

# Choix des Paramètres du Modèle

## 1. Architecture du Réseau

- **Couches Cachées** : Trois couches cachées ont été choisies (128, 256, 128 neurones respectivement) pour capturer la complexité du processus. Les couches plus profondes permettent au modèle de capturer des relations non linéaires plus subtiles entre les variables d'entrée.
- **Couches Denses (Fully Connected Layers)** : Chaque couche est dense, c'est-à-dire que chaque neurone d'une couche est connecté à tous les neurones de la couche suivante. Cela permet de capturer pleinement les interactions potentielles entre toutes les variables d'entrée.

## 2. Fonction d'Activation (ReLU)

- **Pourquoi ReLU (Rectified Linear Unit) ?** La fonction d'activation ReLU est utilisée pour introduire de la non-linéarité dans le modèle. Elle est simple à calculer et aide à éviter le problème du gradient qui disparaît (vanishing gradient), un problème courant lors de l'entraînement de réseaux de neurones profonds. ReLU accélère également la convergence pendant l'entraînement en permettant à certaines sorties de neurones de rester nulles, réduisant ainsi la complexité.

## 3. Couche de Régularisation (Dropout)

- **Pourquoi Dropout ?** La couche Dropout, configurée avec un taux de 0,3, permet de réduire le surapprentissage (overfitting). En "éteignant" aléatoirement 30% des neurones à chaque itération d'entraînement, Dropout aide le modèle à généraliser en empêchant certains neurones de s'adapter excessivement aux données d'entraînement. Cela favorise également une meilleure performance sur les nouvelles données de test.

## 4. Couche de Sortie (Output Layer)

- **Neurone Unique avec Activation Linéaire** : La couche de sortie est un neurone unique sans fonction d'activation (activation linéaire), car le but est de faire de la régression (prédire une valeur continue). Le modèle génère ainsi une sortie sans limite spécifique, correspondant aux valeurs prévues du processus.

## 5. Fonction de Perte (Mean Squared Error - MSE)

- **Pourquoi MSE ?** L'erreur quadratique moyenne (Mean Squared Error) est couramment utilisée pour les tâches de régression, car elle mesure la moyenne des carrés des erreurs entre les valeurs réelles et les prédictions du modèle. MSE donne une importance particulière aux erreurs plus grandes, ce qui pousse le modèle à réduire autant que possible les grandes erreurs, améliorant ainsi la précision globale.

## 6. Optimiseur (Adam)

- **Pourquoi Adam ?** L'optimiseur Adam combine les avantages des algorithmes d'optimisation AdaGrad (qui adapte le taux d'apprentissage pour chaque paramètre) et RMSProp (qui maintient une moyenne des gradients pour stabiliser l'apprentissage). Adam est donc bien adapté à ce type de projet, car il ajuste automatiquement le taux d'apprentissage pour chaque paramètre et gère efficacement les grandes quantités de données et les paramètres.

## ***7. Taille du Lot (Batch Size)***

- **Pourquoi un Batch Size de 64 ?** Une taille de lot de 64 est choisie pour offrir un équilibre entre rapidité et stabilité lors de l'entraînement. Avec un lot de taille moyenne, le modèle peut tirer profit des avantages de la mise à jour par lots sans pour autant nécessiter une mémoire excessive.

## ***8. Nombre d'Époques (Epochs) et Arrêt Précoce (Early Stopping)***

- **Pourquoi Early Stopping avec Patience de 5 ?** Bien qu'un nombre maximal d'époques (epochs) de 100 soit défini, l'early stopping permet de stopper l'entraînement lorsque le modèle ne montre plus d'amélioration sur l'ensemble de validation pendant 5 époques consécutives. Cela empêche un surapprentissage et réduit le temps d'entraînement, en s'assurant que le modèle converge vers un état optimal.

## ***9. Division des Données (Training / Validation / Test Split)***

- **Répartition des Données :** Environ 80% des données sont affectées à l'entraînement, et les 20% restantes servent de test, avec un sous-ensemble de 20% des données d'entraînement pour la validation. Cela permet de surveiller le modèle pendant l'entraînement et d'assurer une bonne généralisation.