前瞻深度學習 Homework01 - Introduce a New NN with Memory

Deep Structured Scene Parsing by Learning with Image Descriptions

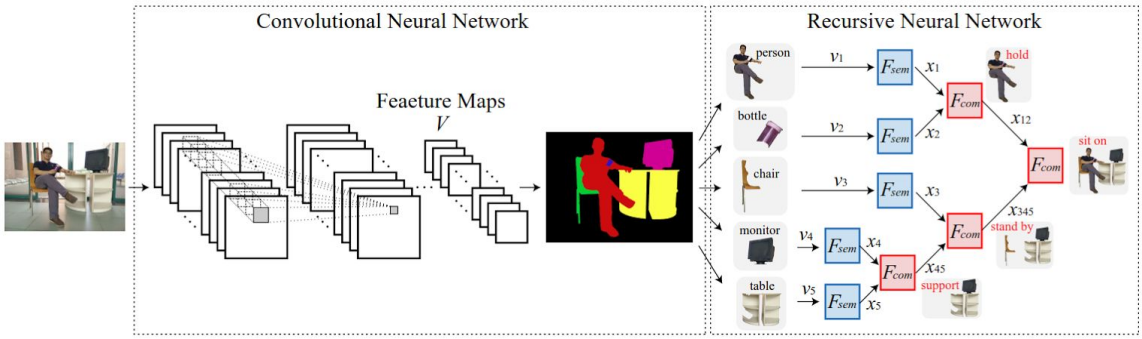
arXiv:1604.02271v2 [cs.CV]13 Aug 2016 ***Reference*** :https://arxiv.org/abs/1604.02271

105062504 魏凱亞 105062518 林怡均

Introduction

This paper addresses a some problem of scene understanding :(1)The representations of nested hierarchical structure in scene images are often ambiguous. (2) Training a scene parsing model usually relies on expensive artificial annotations.

They propose a novel deep architecture that automatically parses an input scene into a structured and meaningful configuration. The model consisting of two networks:CNN and RNN.They use CNN to extract the image representation for pixelwise object labeling. Then,using RNN to construct the hierarchical object structure tree and discover the inter-object relations.By leveraging the descriptive sentences of the training images, they train the model in a weakly-supervised way.Once these scene configurations are determined, then they updated the parameters of both the CNN and RNN by backpropagation.



Proposal

CNN model:

The CNN model is used to layerwise extract features from the input scene image and generate the representations of semantic objects. The input image is fed into the revised VGG-16 network to generate a score map for each entity category. Then, each pixel is labeled with an entity category. Finally, they group the adjacent pixels

with the same label into an object, and obtain the feature representations of objects. The following are the two tasks of the CNN model:

(1) semantic labeling :

They used the fully convolutional network with parameters Wc to yield K + 1 score maps {s^0 , ..., s^k , s^K }, corresponding to one extra background category and K object categories. By using softmax to obtain the corresponding classification score: $\sigma(s_j^t)$ is the probability of j-th pixel belonging to t-th object category.

$$\sigma(s_j^t) = \frac{\exp(s_j^t)}{\sum_{k=1}^{K} \exp(s_j^k)} \quad \text{with} \quad \sum_{t=1}^{K} \sigma(s_j^t) = 1$$

The label of the j-th pixel can be predicted with $\sigma(s_j^t)$ by:

$$c_j = \arg\max_t \ \sigma(s_j^t)$$

c_j is a set of labels of pixels in the image I, $C = \{c_j\}_{j=1}^{M}$,where c_j ∈ {1, ..., K } and M is the number of pixels of image I.

(2) generating feature representations for entities:

After obtaining c_j, they produced a semantic entity category by grouping the adjacent pixels with the same label. They use Log-Sum-Exp to obtain feature representation with fixed length for any entity category:

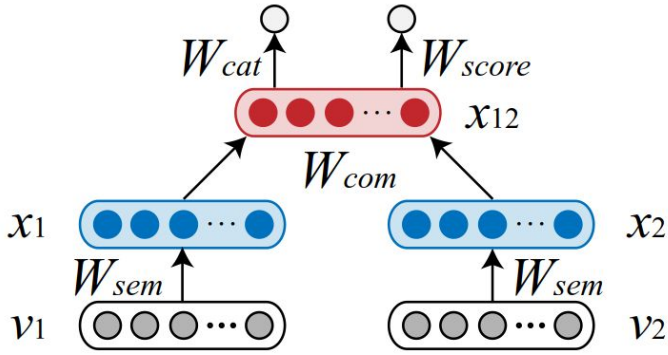
$$v_k = \frac{1}{\pi} \log \left[\frac{1}{Q_k} \sum_{c_j = k} \exp(\pi \bar{v}_j) \right]$$

v_k : the feature representation of the k-th entity category, $\bar{v}_j$: the feature representation of the j-th pixel by concatenating all feature maps at the layer before softmax at position j into a vector, Q_k : the total number of pixels of the k-th object category, π : hyperparameter to control smoothness.

RNN model:

Based on the CNN feature representations, the RNN model produced the structured configuration of the scene. In each recursive iteration, two input objects (child) are merged into a higher-level object (parent). They used RNN to generate the image parsing tree with a greedy algorithm and predict the relation between these two nodes when they are combined into a higher-level node.The model

consists of four sub-networks:{ semantic mapper, combiner, categorizer, scorer }. The parameters denoted as $W_R = \{W_{sem}, W_{com}, W_{cat}, W_{scor}\}$ for each part. F_* is network transformation.



(1) Semantic mapper: It is a one-layer fully-connected network, which mapped object feature v_k onto a semantic space, where x_k is the mapped feature.

$$x_k = F_{sem}(v_k; W_{sem})$$

(2) Combiner: The features of two child nodes from Semantic mapper are fed to the Combiner and generate their parent node feature. x_k and x_l are the two child features and x_{kl} is their parent feature. The parent and child node feature has the same dimensionality, allowing the procedure can be applied recursively.

$$x_{kl} = F_{com}([x_k, x_l]; W_{com})$$

(3) Categorizer: After merging two nodes, the Categorizer which is a softmax classifier determines the relation label y_{kl} .

$$y_{kl} = softmax(F_{cat}(x_{kl}; W_{cat}))$$

(4) Scorer: The Scorer measures the confidence of a merging operation between two nodes. h_{kl} is a real value.

$$h_{kl} = F_{score}(x_{kl}; W_{score})$$

The merging score q_{kl} of node $\{kl\}$ is $q_{kl} = \frac{1}{1 + exp(h_{kl})}$,which used to optimize the structure discovery in training.

Weakly-supervised Model Training:

Describing images by sentences finely accords with the process of human vision, and provides richer semantic and structured contexts. In the inatial stage of training,

they first convert each sentences into a normalized tree, T, including entity labels and relations.

The model aims to perform two task: semantic labeling and scene structure discovery. The loss function is the sum of two terms:

$$\mathcal{J}(W) = \frac{1}{Z}\sum_{i=1}^{Z}(\mathcal{J}_C(W_C; I_i, T_i) + \mathcal{J}_R(W; V_i, T_i)) \quad (8)$$

With a training set containing Z image-tree pairs. I_i is the i-th image and T_i is the tree structure prodeced from the descriptive sentence. V_i is the set of semantic entity feature produced by CNN.

The following term is semantic label loss:

$$\begin{aligned}\mathcal{J}_C(W_C; I, T) = -\frac{1}{M}(\sum_{j=1}^{M}\sum_{k=1}^{K}\mathbf{1}(\widehat{c}_j = k)\log\sigma(s_j^k) \\ + (1 - \mathbf{1}(\widehat{c}_j = k))\log(1 - \sigma(s_j^k))) + \|W_C\|^2\end{aligned} \quad (9)$$

M is the total number of pixels in image I. K denotes the object category. c_j is the estimated category label of pixel j. This loss function measures pixel-wise classification problem by using CNN.

The following term is the **scene structure loss**. This function can be devided into two sub-tasks: tree structure construction and relation categorization:

$$\mathcal{J}_R(W; V_i, T_i) = \mathcal{J}_{struc}(W; V_i, T_i) + \mathcal{J}_{rel}(W; V_i, T_i) \quad (10)$$

The goal of tree structure contruction is to learn a transformation $I \rightarrow P_I$,the parsing tree of image I. P_T is the total number of merging operation in the text parsing tree T. A image parsing tree is valid if the sequence of two regions merges is consistent with the merging order in the text parsing tree. The main idea of the loss function is to encourage correct merging operation a to have larger merging score than that of incorrect merging operation $\overline{a}(hat)$.

$$\begin{aligned}\mathcal{J}_{struc}(W; V, T) = \frac{1}{P_T}\sum_{p=1}^{P_T}[\max_{\widehat{a}_p \notin \mathcal{A}(V,T)} q(\widehat{a}_p) \\ - q(a_p) + \triangle\,] + \frac{\lambda}{2}||W||^2\end{aligned} \quad (11)$$

λ is the weight regularizer. The loss function above maximizes the score of correct merging operation and minimizes incorrect one.

The **relation categorization** can be optimized as a softmax classification problem.

$$\mathcal{J}_{rel}(W;V,T) = -\frac{1}{|U_T|}(\sum_{\{kl\}}\sum_{s=1}^{S}\mathbf{1}(r_{kl}=s)\log G_s(\theta_{kl}(V,W)) + (1-\mathbf{1}(r_{kl}=s))\log(1-G_s(\theta_{kl}(V,W))) + \|W\|^2 \tag{12}$$

$|U_T|$ is the number of relations appeared in the tree structure T. $\{kl\}$ denotes the node mreged from node k and node l. S is the total number of relation categories. r_{kl} is the ground truth relations provided by tree structure T. $G_s(\theta_{kl}(V,W))$ is the categorizer sub-network in the RNN that output the probability the node $\{kl\}$ belongs to relation category s.

Learning Algorithm:

1. Updating intermediate label $\overline{C}(hat)$ and the CNN loss.

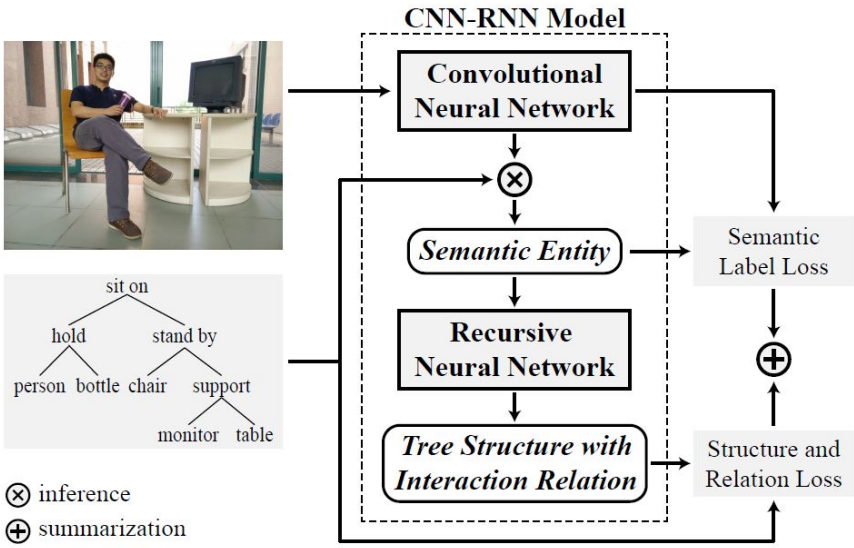
   Given image I and its senmantic tree T, step one is to compute the classification probability of each pixel.

2. Updating latent scene structures and the RNN loss.

   Given the label of each pixel, they first group the pixels into semantic objects and pbtain the feature representation. Then, they use the RNN model to infer the relation and hierarchical configuration bwtween objects and compute the RNN loss.

3. Updating the CNN and RNN parameters.

   Compute the gradient off overall loss (CNN and RNN parameters) with the back-propagation algorithm, given the intermediate label maps and latent scene structure.



unique properties of the new NN

They use a property of the RNNs to recursively learn the representations in a semantically and structurally coherent way.

To avoid relying on the elaborative annotations, they propose to train the model in a weakly-supervised way by leveraging the image descriptions.

Applications of taking advantage of the properties

scene understanding

Image/video captioning

question answering