nature methods

CritiqueARticle

https://doi.org/10.1038/s41592-019-0403-1

Apprentissage profond pour l'analyse d'images cellulaires

Erick Moen[1,3], Dylan Bannon[1,3], Takamasa Kudo[2], WilliamGraf[1], Markus Covert[2] et David Van Valen[1*]

Les progrès récents de la vision par ordinateur et de l'apprentissage automatique sous-tendent une collection d'algorithmes dotés d'une capacité impressionnante à déchiffrer le contenu des images. Ces algorithmes d'apprentissage en profondeur sont appliqués aux images biologiques et transforment l'analyse et l'interprétation des données d'imagerie. Ces avancées sont positionnées pour rendre les analyses difficiles de routine et pour permettre aux chercheurs de mener de nouvelles expériences auparavant impossibles. Nous passons ici en revue l'intersection entre l'apprentissage en profondeur et l'analyse d'images cellulaires et fournissons un aperçu à la fois de la mécanique mathématique et des cadres de programmation de l'apprentissage en profondeur qui sont pertinents pour les scientifiques de la vie. Nous étudions les progrès du domaine dans quatre applications clés: la classification d'images, la segmentation d'images, le suivi d'objets et la microscopie augmentée. Durer,

UNE permettre aux chercheurs d'accéder aux variations temporelles et spatiales inhérentes aux systèmes vivants. Les progrès de l'optique et de la biologie, microscopes capables d'imagerie sur une gamme d'échelles spatiales, des molécules uniques aux organismes entiers. Parallèlement, les améliorations des sondes fluorescentes ont amélioré la luminosité, la photostabilité et la gamme spectrale des protéines fluorescentes et des colorants à petites molécules. Combinées, ces avancées permettent une variété de mesures dynamiques dans les cellules vivantes, à partir de l'imagerie à long terme de molécules uniques.[1,2], aux mesures simultanées de plusieurs biocapteurs[3,4], aux observations du développement d'organismes entiers[5–9]. Ils ont également conduit à des mesures impressionnantes dans des échantillons fixes, la génomique spatiale pilotant désormais la mesure simultanée de dizaines de protéines ou de milliers d'espèces d'ARNm dans des cellules et des tissus fixes tout en préservant les informations spatiales.[dix-12].

Parallèlement à ces progrès technologiques, il y a une demande croissante dans les biosciences pour l'analyse d'images. Les données d'imagerie modernes nécessitent de plus en plus une quantification pour être informatives[13]. Les tâches typiques comprennent l'exploration d'images non supervisée (comparaison des caractéristiques de collections d'images, par exemple, en identifiant les changements de morphologie cellulaire dans un écran de médicament basé sur l'imagerie), la classification d'images (prédiction d'un label pour une image, par exemple, déterminer si un cellule souche s'est différenciée), la segmentation de l'image (identifier les parties d'une image qui correspondent à des objets distincts - par exemple, identifier des cellules individuelles dans des images) et le suivi d'objet (à la suite d'un objet - par exemple, une seule cellule dans une image en direct). embryon - parmi les images d'un film). En réponse à cette demande, les chercheurs et les entreprises ont développé des bibliothèques de logiciels, des implémentations basées sur des cas d'utilisation et des écosystèmes de vision par ordinateur à usage général. MATLAB a été l'une des premières plates-formes commerciales à prendre en charge des solutions de vision par ordinateur et continue de bénéficier d'une utilisation fréquente. Récemment, le développement de bibliothèques open-source de data-science pour Python (par exemple, NumPy[14], SciPy[15], Pandas[16], Scikit-image[17], Scikit-learn[18], Matplotlib[19], et Jupyter[20]) a conduit à une augmentation de la popularité de Python. MATLAB et Python contiennent maintenant des implémentations toutes faites d'algorithmes de vision par ordinateur courants. Traditionnellement, les expérimentateurs écrivaient des outils logiciels qui s'inspiraient de ces bibliothèques. Au fur et à mesure que les tâches d'analyse devenaient plus courantes, plusieurs outils logiciels ont été créés pour améliorer l'accessibilité grâce à une interface graphique. Par exemple, il existe des outils d'analyse monocellulaire de bactéries (SuperSegger[21], Oufti[22], Morphométrie[23]), analyse monocellulaire de cellules de mammifères (CellProfiler[24,25], Ilastik[26], Navigateur d'images de microscopie[27]), et analyse d'images à usage général (ImageJ[28], OMERO[29]). Ces outils et écosystèmes ont transformé la conception expérimentale, rendu les analyses quantitatives et statistiques automatisables et à haut débit, et produit une pléthore de connaissances biologiques critiques.

Fait intéressant, l'apprentissage en profondeur a élargi l'éventail des problèmes que la vision par ordinateur peut résoudre[30]. Ici, le «deep learning» fait référence à un ensemble de techniques d'apprentissage automatique, en particulier les réseaux de neurones qui apprennent des représentations efficaces de données avec plusieurs niveaux d'abstraction.[30]. Notez le contraste avec l'apprentissage machine conventionnel, dans lequel les représentations sont conçues manuellement grâce à l'ingénierie des fonctionnalités. En deep learning, l'apprentissage peut être supervisé ou non. Les approches supervisées, qui ont été les plus réussies, tentent de maximiser les performances sur un ensemble de données annotées. Des approches non supervisées sont utilisées pour reconstruire les données originales après compression dans un espace de faible dimension. Bien que ces techniques existent sous forme mathématique depuis plusieurs décennies, elles ont attiré l'attention lorsqu'une méthode basée sur l'apprentissage en profondeur a remporté le Défi de reconnaissance visuelle à grande échelle ImageNet 2012.[31]. Depuis lors, il y a eu une augmentation importante de la variété des problèmes qui peuvent être résolus avec l'apprentissage en profondeur. En outre, les améliorations apportées au matériel informatique et aux cadres d'apprentissage en profondeur ont placé ces outils à la portée du développeur de logiciel typique. Alors que l'apprentissage profond a été principalement appliqué commercialement, il commence maintenant à émerger dans le domaine physique.[32-34], chimique[35,36], médical[37,38], et sciences biologiques[39-42] avec des applications pour les images et d'autres types de données.

Étant donné le rôle central que l'observation - et donc l'imagination - joue dans les sciences biologiques, l'apprentissage profond a le potentiel de révolutionner la compréhension du fonctionnement interne des systèmes vivants. En effet, une «ruée vers l'or» est actuellement en cours, de nombreux groupes cherchant à appliquer ces méthodes à leurs données afin d'en extraire de nouvelles connaissances biologiques. Néanmoins, l'apprentissage en profondeur n'a pas encore été largement adopté dans les sciences de la vie et les sciences médicales. Surtout, de nombreux outils logiciels mentionnés ci-dessus (à l'exception récente de CellProfiler[25] et ImageJ[43]) ne présente pas encore

[1]Division de biologie et de bioingénierie, California Institute of Technology, Pasadena, Californie, États-Unis. [2]Département de bio-ingénierie, Université de Stanford, Stanford, Californie, États-Unis. [3]Ces auteurs ont contribué à parts égales: Erick Moen, Dylan Bannon. *e-mail:vanvalen@caltech.edu

l'apprentissage en profondeur. À notre avis, pour que l'apprentissage profond transforme véritablement les sciences de la vie, son application doit être aussi routinière que les recherches BLAST. Les obstacles à la diffusion de l'apprentissage profond dans les laboratoires de biologie sont à la fois culturels et techniques. Les mathématiques rendent opaques certains des rouages internes des algorithmes d'apprentissage profond; les exigences uniques de l'apprentissage profond nécessitent une manière différente de penser l'écriture de logiciels. Plus précisément, le besoin de données annotées signifie que les données et les logiciels doivent être développés conjointement - une approche récemment appelée Logiciel 2.0[44] (Figure. 1). La quantité de données et les ressources de calcul requises pour l'apprentissage profond constituent un obstacle important à l'adoption, tout comme les connaissances requises pour optimiser les performances des modèles et interpréter ce que les modèles d'apprentissage profond ont appris. Pour exploiter toute la puissance de ces outils, les scientifiques de la vie doivent se familiariser avec eux pour améliorer leurs flux de travail existants et préparer le terrain pour des analyses actuellement imprévues.

En se concentrant sur des cas d'utilisation courants en biologie cellulaire quantitative, cette revue sert d'introduction pratique à un apprentissage approfondi pour l'analyse d'images biologiques. Il s'appuie sur des examens antérieurs de l'intersection de l'apprentissage profond et des sciences de la vie[42,45-47] en incorporant une discussion sur les expériences conjointes de nos laboratoires dans l'application de ces méthodes aux données d'imagerie cellulaire, et vise à rendre ces méthodes moins opaques pour les nouveaux utilisateurs. Tout d'abord, nous passons en revue les mécanismes pratiques de l'apprentissage profond, y compris les fondements mathématiques, les progrès récents dans les architectures de réseaux neuronaux et les cadres logiciels existants. Ensuite, nous décrivons ce que nous pensons être les composants clés de solutions d'apprentissage en profondeur efficaces à l'échelle du laboratoire. Nous passons ensuite en revue quatre cas d'utilisation: la classification des images, la segmentation des images, le suivi des objets et la microscopie augmentée. Pour chaque cas d'utilisation, nous couvrons la spécification du problème, l'état du champ en ce qui concerne les algorithmes et les applications biologiques, et les ensembles de données accessibles au public.

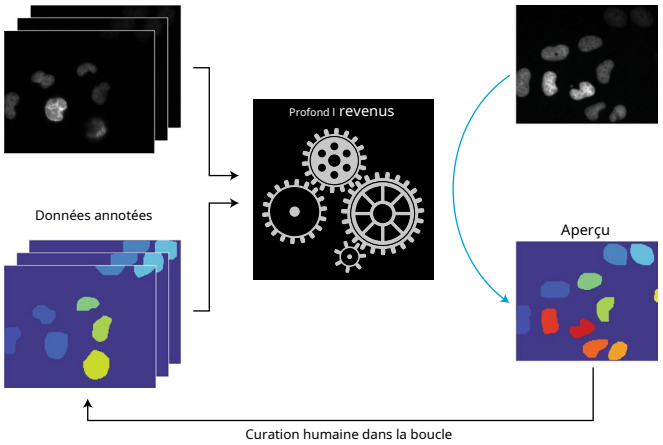


Fig. 1 | Le logiciel 2.0 combine les annotations de données avec l'apprentissage en profondeur pour produire des logiciels intelligents. Les annotations produites par des annotateurs experts ou par une foule peuvent être utilisées pour former des modèles d'apprentissage en profondeur afin d'extraire des informations à partir de données. Une fois formés, ces modèles peuvent être déployés pour traiter de nouvelles données non annotées. L'extension humaine dans la boucle implique l'identification des erreurs de modèle, la correction des erreurs pour produire de nouvelles données d'entraînement et le recyclage sur un ensemble de données mis à jour.

La mécanique pratique de l'apprentissage profond

Dans l'apprentissage en profondeur, un algorithme apprend des représentations efficaces pour une tâche donnée entièrement à partir de données. Une introduction aux mathématiques sous-tendant la formation des modèles d'apprentissage profond est donnée dans l'encadré1, des conseils de dépannage sont donnés dans Box 2, et un glossaire des termes couramment utilisés est donné dans l'encadré 3. Parce que les solutions les plus réussies ont été supervisées, nous pensons qu'il y a trois composantes essentielles à l'application réussie de l'apprentissage profond à l'analyse d'images biologiques: construction d'un jeu de données de formation pertinent et annoté, formation efficace de modèles d'apprentissage profond sur ce sujet. ensemble de données et déploiement de modèles entraînés sur de nouvelles données.

Les données de formation sont essentielles à la réussite des applications de l'apprentissage en profondeur; cette exigence est l'un des principaux inconvénients de cette méthode. D'après notre expérience, assembler suffisamment de données de haute qualité prend souvent autant de temps, sinon plus, que la programmation de la solution d'apprentissage en profondeur. Les solutions robustes nécessitent des ensembles de données qui capturent la diversité des images susceptibles d'être rencontrées lors de l'analyse. Dans la mesure du possible, les annotations pour ces ensembles de données doivent être exemptes d'erreurs, car les erreurs peuvent être apprises. Alors que les données de formation peuvent être limitées, les approches informatiques peuvent extraire le plus d'utilité des données existantes. La normalisation de l'image réduit la variation par rapport aux conditions d'acquisition distinctes[42,48]. Les opérations d'augmentation des données telles que la rotation, le retournement et le zoom peuvent également augmenter la diversité des images dans un ensemble de données limité; ces opérations sont généralement des pratiques standard quelle que soit la taille ou le type de l'ensemble de données[49]. L'apprentissage par transfert est une autre approche pour créer des modèles robustes avec des données limitées. Dans l'apprentissage par transfert, un modèle d'apprentissage en profondeur est formé sur un grand ensemble de données pour apprendre les caractéristiques générales de l'image, puis est affiné sur un ensemble de données plus petit pour apprendre à effectuer une tâche spécifique.[50,51]. Bien que ces approches permettent à des réseaux performants d'émerger à partir d'ensembles de données limités, des améliorations considérables des performances proviennent de grands ensembles de données annotés.[52]. Pour certains utilisations, telles que la détection de taches limitées par la diffraction, il a été possible de produire des images simulées avec une annotation connue[53]. Dans d'autres stratégies, les sorties organisées des pipelines de vision par ordinateur traditionnels ont été utilisées comme données de formation.[54]. Les données de formation ont également été produites manuellement par des experts à l'aide d'outils d'annotation tels que Fidji / ImageJ[48], Cellprofiler[52], et le segment de structure de cellule Allen[55]. Le crowdsourcing, une source rentable de grands ensembles de données, est largement utilisé dans des domaines tels que la conduite automatisée; les outils existants sont en cours d'adaptation pour les images biologiques. Les solutions commerciales d'entreprise incluent Figure Eight, récemment acquise par Appen, et Samasource. Le Quanti.us[56] L'outil dispose d'une interface utilisateur graphique pour l'annotation d'images biologiques à utiliser sur Amazon Mechanical Turk, tout comme l'outil Ground Truth d'Amazon, qui utilise l'apprentissage actif pour réduire les coûts d'étiquetage des données. La gamification a également donné des résultats très prometteurs[57]. Surtout, la communauté reconnaît que les ensembles de données annotés qui alimentent les algorithmes d'apprentissage en profondeur devraient être accessibles au public, car un ensemble complet et étendu de données de formation spécifiques aux problèmes biologiques aiderait considérablement le développement d'algorithmes d'apprentissage en profondeur.

Une fois les données d'entraînement acquises, un modèle d'apprentissage en profondeur peut être formé pour faire des prédictions précises pour les nouvelles données. Cette tâche a plusieurs exigences logicielles et matérielles uniques. Actuellement, Python est le langage le plus populaire pour l'apprentissage en profondeur; les frameworks existants incluent Tensorflow / Keras[58,59], PyTorch[60], MXNet[61], CNTK[62], Theano[63], et Caffe[64]. Bien que ces cadres présentent des différences importantes, il existe également plusieurs points communs. Tout d'abord, tous construisent un graphe de calcul qui décrit tous les calculs effectués par un modèle d'apprentissage en profondeur lorsque les données d'entrée sont transformées en sortie finale. Deuxièmement, ils effectuent tous automatiquement des dérivées, ce qui leur permet d'effectuer des optimisations comme celles décrites dans Box1 sans travail supplémentaire de l'utilisateur une fois le graphe de calcul spécifié. Troisièmement, ils fournissent une passerelle facile pour le matériel spécialisé tel que les unités de traitement graphique (GPU) et les unités de traitement tensoriel.[65-67]. Étant donné que les modèles d'apprentissage en profondeur contiennent souvent des millions de paramètres, un matériel spécialisé est nécessaire pour effectuer ces calculs rapidement. Quatrièmement, ces frameworks contiennent tous des implémentations d'objets mathématiques communs, d'algorithmes d'optimisation, de paramètres d'hyperparamètres et de mesures de performance, ce qui signifie que les utilisateurs peuvent rapidement appliquer un apprentissage en profondeur à leurs données sans avoir à reproduire eux-mêmes ces implémentations. Bien qu'une quantité considérable de

Encadré 1 | Formation d'un classificateur d'image linéaire

Pour illustrer le flux de travail pour entraîner un modèle d'apprentissage en profondeur de manière supervisée, nous considérons ici le cas de la formation d'un classificateur linéaire pour reconnaître des images en niveaux de gris de chats et de chiens. Chaque im-
l'âge est un tableau de taille (N_x, N_y, 1), où N_x et N_y sont le nombre de pixels dans le X et y dimensions, respectivement, et 1 est le nombre de canaux dans l'image. Pour cet exercice, nous réduisons le image dans un vecteur de taille ($N_x N_y$, 1). La tâche de classification consiste à construire une fonction qui prend ce vecteur comme entrée et prédit un étiquette (0 pour les chats, 1 pour les chiens). Un classificateur linéaire effectue cette tâche en produisant des scores de classe qui sont une fonction linéaire de chaque pixel valeur. Mathématiquement, cela s'écrit

$$\begin{matrix} y_0 \\ y_1 \end{matrix} = \begin{matrix} w_{0,0} \cdots & w_{0,NxNy-1} \\ w_{1,0} \cdots & w_{1,NxNy-1} \end{matrix} X \begin{matrix} x_0 \\ \vdots \\ x_{NxNy-1} \end{matrix} = \begin{matrix} \sum w_{0,j}x_j \\ \sum w_{1,j}x_j \end{matrix}$$

où y_0 et y_1 sont les scores de la classe, W est une matrice de poids de classe, et X est le vecteur d'image. La classe avec le score le plus élevé est la classe prédite. La tâche d'apprentissage règle ensuite le $w_{je,j}$ valeurs de sorte qu'une fonction de perte qui mesure les performances du classifieur sur certains L'ensemble de données d'entraînement est minimisé. Une fonction de perte courante est la perte d'entropie croisée ou softmax. Pour arriver à cette fonction de perte, nous commençons par transformer nos scores de classe y_0 et y_1 en probabilités en définissant

$$p_{je} = \frac{e^{\text{Classer } je \text{ But}}}{\sum_{\text{Toute classe es}} e^{\text{Score de classe}}}$$

Ces probabilités reflètent la certitude du modèle qu'une image appartient à la classe je. La perte évaluée pour une collection d'images est définie comme

$$\text{Perte} = -\sum_{\text{Images}} \text{Journal } p_{\text{Corriger}} + \lambda \sum_{je,j} w_{je,j}^2$$

où p_{Corriger} est la probabilité attribuée à la classe correcte pour cette image. Cette équation comporte deux termes. Le premier peut être pensé comme la probabilité logarithmique négative de choisir la classe correcte. Le deuxième terme est appelé régularisation L2; il pénalise les poids importants pour lutter contre le surajustement.

Pour minimiser la fonction de perte, la plupart des algorithmes d'optimisation utilisés dans l'apprentissage en profondeur sont une variante de la descente de gradient stochastique. Premièrement, les poids sont initialisés au hasard à une petite valeur. Le choix de l'initialisation peut affecter considérablement l'apprentissage des modèles profonds; les meilleures pratiques d'initialisation incluent les paramètres d'initialisation de He et al.[167]. Ensuite, nous sélectionnons un petit lot d'images, appelé un minibatch, et identifions la direction dans laquelle changer les poids afin que la fonction de perte soit réduite au maximum lorsqu'elle est évaluée sur ce minibatch. Nous perturbons alors un peu les poids dans cette direction. La direction correcte dans laquelle perturber les poids est capturée par le gradient de la fonction de perte par rapport aux poids. Mathématiquement, ce dégradé conduit à la règle de mise à jour

$$w_{je,j} \rightarrow w_{je,j} - \eta \frac{\partial \text{perte}}{\partial w_{je,j}}$$

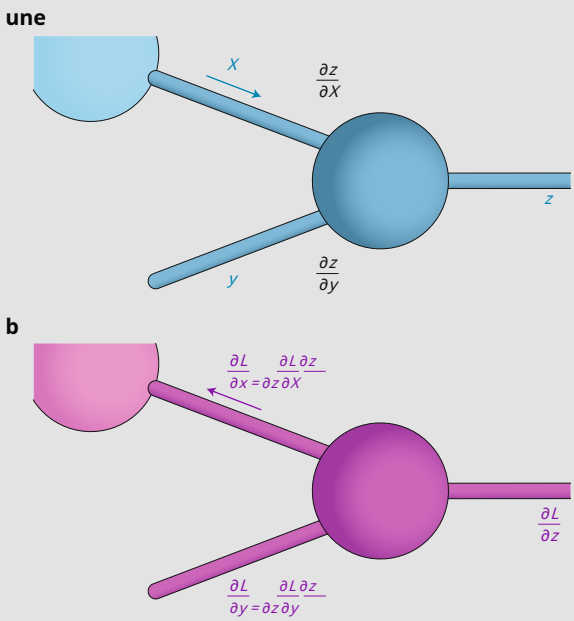
où lr, le scalaire qui met à l'échelle le gradient à chaque étape, est le taux d'apprentissage. Pour le cas du classifieur linéaire, nous pouvons calculer ces gradients de manière analytique. Le dégradé d'une image est donné par

$$\frac{\partial \text{perte}}{\partial w_{je,j}} = x_j (p_{je} - 1(je \text{ est l'étiquette correcte})) + 2\lambda w_{je,j}$$

où 1 () est une fonction indicatrice qui vaut 1 w poule la déclaration à l'intérieur des parenthèses est vrai et 0 lorsqu'il est faux. Le dégradé pour le minibatch d'images est la somme des dégradés de toutes les images du lot. Cette équation fournit des informations sur la manière dont la descente de gradient modifie les poids. Le premier terme conduit à une augmentation des poids qui correspondent à la classe correcte et à une diminution des poids qui correspondent à la classe incorrecte. Si la le modèle est certain ($p_{je} \approx 1$) et correcte, la contribution sera minime; le contraire est vrai si le modèle est certain et faux. le le dégradé est mis à l'échelle par la valeur de pixel appropriée x_j, ce qui oblige le modèle à prêter attention aux pixels brillants. La contribution de la le terme de régularisation pousse les poids vers zéro, empêchant un seul poids de devenir trop gros. Une fois les poids mis à jour, l'utilisateur sélectionne alors un autre lot d'images et répète le processus jusqu'à ce que la perte soit suffisamment minimisée. La précision et la perte de l'algorithme sur l'ensemble de données de validation sont souvent utilisées pour développer des critères d'arrêt. La formation est souvent interrompue lorsque la perte de validation cesse de s'améliorer ou lorsque les courbes d'erreur de formation et de validation commencent à diverger, ce qui signifie un surajustement.

Si le classificateur linéaire met en évidence plusieurs caractéristiques clés de la formation, il existe en pratique des différences importantes. Des variantes de la fonction de perte illustrées ci-dessus ont été développées pour résoudre les problèmes liés au déséquilibre de classe dans les ensembles de données. Plusieurs variantes de descente de gradient stochastique existent, y compris avec momentum[168-170], RMSprop[171], Adagrad[172], Adadelta[173], et Adam[174]. Des travaux récents suggèrent que les réseaux formés avec une descente de gradient stochastique avec momentum ont de meilleures performances en termes de généralisation[175,176]. Nous avons présenté le taux d'apprentissage comme un paramètre statique, mais dans la pratique, il diminue souvent à mesure que la formation progresse.

Surtout, la structure mathématique des modèles d'apprentissage profond est plus compliquée que le modèle linéaire présenté ici. Bien que cette simplification puisse paraître problématique en ce qui concerne le calcul analytique des gradients pour la formation, tous les modèles d'apprentissage profond sont compositionnels. Cela permet d'utiliser de manière itérative la règle de chaîne[177] pour dériver des expressions analytiques pour les dégradés, même pour des fonctions compliquées, comme le montre la figure de cette boîte.



Calcul des dégradés avec rétropropagation. **une,** Au cours de l'avant pass, les dérivées locales sont calculées parallèlement au calcul original. **b,** Lors du passage en arrière, la règle de chaîne est utilisée en conjonction avec les dérivées locales pour calculer la dérivée de la fonction de perte par rapport à chaque poids.

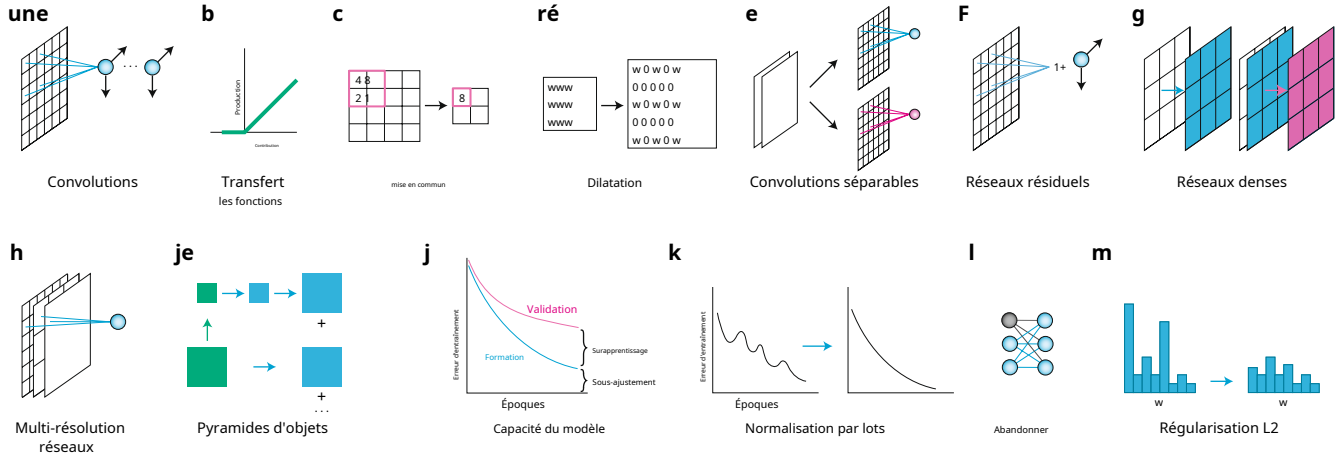


Fig. 2 | Composantes mathématiques courantes des modèles d'apprentissage en profondeur. une, Les convolutions extraient les caractéristiques locales dans les images et les poids de chaque filtre peuvent être ajustés pour extraire la meilleure caractéristique pour un ensemble de données et une tâche donnés. **b,** Des fonctions de transfert telles que celles appliquées par l'unité linéaire rectifiée commune (ReLU) permettent l'apprentissage de relations non linéaires. **c,** Regrouper des opérations telles que le sous-échantillonnage de regroupement maximal pour produire des cartes d'entités spatialement grossières[154]. Les architectures d'apprentissage en profondeur utilisent souvent des cycles itératifs des trois opérations dans **une**-**c** pour produire des représentations d'images en basse dimension. **ré,** Les dilatations permettent aux noyaux convolutifs et groupés d'augmenter leur étendue spatiale tout en gardant le nombre de paramètres fixe[66,155]. Lorsqu'elles sont utilisées correctement, les dilatations permettent aux réseaux de classification formés sur des correctifs d'image d'être utilisés pour une prédiction dense au niveau des pixels. **e**-**je,** Les modèles modernes d'apprentissage en profondeur utilisent plusieurs éléments architecturaux. **e,** Les convolutions séparables effectuent l'opération de convolution sur chaque canal séparément, ce qui réduit la puissance de calcul tout en préservant la précision[156,157]. **F,** Les réseaux résiduels apprennent la cartographie d'identité plus un petit résidu et permettent la construction de réseaux très profonds[68]. **g,** Les réseaux denses permettent à chaque couche de voir chaque couche précédente[69], qui améliore la propagation des erreurs et encourage à la fois la réutilisation des fonctionnalités et l'efficacité des paramètres[69,70]. **h,** Les réseaux multi-résolution permettent aux couches de classification de voir à la fois des cartes d'entités fines et grossières[69]. **je,** Grâce aux pyramides d'objets, les modèles de détection d'objets détectent des objets à des échelles de longueur distinctes[91,158]. **j,** Un graphique de l'erreur de formation pendant la formation révèle les relations entre le surajustement, le sous-ajustement et la capacité du modèle. Le compromis entre ces attributs détermine les architectures de réseau adaptées à une tâche donnée. «Sous-ajustement» se réfère aux modèles avec un pouvoir de représentation insuffisant, et «sur-ajustement» fait référence aux modèles qui ont appris des caractéristiques spécifiques aux données d'entraînement et qui se généralisent donc mal à de nouvelles données non vues. L'augmentation de la capacité du modèle réduit le sous-ajustement mais peut augmenter le risque de sur-ajustement.**k**-**m,** De nombreuses techniques de régularisation garantissent que les modèles d'apprentissage en profondeur apprennent les fonctionnalités générales à partir des données. **k,** La normalisation des lots régularise à la fois les réseaux et réduit le temps nécessaire à la formation[159]. Il a été initialement créé pour atténuer le changement de covariable, mais il a récemment été découvert qu'il adoucissait le paysage de la fonction de perte.[160]. **l,** Dropout désactive aléatoirement les filtres pendant l'entraînement[161], qui régularise le réseau en l'obligeant à ne pas trop s'appuyer sur une seule fonctionnalité pour faire des prédictions. La normalisation et la suppression des lots ne sont généralement pas utilisées ensemble dans le même modèle[162]. **m,** La régularisation L2 pénalise les poids importants et réduit le surajustement.

la programmation est encore nécessaire pour adapter ces cadres aux données d'imagerie cellulaire, ils réduisent considérablement la barrière à l'entrée.

Ces frameworks ont considérablement simplifié la formation et le déploiement de modèles de deep learning. Les aspects de programmation sont souvent réduits à trouver une architecture d'apprentissage en profondeur qui offre les meilleures performances pour une tâche particulière. Des stratégies récentes ont incorporé une recherche dans tout l'espace d'architectures potentielles pour identifier l'architecture modèle la plus efficace.[68-70] (Figure. 2). D'après notre expérience, le choix des caractéristiques architecturales se résume souvent à un compromis entre le sur-ajustement et le sous-ajustement; ceci est également connu sous le nom de compromis biais-variance dans la modélisation statistique[71]

(Figure. 2j). Le surajustement se produit lorsque les modèles fonctionnent bien sur un ensemble de données d'entraînement mais de mauvais résultats sur un ensemble de données de validation retenu, tandis que le sous-ajustement se produit lorsque les modèles fonctionnent mal sur les données d'entraînement parce qu'ils sont incapables de capturer la variation dans un ensemble de données d'entraînement. Ces deux résultats sont souvent (mais pas toujours) les deux faces d'une même pièce, que nous appelons la capacité de modèle: le pouvoir de représentation d'un modèle d'apprentissage en profondeur. Les modèles avec une capacité de modèle élevée fonctionnent bien sur de grands ensembles de données, mais sont sujets au surajustement. Les modèles avec une capacité de modèle inférieure peuvent mieux se généraliser, mais sont à risque de sous-ajustement. Les modèles overfit peuvent ne pas être fiables sur des données invisibles, et les modèles underfit ont des performances sous-optimales[71]. Le surajustement est un problème particulièrement important avec les petits ensembles de données. Les techniques d'atténuation du surajustement sont présentées dans l'encadré2 et Fig. 2j – m. Ces dernières années ont vu des avancées architecturales, telles que les réseaux résiduels[68],

qui ont une capacité de modèle accrue, ce qui peut entraîner un surajustement[72]. Nous recommandons d'utiliser des architectures de modèle avec une capacité de modèle élevée uniquement lorsque suffisamment de données sont présentes pour éviter la fragilité du modèle qui accompagne le surajustement. Lorsque les données sont limitées, les modèles avec une capacité limitée et une formation aux techniques de régularisation sont plus susceptibles d'être solides. Une autre approche consiste à utiliser l'apprentissage par transfert lors de l'adaptation de modèles d'apprentissage profond à de petits ensembles de données. Cette stratégie nécessite souvent que les modèles pré-entraînés soient modifiés pour être compatibles avec le nouvel ensemble de données et la nouvelle tâche (changement du nombre de canaux pour une image d'entrée, etc.); les utilisateurs sont souvent incapables d'apporter des modifications substantielles à l'architecture du modèle. Malgré ces limites, l'apprentissage par transfert peut être très efficace lorsque les données sont limitées. Le fait que les ensembles de données de formation soient suffisamment volumineux peut être évalué avec une analyse de validation croisée. Dans cette approche, on calcule le degré de surajustement sur des modèles entraînés sur différentes fractions des données d'apprentissage disponibles. Si la quantité de données est suffisante, alors le degré de surajustement doit être stable même lorsque la taille de l'ensemble de données d'apprentissage est réduite. Si le surajustement est un problème important, d'autres problèmes pratiques entrent en jeu (Encadré2), y compris l'optimisation des hyperparamètres tels que le taux d'apprentissage, le choix de l'algorithme de formation et les problèmes liés à l'équilibrage des classes.

Une fois formés, des modèles d'apprentissage en profondeur doivent être déployés pour traiter les nouvelles données. Alors que le déploiement peut être réalisé avec des scripts et des notebooks Jupyter[48], une approche alternative et sans doute plus efficace consiste à utiliser des outils de déploiement intégrés dans plusieurs cadres. Par exemple, Tensorflow et MXNet ont tous deux des fonctionnalités de déploiement intégrées qui permettent aux modèles d'être déployés sur un serveur et accessibles via des protocoles de communication Internet standard.[73], qui permet aux modèles d'être partagés au-delà de l'utilisateur d'origine. Les exigences logicielles et matérielles associées signifient que des couches supplémentaires d'ingénierie logicielle au-delà de ce qui est typique des logiciels universitaires sont souvent nécessaires pour qu'une solution de déploiement soit utile. Premièrement, les outils de conteneurisation tels que Docker[74] ont été essentiels pour la

Encadré 2 | Dépannage

Alors que l'apprentissage en profondeur peut résoudre de nombreux problèmes dans l'analyse d'images biologiques, la création de modèles performants nécessite souvent une quantité importante de dépannage. Ici, nous fournissons des conseils sur la navigation dans les problèmes courants qui surviennent lors de la formation des modèles d'apprentissage en profondeur.

Performance d'entraînement. Une très mauvaise performance pendant l'entraînement, définie comme une erreur de classification égale ou inférieure au hasard aléatoire, peut généralement être attribuée à un problème avec les données ou avec les paramètres d'entraînement. Pour les petits ensembles de données, des erreurs dans les données d'entraînement peuvent entraîner de mauvaises performances. Ces erreurs passent souvent inaperçues jusqu'à ce que les données d'entraînement soient inspectées manuellement. Une mauvaise normalisation d'image peut entraîner de mauvaises performances, car les images peuvent perdre leurs fonctionnalités informatives. Des erreurs dans le code qui effectue l'augmentation de l'image et alimente les données dans le pipeline d'entraînement peuvent également entraîner de mauvaises performances. Le taux d'apprentissage est souvent le premier paramètre à ajuster lorsque les données d'entraînement sont exemptes d'erreurs et que les performances sont encore très faibles. La modification de l'architecture du modèle pour augmenter la capacité du modèle peut également être une solution efficace.

Surajustement. Les modèles d'apprentissage en profondeur peuvent apprendre des relations complexes entre les données et les annotations. Par conséquent, on peut se demander si un modèle d'apprentissage en profondeur a appris quelque chose de général qui fonctionnera sur des données réelles ou si l'apprentissage du modèle est unique à l'ensemble de données d'apprentissage. Ce phénomène est appelé surajustement et est généralement mesuré comme la différence entre la précision du modèle pour un ensemble de données d'apprentissage et celle pour un ensemble de données de validation. La quantité de surajustement qui peut être tolérée varie selon la tâche; plusieurs points de pourcentage peuvent être tolérables pour la segmentation, mais peuvent amener un classificateur d'images à mal classer des catégories rares importantes. Plusieurs techniques de régularisation existent pour atténuer le surajustement, mais elles se font souvent au détriment de la capacité du modèle. Normalisation par lots[159] a de fortes propriétés de régularisation, tout comme le décrochage[161]. En règle générale, une seule de ces méthodes est utilisée dans un modèle, car les performances peuvent en souffrir si les deux sont utilisées simultanément[162]. Augmenter la force de la régularisation L2 atténue également le surajustement, mais au détriment de la capacité du modèle. L'augmentation de la gamme des opérations d'augmentation des données crée un ensemble de données d'entraînement plus varié et donc des modèles plus robustes[49]. Parce que le surajustement s'aggrave souvent au fur et à mesure que l'entraînement dure, arrêter l'entraînement prématurément peut également être efficace.[178]. Le choix de l'architecture du modèle est particulièrement important pour les petits jeux de données. Les architectures avec de grandes capacités de modèle peuvent être particulièrement sujettes au surajustement sur de petits ensembles de données, bien que cette tendance puisse être quelque peu atténuée avec l'apprentissage par transfert en pré-entraînement sur des ensembles de données plus volumineux.[50]. Enfin, l'algorithme d'apprentissage utilisé affecte le surajustement: des travaux récents ont démontré que les modèles entraînés avec une descente de gradient stochastique avec momentum se généralisent mieux que les modèles entraînés avec d'autres algorithmes.[175,176].

Déséquilibre de classe. Les ensembles de données de formation pour les tâches de classification ont souvent des nombres d'exemples différents pour chaque étiquette, ce qui peut conduire à des modèles peu performants. À titre d'exemple, considérons un ensemble de données dans lequel 90% des exemples sont l'étiquette A et 10% sont l'étiquette B. Le modèle d'apprentissage en profondeur peut apprendre à tout prédire comme étant de classe A, puis rapporter une précision de classification globale de 90%. Cependant, la précision rapportée est trompeuse et le modèle est trop imprécis sur la classe B pour être utilisé. Il existe plusieurs solutions au déséquilibre de classe. Rééchantillonner les données d'entraînement pour produire un nombre identique d'éléments dans chaque classe est une approche. Cette stratégie peut inclure un sous-échantillonnage pour correspondre à la plus petite taille de classe, un suréchantillonnage pour correspondre à la plus grande taille de classe, ou les deux.

la diversité des données de formation sera fortement réduite. Une autre façon de tenir compte du déséquilibre de classe est d'introduire un terme de poids de classe dans la fonction de perte. Ce terme, qui est souvent pris comme ($N_{\text{Nombre total d'exemples}}$ / $N_{\text{exemples en classe } je}$) × (1/$N_{\text{Des classes}}$) pour chaque classe *je*, multiplie la contribution individuelle de chaque exemple de données à la perte. Ce terme de poids de classe peut être calculé pour l'ensemble de données d'entraînement ou pour chaque mini-match à la volée.

Évaluer les performances. Le suivi des mesures de performance pendant et après la formation est un élément important de la création de modèles d'apprentissage en profondeur. Pour l'évaluation des performances, les données d'entraînement sont souvent divisées en deux parties, une pour la formation et une pour la validation. Si les performances de l'ensemble de données de validation sont utilisées pour modifier les paramètres d'entraînement, il est alors possible de suradapter le modèle aux données de validation, même si ces données n'ont pas été utilisées explicitement pendant l'entraînement. Par conséquent, certains chercheurs ont divisé leurs données en trois parties: une pour la formation, une pour la validation pendant la formation et une pour tester les performances du monde réel. Nos groupes ont obtenu de bons résultats en réservant 10 à 20% des données annotées pour les tests.

Une fois l'ensemble de données divisé, le problème restant est de choisir une mesure de performance. Comme le montre l'exemple de déséquilibre de classe, des mesures simples telles que la précision peuvent déformer les performances. Les métriques utiles varient selon le type de problème. Pour les tâches de classification, l'évaluation de l'exactitude de chaque classe individuelle est plus informative qu'une moyenne de toutes les classes. Une matrice de confusion[179] va plus loin, car il révèle la fréquence de chaque type d'erreur de classification. Pour les tâches de segmentation, à la fois au niveau du pixel (par exemple, les indices Dice et Jaccard[180]) et métriques au niveau de l'instance (précision[181], rappeler[181], et précision moyenne moyenne[182]) peut être utilisé pour mesurer les performances. La quantification des taux auxquels des erreurs spécifiques se produisent, telles que le faux fractionnement ou la fusion de masques d'instances, a fourni des informations importantes sur les modes d'échec des modèles d'apprentissage en profondeur.[52]. Les métriques appropriées doivent être utilisées à la fois sur les ensembles de données de formation et de validation à la fin de chaque période de formation.

Optimisation des hyperparamètres. Un hyperparamètre est un paramètre défini avant le début de l'apprentissage. Les hyperparamètres incluent la force de régularisation L2, le taux d'apprentissage, les paramètres d'initialisation des paramètres et les détails de l'architecture d'apprentissage profond (nombre de couches, types de couches, nombre de filtres dans chaque couche, etc.). L'optimisation des hyperparamètres, partie essentielle du processus de formation aux modèles de deep learning, se compose généralement de trois phases: sélection d'une condition initiale, sélection d'un objectif d'optimisation et recherche des meilleurs hyperparamètres. Bien que le choix des conditions initiales puisse être délicat, des packages comme Keras sont fournis avec les paramètres par défaut des meilleures pratiques pour les hyperparamètres tels que le taux d'apprentissage, la force de régularisation L2 et l'initialisation du poids.[59]. Dans la pratique, nos groupes utilisent souvent les précisions de formation et de validation par classe comme cibles, dans le but de maximiser les précisions de formation entre les classes tout en minimisant l'écart entre les précisions de formation et de validation afin de minimiser le surajustement. Enfin, il existe plusieurs stratégies pour rechercher l'espace des hyperparamètres. Les recherches de grille dans lesquelles le taux d'apprentissage et la force de régularisation sont réglés sont souvent efficaces. L'architecture du modèle peut également être modifiée; nos groupes utilisent souvent le compromis entre la capacité du modèle et le surajustement comme guide pour déterminer les changements à apporter. Nos travaux antérieurs ont montré qu'il est important de faire correspondre la taille du champ réceptif d'un modèle avec la taille de la caractéristique pertinente afin de produire un modèle performant pour les images biologiques.[48].

Le package Python Talos est un outil pratique pour Keras[59] utilisateurs qui aident à automatiser l'optimisation des hyperparamètres grâce à des recherches de grille[183].