7月15日

一、当前学习进程简单总结

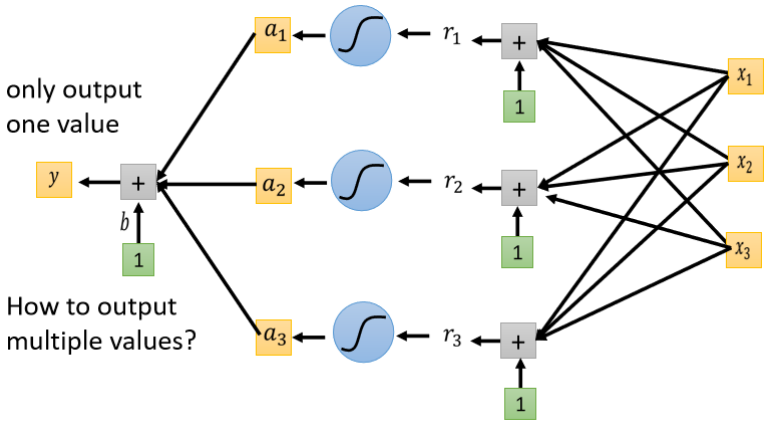
[一文读懂LeNet、AlexNet、VGG、GoogleNet、ResNet到底是什么？ - 知乎 (zhihu.com)]

[一文读懂VGG网络 - 知乎 (zhihu.com)]

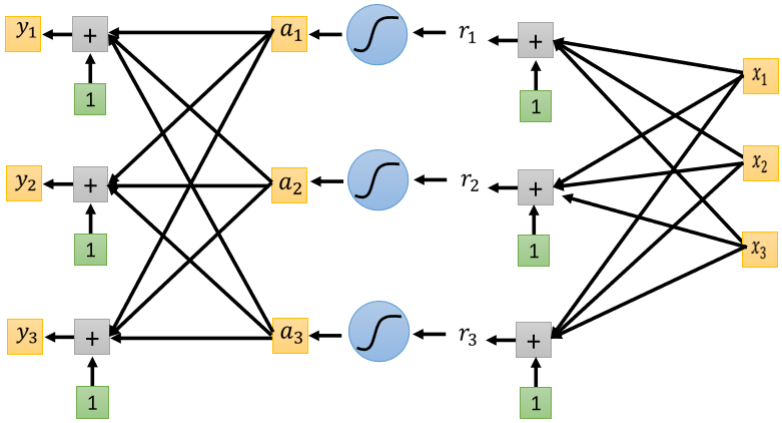
[人人都能看懂的LSTM - 知乎 (zhihu.com)]

简单的概述：

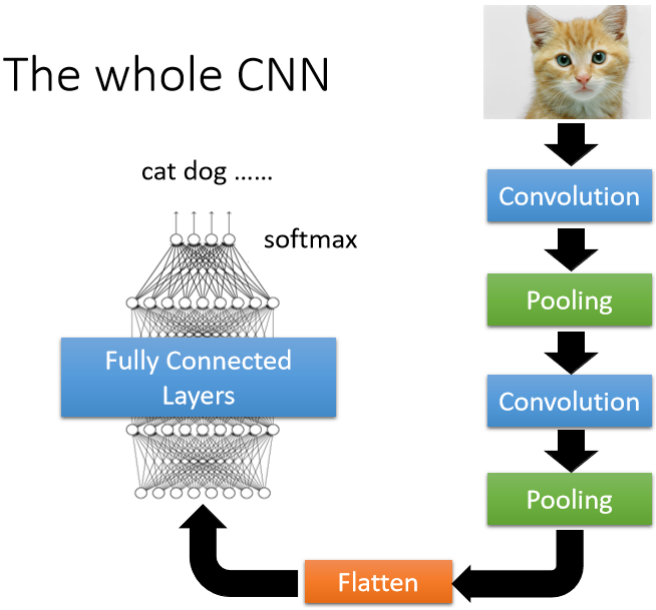
1. **全连接（Fully connected）**网络解决**Regression**问题，**输出是一个数值**



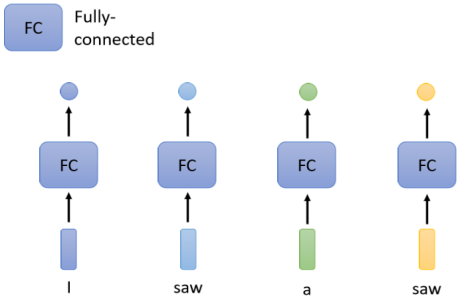
2. 对于**Classification问题**，**输出多个数值**，输出前加上**Softmax**层，且loss函数使用**Cross-Entropy**



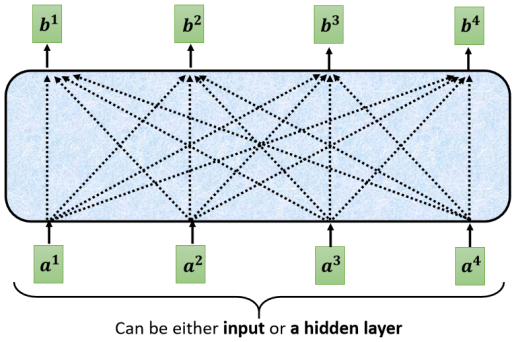
3. 接下来来到**图像处理**的问题，如果直接使用全连接架构，由于图像拉直后的向量维度非常高，使用全连接会导致参数的数目非常大，训练比较困难，且增加了 Overfitting 的风险。为了减少维度，通过**卷积核**对图像进行特征提取来降维，通过**Pooling**操作来降维，最后提取出的表征图像特征的向量维度就小了很多，最后再接上全连接层就可以了，这就是**CNN（Convolutional Neural Network）**的思想。



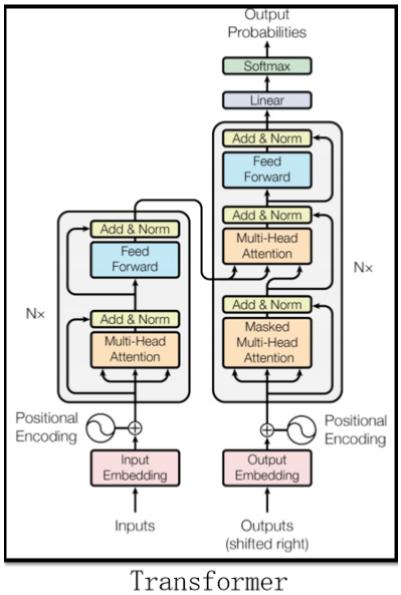
4. 之前处理的输入都是单个向量，接下来考虑输入是多个向量的问题，就比如词性标注的问题，I saw a saw，如果直接使用全连接的架构，你不可能期望两个输入的saw会有不同的输出，这是因为在全连接的架构中不会考虑输入的上下文。



于是想到通过**Self-Attention**来让输出考虑一整个输入序列的信息。如果认为问题的上下文有比较复杂的相关性，可以使用**Multi-head Self-attention**，如果需要考虑输入的位置信息，则需要做**positional encoding**,之后把产生的position vector加到输入中就使得输入包含了位置信息。



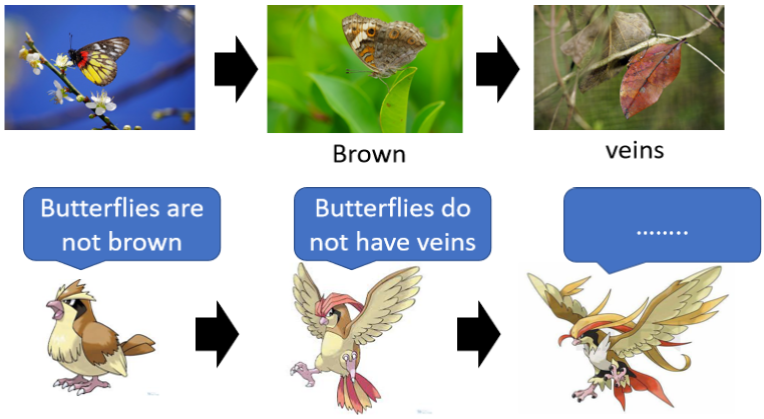
5. 如果输入多个向量，输出多个向量且不知道输出的具体个数（就比如翻译问题），这就需要用到**Sequence-to-sequence**的模型，这一般会使用一个**Encoder-Decoder**的架构，当前最有名的一个**seq2seq's model**就是**Transformer**：



Transformer

https://arxiv.org/abs/1706.03762

6. 假如我们要用我们的network来生成一些东西，要怎么做？这就是**GAN（Generative Adversarial Network）**要做的事情。之所以有对抗（Adversarial ），是因为GAN同时包含了一个generator和一个discriminator，generator是做假钞的，discriminator是验钞机，通过两者的对抗，达到共同的进步。



二、跨域推荐（CDR）入门

1. Embedding

(91 封私信 / 80 条消息) 怎么形象理解embedding这个概念？ - 知乎 (zhihu.com)

总结一句话就是，为了提高召回速度和效率，相当于把召回模型的inference转换成了通过embedding向量内积方法快速得到用户对视频得分的方法。从这个例子可以看出embedding的应用之一：**通过计算用户和物品的Embedding相似度，Embedding可以直接作为推荐系统的召回层或者召回方法之一。**

再对embedding在召回方面的应用浓缩总结一下就是：**通过计算用户和物品或物品和物品的Embedding相似度，来缩小推荐候选库的范围。**

除此之外，通过总结目前主流的CTR预估模型比如wide&deep，deepFM，PNN和DCN等等可以发现，embedding还有一个非常普遍的应用就是**实现高维稀疏特征向量向低维稠密特征向量的转换**，通俗来讲就是把离散特征经过独热编码后的稀疏向量表达转化成稠密的特征向量表达。

或者从另一个角度看，embedding本身就是对事物的多维度特征表示，因此在ctr预估模型中，**训练好的embedding可以当作输入深度学习模型的特征**

总结

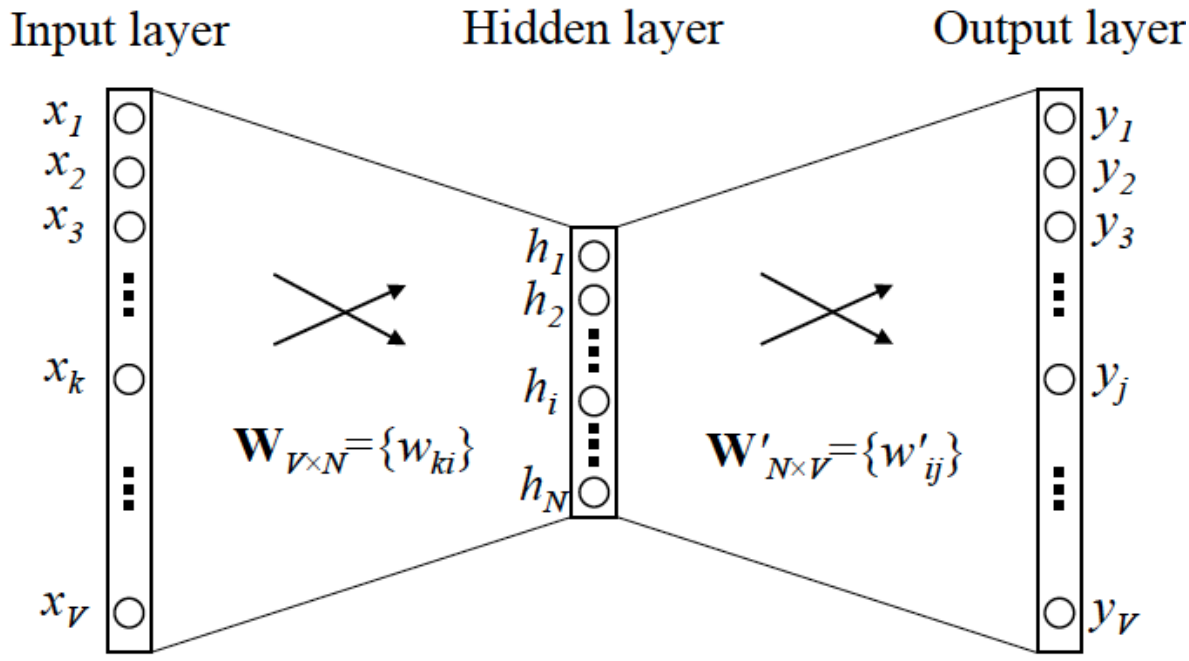
通过这篇文章对embedding的分析，总结embedding有以下三个作用：

- 通过计算用户和物品或物品和物品的Embedding相似度，来缩小推荐候选库的范围。
- 实现高维稀疏特征向量向低维稠密特征向量的转换。
- 训练好的embedding可以当作输入深度学习模型的特征。

无论embedding在模型中最终起到哪些重要的作用，在对embedding的本质理解上，它自始至终都是用一个多维稠密向量来对事物从多维度进行的特征刻画。

2. Word2Vec

[NLP] 秒懂词向量Word2vec的本质 - 知乎 (zhihu.com)



当模型训练完后，最后得到的其实是**神经网络的权重**，比如现在输入一个 x 的 one-hot encoder: [1,0,0,...,0]，对应刚说的那个词语『吴彦祖』，则在输入层到隐含层的权重里，只有对应 1 这个位置的权重被激活，这些权重的个数，跟隐含层节点数是一致的，从而这些权重组成一个向量 vx 来表示x，而因为每个词语的 one-hot encoder 里面 1 的位置是不同的，所以，这个向量 vx 就可以用来唯一表示 x。

注意：上面这段话说的就是 Word2vec 的精髓！！

3. 机器学习与推荐算法公众号文章

MLP or IP：推荐模型到底用哪个更好？ (qq.com)

MLP难以学习向量内积

推荐系统之FM与MF傻傻分不清楚_张小磊的博客-CSDN博客

推荐系统核心技术是协同过滤技术 其中协同过滤技术中最常用的便是矩阵分解技术，矩阵分解将用户与物品嵌入到一个共享的低维隐空间内。

我们将用户和物品构造成一个二维矩阵（后称**U-I矩阵**），其中每一行代表一个用户，每一列代表一个物品，由于U-I矩阵的稀疏性，许多用户对物品没有过相应的评分，那么预测某一个用户对某一个物品的喜爱程度便成了推荐系统的主要任务。**矩阵分解的思想**是**将U-I矩阵分解为两个低秩稠密的矩阵P和Q，其中P为用户的隐因子矩阵，Q为物品的隐因子矩阵，通过这两个矩阵来预测用户对物品的评分**，也即：

$$\hat{r}_{ui} = p_u^T q_i$$

但是考虑一些额外因素：

1、一些用户给分偏高，一些用户给分较低，

2、一些物品本就比较受欢迎，一些物品不被大众所喜爱，

3、评分系统的固有属性，与用户物品都无关。因此，考虑到这些因素，将用户对物品的预测函数升级为：

$$\hat{r}_{ui} = \alpha + \beta_u + \beta_i + p_u^T q_i$$

其中为α全局偏置，β_u为用户偏置，β_i为物品偏置。

本质上，MF模型是FM模型的特例，MF可以被认为是只有User ID 和Item ID这两个特征信息时的FM模型。接下来，举个栗子方便大家理解FM是如何在仅有User ID 和Item ID时退化成MF模型的。假设用户集合为$U=\{A,B,C\}$，物品集合为$I=\{TI,NH,SW,ST\}$，我们以图1中的$x^{(1)}$为例，在仅包含用户ID和物品ID信息时，特征维度$n=|U\cup I|=7$，则特征向量$x^{(1)}=[1,0,0,1,0,0,0]$，即为用户ID和物品ID的one-hot表示的拼接，由于特征向量$x^{(1)}$中第一位和第四位为非零元素，因此二阶FM公式便如下所示：

$$\begin{aligned}\hat{y}(\mathbf{x}) &= w_0+\sum_{i=1}^{n}w_ix_i+\sum_{i=1}^{n}\sum_{j=i+1}^{n}<\mathbf{v}_i,\mathbf{v}_j>x_ix_j\\ &= w_0+\sum_{i=1}^{7}w_ix_i^{(1)}+\sum_{i=1}^{7}\sum_{j=i+1}^{7}<\mathbf{v}_i,\mathbf{v}_j>x_i^{(1)}x_j^{(1)}\\ &= w_0+w_1+w_4+<\mathbf{v}_1,\mathbf{v}_4>\end{aligned}$$

由此，将上述公式和矩阵分解公式联系起来，w_0可看作是MF中的全局偏置，w_1为MF中的用户偏置，w_4则为MF中的物品偏置。因此，MF是FM在仅有用户ID和物品ID信息下的特殊形式。

[由Logistic Regression所联想到的... (qq.com)]

[推荐系统从入门到接着入门 (qq.com)]

为了解决信息过载(Information overload)的问题，人们提出了推荐系统。

SVD++

人们后来又提出了改进的BiasSVD，还是顾名思义，两个加号，我想是一个加了用户项目偏置项，另一个是在它的基础上添加了用户的隐式反馈信息，还是先上公式：

$$\underbrace{arg\,min}_{p_i,q_j}\sum_{i,j}(m_{ij}-\mu-b_i-b_j-q_j^Tp_i-q_j^T|N(i)|^{-1/2}\sum_{s\in N(i)}y_s)^2$$
$$+\lambda(||p_i||_2^2+||q_j||_2^2+||b_i||_2^2+||b_j||_2^2+\sum_{s\in N(i)}||y_s||_2^2)$$

它是基于这样的假设：用户对于项目的历史评分记录或者浏览记录可以从侧面反映用户的偏好，比如用户对某个项目进行了评分，可以从侧面反映他对于这个项目感兴趣，同时这样的行为事实也蕴含一定的信息。其中N(i)为用户i所产生行为的物品集合；ys为隐藏的对于项目j的个人喜好偏置，是一个我们所要学习的参数；至于|N(i)|的负二分之一次方是一个经验公式。

[推荐系统遇上深度学习(二十)--贝叶斯个性化排序(BPR)算法原理及实战 - 简书 (jianshu.com)]