Apprentissage de représentation profondes

Synthèse

Introduction

L'objectif de l'Intelligence Artificielle (IA) est d'effectuer une tâche qui exige de l'intelligence grâce à un système informatique.

Ces tâches se répartissent dans de nombreux domaines:

- Jeux: échecs, le go, jeux de stratégie
- Langage: traduction, résumé, extraction de contenu
- Vision: classification, Segmentation, recherche d'images
- Et d'autres: Régression, décision, analyse de risques

Pour certaines tâches comme les échecs, les performances de l'ordinateur dépassent maintenant celles de l'être humain, tandis que pour d'autres tâches, il reste encore à les approcher. En particulier, de nombreux problèmes résolus facilement par des êtres humains dans les domaines de la vision et du langage se révèlent très difficile à résoudre en utilisant des algorithmes. Bien que le terme IA définit le type de problème, il ne fait pas référence à une méthode spécifique pour le résoudre. Une grande variété d'approches ont émergé au fil des ans, mais aucune n'a réussi à créer une intelligence artificielle générale: une IA avec la même capacité de raisonnement que l'esprit humain.

Considérons maintenant un système d'intelligence artificielle comme décrit ci-dessus. Un tel système devra effectuer des tâches, prendre des décisions et faire des choix. Toutes ces actions s'additionnent pour constituer ce qu'on pourrait appeler le comportement du système. L'apprentissage artificiel est basé sur la réalisation importante qu'un comportement intelligent est trop complexe pour être simplement «programmé». Au lieu de cela, le système va apprendre son comportement à partir de données. Un système doté de cette capacité à apprendre une sorte de comportement intelligent est simplement appelé un modèle, et le processus par lequel nous entraînons un modèle à partir de données est appelé un algorithme d'apprentissage automatique.

Dans la pratique, un modèle est défini par une équation ou un algorithme qui, avant l'apprentissage, dispose d'un ensemble de variables indéterminées appelées paramètres. Au cours de la procédure d'apprentissage, les données sont utilisées pour choisir les valeurs des paramètres qui maximisent la capacité du modèle pour effectuer la tâche voulue. Cette «capacité à exécuter» est mesurée par ce qu'on appelle une fonction cible ou fonction objectif. Pour ce faire, nous nous tournons vers l'optimisation qui est une branche des mathématiques consistant en l'étude de la façon de choisir les paramètres pour optimiser une fonction objectif. Un problème d'apprentissage artificiel peut alors être posé comme un problème d'optimisation où l'objectif est de maximiser une certaine mesure de la performance par rapport à une tâche finale. De toute évidence, l'optimisation est utile pour l'apprentissage artificiel, mais comme nous le verrons, l'apprentissage peut aussi être utile pour l'optimisation. En substance, les algorithmes d'apprentissage peuvent être utilisés pour apprendre le paysage d'une fonction objectif, et ainsi faciliter la recherche de paramètres appropriés.

En ce qui concerne les données à partir desquelles apprendre, elles doivent être informatives pour la tâche cible. Par exemple, dans le domaine de l'apprentissage supervisé, l'objectif est d'apprendre un modèle, étant donné des exemples de ce que le modèle devrait faire dans plusieurs situations. Les données consistent alors en une série d'exemples (stimulus x → réponse y) qui décrivent comment le système devrait idéalement répondre à plusieurs stimuli d'entrée. L'apprentissage supervisé

possède un grand nombre d'applications telles que:

- Prédiction: étant donné une observation x au temps t, quelle sont les observations probables à l'instant t + 1.
- Jeux: étant donné l'état d'un plateau de jeu x, quelle est la meilleure décision y.
- Moteurs de recherche: étant donnée une requête x, quels est le résultat y le plus pertinent.
- Reconnaissance de formes: par exemple dans le cas de caractères manuscrits, étant donné les pixels d'un code postal numérisé x, quel est le code postal y.

Un problème se pose rapidement lorsque la relation entre x et y n'est pas évidente. Cela conduit les praticiens à utiliser une étape de prétraitement dans laquelle un expert doit trouver un ensemble de caractéristiques f(x) tels que l'apprentissage de la relation entre f(x) et y devient plus simple. Cependant, trouver des bonnes caractéristiques est coûteux car cela nécessitent des connaissances spécialisées, souvent acquises après des années d'expérimentation.

Dans l'apprentissage de représentations, un algorithme d'apprentissage est utilisée pour trouver des caractéristiques intéressantes des données. Apprendre des caractéristiques, au lieu d'utiliser une étape de prétraitement comporte de nombreux avantages car cela rend toute l'approche moins dépendante de l'humain, et donc plus générale. Même si cela peut sembler difficile, l'apprentissage de représentations utiles peut être fait en pratique avec de l'apprentissage non supervisé où l'apprentissage se fait sur un ensemble de données d'entraînement x sans réponses y correspondantes. L'apprentissage non supervisé comprend des tâches telles que:

- Le groupage, où l'objectif est de regrouper des observations similaires.
- La compression ou réduction de dimensionnalité, où l'objectif est d'apprendre une représentation de plus petite taille que l'entrée.
- L'estimation de densité, où l'objectif est de trouver une distribution de probabilité qui est susceptible d'avoir généré le jeu de données.

Un aspect particulièrement important pour cette thèse est la possibilité de considérer plusieurs couches de traitement, à savoir une architecture profonde. Bien que l'apprentissage des architectures profondes soulève d'importantes questions computationnelles, le principe a été appliqué avec succès au cours des dernières années en utilisant une approche couche par couche : en essayant d'apprendre les caractéristiques d'une couche à la fois au lieu d'essayer d'apprendre toutes les couches en même temps. L'approche peut être résumée comme suit: Premièrement on entraîne un ensemble de fonctions f1(x) afin de mieux représenter l'entrée x. On peut ensuite considérer l'apprentissage des caractéristiques de niveau supérieur f2(f1(x)) à partir de f1 (x). Dans ce cadre, les caractéristiques f1 sont entraînées pour expliquer les données x et fk+1 est entraîné pour expliquer les données représentées par les caractéristiques fk.

Dans cet esprit, l'apprentissage de représentations profondes se réfère au problème de l'apprentissage de multiples couches de caractéristiques intéressantes pour un ensemble de données.

Cette thèse est composée de trois parties:

Partie I: optimisation Cette partir commence par une définition de ce qui constitue un problème d'optimisation et décrit plusieurs approches pour trouver une solution. Elle décrit ensuite comment les problèmes d'apprentissage artificiel peuvent être posées comme des problèmes d'optimisation. Enfin, nous présentons le point de vue probabiliste sur l'apprentissage artificiel qui a gagné beaucoup d'attention ces dernières années.

Partie II: deep learning Cette partie commence avec une présentation des réseaux de neurones qui sont à l'heure actuelle le moyen le plus efficace pour l'apprentissage en profondeur. Elle décrit ensuite comment les réseaux de neurones peuvent être utilisés dans le contexte de l'apprentissage de représentations profondes. Enfin, la partie se termine par un examen des progrès récents dans

l'apprentissage profond, en introduisant des questions qui motivent les contributions de l'auteur

Partie III: contributions contient les trois principales publications de l'auteur, replacées dans leur contexte et commentées pour préciser leur contribution à la lumière des questions qui sont présentées dans la Partie II.

Enfin, la thèse se termine par un résumé des contributions de l'auteur, une analyse de leur impact sur la compréhension actuelle du paradigme de l'apprentissage en profondeur, et de nouvelles perspectives de recherche qui découlent de l'œuvre accomplie.

Questions

L'apprentissage profond et plus précisément l'apprentissage couche par couche en profondeur, a conduit à de nombreux résultats impressionnants dans plusieurs contextes. Néanmoins, dans cette thèse, nous devons tenir compte de ce qui pourrait être amélioré.

Les RBM empilées semblent avoir la meilleure justification pour l'apprentissage couche par couche jusqu'ici. Cependant la maximisation d'une borne variationnelle ne semble pas se traduire par la maximisation de la vraisemblance d'un modèle génératif profond complet puisque i/ la première couche n'est pas entraînée pour maximiser la log-vraisemblance du modèle profond, et ii/ la garantie d'amélioration pour les couches supérieures ne tient pas pour plus de deux couches. Néanmoins, le principe de l'entraînement couche par couche a d'énormes avantages en théorie ce qui nous amène à poser la question:

Question 1: Est-il possible d'apprendre une couche inférieure optimale avant d'apprendre les couches supérieures ?

Si la réponse est oui, alors il existe aussi un critère valide pour l'entraînement couche par couche. Dans ce cas, nous aimerions savoir :

Question 2: Si il est possible d'apprendre une couche inférieure optimale avant d'apprendre les couches supérieures, quel est le critère à optimiser à chaque étape ?

Si la réponse à la question 1 est non, alors, toute méthode couche par couche rencontrera des problèmes tels que ceux rencontrés dans la maximisation de la borne inférieure variationnelle et aura probablement des difficultés lors du passage à de nombreuses couches.

Un autre point concerne la généralisation de l'entraînement couche par couche de modèles autres que les RBMs et machines de Boltzmann profondes. Dans le cas des auto-encodeurs profonds, la justification actuelle semble être que les auto-associateurs apprennent une approximation des RBM et que l'entraînement d'auto-encodeurs profonds peut donc maximiser une approximation de la borne inférieure variationnelle. Cela a conduit à ce que les RBMs et les machines de Boltzmann profondes soient les premiers choix envisagés pour répondre à un problème d'apprentissage profond. Néanmoins, une étape finale de réglage fin avec la rétro-propagation est souvent considérée dans les applications pratiques que ce soit avec des auto-associateurs ou des RBM empilées. Cela nous amène à examiner la possibilité que les auto-encodeurs sont en fait très adaptés à l'apprentissage des réseaux de neurones profonds et à demander ce qui pourrait justifier cette performance.

Question 3: Y a t-il une justification pour l'apprentissage couche par couche en profondeur qui s'applique directement à d'autres modèles que les RBM et les machines de Boltzmann profondes ?

ou plus précisément dans le cas de empilés auto-associateurs et du réglage fin:

Question 4: Pourquoi l'entraînement couche par couche de RBM empilées et d'auto-associateurs résulte t-il en un apprentissage profond ?

Question 5: Comment pouvons-nous justifier la mise au point non supervisée de modèles

probabiliste avec rétro-propagation ?

Les formulations probabilistes ont conduit à une meilleure compréhension de l'apprentissage en profondeur, et l'évaluation des modèles génératifs profonds devrait idéalement être faite par rapport à la log-vraisemblance du modèle profond. Malheureusement celle ci est incalculable en pratique.

Question 6: Pouvons-nous trouver une mesure de performance calculable pour les modèles génératifs profonds ?

La mesure de la performance peut être utile à la fin de l'entraînement afin d'évaluer si l'entraînement est couronné de succès, mais aussi lors de l'entraînement lui même dans le cadre d'une procédure de sélection de modèle. Pourvu que nous ayons un bon critère pour l'entraînement couche par couche d'un modèle profond, la sélection de modèle pourrait également être faite avec un critère couche par couche en théorie. Cela a le potentiel de réduire considérablement les coûts de calcul actuellement encourus lorsque les modèles ne sont comparées qu'après que toutes leurs couches ont été entraînées.

En outre, être capable de mesurer la performance à chaque niveau donnerait une mesure empirique de ce qui est gagné à chaque fois qu'une couche est ajoutée et pourrait être utilisé pour décider d'arrêter d'ajouter des couches.

Question 7: Peut-on évaluer des architectures profondes et effectuer une sélection de modèle couche par couche ?

Question 8: Est-ce que cela a conduit à un critère pour arrêter l'ajout de couches ?

Enfin, la formation de chaque couche est un problème d'optimisation difficile qui implique généralement des millions de paramètres ou plus. L'impact des métriques et de la paramétrisation sur les stratégies d'optimisation telles que la descente de gradient est bien connu, donc nous demandons:

Question 9: Quel est l'impact des mesures sur l'optimisation de chaque couche ?

Par ailleurs, s'il y a un impact, nous serions très intéressés à améliorer les méthodes d'optimisation actuelles pour réduire les temps de calcul. Cela nous amène à notre dernière question, à savoir:

Question 10: Comment pouvons-nous améliorer l'optimisation de la couche en tenant compte de l'impact des métriques et de la paramétrisation ?

Ayant posé ces questions, nous passons maintenant aux contributions qui tentent de donner quelques réponses.

Contributions

Premier papier

Ludovic Arnold, Hélène Paugam-Moisy, and Michèle Sebag. Unsupervised layer- wise model selection in deep neural networks. In *19th European Conference on Artificial Intelligence (ECAI 2010)*, Lisbon Portugal, August 2010. 915–920

Contexte

Dans l'apprentissage profond, la sélection de modèle pose un problème difficile parce que le grand nombre de paramètres par couche est ensuite multiplié par le nombre de couches, augmentant ainsi la taille de l'espace de recherche de manière exponentielle: un exemple de la malédiction de la dimensionnalité.

Lors de l'examen des réseaux de neurones profonds génératifs comme les RBM empilées ou les DBN, la perspective probabiliste a l'avantage de fournir un cadre théorique complet. Toutefois, elle

suggère l'utilisation de la log-vraisemblance qui est incalculable pour mesurer la performance du problème. Ce problème de sélection de modèle et de l'évaluation de la performance est en fait beaucoup plus important qu'il n'y paraît. Si nous sommes capables d'identifier un bon critère d'évaluation, tel que de meilleurs modèles en fonction de ce critère sont de meilleurs modèles pour une tâche cible, nous avons trouvé non seulement une mesure de performance, mais aussi un critère d'entraînement que nous devons maximiser pendant la phase l'entraînement. La sélection des hyper-paramètre peut être considérée comme une partie intégrante du processus d'entraînement, visant à maximiser le critère d'entraînement sur un espace de recherche plus large qui prend en compte les hyper-paramètres.

Dans le prolongement de cette ligne de raisonnement, nous arrivons naturellement à poser la question suivante: Si les RBM empilées sont entraînées à l'aide d'un critère couche par couche, pourquoi ne pas les évaluer en utilisant le même genre de critère couche par couche ? C'est l'objet de ce premier papier.

Contributions

Parce que les réseaux de neurones profonds sont formés de manière couche par couche, nous proposons de les évaluer d'une manière similaire, à savoir couche par couche: former plusieurs couches possibles à chaque étape par exemple avec divers hyper-paramètres et choisir la meilleure avant de passer aux couches suivantes. Cette stratégie, en cas de succès permet de réduire l'espace de recherche d'exponentiel à linéaire en terme du nombre de couches.

Pour évaluer cette approche, il faut la comparer à l'approche alternative considéré incalculable: évaluation de chaque modèle profond après qu'il ait été complètement entraîné. Cela implique un coût de calcul très élevé qui nous amène à considérer une portée plus limitée de l'étude. À cet égard, la détermination de la topologie optimale, c'est à dire choisir le nombre de couches et le nombre de neurones par couche est un sous-problème intéressant qui nous permet de tester notre hypothèse. Le critère couche par couche que nous proposons pour évaluer les couches dans les réseaux de neurones profonds est l'erreur de reconstruction, un critère naturel pour les auto-associateurs. Comme les auto-encodeurs profonds peuvent être interprétés dans une perspective probabiliste, l'erreur de reconstruction doit être significative pour l'évaluation des modèles génératifs profonds.

Bien que les résultats confirment notre hypothèse, le critère de champ moyen que nous avons utilisé pour former des RBM est une approximation proche du critère d'entraînement des auto-associateurs ce qui peut limiter la généralité de l'approche. Néanmoins, l'étude aboutit à plusieurs résultats intéressants. Tout d'abord, la sélection couche par couche du nombre de neurones avec l'erreur de reconstruction est réussie, donnant une substance aux avantages potentiels revendiqués de l'approche. Deuxièmement, les résultats montrent que les couches supérieures ne peuvent pas récupérer les pertes résultant d'un nombre insuffisant de neurones dans les couches inférieures. Ceci est cohérent avec l'interprétation de chaque couche comme codant l'information en termes de concepts explicatifs de la couche d'en dessous, avec les couches supérieures codant des concepts d'ordre supérieur que les couches inférieures. Dans cette analogie, la compréhension d'un concept est conditionnelle à la compréhension des concepts d'ordre inférieur.

Discussion

Cet article, le premier de l'auteur jamais publié, est naturellement d'une qualité moindre que son travail ultérieur. Il n'en est pas moins un "premier pas" nécessaire dans la compréhension de la formation couche par couche et de l'évaluation des architectures profondes.

Un premier résultat concerne la possibilité d'effectuer la sélection du modèle couche par couche (question 7). Bien que la généralité de l'approche peut être mise en doute en raison de la procédure d'entraînement par champ moyen, l'approche soutient l'hypothèse que l'évaluation couche par couche avec l'erreur de reconstruction est possible et considérablement moins coûteuse que

l'évaluation des réseaux après la formation de toutes les couches.

Une conclusion encore plus intéressante concerne la possibilité d'entraîner des couches inférieures optimales (Questions 1 et 2). Notamment, il semble qu'en pratique la notion de « meilleure couche inférieure » est bien définie, indépendamment des couches suivantes. Pour être plus précis, même si l'erreur de reconstruction d'une couche inférieure ne peut pas être utilisée comme un gage de performance pour l'ensemble du réseau en raison des couches suivantes qui peuvent être mal entraînées, elle semble capable de maximiser la performance éventuelle du futur réseau.

Cela va s'avérer être une idée importante pour le prochain article qui donne une justification théorique et empirique globale pour un nouveau critère proche de l'erreur de reconstruction qui peut être utilisé pour l'entraînement couche par couche d'architectures profondes.

Deuxième papier

L. Arnold and Y. Ollivier. Layer-wise learning of deep generative models. Tech- nical report, ArXiv e-prints, December 2012. URL http://arxiv.org/abs/ 1212.1524 (soumis à publication, février 2013).

Contexte

Tout comme dans le précédent article, nous posons la question de l'évaluation couche par couche et de la sélection de modèle. Toutefois, la principale préoccupation est la justification de l'entraînement couche par couche pour les architectures profondes.

Les méthodes d'entraînement pour les architectures profondes tombent généralement dans l'une des catégories suivantes:

Méthode 1: Maximiser la performance dans un cadre supervisé par exemple avec des réseaux de neurones multicouches. Généralement considéré comme insoluble lorsque le nombre de couches est trop grand, mais récemment montré comme possible (Hinton et al, 2012;. Bengio et Glorot 2010;. Ciresan et al, 2010). Très efficace dans les domaines où il est possible de coder des invariances dans le modèle, par exemple avec des réseaux de neurones convolutionnels (LeCun et Ben-gio, 1995). Cette méthode pourrait sans doute être utilisée pour des tailles de modèles arbitraires, mais en pratique, on peut s'attendre à ce que cela pose un problème d'optimisation de plus en plus difficile.

Méthode 2: Maximiser la vraisemblance d'un modèle génératif profond. Cette approche devrait conduire à des variables latentes optimales, mais est incalculable et doit donc être approchée.

Méthode 3: Maximiser une borne inférieure variationnelle de la log-vraisemblance d'un modèle génératif profond. Cette méthode est appliquée pour le pré-entraînement de presque tous les modèles génératifs profonds jusqu'ici: RBM empilées (Hinton et al, 2006; Bengio et al, 2007), auto-associateurs empilés (Vincent et al, 2008), car ils sont une approximation des RBM (Bengio et Delalleau, 2009), machines de Boltzmann profondes (Salakhutdinov et Hinton, 2009a) et autres variations. Très efficace dans la pratique. Applicable en théorie à n'importe quel modèle probabiliste avec variable latentes qui ne sont pas indépendantes. C'est en fait une approximation de la méthode 2 (maximisation de la vraisemblance d'un modèle génératif profond) avec une borne inférieure variationnelle. Cette méthode bénéficie d'une garantie théorique que l'ajout d'une couche au-dessus d'un modèle à une couche ne peut qu'augmenter la probabilité du modèle profond. Toutefois, la garantie n'est pas généralisable à plus de deux couches.

La méthode 2 étant incalculable, une question essentielle de l'apprentissage profond est de savoir si l'apprentissage couche par couche qui est le principe derrière la méthode 3 est mathématiquement justifié, à savoir: Une estimation du maximum de vraisemblance d'un modèle génératif profond peut-elle être trouvée en maximisant un critère couche par couche ?

Pour que cela soit possible, il faut un critère couche par couche qui représente une certaine notion

d'optimalité pour une couche, ceci avant l'entraînement des couches suivantes. Si un tel critère n'existe pas, alors les couches inférieures optimales dépendent toujours du choix des couches supérieures et l'apprentissage couche par couche est fondamentalement problématique.

Contributions

Dans l'article qui suit, nous proposons un critère qui répond aux exigences ci-dessus et peut en théorie être appliqué pour trouver une solution optimale à un problème d'apprentissage profond couche par couche.

Ce critère: la borne supérieure de la meilleure marginale latente (Best Latent Marginal upper bound), vient avec une garantie d'optimalité pour les couches inférieures (Théorème 1) à condition que le reste de l'entraînement se passe bien. A savoir, la borne supérieure BLM représente le maximum pouvant être atteint pour la log-vraisemblance avec une couche inférieure donnée, tandis que la partie supérieure du modèle génératif profond n'est pas spécifiée. En supposant que la borne supérieure de la BLM a été maximisée avec succès, le problème est alors transféré aux couches supérieures d'une manière similaire à la méthode 3 présentée ci-dessus.

Lorsque la formation des couches supérieures est imparfaite, comme cela peut être le cas en pratique, l'erreur globale admet une borne supérieure qui est exactement le critère optimisé pour les couches supérieures. Un résultat inattendu est l'importance d'avoir des paramètres différents pour les parties génératives et l'inférence du modèle, et de permettre à ce dernier d'être aussi riche que possible. Cette approche a une relation étroite avec les RBM empilées et les auto-encodeurs. Surtout, la borne supérieure de la BLM fournit une justification probabiliste pour empiler des auto-associateurs.

Les expériences sur les deux bases de données profondes distinctes confirment la faisabilité de l'approche pour l'apprentissage couche par couche des modèles génératifs profonds et pour la sélection de modèle couche par couche.

Discussion

Ce document répond à la plupart des questions présentées dans le chapitre 6: Comment entraîner des couches inférieures optimales ? Avec quel critère ? Comment effectuer une sélection de modèle couche par couche ? Comment faire pour arrêter l'ajout de couches?

Tout d'abord, nous montrons qu'il est possible d'apprendre une couche inférieure optimale avant d'apprendre les couches supérieures (Question 1), en ce sens que nous pouvons optimiser pour une seule couche la capacité maximale du modèle quand les couches suivantes ne sont pas connues. Le critère à optimiser à chaque étape (Question 2) est alors la borne supérieure de la BLM. Quant à la question de la généralité (Question 3), la borne supérieure de la BLM s'applique à n'importe quel modèle génératif, qui représente la probabilité d'une entrée x avec des modèles distincts pour p(x|h) et p(h). Dans ce cadre, la borne supérieure de la BLM est le critère à optimiser par rapport aux paramètres de p(x|h), et la log-vraisemblance de l'ensemble de données transformé par une distribution d'inférence *qcond*(h|x) est le critère à optimiser par rapport aux paramètres de p(h). Dans le cas des auto-associateurs et du réglage fin non supervisé (questions 4 et 5), la maximisation de l'erreur de reconstruction peut être considéré comme une maximisation d'une borne inférieure de la borne supérieure de la BLM où chaque exemple correspond à une seule représentation cachée.

Cela donne une justification pour l'empilement d'auto-associateurs, pour le réglage fin non supervisé des auto-encodeurs profonds et pour les modèles de codage clairsemé (Kavukcuoglu et al., 2010a), comme correspondant à l'apprentissage d'un modèle génératif profond partiel où la partie supérieure n'est pas spécifiée.

Ce papier donne également une approche possible pour l'estimation de la log-vraisemblance des modèles génératifs profonds (question 6). Bien que la borne supérieure de la BLM n'est valide que

pour estimer la partie inférieure d'un modèle génératif profond, dans le cas où la couche supérieure est effectivement capable d'apprendre sa distribution cible de manière efficace, la borne supérieure de la BLM donne une très bonne approximation de la log-vraisemblance du modèle complet. Étant un bon critère pour évaluer les couches inférieures, la borne supérieure de la BLM peut également être utilisée pour effectuer une sélection de modèle couche par couche (question 7). L'évaluation de la dernière couche peut alors être abordée en évaluant le modèle par rapport à la vraisemblance pour sa distribution cible qui peut être approchée de manière fiable, car il ne concerne qu'une seule couche cachée. La borne supérieure de la BLM donne également un critère pour arrêter l'ajout de couches (question 8). À savoir, si la borne supérieure de la BLM n'est pas supérieure à la performance réalisée avec un modèle à une couche, ce modèle à une couche peut être choisi comme la dernière couche avec une garantie qu'ajouter des couches ne peut pas mener à de meilleures performances.

Cette étude a conduit également à de nombreuses questions. Tout d'abord, l'évaluation empirique de l'approche pose un sérieux problème parce que le but est de maximiser (même approximativement) une quantité incalculable: la log-vraisemblance d'un modèle génératif profond. Cela rend difficile la comparaison de deux approches concurrentes alors de la mesure de performance ne peut pas elle-même être facilement approximée. Dans ce papier, la solution proposée est d'évaluer la log-vraisemblance pour de petits modèles, et en conséquence de les entraîner sur des problèmes de dimension limitée, tout en essayant de s'assurer que l'évaluation est toujours valide. Ce travail devrait donc bénéficier d'une étude empirique plus poussée impliquant plusieurs ensembles de données complexes et reposant soit sur 1/ de nouvelles façons d'approcher la log-vraisemblance pour les modèles génératifs profonds (comme par exemple dans Murray et Salakhutdinov, 2009), ou 2/ l'évaluation par rapport à une mesure approximative de la performance telle que la précision en classification. Une autre question en suspens concerne l'absence de garantie que le problème est simplifié à chaque couche. En d'autres termes, quand un problème est transféré aux couches supérieures, le problème pourrait en théorie être aussi difficile que le problème initial, voire davantage. Heureusement, cela ne semble pas être le cas dans la pratique. Néanmoins, cette question est étroitement liée à la question de ce qui constitue une bonne représentation cachée. Une représentation qui simplifie le problème serait souhaitable, mais une étude approfondie serait nécessaire pour mieux comprendre cette question. Enfin, bien que nous discutons du critère pour l'optimisation de chaque couche, la maximisation est effectuée en utilisant l'algorithme de rétro-propagation habituel qui correspond à une descente de gradient stochastique. Dans le prochain article, nous examinons la possibilité d'utiliser la descente de gradient naturel pour apprendre les paramètres des RBMs, et potentiellement améliorer la qualité de l'estimation.

Troisième article

Ludovic Arnold, Anne Auger, Nikolaus Hansen, and Yann Ollivier. Information- geometric optimization algorithms: A unifying picture via invariance principles. Technical report, ArXiv e-prints, June 2011. URL http://arxiv.org/ abs/1106.3708.

Contexte

Les articles précédents ont discuté de la procédure d'apprentissage profond elle-même, et plus précisément de la question de la résolution d'un problème d'apprentissage profond par la résolution séquentielle de sous-problèmes.

Cependant, l'apprentissage d'une couche pourrait être amélioré par une meilleure procédure d'optimisation. Dans plusieurs contextes tels que les réseaux de neurones multicouches ou les réseaux de neurones récurrents, la procédure habituelle de descente de gradient ne semble pas conduire à une estimation satisfaisante de l'optimum. Cela rend importante la recherche de meilleures techniques d'optimisation tels que les méthodes de second ordre (Martens, 2010; Martens et Sutskever 2011;. Sutskever et al, 2011), le gradient naturel (Amari, 1998) et leur éventuelle

combinaison (Le Roux et Fitzgibbon, 2010).

Le gradient naturel utilise vraisemblablement la meilleure métrique pour maximiser la log-vraisemblance d'une distribution car il est invariant par rapport à la reparamétrisation et peut également conduire à plus d'invariance par rapport aux transformations de l'entrée. Néanmoins, il est rarement utilisé en pratique car il implique le calcul de la matrice de Fisher à chaque étape. Même si le calcul du gradient naturel exact sera souvent impraticable, des approximations pourraient être assez rapides pour être applicables en pratique, tout en conduisant à un meilleur optimum que ce qui est obtenu avec la descente de gradient habituelle.

Cela nous amène à demander ce qui se passe exactement quand le gradient habituel est utilisée à la place du gradient naturel et comment cela se rapporte à la paramétrisation et aux métriques.

Comme on peut s'attendre à ce qu'une mise en œuvre exacte du gradient naturel ne soit pas compétitive dans le cadre de l'apprentissage artificiel, nous considérons le contexte de l'optimisation boîte noire avec EDA, où les mises à jour du gradient du maximum de vraisemblance sont utilisées pour déplacer une distribution de proposition vers de meilleures valeurs de la fonction objective. Dans ce cadre, le coût de calcul est évalué par rapport au nombre d'évaluations de la fonction cible et ne prend pas en compte le calcul de la matrice de Fisher utilisé dans le gradient naturel.

Contributions

Dans cet article, le gradient naturel est examiné dans le contexte de l'optimisation boîte noire. Les expériences sont effectuées pour comparer les gradients habituels et naturels en essayant d'optimiser une fonction bi-modale simple à l'aide d'une RBM comme distribution de proposition.

Les résultats montrent clairement les avantages de l'utilisation du gradient naturel dans le contexte des EDA. Même en utilisant l'équivalent d'un ensemble 10.000 échantillons à chaque étape, le gradient habituel souffre d'une violation de la symétrie et favorise certaines configurations au détriment des autres. Cette rupture de la symétrie résulte en une perte prématurée de la diversité qui rend le gradient incapable de déplacer la distribution d'une RBM vers deux modes à la fois. Par comparaison, le gradient naturel rétablit la symétrie et est toujours en mesure de préserver la diversité pour tenir compte de la nature multimodale de l'entrée. Dans le cas des mises à jour de gradient stochastique où seuls 10 échantillons sont utilisés à chaque étape, le gradient naturel est encore capable de se déplacer vers les deux modes de la fonction objectif à condition que le taux d'apprentissage soit assez petit.

Ces résultats suggèrent que la pente naturelle peut être plus capable de gérer des distributions multimodales. Dans les applications pratiques où le gradient naturel peut être considéré comme trop coûteux, une voie possible pour atténuer le coût associée au gradient habituel peut être d'utiliser une paramétrisation aussi symétrique que possible. Cela nous amène à proposer une fonction d'énergie centrée pour les RBM.

Discussion

Bien que les résultats de cette étude soient présentés dans le cadre de l'optimisation avec des EDA, ils donnent aussi des indications utiles pour l'entraînement de modèles génératifs avec la descente de gradient qui est couramment utilisée dans le cadre de l'apprentissage artificiel.

Tout d'abord, nous montrons que la dépendance du gradient habituel sur la paramétrisation (Question 9) affecte le potentiel d'une RBM à apprendre des distributions multimodales. En effet, si le gradient habituel à tendance à perdre de la diversité en progressant vers deux modes avec une RBM bi-modale, on s'attend à avoir un problème de plus en plus sévère quand il s'agit de prendre en compte plus de 10^{300} modes comme c'est généralement le cas dans le cadre de l'apprentissage artificiel.

Quant à la question de savoir comment aborder ces difficultés (question 10), une première possibilité serait d'utiliser une approximation du gradient naturel (Le Roux et al, 2007; Desjardins et al, 2013) pour récupérer un certain niveau de symétrie. Toutefois, la mise à jour du gradient naturel exact n'est vraiment invariante à la reparamétrisation que dans la limite d'un taux d'apprentissage infinitésimal. Une possibilité intéressante est alors de restaurer directement symétrie dans la paramétrisation du modèle, par exemple en utilisant une fonction de l'énergie centrée (Montavon et Müller, 2012) tel que celle proposée dans ce papier.

Même si une énergie centrée ne dispose pas de tous les avantages du gradient naturel exact avec un taux d'apprentissage infinitésimal, elle doit permettre une meilleure trajectoire dans l'espace des paramètres avec presque aucune surcharge computationnelle.

Conclusion

L'apprentissage artificiel permet d'apprendre et de généraliser à partir de données en posant un problème d'apprentissage comme un problème d'optimisation.

L'objectif est alors de trouver un modèle qui maximise la performance. Cependant, comme nous considérons des tâches qui exigent de plus en plus d'intelligence, les modèles correspondants ont besoin de plus en plus de paramètres ce qui rend plus difficile l'optimisation: quand le nombre de paramètres augmente, il en va de même pour la dimensionnalité de l'espace de recherche. En dehors de la nécessité de puissance de calcul supplémentaire, l'optimisation dans des espaces en grande dimension requiert habituellement de grandes quantités de données qu'il peut être difficile, voire impossible de trouver dans le cadre supervisé où les étiquettes sont obtenues avec une intervention humaine. Une façon de contourner le problème peut être d'apprendre des représentations avec un algorithme d'apprentissage non supervisé et de tirer profit des grandes quantités de données non étiquetées qui sont presque toujours disponibles. Le problème d'apprentissage est alors résolu en deux étapes: i / apprentissage d'une représentation appropriée et ii / résolution du problème étant donné cette représentation. Cela étant dit, il y a toujours la question de savoir comment apprendre une représentation intéressante. Cela peut être fait avec des données non étiquetées ce qui est une amélioration, mais peut nécessiter beaucoup de paramètres car l'apprentissage non supervisé a tendance à prendre tous les aspects de la distribution des entrées en compte et pas seulement ceux qui sont nécessaires pour résoudre le problème final.

Dans ce contexte, l'apprentissage en profondeur peut conduire à une réponse pratique. Premièrement, les architectures profondes sont capables d'effectuer des opérations complexes avec beaucoup moins de paramètres que les moins profondes, ce qui les rend plus faciles à entraîner. Deuxièmement, la possibilité d'utiliser une procédure d'entraînement couche par couche diminue le coût computationnel en faisant de l'optimisation séquentiellement séparable: au lieu d'apprendre tous les paramètres du modèle à la fois, l'optimisation est faite une couche à la fois.

Concernant la consistance de la procédure d'entraînement couche par couche, la justification actuelle est basée sur la maximisation d'une borne inférieure variationnelle. Cette approche conduit à une garantie que la log-vraisemblance s'améliore quand une couche est ajoutée sur le dessus d'un modèle à une couche, mais la garantie n'est pas valable pour plus de deux couches. En outre, l'optimisation de la borne variationnelle n'est pas consistante car elle ne conduit pas au même optimum que l'optimisation de toutes les couches à la fois. Dans cette thèse, nous proposons un nouveau critère pour l'entraînement couche par couche de modèles génératifs profonds: la borne suérieure de la meilleure marginale latente (Best Latent Marginal upper bound ou plus simplement BLM upper bound). Nous prouvons que la maximisation de ce critère à chaque étape conduit à un modèle génératif profond optimal, à condition que les couches supérieures soit entraînées avec succès. La borne supérieure de la BLM correspond à la plus haute valeur de la log-vraisemblance atteignable en ajoutant des couches. Il en résulte un nouveau paradigme pour l'apprentissage profond couche par couche: la maximisation du potentiel de vraisemblance d'un modèle génératif

profond alors même que les couches supérieures ne sont pas encore connues. La borne supérieure de la BLM a également des liens étroits avec les modèles encodeur-décodeur. À savoir la procédure d'entraînement des auto-encodeurs avec l'erreur de reconstruction correspond à une approximation de la borne supérieure BLM. Cela conduit à une nouvelle justification pour l'empilement d'auto-associateurs comme étant une méthode approximative pour la formation de la partie inférieure d'un modèle génératif profond et suggère que la partie encodeur du modèle devrait être aussi riche que possible.

Ceci est confirmé par nos expériences dans lesquelles des Auto-Encodeurs avec Riche Inférence (AERIes) obtiennent de meilleurs résultats que l'approche variationnelle classique sur deux bases de données distinctes.

Pour l'évaluation de la performance, avoir une mesure calculable est critique pour la sélection de modèle où plusieurs modèles doivent être comparés pour choisir le plus performant. Les approches actuelles typiquement basées sur une tâche supervisée ou sur le calcul de la log-vraisemblance d'un ensemble de validation sont associées à un coût de calcul élevé. Dans cette thèse, nous proposons d'utiliser la borne supérieure de la BLM pour évaluer la performance potentielle des couches inférieures avant que la partie supérieure du modèle ait été entraînée. Dans le contexte de l'apprentissage de représentations, la borne supérieure de la BLM limite supérieure donne un compte rendu précis de la qualité de la représentation, même si un modèle est incomplet. Nous montrons dans nos expériences que la borne supérieure de la BLM est un bon estimateur de la log-vraisemblance finale et décrivons une procédure consistante pour la sélection de modèle: sélectionner chaque couche à son tour, selon la borne supérieure de la BLM, et sélectionner le modèle génératif final selon la log-vraisemblance.

Entraîner efficacement chaque couche peut être un problème d'optimisation difficile. Les approches actuelles reposent souvent sur une descente de gradient avec la métrique euclidienne pour effectuer l'optimisation dans l'espace des paramètres. Lorsque l'objectif est d'estimer une distribution, nos expériences montrent que le choix de la métrique euclidienne introduit une dépendance parasite sur la paramétrisation qui peut entraîner une violation de la symétrie et une difficulté à tenir compte des distributions multimodales. Dans cette thèse, nous montrons l'importance de considérer les métriques pour l'optimisation et montrons que le gradient naturel ou une paramétrisation centrée peuvent être utilisés pour améliorer la trajectoire d'une procédure de descente de gradient.

Ces contributions donnent lieu à plusieurs nouvelles pistes de recherche.

La validation empirique de la BLM n'a pu être obtenue que sur des modèles relativement petits en raison de l'incalculabilité de la log-vraisemblance. Les progrès récents en matière d'échantillonnage MCMC peuvent permettre le calcul de la log-vraisemblance sur de plus grands modèles dans un délai raisonnable. Cela permettrait une comparaison entre la borne supérieure de la BLM et la maximisation traditionnelle de la borne variationnelle sur des ensembles de données nécessitant plus de couches où la BLM est en principe avantagée.

Plusieurs modèles tels que les Spike and Slab RBMs et les machines de Boltzmann profondes peuvent être plus performants que les RBMs et auto-associateurs avec l'aide d'un pré-entraînement couche par couche variationnel. Utiliser ces modèles plus riches en combinaison avec la borne supérieure de la BLM pourrait mener à d'encore meilleurs résultats.

Malgré plusieurs tentatives pour appliquer les principes d'apprentissage en profondeur à d'autres modèles, la plupart des tentatives réussies concernent les réseaux de neurones. Dans cette thèse, nous introduisons une forme générale pour les modèles génératifs profonds qui n'est pas limité aux réseaux de neurones et peut s'appliquer à d'autres classes de modèles. Une possibilité serait donc d'appliquer la BLM à de nouveaux modèles qui ne soient pas des réseaux de neurones.

Utiliser une inférence plus riche dans le cadre de l'encodeur d'un encodeur-décodeur s'est avéré être un bon moyen d'améliorer les performances lors de l'apprentissage des réseaux de neurones profonds. Ce principe est largement applicable et peut conduire à augmenter les performances des

autres modèles comme par exemple ceux couramment utilisés dans le codage clairsemé.

Le succès des méthodes d'apprentissage en profondeur suggère que les auto-associateurs et les RBMs sont capables de simplifier un problème pour les couches supérieures. Cependant, il n'existe aucune justification théorique de ces propriétés de simplification. Comprendre comment une distribution peut être simplifiée peut conduire à une meilleure compréhension de ce qui constitue une bonne représentation de l'apprentissage en profondeur.

Cette thèse met en évidence l'importance de choisir une bonne métrique pour effectuer l'optimisation. Bien que dans plusieurs applications, le gradient naturel exact peut avoir un coût de calcul prohibitif, de meilleures métriques peuvent être trouvées en changeant la paramétrisation ou par approximation du gradient naturel.