基于 AlphaZero 训练中国象棋，核心是用深度神经网络（DNN）替代人类经验棋谱，通过 “自我对弈 + 蒙特卡洛树搜索（MCTS）” 实现从零开始的强化学习。

具体实施需分 4 个关键步骤：

1、环境搭建：构建中国象棋的数字化环境，精准实现棋盘状态（如 32 枚棋子位置、红黑方回合）、走法规则（如马走日、象飞田）和胜负判定逻辑，确保 AI 能理解 “合法操作”。

2、网络设计：训练一个深度神经网络，输入为棋盘状态（通常编码为矩阵），输出分为两部分 ——策略头（预测当前局面下各走法的概率）和价值头（评估当前局面的胜率）。

3、自我对弈生成数据：AI 通过 MCTS 探索走法（结合网络预测的策略和价值），每局对弈记录 “局面 - 走法概率 - 最终胜负” 数据，积累海量高质量训练样本，无需人类棋谱。

4、迭代训练优化：用自我对弈数据不断更新神经网络，再用更新后的网络指导下一轮 MCTS 和对弈，形成 “对弈生成数据→训练优化网络→提升对弈能力” 的闭环，直至 AI 达到顶尖水平。

设计适配中国象棋的 AlphaZero 深度神经网络，核心是让网络能 “看懂” 棋盘并输出关键决策信息，需围绕输入层、特征提取层、策略头与价值头三大模块展开，且需贴合象棋的 “局部战术 + 全局局势” 判断需求。

1. 输入层：将棋盘 “翻译” 成网络能读懂的语言

输入层的核心是棋盘状态编码，需完整且无歧义地传递当前局面信息，避免丢失关键决策依据。

基础维度：将 10×9 的中国象棋棋盘，按 “棋子类型 + 所属方 + 位置” 拆解为多个特征平面（Feature Planes），每个平面是 10×9 的二进制矩阵（1 表示存在对应特征，0 表示不存在）。

棋子类型平面：共 14 个（红方帅、仕、相、马、车、炮、兵各 1 个，黑方将、士、象、马、车、炮、卒各 1 个）。

辅助信息平面：2-4 个，如 “当前回合方”（1 表示红方回合，0 表示黑方）、“是否吃过河卒”（标记兵 / 卒的推进状态）等，帮助网络判断规则细节。

最终输入格式：通常为 [通道数, 10, 9]（通道数 = 14 + 辅助平面数，如 16），例如 PyTorch 中常用的 (batch\_size, 16, 10, 9) 格式，便于后续卷积运算。

2. 特征提取层：让网络 “理解” 棋局规律

这一层是网络的 “大脑”，通过卷积操作捕捉棋盘的局部关联（如 “马后炮”“车捉炮”）和全局趋势（如 “兵力分布”“帅 / 将安全度”），主流采用残差网络（ResNet） 结构。

核心组件：

初始卷积层：用 3×3 小卷积核（适配棋盘局部战术），将输入通道数压缩或调整至目标维度（如 64 或 128），初步提取基础特征。

残差块（Residual Block）：堆叠 10-20 个残差块（数量需平衡性能与训练成本），每个块包含 “卷积→批量归一化（BN）→ReLU 激活” 的两次循环，再通过 shortcut 连接跳过部分层，解决深层网络梯度消失问题，让网络能学习更复杂的棋局模式（如多子联动的全局策略）。

设计原则：不使用全连接层（避免丢失空间位置信息），全程用卷积和残差结构保留棋盘的 “位置关联性”—— 这是区别于图像识别网络的关键（象棋的 “马在哪个位置” 比 “马长什么样” 更重要）。

3. 输出层：分两路输出决策结果

AlphaZero 的网络需同时输出 “走法概率” 和 “局面胜率”，因此采用双输出头结构，两路共享特征提取层的结果，但各自有独立的计算分支。

策略头（Policy Head）：输出 “当前局面下每个合法走法的概率”，指导 MCTS 探索方向。

1-2 层卷积：将特征提取层的输出通道数降至较低维度（如 2 或 3），聚焦 “走法相关特征”。

全连接层：将 10×9 的特征图 flatten 成向量，再映射到 “所有可能走法” 的维度（中国象棋约有 200 种合法走法，具体数量随局面变化）。

Softmax 激活：将输出转化为概率分布，概率越高表示网络认为该走法越优（无需过滤非法走法，后续 MCTS 会处理）。

价值头（Value Head）：输出 “当前局面的胜率评估”（范围 [-1,1]，1 表示必胜，-1 表示必败），快速判断局面优劣。

1 层卷积 + 1 层全连接：先通过卷积压缩特征，再用全连接层将特征映射到低维向量（如 256 维）。

ReLU 激活 + 最终全连接：经过激活函数后，用 1 个神经元输出 scalar 值，再通过 tanh 激活将结果限制在 [-1,1]，直接代表 “当前方的胜率期望”。

关键设计技巧

避免过度参数化：中国象棋的状态空间（约 10^48）远小于围棋（10^170），无需像 AlphaZero 围棋版那样用 40 层残差块，10-15 层残差块即可满足需求，减少训练时间和硬件消耗。

输入特征简化：无需加入 “历史步数”“将军次数” 等复杂特征，基础的 “棋子位置 + 回合方” 已能覆盖 90% 以上的决策信息，过多特征反而会增加网络学习负担。