Compte rendu du projet MAT5016

Deep Neural Networks

Maxence DEBES
Vadim HEMZELLEC-DAVIDSON

18 Janvier 2026

Table des matières

1 Introduction

L'apprentissage profond, bien qu'incontournable aujourd'hui, a longtemps buté sur le problème de vanishing gradient qui empêchait l'entraînement efficace de réseaux profonds. Une proposition pour lever ce verrou a été en 2006 l'introduction du pré-entraînement non supervisé couche par couche (Greedy Layer-Wise Pre-training), une technique utilisant des Restricted Boltzmann Machines (RBM) pour extraire la structure des données avant tout apprentissage supervisé. L'objectif de ce projet est d'implémenter et d'analyser ces architectures (RBM, DBN, DNN) pour vérifier expérimentalement cette théorie. Nous évaluerons d'abord leur pouvoir génératif et leur capacité de reconstruction sur le jeu de données Binary AlphaDigit avant de quantifier l'avantage du pré-entraînement sur des tâches de classification avec MNIST, en testant spécifiquement la robustesse des modèles face à la profondeur du réseau et à la rareté des données d'entraînement.

2 Définitions

Nous rappelons brièvement les définitions des différentes architectures que nous allons étudier :

Un RBM pourra être représenté par un objet/une structure contenant un champ W (matrice de poids reliant les variables visibles aux variables cachées), un champ a (biais des unités d'entrée) et un champ b (biais des unités de sortie) Un réseau de neurones (DNN) et un Deep Belief Network (DBN) pourront être représentés par une liste de RBM, la taille de cette liste étant égale au nombre de couches cachés du réseau (+ couche de de classfication dans le cas du DNN). Chaque élement de cette liste coincidera donc à un RBM et contiendra un champ W (matrice de poids reliant 2 couches consécutives), un champ a (biais des unités d'entrée qui coincident avec les paramètres variationnels estimés) et un champ b.

3 Restricted Boltzmann Machine (RBM)

Cette première étude a été réalisée sur la base de données *Binary AlphaDigit*. L'objectif est de valider la capacité générative d'un RBM entraîné de manière non supervisée et d'analyser l'impact des hyperparamètres sur la qualité des données générées (via l'échantillonnage de Gibbs).

Les paramètres d'apprentissage fixés pour cette étude sont les suivants :

— Learning rate (α) : 0.1
— Nombre d'epochs : 50 ou 100
— Taille du mini-batch : 10
— Algorithme : Contrastive Divergence (CD-1)

3.1 Analyse de la capacité de représentation (nombre de neurones cachés)

Nous avons d'abord cherché à déterminer la taille optimale de la couche cachée (q) nécessaire pour reconstruire correctement un caractère donné. Nous avons fait varier q dans l'ensemble $\{100, 300, 700, 1000\}$ en entraînant le modèle sur une seule classe de caractères.

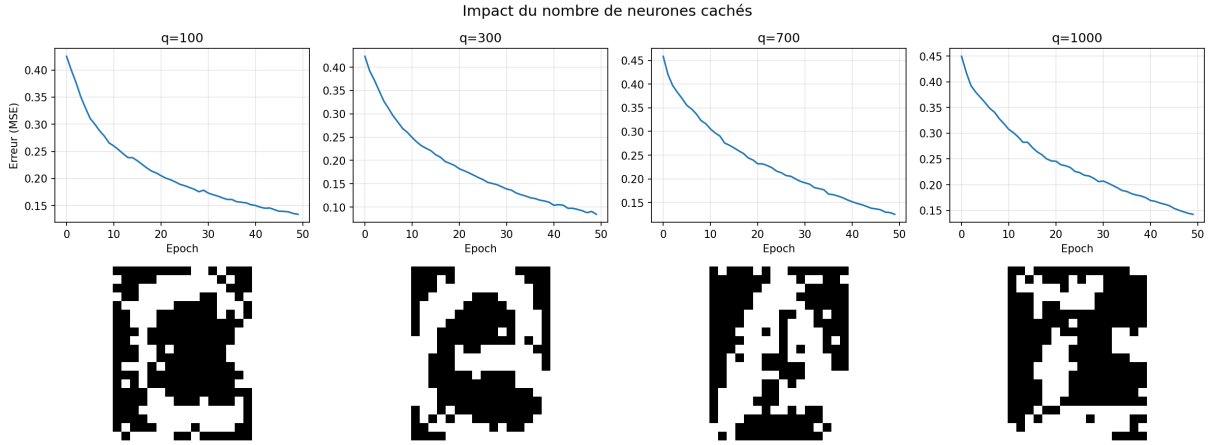

FIGURE 1 – Impact du nombre de neurones cachés (q) sur l'erreur quadratique moyenne (MSE) et la qualité visuelle des images générées après 50 epochs.

Comme l'illustre la Figure 1, nous observons deux phénomènes :

1. Quelle que soit la taille de la couche cachée, l'erreur de reconstruction (MSE) diminue de manière monotone, validant le bon fonctionnement de la descente de gradient.
2. **Qualité visuelle :**
   — Pour $q = 100$, l'image générée est grossière et manque de détails. Le réseau n'a pas assez de capacité mémoire pour capturer la géométrie précise du caractère.
   — Pour $q = 300$, nous observons un net saut qualitatif. La forme est claire, bien définie et l'erreur est minimale (≈ 0.08).
   — Pour $q = 700$ et $q = 1000$, l'amélioration est marginale, voire inexistante. L'erreur stagne autour de la même valeur qu'avec 300 neurones.

Il existe donc une taille critique (ici $q \approx 300$) au-delà de laquelle augmenter la complexité du modèle n'apporte plus de gain significatif pour cette tâche simple.

3.2 Analyse du pouvoir modélisant (diversité des données)

Dans un second temps, nous avons fixé la taille du réseau (par exemple $q = 200$) et nous avons augmenté progressivement la complexité de la tâche en apprenant simultanément plusieurs caractères différents : d'abord 'A', puis 'A' et 'B', jusqu'à l'ensemble 'A-E'.

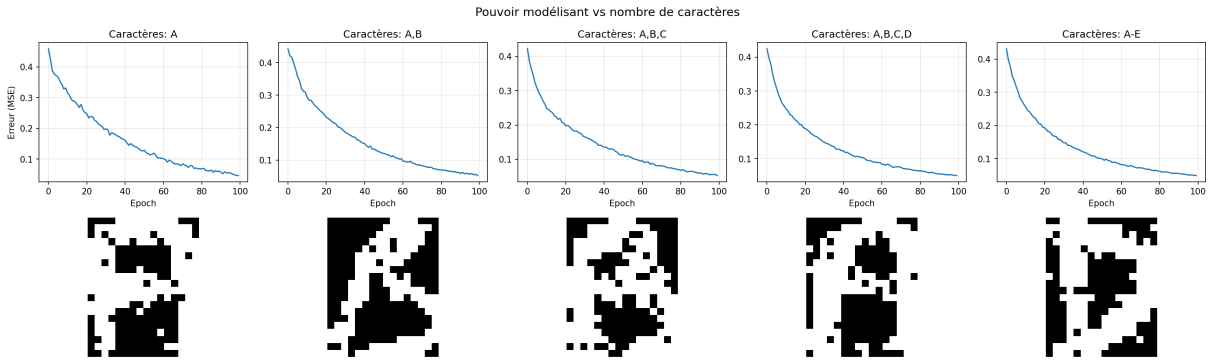

FIGURE 2 – Évolution de la qualité de génération en fonction du nombre de caractères appris simultanément par le RBM.

La Figure 2 met en évidence les limites d'un RBM simple :
— La génération avec un apprentissage mono-classe (seulement A par exemple ici) est excellente, le réseau se spécialise parfaitement.
— Pour l'apprentissage bi-classe (A,B ici), le réseau parvient à séparer les modes et génère soit un A, soit un B de manière distincte.
— À mesure que l'on ajoute des classes (C, D, E), la qualité des images générées se dégrade visiblement. On observe l'apparition de bruit et de formes « chimériques » (superposition moyenne de plusieurs lettres).

Ainsi, à taille constante, la capacité du RBM sature lorsque la diversité des données augmente. Pour modéliser des distributions plus complexes sans augmenter exponentiellement la largeur de la couche cachée, il devient nécessaire de passer à une architecture profonde (DBN) pour hiérarchiser les caractéristiques.

4 Deep Belief Network (DBN)

Afin de dépasser les limites du RBM simple, nous avons étendu l'architecture à un DBN. L'entraînement est effectué via l'algorithme Greedy Layer-Wise Pre-training, empilant plusieurs RBM entraînés séquentiellement.

Les hyperparamètres restent identiques à ceux de la section précédente ($\alpha = 0.1$, batch size=10, 100 epochs par couche), mais la structure du réseau varie selon les expériences.

4.1 Impact de la profondeur (nombre de couches)

Nous avons testé l'influence de la profondeur en ajoutant progressivement des couches cachées, tout en réduisant le nombre de neurones par couche pour forcer une compression de l'information (Architecture type : $320 \to 200 \to 100 \to 50 \to 25$).

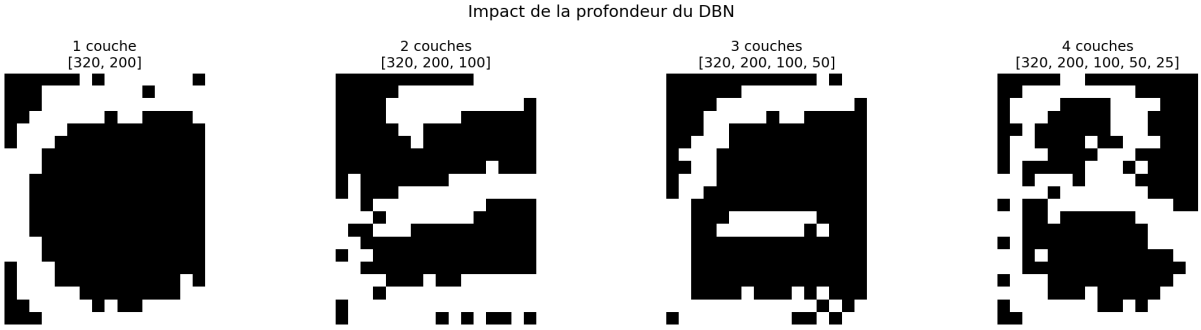

FIGURE 3 – Évolution de la qualité générative selon la profondeur du réseau. De gauche à droite : 1 couche à 4 couches cachées.

La Figure 3 illustre ces phénomènes :
— 1 et 2 couches : Le résultat est comparable à un RBM standard, avec des contours encore moyennement bien décrits.
— 3 couches ([... , 50]) : C'est le résultat visuellement optimal. La forme est nette, les traits sont continus et le bruit de fond est quasiment éliminé. Le réseau a réussi à abstraire la structure géométrique du caractère.
— 4 couches ([... , 25]) : La qualité reste bonne, mais la compression extrême (seulement 25 neurones dans la couche la plus profonde) commence à détériorer légèrement la finesse du tracé.

4.2 RBM vs DBN

Nous avons reproduit l'expérience de diversité (apprentissage de A à E) avec un DBN à 3 couches pour vérifier sa robustesse là où le RBM saturait.

FIGURE 4 – Génération par un DBN à 3 couches en augmentant le nombre de caractères appris (A, puis A-C, puis A-E).

Contrairement au RBM, le DBN maintient une génération de haute qualité même avec 5 caractères. Les lettres sont plus distinguables. Cela confirme que la profondeur augmente la capacité de représentation du modèle.

Enfin, la comparaison directe ci-dessous (Figure 5) permet de conclure sur l'apport du Deep Learning :

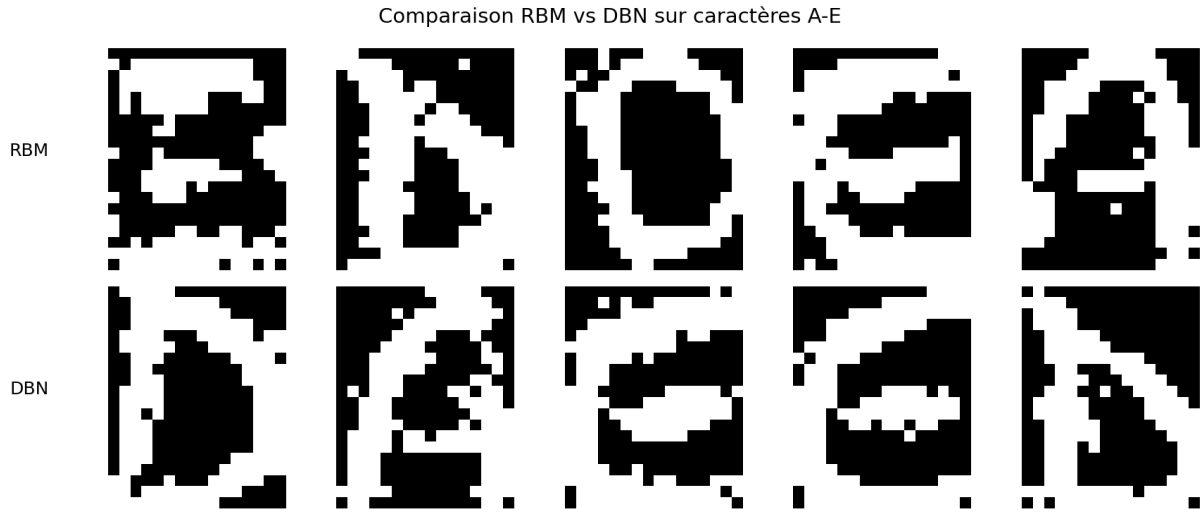

FIGURE 5 – Comparaison directe : RBM (haut) vs DBN (bas) sur l'apprentissage simultané de 5 caractères.

Alors que les images générées par le RBM sont bruitées et présentent des discontinuités, celles du DBN sont beaucoup plus lisses et les lettres sont plus distinguables.

5 Deep Neural Network (DNN) - Étude sur MNIST

Nous nous intéressons ici à la tâche de classification supervisée sur la base MNIST (70 000 chiffres manuscrits, images 28×28 binarisées). L'objectif est de quantifier l'apport du pré-entraînement non supervisé face à la profondeur du réseau, la largeur des couches et la rareté des données annotées.

5.1 Protocole expérimental

Nous comparons systématiquement deux approches pour entraîner un DNN :

1. Avec pré-entraînement : Le réseau est initialisé par un empilement de RBM entraînés couche par couche (non supervisé), puis affiné par rétro-propagation (supervisé).
2. Sans pré-entraînement : Le réseau est initialisé aléatoirement (loi normale $\mathcal{N}(0, 0.01)$), puis entraîné directement par rétro-propagation.

Les hyperparamètres fixés sont : $\alpha = 0.1$, batch size=100, 100 epochs de pré-entraînement (si applicable) et 200 epochs de rétro-propagation.

5.2 Analyse des résultats

5.2.1 Impact de la profondeur du réseau

Nous avons fait varier le nombre de couches cachées de 2 à 5 (avec 200 neurones par couche).

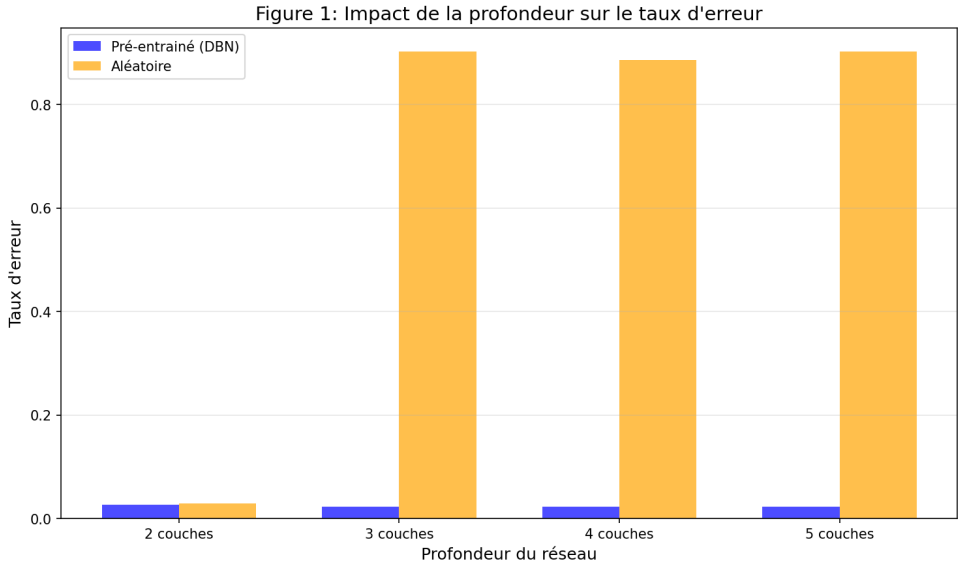

FIGURE 6 – Comparaison du taux d'erreur en fonction du nombre de couches cachées (taille fixe 200).

La Figure 6 illustre le problème de vanishing gradient :

— Dès 3 couches cachées, l'erreur explose à $\approx 90\%$ (soit le hasard pur sur 10 classes), donc l'apprentissage échoue totalement.

— Le pré-entraînement permet de maintenir une performance excellente et stable ($\approx 2.3\%$ d'erreur), quelle que soit la profondeur.

5.2.2 Impact de la largeur des couches

À profondeur fixe (2 couches), nous avons fait varier le nombre de neurones de 100 à 700.

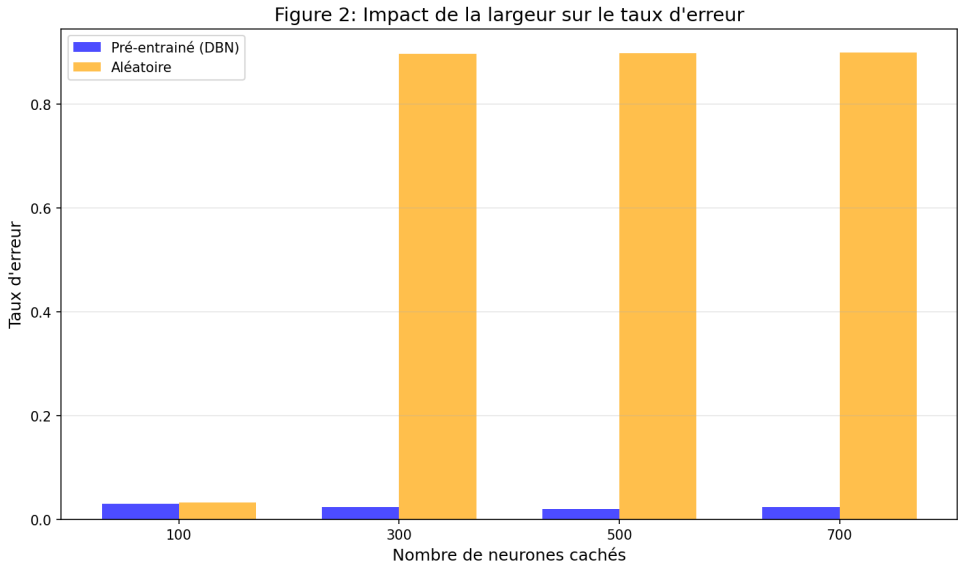

FIGURE 7 – Comparaison du taux d'erreur en fonction de la largeur des couches cachées (2 couches).

La Figure 7 montre que pour des couches larges (≥ 300 neurones), le réseau aléatoire échoue à converger (erreur $\approx 90\%$), en raison de saturation des sigmoïdes, annulant le gradient. Le réseau pré-entraîné contourne ce problème en maintenant une erreur basse ($\approx 2\%$) même avec 700 neurones.

5.2.3 Impact de la taille du jeu d'entraînement

Enfin, nous avons testé la capacité de généralisation en limitant artificiellement le nombre de données d'apprentissage (de 1000 à 56 000 exemples).

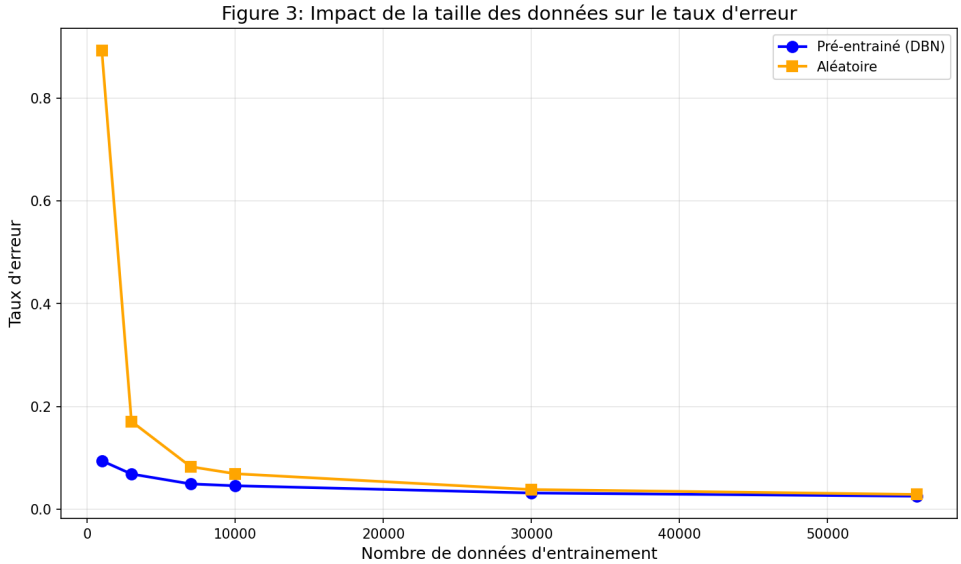

FIGURE 8 – Comparaison du taux d'erreur en fonction du nombre de données d'apprentissage disponibles.

La Figure 8 démontre qu'à faible volume de données (< 3000), l'écart est très important. Avec 1000 exemples, le réseau aléatoire est en échec (89% d'erreur), tandis que le DBN parvient déjà à une performance correcte (9.4% d'erreur). À mesure que la base de données grandit (vers 60 000), l'écart se resserre. Avec suffisamment de données, la rétro-propagation pure finit par trouver une bonne solution, rendant le pré-entraînement moins critique (mais toujours légèrement bénéfique : 2.5% contre 2.9% d'erreur).

5.3 Meilleure performance obtenue

En combinant les enseignements précédents, nous avons entraîné un réseau optimal de grande capacité : architecture [784, 500, 500, 500] sur l'ensemble des données, avec une précision finale sur le test set de **98.03**%, ce qui est excellent.

6 Conclusion

Cette étude a permis de confirmer expérimentalement le rôle crucial du pré-entraînement non supervisé dans la mise en œuvre de réseaux de neurones profonds. Les expériences sur Binary AlphaDigit ont d'abord montré que le DBN offre un pouvoir génératif supérieur à un RBM simple, produisant des données plus structurées et moins bruitées. Par la suite,

l'analyse sur MNIST a démontré que l'initialisation par empilement de RBMs est importante afin d'entraîner efficacement des réseaux profonds ou larges là où l'initialisation aléatoire échoue à cause de vanishing gradient et de la saturation. Enfin, nous avons mis en évidence que cette approche est particulièrement bénéfique dans les scénarios où les données annotées sont rares.