Understanding and expediting learning in Convolution Neural Networks using feature maps

Anand, Mayank
my321532@dal.ca

Bhupathiraju, Akhilesh Varma
ak445438@dal.ca

Chandrala, Rohini
rh359742@dal.ca

February 2022

1 Problem Statement

One of the most significant drawbacks of using neural networks is that they are poor at explaining reasons for the decisions made and are generally referred to as a black box. To provide insights and visualize what is happening in the model can be done using activation maps or feature maps. We get features maps after a filter is passed through the input image. The resulted features maps from prior layers could provide insights and help us accelerate learning in a convolution neural network by classifying new input images based on similarity with previous trained images. So, we plan to perform a similarity measure between the feature maps extracted from earlier trained images and the new input images. Moreover, if that matches, we will give the same class label to that new image. That implies there will be no need to pass that new image to further deeper layers of CNN on the successful classification of similarity, which might reduce the classification time. Second, for better understanding and experimenting, we might get improved classification results by training a CNN from the cached featured maps obtained on training CNN of the same architecture with data set images. In the end, we will have two separate trained models after training the model with original input images and cached feature maps. Furthermore, the results after feeding the same test set to both the models will be compared.

2 Potential Approaches

2.1 Data Description

We have chosen a pretty small and famous MNIST data set for this project to start. However, we will not be exploring only this data set. After visualizing results on the MNSIT data set will try other data set which are bigger in size to explore how our model behaves for more advanced data sets. The MNIST data set is handwritten digits from 0 to 9 written by approximately 250 writers. Also, sets of writers of the training set and test set were disjoint. This dataset contains around 60k examples for training, combining all of the digits from 0 to 9 and 10k examples for testing. The data set has been downloaded from the original website.

2.2 Data Prepossessing

For our first approach, we plan to use approximately half of the images of all classes to train our model initially. Furthermore, will use similarity measure to accelerate our learning for the reaming half. However, for our second approach, we will be using only two classes from the MNIST data set and manually label them as a different class to make it a binary classification problem.

2.3 Modelling Approach

2.3.1 Accelerating Learning using similarity measure.

Our first step is to visualize and cache featured maps. Feature maps are a set of features obtained by convolving a trained kernel on each image.

Visualizing and caching feature maps

Teow [5] in his research used feature maps for MNIST data set for understanding the model and showcased how a CNN learns after each convolution layer. We will be visualizing and caching feature maps using NUMPY [1] and Pytorch [3] the same way Nath [4] did in his blog post.

Feature Vector Extraction

Park and Kim [2] did feature vector extraction in their research. we will be performing something similar to that in which for feature maps, a feature vector will be defined as $F_i(x, y)$ of size $P \times Q$ and Energy E can be defined $E_i \sum_x \sum_y |F_i(x, y)|$ where (i) is index of feature map for that convolution layer. Form here we will calculate mean $\mu_i = \frac{E(i)}{PQ}$ and standard deviation

$$\sigma_i = \frac{\sqrt{\sum_x \sum_y (|F_i(x,y)| - \mu_i)^2}}{PQ}$$

So, feature vector is mean and S.D of features for each convolution layer as:

$$\mathrm{V} = (\mu_0\sigma_0,\ \mu_1\sigma_1,\ \mu_2\sigma_2 \ldots\ldots\ldots \mu_{i-1}\sigma_{i-1})$$

Similarity measure

For similarity measure we will be using cosine similarity to express similarity as distance between $V_{query} = U$ and $V_{calculated} = T$ and can be written as:

$$\mathrm{D}(U, T) = 1 - \frac{\sum_i (U_i \times T_i)}{\sqrt{\sum_i (U_i)^2} \times \sqrt{\sum_i (T_i)^2}}$$

2.3.2 Training Model using feature maps

As a part of experimentation, we would like to consider images of two numbers from the MNIST data set and take all the feature maps obtained after each convolution layer. We will then pass these feature maps as input images to a new model with the same architecture, and this new model is tested against the original images taken from the MNIST data set for two numbers.

3 Anticipated Challenges

In this approach, the images will not be passed through CNN layers if a match is between the feature map extracted from the trained images and the current new image. As a result, this approach might make the CNN either over-fit against the training data or under-fit against the overall data. However, the techniques like Regularization can be applied in case of over-fitting. Another major challenge could be finding the right set of parameters to tune the model. In the experimentation of feature map training, it is quite unsure of how the model would train and what the end results would be. However, we are looking to derive the reason for the model behaviour.

4 Project Timeline

Project Timeline (Feb19-March31)

	Week 1	Week 2	Week 3	Week 4	Week 5	Week 6
Akhilesh	Data Munging Data Validation	Finding and fitting the right Parameters	Training dataset using similarity-based measure	Finding Insights from the plots of various feature maps	Prediction of test data on Model -2	Documentation Video Presentation Poster Preparation
Mayank	Data Transformation Data Exploration	Designing Architecture	Extracting Feature Maps from the layers and saving	Training the feature maps using the 2nd Model	Prediction of test data on Model -1	Documentation Video Presentation Poster Preparation
Rohini	Data Visualization	Coding the Model, Model Fitting and Rerunning	Fine tuning the model parameters	Designing 2nd Model Architecture	Calculation of metrics and Comparison of results on Validation and Test data	Documentation Video Presentation Poster Preparation

Week 1	Data Processing
Week 2	Model Training
Week 3	Model Fine Tuning and Feature Maps Extraction
Week 4	Model Understanding and Feature Map Training
Week 5	Prediction on Unseen Data
Week 6	Documentation and Poster Preparation

5 Future Scope

This idea can be further extended in the music domain where Mel spectrogram, Chromagram can be generated for each audio clip, and a convolution neural network can be built in the same way. After training the network, the feature maps can be obtained from the final convolution layers, and a similarity measure like cosine similarity can be applied to these feature vectors to help find similar music. So, now based on this similarity score, analogous music can be recommended to a user.

References

[1] Charles R. Harris, K. Jarrod Millman, Stéfan J. van der Walt, Ralf Gommers, Pauli Virtanen, David Cournapeau, Eric Wieser, Julian Taylor, Sebastian Berg, Nathaniel J. Smith, Robert Kern, Matti Picus, Stephan Hoyer, Marten H. van Kerkwijk, Matthew Brett, Allan Haldane, Jaime Fernández del Río, Mark Wiebe, Pearu Peterson, Pierre Gérard-Marchant, Kevin Sheppard, Tyler Reddy, Warren Weckesser, Hameer Abbasi, Christoph Gohlke, and Travis E. Oliphant. Array programming with NumPy. *Nature*, 585(7825):357–362, September 2020.

[2] Keunyoung Park and Doo-Hyun Kim. Accelerating image classification using feature map similarity in convolutional neural networks. *Applied Sciences*, 9(1):108, 2019.

[3] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems 32*, pages 8024–8035. Curran Associates, Inc., 2019.

[4] Sovit Ranjan Rath. Visualizing filters and feature maps in convolutional neural networks using pytorch, 2020.

[5] Matthew YW Teow. Understanding convolutional neural networks using a minimal model for handwritten digit recognition. In *2017 IEEE 2nd International Conference on Automatic Control and Intelligent Systems (I2CACIS)*, pages 167–172. IEEE, 2017.