

# Intelligence artificielle 101: Webinaire

## Notes de classe

INSTRUCTEUR: VINCENT BOUCHER  
president@montreal.ai

---

**Samedi, 28 novembre 2020 | 10:00 – 11:30 EST**

### Allocution inaugurale

Bienvenue à "Intelligence Artificielle 101".

L'IA ouvre un monde de nouvelles possibilités.

Estimant que chaque être sensible mérite de jouer un rôle dans l'évolution de l'intelligence artificielle, l'Académie Montréal.IA présente: « ***Intelligence Artificielle 101 : Un survol complet de l'IA destiné au grand public*** ».

*L'intelligence est "la partie informatique de la capacité à prédire et à contrôler un flux d'expériences." — Rich Sutton, 2020*

Dans la Session 1 · Apprentissage profond, nous verrons que l'apprentissage profond est une procédure d'apprentissage général et qu'un pourcentage significatif des percées de l'apprentissage profond provient de constructions réutilisables et du partage de paramètres: les couches CNN réutilisent les poids à plusieurs endroits (partage de paramètres dans l'espace); les couches RNN réutilisent les poids pour plusieurs étapes (partage de paramètres dans le temps); les transformeurs sont des architectures génériques, intéressants et puissants conçus pour traiter un ensemble connecté d'unités (entités dans une séquence (temps), pixels dans une image (espace), etc).

Nous considérerons une classe générale d'algorithmes d'apprentissage non supervisés qui peuvent être construits simplement en prédisant n'importe quelle partie des données à

---

partir de n'importe quelle autre: *"Donnez une étiquette à un robot et vous l'alimentez une seconde; apprenez à un robot à étiqueter et vous l'alimentez pour la vie."* - Pierre Sermanet

Dans la Session 2 · Agents autonomes, considérant que l'évolution et les réseaux de neurones se sont avérés une combinaison puissante dans la nature, nous étudierons les algorithmes 'ouverts' : des algorithmes qui créent sans fin. Reconnaisant que l'évolutivité peut nécessiter plus que des réseaux neuronaux pour s'épanouir, nous explorerons également la possibilité d'apprendre une version de la conception de l'agent lui-même qui est mieux adaptée à sa tâche.

Notant avec satisfaction que des agents autonomes ont commencé à obtenir des RÉSULTATS SUPERHUMAINS, nous étudierons comment l'apprentissage par renforcement profond utilise les réseaux neuronaux profonds dans les algorithmes d'apprentissage par renforcement (permettant d'apprendre la correspondance entre des entrées sensorielles brutes et des sorties motrices brutes) et nous étudierons aussi des agents polyvalents qui sont capables d'adapter continuellement leur comportement sans intervention humaine.

Dans la Session 3 · Environnements, en reconnaissant que la vraie compréhension vient des agents autonomes qui apprennent en combinaison avec la façon dont ils affectent le monde, nous allons voir quelques environnements pour entraîner votre agent autonome : OpenAI Gym, DeepMind Lab et Unity 3D Machine Learning Agents. Comment créer de nouveaux environnements avec l'OpenAI Gym.

*"Ce que je ne peux pas créer, je ne comprends pas."* — Richard Feynman

"VIP AI 101 CheatSheet for All" : <http://www.montreal.ai/ai4all.pdf>

## Session 0 · Démarrage

- Dans le nuage (cloud). "Colab" permet à tous d'écrire et d'exécuter un code python arbitraire par le biais du navigateur. Il s'agit d'un notebook qui ne nécessite aucune installation pour être utilisé (et qui fonctionne entièrement dans le nuage (cloud)), tout en offrant un accès gratuit aux ressources informatiques, y compris les GPU.

- <https://colab.research.google.com/notebooks/welcome.ipynb>
- <https://colab.research.google.com/github/madewithml/basics/>

- 
- <https://colab.research.google.com/github/cs231n/cs231n.github.io/blob/master/python-colab.ipynb>

- Sur une machine locale : Installez Anaconda. (sur l'aide-mémoire "VIP AI 101 CheatSheet for All" - On a Local Machine (2.2)).

- Google's Dataset Search <https://datasetsearch.research.google.com/> est un outil qui permet de découvrir des millions d'ensembles de données sur le web :

*"Dataset Search" a indexé près de 25 millions de ces ensembles de données, vous permettant ainsi de trouver en un seul endroit des ensembles de données et de trouver des liens vers l'endroit où se trouvent ces données." — Natasha Noy*

- TensorFlow ( <https://www.tensorflow.org> ) : Une plateforme d'apprentissage machine complète et open source pour la recherche et la production.

<https://www.tensorflow.org/tutorials>

- Keras. Si vous désirez intégrer des fonctionnalités optimisées d'apprentissage en profondeur dans de vrais produits, vous pouvez simplement utiliser Keras. Introduction à Keras pour les ingénieurs, par François Chollet, Colab :

<https://colab.research.google.com/drive/1IWUGZarIbORaHYUZIF9muCgpPI8pEvve>

- PyTorch ( <https://pytorch.org> ) : Un framework d'apprentissage machine open source qui accélère la voie du prototypage de recherche au déploiement de production.

<https://medium.com/pytorch/get-started-with-pytorch-cloud-tpus-and-colab-a24757b8f7fc>

- TensorFlow Hub est une bibliothèque de modules réutilisables d'apprentissage machine.

<http://tensorflow.org/hub>

*"Je pense que l'apprentissage par transfert est la clé de l'intelligence générale. Et je pense que la clé de l'apprentissage par transfert sera l'acquisition de connaissances conceptuelles abstraites des détails perceptuels de l'endroit où vous les avez apprises." — Demis Hassabis*

- TensorFlow.js ( <https://www.tensorflow.org/js/> ) permet l'apprentissage automatique dans le navigateur Web.

---

- TensorFlow Playground: Entraîner un réseau de neurones s'exécutant dans votre navigateur. <http://playground.tensorflow.org/>

## Session 1 • Apprentissage profond | Réseaux de neurones

L'apprentissage selon Mitchell (1997) :

*"On dit qu'un programme informatique apprend de l'expérience  $E$ , étant donné certaines classes de tâches  $T$  et une mesure de performance  $P$ , si sa performance de réalisation des tâches  $T$ , telle que mesurée par  $P$ , s'améliore avec l'expérience  $E$ ." — Tom Mitchell*

Selon David Foster:

*"L'apprentissage profond est une classe d'algorithme d'apprentissage machine qui utilise plusieurs couches d'unités de traitement de l'information pour apprendre des représentations de haut niveau à partir de données non structurées." — David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play*

*"Cette forme d'apprentissage machine permet d'apprendre par l'expérience sans qu'un humain lui spécifie formellement toutes les connaissances nécessaires." — Deep Learning Book, By Ian Goodfellow, Yoshua Bengio and Aaron Courville*

Débat historique sur l'IA | Clarification de la définition de l'apprentissage profond:

*"L'apprentissage profond s'inspire des réseaux de neurones du cerveau pour construire des machines d'apprentissage qui découvrent des représentations internes riches et utiles, calculées comme une composition de caractéristiques et de fonctions apprises." — Yoshua Bengio*

Le théorème d'approximation universelle stipule essentiellement qu'un réseau à propagation avant avec une seule couche cachée contenant un nombre fini de neurones peut représenter une approximation de n'importe quelle fonction avec un degré de précision arbitraire, à condition qu'il ait suffisamment d'unités cachées.

Apprentissage profond : connecter un ensemble de données, un modèle, une fonction de coût et une procédure d'optimisation.

---

*"Vous avez des éléments de traitement relativement simples qui sont très vaguement des modèles de neurones. Ils ont des connexions qui arrivent, chaque connexion a un poids, et ce poids peut être changé par l'apprentissage. Et ce qu'un neurone fait, c'est prendre les activités sur les connexions multipliées par le poids, tous les additionner, puis décider d'envoyer une sortie. Si le neurone reçoit une somme suffisante, il envoie une sortie. Si la somme est négative, le neurone n'envoie rien. C'est à peu près ça." — Geoffrey Hinton*

<https://www.wired.com/story/ai-pioneer-explains-evolution-neural-networks/>

Les valeurs correctes pour les poids et les biais déterminent la force des prédictions. L'entraînement de réseau de neurones (la rétropropagation) selon Ilya Sutskever: apporter de petites modifications aux poids de votre réseau neuronal jusqu'à ce que sa prédiction satisfasse les données (modifiez les paramètres qui minimisent l'erreur). La procédure de rétropropagation est l'algorithme clé qui rend l'entraînement de modèles profonds hautement efficace (la procédure de rétropropagation pour calculer le gradient d'une fonction objectif par rapport aux poids d'une pile multicouche de modules n'est rien de plus qu'une application pratique de la règle des dérivés en chaîne (Mémoire de 1676 de Gottfried Wilhelm Leibniz ...)).

Règles émergentes à l'aube de l'ère de l'intelligence artificielle. Le 13 mars 2019, Rich Sutton (le père de l'apprentissage par renforcement) a publié : The Bitter Lesson. Pour Rich Sutton, la plus grande leçon que l'on puisse tirer de 70 ans de recherche en IA est la grande puissance des méthodes d'application générale, des méthodes qui continuent à évoluer avec l'augmentation de la puissance de calcul, même si la puissance de calcul disponible devient très importante. Les deux méthodes qui semblent évoluer arbitrairement de cette façon sont la recherche et l'apprentissage. Ainsi, les règles émergentes à l'aube de l'ère de l'intelligence artificielle sont la recherche et l'apprentissage. L'apprentissage profond est une procédure d'apprentissage générale.

Constructions réutilisables. *"Pratiquement tous les observateurs modernes admettraient que les gènes et l'expérience fonctionnent ensemble ; c'est "la nature et l'apprentissage", et non "la nature contre l'apprentissage" ... et des nativistes tels que Pinker, Peter Marler (Marler, 2004) et moi-même (Marcus, 2004) avons fréquemment défendu un point de vue selon lequel une partie importante de l'arsenal inné d'une créature consiste non pas en des connaissances spécifiques*

---

*mais en des mécanismes d'apprentissage, une forme d'innatalité qui permet d'apprendre."* – Gary Marcus, Innateness, AlphaZero, and Artificial Intelligence

## Session 1 · Apprentissage profond | Réseaux de neurones convolutifs

Les couches des réseaux de neurones convolutifs approfondis réutilisent les paramètres à plusieurs endroits.

*"Les ConvNets sont conçus pour traiter des données qui se présentent sous la forme de plusieurs tableaux, par exemple une image couleur composée de trois tableaux 2D contenant des intensités de pixels ..."* — Deep Learning. Yann LeCun, Yoshua Bengio et Geoffrey Hinton. Nature 521, 436–444 (28 mai 2015)

À la fin des années 1980, Geoffrey Hinton et son équipe ont réussi à appliquer la rétropropagation à l'entraînement de réseaux de neurones approfondis. Au cours des années 1990, une équipe dirigée par Yann LeCun, ancien étudiant postdoctoral de Hinton, a entraîné un réseau convolutif approfondi nommé «LeNet», afin de classer les images de chiffres manuscrits avec une grande précision.

### Convolution

Convolution : Par exemple, on glisse un filtre 3x3 sur l'image d'entrée, on multiplie par élément (produit matriciel de Hadamard, parfois désigné produit de Schur) et on ajoute les sorties. Dans les réseaux de neurones convolutifs, les filtres sont appris avec un algorithme d'apprentissage. Bloc de code Tensorflow. Tensorflow code block: **tf.keras.layers.Conv2D**  
[https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/Conv2D](https://www.tensorflow.org/api_docs/python/tf/keras/layers/Conv2D)

### Max Pooling

Agglomération max "Max Pooling" : La couche d'agglomération (pooling) sous-échantillonne le volume spatialement (réduit la dimensionnalité) tout en préservant la structure spatiale: invariance spatiale. Bloc de code Tensorflow:

**tf.keras.layers.MaxPool2D**

[https://www.tensorflow.org/api\\_docs/python/tf/keras/layers/MaxPool2D](https://www.tensorflow.org/api_docs/python/tf/keras/layers/MaxPool2D)

---

CNN Explainer : Une visualisation intuitive et interactive expliquant le fonctionnement de la convolution et de la mise en commun (max pooling), à l'aide d'exemples réels dans le navigateur. Wang et al.: <https://arxiv.org/abs/2004.15004> Interactive visualization in the browser: <http://poloclub.github.io/cnn-explainer/>

Voici un cahier de démarrage colab, par François Chollet (2020), démontrant le flux de travail pour former un bon modèle de classification d'image à partir de zéro sur l'ensemble de données de classification binaire Cats vs Dogs, à partir de 25000 images JPEG, en moins de 2 heures, entièrement gratuitement, sur un GPU Colab :

<https://colab.research.google.com/drive/1umJnCp8tZ7UDTYSQsuWdKRhqbHts38AC>

Apprentissage de représentations: le langage des réseaux de neurones. Le vocabulaire visuel d'un réseau de neurones convolutifs approfondi semble émerger de caractéristiques de bas niveau telles que les bords (edges) et les orientations. Selon Alex Graves, cela est relié aux "Jeux de langage" de Wittgenstein dans les Investigations philosophiques, où un langage fonctionnel émerge de tâches simples avant de définir un vocabulaire.

- <https://tensorspace.org/>
- <https://arxiv.org/abs/1603.07285>

## **Session 1 · Apprentissage profond | Réseaux de neurones récurrents**

Recurrent Neural Networks (RNN) : Constructions réutilisables qui réutilisent les paramètres dans le temps.

*"Les RNN traitent une séquence d'entrée un élément à la fois, en conservant dans leurs unités cachées un 'vecteur d'état' qui contient implicitement des informations sur l'histoire de tous les éléments ..."* — Deep Learning. Yann LeCun, Yoshua Bengio & Geoffrey Hinton. Nature 521, 436 –444 (28 May 2015)

Les RNN sont des réseaux contenant des boucles permettant aux informations de persister dans le temps [Rumelhart et al., 1986].

---

Les RNN sont "Turing complet" (ordinateur universel) : ils peuvent encoder toute la gamme des algorithmes d'apprentissage possibles.

En raison de la simplicité des RNN, la disparition du gradient devient un problème pour les dépendances à long. En utilisant une couche de réseau de neurones sigmoïdes (ou un réseau de neurones profonds) et une opération de multiplication ponctuelle, nous pouvons former des portes (des Gates) pour faire un choix. Avec trois de ces portes, un LSTM peut être construit. Comme les LSTM ont la capacité de rejeter ou d'inclure des informations à l'état de cellule, soigneusement régulées par les portes, les LSTM empêchent le problème de la disparition du gradient.

Les RNN peuvent être augmentés. Les interfaces attentionnelles, permettant de se concentrer sur des parties sélectionnées de l'entrée, sont une technique extrêmement générale et puissante.

Selon Lilian Weng : *"L'auto-attention est un mécanisme d'attention reliant différentes positions d'une séquence afin de calculer une représentation de la même séquence."* - Lilian Weng  
<https://lilianweng.github.io/lil-log/2018/06/24/attention-attention.html>

*"Une fonction d'attention peut être décrite comme établissant une correspondance entre une requête et un ensemble de paires clé-valeur sur une sortie, où la requête, les clés, les valeurs et la sortie sont tous des vecteurs. La sortie est calculée comme une somme pondérée des valeurs, où le poids attribué à chaque valeur est calculé par une fonction de compatibilité de la requête avec la clé correspondante."* "Attention Is All You Need" Vaswani et al. -  
<https://arxiv.org/abs/1706.03762>

L'opération d'auto-attention est l'opération fondamentale de toute architecture de transformeur. Les transformeurs sont des architectures d'apprentissage automatique génériques, simples et excitantes conçues pour traiter un ensemble d'unités connectées (pixels dans une image, etc.) où la seule interaction entre les unités se fait par auto-attention. <http://www.peterbloem.nl/blog/transformers>

## **Session 1 · Apprentissage profond | Apprentissage profond non supervisé**



---

Nous allons considérer une classe générale d'algorithmes d'apprentissage non supervisés qui peuvent être construits simplement en prédisant n'importe quelle partie des données à partir de n'importe quelle autre.

Selon Alexander Graves and Kelly Clancy, DeepMind : *"Our ability to learn about the world without explicit supervision (without labeled data) is fundamental to what we regard as intelligence."* — Alexander Graves and Kelly Clancy, DeepMind

BERT <https://arxiv.org/abs/1810.04805> (Bidirectional Encoder Representations from Transformers), qui a introduit une nouvelle ère en traitement du langage naturel, est pré-entraîné par apprentissage auto-supervisé à l'aide d'une quantité énorme de textes non-annotés sur le Web pour capturer les significations et relations sous-jacentes. Puis, le modèle pré-entraîné peut ensuite être affiné avec l'apprentissage supervisé, ce qui donne des résultats à la fine pointe de la technologie, sur diverses tâches en traitement de langage naturel NLP.

*"Ce que j'aime le plus, c'est à quel point l'idée est simple. Et que cet apprentissage non-supervisé commence vraiment à fonctionner."* — Ilya Sutskever, OpenAI

"Write With Transformer" permet la génération de texte sur des modèles Transformer polyvalents, tels que GPT, GPT-2 et XLNet. Par Hugging Face, 2019 : <https://transformer.huggingface.co>

Modèles génératifs

*"Ce que je ne peux pas créer, je ne comprends pas."* — Richard Feynman

Selon David Foster : *"Un modèle génératif décrit comment un jeu de données est généré, en termes d'un modèle probabiliste. En échantillonnant à partir de ce modèle, nous sommes en mesure de générer de nouvelles données."* — David Foster, Generative Deep Learning: Teaching Machines to Paint, Write, Compose, and Play

GAN: Génération d'échantillons. Comme le résume son concepteur, Ian Goodfellow:

Abstract: *"(...) entraîne simultanément deux modèles: un modèle génératif G qui capture la distribution des données et un discriminateur D qui estime la probabilité qu'un échantillon*

---

*proviennent des données d'apprentissage plutôt que de G. La procédure d'entraînement de G consiste à maximiser la probabilité que D commette une erreur. Ce cadre correspond à un jeu minimax à deux joueurs." Generative Adversarial Nets, Goodfellow et al. -*

<https://arxiv.org/pdf/1406.2661.pdf>

Se référant aux réseaux génératifs par antagonisme (GAN), Yann LeCun, directeur de la recherche en intelligence artificielle de Facebook, a dit à ce propos : « *Les GAN sont l'idée la plus intéressante (en AI) que j'ai vue depuis 10 ans !* »

Un nouveau jour est arrivé dans l'industrie de l'art. *"La question de savoir si une machine peut créer de l'art, ou autre chose, est au cœur de nombreux débats philosophiques depuis des décennies. Est-ce que cela vaut quelque chose sur le marché? Ce point a été définitivement réglé aujourd'hui puisqu'un portrait issu d'une image d'IA a été vendu pour près d'un demi-million de dollars aux enchères."* - Devin Coldewey, TechCrunch

Voici le code: art-DCGAN est une implémentation modifiée de DCGAN ( <https://github.com/robbiebarrat/art-dcgan> ). L'implémentation par Soumith Chintala ( <https://github.com/soumith/dcgan.torch> ) de DCGAN ( <http://arxiv.org/abs/1511.06434> ), avec un accent mis sur la génération d'oeuvres d'art.

GAN en Keras, par François Chollet :

<https://colab.research.google.com/drive/1CQ2XTMoUB7b9i9USUh4kp8BoCag1z-en>

Auto-encodeur: Estimation de la densité d'une distribution

Un auto-encodeur est un réseau de neurones entraîné pour tenter de reproduire en sortie les données d'entrée. Il comprend une couche cachée  $h$  qui décrit un code utilisé pour représenter l'entrée. La reconstruction permet un apprentissage non-supervisé (pas d'étiquettes!).

Les auto-encodeurs apprennent des représentations plus compactes des données d'apprentissage dans un espace vectoriel de variables latentes. Les variables latentes sont échantillonnées dans un espace latent et peuvent être décodées en valeurs observables. De plus, des opérations sémantiques peuvent être effectuées dans l'espace latent à l'aide de l'arithmétique vectorielle.

---

Les propriétés souhaitables d'un espace latent peuvent être résumées comme suit: Expression, réalisme et harmonie. Les autoencodeurs variationnels sont une variante probabiliste des autoencodeurs! On échantillonne de la moyenne et de l'écart type pour calculer l'échantillon latent.

Les propriétés souhaitables d'un espace latent sont similaires à la palette d'un artiste sur laquelle il peut explorer et mélanger les options de couleur d'une peinture.

MusicVAE est un auto-encodeur variationnel pour l'apprentissage d'espaces latents pour les partitions. Échantillonnez et interpolez dans un Colab Notebook:

[https://colab.research.google.com/notebook#fileId=v2/external/notebooks/magenta/music\\_vae/music\\_vae.ipynb](https://colab.research.google.com/notebook#fileId=v2/external/notebooks/magenta/music_vae/music_vae.ipynb)

Idéalement, nous voulons des variables latentes non corrélées les unes avec les autres. Les représentations désenchevêtrées (disentangled representations) sont définies comme celles dans lesquelles un changement dans une unité de la représentation correspond à un changement dans un facteur de variation unique des données tout en restant invariant des autres (Bengio et al. (2013).

Rassemblons tout ce que nous venons de voir.

Dans le réseau multi-objets (MONet, Burgess et al.: <https://arxiv.org/abs/1901.11390>), deux formes d'innéité qui permettent l'apprentissage (un autoencodeur variationnel désenchevêtré et un réseau d'attention récurrent) sont entraînés ensemble de bout en bout -- de manière purement non supervisée -- pour découvrir des représentations internes riches et utiles. De telles représentations peuvent simplifier le raisonnement des agents autonomes.

- <https://arxiv.org/abs/1809.11096>
- <https://nvlabs.github.io/stylegan2/versions.html>
- <http://www.whichfaceisreal.com/>
- <https://generated.photos/>
- <https://arxiv.org/abs/1606.05579>

## Session 2 · Les agents autonomes

---

Les agents autonomes ont commencé à obtenir des RÉSULTATS SUPERHUMAINS.

Dans la Session 1 · Apprentissage profond, nous avons vu que l'apprentissage profond est une procédure d'apprentissage général et qu'un pourcentage significatif des percées de l'apprentissage profond provient de constructions réutilisables et du partage de paramètres. Ainsi, les couches CNN réutilisent les poids à plusieurs endroits (partage de paramètres dans l'espace); les couches RNN réutilisent les poids pour plusieurs étapes (partage de paramètres dans le temps).

Nous avons également considéré une classe générale d'algorithmes d'apprentissage non supervisés qui peuvent être construits simplement en prédisant n'importe quelle partie des données à partir de n'importe quelle autre. Les transformeurs sont des architectures génériques, intéressants et puissants conçus pour traiter un ensemble connecté d'unités (une entité dans une séquence (temps), de pixels dans une image (espace), etc).

L'apprentissage profond est donc un outil puissant pour découvrir des représentations du monde de haut niveau à partir de données non structurées.

Avec la "Session 2 · Agents autonomes" nous présenterons des agents qui puisent des enseignements de leurs expériences pour atteindre des objectifs dans un large éventail d'environnements.

Agent intelligent: tout système qui perçoit son environnement et pose des actions qui maximisent ses chances de succès à atteindre son objectif

Nous allons voir:

- Les stratégies d'évolution
- L'apprentissage par renforcement
- L'apprentissage contre soi (Self Play)
- Méta-apprentissage profond

*"L'intelligence mesure la capacité d'un agent à atteindre des objectifs dans un large éventail d'environnements."* — Shane Legg

- <https://github.com/nottombrown/rl-teacher>

- 
- <https://tossingbot.cs.princeton.edu/>
  - <https://openai.com/blog/reinforcement-learning-with-prediction-based-rewards/>
  - <https://openai.com/blog/learning-dexterity/>
  - <https://cdorgs.github.io/>

## Session 2 · Agents autonomes | Stratégies d'évolution

L'évolution et les réseaux de neurones se sont révélés être une combinaison puissante dans la nature. — Stanley et al.

Dans sa conférence de remise du prix Nobel de chimie de 2018 "Innovation by Evolution : Bringing New Chemistry to Life" (conférence du prix Nobel)†, la professeure Frances H. Arnold a déclaré :

*"La nature... a inventé la vie qui s'épanouit depuis des milliards d'années. (...) Le processus par lequel la nature a créé (...) tout le reste dans le monde biologique est tout aussi impressionnant. Ce processus est l'évolution, la grande machine génératrice de diversité qui a conçu toute la vie sur terre, il y a plus de trois milliards d'années. (...) l'évolution exécute un algorithme simple de diversification et de sélection naturelle, un algorithme qui fonctionne à tous les niveaux de complexité, des molécules protéiques aux écosystèmes entiers."* - Le Prix Nobel de Chimie de 2018

Comme l'a remarqué Wojciech Zaremba : *"L'évolution est un algorithme d'apprentissage lent qui, avec une quantité suffisante de calculs, produit un cerveau humain."* — Wojciech Zaremba

Il est remarquable de considérer que le cerveau humain doté de 100 trilliards de connexions est un produit de l'évolution, un processus naturel dépourvu de surveillance ou de prévoyance intelligente.

Comme Lehman et al. a souligné dans (Lehman, J. & Stanley, K. O. Abandoning objectives: evolution through the search for novelty alone. *Evol. Comput.* 19, 189–223 (2011)), la diversité est le produit phare de l'évolution.

La combinaison de la neuroévolution avec l'apprentissage profond et l'apprentissage par renforcement devrait permettre de nombreuses avancées et peut-être même de concevoir

---

une intelligence artificielle de niveau humain (et des algorithmes qui créent sans fin: des algorithmes ('open-ended') ouverts).

La stratégie d'évolution naturelle est un processus d'estimation et de vérification, dans lequel nous commençons par quelques paramètres aléatoires, puis à plusieurs reprises : (1) Nous ajustons l'estimation par hasard, et (2) Nous modifions légèrement notre estimation vers les ajustements qui ont fonctionné le mieux. Ce processus d'estimation et de vérification (qui est essentiellement la recherche et l'apprentissage) est une méthode d'application générale qui continue à évoluer avec l'augmentation de la puissance de calcul, même si la puissance de calcul disponible devient très importante.

Salimans et son équipe ont montré que la stratégie d'évolution naturelle (en modifiant directement le poids des réseaux de neurones) permet à un robot humanoïde contrôlé par un réseau de neurones à deux couches de marcher (robotique évolutive).

Selon Lehman : *"...l'évolution — qu'elle soit biologique ou informatique — est intrinsèquement créative et devrait de façon routinière nous surprendre, nous ravir et même nous déjouer."* Par Lehman, Clune, Misevic et al. La créativité surprenante de l'évolution numérique: recueil d'anecdotes issues des communautés de recherche en calcul évolutif et en vie artificielle <https://arxiv.org/pdf/1803.03453.pdf>

POET (Paired Open-Ended Trailblazer) est conçu pour inventer en permanence des problèmes de plus en plus complexes et divers, ainsi que leurs solutions correspondantes. <https://arxiv.org/abs/2003.08536> | <http://github.com/uber-research/poet>

Selon Kenneth et al. : *"Alors que la neuroévolution rencontre le potentiel du calcul croissant disponible aujourd'hui, ce domaine pourrait s'avérer un outil essentiel dans la poursuite à long terme de l'intelligence artificielle générale."* — Designing Neural Networks Through Neuroevolution. Kenneth O. Stanley, Jeff Clune, Joel Lehman & Risto Miikkulainen. Nature Machine Intelligence VOL 1, 24 – 35 (January 2019) <https://www.nature.com/articles/s42256-018-0006-z.pdf>

- <https://openai.com/blog/evolution-strategies/>
- <https://colab.research.google.com/github/karpathy/randomfun/blob/master/es.ipynb>

---

## Session 2 · Agents autonomes | Apprentissage par renforcement

Les algorithmes d'apprentissage par renforcement profond ont commencé à atteindre des résultats surhumains dans de nombreux environnements difficiles.

*"L'apprentissage par renforcement (RL) est le sous-domaine de l'apprentissage machine qui concerne la prise de décision et le contrôle moteur. Il étudie comment un agent peut apprendre à atteindre les objectifs dans un environnement complexe et incertain."* — OpenAI

Les stratégies d'évolution modifient directement les paramètres d'un réseau de neurones profond, alors que les algorithmes d'apprentissage par renforcement modifient les actions d'un agent!

Dans l'apprentissage par renforcement (RL), un agent interagit avec un environnement via un processus de décision de Markov (markovien) (MDP). A chaque étape, l'agent est dans un état donné et il choisit une action selon sa politique (possiblement stochastique). En conséquence, il reçoit une récompense de l'environnement et entre dans un nouvel état.

Considérons quelques définitions :

- Environnement: le monde dans lequel l'agent existe et opère.
- Action: une manœuvre que l'agent peut effectuer dans l'environnement.
- État: une situation perçue par l'agent.
- Récompense: la rétroaction qui mesure le succès ou l'échec de l'action de l'agent.

L'apprentissage par renforcement nécessite d'explorer l'environnement pour tenter de découvrir les actions optimales à entreprendre, notamment pour déterminer quelles actions mènent à des récompenses. On peut y arriver, en faisant ce qui suit:

- En suivant la politique (On-policy): "Je ne peux apprendre que de mes actions".
- En agissant hors politique (Off-policy): "Je peux apprendre de quiconque tente d'atteindre n'importe quel objectif".

---

Le réseau DQN (Deep Q-Network (DQN)) a été précurseur à la fin de l'année 2013 ... Dans "Contrôle au niveau humain par l'apprentissage par renforcement" (<https://storage.googleapis.com/deepmind-media/dqn/DQNNaturePaper.pdf>), Mnih et son équipe ont mis au point un agent (réseau DQN), capable de combiner l'apprentissage par renforcement avec une classe de réseaux de neurones artificiels appelés réseaux de neurones profonds. Ce travail représente la première démonstration d'un agent polyvalent capable d'adapter en permanence son comportement (à partir d'entrées sensorielles de grande dimension) sans aucune intervention humaine.

Une taxonomie des algorithmes en apprentissage par renforcement. Par Josh Achiam, chercheur scientifique de l'équipe de sûreté d'OpenAI : <https://spinningup.openai.com/>

Un agent apprentissage par renforcement peut essayer directement ou indirectement d'apprendre à partir de :

Premièrement, par une politique :

Politique: (fonction de comportement de l'agent) - Basé sur la politique

- Apprendre la fonction politique stochastique qui fait correspondre état à action
- Agir en échantillonnant une politique

Deuxièmement, par une fonction de valeur:

Fonction de valeur: (Comment chaque état et / ou action sont bons (good)) - Basé sur la valeur

- Apprendre la valeur d'un état ou la valeur d'une paire action-état (utilité)
- Agir en choisissant la meilleure (utilité) action dans un état

Troisièmement, par un modèle

Modèle: (Représentation que se fait l'agent de l'environnement) - Basé sur le modèle

- Apprendre le modèle du monde, puis planifier en utilisant ce modèle
- Mettre à jour le modèle souvent
- Re-planifier souvent



---

Basé sur la politique. Gradient de la politique (PG): entraînement

- Exécuter une politique pendant un moment
- Augmenter la probabilité des actions conduisant à des récompenses élevées
- Diminuer la probabilité d'actions entraînant peu ou pas de récompenses

Basé sur le modèle. Un agent est entraîné dans un modèle d'environnement appris, c'est-à-dire une hallucination. La politique apprise est ensuite transférée dans l'environnement réel et fonctionne bien. Les "modèles du monde" sont entraînés de manière non-supervisée pour apprendre une représentation spatio-temporelle comprimée de l'environnement. En utilisant les caractéristique extraites du modèle du monde comme entrées d'un agent, une politique compacte est alors formée pour résoudre la tâche requise. L'agent est entièrement entraîné à l'intérieur de son propre modèle du monde halluciné et transfère sa politique dans l'environnement réel. Modèles du monde <https://worldmodels.github.io>

Outils

Algorithmes RL Open Source. Par Sergey Kolesnikov.

[https://docs.google.com/spreadsheets/d/1EeFPd-XIQ3mq\\_9snTIAZSsFY7Hbnmd7P5bbT8LPuMn0/edit#gid=0](https://docs.google.com/spreadsheets/d/1EeFPd-XIQ3mq_9snTIAZSsFY7Hbnmd7P5bbT8LPuMn0/edit#gid=0)

Bases de références d'OpenAI (OpenAI Baselines) : Implémentations de haute qualité d'algorithmes d'apprentissage par renforcement. <https://github.com/openai/baselines>

TF-Agents est une bibliothèque pour l'apprentissage par renforcement dans TensorFlow. <https://github.com/tensorflow/agents>

Suite comportementale pour l'apprentissage par renforcement, Osband et al.

<https://arxiv.org/abs/1908.03568v2> | <http://github.com/deepmind/bsuite> | <https://colab.research.google.com/drive/1RYWJaMEHVeN8yl83QtL35GOSFQBRgLaX>

Vers une architecture d'un agent d'IA général : SuperDyna.

L'ambition : un agent d'IA général pour l'apprentissage par renforcement biologique artificiel. Selon Rich Sutton, au #NeurIPS 2019 :

- 
- Interagir avec le monde.
  - Apprendre de ce qui vient de se passer.
  - Planifier.
  - Découvrir.

Tel que résumé par Stanley et al. dans "Designing neural networks through neuroevolution" (Nature Machine Intelligence) : *"L'évolution ouverte peut nécessiter plus que des réseaux de neurones pour évoluer pleinement—les cerveaux et les corps évoluent ensemble dans la nature, et les morphologies peuvent elles aussi évoluer avec les réseaux de neurones dans les systèmes artificiels, en fournissant une forme d'incarnation."* Dans "Reinforcement Learning for Improving Agent Design", David Ha explore la possibilité d'apprendre une version de la conception de la structure physique de l'agent qui est mieux adaptée à sa tâche, conjointement avec la politique. Blog: <https://designrl.github.io> | Paper: <https://arxiv.org/abs/1810.03779> | Code: <https://github.com/hardmaru/astool/>

Dans "Chip Placement with Deep Reinforcement Learning", Mirhoseini et al. démontrent une approche basée sur l'apprentissage du placement des puces. La méthode peut générer des placements surhumains en moins de 6 heures, tandis que les bases de référence actuelles nécessitent des experts humains et prennent de nombreuses semaines. Mirhoseini et al.: <https://arxiv.org/abs/2004.10746>

- <https://arxiv.org/abs/1909.07528>

## Session 2 • Agents autonomes | Self Play

*"Le "Self-Play" est une création automatisée de connaissances."* — Carlos E. Perez

Lancé en 1959 par Arthur Lee Samuel dans son jeu de dames, le Self Play est une approche réussie dans laquelle un système (A) joue contre une copie (figée) de lui-même (B), et améliore progressivement sa politique en utilisant un algorithme d'apprentissage par renforcement. Lorsque A peut battre B de façon constante,  $B := A$  (opérateur d'assignation). Répéter.

Étapes importantes et chronologie :

- 
- Programme de jeu de dames d'Arthur Lee Samuel (fonction d'approximation utilisée pour l'apprentissage des fonctions de valeur). Samuel a suivi la suggestion de Shannon (1950) selon laquelle une fonction de valeur n'avait pas besoin d'être exacte pour être utile) (1959)
  - TD-Gammon ("TD-Gammon est un réseau de neurones qui s'entraîne à être une fonction d'évaluation pour le jeu de backgammon en jouant contre lui-même et en apprenant du résultat. - Gerald Tesauro (1992))  
<https://www.bkgm.com/articles/tesauro/tdl.html>
  - AlphaGo et AlphaZero

En 2017, Silver et al. a introduit AlphaGo Zero, un algorithme basé uniquement sur l'apprentissage par renforcement, sans données humaines, sans guidage ni connaissance du domaine au-delà des règles du jeu. En quelques jours, à partir de zéro, AlphaGo Zero a atteint des performances surhumaines, et a gagné 100-0 contre AlphaGo, vainqueur de champions précédemment publié.

Propriétés attrayantes de l'apprentissage contre soi-même (Self Play). Selon Ilya Sutskever (Source: MIT AI: OpenAI Meta-Learning and Self-Play (Ilya Sutskever)

[https://youtu.be/9EN\\_HoEk3KY](https://youtu.be/9EN_HoEk3KY)) :

- Environnement simple → stratégie extrêmement complexe.
- L'apprentissage contre soi-même permet de transformer les calculs en données!
- Curriculum parfait (l'apprentissage contre soi-même assure que l'environnement a le bon degré de difficulté pour qu'un agent IA s'améliore).

Comment passer de l'apprentissage contre soi-même (Self-Play) au monde réel ?

Apprentissage par transfert. Apprendre dans des simulations ('ENVIRONNEMENTS') sur de multiples problèmes via l'apprentissage contre soi-même et continuer à apprendre dans le monde réel. Aujourd'hui, l'apprentissage de plusieurs problèmes via Self-Play pourrait être accompli, pour les jeux par exemple, avec OpenSpiel : une collection d'environnements et d'algorithmes pour la recherche sur l'apprentissage par renforcement général et la planification dans les jeux.

"OpenSpiel: A Framework for Reinforcement Learning in Games" Lanctot et al.

<https://arxiv.org/pdf/1908.09453v4.pdf>

---

Vue d'ensemble des jeux mis en œuvre : Echecs, Go, Pentago, Hanabi, Tic-Tac-Toe.

Pour essayer OpenSpiel dans Google Colaboratory :

[https://colab.research.google.com/github/deepmind/open\\_spiel/blob/master/open\\_spiel/colabs/install\\_open\\_spiel.ipynb](https://colab.research.google.com/github/deepmind/open_spiel/blob/master/open_spiel/colabs/install_open_spiel.ipynb)

Le self-play reflète les mêmes aperçus que la coévolution.

- <https://deepmind.com/blog/article/alphago-zero-starting-scratch>

## Session 2 • Agents autonomes | Apprendre à apprendre

Méta-apprentissage = apprendre à apprendre (en résolvant de nombreuses tâches).

*"Un algorithme de méta-apprentissage prend en charge une distribution de tâches, où chaque tâche est un problème d'apprentissage, et il produit un apprenant rapide — un apprenant qui peut généraliser à partir d'un petit nombre d'exemples."* — Reptile: A Scalable Meta-Learning Algorithm. Alex Nichol & John Schulman. OpenAI.

On peut réduire le méta-apprentissage (learning-to-learn) à l'apprentissage profond conventionnel en posant: Cas d'entraînement = tâche d'entraînement.

- Entraîner un système à de nombreuses tâches
- Le système qui en résulte peut résoudre rapidement de nouvelles tâches

(Source: MIT AI: OpenAI Meta-Learning and Self-Play (Ilya Sutskever)

[https://youtu.be/9EN\\_HoEk3KY](https://youtu.be/9EN_HoEk3KY) )

Un aspect clé de l'intelligence est la versatilité – la capacité de faire beaucoup de choses différentes. Model-Agnostic Meta-Learning cherche à trouver un ensemble de paramètres  $\theta$  qui sont hautement adaptables en associant un apprenant (un réseau de neurones avec un ensemble de poids  $\theta$ ) à un algorithme d'apprentissage (comme la descente de gradient stochastique (SGD)) pour résoudre une tâche sélectionnée dans une distribution de tâches aussi rapidement que possible.

Tel qu'expliqué par Finn et al. (<https://arxiv.org/abs/1703.03400>) : Au cours du méta-apprentissage, méta-apprentissage modèle agnostique MAMA optimise pour un

---

ensemble de paramètres de telle sorte que lorsqu'un pas de gradient est effectué par rapport à une tâche particulière  $i$ , les paramètres sont proches des paramètres optimaux  $\theta^*_i$  pour la tâche  $i$ . Le méta-apprentissage (apprendre à apprendre) (Chelsea Finn and Sergey Levine. Meta-learning and universality: Deep representations and gradient descent can approximate any learning algorithm. arXiv preprint arXiv:1710.11622, 2017) est un tremplin clé vers des agents versatiles qui peuvent continuellement apprendre une grande variété de tâches tout au long de leurs vies.

Inspirés par les efforts précédents pour concevoir une interface unifiée entre les tâches, comme l'OpenAI Gym (Brockman et al., 2016) dans l'apprentissage par renforcement, Deleu et al., 2019, présentent Torchmeta. Torchmeta est une collection d'extensions et de chargeurs de données pour le méta-apprentissage. <https://arxiv.org/abs/1909.06576> | <https://github.com/tristandeleu/pytorch-meta>

Dans "AI-GAs: Algorithmes générateurs d'IA, un paradigme alternatif pour la production d'une intelligence artificielle générale" <https://arxiv.org/abs/1905.10985>, Jeff Clune décrit une voie passionnante qui pourrait ultimement réussir à produire une IA générale. Elle est basée sur la tendance évidente en apprentissage automatique selon laquelle les solutions conçues à la main sont éventuellement remplacées par des solutions apprises plus efficaces. L'idée est de créer un algorithme de génération d'IA (AI-GA), qui apprend automatiquement comment produire une IA générale.

Trois piliers sont essentiels pour cette approche:

1. méta-apprentissage des architectures,
2. méta-apprentissage des algorithmes d'apprentissage eux-mêmes, et
3. génération d'environnements d'apprentissage efficaces.

Darwin Complete (Darwin complet): un encodage environnemental qui permet la création de tout environnement d'apprentissage possible.

## Session 3 • Environnements

Pour Endsley,

---

*"La conscience d'une situation est la perception des éléments de l'environnement dans un volume de temps et d'espace, la compréhension de leur signification et la projection de leur statut dans un avenir proche." — Endsley (1987)*

La véritable compréhension provient d'agents autonomes qui apprennent en combinaison avec la manière dont ils affectent le monde.

L'OpenAI Gym est une ressource pour développer et comparer les algorithmes d'apprentissage par renforcement. Ce qui rend le gym si génial, c'est une interface de programmation applicative (application programming interface (API)) commune autour des environnements. <https://gym.openai.com>

*"En intégrant leur approche dans le cadre du populaire OpenAI Gym, les firmes de design peuvent créer des environnements plus réalistes – par exemple, intégrer la résistance des matériaux, les facteurs de sécurité, ..., et intégrer des algorithmes existants dans ce cadre afin d'optimiser également les aspects liés à la conception. tels que la consommation d'énergie, la facilité de fabrication ou la durabilité." — David Ha*

Voici quelques exemples:

- TensorTrade. <https://github.com/tensortrade-org/tensortrade>
- Learning Dexterity. <https://blog.openai.com/learning-dexterity/>
- Environnement d'assemblage de meubles IKEA pour les tâches de manipulation complexes à long terme. <https://clvrai.github.io/furniture/>

Créer de nouveaux environnements avec l'OpenAI Gym.

<https://github.com/openai/gym/blob/master/docs/creating-environments.md>

Comment créer de nouveaux environnements avec l'OpenAI Gym. Exemple minimal avec code et agent (CheatSheet <http://www.montreal.ai/ai4all.pdf> : OpenAI Gym (6.1)) :

1. Télécharger gym-foo  
<https://drive.google.com/file/d/1r2A8j9CjIjQNwss246gATeD0LLMzpUT-/view?usp=sharing>
2. `cd gym-foo`
3. `pip install -e .`

---

#### 4. python ES-foo.py

MMO neuronal : Pour des simulations complexes avec de nombreux agents à longue durée de vie. C'est un environnement de jeu massivement multi-agents qui supporte un grand nombre variable d'agents dans une tâche persistante et ouverte. Blog:

<https://blog.openai.com/neural-mmo/>

DeepMind Lab: Une plateforme 3D personnalisable pour la recherche en IA basée sur les agents. DeepMind Lab fournit une suite de tâches de navigation 3D difficiles et de résolution d'énigmes pour les agents intelligents. <https://github.com/deepmind/lab>

Les agents d'apprentissage machine Unity ( Unity Machine Learning Agents ) permettent de créer des environnements où des agents intelligents (agent unique, multi-agents coopératifs et compétitifs et écosystème) peuvent être entraînés en utilisant l'apprentissage par renforcement, la neuroévolution ou d'autres méthodes d'apprentissage machine <https://unity3d.ai> ("Unity Machine Learning Agents Toolkit" : <https://github.com/Unity-Technologies/ml-agents>). "Unity : A General Platform for Intelligent Agents". Juliani et al.: <https://arxiv.org/abs/1809.02627>

Un bon endroit pour commencer: les environnements de marathon pour les agents d'apprentissage automatique Unity :

<https://github.com/Unity-Technologies/marathon-envs> Blog par Joe Booth : <https://towardsdatascience.com/gettingstartedwithmarathonenvs-v0-5-0a-c1054a0b540c>

Arena : une plateforme générale d'évaluation et des outils pour créer une intelligence multi-agent Song et al., 2019 : <https://arxiv.org/abs/1905.08085>

Habitat IA permet l'entraînement d'agents d'intelligence artificielle incarnés (robots virtuels) dans un simulateur 3D hautement photoréaliste et efficace, avant de transférer les compétences acquises dans la réalité. Par Facebook AI Research: <https://aihabitat.org/>

Jetson Nano. Un petit ordinateur IA pour créer des systèmes intelligents

<https://www.nvidia.com/en-us/autonomous-machines/embedded-systems/jetson-nano/>

NVIDIA Jetson Nano de NVIDIA permet le développement de nouveaux petits systèmes IA à faible consommation. Il ouvre de nouveaux mondes d'applications pour l'internet des objets (IoT), y compris les robots domestiques.

---

Edge TPU – Dev Board. Plateforme de développement pour prototyper rapidement des produits d'apprentissage machine sur une machine locale :

<https://coral.ai/products/dev-board/> TensorFlow Lite models can be compiled to run on the Edge TPU. Les modèles TensorFlow Lite peuvent être compilés pour fonctionner sur le TPU Edge. Le coprocesseur Edge TPU embarqué est capable d'effectuer 4 trillions d'opérations (tétra-opérations) par seconde (TOPS).

## ALLOCUTION SPÉCIALE | Considérations éthiques

Macrostratégie IA : Et si nous réussissions (AGI) ?

Aligner l'IA sur les intérêts humains.

*"Je crois que la réponse est de trouver comment créer une intelligence artificielle superintelligente de sorte que même si - ou quand - elle se libère, elle reste sécuritaire parce qu'elle est fondamentalement de notre côté, car elle partage nos valeurs. Je ne vois pas d'autres façons de faire face à ce problème difficile." — Nick Bostrom*

*"(...) Dans le contexte sociétal, nous avons mis en place des lois pour empêcher les gens de faire de mauvaises actions (...). Concevoir des fonctions d'objectifs pour les gens est quelque chose que nous savons faire. (...) En effet, le Code civil s'appelle "code" et ça vous dit quelque chose. Et l'essence de ce qu'est le code juridique est de concevoir cette fonction d'objectif pour les justiciables. Ce code vous dit ce que vous pouvez faire, ce que vous ne pouvez pas faire et ce que vous devrez payer si vous le faites : c'est une fonction d'objectif. (...). C'est là que la science juridique et l'informatique se rejoignent." — Yann LeCun <https://youtu.be/SGSOCuByo24>*

De leur côté, Fjeld et son équipe ont proposé une carte des approches éthiques et de celles fondées sur le droit ("A Map of Ethical and Right-Based Approaches." Fjeld et al.: <https://ai-hr.cyber.harvard.edu/primp-viz.html> ).

Naturellement, en tant que créateurs de cette nouvelle technologie, il est de notre responsabilité collective de guider l'IA afin qu'elle ait un impact positif sur notre planète.

*"Les droits universels de l'Homme doivent être intégrés dans la conception des structures technologiques de l'IA." — Dr Eileen Donahoe*



- 
- <https://arxiv.org/abs/2004.13332>
  - <https://cyber.harvard.edu/publication/2020/principled-ai>

## Remarques de clôture

Nous sommes à l'aube de l'ère de l'intelligence artificielle.

Selon Bill Braun, CIO de Chevron:

*"C'est le printemps pour l'IA, et nous prévoyons un long été."* — Bill Braun, CIO of Chevron

Pour le reste du webinaire, Vincent Boucher, président de MONTRÉAL.IA, répondra (en direct) à vos questions en direct en ligne.

**LA FIN - MERCI..**

\*\* Le contenu de ce cours est destiné à votre usage personnel et ne doit pas être partagé ou diffusé.