

论文题目：《基于信用分配的决策优化——以中国象棋博弈为例》

报告人：hohoho

## 1. 背景与目标

信用分配（credit assignment）即奖励分配，是强化学习领域的一个重要问题。对于单智能体强化学习，智能体可能在中间的时间步没有收到任何奖励信号，而在最终的时间步才收到奖励信号，信用分配就是要解决这种奖励延迟的问题，合理量化智能体每个时间步应得的奖励，从而鼓励智能体执行价值高的动作，避免无价值的动作。所以、本论文旨在深入研究单智能体强化学习领域中的信用分配问题，以中国象棋这一典型的完全信息博弈为切入点，详细探讨如何有效地对中国象棋中的每个对弈步进行奖励分配，以优化智能体的学习效果。通过对比分析多种信用分配方法的特点，寻找一种切实可行且高效的奖励机制。

本文将提出一种基于深度神经网络评估的奖励机制。该机制将充分考虑当前实际对弈结果以及涉及的各个棋子的棋力，从而为每一步着法赋予恰当的价值。这一奖励机制的引入将有助于激励智能体选择那些具有更高价值的动作，从而加速智能体的学习过程。

## 2. 项目进展情况

当前进行奖励分配的主要方法是对奖励函数进行重塑（reward shaping）。Jose A. Arjona-Medina 等[1]提出 RUDDER 方法进行奖励重塑，该方法指出可以使用 LSTM 进行关于奖励的模式识别，从而对奖励分解后再重分配。如图 2-1 所示，得益于 LSTM 的网络架构：遗忘门、记忆门、输出门，如果信息没变化，网络不会学到新的模式，当有新的信息，相应的模式就会被学习到。通过分析 LSTM 的“记忆”，可以重构出关键事件这些信息，并根据最后预测的奖励的贡献度分

配给每个状态-动作配对，如图 2-2 所示。

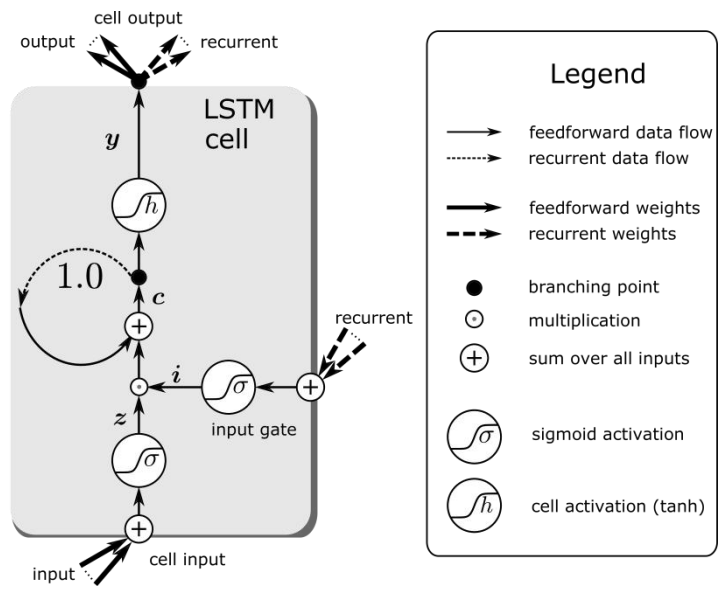


图 2-1 LSTM 单元结构<sup>1</sup>

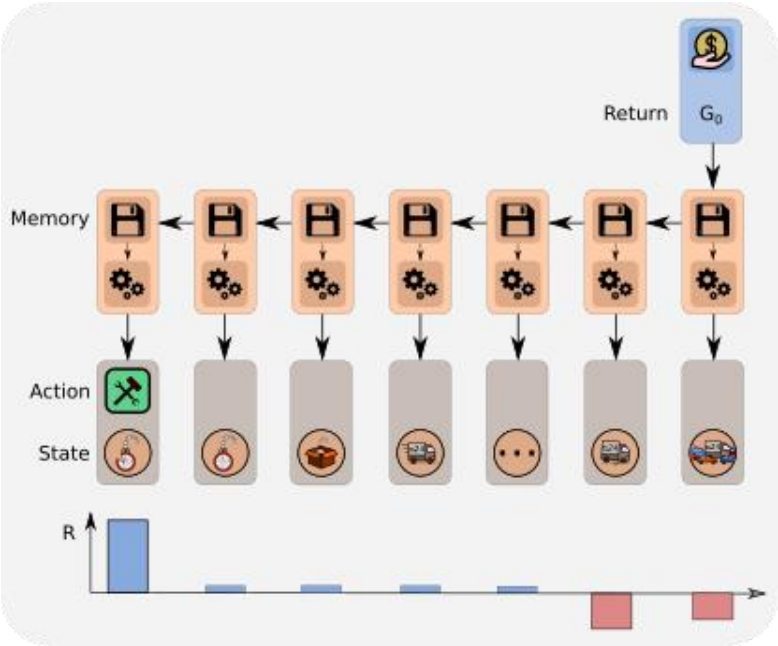


图 2-2 使用 LSTM 检测关键事件<sup>2</sup>

多触点归因 (MTA: Multi Touch Attribution) 是一种确定每个触点对最终

<sup>1</sup> 图片来自 <https://ml-jku.github.io/rudder/>  
<sup>2</sup> 图片来自 <https://ml-jku.github.io/rudder/>

转化效果的价值贡献进行分析的方法，常应用于数字营销场景，如顾客从开始对某种产品产生初始印象到最终购买该产品期间，可能会涉及多个渠道（称为触点），如何把“功劳”公平归因于各个渠道，是多触点归因要解决的问题。Ning li 等[2]提出了一种使用带注意力机制的深度神经网络模型 DNAMTA，使用 LSTM 来拟合顾客所触达的广告序列路径，学习每个广告渠道的权重，并且将顾客个人静态信息通过全连接神经网络编码融入到整体的训练模型中。

受此启发，中国象棋的博弈过程可以类比于数字营销中顾客所触达的广告渠道路径：下棋者落子的每部棋可类比于每个广告渠道，下棋的最终输赢结果则类比于整体广告效果是否让顾客最终购买了该产品，从而中国象棋的信用分配问题可以借鉴广告的多触点归因分析来解决。本文将使用 LSTM 为主要神经网络架构来实现多触点归因分析，识别每一步落子的贡献度大小。

## 2.1 数据收集与整理

通过与象棋程序进行自动博弈来收集博弈数据。数据经过去除重复值、去除空值和异常值处理后，还基于原始数据组合新的数据特征。数据字段说明如表 2-1 所示。

表 2-1 博弈数据字段说明

| 字段名             | 说明                           |
|-----------------|------------------------------|
| match_id        | 博弈场次 id，标识每一场博弈              |
| round_id        | 轮次 id，标识每一场博弈中每一轮对战          |
| state           | 当前棋局状态                       |
| action          | 落子动作                         |
| next_state      | 下一个棋局状态                      |
| done            | 当前博弈局是否结束：0 表示未终局，1 表示已终局    |
| chapture_reward | 吃子奖励：正数为红方吃子，负数表示黑方吃子        |
| win             | 胜/负/和局标识：1 表示胜，-1 表示负，0 表示和局 |
| converted       | 胜或负为 1，和局为 0                 |
| jid             | 每次落子的唯一标识                    |

其中 state 所表示棋局状态遵循中国象棋电脑应用规范所示使用 FEN 格式串表示<sup>3</sup>，action 所示落子动作遵循中国象棋电脑应用规范所示着法表示<sup>4</sup>。详细数据样例可参考图 2-3

| match_id | round_id       | state  | action | next_state   | done | captures | score_red | win | converted | fid            |
|----------|----------------|--|--------|--|------|----------|-----------|-----|-----------|----------------|
| 0-0      | 1692660737310  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-0       | 1692660737310  |
| 0-0      | 1692660737311  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-0       | 1692660737311  |
| 0-1      | 1692660737320  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-1       | 1692660737320  |
| 0-1      | 1692660737321  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-1       | 1692660737321  |
| 0-2      | 1692660737330  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-2       | 1692660737330  |
| 0-2      | 1692660737331  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-2       | 1692660737331  |
| 0-3      | 1692660737340  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-3       | 1692660737340  |
| 0-3      | 1692660737341  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-3       | 1692660737341  |
| 0-4      | 1692660737350  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | a0a1   | ENBAKBNR/78/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr  | 0/N  | -6, 28   | 0         | 0   | 0-4       | 1692660737350  |
| 0-4      | 1692660737351  | ENBAKBNR/78/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr  | h2b4   | ENBAKBNR/78/1C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr    | 0    | 0        | 0         | 0   | 0-4       | 1692660737351  |
| 0-4      | 1692660737352  | ENBAKBNR/78/1C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr    | a0a1   | ENBAKBNR/78/1C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr    | 0    | 0        | 0         | 0   | 0-4       | 1692660737352  |
| 0-4      | 1692660737353  | ENBAKBNR/78/1C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr    | h2b4   | ENBAKBNR/78/9/P1P1P1P/1C5C1/9/p1p1p1p/1c5c1/9/rnbakbnnr  | 0    | 0        | 0         | 0   | 0-4       | 1692660737353  |
| 0-4      | 1692660737354  | ENBAKBNR/78/9/P1P1P1P/1C5C1/9/p1p1p1p/1c5c1/9/rnbakbnnr  | h4b6   | ENBAKBNR/78/9/P1P1P1P/4C2C1/9/p1p1p1p/1c5c1/9/rnbakbnnr  | 0/C  | -25, 41  | 0         | 0   | 0-4       | 1692660737354  |
| 0-4      | 1692660737355  | ENBAKBNR/78/9/P1P1P1P/4C2C1/9/p1p1p1p/1c5c1/9/rnbakbnnr  | h4f4   | ENBAKBNR/78/9/P1P1P1P/4C1/9/p1p1p1p/1c5c1/9/rnbakbnnr    | 0    | 0        | 0         | 0   | 0-4       | 1692660737355  |
| 0-4      | 1692660737356  | ENBAKBNR/78/9/P1P1P1P/4C1/9/p1p1p1p/1c5c1/9/rnbakbnnr    | i1i4   | ENBAKBNR/78/9/P1P1P1P/4C2P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | 0    | 0        | 0         | 0   | 0-4       | 1692660737356  |
| 0-4      | 1692660737357  | ENBAKBNR/78/9/P1P1P1P/4C2P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | a1f1   | ENBAKBNR/58/9/P1P1P1P/4C2P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | 0    | 0        | 0         | 0   | 0-4       | 1692660737357  |
| 0-4      | 1692660737358  | ENBAKBNR/58/9/P1P1P1P/4C2P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | i0a2   | ENBAKBNR/58/484/P1P1P1P/4C2P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-4       | 1692660737358  |
| 0-5      | 1692660737360  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-5       | 1692660737360  |
| 0-5      | 1692660737361  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-5       | 1692660737361  |
| 0-6      | 1692660737370  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-6       | 1692660737370  |
| 0-6      | 1692660737371  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-6       | 1692660737371  |
| 0-7      | 1692660737380  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-7       | 1692660737380  |
| 0-7      | 1692660737381  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-7       | 1692660737381  |
| 0-8      | 1692660737390  | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-8       | 1692660737390  |
| 0-8      | 1692660737391  | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-8       | 1692660737391  |
| 0-9      | 16926607373900 | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | f0e1   | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-9       | 16926607373900 |
| 0-9      | 16926607373901 | ENBAKBNR/444/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | h2g2   | ENBAKBNR/444/1C4C2/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 1/E  | -4, 92   | -1        | 1   | 0-9       | 16926607373901 |
| 0-10     | 16926607373810 | ENBAKBNR/9/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr   | a0a1   | ENBAKBNR/484/1C5C1/P1P1P1P/P/9/p1p1p1p/1c5c1/9/rnbakbnnr | 0/N  | -6, 28   | 0         | 0   | 0-10      | 16926607373810 |

图 2-3 博弈数据样例

原始数据经过清洗后，由于 state 棋局状态特征为字符串型，现使用 Sentence-BERT[3]的方法将编码为数值型向量，而 action 落子动作特征则编码为 one-hot 向量。最终将每个样本特征拼接为 2871 维的向量，作为训练数据。当前已收集了 11082 条博弈记录（仅红方），根据 match\_id 进行分组，形成形如[batch, step, feature]的训练样本 2462 条，测试样本 308 条，验证样本为 308 条。

## 2.2 模型开发与训练

本论文借鉴 DNAMTA[2]的神经网络归因模型，主要使用 LSTM 作为模型的主要架构（图 2-4），并通过注意力层学习奖励分配权重。训练过程如下：

- （1）使用 6 层 LSTM 对博弈数据特征进行编码，分别保留每一步的输出结果；
- （2）每一步的编码结果输入到注意力网络层，输出注意力权重；
- （3）注意力权重与 LSTM 的每一步编码结果进行加权求和，最后通过一个全连接层输出预测结果（converted=1 或 converted=0）。

<sup>3</sup> 中国象棋电脑应用规范（三）——FEN 文件格式：[https://www.xqbase.com/protocol/cchess\\_fen.htm](https://www.xqbase.com/protocol/cchess_fen.htm)

<sup>4</sup> 中国象棋电脑应用规范（二）——着法表示：[https://www.xqbase.com/protocol/cchess\\_move.htm](https://www.xqbase.com/protocol/cchess_move.htm)

```

LstmAttentionModel(
    (lstm1): LSTM(2871, 256, batch_first=True, dropout=0.05)
    (lstm2): LSTM(256, 256, batch_first=True, dropout=0.05)
    (lstm3): LSTM(256, 256, batch_first=True, dropout=0.05)
    (lstm4): LSTM(256, 256, batch_first=True, dropout=0.05)
    (lstm5): LSTM(256, 256, batch_first=True, dropout=0.05)
    (lstm6): LSTM(256, 256, batch_first=True, dropout=0.05)
    (attention): AttentionLayer()
    (output_layer): Linear(in_features=256, out_features=1, bias=True)
)

```

图 2-4 本论文模型架构

## 2.3 模型下游应用

模型下游应用主要是奖励分配——分配适当的奖励到每个状态-动作配对上。从训练完多模型提取注意力层的权重，基于历史博弈数据可统计出每个落子动作对应的奖励贡献权重，如算法 1 所示。

---

### 算法 1 计算奖励权重比例

---

输入: *AttentionWeight* 模型注意力层权重, *Xdata* 博弈数据, *actionCount* 落子动作的数量

输出: 奖励权重比例

```

1: function GETACTIONINDEX(XStep, actionCount)
2:   startIndex  $\leftarrow$  768
3:   endIndex  $\leftarrow$  startIndex + actionCount - 1
4:   return  $\arg \max(XStep[startIndex : endIndex])$ 
5: end function
6:
7: function GETREWARDWEIGHT(AttentionWeight, XData)
8:   attributions  $\leftarrow$  [0, 0, 0, ..., 0]
9:   actionsFreq  $\leftarrow$  [0, 0, 0, ..., 0]
10:  stepCount  $\leftarrow$   $\text{len}(AttentionWeight)$ 
11:  for i = 0  $\rightarrow$  stepCount do
12:    stepList  $\leftarrow$  AttentionWeight[i]
13:    contributionCount  $\leftarrow$   $\text{len}(stepList)$ 
14:    for j = 0  $\rightarrow$  contributionCount do
15:      actionIndex  $\leftarrow$  GETACTIONINDEX(XData[i][j], actionCount)
16:      stepContribution  $\leftarrow$  stepList[j]
17:      attributions[actionIndex]  $\leftarrow$  attributions[actionIndex] + stepContribution
18:      actionFreq[actionIndex]  $\leftarrow$  actionFreq[actionIndex] + 1
19:    end for
20:  end for
21:  result  $\leftarrow$  attributions / actionsFreq
22:  return result
23: end function

```

---

当前基于算式（2-1）进行奖励分配。

$$reward_{action_i} = \frac{weight_{action_i}}{\sum_{k=1}^N weight_{action_k}} \times reward_{total} \quad (\text{算式 2-1})$$

其中 $weight_{action_i}$ 为算法 1 的计算结果。

### 3. 当前成果

（1）已完成了神经网络模型训练并已验证训练收敛，如图 3-1 所示。

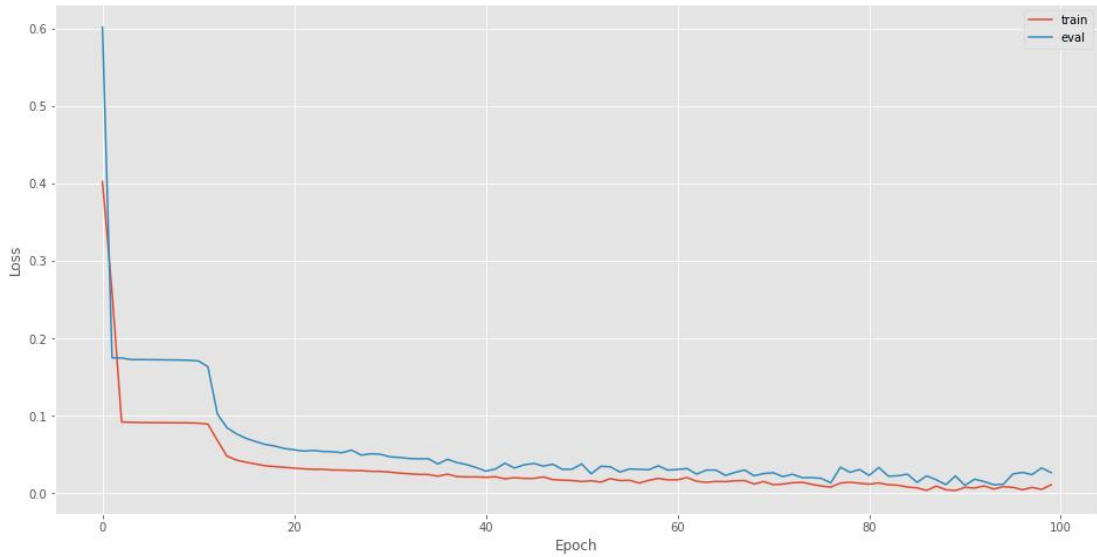


图 3-1 模型训练过程

（2）已输出落子动作对应的奖励权重，如图 3-2 所示，横坐标为落子动作的 ID，纵坐标为动作对应的贡献度。

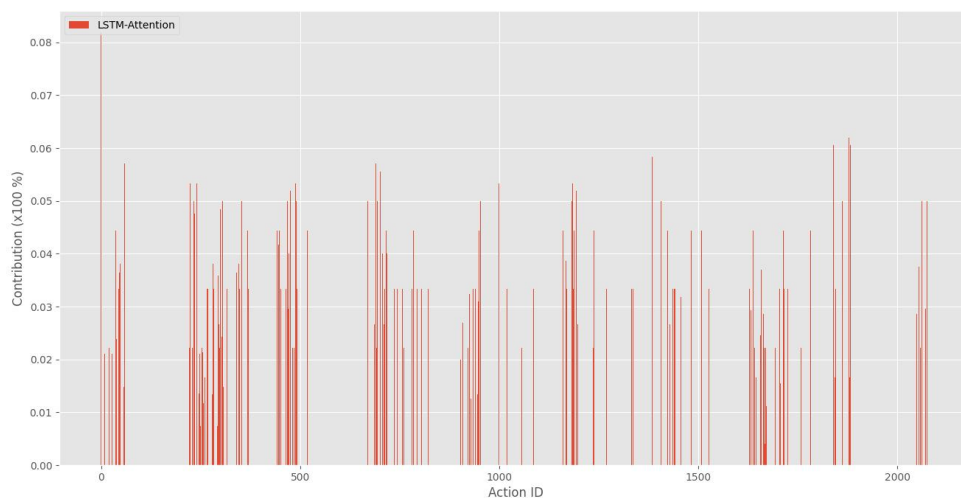


图 3-2 落子动作对应奖励权重的分布

(3) 已输出 AlphaGo Zero[4]的奖励分配方式（所有动作都得到相同的奖励分配，即平均分配方式）与本论文的奖励分配方式对比结果，如图 3-3、图 3-4 所示，作为后续分析使用。



图 3-3 橙色是本论文奖励重分配结果，蓝色是赢时 AlphaGo Zero 的分配方式（所有落子动作都赋值奖励 1）





图 3-4 橙色是本论文奖励重分配结果,蓝色是赢时 AlphaGo Zero 的分配方式(所有落子动作都赋值奖励-1)

## 4. 遗留问题与解决方案

(1) 本论文所提出的奖励分配方式缺少充分的对比验证。解决方案：增加多种奖励分配方式进行对比验证。

(2) 关于带有注意力机制的 LSTM 如何识别关键落子动作,以及与其他模型架构比较本论文所展现的架构有何优势等问题,缺少科学论证与事实说明。解决方案：将加入其他模型架构进行结果对比,并通过消融实验,说明本论文解决方案的科学性。

(3) 暂缺本论文所提解决方案对于强化学习效率提升的解释性说明。解决方案：本模型奖励分配架构与象棋博弈系统有待集成。



## 5. 下阶段计划

针对以上遗留问题，在接下来的研究阶段将进一步优化奖励分配的解决方案。具体计划如下：

### （1）对比验证奖励分配方式

针对奖励分配方式缺乏充分对比验证的问题，将进行更广泛的对比实验，以评估本论文提出的奖励分配方式的有效性。除了与平均分配方式的对比外，还将考虑包括但不限于以下方式进行对比验证：

- 基于最后触点的分配： 将实施该奖励分配方案，其中奖励主要集中在每个落子序列的最后一个动作，以验证这种方法是否能够提高性能。
- 基于时间衰减的分配： 将尝试根据时间衰减策略分配奖励，逐渐降低早期动作的奖励权重，以探索这种方式是否能够更好地平衡长期和短期奖励。

### （2）探索关键落子动作识别与模型比较

为了解决带有注意力机制的 LSTM 如何识别关键落子动作的问题，将采取以下步骤：

- 引入其他模型架构： 我们将引入其他带有注意力机制的模型，例如 Transformer 等，与我们的架构进行比较，以揭示不同架构之间的优势和劣势。
- 消融实验： 将对本论文提出的架构进行消融实验，逐步去除注意力机制等关键组件，以验证这些组件对于模型性能贡献。

### （3）提升强化学习效率的可解释性

为了更好地解释我们的奖励分配架构如何提升强化学习效率，将采取以下措施：

- 集成象棋博弈系统： 将我们的奖励分配架构应用于象棋博弈系统，通过在实际场景中的应用来验证其效果，同时从实际案例中提取具体的效率提升情况和原因。
- 解释性分析： 我们将进行对模型决策的解释性分析，例如通过可视化注意力权重等方式，揭示模型在不同决策点上的关注重点，从而增加模型解释性。

基于以上计划，将继续完成论文撰写，预计 2023 年 11 月完成论文初稿。

## 参考文献

- [1] J. A. Arjona-Medina, M. Gillhofer, M. Widrich, T. Unterthiner, J. Brandstetter, and S. Hochreiter, ‘RUDDER: Return Decomposition for Delayed Rewards’. arXiv, Sep. 10, 2019. doi: 10.48550/arXiv.1806.07857.
- [2] N. li, S. K. Arava, C. Dong, Z. Yan, and A. Pani, ‘Deep Neural Net with Attention for Multi-channel Multi-touch Attribution’. arXiv, Sep. 06, 2018. Accessed: Aug. 23, 2023. [Online]. Available: <http://arxiv.org/abs/1809.02230>
- [3] N. Reimers and I. Gurevych, ‘Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks’. arXiv, Aug. 27, 2019. doi: 10.48550/arXiv.1908.10084.
- [4] D. Silver *et al.*, ‘Mastering the game of Go without human knowledge’, *Nature*, vol. 550, no. 7676, pp. 354 - 359, Oct. 2017, doi: 10.1038/nature24270.