Partial Least Squares: A Deep Space Odyssey

Artur Jordão
William Robson Schwartz (Advisor)

UFMG

CNPq
Conselho Nacional de Desenvolvimento Científico e Tecnológico

CAPES

senselab

DEEPEyes

Introduction

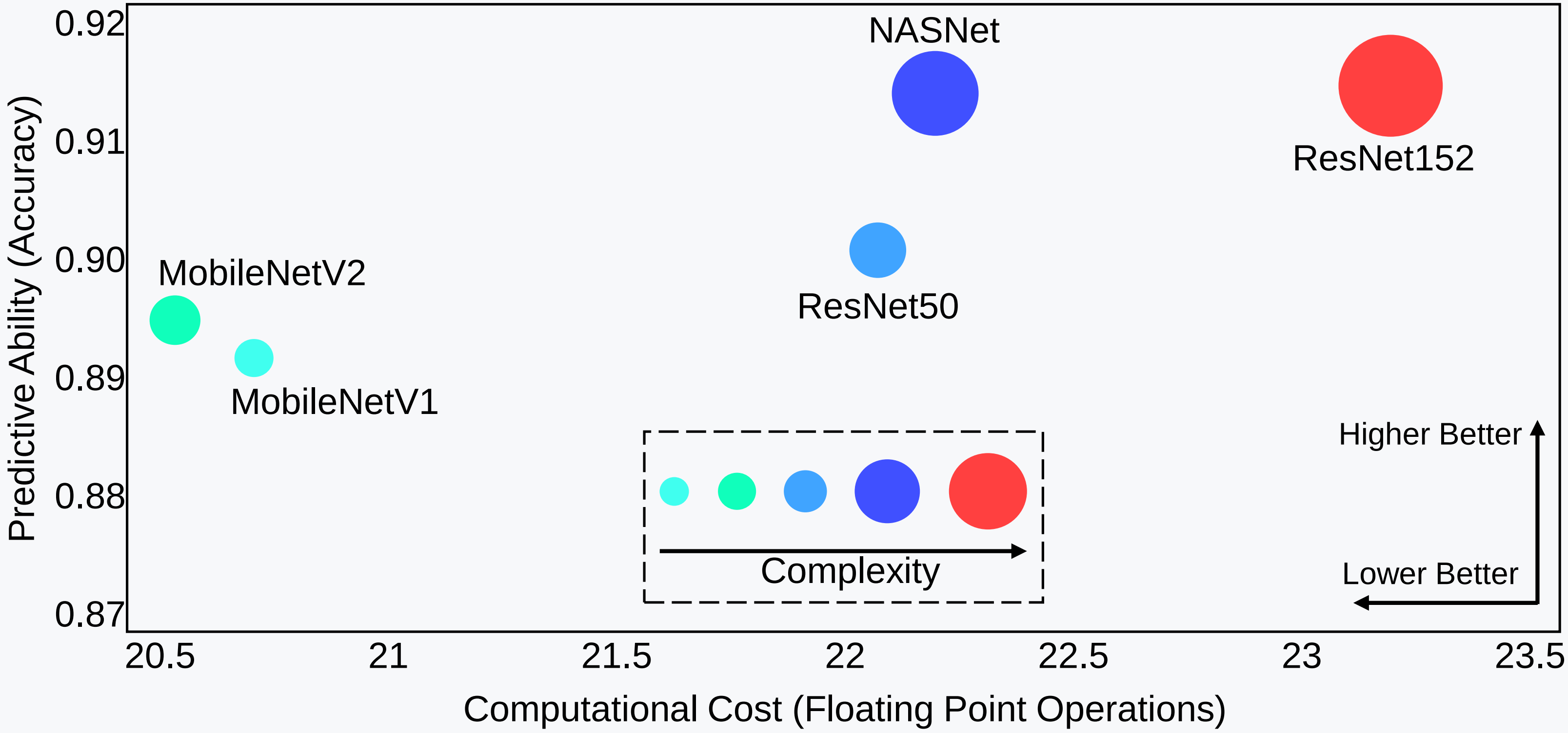
- Pattern recognition plays an important role in cognitive and decision-making tasks

- Pattern recognition methods have led to a series of breakthroughs
  - Often surpassing human performance [Deng et al., 2009; Badia et al., 2020]

Deng et al. (2009). *ImageNet: A Large-Scale Hierarchical Image Database*. In CVPR.
Badia et al. (2020). *Agent57: Outperforming the atari human benchmark*. In ICML.

Convolutional Networks

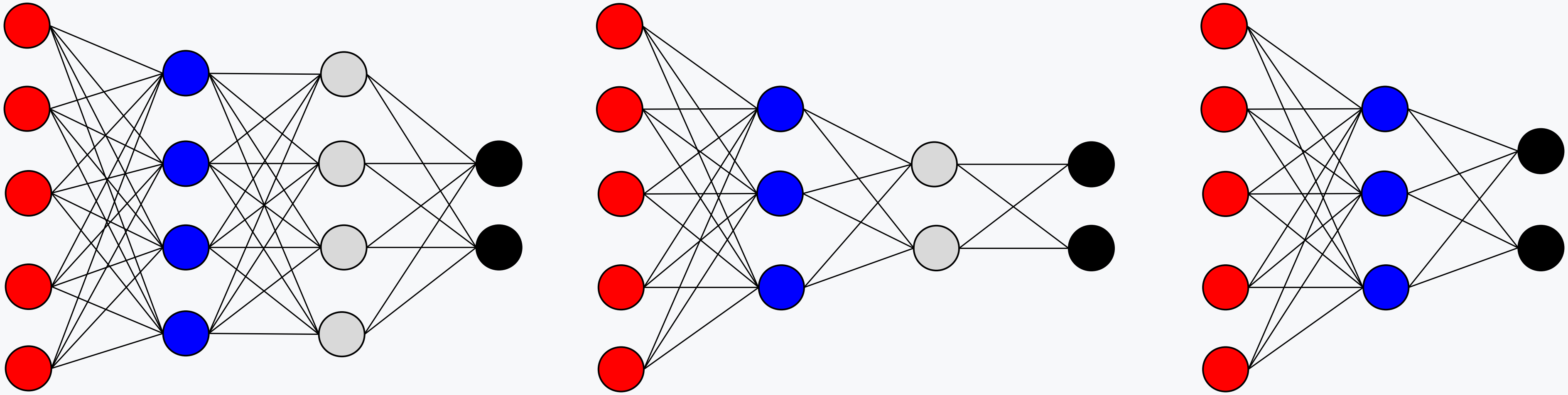
- Visual pattern recognition models
  - Convolutional networks
  - Large architectures (large circles) lead to better results



Convolutional Networks

- Pruning approaches
  - Locate and remove structures (i.e., filters or layers) from the architecture

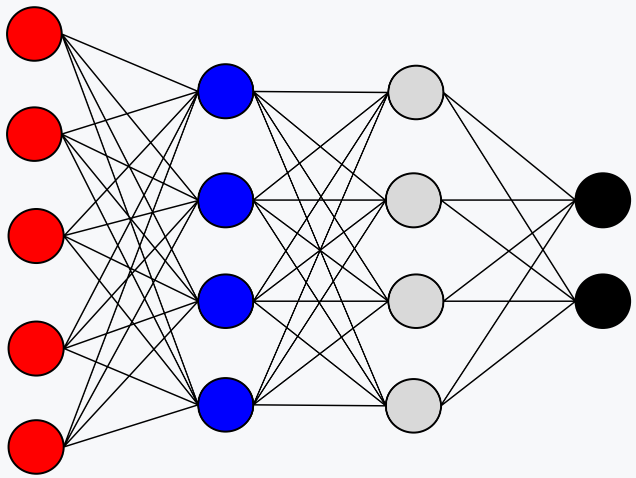
- Existing criteria for pruning convolutional networks are ineffective since the accuracy of the original (unpruned) network is degraded



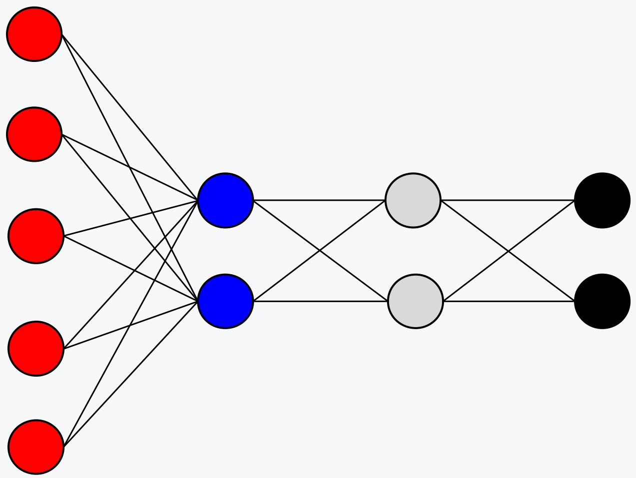
Original, unpruned, Network   Pruning Neurons   Pruning Layers

Convolutional Networks

- Neural Architecture Search (NAS)
  - Automatically design efficient and accurate
- Current strategies analyze a large set of possible candidate architectures
  - Require vast computational resources and take many days to process



Human-designed

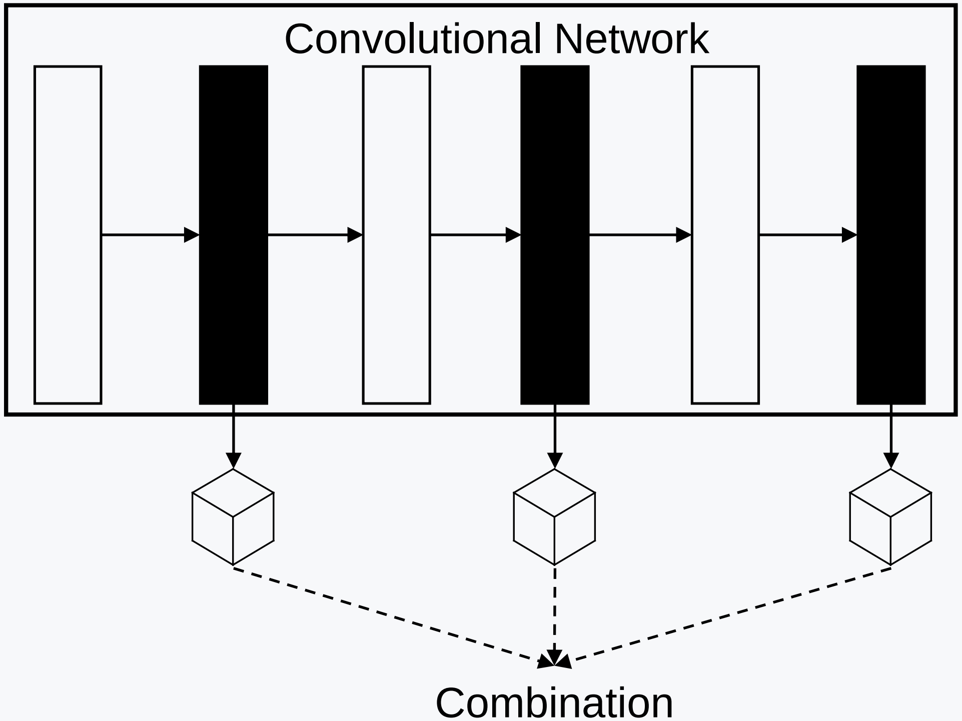


Automatically designed

Convolutional Networks

- HyperNets approaches
  - Explore early and deep layers to improve data representation

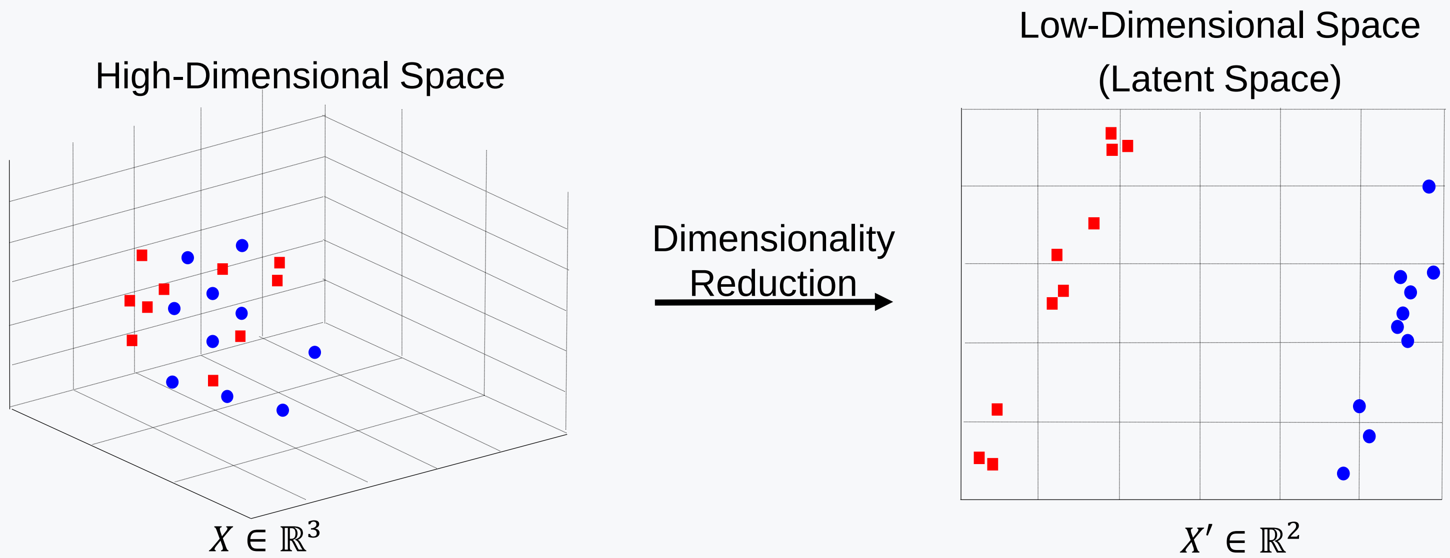
- HyperNets approaches insert time-consuming operations



Dimensionality Reduction

- Dimensionality reduction is able to yield discriminative representations besides reducing computational cost

- Partial Least Squares (PLS) has presented remarkable results
  - Discriminative
  - Robust to sample size problem (singularity)
  - Operate as a feature selection method



Dimensionality Reduction

- Unfeasible for large datasets (e.g., ImageNet) since all the data need to be available in advance
  - Memory constraints

- Incremental dimensionality reduction methods
  - Find the latent space using a single data sample at a time
  - Keep some properties of the traditional dimensionality reduction methods

- Most incremental Partial Least Squares are computationally inefficient and do not preserve all the properties of PLS

Hypotheses

Hypothesis

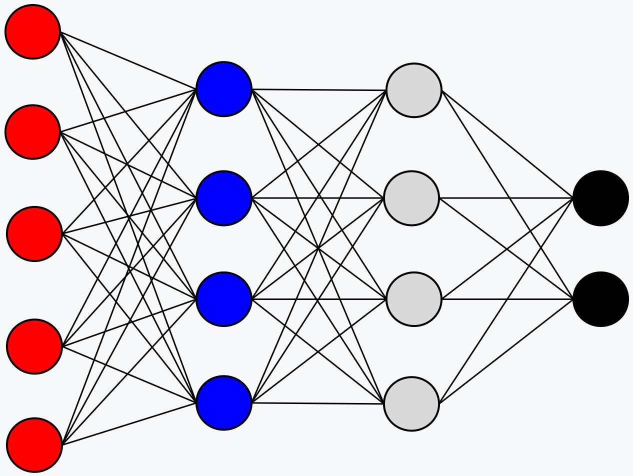
- The importance of a structure composing the convolutional architecture can be effectively estimated using **Partial Least Squares**

- Our central hypothesis is that Partial Least Squares learns the importance inherent to predictive ability of the network

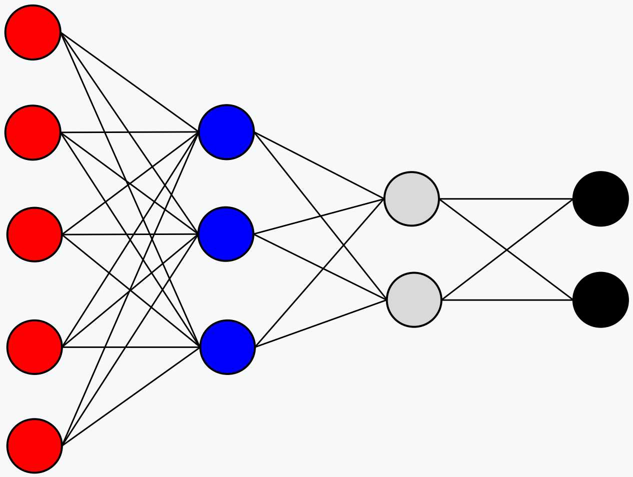
Hypothesis

- The importance of a structure composing the convolutional architecture can be effectively estimated using **Partial Least Squares**

- We can remove neurons and layers from convolutional networks to decrease the computational cost



Original, unpruned, Network

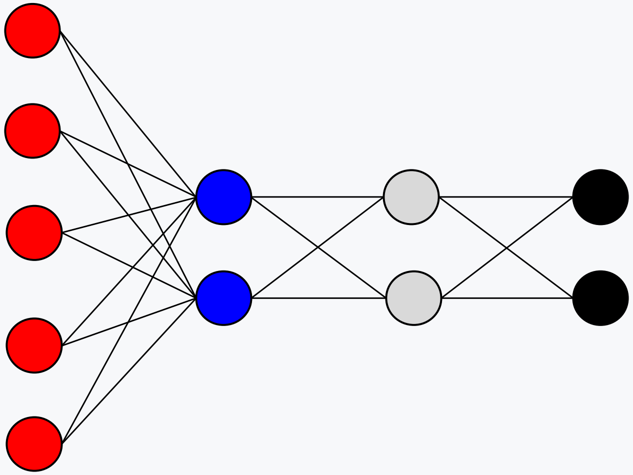


Network after Pruning

Hypothesis

- The importance of a structure composing the convolutional architecture can be effectively estimated using **Partial Least Squares**

- We can insert structures to automatically design high-performance architectures

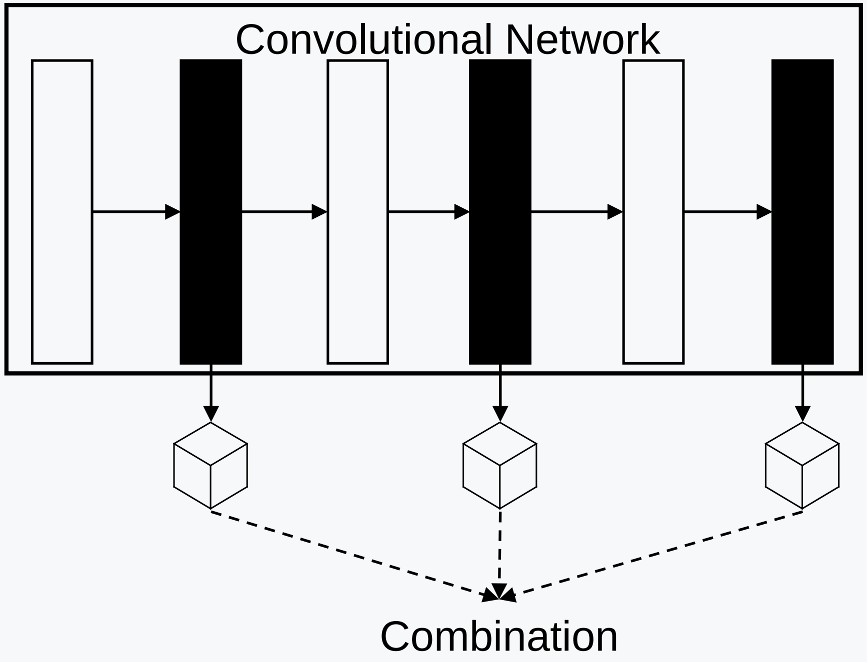


Automatically designed

Hypothesis

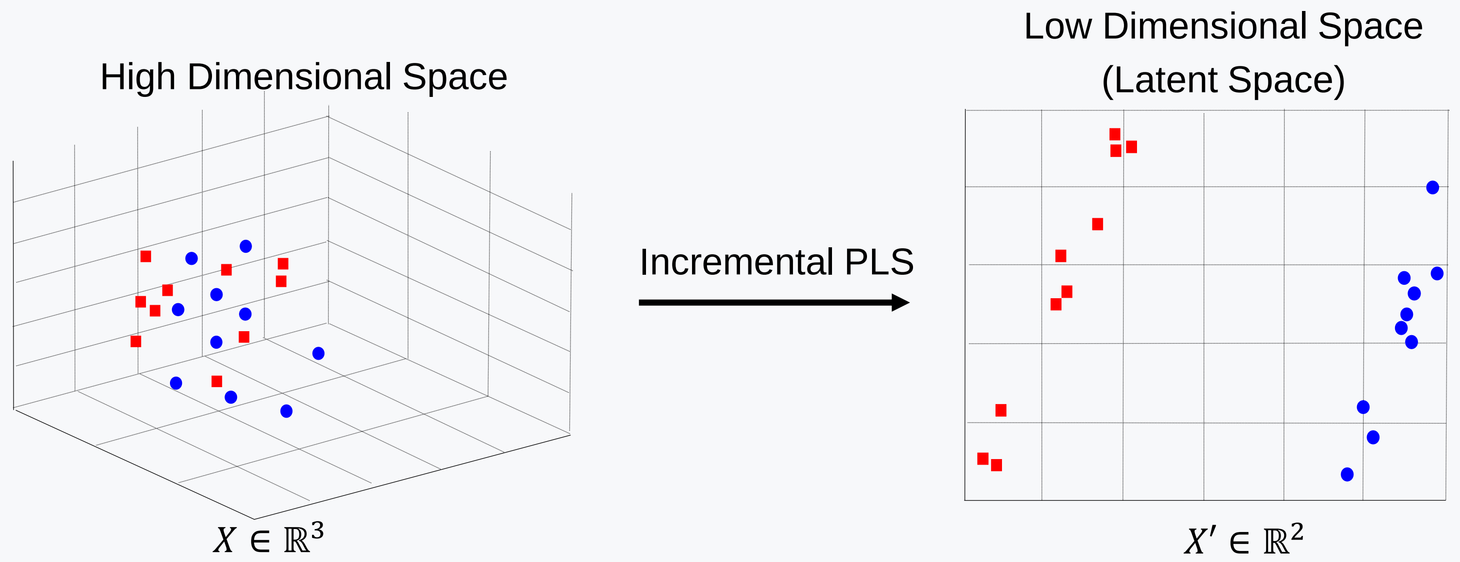
- The importance of a structure composing the convolutional architecture can be effectively estimated using **Partial Least Squares**

- We can combine multiple levels of representation to improve data representation



Hypothesis

- It is possible to compute all components of PLS incrementally using simple algebraic decomposition
  - Low time complexity
  - Preserves the proprieties of PLS across all components



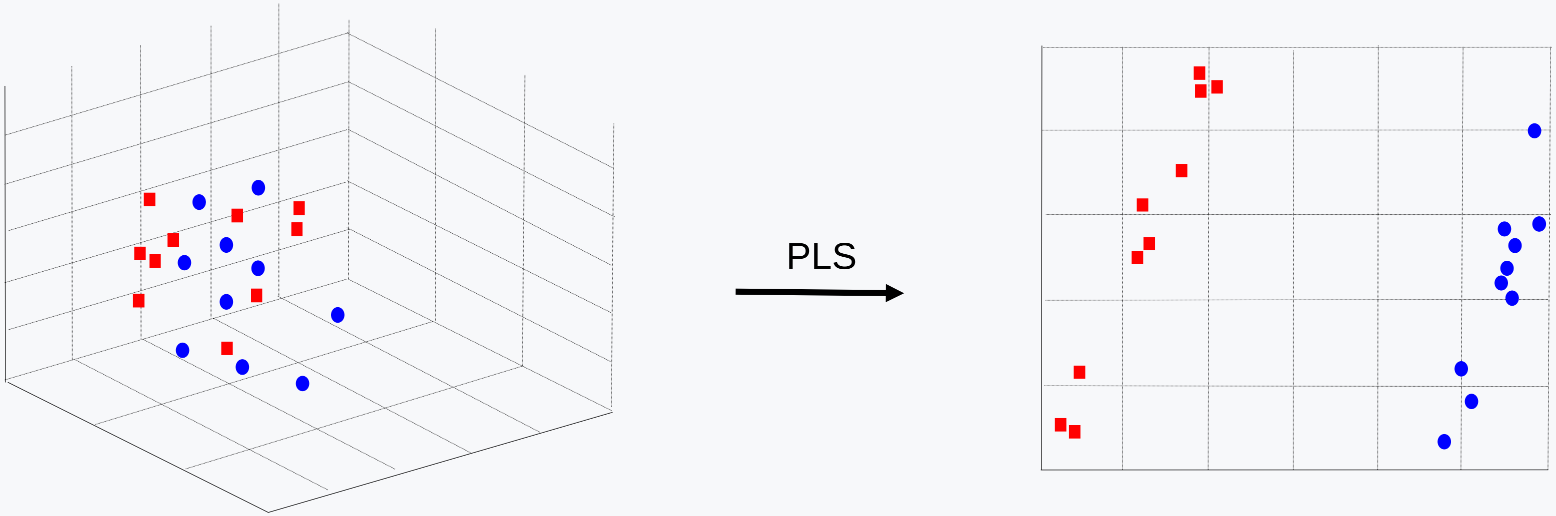
Summary

- Theoretical Concepts
- Pruning Approaches
  - Pruning Filters
  - Pruning Layers
- Neural Architecture Search
- HyperNet
- Incremental Partial Least Squares

Theoretical Concepts

Partial Least Squares

- Find a projection matrix $\mathrm{W}(\mathrm{w}_1, \mathrm{w}_2, \ldots \mathrm{w}_\mathrm{c})$ that projects the high dimensional ($\mathbb{R}^\mathrm{m}$) space onto a low *c*-dimensional space ($\mathbb{R}^\mathrm{c}$ latent space)
  - $\mathrm{c} \ll \mathrm{m}$

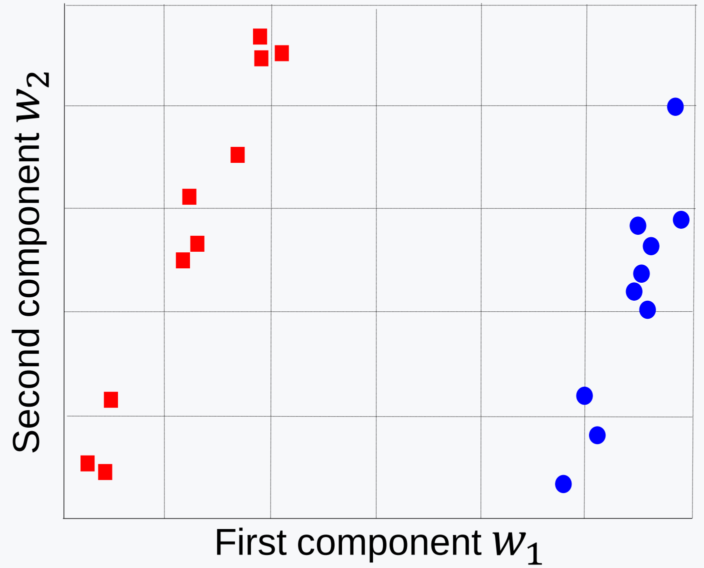


$X \in \mathbb{R}^\mathrm{m}$

$X' \in \mathbb{R}^c$

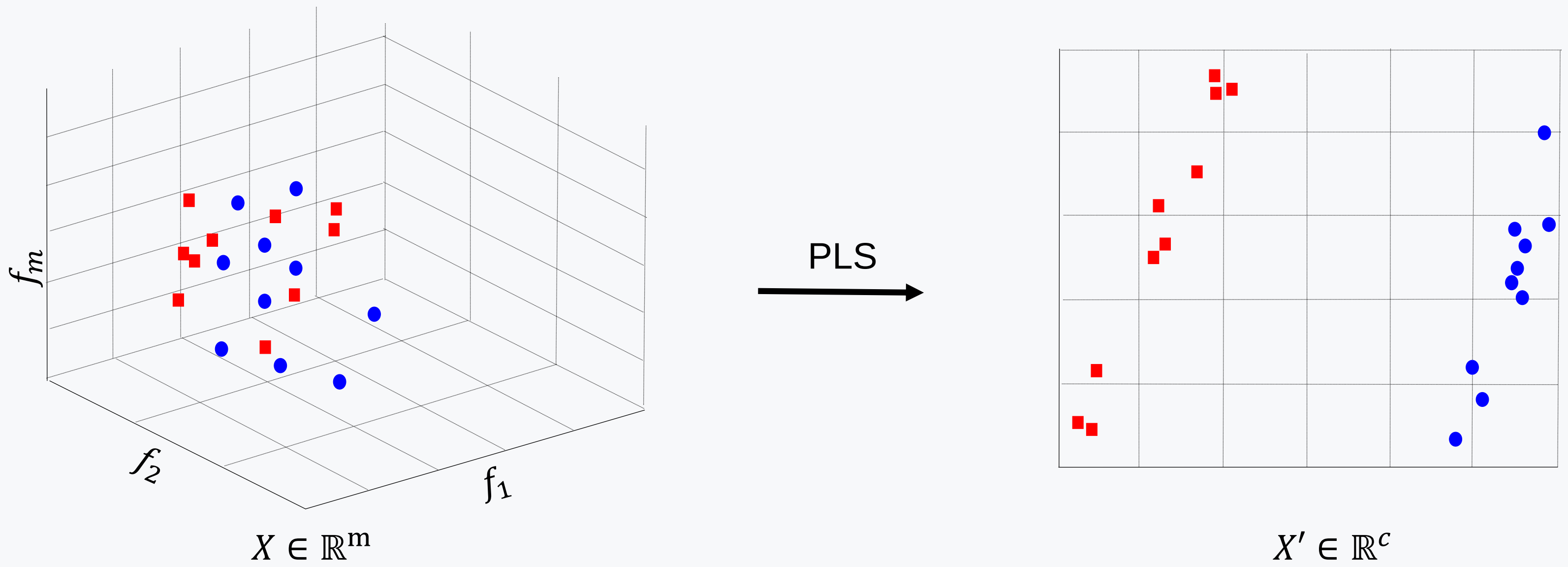
Partial Least Squares

- Compute the component w_i in terms of
  - $maximize(\ COV(Xw, Y)\ ) =\ X^T Y \Rightarrow w_i = X^T Y$



Variable Importance in Projection (VIP)

- VIP estimates the importance of each feature $f_i \in \mathbb{R}^m$
  - PLS as a feature selection method



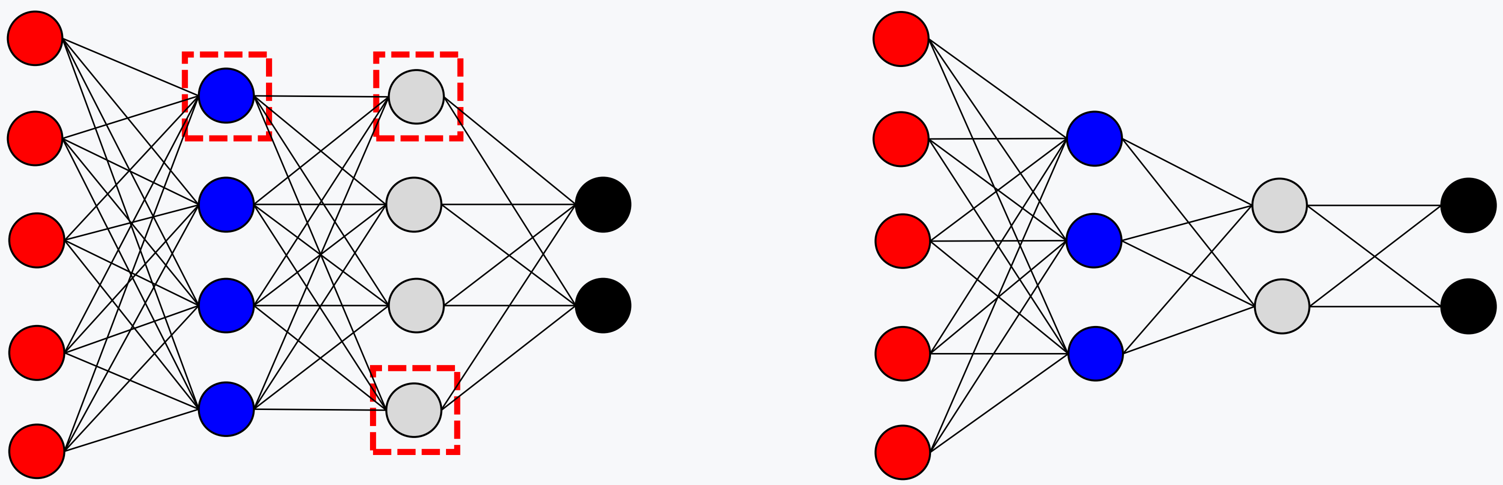
Summary

- Theoretical Concepts
- Pruning Approaches
  - **Pruning Filters**
  - Pruning Layers
- Neural Architecture Search
- HyperNet
- Incremental Partial Least Squares

Pruning Filters

Problem Definition

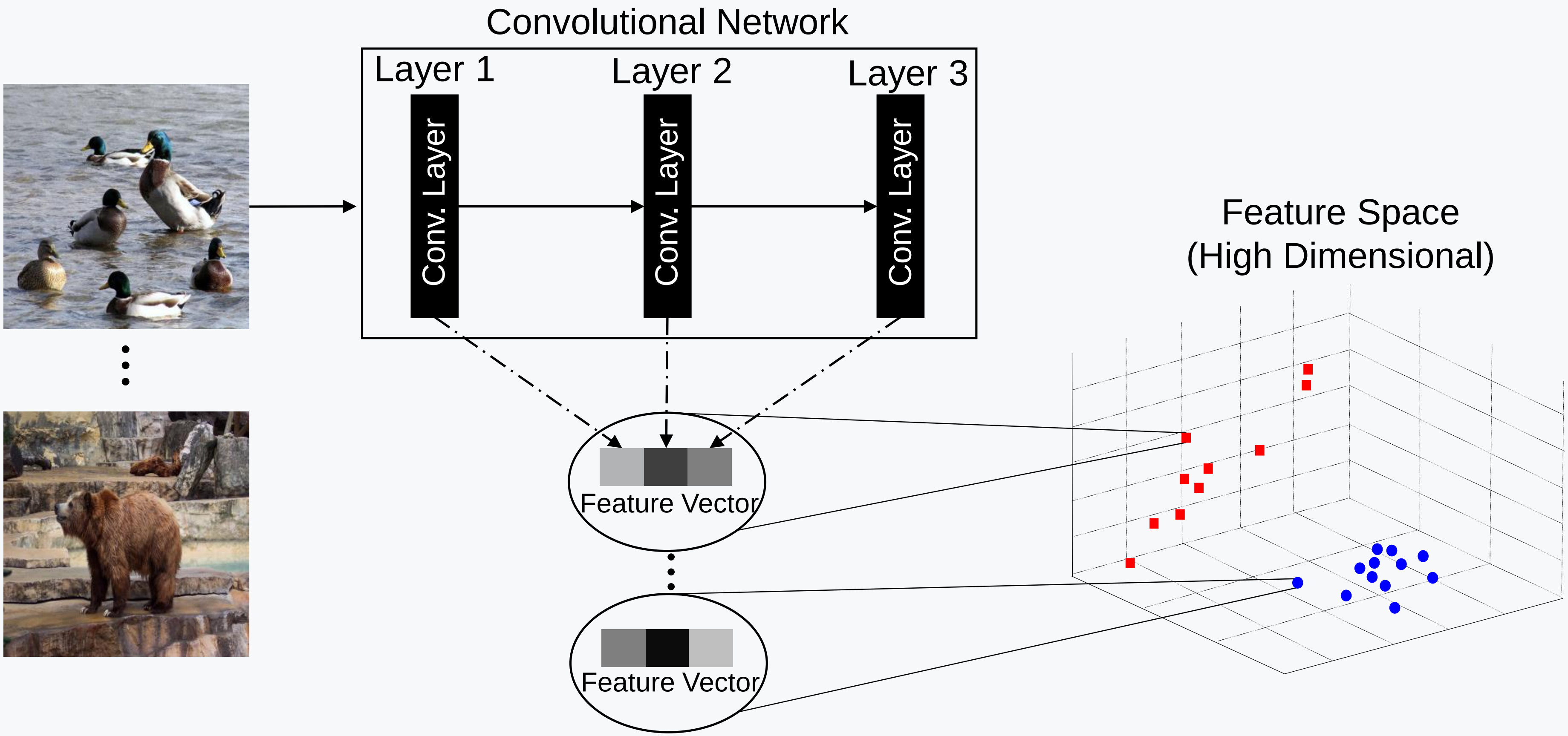
- Identify and remove (red dashed squares) neurons that preserve as much accuracy as possible



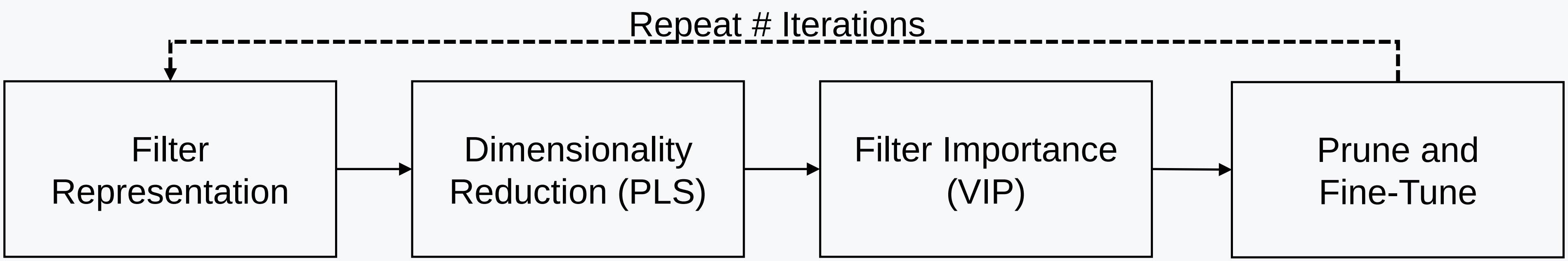
Original, unpruned, Network

Network after Pruning

Pruning Filters



Overview


Repeat # Iterations
Filter Representation
Dimensionality Reduction (PLS)
Filter Importance (VIP)
Prune and Fine-Tune

Experiments
Pruning Filters

Applications and Datasets

- Activity Recognition
  - 5 - 21 classes
  - Cross-validation
- Face Verification
  - Two classes
  - Cross-validation
- Image Classification
  - 10 - 1,000 classes
  - Hold-out


Labeled Faces in the Wild (LFW)

ImageNet

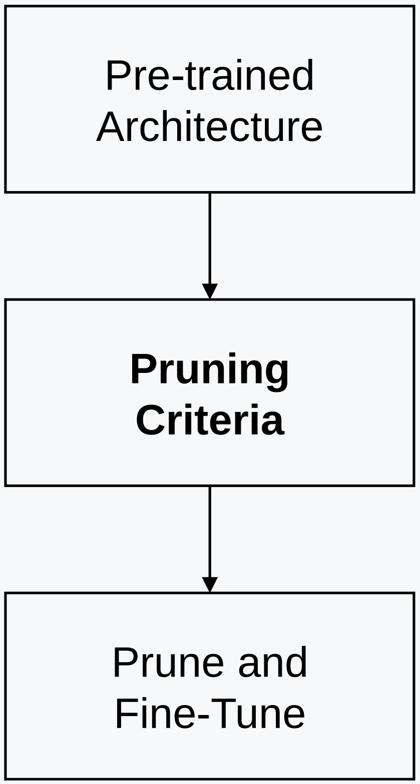
Experimental Setup

- Parameter Assessment
  - Validation set
- Convolutional Networks
  - VGG16
  - ResNets
- Computational Cost
  - Number of Floating Point Operations (FLOPs)
- Statistical Test
  - Paired t-test using 95% confidence

Comparison with other Pruning Criteria

- Pruning criteria
  - ℓ_1-Norm
  - KL [Luo and Wu 2020]
  - HRANK [Lin et al. 2020]
  - ABS [Tan and Montani 2020]

- Feature selection techniques
  - infFS [Roffo et al. 2015]
  - ilFS [Roffo et al. 2017]
  - infFSU [Roffo et al. 2020]



Roffo et al. (2015). Infinite feature selection. In ICCV.
Roffo et al. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In ICCV.
Roffo et al. (2020). Infinite feature selection: a graph-based feature filtering approach. In PAMI.
Luo and Wu (2020). *Neural network pruning with residual-connections and limited-data.* In CVPR.
Lin et al. (2020). *Hrank: Filter pruning using high-rank feature map*. In CVPR.
Tan and Montani (2020). *Dropnet: Reducing neural network complexity via iterative pruning*. In ICML.

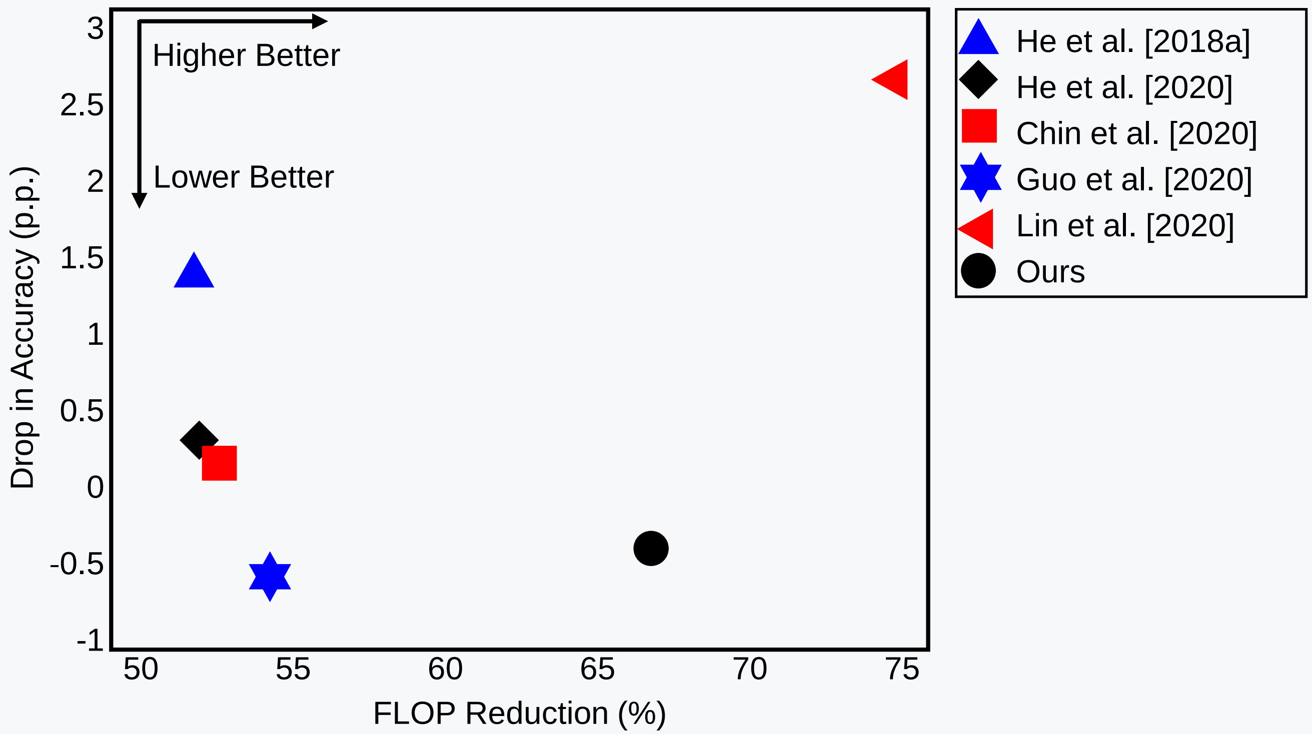
Comparison with other Pruning Criteria

- VGG16

Filter Importance Criteria	CIFAR-10 Acc. Drop↓	ImageNet (32x32) Acc. Drop↓	ImageNet (224x224) Acc. Drop↓
ℓ_1-Norm	-0.69	6.22	**-0.62**
infFS	-0.69	6.31	-0.50
ilFS	-0.65	6.04	-0.36
infFSU	0.48	6.30	-0.33
KL	-0.59	6.37	-0.41
HRANK	-0.84	6.70	-0.47
ABS	-0.62	6.58	-0.42
PLS+VIP	**-0.89**	**5.81**	-0.58

Comparison with other Pruning Approaches

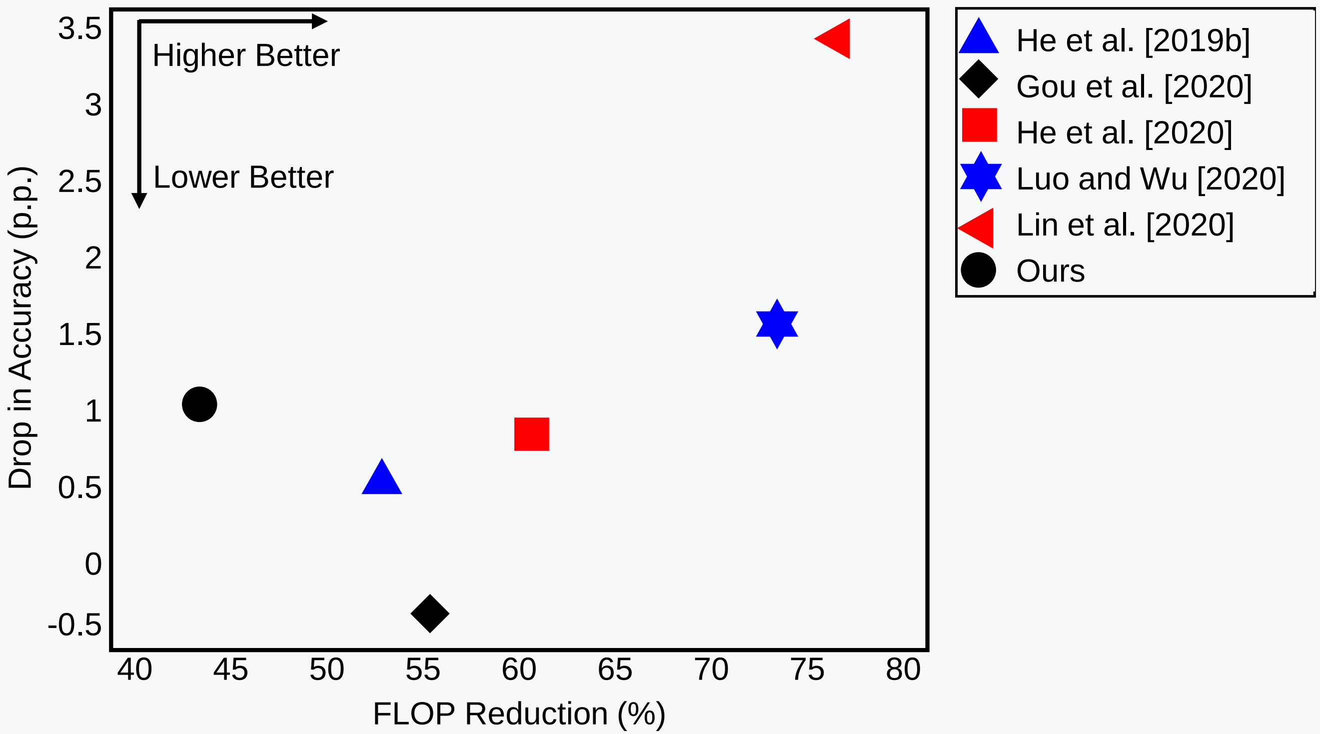
- ResNet56 on CIFAR-10



He et al. (2018a). *Soft filter pruning for accelerating deep convolutional neural networks*. In CVPR.
He et al. (2020). *Learning filter pruning criteria for deep convolutional neural networks acceleration*. In CVPR
Chin et al. (2020). *Towards efficient model compression via learned global ranking*. In CVPR
Guo et al (2020). *A unified framework for model compression*. In CVPR.
Lin et al. (2020). *Hrank: Filter pruning using high-rank feature map*. In CVPR.

Comparison with other Pruning Approaches

- ResNet50 on ImageNet (224x224)



He et al. (2019b). *Filter pruning via geometric median for deep convolutional neural networks acceleration*. In CVPR.
Guo et al (2020). *A unified framework for model compression*. In CVPR.
He et al. (2020). *Learning filter pruning criteria for deep convolutional neural networks acceleration*. In CVPR
Luo and Wu (2020). *Neural network pruning with residual-connections and limited-data*. In CVPR.
Lin et al. (2020). *Hrank: Filter pruning using high-rank feature map*. In CVPR.

Conclusions

- We demonstrate that is possible to remove unimportant, or least important, filters by estimating their importance using PLS

- Compared to existing criteria for determining filter importance, PLS achieves the lowest drop in accuracy

- Compared to state-of-the-art pruning approaches, our strategy for removing filters achieves one of the best trade-offs between FLOP reduction and accuracy drop

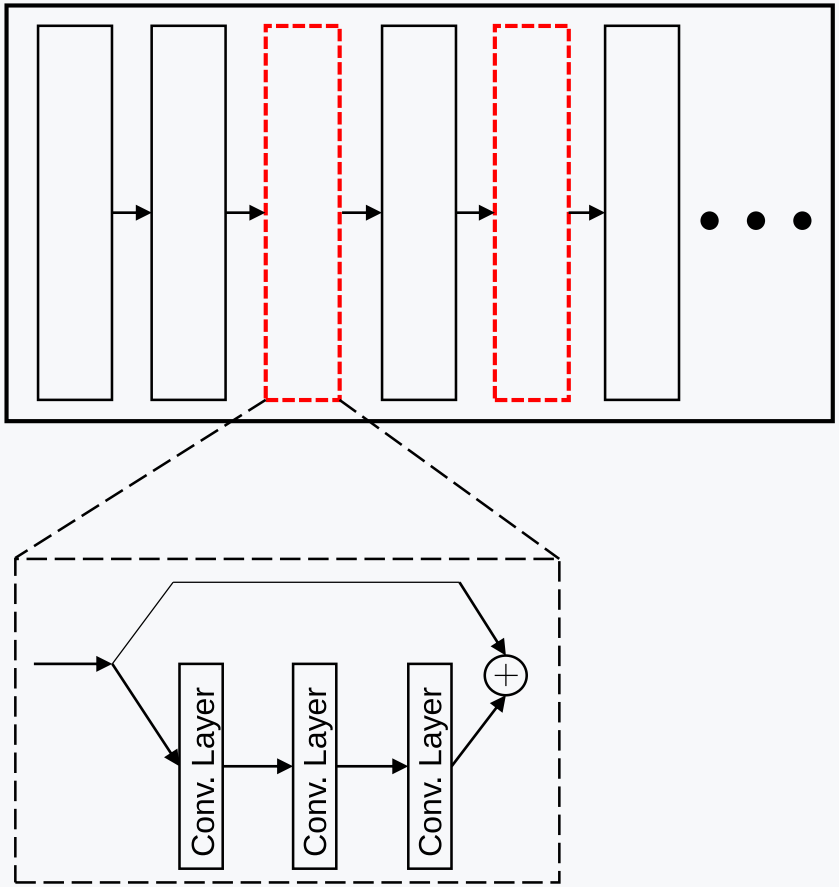
Summary

- Theoretical Concepts
- Pruning Approaches
  - Pruning Filters
  - **Pruning Layers**
- Neural Architecture Search
- HyperNet
- Incremental Partial Least Squares

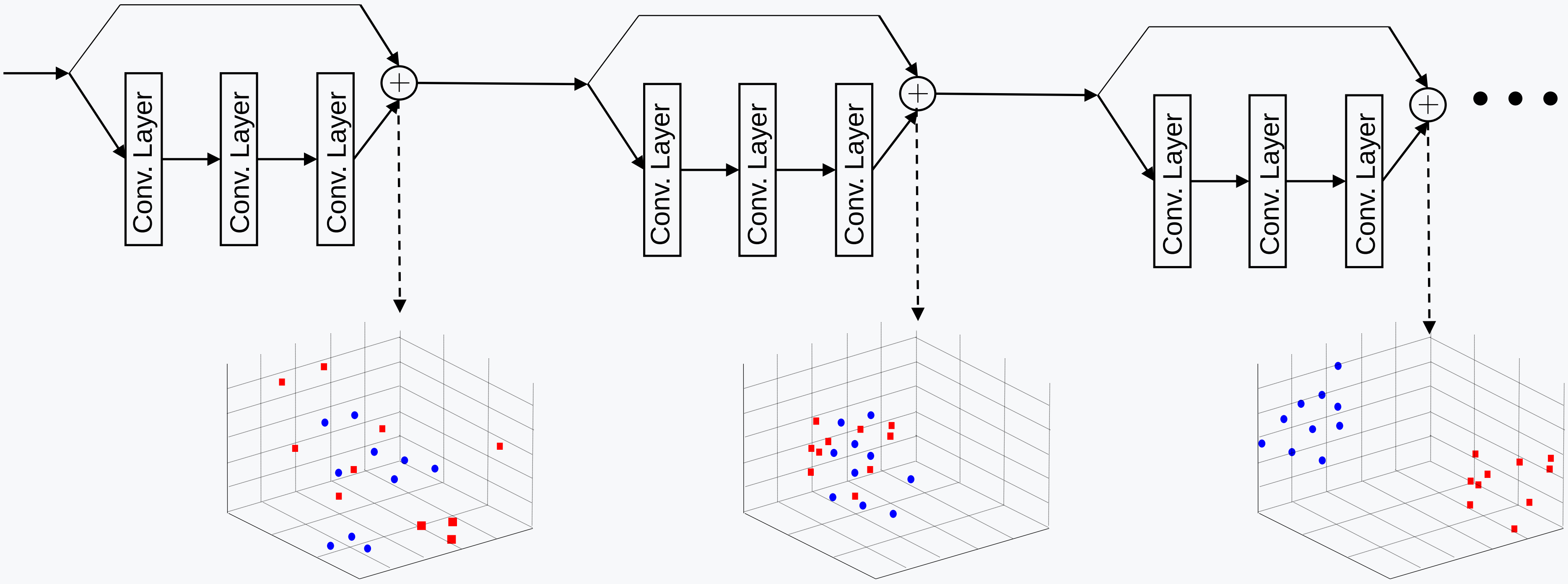
Pruning Layers

Problem Definition

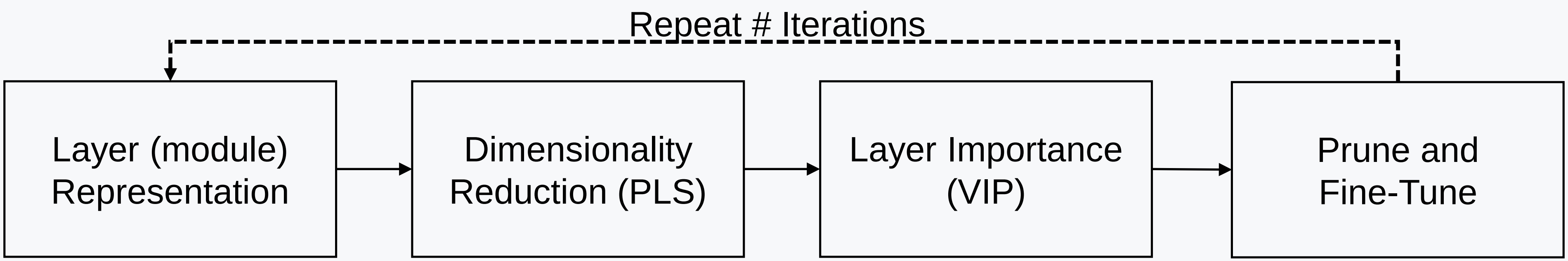
- Identify and remove (red dashed rectangles) layers that preserve as much accuracy as possible



Pruning Layer



Overview

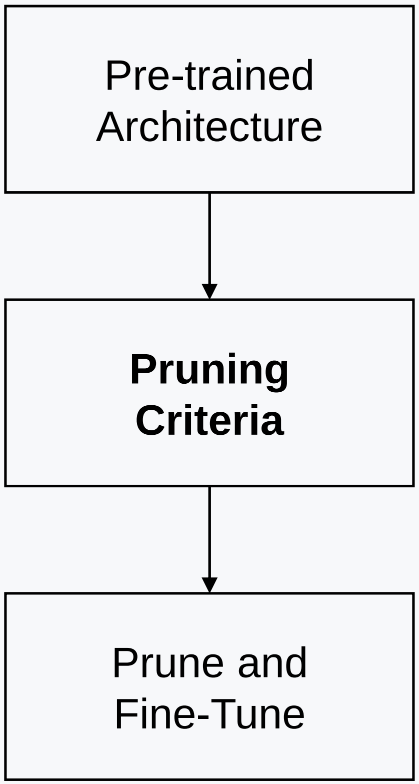

Repeat # Iterations
Layer (module) Representation
Dimensionality Reduction (PLS)
Layer Importance (VIP)
Prune and Fine-Tune

Experiments
Pruning Layers

Comparison with other Pruning Criteria

- Pruning criteria
  - KL [Luo and Wu 2020]
  - HRANK [Lin et al. 2020]
  - ABS [Tan and Montani 2020]

- Feature selection techniques
  - infFS [Roffo et al. 2015]
  - ilFS [Roffo et al. 2017]
  - infFSU [Roffo et al. 2020]



Roffo et al. (2015). Infinite feature selection. In ICCV.
Roffo et al. (2017). Infinite latent feature selection: A probabilistic latent graph-based ranking approach. In ICCV.
Roffo et al. (2020). Infinite feature selection: a graph-based feature filtering approach. In PAMI.
Luo and Wu (2020). *Neural network pruning with residual-connections and limited-data.* In CVPR.
Lin et al. (2020). *Hrank: Filter pruning using high-rank feature map*. In CVPR.
Tan and Montani (2020). *Dropnet: Reducing neural network complexity via iterative pruning*. In ICML.

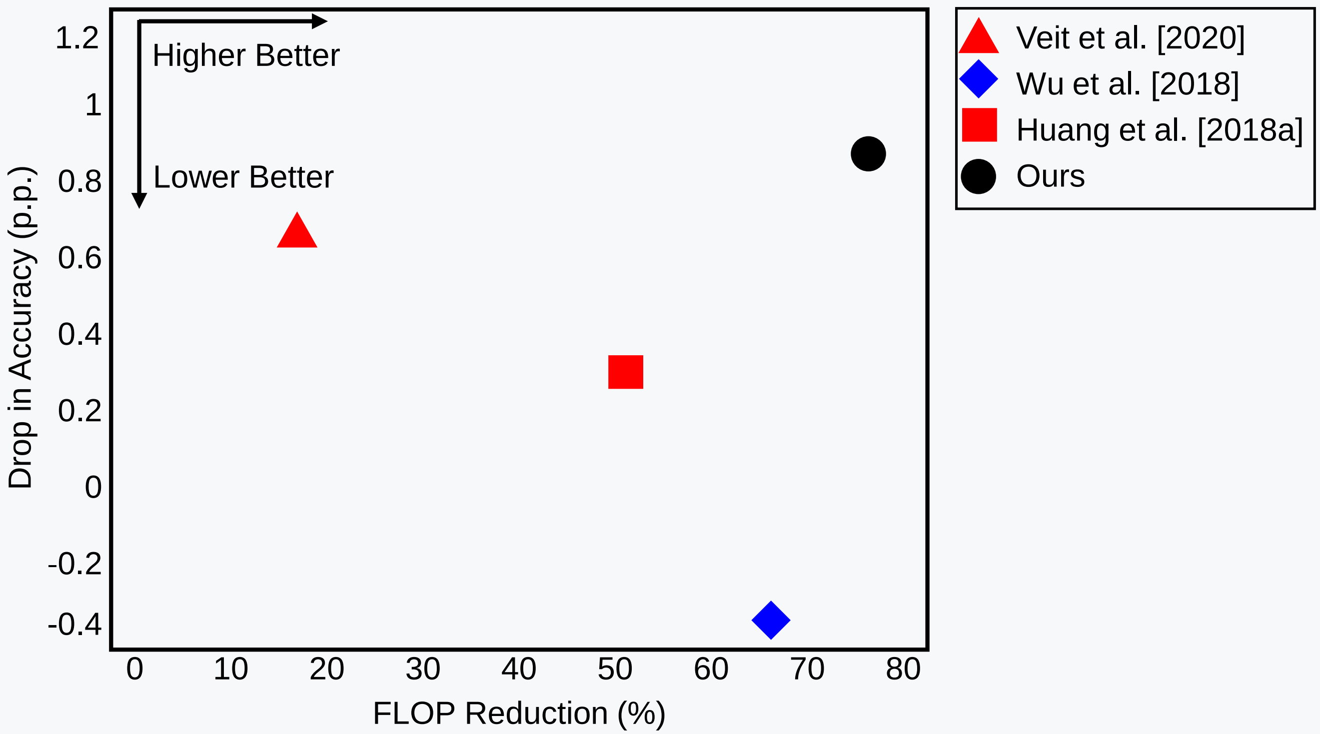
Comparison with other Pruning Criteria

- ResNet56 (CIFAR and ImageNet 32x32) and ReNet50 (ImageNet 224x224)

Layer Importance Criteria	CIFAR-10 Acc. Drop↓	ImageNet (32x32) Acc. Drop↓	ImageNet (224x224) Acc. Drop↓
infFS	-0.68	1.50	-2.03
ilFS	-0.46	1.12	**-2.11**
infFSU	-0.50	2.03	-2.03
KL	-0.32	1.00	-2.06
HRANK	-0.73	2.35	-2.03
ABS	-0.54	**0.96**	**-2.11**
PLS+VIP	**-0.84**	2.25	-1.92

Comparison with other Pruning Approaches

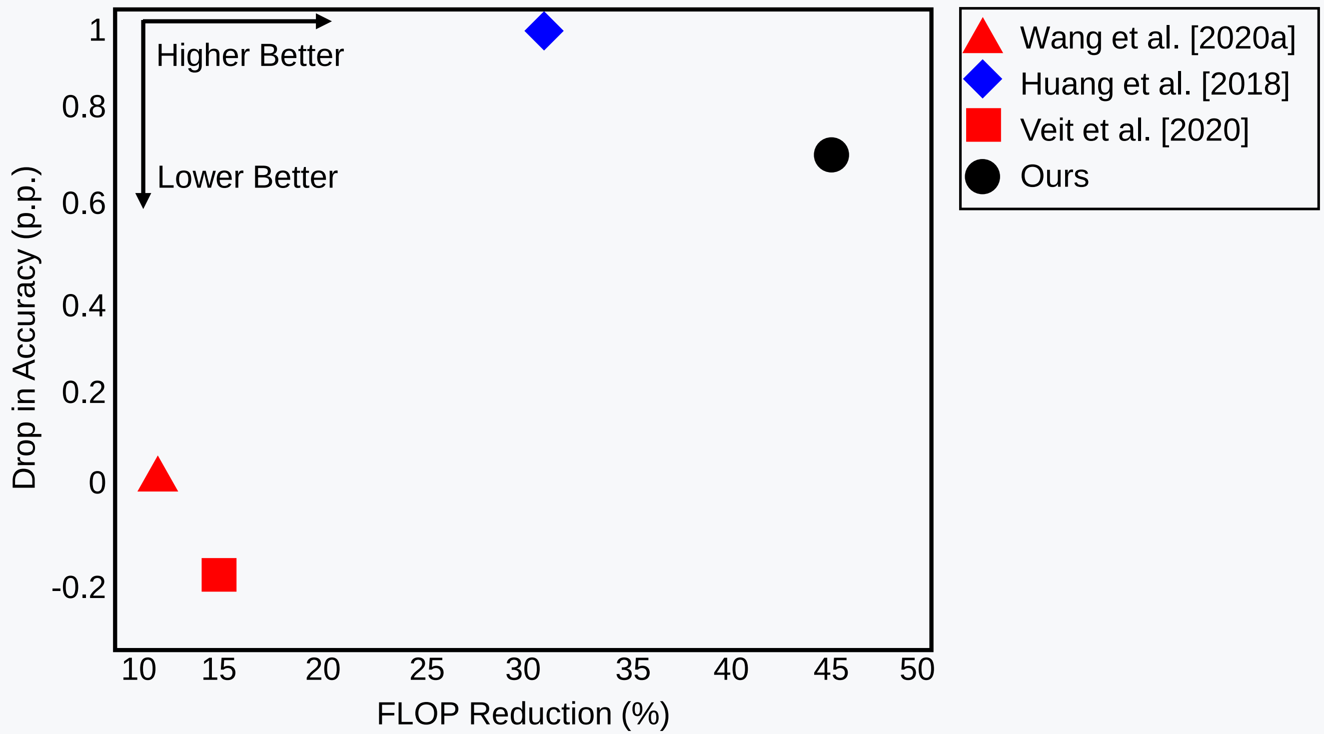
- ResNet110 on CIFAR-10



Veit et al. (2020). *Convolutional networks with adaptive inference graphs*. In IJCV.
Wu et al. (2018a). *Blockdrop: Dynamic inference paths in residual networks*. In CVPR.
Huang et al. (2018). *Data-driven sparse structure selection for deep neural networks*. In ECCV.

Comparison with other Pruning Approaches

- ResNet50 on ImageNet 224x224



Veit et al. (2020). *Convolutional networks with adaptive inference graphs*. In IJCV.
Wu et al. (2018a). *Blockdrop: Dynamic inference paths in residual networks*. In CVPR.
Huang et al. (2018). *Data-driven sparse structure selection for deep neural networks*. In ECCV.

Conclusions

- We demonstrate that is possible to remove unimportant, or least important, layers by estimating their importance using PLS

- Compared to existing criteria for assigning layer importance, PLS achieves competitive results while being more efficient

- Compared to state-of-the-art pruning approaches, our strategy for removing layers achieves the best trade-offs between FLOP reduction and accuracy drop

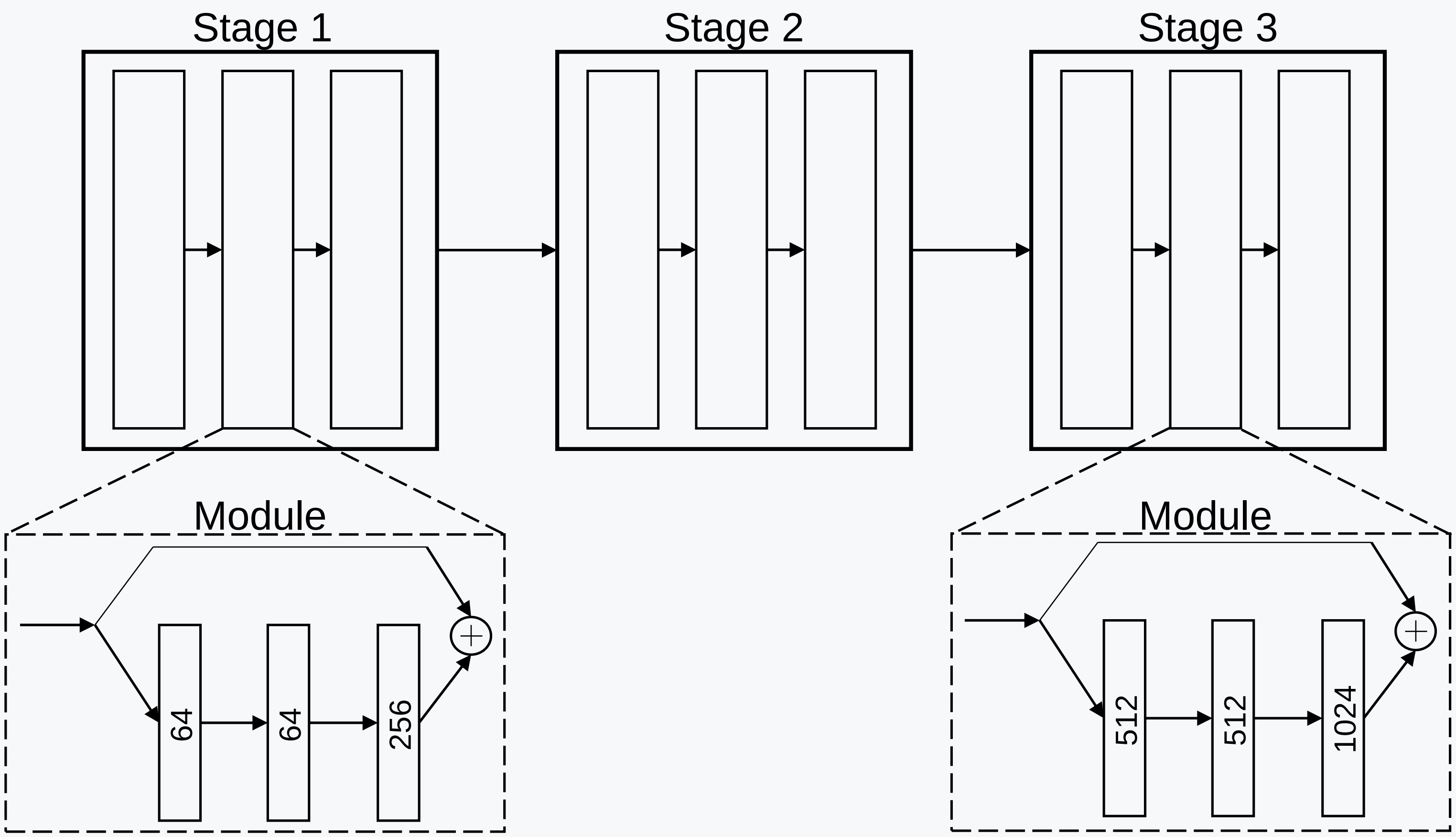
Summary

- Theoretical Concepts
- Pruning Approaches
  - Pruning Filters
  - Pruning Layers
- **Neural Architecture Search**
- HyperNet
- Incremental Partial Least Squares

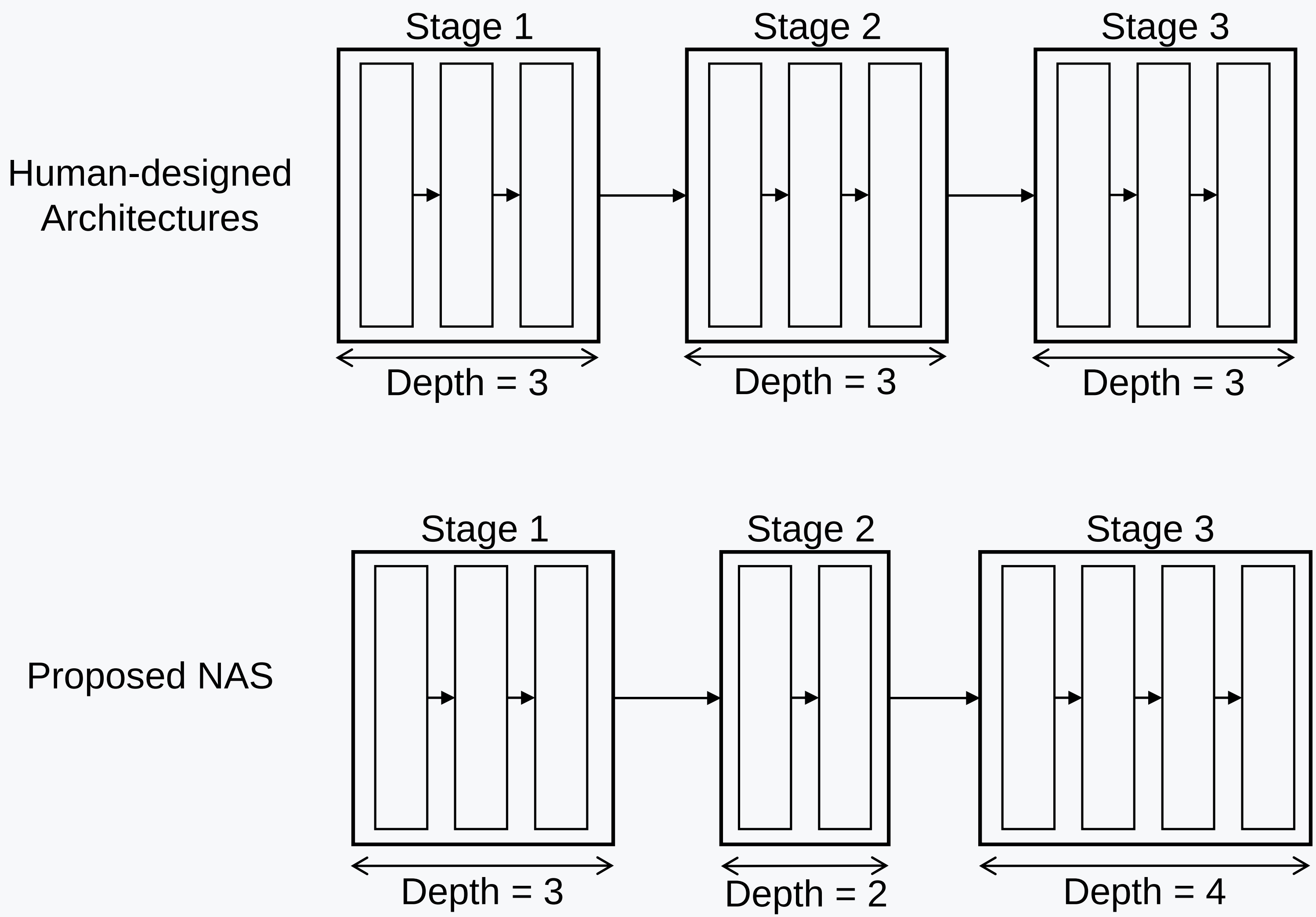
Neural Architecture Search

Problem Definition

- Modern architectures are composed of stages
  - Each stage consists of b modules

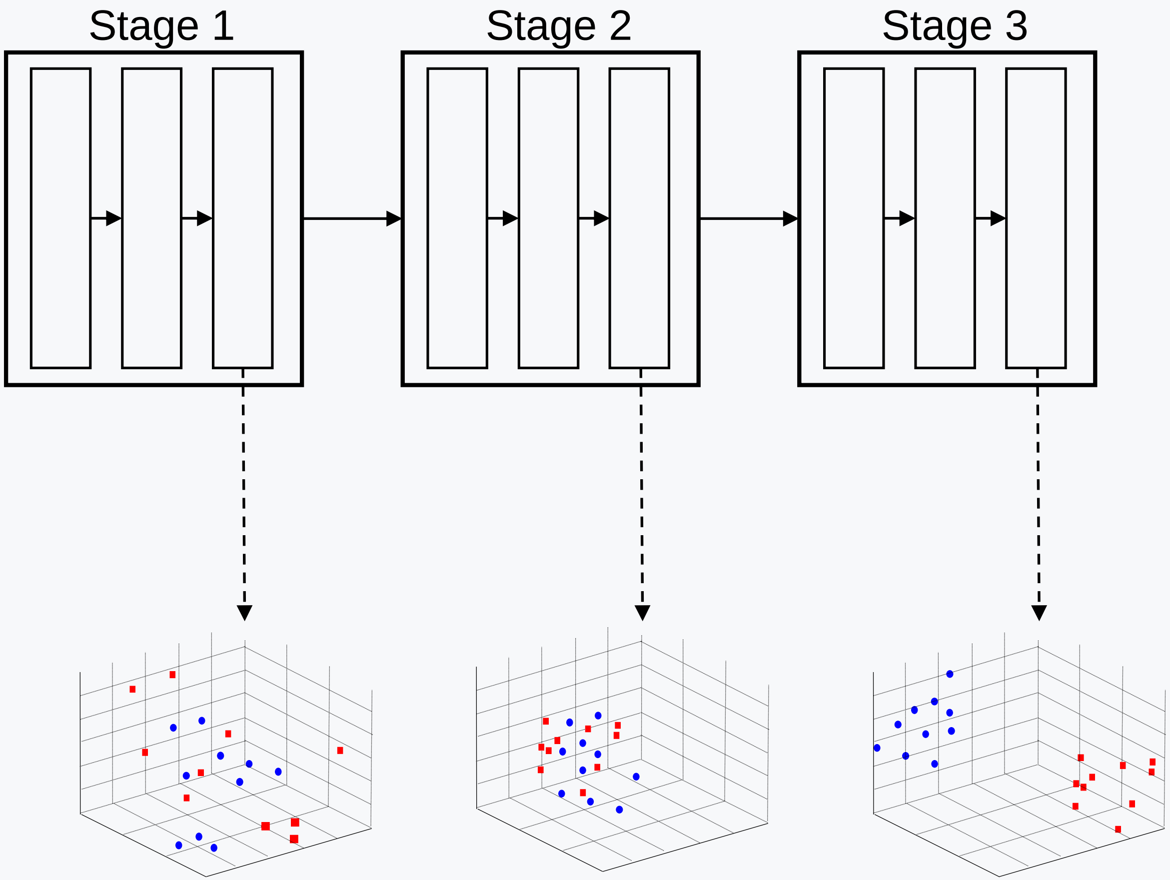


Problem Definition

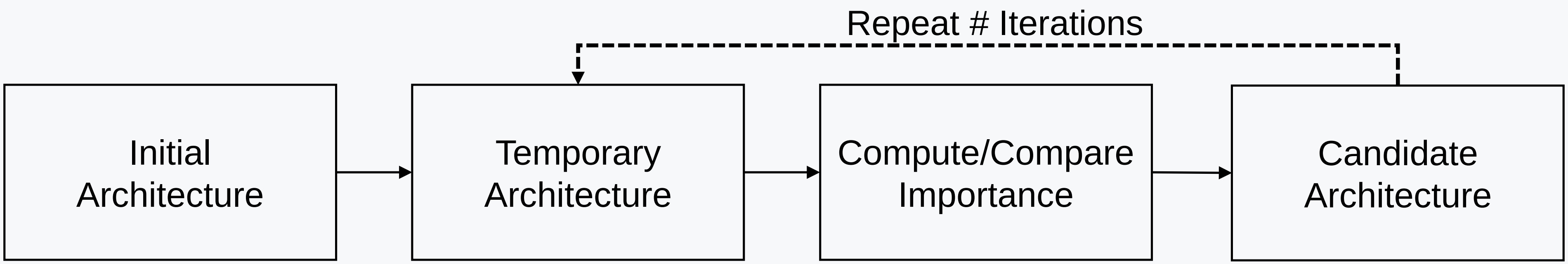


Proposed Approach

Proposed Approach



Overview


Repeat # Iterations
Initial Architecture
Temporary Architecture
Compute/Compare Importance
Candidate Architecture

Experiments Neural Architecture Search

Importance Criteria

- CIFAR-10

Criterion	Iteration (ith Candidate Arch.)				
	1	2	3	4	5
infFS [Roffo et al. 2015]	91.59	92.09	92.02	92.36	92.45
ilFS [Roffo et al. 2017]	91.94	92.06	92.10	92.08	92.52
infFSU [Roffo et al. 2020]	90.42	92.26	91.95	92.41	**92.64**
PLS+VIP	**92.03**	**92.38**	**92.62**	**92.53**	92.58

Roffo et al. (2015). *Infinite feature selection*. In ICCV.
Roffo et al. (2017). *Infinite latent feature selection: A probabilistic latent graph-based ranking approach*. In ICCV.
Roffo et al. (2020). *Infinite feature selection: a graph-based feature filtering approach*. In PAMI.

Comparison with human-designed architectures

- CIFAR-10
  - * indicates human-designed architectures

Architecture	Depth	Param. ↓ (Million)	FLOP↓ (Million)	Accuracy↑
ResNet44*	44	0.66	97	92.83
Ours (it=1)	**43**	**0.60**	**92**	**93.38**
ResNet56*	**56**	0.86	**125**	93.03
Ours (it=3)	59	**0.69**	130	**93.36**
ResNet110*	110	1.7	253	93.57
Ours (i=5)	**67**	**0.88**	**149**	**94.27**

Comparison with state-of-the-art NAS

- CIFAR-10

Model	Evaluated↓ Models	GPUs ↓	Param.↓ (Million)	Accuracy↑
Zoph et al. [2018]	20, 000	800	2.5	94.51
Real et al. [2017]	1, 000	250	5.4	94.60
Dong and Yang [2019]	240	**1**	2.6	96.25
Yang et al. [2020b]	128	**1**	3.6	**97.38**
Jin et al. [2019]	60	**1**	---	88.56
Ours (it=5)	**11**	**1**	**2.3**	94.74

Zoph et al. (2018). *Learning transferable architectures for scalable image recognition*. In CVPR.
Real et al. (2017). *Large-scale evolution of image classifiers*. In ICML.
Dong and Yang et al. (2019). *Searching for a robust neural architecture in four GPU hours*. In CVPR.
Yang et al. (2020b). *CARS: continuous evolution for efficient neural architecture search*. In CVPR.
Jin et al. (2019). *Auto-keras: An efficient neural architecture search system*. In SIGKDD.

Conclusions

- We demonstrate that it is possible to design high-performance convolutional architectures by inserting layers based on their importance
  - Layer importance is assigned by PLS

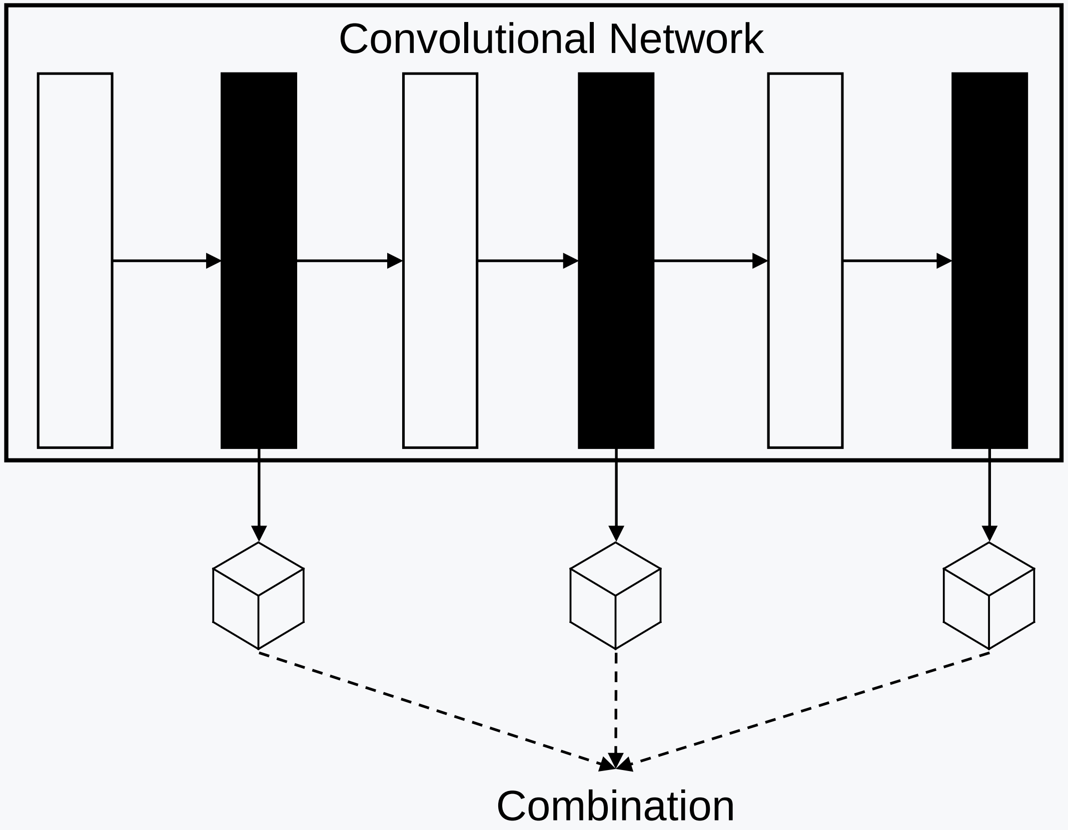
- Compared to NAS strategies, our method is extremely more efficient, as it evaluates one order of magnitude fewer models and discovers architectures on par with the state of the art

Summary

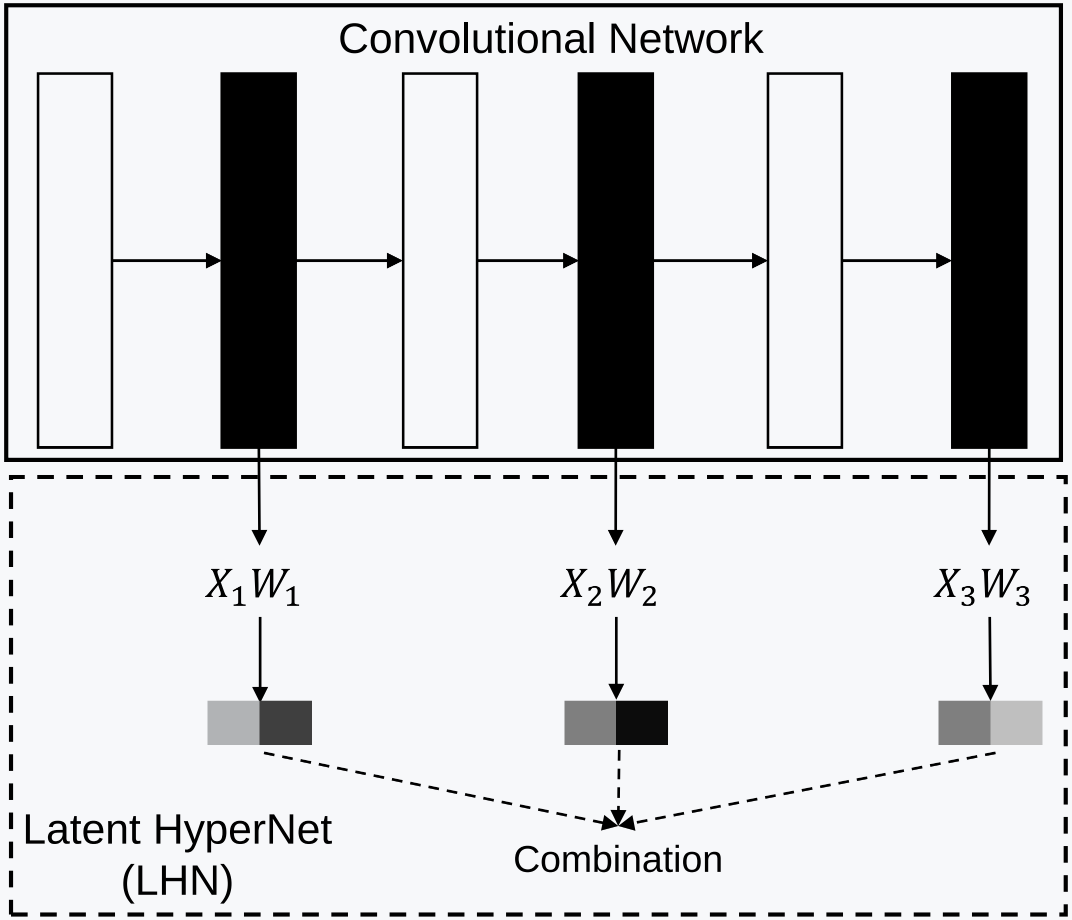
- Theoretical Concepts
- Pruning Approaches
  - Pruning Filters
  - Pruning Layers
- Neural Architecture Search
- **HyperNet**
- Incremental Partial Least Squares

HyperNet Approach

Problem Definition



Proposed Approach



Experiments
Latent HyperNet

Improvements

- Improvement in accuracy

Architecture	Method	CIFAR-10↑	ImageNet 32x32↑
VGG16	Kong et al. [2016]	-0.22	0.01
	LHN (Ours)	**0.05**	**0.66**
ResNet20	Kong et al. [2016]	**-0.02**	**3.60**
	LHN (Ours)	-0.13	2.65

Kong et al. (2016). *Hypernet: Towards accurate region proposal generation and joint object detection*. In CVPR.

Computational Cost

- Floating Point Operations
  - Million

Architecture	Method	CIFAR-10↓	ImageNet 32x32↓
VGG16	Kong et al. [2016]	313.54	314.05
	LHN (Ours)	**313.22**	**313.72**
ResNet20	Kong et al. [2016]	43.91	44.42
	LHN (Ours)	**40.85**	**41.36**

Kong et al. (2016). *Hypernet: Towards accurate region proposal generation and joint object detection*. In CVPR.

Conclusions

- We demonstrate that an efficient yet effective way of combining multiple levels of features is to project them on the latent space of PLS

- Compared to time-consuming operations, we demonstrate that the PLS projection enhances data representation at negligible additional cost

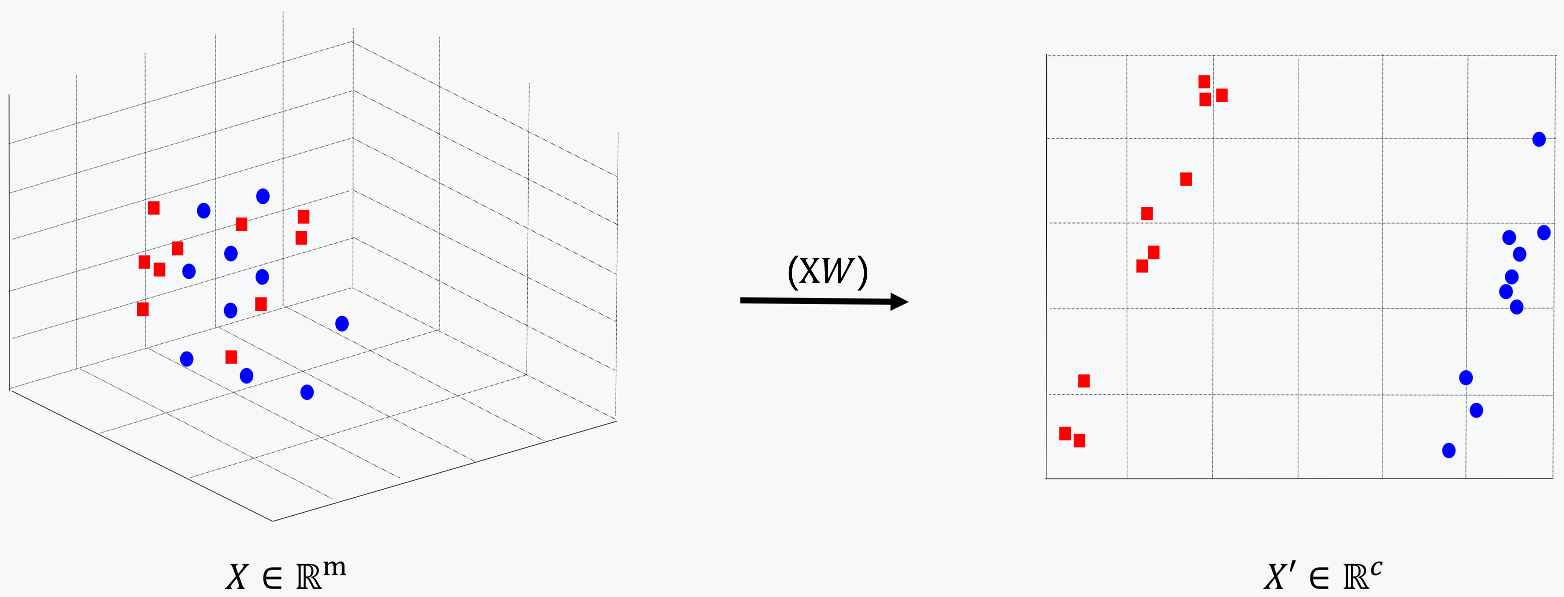
Summary

- Theoretical Concepts
- Pruning Approaches
  - Pruning Filters
  - Pruning Layers
- Neural Architecture Search
- HyperNet
- **Incremental Partial Least Squares**

Incremental PLS Approach

Problem Definition

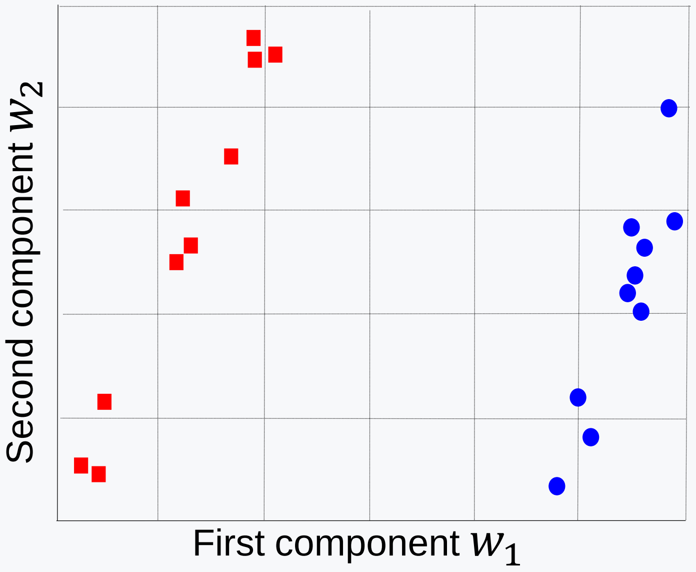
- Find a projection $W(w_1, w_2, \ldots, w_c)$ using a single sample x and its respective label y
  - Keep the property of maximizing the covariance across all c-components



Proposed Approach

- Partial Least Squares estimates the *ith* component in terms of
  - $w_i = X^T Y$

- Compute the *ith* component by decomposing $X^T Y$ as
  - $X^T Y = \sum x^T y \Rightarrow w_i = w_i + (x^T y)$ [Zeng et al. 2014]



Zeng et al. (2014). *Incremental Partial Least Squares Analysis of Big Streaming Data*. In Pattern Recognition.

Proposed Approach

- Decomposition
  - $X^T Y \Rightarrow w_i = w_i + (x^T y)$

Traditional PLS deflation		Proposed Deflation
$t = Xw_i$	Single Sample Projection $\longrightarrow$	$t = xw_i$
$p_i = X^T t$	**Decomposition** $\longrightarrow$	$\boldsymbol{p_i} = \boldsymbol{p_i} + (\boldsymbol{x^T t})$
$q_i = Y^T t$	**Decomposition** $\longrightarrow$	$\boldsymbol{q_i} = \boldsymbol{q_i} + (\boldsymbol{y^T t})$
$X = X - tp_i^T$	Single Sample Deflation $\longrightarrow$	$x = x - tp_i^T$
$Y = Y - tq_i^T$	Single Label Deflation $\longrightarrow$	$y = y - tq_i^T$

Overview

CIPLS Algorithm

Foreach $x \in X \ and \ y \in Y$ **do**

for $i = 1$ **to** c **do**

$$\boldsymbol{w_i} = \boldsymbol{w_i} + (\boldsymbol{x^T y})$$

$$t = x w_i$$

$$p_i = p_i + (x^T t)$$

$$q_i = q_i + (y^T t)$$

$$x = x - t p_i^T$$

$$y = y - t q_i^T$$

end

end

PLS Algorithm

for $i = 1$ **to** c **do**

$$\boldsymbol{w_i} = \boldsymbol{X^T Y}$$

$$t = X w_i$$

$$p_i = X^T t$$

$$q_i = Y^T t$$

$$X = X - t p_i^T$$

$$Y = Y - t q_i^T$$

end

Experiments
CIPLS

Comparison with other Incremental Methods

- Face verification - Labeled Faces in the Wild (LFW)

Method	Accuracy↑	Difference to PLS↓
SGDPLS [Arora et al., 2016]	90.60 [89.95 91.24]	1.87
IPLS [Zeng and Li, 2014]	90.30 [89.60 90.99]	2.17
PLS	**92.47** [91.87 93.05]	---
CIPLS (Ours)	91.78 [91.08 92.47]	**0.69**

Zeng et al. (2014). *Incremental partial least squares analysis of big streaming data.* Pattern Recognition.
Arora et al (2016). *Stochastic optimization for multiview representation learning using partial least squares*. In ICML.

Comparison with other Incremental Methods

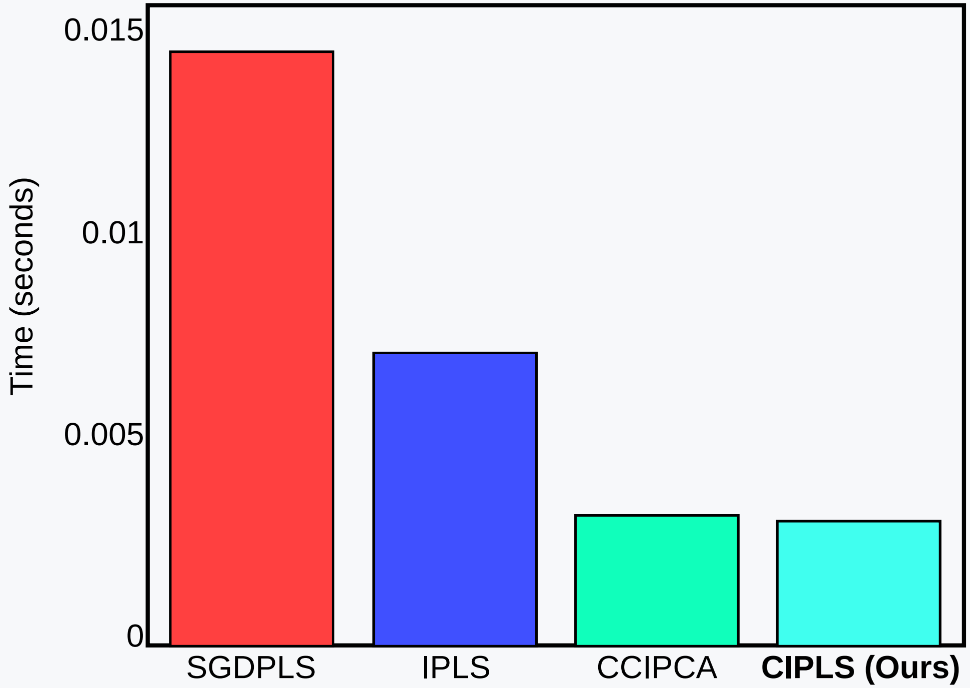
- Image classification - ImageNet

Method	Accuracy ↑ ImageNet 32x32	Accuracy ↑ ImageNet 224x224
SGDPLS [Arora et al., 2016]	---	---
IPLS [Zeng and Li, 2014]	43.24	64.74
PLS	---	---
CIPLS	**43.31**	**67.09**

Zeng et al. *Incremental partial least squares analysis of big streaming data.* Pattern Recognition, 2014.
Arora et al. *Stochastic optimization for multiview representation learning using partial least squares*. In ICML. 2016.

Computational Cost

- Time (in seconds) for estimation the projection matrix



Conclusions

- We show that it is possible to compute all components of PLS incrementally using simple algebraic decomposition
  - Preserves all the properties of PLS across all components
  - Computationally efficient and low time complexity

- Our CIPLS is the most accurate and fastest incremental PLS

Publications

Journal Papers

- Jordao, A., Yamada, F., and Schwartz, W. R. **Deep Network Compression based on Partial Least Squares**. Neurocomputing, 2020

- Jordao, A., Lie, M., and Schwartz, W. R. **Discriminative Layer Pruning for Convolutional Neural Networks**. Journal of Selected Topics in Signal Processing, 2020

Publications

Conference Papers

- Jordao, A., Kloss, R. B., and Schwartz, W. R. Latent hypernet: **Exploring the layers of Convolutional Neural Networks**. In International Joint Conference on Neural Networks (IJCNN), 2018

- Jordao, A., Kloss, R., Yamada, F., and Schwartz, W. R. **Pruning Deep Neural Networks using Partial Least Squares**. British Machine Vision Conference (BMVC) Workshops: Embedded AI for Real-Time Machine Vision, 2019

- Jordao, A., Lie, M., Yamada, F., and Schwartz, W. R. **Stage-Wise Neural Architecture Search**. International Conference on Pattern Recognition (ICPR). Accepted for publication, 2020

- Jordao, A., Lie, M., de Melo, V. H. C., and Schwartz, W. R. **Covariance-free partial least squares: An Incremental Dimensionality Reduction Method.** Winter Conference on Applications of Computer Vision (WACV). Accepted for publication, 2021

Acknowledgments

Codes

- Code is available at:
  - https://github.com/arturjordao

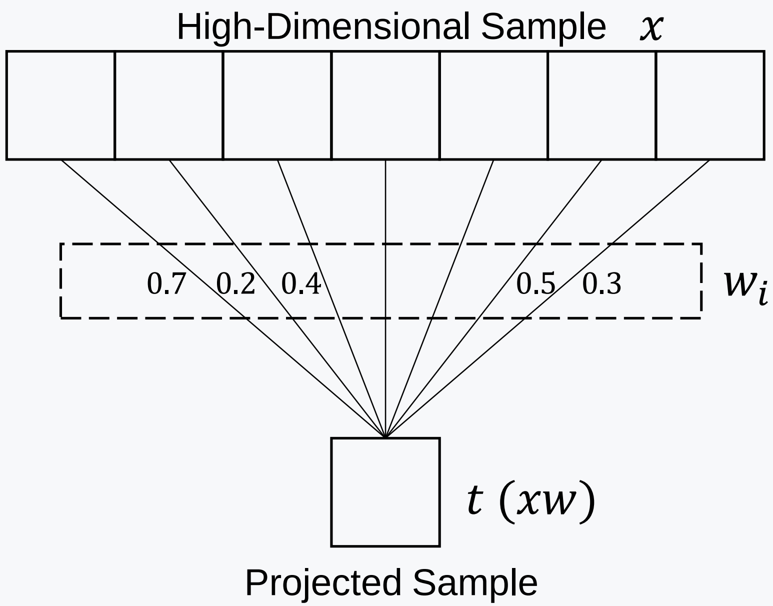
Latent HyperNet | Pruning Filters | Pruning Layers | CIPLS

Additional Slides

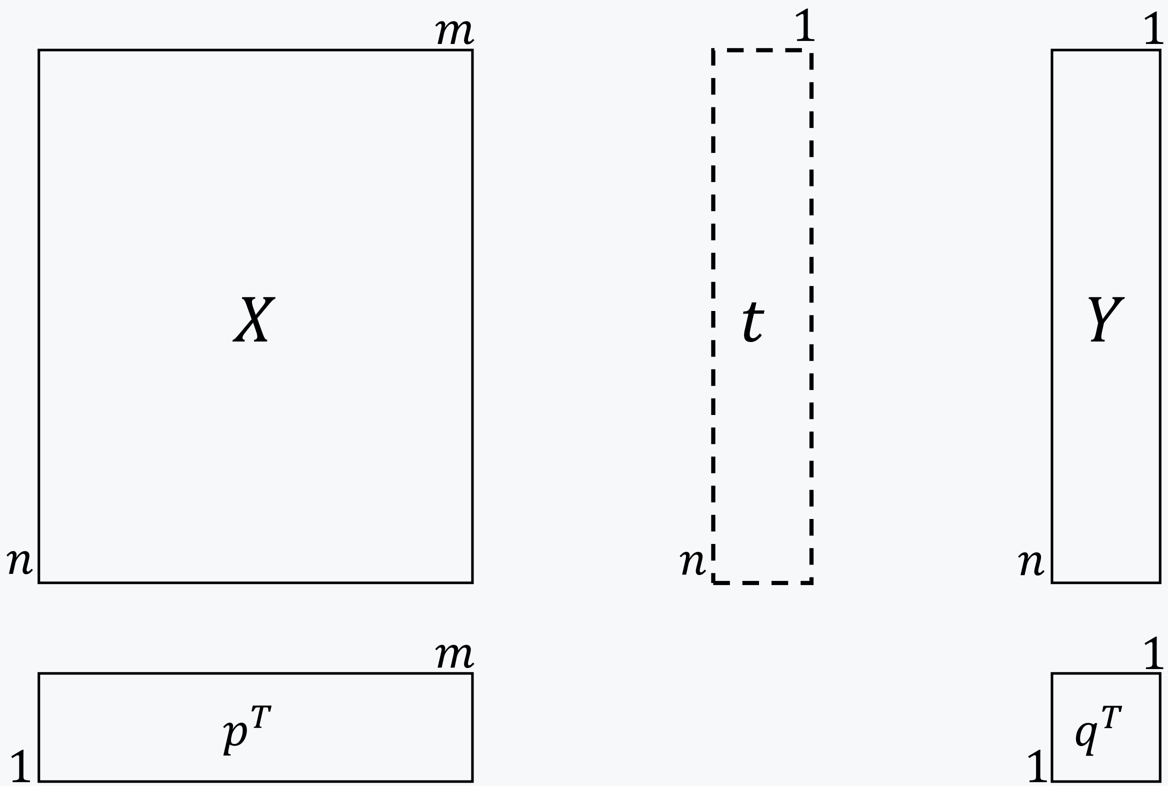
Thesis Statement

The importance of structures (neurons or layers) composing a convolutional network can be effectively estimated with Partial Least Squares, which in turn can be computed incrementally without degrading its discriminative information. With the estimation of this importance, it is possible to obtain high-performance convolutional networks by removing, inserting or combining structures

Projection


High-Dimensional Sample x
0.7 0.2 0.4 0.5 0.3
w_i
$t\ (xw)$
Projected Sample

Partial Least Squares



Partial Least Squares

PLS1 Algorithm

for $i = 1$ **to** c **do**

$$w_i = X^T Y$$

$$t = X w_i$$

$$p_i = X^T t$$

$$q_i = Y^T t$$

$$X = X - t p_i^T$$

$$Y = Y - t q_i^T$$

end

PLS2 Algorithm

for $i = 1$ **to** c **do**

$initialize\ u \in R^{nx1}$

$Repeat\ until\ convergence$

$$w_i = X^T u$$

$$t = X w_i$$

$$q_i = Y^T t$$

$$u = Y q_i$$

end

$$p_i = X^T t$$

$$X = X - t p_i^T$$

$$Y = Y - t q_i^T$$

end

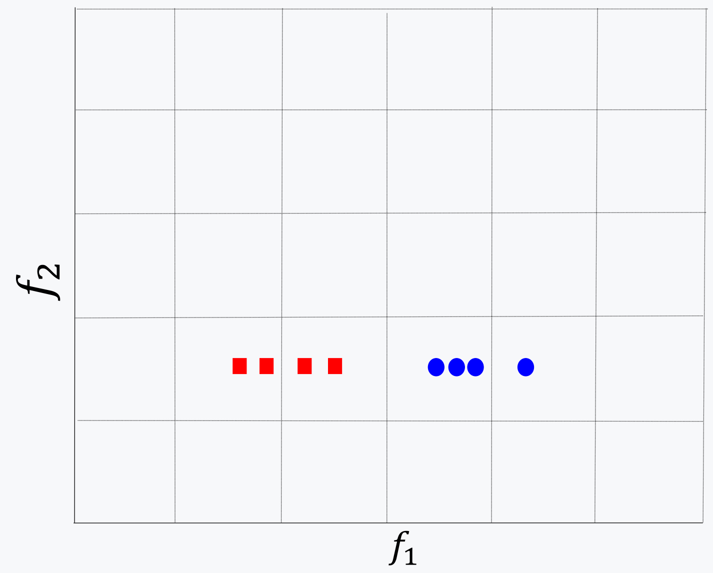
Variable Importance in Projection

$$f_j = \sqrt{m \sum_{i=1}^{c} SS_i (w_{ij} / \|w_i\|^2) / \sum_{i=1}^{c} SS_i}$$

$$SS_i = q_i^2 t_i^T t_i$$

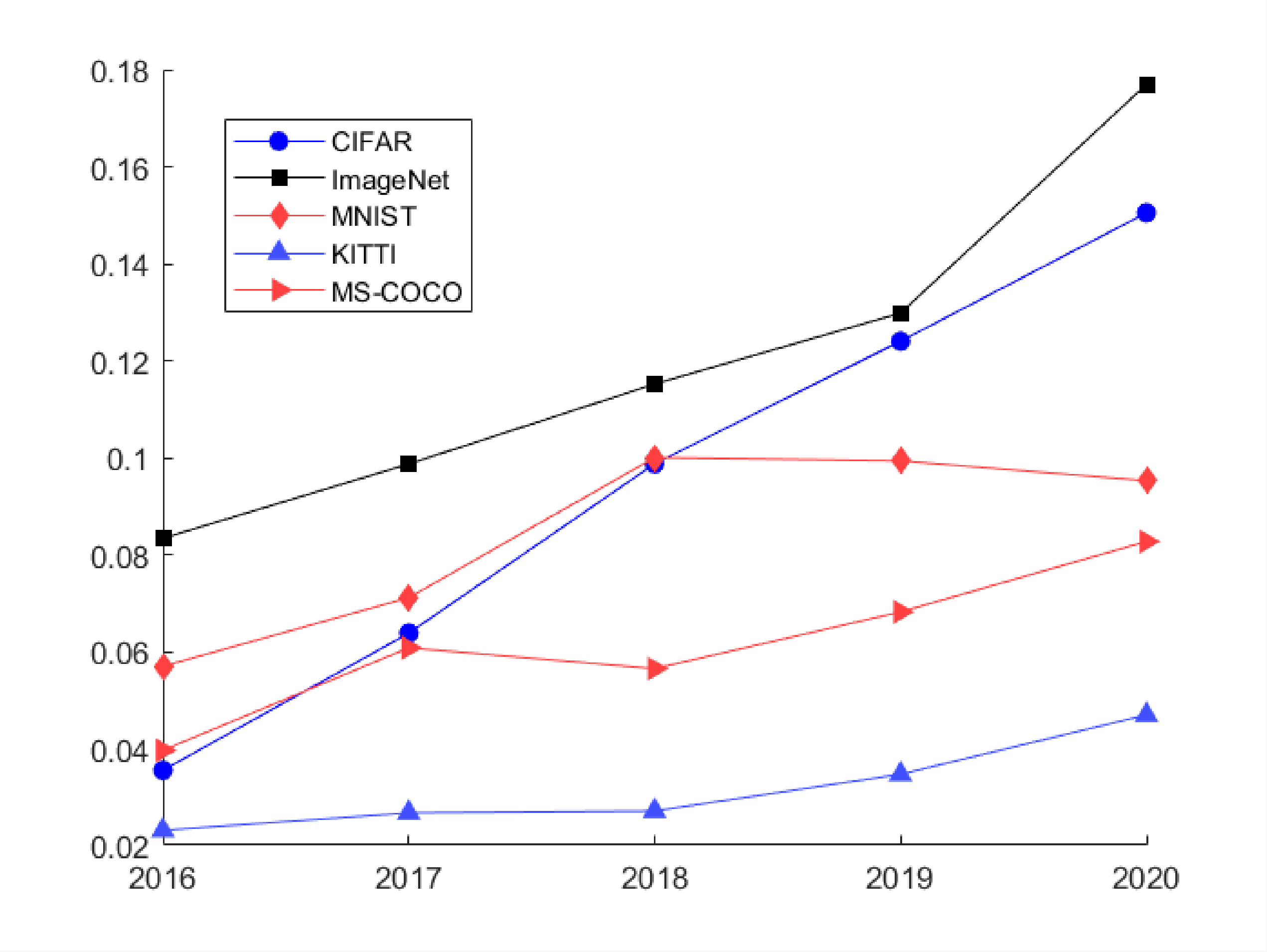
PLS vs. CCA

$$corr(X^i, Y^i) = \frac{cov\ (X^i, Y^i)}{var(X^i) * var(Y^i)}$$



$$cov(X^i, Y^i) = var(X^i) * var(Y^i) * corr(X^i, Y^i)$$

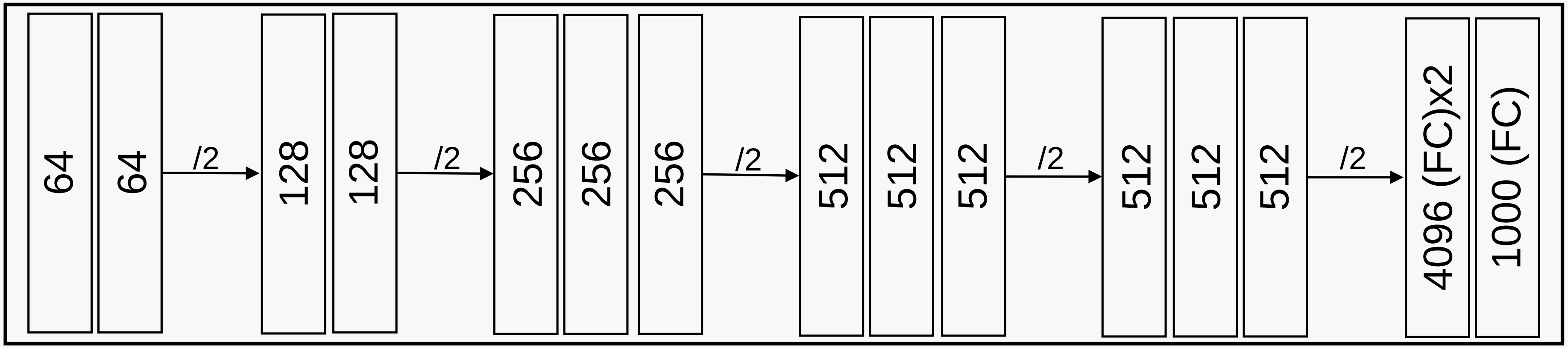
Benchmarks


CIFAR
ImageNet
MNIST
KITTI
MS-COCO
0.18
0.16
0.14
0.12
0.1
0.08
0.06
0.04
0.02
2016
2017
2018
2019
2020

VGG16

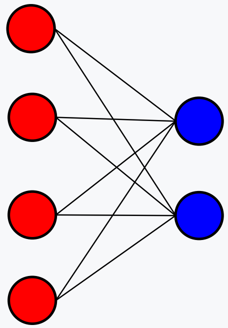
- 3x3 filters

VGG16 Architecture



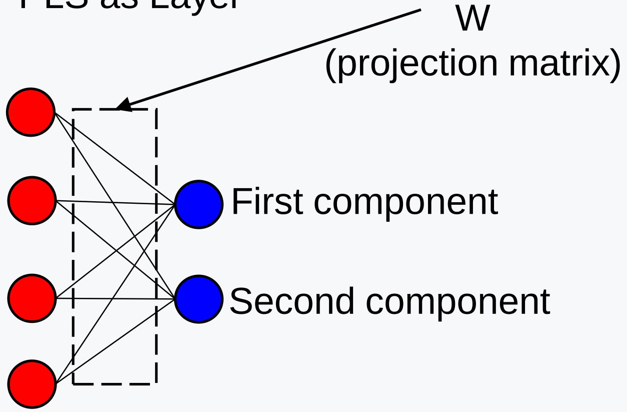
PLS GPU

Fully Connected Layer



$$f(x) = xW + b$$

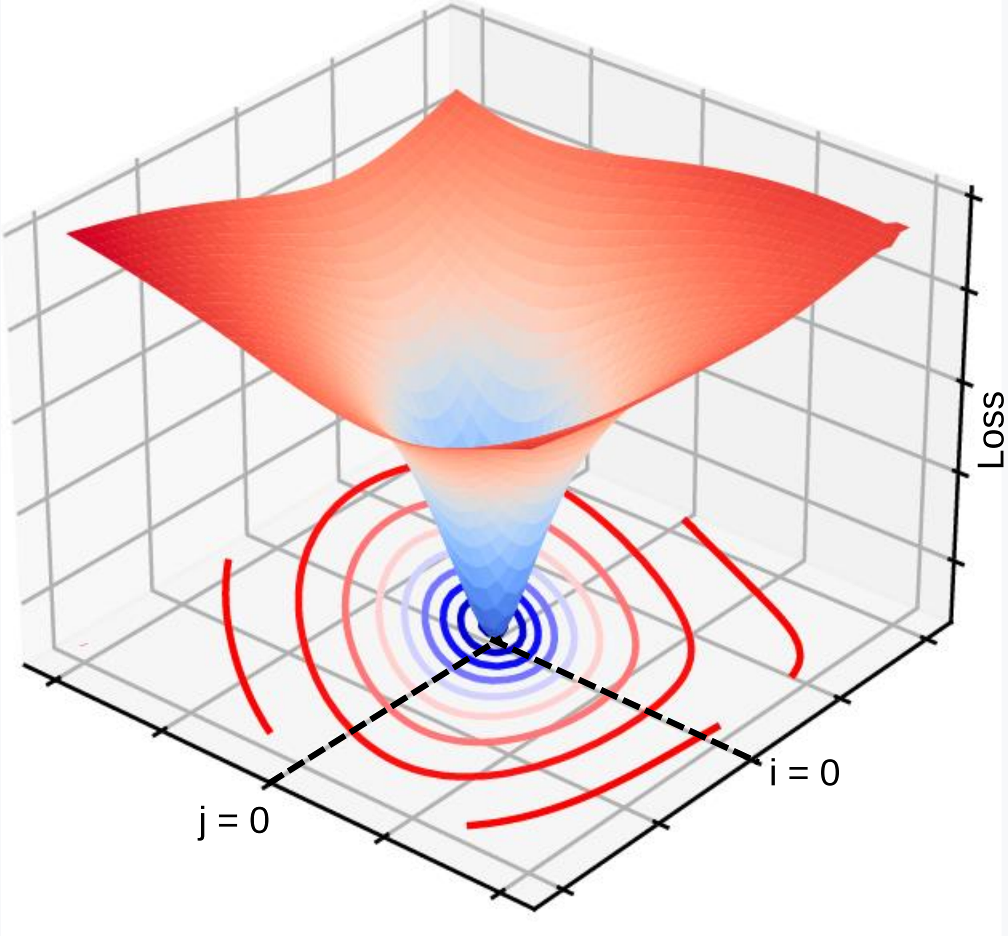
PLS as Layer



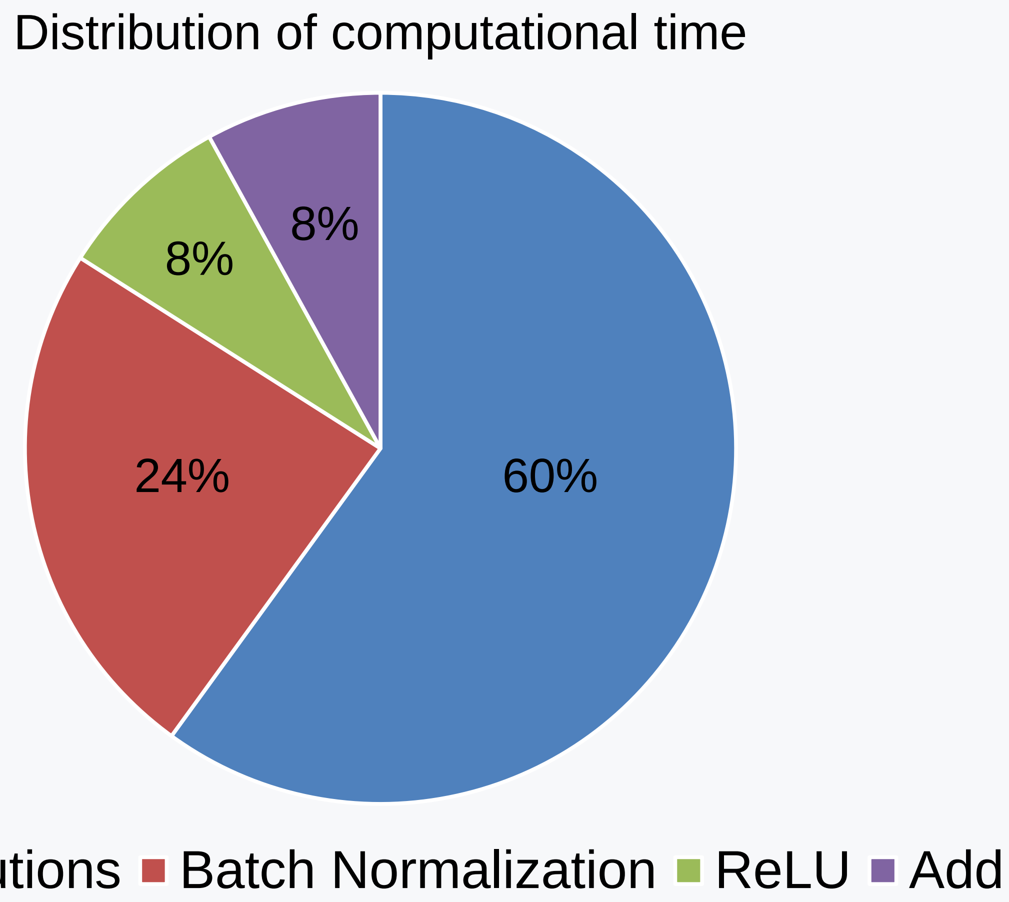
$$f(x) = xW$$

Loss Landscape

- $\theta + (i * \alpha) + (j * \beta)$
  - θ network parameters
  - α, β random distributions



Computational Time

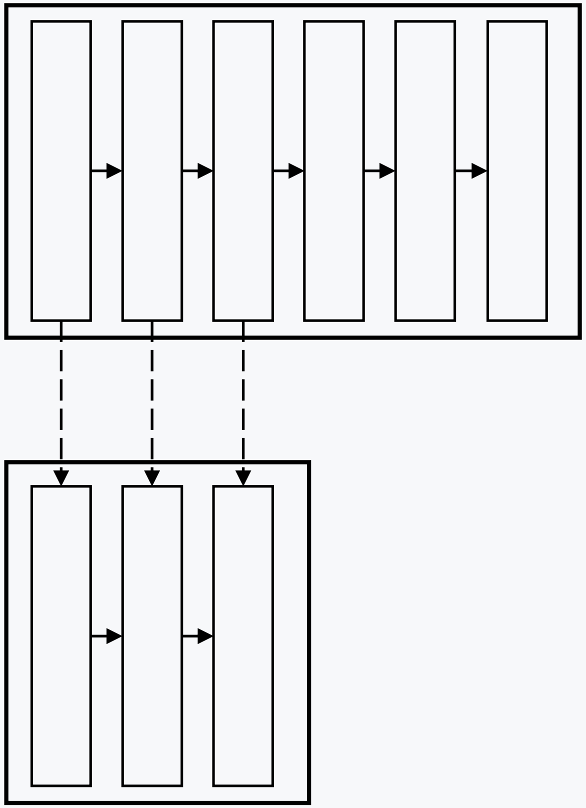


Pruning Approach Additional Slides

NAS
Additional Slides

Weight Transfer

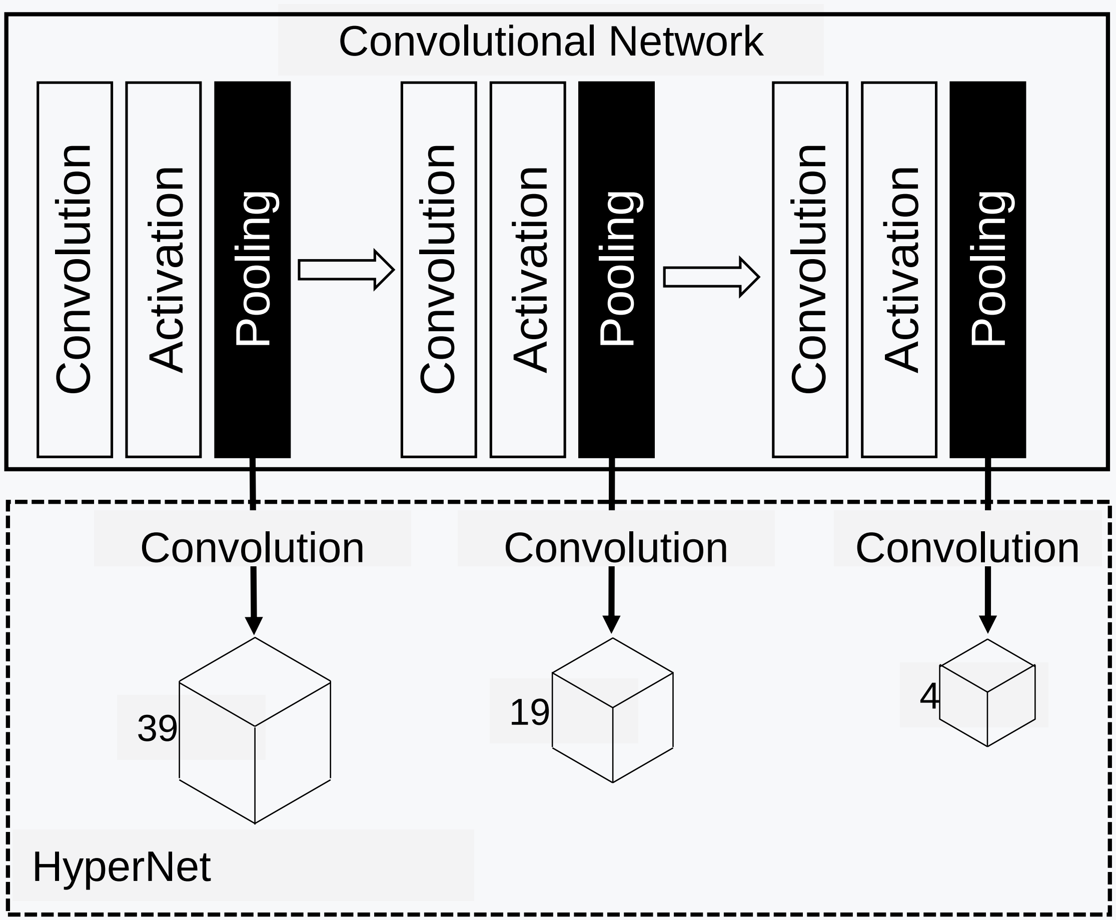
Human-designed
Architectures



Candidate
Architecture

Latent HyperNet Additional Slides

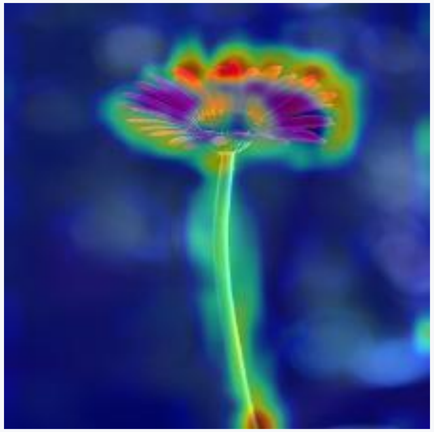
Baseline



Kong et al. Hypernet: Towards accurate region proposal generation and joint object detection. In CVPR, 2016.

HyperNet


Original Image


Early Layer


Deep Layer

Incremental PLS Additional Slides

IPLS Overview

- Compute the component w_i in terms of
  - $maximize\big(cov(Xw, Y)\big) = X^TY \Rightarrow w_i = X^TY$
  - X and its respective Y are not in memory in advance

- Decomposition[11]
  - $X^TY = \sum(x_n y_n)$
  - $w_i = w_i + (x_n y_n)$

IPLS Algorithm

foreach $x_n \in X \; and \; y_n \in Y$ **do**

$w_0 = \bar{x}_n y_n + w_{0(n-1)}$

$CCIPCA(\bar{x})$[12]

for $i = 2$ **to** $c - 1$ **do**

$w_i = C^{i-1} w_0$

end

end

[11] Zeng et al. *Incremental partial least squares analysis of big streaming data.* Pattern Recognition, 2014.
[12] Weng et al. *Candid covariance-free incremental principal component analysis*. In PAMI, 2003.

SGDPLS Overview

SGDPLS Algorithm

foreach $x_n \in X \text{ and } y_n \in Y$ **do**

  for $ep = 1$ **to** $Epochs$ **do**

    $W_n = \alpha(x_n y_n)\beta_{n-1}$

    $\beta_n = \alpha(y_n x_n)W_{n-1}$

  end

end

CCIPCA Overview

CCIPCA Algorithm

foreach $x_n \in X \; and \; y_n \in Y$ **do**

$k = \min(n, L)$

for $i = 1$ **to** k **do**

$$\lambda = \frac{n - 1 - l}{n}$$

$$\theta = \frac{n + l}{n}$$

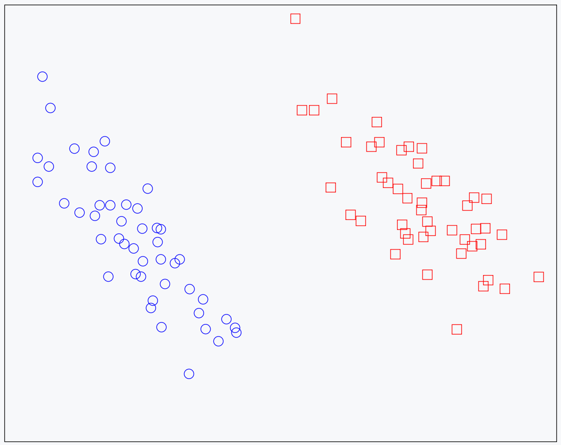
$$w_i = \lambda w_i + \theta x_n (x_n^T w_i)$$

$$x_n = x_n - \; (x_n^T w_i) w_i$$

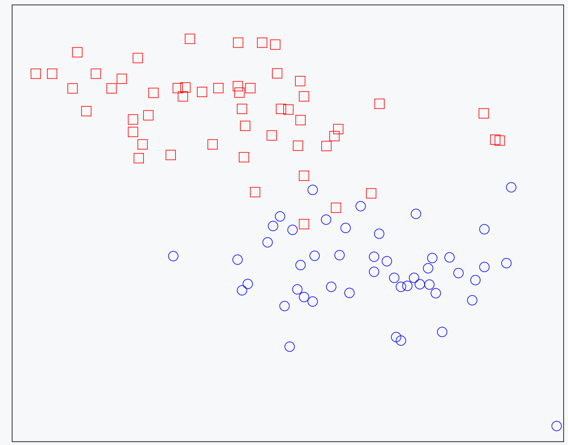
end

end

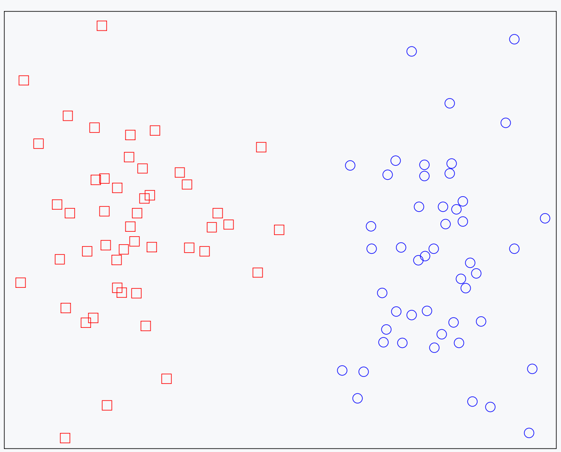
Higher-order Components



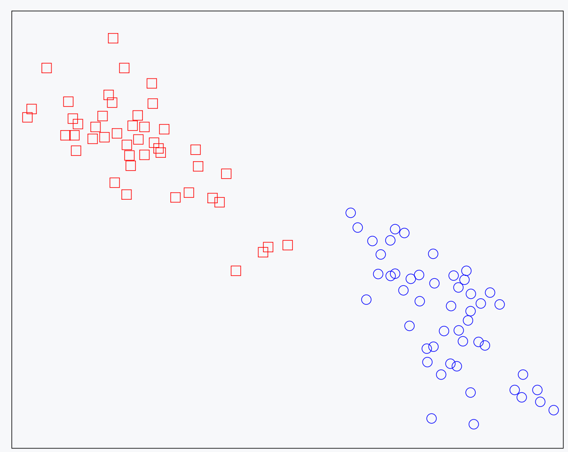
PLS



SGDPLS



IPLS



CIPLS

Introduction

- Pattern recognition methods have led to a series of breakthroughs
  - Improvement in data representation (features)
  - Learn features from raw data (convolutional networks)
  - Transformations on the pre-computed features (dimensionality reduction)