## 2.2 Modélisation par optimisation sans contrainte

La modélisation par apprentissage consiste à trouver le jeu de paramètres qui conduit à la meilleure approximation possible de la fonction de régression, à partir des couples entrées / sorties constituant l'ensemble d'apprentissage. Le plus souvent, ces couples sont constitués d'un ensemble de vecteurs de variables (descripteurs dans le cas de molécules) $\{x^i, i = 1...N\}$, et d'un ensemble de mesures de la grandeur à modéliser $\{y(x^i), i = 1...N\}$.

La détermination des valeurs de ces paramètres nécessite la mise en œuvre de méthodes d'optimisation qui diffèrent selon le type de modèle choisi. Nous allons tout d'abord présenter les principaux types de modèles faisant appel à l'optimisation paramétrique, sans contrainte, qui consiste à déterminer les paramètres optimaux par minimisation directe d'une fonction de coût par rapport aux paramètres du modèle.

### 2.2.1 Réseaux de neurones

Les réseaux de neurones formels [38] étaient, à l'origine, une tentative de modélisation mathématique des systèmes nerveux, initiée dès 1943 par McCulloch et Pitts [39].

Un ***neurone formel*** est une fonction non linéaire paramétrée, à valeurs bornées, de variables réelles. Le plus souvent, les neurones formels réalisent une combinaison linéaire des entrées reçues, puis appliquent à cette valeur une « fonction d'activation » *f*, généralement non linéaire. La valeur obtenue y est la sortie du neurone. Un neurone formel est ainsi représenté sur la Figure 2.1.

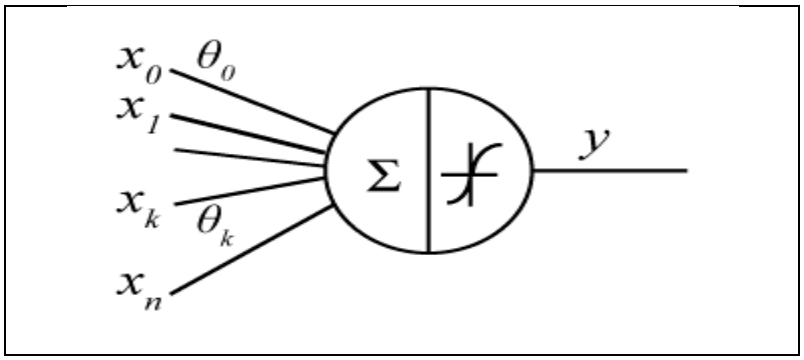


**Fig.2.1 : Représentation d'un neurone formel**

Les $\{x_k\}_{k=1\ldots n}$ sont les variables, ou ***entrées*** du neurone, et les $\{\theta_k\}_{k=0\ldots n}$ sont les ***paramètres***, également appelés synapses ou poids. Le paramètre $\theta_0$ est le paramètre associé à une entrée fixée à 1, appelée biais. L'équation du neurone est donc :

$$y = f(\theta_0 + \sum_{k=1}^{n} \theta_k x_k) \qquad (1)$$

Les fonctions d'activation les plus couramment utilisées sont la fonction tangente hyperbolique, la fonction sigmoïde et la fonction identité.

Les neurones seuls réalisent des fonctions assez simples, et c'est leurs compositions qui permettent de construire des fonctions aux propriétés particulièrement intéressantes. On appelle ainsi ***réseau de neurones*** une composition de fonctions « neurones » définies ci-dessus.

La Figure 2.2 représente un réseau de neurones non bouclé, organisé en couches (perceptron multicouche), qui comporte $N_e$ variables, une couche de $N_c$ neurones cachés, et $N_s$ neurones de sortie.

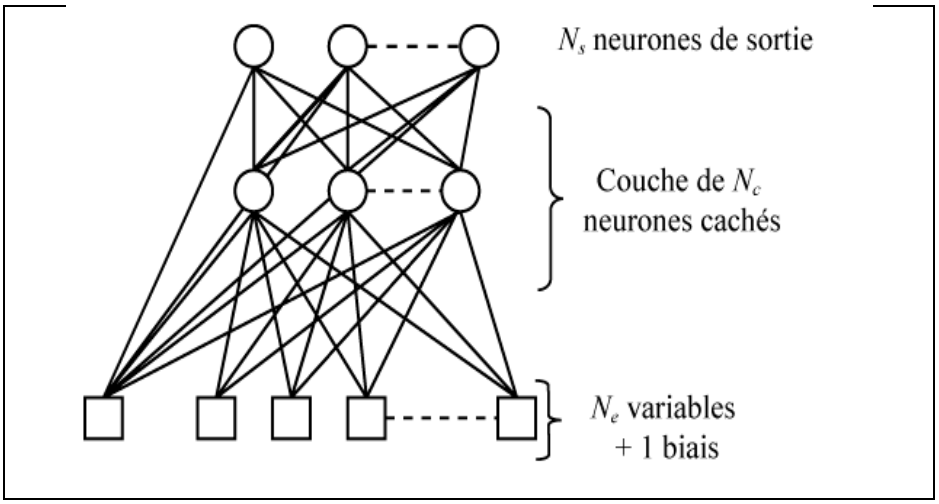


**Fig.2.2 : Représentation d'un réseau de neurones**

À chaque connexion est associé un paramètre. Les sorties du réseau sont donc des fonctions non-linéaires de ses variables et de ses paramètres. Le nombre de degrés de liberté, c'est-à-dire de paramètres ajustables, dépend du nombre de neurones de la couche cachée ; il est donc possible de faire varier la complexité du réseau en augmentant ou en diminuant le nombre de neurones cachés.

### 2.2.1.1 Propriétés des réseaux de neurones

Les réseaux de neurones ont pour but de modéliser des processus, à partir d'exemples de couples entrées / sorties. Ils ont la propriété d'***approximation universelle*** : un réseau de neurones comportant un nombre fini de neurones cachés, de même fonction d'activation, et un neurone de sortie linéaire, est capable d'approcher uniformément, avec une précision arbitraire, toute fonction bornée suffisamment régulière, sur un domaine fini de l'espace de ses variables. De plus, il s'agit d'***approximateurs parcimonieux*** : une approximation par un réseau de neurones nécessite en général moins de paramètres que les approximateurs usuels.
Le nombre de paramètres nécessaires pour obtenir une précision donnée augmente en effet linéairement avec le nombre de variables pour un réseau de neurones, alors qu'il croît exponentiellement pour un modèle linéaire par rapport aux paramètres. Cette propriété est très importante, car les réseaux de neurones demandent de ce fait moins d'exemples que d'autres approximateurs pour l'apprentissage.

### 2.2.1.2 Apprentissage des réseaux de neurones

Considérons un ensemble d'apprentissage, constitué de *N* couples entrées / sorties, c'est-à- dire d'un ensemble de variables $\{\mathbf{x}^i, i = 1...N\}$ et d'un ensemble de mesures de la grandeur à modéliser $\{y(\mathbf{x}^i), i = 1...N\}$. Pour une complexité donnée (le choix de cette complexité est étudié dans la section 2.2.2 de ce chapitre), l'apprentissage s'effectue par minimisation de la fonction de coût des moindres carrés, définie par :

$$j(\theta) = \frac{1}{2} \sum_{i=1}^{N} [y(x^i) - g(x^i, \theta)]^2 \qquad (2)$$

La minimisation de cette fonction s'effectue par une descente de gradient. Cet algorithme a pour but de converger, de manière itérative, vers un minimum de la fonction de coût, à partir de valeurs initiales des poids aléatoires. À chaque étape, le gradient de la fonction est calculé, à l'aide de l'algorithme de ***rétropropagation***. Puis les paramètres sont modifiés en fonction de ce gradient, dans la direction de la plus forte pente, vers un minimum local de *J*. Cette descente peut être effectuée suivant plusieurs méthodes : gradient simple ou méthodes du second ordre, dérivées de la méthode de Newton. Les méthodes du second ordre, généralement plus efficaces, sont les plus utilisées. La procédure de minimisation est arrêtée

lorsqu'un critère est satisfait : le nombre maximal d'itérations est atteint, la variation du module du vecteur des paramètres ou du gradient de la fonction de coût est trop faible…

### 2.2.2 Sélection du modèle

La modélisation vise à fournir un modèle qui soit non seulement ajusté aux données d'apprentissage, mais aussi capable de prédire la valeur de la sortie sur de nouveaux exemples, c'est-à-dire de généraliser.

Soit $R_i$ l'erreur commise par le modèle considéré sur l'exemple $i$ (également appelé résidu) :

$$R_i = y(x^i) - g(x^i, \theta) \quad (3)$$

où $y$ ($x^i$) est la valeur mesurée de la grandeur à modéliser pour l'exemple $i$, et g($x^i$ ,θ) est l'estimation du modèle pour ce même exemple. L'erreur du modèle sur les données d'apprentissage, $E_A$ , peut être évaluée par l'erreur quadratique moyenne en apprentissage, appelée également EQMA :

$$E_A = \sqrt{\frac{1}{N_A}\sum_{i=1}^{N_A}(R_i)^2} \quad (4)$$

où $N_A$ est le nombre d'exemples servant à établir le modèle. La qualité du modèle en apprentissage est souvent visualisée sur un *diagramme de dispersion*, sur lequel sont portées les valeurs de la grandeur d'intérêt estimées par le modèle, en fonctions des valeurs mesurées de cette grandeur. La qualité de la modélisation est d'autant meilleure que les points de ce graphique sont proches de la première bissectrice. L'ajustement des points à cette droite peut être évalué par le ***coefficient de détermination*** :

$$R^2 = \frac{\sum_{i=1}^{N_A}(y^i - \overline{y})^2}{\sum_{I=1}^{N_A}(g(x^i,\theta) - \overline{y})^2} \quad (5)$$

Où $\overline{y}$ est la moyenne des valeurs mesurées. Ce coefficient est égal au rapport de la variance expliquée à la variance totale de la sortie. Plus il est proche de 1, plus la corrélation entre les

valeurs mesurées et prédites est forte. Les valeurs intermédiaires renseignent sur le degré de dépendance linéaire entre les deux variables.

L'erreur sur la base d'apprentissage seule n'est pas un bon indicateur de la qualité du modèle ; un bon modèle doit être capable de rendre compte de la relation déterministe entre les variables et la grandeur modélisée, sans s'ajuster au bruit des données d'apprentissage. Il convient donc de trouver comment estimer cette capacité.

#### 2.2.2.1 Méthode générale de sélection

Lors de la modélisation, nous recherchons à la fois la complexité la mieux adaptée au problème étudié, et les paramètres du modèle correspondant. Le modèle doit en effet être de complexité suffisante pour rendre compte de la relation entre les variables explicatives et la grandeur modélisée, mais il ne doit pas être trop complexe, afin de ne pas être trop sensible au bruit présent dans les données. En effet, lorsque la complexité du modèle envisagé augmente, on observe généralement une diminution du coût d'apprentissage, car le modèle s'ajuste de lus en plus précisément aux données d'apprentissage. En revanche, l'erreur de généralisation diminue pour atteindre un minimum, puis augmente avec la complexité du modèle, car celui-ci est alors surajusté aux données d'apprentissage, et au bruit présent dans ces données. Ce compromis est également connu sous le nom de dilemme biais-variance : le biais caractérise l'écart entre les estimations et les mesures, et la variance reflète l'influence du choix de la base d'apprentissage sur le modèle **[40]**. De plus, lorsque les modèles envisagés ne sont pas linéaires par rapport à leurs paramètres, la sélection du modèle optimal ne consiste pas uniquement à choisir la meilleure famille de fonctions, comme c'est le cas pour des modèles linéaires (modèles polynomiaux ou combinaisons linéaires de fonctions non paramétrées). La sélection s'effectue donc souvent en plusieurs étapes.

- Pour les modèles non linéaires en leurs paramètres, la première sélection s'effectue au sein d'une famille de fonctions donnée. En effet, la fonction de coût n'étant pas quadratique en les paramètres du modèle, elle ne possède pas un minimum unique.
  L'optimisation de cette fonction peut donc conduire à des minima différents, donc à différents modèles de même complexité. Il est donc nécessaire de réaliser plusieurs apprentissages, pour une complexité donnée, et de choisir celui qui est susceptible de posséder les meilleures propriétés de généralisation.

- Il faut ensuite sélectionner, parmi les meilleurs modèles de complexités différentes, celui qui présente les meilleures capacités de généralisation.

Nous allons maintenant détailler les critères de sélection les plus fréquemment utilisés, en étudiant dans un premier temps comment l'erreur de généralisation peut être évaluée.

### 2.2.2.2 Sélection du modèle par estimation de l'erreur de généralisation

L'erreur de généralisation ne peut pas être évaluée exactement ; la base de données disponible est en effet de taille limitée, et la distribution de probabilité des données, dont dépend cette erreur, est généralement inconnue. Il est donc nécessaire de trouver une approximation de cette erreur de généralisation. Nous allons ainsi présenter les méthodes les plus utilisées pour effectuer ces sélections.

La méthode la plus commune, dite méthode de validation simple ou hold-out, consiste à construire, à partir de l'ensemble des données, deux ensembles : les paramètres du modèle sont ajustés sur la ***base d'apprentissage***, et l'erreur de généralisation est évaluée sur la ***base de validation.***

L'erreur en validation est définie de façon analogue au coût d'apprentissage :

$$E_V = \sqrt{\frac{1}{N}\sum_{i=1}^{N_V}(R_i)^2} \qquad (6)$$

où NV est le nombre d'exemples dans la base de validation.

Pour des modèles de complexité donnée, la méthode de sélection consiste à effectuer plusieurs apprentissages à partir de différentes initialisations des paramètres, lorsque l'on a affaire à un modèle non linéaire en ses paramètres. Il faut alors choisir, parmi les modèles obtenus, celui qui fournit la plus faible erreur en validation. Il s'agit d'une méthode très rapide, mais d'efficacité limitée, surtout lorsque le nombre d'exemples disponible est faible. En effet, si le nombre d'exemples est petit, la variance de l'estimation de l'erreur de généralisation est grande ; le coût de validation dépend alors fortement du choix de la base de validation [41].

De plus, les exemples de la base de validation ne sont pas utilisés pour l'estimation des paramètres du modèle, et peuvent faire défaut : le modèle n'étant pas ajusté à certains types d'exemples, ses capacités de généralisation sont limitées. Par conséquent, l'erreur de

généralisation est plus souvent estimée à l'aide d'autres méthodes, telles que la validation croisée ou le leave-one-out.

#### 2.2.2.2.1 La validation croisée

L'ensemble des $N$ exemples disponibles est cette fois-ci divisé en $K$ sous-ensembles disjoints. La valeur $K = 10$ est souvent retenue, mais il n'y a pas de méthode de choix optimal de $K$. On construit alors une base d'apprentissage à partir de $K - 1$ de ces sous ensembles.
Le modèle est ajusté sur cette base, puis l'erreur de prédiction $R_i$ est calculée pour chaque exemple du sous-ensemble restant. Cette étape est réalisée $K$ fois, en choisissant un sous-ensemble de validation différent à chaque itération. Ainsi, chaque exemple se trouve une fois et une seule dans une base de validation. Cette procédure est illustrée sur la Figure 2.3, où $K = 5$. Les sous-ensembles $E_1 \ldots E_5$ constituent successivement la base de validation.

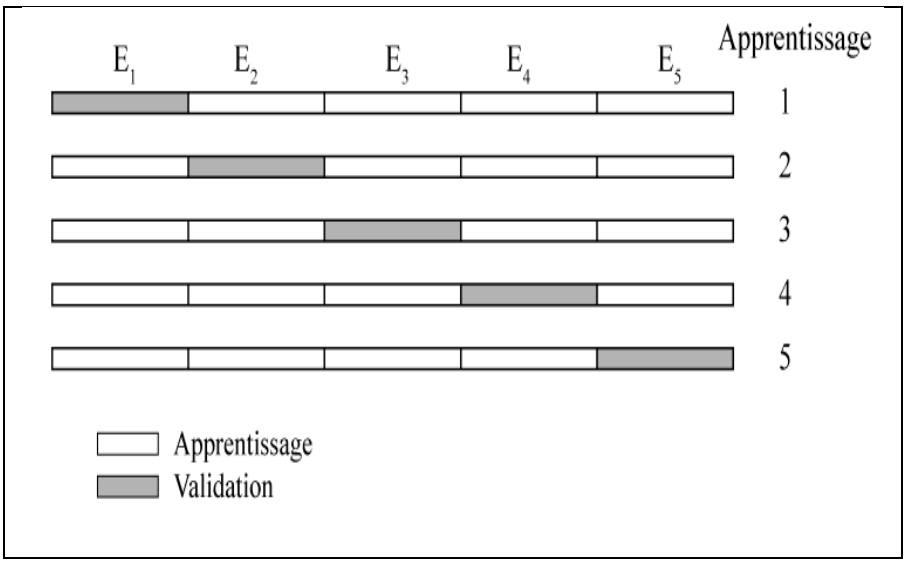


**Fig.2.3 : Principe de la validation croisée.**

La performance en généralisation de la famille de modèles est ensuite estimée en calculant le score de validation croisée :

$$\sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i)^2} \qquad (7)$$

Cette technique a l'avantage d'être moins sensible au choix des bases de validation (la variance de l'erreur est plus faible), et permet d'utiliser toutes les données en apprentissage.

Elle nécessite cependant d'effectuer $K$ apprentissages, qui produisent ainsi $K$ modèles différents. Une fois la meilleure complexité déterminée, on effectue un apprentissage avec l'ensemble des données disponibles.

#### 2.2.2.2.2 Méthode du leave-one-out

Le cas particulier où $K = N$ ($N$ est le nombre d'exemples disponibles) est appelé ***leave-one-out*** : à chaque itération, un exemple $i$ est extrait de l'ensemble d'apprentissage. Une série d'apprentissage est réalisée à l'aide des $N - 1$ exemples restants, et l'erreur de prédiction sur l'exemple $i$ est calculée pour chacun des modèles obtenus. On retient l'erreur la plus faible, notée $R_i^{-i}$ . Le score de leave-one-out est alors défini par :

$$E_t = \sqrt{\frac{1}{N}\sum_{i=1}^{N}(R_i^{-i})^2} \qquad (8)$$

Ce score $E_t$ est un estimateur non-biaisé de l'erreur de généralisation [42]. Cette méthode présente ainsi l'avantage de tirer pleinement profit des données disponibles, et d'être indépendante du choix de bases de validation. Cependant, le temps de calcul peut devenir très grand, car il est nécessaire de procéder à $N$ séries d'apprentissage.

### 2.2.2.3 Le leave-one-out virtuel

La méthode du leave-one-out virtuel [43] consiste à estimer les erreurs $R_i^{-i}$ à partir d'un seul apprentissage réalisé sur l'ensemble des $N$ exemples, ce qui permet d'estimer le score de leave-one-out sans avoir à réaliser N apprentissages. Cette estimation est appelée score de ***leave-one-out virtuel***, et repose sur le calcul de la matrice jacobienne du modèle $g$ (x,θ*), obtenu à partir des $N$ exemples. Un développement de ce modèle au premier ordre par rapport aux paramètres, au voisinage de θ*, est :

$$g(x,\theta) \cong g(x,\theta^*) + Z(\theta - \theta^*) \qquad (9)$$

Ce développement fait apparaître **Z**, matrice jacobienne du modèle. Cette matrice est de taille ($N$, $q$), où $q$ est le nombre de paramètres du modèle. Les $N$ éléments de la colonne $j$ sont égaux aux dérivées partielles de la sortie par rapport au paramètre $j$. Les éléments de ***Z*** s'écrivent donc :

$$(Z)_{ij} = \left(\frac{\partial g(x_i,\theta_{mc})}{\partial \theta_j}\right)_{\theta=\theta_{mc}} \quad (10)$$

Les modèles dont la jacobienne n'est pas de rang plein, c'est-à-dire de rang inférieur à *q*, doivent être rejetés. En effet, cela signifie qu'au moins deux paramètres ont des effets qui ne sont pas indépendants sur la sortie. Ces modèles comportent donc trop de paramètres, et ont de grandes chances d'être surajustés.

Considérons la matrice de projection orthogonale sur le sous-espace défini par les colonnes de la matrice $\mathbf{Z}$ : $\boldsymbol{H} = \boldsymbol{Z}(\boldsymbol{Z^T Z})^{-1}\boldsymbol{Z^T}$ qui est une matrice carrée *(N, N)*.

Les termes diagonaux de cette matrice sont les ***leviers de plan tangent*** (appelés simplement leviers dans la suite du mémoire) des exemples. Le levier d'un exemple i est ainsi égal à :

$$h_{ii} = \boldsymbol{z}^{iT}(\boldsymbol{Z}^T\boldsymbol{Z})^{-1}\boldsymbol{z}^i \quad (11)$$

où $\mathbf{z}^i$ est le vecteur dont les composantes sont celles de la i-ième ligne de **Z**.

Le calcul de ces leviers permet d'estimer l'erreur de prédiction sur un exemple i lorsqu'il est retiré de la base d'apprentissage (c'est-à-dire $R_i^{-i}$ ), à partir de l'erreur commise sur cet exemple lorsqu'il est dans cette base (c'est-à-dire $R_i$) :

$$R_i^{-i} \cong \frac{R_i}{1-h_{ii}} \quad (12)$$

Ce résultat est d'ailleurs exact si le modèle est linéaire ; il est alors appelé PRESS (Predicted REsidual Sum of Squares).

Dès lors, il est possible de calculer le score de leave-one-out virtuel, qui constitue une très bonne approximation de l'erreur de généralisation si le développement limité au premier ordre sur lequel elle est fondée est suffisamment précis :

$$E_p = \sqrt{\frac{1}{N}\sum_{i=1}^{N}\left(\frac{R_i}{1-h_{ii}}\right)^2} \quad (13)$$

Ce calcul permet non seulement d'estimer rapidement l'erreur de généralisation, mais également de révéler d'éventuels surajustements du modèle à des exemples particuliers. Il

peut ainsi constituer la base d'une méthode de planification expérimentale : s'il est possible d'enrichir la base de données de mesures supplémentaires, celles-ci doivent être effectuées au voisinage des points qui ont des leviers importants.

#### 2.2.2.4 Sélection de modèle à l'aide des leviers:

##### 2.2.2.4.1 Interprétation des leviers:

Nous avons vu que les leviers sont les termes diagonaux de la matrice de projection orthogonale sur le sous-espace des colonnes de la matrice jacobienne. Lorsque la matrice **Z** est de rang plein, les leviers vérifient les propriétés suivantes :

$$\begin{cases} \sum_{i=1}^{N} h_{ii} = q \\ 0 < h_{ii} < 1 \quad \forall i \in [\![1, N]\!] \end{cases} \tag{14}$$

où *q* est le nombre de paramètres du modèle. Le levier $h_{ii}$ relatif à un exemple *i* peut ainsi être interprété comme la proportion des paramètres utilisée par le modèle pour s'ajuster à l'exemple *i*. Par conséquent, si tous les exemples ont la même influence sur le modèle, les leviers sont tous égaux à $\frac{q}{N}$. Si le levier d'un exemple est proche de 1, le modèle est particulièrement ajusté sur cet exemple, et il est presque parfaitement appris. Par contre, l'erreur en prédiction sur cet exemple lorsqu'il n'appartient pas à la base d'apprentissage, qui peut être estimée par $R_i^{-i} \cong \frac{R_i}{1-h_{ii}}$, est très grande. En revanche, un exemple dont le levier est proche de 0 n'a aucune influence sur le modèle. Si un modèle consacre une grande partie de ses degrés de liberté à un ou plusieurs exemples, il doit être rejeté, car il est susceptible d'être ajusté au bruit présent dans ces mesures [44]. Les leviers constituent donc un outil supplémentaire de sélection.

##### 2.2.2.4.2 Calcul des intervalles de confiance

Les leviers permettent également le calcul des intervalles de confiance sur les prédictions du modèle. Un intervalle de confiance au seuil de confiance $(1-\alpha)$ est un intervalle tel qu'on peut dire qu'il contient la vraie valeur de la grandeur prédite avec une probabilité $(1-\alpha)$. L'étendue de l'intervalle de confiance caractérise la confiance que l'on peut avoir dans la prédiction faite par le modèle : plus l'étendue est faible, plus cette confiance est grande.

Lorsque le modèle est non-linéaire, un intervalle de confiance approché pour l'espérance mathématique de la sortie $Y_p$ peut être calculé à l'aide des leviers.
Pour un exemple *i* de la base d'apprentissage, son expression est :

$$E\left(Y_p \middle| x^i\right) \in g\left(x^i, \theta\right) \pm t_{\propto}^{N-q} \, s\sqrt{z^{iT}(Z^T Z)^{-1} z^i} = g\left(x^i, \theta\right) \pm t_{\alpha}^{N-q} \, s\sqrt{h_{ii}} \qquad (15)$$

Où g($\mathbf{x}^i$ ,$\boldsymbol{\theta}$) est la prédiction du modèle pour cet exemple, $t_{\propto}^{N-q}$ est la valeur d'une variable de Student à *N*-*q* degrés de liberté et un niveau de confiance (1−α), et *s* une estimation de la variance de l'erreur de prédiction de modèle.
L'intervalle de confiance peut également être estimé pour un exemple qui n'appartient pas à la base d'apprentissage :

$$E^{(-i)}\left(Y_p \middle| x^i\right) \in \; g\left(x^i, \theta\right) \pm t_{\propto}^{N-q-1} \; s^{-i}\sqrt{\frac{h_{ii}}{1-h_{ii}}} \qquad (16)$$

Cet intervalle est donc d'autant plus large que $h_{ii}$ est proche de 1 : dans ce cas, la prédiction sur cet exemple est très peu fiable.
Les leviers se révèlent donc être un outil efficace pour détecter le surajustement de modèles à des exemples particuliers, et pour repérer ces exemples. Ils permettent la sélection d'un modèle parmi des modèles de complexités différentes, lorsque leurs performances en généralisation sont comparables : le meilleur est généralement celui pour lequel la distribution des leviers des exemples autour de $\frac{q}{N}$ est la plus étroite. La sélection de modèle peut donc se dérouler de la façon suivante :

- Pour des modèles de complexité donnée, on réalise plusieurs apprentissages, à partir d'initialisations différentes.
- Les modèles dont la matrice jacobienne n'est pas de rang plein sont écartés ; on calcule alors les scores d'apprentissage et de leave-one-out virtuel des modèles obtenus. Le modèle présentant les meilleures capacités de généralisation est retenu.
- Cette sélection est effectuée pour des modèles de complexité croissante. On compare alors les modèles retenus pour chaque complexité, c'est-à-dire leurs scores d'apprentissage et de leave-one-out virtuel, ainsi que la répartition des leviers des exemples. Ces critères permettent de sélectionner le meilleur modèle.

Les performances de ce modèle peuvent alors être mesurées sur une base de test : elle est composée d'exemples n'ayant pas servi à établir ni à choisir le modèle. Le score de test permet ainsi d'évaluer les performances du modèle en généralisation.

## 2.3 Autres méthodes de QSPR/QSAR

La modélisation d'une propriété ou d'une activité moléculaire nécessite de disposer d'informations caractérisant les molécules, informations à partir desquelles la grandeur en question est prédite. Il peut s'agir de descripteurs, mais il existe des méthodes alternatives de caractérisation des molécules. Nous présenterons dans un premier temps la méthode de contribution de groupes, qui, bien que datant des débuts de la modélisation QSAR, est toujours utilisée pour des applications particulières. Nous décrirons ensuite comment les techniques de calcul et de comparaison de champs moléculaires se révèlent particulièrement efficaces pour la modélisation d'activités biologiques. Nous introduirons finalement des méthodes récentes qui, de façon analogue aux *graph machines*, considèrent que les structures des molécules contiennent des informations dont il est possible de tirer directement profit pour la modélisation.

### 2.3.1 Méthode de contribution de groupes

Les méthodes de contribution de groupes consistent à évaluer une propriété en décomposant la molécule en un ensemble de groupes fonctionnels, et en sommant les contributions relatives à des fragments de molécules **[45,46]**. Ces contributions sont déterminées à partir d'une base d'exemples de molécules, dont les valeurs de la propriété sont connues. Plusieurs types de groupes fonctionnels peuvent être définis. Ils sont généralement organisés en un système hiérarchique :

- Les ***groupes d'ordre 0*** sont des atomes, et le calcul d'une propriété est effectué en sommant les contributions de chacun des atomes de la molécule considérée.
- La décomposition en ***groupes d'ordre 1*** consiste à découper la molécule en groupes 'atomes (tels que -CH2-, -CH3 ou -OH). Leurs contributions à une propriété donnée sont sommées sans que l'environnement de chacun des groupes dans la molécule ne soit pris en considération. Ainsi, le groupe -$CH_2$- a une contribution fixe, qu'il soit relié à un carbone ou à un groupe oxygéné. Ces groupes sont assez souvent employés,

car ils permettent d'estimer rapidement la valeur d'une propriété, avec une précision parfois suffisante (par exemple pour l'enthalpie de formation). Cependant, les résultats obtenus, pour la température d'ébullition par exemple, ne sont pas toujours satisfaisants. De plus, certains isomères peuvent conduire à la même décomposition : il est alors impossible de les distinguer par cette méthode.

- ***Les groupes d'ordre 2*** **[47,48]** sont constitués des atomes centraux de la molécule (autres que H), accompagnés de leurs plus proches voisins, c'est-à-dire de tous les atomes auxquels ils sont reliés. Contrairement aux groupes d'ordre 1, ceux d'ordre 2 tiennent compte de l'environnement des atomes.

Le Tableau 1 présente une comparaison des décompositions des molécules de butan -2-ol et 2-

Méthylpropan -1-ol, représentées sur la Figure 2.4.

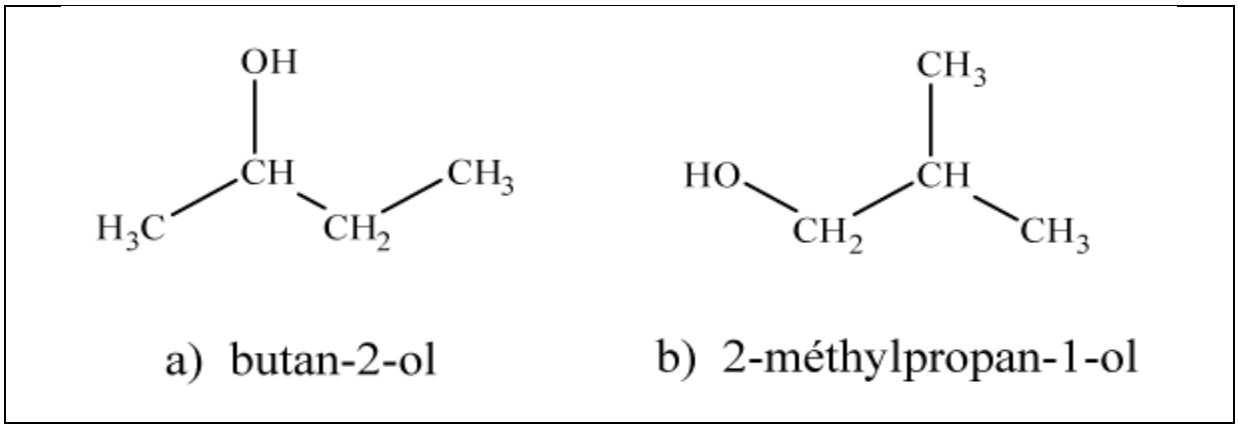


**Fig.2.4 : Exemple de deux molécules, isomères de constitution**

| Méthode | Groupe | Nombre de groupes a) | Nombre de groupes b) |
|---|---|---|---|
| **Ordre 0** | C | 4 | 4 |
| | H | 10 | 10 |
| | O | 1 | 1 |
| **Ordre 1** | CH | 1 | 1 |
| | $-CH_2-$ | 1 | 1 |
| | $-CH_3$ | 2 | 2 |
| | -OH | 1 | 1 |
| **Ordre 2** | $C-(C)(H)_3$ | 2 | 2 |
| | $C-(C)_2(H)_2$ | 1 | 0 |
| | $C-(C)(O)(H)_2$ | 0 | 1 |
| | $C-(C)_3(H)$ | 0 | 1 |
| | $C-(C)_2(O)(H)$ | 1 | 0 |

**Tableau 1 : Décomposition en groupes de deux isomères de constitution**

On observe que seule la décomposition en groupes d'ordre 2 permet de distinguer ces deux molécules.

De nombreuses méthodes s'appuient donc sur des groupes des trois ordres pour améliorer la précision des prédictions et différencier les isomères. Elles sont principalement utilisées pour prédire des propriétés thermodynamiques, par exemples des propriétés critiques (température ou pression critique) et de nombreuses grandeurs énergétiques. Elles présentent également l'avantage de permettre l'estimation de propriétés de mélanges, par addition des contributions des composants du mélange.

### 2.3.2 Analyse comparative de champs moléculaires (CoMFA)

La méthode CoMFA (pour Comparative Molecular Field Analysis) a été développée à partir de 1988 par Cramer **[49]**. Elle est en particulier utilisée pour la modélisation d'activités biologiques, pour lesquelles les méthodes classiques se révèlent parfois peu performantes.

L'activité biologique d'une molécule dépend généralement de son interaction avec un récepteur donné. La modélisation de cette activité peut donc être réalisée en calculant les

interactions de chaque molécule (ligand) avec ce récepteur, et en établissant une relation entre ces interactions et l'activité étudiée. Cette modélisation, qui repose sur le calcul de potentiels d'interactions moléculaires, s'effectue en plusieurs étapes.

#### 2.3.2.1 Recherche des conformations moléculaires les plus stables

Des programmes tels que Concord ou Corina **[50]** permettent d'accéder à une conformation de basse énergie, à partir de laquelle sont créées plusieurs conformations bioactives.

#### 2.3.2.2 Alignement des ligands

Les potentiels d'interactions moléculaires dépendent des orientations des structures utilisées pour leur calcul. La comparaison des potentiels de différentes molécules nécessite donc un alignement préalable de ces structures. Il s'agit d'une étape assez délicate, en particulier lorsque les composés ont des structures diverses. Cette étape est même susceptible de limiter les possibilités d'application de la méthode. L'alignement est réalisé par rapport aux groupes fonctionnels identifiés comme potentiellement pharmacophores. Il existe plusieurs méthodes pour réaliser cet alignement. La plus simple est la méthode de superposition par sous-structures, qui consiste à superposer les molécules qui partagent un squelette commun, selon ce squelette. L'approche par superposition de pharmacophores ne nécessite pas cette supposition, mais part du fait que les pharmacophores de chaque molécule sont identifiés. Il s'agit ensuite de maximiser la superposition de ces groupements entre les molécules. Il est également possible d'aligner les molécules par rapport à leurs moments dipolaires ou à leurs champs électrostatiques.

#### 2.3.2.3 Calcul des champs électrostatiques et stériques

Les champs électrostatiques et stériques sont ensuite évalués. Pour cela, on définit autour de chaque molécule, préalablement alignée, une grille 3D. Puis on calcule, en chaque point de la grille, l'énergie d'interaction de la molécule avec un atome sonde.

#### 2.3.2.4 Corrélation entre les champs et les activités biologiques

Les valeurs calculées des champs peuvent alors être utilisées comme variables du modèle. Le nombre de ces variables peut être très grand (quelques milliers), lorsque la résolution de la grille est fine par rapport à la taille des molécules. Il n'est alors pas possible de corréler directement ces valeurs de champs avec l'activité étudiée. La modélisation est généralement effectuée par la méthode des moindres carrés partiels, et le nombre optimum de variables final déterminé par validation croisée.

#### 2.3.2.5 Visualisation graphique des résultats

Une visualisation graphique des résultats permet enfin de situer autour d'une molécule les régions pour lesquelles une substitution augmente ou diminue l'affinité envers le récepteur. Cette méthode a montré son efficacité en QSAR, en particulier pour la modélisation d'interactions ligand-protéine. Elle permet en effet de décrire efficacement les interactions ligand-récepteur, car les propriétés des ligands sont calculées à partir de leurs conformations bioactives. Les problèmes sont néanmoins multiples :

- Les résultats de modélisation dépendent de la conformation bioactive choisie. Or, la recherche de cette dernière est parfois difficile, en particulier lorsque la molécule est flexible : il existe alors un nombre important de conformations possibles près du minimum d'énergie.
- Il n'existe pas de règle générale pour l'alignement des molécules. Il s'agit d'un défaut majeur, car le modèle est très sensible à cet alignement.
- De plus, le temps de calcul est généralement assez long (30 à 60 min pour une vingtaine de molécules possédant de 30 à 50 atomes).

Puisque l'alignement des molécules est une limitation de la méthode CoMFA, de nouvelles techniques ne nécessitant pas cette étape ont été développées. La **méthode CoMSA** (pour Comparative Molecular Surface Analysis), ne repose pas sur la comparaison de champs calculés en une série donnée de points, mais celle du potentiel électrostatique moyen de régions définies de la surface de la molécule. Ces régions sont nombreuses, et la méthode des cartes auto-organisatrices de Kohonen **[51]** permet de réduire la dimension des données tout en préservant leur topologie. Cette méthode est particulièrement adaptée pour transformer la surface tridimensionnelle de la molécule en une carte bidimensionnelle du potentiel électrostatique. La transformation de Kohonen permet en effet à la fois de diminuer la taille

des données, et de retrouver les données 3D à partir de leur représentation 2D. Les vecteurs obtenus sont alors analysés et corrélés à l'activité étudiée, par la méthode des moindres carrés partiels.

## 2.4 Conclusion

De nombreuses recherches ont été menées, au cours des dernières décennies, pour trouver la meilleure façon de représenter l'information contenue dans la structure des molécules, et ces structures elles-mêmes, en un ensemble de nombres réels appelés descripteurs ; une fois que ces nombres sont disponibles, il est possible d'établir une relation entre ceux-ci et une propriété ou activité moléculaire, à l'aide d'outils de modélisation classiques.

Dans ce chapitre nous avons commencé par la présentation des descripteurs moléculaires les plus courants, en commençant par les descripteurs les plus simples, qui nécessitent peu de connaissances sur la structure moléculaire, mais véhiculent peu d'informations. Nous avons vu ensuite comment les progrès de la modélisation moléculaire ont permis d'accéder à la structure 3D de la molécule, et de calculer des descripteurs à partir de cette structure.
Par la suite nous avons présenté les techniques de modélisation par apprentissage qui consiste à trouver le jeu de paramètres qui conduit à la meilleure approximation possible de la fonction de régression, à partir des couples entrées / sorties constituant l'ensemble d'apprentissage,

Ensuite des Méthodes de sélection de modèle qui vise à fournir un modèle qui soit non seulement ajusté aux données d'apprentissage, mais aussi capable de prédire la valeur de la sortie sur de nouveaux exemples (c.-à-d. généraliser) ont été présentées ainsi que d'autres méthodes de QSPR/QSAR tel que la méthode de contribution de groupes.

Les techniques alternatives que nous venons de décrire répondent donc généralement à des besoins précis, mais elles ont pour cette raison des domaines d'application limités. Nous montrerons dans le chapitre suivant que la méthode des *graph machines*, tout en s'affranchissant des inconvénients liés à l'utilisation de descripteurs, ne présentent pas cette limitation.

Les méthodes traditionnelles de modélisation que nous avons présentés, tels que les réseaux de neurones, réalisent pour la plupart une association entre un vecteur de nombres réels, qui constituent les variables du modèle, et un vecteur de sorties également réelles. Cependant, dans de nombreux problèmes de modélisation, les données se présentent sous la forme de structures dont il faut extraire un vecteur de réels si l'on veut avoir recours à des modèles classiques. Cette étape nécessite de sélectionner les données pertinentes relatives au problème et de les calculer à partir des structures ; elle se traduit par une perte d'information. Il semblerait donc plus pertinent de tirer directement profit de la structure des données, par l'intermédiaire d'un autre type de modèle, capable d'établir de façon directe une association entre ces structures et un vecteur de sorties. Les données structurées se représentent aisément sous la forme de graphes. Nous commencerons par présenter les graphes en tant qu'objets mathématiques, ensuite nous introduirons les mémoires récursives auto-associatives, premiers modèles à réaliser un codage de données structurées par apprentissage artificiel. Nous présenterons les *graph machines* ainsi que leur apprentissage. En fin un exemple de modélisations de propriétés physico-chimiques de molécule sera présenté.

## 3.1 Les graphes (définition et caractéristiques)

La théorie des graphes date du 18$^{ème}$ siècle, et s'est considérablement développée au 20$^{ème}$ siècle, car elle permet de résoudre de nombreux types de problèmes [52]. Nous allons présenter les bases de cette théorie, et les définitions essentielles relatives aux graphes.

### 3.1.1 Graphes simples

Un graphe simple G est un groupe {S, A} où :

- S est ensemble d'objets { $s_1$,$s_2$ ,...,$s_n$ }, appelés sommets du graphe ;
- A est un sous-ensemble de S × S, dont les éléments { $a_1$,$a_2$ ,...,$a_m$ } sont les arêtes du graphe.

Un graphe est ***connexe*** s'il est possible, à partir de n'importe quel sommet, de rejoindre tous les autres. L'arête $a_k = (s_i, s_j)$ est incidente aux nœuds $s_i$ et $s_j$, qui sont dits adjacents. Le degré d'un nœud est défini comme le nombre d'arêtes qui lui sont incidentes. Un ***cycle*** est une chaîne $(s_1, s_2, \ldots, s_k)$ dont le premier et le dernier sommet sont identiques et tous les autres sommets distincts. Si le graphe possède *m* arêtes $\{ a_1, a_2, \ldots, a_m \}$, on peut faire correspondre à tout cycle $\mu$ un vecteur $\mu = (\mu_1, \mu_2; \ldots\ldots \mu_m)$ tel que :

$$\mu_\iota = \begin{cases} 1 \; si \; ai \in \mu \\ 0 \; si \; ai \notin \mu \end{cases} \qquad (17)$$

On dit que *p* cycles $(\mu^1, \mu^2, \ldots, \mu^p)$ sont dépendants s'il existe entre leurs vecteurs associés une relation vectorielle de la forme :

$$\sum_{i=1}^{p} \lambda_i \mu^i = 0 \qquad (18)$$

telle que $(\lambda_1, \lambda_2, \ldots, \lambda_p) \in \mathbb{R}^p$ et $(\lambda_1, \lambda_2, \ldots, \lambda_p) \neq (0,0,\ldots,0)$.

On appelle ***distance*** entre deux sommets la longueur de la plus courte chaîne reliant ces sommets ; le ***diamètre*** d'un graphe est alors la plus grande distance entre les sommets d'un graphe.

### 3.1.2 Graphes orientés

Un ***graphe orienté*** est un groupe G = {S, A}, où A est un ensemble d'arcs de la forme $(s_i, s_j)$, pour lequel l'arc part de $s_i$ et arrive en $s_j$. Le degré entrant $d^-$ d'un sommet est le nombre d'arcs qui arrivent à ce sommet, et le degré sortant $d^+$ le nombre d'arcs qui en partent.

Un ***arbre*** est un graphe connexe sans cycle. Dans un arbre orienté, si les sommets $s_i$ et $s_j$ sont reliés par l'arc $(s_i, s_j)$, $s_i$ est ***parent*** du nœud $s_j$, qui est un ***enfant*** du nœud $s_i$.

Les nœuds qui possèdent à la fois des nœuds parents et enfants sont appelés des ***branches***. Un ***nœud racine*** est un nœud sans parent, tandis que les ***feuilles*** sont les nœuds sans enfant. Un arbre est un arbre *n*-aire si tous les nœuds de l'arbre ont au plus *n* successeurs.

Une ***arborescence*** {S, A, r} de racine r est un graphe {S, A} où r est un élément de S tel

que, pour tout sommet s, il existe un unique chemin d'origine r et d'extrémité s. On montre facilement que, dans une arborescence, la racine r n'admet pas de prédécesseur, et que tout sommet différent de r admet un seul prédécesseur.

La Figure 3.1, qui représente une arborescence, illustre les concepts que nous venons de définir.

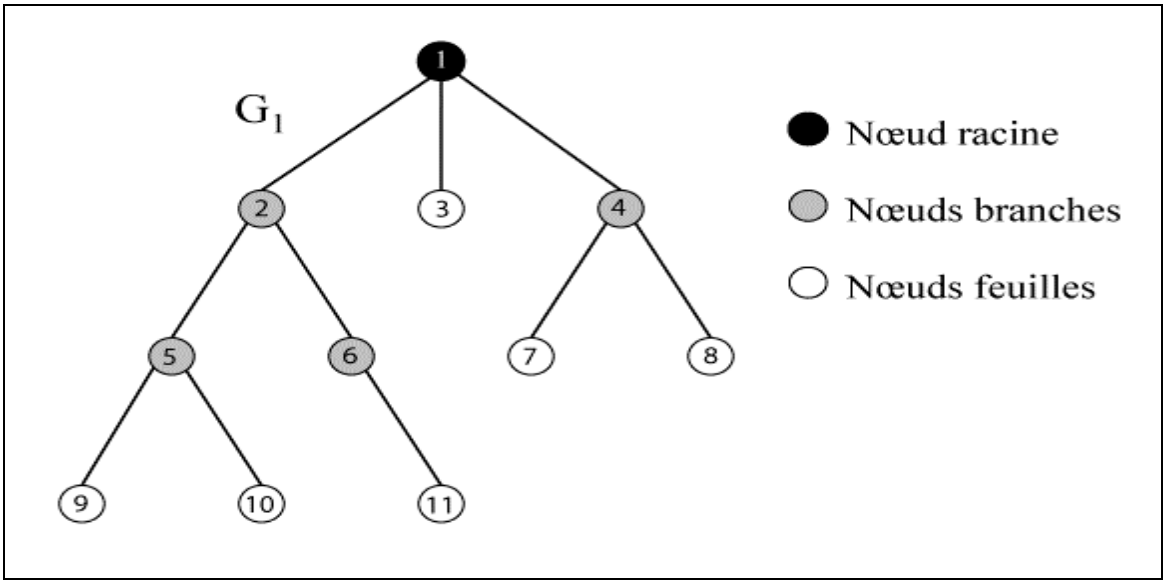


**Fig.3.1 : Exemple d'arborescence**

### 3.1.3 Graphes étiquetés

Il est possible d'affecter aux nœuds et aux arcs d'un graphe orienté un ensemble d'étiquettes : le graphe est alors ***étiqueté***. Étant donné un ensemble fini d'étiquettes de sommets $L_V$, et un ensemble fini d'étiquettes d'arcs $L_A$, un graphe étiqueté est défini par un triplet $(S, r_S, r_A)$ où :

- S est l'ensemble des sommets ;
- $r_S \subseteq S \times L_S$ est l'ensemble des couples $(s_i, l_s)$ tels que le sommet $s_i$ a pour étiquette $l_s$ ;
- $r_A \subseteq S \times S \times L_A$ est l'ensemble des triplets $(s_i, s_j, l_a)$ tels que l'arc (si ,s j) a pour étiquette $l_a$.