IE 506: Course Project Endterm Review

Learning to Learn: Model Regression Networks for Easy Small Sample Learning

Team AI-nthusiasts

Om Prabhu (19D170018)
Naman Gupta (190010048)

OUTLINE

- MIDTERM PROJECT REVIEW:
  - Overview of Problem Statement
  - Work Done before Midterm Review
  - Addressing Major Comments
- MODEL IMPLEMENTATION:
  - Analysis of Dataset
  - Data Processing
  - Model Architecture
- RESULTS:
  - Model Performance
  - Comparison Against Existing Models
- CONCLUSIONS:
  - Key Takeaways & Concluding Remarks
  - Future Work

Overview of Problem Statement

- **Objective:** Using experience from already learned example classes with large data to facilitate learning of novel classes with smaller number of data samples
- **Motivation:** The authors hypothesize the existence of a generic, category agnostic transformation from small-sample models to underlying large-sample models, which is empirically validated in the paper. A major motivation is the transferability of feature extraction in deep CNNs trained on large object categories to novel small categories.
- **Background:** Over the past decade, ML techniques to deal with big data have emerged - however, data acquisition is either expensive or very tough due to its limited nature in most practical applications. The most common applications with limited training data involve visual phenomena such as object recognition & error handling by robots in natural environments (e.g., emergency response systems in self-driving cars).

Work Done before Midterm Review

- General literature review of few-shot ML techniques and transfer learning algorithms
- Detailed review of the paper given:
  - Understanding the theory and mathematical background behind the model described
  - Analysing the model implemented in the paper - data processing, model architecture, results

$$L(\Theta) = \sum_{j=1}^{J} \left\{ \frac{1}{2} \left\| \mathbf{w}_j^* - T\left(\mathbf{w}_j^0, \Theta\right) \right\|_2^2 + \lambda \sum_{i=1}^{M+N} \left[1 - y_i^j \left(T\left(\mathbf{w}_j^0, \Theta\right)^T \mathbf{x}_i^j \right) \right]_+ \right\}$$

- Analysis of previous works:
  - Tabula Rasa: Model Transfer for Object Category Detection (Aytar, Zisserman - 2011)
  - Uncovering Shared Structures in Multiclass Classification (Amit, Fink, Srebro, Ullman - 2007)
- Brief case study of the ILSVRC 2012 challenge (tasks, datasets, etc)

Work Done before Midterm Review (Contd.)

- Replication of CNN models on the ILSVRC 2012 training data:
  - Model using AlexNet architecture implemented in PyTorch
  - Same model implemented in TensorFlow with some additional data processing
- Modifications proposed to the existing model:
  - Use convolutional layers for feature extraction & training instead of fully connected layers - could potentially lead to better performance and accuracy
  - Use an alternative loss function instead of the SVM loss

Comments during Midterm Review

- Implement CNNs instead of dense layers in the classification model
  - Unfortunately, the ILSVRC 2012 dataset is too large to train a model without access to high-end computing hardware
  - We have implemented models using CNNs on 3 different alternative datasets having similar hierarchy as the ILSVRC 2012 dataset

Outline

- Midterm Project Review:
  - Overview of Problem Statement
  - Work Done before Midterm Review
  - Addressing Major Comments
- Model Implementation:
  - Analysis of Dataset
  - Data Processing
  - Model Architecture
- Results:
  - Model Performance
  - Comparison Against Existing Models
- Conclusions:
  - Key Takeaways & Concluding Remarks
  - Future Work

Analysis of the ILSVRC 2012 Dataset

- 1000 object categories - more fine-grained classes compared to the PASCAL VOC dataset (e.g. PASCAL VOC has only “aeroplane”, while ILSVRC has “warplane”, “airship”, “airliner”, “space shuttle”)

Image classification annotations (1000 object classes)

Year	Train images (per class)	Val images (per class)	Test images (per class)
ILSVRC2010	1,261,406 (668-3047)	50,000 (50)	150,000 (150)
ILSVRC2011	1,229,413 (384-1300)	50,000 (50)	100,000 (100)
ILSVRC2012-14	1,281,167 (732-1300)	50,000 (50)	100,000 (100)

Additional annotations for single-object localization (1000 object classes)

Year	Train images with bbox annotations (per class)	Train bboxes annotated (per class)	Val images with bbox annotations (per class)	Val bboxes annotated (per class)	Test images with bbox annotations
ILSVRC2011	315,525 (104-1256)	344,233 (114-1502)	50,000 (50)	55,388 (50-118)	100,000
ILSVRC2012-14	523,966 (91-1268)	593,173 (92-1418)	50,000 (50)	64,058 (50-189)	100,000

- Very computationally expensive to train a model on a local machine - we have instead used the CIFAR-10 dataset (comes with keras) and the Ants & Bees dataset by Gaurav Dutta on Kaggle

Analysis of the Datasets Used

- Ants and Bees Dataset:
  - Relatively smaller dataset - contains 398 images with labels “ant” or “bee”
  - Training set contains 124 ant & 121 bee images, test dataset contains 70 ant & 83 bee images
- CIFAR-10 Dataset:
  - Comprised of 60,000 32x32 pixel colored images with 3 channels
  - There are 10 classes, labelled 0 to 9 respectively (0: airplane, 1: automobile, 2: bird, 3: cat, 4: deer, 5; dog, 6: frog, 7: horse, 8: ship, 9:truck)
  - Training dataset contains 50,000 images and test dataset contains 10,000 images
  - Images are very low resolution, which is one of the reasons even top-of-the-line models fail to achieve significant accuracy on the dataset

Analysis of the Datasets Used

- Omniglot Dataset:
  - Contains 1,623 characters from 50 different alphabets with 20 examples for each character.
  - The dataset was created by drawing 20 samples for each character online via Amazon's Mechanical Turk.
  - Few-shot learning task involves randomly selecting k samples from n randomly chosen classes.
  - The *n* numerical values are used to create a new set of temporary labels to test the model's ability to learn a new task given few examples.
  - If training on 5 classes, the new class labels will be either 0, 1, 2, 3, or 4.
  - Omniglot is an excellent dataset for few-shot learning as it offers many different classes to draw from with a reasonable number of samples for each class.

Data Processing

- The Ants & Bees dataset is used as is
- Processing on the CIFAR-10 dataset:
  - Dataset contains 10 classes represented as unique integers
  - Each image has pixel values represented as unsigned integers
  - We hence use one hot encoding for the class element of each sample
    - Transforms the integer class label into a 10 element binary vector with 1 for the index of the corresponding class value and 0 for the other elements
  - We then normalize the pixel values
    - Pixel values are converted from integers to floats, and rescaled to the range [0, 1] by dividing by 255
- Omniglot dataset:
  - Images are shrinked and converted into a grayscale image

Model Architecture

- Model 1: (uses the Ants & Bees dataset, all layers use ReLU as the activation function)
  - AlexNet CNN architecture: 5 sequential convolutional layers for feature extraction (kernel size decreases from 11, 11 to 3, 3) followed by 3 linear layers for binary classification
  - VGG CNN architecture: 13 sequential convolutional layers for feature extraction (constant kernel size of 3, 3) followed by 3 linear layers for binary classification
- Model 2: (uses the CIFAR-10 dataset, all layers use ReLU as the activation function)
  - There are 4 models - we start with one block of 2 convolutional layers, and go on adding one block for each model
  - Adding blocks increases the complexity of the regression network and, as a result, offers a better training & test accuracy
  - For testing purposes, a new class exclusive to the CIFAR-100 dataset is introduced for retraining the network
  - Output layer is fully connected

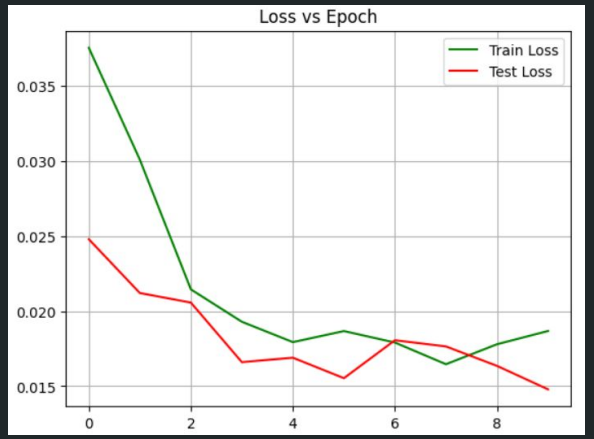
Model Architecture (contd.)

- Model 3: (uses the Omniglot dataset)
  - Approximation of MAML algorithm for few-shot learning
  - Model architecture: 4-layer CNN followed by a fully connected layer
  - Training loop: inner loop and outer loop
  - Inner loop: mini dataset sampled from full dataset using SGD with fixed number of inner iterations; cross-entropy loss
  - Outer loop: weights updated using meta-learning algorithm with multiple steps; mini dataset sampled using fixed number of inner iterations and trained on model; weights updated using meta step size, which decreases as number of meta-iterations increases; evaluation after fixed number of meta-iterations.

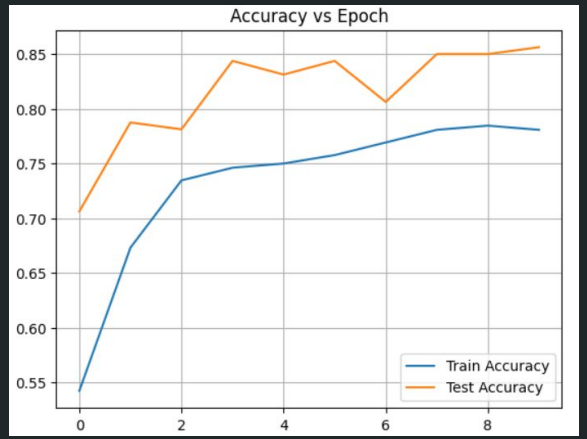
OUTLINE

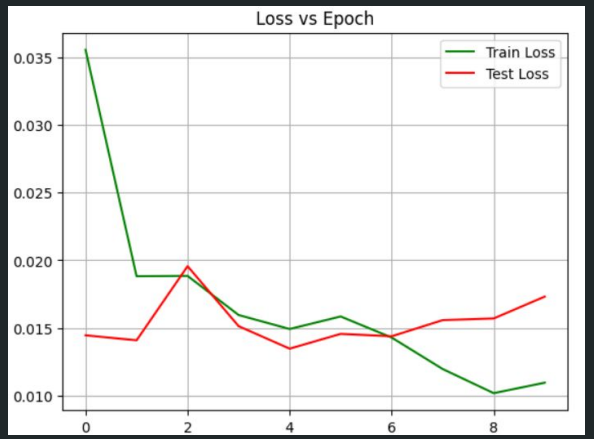
- MIDTERM PROJECT REVIEW:
  - Overview of Problem Statement
  - Work Done before Midterm Review
  - Addressing Major Comments
- MODEL IMPLEMENTATION:
  - Analysis of Dataset
  - Data Processing
  - Model Architecture
- RESULTS:
  - Model Performance
  - Comparison Against Existing Models
- CONCLUSIONS:
  - Key Takeaways & Concluding Remarks
  - Future Work

Model 1 Performance (Ants & Bees Dataset)

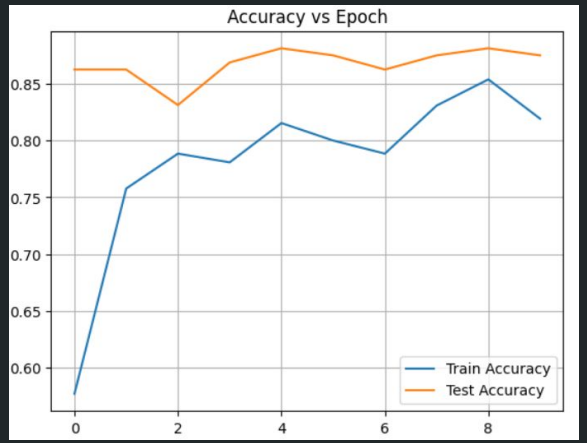


CNN model with AlexNet architecture
(test accuracy = 0.9)





CNN model with VGG16 architecture
(test accuracy = 0.95)



Model 2 Performance (CIFAR-10 Dataset)

- Test accuracies for all the 4 models:

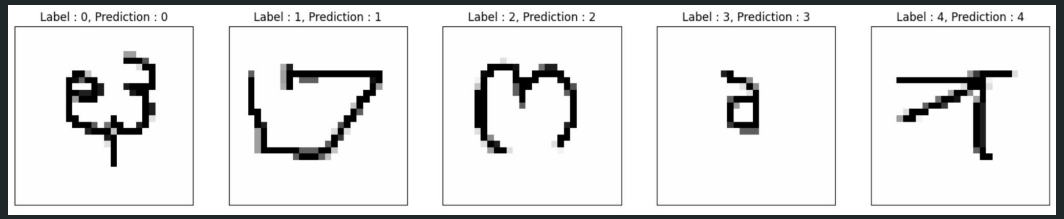
1 Block	2 Blocks	3 Blocks	4 Blocks
66.38%	70.96%	73.55%	71.54%

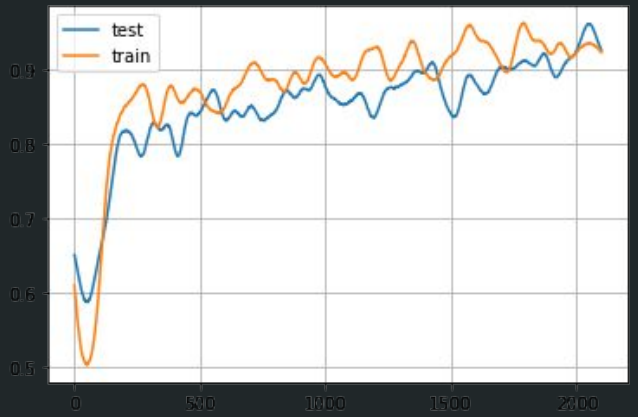
- An explanation for the decrease in test accuracy from 3 to 4 blocks of convolutional layers is that the model starts to become too complex and overfits to the training data
- Comparison against existing models:

ANODE (Augmented Neural ODEs)	60.6%
ViN (Vision Nystromformer)	65.06%
SmoothNetV1	73.5%
Hybrid ViN	75.26%

- SmoothNetV1 uses an averaging metric for the individual test scores for all the remaining classes that are exclusive to the CIFAR-100 dataset; we have tested only over the 11th class in CIFAR-100

Model 3 Performance (Omniglot Dataset)





Algorithm	1-shot 5-way	5-shot 5-way	1-shot 20-way	5-shot 20-way
MAML + Transduction	$98.7 \pm 0.4\%$	$99.9 \pm 0.1\%$	$95.8 \pm 0.3\%$	$98.9 \pm 0.2\%$
1^{st}-order MAML + Transduction	$98.3 \pm 0.5\%$	$99.2 \pm 0.2\%$	$89.4 \pm 0.5\%$	$97.9 \pm 0.1\%$
Reptile	$95.39 \pm 0.09\%$	$98.90 \pm 0.10\%$	$88.14 \pm 0.15\%$	$96.65 \pm 0.33\%$
Reptile + Transduction	$97.68 \pm 0.04\%$	$99.48 \pm 0.06\%$	$89.43 \pm 0.14\%$	$97.12 \pm 0.32\%$

Code: https://colab.research.google.com/drive/13f16KNCdEb4PwZ2p46ixcW24N_9MpSmt#scrollTo=pvh_1i94oc39

OUTLINE

- MIDTERM PROJECT REVIEW:
  - Overview of Problem Statement
  - Work Done before Midterm Review
  - Addressing Major Comments
- MODEL IMPLEMENTATION:
  - Analysis of Dataset
  - Data Processing
  - Model Architecture
- RESULTS:
  - Model Performance
  - Comparison Against Existing Models
- CONCLUSIONS:
  - Key Takeaways & Concluding Remarks
  - Future Work

Key Takeaways & Concluding Remarks

- Gained a deeper insight into few-shot learning and model regression networks for learning new models/categories from existing small sample models/categories
- Explored several deep learning architectures for image classification including AlexNet & VGG CNN models as well as dense layer networks
- Replicated the CNN model that won the ILSVRC 2012 image classification challenge (midterm)
- Demonstrated the effectiveness of transfer learning for improving the performance of image classification models
- Implemented our own modified versions of transfer learning algorithms using different CNN architectures including training and testing on extensive datasets with similar structure to that of the ILSVRC 2012 data
- Compared the performance of different deep learning architectures as well as impact of increasing complexity in convolutional network models

Future Work

There are a lot of novel applications which can be explored in the future. Some of these include:

- Emergency event detection and response: not a lot of data available due to the intrinsic rarity of emergency events in autonomous systems
- Train classifiers for unknown classes: can be used to reasonably predict the kingdom/class of unknown organisms/unexplored earth territories that have been theorized but not explored yet
- Explore other avenues such as text classification or audio processing

It is also possible to finetune the model further in the following ways:

- Use known relationships between certain groups of object categories to add some sort of correlation between train and test data and potentially improve model performance
- Improve the efficiency of process by developing methods to select the most informative samples for transfer learning

REFERENCES

- Research Papers
  - Learning to Learn: Model Regression Networks for Easy Small Sample Learning (Wang, Habert - 2016)
  - Tabula Rasa: Model Transfer for Object Category Detection (Aytar, Zisserman - 2011)
  - Uncovering Shared Structures in Multiclass Classification (Amit, Fink, Srebro, Ullman - 2007)
  - Detecting Avocados to Zucchinis (Russakovsky, Deng, Huang, Berg - Stanford, 2013)
  - On First-Order Meta-Learning Algorithms (Alex, Joshua, John - OpenAI, 2018)
- GitHub Repositories
  - longrootchen/ILSVRC-2012-classification-pytorch
  - https://github.com/tensorflow/models/blob/master/research/slim/README.md
  - NvsYashwanth/CIFAR-10-Image-Classification: CIFAR-10 Image Classification using PyTorch
  - ant_bees/ant_bees.ipynb at main · likarajo/ant_bees (github.com)