R-CNN

Daniel Schwartz

May 2019

1 Notes

1.1 Representor Theorem

1.1.1 Statistical Learning Theory

- From Statistics and Functional Analysis

1.1.2 Group of algorithms that cater to pattern analysis

- Focus on identifying patterns in data
- Most require data converted into feature vectors

1.1.3 Kernel Methods

- Do not require feature vectors but similarity functions
- Have advantage of operating on a feature space
  - Computes values of pairs of data without considering coordinates of that data space (dot product)
- Employ memory-based learning
  - Instead of generalization approach it compares unknown and new instances to training instances stored in memory

1.1.4 RKHS

- Derived from Hilbert Space
  - Vector space that generalizes 2D and 3D objects
  - Vector space with an inner product (f, g) such that the norm is

$$|f| = \sqrt{f, f} \tag{1}$$

turns into a complete metric space

- Establishes a linear relationship in a Hilbert space
- Norm should be as small as possible for it to be linearly functional

1.1.5 Representer Theorem

- Applications
  - Pattern analysis and SVM
- Problem
  - Kernels possess problem of infinite dimensional space that may seem mathematically feasible but not practically viable
    * Especially for training a learning machine dealing with optimization

1.1.6 Without prior assumptions (non-parametric)

- Given a non-empty set X, a positive definite real-valued kernel k on $X \times X$, a training sample

$$(x_1, y_1), \ldots, (x_m, y_m) \in X \times R \tag{2}$$

  a strictly monotonically increasing real-valued function g on $[0, \infty]$, an arbitrary cost function

$$c : (X \times R2)^m \to R \cup \infty \tag{3}$$

  and a class of functions

$$F = f \in R^x | f(\bullet) = (\Sigma)_{i=1}^{\infty} \beta_i k(\bullet, z_i), \beta_i \in R, z_i \in X, ||f|| < \infty \tag{4}$$

- Here is the norm in the RKHS associated with k, i.e. for any $z_i \in X$
- Then any $f \in F$ minimizing the regularized risk functional

$$c((x_1, y_1, f^{'}(x_1), \ldots, (x_m, y_m, f^{'}(x_m)))) + g(||f||) \tag{5}$$

- Admits a representational form

$$f(\bullet) = \sum_{i=1}^{m} (a_i k(\bullet, x_i)^{"}) \tag{6}$$

1.1.7 With partial assumptions (semi-parametric)

- Suppose that in addition to the assumptions of the previous theorem we are given a set of M real-valued functions

$$\psi p M p = 1 \text{ on } X \tag{7}$$

$$\text{with the property that the } m \times M \text{ matrix } (\psi p(xi))ip \text{ has rank } M \tag{8}$$

$$\text{then any } f' := f + h \text{ with } f \in F \text{ and } h \in \text{ span } \psi p \tag{9}$$

- minimizing the regularized risk

$$c((x_1, y_1, f^{'}(x_1), \ldots, (x_m, y_m, f^{'}(x_m)))) + g(||f||) \tag{10}$$

- Admits a representational form

$$f(\bullet) = \sum_{i=1}^{m}(a_i)k(x_i, \bullet) + \sum_{p=1}^{M}(\beta_p \psi_p(\bullet), \tag{11}$$

- with unique coefficients $\beta_p \in R$ for all $p = 1, \ldots, M$

$**$Above theorems minimise factors such as real-valued function g and cost function c. In ML context, these theorems give provisions for kernels in the training data.

1.2 Capsule Network

1.2.1 Advantages

- Require less training
- Affine transformations
- Activation vectors are easy to interpret

1.2.2 Capsule

- Any function that tries to predict the presence and instantiation parameters of a particular object at a given location
- Activation vector
  - Length: estimated probability of presence
  - Orientation: object's estimated pose parameters
- Implementing
  - Squash all vectors length to be between 0 and 1

1.2.3 Equivariance

- Allows image segmentation as opposed to CNN losing data through pooling layers

1.2.4 Every capsule in the first layer tries to predict the output of every capsule in the next layer

- Dot product of transformation matrix with its own activation vector
- First layer learns all part/whole relationships

1.2.5 Routing by agreement

- If lower-order capsules agree on a higher-order capsule then only consider higher-order capsule
- Allows for a cleaner input signal and more accurately determine pose of object
- Easily navigate hierarchy of parts and know which part belongs to which object
- Helps with overlapping objects and crowded scenes

1.2.6 RBA Implementation - Clusters of Agreement

- Set raw weights to 0 for all features
- Apply softmax function to raw weights for each primary capsule
- Compute mean of all predictions
- Measure distance between each predicted vector and mean vector (Scalar product) to see how much agree
- Result is weight of vector
- Then calculate weighted mean
- Reassign weights
- Repeat weighted mean process 3-5 times
- Find weighted sum
- Squash sum

1.3 Faster R-CNN

- Fully convolutional region proposed network to generate object-like regions
- Classifier after RPN to further infer candidate regions

1.4 Convolution layers

- At the convolution layer, the previous layer's feature maps are convolved with learnable kernels
- Trainable bias parameter is added
- Result is processed by the activation function to form the output feature map

1.5 Feature pooling layer

- This layer treats each feature map separately
- In general, this layer is called the subsampling layer
  - Produces down-sampled versions of the input maps
  - This means that the number of input and output maps is the same
  - Output maps are smaller in size
- Results are robust to small variations in the location of features in the previous layer

1.6 Fully connected (FC) layers

- After data processing by several convolutional and subsampling layers
- High-level reasoning in the neural network is performed via FC layers
- Neurons in an FC layer have full connections to all activations in the previous layer
- Their activations can hence be computed with a matrix multiplication followed by a bias offset

2 Research Papers

2.1 Accurate object detection

2.1.1 Neocognitron

- Hierarchical and shift-invariant model for pattern recognition
- Fukushima's method had limited empirical success in part because it lacked a supervised training algorithm
- LeCun and colleagues demonstrated that stochastic gradient descent via backpropagation was effective for training deeper networks for challenging real-world handwritten character recognition problems.

2.1.2 Their method generates around 2000 category-independent region proposals for the input image

- Extracts a fixed-length feature vector from each proposal using a CNN
- Classifies each region with category-specific linear SVMs
- Use a simple warping technique (anisotropic image scaling) to compute a fixed size CNN input from each region proposal, regardless of the region's shape

2.1.3 Object detection system consists of three modules

- The first generates category-independent region proposals These proposals define the set of candidate detections available to our detector
- The second module is a convolutional network that extracts a fixed-length feature vector from each region
- The third module is a set of class-specific linear SVMs.

2.1.4 While R-CNN is agnostic to the particular region proposal method

- Use selective search to enable a controlled comparison with prior detection work

2.1.5 Run selective search on the test image to extract around 2000 region proposals

- Use selective search's "fast mode" in all experiments
- Warp each proposal and forward propagate it through the CNN in order to compute features
- For each class, score each extracted feature vector using the SVM trained for that class
- Given all scored regions in an image, we apply a greedy non-maximum suppression for each class independently
  - Rejects a region if it has an intersection-over-union (IoU) overlap with a higher scoring selected region larger than a learned threshold.

2.1.6 To adapt the CNN to the new task (detection) and the new domain (warped proposal windows)

- Continue stochastic gradient descent (SGD) training of the CNN parameters using only warped region proposals

- First-layer filters can be visualized directly and are easy to understand and they capture oriented edges and opponent colors
- Single out a particular unit (feature) in the network and use it as if it were an object detector in its own right
  - Compute the unit's activations on a large set of held-out region proposals (about 10 million)
  - Sort the proposals from highest to lowest activation
  - Perform non-maximum suppression
  - Display the top-scoring regions

2.1.7 Computing features on CPMC regions (all of which begin by warping the rectangular window around the region to 227 x 227)

- The first strategy (full) ignores the region's shape and computes CNN features directly on the warped window
  - These features ignore the non-rectangular shape of the region
  - Two regions might have very similar bounding boxes while having very little overlap
- The second strategy (fg) computes CNN features only on a region's foreground mask
  - Replace the background with the mean input so that background regions are zero after mean subtraction
- The third strategy (full+fg) simply concatenates the full and fg features

2.2 Optical remote sensing images

2.2.1 Detection of dense objects in optical remote sensing images

2.2.2 Adopt dilated convolutions instead of traditional convolutions to improve precision

- As certain objects in satellite remote sensing images are small and difficult to detect

2.2.3 Adopt a bootstrapping strategy called Online Hard Example Mining for mining hard negative examples, and we add it to Faster RCNN.

- Use a multi-scale representation and its combinations in a new manner
  - Propose a fully convolutional neural network instead of the fully connected layers in the Faster RCNN framework.

- The object detection accuracy and recall show significant improvement with their approach

2.2.4 RCNN

- Combines CNNs and a support vector machine (SVM) as well as bounding boxes to detect objects
- RCNNs can be used to detect objects with high accuracy
- Used unsupervised pre-training followed by supervised fine-tuning
  - When labeled data is scarce

2.3 Large-scale remote sensing images

2.3.1 A unified and self-reinforced CNN R^2-CNN

- Composed of the backbone Tiny-Net, intermediate global attention block, and final classifier and detector
- Enabling the entire network efficient in both computation and memory consumption
- Robust to false positives
- Strong to detect tiny objects

2.3.2 Algorithm

- First, as a unified and self-reinforced framework,
- R^2-CNN first crops large-scale images with a much more smaller scale (such as 640 x 640 pixels) with 20 percent overlap to tackle the oversized input size
  - By processing the patches asynchronously, the limited memory is not a problem anymore
- A convolutional backbone structure is then applied to inputs, which enables powerful features extraction
- Based on the discriminative features, a classifier first predicts the existence of detection target in the current patch
- Detector is followed to locate them accurately if available
- Classifier and detector are mutually reinforced each other under the end-to-end training framework

2.3.3 Self-reinforced architecture

- Since, in large-scale remote sensing images, most crops do not contain valid target so that about 99 percent of the total patches do not need to pass the heavy detector branch
- Light classifier branch can filter out a blank patch without heavier detector cost
- As most false positives commonly occur with massive backgrounds, benefited from the self-reinforced framework, the classifier can identify the difficult situation even when there is only one tiny object in the patch given the fine-grained features from the detector
- The detector receives less false positive candidates since most of them are filtered out by the classifier
- Even if the patches are distinguished incorrectly by a classifier, the detector can still rectify the results later

2.3.4 Inserted an efficient zoom-out and zoom-in architecture in Tiny-Net to enlarge the feature map without margin cost

- Improves the recall of tiny objects obviously
- Position-sensitive RoI pooling is also used to get more spatial information

2.4 Vehicle detection

2.4.1 Cons with Faster R-CNN

- Poor performance for locating small-sized vehicles accurately
- Classifier after RPN cannot distinguish vehicles and complex backgrounds

2.4.2 Hyper Region Proposal Network

- Use pre-trained ZF model based on ImageNet
- Predicts all possible bounding boxes of vehicle-like objects with high recall rate with a combination of hierarchical feature maps
- Replace classifier after RPN by cascade of boosted classifiers to verify candidate regions

2.4.3 HRPN Layers

- First convolutional layer takes training images as input and has 96 kernels (7 x 7 x 34)
- Second convolutional layer output of previous and filters it with a stride of 2 pixels by 256 kernels (5 x 5 x 96)
- Third, Fourth, and Fifth convolutional layers are directly connected to each other with 384 kernels (3 x 3 x 384) and the last of 256 kernels (3 x 3 x 256)

2.4.4 Algorithm

- Crop large-scale images into blocks and rotate blocks every 90 degrees
- Send blocks to HRPN
- Generate candidate regions
  - Use sliding window operation on hyper feature maps
  - Parameters of weight are set by Gaussian and parameters of bias are set by constant
  - Extract 256-d feature vector for each 256 region proposal
- If a predicted region has Intersection-over-Union bigger than 0.7 with ground truth, assign a positive label, else if less than 0.1 we assign negative, else we discard
  - All positive and negative region proposals are fed to loss function

2.4.5 Detection Task Related Work

- Generation of candidate regions
- Sliding-window search algorithm
- Region-proposal methods: merge segments that are likely to include objects

2.4.6 Feature extraction

- Haar-like features
- Local binary patterns
- Scale-invariant feature transform descriptors

2.4.7 Classification

- SVM
- AdaBoost

2.5 Optic Disc R-CNN

2.5.1 Regional CNN using an object detection based method

- Region proposal network
- Feature maps from feature extraction are fed into RPN
- Proposes score to indicate probability contains optic disc/cup
- 9 anchors are generated (3 scales x 3 aspect ratios) from sliding windows

2.5.2 Region of interest pooling

- Crops small regions of feature maps according to coordinates of candidate bounding boxes
- Max pooling is applied to blocks of size k x k

2.5.3 Classifier

- Deep convolutional layers which generate encoded forms of bounding box's coordinates

2.5.4 End-to-end deep learning framework where feature maps are shared for segmentation and attention mechanisms

- Feature Extraction is made up of deep convolutional layers to extract feature representations for original images
- Introduce atrous convolution to extract more dense features to improve bounding box accuracy

2.5.5 Object detection

- Finds minimal bounding boxes of ellipses
- Ignores rotational angle

2.6 Emotion classification

2.6.1 Current classification only focuses on whole level image ignoring sentimental response of multi-level visual features from local regions which contribute to diverse emotion reactions

- Feature pyramid network to extract multi-scale deep feature maps related to image emotion
- Extracted from different convolutional layers combine high-level semantic features with low-level deep features
- Consists of:
  - Bottom-up pathway: feed-forward computation of normal backbone convolutional network
  - Top-down pathway: Combine different levels of feature maps extracted from bottom-up pathway
  - Lateral connections between both

2.6.2 Region-based CNN that can effectively extract local emotional info from emotional regions of image

- Ignores noisy info generating non-emotional regions
- Faster R-CNN: Extracts emotional region from image
  - Two-stage detector mainly consisting of three major parts
    - Shared bottom convolutional layers (FPN)
    - Region proposal network
    - Classifier built for region-of-interest

2.6.3 Algorithm

- Set of local deep representations of emotional regions is collected
- Global deep representation of whole image is concatenated with local deep representations
- Followed by a softmax layer transformed into a probability distribution of different emotions
- Considers emotion class probability
  - Emotions are subjective and one class may not be confident