Fondamentaux du Deep learning

Louis Thiry (INRIA Rennes)

louis.thiry@outlook.fr

Formation Continue, Grenoble, 30 Juin 2022

Deep Learning

- Machine learning.
- Réseaux de neurones **profonds**.
- Différentiation automatique !
- Optimisation compliquée: instabilités, lenteur, sur-apprentissage.
- Choix délicat de nombreux hyper-paramètres
- Problèmes CSV: améliorations pas spectaculaire sur les méthodes classiques.
- **Succès en traitement un signal**

Réseaux convolutifs.

Champ réceptif d'un CNN

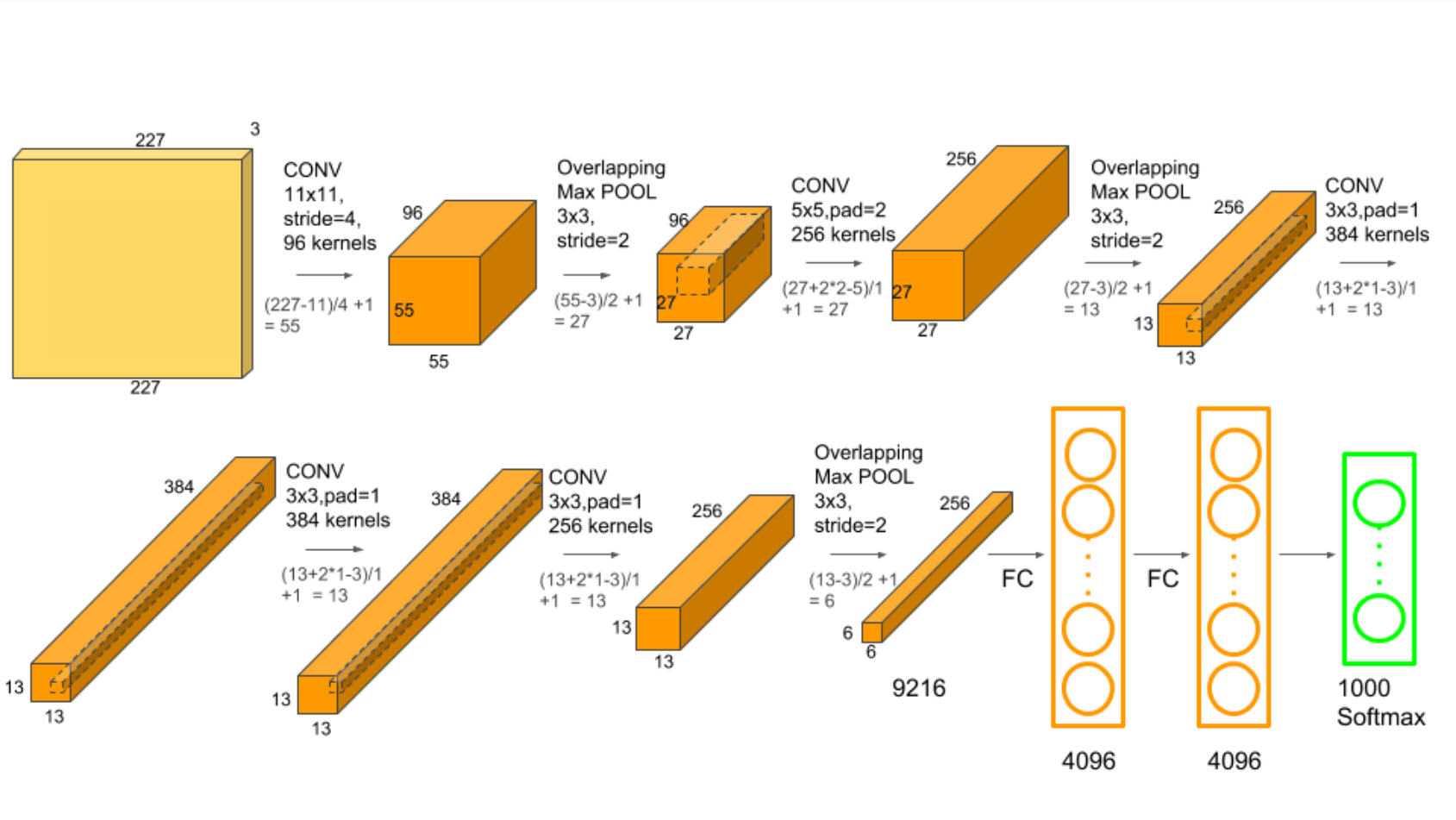


Figure 1: Architecture d'AlexNet

Champ réceptif d'un CNN

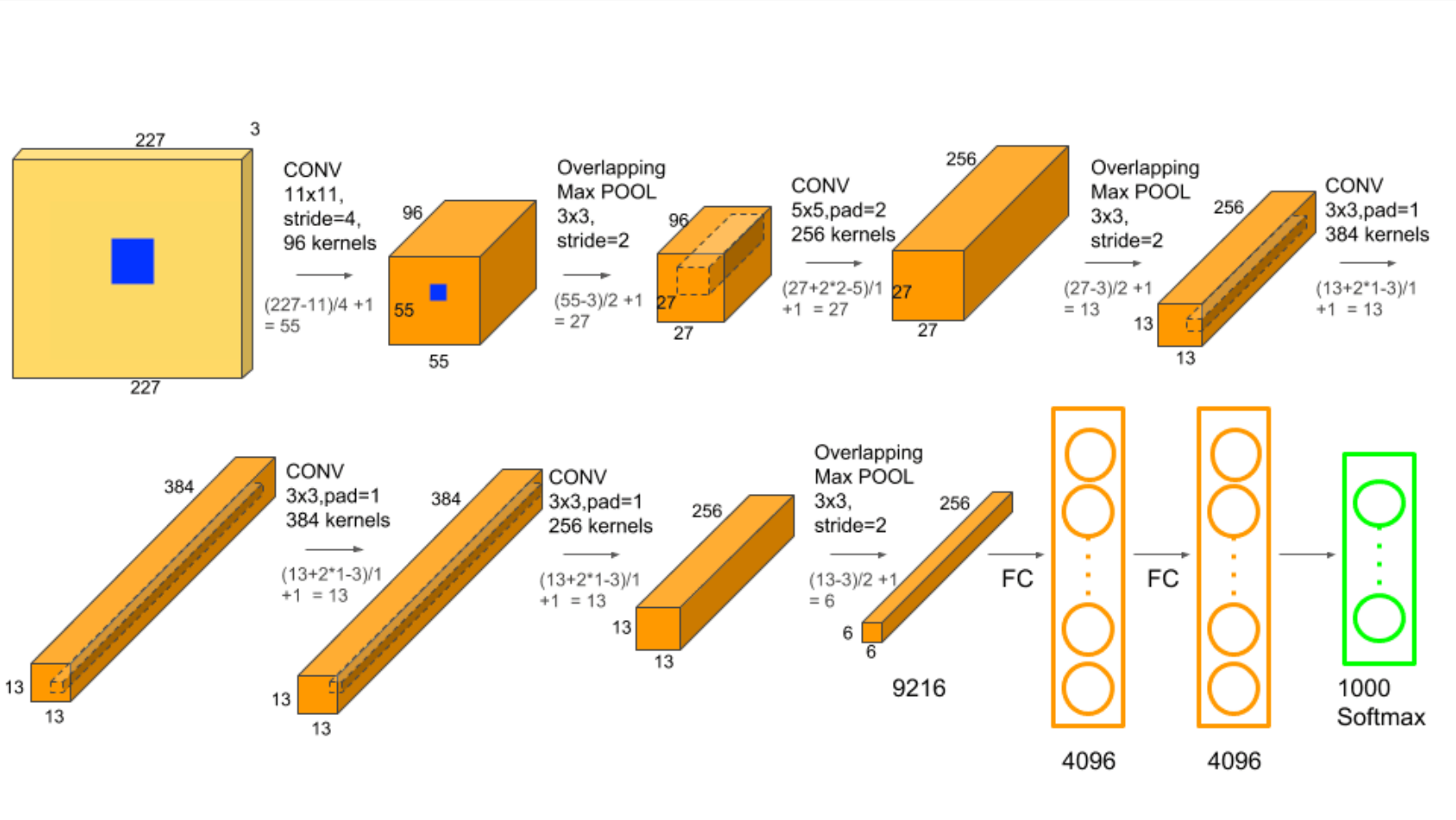


Figure 2: Architecture d'AlexNet

Champ réceptif d'un CNN

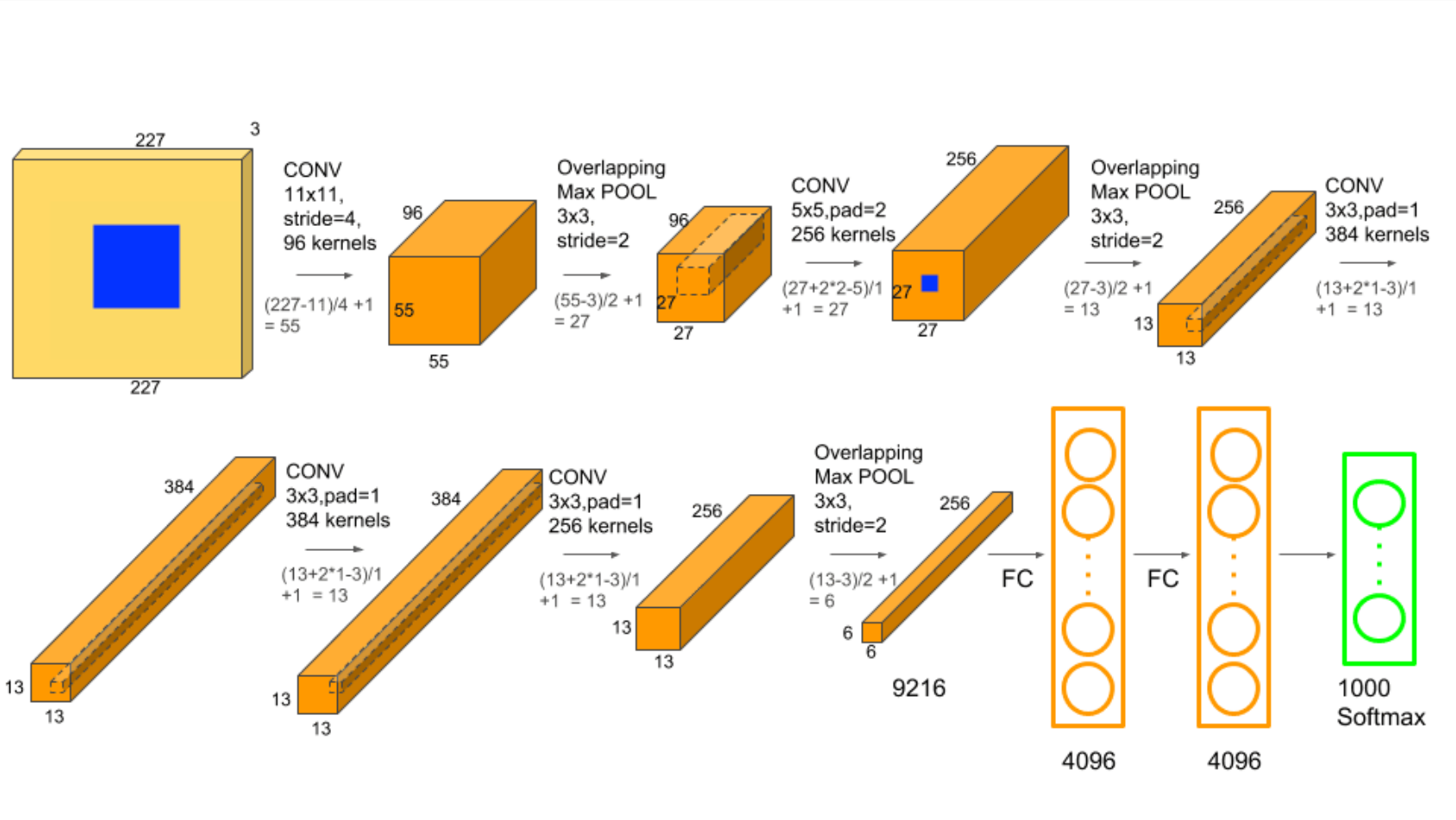


Figure 3: Architecture d'AlexNet

Champ réceptif d'un CNN

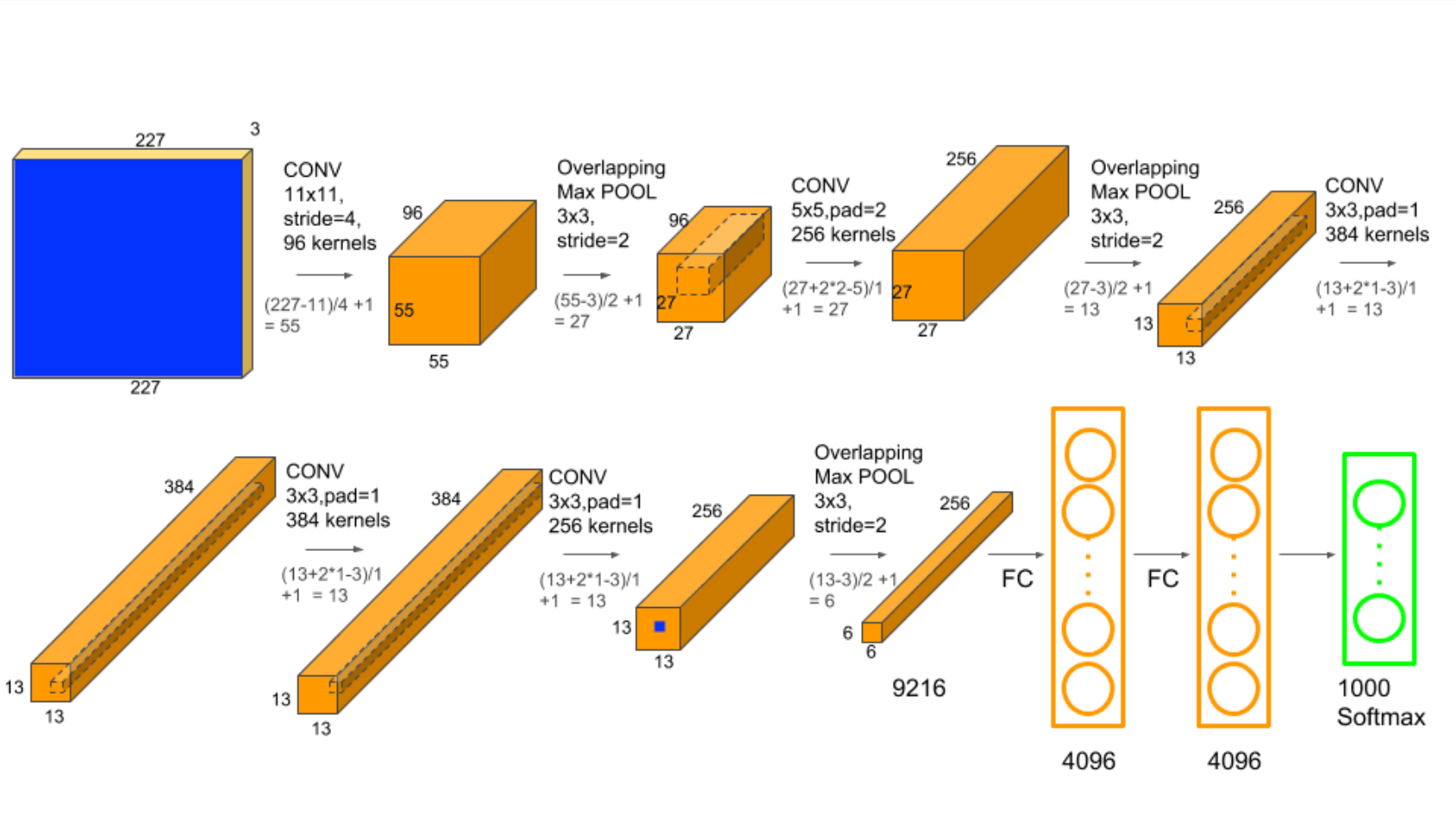


Figure 4: Architecture d'AlexNet

Fonctionnement d'un CNN

LeCun et al., 2015 Hypothèse:

- *" The first layer [of the convolutional network] represents the presence or absence of edges in the image. "*
- *" The second layer typically detects motifs"*
- *" The third layer may assemble motifs into [...] parts of familiar objects."*
- *" Subsequent layers would detect objects as combinations of these parts."*

→ **Structure hiérarchique, multi-échelle**, les premières couches sont générales, ler dernières plus *spécialisées*.

3 papiers centraux

$> 110,000$ citations.

- ImageNet Classification with Deep Convolutional Neural Networks
- Very Deep Convolutional Networks for large-scale visual recognition.
- Deep Residual Learning for Image Recognition

évolution des performances sur ImageNet

Les exemples adversariaux

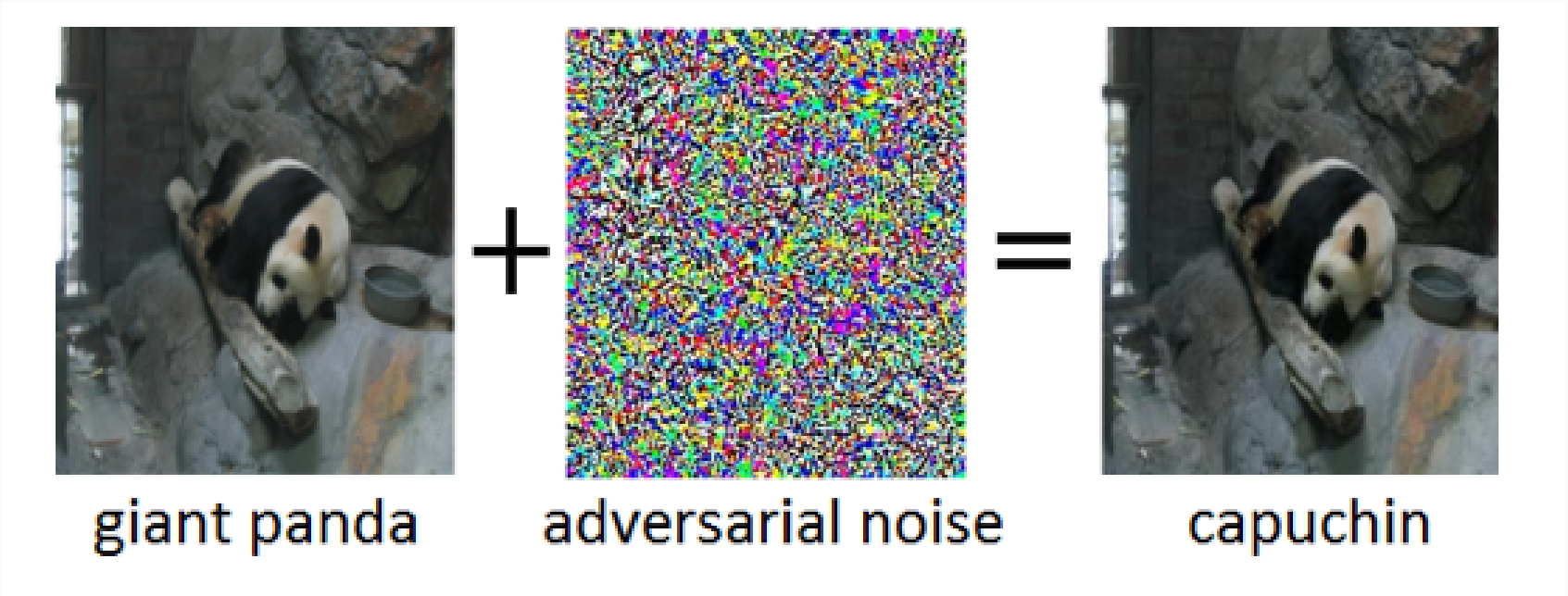


Figure 5: Image classique d'une très faible perturbation (pas de structure dans la perturbation) qui trompe un classifieur.

Performance des modèles sous attaque

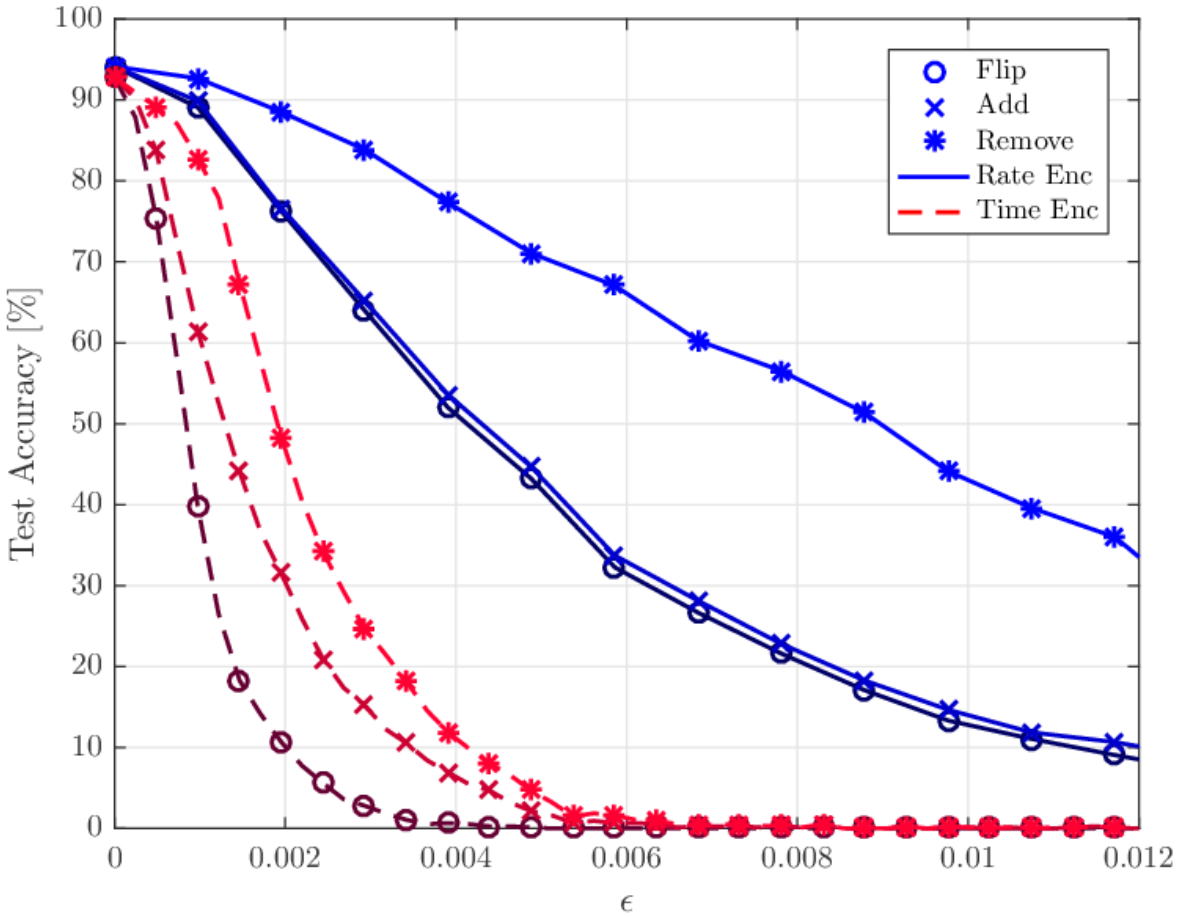


Figure 6: Evolution de la robustesse d'un modèle sous attaque (Bagheri et al., 2018). Les pixels sont entre $[0, 1]$.

Exemple: FGSM

Fast gradient sign method

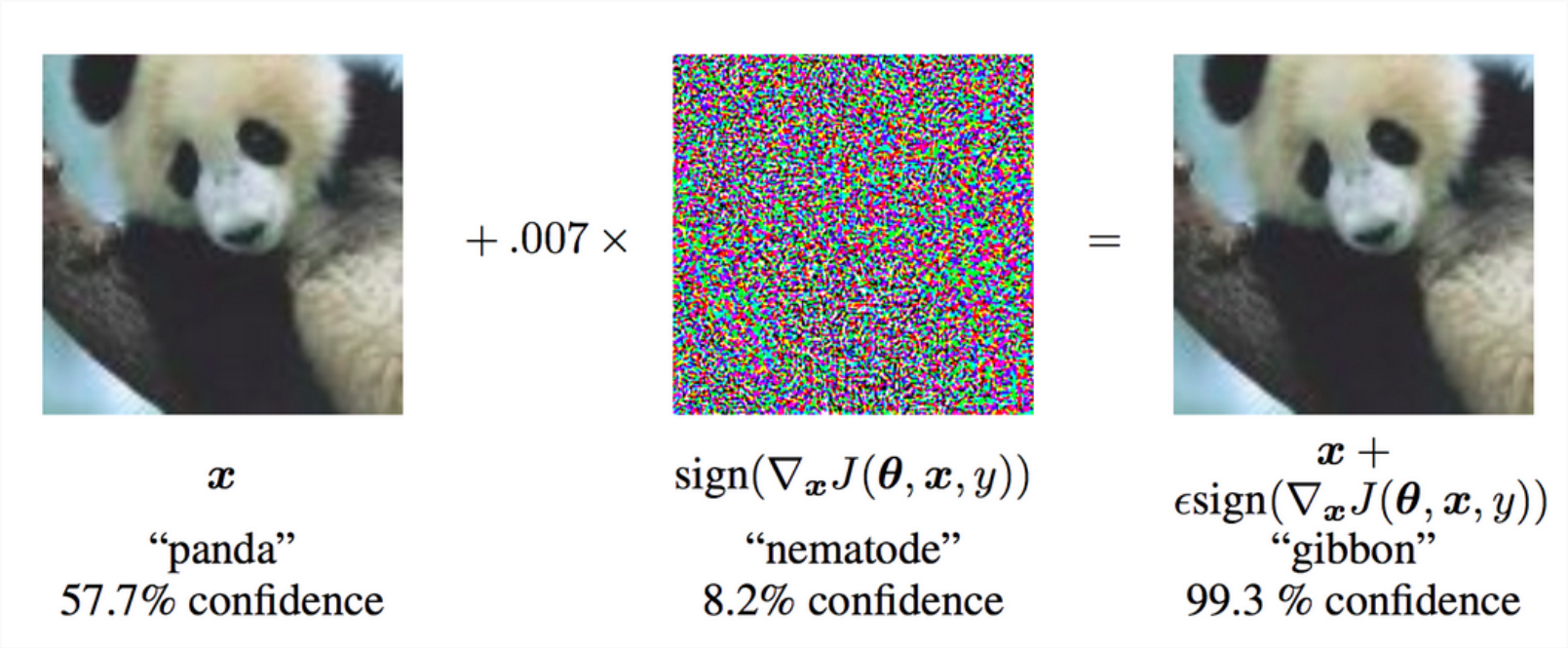


Figure 7: La méthode FGSM en pratique.

Mise en pratique sur le notebook CNN.

Utiliser des réseaux pré-entraînés

- Feature extraction: utiliser les réseaux pre-entrainés pour finalement avoir des nouvelles façons de représenter des images.
- Fine-tuning: prendre un réseau pré-entraînés et l'adapter à notre tâche spécifique.
- Utilisation détournée des modèles pre-entraînes: deep dream ou style transfer.

Feature Extraction

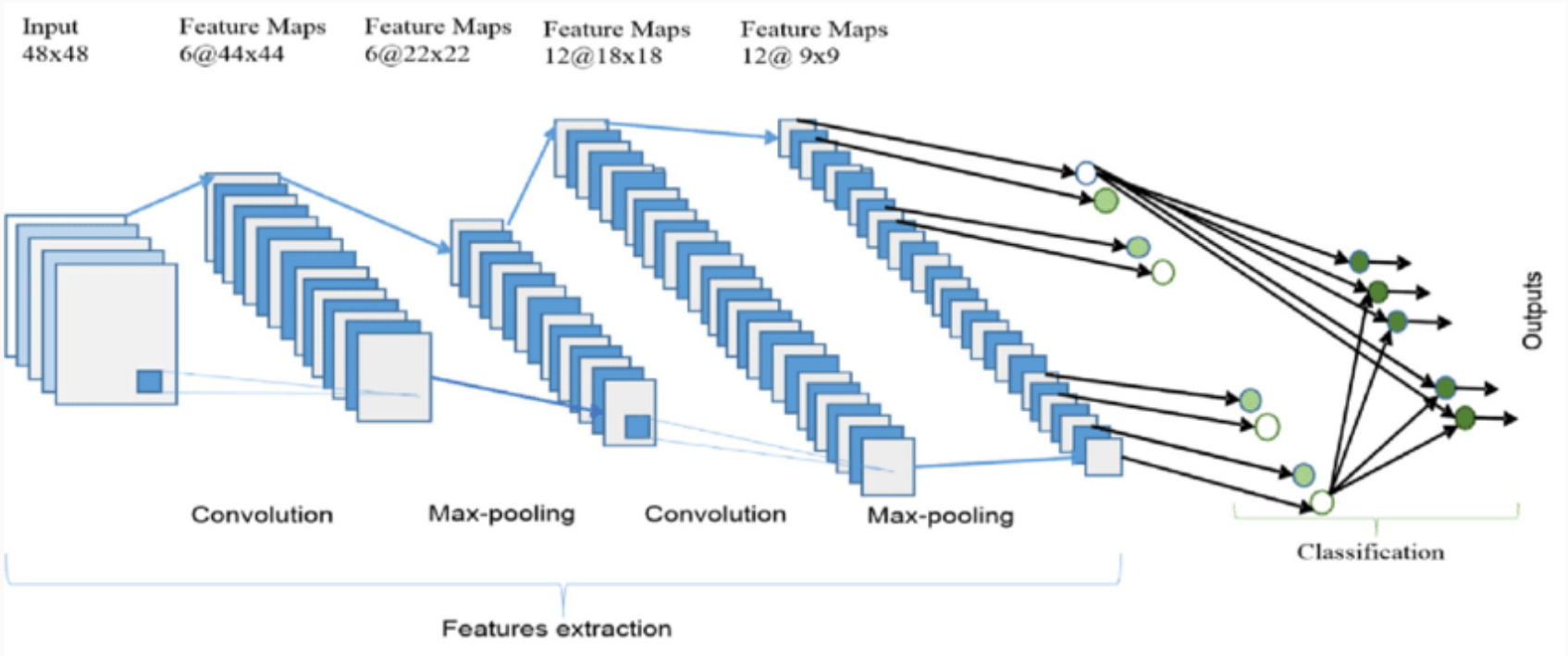


Figure 8: La partie qui est une suite de convolution et pooling est générique, peu spécifique de la tâche pour laquelle le réseau a été entrainé. C'est donc réutilisable.

Feature Extraction

Un fait important

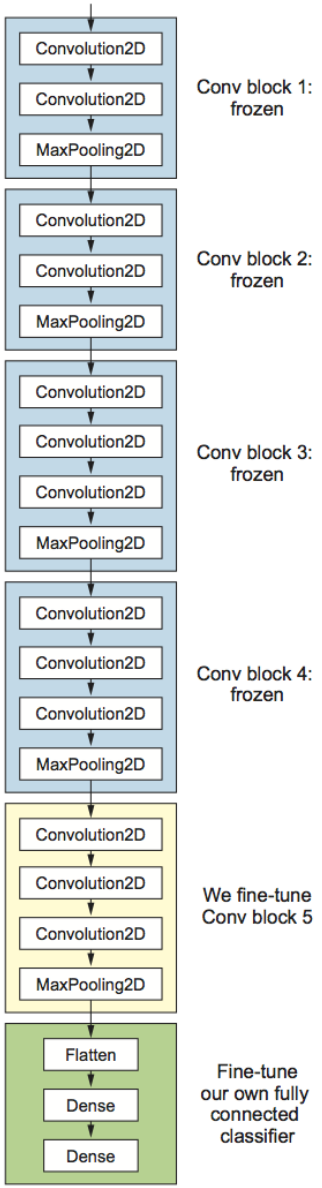
Le niveau de spécificité (à la tache) de la représentation dépend de la profondeur de la couche dans le réseau.

- Moins la couche est profonde, plus elle extrait des informations localisées et génériques sur l'image: les contours, couleur et textures.
- Plus c'est profond plus le concept est abstrait: oreille d'écureuil.

Heuristique

En fonction de la similarité entre le dataset de pre-entrainement et celui d'entraînement on juge de la bonne profondeur à laquelle pratiquer l'extraction.

Fine-tuning



- Etape après la feature extraction.
- On entraine simultanéement le classifieur et les couches les plus profondes de convbase.
- Attention à l'ordre des opérations!

Fine-tuning

- Les couches les plus profondes de convbase contiennent des informations plus spécifiques au jeu de donnée sur lesquel le modèle a été pre-entrainé.
- La quantité de couches de convolution fine-tuned joue son rôle pour prévenir l'overfitting.
- Le fine-tuning se fait avec un learning rate petit pour éviter de trop perturber l'équilibre.

Une commodité

Coder en deep learning est devenu *facile*

- Les logiciels comme *PyTorch* sont *stables*, avec plusieurs années de retour sur expérience et beaucoup d'utilisateurs.

- Pleins de bases de données, de réseaux pre-entrainés, plus besoin de passer des années à *préparer les données* pour commencer à avoir des résultats.

- **Best practice:** Prendre un code qui fonctionne sur github et le modifier à votre guise.

- Accès aux GPU de plus en plus facile.

Quelques limitations

- Pas pour les problèmes *critiques*.
- Beaucoup d'énergie (en embarqué).
- Beaucoup de mémoire (en embarqué) (Iandola et al., 2016; Howard et al., 2017).

Hardware

Trois composantes importantes d'un ordinateur: RAM, CPU et GPU.

- CPU: Central Processing Unit. Classique et developpé par Intel, ca fait un peu toutes les tâches.
- GPU: Graphics Processing Unit. C'est spécialisé à faire beaucoup de calculs simples en parallèle. Pour les jeux videos puis maintenant le deep.
- RAM: Random Access Memory: La mémoire vive, "ce dont on peut parler sans avoir à réflechir". En deep: paramètres du réseau + base de données.
- Plus récemment ASIC (Application Specific Integrated Circuit), TPU (Tensor Processing Unit) : tout est dans le nom!

Tensor 1.0: Efficacité des GPU

- Matrice = Tenseur à 2 dimensions. Tenseur = stockage de nombres organisé en dimensions.
- Deep Learning = beaucoup de produits de tenseurs en parallèle, e.g., convolution = produits de tenseurs.
- Les GPUs sont efficaces pour des faire des calculs simples en parallèle, e.g. produit de matrices.

Important de regarder sa base de données.

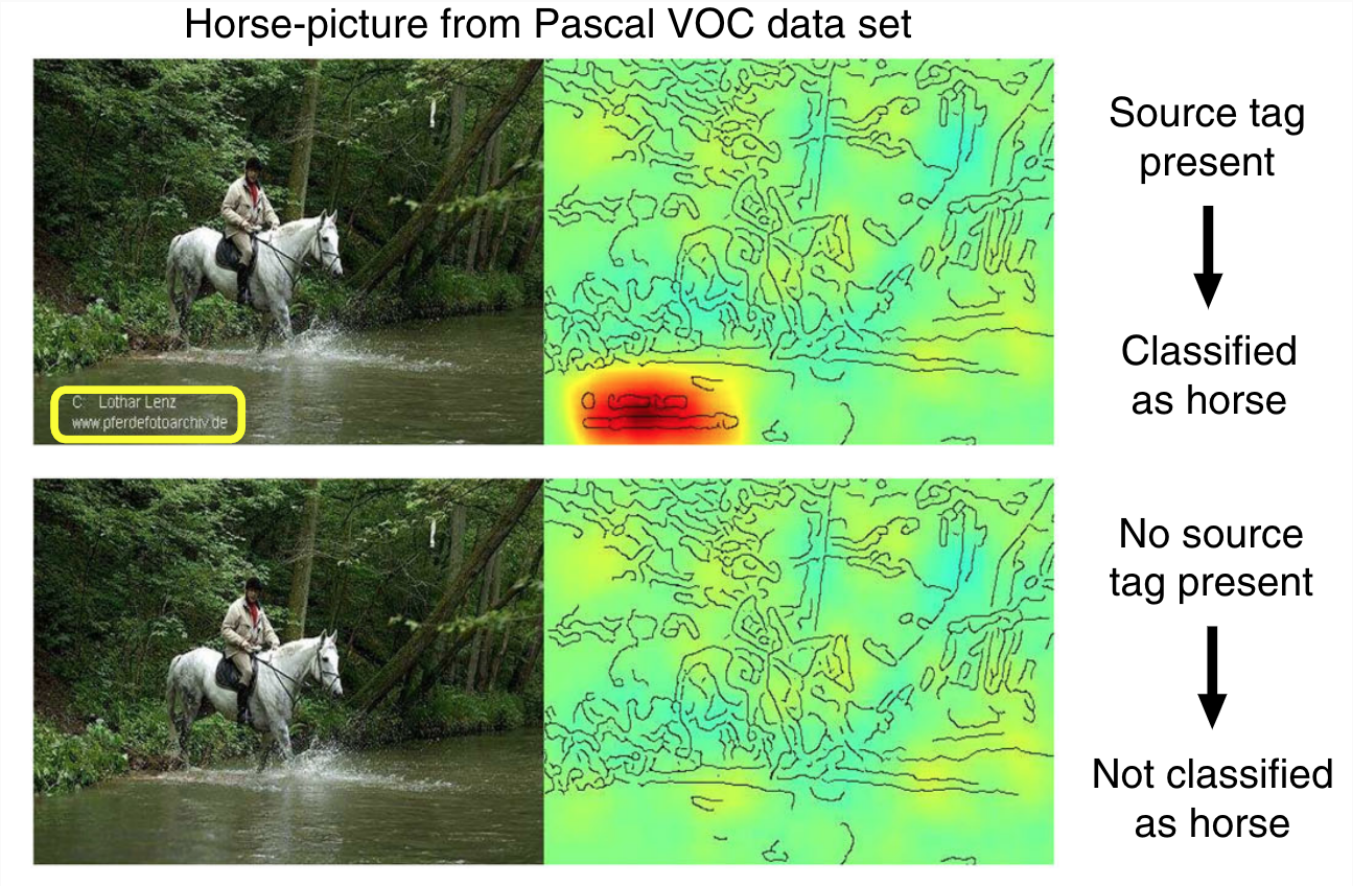
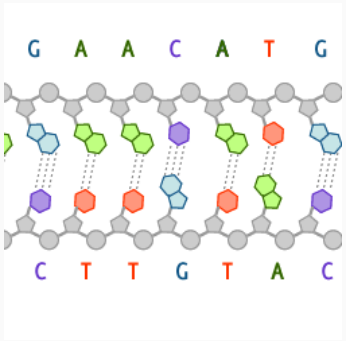
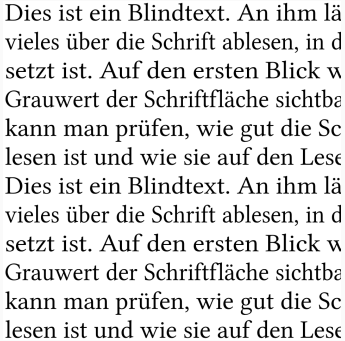


Figure 9: Pour Pascal VOC, c'est en regardant les pixels importants pour la classification que (Lapuschkin et al., 2019) ont remarqués que des *water-mark* étaient présentes pour les photos de cheval uniquement.

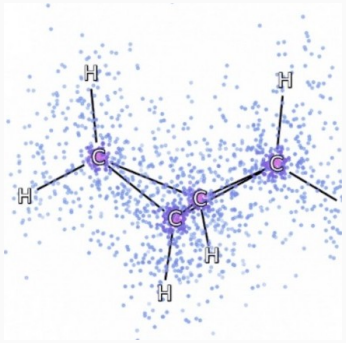
Les Réseaux de Neurones: pas que de l'image.



(a) ADN


Dies ist ein Blindtext. An ihm lä
vieles über die Schrift ablesen, in d
setzt ist. Auf den ersten Blick w
Grauwert der Schriftfläche sichtba
kann man prüfen, wie gut die Sc
lesen ist und wie sie auf den Lese
Dies ist ein Blindtext. An ihm lä
vieles über die Schrift ablesen, in d
setzt ist. Auf den ersten Blick w
Grauwert der Schriftfläche sichtba
kann man prüfen, wie gut die Sc
lesen ist und wie sie auf den Lese

(b) Texte



(c) Atome

(d) Audio

- Fonctionne pour tout type de donnée *homogène* et où beaucoup d'exemples sont disponibles (e.g. romans).
- Résoudre les équations fondamentales de la physique (Pfau et al., 2020).
- Le repliement des protéines: prometteur, ici.

Mise en pratique

Le *Deep Learning* comporte une forte dimension expérimentale. La pratique est indispensable pour saisir les notions.

- Former des groupes de 2 a 4
- Trouver des données de type CSV ou image: *Best Public Datasets for Machine Learning*
- Les transformer en nombres si-besoin: pandas.get_dummies.
- D'abord une méthode simple: régression linéaire.
- Améliorer avec feature engineering et réseaux de neurones
- Présentez vos résultats sous forme d'un notebook.

Pour aller plus loin

- Papier BagNet sur la classification a base de patches
- Biais texture / forme des réseaux de neurones convolutionels sur ImageNet.
- Les exemple adversariaux
- La cheat-sheet de Scikit-Learn