



UNIVERSITÉ جامعة  
**MUNDIAPOLIS**  
HONORIS UNITED UNIVERSITIES

# Techniques de Regularisation

Réaliser par :

Baali Ghizlane  
Imane Taghzout

Encadré par :

Pr. Zouhair ELAMRANI  
ABOU ELASSAD

# Plan :

- 1 Introduction
- 2 Problème de l'Overfitting
- 3 Solution pour l'Overfitting
- 4 Définition de Régularisation
- 5 Comment fonctionne la régularisation ?

## Plan :

6

Les Types de Techniques de régularisation ?

---

7

Choix de type de Régularisation

---

8

Comparaison entre les types de régularisations

---

9

Exemples pratiques du régularisation

---

10

Cas d'utilisation et exemples

---



# Introduction

La régularisation en Machine Learning est une technique essentielle pour éviter le surajustement "Overfitting". Pour améliorer la performance du modèle dans de tels cas, diverses méthodes comme l'augmentation des données d'entraînement, la validation croisée et l'ingénierie des caractéristiques peuvent être utilisées. Cependant, ces techniques ont leurs limites.

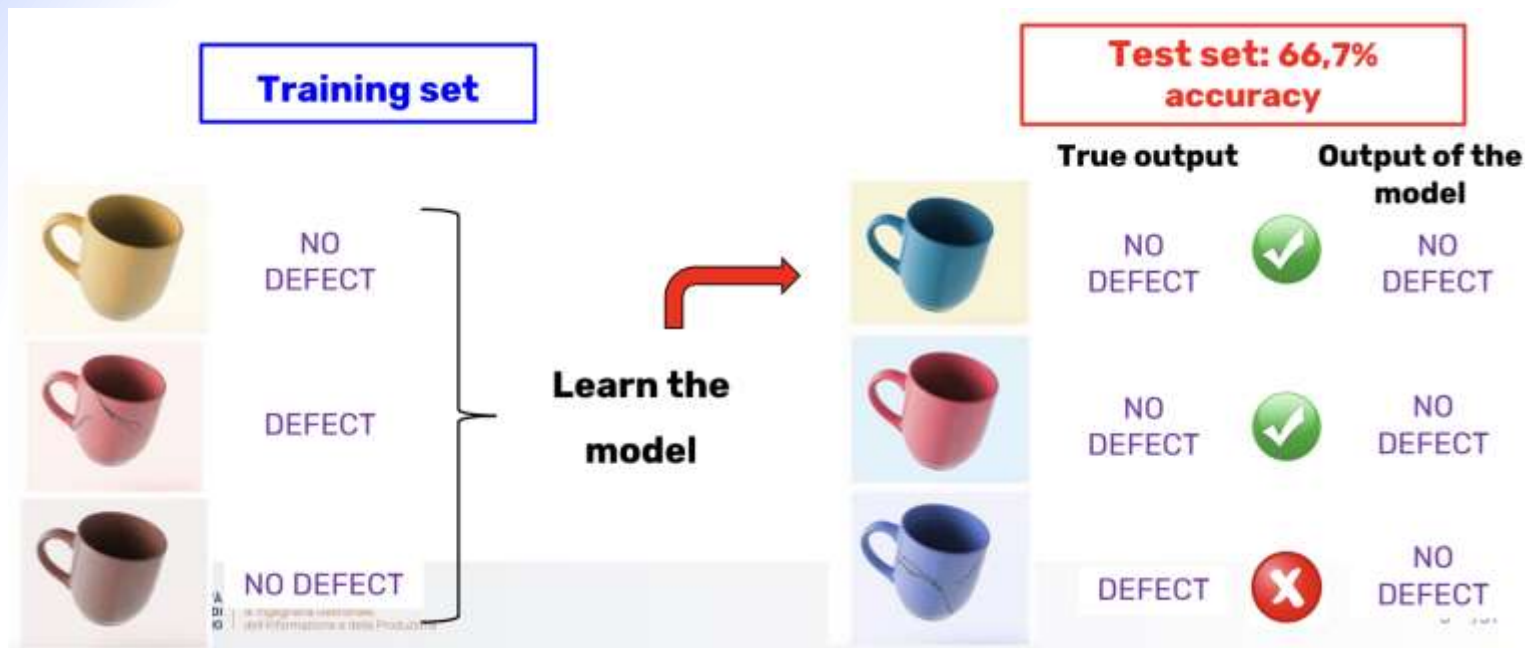
La régularisation se distingue comme une solution très efficace. Elle consiste à introduire des contraintes sur le modèle lors de l'entraînement pour l'empêcher de devenir trop complexe.





# Problème de l'Overfitting

"Un modèle appris doit être testé sur un ensemble de données distinct de celui utilisé pour son entraînement."

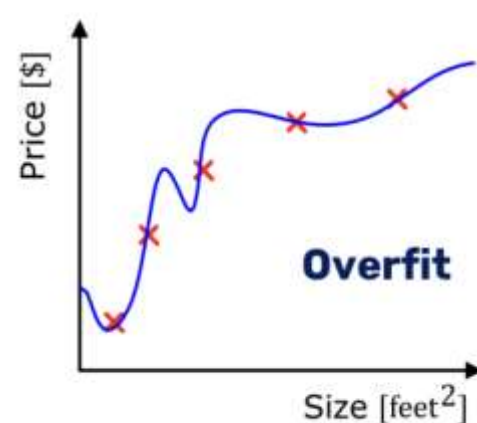
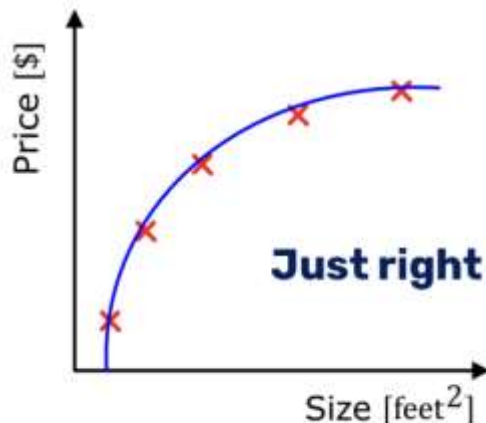
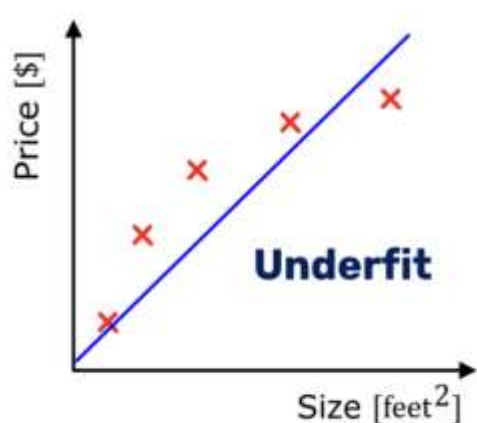




L'objectif de l'utilisation d'un test set est d'évaluer les performances du modèle appris sur des données non vues.

→ C'est un peu comme passer un examen pour évaluer si vous avez bien compris le matériel, et non simplement si vous l'avez mémorisé !

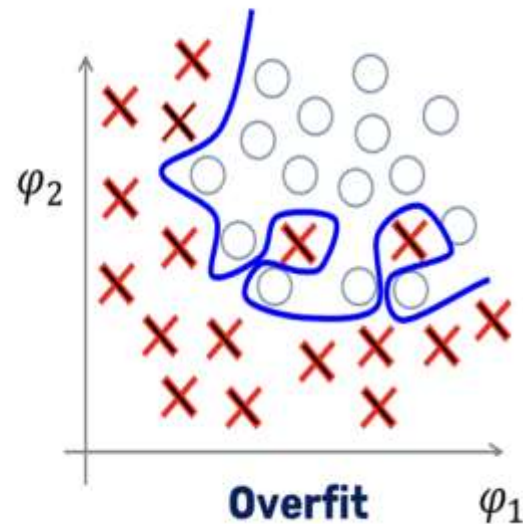
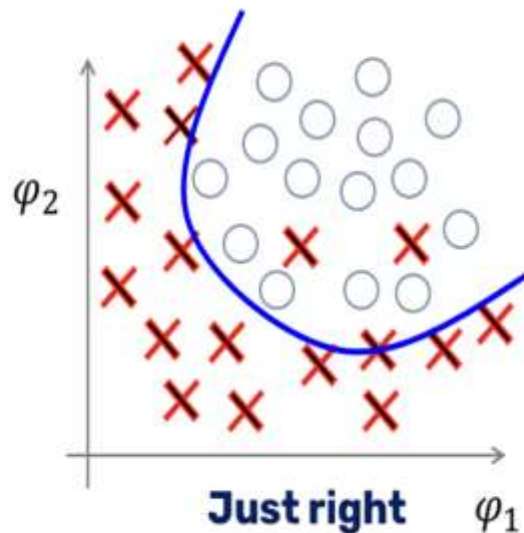
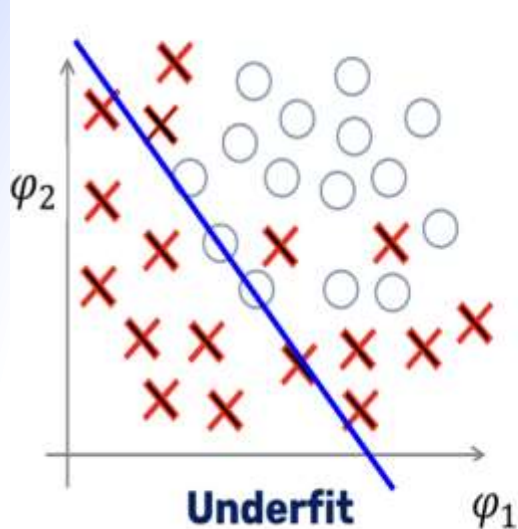
### **Example: House regression problem**





**Overfitting** : Si nous avons trop de caractéristiques “features” , l'hypothèse apprise peut s'adapter très bien à l'ensemble d'entraînement, mais échouer à généraliser à de nouveaux exemples.

### Example: Classification problem





# Solution pour l'Overfitting

La **régularisation** est la première ligne de défense contre le overfitting.

Nous avons constaté que les **modèles complexes** sont plus susceptibles au overfitting.

- Cela s'explique par le fait qu'ils sont plus puissants et qu'ils peuvent donc s'adapter au bruit.

Les **modèles simples** ont une expressivité limitée et sont donc moins enclins à s'adapter au bruit.

- Cependant, si nous nous en tenons uniquement à des modèles simples, nous risquons de ne pas obtenir une approximation satisfaisante de la fonction cible (target function).

Comment conserver les avantages des deux mondes ?

$$E_{aug}(\theta) = \frac{1}{N} \sum_{i=1}^N (y(i) - h(\varphi(i); \theta))^2 + \lambda \cdot \sum_{j=0}^{d-1} (\theta_j)^2$$

Au lieu de minimiser l'erreur dans l'échantillon  $E_{in}$  (c'est-à-dire la fonction de coût  $J(\theta)$ ), on peut minimiser l'erreur augmentée :

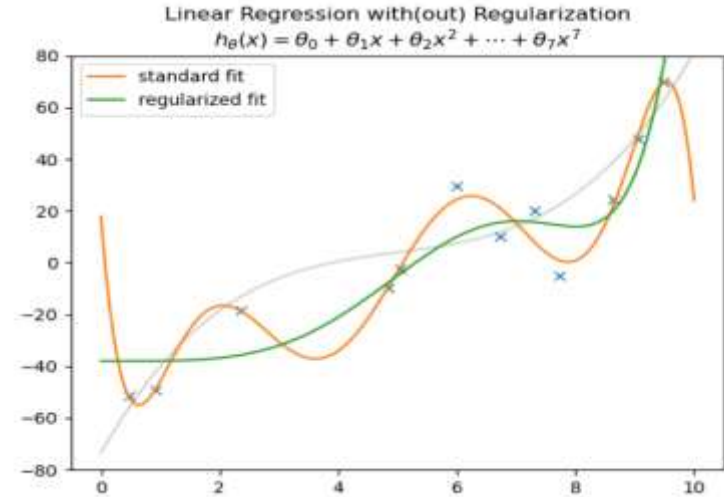
• The term  $\Omega = \sum_{j=0}^{d-1} (\theta_j)^2$  is called **regularizer**

• The term  $\lambda$  (**regularization hyper-parameter**) weights the importance of minimizing  $J(\theta)$ , with respect to minimizing  $\Omega$ .



# Définition de Régularisation

La régularisation fait référence à une technique qui réduit le overfitting d'un modèle en pénalisant les estimations de coefficients afin que le modèle ne capte pas le bruit dans les données et évite ainsi overfitting ou le underfitting. En utilisant ces techniques, nous pouvons éviter les erreurs et augmenter les performances de notre modèle.



L'illustration montre l'amélioration des performances du modèle après régularisation, rendant la courbe plus lisse que celle du standard fit. La régularisation a réduit l'ampleur des caractéristiques tout en maintenant le même nombre, aidant à résoudre des problèmes de haute dimensionnalité.

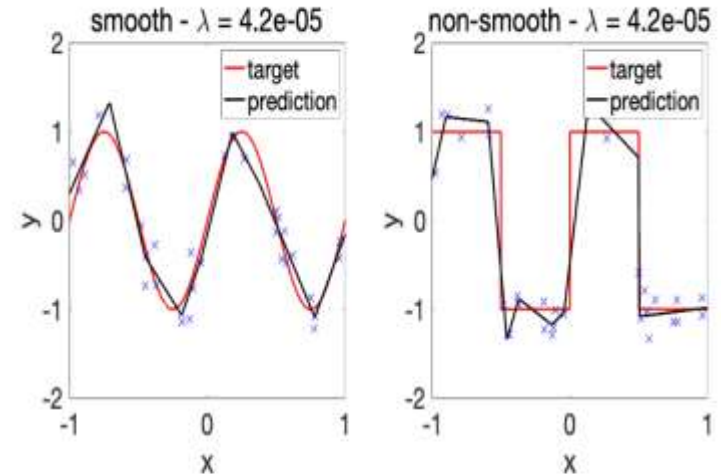
# Comment fonctionne la régularisation ?



La technique de régularisation fonctionne en pénalisant les paramètres d'apprentissage. Elle réduit l'erreur, et par conséquent overfitting et underfitting, en diminuant les paramètres de régularisation.

Dans l'illustration, nous pouvons clairement voir que la courbe devient plus lisse lorsque la régularisation est appliquée et que la courbe devient irrégulière lorsque le paramètre lambda est augmenté.

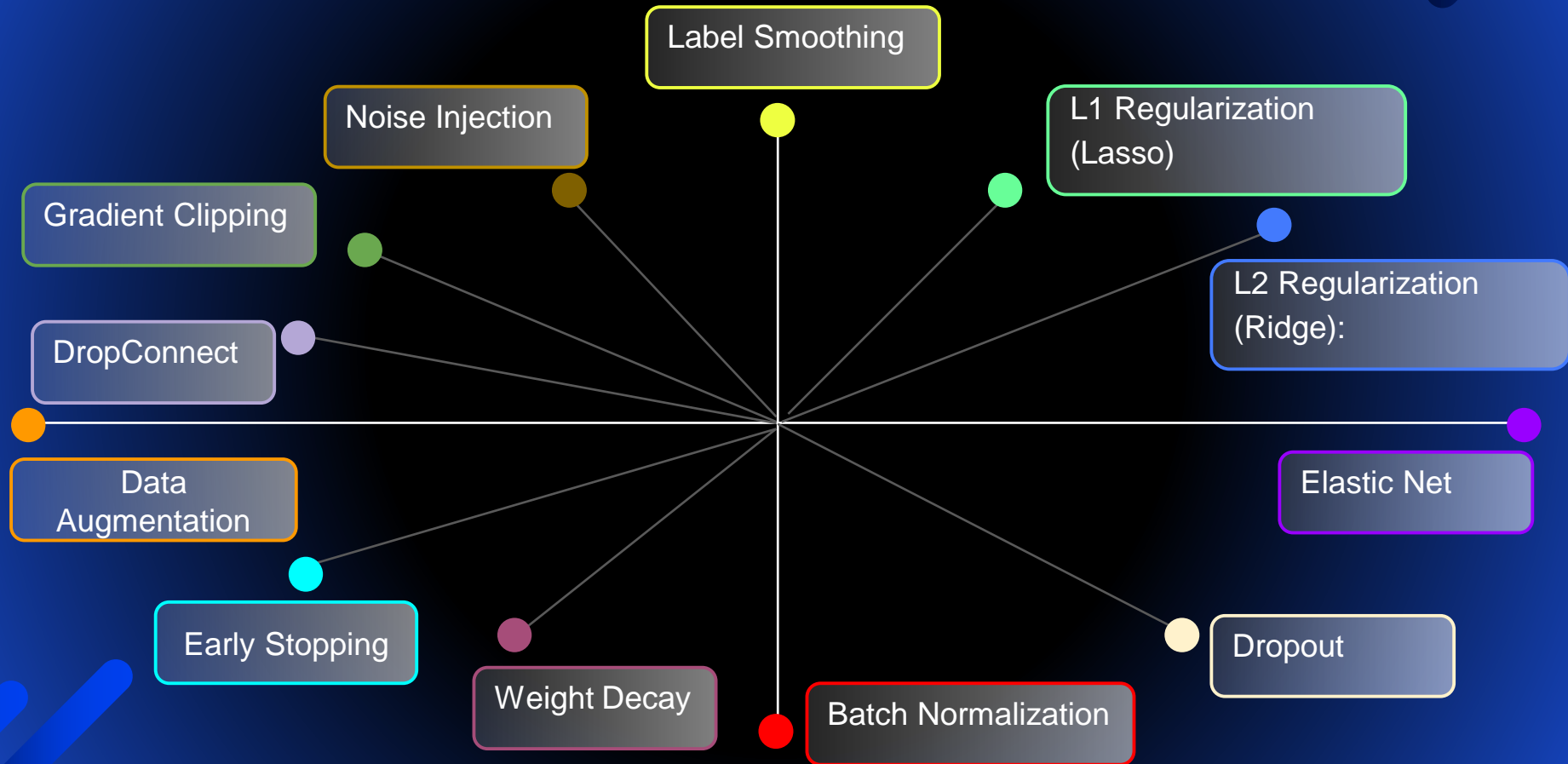
Nous nous pouvons aussi observer comment ces régularisations changent au fil du temps en pénalisant les paramètres.



source: Image



# Les Types de Techniques de Régularisation



## **L1 Regularization (Lasso)**

Ajoute les valeurs absolues des poids à la fonction de perte, pénalisant les poids importants. Conduit à des vecteurs de poids éparse (sparse) car elle encourage certains poids à être exactement égaux à zéro.

---

## **L2 Regularization (Ridge)**

Ajoute les valeurs au carré des poids à la fonction de perte, pénalisant les poids importants. Encourage de petites valeurs de poids et des vecteurs de poids plus lisses.

---

## **Elastic Net**

Une combinaison de la régularisation L1 et L2, vous permettant de contrôler l'équilibre entre les deux.

---

## **Dropout**

Pendant l'entraînement, met de manière aléatoire une fraction des sorties des neurones à zéro, les désactivant efficacement. Empêche la coadaptation des neurones et encourage le réseau à être plus robuste.

---

## **Batch Normalization**

Normalise les activations de chaque couche au sein d'un mini-lot pour avoir une moyenne nulle et une variance unitaire. Aide à stabiliser et accélérer l'entraînement en réduisant le décalage covariant interne.

---

## Weight Decay

Ajoute un terme de pénalité à la fonction de perte qui décourage les grandes valeurs de poids. Équivalent à la régularisation L2.

## Early Stopping

Surveille les performances de validation pendant l'entraînement et arrête l'entraînement lorsque les performances commencent à se dégrader, empêchant le modèle de surajustement.

## Data Augmentation

Génère des données d'entraînement supplémentaires en appliquant diverses transformations aux données existantes, telles que la rotation, le redimensionnement, le retournement et le rognage. Augmente la diversité de l'ensemble d'entraînement et réduit le surajustement.

## DropConnect

Similaire au dropout mais applique le masque de dropout aux connexions entre les neurones plutôt qu'aux sorties des neurones. Met de manière aléatoire les poids à zéro pendant l'entraînement.

## Gradient Clipping

Limite l'amplitude des gradients pendant l'entraînement pour éviter les problèmes de gradient explosif.

## Noise Injection

---

Ajoute du bruit aux entrées ou aux couches intermédiaires pendant l'entraînement. Agit comme une forme de régularisation en rendant le réseau plus robuste au bruit.

## Label Smoothing

---

Au lieu d'utiliser des étiquettes binaires dures, attribue une distribution d'étiquettes souples à chaque classe. Réduit la confiance du modèle et l'encourage à apprendre des caractéristiques plus généralisables.

# Choix de type de Régularisation

Le choix des techniques de régularisation dépend souvent des caractéristiques spécifiques de l'ensemble de données et de la complexité du modèle.

- les plus utilisés sont :

1. L1 Regularization (Lasso)
2. L2 Regularization (Ridge)
3. Elastic Net
4. Dropout

- **$L_2$  regularizer:** also called **Ridge** regression  $\Omega(\theta) = \sum_{j=0}^{d-1} (\theta_j)^2 = \theta^T \theta = \|\theta\|_2^2$

- **$L_1$  regularizer:** also called **Lasso** regression  $\Omega(\theta) = \sum_{j=0}^{d-1} |\theta_j| = \|\theta\|_1$

- **Elastic-net regularizer:**  $\Omega(\theta) = \sum_{j=0}^{d-1} \beta (\theta_j)^2 + (1 - \beta) \sum_{j=0}^{d-1} |\theta_j|$

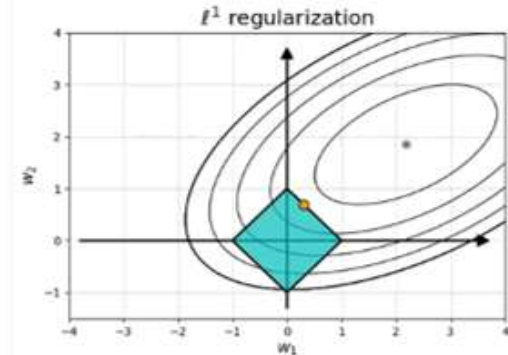
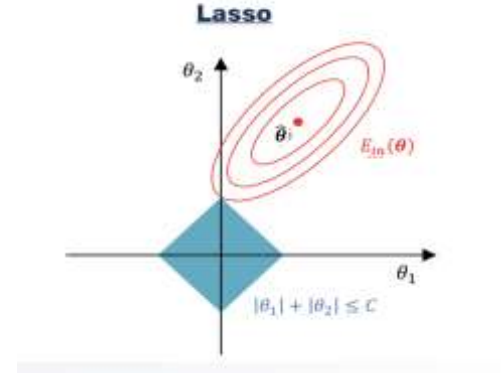
# L1 Regularization (LASSO)



La régularisation L1, utilisant la régression Lasso, réduit le nombre de paramètres en poussant les données vers un point central, souvent zéro. Elle ajoute une pénalité basée sur la valeur absolue des coefficients à la fonction de perte, éliminant ainsi les caractéristiques moins importantes et réduisant le overfitting.

Cette technique est particulièrement utile pour les modèles avec un grand nombre de caractéristiques, comme la régression logistique, où elle aide à éliminer les caractéristiques non pertinentes et à améliorer les performances du modèle.

$$\text{Lasso Regression Cost Function} = \text{Loss Function} + \lambda \sum_{j=1}^m |w_j|$$





# L2 Regularization (RIDGE)

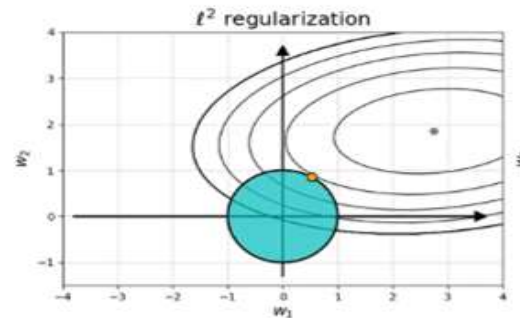
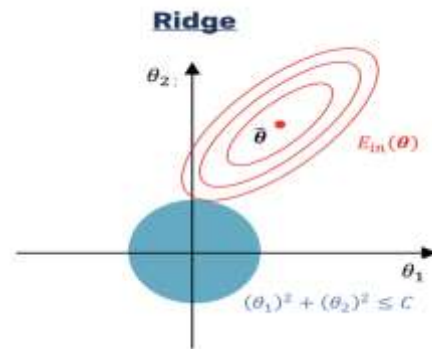


La régularisation L2 utilise la régression Ridge, adaptée à l'analyse de données présentant une multicollinéarité. Contrairement à L1 Lasso qui pénalise les poids par la somme de leurs valeurs absolues, la régression Ridge les pénalise par la somme de leurs valeurs au carré.

Cette méthode permet de réduire le biais tout en diminuant la variance, ce qui améliore la précision des prédictions.

La régression Ridge est efficace lorsque les variables sont fortement corrélées. Contrairement à la régularisation L1, où les coefficients tendent à devenir nuls, la régularisation L2 répartit les coefficients de manière plus uniforme, ce qui donne des modèles non parcimonieux.

$$\text{Ridge Regression Cost Function} = \text{Loss Function} + \frac{1}{2} \lambda \sum_{j=1}^m w_j^2$$



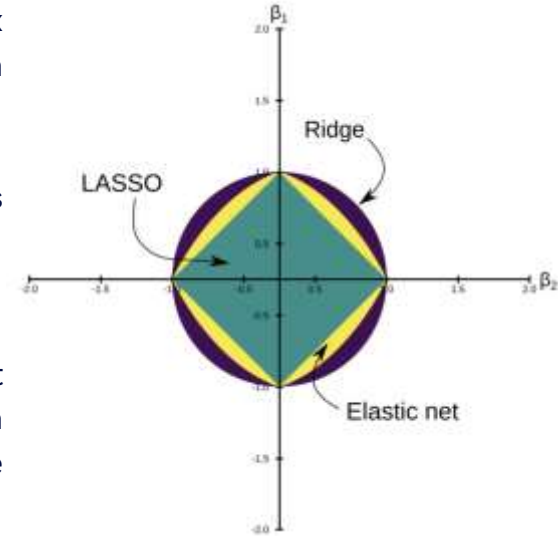
# Elastic Net



La régularisation Elastic Net est une technique de régression linéaire qui combine à la fois les méthodes de régularisation L1 et L2 . Elle utilise deux hyperparamètres, alpha et lambda, pour sélectionner les caractéristiques tout en réduisant les coefficients.

- ❖ L1 facilite la sélection des caractéristiques en définissant certains coefficients à zéro, ce qui favorise la sparsity.
- ❖ L2 pénalise les coefficients non nuls pour éviter le overfitting.

Elastic Net est particulièrement utile pour les ensembles de données comportant un grand nombre de caractéristiques et des problèmes de multicolinéarité. En combinant L1 et L2, elle offre une approche équilibrée qui améliore la robustesse et les performances de généralisation du modèle.



Elastic Net cost function is:

$$\frac{1}{n} \left( \left[ \sum_{i=1}^n (Y_i - \hat{Y}_i)^2 \right] + \left[ \lambda \left( \alpha \sum_{j=1}^p |\beta_j| + (1 - \alpha) \sum_{j=1}^p \beta_j^2 \right) \right] \right)$$

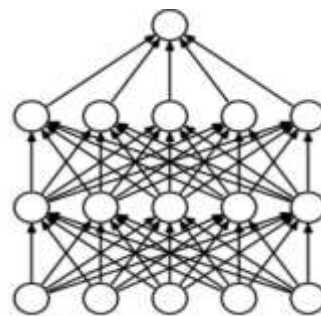
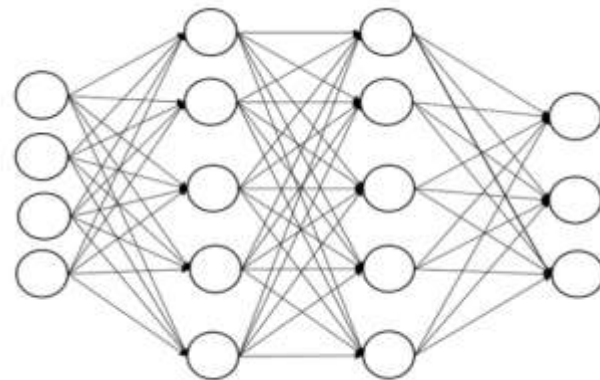
# Dropout



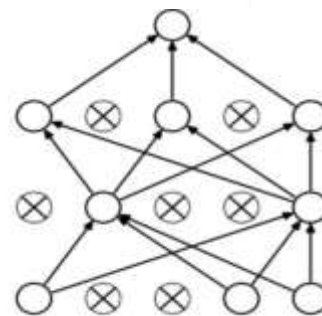
Dropout est une technique de régularisation largement utilisée en apprentissage profond. En pratique, elle consiste à désactiver aléatoirement certains neurones pendant l'entraînement d'un réseau neuronal.

Cette méthode aide à prévenir le overfitting en obligeant le réseau à ne pas trop dépendre de certains neurones ou combinaisons de neurones. En conséquence, le réseau devient plus robuste et généralise mieux sur de nouvelles données.

- Ajoute beaucoup de bruit
- Préviend le overfitting en brisant les prédictions fragiles
- Distribue le signal d'apprentissage
- Renforce la représentation distribuée
- Oblige les "branches mortes" à apprendre
- En quelque sorte, approxime les méthodes d'ensemble



(a) Standard Neural Net



(b) After applying dropout.

L1 Regularization	L2 Regularization	Drop-Out Regularization
Il modifie le modèle en ajoutant une pénalité équivalente à la somme des valeurs absolues des coefficients	Il modifie le modèle en ajoutant une pénalité équivalente à la somme des carrés de la magnitude des coefficients	Il modifie le réseau neuronal en désactivant les neurones de manière aléatoire.
Effectue une minimisation des coefficients en prenant les vraies valeurs des coefficients.	Effectue une minimisation en élevant au carré les amplitudes des coefficients.	Il élimine des neurones pour minimiser les paramètres d'apprentissage
Il est éparsé (sparse).	Il n'est pas éparsé (non-sparse).	Il est éparsé (sparse).
Robuste aux valeurs aberrantes.	Pas robuste aux valeurs aberrantes.	Non affectée par les valeurs aberrantes.

Comparaison entre les types de régularisations

# Exemples pratiques du régularisation



1. Régression linéaire : Utilise la régularisation (L1 ou L2) pour éviter l'overfitting et rendre le modèle plus robuste aux données bruitées, comme dans la prédiction des prix de l'immobilier.
2. Régression logistique : Adaptée à la classification binaire, utile pour prédire les achats des clients, en particulier lorsque les caractéristiques sont nombreuses ou corrélées.
3. Réseaux neuronaux : La régularisation (L2, dropout) est cruciale pour éviter l'overfitting dans les modèles d'apprentissage profond, comme la classification d'images.
4. Machines à vecteurs de support (SVM) : Il est également possible d'appliquer une régularisation L1 aux SVM pour effectuer une sélection des caractéristiques en poussant certains poids de ces dernières vers zéro.



# Cas d'utilisation et Exemples

Lien Github: [https://github.com/Ghizlane8/Regularization\\_Techniques](https://github.com/Ghizlane8/Regularization_Techniques)

# Référence

<https://www.einfochips.com/blog/regularization-make-your-machine-learning-algorithms-learn-not-memorize/>

<https://towardsdatascience.com/types-of-regularization-in-machine-learning-eb5ce5f9bf50>

<https://medium.com/analytics-vidhya/types-of-regularization-and-when-to-use-them-f0350ce651a7>

<https://www.geeksforgeeks.org/regularization-in-machine-learning/>

<https://merioh.com/a/4f5f8114639b/regularization-in-machine-learning-a-comprehensive-guide>

<https://buggyprogrammer.com/regularization-in-machine-learning/>

<https://www.geeksforgeeks.org/regularization-in-machine-learning/>

<https://techvidvan.com/tutorials/regularization-in-machine-learning/>

Thank You!

