

1. Sujet du projet

- **Nom du projet** : AI
- **Auteurs** : ISSAD Lisa

ACHAB Ouardia

DE ANGELIS enzo

Date : 25/01/2025

2. Introduction au projet

- **Objectif** :
 - Le projet a pour but de simuler les principes fondamentaux de l'intelligence artificielle (IA) afin d'éduquer les utilisateurs intéressés par ce domaine.
 - Il s'agit de créer un environnement de simulation qui illustre des solutions d'IA typiques, telles que l'algorithme A*, les réseaux de neurones, etc.
- **Motivations** :
 - L'intelligence artificielle est un domaine en pleine expansion qui influence de nombreux secteurs.
 - Ce projet vise à rendre ces concepts accessibles à un public plus large, en fournissant une plateforme interactive pour l'apprentissage.

3. Spécification du projet

3.1 Contexte et Justification

L'intelligence artificielle (IA) est devenue un domaine incontournable dans le monde moderne, touchant des secteurs variés tels que la santé, les transports, la finance et bien d'autres. Cependant, la complexité de ses concepts peut constituer un obstacle pour les novices. Ce projet vise à créer un environnement de simulation qui permet aux utilisateurs de visualiser et d'interagir avec des algorithmes d'IA fondamentaux. En rendant ces concepts accessibles, nous espérons susciter l'intérêt et encourager l'apprentissage dans ce domaine fascinant.

3.2 Objectifs du projet

Les objectifs principaux de ce projet sont les suivants :

1. **Éducation** : Fournir une plateforme d'apprentissage interactive pour les utilisateurs souhaitant découvrir les principes de l'IA.
2. **Visualisation** : Illustrer le fonctionnement des algorithmes d'IA à travers des simulations visuelles.
3. **Interactivité** : Permettre aux utilisateurs de modifier les paramètres des algorithmes et d'observer les effets en temps réel.
4. **Accessibilité** : Créer une interface utilisateur intuitive qui facilite l'apprentissage.

3.3 Algorithmes à Simuler

3.3.1 Algorithme A*

L'algorithme A* est un algorithme de recherche de chemin qui est largement utilisé dans les jeux vidéo et les applications de navigation. Il combine les avantages de la recherche en largeur et de la recherche en profondeur, en utilisant une heuristique pour guider la recherche vers la solution optimale.

- **Fonctionnement :**

- L'algorithme A* utilise une fonction de coût qui combine le coût du chemin parcouru et une estimation du coût restant pour atteindre la destination.
- Les utilisateurs pourront visualiser le processus de recherche, y compris les nœuds explorés et le chemin final trouvé.

- **Applications :**

- Navigation GPS, jeux vidéo, robotique.

3.3.2 Perceptron

Le perceptron est un modèle fondamental dans le domaine de l'apprentissage automatique et constitue la base des réseaux de neurones. Il a été introduit par Frank Rosenblatt dans les années 1950 et est souvent utilisé pour des tâches de classification binaire. Le perceptron est un classificateur linéaire qui prend des décisions en fonction d'une combinaison linéaire de ses entrées.

Fonctionnement du Perceptron

1. Structure du Perceptron :

- Un perceptron est composé de plusieurs entrées (features), chacune associée à un poids. Les entrées peuvent représenter des caractéristiques d'un ensemble de données, comme la taille, le poids, ou d'autres attributs pertinents.
- Le perceptron a également un biais, qui est un paramètre supplémentaire permettant d'ajuster la sortie indépendamment des entrées.

2. Calcul de la Sortie :

- La sortie du perceptron est calculée en effectuant une somme pondérée des entrées : $[z = w_1x_1 + w_2x_2 + \dots + w_nx_n + b]$ où (w_i) sont les poids, (x_i) sont les entrées, et (b) est le biais.
- Ensuite, cette somme est passée à travers une fonction d'activation, généralement une fonction de seuil (step function), qui détermine si le neurone doit s'activer ou non. Si (z) dépasse un certain seuil, la sortie est 1 (ou "activée"), sinon elle est 0 (ou "non activée").

3. Apprentissage :

- Le perceptron apprend à partir d'exemples d'entraînement. Lors de l'entraînement, il ajuste les poids et le biais en fonction des erreurs de classification.

- L'algorithme d'apprentissage du perceptron utilise la règle de mise à jour des poids : $w_i = w_i + \eta \cdot (y - y^{\wedge}) \cdot x_i$ est le taux d'apprentissage, (y) est la sortie réelle, et y^{\wedge} est la sortie prédite.

Applications du Perceptron

- **Classification Binaire** : Le perceptron est principalement utilisé pour des tâches de classification binaire, où il doit décider entre deux classes. Par exemple, il peut être utilisé pour classer des emails comme "spam" ou "non spam".
- **Reconnaissance de Formes** : Dans des applications simples de reconnaissance de formes, le perceptron peut être utilisé pour identifier des objets en fonction de leurs caractéristiques.
- **Problèmes Linéaires** : Le perceptron est efficace pour résoudre des problèmes où les classes sont linéairement séparables. Cependant, il a des limitations dans les cas où les données ne peuvent pas être séparées par une ligne droite (ou un hyperplan dans des dimensions supérieures).

Objectifs Pédagogiques

1. Compréhension des Concepts Fondamentaux :

- À travers la simulation du perceptron, les utilisateurs apprendront les concepts de base de l'apprentissage supervisé, y compris la notion de poids, de biais, et de fonction d'activation.
- Les utilisateurs découvriront comment les modifications des poids influencent la capacité du perceptron à classer correctement les données.

2. Expérimentation :

- Les utilisateurs pourront interagir avec le perceptron en ajustant les poids et le biais, et en observant les effets de ces ajustements sur la classification des données d'entrée.
- Ils pourront également tester le perceptron sur différents ensembles de données, ce qui leur permettra de comprendre les limites du modèle et les situations où il fonctionne bien ou moins bien.

3. Visualisation des Processus :

- La simulation inclura des visualisations graphiques des données d'entrée, des poids, et des résultats de classification. Cela aidera les utilisateurs à visualiser comment le perceptron prend des décisions et apprend à partir des données.
- Les utilisateurs pourront voir comment le perceptron évolue au fil des itérations d'apprentissage, ce qui rendra le processus d'apprentissage plus tangible et compréhensible.

3.4 Fonctionnalités de l'Application

3.4.1 Interface Utilisateur

L'interface utilisateur doit être conçue pour être intuitive et accessible. Les principales caractéristiques incluront :

- **Menu de sélection d'algorithmes** : Permettre aux utilisateurs de choisir l'algorithme qu'ils souhaitent simuler.
- **Zone de visualisation** : Afficher la simulation en temps réel, avec des graphiques et des animations pour illustrer le fonctionnement des algorithmes.
- **Panneau de paramètres** : Offrir des options pour ajuster les paramètres des algorithmes (par exemple, les poids dans un réseau de neurones, les heuristiques dans A*).

3.4.2 Documentation et Aide

Pour faciliter l'apprentissage, l'application inclura :

- **Tutoriels interactifs** : Des guides pas à pas pour aider les utilisateurs à comprendre comment utiliser l'application et les algorithmes.
- **Explications des algorithmes** : Des descriptions détaillées des algorithmes, de leur fonctionnement et de leurs applications.
- **FAQ et support** : Une section d'aide pour répondre aux questions courantes et fournir des conseils.

3.5 Technologies et Outils

Pour le développement de ce projet, plusieurs technologies et outils seront utilisés :

- **Langage de programmation** : Java sera utilisé pour le développement de l'application. Java est un langage robuste et orienté objet, idéal pour créer des applications interactives et multiplateformes. Il offre également une bonne gestion de la mémoire et des performances optimales, ce qui est essentiel pour les simulations d'algorithmes d'IA.
- **Framework de développement** : Un framework comme JavaFX ou Spring Boot sera utilisé pour créer une interface utilisateur graphique (GUI) interactive. JavaFX permet de créer des applications riches en fonctionnalités avec des éléments graphiques modernes, tandis que Spring Boot peut faciliter le développement d'applications web.