Attention Network

Joohyung Lee

References

- A Beginner's Guide to Attention Mechanisms and Memory Networks
- Attention in NN and How to Use It
  -Adam Kosiorek, Oct 2017
  - Ability to 'focus' on a subset of its inputs (or features)
- Hassabis, Demis, et al. "Neuroscience-Inspired Artificial Intelligence," Neuron, Elsevier, 2017

R-CNN

Joohyung Lee

Introduction 1

- Objective: object detection & semantic segmentation
- Two achievements:
  - CNN for region proposals for localization & segmentation
  - 'Transfer learning' for an auxiliary task followed by domain-specific fine-tuning
- Inspiration
  - Though histogram representation(HOG and SIFT) may associate with V1, recognition occurs several states downstream suggesting hierarchical, multi-stage processes for feature computation.
  - AlexNet syndrome
- First paper that apply CNN to object detection

Introduction 2

- Approaches
  - Regression problem (Szegedy et al, low performance)
  - Sliding window (popular, few layers for better resolution)
    - Precise localization remains an open technical challenge
- “recognition using regions”
  - region proposal method generates around 2k candidates
  - proposed regions → affine warped(fix image size) → CNN for feature extraction → linear SVM for classification (category-specific)
- Scarce labeled data
  - supervised pre-training(ILSVRC) followed by a domain-specific fine-tuning works (8% boost on PASCAL)
  - AlexNet can be used without any fine-tuning (empirical find-out)

Module 1/3: Region Proposal

- R-CNN is agnostic to the particular region proposal method
- Selective search to enable a controlled comparison with prior detection work

Module 2/3: Feature extraction

- AlexNet (requires 227*227)
- Dilate bounding box (16 pixels)

Test-time detection

- For each class, filter out all overlapping regions candidates with less score (if score is higher than a learned threshold)
- Efficient
  - Features are 1) in low dimension, 2) small in quantity(360k vs. 4k)
  - features are batched: 2000 →2000*4096 X 4096*N (# of classes)

Training (1)

- 1) Supervised pre-training
  - ILSVRC 2012 (image-level annotation)
  - Performance similar to that of AlexNet
- 2) Domain-specific fine-tuning
  - Replace FC with (N+1) way classification layer (+1 for background)
  - Proposed region is positive if ≥ 0.5IoU
  - initial learning rate 0.001 (10% of initial pre-training rate)
  - mini-batch: 128=32 positive windows(over all classes)+96 background
  - Sampling bias towards positive windows which are very rare

Training (2)

- 3) Object category classifiers
  - Proposed region is negative if IoU $\leq$ 0.3
  - 0.3 from grid search 0:0.1:0.5 using validation set
  - One linear SVM per class
- 4) Bounding-box regression
  - class-specific
  - after CNN (parallel to SVM)
  - Px,Py,Pw,Ph
  - Optimized by L2norm between $W^T P$ and t_*, which is f(G,P)
  - least square regularization
  - applied only once for iterative fashion does not improve the performance

Reference

- Girshick Ross, et al. "Rich feature hierarchies for accurate object detection and semantic segmentation." CVPR '14, 580-587

SPP-Net

Joohyung Lee

Abstract

- Purpose
  - To eliminate the constraint that the input image shall be in fixed-size format for CNN.
- Approach: SPP-net
  - generates fixed-length representation regardless of image size/scale
  - robust to object deformations
- Results
  - Classification
    - Improvement on all CNN based classification methods
  - Object detection
    - Significant improvement in the speed (x102 faster)
    - Near state-of-the-art performance (rank #2,#3)

Introduction (1)

- Revolutionary change in vision community
  - CNN, large scale training data
- Technical issue in 1) training, 2) testing
  - fixed input image size which limits both the aspect ratio and the scale of the input image
- Current approach
  - cropping or warping which cause distortion or unfit ROI
- Why?
  - CNN consists of 1) conv layer, 2) fully-connected layer (problematic)

Introduction (2)

- Our approach: spatial pyramid pooling (SPP)
  - SPP between the last conv layer and fully connected layer
- History of the SPP-net
  - Spatial Pyramid Matching as an extension of the Bag-of-Word model
- Advantage of SPP over traditional Conv
  - fixed length output regardless of the input size
  - multi-level pooling (robust to object deformation) instead of a single window size
- Allows to feed images with varying sizes or sales during training
  - increase scale-invariance, reduces over-fitting
  - switch the input-size per epoch.
  - Use different models which however share the parameter values

Introduction (3)

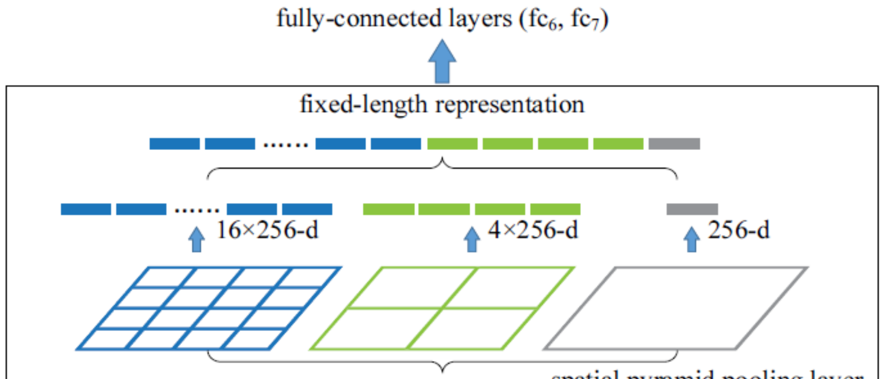
- SPP improves classification networks
  - Improved 4 networks for ImageNet
  - It is thus reasonable to conjecture that SPP should improve more sophisticated convolutional architectures
  - State-of-the-art on Caltech101, Pascal VOC 2007
- SPP is strong in object detection task
  - Run conv net ONLY ONCE per image, extract features by SPP-net
  - 24~102x faster, better or comparable accuracy

Convolutional layers and feature mapping

- pooling is also conv due to sliding window fashion
- idea of *feature maps* in CNN
- filter can be activated by some semantic content
- traditional perspective adapted in CNN
  - CNN -> FC
  - feature representation -> BoW or spatial pyramid
    - kernel method with different cardinality (orderless / order)

The Spatial Pyramid Pooling Layer

- Discrepancy: conv (various->various), classifier (fixed->fixed)
- Discrepancy can be treated by BoW approach (word is, in this case, spatial bin?)
- Replace the last pooling later with a SPP layer.
- The outputs of the spatial pyramid pooling are kM-dimensional vectors (k: # of filters, M: # of bins)
- Global average pooling corresponds to the traditional BoW method



Training

- Single-size training
  - 224*224 only
- Multi-size training
  - 180*180, 224*224

Object Detection

- Feature extraction by CNN followed by selective search for object proposal
- SPP followed by FC for each proposals
- binary linear SVM for each category on the features just like RCNN.
- bounding box regression to post-process the prediction windows following RCNN.
  - regression layer after SPP but parallel to SVM layer

Reference

- He, Kaiming et al, “Spatial Pyramid Pooling in Deep Convolutional Networks for Visual Recognition”

Fast R-CNN

Joohyung Lee

Abstract

- Objective: object detection
- Fast (training, test), good accuracy

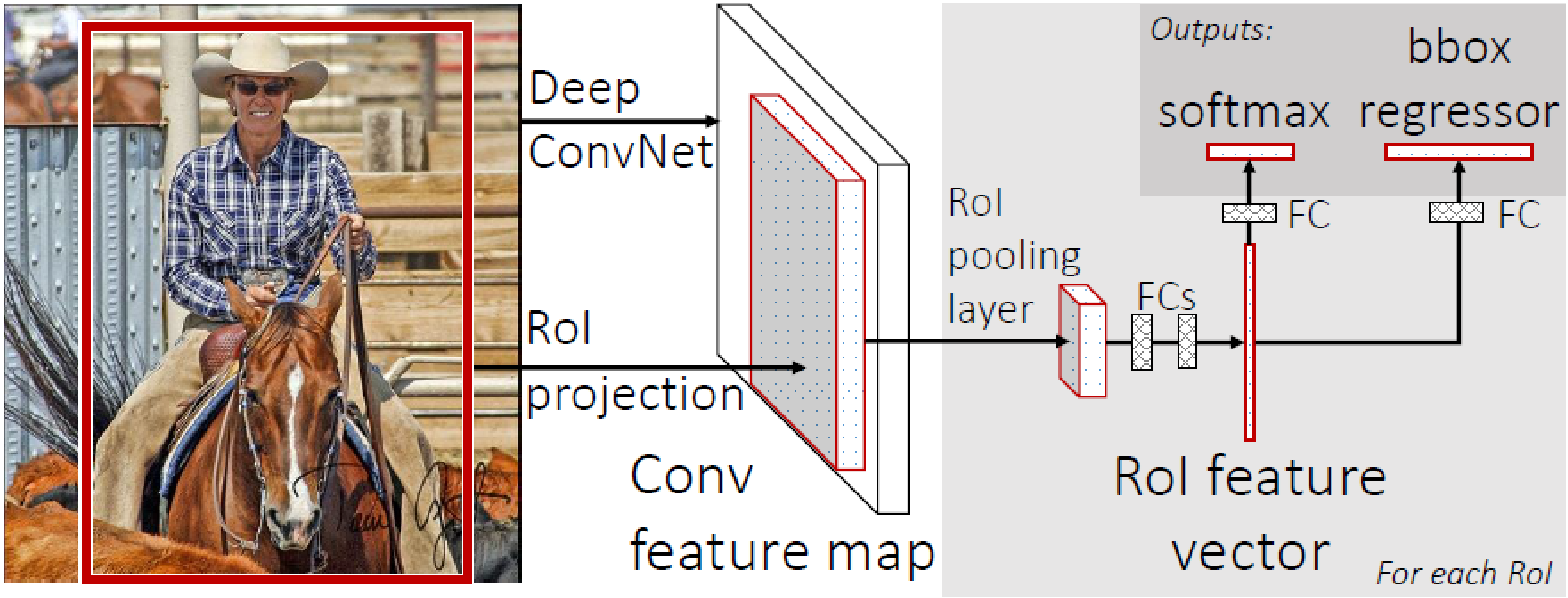
Introduction

- Two primary challenges for accurate localization:
  - Numerous candidates for object locations shall be processed
  - It needs further refinement to boost accuracy in that the candidates only offers a rough localization
- Single-state training for 1) classification of the proposed objects and 2) further refinement

Limitation & Improvement

- RCNN
  - multi-stage pipeline: CNN → SVM → Bounding-Box
  - Expensive training: features for each proposal are stored in disk for SVM&BB
  - slow detection (inference): feature extraction from each object proposal per image
- SPPnet
  - Still multi-stage pipleline
  - fine-tuning cannot update the conv layers before SPP
- FRCNN
  - single-stage using multi-task loss: 1)no need for disk storage, 2) all layer update
  - higher mAP than RCNN and SPPnet

Architecture and training



Architecture 1

- RoI pooling layer
  - single level spatial pyramid pooling
- Initializing networks
  - AlexNet, VGG-Net
  - ROI pooling instead of the last max pooling layer. H&W compatible with the net's first FC layer (7*7 for VGG16)
  - parallel FC+Softmax and bb regressor
  - Network receives two data inputs: list of images and RoIs
- Fine-tuning
  - hierarchical sampling: sampling N images -> sampling R/N RoIs from each image. No slow convergence effect due to inter-correlation
  - Streamlined training process

Architecture 2

$$L(p, u, t^u, v) = L_{\text{cls}}(p, u) + \lambda[u \geq 1] L_{\text{loc}}(t^u, v),$$

$$L_{\text{loc}}(t^u, v) = \sum_{i \in \{\text{x,y,w,h}\}} \text{smooth}_{L_1}(t_i^u - v_i),$$

in which

$$\text{smooth}_{L_1}(x) = \begin{cases} 0.5x^2 & \text{if } |x| < 1 \\ |x| - 0.5 & \text{otherwise,} \end{cases}$$

- Multi-task loss
  - normalize v_i
    - zero mean, unit variance
  - λ=1
- Mini-batch sampling
  - N=2, R=128 -> 64 RoIs per image
  - 25% RoIs from proposal with at least 0.5 IoU with gt bb
  - foreground object class only
- Back-propagation through RoI pooling layer
  - ?
- Scale invariance
  - following SPP-net (pre-defined pixel size)
  - random pixel size (with constraint?)-data augmentation

Others

- mtl improves both classifier and bb regression
- multi scale slightly outperforms single scale but considerable lose in computation time
- softmax slightly outperforms SVM
- More proposal? not much change…. rises at the beginning and falls if proposal increases more

Reference

- Girshick, Ross, “Fast R-CNN,” arxiv:1504.08083v2, 2015.

Faster R-CNN

이주형

Abstract

- Novelty: Region Proposal Network (RPN) that shares full-image convolutional features with the detection network
- RPN is an FCN that simultaneously predicts object bounds and objectness scores at each position.
- Proposed regions are used by Fast R-CNN for detection.
- We further merged RPN and Fast R-CNN into a single network by sharing convolutional features

Introduction (1)

- Region-based CNN’s high computation cost has been reduced by sharing convolutions across proposals
- Now, proposal itself became the computation bottleneck
- Selective search is popular. EdgeBoxes currently provides the best tradeoff between the quality and speed but still slow
- Proposal may consume much time for it is held in CPU whereas CNN is computed using GPU
- Introduce ‘Region Proposal Networks’ (RPNs)
  - share convolutional layer with object detection network

Introduction (2)- RPN

- Output:
  - Region bound
  - Objectness score at each location
- FCN and thus can be trained end-to-end
- Predict region proposals with a wide range of scales and aspect ratios
- Introduce “anchor” boxes which is a pyramid of regression references
  - multiple scales and aspect ratios
- Training scheme that alternates b/w fine-tuning for proposal and detection
- Good results in various open competitions

Related Works

- Object Proposals
  - based on 1) super-pixels, 2) sliding windows. External, indep modules
- Deep Network for Object Detection
  - Since RCNN does NOT predict object bound(except bb refinement), its accuracy depends on the performance of the region proposal
  - Overfeat: FC for bb coordinate regression (classification -> where?)
  - MultiBox: multiple class-agnostic boxes. Does NOT share features
  - DeepMask: learning segmentation proposals
- Shared computation of convolutions
  - OverFeat: conv from an image pyramid
  - SPP: efficient region-based object detection and semantic segmentation

Architecture - Overall

- Two modules
  - region proposal module
  - Fast R-CNN detector
  - entire system is a single, unified network for object detection.
  - with ‘attention’ mechanism, RPN module tells the Fast R-CNN module where to look

Architecture - RPN

- Input: image (of any size)
- Output: set of rectangular object proposals, each with objectness score
- Assumption: RPN and object detection network will share a common set of conv layers
- 1) n*n(e.g. 3*3) filter to generate feature map from last conv
- 2) two 1*1 filters to generate both box-regression layer, the coordinate relative to the anchor location(4k) and box-classification layer(2k) for k proposals(i.e. anchors)

RPN

- Translation-Invariant
  - RPN is T-I unlike MultiBox(k-means for anchor proposal)
  - Reduces the model size -> expect to have less risk of overfitting on small dataset
- Multi-scale anchors as regression references
  - Image/feature pyramid: time-consuming
  - Sliding windows of multiple scales(a/o aspect ratio): joint with above?
  - Pyramid of anchors: classifies and regresses bb with reference to anchor boxes of multiple scales and aspect ratio
- Loss Function
  - Positive label if, 1) anchor with the highest IoU to gt or 2) anchor with IoU higher than 0.7
  - Negative label if IoU is lower than 0.3 for all gt
  - If neither of the above cases, not used for training
  - Multi-task loss (sum of the normalized loss of cls and reg). Ratio is controlled by the hyperparameter

Others

- Training RPNs
  - “Image-centric” sampling strategy
  - Negative and positive anchors sampled per image to keep 1:1 ratio between negative and positive
  - 128:128 = # of positive anchors: # of negative anchors
- Training both RPN and Fast R-CNN
  - alternate training
  - joint (merge the loss)
  - ‘almost’ joint

Reference

- Shaoqing. Ren, et al (2016) “Faster R-CNN: Towards Real-Time Object Detection with Region Proposal Networks” arXiv:1506.01497v3 [cs.CV]

Mask R-CNN

이주형

Abstract

- Extends Faster R-CNN by adding a segmentation branch in parallel with the existing branch (detection).
- Fast: 5fps
- Easy to generalize to other task (i.e., human pose estimator)
- Top results
  - COCO Instance segmentation
  - COCO bounding box
  - COCO person keypoint detection
- Solid baseline & future research in instance-level recognition
- Code available at GITHUB

Introduction (approach)

- Recent improvement on baseline systems: RCNN, FCN
- Our goal: framework for 'instance segmentation'
  - Object detection + semantic segmentation
- Mask R-CNN = Faster R-CNN + small FCN in parallel
- Approach 1: preserving spatial location ('RoIAlign')
  - Problem: pixel-to-pixel alignment ('RoIPool')
  - Drastic improvement (more with stricter localization metric)
- Approach 2: decouple mask and class prediction
  - No competition among classes
  - Experimentally, conventional FCN's per-pixel multi-class categorization (coupled two tasks) worked poorly for instance segmentation

Introduction (result)

- Surpassed all previous state-of-the-art single-model
  - Instance segmentation, object detection, ablation study for various component
- Fast
  - Inference: 200ms per frame on a GPU
  - Training (COCO): 2 days on a single 8-GPU machine
- Generality of the framework
  - Human pose estimation (COCO key-point dataset)
    - Surpassed the state-of-the-art
    - 5 fps
- Mask R-CNN is a flexible framework for instance-level 'recognition'

Related Works

- R-CNN
- Instance Segmentation
  - 1) segment proposal <u>before</u> classification module, e.g. Fast RCNN
  - 2) segment proposal -> bounding-box -> classification (cascade)
  - 3) FCIS (simultaneous classification, bb, mask) exhibits error (mask)
  - 4) semantic segmentation -> classification (*segmentation-first* strategy)
  - Mask RCNN is an *instance-first* strategy

Mask R-CNN

- First stage: RPN (identical to Faster R-CNN)
- Second stage: additional binary mask for each RoI in parallel
  - per class network (loss contribution only if the class were detected)
  - multi-task loss (addition): $L = L_{cls}+L_{box}+L_{mask}$
  - L_{mask}: per pixel sigmoid, binary cross-entropy loss

RoIAlign

- For example, if CNN contains 4 poolings with stride 2, the feature map stride will be 16.
  - RoIPool computes [x/16] to align the input image and the feature map
  - RoIAlign 1) computes x/16 and utilizes bilinear interpolation to compute the exact feature values with better alignment and 2) aggregate the result using max or average pooling as in Faster-RCNN

Reference

- Kaiming. He, et al (2018) “Mask R-CNN”. arXiv:1703.06870v3 [cs.CV]