Histoire des réseaux de neurones et du deep learning en traitement de signaux et des images

Nicolas THOME[1] Christian WOLF[2]

[1]Sorbonne Université, CNRS, ISIR, Paris, France

[2]Naver Labs Europe, Meylan, France

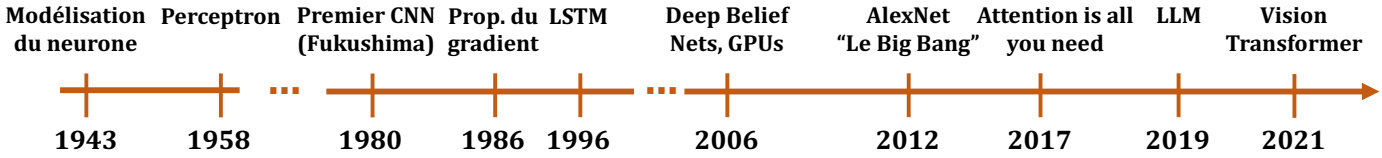


FIGURE 1 : Quelques évènements clé du développement historique des réseaux de neurones.

Résumé – Cette soumission trace un historique des réseaux de neurones informatiques pour le traitement du signal et des images, depuis leurs fondements jusqu'aux succès du deep learning moderne. Nous présentons leurs éléments clés des modèles en termes d'architecture et d'entraînement, et soulignons l'importance du *Big Data* et de l'accélération matérielle des GPU. Enfin, nous mentionnons les tendances fortes actuelles à travers l'apprentissage auto-supervisé, les mécanismes attentionnels et les transformers.

Abstract – This paper gives an overview of the history of neural networks for signal and image processing, from their foundations until the successes of modern deep learning. We present their key elements in terms of architecture and training, and highlight the importance of the big data aspects and GPU hardware acceleration. Finally, we mention current strong trends towards self-supervised learning, attention mechanisms and transformers.

1 Introduction

L'histoire des réseaux de neurones (NN — *Neural Networks*) s'inscrit à la frontière de plusieurs domaines scientifiques : l'informatique, les neurosciences, le traitement de signal, l'automatique, ou la vision par ordinateur. La Figure 1 montre que les grandes lignes de la méthodologie s'établissent très tôt avec l'introduction des premiers modèles linéaires, puis multi-couches, de la retropropagation du gradient, des modèles récurrents et des couches convolutives. L'ère de l'apprentissage profond moderne débute vers 2006 et se généralise à partir de 2012. Le concept de l'attention introduit en 2017 établit un modèle unifié capable de répondre à une multitude de modalités et de problèmes. Avec la fin des années 2010, l'impact des modèles de fondation et les grands modèles de langage (LLM) se fait sentir au delà du traitement automatique du language.

Pendant longtemps, le développement des réseaux de neurones a été mené parallèlement au développement des modèles graphiques probabilistes (PGM — *Probabilistic Graphical Models*), qui, dans le domaine du traitement du signal et des images, constituaient l'une des méthodologies dominantes. Les PGM modélisent la distribution conjointe $p(x, y)$ des variables observées y et cachées x, ou la distribution conditionnelle $p(x|y)$ selon la variante, la factorisation étant définie par la structure du graphe associé[1]. Contrairement aux réseaux de neurones, la littérature des PGM met l'accent sur la minimisation globale et exacte des fonctions d'énergie sous-jacentes, *e.g.* par programmation dynamique, propagation de croyances ou coupures de graphes. Ceci limite le pouvoir expressif des PGM à des interactions simples entre variables.

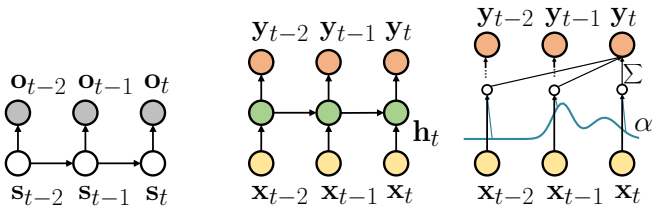
En revanche, la littérature des NN met l'accent sur des modèles à haute capacité comprenant des interactions riches et expressives entre les variables, avec une optimisation réalisée de manière approximative par descente de gradient. Dans leur forme la plus pure, les NN n'effectuent des étapes d'optimisation uniquement pendant la phase d'apprentissage, tandis que les PGM effectuent généralement des étapes d'optimisation au moment de l'apprentissage *et* du test, ajustant le modèle à l'observation courant grâce à l'estimation des variables cachées. Comme on le verra dans la suite, dans de nombreuses applications, ce compromis entre capacité et facilité d'optimisation a fait pencher la balance en faveur des réseaux de neurones profonds. L'histoire a montré que leur grande capacité compense largement les difficultés qu'ils génèrent en termes d'optimisation exacte et globale.

En pratique, les deux familles de modèles ne sont pas strictement séparées. L'histoire des réseaux profonds a même débuté par l'introduction des machines de Boltzman restreintes et leur apprentissage non supervisé sur les GPU par Hinton en 2006 [17]. Techniquement parlant, ces modèles sont des PGM, plus spécifiquement des champs aléatoires de Markov (MRF).

2 Avant l'ère du deep learning

Les débuts des réseaux de neurones ont été inspirés par les recherches en neurosciences : on peut ainsi citer la première modélisation d'un neurone par McCullogh et Pitts en 1943 [22]

[1]La notation de la littérature en apprentissage est inversée par rapport à la littérature classique en TSI. Dans cette litérature, les CRF modélisent $p(y|x)$, x étant le champs observé, y le champs caché.



(a) (b) (c)

FIGURE 2 : Examples de modèles pour traitement de séries temporelles avant et durant l'ère du deep learning. (a) : modèles de Markov cachées (HMM) : Les sommets du graphe correspondent aux variables aléatoires observées et cachées, les arêtes encodent les propriétés d'indépendance conditionnelle de la distribution jointe ; (b) réseaux de neurones récurrents : les sommets du graphe correspondent aux neurones, les arêtes aux flux de traitement lors d'un passage ; (c) les "transformers" calculent des distributions d'attention entre toute paire de sommets, qu'ils soient voisins ou pas.

et la proposition par Rosenblatt du *Perceptron* en 1958, un modèle linéaire. Puis, la recherche sur les NN a été plongée dans un état quasi dormant par la compréhension que ces modèles ne peuvent résoudre que des problèmes linéairement séparables, c.à.d. très simples. Ce blocage a été résolu par l'apparition des réseaux de neurones non-linéaires sous la forme de perceptrons multicouches (MLP — *Multi Layer Perceptrons*) et la découverte de leur capacité d'approximation universelle [4]. Un algorithme efficace a été proposé pour leur apprentissage, la rétro-propagation du gradient. Celui-ci permet de calculer le gradient de la fonction objectif (la "perte") par rapport à l'ensemble des paramètres du modèle, quelque soit leur couche d'appartenance, en appliquant, de manière itérative, le théorème de la dérivation des fonctions composées.

Pour illustrer ce processus, considérons une unité (un "neurone") j, recevant des entrées z_i venant d'unités i et livrant sa sortie a_j à des unités k de la couche suivante. Le gradient de la perte $\mathcal{L}$ par rapport au paramètre w_{ij} liant unités i et j peut être exprimé avec la récurrence suivante,

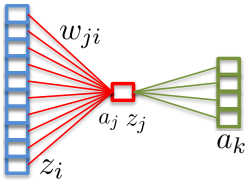


FIGURE 3 : Un neurone a_j intégré dans un réseau.

$$\frac{\partial \mathcal{L}}{\partial w_{ji}} = \frac{\partial \mathcal{L}}{\partial a_j} z_i, \quad \frac{\partial \mathcal{L}}{\partial a_j} = h'(a_j) \sum_k w_{kj} \frac{\partial \mathcal{L}}{\partial a_k} \qquad (1)$$

où h' est le gradient de la fonction d'activation h appliquée à la sortie de l'unité.

L'application des NN au traitement d'informations visuelles a été également inspirée par des travaux précurseurs en neurosciences. Le travail lauréat du prix Nobel obtenu par Hubel et Wiesel sur les circuits de traitement visuel en 1962 a donné lieu à l'introduction des convolutions dans les réseaux de neurones (c.f Figure 4, gauche) par Fukushima en 1980 [11], et puis à leur forme moderne introduite par LeCun en 1989, introduisant la rétropropagation des gradients pour les couches convolutives [21].

Parallèlement, l'adaptation des NN au traitement des séries temporelles a permis des applications en traitement du signal. Les premières tentatives sous la forme de réseaux récurrents sont illustrées dans la Figure 2b : un "état caché" permet de garder une mémoire arbitrairement riche, car réalisée sous forme d'un vecteur $\mathbf{h}_t$ maintenu durant l'évolution du temps t. La dynamique du processus en question est modélisée par des fonctions de mise à jour de cet état. L'apprentissage se fait par rétropropagation du gradient à travers le temps. Ces modèles souffraient initialement de difficultés d'entraînement causées par des instabilités numériques, connues sous les noms de disparition ou d'explosion des gradients. Une percée a été franchie lorsque Hochreiter et Schmidhuber ont introduit les réseaux LSTM ("*Long-Short Term Memory*") en 1996 [18]. La solution consiste à introduire un ensemble de fonctions dites de "*gating*", permettant au réseau de réguler le flux d'informations dans chaque cellule et de contrôler la vitesse de la mise à jour de la mémoire neuronale. Des associations entre CNN et LSTM ont été proposées pour le traitement de la vidéo [2]. Les réseaux LSTM ont été l'architecture dominante pour la décision séquentielle pendant de nombreuses années jusqu'à l'introduction des "transformeurs" en 2017 (section 5).

3 Deep learning et big data

Les NN ont connu des périodes de succès et de déclin au cours des quatre dernières décennies. Une période de réticence à leur égard a eu lieu grossièrement au cours de la période allant des années 1990 à 2010, comme illustré dans la figure 1, qui a été marquée par un âge d'or des méthodes à noyau ("*Kernel methods*") et de la conception manuelle de descripteurs, notamment le modèle sac de mot ("*Bag of Words*", BoW), qui, venu du domaine du traitement du langage et de la recherche d'information a imprégné rapidement d'autres domaines connexes comme la reconnaissance visuelle ou audio.

Un renouveau majeur a eu lieu au début des années 2010, qui a été impulsé par les succès expérimentaux majeurs obtenus par les NN pour des tâches de perception sur des données de natures diverses. Une première impulsion a été observée en 2011 dans le domaine de la reconnaissance de la parole [5], où des réseaux de neurones ont surclassé les méthodes traditionnelles reposant sur des descripteurs manuels. Un succès retentissant a eu lieu un an plus tard sur la base de données *ImageNet* et le challenge de classification d'images large échelle (*ILSVRC'12*), où des réseaux convolutifs profonds [20] ont permis un gain substantiel par rapport aux méthodes dominantes et basées sur le modèle BoW. Ce "*Big Bang*" du deep learning moderne a marqué durablement l'histoire des NN. Les réseaux utilisés par [20] à *ILSVRC'12* mimaient fortement leurs ancêtres des années 80 [21]. Ils ont montré la capacité des deep NN à apprendre des représentations très puissantes à partir de grandes masses de données étiquetées, ainsi que la très forte accélération de l'entraînement rendu possible par le portage des calcul sur cartes graphiques (GPU). Après 2012, des améliorations des modèles en termes d'architecture, comme les connexions résiduelles avec les ResNets [15], ont permis d'entraîner des modèles de plus en plus profonds (plusieurs centaines ou milliers de couches) et performants. Un dernier élément clé concerne les capacité de "Transfert" des NN préentraînés sur *ImageNet* [1], qui constituent la référence en vision pour les appliquer sur des volumétries de données plus modestes.

Au-delà des tâches de classification, les deep NN ont aussi été adaptés pour effectuer des prédictions au niveau de chaque pixel d'une image, tâche connue sous le nom de prédiction dense. Notamment, des approches dédiées ont été proposées pour la détection d'objets [12], la segmentation sémantique d'images [3] ou l'estimation de pose [13]. Des architecture à base d'"encodeur-décodeurs" se sont imposées

comme le très populaire réseau U-Net [27] pour la segmentation d'images médicales et incluant des connexions directes ("*skip connexions*") entre l'encodeur et le décodeur pour faciliter la super-résolution spatiale dans le décodeur.

4 Apprentissage auto-supervisé

Pour des problèmes de reconnaissance sémantique, il est nécessaire de s'abstraire de l'information d'entrée pour apprendre des représentations pertinentes. Dans ce contexte, les méthodes standard pour entraîner des NN de manière non supervisée et reposant sur l'erreur de reconstruction (MSE) ou la maximisation de la vraisemblance ont été remises en cause [25, 26].

Une famille de méthodes qui a imprégné le deep learning depuis 10 ans concerne l'apprentissage auto-supervisé. L'idée est de convertir une tâche non supervisée en une tâche supervisée. Cette dernière, appelée "tâche prétexte", correspond le plus souvent à une tâche de classification dont l'intuition est de fournir un signal d'apprentissage nécessitant l'extraction de représentations sémantiques pour la résoudre. Les représentations apprises de manière auto-supervisée sont ensuite utilisées pour différentes tâches de reconnaissances spécifiques.

Diverses tâches prétextes ont été introduites pour la littérature, comme la prédiction de la rotation d'une image, la colorisation, la prédiction de la position relative ou absolue d'un patch ("puzzle"), des méthodes de reconstruction de données manquantes ("inpainting") consistant à prédire une région d'image à partir du contexte. Ces approches sont inspirées des méthodes de masquage utilisées pour entraîner les modèles de traitement du langage, et on été appliqués récemment sous la forme de masked autoencoders [14] pour l'analyse d'images.

D'autres méthodes consistent dites "contrastives" utilisent la discrimination d'instance comme tâche prétexte. Elles consistent à construire des paires d'images positives, *i.e.* correspondant à la même instance, et négatives. Ces approches ont été appliquées en générant différentes augmentations de données préservant la classe dans SimCLR, et adaptées pour contrôler le nombre de négatifs dans un batch avec des mécanismes de mémoire dans MoCO [16]. D'autres méthodes abandonnent l'utilisation de paires négatives, comme dans BYOL [9] où le modèle apprend à accroître la similarité entre deux version faiblement et fortement augmenté d'une instance dans un schéma d'apprentissage "teacher/student".

Les méthodes d'apprentissage auto-supervisées permettent d'apprendre avec des volumétries de données toujours plus importantes. Elles se sont montrées compétitives par rapport des méthodes supervisées pour l'entraînement des NN. Elles présentent également des propriétés de robustesse intéressantes, et se montrent souvent supérieures à leur concurrents supervisés pour la détection d'anomalies (points hors de la distribution d'entraînement, OOD).

5 Attention et transformers

Durant une certaine période, jusqu'à 2017, les modèles neuronaux dominant pour le TSI étaient les RNN dans la variante LSTM pour les séries temporelles, et les réseaux de neurones convolutifs (CNN) pour le traitement d'images. Les deux types de réseaux disposaient de symétries intéressantes, les premiers étant Markoviens, et le deuxième jouissant d'équivariance en

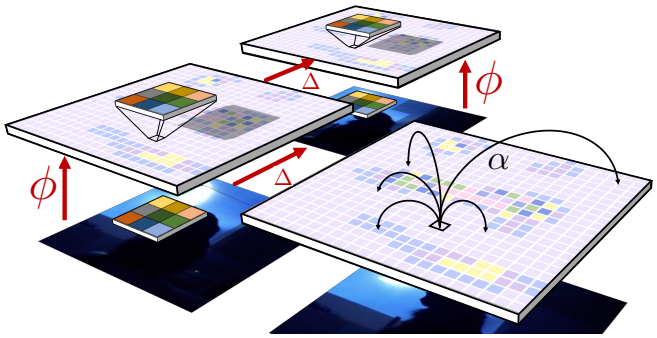


FIGURE 4 : Les couches convolutives (gauche, indiquées par ϕ) sont caractérisées par un champs réceptif limité et par la proprité de l'équivariance par rapport à une translation Δ, c.à.d. $\phi(\mathbf{x}+\Delta) = \phi(\mathbf{x}) + \Delta$ [Lien-animation]] . Les couches basées sur l'attention (droite), d'un réseau ViT [7], une adaptation de [31] au traitement des images, permettent de gèrer les interactions spatiales de longues portées par une fonction de similarité apprise.

translation. Plus particulièrement, pour une entrée $\mathbf{x}$, un opérateur de décalage $+$ et une couche convolutive ϕ, nous obtenons $\phi(\mathbf{x}+\Delta) = \phi(\mathbf{x}) + \Delta$, voir la Figure 4a. Conformément à la théorie classique de l'apprentissage [29], il était généralement accepté que la réduction de l'expressivité d'une classe de modèles par ajout de biais inductifs, tel que la propriété d'équivariance, pouvait mener à une meilleure généralisation aux données non vues. Or, des 2017 avec l'introduction des modèles de type "transformers" par Vaswani et al. [31], cette tendance se verra inverser. D'abord introduit en traitement automatique de la langue pour remplacer les LSTM, ces modèles sont équivariants à la permutation, mais non-équivariant à la translation — ils traitent une séquence de données comme un "sac" d'éléments non ordonnés, conduisant même à l'ajout d'un pré-traitement des données par encodage de leurs positions pour rendre possible la prise en compte de l'ordre.

Dans sa version la plus simple, un transformer utilise le principe de la "*self-attention*", le calcul d'une distribution d'attention $\alpha_{i,j}$ mesurant une similarité entre toute paire (i,j) de la séquence $\{\mathbf{x}_t\}_{t=1\ldots T}$ (voir également la Figure 2c) :

$$\alpha_{i,j} = \frac{\exp\left(e_{i,j}\right)}{\sum_{i'=1}^{H}\exp\left(e_{i',j}\right)}, \quad e_{i,j} = \frac{\left(\mathbf{x}_j^T W_Q\right)^T\left(\mathbf{x}_i^T W_K\right)}{\sqrt{D}}. \tag{2}$$

où D est une constante dépendant de la dimension du problème et W_Q et W_K sont des projections linéaires nommées "*Query*" et "*Key*". L'attention est utilisée pour pondérer une somme sur une projection des données d'entrée :

$$\mathbf{y}_j = \sum_i \alpha_{i,j}\mathbf{x}_i^T W_V \ + \mathbf{x}_j \tag{3}$$

où W_V est une troisième projection nommée "*Value*". Intuitivement, l'apprentissage des projections *Query* et *Key* permet de définir un espace de similarité entre les vecteurs de la séquence, alors que la projection *Value* va définir la transformation des données par le modèle, pondérée par les calculs de similarité. En pratique, ces calculs sont cumulés en plusieurs têtes, dont les résultats sont sommés, et puis en couches.

Il s'avère que ce choix de conception était plus que pertinent, les transformers se sont imposés dans plusieurs domaines scientifiques tels que le traitement automatique de la langue, la vision par ordinateur, le traitement de signal et la robotique pour la prise de décisions séquentielles. L'application en traitement de signal est direct, s'agissant de séries temporelles. En vision par ordinateur, le ViT ("*Vision Transformer*")

[7] applique le principe d'attention aux patches d'une image, après avoir appliqué une transformation linéaire sur le signal brut, voir la Figure 4, droite. La "cross-attention" généralise le concept d'attention aux signaux multi-modaux, généralement en calculant les projections *Key* et *Value* sur une modalité et la projection *Query* sur une autre, en alternant. Cela permet de définir des modèles mélangeant vision et langage [30], des vues géométriques [28], ou des données issues de la physique [19].

La recherche actuelle explore l'usage massive de données et de moyens de calculs pour apprendre des modèles de capacités hors mesure : Au moment de l'écriture de cet article en 2023, un grand modèle de langue (LLM — *Large Language Model*) typiquement dispose de centaines de milliards de paramètres [6], jusqu'à un billion paramètres (10^{12}), les modèles pour la vision atteignant jusqu'à 20 milliards de paramètres. Entraînés de manière self-supervisée (section 4), les résultats sont au rendez-vous : gains de performances, et pour les LLM, raisonnement de haut niveau, parfois bluffant sur certains cas, tout en générant des erreurs assez surprenantes dans d'autres cas. Surtout, ces modèles ont démontré des capacités d'adaptation à des nouvelles tâches surprenantes, leur donnant le nom "*Foundation Models*". Dans le cas des LLM, cela se fait sans re-entrainement, la tâche étant spécifiée en ajoutant un préfix textuel ("prompt") aux entrées lors de la phase de test [6]. Les modèles s'avèrent également puissant en vision par ordinateur [10], reconnaissance de parole [24], robotique [8] etc.

Conclusion L'inflation actuelle concernant l'augmentation massive de la capacité des modèles, de la quantité des données d'entraînement et des moyens de calculs soulève des questions à de nombreux niveaux, tant sur des aspects de souveraineté numérique que d'empreinte écologique des modèles. Cependant, une alternative importance se dessine pour la conception de modèles hybrides permettant un apprentissage plus frugal et économe, en exploitant des connaissances du domaine en traitement du signal et des images, , *e.g.* sur la géométrie [23] ou les modèles physiques [32].

Références

[1] H. AZIZPOUR, A.S. RAZAVIAN, J. SULLIVAN, A. MAKI et S. CARLSSON : Factors of transferability for a generic convnet representation. *IEEE-T-PAMI*, 38(9):1790–1802, 2016.

[2] M. BACCOUCHE, F. MAMALET, C. WOLF, C. GARCIA et A. BASKURT : Sequential deep learning for human action recognition. *In HBU*, 2011.

[3] L.-C. CHEN, G. PAPANDREOU, I. KOKKINOS, K. MURPHY et A.L. YUILLE : Semantic image segmentation with deep convolutional nets and fully connected crfs. *In ICLR*, 2015.

[4] G CYBENKO : Approximations by superpositions of sigmoidal functions. *Mathematics of Control, Signals, and Systems*, 2(4): 303–314, 1989.

[5] G. E. DAHL, Dong YU, Li DENG et A. ACERO : Context-dependent pre-trained deep neural networks for large-vocabulary speech recognition. *T-ASL*, 20(1), 2012.

[6] Brown et AL. : Language models are few-shot learners. *In NeurIPS*, 2020.

[7] Dosovitskiy et AL. : An image is worth 16x16 words : Transformers for image recognition at scale.

[8] Driess et AL. : Palm-e : An embodied multimodal language model. *arXiv :2303.03378*, 2022.

[9] J.B. Grill et. AL. : Bootstrap your own latent - a new approach to self-supervised learning. *In NeurIPS*, 2020.

[10] Yuan et AL. : Florence : A new foundation model for computer vision. *arxiv :2111.11432*, 2021.

[11] K. FUKUSHIMA : Neocognitron : a self organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biol. Cybern.*, 36:30–42, 1980.

[12] R. GIRSHICK, J. DONAHUE, T. DARRELL et J. MALIK : Rich feature hierarchies for accurate object detection and semantic segmentation. *In CVPR*, 2014.

[13] R. Alp GÜLER, N. NEVEROVA et I. KOKKINOS : Densepose : Dense human pose estimation in the wild. *In CVPR*, 2018.

[14] K. HE, X. CHEN, S. XIE, Y. LI, P. DOLLÁR et R. GIRSHICK : Masked autoencoders are scalable vision learners. *In CVPR*, 2022.

[15] K. HE, X. ZHANG, S. REN et J. SUN : Deep residual learning for image recognition. *In CVPR*, 2016.

[16] Kaiming HE, Haoqi FAN, Yuxin WU, Saining XIE et Ross GIRSHICK : Momentum contrast for unsupervised visual representation learning. *In CVPR*, 2020.

[17] G.E. HINTON et R.R. SALAKHUTDINOV : Reducing the dimensionality of data with neural networks. *Science*, 2006.

[18] S. HOCHREITER et J. SCHMIDHUBER : Long short-term memory. *Neural Comput.*, 9(8), 1997.

[19] S. JANNY, A. BENETTEAU, M. NADRI, J. DIGNE, N. THOME et C. WOLF : Eagle : Large-scale learning of turbulent fluid dynamics with mesh transformers. *In ICLR*, 2023.

[20] A. KRIZHEVSKY, I. SUTSKEVER et G.E. HINTON : Imagenet classification with deep convolutional neural networks. *In NIPS*, 2012.

[21] Yann LECUN, Bernhard BOSER, John S DENKER, Donnie HENDERSON, Richard E HOWARD, Wayne HUBBARD et Lawrence D JACKEL : Backpropagation applied to handwritten zip code recognition. *Neural computation*, 1(4):541–551, 1989.

[22] W.S. MCCULLOCH et W. PITTS : A logical calculus of the ideas immanent in nervous activity. *Bull. Math. Biophys*, 5, 1943.

[23] B. MILDENHALL, P. SRINIVASAN, M. TANCIK, J.T. BARRON, R. RAMAMOORTHI et R. NG : NeRF : Representing scenes as neural radiance fields for view synthesis. *In ECCV*, 2020.

[24] A. RADFORD, J.W. KIM, T. XU, G. BROCKMAN, C. MCLEAVEY et I. SUTSKEVER : Robust speech recognition via large-scale weak supervision, 2022.

[25] Antti RASMUS, Mathias BERGLUND, Mikko HONKALA, Harri VALPOLA et Tapani RAIKO : Semi-supervised learning with ladder networks. *In NeurIPS*, volume 28, 2015.

[26] T. ROBERT, N. THOME et M. CORD : Hybridnet : Classification and reconstruction cooperation for semi-supervised learning. *In ECCV*, 2018.

[27] O. RONNEBERGER, P. FISCHER et T. BROX : U-net : Convolutional networks for biomedical image segmentation. *In MICCAI*, 2015.

[28] A. SAHA, O. MENDEZ, C. RUSSELL et R. BOWDEN : Translating Images into Maps. *In ICRA*, 2022.

[29] S. SHALEV-SHWART et S. BEN-DAVID : *Understanding Machine Learning - From Theory to Algorithms*. Cambridge University Press, 2014.

[30] H.H. TAN et M. BANSAL : LXMERT : Learning Cross-Modality Encoder Representations from Transformers. *In EMNLP*, 2019.

[31] A. VASWANI, N. SHAZEER, N. PARMAR, J. USZKOREIT, L. JONES, A. GOMEZ, L. KAISER et I. POLOSUKHIN : Attention is all you need. *In NeurIPS*, 2017.

[32] Y. YIN, V. Le GUEN, J. DONA, I. AYED, de E. BÉZENAC, N. THOME et P. GALLINARI : Augmenting physical models with deep networks for complex dynamics forecasting. *In ICLR*, 2021.