Strategic Freezing

Zachary Seligman, David Patrick, Amanda Fernandez
The University of Texas at San Antonio, San Antonio TX, 78249

UTSA Computer Science College of Sciences

Vision • Artificial • Intelligence • Laboratory

CONNECT

Abstract

Convolutional neural networks (CNNs) are notoriously data-intensive, requiring significantly large datasets for training accurately in an appropriate runtime. Recent approaches aiming to reduce this requirement focus on removal of low-quality samples in the data or unimportant filters, leaving a vast majority of the training set and model in tact. We propose Strategic Freezing, a new training strategy which strategically freezes features in order to maintain class retention. Preliminary results of our approach are demonstrated on the Imagenette dataset using ResNet34.

Introduction

- Deep Neural Networks (DNNs) require significant data.
  - Most approaches to reduce training data are vulnerable to *Catastrophic Forgetting.*
  - Approaches to remove filters that aren't unimportant are vulnerable to *model drift.*
- We propose a new training strategy: Strategic Freezing
  - Provides a definitive end to the training process.
  - Leverages freezing on filters and residue to prevent the *Catastrophic Forgetting* problem.

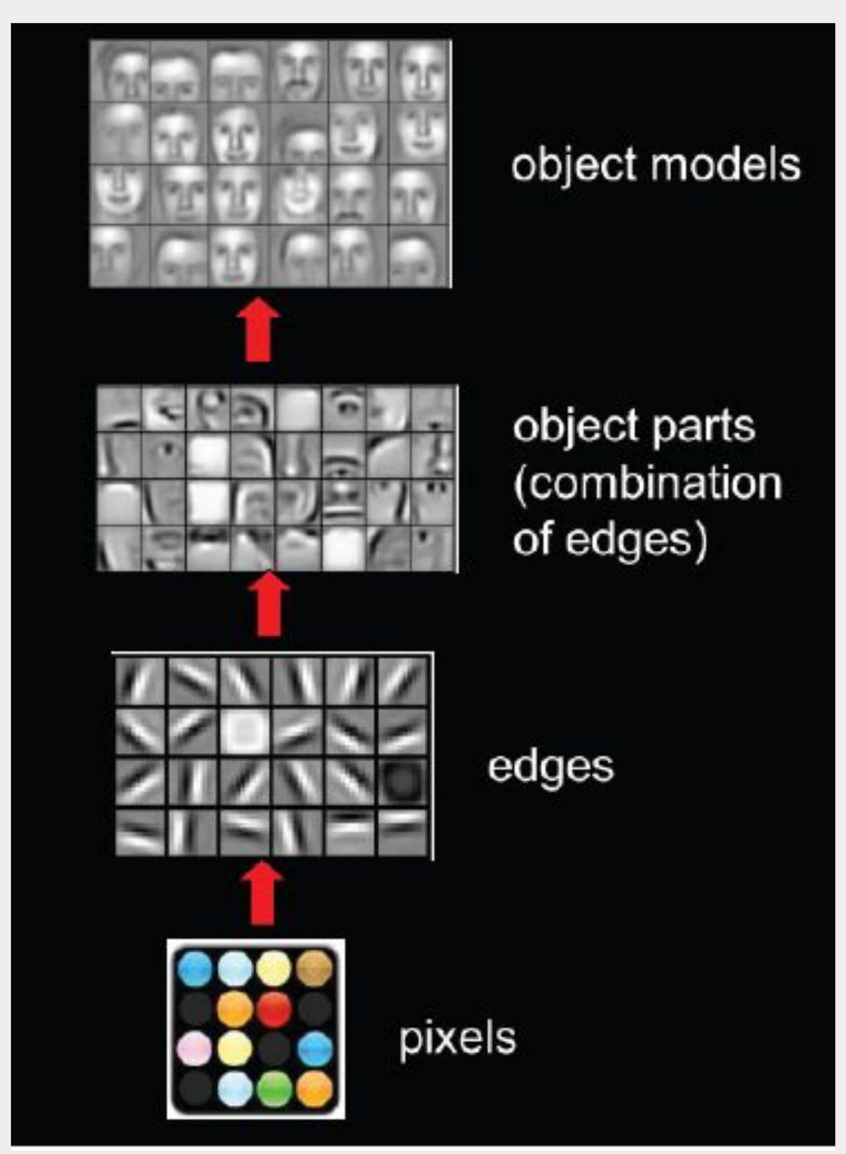

Figure 1: Visualization of high and low level filters

Method

Algorithm 1: Proposed Training Strategy - Selected Freezing

Input: f: Network, n: Max epochs, c: Number of classes
Data: X: Training dataset, C: Class above threshold

while $epochs < n$ *and* $len(X) \neq 0$ **do**
  Train f on X to obtain per class $f1scores*$
  Validate f on X
  for $f1score$ *in* $f1scores$ **do**
    if $f1score >= threshold$ **then**
      for *layer in layers_to_freeze* **do**
        Get *activations* from *layer*
        for *activation in activations* **do**
          for *filter in activation* **do**
            filter_score[*image*, *filter*] = *max*(*filter*)
        Sum *filter* score across X_C
        Return *filter* ranks for X_C
      for *layer in layers_to_freeze* **do**
        if *layer contains filter to freeze* **then**
          Add top% of filters to *filters_to_freeze*

Algorithm 2: Gradient Removal During Training

for *convolution in convolutions* **do**
  if *convolution contains freeze* **then**
    Set gradient for *filters_to_freeze* to 0

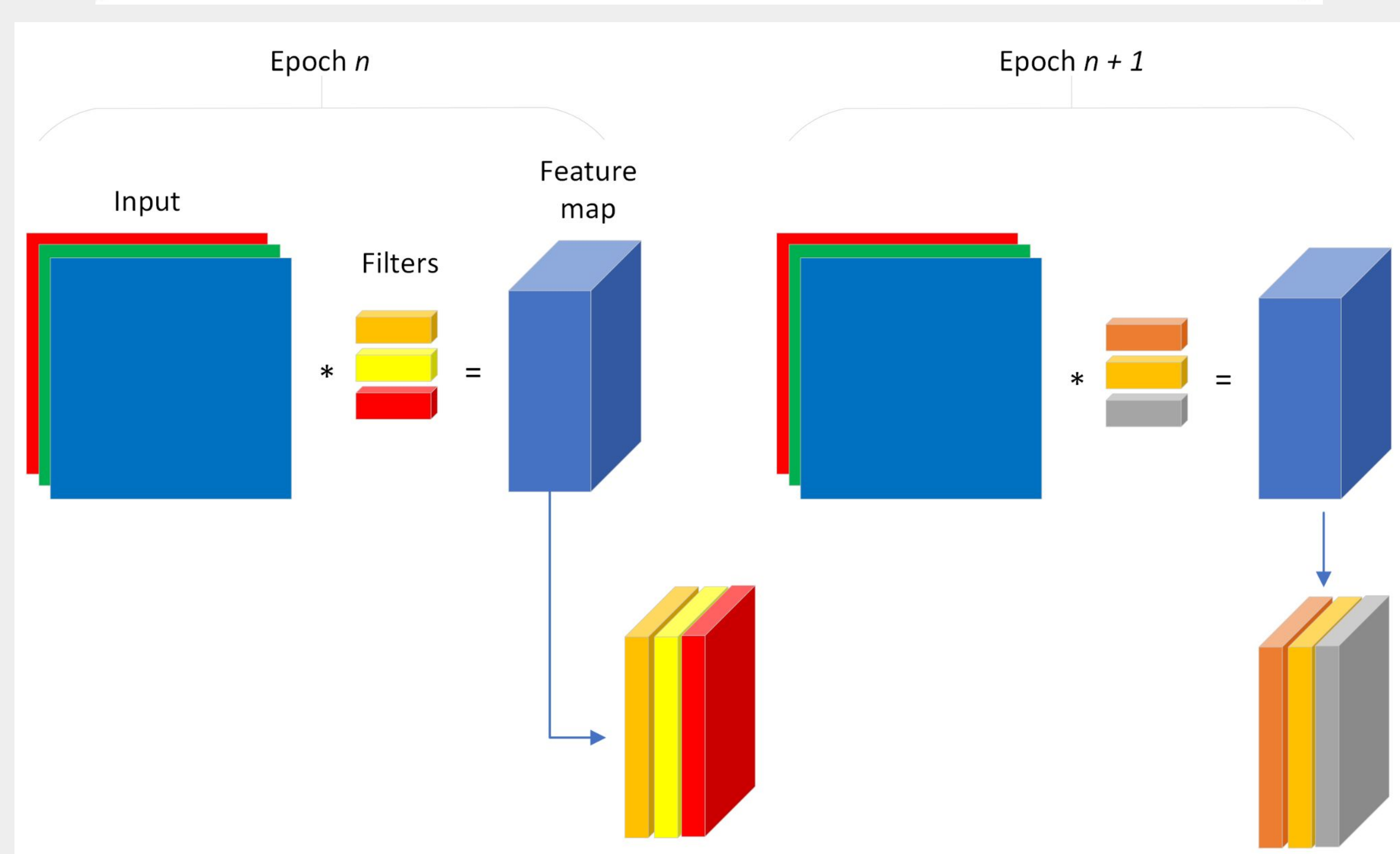

Figure 2: Visualization of ranked filters and their feature spaces

Preliminary Results

	Baseline	Dropout	Dropout w/ residue	Dropout w/ freeze	Dropout w/ freeze and residue
Accuracy	0.73	0.33	0.57	0.32	0.39
Runtime (min/epoch)	0:36	0:36	0:28	0:31	00:55
Total Runtime (min)	18:13	6:53	09:12	09:02	11:16

Table 1: Validation accuracy and runtime on Imagenette dataset

Conclusion/Open Questions

- Max activation to rank filters and/or summation of ranks across a dataset does not give desirable accuracy.
- How do we further encourage the network to not catastrophically forget?
- Algorithmic or architectural approach to fix model drift?
- Can this method be extended to other types of layers?
- How does Distributed Data Parallel (DDP) impact freezing and/or data dropout?
- Activations v.s. gradients v.s. clustering activations to approach filter rankings

Acknowledgements

This work was supported by the National Nuclear Security Administration, Minority Serving Institutions Partnership Program DE-NA0003948.

References

1. Honglak Lee, Roger Grosse, Rajesh Ranganath, and Andrew Y. Ng. 2009. Convolutional deep belief networks for scalable unsupervised learning of hierarchical representations. In Proceedings of the 26th Annual International Conference on Machine Learning (ICML '09). Association for Computing Machinery, New York, NY, USA, 609–616.
2. Jung, Hyungsik, and Youngrock Oh. "Towards better explanations of class activation mapping." Proceedings of the IEEE/CVF International Conference on Computer Vision. 2021.
3. Nguyen, Anna, et al. "Explaining Convolutional Neural Networks by Tagging Filters." arXiv preprint arXiv:2109.09389 (2021).
4. Mousa-Pasandi, Morteza, et al. "Convolutional neural network pruning using filter attenuation." 2020 IEEE International Conference on Image Processing (ICIP). IEEE, 2020.