



*Département d'ingénierie et d'informatique
Université du Québec en Outaouais*

Prédiction de la consommation d'énergie avec les RNA
Projet/laboratoire Hiver 2020

Par

Antoine Fortin

FORA16039708

GEN1663
Réseaux intelligents
Présenté à
Marek Zaremba

Pavillon Lucien-Brault
12 mars 2020

Table des matières

Introduction	3
Familiarisation avec Matlab	4
Analyse de résultats d'une régression	6
Analyse de résultats de la prédiction de la consommation de l'énergie	8
1 heure	8
6 heures	10
24 heures	12
Conclusion	15

Table des Figures

Figure 1: Page d'entraînement de nftool.....	4
Figure 2: Page d'entraînement et de résultats.....	5
Figure 3: Résultat des régressions de bases	6
Figure 4: Erreur quadratique moyenne des régressions de bases	7
Figure 5: Résultats des régressions pour prédiction 1 heure.....	8
Figure 6: Erreur quadratique moyenne des régressions pour prédiction 1 heure.....	9
Figure 7: Résultats des régressions pour prédiction 6 heures	10
Figure 8: Erreur quadratique moyenne des régressions pour prédiction 6 heures.....	11
Figure 9.1: Résultats des régressions pour prédiction 24 heures avec division 85/10/5	12
Figure 9.2: Résultats des régressions pour prédiction 24 heures avec division 70/15/15.....	13
Figure 10: Erreur quadratique moyenne des régressions pour prédiction 24 heures.....	14

Introduction

Depuis la dernière décennie, les réseaux de neurones artificielles (RNA) sont de plus en plus utilisés dans plusieurs situations : la classification et le traitement sont des exemples. Cependant, ce projet touche uniquement l'approximation d'une fonction. En effet, avec 9568 vecteurs de données, il sera possible d'approximer une fonction pour calculer l'énergie produite par une centrale à cycle combiné composée de turbines à gaz et de turbines à vapeur. La fonction de l'énergie sera analysée à partir de la température ambiante, la pression du gaz d'échappement, la pression ambiante et l'humidité relative.

Pour se faire, le projet est divisé en quatre parties : la familiarisation avec les outils de réseaux neuraux de Matlab, la création de réseaux de neurones avec des structures différentes, l'étude de l'influence des paramètres sur l'apprentissage du réseau et l'analyse de la prédiction de la consommation d'énergie. Ce projet se concentra surtout sur la prédiction de la consommation d'énergie pour 1 heure à l'avance et sur une période de 6 heures. D'ailleurs, la familiarisation avec Matlab, plus précisément *nftool*, sera abordée, ainsi que l'analyse des résultats d'une régression pour déterminer la puissance à un instant connaissant la valeur des 4 entrées au même instant.

Familiarisation avec Matlab

Matlab inclut plusieurs outils pour utiliser et créer des réseaux de neurones artificielles depuis plusieurs années. Un de ces outillages est *nftool* (*neural net fitting tool*) qui permet de résoudre les problèmes d'approximation d'entrée-sortie. Les premières étapes d'utilisation de cette outil consiste à choisir les données, la répartition entre les divers types de tests, le nombre de couche de neurone et l'algorithme d'entraînement

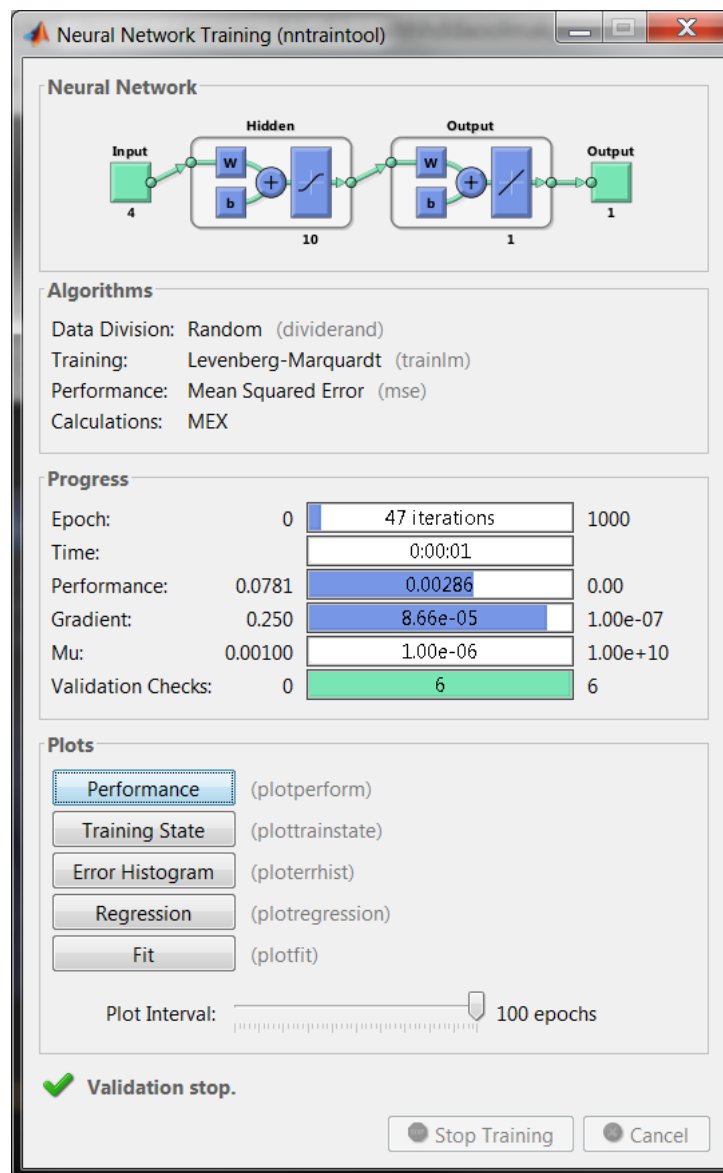


Figure 1: Page d'entrainement de nftool

Une fois le nombre de données d'entraînement, de validation de test et le nombre de couches de neurone choisi, le réseau est prêt à être entraîné. Une fois l'entraînement démarré, la page d'entraînement apparaît en indiquant le nombre d'itérations, l'erreur quadratique moyenne et le schéma du réseau de neurone. De plus, il est possible d'afficher les graphiques des divers tests et le graphique d'évolution de l'erreur quadratique moyenne.

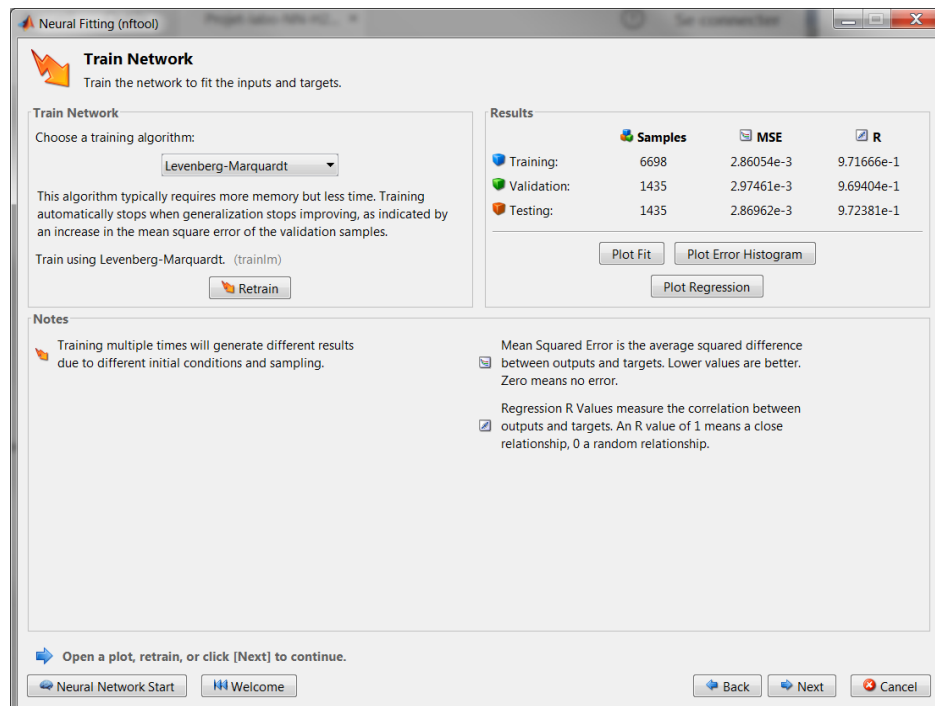


Figure 2: Page d'entraînement et de résultats

Une autre page pertinente est celle de la figure 2, la page d'entraînement et de résultats. Cette page montre un aperçu des résultats en affichant le nombre d'échantillon de chaque test ainsi que des données de performances comme l'erreur quadratique moyenne et le facteur de corrélation R pour chaque test. Pour ce projet, cet outil sera utilisé pour les approximations.

Analyse de résultats d'une régression

Avant d'utiliser *nftool* pour effectuer une régression de base, il faut importer et normaliser les données de chaque entrée et chaque sortie. Pour se faire, il faut étalonner les données de 0 à 1 ou 0 correspond au maximum et 0 au minimum d'un des types de données. Par exemple, dans le cas du projet, puisque la puissance varie entre 420.26 MW et 495.76 MW, toutes les valeurs utilisées entre 420.26 et 495.76 seront entre 0 et 1.

Suite à la normalisation, les données sont prêtes à être utilisé avec *nftool*. Ce cas-ci a utilisé une distribution de 70/15/15. C'est-à-dire, 70% des données seront utilisées pour l'entraînement, 15% pour la validation et le reste pour le test final. De plus, le réseau contient 10 couches de neurones et utilise l'algorithme Levenberg-Marquardt. Cette régression permettra de produire un réseau de neurones avec comme entrées la température, la pression, l'humidité relative et la pression du gaz d'échappement et comme sortie la production d'énergie. En d'autres mots, une fonction permettra d'approximer la puissance en fonction des 4 autres paramètres.

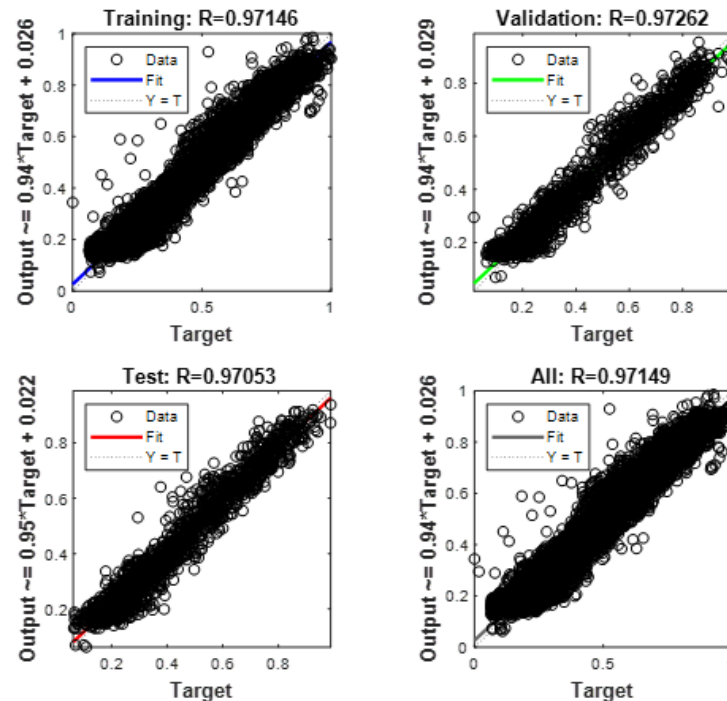


Figure 3: Résultat des régressions de bases

La figure 3 présente 4 régressions : l'entraînement, la validation, le test final et les 3 combinés. Visiblement, les 3 régression se ressemblent et ont un facteur de corrélation presque identique. Ces résultats présentent une régression réussie, puisque les étapes de l'entraînement et de la validation permettent de créer un réseau de neurones artificielles qui approximent bien la production énergétique. La régression de test confirme le tout. De plus, ces régressions seront utilisées comme références de performances

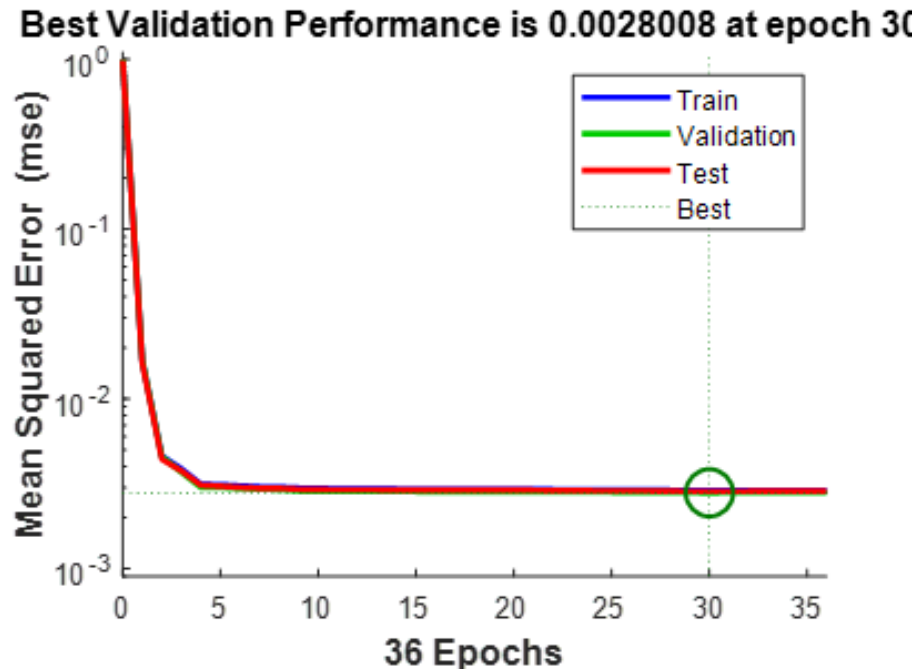


Figure 4: Erreur quadratique moyenne des régressions de bases

La figure ci-haut montre l'évolution du MSE au cours des itérations. Évidemment, plus il y a d'itérations, plus le réseau devient performant. De plus, les trois régressions ont un MSE très semblable tout au long de leur évolution ce qui, avec les autres résultats, confirme que le réseau est satisfaisant.

Analyse de résultats de la prédiction de la consommation de l'énergie

Pour cette partie, les données doivent être manipulées. La prédiction consiste à utiliser les entrées présentes, mais aussi les entrées et les sorties passées. Ainsi, le réseau sera en mesure de prédire ce s'en vient.

1 heure

Pour une prédiction de 1 heure, le vecteur de sortie contient 1 élément et le vecteur d'entrées est de longueur 9 : 4 entrées passées, 1 sorties passées et 4 entrées présentes. De plus, cela aura comme effet de diviser le nombre d'échantillon par 2.

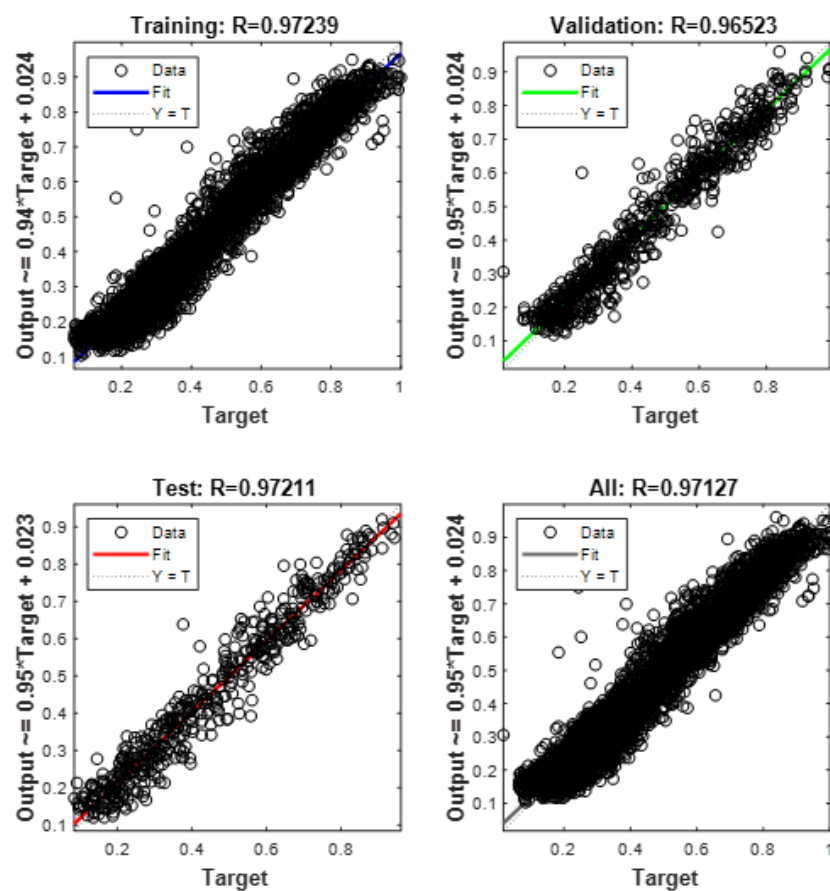


Figure 5: Résultats des régressions pour prédiction 1 heure

Les régressions sont tout à fait satisfaisantes pour approximer la production d'énergie. En effet, le facteur de corrélation est aussi élevé que la régression de base avec moins d'itérations (21 contre 36) et ce, malgré le nombre d'échantillon réduit. Pour tenter de pallier ce problème, seulement 10% des échantillons sont utilisés pour le test final.

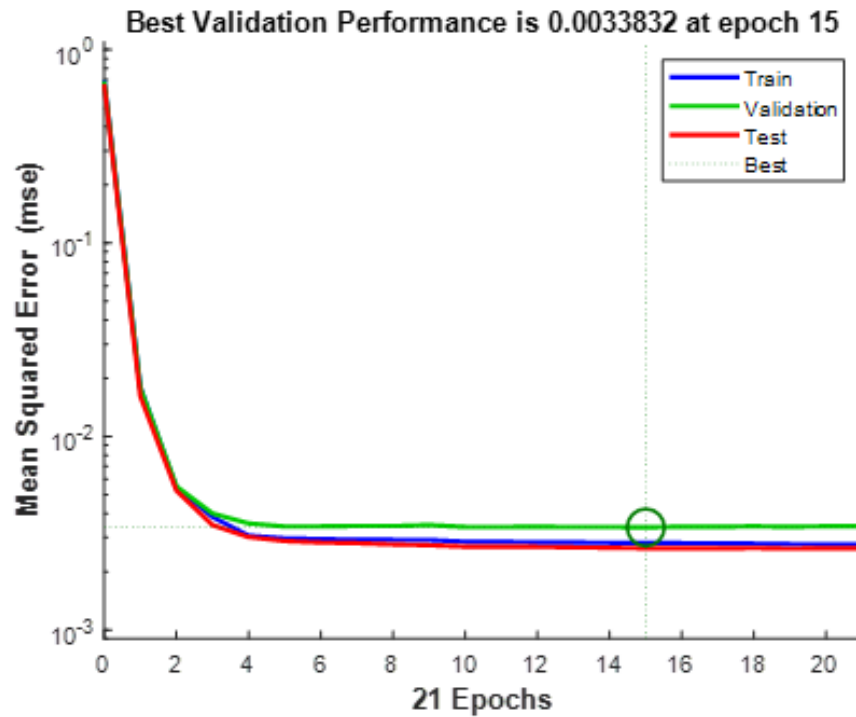


Figure 6: Erreur quadratique moyenne des régressions pour prédiction 1 heure

Le graphique du MSE présente elle aussi des valeurs extrêmement basses en fin d'évolution. Cependant, le MSE de la validation est légèrement plus élevé, mais pas assez pour semer le doute dans les résultats.

6 heures

Similairement à la partie précédente, les entrées et sorties passées sont combinées ensemble pour former un vecteur d'entrée de 29 éléments et une sortie d'un élément. Donc, 1594 échantillons sont présents. La division des échantillons est, encore une fois, 70/15/15.

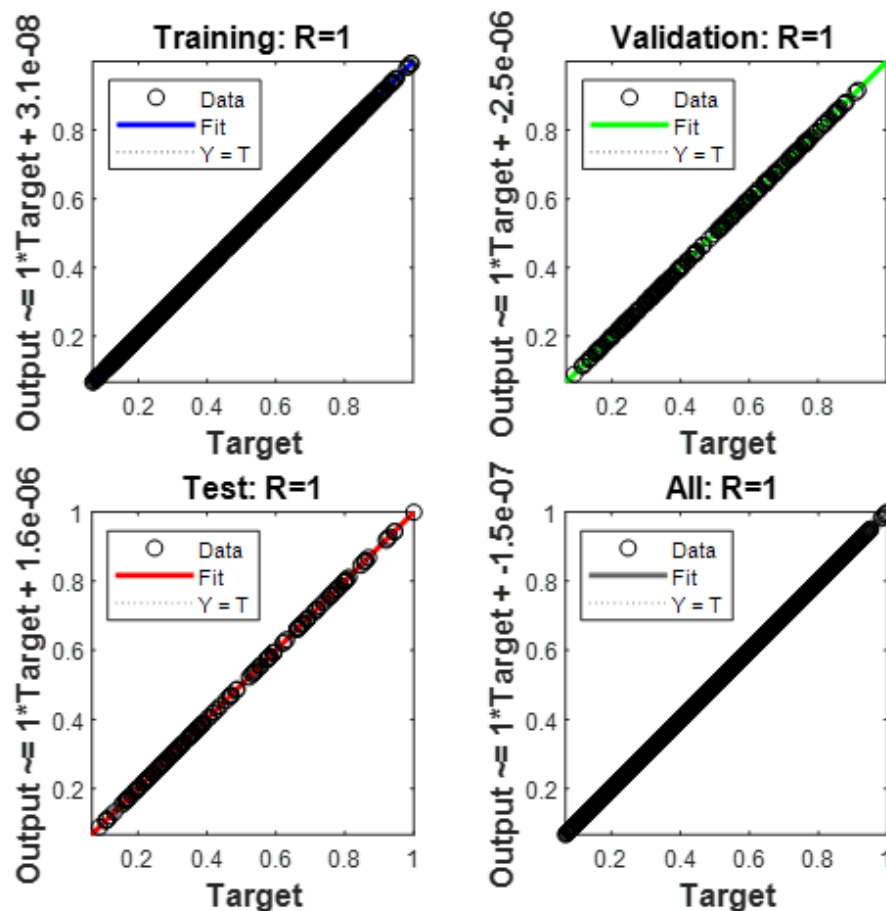


Figure 7: Résultats des régressions pour prédiction 6 heures

Comme le montre la figure ci-haut, les régressions sont presque parfaites et ce, peu importe le nombre de couche de neurones ou l'algorithme utilisé. De tels résultats sont particulièrement étrange, mais quelques scénarios peuvent offrir des pistes d'explication. Par exemple, il est possible que le réseau soit si performant que ça. En effet, Puisque le système peut se baser sur 5 sorties précédentes, il est peut-être capable d'être extrêmement efficace. Une autre piste d'explication serait le surapprentissage, mais, si c'était un cas de surapprentissage, la régression de test final serait un échec.

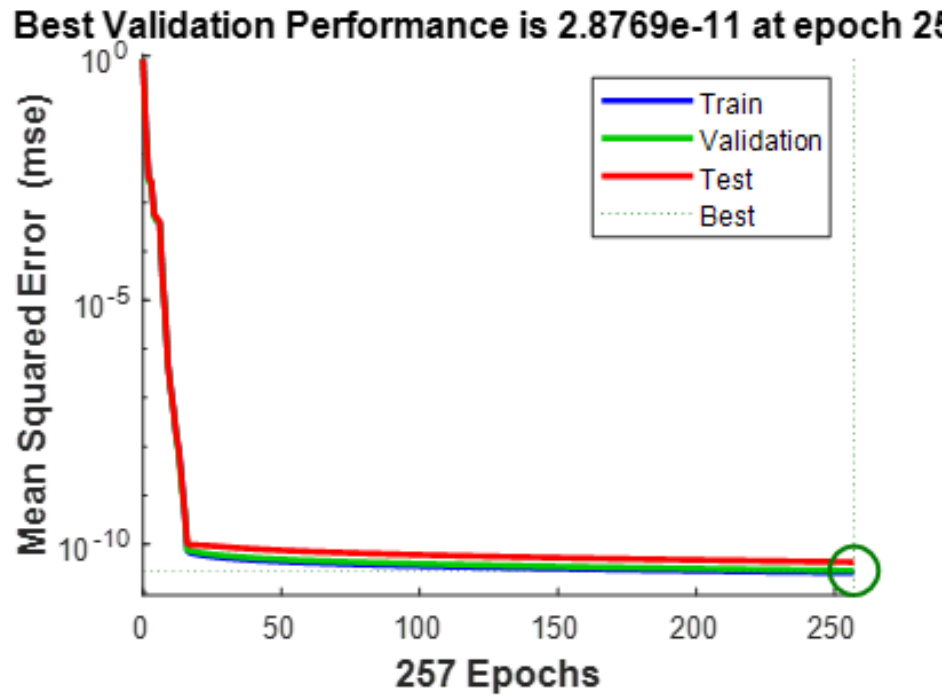


Figure 8: Erreur quadratique moyenne des régressions pour prédiction 6 heures

L'erreur quadratique moyenne est extrêmement faible et ce, dès la vingtaine d'itérations ce qui se reflète dans les régressions et le facteur de corrélation. Des erreurs quadratiques moyennes aussi basses pourrait confirmer que le réseau est si performant que ça, ou encore, indiquer que la source du problème affecte le MSE et les régressions.

24 heures

Encore une fois, les entrées et sorties passées sont combinées ensemble pour former un vecteur d'entrée de 119 éléments et une sortie d'un élément. Donc, 398 échantillons sont présents. La division des échantillons est 85/10/5 pour que le réseau puisse bien s'entraîner. Si moins d'échantillons sont offerts à l'entraînement, par exemple 70%, la validation et le test final sont légèrement moins performants

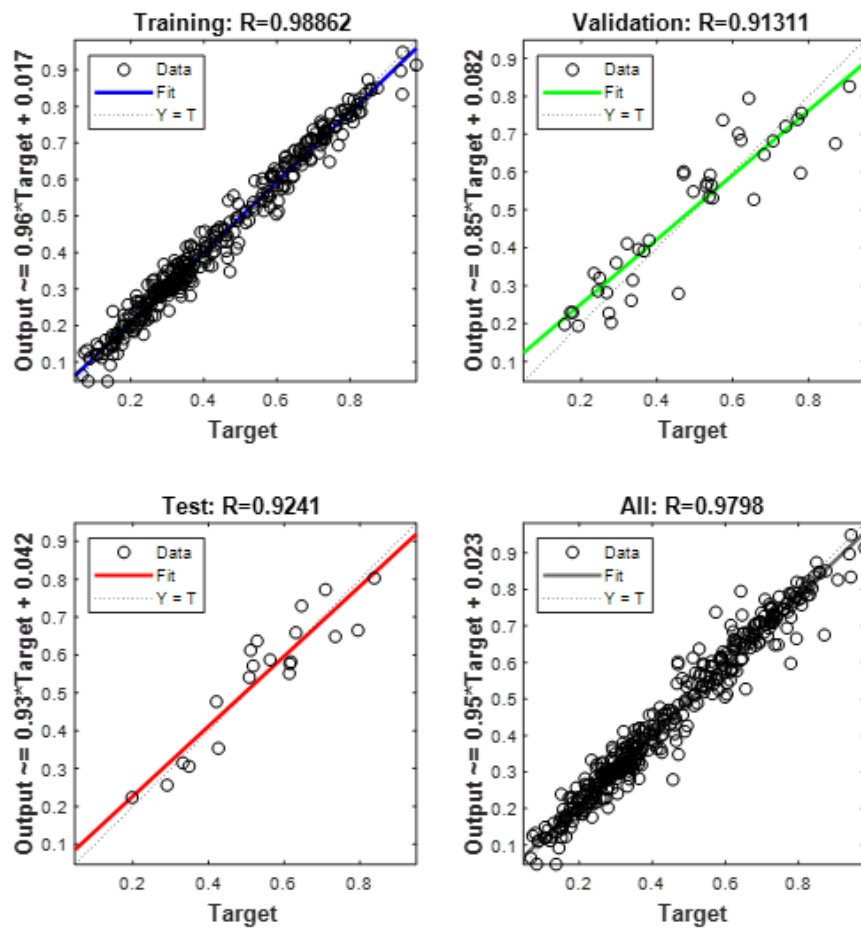


Figure 9.1: Résultats des régressions pour prédiction 24 heures avec division 85/10/5

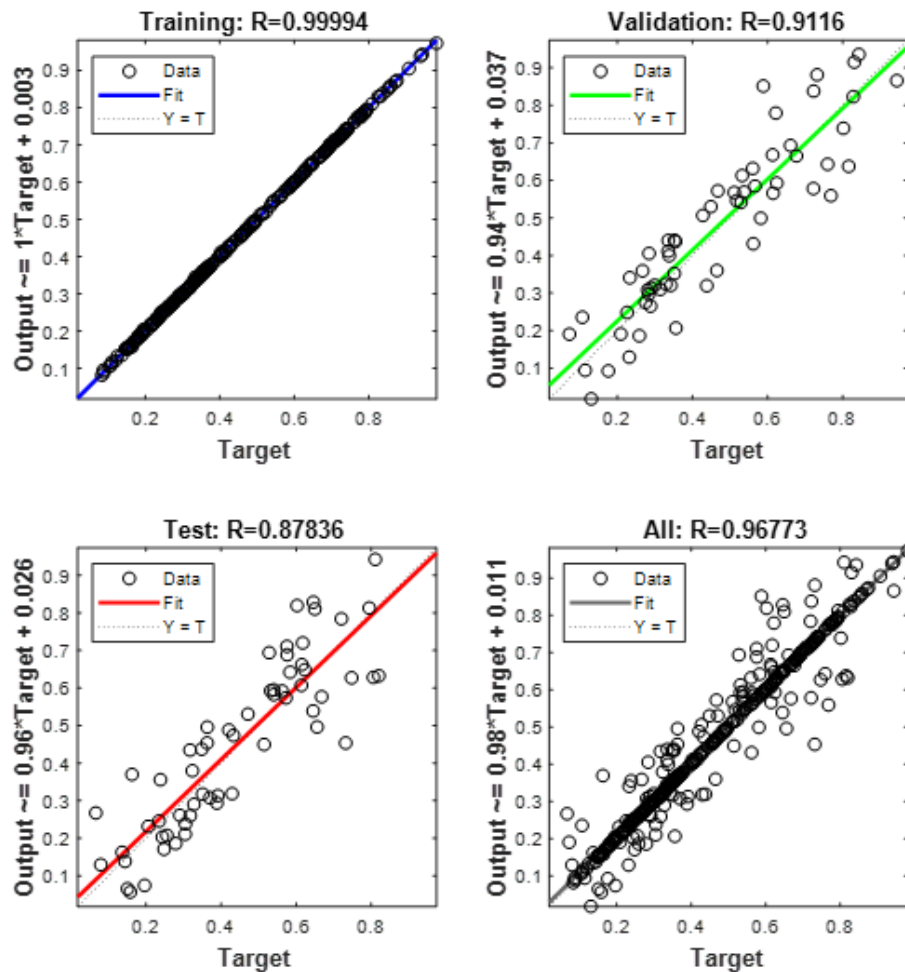


Figure 9.2: Résultats des régressions pour prédiction 24 heures avec division 70/15/15

Les figures 9.1 et 9.2 confirment que la division 85/10/5 produit de meilleures approximations d'énergies, puisque le facteur de corrélation est meilleur (0.924 contre 0.878). Ce ne semble pas si grand comme différence, mais la différence visuelle sur les graphiques est définitivement visible. De plus, pour donner suite à l'entraînement, le réseau avec de la figure 9.2 obtient un facteur de corrélation presque parfait, mais comme mentionné plus haut, le réseau est moins performant malgré tout.

Encore une fois, l'entraînement permet d'obtenir un réseau relativement efficace en moins d'itérations comparé aux réseaux de base et de 1 heure. Ainsi, utiliser les sorties et entrées passées plutôt qu'uniquement les entrées présentes permettent d'effectuer moins d'itérations.

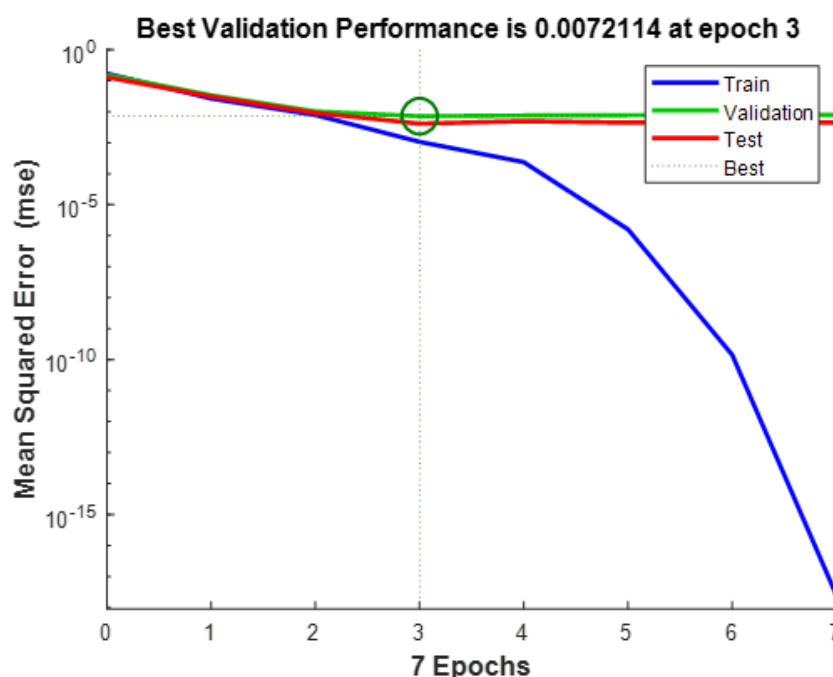


Figure 10: Erreur quadratique moyenne des régressions pour prédiction 24 heures

Visiblement, l'évolution de l'erreur quadratique moyenne est totalement différente que celle de la validation et du test final. Malgré la fin des itérations et du facteur de corrélations relativement satisfaisant, il semble que l'erreur quadratique moyenne de l'entraînement aurait pu chuter encore plus, similairement à la prédiction de 6h. Cependant, l'erreur quadratique moyenne de la validation et du test final sont tout à fait normal, quoique, légèrement plus élevé que le réseau de base et le réseau de prédiction de 1 heure. Comparer aux autres situations, celui-ci contient beaucoup moins d'échantillons (398 contre 1594, 4784 ou 9568). Cela explique la rapidité d'exécution de l'algorithme.

Conclusion

Finalement, le projet permet d'explorer la conception de RNA de base. En effet, grâce à l'outil de Matlab nftool, il est tout à fait possible de réaliser un RNA simple qui résout les problèmes de régressions de base, comme la situation initiale, ou, une fois le réseau conçu, il est possible d'approximer la sortie avec l'entrée.

De plus, il est possible de réaliser un RNA aussi précis que celui de base avec moins d'itérations. Cependant, il faut plus de données d'entrées et il faut trouver un juste équilibre entre le nombre d'entrées passées utiliser et la précision de l'approximation. Si trop de données passées sont utilisées, comme le cas de prédiction de 24 heures, l'approximation devient moins fiable puisque le nombre total d'échantillons utilisés pour l'entraînement est drastiquement réduit.