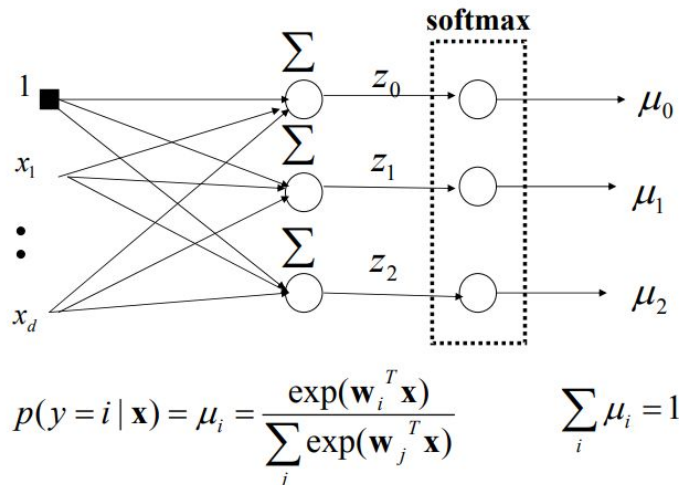


# Improving Generalization via Scalable Neighborhood Component Analysis

Обзор статъи

14.08.2018  
18.09.2018

- Основные задачи computer vision обычно формулируются как задачи классификации
- В deep neural network классификаторы обычно реализуются, используя multi-way parametric softmax:



- Это означает, что рассматриваемые классы при обучении и при тестировании одни и те же
- Однако такое предположение о замкнутости мира плохо работает на реальных данных

# Open-set nature

Как можно решить эту проблему?

Получить новый классификатор

- ресурсоемко

# Open-set nature

## Как можно решить эту проблему?

### Transfer learning

отрезать последний слой  
(softmax classification layer)  
и дообучить предпоследний слой

- полученный классификатор имеет смысл только для разделения пространства тренировочных данных и может быть непригодным в пространстве новых данных

# Open-set nature

## Как можно решить эту проблему?

### Non-parametric embedding

to directly optimize a feature representation which preserves distance metrics in a non-parametric fashion

Хорошо подходят для meta-knowledge transfer (в отличие от параметрических моделей)

Пока конкурентоспособных моделей, использующих данный подход, не было предложено

# Non-parametric embedding

## Идея

1. Обучаемся на данных  $D$
2. Получаем на вход изображение  $x$  из другого набора данных  $D'$
3. Вычисляем  $v = f(x)$ ,  
где  $f(x)$  - это  $DNN$
4. Находим в  $D$  изображения, которые максимально похожи на  $v$
5. Информация из выбранных изображений может быть перенесена на  $x$

# Neighbourhood components analysis

## Используется для обучения сети

- Классификация многомерных данных в соответствии с заданной метрикой расстояния
- Почти тоже, что и *kNN*
- “Обучает” метрику расстояния путем нахождения линейного преобразования входных данных так, чтобы в полученном пространстве эффективность классификации увеличивалась

# Non-parametric formulation of classification

Имеем набор данных  $x_1, x_2, \dots, x_n$  принадлежащие классам  $y_1, y_2, \dots, y_n$

Каждый экземпляр преобразуется в вектор  $v_i = f_\theta(x_i)$

Дальше - вычисляем матрицу подобия  $s_{ij} = \cos(\phi) = \frac{v_i^T v_j}{\|v_i\| \|v_j\|} = v_i^T v_j$

Элемент  $x_i$  будет соседом  $x_j$  с вероятностью  $p_{ij} = \frac{\exp(s_{ij}/\sigma)}{\sum_{k \neq i} \exp(s_{ik}/\sigma)}$

Определим множество  $\Omega_i = \{j | y_j = y_i\}$

Тогда вероятность, что элемент  $x_i$  будет правильно классифицирован, можно найти по формуле

$$p_i = \sum_{j \in \Omega_i} p_{ij}$$



# Non-parametric formulation of classification

Целевая функция  $J = \frac{1}{n} \sum_i J_i = -\frac{1}{n} \sum_i \log(p_i)$

Градиент данной целевой функции находится по формуле

$$\frac{\partial J_i}{\partial v_i} = \frac{1}{\sigma} \sum_k p_{ik} v_k - \frac{1}{\sigma} \sum_{k \in \Omega_i} \tilde{p}_{ik} v_k$$

$$\frac{\partial J_i}{\partial v_j} = \begin{cases} \frac{1}{\sigma} (p_{ij} - \tilde{p}_{ij}) v_i, & j \in \Omega_i \\ \frac{1}{\sigma} p_{ij} v_i, & j \notin \Omega_i \end{cases}$$

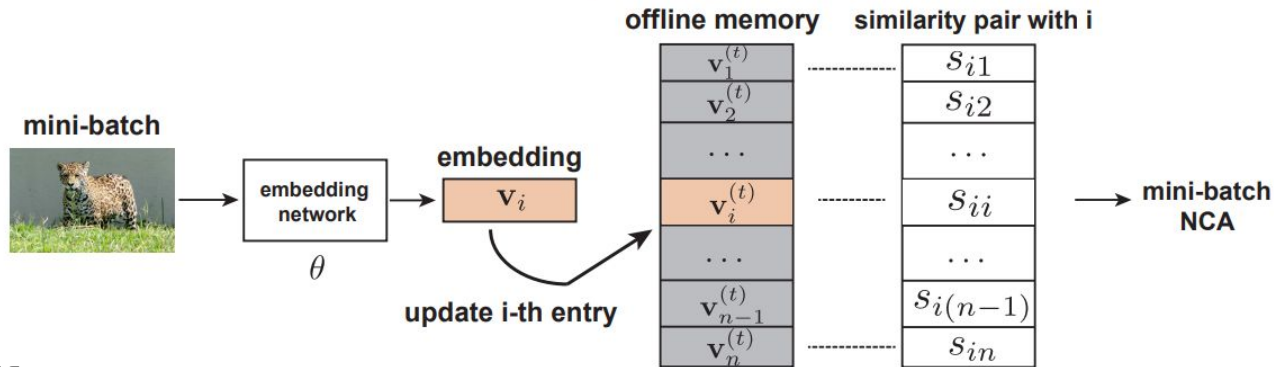
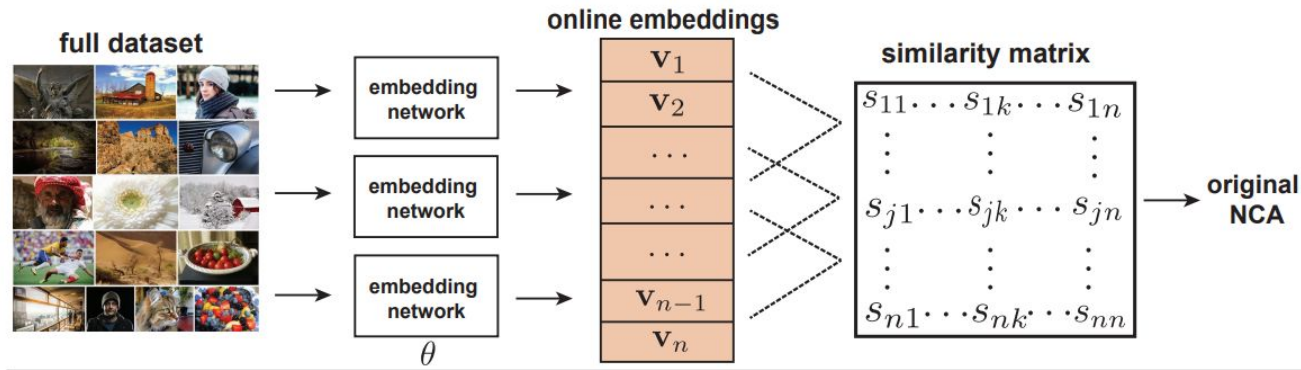
$$\tilde{p}_{ik} = p_{ik} / \sum_{j \in \Omega_i} p_{ij}$$

# Defeat bottleneck

1. Для обучения используется *SGD*

2. В качестве градиента целевой функции берут  $\frac{\partial J_i}{\partial v_i} = \frac{1}{\sigma} \sum_k p_{ik} v_k - \frac{1}{\sigma} \sum_{k \in \Omega_i} \tilde{p}_{ik} v_k$

3. Используют расширенную память для хранения  $v_i$



$t$ -ая итерация

$$v_i^{(t)} \approx f_{\theta^{(t)}}(x_i) \quad \frac{\partial J_i}{\partial v_i} = \frac{1}{\sigma} \sum_k p_{ik} v_k^{(t)} - \frac{1}{\sigma} \sum_{k \in \Omega_i} \tilde{p}_{ik} v_k^{(t)} \quad p_{ij} = \frac{\exp(s_{ij}/\sigma)}{\sum_{k \neq i} \exp(s_{ik}/\sigma)}$$

$$v_i^{(t+1)} \leftarrow m \cdot v_i^{(t)} + (1 - m) \cdot v_i$$

# Image Recognition

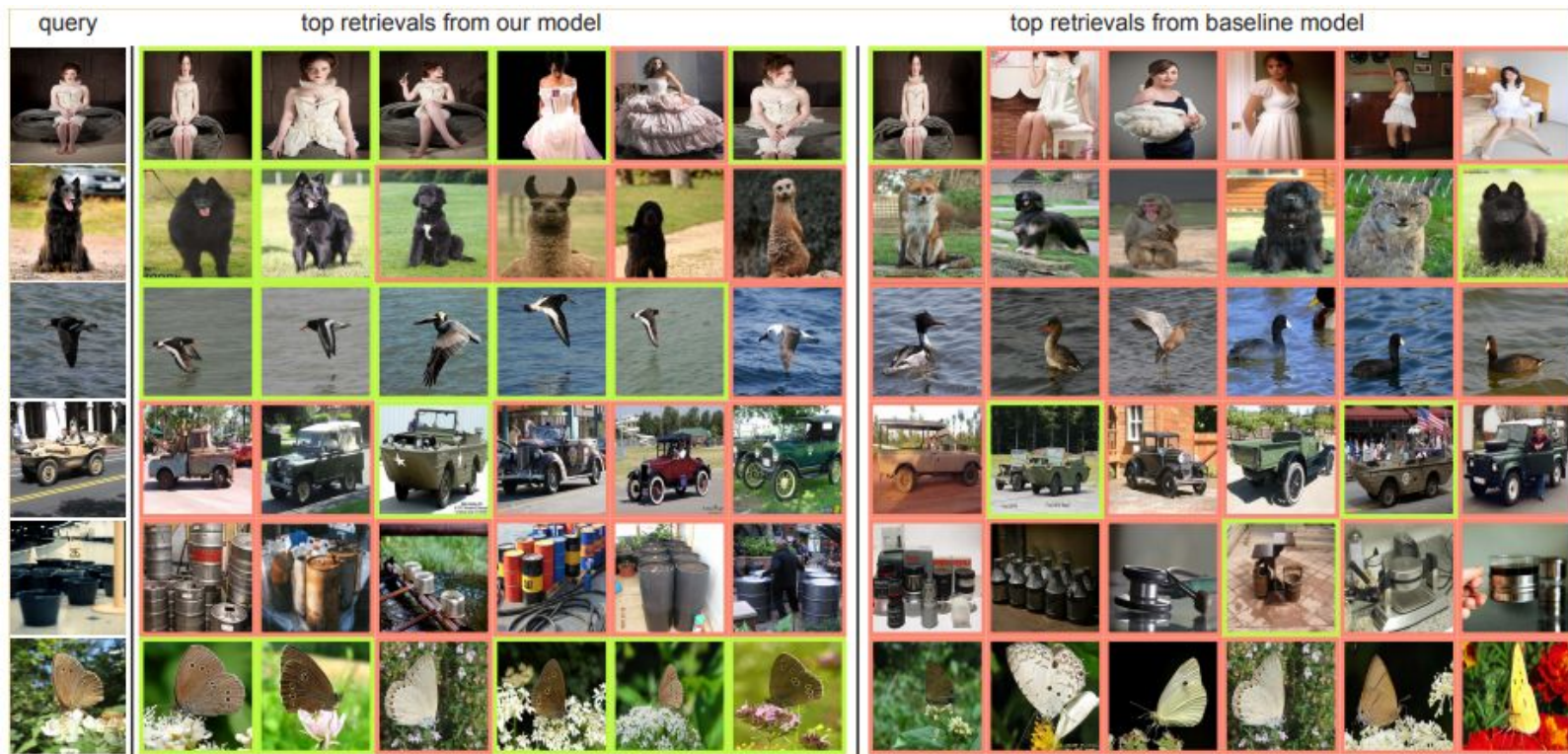
- Лучше, чем baseline на 3% при  $k = 1$
- Лучше на 1.1 %, чем *ResNet34*,  
и на 0.7%, чем *ResNet50* при  $k = 30$

# Discovering Sub-Categories

- Для каждого класса обнаруживает примерно *100-500* соседей
- *ImageNet* при этом имеет около *1000* изображений в каждом классе

CIFAR			ImageNet		
Task	20 classes	100 classes	Task	127 classes	1000 classes
Baseline	81.53	54.17	Baseline	81.48	48.07
Ours	81.42	<b>62.32</b>	Ours	81.62	<b>52.75</b>

# Discovering Sub-Categories





# Few-shot Recognition

- В связи с подходом модель не так сильно зависит от количества представителей каждого класса при обучении

Table 5: Few-shot recognition on Mini-ImageNet dataset.

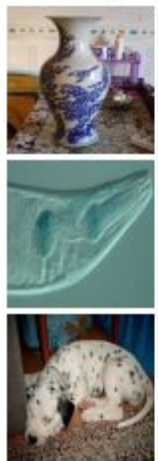
Method	Network	FineTune	5-way Setting		20-way Setting	
			1-shot	5-shot	1-shot	5-shot
NN Baseline [42]	Small	No	41.1±0.7	51.0±0.7	-	-
Meta-LSTM [29]	Small	No	43.4±0.8	60.1±0.7	16.7±0.2	26.1±0.3
MAML [6]	Small	Yes	48.7±0.7	63.2±0.9	16.5±0.6	19.3±0.3
Meta-SGD [20]	Small	No	50.5±1.9	64.0±0.9	17.6±0.6	28.9±0.4
Matching Net [42]	Small	Yes	46.6±0.8	60.0±0.7	-	-
Prototypical [36]	Small	No	49.4±0.8	<b>68.2±0.7</b>	-	-
RelationNet [39]	Small	No	<b>51.4±0.8</b>	61.1±0.7	-	-
Ours	Small	No	50.3±0.7	64.1±0.8	<b>23.7±0.4</b>	<b>36.0±0.5</b>
SNAIL [27]	Large	No	55.7±1.0	68.9±0.9	-	-
RelationNet [39]	Large	No	57.0±0.9	71.1±0.7	-	-
Ours	Large	No	<b>57.8±0.8</b>	<b>72.8±0.7</b>	<b>30.5±0.5</b>	<b>44.8±0.5</b>

*t-way* - количество классов

*s-images* - количество изображений в каждом классе

# Few-shot Recognition

query



five learning examples



query



five learning examples

