设计思路

问题分析

此次任务是一个二分类问题。青少年特征和行为作为features，是否成年的信息作为tag，tag的取值空间为$\{0,1\}$。任务是构建从输入features到输出tags的映射。

数据预处理

Hopfield Module

我们的模型参考了论文《Hopfield Network is All You Need》，采用了论文中所提出的模型*Hopfield* 。该模型在诸多*Multiple Instance Learning*任务中有着非常好的表现，因此我们选择将该模型应用到此次的分类任务中。模型的框图如下所示：

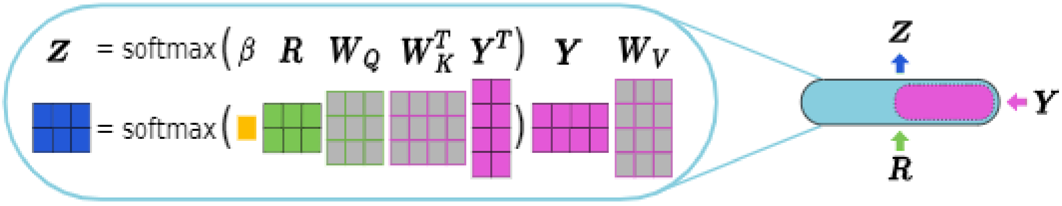


Figure 3: The layer `Hopfield` allows the association of two sets R (■) and Y (■). It can be integrated into deep networks that propagate sets of vectors. The Hopfield memory is filled with a set from either the input or previous layers. The output is a set of vectors Z (■).

该模型允许对两个数据集R和Y的关联，其中R为state pattern，作为模型的输入；Y为stored pattern，可以由外界输入，并存储在模型中，同时允许被训练。该模型隶属于*Encode-Decode*模型，并且运用了*Attention*机制。将其中RW_Q看作*Query*，YWk看作*Key*，YW_V看作*Value*，可见输出Z本质上就是*state pattern*基于当前*stored pattern*的*attention value*。*Hopfield Module*的流程图如下：

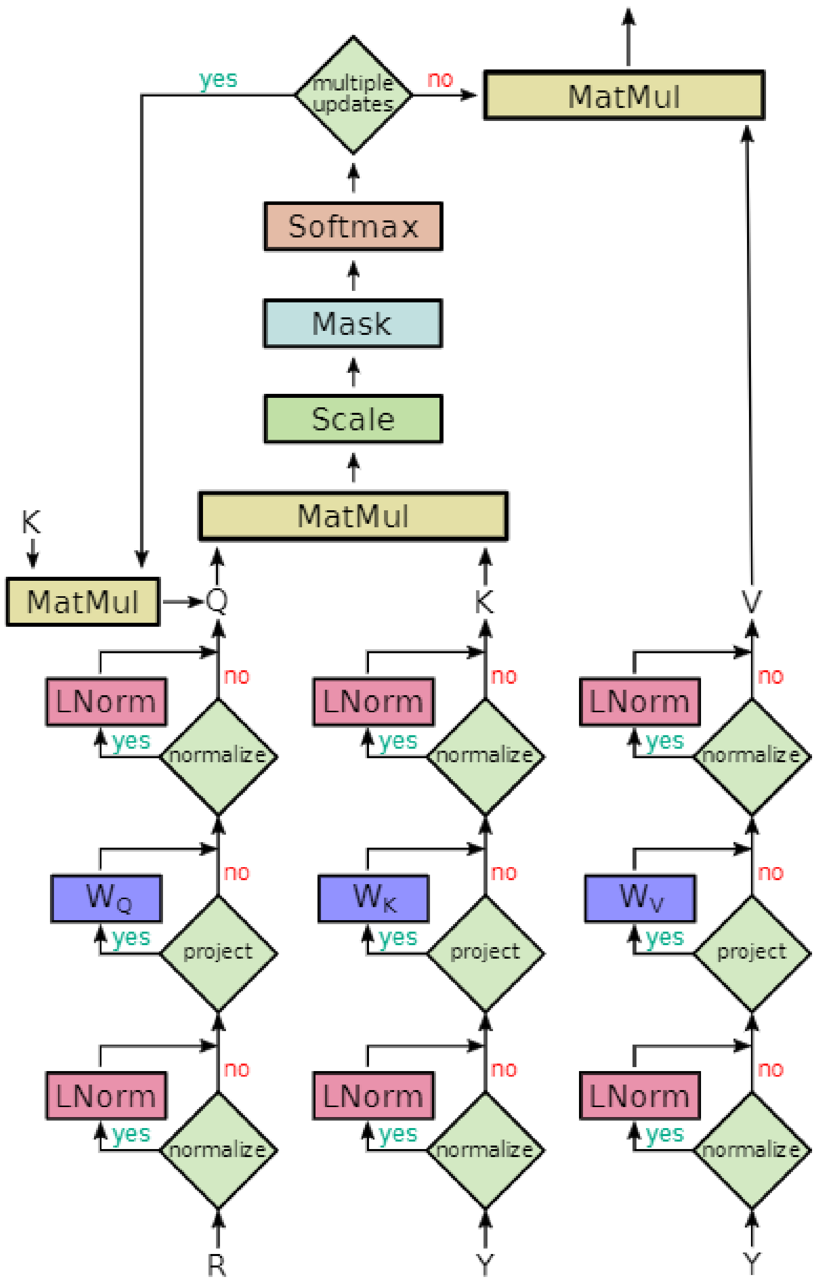
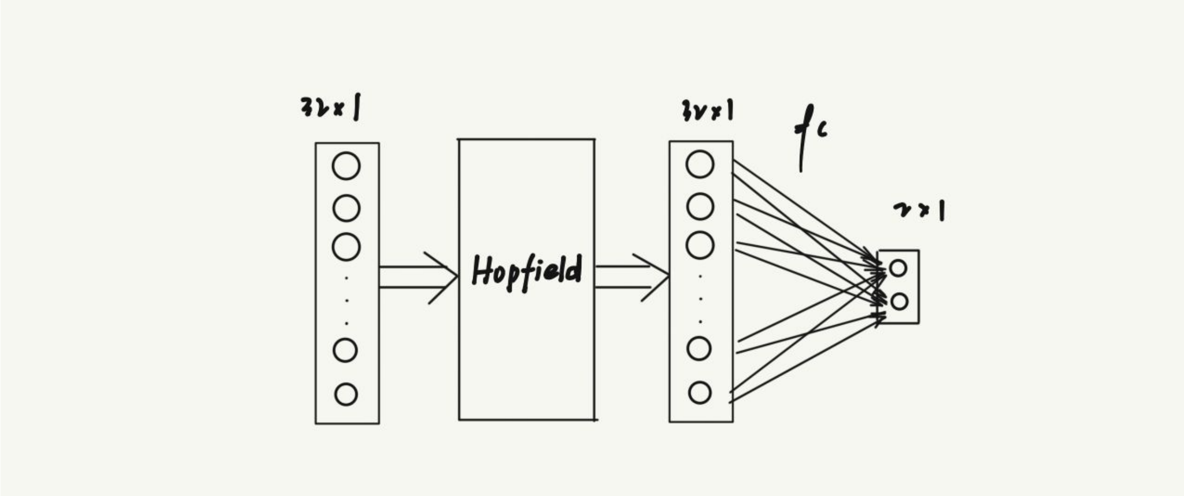


Figure A.7: A flowchart of the Hopfield layer. First, the raw state (query) patterns $\boldsymbol{R}$ and the raw stored (key) patterns $\boldsymbol{Y}$ are optionally normalized (with layer normalization), projected and optionally normalized (with layer normalization) again. The default setting is a layer normalization of the input patterns, and no layer normalization of the projected patterns. The raw stored patterns $\boldsymbol{Y}$ can in principle be also two different input tensors. Optionally, multiple updates take place in the projected space of $\boldsymbol{Q}$ and $\boldsymbol{K}$. This update rule is obtained e.g. from the full update Eq. (423) or the simplified update Eq. (424) in the appendix.

网络搭建

应用*Hopfield Module*搭建神经网络如下图所示：



其中，输入为代表一个青少年的32个*features*的向量，由于*Hopfield Module*是对*pattern*的更新，因此输出依旧保留原来的维度，也就是32×1，再将*Hopfield Module*输出的结果经一个*fully-connected layer*映射成一个二维向量，代表分类结果为0和1的概率。注意，虽然*Hopfield Module*的*stored pattern*和*state pattern*都接受输入，但在此模型中并没有输入*stored pattern*，而是通过*Hopfield Module*自行初始化一个*stored pattern*，输入的只有*state pattern*。

训练&测试

测试采用*k*折交叉验证，根据参数*k*划分数据集和测试集。选取*Loss function*为*CrossEntropyLoss*，选取optimizer为*AdamW*。每一次训练首先传入数据集，经网络前馈得到结果，利用*Loss Function*计算损失，反向传播至各个节点，并利用optimizer对节点参数进行优化。测试过程传入测试集，得到预测结果为0和1的概率，取概率更大的为预测结果，对比预测结果和真实tag。一个训练周期结束后，通过预测正确的次数和总次数的比值计算此次测试的准确率。