Variational Feature Pyramid Networks

Panagiotis Dimitrakopoulos [1] Giorgos Sfikas [1,2] Christophoros Nikou [1]

[1] Dept. of Computer Science & Engineering, University of Ioannina, Greece

[2] School of Electrical and Computer Engineering National Technical University of Athens, Greece

June 22, 2022

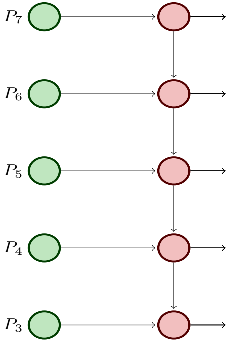
Introduction

- Recent architectures for object detection adopt a feature pyramid network as a backbone for deep feature extraction:
- In this work, we opt to learn a dataset-specific architecture for efficient feature pyramid networks
- Starting by a complex network, we adopt variational inference to prune redundant connections.

Feature Pyramid Networks

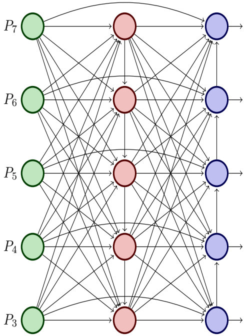
- Feature pyramid networks (FPNs) were designed as a solution for detecting the objects of an image at different scales

    - The bottom-up pathway (green nodes) is the feed-forward computation of the backbone CNN,
    - A building block is responsible for constructing the top-down feature maps (red nodes)
    - Recent works propose more sophisticated modules and architectures.



Proposed Pyramid Network

- Initial architecture of our network:
  Bottom-up pathway, a hidden and an output layer.
  - Each intermediate block:
    - Input features $F_{level}^{layer} = \{F_1, \ldots, F_N\}$
    - Input Connections weighted by:
      $W_{level}^{layer} = \{w_1, \ldots, w_N\}$
    - Output: $F_{out} = \text{Conv}(\frac{\sum_{i=1}^{N} w_i F_i}{\sum_{i=1}^{N} w_i + \epsilon})$



Variational Inference

- In our method, we treat each weight $\mathbf{w}$ associated with each connection on the network as a stochastic variable coming from a parametric distribution $p(\mathbf{W})$.

- The goal is to find an approximation for the posterior $p(\mathbf{W}|\mathbf{D})$

- Using Variational Inference and the SGVB method the loss becomes:

$$\tilde{\mathcal{L}}(\mathbf{W}) = \frac{1}{L}\sum_{i=1} \log p(\mathbf{Y}|\mathbf{X}, \mathbf{W} = f(\mathbf{w}, \epsilon)) - KL(q_\phi(\mathbf{W})||p(\mathbf{W})). \tag{1}$$

Choice of Prior Distribution (1)

- The mechanism of Automatic Relevance Determination using factorized Gaussians

  - Prior Distribution

    $$p(\mathbf{W}) = \prod_i p(\mathbf{w}_i) \text{ where } \mathbf{w}_i \sim \mathcal{N}(0, \hat{\sigma}_i^2)$$

  - Approximate Posterior Distribution

    $$q(\mathbf{W}) = \prod_i q(\mathbf{w}_i) \text{ where } \mathbf{w}_i \sim \mathcal{N}(\mu_i, \sigma_i^2)$$

- The optimal hyperparameter $\hat{\sigma}$ of the prior distribution can be calculated

$$\frac{\partial \tilde{\mathcal{L}}(\mathbf{W})}{\partial \hat{\sigma}_i^2} = 0 \text{ which yields } \hat{\sigma}_i^2 = \mu_i^2 + \sigma_i^2$$

Choice of Prior Distribution (2)

- We extended the mechanism of Automatic Relevance Determination (ARD) in order to study the correlation between the connection weights

- Prior Distribution

$$p(\mathbf{W}) = \mathcal{N}(\mathbf{w}|0, \hat{\Sigma}),$$

- Approximate Posterior Distribution

$$q(\mathbf{W}) = \mathcal{N}(\mathbf{w}|\mu, \Sigma),$$

where $\Sigma = LL^T$ (Cholesky decomposition)

- The optimal hyperparameter $\hat{\Sigma}$ can be calculated directly by optimizing the VLB Empirical Bayes

$$\frac{\partial \tilde{\mathcal{L}}(\mathbf{W})}{\partial \hat{\Sigma}} = 0 \text{ which yields } \hat{\Sigma} = \mu\mu^T + \Sigma$$

Evaluating model's Performance

- Numerical results for object segmentation trials on COCO

Network	Model	AP	Params	Inference
Mask RCNN	BiFPN	0.271	1.60M	7.8 ± 0.01
	PANet	0.268	1.74M	6.7 ± 0.01
	NAS-FPN	0.280	1.53M	5.4 ± 0.10
	PConv	0.279	$\mathbf{1.25}$M	8.4 ± 0.77
	HRNet	0.288	1.32M	$\mathbf{3.2 \pm 0.17}$
	ARD	0.290	1.67M	6.5 ± 0.01
	FullARD	$\mathbf{0.299}$	1.74M	6.8 ± 0.02

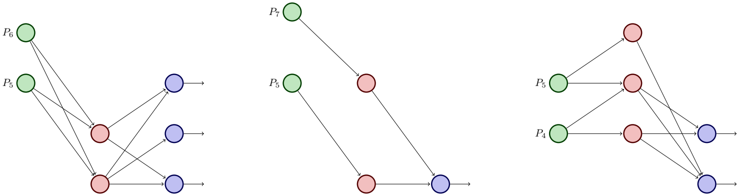
Evaluating Probabilistic Pruning

- Numerical results for instance segmentation trials on COCO

Model	AP	Cons	Inference	Params
No Pruning	0.299	63	14.2 ± 0.1	1.74
Rand. Pruning	0.222	16	8.1 ± 0.04	1.60
Lasso-based	0.283	9	$\mathbf{4.8 \pm 0.02}$	**1.32**
Molchanov	0.286	9	6.1 ± 0.03	1.38
Frankle	0.280	9	7.1 ± 0.02	1.40
ARD	0.290	**9**	6.5 ± 0.01	1.39
FullARD	**0.299**	16	6.8 ± 0.02	1.60

Evaluating Model's Architecture

- Different resulting architectures for the trained model, combined with the proposed FullARD prior on Faster RCNN on three different datasets

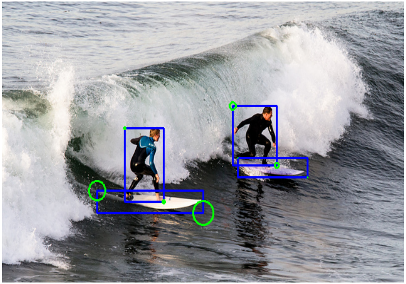


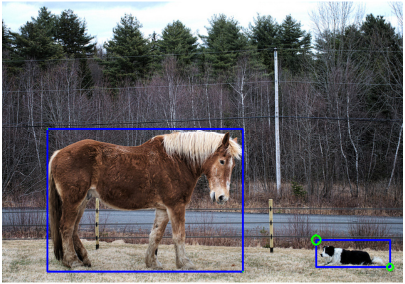
"COCO" "Cards" "Plants"

Evaluating Model's Uncertainty

- By sampling several $\mathbf{w} \sim q(\mathbf{w}|D)$ we can ensemble the resulting architectures and acquire uncertainty estimates.

- Quantitative evaluation of uncertainty estimates for Faster RCNN trained on COCO.





Thank You!