作业3：深度学习

李堃秀 2022012110 Likx22@mails.tsinghua.edu.cn

1 简答题

1. **什么是交叉熵（Cross Entropy）？在学习一个类别分布（Categorical Distribution）时，使用交叉熵作为损失函数比绝对值损失函数（Absolute Error，$L_{abs} = |y_i - \hat{y}_i|$）有什么好处?**
   交叉熵是一种用于衡量两个概率分布之间差异的损失函数。
   使用交叉熵作为损失函数的好处：
   1)**能够更好地处理概率分布之间的细微差别**：交叉熵直接衡量的是概率分布之间的差异，而绝对值损失函数只衡量了预测值和真实值之间的差异，但忽略了概率分布的整体形状。
   2)**能够更快的接近正确的概率分布**：交叉熵损失函数在概率接近真实值时会产生更大的梯度，促使模型更快地调整参数以接近正确的概率分布，而绝对值损失函数在接近最优解时梯度较小，可能导致训练速度减慢或陷入局部最优。
   3)**数值更加稳定**

2. **多层感知机（Multilayer Perceptron）相比线性模型有哪些优势？相较于训练浅而宽的神经网络（“宽度学习”），训练相对窄而深的神经网络有什么好处?**
   相比线性模型，多层感知机**具有处理非线性关系的能力**，也就意味着其**拥有更强的表达能力和适应性。**
   相比“宽度学习”，训练窄而深的神经网络的好处：
   1)**具有更好的泛化与表现能力**：深的神经网络能够通过多层隐藏层能提取不同层次的特征，从而增强了模型的抽象能力，这意味着其能够提高泛化能力并适应更为复杂的数据模式，避免过拟合。
   2)**能够更加有效利用参数**

3. **卷积（Convolution）和互相关（Cross-correlation）分别是什么意思？在卷积神经网络中，卷积核通常进行的是卷积还是互相关操作?**
   卷积：数学运算，通常指将一个函数（或矩阵）与另一个函数（或矩阵）按某种方式结合的过程。该过程通常包括：一个函数（例如卷积核）在输入信号上滑动，并进行元素逐点乘积后求和的操作，卷积核会在每个位置与输入数据的局部区域进行反向对齐。
   互相关：类似于卷积的一种操作，但不同之处在于，互相关操作中，卷积核直接与输入信号对齐，不会反向翻转卷积核。
   在卷积神经网络中，卷积核通常进行的是**互相关操作**

4. **批量大小（Batch Size）对于优化器(比如随机梯度下降) 影响巨大。为了减小现存占用，小宣提出将每次前向传播的批量大小减半，梯度累积两次再进行反向传播。请问这种方法能确保训练得到的模型效果参数一致吗（假设随机状态、batch 划分、dropout 的神经元相同）？若有影响，请指出原因（例如优化器、模型中的某些层）；若无影响，请论证。**
   这种方法**不**能确保训练得到的模型效果参数一致
   影响原因：
   1)**影响优化器行为**：在使用梯度累积时，每次更新的梯度实际上是多个小批量梯度的平均值，因此改变批量大小可能影响每次更新的梯度值，进而影响优化器行为。
   2)**影响学习率**：如果在减小批量大小时没有相应调整学习率，可能会导致模型更新方式的不同，从而影响最终结果。即使梯度累积两次，也不一定能消除这种差异

3） **计算过程存在数值差异**：在进行梯度累积时，由于每个批量的梯度可能具有较大差异，计算的数值精度和反向传播过程中的小误差可能会在训练过程中累积，从而影响模型的最终结果。

5. **为什么说残差连接（Residual Connection）有利于训练更深层的深度网络？残差链接能够缓解梯度消失（Gradient Vanishing）的问题吗?**
   残差连接有利于训练更深层的深度网络的原因：
   1)**能够缓解梯度消失问题**：残差连接通过直接将输入信息传递到网络的更深层，避免了在反向传播时梯度逐渐消失的问题。
   2)**其优化过程更加高效**
   3)**能够提升训练速度和准确度**：通过缓解梯度消失问题，残差连接使得模型更容易训练，通常能提高收敛速度并达到较高的准确度。
   残差链接**能够**缓解梯度消失的问题

2 解答题

2.1 卷积神经网络
一个卷积神经网络的前向传播过程如下图所示，它依次通过以下各层将一张尺寸为55×55×1的单通道图片转换为1024×1 的输出：卷积层（Convolution）、最大池化层（Max Pooling）、ReLU 、全连接层（各层参数已在图中列示），且在该网络中，我们将不使用任何偏置参数（Bias Parameters）。则对于该网络：
1. 卷积层共有多少个可学习参数?
$3*5*5=75$

2. 最大池化层的输出的尺寸为多少?
卷积层输出尺寸：$\frac{55-5}{2}+1=26$, 有3个filter
最大池化层输出尺寸：$\frac{26-2}{2}+1=13$，故为$13*13*3$

3. 在前向传播过程中，对于每个样本需要进行多少次ReLU 函数计算
$$26*26+13*13+1024*1=1869$$

$

4. 为了给模型加入非线性，需要在网络中加入激活函数，请列举两个激活函数。
ReLU、Sigmoid

2.2 注意力机制
本题中我们将探究利用GPT 类架构进行机器翻译过程中的注意力机制计算过程。
假设我们想要翻译"他| 喜欢| 苹果"这一中文句子（3 个token 使用竖线分隔），在GPT 的某一自注意力层中3 个token 的query、key、value 向量分别记作
$Q=q_1,\ldots,q_3,K=k_1,\ldots,k_3,V=v_1,\ldots,v_3,q_i,k_i,v_i\in R^d$ **。**
1. 请写出经过带掩码的自注意力层$Y=Attention(Q,K,V)=Softmax\left(\frac{QK^T}{\sqrt{d}}\right)V$**之后每个token 对应的输出yi 的表达式。（注意以上公式中省略了掩码）**

$$y_i=\sum_{j=1}^{3}\frac{e^{\frac{q_ik_j^T}{\sqrt{d}}}}{\sum_{n=1}^{3}e^{\frac{q_ik_n^T}{\sqrt{d}}}}v_j$$

2. 设d = 4, $K = \left\{ \begin{bmatrix} 0 \\ 1 \\ -1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ -1 \\ 0 \\ -1 \end{bmatrix} \right\}$。当GPT 模型预测翻译的第一个token（英文单词）“He”的时候，它应该需要尽量多的来自token“他”的信息。请写出向量q_3 的一个取值，满足q_3的2 范数不超过1，且与第一个token 的自注意力权重（相比别的token）最大，并写出此时y_3 关于v_1，v_2，v_3 的表达式。

$q_3 = (0,1,0,0)^T$

则:

$$\frac{q_3 k^T}{\sqrt{d}} = \frac{1}{2}(1,0,1,1) = (\frac{1}{2}, 0, \frac{1}{2}, \frac{1}{2})$$

运用Softmax函数后得到

$$y_3 = \frac{\sqrt{e}}{3\sqrt{e}+1} v_1 + \frac{1}{3\sqrt{e}+1} v_2 + \frac{2\sqrt{e}}{3\sqrt{e}+1} v_3$$

3. 设生成了token“He”之后，计算到这一层时，$k_4 = (-1,0,-1,-1)^T$。当GPT模型预测翻译的第二个token（英文单词）“likes”的时候，它同时需要“喜欢”和“He”的信息（因为“likes”是第三人称单数形式），此时多头自注意力机制可以胜任。假设以query，key，value向量的前两维和后两维作为两个自注意力头（Heads）的特征向量，请写出向量q_4 的一个取值，满足q_4 的2 范数不超过1，且在第一个自注意力头中与第二个token（“喜欢”）的自注意力权重（相比别的token）最大，在第二个自注意力头中与第四个token（“He”）的自注意力权重最大，并写出此时y_4 的表达式。

$q_4 = (0,1,0,0)^T$

类似b中过程，第一个自注意力头，有：

$$y_4^{(1)} = \alpha v_1 + \beta v_2 + 2\alpha v_3$$

第二个自注意力头，有：

$$y_4^{(2)} = \gamma v_1 + \delta v_2 + \gamma v_3$$

有

$$y_4 = y_4^{(1)} + y_4^{(2)} = (\alpha+\gamma)v_1 + (\beta+\delta)v_2 + (2\alpha+\gamma)v_3$$

其中

$$\alpha = \frac{e^{0.5}}{1+3e^{0.5}}, \beta = \frac{1}{1+3e^{0.5}}, \gamma = \frac{e}{2(1+2e)}, \delta = \frac{1}{3+e}$$

3 深度学习与AlphaGoZero

1. 将之前作业中完成的代码填入对应位置，运行训练，并汇报使用MLP 模型的AlphaZero 算法训练过程中对Random Player 的胜率，和训练过程的elo 分数曲线图。

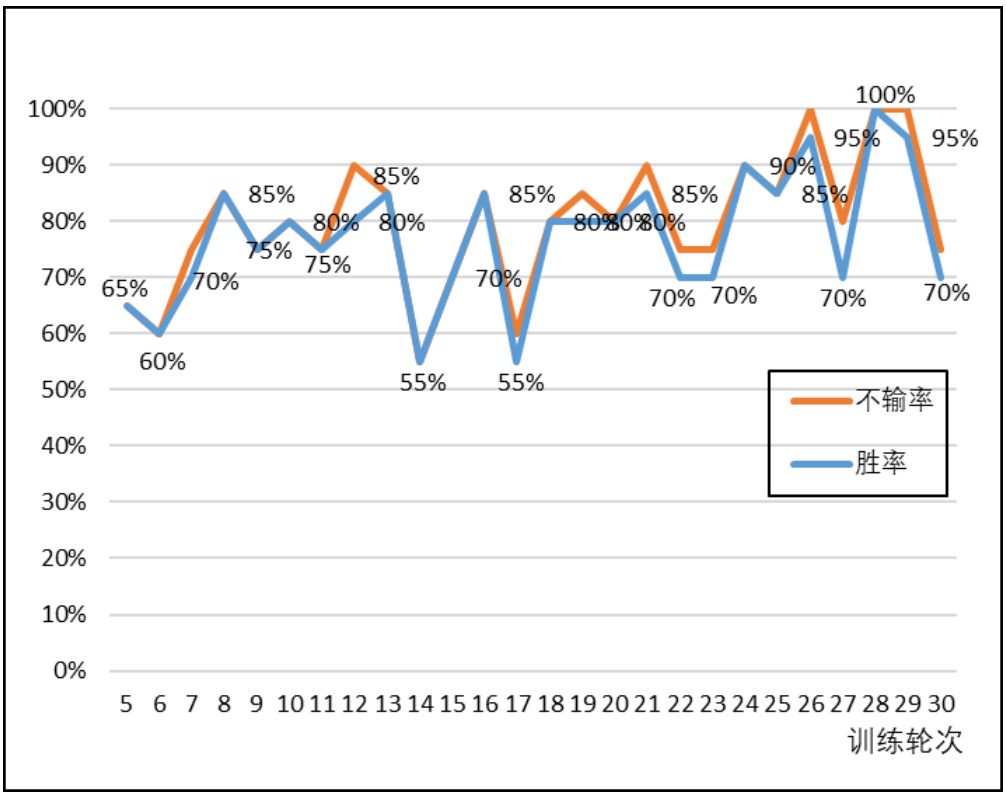

图1：MLP模型训练中对战Random Player

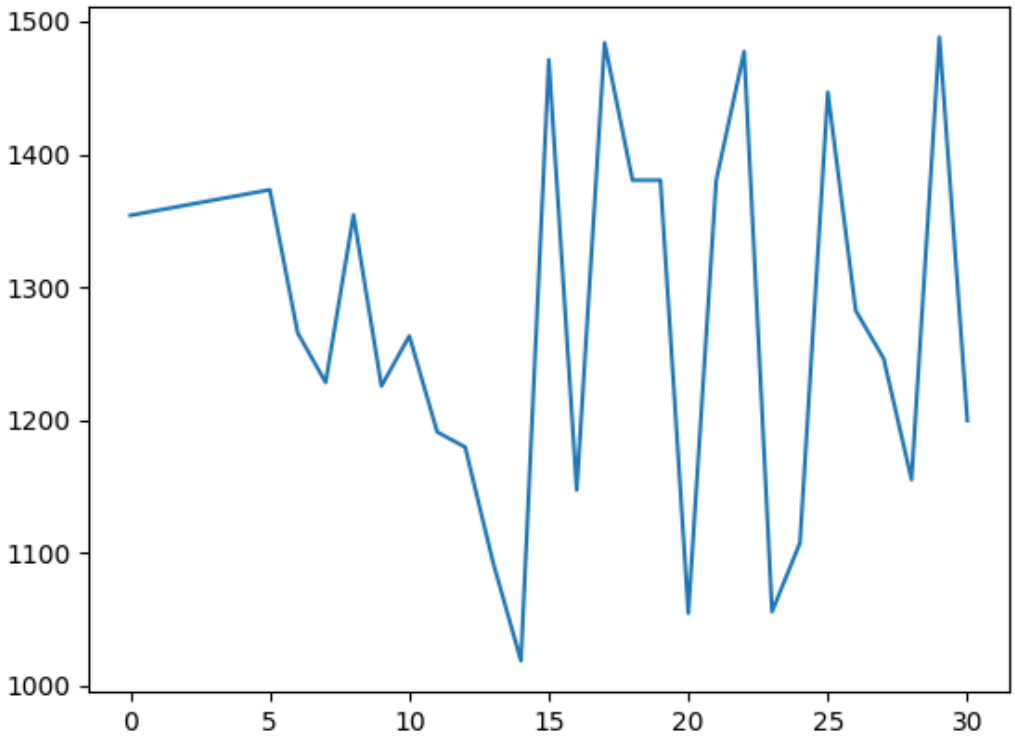

图2：MLP模型训练elo曲线

2. 参考示例代码，设计并实现一个不一样的深度模型，要求至少需要使用一个卷积层处理二维棋盘特征。请绘制网络结构图，并简要说明设计的理由。

MyNet通过结合卷积层和全连接层，实现二维棋盘特征的提取与处理。

卷积层：用于捕捉局部特征，批规范化和ReLU激活函数能够加速训练并提升模型的非线性能力。

展平：将卷积层提取的特征展平成一维向量。

全连接层：防止模型过拟合。
输出层：采用LogSoftmax激活函数输出动作的概率分布。

```
Input: batch size x board x x board y
                    |
添加通道维度: batch size x 1 x board x x board y
                    |
卷积层1: Conv1 -> BatchNorm1 -> ReLU (Output: batch size x 32 x board x x board y)
                    |
卷积层2: Conv2 -> BatchNorm2 -> ReLU (Output: batch size x 64 x board x x board y)
                    |
展平: Flatten (Output: batch size x (board x * board y * 64))
                    |
全连接层1: Linear1 -> ReLU -> Dropout (Output: batch size x 256)
                    |
全连接层2: Linear2 -> ReLU -> Dropout (Output: batch size x 128)
                    |
                   / \
                  /   \
策略输出（Policy Head）:          价值输出（Value Head）:
LinearPolicy -> LogSoftmax        LinearValue -> Tanh
(Output: batch size x action space size)    (Output: batch size x 1)
```

图3：MyNet模型网络结构图

3. 使用自己设计的深度模型，运行训练，并汇报训练过程中AlphaZero 算法对Random Player的胜率以及elo 分数曲线图。要求训练过程中，对Random Player 的胜率至少有一次不低于90%。

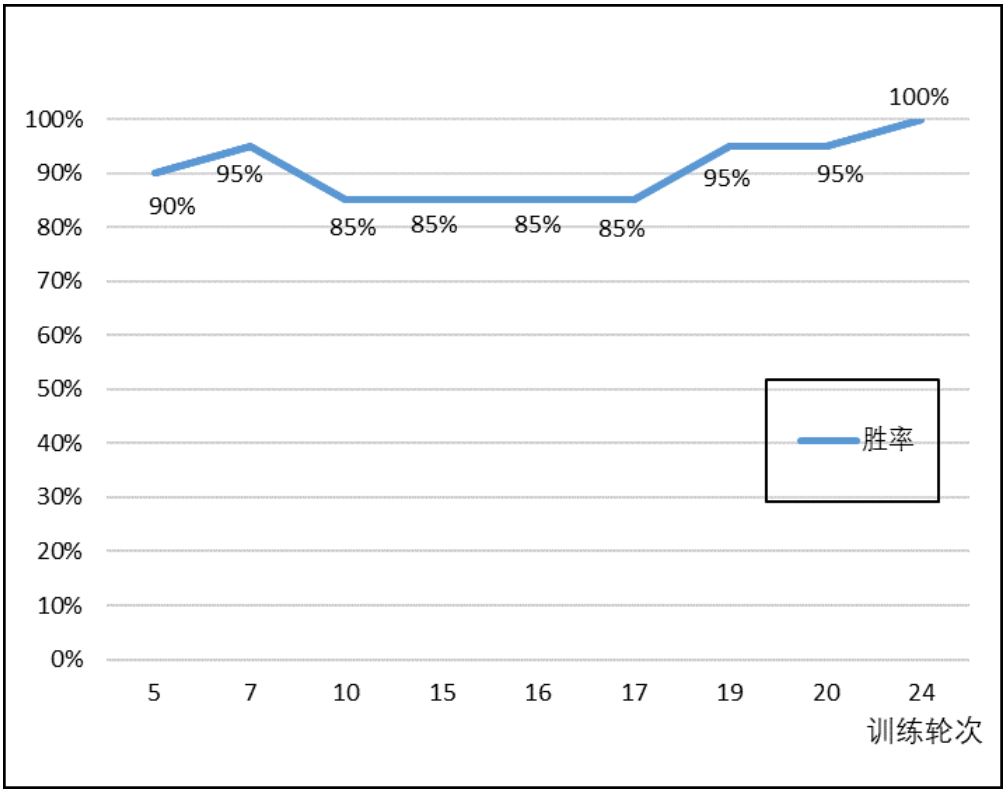

图4：MyNet模型训练中对战Random Player

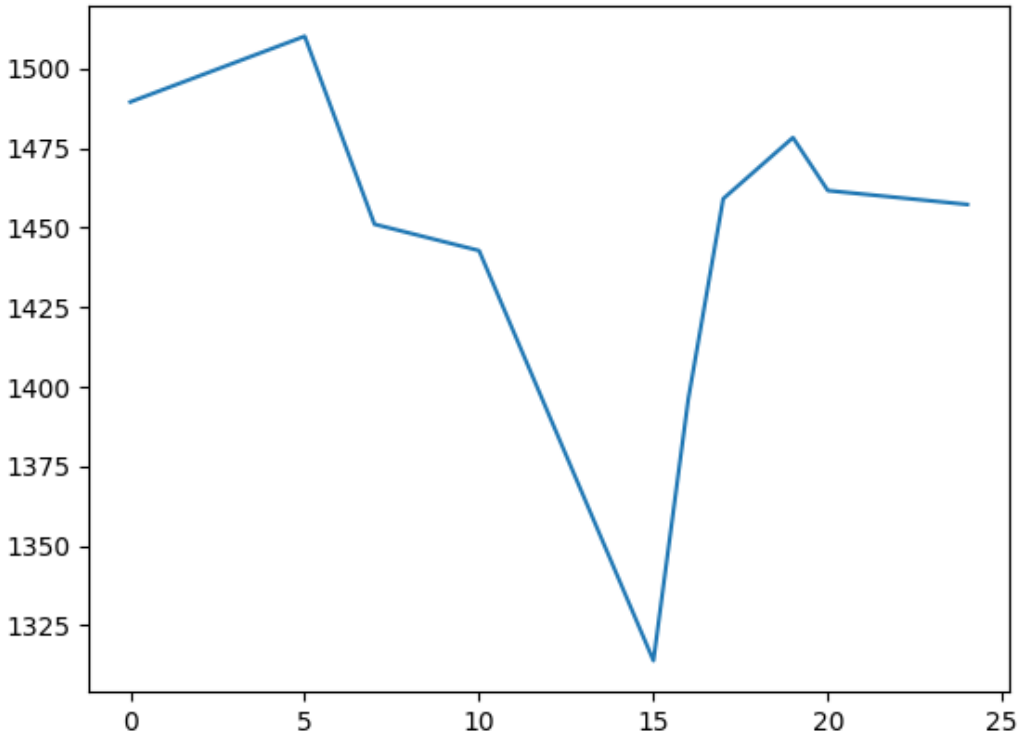

图5：MyNet模型训练elo曲线

4. 修改pit_puct_mcts.py，加载训练后的MLP 模型和自己设计的模型进行对弈，汇报对局的胜率。

```
D:\code>python pit_puct_mcts.py
W0511 18:10:39.117000 5196 site-packages\torch\distributed\elastic\multiprocessing\redirects.py:29] NOTE: Redirects are
currently not supported in Windows or MacOs.
100%|                                                                                | 15/15 [05:34<00:00, 22.28s/it]
100%|                                                                                | 15/15 [05:32<00:00, 22.19s/it]
Player 1 (MLP_net) win: 11 (36.67%)
Player 2 (My_net) win: 18 (60.00%)
Draw: 1 (3.33%)
Player 1 not lose: 12 (40.00%)
Player 2 not lose: 19 (63.33%)
```

图6：MLP对战MyNet结果

声明：

在进行训练前，参考优秀作业案例的思路，对第二次作业中的puct_mcts相关实现进行了修改完善。

本次作业所需实现的MyNet相关部分为独立完成。