特征映射的深度核函数

邓国伟　　孙一鸣　　吴金柱

***Abstract*—In some complex classification tasks, due to different classification requirements of different scenarios, the classifiers trained only on training dataset cannot be directly applied to reality. What we care about is not the result of classification, but the process of feature extraction. The process of feature extraction can be seen as a process of kernel mapping, which maps origin samples into high-dimensional spaces that are easily classified. In image recognition, due to the high dimensionality of image data, because complex data requires more complex kernel functions to map, the classified results obtained by directly using traditional kernel functions may be poor. Deep learning, with its complex function form and numerous parameters, can be infinitely approximated to a wide variety of functions. Therefore, we can classify different images by training a kernel function in the form of a deep neural network.**

keywords: deep learning　deep kernel　classifier

摘要–在一些复杂的分类任务中，由于不同的场景的分类需求不同，仅仅通过数据集训练所得到的分类器无法直接应用于实际。我们往往关心的不是分类的结果，而是特征提取的过程。这个特征提取的过程，可以看作是一个核映射的过程，即将难以区分的样本映射到容易区分的高维空间中。在图像识别中，由于图像数据维度较高，直接使用传统的核函数得到的分类结果较差，因为复杂的数据需要更复杂多变的核函数来映射。而深度学习以其复杂的函数形式和众多的参数，可以无限逼近于各种各样的函数。因此，我们可以通过训练一个深度神经网络形式的核函数，来对不同的图像进行分类。

关键词：深度学习　深度核　分类器

I. 引言

目前，在深度学习领域，虽然其不可解释性一直被诟病，然而，有一点被大众广泛认可，那就是深度学习中的每一层隐含层，都是对数据特征的提取。较浅的隐含层学习到的是低级特征，而较深的隐含层学习到的是高级特征。最后，将提取到的特征输入到分类层中（如softmax层），即可得到分类结果。然而，在一些复杂的分类任务中，如人脸识别情景，假如我们有10万个不同的人的人脸数据库，每个人有若干张照片，那么我们就可以训练一个10万分类模型，对于给定的照片，我们可以判断它是10万个中的哪一个。但这仅仅是训练场景，无法直接应用。到了具体的应用环境，比如一个公司内部，可能有只有几百人；在公共安全检测场景，可能有数百万人，所以前面做好的10万分类模型基本上是没有意义的，但是在这个模型softmax层之前的特征，可能还是很有意义的。如果对于同一类数据，其提取特征的方式相似，那么实际应用中，我们就可以把训练好的模型当作特征提取工具，然后把提取出来的特征输入到其他的分类算法中，如SVM与KNN中，即训练出的模型，可以应用于不同的分类任务。许多分类器的分类原理是利用核函数将样本映射到其他维度的空间形成可分的映射样本，然后利用分类器分类。以支持向量机为例，其核函数可以将线性不可分的样本，映射到线性可分的高维空间中，完成分类任务。深度神经网络以其复杂的函数形式和众多的参数，可以无限逼近于各种各样的函数，因此我们拟用深度神经网络构造一个核函数，使其对于同类样本都具有很好的映射效果，从而使其他的分类算法可以很好的完成分类任务。

Identify applicable funding agency here. If none, delete this.

II. 整体思路

对于不同的数据集，需要使用不同的深度网络类型来学习。比如，对于维度较小的数据集，几层普通的全连接层就可以达到较好的效果；而对于图像数据，使用使用卷积神经网络可以有效地提高精度，同时减少参数数量；同样，对于文本语言数据，为了考虑文字间的时序关系，LSTM和GRU层能够达到较好的效果。但是，万变不离其宗，这些不同的神经网络，都可以看作为一个核函数$\phi(x)$,其输入为数，输出的是特征。该课题的重点在于，如何训练这个核函数，使得该核函数在某类数据集中，有普适的效果。对于一包括n类样本的数据集$D = \{x_i\}_{i=1}^{n}$，其中，训练集记为D_1，测试集记为D_2，对于核函数$\phi(x)$，如果能达到以下效果，就认为该核函数具有普适性。

(1) 对于一数据集$D = \{x_i\}_{i=1}^{n}$，其中，训练集记为D_1，测试集记为D_2，使用训练集D_1训练核函数$\phi(x)$。使用训练好的核函数，运用于分类器中。如果仅使用D_1中一小部分训练分类器，能够使分类器在测试集D_2中有较好的分类效果，则核函数具有普适性。

(2) 对于一含有n类标签的数据集$D = \{x_i\}_{i=1}^{n}$，使用其中m类作为核函数训练集D_1，另外$n-m$类数据中一部分作为分类器的训练集D_2，另外一部分作为分类器的测试集D_3。使用训练集D_1，训练核函数$\phi(x)$。同样，使用训练好的核函数，运用于分类器中。如果使用D_2训练分类器，能在D_3有较好的分类效果则核函数具有普适性。

III. 理论基础

训练深度核函数，并运用于不同的分类器，其本质是基于特征的迁移学习。迁移学习中，定义了源域D_s，一个对应的源任务T_s，还有目标域D_t，以及目标任务T_t。基于特征的迁移学习是，通过源域 D_s学习一个普适性强的的的特征表示，把知识通过特征的形式进行编码，并从源域D_s传递到目标域D_t，提升目标域 Dt任务效果。在我们这个课题中，源域即为核函数的训练集，目标域即为分类器的训练集与测试集。通过对核函数进行训练，获得一个普适性强的特征提取器，从而迁移到分类器上，使得分类器训练速度加快，分类效果变好。

IV. 深度核函数的训练

A. 支持向量机(*Support Vector Machine*)

支持向量机(SVM)在高维或无限维空间中构造超平面或超平面集，其可用于分类，回归或其他任务。直观来说，通过寻找任何类的最近训练数据点具有最大距离的超平面（所谓的功能边界或支持向量）实现良好的分离，通常边距越大，分类器的泛化误差越低。

$$\min_{w,b,\zeta} \frac{1}{2} w^T w + C \sum_{i=1}^{n} \zeta_i,$$

$$s.t. \begin{cases} y_i(w^T \phi(x_i) + b) \geq 1 - \zeta_i, \\ \zeta_i \geq 0, i = 1, ..., n \end{cases}$$

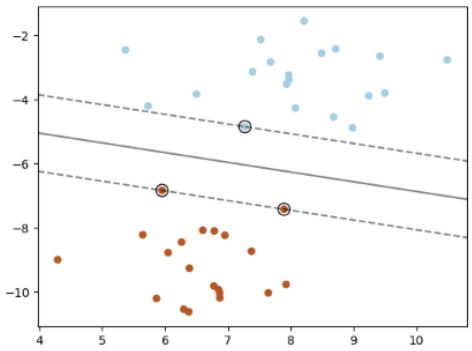

Fig. 1. SVM

通常，支持向量机的训练过程是基于核函数已知时，使用以上数学表达式，运用KKT条件，对参数w和b进行优化，寻找到最优的分类超平面。而在核函数$\phi(x)$未知时，整个网络的需要训练的参数，应该包括核函数中的可训练参数和SVM中的参数w和b。训练过程使用反向传播的原则，对所有参数进行训练。整个网络的损失函数，使用hinge loss函数：

$$loss = max(0, 1 - y_{pred} * y_{true})$$

使用svm的优化方法，能够很好的解决二分类的问题。若要使用svm解决多分类问题，有两种思路：one vs. one和 one vs. rest，对于n分类问题，前者构造$(n(n-1))/2$个分类超平面，后者构造n个分类超平面，其原理与二分类相似，只是稍显繁琐。

B. *softmax+center loss*

为了更好的解决多分类问题，深度学习中一般使用softmax分类器，其表达式为：

$$y_i = \frac{e^{a_i}}{\sum_{k=1}^{C} e^{a_k}} \quad i \in 1, ..., C$$

Softmax能够很好地解决多分类的问题，并且其输出结果可以看作是样本属于某一类的概率。起初，我们直接在核函数后面加上softmax层，对核函数进行训练。虽然整个网络的分类效果较好，但是，训练出来的核函数应用于其他分类器，效果却并不理想。

通过对softmax函数进行深入考究之后，发现直接训练softmax的话，得到的特征并不具有聚类的特性，相反，它们会尽量布满整个空间。例如，在二维空间中，对于不同的向量$(1, 2)$和$(2, 3)$或者任意一个满足$(x, x+1)$的向量，输入到softmax函数中，其结果相等。也就是说，在直线$y = x+1$上的每一个点，输入到softmax层之后，可能得到同样的结果。这将会影响核函数表达特征的能力。为了减轻这一影响，在训练核函数时，引入一个聚类惩罚项。因此，整个网络中包含两个损失函数，softmax对应的交叉熵函数和聚类惩罚项。前者使不同类的间距尽可能大，后者使同类样本间距尽可能小。

交叉熵

$$L_s = -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}}$$

中心误差

$$L_C = \frac{1}{2} \sum_{i=1}^{m} ||x_i - c_{y_i}||_2^2$$

其中c_{y_i}表示第y_i个类别的特征中心，x_i表示全连接层之前的特征。后面会讲到实际使用的时候，m表示mini-batch的大小。因此这个公式就是希望一个batch中的每个样本的feature离feature的中心的距离的平方和要越小越好，也就是类内距离要越小越好。

整个网络的损失函数，为交叉熵和中心误差的加权和：

$$L = L_S + \lambda L_C$$

$$= -\sum_{i=1}^{m} log \frac{e^{W_{y_i}^T x_i + b_{y_i}}}{\sum_{j=1}^{n} e^{W_j^T x_i + b_j}} + \frac{\lambda}{2} \sum_{i=1}^{m} ||x_i - c_{y_i}||_2^2$$

对于不同的λ，其效果如下：

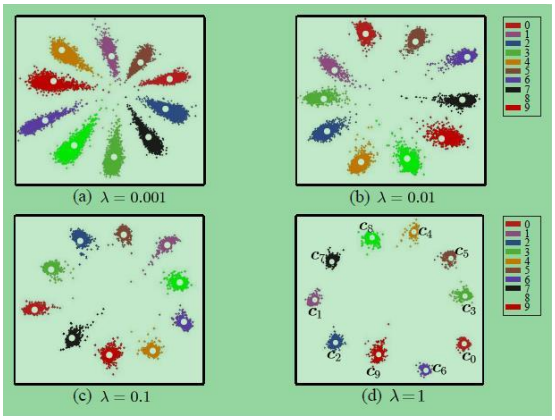

Fig. 2. 效果图

为了实现中心误差，网络必须要能够存储各类特征的中心。为此，我们在网络中添加了一层embedding层。Embedding层通常运用在自然语言处理）NLP中，但其作用并不限于word embedding。Embedding层可以有两种理解：

(1) 是one hot输入的全连接层的加速版本，也就是说，它就是一个以one hot为输入的Dense层，数学上完全等价
(2) 它就是一个矩阵查找操作，输入一个整数，输出对应下标的向量，只不过这个矩阵是可训练的

因此，整个网络的结构如下图:

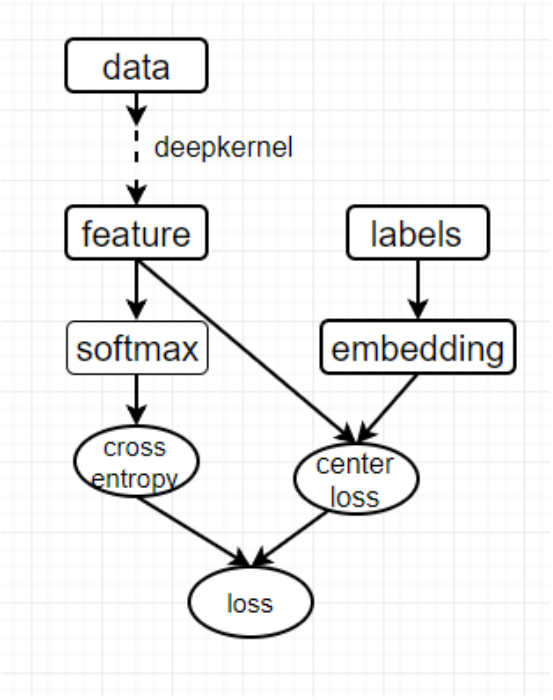

Fig. 3. 结构图

C. LDA

线性判别分析作为经典的二分类算法，在二分类上具有独特的优势。线性性别分析基于“同类尽可能近，异类尽可能远”的思路构造如下损失函数。

$$J = \frac{||\mu_0 - \mu_1||^2}{\sigma_0 + \sigma_1}$$

其中，

$$\mu_0 = \frac{1}{n_0} \sum_{y_i=0} \phi(x_i)$$

$$\mu_1 = \frac{1}{n_1} \sum_{y_i=1} \phi(x_i)$$

$$\sigma_0 = \frac{1}{n_0 - 1} \sum_{y_i=0} \phi(x_i)\phi^T(x_i)$$

$$\sigma_1 = \frac{1}{n_1 - 1} \sum_{y_i=1} \phi(x_i)\phi^T(x_i)$$

我们使用线性判别分析的原理构造多分类问题的损失函数。训练样本$D = \{x_i\}_{i=1}^n$来自c类，同类尽可能近，使同类样本映射后的值方差尽可能小，异类尽可能远，使任意两类样本中心的欧式距离最大，定义新的损失函数如下。

$$J = \frac{\sum_{i=1 j=1 i \neq j}^{c} ||\mu_i - \mu_j||^2}{\sum_{i=1}^{c} \sigma_i}$$

其中，

$$\mu_k = \frac{1}{n_k} \sum_{y_i=k} \phi(x_i)$$

$$\sum_k = \frac{1}{n_k - 1} \sum_{y_i=k} \phi(x_i)\phi^T(x_i)$$

基于神经网络最小化损失函数的原则，我们取1/J作为深度网络的损失函数进行反向传导。设置合适的epoch训练

深度网络，使得深度网络全连接层输出的映射样本在该维度上可分。即实现lda的深度核函数的训练，后续将该核函数的映射结果使用SVM进行分类。

D. 小结

上述三种深度核函数网络训练方法的训练过程如图4。

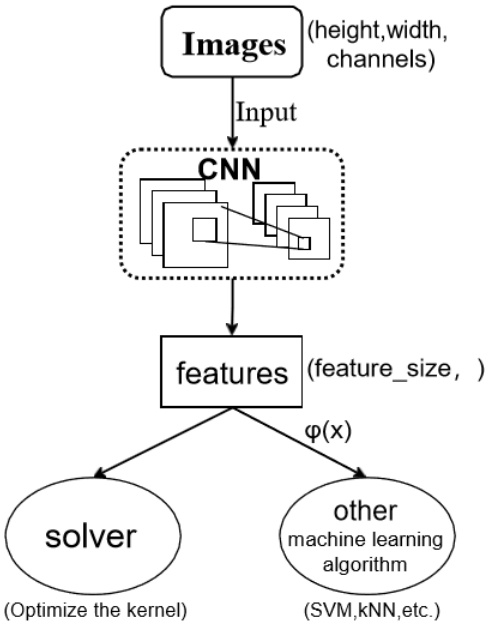

Fig. 4. 训练过程

V. 测试结果

为了验证深度核函数的普适性与优越性，我们分别在四个数据集上进行了测试：Mnist，Mnist-Fashion，cifar-10，fer2013。其中Mnist为手写数字数据集，尺寸为（70000，28，28），Mnist-Fashion为10类物品的数据集（70000，28，28），cifar-10包含了10类动物与交通工具的彩色图像（60000，32，32，3），fer2013是在kaggle下载的人脸表情识别的数据集（42000，48，48）。用SVM使用不同的核函数对四个数据集进行测试，结果如表I。

TABLE I
不同核函数SVM分类结果

kernel	rbf	deepkernel1	deepkernel2
mnist	93.89%	99.09%	98.38%
mnist-fashion	85.57%	91.33%	93.16%
fer2013	14.26%	79.91%	43.86%
CIFAR-10	36.1%	70.7%	74.13%

* deepkernel1为使用数据集的前80%训练所得的核函数，同时，使用10%作为svm的训练集，剩余10%作为测试集

** deepkernel2为使用数据集中的前8类作为训练集，后两类的50%作为svm的训练集，剩余50%作为训练集

我们测试了的三种分类器：SVM（kernel:‘rbf’）指使用内置核函数“rbf”的支持向量机；SVM(deepkernel)[1]表示使用我们的深度核函数，同时指明是第一种训练方式，即使用所有类的样本的80％作为训练样本，训练得到深度核函数，再将剩下20％的样本送入支持向量机（这些样本同样将被划分为训练集和测试集），检验其精度；与之相对应，SVM(deepkernel)[2]指第二种训练方式，即使用一部分类的所有样本训练得到深度核函数，再将剩下类的所有样本送入支持向量机，划分为训练集和测试集，检验精度。由于我们选取的样本都是十分类数据，我们测试时，统一选取前八类来训练深度核函数，后两类用来检验精度。

实验结果表明深度核函数相比SVM传统的核函数具有更好的效果，并且采用deep kernel的SVM分类器只训练了SVM的参数，网络中的超参数还没有花太多时间去调整，因此其准确率还有提升的空间。

在训练时间上，对于较简单的数据集，如Mnist和Mnist-Fashion，使用rbf为核函数的svm能够很快的收敛，也有较好的精度。但是，面对维度更高，更复杂的数据集fer2013和cifar-10数据集，训练时间过长，特别对于fer2013，其训练时间超过1小时，并且很难收敛。而深度网络在训练时间上也有很大优势，如表II

TABLE II
不同核函数SVM训练时间

kernel	rbf	deepkernel	deepkerne2
mnist	$\leq$ 2min	$\leq$ 30s	$\leq$ 30s
mnist-fashion	$\leq$ 2min	$\leq$ 30s	$\leq$ 30s
fer2013	$\geq$ 1h	$\leq$ 1min	$\leq$ 1min
cifar-10	$\leq$ 10min	$\leq$ 1min	$\leq$ 1min

VI. 总结

利用深度神经网络作为分类器的核函数，将不可分的样本映射到可分的空间，再用分类器对样本进行分类。在映射的过程中，深度核函数相较于传统的核函数具有极大的优势，一方面是深度核函数的普适性，即对各种样本都可以得到可分的映射空间；另一方面是深度核函数的优越性，采用深度核函数的分类器在分类时，在精度和训练时间上都优于一般的核函数。因此，深度核函数具有其独到的优势。除此之外，深度核函数在映射的过程中可以学到样本的特征，因此也可视作一种特征降维的方法。

REFERENCES

[1] 李玉,张婷,胡海鹤, ”基于多层感知器的深度核映射支持向量机,” 北京工业大学学报, 2016年11月.

[2] Yangon Wen, Kaipeng Zhang, Zhifeng Li, You Ciao, ”A Discriminative Feature Learning Approach for Deep Face Recognition, ‘ ’ ECCV2016, pp. 503-505.

[3] 苏剑林 ,《keras中自定义复杂的loss函数》, 2017年7月.

[4] 周志华, 《机器学习》, 清华大学出版社, 2016.

[5] ”https://scikit-learn.org/stable/”.