Artificial Intelligence 25/26
Homework **Predictive MinMax**

The classical **action(s)**=**MinMax(s,H)** is perfect for toy games, does a poor job in realistic ones:

- the tree is too large
- the evaluation **H** of outcome at non-leaves is not trivial.

We could remedy by making the leaf evaluator **H** adaptive, and turn it into a good predictor of the game outcome $\mathbf{H}_{true}$ by using **action(s) := MinMax(s , $\mathbf{H}_{true}$,L,K)** where **L** is a cut in search-depth, and **K** is a cut in explored state neighborhood-width. We address both problems , we limit the search space and we simplify the evaluation.

This turns a static search procedure into an *adaptive-predicting* one. The pipeline becomes:

Search: Use **action(s) := MinMax(s , $\mathbf{H}_{true}$,L,K)** to evaluate positions at the leaves. $\mathbf{H}_{true}$ can be a simple MultilayerPerceptron.

Train loop

1. **Play** loop
   s:=MinMax(s , H_{true} ,L,K)

   s:=MinMax(s , $-H_{true}$,L,K)

   until end game

2. **Observe(outcome)**
   After the game ends with a true outcome $z \in \{-1,0,+1\}$, collect the states visited as a batch.

3. **Learn:**
   Treat each visited state s as a supervised example with target z and train H_{true} with the game batch.

This simple loop slowly improves the MLP's predictive accuracy, which in turn improves the quality of the minimax planning — and thus the quality of the data generated by self-play.
It is a **self-bootstrapping learner**. No reinforcement learning is needed, no prior data to train it. By running it improves, the more it runs the faster improves.

Homework 1: **object** By experimenting, define a strategy [L(t),K(t)] for choosing L and K at the t^{th} iteration of the train loop.

Implement the Predictive MinMax as explained, run an experimentation of your strategy, executed possibly to tune it at best. Turn in the notebook of the experimentation, a complete pdf report, the chatgpt,claude,... chat that you used.

Grading	Adherence to the object: 40%	
	Experimentation logics: 30%	
	Report	30%