Master de Chimie · Modélisation moléculaire

TP de Chimiométrie / Machine Learning

Régression de la solubilité aqueuse par le perceptron linéaire (Adaline)

3 novembre 2026

1 Introduction et contexte chimique

En chimiométrie, les modèles **QSPR** (*Quantitative Structure–Property Relationship*) cherchent à prédire une *propriété quantitative* d'une molécule (solubilité, température d'ébullition, pK_a, etc.) directement à partir de descripteurs issus de sa structure. Contrairement à la classification (molécule active ou inactive), la sortie du modèle est ici une **valeur réelle**.

Dans ce TP, nous prédisons la **solubilité aqueuse** $\log S$ (en $\log_{10}$ mol / L) de molécules à l'aide d'un **perceptron linéaire**, aussi appelé **Adaline** (*Adaptive Linear Neuron*), entraîné par la règle de Widrow–Hoff.

Du perceptron de Rosenblatt au perceptron linéaire

Les deux modèles partagent la même architecture : une somme pondérée des descripteurs plus un biais. Ce qui change, c'est la fonction d'activation et la manière de mesurer l'erreur.

Critère	Rosenblatt (classification)	Linéaire (régression)
Sortie	étiquette binaire ± 1	valeur réelle $\hat{y} \in \mathbb{R}$
Activation	fonction seuil (signe de z)	identité : $f(z) = z$
Erreur	$y - y_{\text{pred}} \in \{-2,\ 0,\ +2\}$	$e = y - \hat{y} \in \mathbb{R}$ (continue)
Mise à jour	seulement en cas d'erreur de classification	à chaque molécule, proportionnelle à l'erreur
Exemple	molécule active / inactive	$\log S$, $T_{\text{éb}}$, pK_a…

1.1 Descripteurs moléculaires et propriété cible

Nous conservons les 3 descripteurs physico-chimiques normalisés du TP précédent :

1. x_1 **: le coefficient de partage lipophile** ($\log P$) $\rightarrow$ mesure l'hydrophobie de la molécule.
2. x_2 **: la surface polaire (PSA)** $\rightarrow$ exprime l'accessibilité aux liaisons hydrogène.
3. x_3 **: le volume molaire normalisé** $\rightarrow$ traduit l'encombrement stérique de la molécule.

La propriété à prédire est $y = \log S$: plus elle est élevée, plus la molécule est soluble dans l'eau.

1.2 Fondements mathématiques

Pour chaque molécule k décrite par $\mathbf{x}^{(k)} = [x_1, x_2, x_3]^\top$, le modèle calcule une combinaison linéaire pondérée, *sans seuil* :

$$\hat{y}^{(k)} = \sum_{i=1}^{3} w_i x_i^{(k)} + b = \mathbf{w}^\top \mathbf{x}^{(k)} + b$$

L'erreur de prédiction commise sur la molécule k est mesurée par le coût quadratique

$$E^{(k)} = \tfrac{1}{2}\big(y^{(k)} - \hat{y}^{(k)}\big)^2$$

dont les dérivées partielles valent

$$\frac{\partial E^{(k)}}{\partial w_i} = -\big(y^{(k)} - \hat{y}^{(k)}\big)\, x_i^{(k)} \qquad \text{et} \qquad \frac{\partial E^{(k)}}{\partial b} = -\big(y^{(k)} - \hat{y}^{(k)}\big)$$

Une descente de gradient de pas η sur ce coût donne directement la règle d'apprentissage :

Règle delta de Widrow–Hoff

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta\,\big(y^{(k)} - \hat{y}^{(k)}\big)\,\mathbf{x}^{(k)}$$
$$b^{(t+1)} = b^{(t)} + \eta\,\big(y^{(k)} - \hat{y}^{(k)}\big)$$

On retrouve la même forme que la règle du perceptron de Rosenblatt. La différence essentielle est que l'erreur $y - \hat{y}$ est maintenant un nombre réel : **chaque molécule corrige légèrement les poids**, même lorsque la prédiction est déjà proche de la valeur mesurée.

Indicateurs de qualité du modèle

Sur l'ensemble des N molécules, on évalue l'**erreur quadratique moyenne** (EQM) et le **coefficient de détermination** R^2 :

$$\mathrm{EQM} = \frac{1}{N}\sum_{k=1}^{N}\big(y^{(k)} - \hat{y}^{(k)}\big)^2 \qquad R^2 = 1 - \frac{\sum_k \big(y^{(k)} - \hat{y}^{(k)}\big)^2}{\sum_k \big(y^{(k)} - \bar{y}\big)^2}$$

2 Jeu de données expérimental

Le tableau ci-dessous regroupe 6 molécules d'apprentissage avec leurs 3 descripteurs et leur solubilité aqueuse mesurée :

Molécule	x_1 ($\log P$)	x_2 (PSA)	x_3 (Vol. molaire)	y ($\log S$ mesuré)
M1	1.0	2.0	0.5	-0.6
M2	2.0	1.0	1.5	-2.8
M3	0.5	2.5	0.2	$+0.4$
M4	2.5	2.0	1.0	-2.2
M5	1.5	1.2	0.8	-1.7
M6	3.0	2.5	2.0	-3.1

Une 7e molécule, $\mathbf{M7} = [2.0,\ 1.8,\ 1.2]^\top$, n'est pas utilisée pendant l'apprentissage : elle servira à tester la capacité de prédiction du modèle.

3 Résolution numérique détaillée (première époque)

Note méthodologique importante

Comme pour le perceptron de Rosenblatt, l'algorithme réalise un parcours séquentiel unique de M1 à M6 (**une époque**) et les poids sont mis à jour **immédiatement après chaque molécule**. La différence est qu'ici la mise à jour a lieu **à chaque molécule**, puisque l'erreur n'est presque jamais exactement nulle. Nous prenons un taux d'apprentissage $\eta = 0.05$.

Initialisons les poids et le biais à zéro :

$$\mathbf{w}^{(0)} = [0, 0, 0]^\top \quad \text{et} \quad b^{(0)} = 0$$

Étape 1 · Molécule M1

$\mathbf{x}_1 = [1.0, 2.0, 0.5]^\top$ et $y_1 = -0.6$:

$$\hat{y}_1 = (0 \times 1.0) + (0 \times 2.0) + (0 \times 0.5) + 0 = 0$$
$$e_1 = y_1 - \hat{y}_1 = -0.6 - 0 = -0.6$$

Analyse : le modèle prédit $\log S = 0$ alors que la valeur mesurée est -0.6 : il surestime la solubilité. On corrige les paramètres :

$$\begin{aligned}\mathbf{w}^{(1)} &= \mathbf{w}^{(0)} + \eta\, e_1\, \mathbf{x}_1 = [0,0,0] + 0.05 \times (-0.6) \times [1.0, 2.0, 0.5] \\ &= [0,0,0] + (-0.03) \times [1.0, 2.0, 0.5] = [\mathbf{-0.03},\ \mathbf{-0.06},\ \mathbf{-0.015}]^\top \\ b^{(1)} &= b^{(0)} + \eta\, e_1 = 0 + 0.05 \times (-0.6) = \mathbf{-0.03}\end{aligned}$$

Étape 2 · Molécule M2

$\mathbf{x}_2 = [2.0, 1.0, 1.5]^\top$ et $y_2 = -2.8$, avec les nouveaux paramètres $\mathbf{w}^{(1)}$ et $b^{(1)}$:

$$\begin{aligned}\hat{y}_2 &= (-0.03 \times 2.0) + (-0.06 \times 1.0) + (-0.015 \times 1.5) - 0.03 \\ &= -0.06 - 0.06 - 0.0225 - 0.03 = -0.1725 \\ e_2 &= y_2 - \hat{y}_2 = -2.8 - (-0.1725) = -2.6275\end{aligned}$$

Analyse : l'erreur est beaucoup plus grande (la molécule M2 est nettement moins soluble que prévu), la correction sera donc plus importante :

$$\begin{aligned}\mathbf{w}^{(2)} &= \mathbf{w}^{(1)} + \eta\, e_2\, \mathbf{x}_2 = [-0.03, -0.06, -0.015] + (-0.131375) \times [2.0, 1.0, 1.5] \\ &= [-0.03 - 0.26275,\ -0.06 - 0.131375,\ -0.015 - 0.1970625] \\ &= [\mathbf{-0.29275},\ \mathbf{-0.191375},\ \mathbf{-0.2120625}]^\top \\ b^{(2)} &= b^{(1)} + \eta\, e_2 = -0.03 + (-0.131375) = \mathbf{-0.161375}\end{aligned}$$

En poursuivant la même procédure pour M3, M4, M5 et M6, on obtient le récapitulatif suivant de la première époque (valeurs arrondies à 3 décimales) :

Molécule	$\hat{y}$ (avant)	$e = y - \hat{y}$	$\mathbf{w}$ (après)	b (après)
M1	0.000	−0.600	[−0.030, −0.060, −0.015]	−0.030
M2	−0.173	−2.628	[−0.293, −0.191, −0.212]	−0.161
M3	−0.829	+1.229	[−0.262, −0.038, −0.200]	−0.100
M4	−1.030	−1.170	[−0.408, −0.155, −0.258]	−0.158
M5	−1.163	−0.537	[−0.449, −0.187, −0.280]	−0.185
M6	−2.558	−0.542	[−0.530, −0.255, −0.334]	−0.212

À la fin de cette première époque, l'EQM sur les 6 molécules vaut encore environ 0.636. Les époques suivantes réduisent progressivement l'erreur, jusqu'à ce que l'EQM passe sous le seuil fixé.

4 Code MATLAB d'implémentation

perceptron_regression.m

```
% ========================================================================
% TP Chimiometrie : Perceptron lineaire (Adaline) pour la regression
% Prediction de la solubilite aqueuse (logS) de molecules
% ========================================================================
clear; clc; close all;

% 1. Definition du jeu de donnees (6 molecules, 3 descripteurs)
% Descripteurs : [logP, PSA, Volume Molaire]
X = [
    1.0, 2.0, 0.5;   % M1
    2.0, 1.0, 1.5;   % M2
```

```
    0.5, 2.5, 0.2;   % M3
    2.5, 2.0, 1.0;   % M4
    1.5, 1.2, 0.8;   % M5
    3.0, 2.5, 2.0    % M6
];

% Solubilite aqueuse mesuree (logS)
Y = [-0.6; -2.8; 0.4; -2.2; -1.7; -3.1];
N = size(X, 1);

% 2. Initialisation des poids, du biais et des hyperparametres
w = [0; 0; 0];
b = 0;
eta = 0.05;              % taux d'apprentissage
max_epochs = 500;
tol = 1e-3;              % seuil d'arret sur l'EQM
mse_hist = zeros(max_epochs, 1);

% 3. Boucle d'apprentissage (regle delta de Widrow-Hoff)
for epoch = 1:max_epochs

    % Parcours sequentiel de chaque molecule (une seule fois par epoque)
    for i = 1:N
        xi = X(i, :)';
        yi = Y(i);

        % Sortie lineaire (activation = identite) et erreur reelle
        y_pred = dot(w, xi) + b;
        erreur = yi - y_pred;

        % Mise a jour immediate des parametres (a chaque molecule)
        w = w + eta * erreur * xi;
        b = b + eta * erreur;
    end

    % Erreur quadratique moyenne (EQM) sur l'epoque complete
    Y_hat = X * w + b;
    mse_hist(epoch) = mean((Y - Y_hat).^2);

    % Condition d'arret : EQM inferieure au seuil
    if mse_hist(epoch) < tol
        fprintf('Convergence atteinte a l''epoque %d (EQM = %.2e)\n', ...
                epoch, mse_hist(epoch));
        break;
    end
end
mse_hist = mse_hist(1:epoch);

% 4. Qualite du modele : coefficient de determination R^2
Y_hat  = X * w + b;
SS_res = sum((Y - Y_hat).^2);
SS_tot = sum((Y - mean(Y)).^2);
R2 = 1 - SS_res / SS_tot;

% 5. Solution analytique de reference (moindres carres)
A = [X, ones(N, 1)];
```

```
theta = A \ Y;            % theta = [w1; w2; w3; b]

% 6. Prediction pour une nouvelle molecule M7 (non utilisee a l'apprentissage)
x_new = [2.0, 1.8, 1.2];
y_new = x_new * w + b;

% 7. Affichage des resultats finaux
fprintf('\n----------------------------------------\n');
fprintf('Poids appris w    : [%.4f, %.4f, %.4f]\n', w);
fprintf('Biais appris b    : %.4f\n', b);
fprintf('EQM finale        : %.2e\n', mse_hist(end));
fprintf('Coefficient R^2   : %.4f\n', R2);
fprintf('----------------------------------------\n');
fprintf('Solution analytique (moindres carres) :\n');
fprintf('w = [%.4f, %.4f, %.4f],  b = %.4f\n', theta(1:3), theta(4));
fprintf('Prediction pour M7 : logS = %.2f\n', y_new);

% 8. Visualisation : courbe d'apprentissage et valeurs predites vs mesurees
figure;
subplot(1, 2, 1);
semilogy(1:epoch, mse_hist, 'LineWidth', 1.5);
xlabel('Epoque'); ylabel('EQM'); title('Courbe d''apprentissage');
grid on;

subplot(1, 2, 2);
plot(Y, Y_hat, 'o', 'MarkerFaceColor', 'b'); hold on;
plot([min(Y) max(Y)], [min(Y) max(Y)], 'k--');   % droite y = x
xlabel('logS mesure'); ylabel('logS predit');
title('Predit vs mesure'); grid on;
```

5 Résultats attendus et interprétation

L'exécution du script affiche dans le terminal :

Sortie console

```
Convergence atteinte a l'epoque 95 (EQM = 9.82e-04)

----------------------------------------
Poids appris w    : [-0.8435, 0.5897, -0.7744]
Biais appris b    : -0.4947
EQM finale        : 9.82e-04
Coefficient R^2   : 0.9993
----------------------------------------
Solution analytique (moindres carres) :
w = [-0.8210, 0.6121, -0.7956],  b = -0.5770
Prediction pour M7 : logS = -2.05
```

Les poids obtenus par descente de gradient sont très proches de la solution analytique des moindres carrés ; le léger écart vient du fait que l'apprentissage s'arrête dès que l'EQM passe sous le seuil 10^{-3}.

Molécule	$\log S$ mesuré	$\log S$ prédit	Résidu
M1	-0.600	-0.546	-0.054
M2	-2.800	-2.754	-0.046
M3	$+0.400$	$+0.403$	-0.003
M4	-2.200	-2.198	-0.002
M5	-1.700	-1.672	-0.028
M6	-3.100	-3.100	0.000

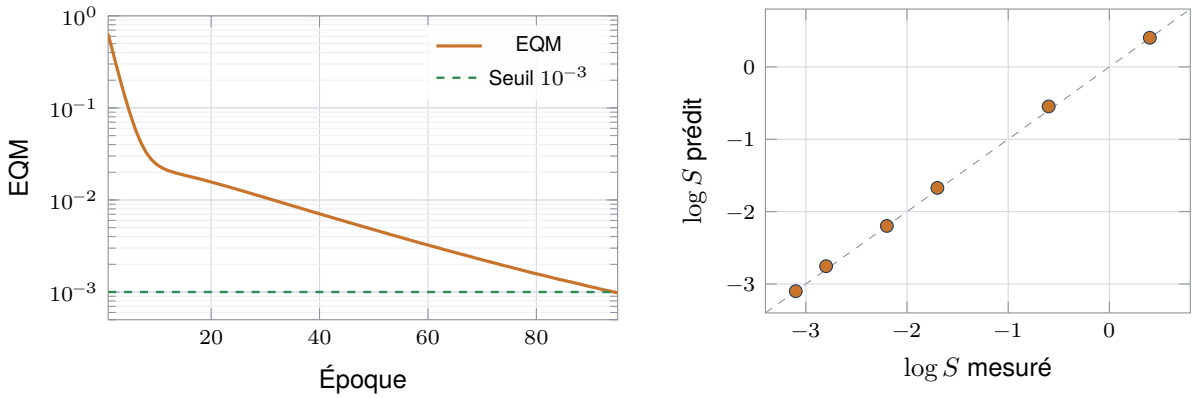


Figure 1. *À gauche : courbe d'apprentissage (EQM en échelle logarithmique) ; à droite : valeurs prédites en fonction des valeurs mesurées (la droite en pointillés représente la prédiction parfaite).*

Interprétation chimique des poids

Les signes des poids appris sont cohérents avec la physico-chimie de la dissolution dans l'eau :

- $w_1 < 0$: plus une molécule est **lipophile** ($\log P$ élevé), moins elle est soluble ;
- $w_2 > 0$: plus sa **surface polaire** est grande, plus elle forme de liaisons hydrogène avec l'eau et plus elle est soluble ;
- $w_3 < 0$: plus sa **taille** est importante, plus le coût de création d'une cavité dans l'eau augmente et moins elle est soluble.

6 Exercices

1. Exécutez le script avec $\eta \in \{0.01,\ 0.05,\ 0.2,\ 0.3\}$. Comparez le nombre d'époques nécessaires et l'allure de la courbe d'apprentissage. Que se passe-t-il pour les grandes valeurs de η ?
2. Remplacez le seuil `tol` par 10^{-5} (et augmentez `max_epochs` si nécessaire). Les poids se rapprochent-ils de la solution analytique ?
3. Pourquoi est-il recommandé de **normaliser** les descripteurs avant l'apprentissage ? Quelle difficulté apparaîtrait si x_3 était exprimé dans une unité mille fois plus grande ?

4. Proposez une interprétation chimique de l'ordre de grandeur des poids $|w_1|$, $|w_2|$ et $|w_3|$. Quel descripteur influence le plus la solubilité dans ce jeu de données ?

5. Quelle est la limite d'un modèle linéaire si la relation entre $\log S$ et les descripteurs est non linéaire ? Quelle extension du perceptron permettrait de la lever ?