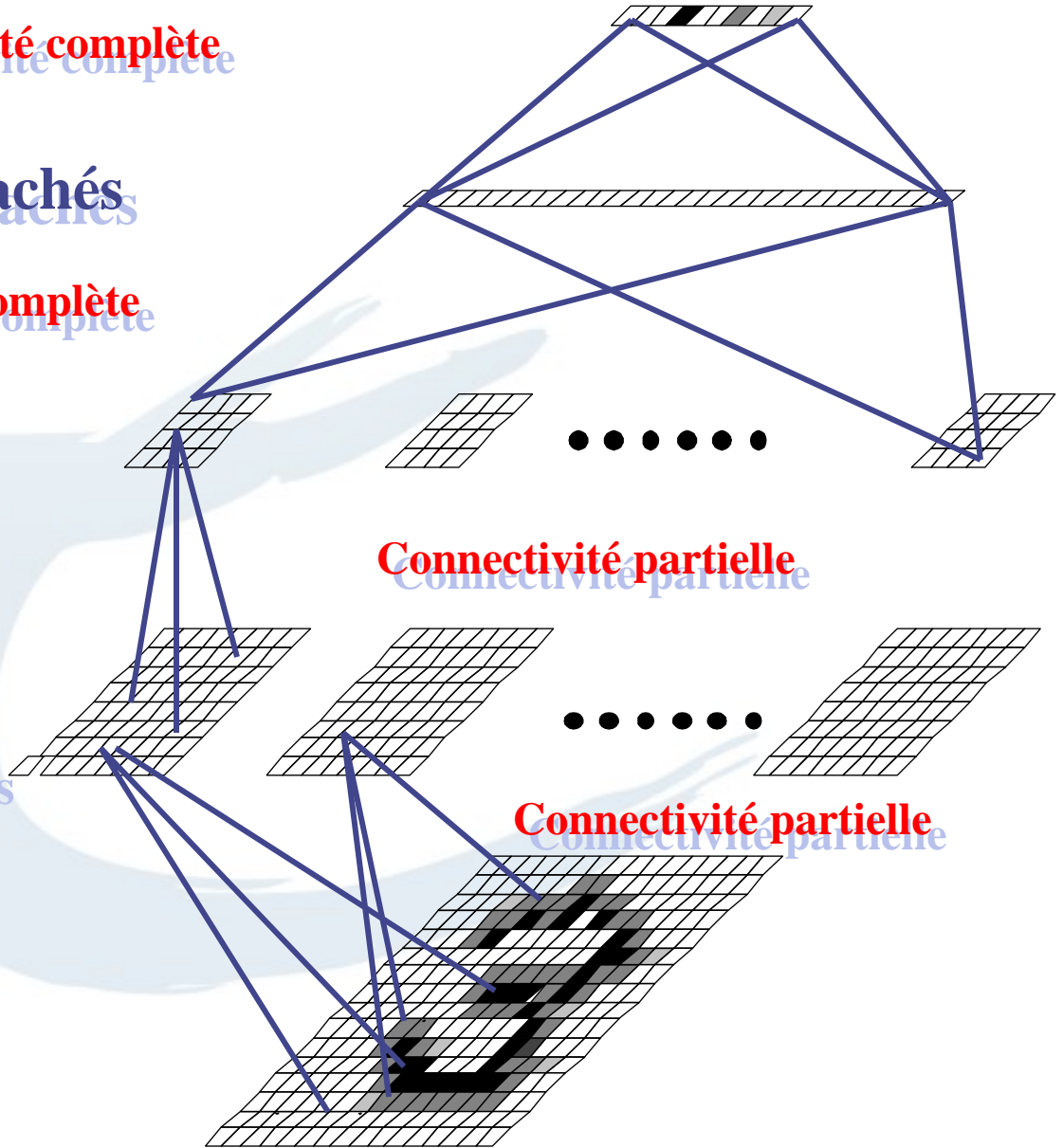
Connectivité partielle

12x64 Neurones cachés

Carte de caractéristiques

Connectivité partielle

256 entrées



Reconnaissance de codes postaux

- Architectures
 - « *LeNet* » [LeCun,1991]
 - [Knerr,1992]
 - Pré-traitements figés
 - détection de contours
 - normalisation
 - 4 cartes de caractéristiques de 64 éléments
 - les dix classes sont séparées deux à deux
 - 45 classifieurs différents
 - Apprentissage effectué séparément
 - Tous les exemples de l'ensemble d'apprentissage sont linéairement séparables deux à deux $C_2^{10} = \frac{10!}{8! \times 2!} = \frac{10 \times 9}{2}$
 - Chacun des 45 classifieurs est donc en fait constitué d'un seul neurone.

Reconnaissance de codes postaux

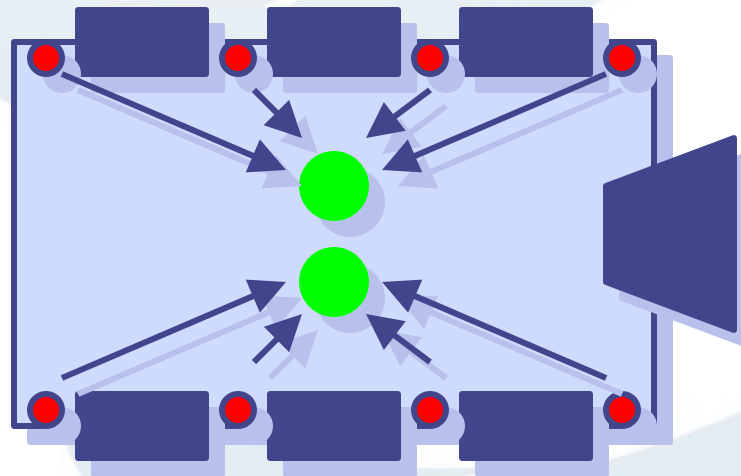
- Résultats

	Chiffres Bien classés	Taux de rejet	Exemples mal classés
LeNet	70,9%	28,1%	1%
Knerr	90,3%	8,7%	1%

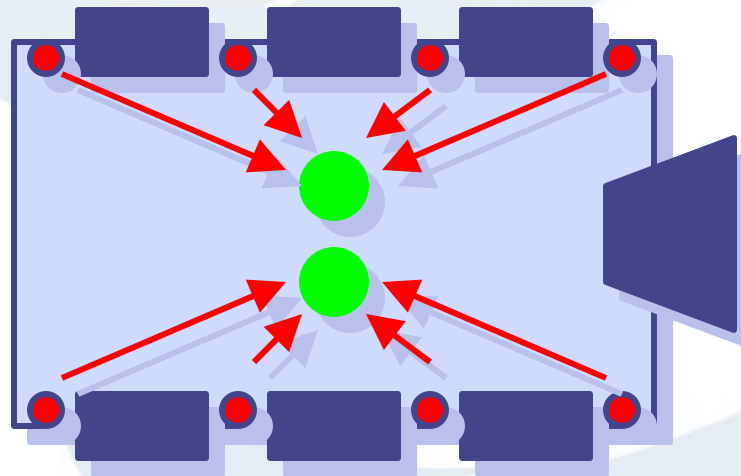
Véhicules de Braitenberg

- Principe
 - Utiliser directement les capteurs pour produire des comportements
 - Approche bottom-up

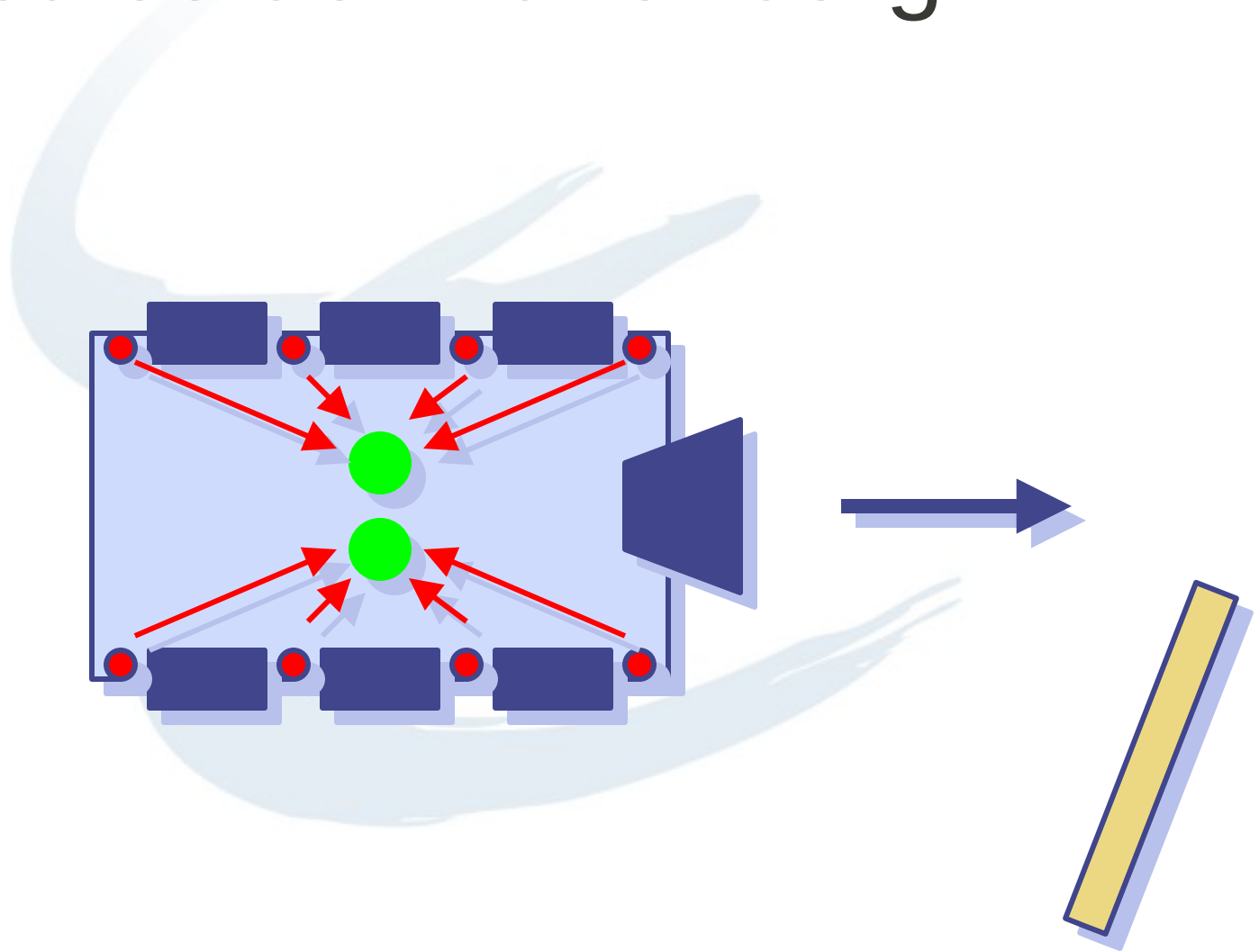
Véhicules de Braitenberg



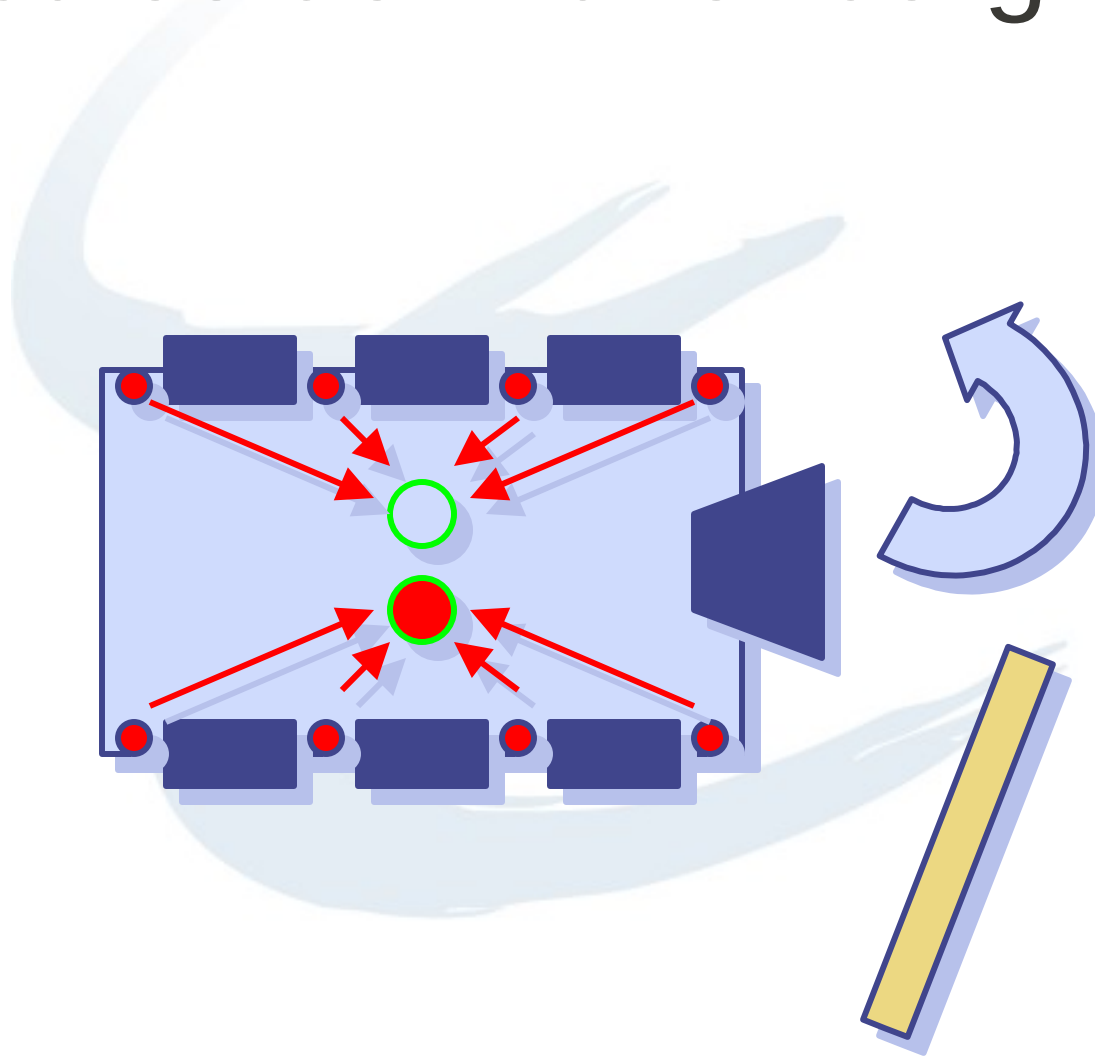
Véhicules de Braitenberg : évitement d'obstacles



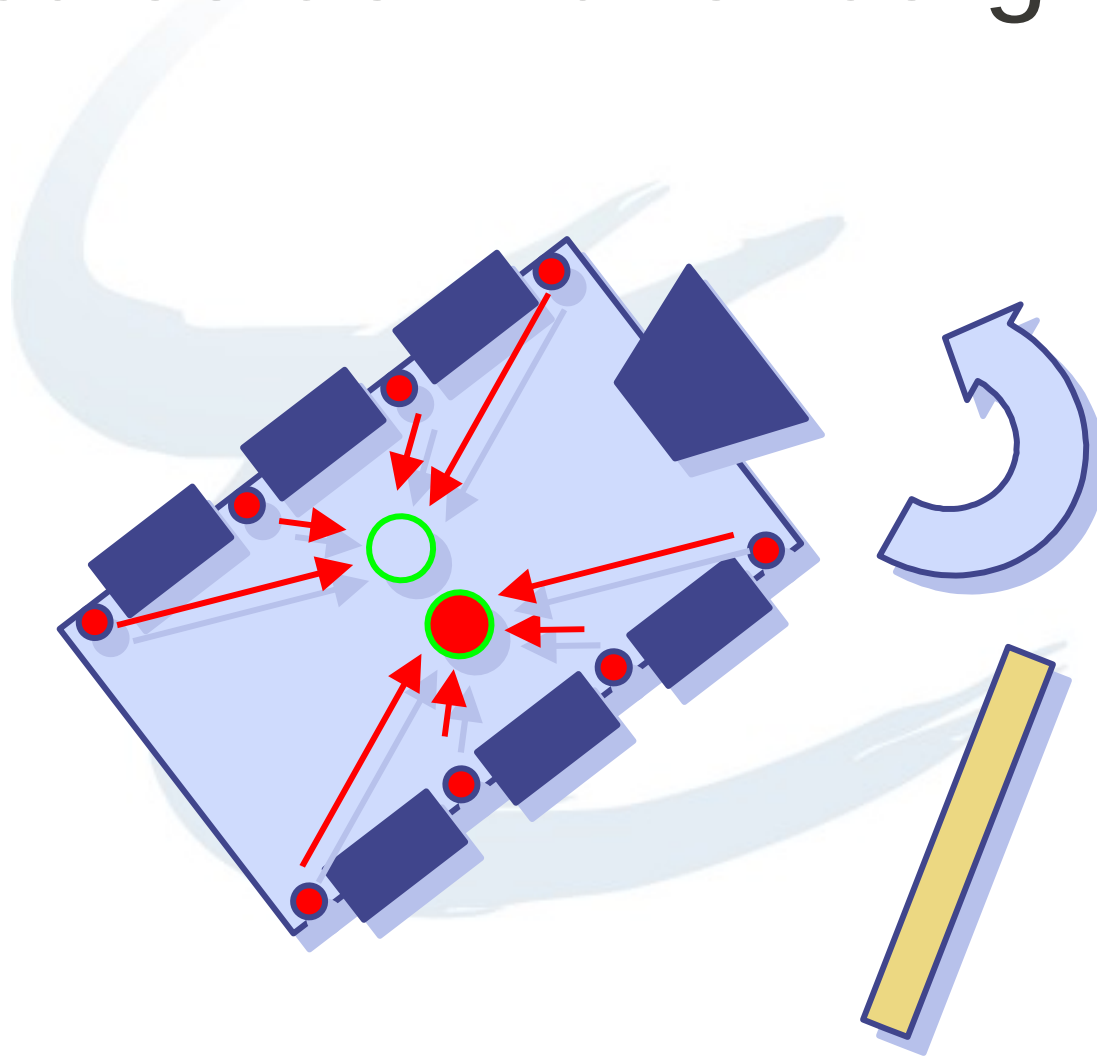
Véhicules de Braitenberg



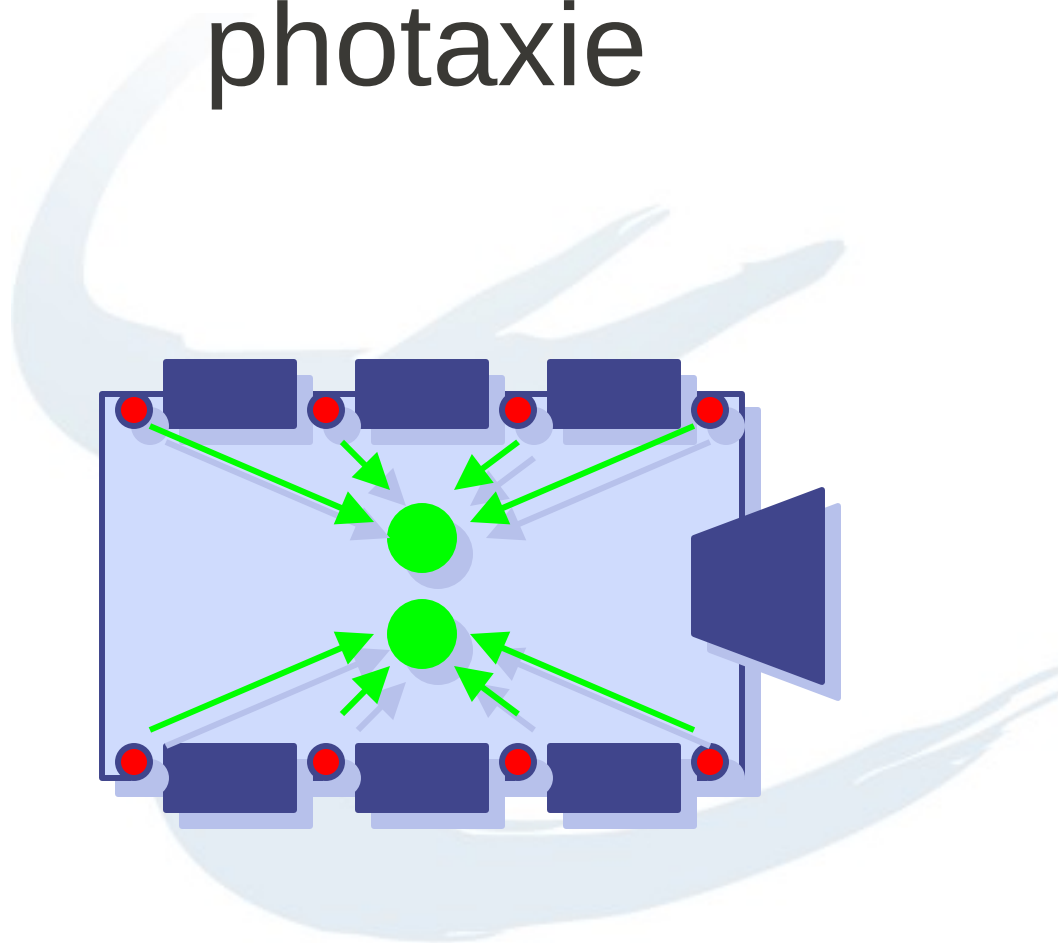
Véhicules de Braitenberg



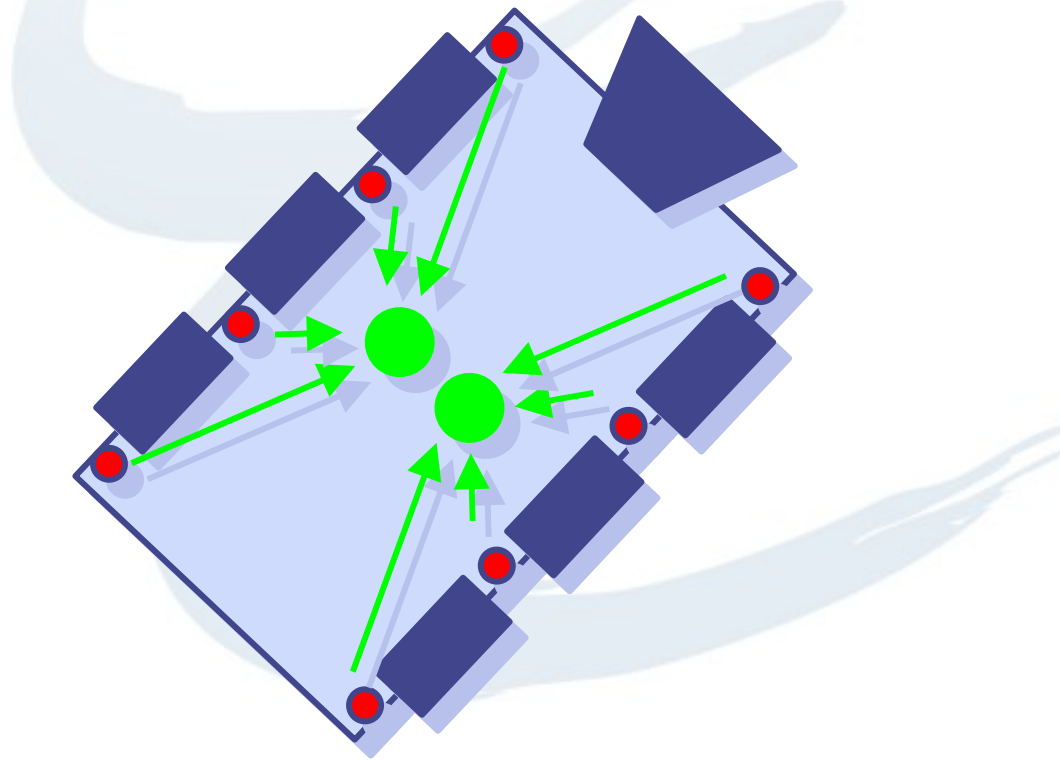
Véhicules de Braitenberg



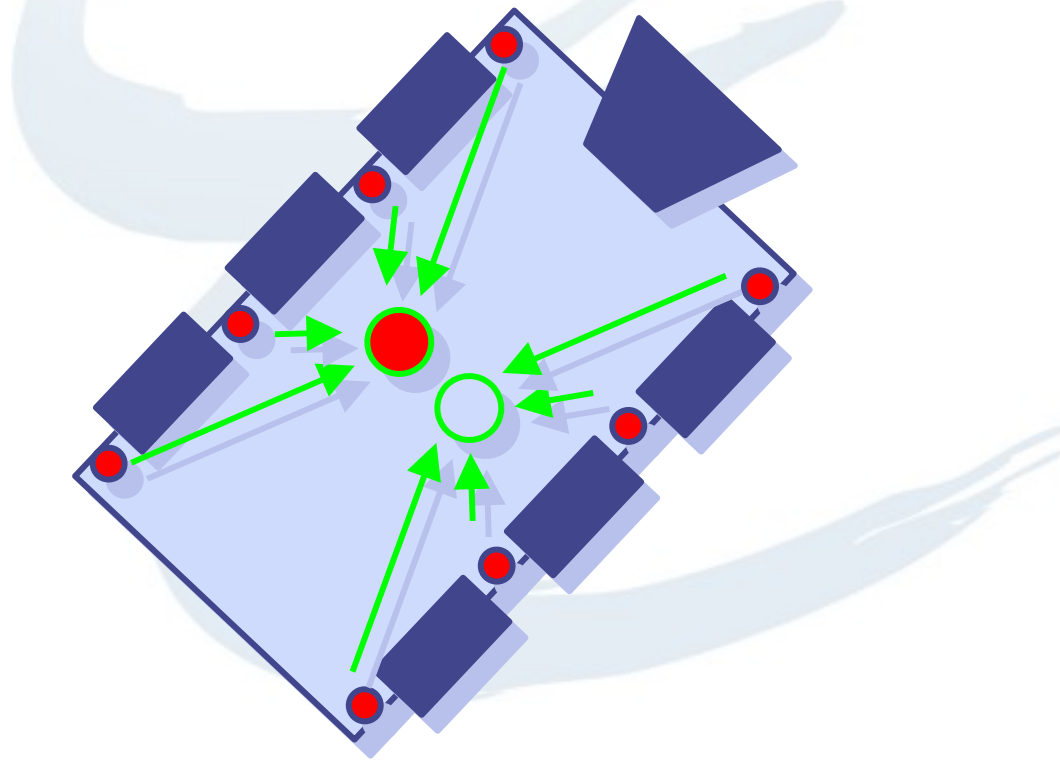
Véhicules de Braitenberg : photaxie



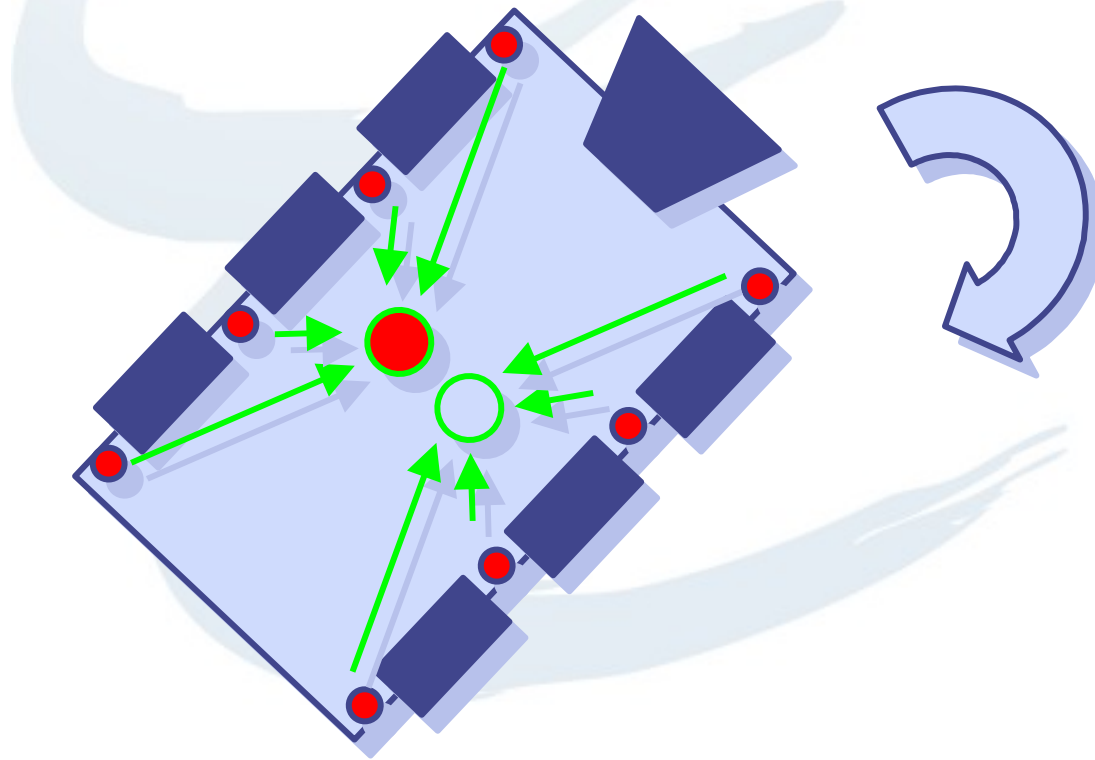
Véhicules de Braitenberg : photaxie



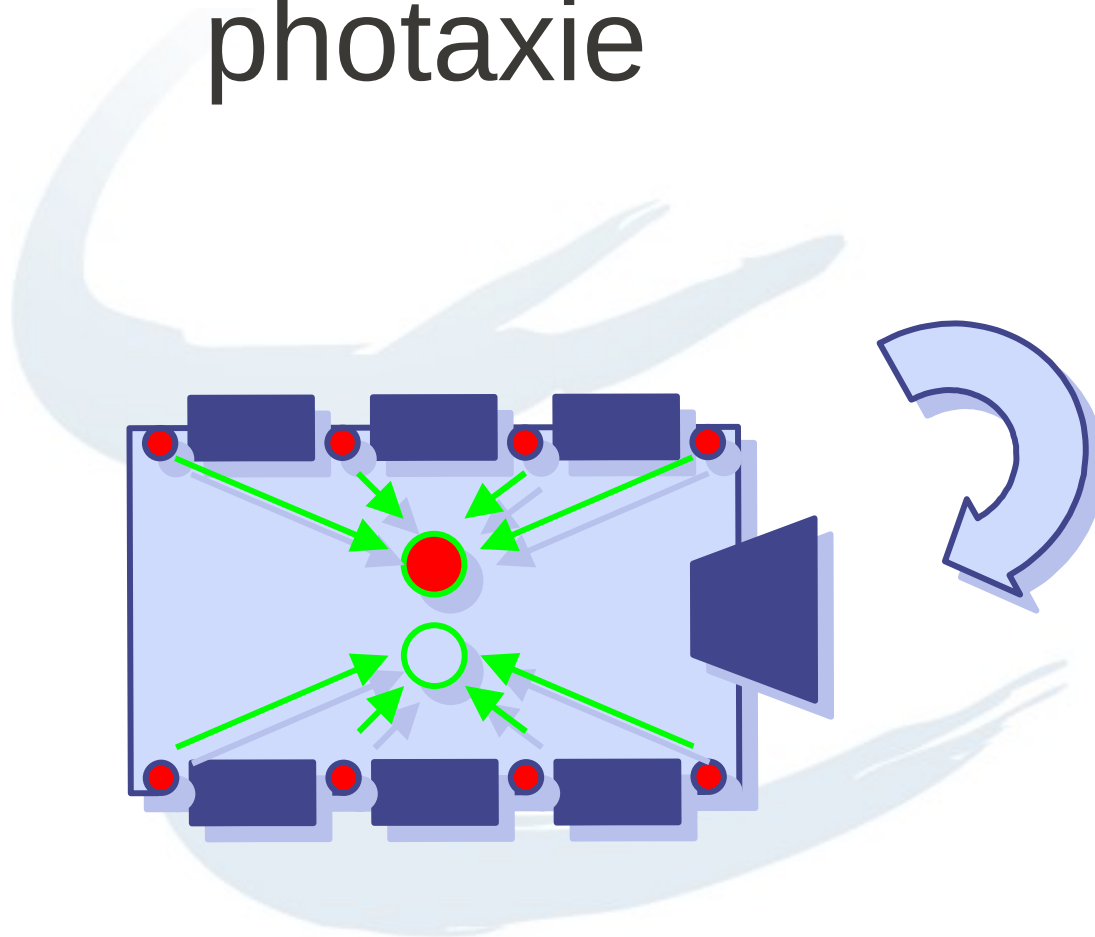
Véhicules de Braitenberg : photaxie

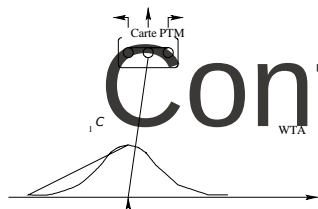


Véhicules de Braitenberg : photaxie

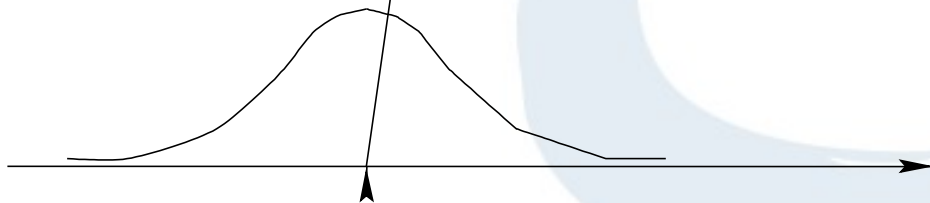
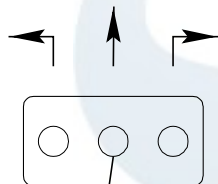


Véhicules de Braitenberg : photaxie





Contrôle flou

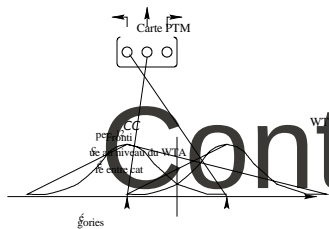


Categorie C_1

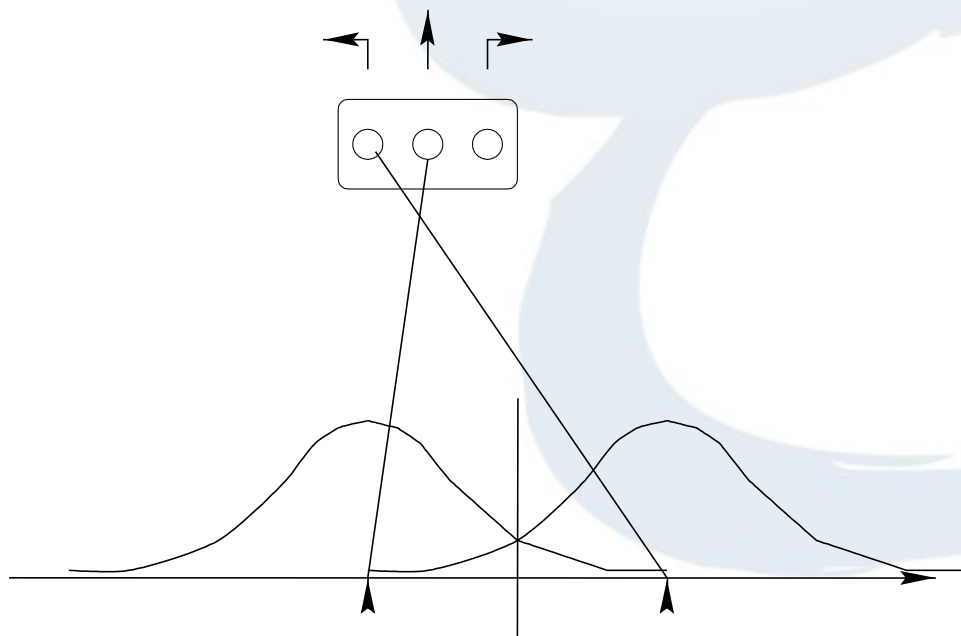
Reconnue

Mouvement

avancer



Contrôle flou

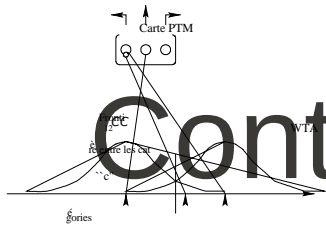


Categorie C_2
Reconnue

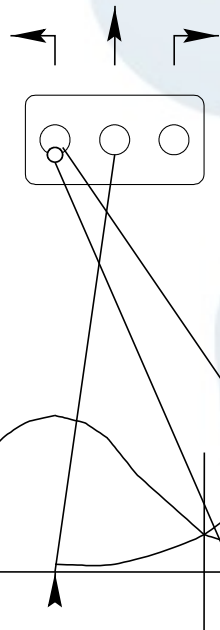
mouvement
tourner

Categorie C_1
Reconnue

Mouvement
avancer



Contrôle flou

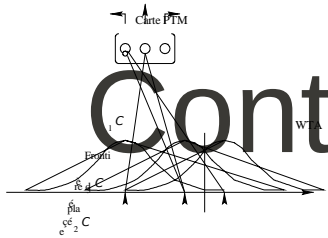


collision

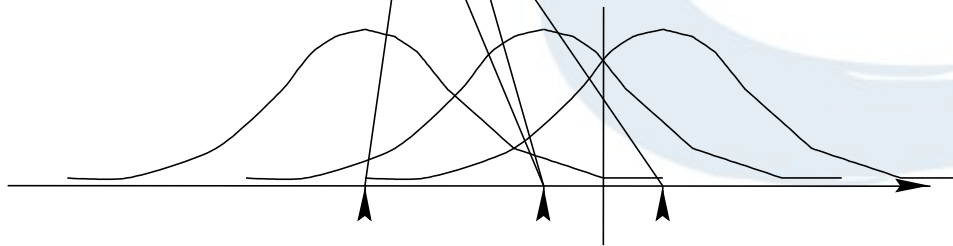
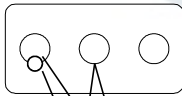
Mouvement
tourner

"c"

Mouvement
avancer



Contrôle flou

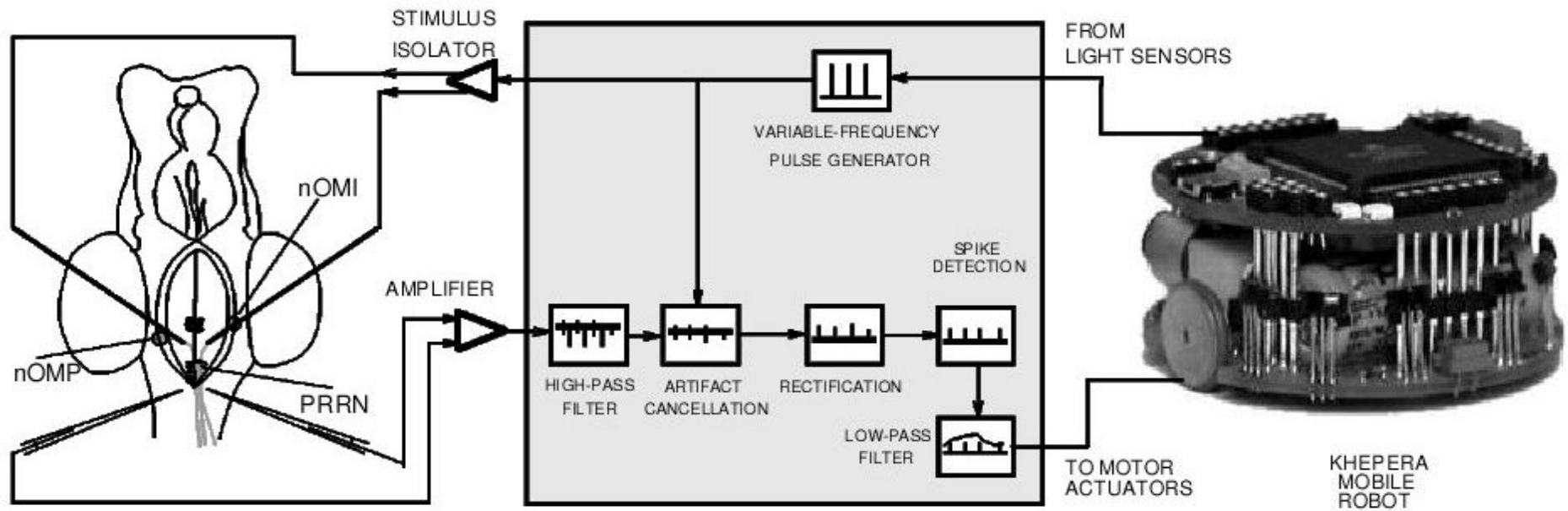


Mouvement
tourner

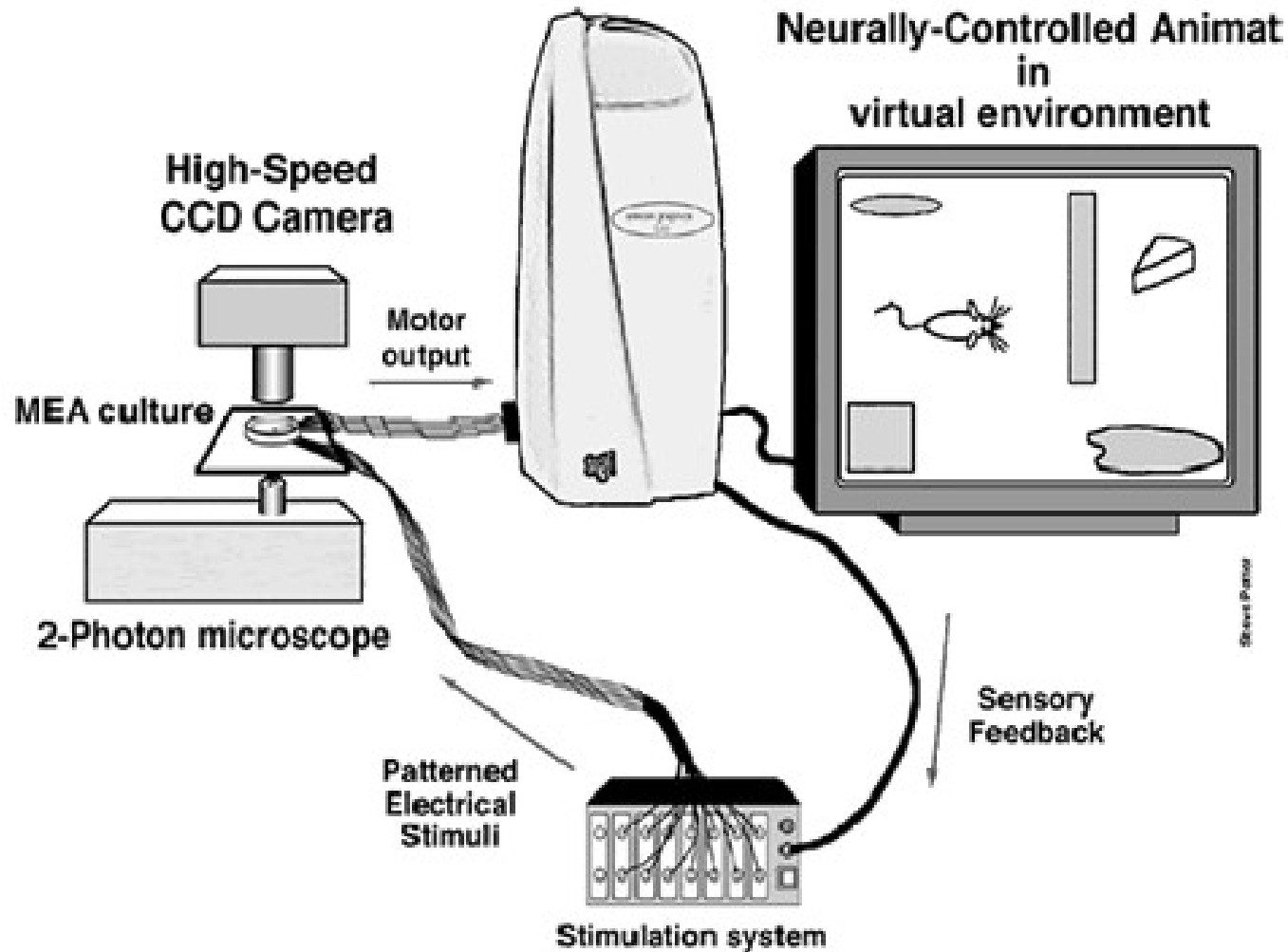
Mouvement
avancer

Mouvement
avancer

Contrôle par un «vrai» cerveau



Contrôle par un «vrai» cerveau [Potter03]



Réseaux de neurones

Application à la commande

Application à la commande

- Cadre



Commande Adaptative

- **Hypothèses**
 - Système contrôlable et observable
 - Représenté par une fonction de transfert
 - Le système est stable (sinon il faut le stabiliser préalablement)

Identification

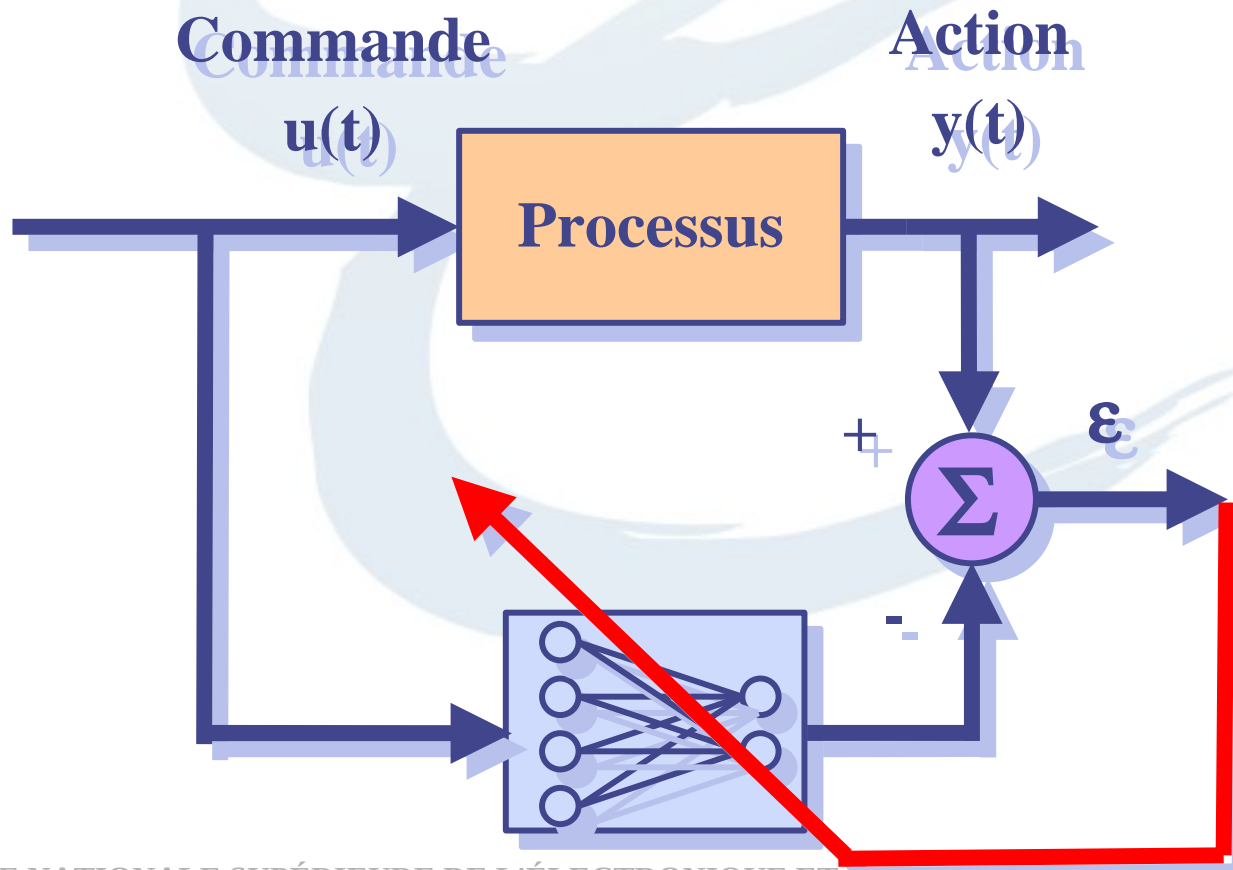
- Principe
 - On dispose d'un processus inconnu
 - On veut modéliser le processus par un réseau de neurones

Identification de processus

- Principe
 - faire apprendre un réseau de telle sorte qu'il réagisse comme le système.

Identification de processus

- Principe



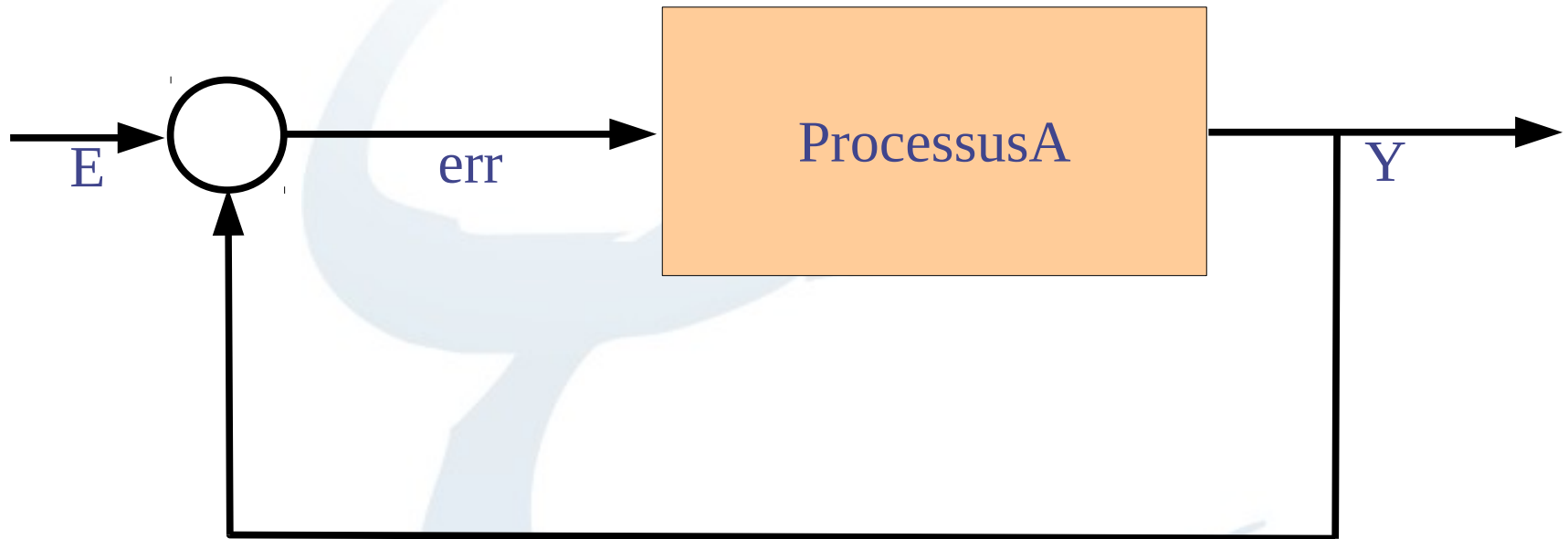
Méthode

- On échantillonne la réponse du processus à différentes commandes de gains faibles
- On constitue ainsi un ensemble d'associations (Commande \leftrightarrow Sortie)
- On sépare cet ensemble en deux groupes appelés entraînement et validation
- On optimise le réseau sur entraînement tant que l'erreur baisse sur validation

Structure du réseau ?

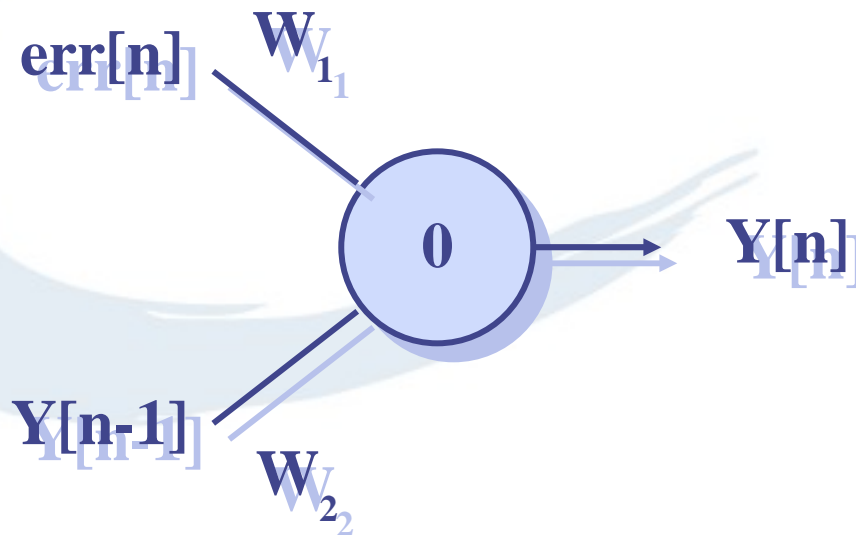
- La sortie du réseau doit être celle du processus, mais quelles en sont les entrées ?
 - $X[n]$ l'entrée au temps n (évident)
 - Entrées précédentes ($X[n-1]$, $X[n-2]$, ...)
 - Sortie précédentes ($Y[n-1]$, $Y[n-2]$, ...)
- Le choix de la structure du réseau est toujours délicat

Exemple



Structure du réseau choisie

- Fonction d'activation linéaire
- Sans seuil
- Combinaison de l'entrée et de la sortie retardée



Échantillonnage des réponses indicielles

```
% on génère la consigne (indice)
Consigne = ones(1, fin);
Consigne(1:10) = 0;

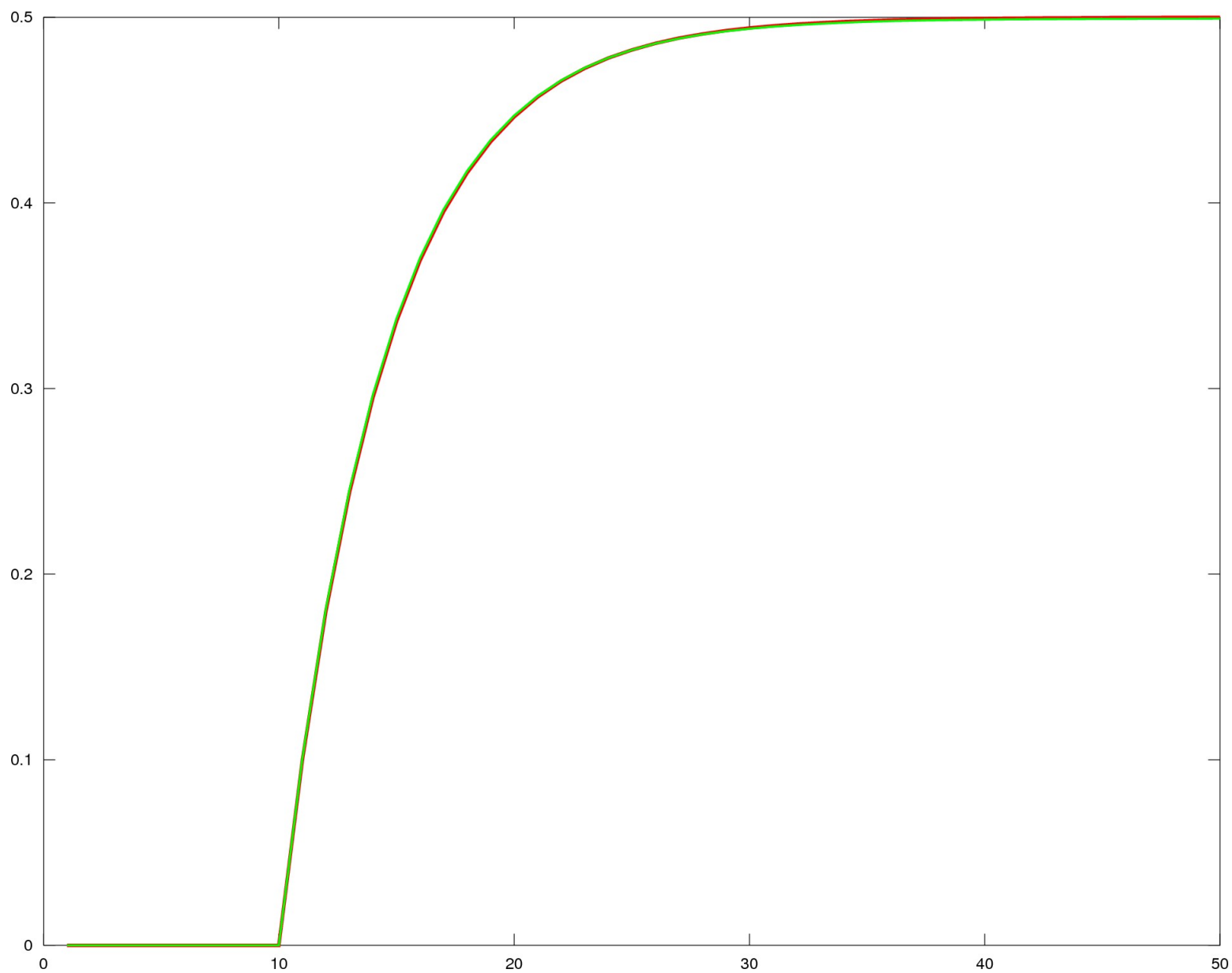
% gain allant de 0.01 à 0.5 par pas de 0.05
g = [0.01:0.05:0.5]
for ig = 1:length(g)
    % on génère la sortie en BF
    S = sampleProcessusA(Consigne, g(ig));
    % on génère la base d'apprentissage
    err = zeros(1, 1:fin);
    for i = [debut:fin]
        err(i) = Consigne(i) - S(i-1);
        X(:, i+1-debut) = [ g(ig)*err(i) S(i-1) ]';
    end
    % assemblage des réponses
    Xt = [Xt X];
    St = [St S(debut:fin)];
end
```

Apprentissage du réseau

- Initialisation des poids W_i aléatoirement
- Rétropropagation du gradient
 - Avec décroissance de la constante d'apprentissage
- 10 initialisations aléatoires
 - On garde la meilleure sur l'ensemble de validation

Test

- Échantillonnage du processus pour un gain différent (1.0)
- Comparaison entre la sortie du processus avec ce gain et la sortie du réseau avec ce gain
- Erreur = $3.47e-07$



Connaissances a priori

- Parfois, on connaît le modèle du système, mais pas les grandeurs numériques
- On peut utiliser un réseau de neurone pour découvrir les paramètres du système
- On construit le réseau tel que sa structure est proche de celle de la fonction de transfert

Exemple

- Le processus A de l'exemple précédent est en fait

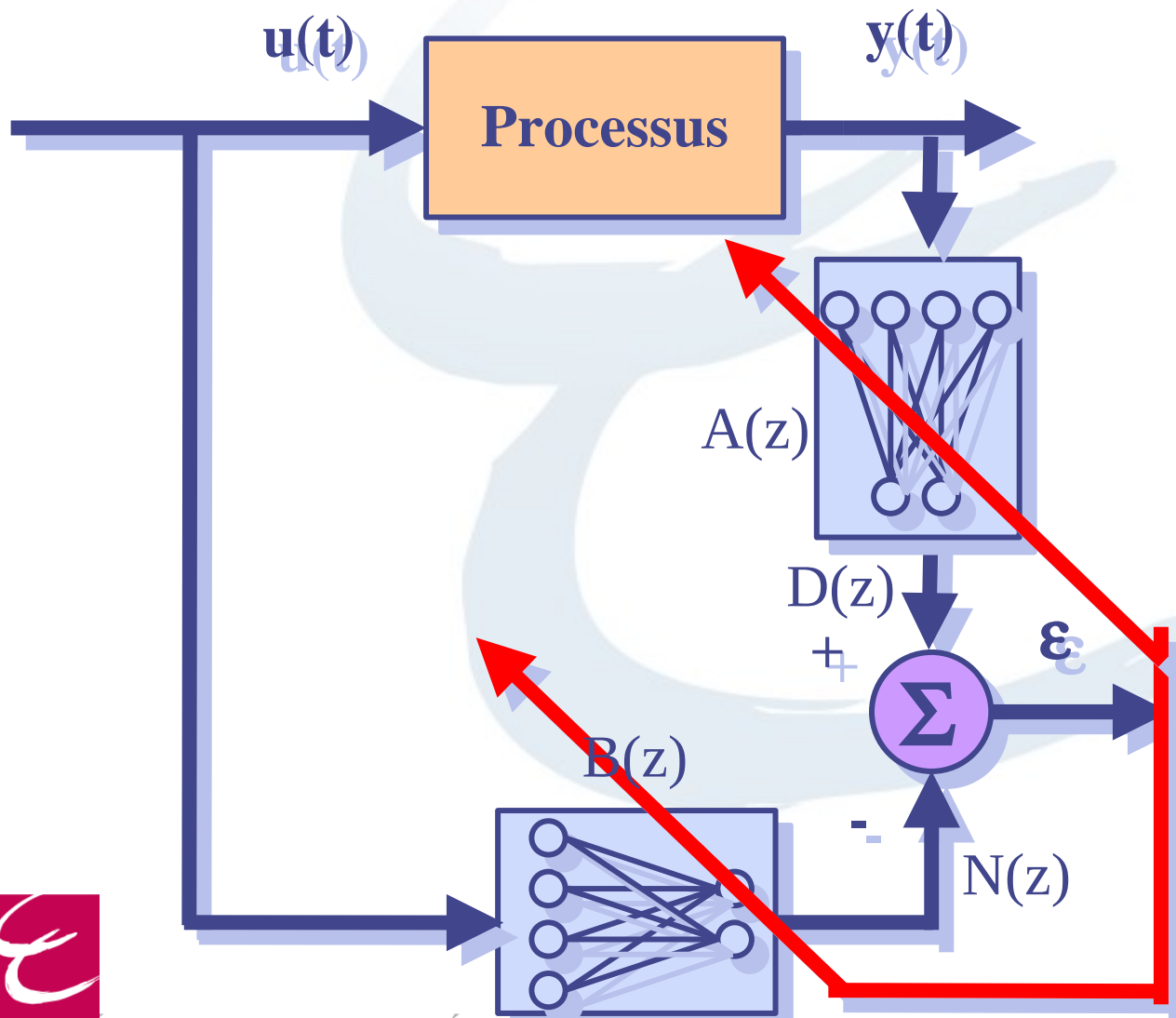
$$H(z) = \frac{0.1}{1 + 0.9z^{-1}}$$

- Le réseau modélisait donc bien le processus
- Les poids obtenus : [0.10086 0.89887]
(proches des vraies valeurs)

Identification série-parallèle

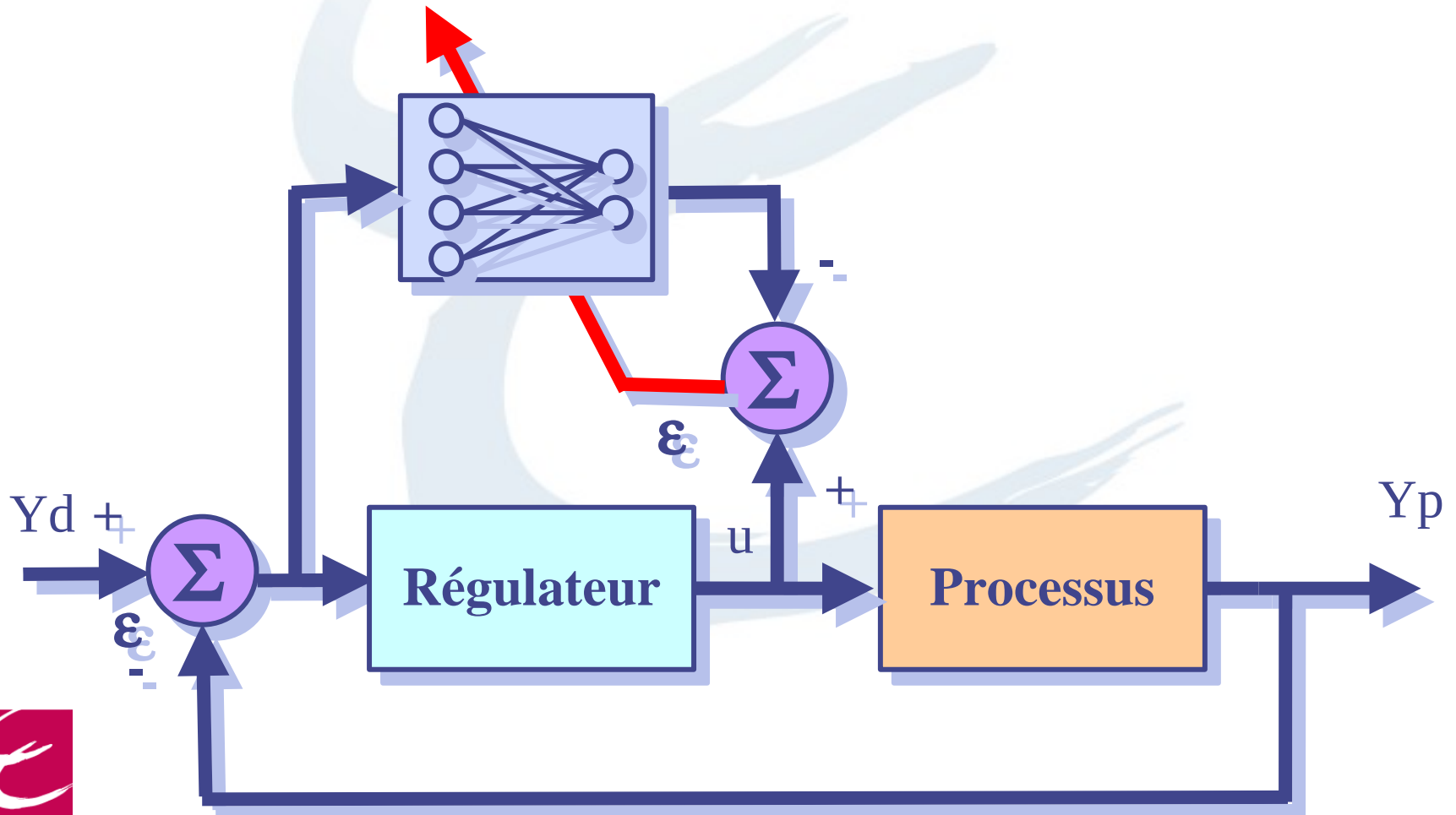
- Principe
 - On décompose la fonction de transfert
 - $H(z)=Y(z)/U(z)=B(z)/A(z)$
 - $B(z).U(z)=A(z).Y(z)$

Identification série/parallèle



Commande de processus

- Copie d'un régulateur



Commande Adaptative Inverse

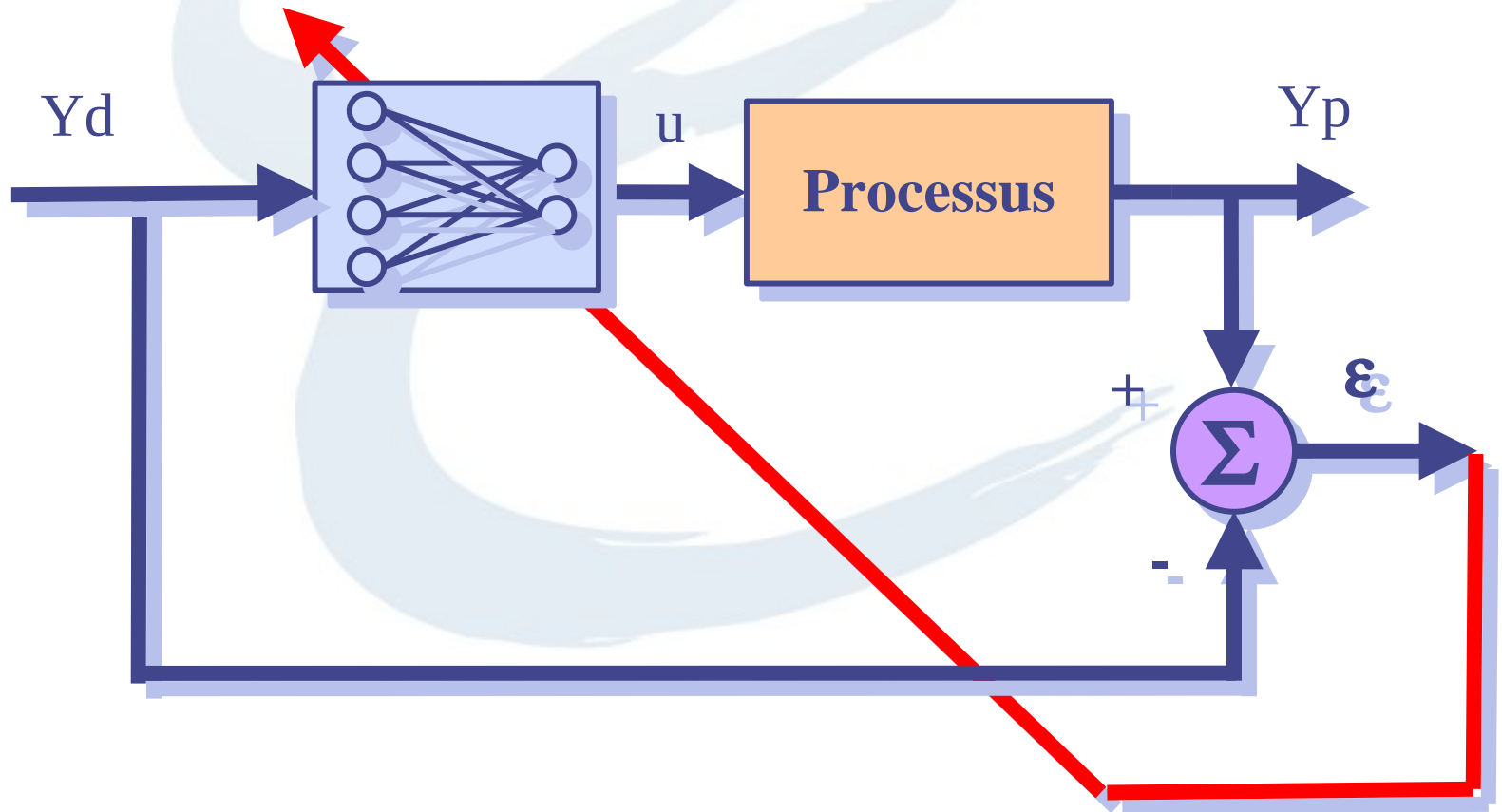
- Identification
 - Modélisation directe ou identification
 - Techniques de filtrages adaptatifs
 - Modélisation inverse ou déconvolution
 - mêmes techniques
 - commande de systèmes dynamiques
 - Minimisation des perturbations
 - Modèles directs et inverses

Commande Adaptative Inverse

- **Commande Adaptative**
 - Les algorithmes de filtrage adaptatifs peuvent être utilisés pour réaliser la commande adaptative de systèmes inconnus éventuellement variant dans le temps
 - Les paramètres du système peuvent être ajustés ou adaptés pour satisfaire l'inconnu ou les variations du système à commander

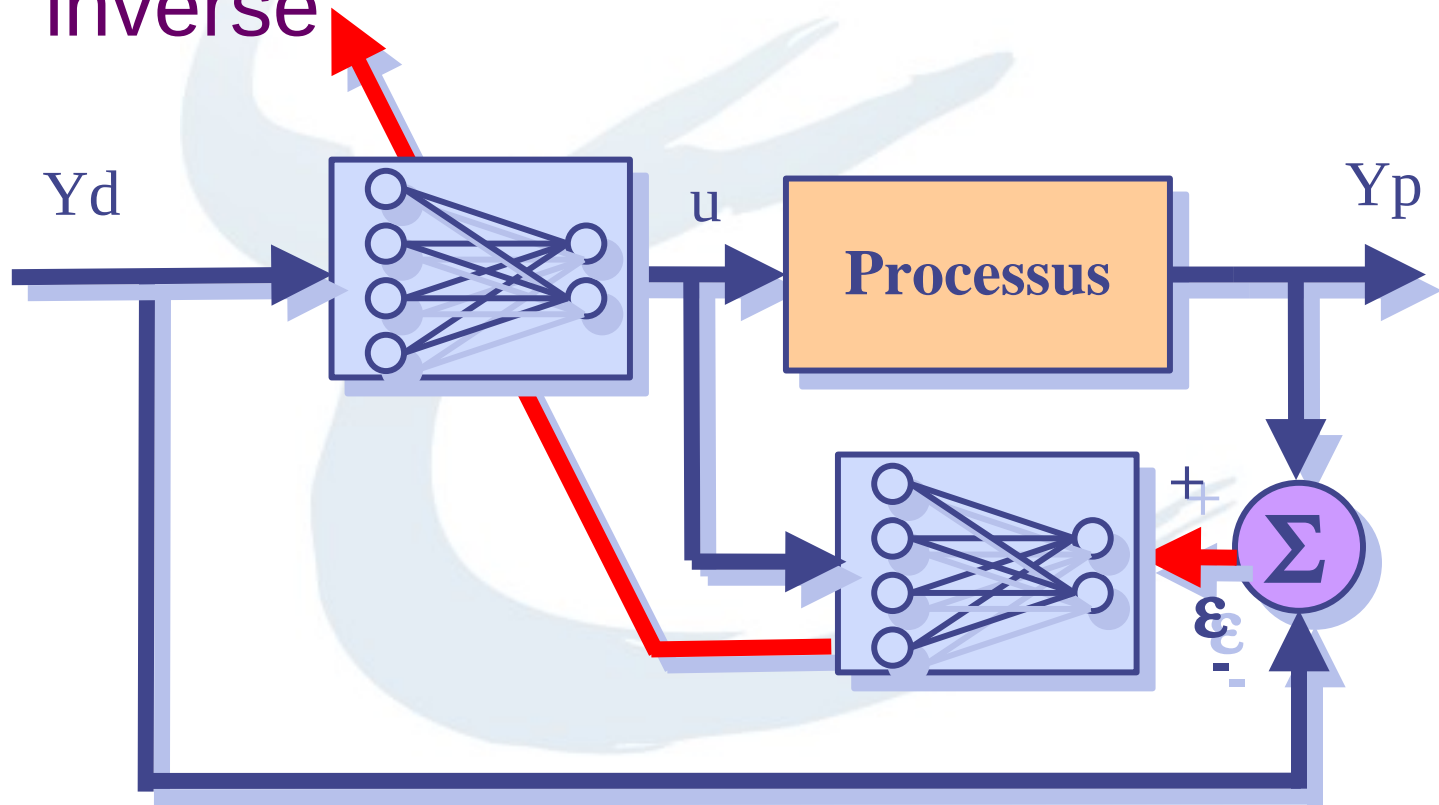
Commande de processus

- Commande par modèle inverse



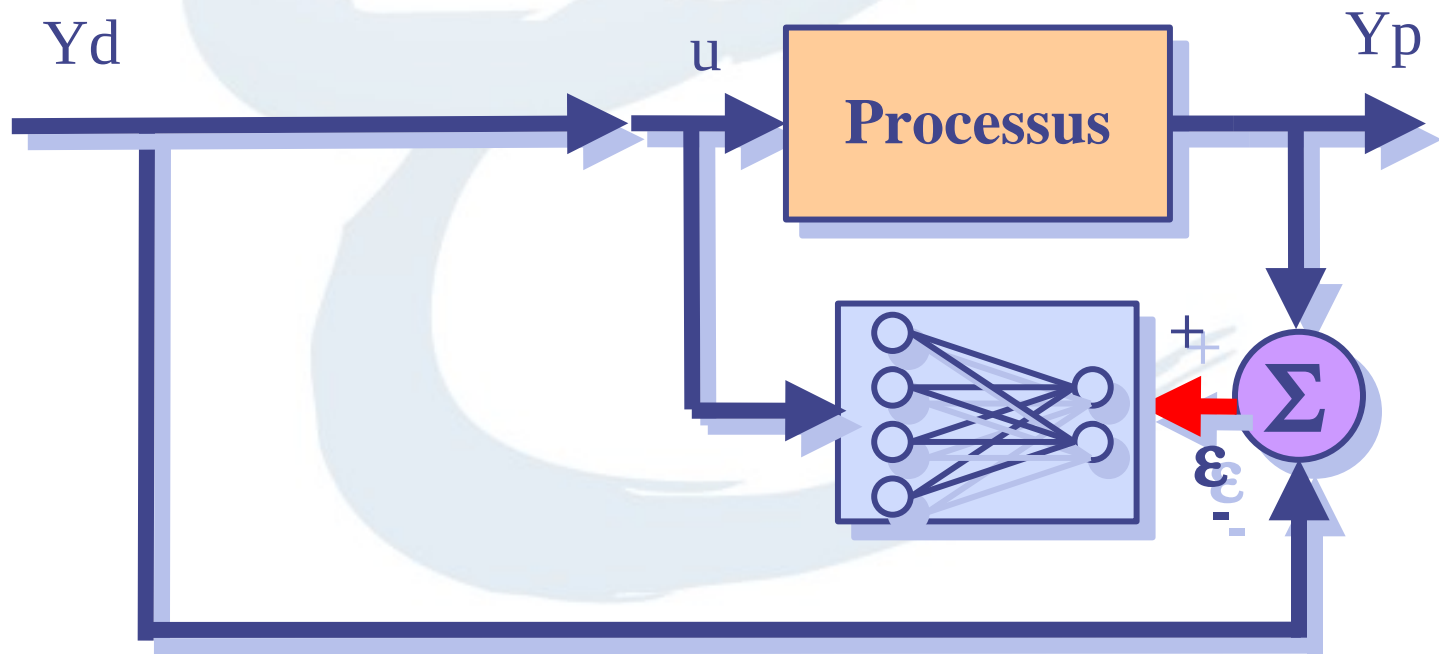
Commande de processus

- Commande par modèle direct et inverse



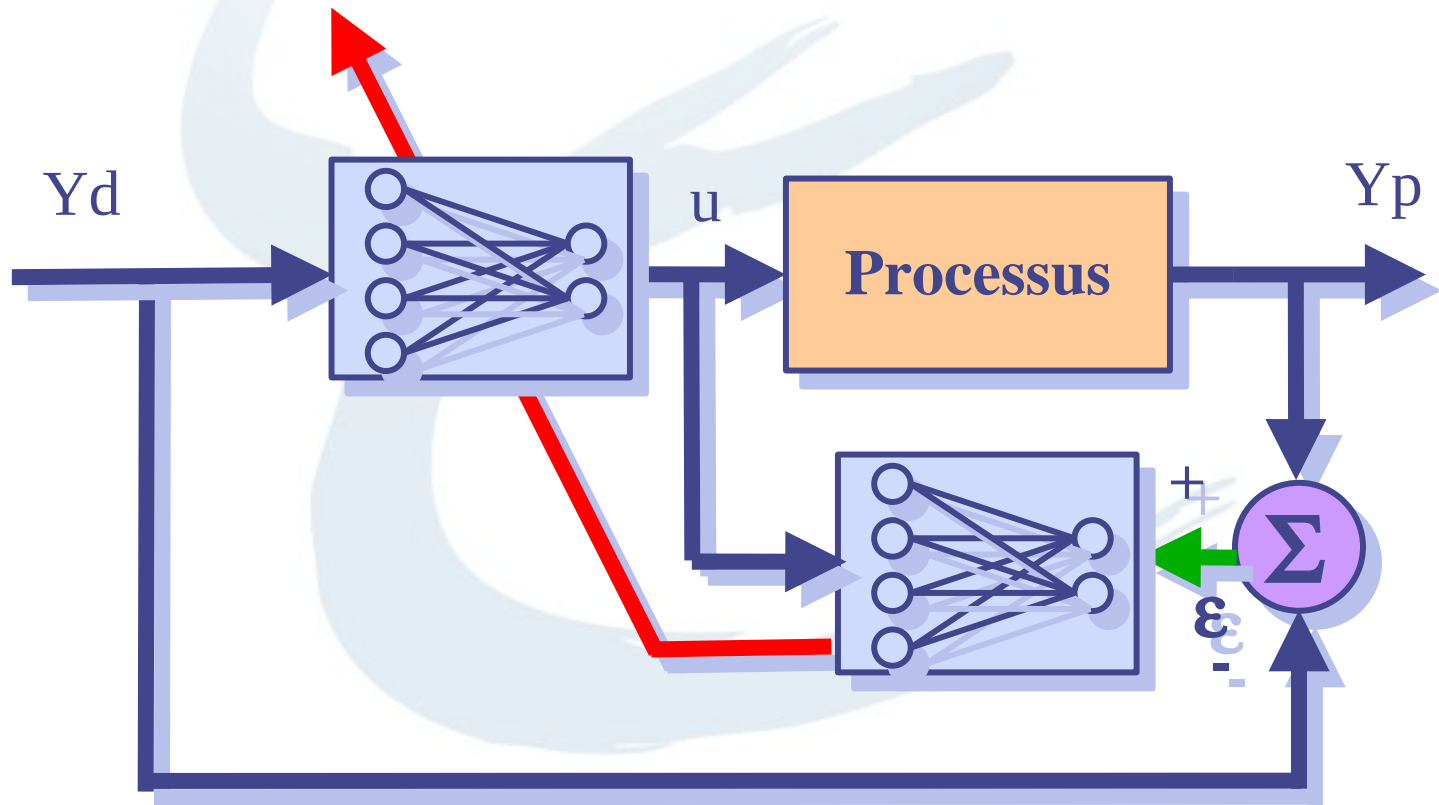
Commande de processus

- Apprentissage en 2 temps



Commande de processus

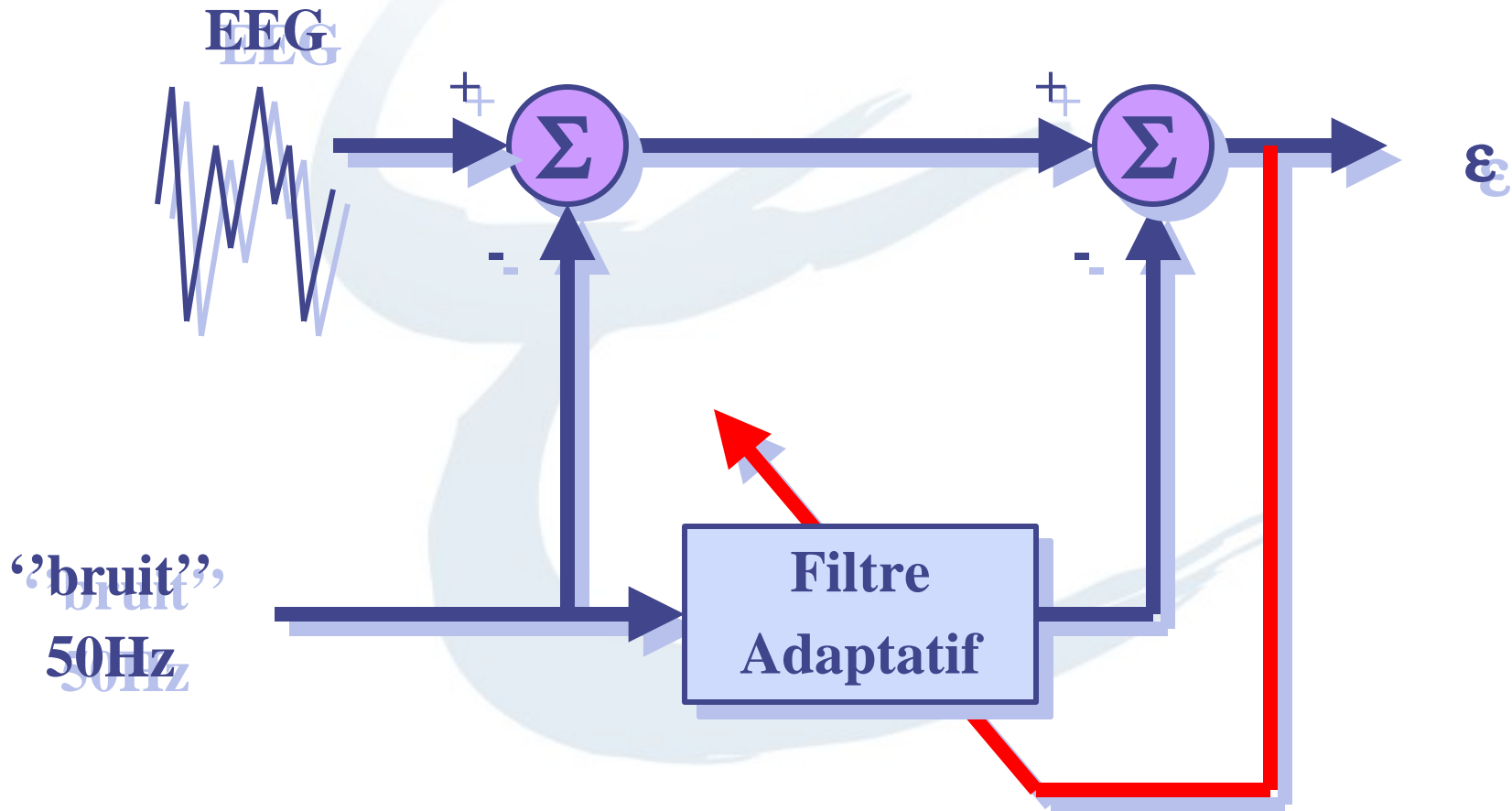
- Apprentissage en 2 temps



Applications en commande

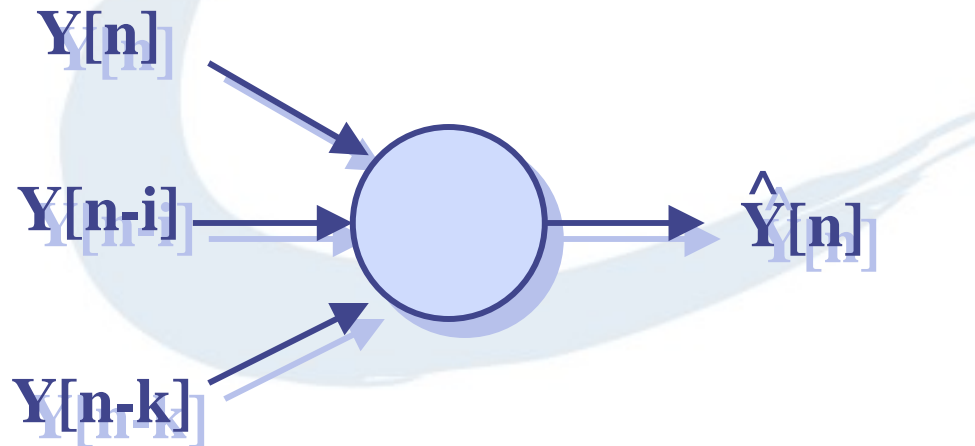
- Cadre de la commande adaptative
 - systèmes d'antennes adaptatives
 - égalisation de canaux
 - annulation d'interférence
 - analyse spectral
 - synthèse de la parole

Application : Compensation adaptatif du bruit



Application : Compensation adaptatif du bruit

- Utilisation d'un RN
 - Architecture : 1 neurone
 - Apprentissage : Widrow & Hoff



Fonctionnement en prédicteur

Applications en robotique

- Approche classique
 - modèles des systèmes
 - bras manipulateur, tête (modèles géométrique, cinématique, ...)
 - caméra(s) (paramètres intrinsèques)
 - reconstruction 3D de l'objet cible
 - commande du robot

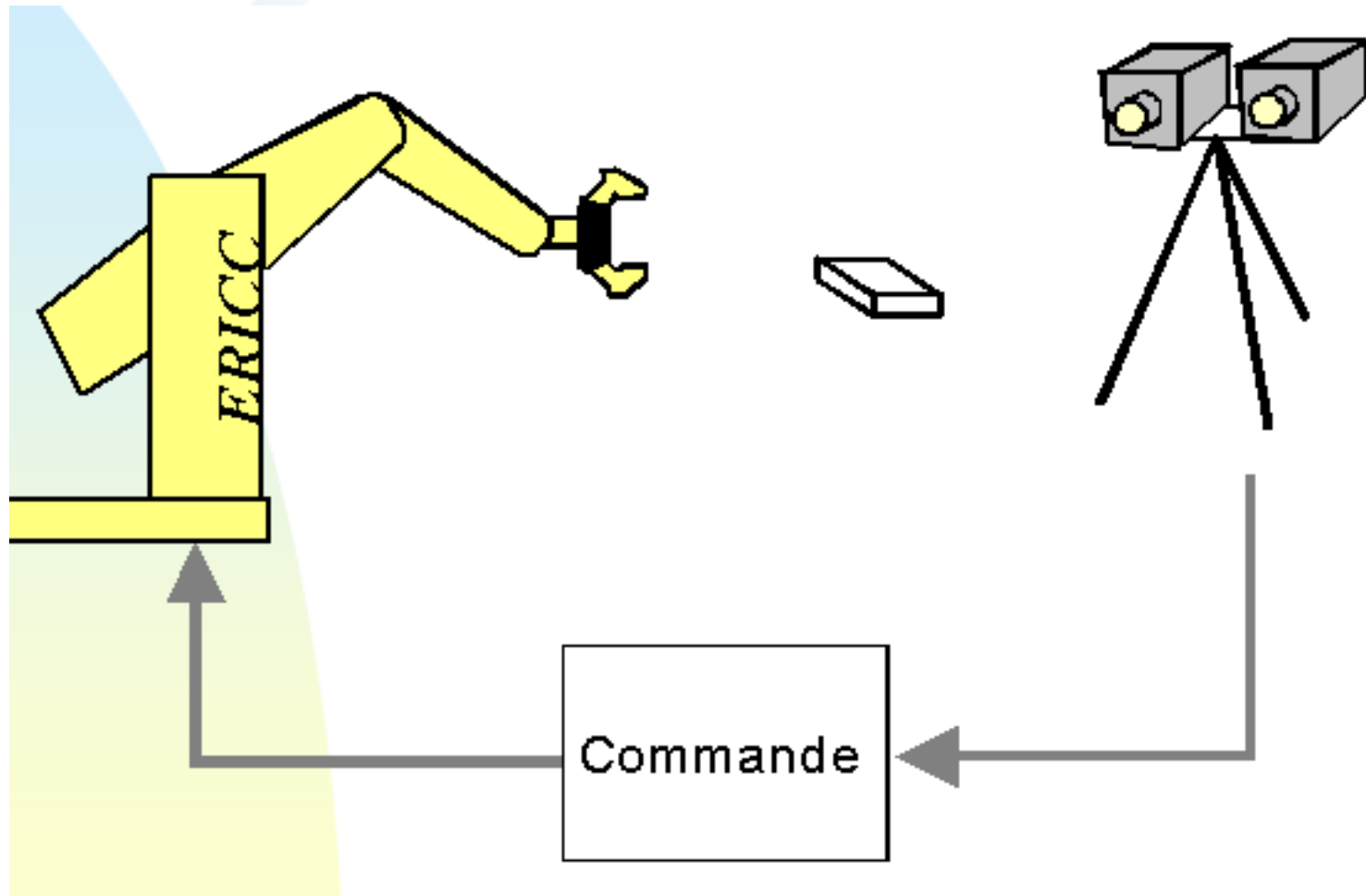
Applications en robotique

- > 80 % des robots industriels sont programmés par un « expert »
- très peu d'interaction avec l'environnement (« intelligence »?)
- très peu de progrès en vision robotique

Applications en robotique

- Difficultés
 - Modélisation
 - équations diff. non-linéaires
 - couplées
 - fonction du temps

Exemple : Contrôle visuo-moteur



Exemple : Contrôle visuo-moteur

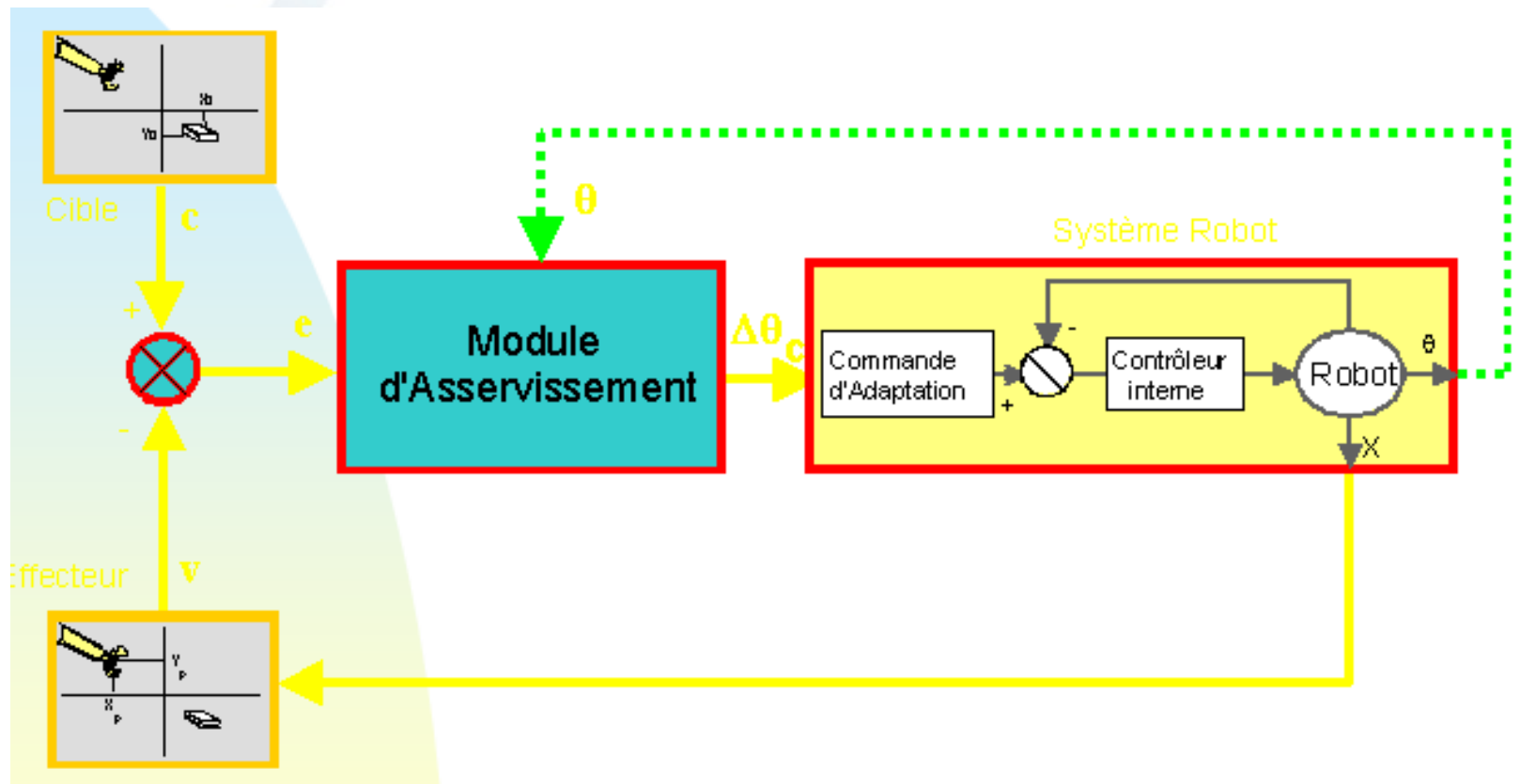
- Limites de la commande par les modèles
 - calibrage difficile
 - imperfections des modèles et capteurs
 - boucle ouverte
 - **Absence d'informations de retour sur le positionnement réellement effectué**

Utilisation de réseaux de neurones

- Caractéristiques
 - Apprentissage
 - Capacités d'apprentissage de relations complexes non-linéaires
 - Adaptabilité
 - Comportement robuste face à des changements imprévus des paramètres du système

Contrôle visuo-moteur

- 2 propriétés :
 - **regarder** ce que fait la main pour *corriger* son mouvement
 - **mémoriser** les adaptations pour acquérir une certaine *expérience* à partir des mouvements effectués



Contrôle visuo-moteur

- Projection des sorties motrices dans l'espace des entrées visuelles.

Conclusion

- Utilisation des RN très vaste
- Intérêt surtout quand
 - Pas « d'expert »
 - Systèmes non-linéaires
 - Systèmes versatiles