Rapport 2

Machine Learning

Julien MOLINIER

Rappel du sujet 2 : une topologie fixe (mais raisonnable), étudier l'impact de l'algorithme d'optimisation (Momentum, adam, RMSProp, …).

Choix de topologie :

- Une couche dense de 100 neurones avec une dimension des entrées égale à 784 et une sigmoïde pour fonction d'activation
- Une couche dense de 100 neurones avec une dimension des entrées égale à 100 et une sigmoïde pour fonction d'activation
- Une couche dense de 10 neurones avec softmax pour fonction d'activation

L'apprentissage s'effectue avec 15 époques et un batch size de 128. On conserve 25% des images prévues initialement pour l'apprentissage pour la validation à la fin de chaque époque.

I. Comparatif des optimiseurs (MNIST) :

	sgd	RMSProp	adam	nadam	adadelta	adamax	adagrad
Accuracy (%)	77.12	97.15	97.45	97.53	96.70	96.79	85.66
Temps d'apprentissage (s)	13.17	14.70	15.64	16.26	15.43	14.83	14.02

Les résultats précédents montrent clairement que l'optimiseur sgd est plus rapide que les autres mais il obtient surtout le pire résultat. Nadam obtient le meilleur résultat mais au détriment du temps d'exécution.

Lors du lancement de l'algorithme on obtient la matrice de confusion suivante pour l'optimiseur nadam :

$$\begin{pmatrix} 967 & 0 & 2 & 1 & 0 & 2 & 5 & 2 & 1 & 0 \\ 0 & 1127 & 2 & 1 & 0 & 1 & 2 & 1 & 1 & 0 \\ 6 & 0 & 1014 & 1 & 1 & 0 & 3 & 4 & 3 & 0 \\ 0 & 1 & 6 & 984 & 0 & 6 & 0 & 4 & 6 & 3 \\ 0 & 0 & 3 & 1 & 951 & 0 & 6 & 2 & 3 & 16 \\ 4 & 0 & 0 & 11 & 2 & 860 & 4 & 1 & 7 & 3 \\ 7 & 3 & 5 & 1 & 3 & 3 & 933 & 0 & 3 & 0 \\ 0 & 2 & 11 & 6 & 0 & 1 & 0 & 999 & 1 & 8 \\ 5 & 0 & 6 & 3 & 3 & 5 & 3 & 4 & 943 & 2 \\ 2 & 3 & 0 & 7 & 9 & 3 & 0 & 6 & 4 & 975 \end{pmatrix}$$

Et voici les courbes d'évolution de la fonction de coût en fonction des itérations d'apprentissage pour les optimiseurs nadam et sgd :

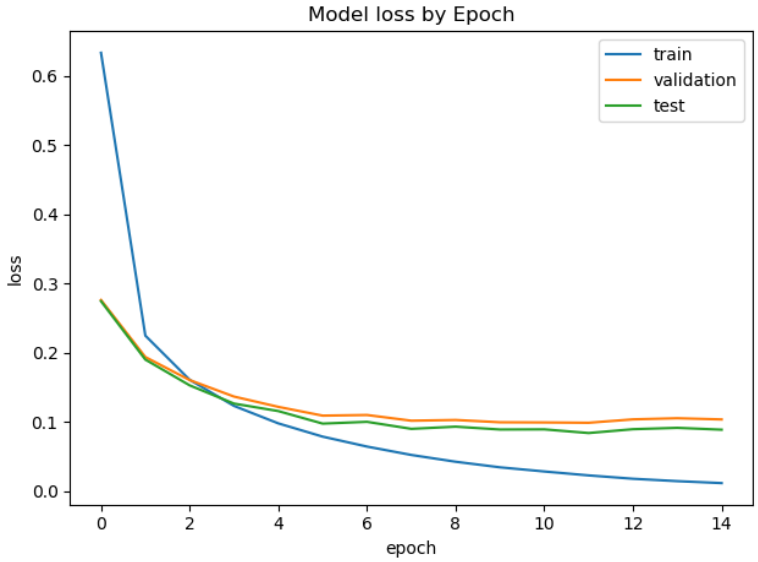

Figure 1 : Evolution de la fonction de coût avec l'optimiseur nadam

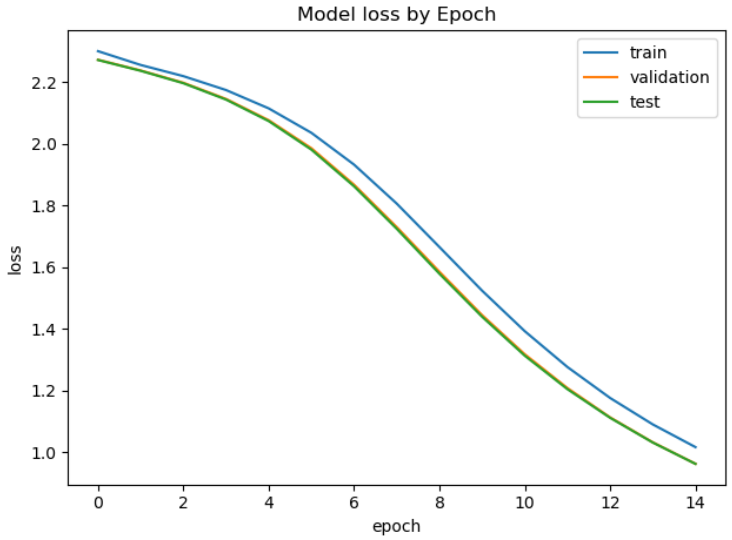

Figure 2 : Evolution de la fonction de coût avec l'optimiseur sgd

La première courbe montre clairement un sur-apprentissage à partir de la 3ème époque. Le modèle devient trop fin par rapport aux données de test dont la précision stagne à partir de la 5ème époque. Au contraire la seconde courbe montre que le modèle n'est pas assez entrainé. Pour finir l'optimiseur nadam est plus rapide pour progresser au début de l'entrainement. Sgd évolue plus linéairement.

II. Comparatif des optimiseurs (FMNIST) :

	sgd	RMSProp	adam	nadam	adadelta	adamax	adagrad
Accuracy (%)	68.56	87.44	87.11	87.62	85.84	86.83	85.62
Temps d'apprentissage (s)	13.05	14.29	14.81	15.64	15.12	14.93	13.56

On remarque que les images de la base de données fashion mnist portent plus facilement à confusion ce qui impacte fortement la précision de l'algorithme sur les données de test.

Lors du lancement de l'algorithme on obtient la matrice de confusion suivante pour l'optimiseur nadam :

$$\begin{pmatrix} 759 & 2 & 16 & 54 & 3 & 1 & 155 & 0 & 10 & 0 \\ 1 & 965 & 1 & 25 & 3 & 0 & 4 & 0 & 1 & 0 \\ 5 & 1 & 812 & 18 & 75 & 0 & 87 & 0 & 2 & 0 \\ 15 & 6 & 16 & 922 & 15 & 0 & 21 & 0 & 5 & 0 \\ 0 & 1 & 114 & 64 & 742 & 0 & 75 & 0 & 4 & 0 \\ 0 & 0 & 0 & 1 & 0 & 950 & 0 & 23 & 2 & 24 \\ 78 & 3 & 75 & 46 & 57 & 0 & 727 & 0 & 14 & 0 \\ 0 & 0 & 0 & 0 & 0 & 26 & 0 & 948 & 0 & 26 \\ 3 & 1 & 3 & 4 & 3 & 2 & 2 & 4 & 978 & 0 \\ 0 & 0 & 0 & 0 & 0 & 7 & 1 & 33 & 0 & 959 \end{pmatrix}$$

Cette matrice montre une confusion non négligeable entre la classe 1 et la classe 7, c'est-à-dire entre les tee-shirts et les chemises. En effet ces vêtements sont très ressemblants et la qualité en termes de pixels de l'image n'est pas suffisante pour éviter la confusion. Ainsi 78 tee-shirts sont reconnus comme des chemises et 155 chemises comme des tee-shirts. On observe le même problème entre les classes 3 et 5 respectivement pullovers et vestes.

Voici les courbes d'évolution de la fonction de coût en fonction des itérations d'apprentissage pour les optimiseurs nadam et sgd :

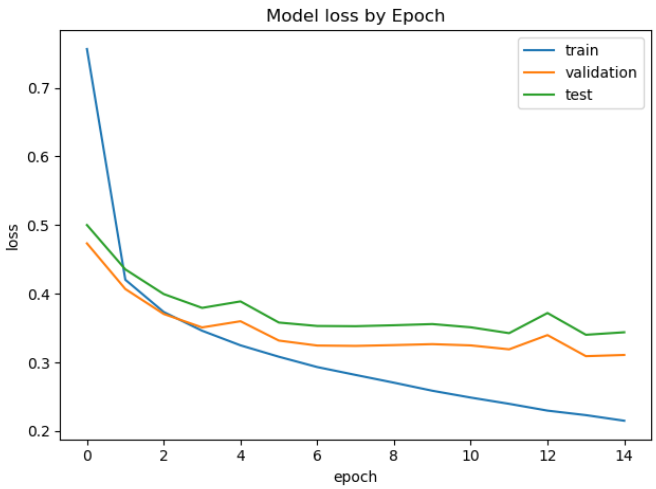

Figure 3 : Evolution de la fonction de coût avec l'optimiseur nadam

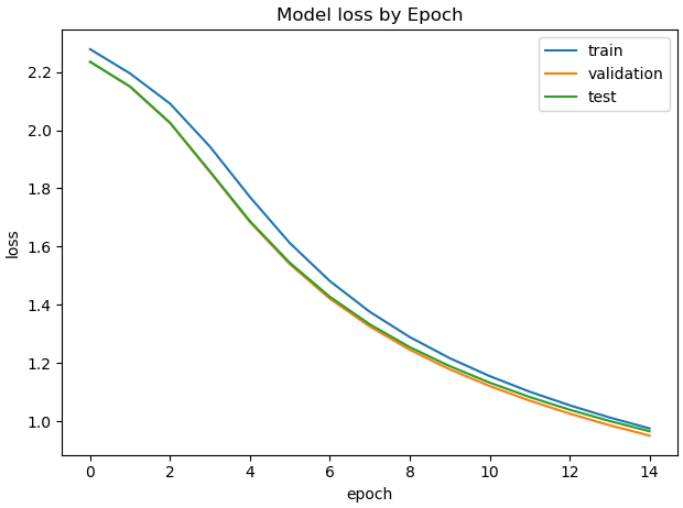

Figure 4 : Evolution de la fonction de coût avec l'optimiseur sgd

Le constat est le même que pour les graphes avec la base MNIST. En revanche le premier graphe montre des sursauts du taux d'erreur signe d'une sortie d'un minimum local lors de la descente de gradient.

III. Conclusion

Pour conclure nous pouvons dire le choix d'optimiseur dépend des qualités que nous souhaitons. En effet l'optimiseur nadam avec la meilleure précision possède le plus long temps d'exécution. Inversement sgd est le plus rapide mais possède la pire précision. Si on décide de faire un compromis il apparaît que l'optimiseur RMSProp soit une bonne solution ici.