

# 1 Introduction

此文在CNN的基础上对hidden layer进行supervision。实际上是对CNN的隐层采用squared hinge loss (L2 Loss) 使得CNN的隐层具有discriminative。本文的方法有一个先验的经验，那就是如果能够让特征discriminative，则分类效果一定就好。（不过这一点我觉得不能完全认同）

## 2 算法说明

### 2.1 符号规定

input training data:  $S = \{(X_i, y_i), i = 1, \dots, N\}$  其中  $X_i \in \mathcal{R}^n$  表示原始数据,  $y_i \in \{1, \dots, K\}$  表示对应的label。

网络层数:  $M$

学习到的filters/weights:  $W^{(m)}, m = 1, \dots, M$

$m - 1$ 层产生的feature map:  $Z^{(m-1)}$

convolved/filtered responses:  $Q^{(m)}$

Pooling function:  $f()$

输出层的SVM weights:  $w^{(out)}$

### 2.2 公式说明

对每一层  $m = 1, \dots, M$  有:

$$Z^{(m)} = f(Q^{(m)}), \text{ and } Z^{(0)} \equiv X, \quad (1)$$

$$Q^{(m)} = W^{(m)} * Z^{(m-1)}, \quad (2)$$

亦即从网络的结构上来看文中的结构与传统的CNN网络并无差别。合并所有的层的weights有:

$$W = (W^{(1)}, \dots, W^{(M)}),$$

对于each hidden layer有:

$$w = (w^{(1)}, \dots, w^{(M-1)}),$$

%%这个w是怎么来的我目前还没完全确定，文章写的不是十分清楚，智能说一下自己的理解，这个w很可能是将  $Z^{(m-1)}$  作为输入，输出为各不相同的label到L2SVM中，也就是说利用svm将一层的Z完全区分开来，这样学到一个对应的w。因此在训练之前这个w是未知的，他与每一次filters update时一起update，其update方法与CNN+SVM的策略相同，在《Deep Learning using Linear Support Vector Machines》中有详细介绍（本文后面附有链接）。

本文的核心motivation就是在hidden layer加入约束

总体上的目标函数是:

$$\|w^{(out)}\|^2 + \mathcal{L}(W, w^{(out)}) + \sum_{m=1}^{M-1} \alpha_m \left[ \|w^{(out)}\|^2 + \ell(W, w^{(m)}) - \gamma \right]_+, \quad (3)$$

这其中:

$$\mathcal{L}(W, w^{(out)}) = \sum_{y_k \neq y} \left[ 1 - \langle w^{(out)}, \phi(Z^{(M)}, y) - \phi(Z^{(M)}, y_k) \rangle \right]_+^2 \quad (4)$$

$$\ell(W, w^{(m)}) = \sum_{y_k \neq y} \left[ 1 - \langle w^{(m)}, \phi(Z^{(m)}, y) - \phi(Z^{(m)}, y_k) \rangle \right]_+^2 \quad (5)$$

对于这个目标函数有如下解释:

- (3)左边两项  $\|w^{(out)}\|^2$  和  $\mathcal{L}(W, w^{(out)})$  与传统CNN相似，不同的是在这里不是用传统CNN的softmax做分类器而是用SVM。
- (3)右边一项括号中的  $\|w^{(out)}\|^2$  和  $\ell(W, w^{(m)})$  是本文中提出的方法
- 从整体上来看这个函数，一方面它照顾到了经典CNN中误差反向传导所必须的自下至上的传递结构，另一方面还使得filters之间满足response discriminative的特性。

未完待续。。。。