Travaux pratiques d'IA

Série 0: Préparation

1 Environnement de travail

Nous utiliserons Python comme langage de programmation pour les TPs. Plusieurs environnements de travail sont envisageables:

- Google Colab (recommandé).
- Repl.it
- Une installation locale avec Conda par exemple.

2 BFS/DFS

Implémenter et tester les fonctions suivantes:

- sample_datapoints(n: int) → list[tuple[float]]: génère n points dans le cercle unité.
- build_graph(n: int, connec_factor: float) → dict[int, list[int]: génère une liste d'adjacence entre n points avec un facteur de connectivité connec_factor, i.e. en moyenne un graphe non dirigé de ce type contient n*connec_factor arrêtes.
- visualize_graph(datapoints: list[tuple[float]], graph: dict[int, list[int]) → None: produit une visualisation du graphe.
- shortest_path_length_bfs(graph: dict[int, list[int], start: int, goal: int) → int: calcule la longueur du plus court chemin entre start et goal à l'aide de BFS.
- shortest_path_length_dfs(graph: dict[int, list[int], start: int, goal: int) → int: calcule la longueur du plus court chemin entre start et goal à l'aide de DFS.
- connected_components_bfs(graph: dict[int, list[int], start: int) → list[int]: calcule la composante connexe associée au noeud start à l'aide de BFS.
- connected_components_dfs(graph: dict[int, list[int], start: int) → list[int]: calcule la composante connexe associée au noeud start à l'aide de DFS.