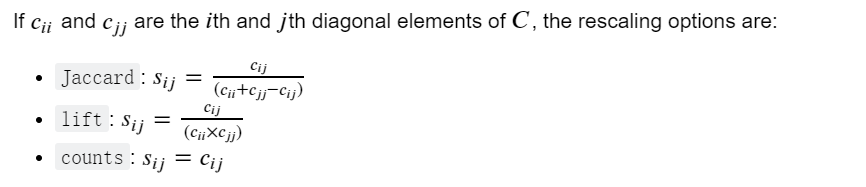
# 人工智能第三次汇报（12.18）

首先根据运行环境将提供的算法分为三类：cpu、cpu/gpu、pyspark并按照setting.md进行local环境的配置。环境配置成功后先从cpu环境的算法开始依次学习了解并运行。

#### Simple Algorithm for Recommendation (SAR) 简单推荐算法

SAR算法是基于项与项的相似性矩阵以及用户与项的亲和度矩阵实现个性化推荐。根据用户与各个项目的数据从中得到项与项根据共现能力的相似度矩阵以及用户与项的亲和度矩阵。项与项的共现能力指两个项目同时出现的次数，得到相似度矩阵之后可以根据Jaccard(中和), lift(小众)和counts(值最高)等系数进行缩放矩阵。



用户与项的亲和力可以受到权重与时间的影响，可以根据用户需求进行项目权重的修改例如设定用户与项目进行交互的时间越早权重越低，例如下式：



同时可以选择性选择不再推荐用户已经进行交互过的项目。最后通过这两个矩阵得到针对用户的个性化推荐排名。

#### **Bayesian Personalized Ranking (BPR)**

BPR基于隐含用户行为（购买点击等）利用可用的观测值作为正反馈为用户提供项目的推荐列表，核心算法是贝叶斯概率。BPR通过优化似然函数和先验概率p(Θ)，使模型参数Θ的后验概率最大化。

所有用户的似然联合概率为

用户偏好项目i而非项目j的个人概率：，

其中。

最大后验估计量的最终目标为，

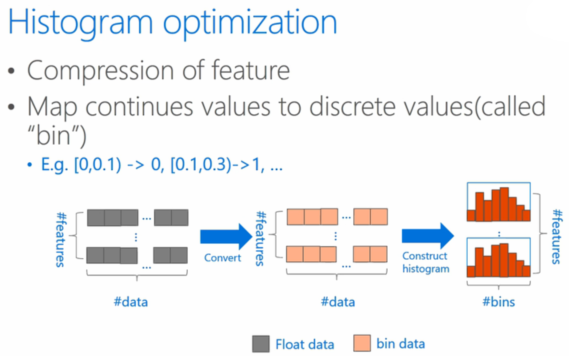
将其利用随机梯度下降法优化求解：



#### **LightGBM/Gradient Boosting Tree**

LightGBM是一个基于树的梯度提升算法。与其他基于树的梯度提升算法不同的是LightGBM采用按叶子生长（leaf-wise）的决策树生长策略，每次从当前所有叶子中找到分裂增益最大的叶子然后分裂进行循环，与按层生长的决策树生长策略（同时分裂同一层的叶子但不加区分的对待同一层的叶子）相比，可以得到更好的精度，缺点是可能会长出比较深的决策树产生过拟合因此算法中加入了最大深度限制防止过拟合。

另一个优点是直方图算法。将连续的浮点特征离散成k个离散值，并构造宽度为k的直方图。然后遍历训练数据，统计每个离散值在直方图中的累计统计量。在进行特征选择时，只需要根据直方图的离散值，遍历寻找最优的分割点。利用直方图算法代替预排序使得计算代价与内存消耗大大降低。



#### **Riemannian Low-rank Matrix Completion algorithm（黎曼低秩矩阵补全算法）RLRMC**

黎曼低秩矩阵完备化（RLRMC）是一种基于矩阵分解（vanilla）的矩阵补全算法。RLRMC算法假设评级矩阵M（大小为d×T）部分已知,d为项目的个数，T为用户的个数。M（i,j）处的条目表示第j个用户对第i个项目的评级。矩阵公式为M = LRT,L是d\*r的矩阵，R是T\*r的矩阵，r是需要提供的秩超参数。通常假设r<<d,T。它使用黎曼共轭梯度算法来解决优化问题。共轭梯度算法是介于最速下降法与牛顿法之间的方法。具体迭代思路如下：

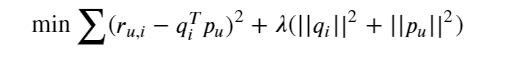
设我们要求解下列线性系统 Ax = b,令，rk为第k步的残差，令迭代方向，则下一个位置的迭代公式为，其中。

#### **Surprise Singular Value Decomposition (SVD)**

矩阵表示预测评级的公式为

其中表示用户u与项目i的预测评级，qiT和pu是项目与用户的潜在因素

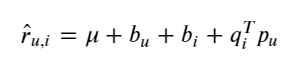
矩阵分解优化求解公式为：



其中为正则参数。

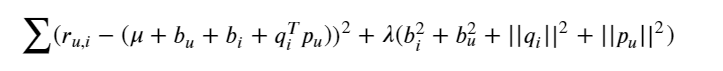
在SVD算法中引入了两个新变量bu,bi，用来表示用户偏差与项目偏差，旨在捕捉一些评级高于或低于均值的用户与项目。

因此模型公式更改为：

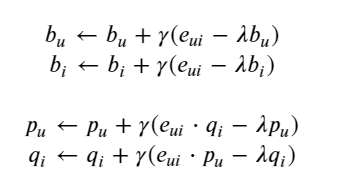


表示数据集中所有评级的均值。

正则优化问题修改为：



利用随机梯度下降方法求解的迭代公式为



其中是学习率，eui为模型对(u,i)的误差，

1604025043(1)

1. **[Vowpal Wabbit](https://github.com/VowpalWabbit/vowpal_wabbit" \t "http://localhost:8888/notebooks/Recommenders/examples/02_model_content_based_filtering/_blank)**

vovpal-Wabbit是一个快速的在线机器学习库，它实现了与推荐用例相关的几个算法。采用在线学习的方式，优化方法通常采用随机梯度下降法，主要优势是运行速度非常快，可以支持超大数据集的分布式训练场景。通过离线探索各种算法，在大量历史数据基础上训练一个高度精确的模型，然后将该模型运用到实际中，实时生成快速预测。

可以通过命令行参数实现不同的回归模型模型

（1）实现具有交互特征的线性回归，使用二次命令行参数并指定应组合的名称空间：‘-q ui’根据每个名称空间的第一个字母组合用户和项目名称空间。

（2）实现多项式回归，基本的多类logistic回归可以使用‘--oaa N’选项指定的One-over-All方法来完成，其中N是类的数量。

（3）实现调整输入数据为二进制结果的逻辑回归，为了进行逻辑回归，loss\_函数参数更改为“logistic”，目标标签切换到[0，1]。另外，确保在预测期间设置“-link logistic”，以将logit输出转换回概率值。

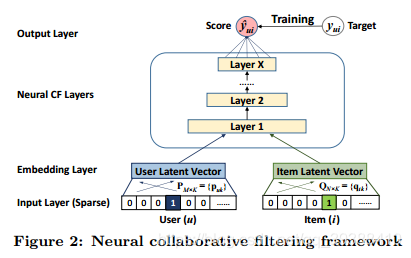
除回归模型外还支持两种不同方式的矩阵分解：

（1）svd算法（基于奇异值分解），使用“-rank”命令行参数调用。

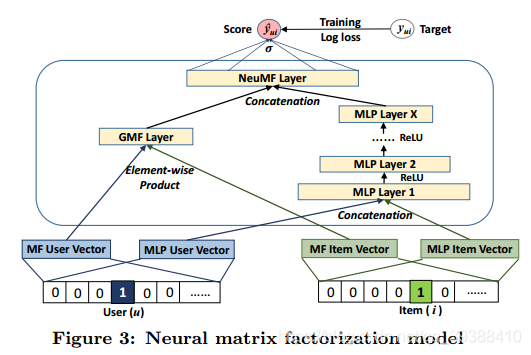
（2）基于因式分解机的矩阵分解，使用“-lrq”命令行参数调用。

1. **Neural Collaborative Filtering**

Ncf基于神经网络的协同过滤算法是一种运用神经网络实现推荐功能的算法,接受隐式反馈，通过算法提供给用户0-1区间内的项目推荐分数。首先通过embedding层将输入层的稀疏表示映射为一个新的潜在向量将得到的用户嵌入与项目嵌入输入神经协同过滤层中，将潜在向量映射为预测分数。



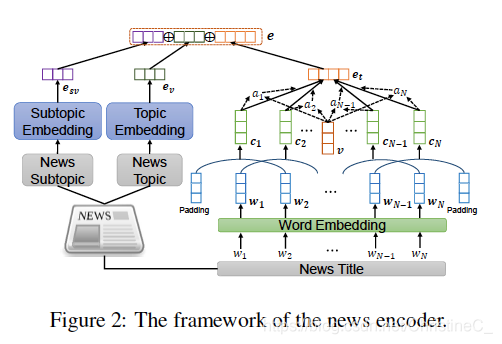
神经协同过滤层有三种实现方式，第一种为gmf，广义矩阵分解将向量对应元素相乘通过全连接层(tf.contrib.layers.fully\_connected)和激活函数得到预测的分数，第二种为mlp,使用多层感知机作为全连接层,Relu函数作为激活函数，第三种为NeuMF模型，他将gmf和mlp模型结合起来，为了使融合模型更加灵活，采用分别学习GMF和MLP的嵌入层矩阵的方法，并在两个模型的最后一层前将他们的输出向量连起来，然后传入最后的NeuMF layer，预测最终的分数，为了使范围控制在0-1使用了tf.sigmoid激活函数。



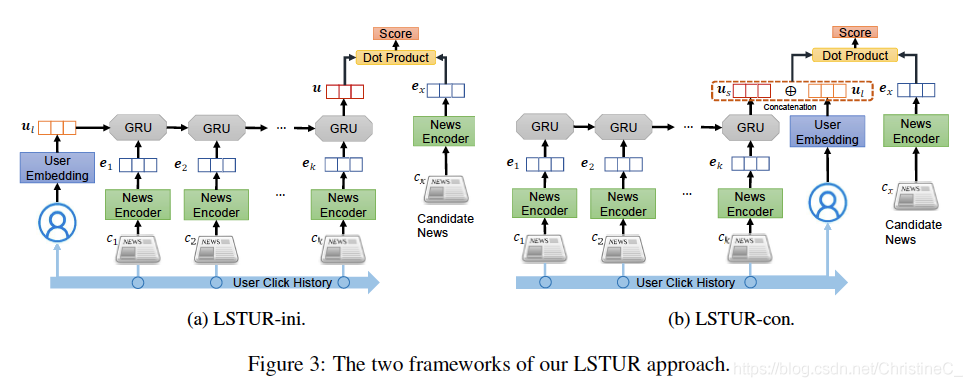
1. **LSTUR同时学习用户长期和短期兴趣的方法来推荐新闻**

主要包括两个编码器：新闻编码器、用户编码器。

新闻编码器从新闻标题和一级、二级类目中学习新闻的表示并使用注意力机制来选择重要的词。新闻标题编码器主要用到的思路为word embedding,cnn以及attention机制最终得到新闻标题的向量表示。类目编码器基于一级类目和二级类目进行embedding得到ev,esv,最终新闻整体表示为

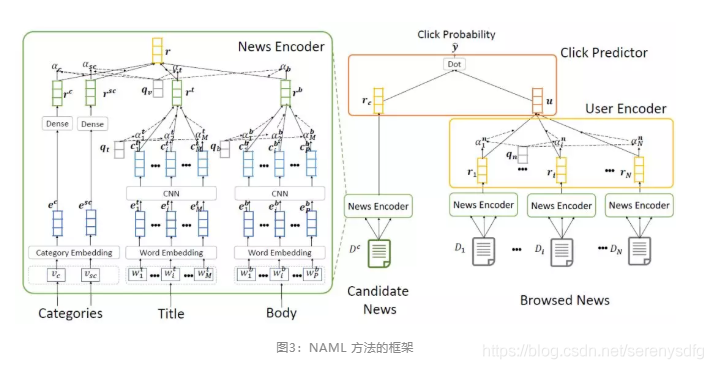


用户编码器根据用户的embedding中学习用户的长期兴趣，使用gru网络从用户最近浏览的新闻中学习短期用户兴趣，有两种方法用来结合长短期的用户兴趣：lstur-ini,lstur-con.第一种是使用长期用户兴趣来初始化短期用户兴趣学习中gru网络的隐藏状态，第二种是将长期和短期用户兴趣向量连接起来作为用户向量。用户过去一段时间内浏览的新闻按时间升序排列，用c={c1,c2,....,ck}表示，每一篇新闻用新闻编码器学习到的embedding表示{e1,e2,.....,ek}



1. **NAML: Neural News Recommendation with Attentive Multi-View Learning**

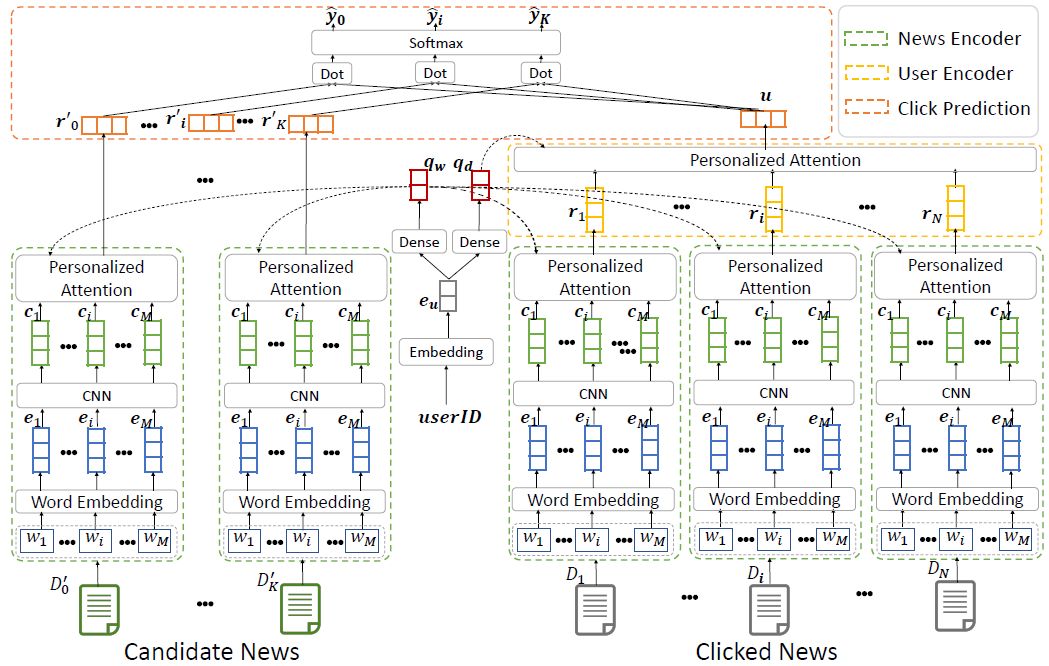
该模型提出了一种新视角学习的角度，通过融合多角度的新闻信息，结合用户侧与新闻测的数据特征来实现的深度学习模型。该模型主要包括两个编码器，一个是新闻编码器一个是用户编码器，最后通过一个dot production实现预测。点击预测模块用来进行新闻侧与用户侧的点积运算。用户编码器将用户浏览的历史记录输入到新闻编码器中进行编码之后利用attention机制对不同新闻加权处理。新闻编码器分为三个部分category，title，body。Category将主类和二级类的特征作为输入通过embedding层获取预训练的词向量然后通过一个dense处理得到向量表示，title和body部分分别以新闻标题与新闻内容做为输入，与lstur的新闻编码器相同通过embedding，cnn以及attention机制得到最终向量表示。



1. **NPA: Neural News Recommendation with Personalized Attention**

npa是指基于多视角学习和个性化注意力机制的新闻推荐算法。新闻编码器包括三层：一个将词语序列转化为语义向量序列的词嵌入层，一个用于建模局部上下文的 CNN 层，和一个词语级的个性化注意网络。在个性化注意力网络中，我们使用一个偏好查询向量和一系列输入表示计算相似度，构建个性化的输出表示。在词语级别，用户的偏好向量是由用户的 ID 嵌入向量经过一个非线性变换生成的。

在用户编码器内，我们使用一个新闻级别的个性化注意力网络。我们通过另一个非线性变换来得到用户的新闻偏好向量。最终在点击预测模块，候选新闻的点击分数由用户和候选新闻表示向量的内积计算。



1. **Attentive Asynchronous Singular Value Decomposition**

注意异步奇异值分解：  
A2SVD是对ASVD算法的拓展，在其基础上加入了attentive注意机制。

ASVD算法结合基准推荐模型、最近邻方法和因子分解模型对其进行改进得到新的推荐算法

基准推荐模型计算公式为bui=μ+bu+bi

最近邻方法的思路是根据用户个人偏好推荐与用户偏好最相关的物品，将利用数据进行统计得到的物品相关度改进为设置参数通过训练得到；*rui*=*bui*+∑*j*∈*R*(*u*)*wij*(*ruj*−*buj*)

将隐式数据加到模型中，其中N(u)为隐式反馈数据中与用户U相关的物品集合：  
*rui*=*bui*+∑*j*∈*R*(*u*)*wij*(*ruj*−*buj*)+∑*j*∈*N*(*u*)*cij*

为了解决模型中的参数两极分化问题，对邻域个数进行归一

*rui*=*bui*+|*R*(*u*)|−12∑*j*∈*R*(*u*)*wij*(*ruj*−*buj*)+|*N*(*u*)|−12∑*j*∈*N*(*u*)*cij*

因子分解模型可以将用户和物品分别映射到一个低维空间中，这样根据数据可以得到一个全局的向量表示，用于计算用户与用户、物品与物品以及用户和物品之间的相似度。   
可以通过SVD分解得到向量表示

异步svd算法将模型中的wij,ci转换为低维空间内的表示，主要思路是将用户表示为其相关物品向量的累加

*rui*=*bui*+*qTi*(|*R*(*u*)|−12∑*j*∈*R*(*u*)*xj*(*ruj*−*buj*)+|*N*(*u*)|−12∑*j*∈*N*(*u*)*yj*)

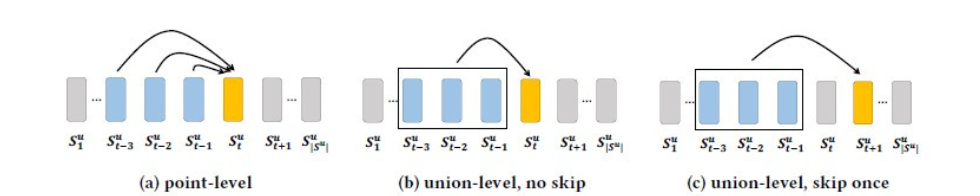
S2SVD在此模型基础上加入了用户personalized attention机制用来分析用户的长短期喜好

1. **CASER（ConvolutionAl Sequence Embedding Recommendation）**

Top-n序列推荐模型

由于用户的表现即有因为喜好而存在的长期静态的行为，也有根据用户最近的活动或交互产生的活动与交互，因此推荐系统中除了用户的日常喜好也需要关心事件序列的发生新近关系。

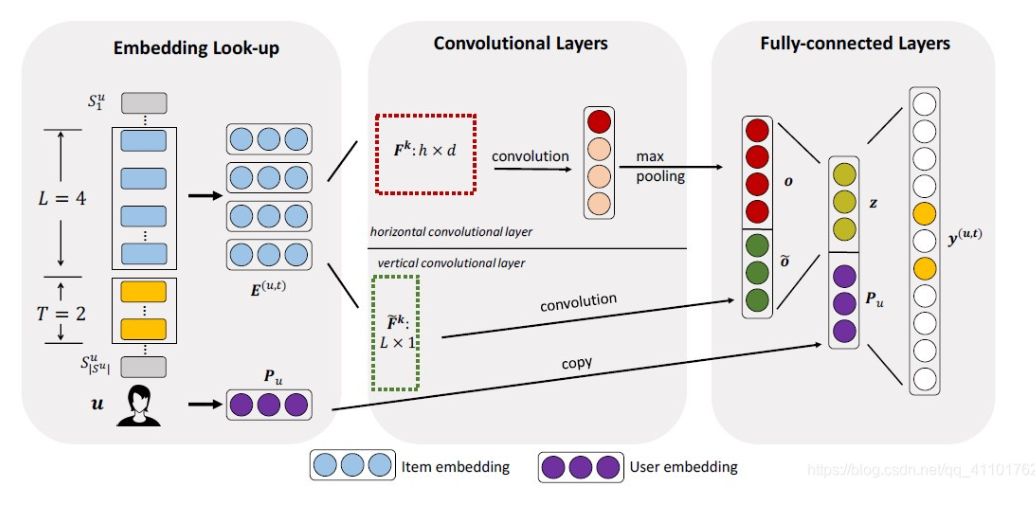
原先的top-n序列（图a）推荐只考虑了基于前L个动作推荐下一个动作，没有考虑联合序列的影响（图b）,以及包括跳过的序列行为影响（图c）即过去的行为不是立即对下一个行为产生影响但对后续的行为产生影响。



Caser(卷积序列嵌入推荐)由三个组件组成:嵌入查找层、卷积层和全连接层。嵌入查找层通过向神经网络嵌入先前的项目来捕获潜在空间中的序列特征。将长为L的序列映射成embedding，将其堆积在一起。将前L个item表示为一个L×d的矩阵E，d是潜在维度,矩阵E的行保持着交互物品的顺序。然后我们将这个矩阵E视为L个物品的一个image，通过不同的卷积过滤器去学习 image的局部特征。用户的特征是用LFM生成的。

第二个组件是卷积层，水平卷积网络中的Fk滤波器会从上到下滑动，并与物品的横向维度发生相互作用，水平滤波器通过嵌入e与每一个连续的h项进行交互。通过学习嵌入和过滤器，可以将编码目标项预测误差的目标函数最小化。垂直卷积网络中的F~k滤波器与列交互从左到右滑d次在矩阵E上，产生垂直c˜k卷积结果。

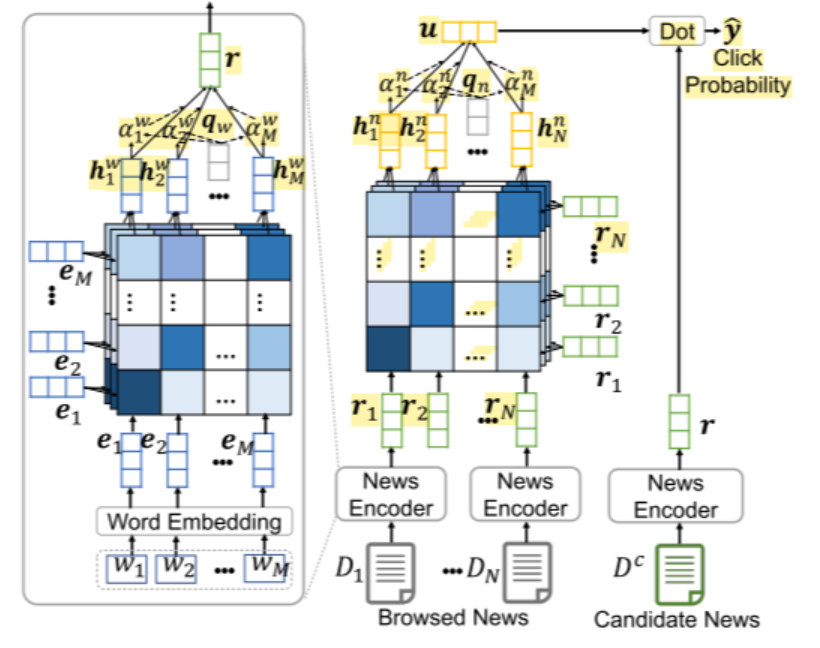
第三个组件为全连接层。将两个卷积层的输出连接起来投影到z得到卷积序列嵌入。第一个组件得到的用户嵌入pu与z连接起来投射到输出层中，取输出层最高的n个值得到top-n推荐。



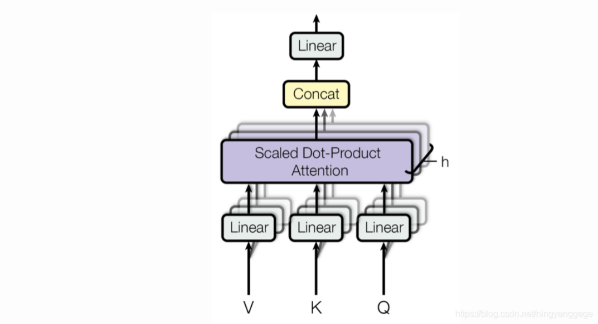
1. **NRMS model(Neural News Recommendation with Multi-Head Self-Attention)**

多头自我注意机制

NRMS包括两个编码器，新闻编码器和用户编码器，新闻编码器包括两层词嵌入embedding层和self-attention层。把新闻转换为词矩阵后直接使用self-attention计算得到新闻表达。用户编码器是把浏览过的新闻通过新闻编码器计算之后又加了一层self-attention得到用户侧的表达，最后通过click将新闻编码器与用户编码器计算得到的两个vector做一次内积得到分数。



算法的核心是多头部的自我注意机制。注意力机制用于决定输入的哪个部分需要更加关注并从关键的部分进行特征提取得到重要的信息。多头注意力机制利用多个查询Q=[q1,q2,.....]平行的从输入信息中选取多个信息，每个注意力关注输入信息的不同部分然后拼接得到提取的特征信息。自注意力机制中K=V=Q,Q是对自我输入的变换而非来自于外部。



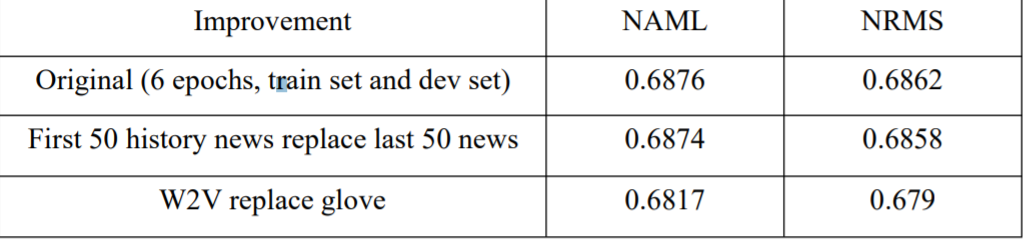
1. **改进方案**

**14.1局部融合方案**

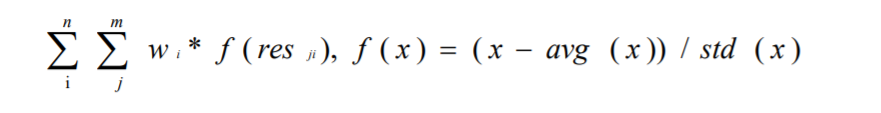
首先提出两种方案：

1. 后五十个记录代替前五十个记录进行训练，目的是为了让用户兴趣更接近当前时段
2. 表示词向量时利用w2c代替glove embedding

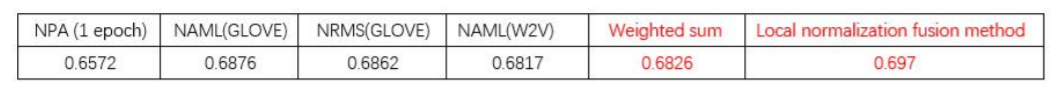
这两种方法得到的评价结果如下：



由于这两种方案的auc分数均低于原方案，因此基于原方案提出融合三种不同的模型npa(基于个性化注意力机制的推荐算法),naml(基于多视角学习的推荐算法),nrms(基于多头部自我注意力机制的推荐算法)的推荐算法，因为三种算法的分布各不相同，所以不能采取简单的归一化方式，具体融合的公式如下：



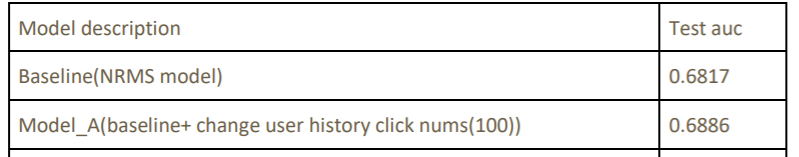
融合的结果如下：



**14.2 多模型融合**

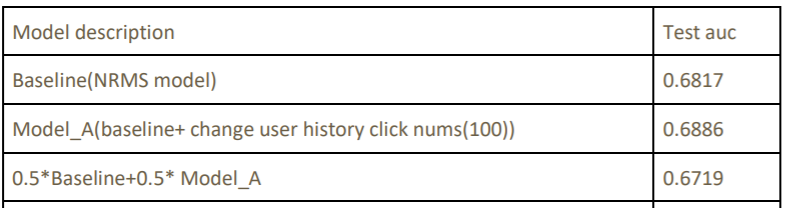
首先基于nrms对其做出一个改进：将用于刻画用户兴趣的用户点击新闻纪录从50条增加到一百条，目的是利用更大的数据更好的刻画用户的新闻偏好与兴趣。

评估结果如下，可以看到用于用户表达的数据增加能够提高评估的效果。

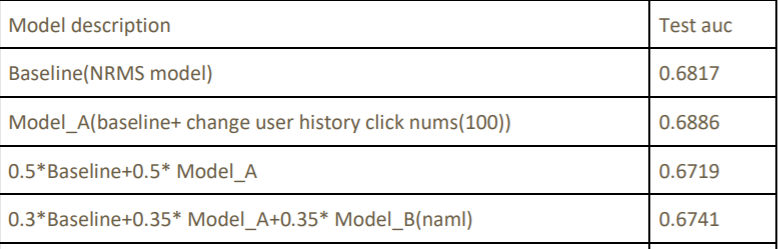


其次使用多模型推荐系统可以提高推荐系统的准确性

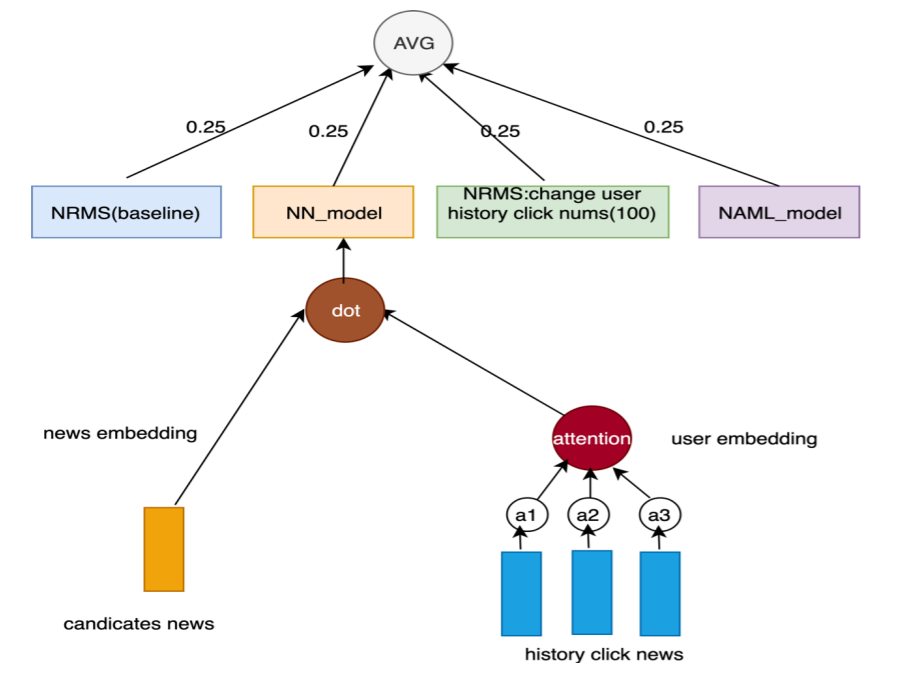
1. 对基础算法nrms与改进的算法做一个融合，得到评价结果如下：



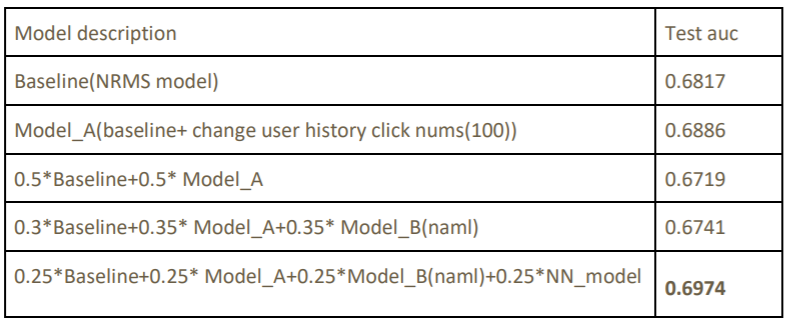
1. 在上述两种算法上加上另一种推荐算法naml进行融合，得到的评价结果如下：



1. 引入一种新的模型，并将新模型与上述三种模型的结果求平均，模型构架如下：



得到的评价结果如下，可以看出将模型融合之后得到的分数比之前更高。

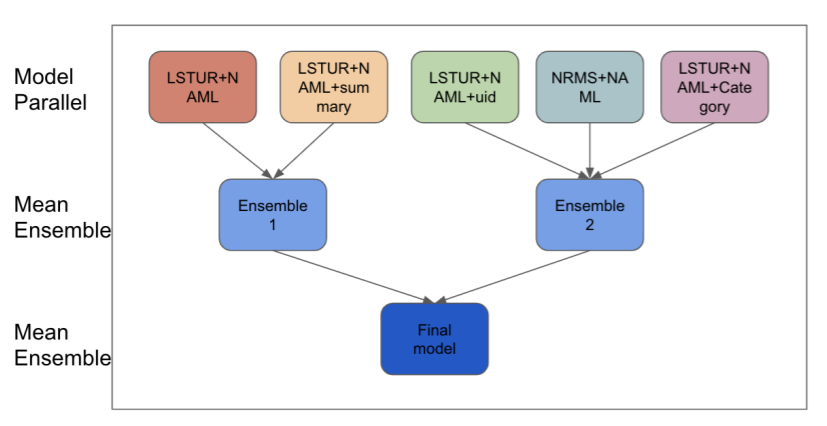


**14.3 多模型集成融合**

基于提供的算法lstur(基于长短期用户兴趣的推荐算法)，naml和nrms算法进行集成融合并且寻找更多有用的信息加入模型中。

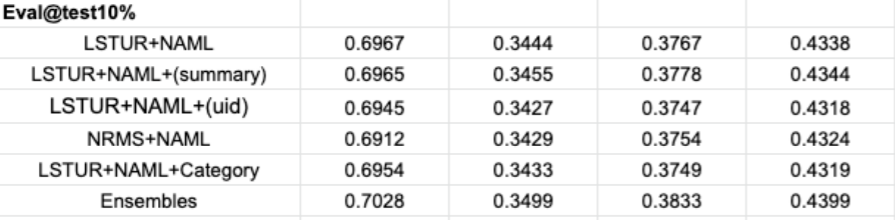
1. lstur+naml:使用专注的多视角学习方式来编码新闻，利用lstur的长短期学习用来表示用户行为
2. Lstur+naml+(summary):将原naml中的abstrct改为summary
3. Lstur+naml+(user-id):使用最大熵策略在整个用户列表中选择可学习的user-id
4. Nrms+naml:使用专注的多视角学习方式来编码新闻，利用nrms的多头自注意机制用来表示用户行为
5. Lstur+naml+category：增加新闻类别属性

融合模型的框架：



模型的评估结果如下：

1608188139(1)



**14.4多视角学习的nrms算法**

对基于多头自注意机制的nrms按照naml的多视角学习方式进行优化，即将nrms新闻编码器中输入的新闻内容替换为各个属性例如新闻标题、类别、正文等等，基于这些属性编码的结果利用attentive pooling将上述单元输出的邻域点特征集聚合到一起同时可以自动学习和聚合特征集中有用的信息。聚合之后利用mlp计算每个点击预测分数。

模型的总体框架如下：



其中进行测试的模型组合有：

Title + abstract + category + sub-category (TACS)

Title + abstract + body + category + sub-category (TABCS)

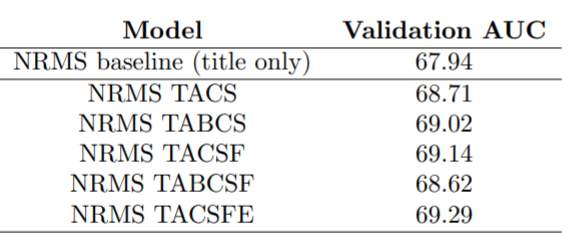
Title + abstract + category + sub-category + features (TACSF)

Title + abstract + body + category + sub-category + features (TABCSF)

Title + abstract + category + sub-category + features + entities (TACSFE)

其中features是新设置的一个新闻特征，旨在捕捉新闻的流行程度，这些特征包括：有多少用户点击了一篇文章，这些用户每次点击的新闻数量的平均值和标准差，以及这些用户在这段时间内点击的新闻总数的平均值和标准差。

评估结果如下：



1. **总结与感想**

通过本次人工智能实验——新闻推荐算法的学习，我学到了很多推荐算法以及深度学习、机器学习、神经网络等等算法在推荐系统中的应用，同时也在学习算法的过程中学习了tensorflow的python库的使用。在提供的所有算法中印象最深刻接触也最多的就是naml,npa,nrms,lstur等等新闻推荐算法，虽然我们在此基础上的改进算法没有能够运行成功但是我们也在算法的改进过成功体会到了这些算法的融合过程以及各个优化层，嵌入层，编码器之间的联系，收获颇丰。