Polycopié sur l'Introduction aux Réseaux de Neurones et au Deep Learning

Basé sur les slides de l'IntroNN 2025

Table des matières

1 Introduction et Concepts Généraux

1.1 Définition de l'Apprentissage Automatique

Définition 1.1 (Machine Learning)**.** *Un programme informatique est dit apprendre* ***si et seulement si*** *:*

Pour une expérience E, une tâche T et une mesure de performance P, la performance de P s'améliore avec

Cette définition, proposée par Tom Mitchell (1997), met l'accent sur l'amélioration progressive via l'expérience.

1.2 Types d'Apprentissage

- **Apprentissage Supervisé** : Chaque point de données $\mathbf{x}_i \in \mathbb{R}^d$ est associé à une étiquette y_i. L'objectif est de trouver une fonction f telle que $y_i \approx f(\mathbf{x}_i)$.
- **Apprentissage Non Supervisé** : Les données ne sont pas étiquetées. L'objectif est de découvrir une structure sous-jacente (exemple : clustering).
- **Autres variantes** : Semi-supervisé, apprentissage par renforcement, etc.

2 Le Perceptron et son Algorithme d'Apprentissage

2.1 Définition du Perceptron

Définition 2.1 (Perceptron)**.** *Un perceptron est un classificateur linéaire défini par :*

$$f(\mathbf{x}; \theta) = \text{sign}(\mathbf{w}^\top \mathbf{x} + b),$$

où $\theta = \{\mathbf{w}, b\}$ représente l'ensemble des paramètres.

2.2 Algorithme d'Apprentissage du Perceptron

Mise à jour des poids : Pour un exemple $(\mathbf{x}_i, y_i)$, si la prédiction est incorrecte, la mise à jour est effectuée par :

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} + \eta\, y_i\, \mathbf{x}_i,$$

où $\eta > 0$ est le taux d'apprentissage.

2.3 Théorème de Convergence du Perceptron

Proposition 2.2 (Convergence du Perceptron)**.** *Si les données sont* ***linéairement séparables****, alors l'algorithme du perceptron converge en un nombre fini d'itérations.*

Esquisse de démonstration. Soit $\mathbf{w}^*$ un vecteur séparateur tel que pour tout i,

$$y_i \left(\mathbf{w}^* \cdot \mathbf{x}_i\right) \geq \gamma \|\mathbf{w}^*\| \quad (\gamma > 0),$$

et supposons qu'il existe $R > 0$ tel que $\|\mathbf{x}_i\| \leq R$ pour tout i. On peut montrer par récurrence que la norme $\|\mathbf{w}^{(t)}\|$ croît d'une quantité bornée, ce qui permet d'obtenir une borne supérieure sur le nombre d'itérations (proportionnelle à R^2/γ^2). □

2.4 Limites du Perceptron

Remarque 2.3. *Le perceptron présente deux limites principales :*

1. ***Problèmes non linéaires** : Il ne peut résoudre des problèmes non linéairement séparables (exemple classique : XOR).*
2. ***Qualité de la frontière** : Même en présence de séparabilité linéaire, la frontière obtenue n'est pas nécessairement optimale (optimisation de la marge, par exemple dans les SVM).*

3 Fonctions de Perte et Apprentissage Probabiliste

3.1 Fonctions de Perte

Pour mesurer la qualité d'un classificateur, plusieurs fonctions de perte sont utilisées :

— **Zero-One Loss** :

$$\ell_{0-1}(y, \hat{y}) = \mathbf{1}_{\{y \neq \hat{y}\}}.$$

— **Hinge Loss** (utilisée notamment pour les SVM) :

$$\ell_{\text{hinge}}(y, f(\mathbf{x})) = \max(0, 1 - y \cdot f(\mathbf{x})).$$

— **Mean Squared Error (MSE)** :

$$\ell_{\text{MSE}}(y, f(\mathbf{x})) = (y - f(\mathbf{x}))^2.$$

3.2 Modèle Probabiliste et Sigmoïde

Pour obtenir une interprétation probabiliste, on remplace la fonction sign par la fonction sigmoïde :

$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$

Ainsi, la prédiction devient :

$$\hat{y} = f(\mathbf{x}; \theta) = \sigma(\mathbf{w}^\top \mathbf{x} + b) \in [0, 1],$$

que l'on peut interpréter comme une probabilité.

3.3 Maximum de Vraisemblance (MLE)

Pour un jeu de données $\{(\mathbf{x}_i, y_i)\}_{i=1}^n$, la vraisemblance conditionnelle du modèle est donnée par :

$$L(\theta) = \prod_{i=1}^{n} [f(\mathbf{x}_i; \theta)]^{y_i} [1 - f(\mathbf{x}_i; \theta)]^{1-y_i} .$$

La log-vraisemblance s'écrit alors :

$$\log L(\theta) = \sum_{i=1}^{n} [y_i \log f(\mathbf{x}_i; \theta) + (1 - y_i) \log(1 - f(\mathbf{x}_i; \theta))] .$$

La minimisation du négatif de cette log-vraisemblance permet de déterminer les paramètres optimaux.

4 Apprentissage par Gradient

4.1 Descente de Gradient

Pour minimiser la fonction de perte $\ell(\theta)$, on utilise la descente de gradient :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \ell(\theta),$$

où η est le taux d'apprentissage.

4.2 Descente de Gradient Stochastique (SGD)

Afin de réduire le coût computationnel, on peut utiliser des mini-batchs de données :

$$\theta^{(t+1)} = \theta^{(t)} - \eta \nabla_\theta \ell_{\text{mini-batch}}(\theta).$$

4.3 Calcul des Gradients

Pour le modèle probabiliste, on calcule la dérivée de la fonction de perte par rapport aux paramètres en utilisant la règle de la chaîne. Par exemple, pour un poids w_j :

$$\frac{\partial \ell}{\partial w_j} = \sum_{i=1}^{n} (f(\mathbf{x}_i; \theta) - y_i) \, x_{ij}.$$

Ce calcul est généralisé de façon vectorielle dans le cadre de l'optimisation.

5 Réseaux de Neurones à Couches Multiples (MLP)

5.1 Architecture d'un MLP

Un réseau de neurones à couches multiples (MLP) se compose de :

1. **La couche d'entrée** : reçoit le vecteur $\mathbf{x}$.
2. **Les couches cachées** : appliquent des transformations non linéaires.
3. **La couche de sortie** : produit la prédiction finale.

5.2 Modélisation d'une Couche Cachée

La sortie d'une couche cachée est donnée par :

$$\mathbf{h}^{(1)} = \phi\Big(\mathbf{W}^{(1)}\mathbf{x} + \mathbf{b}^{(1)}\Big),$$

où ϕ est une fonction d'activation (sigmoïde, tanh, ReLU, etc.).

5.3 La Fonction Softmax pour la Classification Multi-classes

Pour un problème à K classes, la couche de sortie utilise la fonction softmax :

$$\hat{o}_k = \frac{e^{h_k^{(l)}}}{\sum_{j=1}^{K} e^{h_j^{(l)}}}, \quad k = 1, \ldots, K,$$

avec $\sum_{k=1}^{K} \hat{o}_k = 1$. Cela permet d'interpréter les sorties comme des probabilités.

6 Optimisation et Rétropropagation

6.1 Principe de la Rétropropagation

La rétropropagation permet de calculer efficacement les gradients de la fonction de perte par rapport à tous les paramètres du réseau via la règle de la chaîne.

6.2 Schéma de Calcul du Gradient

Pour un réseau à une couche cachée, le gradient d'un poids w s'exprime par :

$$\frac{\partial \ell}{\partial w} = \frac{\partial \ell}{\partial o} \cdot \frac{\partial o}{\partial h} \cdot \frac{\partial h}{\partial w}.$$

En pratique, ces dérivées sont calculées de manière vectorisée, en utilisant des produits matriciels (jacobiennes).

6.3 Exemple Simple de Backpropagation

Considérons un réseau simple avec :

$$z = f(x, y) = (x + y)z.$$

Le calcul des gradients s'effectue en deux phases :

— **Phase Forward** : Calcul de la sortie f à partir des entrées.
— **Phase Backward** : Calcul des dérivées partielles $\frac{\partial f}{\partial x}$, $\frac{\partial f}{\partial y}$ via la règle de la chaîne.

7 Capacité de Représentation et Théorème d'Approximation Universelle

7.1 Théorème d'Approximation Universelle

Théorème 7.1 (Approximation Universelle)**.** *Un MLP à une seule couche cachée, avec un nombre suffisant de neurones et une fonction d'activation non linéaire, peut approcher toute fonction continue définie sur un compact avec une précision arbitraire.*

7.2 Interprétation

Ce théorème, démontré notamment par Hornik et al. (1989), justifie la capacité expressive des réseaux de neurones, bien que le théorème ne fournisse pas de méthode pour déterminer le nombre optimal de neurones.

8 Sélection de Modèle et Choix d'Hyperparamètres

8.1 Sous-Apprentissage et Sur-Apprentissage

— **Sous-apprentissage (Underfitting)** : Le modèle ne parvient pas à capturer la structure des données (capacité insuffisante).
— **Sur-apprentissage (Overfitting)** : Le modèle s'adapte trop aux données d'entraînement et perd en capacité de généralisation.

8.2 Validation Croisée et Séparation des Données

La stratégie classique consiste à diviser les données en trois ensembles :

1. **Entraînement** : Pour ajuster les paramètres.
2. **Validation** : Pour sélectionner les hyperparamètres.
3. **Test** : Pour évaluer la performance finale.

9 Régularisation et Initialisation

9.1 Régularisation

Afin d'éviter le sur-apprentissage, on ajoute un terme de régularisation dans la fonction de perte, par exemple le weight decay :

$$\ell_{\text{total}}(\theta) = \ell(\theta) + \lambda \|\theta\|^2,$$

où λ contrôle la force de la régularisation.

9.2 Initialisation des Paramètres

L'initialisation des poids est cruciale pour la convergence de l'optimisation. Deux stratégies courantes sont :

— **Initialisation aléatoire selon une loi normale** : $w \sim \mathcal{N}(0, \frac{1}{d})$.
— **Initialisation uniforme** dans un intervalle dépendant de la taille des couches.

10 Avantages et Difficultés des Réseaux de Neurones

10.1 Avantages

— **Flexibilité** : Adaptables à divers types de données (images, texte, etc.).
— **Capacité Expressive** : Grâce au théorème d'approximation universelle, un MLP peut modéliser des fonctions très complexes.
— **Optimisation Efficace** : La rétropropagation et le SGD permettent de traiter de très grands jeux de données.

10.2 Difficultés

— **Optimisation Non Convexe** : Le paysage de la fonction de perte comporte de nombreux minima locaux.
— **Problèmes de Gradient** : Apparition du phénomène de *vanishing* ou de *gradients saturés* dans les réseaux profonds.
— **Sélection d'Hyperparamètres** : Le choix du nombre de couches, de neurones, du taux d'apprentissage, etc., repose souvent sur des expérimentations empiriques.

11 Conclusion

Ce polycopié a pour but de synthétiser les concepts essentiels abordés dans les slides d'introduction aux réseaux de neurones et au deep learning. Nous avons ainsi couvert :

— La définition et les concepts fondamentaux de l'apprentissage automatique.
— Le modèle du perceptron, son algorithme d'apprentissage et sa convergence.
— Les fonctions de perte et l'approche probabiliste via le MLE.
— Les méthodes d'optimisation par descente de gradient et rétropropagation.
— L'architecture des réseaux de neurones à couches multiples et le théorème d'approximation universelle.
— La sélection de modèle, la régularisation et l'initialisation des paramètres.
— Les avantages et les difficultés inhérents aux réseaux de neurones.

Ce document, rédigé dans un style mathématique, permet d'avoir une vue d'ensemble complète et détaillée des thèmes abordés dans les slides. Il peut être enrichi et adapté selon les besoins d'approfondissement.