结合线性支持向量机的深度学习

2017110220 曹亚帅*
yashcao@bupt.edu.cn

14th December 2017

1 Introduction

对于分类任务，大多数的深度学习模型中都采用 Softmax 激活函数进行预测，训练深度网络的方法往往是最小化交叉熵损失。支持向量机 (Support Vector Machines, SVM) 的分类功能可以用于替代 Softmax 分类。过去已有将 SVM 与深度卷积网络结合使用的方法提出：训练深度卷积网络来学习良好的隐层表示，然后将数据样本对应的隐层表示变量作为 SVM 的输入进行分类。虽然这种技术通常可以提高性能，但缺点是对于 SVM 来说较低级别的特征并未得到有效的精调 (fine-tuned)。

论文《Deep Learning using Linear Support Vector Machines》解决的问题是如何使梯度可以反向传播，以学习更低层次的特征。

主要贡献:

1. 论文表明了在一些深度的结构顶层中使用线性的 SVM 而非 Softmax，可以理解为对于大数据集，使用 SGD 求解 SVM 达到训练快速且可以在线学习的目的。

2. 论文提出的模型优化的目标是 L2-SVM 的 loss，而不是标准的 hinge loss。与标准支持向量机的 hinge loss 不同，L2-SVM 的 loss 是可微的，并且起到了重要的惩罚项作用。用 L2-SVM 的实验结果表明，通过简单地用线性支持向量机替代 Softmax 在数据集 MNIST 上带来了显著的提升。

本文的整体结构如下: 第1节说明了本文要解决的问题。在第2节中简述 Deep Learning 和 SVM 的基础理论，联系了本学期课程。第3节分析了提出方法与传统方法之间的区别。第4节尝试了算法的实现与思路的拓展。

*参考原文：《An Architecture Combining Convolutional Neural Network (CNN) and Support Vector Machine (SVM) for Image Classification》 https://arxiv.org/abs/1712.03541v1

2 Theorem

2.1 Deep Neural Networks

深度神经网络是由许多相连的神经元组成的，一般包含输入层，隐层，输出层。假设输入样本为 $\boldsymbol{x}$，中间经过权重 $\boldsymbol{w}$ 和偏置 b 的计算得到加权输出 $\boldsymbol{w}^T\boldsymbol{x}+b=z$，每一层对 z 经过激活函数 $f(\cdot)$ 的处理得到预测 $y=f(z)$。

数据集样本 $\boldsymbol{x}$ 对应的标签为 $\boldsymbol{t}$，通过 t 和 y 的误差得到 loss 函数，loss 函数经过梯度下降等方法优化 $\boldsymbol{w}$、b 最终使深度神经网络可以更加接近准确的预测。

2.2 Softmax

Softmax 分类器是常用的分类器之一。Softmax 函数的本质就是将一个 n 维的任意实数向量压缩（映射）成另一个 n 维的实数向量，其中映射后的向量中每个元素取值都介于 $(0,1)$ 之间。

Softmax 函数形式如下：

$$\sigma(z)_j=\frac{e^{z_j}}{\sum_{i=1}^{n}e^{z_i}}\qquad j=1,2,...,n \tag{1}$$

Softmax 函数经常用在神经网络的最后一层，称为 Softmax 层，上式中的 $z=\boldsymbol{w}^T\boldsymbol{x}+b$ 是 Softmax 层的输入。Softmax 层作为输出层进行多分类任务，则输出的结果为：

$$\begin{aligned}(p_1,p_2,\ldots,p_n)=&\text{softmax}(z_1,z_2,\ldots,z_n)\\=&\left(\frac{e^{z_1}}{Z},\frac{e^{z_2}}{Z},\ldots,\frac{e^{z_n}}{Z}\right)\end{aligned} \tag{2}$$

其中，$Z=e^{z_1}+e^{z_2}+\cdots+e^{z_n}$。如果预测分类结果在 n 个类别中取值，这个多分类问题的预测结果为：

$$y^*=arg\ \underset{i}{max}\ p_i \tag{3}$$

2.3 Support Vector Machines

SVM 是非常经典的二分类模型，其核心在于找到具有最大间隔的超平面 $\boldsymbol{w}^T\boldsymbol{x}+b=0$。由于只需要数个向量就可以描述一个边界，故这些向量被称为支持向量（support vector）。最大化分类间隔可以用以下公式描述：

$$\begin{cases}\underset{\boldsymbol{w}}{min} & \dfrac{1}{2}\|\boldsymbol{w}\|^2\\ s.t. & 1-y_i(\boldsymbol{w}^T\boldsymbol{x_i}+b)\leqslant 1\ \ i\in[1,m]\end{cases} \tag{4}$$

下面是线性 SVM 的一般形式（考虑松弛变量 ξ_i），其中目标分类 $y\in\{-1,1\}$，C 为给定的惩罚系数：

$$\begin{cases}\min\limits_{\omega,\xi}\left[\dfrac{1}{2}\|\omega\|_2^2+C\sum\limits_{i=1}^{n}\xi_i\right]\\ s.t.\ (\omega^Tx_i+\gamma)y_i\geq 1-\xi_i,\ \forall i=1,\ldots,n\\ \qquad \xi_i\geq 0,\ \forall i=1,\ldots,n\end{cases} \tag{5}$$

SVM 可以利用核函数（kernel function）将低维的线性不可分问题转变为高维的线性可分问题，它仅仅是做隐式映射，并不显式地计算结果，这就基本避免了维度提升带来的计算复杂度。本文考虑到非线性扩展的复杂度，所以在标准深度学习模型中只使用线性 SVM。

3 Model Analysis

3.1 L2-SVM

SVM 根据 loss 函数的不同，可以分为 L1-SVM 和 L2-SVM。

记 $m \triangleq f_\theta(x)y$（其中 $y \in \{-1, 1\}$，$f_\theta(x) = \boldsymbol{w}^T x + b$）。对于 2 分类问题，最理想的损失函数是只包含 0/1 的损失：当 $f_\theta(x)$ 与 y 同号时，损失为 0；而当 $f_\theta(x)$ 与 y 异号时，损失为 1。但这种损失函数既不是严格可导的，也不是凸函数，所以直接优化很难。Hinge loss 是 0/1 损失的一种近似：

$$J_{\text{hinge}}(m) = \max\{0, 1 - m\} \tag{6}$$

函数图像如下所示：

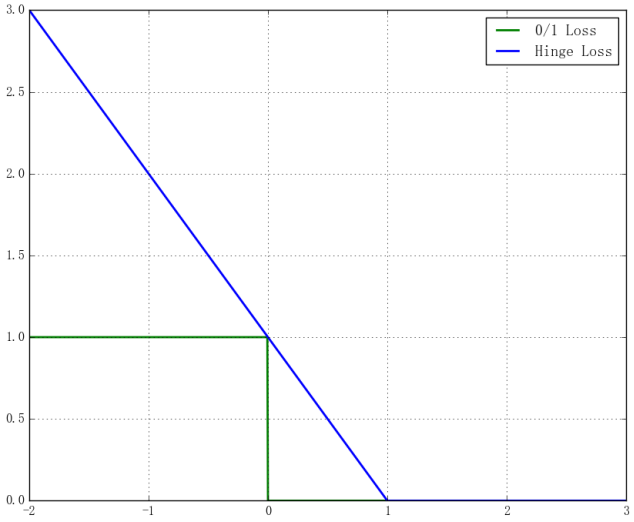

Figure 1: Hinge Loss

应用拉格朗日乘数法，L1-SVM 的优化目标原始形式为：

$$\min_{\omega} \left[\frac{1}{2} \|\omega\|_2^2 + C \sum_{i=1}^{n} \max\{0, 1 - m\} \right] \tag{7}$$

可知 L1-SVM 的 loss function 是标准的 hinge-loss function。但是这种损失函数不是处处可导，因此采用松弛变量平方和的损失函数，即 squared hinge loss SVM(也叫 L2-SVM)，

目标函数如下：

$$\min_{\omega}\left[\frac{1}{2}\|\omega\|_2^2 + C\sum_{i=1}^{n}\max\{0,1-m\}^2\right] \tag{8}$$

L2-SVM 是可微的，并且对于错误的点施加更大的惩罚。

3.2 多分类的 SVM

SVM 最早被研究用于二分类，然而实际生活中总是充斥着多分类问题。二分类 SVM 的组合可以实现一对多的分类。扩展 SVM 简单的方法是使用所谓的“one-vs-rest”方法。对于 n 类问题，n 个线性 SVM 将被独立训练，其中来自其余类的数据形成负例。

训练时依次把某个类别的样本归为一类, 其他剩余的 $n-1$ 个样本归为另一类，这样 n 个类别的样本就构造出了 n 个不同的 SVM 分类器。分类时将未知样本 x 应用到 n 个训练模型 $y_k(x)$ 中, 得到 n 个分类得分，测试结果为具有最大分类值的那类。预测公式如下：

$$y(x) = \max_{k} y_k(x) \tag{9}$$

3.3 使用 SVM 的深度学习

在论文中使用 L2-SVM 的目标来训练深度神经网络进行分类。通过反向传播顶层线性 SVM 中的梯度来学习较低层权重，首先我们需要做 SVM 的目标函数 $\boldsymbol{l(w)}$ 对倒数第二层的激活值 $\boldsymbol{h}$ 的微分。对于 L1-SVM（公式 (7)）来说，

$$\frac{\partial \boldsymbol{l(w)}}{\partial \boldsymbol{h}_n} = -Cy_n\boldsymbol{w}, \qquad if(1 > (\boldsymbol{w}^T\boldsymbol{h}_n + b)y_n) \tag{10}$$

同样的，对于 L2-SVM（公式 (8)）可得：

$$\frac{\partial \boldsymbol{l(w)}}{\partial \boldsymbol{h}_n} = -2Cy_n\boldsymbol{w}(\max(1-(\boldsymbol{w}^T\boldsymbol{h}_n + b)y_n, 0)) \tag{11}$$

根据作者的发现，大多数情况下 L2-SVM 略好于 L1-SVM。然后最终是要训练整个网络的，所以可以执行如下公式：

$$\frac{\partial \boldsymbol{l(w)}}{\partial \boldsymbol{w}_{cnn}} = \frac{\partial \boldsymbol{l(w)}}{\partial \boldsymbol{h}_n} \cdot \frac{\partial \boldsymbol{h}_n}{\partial \boldsymbol{w}_{cnn}} \tag{12}$$

4 Algorithm Implementation

4.1 SGD 训练 SVM 算法

以前的工作中，深度网络学习出来的特征向量被输入到 SVM 中，进行 SVM 对特征向量的训练。本文不同之处是通过顶层 SVM 反向传播梯度来训练所有层次的深层网络，学习所有层次的特征。其余的地方和普通的神经网络没有区别。

处理方法：

1. 使用 CNN 实现多分类时，最后一层的神经元个数往往是类别数目。使用 Softmax 分类时，最后一层的激活函数设置为 Softmax 即可。现在不使用 Softmax 激活函数，将激活函数设置为 linear，即 $f(x) = x$。

2. 在最后一层后面加上一个 linear one-vs-all SVM。

3. 自定义损失函数，将 hinge loss 改为 squared hinge loss。再使用标准的 SGD 方法对自定义的损失函数进行训练优化。

4.2 思考拓展

1. 分类问题中，深度神经网络输出层除 SVM 外，还可以换成什么分类器？

2. 一个二分类 SVM 可以视为一个简单的感知机（神经元），那么是否可以构造多层的 SVM 构成更为强大的深度 SVM 网络呢？

机器学习中常用的分类器有：朴素贝叶斯、逻辑回归、线性回归、决策树、K 近邻分类器、SVMs、Adaboosting。每种分类器有它适用的地方和局限性。

SVM 本质就是个分类器，而非特征映射，所以直接拼起来并不能视作深度模型。关于支持向量机的探索参考《An Ensemble of Deep Support Vector Machines for Image Categorization》论文和相关文章*。深层的 SVM(D-SVM) 以标准方式训练，然后使用支持向量的核激活作为输入来训练下一层的 SVM。通过这种方式，可创建核激活的非线性组。D-SVM 的主要思想是以非线性方式组合核激活。支持向量集包含构造分类决策函数的所有信息，但是它们的核激活在标准支持 SVM 中以线性方式组合，否则优化问题会变得相当复杂。

一个针对**回归**问题的 two-layer SVM 结构如下所示，有 D 个输入，隐层的 SVM 用 S_α 表示，每个 S_α 可以学到隐层的表示 $f(x|\theta)_\alpha$。最后一层的 main SVM 用 M 表示，main SVM 将隐层的表示输出为 $g(f(x|\theta))$。

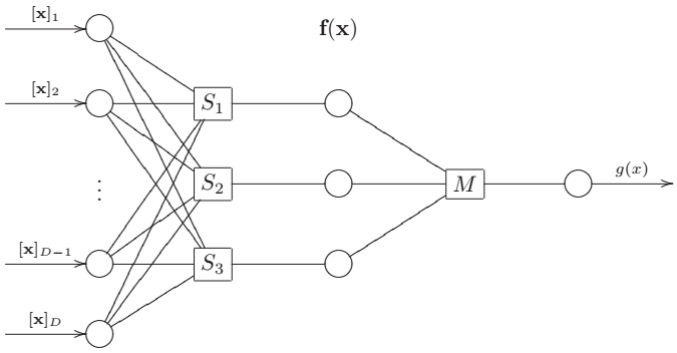

Figure 2: ML-SVM

*参考原文：《Multi-Layer Support Vector Machines》http://www.ai.rug.nl/~mwiering/GROUP/ARTICLES/Multi-Layer-SVM.pdf

回归问题的多层 SVM 算法：

1. 初始化输出层的 SVM 参数。
2. 初始化隐层的 SVMs 参数。
3. 计算隐层的 SVMs 的核矩阵。
4. 使用微扰数据集 (perturbed dataset) 训练隐层的 SVMs。
5. 重复以下步骤直到最大训练轮数 (epochs)：
   (1) 计算输出层的 SVM 的核矩阵。
   (2) 训练输出层的 SVM。
   (3) 使用反向传播方法，为隐层的 SVMs 创建训练集。
   (4) 训练隐层的 SVMs。

由于 min-max 优化问题和两层非线性的核函数，ML-SVM 不再是凸优化的问题。不过输出层 SVM 仍是一个凸优化问题。另外还可以在基于 ML-SVM 的分类问题、降维问题上再做思路的拓展。

---

5 后记

本学期学习了《Machine Learning》，也非常感谢有这个机会去接触机器学习领域的一些概念，但是深感自己在机器学习方面还处于幼儿园阶段。最后的报告选材我选择了 Deep Learning 和 SVM 相关的论文*，因为在本学期介绍的各种理论给我留下最深印象的就是支持向量机，它在深度学习（神经网络）复出前也是红极一时。最美的地方在于 SVM 的理论包含了矩阵分析、核函数理论、拉格朗日对偶等等方面，而且它和深度神经网络不同的地方就是 SVM 的解一定是全局最优的，但神经网络很可能是局部最优解（但是效果也相当出色）。所以我萌生的想法就是能否将它们结合，取长补短达到 state of the art。当然我这个粗鄙的想法早被学术界的大牛们想到而且研究过了，但我还是怀着一颗好奇的心捧起他们做过的工作，当然还有今年复试被老师问及 SVM 的推导而尴尬时，我觉得这是个机会让我弥补这个 SVM 的遗憾。感谢老师，感谢学校今年开了机器学习的课程。

*参考原文：《Deep Learning using Linear Support Vector Machines》`https://arxiv.org/abs/1306.0239`