École Nationale Supérieure d'Arts et Métiers Meknes

Convolutional Neural Network

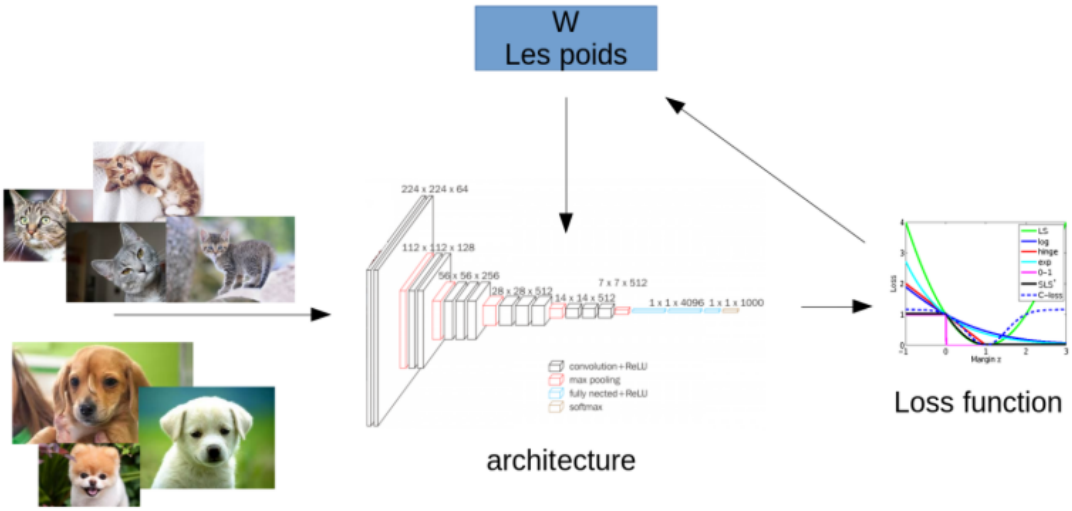
Intelligence Artificielle

Projet -Version Française-

Student :
Imad MAALOUF

Supervisor :
Pr. Tarik HAJJI

Convolutional Neural Network



Dans cet article, je vais expliquer le concept des réseaux neuronaux convolutionnels (CNN) en mettant en œuvre de nombreux exemples avec des images et en argumentant l'utilisation des CNN par rapport aux réseaux neuronaux multicouches classiques pour le traitement d'images. Plongeons et discutons des CNN (réseaux neuronaux convolutionnels) en détail, ce qui vous sera plus utile.

Table des matières

Table des figures

Listings

1 INTRODUCTION GENERALE

1.1 Introduction

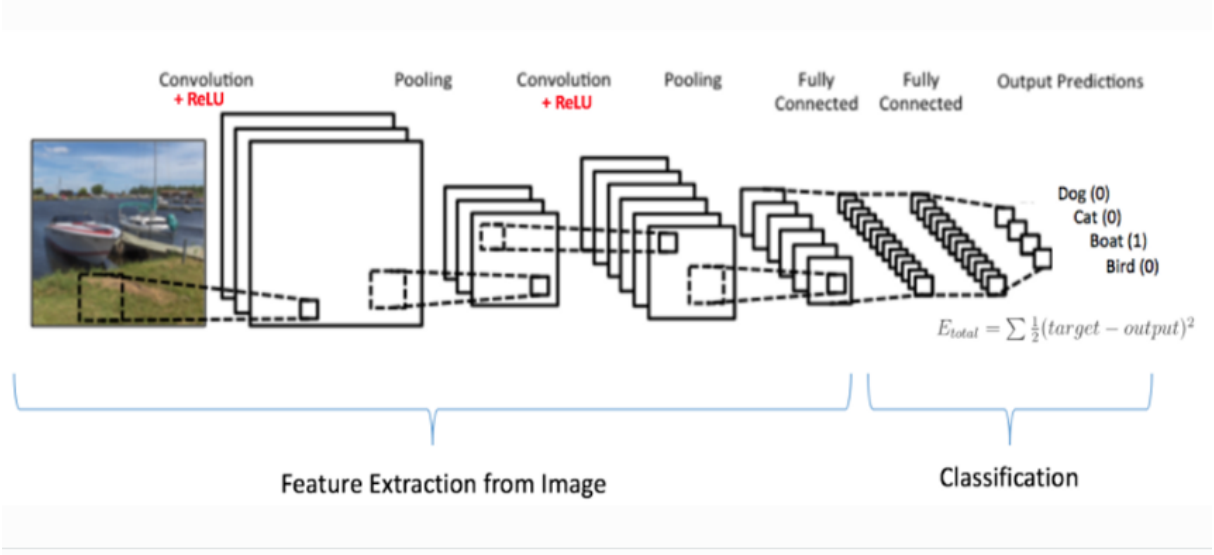


FIGURE .1 – Image explicatif pour CNN

1.2 Qu'est-ce qu'un CNN ?

Un réseau neuronal convolutionnel (CNN), également connu sous le nom de ConvNet, est un type spécialisé d'algorithme d'apprentissage profond principalement conçu pour les tâches nécessitant la reconnaissance d'objets, notamment la classification, la détection et la segmentation d'images. Les CNN sont utilisés dans divers scénarios pratiques, tels que les véhicules autonomes, les systèmes de caméras de sécurité et autres.

1.3 L'importance des CNN

Il y a plusieurs raisons pour lesquelles les réseaux de neurones convolutifs (CNN) sont importants dans le monde moderne,comme le souligne ci-dessous :

1. Les CNN se distinguent des algorithmes classiques d'apprentissage automatique tels que les SVM et les arbres de décision par leur capacité à extraire automatiquement des caractéristiques à grande échelle, évitant ainsi le besoin d'ingénierie manuelle des caractéristiques et améliorant ainsi l'efficacité.
2. Les couches de convolution confèrent aux CNN leurs caractéristiques d'invariance à la translation, les rendant capables d'identifier et d'extraire des motifs et des caractéristiques à partir des données, indépendamment des variations de position, d'orientation, d'échelle ou de translation.
3. Une variété d'architectures de CNN pré-entraînées, notamment VGG-16, ResNet50, Inceptionv3 et EfficientNet, ont démontré des performances de premier plan. Ces modèles peuvent être adaptés à de nouvelles tâches avec relativement peu de données grâce à un processus appelé "fine-tuning".

4. Au-delà des tâches de classification d'images, les CNN sont polyvalents et peuvent être appliqués à une gamme d'autres domaines, tels que le traitement du langage naturel, l'analyse de séries temporelles et la reconnaissance vocale.

1.4 Actions utilisées dans un CNN

Un CNN applique généralement 3 types d'opérations différentes à une image afin d'en extraire les informations pertinentes.

Ces 3 types d'opérations sont les suivantes :

1. La convolution
2. Le pooling
3. La fonction d'activation de type ReLU
4. La couche fully-connected

Nous allons nous intéresser à chacune de ses opérations.

2 ARCHITECTURE D'UN CNN

2.1 Architecture d'un Convolutional Neural Network-CNN

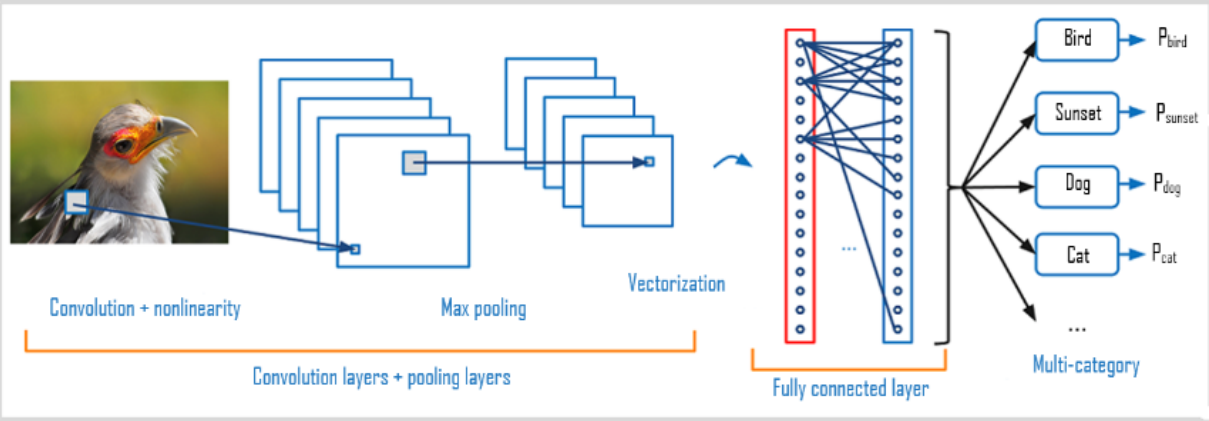


FIGURE .2 – Architecture d'un CNN

Les CNN désignent une sous-catégorie de réseaux de neurones et sont à ce jour un des modèles de classification d'images réputés être les plus performant.

Leur mode de fonctionnement est à première vue simple : l'utilisateur fournit en entrée une image sous la forme d'une matrice de pixels.

Celle-ci dispose de 3 dimensions :

1. Deux dimensions pour une image en niveaux de gris.
2. Une troisième dimension, de profondeur 3 pour représenter les couleurs fondamentales (Rouge, Vert, Bleu).

Contrairement à un modèle MLP (Multi Layers Perceptron) classique qui ne contient qu'une partie classification, l'architecture du Convolutional Neural Network dispose en amont d'une partie convolutive et comporte par conséquent deux parties bien distinctes :

- Une partie convolutive : Son objectif final est d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale. En résumé, l'image fournie en entrée passe à travers une succession de filtres, créant par la même occasion de nouvelles images appelées cartes de convolutions. Enfin, les cartes de convolutions obtenues sont concaténées dans un vecteur de caractéristiques appelé code CNN

- Une partie classification : Le code CNN obtenu en sortie de la partie convolutive est fourni en entrée dans une deuxième partie, constituée de couches entièrement connectées appelées perceptron multicouche (MLP pour Multi Layers Perceptron). Le rôle de cette partie est de combiner les caractéristiques du code CNN afin de classer l'image.

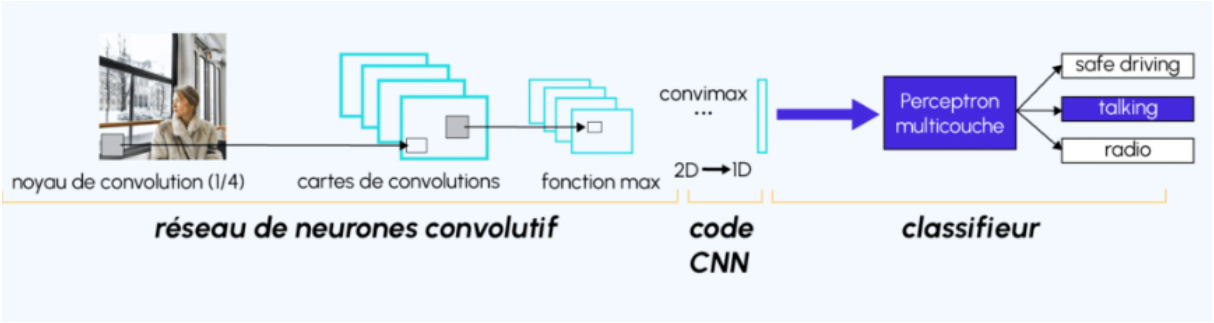


FIGURE .3 – réseau de neurones convolutif

Note

Il existe quatre types de couches pour un réseau de neurones convolutif : la couche de convolution, la couche de pooling, la couche de correction ReLU et la couche fully-connected. Dans ce chapitre, je vais vous expliquer le fonctionnement de ces différentes couches

3 La couche de convolution

La convolution est une opération mathématique simple généralement utilisée pour le traitement et la reconnaissance d'images.

3.1 À quoi sert la convolution ?

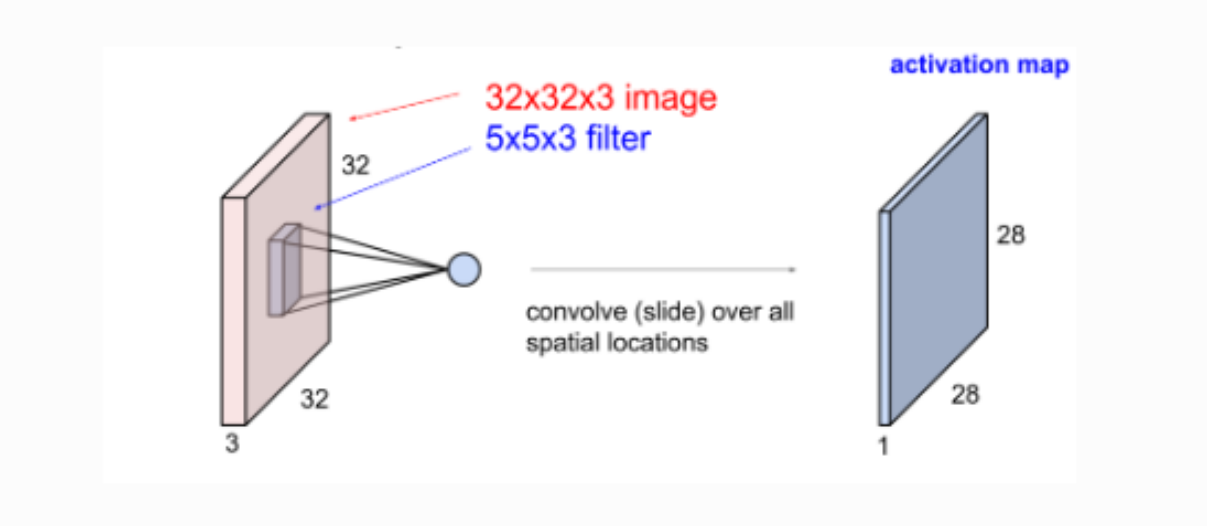


FIGURE .4 – La couche de convolution

La couche de convolution est la composante clé des réseaux de neurones convolutifs, et constitue toujours au moins leur première couche.

Son but est de repérer la présence d'un ensemble de features dans les images reçues en entrée.

Pour cela, on réalise un filtrage par convolution : le principe est de faire "glisser" une fenêtre représentant la feature sur l'image, et de calculer le produit de convolution entre la feature et chaque portion de l'image balayée.

Une feature est alors vue comme un filtre : les deux termes sont équivalents dans ce contexte.

Remarque

Cette technique est très proche de celle étudiée dans la partie précédente pour faire du template matching : ici, c'est le produit convolution qui est calculé, et non la corrélation croisée.

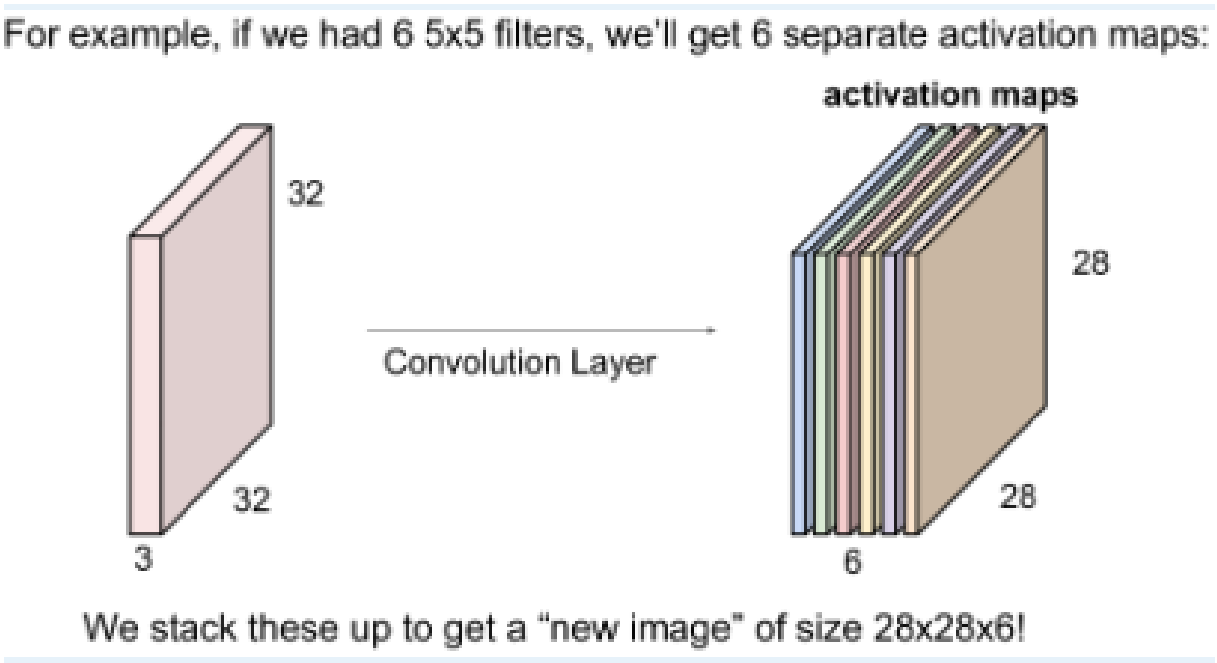


FIGURE .5 – filtrage par convolution

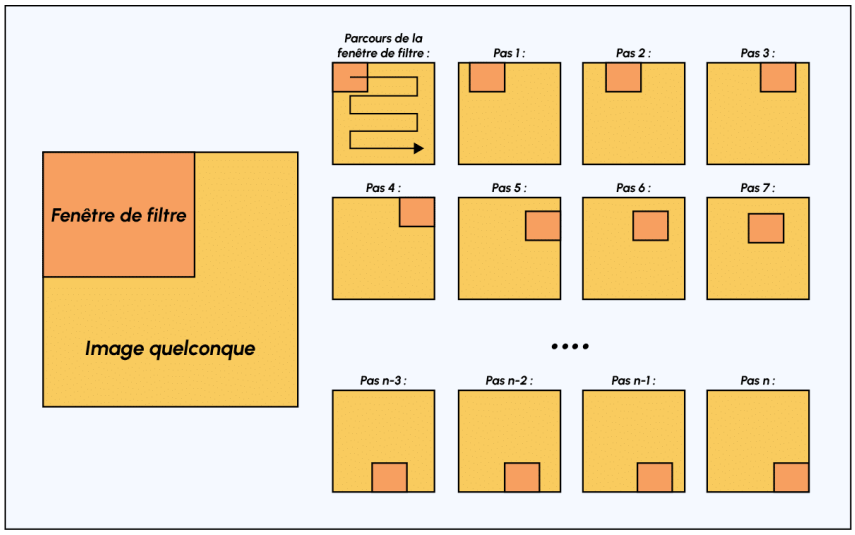


FIGURE .6 – filtrage par convolution

Dans un premier temps, on définit la taille de la fenêtre de filtre située en haut à gauche.

La fenêtre de filtre, représentant la feature, se déplace progressivement de la gauche vers la droite d'un certain nombre de cases défini au préalable (le pas) jusqu'à arriver au bout de l'image.

À chaque portion d'image rencontrée, un calcul de convolution s'effectue permettant d'obtenir en sortie une carte d'activation ou feature map qui indique où est localisées les features dans l'image : plus la feature map est élevée, plus la portion de l'image balayée ressemble à la feature.

3.2 Exemple d'un filtre de convolution classique

Lors de la partie convolutive d'un Convolutional Neural Network, l'image fournie en entrée passe à travers une succession de filtres de convolution. Par exemple, il existe des filtres de convolution fréquemment utilisés et permettant d'extraire des caractéristiques plus pertinentes que des pixels comme la détection des bords (filtre dérivateur) ou des formes géométriques. Le choix et l'application des filtres se fait automatiquement par le modèle.

Parmi les filtres les plus connus, on retrouve notamment le filtre moyenneur (calcule pour chaque pixel la moyenne du pixel avec ses 8 proches voisins) ou encore le filtre gaussien permettant de réduire le bruit d'une image fournie en entrée :

Voici un exemple des effets de ces deux différents filtres sur une image comportant un bruit important (on peut penser à une photographie prise avec une faible luminosité par exemple). Toutefois, un des inconvénients de la réduction du bruit est qu'elle s'accompagne généralement d'une réduction de la netteté :

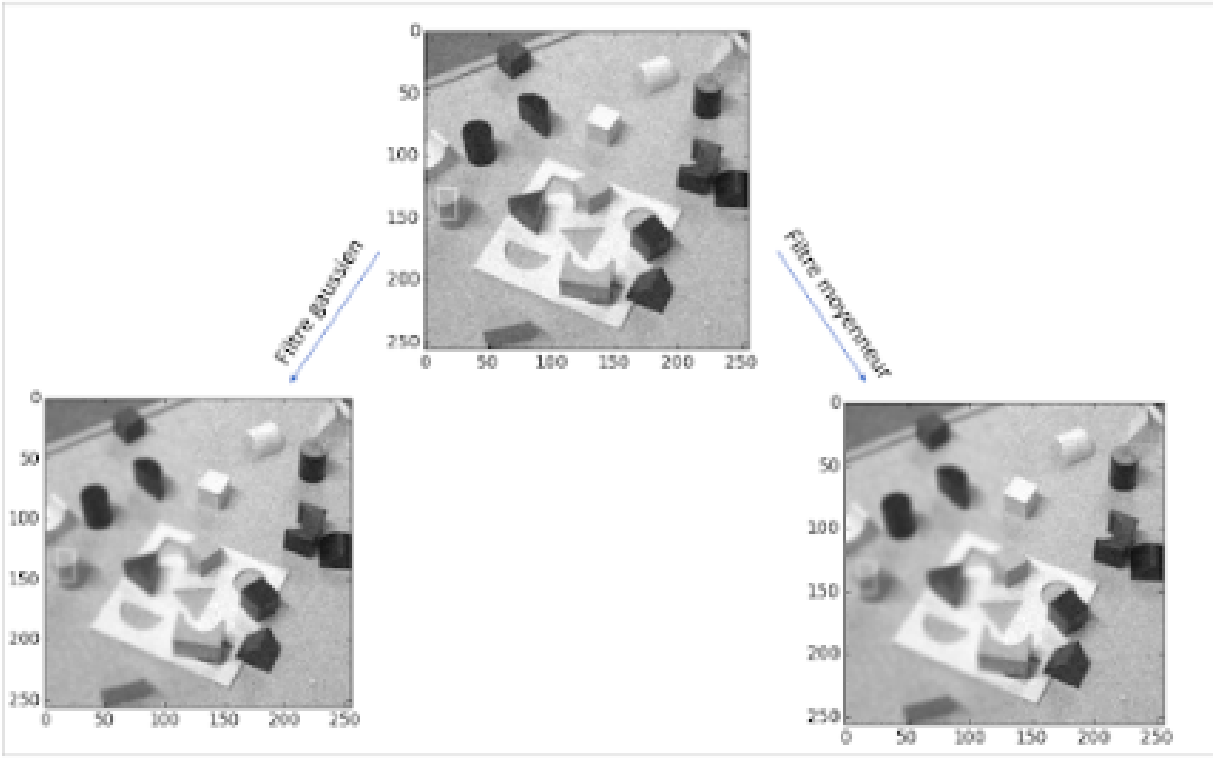


FIGURE .7 – les effets des différents filtres

Comme on peut l'observer, contrairement au filtre moyenneur, le filtre gaussien réduit le bruit sans pour autant réduire significativement la netteté.

Outre sa fonction de filtrage, l'intérêt de la partie convolutive d'un CNN est qu'elle permet d'extraire des caractéristiques propres à chaque image en les compressant de façon à réduire leur taille initiale, via des méthodes de sous-échantillonnage tel que le Max-Pooling.

4 La couche de pooling

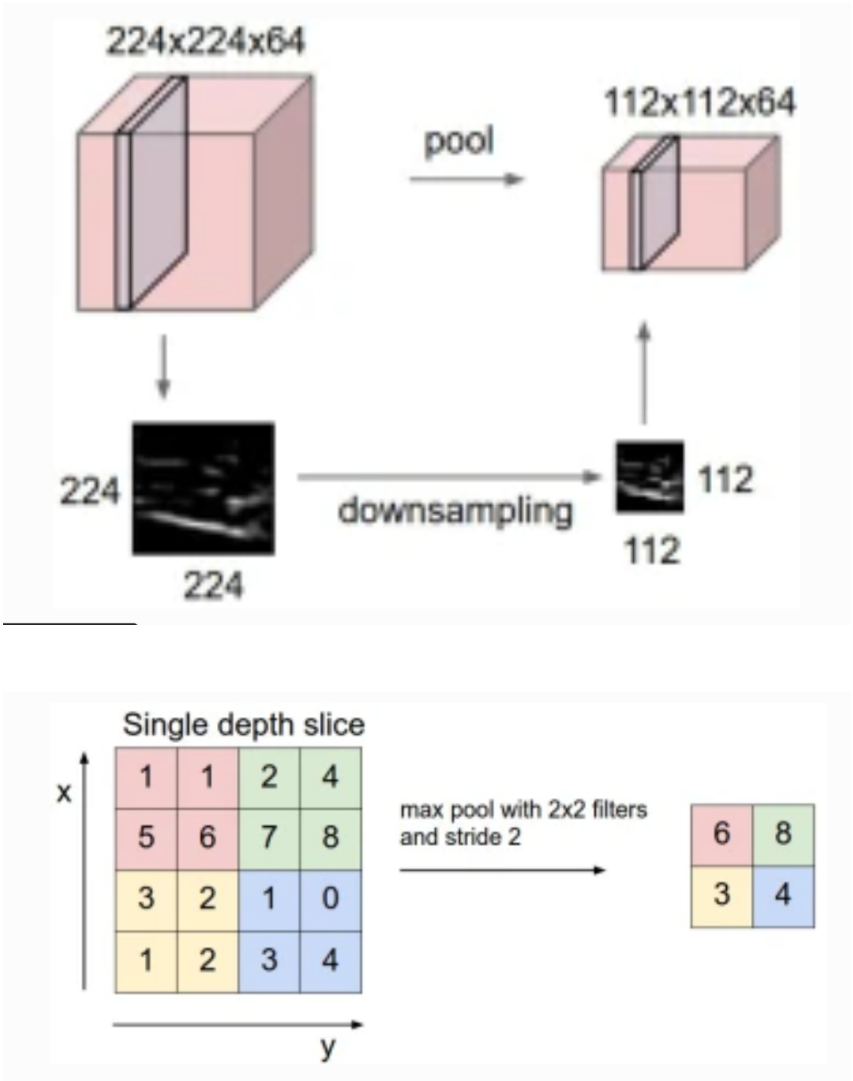


FIGURE .8 – La couche de pooling

But

Le but du Pooling est de réduire la taille des "images" mais également de pallier le phénomène d' « overfitting ». Tout comme pour la convolution on applique un filtre qu'on fait glisser sur l'image et (dans la version "max pooling") on garde la valeur max sur chaque fenêtre. On parle aussi de « Down-Sampling » ou « Sub-Sampling »

4.1 À quoi sert la pooling ?

Ce type de couche est souvent placé entre deux couches de convolution : elle reçoit en entrée plusieurs feature maps, et applique à chacune d'entre elles l'opération de pooling.

L'opération de pooling (ou sub-sampling)consiste à réduire la taille des images, tout en préservant leurs caractéristiques importantes.

Pour cela, on découpe l'image en cellules régulière, puis on garde au sein de chaque cellule la valeur maximale. En pratique, on utilise souvent des cellules carrées de petite taille pour ne pas perdre trop d'informations. Les choix les plus communs sont des cellules

adjacentes de taille 2 × 2 pixels qui ne se chevauchent pas, ou des cellules de taille 3 × 3 pixels, distantes les unes des autres d'un pas de 2 pixels (qui se chevauchent donc). On obtient en sortie le même nombre de feature maps qu'en entrée, mais celles-ci sont bien plus petites.

La couche de pooling permet de réduire le nombre de paramètres et de calculs dans le réseau. On améliore ainsi l'efficacité du réseau et on évite le sur-apprentissage.

Il est courant d'insérer périodiquement une couche Pooling entre les couches Conv successives dans une architecture ConvNet. Sa fonction est de réduire progressivement la taille spatiale de la représentation afin de réduire la quantité de paramètres et de calculs dans le réseau, et donc de contrôler également le sur-ajustement. La couche de pooling fonctionne indépendamment sur chaque tranche de profondeur de l'entrée et la redimensionne spatialement, en utilisant l'opération MAX.

Ainsi, la couche de pooling rend le réseau moins sensible à la position des features : le fait qu'une feature se situe un peu plus en haut ou en bas, ou même qu'elle ait une orientation légèrement différente ne devrait pas provoquer un changement radical dans la classification de l'image.

Note :

Pooling : réduire la pile d'images

4.2 Exemple

les etapes de pooling

1. - Choisissez une taille de fenêtre (généralement 2 ou 3).
2. - Choisissez un pas (généralement 2).
3. - Parcourez votre fenêtre à travers vos images filtrées.
4. - De chaque fenêtre, prenez la valeur maximale.

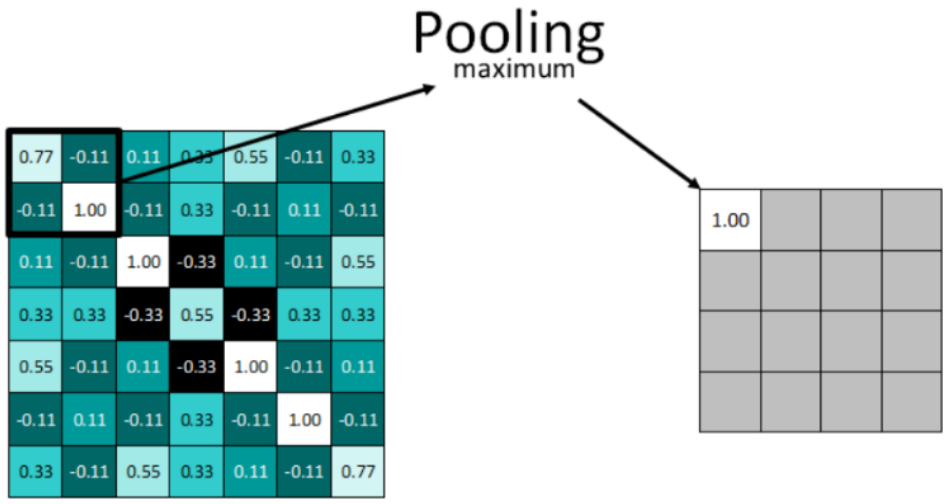


FIGURE .9 – les etapes de pooling

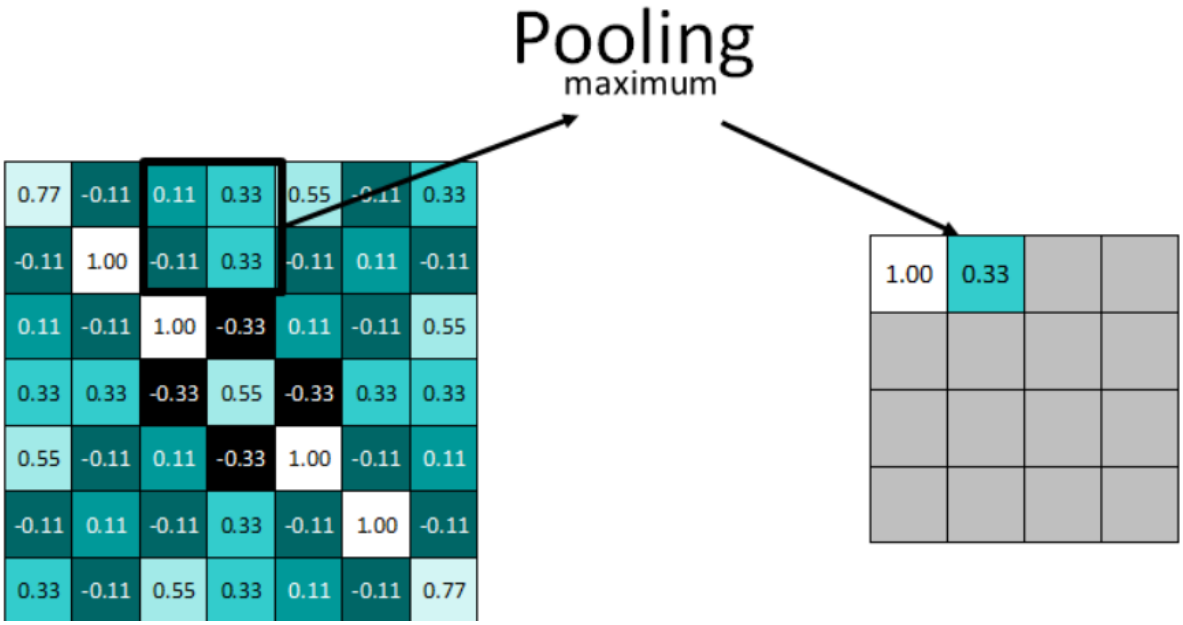


FIGURE .10 – les etapes de pooling

Après avoir procédé au pooling, l'image n'a plus qu'un quart du nombre de ses pixels de départ.

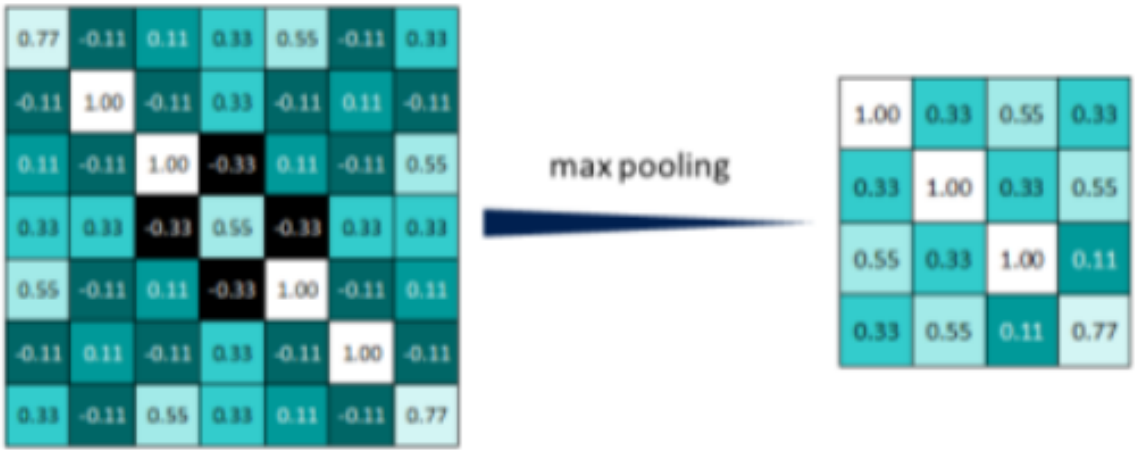


FIGURE .11 – max pooling

Parce qu'il garde à chaque pas la valeur maximale contenue dans la fenêtre, il préserve les meilleurs caractéristiques de cette fenêtre. Cela signifie qu'il ne se préoccupe pas vraiment d'où a été extraite la caractéristique dans la fenêtre.

Le résultat est que le CNN peut trouver si une caractéristique est dans une image, sans se soucier de l'endroit où elle se trouve. Cela aide notamment à résoudre le problème liés au fait que les ordinateurs soient hyper-littéraires.

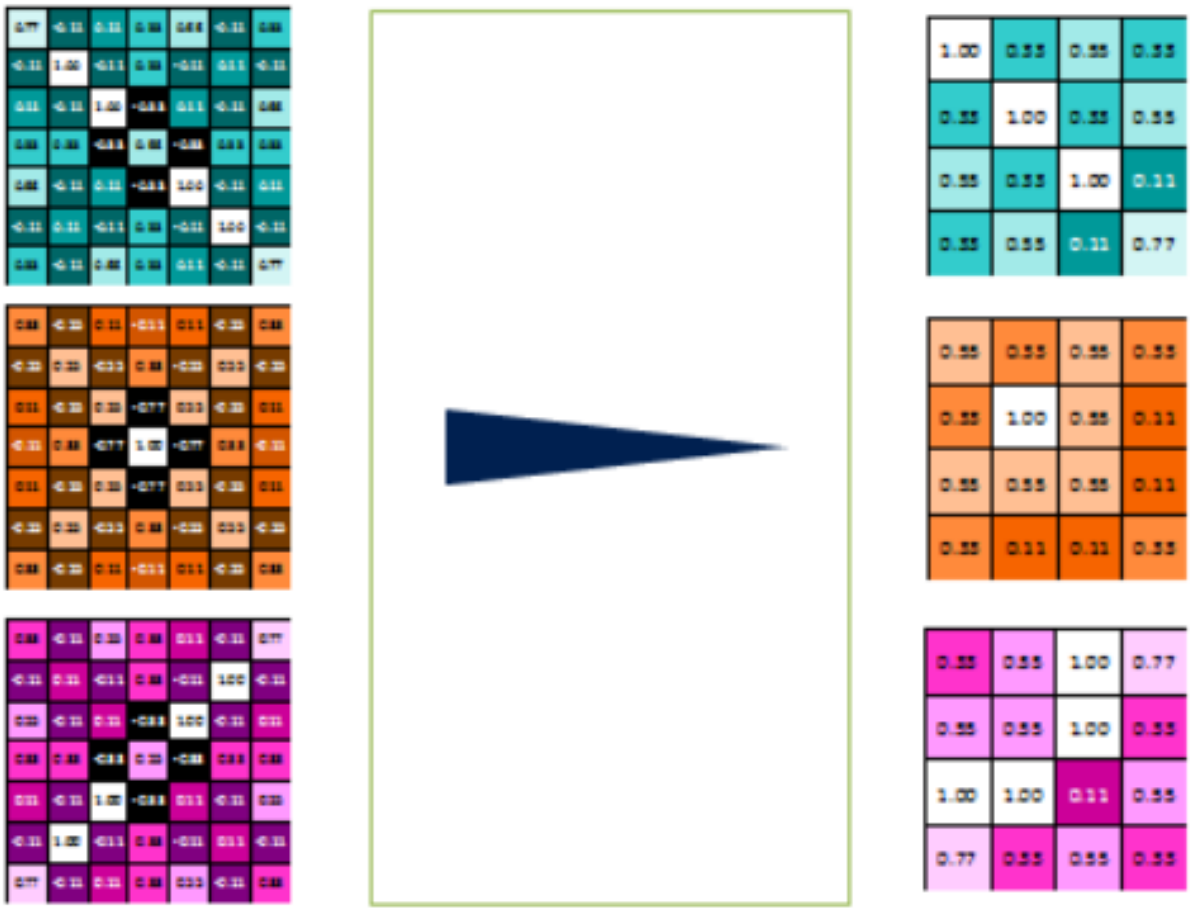


FIGURE .12 – couche de pooling traiter

Au final, une couche de pooling est simplement un traitement de pooling sur une image ou une collection d'images. L'output aura le même nombre d'images mais chaque images aura un nombre inférieur de pixels. Cela permettra ainsi de diminuer la charge de calculs. Par exemple, en transformant une image de 8 mégapixels en une image de 2 mégapixels, ce qui nous rendra la vie beaucoup plus facile pour le reste des opérations à effectuer par la suite.

5 La couche de correction ReLU

5.1 Introduction

Pour améliorer l'efficacité du traitement en intercalant entre les couches de traitement une couche qui va opérer une fonction mathématique (fonction d'activation) sur les signaux de sortie. dans ce cadre on trouve

ReLU (Rectified Linear Units) désigne la fonction réelle non-linéaire définie par ReLU(x)=max(0,x).

La couche de correction ReLU remplace donc toutes les valeurs négatives reçues en entrées par des zéros. Elle joue le rôle de fonction d'activation

5.2 Normalization

Changer tout négatif à zéro

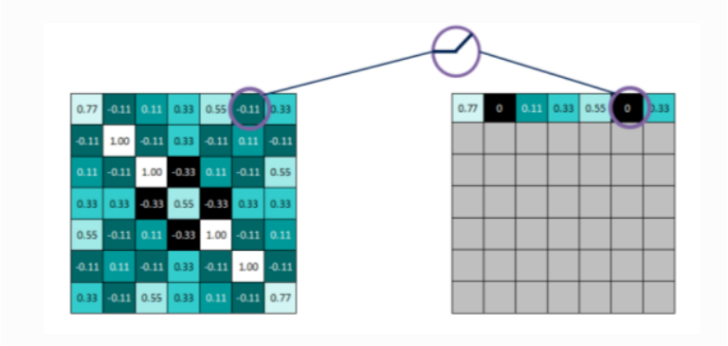


FIGURE .13 – Normalization

Un élément important dans l'ensemble du processus est l'Unité linéaire rectifiée ou ReLU. Les mathématiques derrière ce concept sont assez simples encore une fois : chaque fois qu'il y a une valeur négative dans un pixel, on la remplace par un 0. Ainsi, on permet au CNN de rester en bonne santé (mathématiquement parlant) en empêchant les valeurs apprises de rester coincer autour de 0 ou d'exploser vers l'infinie.

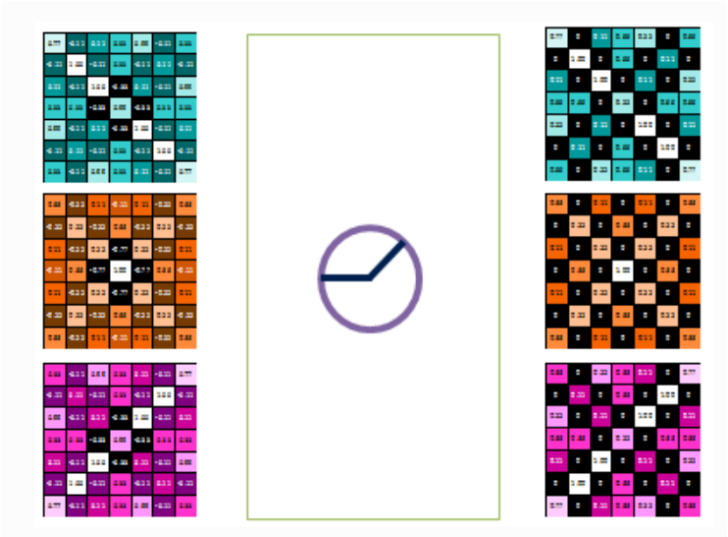


FIGURE .14 – ReLU "mathématiques"

C'est un outil pas vraiment sexy mais fondamental car sans lequel le CNN ne produirait pas vraiment les résultats qu'on lui connaît.

Le résultat d'une couche ReLU est de la même taille que ce qui lui est passé en entrée, avec simplement toutes les valeurs négatives éliminées.

La sortie de l'un devient l'entrée du suivant.

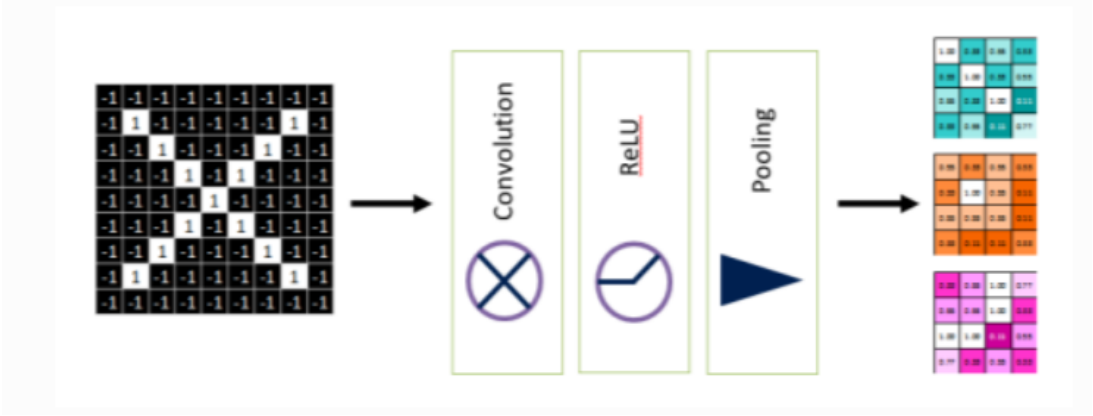


FIGURE .15 – La sortie de la couche ReLU

6 La couche fully-connected

Les CNNs ont une autre flèche dans leur carquois. En effet, les couches entièrement connectés prennent les images filtrées de haut niveau et les traduisent en votes. Dans notre exemple, nous devons seulement décider entre deux catégories, X et O.

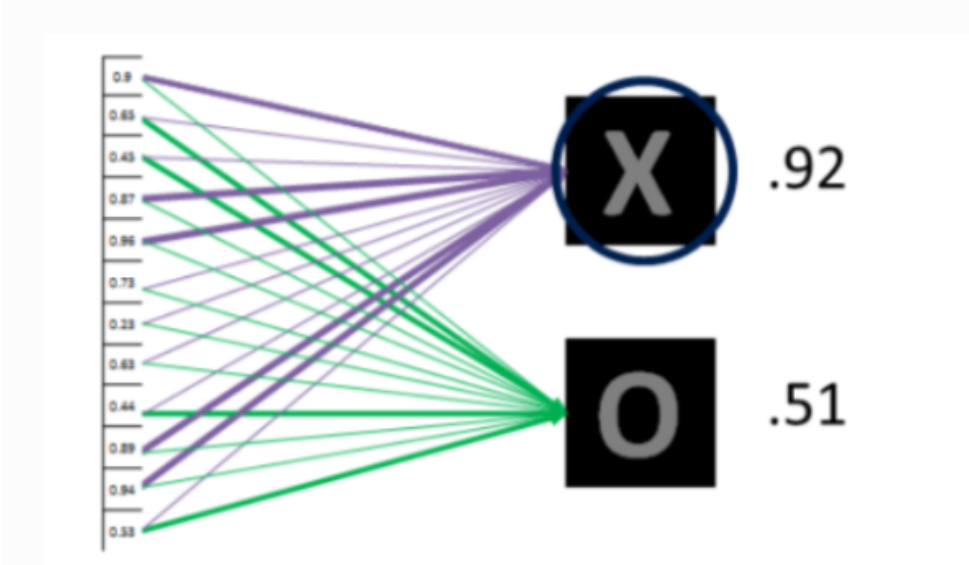


FIGURE .16 – La couche fully-connected

Les couches entièrement connectées sont les principaux blocs de construction des réseaux de neurones traditionnels. Au lieu de traiter les inputs comme des tableaux de 2 Dimensions, ils sont traités en tant que liste unique et tous traités de manière identique. Chaque valeur a son propre vote quant à si l'image est un X ou un O. Cependant, le

process n'est pas complètement démocratique. Certaines valeurs sont bien meilleures à détecter lorsqu'une i mage est un X que d'autres, et d'autres sont bien meilleures à détecter un O. Celles-ci ont donc davantage de pouvoir de vote que les autres. Ce vote est appelé le poids, ou la force de la connection, entre chaque valeur et chaque catégorie.

Lorsqu'une nouvelle image est présentée au CNN, elle se répand à travers les couches inférieures jusqu'à atteindre la couche finale entièrement connectée. L'élection a ensuite lieu. Et la solution avec le plus de vote gagne et est déclarée la catégorie de l'image.

Les couches entièrement connectées, tout comme les autres couches, peuvent être ajoutées les unes à la suite des autres car leur valeur en sortie (une liste de votes) ressemble énormément à leur valeur en entrée (une liste de valeur). En pratique, plusieurs couches entièrement connectées sont souvent ajoutées les unes à la suite des autres, avec chaque couche intermédiaire votant pour des catégories "cachées" fantômes. En effet, chaque couche additionnelle laisse le réseau apprendre des combinaisons chaque fois plus complexes de caractéristiques qui aident à améliorer la prise de décision.

FIGURE .17 – La couche fully-connected

I Avantages et Défis des Réseaux de Neurones Convolutifs (CNNs)

1 Avantages des CNN

Les CNNs présentent de nombreux avantages qui les rendent particulièrement performants pour les tâches de vision par ordinateur. Voici quelques-uns de leurs points forts :

1.1 Extraction automatique de caractéristiques

Contrairement aux méthodes traditionnelles qui nécessitaient une définition manuelle des caractéristiques d'une image, les CNNs ont la capacité d'extraire automatiquement des caractéristiques pertinentes à partir des données d'entrée. Ceci est réalisé grâce à leurs couches convolutives, qui appliquent des filtres à différents niveaux de granularité pour identifier des motifs simples (lignes, contours) aux structures plus complexes (formes, objets).

1.2 Invariance spatiale

Les CNNs sont robustes aux variations de position, de taille et d'orientation des objets dans une image. Cette invariance spatiale est obtenue grâce aux opérations de convolution et de pooling, qui permettent aux CNNs de reconnaître des objets quelle que soit leur disposition dans l'image.

1.3 Robustesse au bruit

Les CNNs sont conçus pour tolérer le bruit et les imperfections dans les images, ce qui les rend adaptés aux situations réelles où la qualité des images peut varier. Les couches de pooling et l'apprentissage par échantillonnage aléatoire (data augmentation) contribuent à cette robustesse en réduisant l'impact du bruit et en permettant aux CNNs de s'adapter à des données imparfaites.

1.4 Transfert d'apprentissage (Transfer Learning)

Plutôt que de partir de zéro, les CNNs peuvent tirer parti de modèles pré-entraînés sur des ensembles de données massifs (comme ImageNet) pour de nouvelles tâches. En ne réentrainant que les dernières couches du modèle sur un nouvel ensemble de données spécifique, on peut obtenir des performances élevées avec beaucoup moins de données et de temps de calcul.

1.5 Performances de pointe

Les CNNs ont atteint des performances inégalées sur un large éventail de tâches de vision par ordinateur, surpassant souvent les approches traditionnelles. Ils excellent notamment dans la classification d'images, la détection d'objets et la segmentation sémantique.

2 Défis et Limitations des CNN

Malgré leurs nombreux avantages, les CNNs ne sont pas sans défis et limitations. Voici quelques points importants à prendre en compte :

2.1 Dépendance aux données

Les CNNs nécessitent une grande quantité de données d'entraînement étiquetées pour fonctionner efficacement. La collecte et l'annotation de ces données peuvent être coûteuses et prendre du temps, ce qui limite l'applicabilité des CNNs aux domaines où il est difficile ou onéreux d'obtenir des données de qualité.

2.2 Nature "boîte noire"

Qu'est-ce que la "boîte noire" des CNNs ?
L'architecture complexe des CNNs, composée de nombreuses couches et d'un grand nombre de paramètres, rend difficile la compréhension des processus internes qui mènent à leurs décisions. En d'autres termes, il est difficile de savoir "comment un CNN pense".
Pourquoi la "boîte noire" est-elle problématique ?
Cette "boîte noire" pose plusieurs problèmes :

1. **Limite la confiance dans les modèles** : Si l'on ne comprend pas comment un CNN arrive à sa conclusion, il est difficile de lui accorder une confiance totale, surtout dans des situations critiques où une interprétation claire est essentielle.
2. **Entraîve des difficultés de débogage :** Lorsque des erreurs se produisent, il peut être ardu de les identifier et de les corriger sans une compréhension fine du fonctionnement interne du CNN. Cela peut ralentir le développement et l'amélioration des modèles.
3. **Empêche l'amélioration des performances :**Si l'on ne comprend pas comment un CNN "pense", il devient difficile d'identifier les points faibles et de mettre en place des stratégies pour améliorer ses performances.

2.3 Sensibilité aux Variations des Données

Les CNNs peuvent être sensibles aux variations subtiles dans les données d'entrée. De petits changements dans l'image d'entrée, comme le bruit ou les transformations géométriques, peuvent parfois entraîner des prédictions erronées. Cette sensibilité nécessite une préparation et une augmentation des données minutieuses pour s'assurer que le modèle est robuste et capable de généraliser à des scénarios réels variés.

2.4 Ressources Computationnelles Intenses

L'entraînement des CNN est computationnellement intensif, nécessitant des ressources matérielles puissantes et spécialisées comme les GPU. Cette exigence en matière de calcul peut limiter l'accessibilité des CNN, surtout pour les petites entreprises ou les chercheurs indépendants qui ne disposent pas des infrastructures nécessaires pour entraîner ces modèles à grande échelle.

3 Conclusion

En résumé, les CNNs offrent une combinaison unique d'avantages qui les positionnent comme des outils de pointe pour la vision par ordinateur. Leur capacité à extraire automatiquement des caractéristiques, leur invariance spatiale, leur robustesse au bruit, leur transférabilité et leurs performances exceptionnelles en font des éléments essentiels pour de nombreuses applications innovantes.

Cependant, il est important de reconnaître les défis et limitations auxquels les CNNs sont confrontés, tels que la dépendance aux données, la nature "boîte noire", la sensibilité aux variations des données et les exigences computationnelles élevées. En s'attaquant à ces défis et en développant des techniques pour les surmonter, les chercheurs et les développeurs peuvent continuer à améliorer les capacités des CNNs et étendre leur champ d'application dans divers domaines de la vision par ordinateur et de l'intelligence artificielle.

II Conclusion Générale

Les Convolutional Neural Networks (CNN) ont révolutionné le domaine de l'intelligence artificielle, particulièrement dans le traitement et la reconnaissance d'images. Leur architecture unique, composée de couches de convolution, de pooling, de correction ReLU, et de fully-connected, leur permet de traiter efficacement des données visuelles en extrayant automatiquement des caractéristiques pertinentes.

Les CNN se distinguent par leur capacité à automatiser l'extraction des caractéristiques, ce qui élimine le besoin d'ingénierie manuelle et améliore l'efficacité des modèles. Leur invariance à la translation permet d'identifier des motifs dans les images indépendamment de leur position ou orientation. De plus, les architectures pré-entraînées peuvent être adaptées à de nouvelles tâches avec relativement peu de données, grâce au fine-tuning, ce qui montre leur grande adaptabilité.

L'utilisation des couches de convolution pour détecter les caractéristiques, des couches de pooling pour réduire la taille des images tout en préservant les informations essentielles, des couches ReLU pour normaliser les valeurs, et des couches fully-connected pour classer les images, permet aux CNN de fonctionner de manière efficace et robuste. Cela les rend particulièrement adaptés non seulement pour la classification d'images, mais aussi pour d'autres domaines comme le traitement du langage naturel et la reconnaissance vocale.

En somme, les CNN sont un outil puissant et polyvalent dans l'intelligence artificielle, capable de résoudre une grande variété de problèmes avec une haute précision et une efficacité remarquable. Leur capacité à apprendre automatiquement et à s'adapter à différents types de données fait d'eux un choix privilégié pour de nombreuses applications technologiques modernes.

III References

1. **petite documentation personnalisée dans laquelle je présente tous les aspects de mon travail**
   `https://convolutional-neural-network.readthedocs.io/en/latest/index.html`
2. **Lien vers mon repository GitHub où se trouve le code et le jeu de données "dataset" sur lesquels j'ai travaillé.**
   `https://github.com/imadmlf/Convolutional-Neural-Network`
3. **Accédez à un notebook sur Colab pour mettre en pratique tout ce qui a été expliqué dans la documentation.**
   `https://colab.research.google.com/github/imadmlf/Convolutional-Neural-Network/blob/main/CNN_simple_data.ipynb`