Évolution des Architectures CNN

ARN
8 - CNN Architecture

Abstract

Table des matières

1. Évolution des Architectures de Réseaux de Neurones Convolutionnels

1.1. Introduction et Contexte Historique

1.1.1. Le Défi ImageNet (ILSVRC)

ImageNet : Base de données massive de 15 millions d'images réparties en 22 000 catégories, étiquetées manuellement entre 2007-2010 via Amazon Mechanical Turk par 49 000 travailleurs de 167 pays.

Challenge ILSVRC : Compétition annuelle utilisant 1000 images de 1000 catégories, évaluée sur les taux d'erreur top-1 et top-5.

Performance humaine : 5% d'erreur top-5, due à la méconnaissance de certaines classes et au besoin de reconnaissance fine.

1.2. Les Architectures Pionnières

1.2.1. LeNet-5 (1989) - Les Fondations

Créateurs : Yann LeCun et collègues (AT&T Bell Labs)

Innovation majeure : Premier démonstration que des réseaux profonds entraînés par rétropropagation peuvent résoudre des problèmes de reconnaissance d'images réels sans préprocessing complexe.

Performance : Reconnaissance de codes postaux manuscrits, entraîné sur 9298 chiffres pendant 3 jours sur Sun Sparc Station 1.

Architecture : Alternance de couches convolutionnelles et de sous-échantillonnage, suivies de couches entièrement connectées.

1.2.2. AlexNet (2012) - La Révolution

Créateurs : Alex Krizhevsky, Ilya Sutskever, et Geoffrey Hinton

Résultats : Victoire écrasante ILSVRC 2012 avec 15.4% d'erreur (contre 26.2% pour le second)

Innovations techniques clés :
- **ReLU** au lieu de tanh : résolution du problème du gradient qui disparaît
- **Data augmentation** : rotation, translation, recadrage
- **Dropout** : régularisation pour éviter le surapprentissage
- **GPU training** : 62 millions de paramètres entraînés sur 2 GTX 580 pendant 5-6 jours

Architecture : 8 couches (5 convolutionnelles + 3 entièrement connectées)

1.3. L'Ère de l'Optimisation

1.3.1. ZF Net (2013) - L'Amélioration Incrémentale

Créateurs : Matthew Zeiler et Rob Fergus (NYU)

Performance : ILSVRC 2013 avec 11.2% d'erreur

Améliorations :
- Noyaux plus petits (7×7 au lieu de 11×11)
- Augmentation progressive du nombre de filtres
- Entraînement sur 1.3M images seulement

1.3.2. VGG Net (2014) - La Simplification

Créateurs : K. Simonyan et A. Zisserman (Visual Geometry Group, Oxford)

Performance : 7.3% d'erreur (n'a pas gagné mais architecture influente)

Principe fondamental : **Stricte utilisation de filtres 3×3** avec stride et padding de 1

Innovation conceptuelle : Démonstration qu'un filtre 5×5 peut être remplacé par deux couches de filtres 3×3 (25 → 18 paramètres), et un filtre 11×11 par cinq couches 3×3 (121 → 45 paramètres)

Variantes : VGG-16 (138M paramètres) et VGG-19 (143M paramètres)

Avantages :
- Réduction drastique des paramètres
- Architecture simple et reproductible
- Plus de non-linéarités avec plus de couches

1.4. Les Architectures Révolutionnaires

1.4.1. GoogLeNet/Inception (2015) - L'Innovation Parallèle

Performance : Victoire ILSVRC 2014 avec 6.7% d'erreur

Révolution architecturale : 100+ couches mais **sans couches entièrement connectées**

Module Inception : Traitement parallèle avec multiples tailles de noyaux (1×1, 3×3, 5×5) puis concaténation

Convolutions 1×1 : Innovation pour la réduction de dimensionnalité et le pooling de caractéristiques

Exemple d'efficacité : Application de 16 filtres 1×1×192 avant 32 filtres 5×5 → 12.4M multiplications au lieu de 120M

Avantages :
- Détection de caractéristiques à différentes échelles
- 12× moins de poids qu'AlexNet
- Efficacité computationnelle remarquable

1.4.2. ResNet (2015) - Les Connexions Résiduelles

Créateurs : Équipe Microsoft Research

Performance : Victoire ILSVRC 2015 avec 3.6% d'erreur (152 couches)

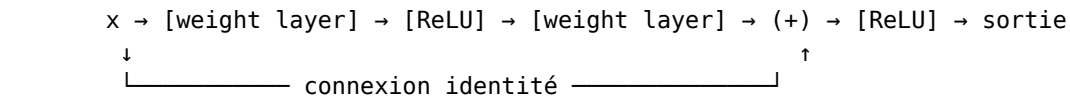
Problème résolu : Dégradation des performances lors de l'augmentation de profondeur (20 → 50+ couches)

Innovation clé : **Connexions de court-circuit (shortcut connections)**

Principe mathématique : Apprentissage de $F(x) + x$ au lieu de $H(x)$ directement

$$y = F(x, \{W_i\}) + x$$

Bloc résiduel :



Avantages :
- Permet l'entraînement de réseaux très profonds (152+ couches)
- Résolution du problème du gradient qui disparaît
- Performances supérieures aux réseaux "plain" de même complexité

1.5. Les Architectures Modernes

1.5.1. DenseNet (2017) - La Réutilisation Maximale

Créateurs : Cornell, Tsinghua Universities & Facebook

Principe révolutionnaire : Chaque couche reçoit en entrée les sorties de **toutes** les couches précédentes

Formulation : Pour la couche ℓ :

$$x_\ell = H_{\ell([x_0, x_1, \ldots, x_{\{\ell-1\}}])}$$

Avantages :
- Atténuation du problème du gradient qui disparaît
- Renforcement de la propagation des caractéristiques

- Encouragement de la réutilisation des caractéristiques
- Réduction substantielle du nombre de paramètres

Performance : Résultats similaires à ResNet avec beaucoup moins de paramètres et de calculs

1.5.2. EfficientNet (2020) - L'Optimisation Systémique

Créateurs : Mingxing Tan et Quoc Le (Google Brain)

Innovation méthodologique : Étude systématique de la mise à l'échelle des réseaux

Facteur de mise à l'échelle composé :

$$d = \alpha^\varphi, \quad w = \beta^\varphi, \quad r = \gamma^\varphi$$

où $\alpha, \beta, \gamma \geq 1$ et $\alpha \cdot \beta^2 \cdot \gamma^2 \approx 2$

Contrainte FLOPS : $\text{FLOPS} \propto d \cdot w^2 \cdot r^2 \propto 2^\varphi$

Performances remarquables :
- EfficientNet-B0 : 5× moins de paramètres et 11× moins de FLOPS que ResNet-50
- EfficientNet-B1 : 7.5× moins de paramètres et 16× moins de FLOPS que ResNet-152

1.6. Techniques Transversales

1.6.1. Batch Normalization (2015)

Créateurs : Sergey Ioffe et Christian Szegedy (Google)

Problème adressé : Instabilité due aux différentes échelles des entrées

Algorithme : Pour un mini-batch $B = \{x_1, ..., x_m\}$:

$$\mu_B \leftarrow \frac{1}{m}\sum_{i=1}^{m} x_i \quad // \text{ moyenne du mini-batch}$$

$$\sigma_B^2 \leftarrow \frac{1}{m}\sum_{i=1}^{m} (x_i - \mu_B)^2 \quad // \text{ variance du mini-batch}$$

$$\hat{x}_i \leftarrow \frac{x_i - \mu_B}{\sqrt{\sigma_B^2 + \varepsilon}} \quad // \text{ normalisation}$$

$$y_i \leftarrow \gamma \hat{x}_i + \beta \quad // \text{ mise à l'échelle et décalage}$$

Avantages :
- Accélération de la convergence
- Possibilité d'utiliser des taux d'apprentissage plus élevés
- Réduction du besoin de dropout
- Entraînement possible avec des fonctions saturantes

1.6.2. Augmentation des Données

Transformations géométriques :
- Rotation, translation, zoom, recadrage
- Retournement horizontal/vertical

Transformations colorimétriques :
- Perturbation des canaux RGB
- Modification luminosité/contraste

1.7. Impact et Évolution

1.7.1. Démocratisation du Deep Learning

Tendance observée : De 2012 à 2017, passage d'une seule équipe sous 25% d'erreur à 29 équipes sur 38 sous 5% d'erreur sur ImageNet.

Facteurs de démocratisation :
- Disponibilité des architectures pré-entraînées
- Frameworks de développement accessibles
- Augmentation de la puissance de calcul

1.7.2. Amélioration de l'Efficacité d'Entraînement

Observation OpenAI : Le calcul nécessaire pour entraîner un réseau diminue de moitié tous les 16 mois grâce à :
- Amélioration des procédures d'entraînement
- Recherche d'hyperparamètres plus efficace
- Architectures optimisées

1.8. Leçons et Principes Généraux

1.8.1. Innovations Architecturales Clés

Chronologie des innovations :
1. **2012** : ReLU, dropout, augmentation des données (AlexNet)
2. **2014** : Filtres 3×3 systématiques (VGG)
3. **2015** : Modules parallèles et convolutions 1×1 (Inception)
4. **2015** : Connexions résiduelles (ResNet)
5. **2015** : Normalisation par batch
6. **2017** : Connexions denses (DenseNet)
7. **2020** : Mise à l'échelle systémique (EfficientNet)

1.8.2. Tendances Émergentes

Efficacité computationnelle : Recherche du meilleur rapport performance/ressources

Architectures modulaires : Blocs réutilisables et composables

Automatisation : Recherche automatique d'architectures (Neural Architecture Search)

1.9. Conclusion

L'évolution des architectures CNN de LeNet à EfficientNet illustre une progression remarquable de l'ingénierie des réseaux de neurones. Cette évolution s'articule autour de plusieurs axes d'innovation :

Innovations techniques fondamentales :
- Résolution du problème du gradient qui disparaît (ReLU, ResNet)
- Optimisation de l'efficacité computationnelle (Inception, EfficientNet)
- Stabilisation de l'entraînement (Batch Normalization)
- Prévention du surapprentissage (Dropout, augmentation des données)

Impact transformateur : Ces architectures ont démocratisé l'accès aux performances de pointe en vision par ordinateur, permettant le passage d'une technologie expérimentale à un outil largement déployé dans l'industrie.

Perspective future : L'accent se déplace vers l'efficacité, la recherche automatique d'architectures, et l'adaptation à des contraintes de ressources variées, ouvrant la voie à une nouvelle génération de réseaux optimisés.