# IFT6390 - Devoir 3

Pierre-Antoine Bernard (20077521)
Simo Hakim (20096040)

Décembre 2023

**Q1**: Pour cette question, il était demandé d'entraîner pour 50 époques un modèle MLP avec 2 couches cachées contenant 128 neurones, avec différents taux d'apprentissage $\ell$. Ce travail a été réalisé pour $\ell \in \{0.01, 1 \times 10^{-4}, 1 \times 10^{-8}\}$ et l'erreur quadratique moyenne pour les prédictions sur les données de validation sont illustrées à chaque époque dans la figure ci-dessous. On note que la

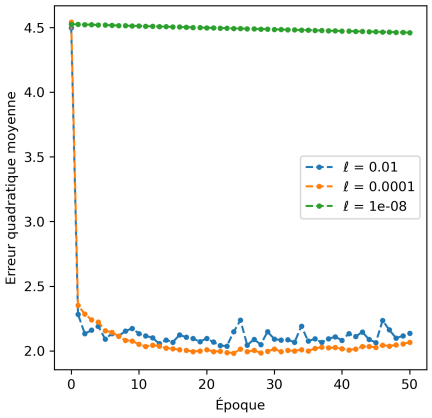


Figure 1: Erreur quadratique moyenne en fonction des époques pour un modèle MLP sur les données CIFAR10, à différents taux d'apprentissage $\ell$

meilleure performance est obtenue avec un pas de $\ell = 1 \times 10^{-4}$. Pour un pas trop petit, comme $\ell = 1 \times 10^{-8}$, l'erreur quadratique diminue de manière monotone avec les époques. Toutefois, son taux de réduction à chaque époque est très faible en comparaison à celui atteint avec le pas optimal de $\ell = 1 \times 10^{-4}$. Choisir un pas trop petit mène donc à une augmentation non-nécessaire du temps de calcul pour atteindre des résultats équivalents.

À l'opposée, un pas trop grand peut aussi mener à un apprentissage sous-optimal. En effet, pour $\ell = 0.01$, on note un comportement non-monotone et plus chaotique de l'erreur quadratique en fonction des époques. Puisque l'apprentissage est basé sur une descente par gradient, il est important que le pas $\ell$ soit assez petit pour qu'à chaque étape les nouveaux paramètres restent dans l'intervalle où la fonction à de coût à optimiser est *linéaire*. Comme illustré dans la Figure 1, un pas trop grand peut faire en sorte que la descente de gradient dépasse les paramètres optimaux et que l'erreur moyenne varie significativement à chaque époque.

**Q2:** Ici, on nous demandait d'entraîner un réseau CNN sur 50 époques avec une taille de batch 128, un taux d'apprentissage de $\ell = 0.001$ et des fonctions d'activation ReLU. Le réseau demandé contient trois couches convolutionnelles cachées avec un nombre de filtres de $(16, 32, 45)$, des noyaux de taille 3, et un pas de convolution de 1. L'erreur quadratique moyenne sur les données de validation obtenue à chaque époque est illustrée ci-dessous.

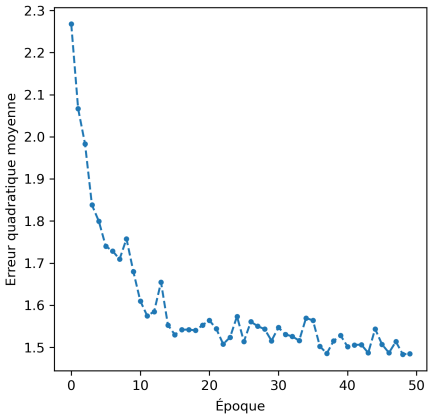


Figure 2: Erreur quadratique moyenne pour un CNN sur les données CIFAR10, en fonction du nombre d'époques d'entraînement

**Q3.a**: Pour un réseau ayant l'architecture décrite, la partie convolutive se décompose comme une combinaison de réseaux complètement connectés *indépendants*. Pour chaque paire de deux couches consécutives, on a pour chaque attribut de la première couche un réseau complètement connecté à un nombre de neurones de la deuxième couche correspondant à son nombre de filtres. Il est important de noter que chaque attribut de la première couche est connecté à des neurones de la deuxième par un *unique* réseau complètement connecté. Aussi, chaque neurone de la deuxième couche est uniquement connecté à un seul attribut de la couche précédente, formant donc des réseaux indépendants.

**Q3.b**: Pour le type d'architecture décrit, on a que chaque filtre de chaque couche convolutive correspond à une couche complètement connectée. En effet, puisque la taille du noyau est la même que la taille de l'image il n'y a pas de partage de poids car la convolution ne fait aucun pas.

On demande le nombre de filtre pour que ce modèle ait le même nombre de paramètres que celui de la question 2. Le modèle de la question 2 possède:

1. $(3 \times 3 \times 3) \times 16 + 16 = 448$ paramètres pour la première couche;
2. $(16 \times 3 \times 3) \times 32 + 32 = 4640$ paramètres pour la deuxième couche;
3. $(32 \times 3 \times 3) \times 45 + 45 = 13,005$ paramètres pour la troisième couche;
4. $(45 \times 4 \times 4) \times 128 + 128 = 92,288$ pour la couche dense;
5. $128 + 1 = 129$ pour la couche de sortie.

Au total le modèle de la question 2 possède donc 110510 paramètres. Or, le modèle superficiel de cette question contient pour $L$ filtres:

1. $3 \times 32 \times 32) \times L + L = 3073L$ paramètres pour la première couche
2. $(L \times 4 \times 4) \times 128 + 128 = 2048L + 128$ paramètres pour la couche dense
3. $128 + 1 = 129$ pour la couche de sortie.

Il possède donc $5121L + 257$ paramètres. Il faut donc environ $L = 22$ filtres.
**Q3.c**: Le résulat de l'entraînement des différents modèles est présenté dans la Figure 3 de la page suivante.

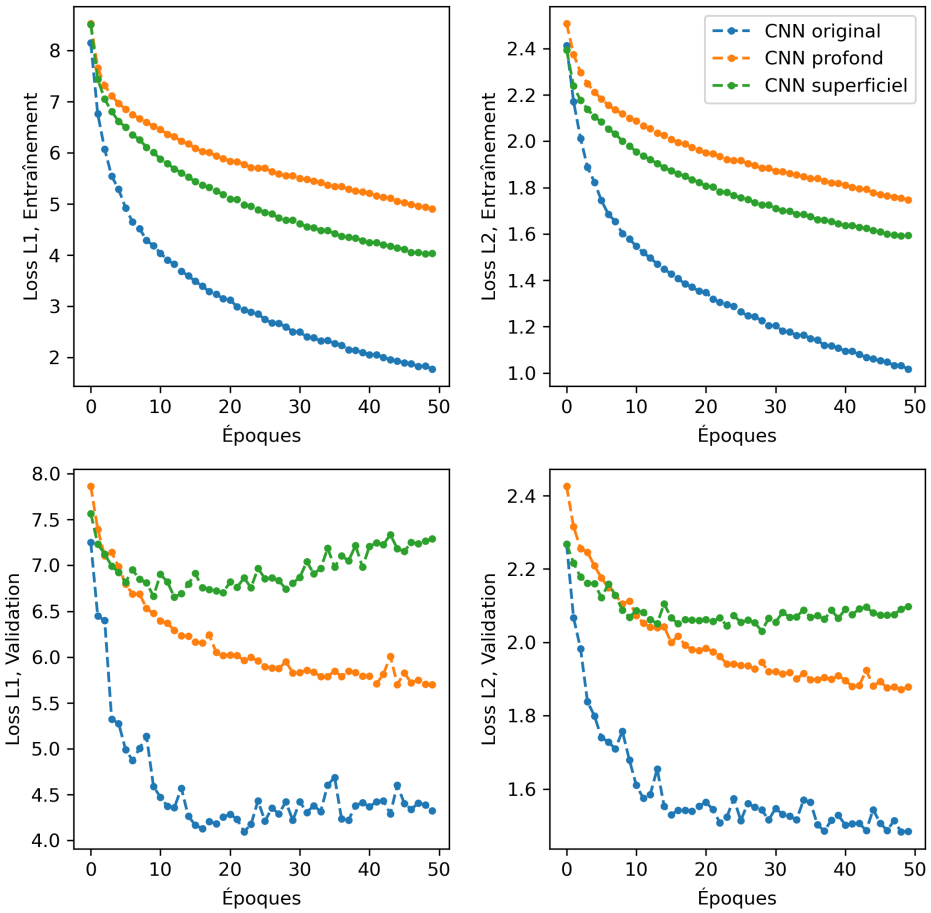


Figure 3: Fonction de coût $L1$ et $L2$ sur les données d'entraînement et de test, pour différentes architectures de réseaux convolutifs entraînés sur 50 époques

Parmi les trois modèles, on note que les performances à l'entraînement sont optimale le réseau original qui termine avec un coût L1 de 1.77 pour un coût L1 initial de l'ordre de 8. En contraste, les réseaux profond et superficiel termine avec un coût de l'ordre de $4 - 5$ pour un coût initial similaire. Des résultats semblables ont été obtenus pour le coût L2.

Pour la validation, on note que les performances des trois modèles sont assez différentes. Le modèle offrant la meilleure généralisation semble être le modèle original, doté de convolutions non-triviales (distinctes de couches complètement connectées). En effet, ce dernier termine les 50 époques avec une MAE de validation de 1.48 pour une MAE initiale de l'ordre de 2.5. Cette performance est naturellement inférieure à celle obtenue sur les données d'entraînement, mais reste meilleure que celle des deux autres modèles pour la validation. En effet, le modèle profond et superficiel terminent respectivement avec une MAE 1.87 et 2.09.

De ces observations, on peut donc déduire que, pour les modèles traitant des données ayant une structure spatiale (e.g. les images de CIFAR10), un CNN avec de vraies couches convolutives offre les meilleures performances. Puisque le modèle profond offre une meilleure performance sur les données de validation que le modèle superficiel, on note aussi qu'un réseau avec une structure plus profonde semble avoir un plus grand potentiel de généralisation, pour un même nombre de paramètres.