# Le calcul neuromimétique

- Le problème de l'interaction (problème XOR)
- La logique neuromimétique
- Définition de la linéarité

par Eric Keller, IMM-Lettres, Université de Lausanne

# Le problème XOR

- Le problème XOR figure de manière pro-éminente dans l'histoire des réseaux neuromimétiques
- La relation XOR est plus complexe qu'une simple corrélation multidimensionelle (p.ex. AND, OR)
  - AND et OR s'appliquent *sans interaction*
  - XOR s'applique *avec interaction*

$\rightarrow$ *Que veut dire "interaction" ici?*

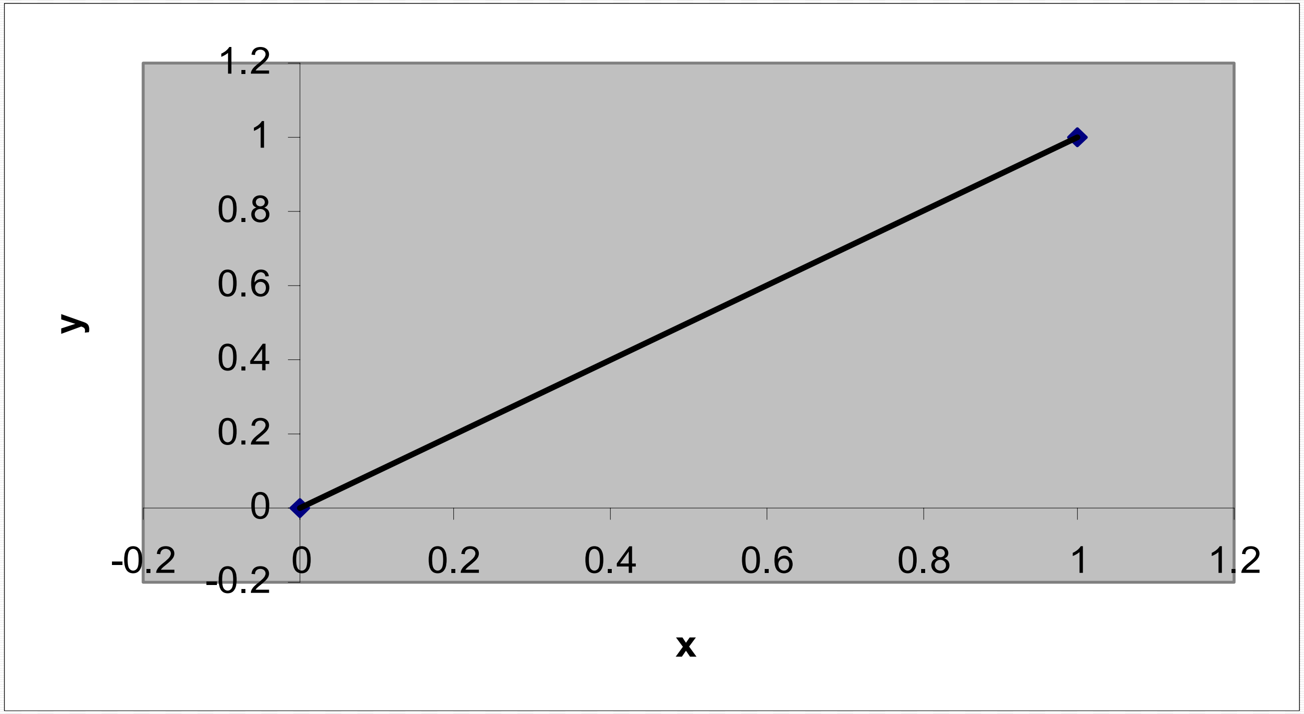
# Corrélation unidimensionnelle entre x et y

- Voici une corrélation unidimensionnelle et simpliste, où x ne peut avoir que deux valeurs, et où x prédit y parfaitement:

  **x** → **y**

  0 → 0

  1 → 1



# Corrélation multidimensionnelle gouvernée par AND et OR

- Les relations AND et OR définissent des corrélations multidimensionnelles et conditionnelles, à savoir

| | | | |
|---|---|---|---|
| Dimension 1: | 1 | 0 | 1 |
| Relation AND | $\rightarrow$ 1 | $\rightarrow$ 0 | $\rightarrow$ 0 |
| Dimension 2: | 1 | 0 | 0 |

| | | | |
|---|---|---|---|
| Dimension 1: | 1 | 0 | 1 |
| Relation OR | $\rightarrow$ 1 | $\rightarrow$ 0 | $\rightarrow$ 1 |
| Dimension 2: | 1 | 0 | 0 |

- Nous lisons ces formules comme suit:

  AND: "0 prédit 0 et (1 prédit 1, **si** les deux dimensions contiennent 1)"

  OR: "0 prédit 0 et (1 prédit 1, **si** au moins une dimension contient 1)"

- Les relations logiques AND ou OR gouvernent donc l'application d'une partie de la corrélation.

# AND et OR - sans interaction

- Dans ces cas, il est possible d'examiner l'effet de chaque dimension séparément.
- Ceci nous permet de dire que AND et OR fonctionnent sans interaction entre les dimensions.

```
// Règles OR
// les règles s'appliquent de manière séquentielle et indépendante
res = 0;
if (dimension_1 == 1) res = 1; else res = 0; // règle 1
if (dimension_2 == 1) res = 1; else res = 0; // règle 2
```

**La présence d'un 0 sur la dimension 2 peut défaire l'effet de la règle 1**

```
// Règles AND
// les règles sont imbriquées, mais ont des effets indépendants
res = 0;
if (dimension_1 == 1) { // règle 1
    res = 1;
    if (dimension_2 == 0) res = 0; } // règle 2
```

**La présence d'un 0 sur la dimension 2 peut défaire l'effet de la règle 1**

# XOR pas si facile, car interaction

| | | | | |
|---|---|---|---|---|
| Dimension 1: | 1 | 0 | 1 | 0 |
| Relation XOR | $\rightarrow$ 0 | $\rightarrow$ 0 | $\rightarrow$ 1 | $\rightarrow$ 1 |
| Dimension 2: | 1 | 0 | 0 | 1 |

En d'autres mots, afin de connaître l'effet des deux dimensions, il faut les examiner **ensemble**.

On ne peut les juger séparément.

// règle XOR

if (**dim_1==1 && dim_2==0**) res=1; else res=0;

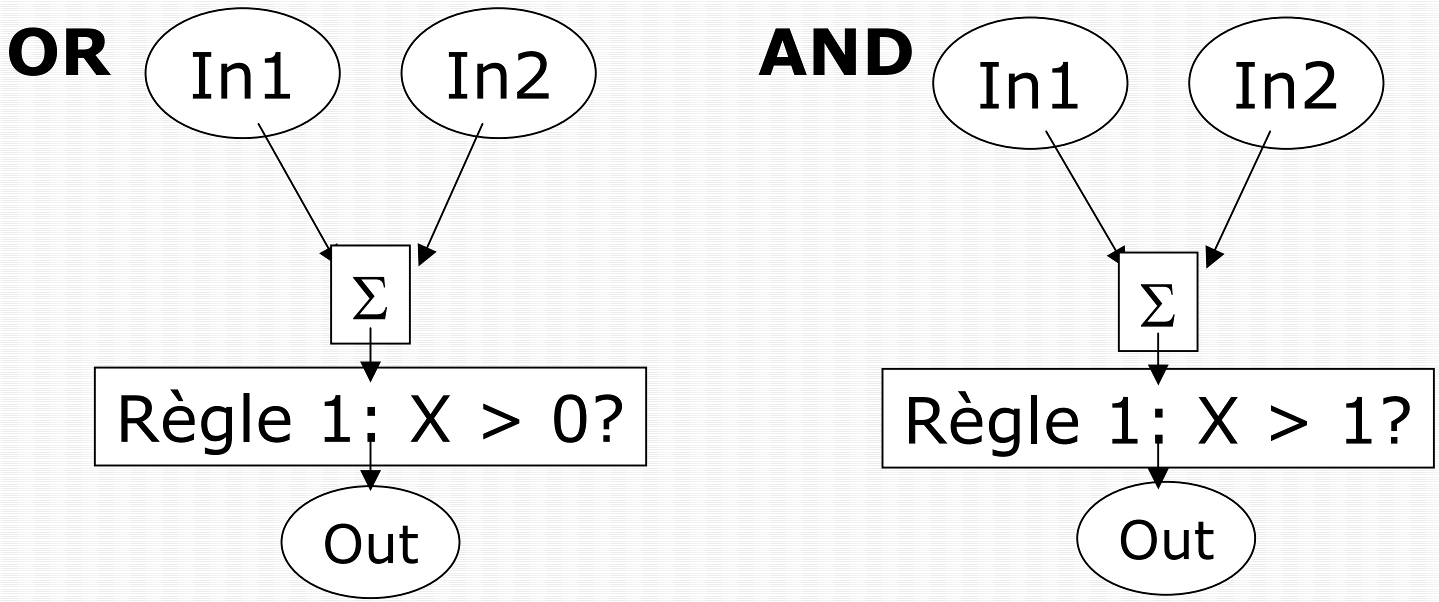
if (**dim_1==0 && dim_2==1**) res=1; else res=0;

# Réseau neuromimétique

- Supposons qu’on souhaite simuler les relations OR, AND et XOR au moyen d’un système neuromimétique traditionnel.
- Ce type de système incorpore plusieurs outils formels, dont:
  1. Le **cumul** de poids (d’influences) s’effectuant de l’entrée vers la sortie
  2. L’utilisation de **multiplicateurs,** variables et stables (biais), pour calculer les poids.
  3. La **binarisation** de ces poids après chaque cumul, c.-à-d., la transformation du cumul en 0 ou 1.
  4. La binarisation s’effectue typiquement au moyen d’une **fonction de transfert** (p.ex. la fonction sigmoïde), mais elle peut aussi se réduire à une simple règle si…alors.
- Pour examiner nos relations logiques, nous n’utiliserons pour l’instant que le **cumul** et la **binarisation**, mais ni **biais,** ni **fonction de transfert**. Nous reviendrons sur ces éléments supplémentaires plus tard.

# AND et OR en tant que problème de prédiction

- Nous verrons ici que la présence ou l'absence d'une **interaction** a des effets importants sur l'architecture d'un réseau neuromimétique servant à découvrir des relations entre les entrées multidimensionnelles et la sortie d'un système.
- Voici les réseaux simplifiés pour AND et OR:



# Calcul des solutions OR et AND au moyen des réseaux simplifiés

**OR**

**Supposons in1==0 et in2==0**
**x = 0 + 0 = 0**
**Règle 1 (x>0?): y = 0**
**Out = 0**

**Supposons in1==1 et in2==0**
**x = 1 + 0 = 1**
**Règle 1 (x>0?): y = 1**
**Out = 1**

**Supposons in1==1 et in2==1**
**x = 1 + 1 = 2**
**Règle 1 (x>0?): y = 1**
**Out = 1**

**AND**

**Supposons in1==0 et in2==0**
**x = 0 + 0 = 0**
**Règle 1 (x>1?): y = 0**
**Out = 0**

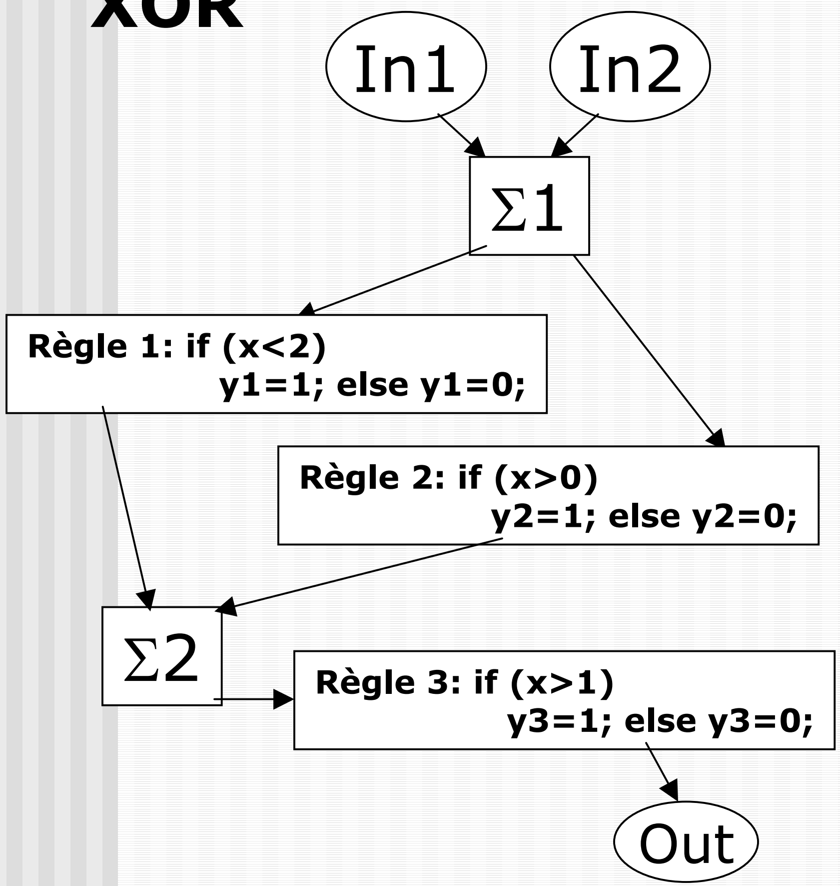
**Supposons in1==1 et in2==0**
**x = 1 + 0 = 1**
**Règle 1 (x>1?): y = 0**
**Out = 0**

**Supposons in1==1 et in2==1**
**x = 1 + 1 = 2**
**Règle 1 (x>1?): y = 1**
**Out = 1**

Suite à un simple ajustement du **seuil de la règle de binarisation,** le système (1) peut distinguer correctement entre la prédiction de 0 et 1, et (2) peut juger soit la condition AND soit la condition OR.

# Le cas nettement plus complexe de l'interaction entre deux dimensions

**XOR**




**Supposons in1==1 et in2==0**
**Σ1 = 1 + 0 = 1**
**Règle 1: y1 = 1**
**Règle 2: y2 = 1**
**Σ2 = y1+y2 = 2**
**Règle 3: y3 = 1**

**Supposons in1==1 et in2==1**
**Σ1 = 1 + 1 = 2**
**Règle 1: y1 = 0**
**Règle 2: y2 = 1**
**Σ2 = y1+y2 = 1**
**Règle 3: y3 = 0**

**Supposons in1==0 et in2==0**
**Σ1 = 0 + 0 = 0**
**Règle 1: y1 = 1**
**Règle 2: y2 = 0**
**Σ2 = y1+y2 = 1**
**Règle 3: y3 = 0**

→ Besoin de deux étapes

# Les deux étapes requises par le cas interactif

- Règles 1 et 2: un premier “filtre” de deux règles binarisantes, afin d’accepter les seules sommes acceptables:

  $\Sigma 1 > 0$ et $\Sigma 1 < 2$
- Règle 3: une deuxième règle binarisante, afin de juger si les deux règles ont été activées:

  $\Sigma 2 > 1$
- Ceci n’est pas la seule solution neuromimétique au problème XOR, mais elle a le bénéfice d’être explicite.
- Nous concluons de ces illustrations que l’examen de variables indépendantes en interaction requiert une étape de logique intermédiaire, donc une **couche de noeuds cachés**.

# Une solution au problème XOR obtenu au moyen d'un réseau classique

Pour examiner l'apprentissage neuromimétique du problème XOR…

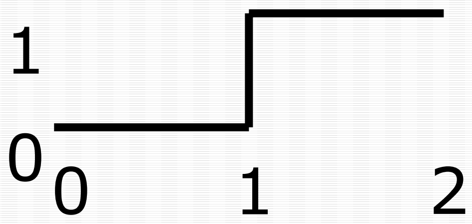
- Nous avons utilisé NevProp
- Nous avons effectué l'apprentissage suivant:

| **x** | **y** | **sortie** |
|---|---|---|
| 0 | 0 | 0 |
| 1 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 1 | 0 |

- Nous avons utilisé un réseau entièrement connecté
- +- 50 itérations
- L'apprentissage a été satisfaisant
- Nous avons sauvé les poids
- Nous avons simulé le résultat en Excel

Mais avant de passer à l'examen de la simulation, nous devons expliquer la **fonction de transfert**.

# Une transition douce au moyen d'une fonction de transfert




- Les règles de binarisation illustrées précédemment constituent des transitions abruptes.
- Une règle comme

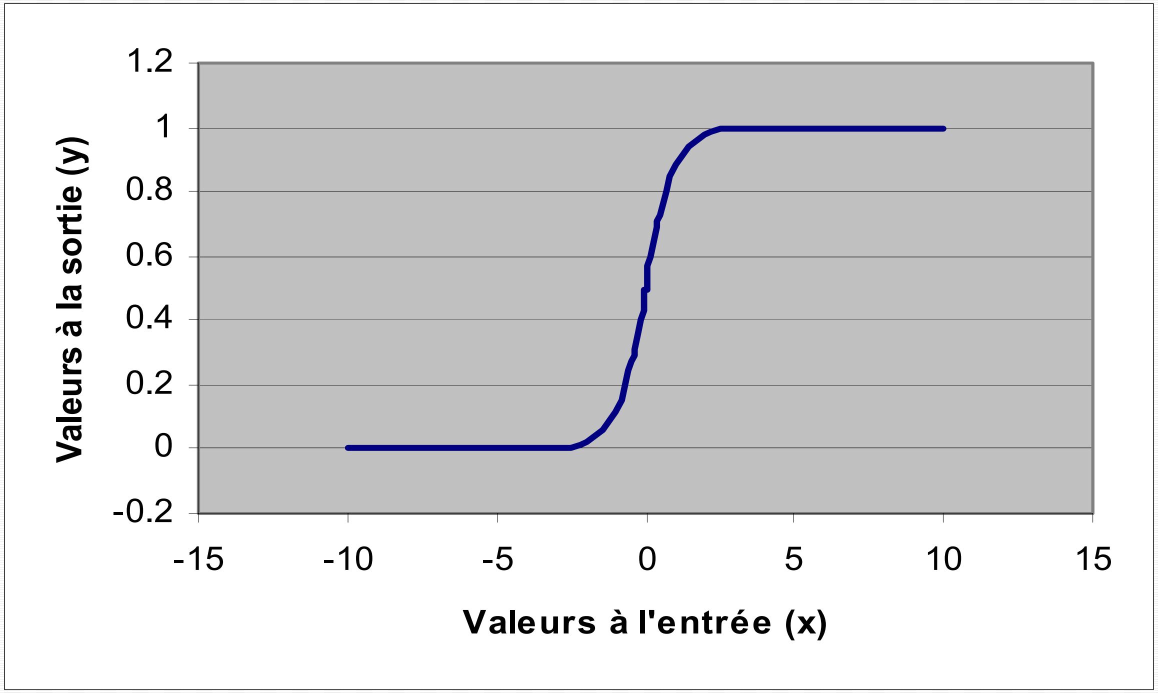
```
if (x<=1) y = 0; else y = 1;
```

  introduit un changement abrupt de 0 à 1 au dessus de 1.
- Ceci peut avoir des effets néfastes quand un fort nuage de points est situé de part et d'autre de la transition.
- Une "transition douce" réduirerait ces effets. Ceci devient possible par l'application d'une **fonction de transfert** appropriée.
- Ce type de fonction de transfert sert à transformer l'espace numérique de $-\infty$ à $+\infty$ en un espace situé entre 0 et 1, avec une transition douce au milieu.
- Les fonctions de transfert les plus fréquentes sont les fonctions **sigmoïde** et **logit**.

# La binarisation:
# La fonction de transfert sigmoïde

- **En Excel, la sigmoïde se calcule avec la formule "=1/(1+EXP(-x))"**
- **EXP(x) = e multiplié par soi-même x fois, où e = 2.7182818. EXP(x) est l'opposé de LN(x)**
- **Fournit une transition assez rapide (pente raide). Appropriée pour les cas où les données sont assez précises et bien distribuéees.**

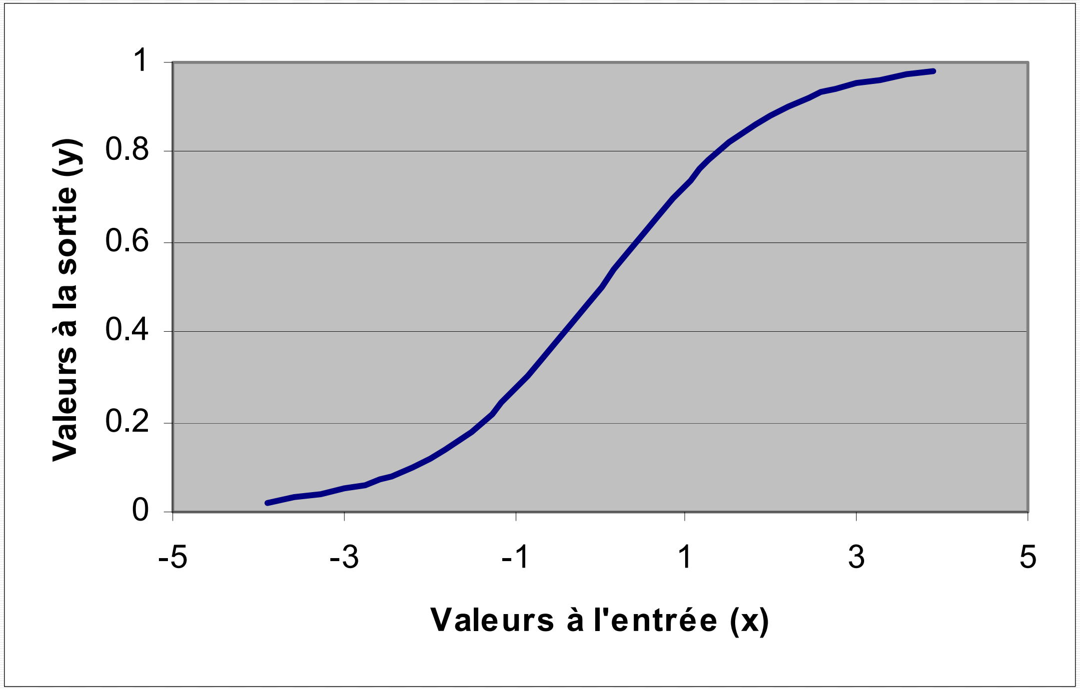
| X | sigmoïde(x) |
|---|---|
| 10 | 0.999999998 |
| 1 | 0.880797078 |
| 0.5 | 0.731058579 |
| 0.1 | 0.549833997 |
| 0.05 | 0.524979187 |
| 0.01 | 0.504999833 |
| 0 | 0.5 |
| -0.01 | 0.495000167 |
| -0.05 | 0.475020813 |
| -0.1 | 0.450166003 |
| -0.5 | 0.268941421 |
| -1 | 0.119202922 |
| -5 | 4.53979E-05 |
| -10 | 2.06115E-09 |



# La binarisation: La fonction de transfert logit

- La fonction logit fournit une transition nettement plus douce (pente moins raide).
- Approprié pour les cas où les données sont moins précises et/ou la distribution des données est plutôt inégale.

| | |
|---|---|
| 3.89182 | 0.98 |
| 1.045969 | 0.74 |
| 0.847298 | 0.7 |
| 0.663294 | 0.66 |
| 0.489548 | 0.62 |
| 0.322773 | 0.58 |
| 0.160343 | 0.54 |
| 0 | 0.5 |
| -0.16034 | 0.46 |
| -0.32277 | 0.42 |
| -0.48955 | 0.38 |
| -0.66329 | 0.34 |
| -0.8473 | 0.3 |
| -1.04597 | 0.26 |
| -3.89182 | 0.02 |



# Solution XOR transféré sur Excel

Examinons maintenant la solution neuromimétique pour le problème XOR:

| | | | | | | | | | |
|---|---|---|---|---|---|---|---|---|---|
| | *pente* | 1.0000 | | | | | | | |
| | *biais/constante* | 1.0000 | | | | | | | |
| | | | | | | | | | |
| | **1 noeud caché, modèle entièrement connecté** | | | | | | | | |
| | | *c1* | *pds c1-c2* | *c2-brut* | *c2-sigm* | *pds c1/2-c3* | *c3-brut* | *c3-sigm* | *c3-bin* |
| ***Influences*** | *1.1->2.1* | **0** | -7.6350 | 0.0000 | | | | | |
| ***sur la*** | *1.2->2.1* | **1** | -7.6353 | -7.6353 | | | | | |
| ***couche cachée*** | *2.2 b -> 2.1* | **biais** | **11.7365** | **11.7365** | **sigmoide** | | | | |
| | *somme c2* | | | **4.1012** | **0.9837** | | | | |
| ***Influences*** | *2.1 -> 3.1* | | | **somme** | | 11.2416 | 11.0586 | | |
| ***sur la*** | *1.1 -> 3.1* | | | | | 5.1013 | 0.0000 | | |
| ***couche de sortie*** | *1.2 -> 3.1* | | | | | 5.1007 | 5.1007 | | |
| | *2.3 b -> 3.1* | | | | | **-13.4433** | **-13.4433** | **sigmoide** | |
| | *somme-c3* | | | | | **somme** | **2.7159** | **0.9380** | **1** |
| | | | | | | | | | |

Pour comprendre ce tableau, il est suggéré (1) de dessiner ce modèle entièrement connecté, consistant de 2 noeuds d'entrée, un noeud caché et un noeud de sortie, et (2) de "jouer" avec les entrées afin de voir les calculs changer. Le fichier Excel original disponible sur ce site permet d'ajuster les valeurs d'entrée c1. (c1…3 = colonne 1…3, pds = poids, bin = binarisé sur 0 ou 1)

# Prédiction linéaire et non linéaire

- Nous savons que les systèmes non linéaires profitent souvent d'une exploration neuromimétique, car les systèmes neuromimétiques peuvent y découvrir des relations fiables, même si la relation entre observations d'entrée et de sortie est non linéaire
- Mais que veut dire "linéaire" précisément?

# Prédiction linéaire et non linéaire

La relation entre l'entrée et la sortie d'un système est linéaire, si elle montre

1. Homogénéité
2. Additivité

1. Homogénéité entre deux vecteurs

   **Si x $\rightarrow$ y et x * z $\rightarrow$ y *z alors la relation est homogène.**

   **En d'autres mots: un effet multiplicatif doit s'appliquer de la même manière à la variable indépendante et à la variable dépendante.**

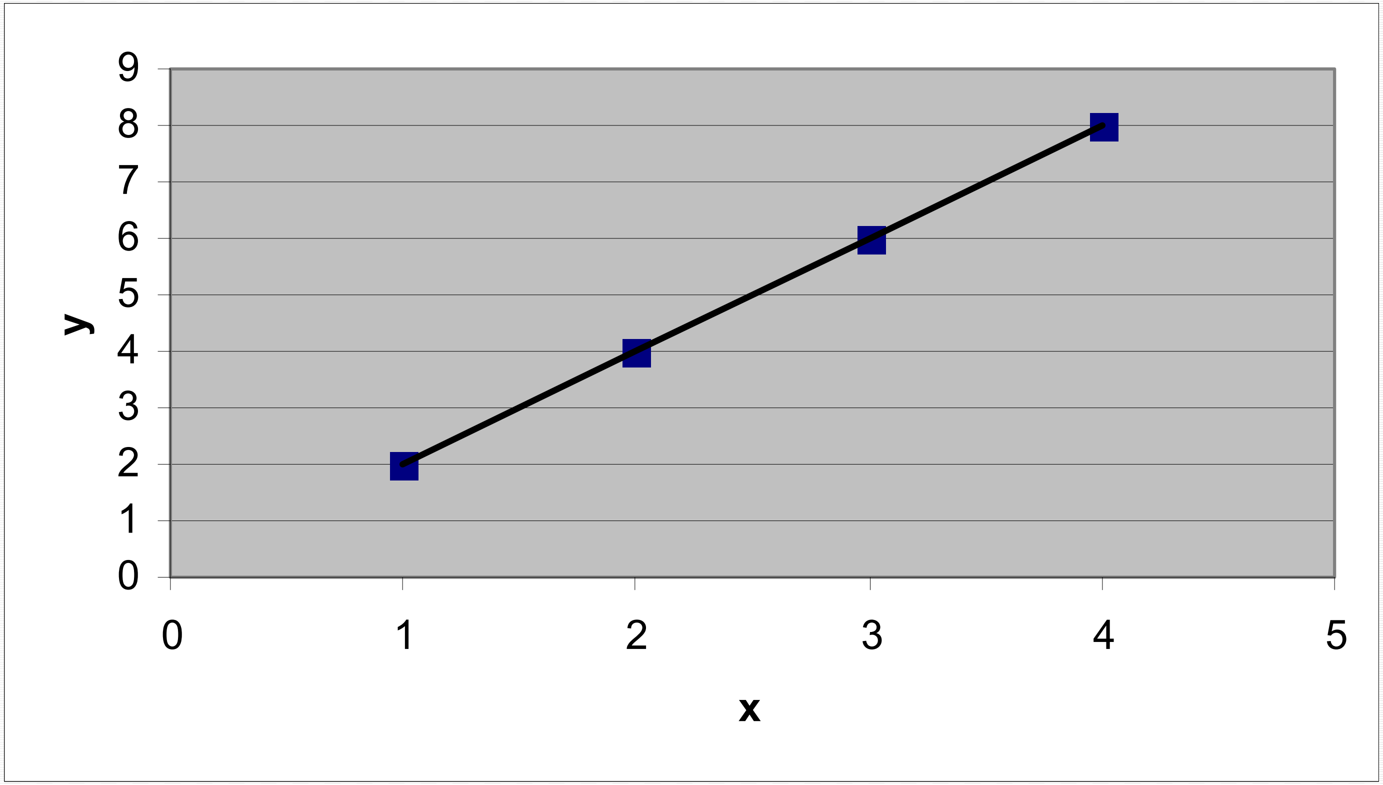
2. Additivité entre deux composantes d'un système

   **Si un système consiste de deux relations homogènes mises en séquence, et si la sortie de ce système peut être prédite par l'addition des deux relations, alors le système montre de l'addivité.**

# Exemple: homogénéité

Un système homogène (multiplication par 2)

| **x** | **y** |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 6 |
| 4 | 8 |

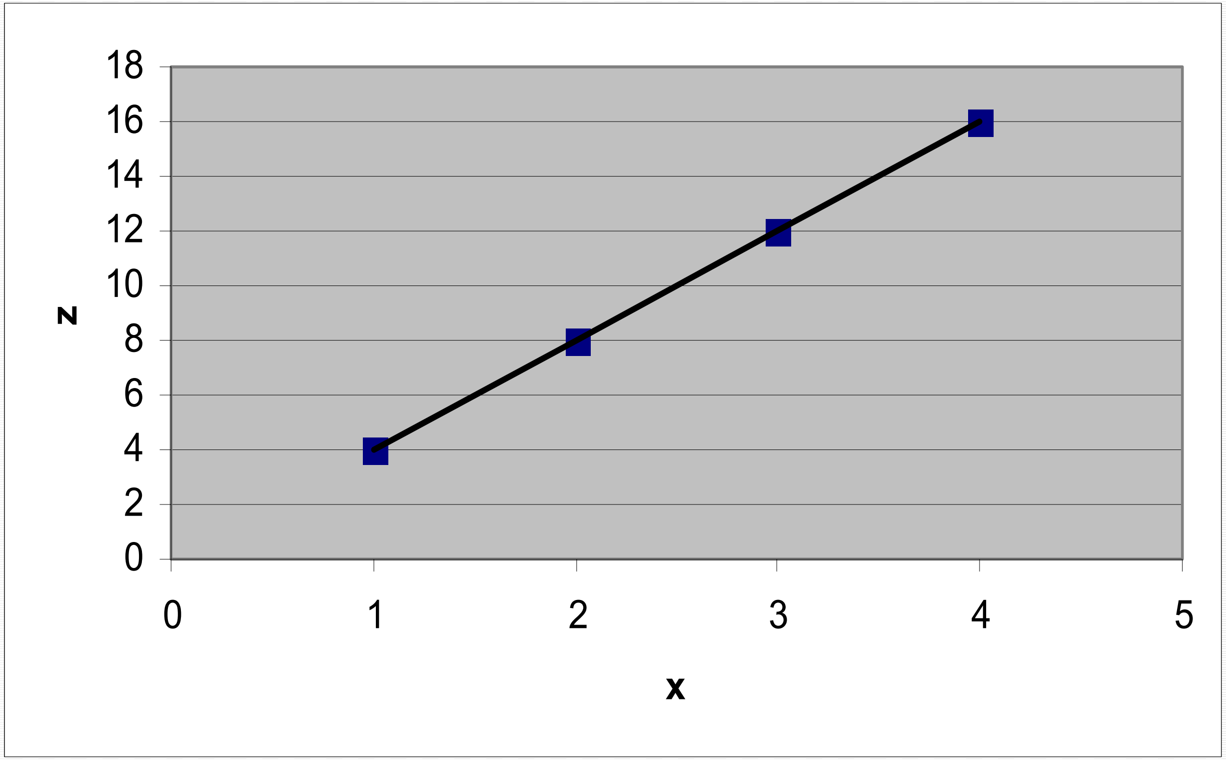


# Exemple 1: Additivité

2 systèmes homogènes et additifs

(y = 2 * x, z = 2 * y)

| **x** | **y** | **z** |
|---|---|---|
| 1 | 2 | 4 |
| 2 | 4 | 8 |
| 3 | 6 | 12 |
| 4 | 8 | 16 |

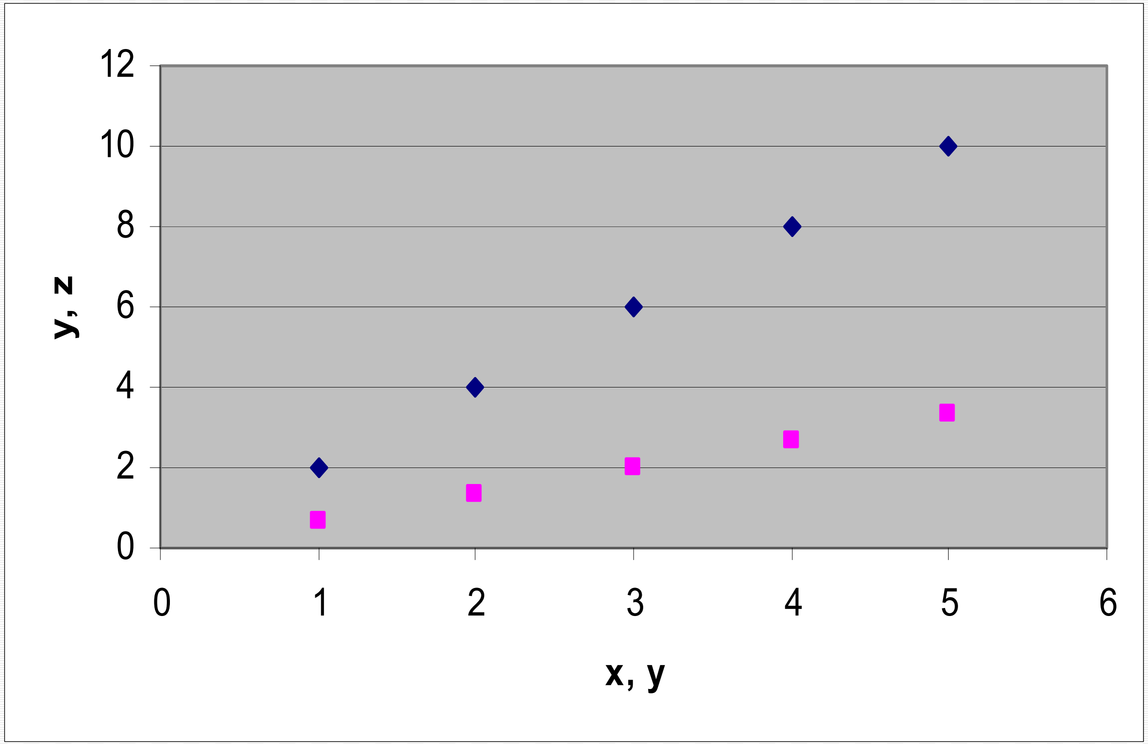


# Exemple 2: Additivité

2 systèmes homogènes et additifs

(y = 2 * x, z = 0.33 * y)

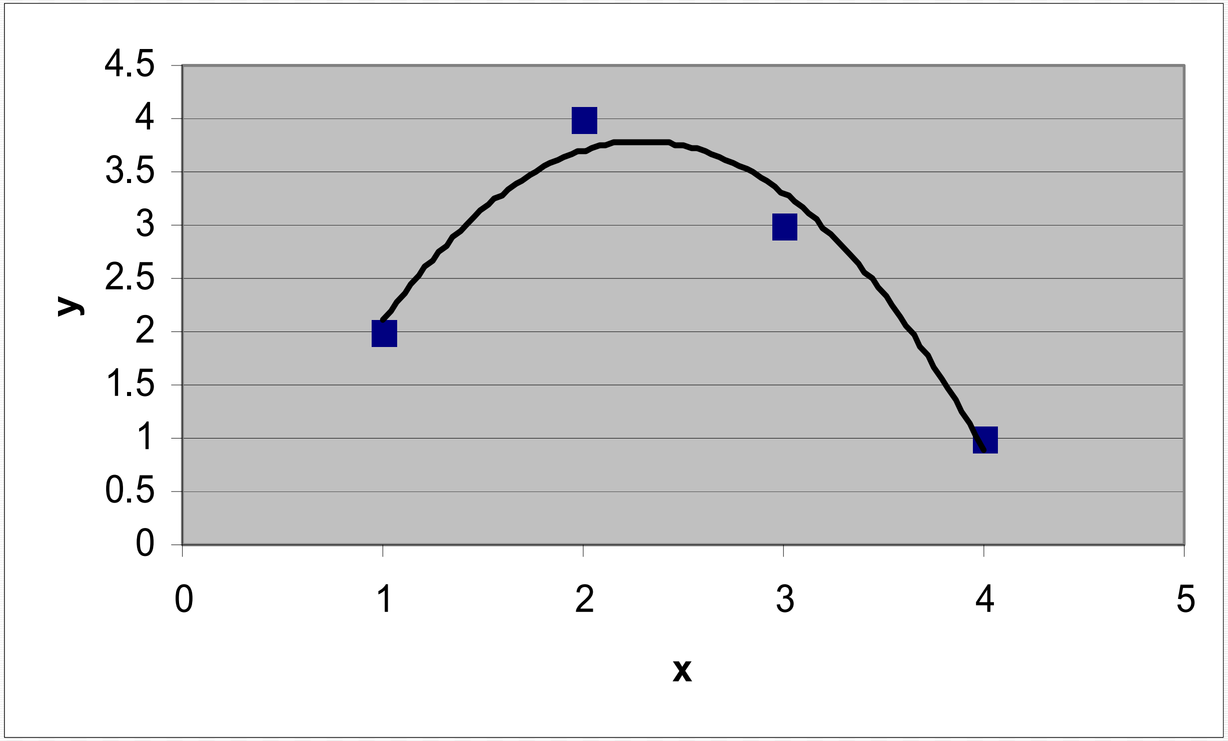
| **x** | **y** | **z** |
|---|---|---|
| 1 | 2 | 0.66 |
| 2 | 4 | 1.32 |
| 3 | 6 | 1.98 |
| 4 | 8 | 2.64 |
| 5 | 10 | 3.3 |



# Exemple: non linéarité

(pas de multiplicateur évident)

| x | y |
|---|---|
| 1 | 2 |
| 2 | 4 |
| 3 | 3 |
| 4 | 1 |



# Conclusions sur la linéarité

- Afin de savoir s’il est utile d’essayer une exploration neuromimétique…
  1. Examiner l’homogénéité de chaque sous système (typiquement par une représentation graphique de x sur y, mais aussi par une corrélation appropriée).
  2. Examiner la possibilité de “linéariser” la relation x sur y par l’application d’une fonction (p.ex. racine carrée, log) avant d’entamer une solution neuromimétique.
  3. Examiner l’additivité des multiples sous systèmes (typiquement par une représentation graphique de x sur z, mais aussi par le calcul d’une corrélation appropriée).