下面开始正式（瞎）讲（b）解（b）表示学习的一些很基础很基础的模型和算法了。如果谁一个知识门类有基础的，有在基础上衍生的，我们可以把基础的比喻为“树根”，然后树根衍生出来树干和树叶，我们知道了基础的之后，再看到树干树叶，就可以灵光一现（我猜的，我是没现过）：啊！这个论文是根据这个树根bb出来的！今天介绍的就是**表示学习**的一些树根，只是我看到的，**包括但是不限于**啊，不限于……

还记得上回书说了啥不？忘了也没啥关系，回去稍微看下啥是知识图谱三元组，记住字母E代表的就和实体（entity）有关，其中h是头实体，t是尾实体；字母R代表的就和关系（relation）有关，然后知道表示学习是把一个三元组形式表示的知识图谱倒腾成一堆向量，这些向量里的数字是你没必要具体知道的隐藏语义信息就行了（emmm，如果我这里说的看不懂，就回去看一下上一篇文章吧，此处应该有链接）。还有！这里只介绍到**模型**！也就是参数堆给你我就跑了，我不介绍求解啊（其实是因为我不太会，逃……）

在我这里，仅在我这里，我认为目前的表示学习的模型和算法可以分成这么几类：

1. 张量分解的，包括RESCAL和LFM
2. 神经网络相关的，包括SLM,NTN
3. 距离模型、翻译模型相关的，包括SE，TransE，TransR，TransH这种火到炸的trans系列

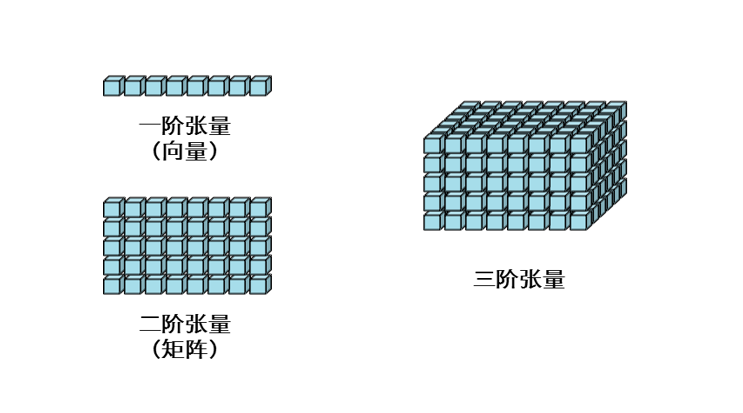
SME,没想好归在哪里，但是总在说

下面，开始一一瞎说阿不介绍，按照这个时间的推移，先是有张量分解，然后神经网，最后是翻译模型，近几年就是神经网+翻译模型非常非常火，火到一眼望去各种trans啥啥啥，表示学习不止翻译模型！表示学习不止翻译模型！表示学习不止翻译模型！重要的事情说三遍！

**张量分解**

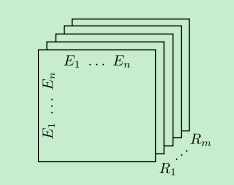
看到这四个小红字，你脑袋里出现的第一个问题，我猜和我一样：啥是张量？

这个张量啊，你去百度，以及知乎，会得到很多，我觉得很玄学的答案，从物理到数学各种专业且高大上的解释，让我这个本科差点拿理学学位，硕士跑路到计科过的跟数学系似的半吊子感觉十分对不起母校……咳咳，其实不理解那么高深也不影响理解。张量（Tensor），这个词在论文里只要出现，你就可以认为这里要出现一个三维数组了，就是是a\*b\*c这种规模的数组。网上扒的图（<https://blog.csdn.net/yixianfeng41/article/details/73009210>）：



论文中出现的张量（Tensor），你就可以理解成这是个长成图里的三阶张量样子的东西。

RESCAL和LFM（latent factor model）这类模型，是为了干这样一件事，给我一对头尾实体和一个关系，如果这个头实体和尾实体之间存在这个关系，那我就打个标签“1”，如果没有这个关系，那我就打个标签“0”。然后我们手里现在有一堆实体和一堆关系，头实体尾实体关系排列组合整吧整吧就是一个“张量”的样子。这个思想在RESCAL中有一个很帮助理解的图：



这个图就是一个三阶张量的样子，每个面表征一个关系，每个面上，横坐标和纵坐标都是实体，每个面上的每个数就是这个位置的头尾实体和关系的标签是0还是1。

好了，现在我们知道要干的是啥了。下面问题来了，现在这个张量的德行，你给我头尾实体和关系，我就静静的看着你表演如何打0和1的标签。这就跟尹志平欺负杨过只教心法不教招式一样，这个张量的意思很重要，然而给你这个你也整不出来01标签。

招式是什么？就是张量分解。这个0和1的标签怎么来？我们用一个长成



这样的东西来算。这是个啥听我慢慢讲……

对于**每一个实体**，我都给它整成一个长度是r的向量；对于所有的n个实体，它们就组成了一个矩阵，就是这里的lh和lt，分别表示头实体的矩阵和尾实体的矩阵，量纲是n\*r这种的（其中n是实体总数，r是每个实体都是一个长度为r的向量）；然后**每一个关系**，都整成M这样，量纲是r\*r这种的，这三个矩阵乘起来，n\*r的r\*r的r\*n的，最后得到的是一个n\*n的矩阵，每一个元素都是一个介于0和1之间的数，表征对应位置的头尾实体有这个关系的概率是多大。以上是对一个关系来说的，我们如果一共有**m个关系**，每个关系都这么操作，最后就能得到一个n\*n\*m的张量。所以现在我们要求的参数们（上一篇文章我说要堆这的W们）也就出来了，就是上述的这些矩阵中的每个元素。长成这样的这个公式，就可以理解为**张量分解**。

以上这个关于张量分解是啥我都是把着RESCAL说的。这个LFM和RESCAL核心思想都是这种张量分解的模型，只不过在这个模型的基础上还可以各种作妖，比如LFM在外面还套了sigmoid函数，然后sigmoid函数里也不止lh\*Mr\*lt，还考虑了只头实体和关系的关系，只尾实体和关系的关系等等等等；此外它还改进了M，把关系的矩阵M认为是一堆向量的组合，这样能减少参数……总之就是各种以张量分解为基础的各种作妖，但是核心还是不变的。

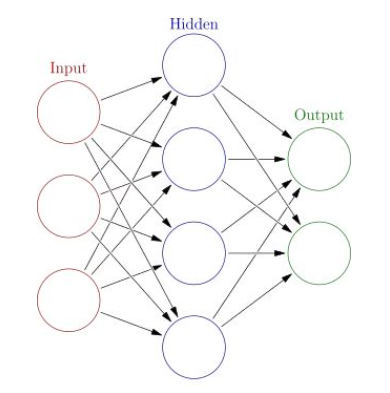
多说一句，有的还会把这样子的模型叫做“**双线性模型**“，这个双线性最开始和张量一样，也是搞得我想死，一群搞数学的用的名词为啥查着查着总是和物理有关……后来我突然想通，这个双线性，其实就是这个意思：你看，y=wx这种，是线性模型，因为w是对x的一个线性操作，y=wxm这种，就是双线性模型了，第一个线性是w对x的线性操作，第二个线性是m对x的线性操作，所以你就可以把双线性，理解成，两个线性，就是这么简单粗暴……

还有一个是我的小疑问，我自己的理解，强烈的不知道对不对的那种，要是有大佬知道一定要告诉我啊！我看论文得到的结论，LFM对于同一个实体，做为头尾实体向量表示不一样，但RESCAL是一样的，就是这个实体的表示就是一个向量，无论是头实体还是尾实体，但是我不确定啊！论文这里没看太明白瞎猜的！欢迎有明白人救救我！

**神经网**

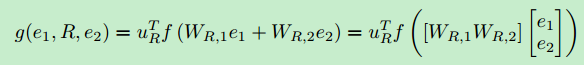
这里用SLM和NTN来举例子。这两个模型用神经网的思路是这样婶的，对于每一种关系，我都用一个神经网来表示，判断两个实体有没有某种关系，就把这两个实体丢进这个关系的网络中，看输出。理想状态有关系输出1，没关系输出0，实际上这个网络会输出一个介于0和1之间的数，来表示这俩实体有这种关系的概率。

这个神经网络啊，基本上搜教程看到的都是长这样的：



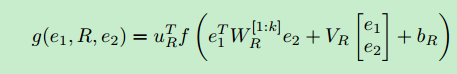
就是很多层，一层很多神经元。我们要介绍的SLM全称是single layer model，单层模型，一会把模型摆上你会发现，这个东西相当于只用了一个神经元，所以我个人认为这个叫神经网有点过了，应该叫感知机……

关于神经网，这里就不多说了……下面关门放SLM：

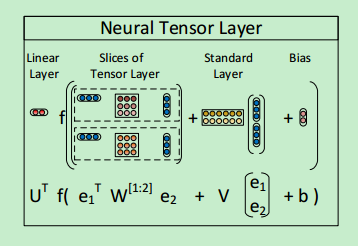


看到这个别懵，听我慢慢讲啥意思。对于一个神经元，常规操作是把输入乱七八糟组合出花，然后组合丢入神经元，之后再用一个函数（基本都是sigmoid这种平滑可导单调的函数）操作下这堆组合，就得到了这个神经元的输出。SLM思路很简单，一个神经元的输入直接就是头实体和尾实体（e1和e2）向量表示的线性组合（有的不叫线性，说这是个concatenate的操作，然而我觉得，这个就是线性组合，嗯），f是tanh，这个线性组合tanh一下就是神经元的输出；然后这个前面的Ur可以当作一个系数，一个线性的系数，乘上了这个输出。你看，相当于只有一个神经元，一层也没毛病，所以这个叫做single layer model。这个式子是一个关系的网络模型，是一个关系的表示，这个关系的表示，就可以看成是这个网络的参数们（Ur，W们）。

下面说NTN，NTN全称是neural tensor network，神经张量网，乍一看这个名字，啊有神经网有张量，没错，其实NTN就是SLM+张量分解。关门放NTN模型：



那个br可以不理它，意义就是一个偏置，弄参数总觉得wx有点单调，所以看到的大部分都是wx+b这样的，功能都是一样的，就是个偏置；这个公式和上一个SLM的差啥？聪明的你应该发现了差了括号里的第一项。这个东西是不是感觉在大明湖畔见过？往上划一划，这个是张量分解啊朋友们！k是这个张量有多少层，就是上面张量的那个图前后有多少面，然后一个实体是d\*1的量纲，Wr是d\*d的量纲，然后一个e1\*Wr\*e2就得到了一个数，现在有k层Wr，就得到了一个k\*1的向量；Vr长啥样，对这个量纲就行，我们要让括号里第二项算完也是k\*1的，Vr就得是k\*2d的，同理，br得是k\*1的，Ur也是k\*1（这是一个线性代数的常规操作，用你聪明的小脑袋算算就能算出来）



这是一个k=2的时候的可视化的图。你会发现SLM就是k=0的情况。

你可能发现一个问题，这两个模型的实体的向量表示是哪里来的？emmm，这个得告诉你，他们用了预先训练好的实体的向量表示，这一堆神经网相当于只是做了relation的表示，我觉得这算是作弊了，恩……

这个神经网只是举了两个很经典的例子，包括但不限于啊~如果看到长得不一样的不要强行对号入座啊！

距离模型，翻译模型：

这几种可以混合着用，不是非要看成，三种，彼此交叉很大的，哪里都能有张量分解，哪里也都能有神经网络，哪里也都能有transE。张量分解只是一种矩阵表示形式，神经网络也可以看成是一个做分类的工具……比如NTN就是SLM加上了张量分解。

NTN和SLM都是一层的神经网（单层的神经网应该叫感知机），你还可以加层数（没看到这样的论文，但是我觉得一定有），而且这里面实体的vector是训练好的了，我可以用transE整啊~总之排列组合，只有想不到，没有做不到

附上参考文献