Deep Convolutional Ranking for Multilabel Image Annotation

`Image annotation/tagging`

Model

Compare Some Multilabel Ranking Loss functions

- **n means number of images**
- **c means number of tags**

1. Softmax

   The loss has been used for multilabel annotation or a single image classification.

   $$P_{ij} = \frac{exp(f_j(x_i))}{\sum_{k=1}^{c} exp(f_k(x_i))}$$

   Cost function is:

   $$J = -\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{c}\bar{p}_{ij}log(p_{ij}) = -\frac{1}{m}\sum_{i=1}^{n}\sum_{j=1}^{c_+}\frac{1}{c_+ log(p_{ij})}$$

   Where c_+ denotes the number of positive labels for each image.
2. Pairwise Ranking

   Thoughts: rank the positive labels to have higher scores than negative labels, which led to the following minimization problem:

   $$J = \sum_{i=1}^{n}\sum_{j=1}^{c_+}\sum_{k=1}^{c_-} max(0, 1 - f_j(x_i) + f_k(x_i))$$

3. Weighted Approximate Ranking (WARP)
   optimizes the top-k accurary for annotation by using a stochastic sampling approach, similar to **Pairwise Ranking** except the weighted function $L(r_j)$

$$J = \sum_{i=1}^{n} \sum_{j=1}^{c_+} \sum_{k=1}^{c_-} L(r_j) max(0, 1 - f_j(x_i) + f_k(x_i))$$

   where $L(\cdot)$ is a weighted function for different ranks, r_j is the rank for the j^{th} class for image i.
   The define of $L(\cdot)$ is

$$L(r) = \sum_{j=1}^{r} \alpha_j, with\ \alpha_1 \geq \alpha_2 \geq \alpha_3 \geq \cdots \geq 0.$$

   where α_j is equal to $\frac{1}{j}$
   How to estimate the rank r_j?
   Using a random sampling method: **for a positive label, we continued to sample negative labels until we found a violation; then we recorded the number of trials s we sampled for negative labels. The rank was estimated by the ollowing formulation**

$$r_j = \lfloor \frac{c-1}{s} \rfloor$$

   where s means the numbers of sampling trials

Network details

Train a CNN net from scratch.
Setting of Net: **Based on AlexNet, image size: 256 x 256; patches: 220 x 220; filter size: 11,9,5; FC-layer:4096; Dropout:0.6;ReLU**
Setting of Training: **asynchronized stochastic gradient descent with a momentum term with weight 0.9; mini-batch size:32; global lr: 0.002; staircase weight decay**

Results

method / metric	per-class recall	per-class precision	overall recall	overall precision	$N+$
Upper bound	97.00	44.87	82.76	66.49	100.00
Visual Feature + kNN	19.33	**32.59**	53.44	42.93	91.36
Visual Feature + SVM	18.79	21.51	35.87	28.82	82.72
CNN + Softmax	31.22	31.68	59.52	47.82	98.76
CNN + Ranking	26.83	31.93	58.00	46.59	95.06
CNN + WARP	**35.60**	31.65	**60.49**	**48.59**	**96.29**

Table 1: Image annotation results on NUS-WIDE with $k = 3$ annotated tags per image. See text in section 5.4 for the definition of "Upper bound".

method / metric	per-class recall	per-class precision	overall recall	overall precision	$N+$
Upper bound	99.57	28.83	96.40	46.22	100.00
Visual Feature + kNN	32.14	**22.56**	66.98	32.29	95.06
Visual Feature + SVM	34.19	18.79	47.15	22.73	96.30
CNN + Softmax	48.24	21.98	74.04	35.69	98.76
CNN + Ranking	42.48	22.74	72.78	35.08	97.53
CNN + WARP	**52.03**	22.31	**75.00**	**36.16**	**100.00**

Table 2: Image annotation results on NUS-WIDE with $k = 5$ annotated tags per image. See text in section 5.4 for the definition of "Upper bound".