# Comparaison des capacités prédictives de réseaux de neurones, application à la masse sèche de cellules

Romain BAILLY[1, 4] Marielle MALFANTE[1] Cédric ALLIER[2, 3] Lamya GHENIM[4] Jérôme MARS[5]

[1]Univ. Grenoble Alpes, CEA, List, F-38000 Grenoble, France

[2]Univ. Grenoble Alpes, CEA, Leti, F-38000 Grenoble, France

[3]Janelia Research Campus, Howard Hughes Medical Institute, Ashburn, VA, USA

[4]Univ. Grenoble Alpes, INSERM, CEA-IRIG, BGE, Biomics, Grenoble, F-38000, France

[5]Univ. Grenoble Alpes, CNRS, Grenoble INP, GIPSA-Lab, 38000 Grenoble, France

**Résumé –** Depuis une dizaine d'années, de nombreuses architectures de réseaux de neurones ont été proposées pour résoudre des problèmes complexes, jusqu'alors insolubles. Bien que chaque architecture tente d'augmenter les performances de ses prédécesseurs, nous constatons aujourd'hui un manque d'évaluation comparée de leurs performances. Nous présentons dans ce papier une étude comparative de différentes architectures des réseaux de neurones pour la prédiction de séries temporelles. En particulier, celle de masse sèche de cellules. Quatre types d'architectures (Perceptrons Multicouches, CNN-1D, LSTM et réseaux à connexions résiduelles) sont comparées selon leurs capacités prédictives, leur nombre de paramètres, leur temps d'entraînement et leur temps d'inférence. Les expériences réalisées mettent en avant une prédominance des perceptrons multicouches à extraire une représentation utile pour la prédiction de la masse sèche de cellule par un réseau totalement connecté, et ce, sur toutes les métriques étudiées.

**Abstract –** Over the last decade, many neural network architectures have been proposed to solve complex, previously unsolvable problems. Although each architecture tries to increase the performances of its predecessors, we note today a lack of comparative evaluation of their performances. We present in this paper a comparative study of different neural network architectures for time series prediction. In particular, that of dry mass of cells. Four types of architectures (Multilayer Perceptrons, CNN-1D, LSTM and CNN with skipped connections) are compared according to their predictive capabilities, their number of parameters, their training time and their inference time. The experiments carried out show a predominance of multilayer perceptrons for the prediction of cell dry mass according to all the metrics studied.

## 1 Introduction

Cette étude se place dans le contexte de la vidéo-microscopie sans lentille [2], une technique émergente permettant de capturer des images contenant des dizaines de milliers de cellules, par opposition aux méthodes d'acquisition classique ne permettant l'acquisition que de quelques cellules. Ainsi, des séries temporelles décrivant la vie de ces cellules pendant plusieurs jours peuvent être extraites [1].

La prédiction du futur de la cellule et en particulier sa division est d'une importance capitale. En effet, bien que la division de la cellule soit un mécanisme compris aujourd'hui, il n'y a pas encore de consensus concernant la croissance en masse des cellules entre deux divisions. C'est pourquoi nous cherchons à prédire la masse future de cellules.

En plus de l'étude absolue des résultats de prédiction, cette étude se positionne dans un contexte d'apprentissage auto-supervisé pour lequel la tâche de prédiction n'est qu'une tâche prétexte permettant d'apprendre une représentation des séries temporelles sans aucun label. Cette représentation peut ensuite être utilisée pour réaliser une autre tâche d'intérêt comme de la classification ou de la détection d'anomalies [3]. C'est pourquoi, nous proposons d'étudier des architectures de réseaux de neurones pour prédire le futur de la masse sèche des cellules et pas d'autres méthodologies.

Cette étude a pour but de comparer les perceptrons multicouches (sec 3.1), les réseaux convolutionnels à une dimension (sec 3.2), les réseaux LSTM (sec 3.3) et les réseaux à connexions résiduelles (sec 3.4) utilisables pour cette **tâche de prédiction de série temporelle**. Ces architectures sont à la fois comparées selon leurs **performances prédictives**, mais aussi leur **empreinte mémoire** et leur **temps d'entrainement** et d'**inférence**. Des architectures plus lourdes (comme les transformers par exemple) ne sont pas étudiées ici au regard de l'aspect embarqué impliqué par le microscope sans lentille.

## 2 Jeu de données

Les séries temporelles utilisées pour cette étude sont extraites d'un jeu de données d'images de cellules HeLa grâce à la méthode présentée dans [1]. Ce jeu de données contient différentes séries temporelles décrivant la vie de plus de 29 000 cellules suivies pendant une durée de 80 heures à raison d'un point toutes les 10 minutes. Nous nous concentrons dans ce papier sur une série temporelle en particulier : la masse sèche des cellules, c'est-à-dire la masse des cellules si elles avaient été privées de leur eau.

La masse sèche d'une cellule suit globalement la tendance présentée figure 1 : la masse de la cellule augmente jusqu'à sa division où elle est environ divisée par deux entre les deux cellules filles (en vert sur la figure 1).

Nous considérerons des fenêtres de 30 heures, découpées en

20 heures d'entrées, connues par le réseau de neurones, puis 10 heures en sortie à prédire. Un fenêtrage glissant [4] est utilisé sur les séries temporelles de plus de 30 heures. Ainsi, le jeu de données d'entraînement contient 189 565 séries temporelles provenant de 3 879 cellules différentes et le jeu de validation contient 53 056 fenêtres provenant de 557 cellules.

# 3 Méthodes

Les sections 3.1 à 3.4 présentent les architectures mises en place dans cette étude comparative, ainsi que les différents hyperparamètres utilisés. Le tableau 1 résume les valeurs d'hyperparamètres choisies pour chaque architecture.

## 3.1 Perceptrons Multicouches

Les perceptrons multicouches sont constitués d'un nombre variable de neurones par couches cachées. Ces neurones réalisent une combinaison linéaire de leurs entrées pondérées par les poids du réseau.

**Paramètres** Les paramètres identifiés pour ces réseaux sont le nombre de couches et le nombre de neurones par couches $\mathcal{D}$. Des fonctions d'activation `reLu` sont utilisées entre les couches denses.

La couche de sortie de tous les réseaux présentés dans cette étude est une couche dense (sans fonction d'activation) de taille 60 utilisée pour prédire, en une fois, les 10 prochaines heures de la série temporelle.

## 3.2 Réseaux convolutifs (CNN) 1D

### CNN à maxpooling

Les réseaux convolutifs consistent en un empilement de couches de convolutions apprenant des filtres et de couches de maxpooling. Ces filtres appris sont alors particulièrement pertinents pour décrire le phénomène observé. Les réseaux convolutifs à une dimension [8] sont les équivalents à une dimension des CNN-2D utilisés en vision par ordinateur [7]. Les CNN-1D fonctionnement comme leur version à deux dimensions en apprenant des filtres qui sont convolués avec les entrées, la seule différence réside dans la forme des filtres, qui

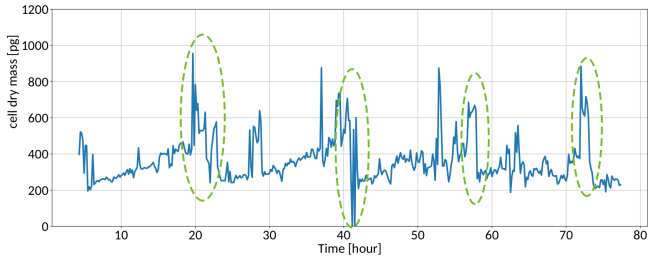


FIGURE 1 : Exemple de série temporelle de masse sèche de cellule. La masse croit régulièrement jusqu'à une division (entourée en vert) où elle chute abruptement. Le signal idéal étant en dent de scie, on constate que les signaux peuvent être très bruités.

TABLE 1 : Tableau récapitulatif de l'espace de recherche des hyperparamètres. Un hyper-paramètre est optimisé avec une valeur fixée des autres hyper-paramètres

| Param | Valeurs possibles |
|---|---|
| | Réseaux denses |
| $\mathcal{D}$ | $32/32/32 - 32/64/128 - 64/64/64/64/64$<br>$64/64/64 - 256/128/64 - 512/256/128/64/32$<br>$128/128/128 - 32/32/32/32/32$ |
| | Réseaux convolutifs à maxpooling |
| $\mathcal{C}$ | $3 - 5 - 6 - 9 - 12 - 16$ |
| $\mathcal{K}$ | $3 - 5 - 8 - 12 - 16$ |
| $\mathcal{F}$ | $64 - 32 - 128 - 256$ |
| $\mathcal{MP}$ | $3 - 2 - 1$ |
| $\mathcal{D}$ | $64-16-32-128-256-16/16-32/32-64/64-$<br>$128/128-64/32-128/64-256/128-64/32/16-$<br>$128/64/32$ |
| | Réseaux convolutifs à dilatation |
| $\mathcal{C}$ | $3 - 5 - 9 - 12$ |
| $\mathcal{K}$ | $3 - 5 - 8 - 12 - 16$ |
| $\mathcal{F}$ | $64 - 128 - 256$ |
| $\mathcal{DR}$ | $2^i - 1 - 4 - 8 - 16$ |
| | LSTM |
| $\mathcal{L}$ | $1 - 2 - 3$ |
| $\mathcal{N}$ | $64 - 32 - 128 - 256 - 512$ |
| | Réseaux à connexion résiduelles |
| $\mathcal{B}$ | $1 - 2 - 3 - 5$ |
| $\mathcal{DR}$ | $2i - 2 - 4 - 8 - 16$ |

ne sont qu'à **une** dimension. Afin de conserver des architectures CNN-1D simples, aucune couche de normalisation par batch n'est utilisée.

**Paramètres** Les paramètres à étudier sont donc : le nombre de couches convolutives $\mathcal{C}$, la taille des noyaux de convolutions $\mathcal{K}$, le nombre de filtres dans chacune des couches convolutives $\mathcal{F}$, le nombre de couches convolutives avant une couche de maxpooling $\mathcal{MP}$ et le réseau dense utilisé pour la reconstruction du signal $\mathcal{D}$. Des fonctions d'activation **tangentes hyperboliques** sont utilisées entre chacune des couches. Ce choix permet d'éviter le problème d'explosion du gradient que nous avons observé lors d'une étude préliminaire.

### CNN à dilatation

Par ailleurs, une manière alternative de construire des réseaux convolutifs consiste à remplacer les couches de maxpooling par des convolutions à dilatation ou *à trous* [9].

**Paramètres** Le pas de dilatation (*dilation rate*) des convolutions est défini par le paramètre $\mathcal{DR}$. Les convolutions de la section précédente correspondent alors à $\mathcal{DR} = 1$. Ce pas de dilatation permet, de la même manière que les couches de maxpooling, d'augmenter le champ réceptif du réseau de neurones sans pour autant diminuer la taille de l'espace latent. La valeur $\mathcal{DR} = 2^i$ correspond à un taux de dilatation à la $i^{eme}$ couche convolutive égale à $2^i$ comme proposé dans [9].

### 3.3 Long Short Term Memory (LSTM)

Les réseaux de neurones récurrents sont des réseaux de neurones particuliers présentant au moins un cycle dans leur structure. Ils sont particulièrement utilisés pour l'analyse de séries temporelles, mais sont relativement difficiles à entraîner. Les réseaux Long-Short Term Memory (LSTM) [6] sont une sous-classe des réseaux récurrents tentant de pallier le problème d'évanescence du gradient des réseaux récurrents classiques.

Les LSTM sont caractérisés à la fois par leur **nombre de couches** $\mathcal{L}$ et leur **nombre de nœuds** par couches $\mathcal{N}$.

### 3.4 Réseaux à connexions résiduelles

L'architecture à résidus [5] ajoute des connexions directes entre des couches non consécutives, des *skipped connections*. Ces dernières permettent de shunter certaines couches neuronales afin d'éviter la disparition du gradient.

Dans cette étude, les blocs résiduels sont constitués de deux paires de couches de CNN-1D associées à une couche de *BatchNormalization*. La connexion résiduelle somme l'entrée de la première couche CNN-1D à la sortie de la $2^{ieme}$ couche de *BatchNormalization*.

L'architecture résiduelle reprenant celle de la section 3.2, les paramètres $\mathcal{K}$, et $\mathcal{F}$ seront fixés aux meilleures valeurs obtenues lors de l'entraînement des CNN à dilatation pour limiter le nombre d'entraînements à effectuer. Les hyperparamètres restant testés sont le **nombre de blocs** résiduels $\mathcal{B}$ et le **taux de dilatation** $\mathcal{DR}$.

### 3.5 Paramètres expérimentaux

Pour chacune des architectures présentées, les réseaux de neurones sont entraînés 10 fois. Les différents hyperparamètres sont optimisés successivement en sélectionnant la valeur de l'hyperparamètre permettant d'obtenir la meilleure valeur d'erreur quadratique moyenne entre la prédiction et le signal à prédire. L'optimisation est réalisée grâce à l'algorithme `ADAM`, avec un *learning rate* fixé à $0{,}001$ et un *early stopping* permettant de prévenir le sur-apprentissage.

Les entraînements des réseaux sont réalisés sur une carte graphique `nvidia A100` grâce à la librairie `tensorflow`. Cette architecture matérielle est à prendre en compte dans l'analyse des temps d'entraînement et d'inférence.

## 4 Résultats et discussions

Les différentes architectures sont comparées selon 4 différentes métriques, rangées par ordre d'importance pour notre application :

1. L'Erreur quadratique moyenne (EQM) **minimale** et **moyenne sur 10 entrainements**
2. Le nombre de paramètres
3. Le temps d'inférence moyen
4. Le temps d'entrainement moyen

Le tableau 2 présente uniquement les meilleures architectures dans chacune des catégories mentionnées ci-dessus.

Tout d'abord, nous constatons que, bien qu'il soit possible de différencier les familles de réseaux sur leur valeur d'EQM, ces dernières sont comprises entre $0{,}454$ pour le meilleur réseau, un réseau dense, contre $0{,}471$ pour la moins bonne famille, les réseaux à connexions résiduelles. La figure 2 montre deux exemples de prédictions très satisfaisantes d'un point de vue applicatif. Ces prédictions sont réalisées par le meilleur réseau dense. Le réseau de neurone est capable de prédire la tendance globale du signal. Outre la valeur absolue d'EQM, cet écart de seulement $3{,}7\%$ souligne des performances similaires pour toutes les architectures. On constate donc que les CNN, LSTM et réseaux à résidus ont étés moins performants dans l'apprentissage d'une représentation capable de capturer les variations du signal utiles pour prédire son futur, contrairement aux extracteurs de *features* dense.

Il est plus facile de comparer les réseaux sur leur nombre de paramètres. Les CNN à dilatation ne diminuant jamais la taille de l'entrée, ils ont un nombre de paramètres dépassant le million pour un gain de performances prédictives peu significatif comparé aux autres architectures. En comparaison, les réseaux denses, peuvent obtenir des performances seulement $2\%$ plus faibles pour un nombre de paramètres 127 fois plus faible. Les réseaux permettant d'obtenir le meilleur compromis nombre de paramètres/EQM sont les réseaux denses pouvant atteindre des EQM de l'ordre de $0{,}454$ avec seulement $56\,000$ à $76\,000$ paramètres alors qu'aucune autre famille de réseaux n'est capable de dépasser la barrière de $0{,}46$ d'EQM.

Les LSTM et les réseaux denses peuvent atteindre des temps d'entrainements similaires d'à peine plus de 2 minutes. Par opposition, les CNN, avec ou sans *skip connections*, peuvent atteindre des temps d'entrainement 2 à 3 fois plus long pour des performances moindres. Il faut cependant remettre en perspective les temps d'entrainements observés dans cette étude, aucun n'ayant duré plus de dix minutes, face à des temps d'entrainements classiques dans le domaine de l'intelligence artificielle compris entre plusieurs heures et plusieurs semaines.

Concernant les temps d'exécution, les réseaux denses sont largement plus performants que les autres architectures avec $0{,}83$ s de moins que les LSTM en seconde position. Cet écart

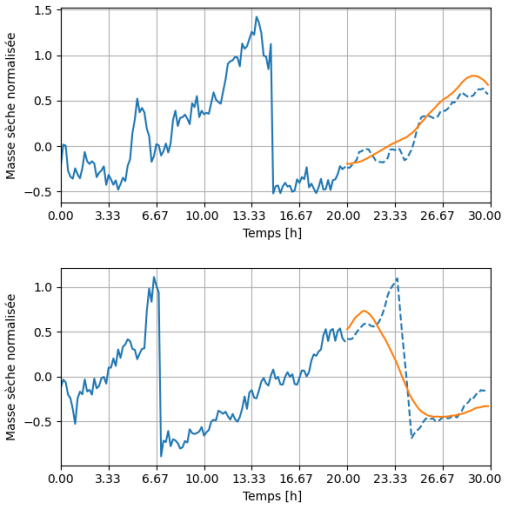


FIGURE 2 : Deux exemples de prédictions du futur d'une série temporelle de masse sèche effectuée par le meilleur réseau dense. La courbe bleue correspond à l'entrée du réseau, celle en bleue pointillée la masse future à prédire et en orange la prédiction. La prédicion est réalisée par le meilleur réseau dense (en **gras souligné** tableau 2) L'EQM atteinte sur ces prédiction est respectivement de $0{,}015$ et $0{,}085$.

TABLE 2 : Tableau récapitulatif des métriques obtenues pour les meilleurs réseaux de neurones dans chaque famille et pour chaque métrique. Les valeurs soulignées sont les meilleures intra famille tandis que celles en **gras** sont les meilleures inter familles

| Paramètres | | | | | EQM min | EQM moyenne | # param | Temps d'entrainement | Temps d'inférence |
|---|---|---|---|---|---|---|---|---|---|
| $\mathcal{D}$ | | | | | Perceptrons Multicouches | | | | |
| 256/128/64 | | | | | **0,454** | **0,458 ± 0,001** | 76 028 | 182 ± 13 s | 5.94 ± 0,25 s |
| 32/32/32 | | | | | 0,474 | 0,478 ± 0,002 | **7 964** | 195 ± 33 s | 4.70 ± 0,04s |
| 32/64/128 | | | | | 0,477 | 0,482 ± 0,004 | 22 044 | **131 ± 26** s | 6.70 ± 0,25 s |
| 128/128/128 | | | | | **0,454** | 0,460 ± 0,002 | 56 252 | 161 ± 23 s | **4.67 ± 0,07** s |
| $\mathcal{C}$ | $\mathcal{K}$ | $\mathcal{F}$ | $\mathcal{MP}$ | $\mathcal{D}$ | CNN-1D à Maxpooling | | | | |
| 9 | 3 | 64 | 3 | 128 | 0,460 | 0,470 ± 0,002 | 229 820 | 327 ± 26 s | 8.46 ± 0,17 s |
| 9 | 3 | 32 | 3 | 64 | 0,471 | 0,478 ± 0,004 | 59 644 | 291 ± 56 s | 6.75 ± 0,11 s |
| 5 | 3 | 64 | 3 | 64 | 0,468 | 0,479 ± 0,002 | 299 388 | 172 ± 7 s | 5.91 ± 0,05 s |
| 3 | 3 | 64 | 3 | 64 | 0,467 | 0,482 ± 0,004 | 274 684 | 180 ± 26 s | 5.58 ± 0,04 s |
| $\mathcal{C}$ | $\mathcal{K}$ | $\mathcal{F}$ | $\mathcal{DR}$ | | CNN-1D à taux de dilatation | | | | |
| 5 | 3 | 64 | $2^i$ | | 0,461 | 0,471 ± 0,003 | 1 040 572 | 239 ± 13 s | 6.50 ± 0,12 s |
| 3 | 3 | 64 | $2^i$ | | 0,466 | 0,476 ± 0,003 | 1 015 868 | 192 ± 36 s | 5.85 ± 0,15 s |
| $\mathcal{L}$ | | $\mathcal{N}$ | | | LSTM | | | | |
| 1 | | 32 | | | 0,465 | 0,468 ± 0,001 | 16 316 | 191 ± 14 s | 5.61 ± 0,12 s |
| 1 | | 256 | | | 0,478 | 0,483 ± 0,002 | 304 828 | 132 ± 6 s | 5.50 ± 0,09 s |
| $\mathcal{B}$ | | $\mathcal{DR}$ | | | Réseaux à connexions résiduelles | | | | |
| 5 | | 8 | | | 0,471 | 0,481 ± 0,003 | 130 044 | 510 ± 28 s | 9.40 ± 0,12 s |
| 1 | | $2^i$ | | | 0,543 | 0,561 ± 0,005 | 29 180 | 238 ± 25 s | 6.01 ± 0,27 s |

représente un temps d'inférence par échantillon de $88\,\mu s$ et $103\,\mu s$. De plus, on constate que le temps d'inférence n'est pas totalement proportionnel au nombre de paramètres, les CNN résiduels à 29 180 paramètres étant plus longs de 29% en inférence que les réseaux denses à 56 252 paramètres. L'opération de convolution est plus longue que la *simple* multiplication opérée par une couche dense.

# 5 Conclusion

Cette étude compare différentes familles d'architectures de réseaux de neurones pour la tâche de prédiction de séries temporelles. Elle porte en particulier sur un jeu de données applicatif de masse sèche de cellules.

Quatre types d'architectures - Perceptrons multicouches, CNN-1D, LSTM et réseaux à connexions résiduelles - sont comparées selon leurs capacités prédictives, leur nombre de paramètres, leur temps d'entraînement et leur temps d'inférence. Pour toutes ces métriques, les perceptrons multicouches sons ceux ayant permis d'extraire la meilleure représentation pour pouvoir ensuite prédire le futur du signal au mieux.

Une des perspectives consiste à étudier la transférabilité des représentations apprise à d'autres tâches relative à ces séries temporelles ou à d'autres séries temporelles, qu'elles soient par exemple de lignées cellulaires différentes ou qu'elles concernent un tout autre phénomène.

# Références


[1] C. ALLIER, L. HERVÉ, O. MANDULA, P. BLANDIN, Y. USSON, J. SAVATIER, S. MONNERET et S. MORALES : Quantitative phase imaging of adherent mammalian cells : A comparative study. *Biomedical Optics Express*, 10(6) :2768–2783, juin 2019.

[2] C. ALLIER, S. MOREL, R. VINCENT, L. GHENIM, F. NAVARRO, M. MENNETEAU, T. BORDY, L. HERVÉ, O. CIONI, X. GIDROL, Y. USSON et J.-M. DINTEN : Imaging of dense cell cultures by multiwavelength lens-free video microscopy. *Cytometry Part A*, 91(5) :433–442, 2017.

[3] R. BAILLY, M. MALFANTE, C. ALLIER, L. GHENIM et J. MARS : Deep anomaly detection using self-supervised learning : Application to time series of cellular data. *In ASPAI 2021 - 3rd International Conference on Advances in Signal Processing and Artificial Intelligence*, novembre 2021.

[4] Z. CUI, W. CHEN et Y. CHEN : Multi-Scale Convolutional Neural Networks for Time Series Classification. *arXiv:1603.06995 [cs]*, mai 2016.

[5] K. HE, X. ZHANG, S. REN et J. SUN : Deep residual learning for image recognition. *In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, juin 2016.

[6] S. HOCHREITER et J. SCHMIDHUBER : Long Short-Term Memory. *Neural Computation*, 9(8) :1735–1780, novembre 1997.

[7] A. KRIZHEVSKY, I. SUTSKEVER et G. E HINTON : ImageNet Classification with Deep Convolutional Neural Networks. *In* F. PEREIRA, C. J. C. BURGES, L. BOTTOU et K. Q. WEINBERGER, éditeurs : *Advances in Neural Information Processing Systems 25*, pages 1097–1105. Curran Associates, Inc., 2012.

[8] Y. LECUN, B. BOSER, J. S. DENKER, D. HENDERSON, R. E. HOWARD, W. HUBBARD et L. D. JACKEL : Backpropagation Applied to Handwritten Zip Code Recognition. *Neural Computation*, 1(4) :541–551, décembre 1989.

[9] A. VAN DEN OORD, S. DIELEMAN, H. ZEN, K. SIMONYAN, O. VINYALS, A. GRAVES, N. KALCHBRENNER, A. W. SENIOR et K. KAVUKCUOGLU : WaveNet : A generative model for raw audio. *CoRR*, abs/1609.03499, 2016.