ML Projet

Isai Gordeev

École Polytechnique

20 avril 2023

Plan

ML ou pas ?

Vous êtes sûr en emploi de ML ?
Votre problème peut être résolu **algorithmiquement**

Oui, je suis sûr et alors quoi ?

ML essaie de trouver une fonction f tel que

$$f(X) = y$$

$$f(X) := f^L(W^L f^{L-1}(W^{L-1} \cdots f^1(W^1 x) \cdots))$$

où X est notre attributs, y est caractéristique

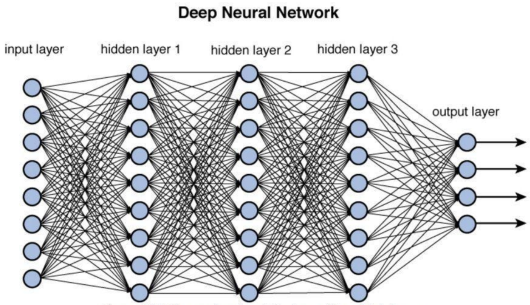
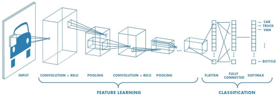
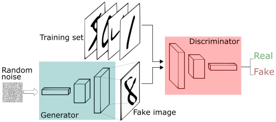


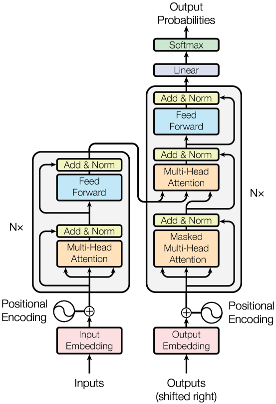
Figure 12.2 Deep network architecture with multiple layers.

Architecture

- Réseau neuronal convolutif (CNN) – le traitement des images
- Réseaux antagonistes génératifs (GAN) - la génération des objets
- Transformer – l'architecture d'utilisation générale

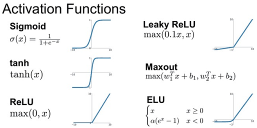


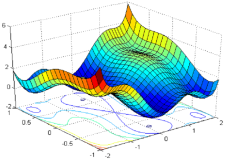




Metaparaméters

- Le réseau
  - Les fonctions d'activations
  - Les couches
  - Autres
- Vraisemblance
  - La fonction de perte $L(W, y, \hat{y})$
  - Régularization
  - Autres





L1 Regularization

$$\text{Cost} = \sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2 + \lambda \sum_{j=0}^{M} |W_j|$$

L2 Regularization

$$\text{Cost} = \underbrace{\sum_{i=0}^{N} (y_i - \sum_{j=0}^{M} x_{ij} W_j)^2}_{\text{Loss function}} + \underbrace{\lambda \sum_{j=0}^{M} W_j^2}_{\text{Regularization Term}}$$

Données

- Il faut collecter les données (ou notre tensor X)
- Il fait éviter les attributs dépendants
- Pour l'apprentissage 80% de données
- Pour la validation 20% de données

Par exemple on a n image de chat noir
Il faut dire si une image contient un chat noir

Apprentissage

1. Rétropropagation

$$
\begin{aligned}
\delta^1 &= (f^1)' \circ (W^2)^T \cdot (f^2)' \circ \cdots \circ (W^{L-1})^T \cdot (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \\
\delta^2 &= (f^2)' \circ \cdots \circ (W^{L-1})^T \cdot (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \circ \nabla_{a^L} C \\
&\vdots \\
\delta^{L-1} &= (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \circ \nabla_{a^L} C \\
\delta^L &= (f^L)' \circ \nabla_{a^L} C,
\end{aligned}
$$

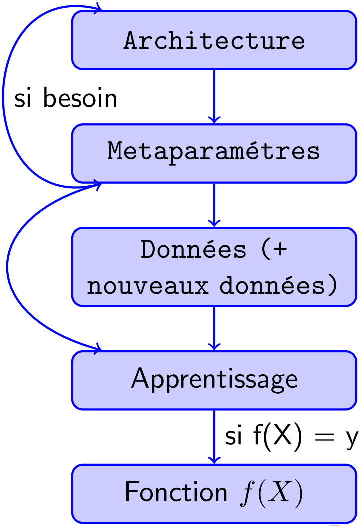
2. Algorithme du gradient stochastique

$$
W^i = W^i - \lambda \nabla L(W, y, \hat{y})_{W^i} = W^i - \lambda \delta^i_W
$$

$$
L(W, y, \hat{y}) = arg(min_X L(W, y, \hat{y}))
$$

Réglage

On maintient le modèle en répétant les étapes 3 - 5 (2 - 5)

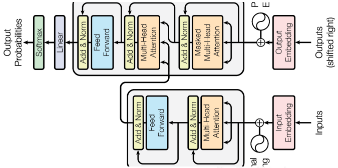


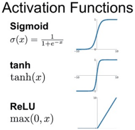
Résultat

On a obtenu la fonction f permettant prévoir pour un élément X_i de X son caractéristique et la façon de la généraliser et maintenir

$$f(X_i) = y_i$$

Bilan





Deep Neural Network

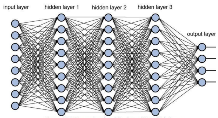


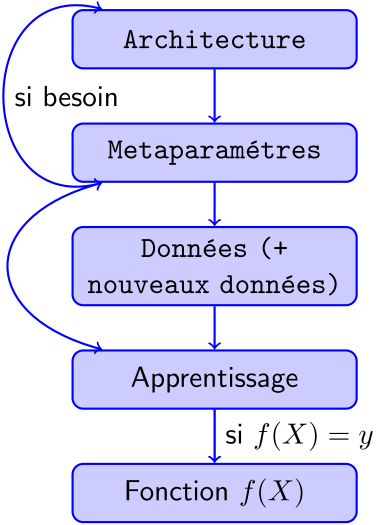
Figure 12.2 Deep network architecture with multiple layers.

1. Rétropropagation

$$\delta^1 = (f^1)' \circ (W^2)^T \cdot (f^2)' \circ \cdots \circ (W^{L-1})^T \cdot (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \circ$$
$$\delta^2 = (f^2)' \circ \cdots \circ (W^{L-1})^T \cdot (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \circ \nabla_{a^L} C$$
$$\vdots$$
$$\delta^{L-1} = (f^{L-1})' \circ (W^L)^T \cdot (f^L)' \circ \nabla_{a^L} C$$
$$\delta^L = (f^L)' \circ \nabla_{a^L} C,$$

2. Algorithme du gradient stochastique

$$W^i = W^i - \lambda \nabla L(W, y, \hat{y})_{W^i} = W^i - \lambda \delta^i_W$$
$$L(W, y, \hat{y}) = arg(min_X L(W, y, \hat{y}))$$



Merci pour votre attention