Induction Networks for Few-Shot Text Classification

代码：Github

数据集：A Large-Scale Few-Shot Relation Extraction Dataset

摘要

当数据不足或需要适应看不见的类时，文本分类往往会遇到困难。在这种具有挑战性的情况下，最近的研究使用元学习来模拟小样本任务，在该任务中，将新问题与样本级别的小型支持集进行比较。但是，这种基于样本的比较可能会受到同一类中各种表达式的严重干扰。因此，我们应该能够学习支持集中每个类的一般表示，然后将其与新问题进行比较。在本文中，我们提出了一种新颖的归纳网络，通过在元学习中创新地利用动态路由算法来学习这种广义的类表示。

1.介绍

小样本学习致力于通过从极少数带有标签的示例中识别新颖的类来解决数据不足问题。仅一个或几个示例的局限性就挑战了深度学习中的标准微调方法。早期研究仅在有限的程度上应用了数据扩充和正则化技术来缓解因数据稀疏而导致的过拟合问题。相反，研究人员探索了元学习，以在人类学习的启发下将分布分配到类似任务上。当代的小样本学习方法通常遵循测试和训练条件必须匹配的原则，将训练过程分解为辅助元学习阶段，该阶段包括许多元任务。他们通过将元任务从一个小批量转换为另一个小批量来提取一些可转让的知识。这样，小样本模型可以仅通过一个带有标签的小型支持集对新类进行分类。

然而，现有的方法对小样本学习仍然面临许多重要问题，包括实行强有力的先验，任务间复杂的梯度转移，和微调目标的问题。Snell 等人提出的方法将非参数方法和度量学习相结合，为其中一些问题提供了潜在的解决方案。非参数方法可以使新示例快速被吸收，而不会遭受灾难性的过度拟合。这样的非参数模型仅需要学习样本的表示和度量标准。但是，相同类别的实例是相互关联的，并且具有统一的分数和特定的分数。在以前的研究中，通过简单地将支持集中样本的表示形式相加或取平均值来计算类级别的表示形式。这样做可能会在同一类别的各种形式的样本所带来的噪声中丢失基本信息。请注意，小样本学习算法无法在支持集上进行微调。当增加支持集的大小，由一个大数据量带来的改善也将会被更多的样本级别的噪音所抵消。

相反，我们通过在类级别上进行归纳来探索一种更好的方法：忽略无关的细节，并从同一类中具有各种语言形式的样本中封装一般语义信息。因此，需要一种全面的体系结构，其可以重构支持集的分层表示并且将样本表示动态地转换为类表示。

胶囊网络通过使用执行动态路由的“胶囊”来编码个体和整体之间的内在空间关系从而构成视点不变的知识。遵循类似的灵感，我们可以将样本视为个体，将类视为整体。我们提出了归纳网络，其目的是基于动态路由过程，对从小型支持集中的样本中学习广义类级别表示的能力进行建模。首先，编码器模块为问题和支持样本生成表示形式。接下来，归纳模块执行动态路由过程，其中矩阵变换可以看作是从样本空间到类空间的映射，然后类表示的生成全都取决于按协议进行的路由而不是任何参数，这为所提出的模型提供了强大的归纳能力，可以处理看不见的类。通过将样本的表示形式视为胶囊输入，将类的类别视为胶囊输出，我们希望识别出与样本级噪声无关的类的语义。最终，对问题和类之间的交互进行了建模-关系模块将比较它们的表示，以确定问题是否与类匹配。整体模型定义了基于 episode 的元训练策略，具有端到端的元训练能力，具有通用性和可伸缩性，可以识别看不见的类别。

具体贡献如下：

为归纳文本分类提出了了归纳网络。为了处理小样本学习任务中的样本多样性，该模型显式地建模了从小型支持集中推导类级别表示。

提出的归纳模块将动态路由算法与典型的元学习框架结合在一起。矩阵转换和路由过程使我们的模型能够很好地泛化以识别看不见的类。

2.问题定义

2.1 小样本分类

小样本分类是一项任务，在这个任务中，仅给出每个新类别的几个样例的条件下，必须对分类器进行调整以适应训练中未见的新类别。我们有一个带有标签的具有类别集合 Ctrain 的大型训练集。但是，经过训练后，我们的最终目标是在包含不相交的新的类别集合 Ctest

的测试集上生成分类器，在这个测试集中，只有一小部分带标签的支持集可用。如果支持集 C 个类别中的每个仅包含 K 个标记样例，则目标小样本问题称为 C-way K-shot 问题。通常，K 太小而无法训练监督分类模型。因此，我们的目标是在训练集上进行元学习，并提取可转让的知识，这将使我们能够在支持集上进行更好的小样本学习，从而对测试集进行更准确的分类。

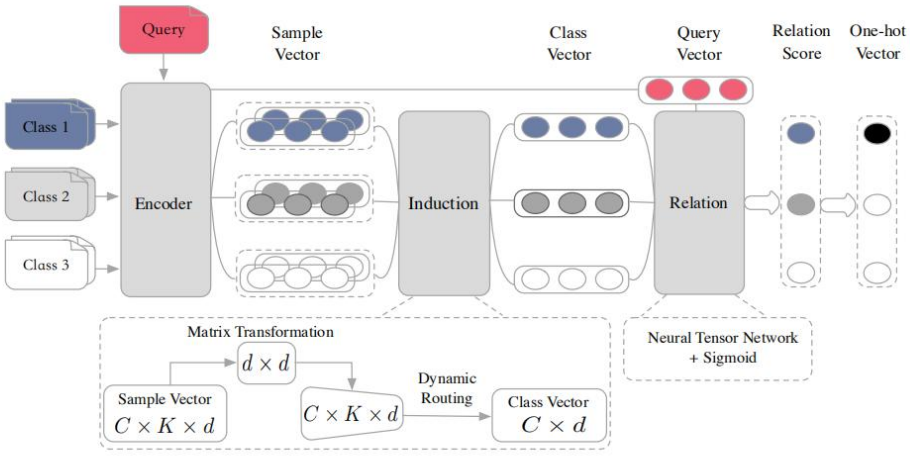
2.2 小样本问题的训练过程

Algorithm 1 Episode-Based Meta Training

1: **for** each *episode iteration* **do**
2: Randomly select C classes from the class space of the training set;
3: Randomly select K labeled samples from each of the C classes as support set $S = \{(x_s, y_s)\}_{s=1}^{m}$ $(m = K \times C)$, and select a fraction of the reminder of those C classes' samples as query set $Q = \{(x_q, y_q)\}_{q=1}^{n}$;
4: Feed the support set S to the model and update the parameters by minimizing the loss in the query set Q;
5: **end for**

首先通过从训练集中随机选择类别的子集，然后在每个选定的类别中选择示例的子集作为支持集 S 和其余示例的子集作为问题集 Q，形成元 episode。元训练过程明确学习到给定的支持集 S，以最大程度地减少问题集 Q 上的损失。我们将此策略称为基于 episode 的元训练，其详细信息如算法 1 所示。值得注意的是，要对模型进行训练有成千上万的潜在元任务，因此很难过拟合。

3.模型



igure 1: Induction Networks architecture for a C-way K-shot ($C = 3$, $K = 2$) problem with one query example

归纳网络如图 1 所示（3-way 2-shot 模型），它由三个模块组成：编码器模块，归纳模块和关系模块。

3.1 编码器模块

这个模块是一个双向循环神经网络，具有 self-attention 能力，给定输入文本，其由单词嵌入序列表示 $x = (w_1, w_2, ..., w_T)$。我们使用双向 LSTM 处理文本：

$$\overrightarrow{h_t} = \overrightarrow{LSTM}(w_t, h_{t-1}) \qquad (1)$$

$$\overleftarrow{h_t} = \overleftarrow{LSTM}(w_t, h_{t-1}) \qquad (2)$$

然后我们将 $\overrightarrow{h_t}$ 和 $\overleftarrow{h_t}$ 连接起来，获得隐藏状态 h_t 。令每个 LSTM 的隐藏状态大小为 u。为简单起见，将所有 T 个 h_t 记为 $H=(h_1,h_2,...,h_T)$ 我们的目标是将可变长度的文本编码为固定大小的嵌入。我们通过在 H 中选择 T LSTM 隐藏向量的线性组合来实现。计算线性组合需要 self-attention 机制，该机制将整个 LSTM 隐藏状态 H 作为输入，并输出权重为 a 的向量：

$$a=\text{softmax}\left(W_{a2}\tanh\left(W_{a1}H^T\right)\right) \qquad (3)$$

其中，$W_{a1}\in R^{d_a\times 2u}$ 和 $W_{a2}\in R^{d_a}$ 是权重矩阵，d_a 是超参数。文本的最终表示 e 是 H 的加权：

$$e=\sum_{t=1}^{T}a_t\cdot h_t \qquad (4)$$

3.2 归纳模块

本节介绍了提出的动态路由归纳算法。我们将通过等式 4 从支持集 S 获得的这些向量 e 视为样本向量 e^s ，将来自问题集 Q 的向量 e 作为问题向量 e^q 。最重要的步骤是提取支持集中每个类的表示形式。归纳模块的主要目的是设计从样本向量 e_{ij}^s 到类向量 c_i 的非线性映射：

$$\left\{e_{ij}^s \quad R^{2u}\right\}_{i=1,..C,j=1..K}\mapsto\left\{c_i \quad R^{2u}\right\}_{i=1}^{C}$$

在输出胶囊的数量为一个的情况下，我们在此模块中应用动态路由算法。为了在我们的模型中接受任何方式的任意输入，在支持集中的所有样本向量上采用了权重可转换的形式。支持集中的所有样本向量共享相同的变换权重 $W_s\in R^{2u\times 2u}$ 和偏差 b^s，因此该模型足够灵活，可以处理任何规模的支持集。每个样本预测向量 $\hat{e}_{ij}^s$ 的计算公式为：

$$\hat{e}_{ij}^s=\text{squash}\left(W_s e_{ij}^s+b_s\right) \qquad (5)$$

其中，squash 是整个矢量的非线性压缩函数，它使矢量的方向保持不变，但减小了其大小。给定输入向量 x，squash 定义为：

$$\text{squash}(x)=\frac{\|x\|^2}{1+\|x\|^2}\frac{x}{\|x\|} \qquad (6)$$

等式 5 编码了较低级别的样本特征与较高级别的类别特征之间的重要不变语义关系。为了确保类向量自动封装此类的样本特征向量，将反复地应用动态路由。在每次迭代中，该过程都会动态修改连接强度，并通过“路由 softmax”确保类 i 与该类中所有支持样本之间的耦合系数 d_i 的和为 1：

$$d_i=softmax(b_i) \qquad (7)$$

其中 b_i 是耦合系数的对数，在第一次迭代中初始化为 0，给定每个样本的预测向量 $\hat{e}_{ij}^s$，每个类别候选向量 $\hat{c}_i$ 是类别 i 中所有样本预测向量 $\hat{e}_{ij}^s$ 的加权和，然后应用非线性“*squashing*”函数以确保路由过程的矢量输出的长度不会超过 1

$$\hat{c}_i=\sum_j d_{ij}\cdot\hat{e}_{ij}^s \qquad (8)$$

$$c_i=squash(\hat{c}_i) \qquad (9)$$

每次迭代的最后一步是通过“协议路由”方法调整耦合系数 b_{ij} 的对数。如果产生的类别候选向量具有一个样本预测向量的大标量输出，则存在自上而下的反馈，该反馈会增加该样本的耦合系数，而降低其他样本的耦合系数。这种调整类型对于小样本学习的情况非常有效且稳定，因为它不需要恢复任何参数。每个 b_{ij} 通过以下方式更新：

$$b_{ij}=b_{ij}+\hat{e}_{ij}^s\cdot c_i$$

上述归纳方法成为动态路由归纳，在算法 2 中进行总结

Algorithm 2 Dynamic Routing Induction

Require: sample vector e_{ij}^{s} in support set S and initialize the logits of coupling coefficients $b_{ij}=0$

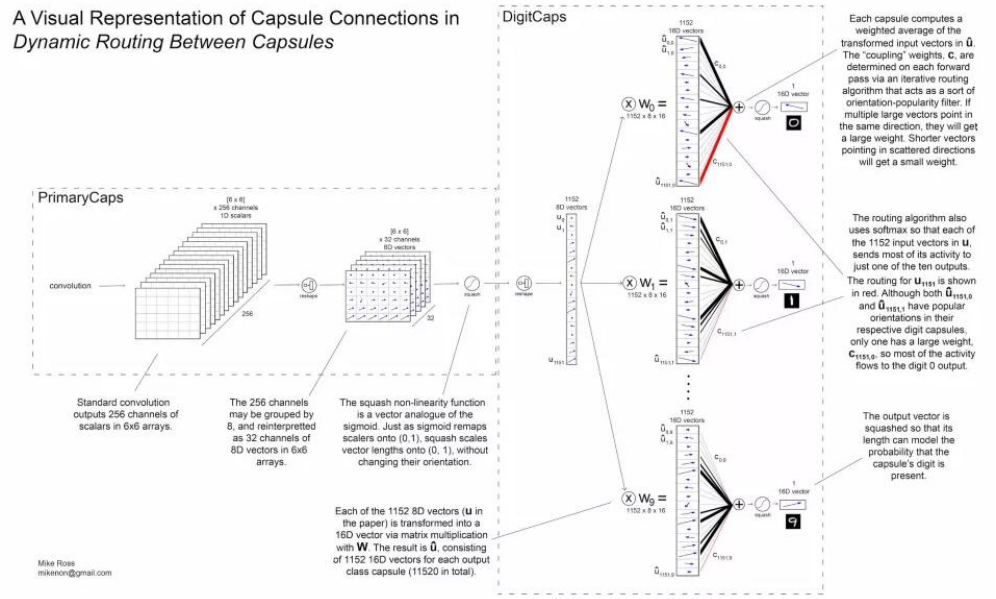
Ensure: class vector c_i

for all samples $j=1,...,K$ in class i:
  $\hat{e}_{ij}^{s}=squash(W_s e_{ij}^{s}+b_s)$
for $iter$ iterations **do**
    $d_i=softmax(b_i)$
    $\hat{c}_i=\sum_j d_{ij}\cdot\hat{e}_{ij}^{s}$
    $c_i=squash(\hat{c}_i)$
    for all samples $j=1,...,K$ in class i:
      $b_{ij}=b_{ij}+\hat{e}_{ij}^{s}\cdot c_i$
end for
Return c_i

以下是胶囊网络的网络模型，本文主要借鉴了 Digitcaps 部分

论文链接：https://arxiv.org/abs/1710.09829

胶囊网络解读：https://blog.csdn.net/shine19930820/article/details/88959982



3.3 关系模块

在归纳模块生成类向量 c_i 并将问题集中的每个问题文本由编码器模块编码为问题向量 e^q 之后，接下来的基本过程是测量每个[问题-类别]对之间的相关性。关系模块的输出称为关系得分，代表 c_i 和 e^q 之间的相关性，是 0 到 1 之间的标量。在此模块中使用神经张量层，该模块在建模两个向量之间的关系方面显示了巨大的优势，在本文中，我们选择它作为交互函数。张量层输出如下的关系向量：

$$v\left(c_i,e^q\right)=f\left(c_i^T M^{[1:h]}e^q\right) \qquad (11)$$

其中 $M^k\in R^{2u\times 2u},k\in[1,...,h]$ 是张量参数的一个切片， f 是 RELU 函数。

第 i 个类和第 q 个问题之间的最终关系得分 r_{iq} 由 sigmoid 函数激活的完全连接层计算得出。

$$r_{iq}=\text{sigmoid}\left(W_r v\left(c_i,e^q\right)+b_r\right) \qquad (12)$$

3.4 目标函数

我们使用均方误差（MSE）损失来训练我们的模型，将关系得分 r_{iq} 回归为真实得分 y_q：匹配对具有相似性 1，错配对具有相似性 0。在一个 episode 中，给定具有 C 个类别的支持集 S 和查询集 $Q=(x_q,y_q)_{q=1}^{n}$，损失函数定义为：

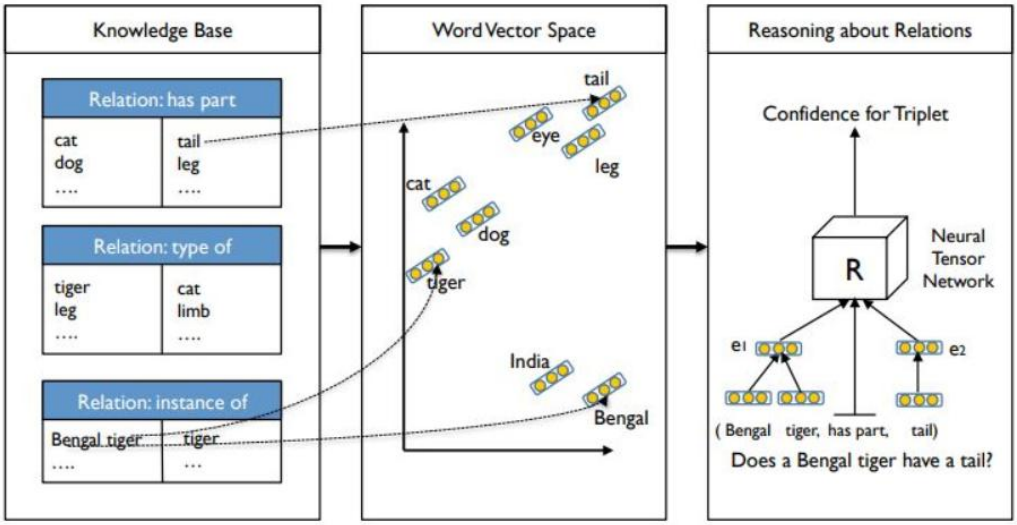
$$L(S,Q)=\sum_{i=1}^{C}\sum_{q=1}^{n}\left(r_{iq}-1\left(y_q==i\right)\right)^2 \qquad (13)$$

预测关系分数，可以将其视为回归问题，并且真实得分在 $\{0,1\}$ 空间内。

这三个模块的所有参数都通过反向传播共同训练。每个训练阶段的所有参数均使用 Adagrad。由于其泛化性质，我们的模型不需要对从未见过的类进行任何微调。归纳和比较能力与训练 episodes 一起累积在模型中。

关于神经张量网络：

在知识库完成中，任务是确定两个实体对之间的关系。例如，考虑两个实体对 -<cat, tail>和<supervised learning, machine learning>。如果我们被要求确定给定的两对之间的关系 -<cat，R，tail>和<supervised learning, R, machine learning> - 那么第一个关系可以最好的归结为有型，而第二个关系可以被归结为实例。所以，我们可以将这两个对重新定义为 <cat，has，tail>和<supervised learning，instance of，machine learning>。神经张量网络（NTN）在实体 - 关系对的数据库上训练，用于探究实体之间的附加关系。这是通过将数据库中的每个实体（即每个对象或个体）表示为一个向量来实现的。这些载体可以捕获有关该实体的事实，以及它是如何可能是某种关系的一部分。每个关系都是通过一个新的神经张量网络的参数来定义的，这个神经张量网络可以明确地涉及两个实体向量：

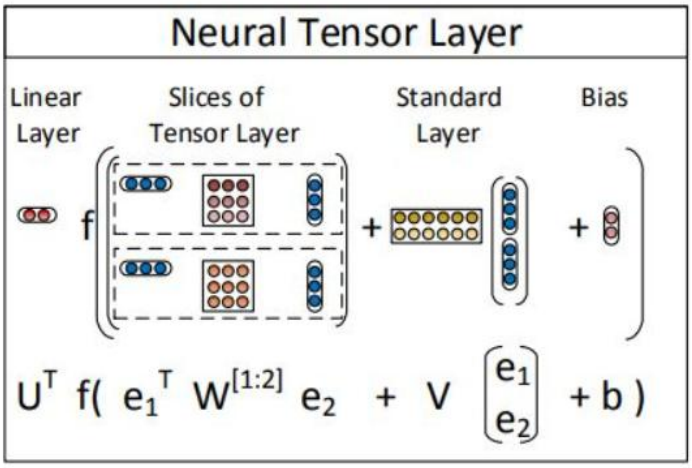


关系推理的神经模型

能够认识到某些事实纯粹是由于其他现有的关系而存在的，是学习常识推理的模型的目标。NTN 旨在发现实体<e1,e2>之间的关系，即对于<e1，e2>确定性地预测关系 R. 例如，(e1,R,e2) = (Bengal tiger, has part, tail) 这个关系是否真实且具有确定性。神经张量网络（NTN）用一个双线性张量层代替一个标准的线性神经网络层，它直接关联了多个维度上的两个实体向量。该模型通过下列基于 NTN 的函数计算两个实体处于特定关系的可能性分数：

$$g\left(e_1,R,e_2\right)=u_R^T f\left|e_1^T W_R^{[1:k]} e_2+V_R\begin{matrix}e_1\\e_2\end{matrix}+b_R\right|$$

可视化神经张量网络



4.实验分析

转换的影响 首先是可以看到 Induction Network 的 transformation 效果，可以发现 transformation 之前的 5 个类别中的样本分布相对比较紊乱，而 transformation 之后 support set 中的 5 个类别样本表征边界相对清晰。 我们可以清楚地看到矩阵变换后的向量更加可分，证明了矩阵变换对低级样本特征和更高级别特征之间的语义关系进行编码的有效性。

图2：在5-way 10-shot场景下变换的影响 (a)在矩阵变换之前的支持样本向量 (b)在矩阵变换之后的支持样本向量

查询文本向量可视化 我们发现我们的归纳模块不仅可以通过生成有效的类级别特征来完美地工作，而且还有助于编码器学习更好的文本向量，因为它在反向传播期间为实例和特征赋予不同的权重。 图 3 显示了在 5-way 10-shot 场景下，关系网络和我们的归纳网络学习的文本向量的 t-SNE [Maaten 和 Hinton，2008]可视化。 具体来说，从 ODIC 测试集中选择 5 个类，然后使用 t-SNE 将嵌入的文本投影到二维空间中。 很明显，归纳网络学习的文本向量在语义上比关系网络更好，既 Query Set 中的 query 经过 Induction Network 之后明显边界会更清晰

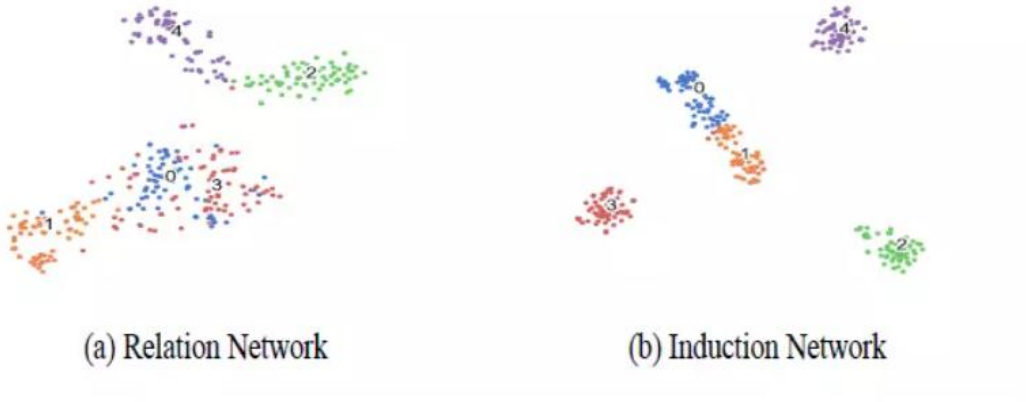


图3：查询文本向量可视化被学习：(a)关系网络 (b)归纳网络

5.结论

在本文中，介绍了归纳网络，一种针对小样本文本分类的新神经模型。 所提出的模型重建支持训练样本的分层表示，并动态地将样本表示引入类表示。 我们将动态路由算法与典型的元学习框架相结合，来模拟人类归纳能力