Exploring Class Ordering Heuristics for Incremental Learning

Séraphin Bassas  
Department of Computer Science  
McGill University  
845 Rue Sherbrooke O  
Montréal, QC H3A0G4  
seraphin.bassas@mail.mcgill.ca

Jugal Kalita  
Department of Computer Science  
University of Colorado at Colorado Springs  
1420 Austin Bluffs Pkwy  
Colorado Springs, CO 80918  
jkalita@uccs.edu

Abstract

Incremental Learning research is principally focused on overcoming the challenge of “catastrophic forgetting” to increase model accuracy. However, in this endeavor of increasing model accuracy, one factor that is almost consistently overlooked is the way training data is ordered with respect to their classes. Although the entirety of a dataset is always available to the authors before training, random or seeded orderings are typically used, but this is problematic when trying to reproduce studies and peak accuracy results. We bring to the table a method to sort classes deterministically such that the order in which they are presented to the IL algorithm will, on average, consistently yield greater mean accuracy than random orderings do. Thus, through this novel ordering heuristic, we provide the computer vision community with a tool to perform more easily reproduced and benchmarked results, as well as a trick to boost model accuracy by as much as 5%.

1. Introduction

When AlexNet (Krizhevsky, Sutskever, and Hinton 2012) emerged in 2012, the Computer Vision, NLP, and Neural Machine Translation communities, to name a few, shifted gears. As mentioned in (Can and Ezen-Can 2020), while before scientists used to spend their time crafting features to extract information from images, documents, and other data types, the bursting of deep neural networks into the limelight has shifted the paradigm towards tuning the networks’ many parameters. As a direct consequence, substantial progress has been made in engineering more efficient models and parameters since then.

Nevertheless, one of the “parameters” that has been underwhelmingly addressed in the literature is the tuning of the order in which data is being fed to the networks to maximize accuracy. In the literature, the relationship between the ordering of data, classes, or tasks and network accuracy is poorly qualified and/or quantified. Despite inadequate understanding of this relationship, the notion that the order of the data may have an effect on classification is not new. In fact, (MacGregor 1988; Giraud-Carrier 2000; Wenzel and Hotz 2010) dabbled with these concepts in the late 80s, early 2000s, and early 2010’s respectively. While the problems these authors tried to solve were very different, (Wenzel and Hotz 2010) provides a formal definition of the notion of ordering sensitivity which varies across classification algorithms. What’s more, the author establishes a measure to estimate this sensitivity which is computed as the variance of a network’s accuracy when trained across randomly picked data orderings.

While descendants of AlexNet now excel at the ILSVRC competition, more challenging problems such as Incremental Learning (IL) are being addressed. IL experiments often use the same Imagenet dataset (Deng et al. 2009) that AlexNet won ILSVRC-2012 on. In IL, a network is trained on an initial set of classes, and at each incremental step thereafter, the model keeps getting retrained on data from different classes without previous data available. As such, one can see how data or class orderings are particularly relevant to the IL problem.

With that in mind, we turn the attention towards Wenzel and Hotz (2010), who brought unto the computer vision community the intuition that similar classes should be introduced in proximity to each other or according to some measure of relatedness. This idea is inspired from the way humans learn in school for instance, where the incremental learning of new concepts is timed precisely through carefully designed curricula. However, the opposite intuition can also be defended wherein classifiers will more easily classify classes that are further apart in feature space. For instance, linear classifiers such as support vector machines (Drucker et al. 1997) have more ease drawing an accurate hyperplane between two classes that are further apart in feature space.

Regardless of which approach is the “correct” one, the proposed research aims to investigate how both these ordering strategies can be fitted to various heuristics for class ordering applied to the incremental image classification setting.

To summarize, we seek to address the gap in the literature regarding the effects of class orderings on model accuracy in the IL setting. Our contributions are the following:

1. Creating diverse class ordering heuristics inspired from various learning philosophies,
2. Finding a heuristic that consistently yields higher model accuracies independent of model, dataset, or hyperparameters,
3. Starting a discussion about the way data ordering is used in comparison studies and state-of-the-art papers in IL and
within training batches has a significant impact on network accuracy regardless of learning rate, batch size, or model. The authors provide statistical analyses to demonstrate the significance in accuracy disparities across different data orderings on Imagenet, but do not provide any insight into generating favorable orderings.

2.4 Class Ordering

An early approach before the aforementioned “deep learning boom” to order classes for incremental classification was based on finding the distance between classes. They used Baye’s error as a measure of distance and then ordered the classes on the basis of which ordering minimized that distance. The most relevant work to our proposed approach is that of Masana, Twardowski, and Van de Weijer (2020) in which they investigate the effect of class orderings on the accuracies of state-of-the-art IL models. They obtain a fitness score for various class orderings which is computed by taking the trace of the matrix multiplication of a weight matrix by the confusion matrix obtained after model training and testing. In their approach, each weight matrix represents a different heuristic for class ordering, and the authors use a simulated annealing algorithm to generate class orderings that will maximize a fitness function for a given heuristic. That being said, the authors did not find a heuristic that consistently outperformed random class orderings across models. Occasionally, mention of class orderings can be spotted in IL papers (Rebuffi et al. 2017; Castro et al. 2018). Rebuffi et al. (2017) the authors use a shuffled version of the data that is seeded so that it’s always the same, and Castro et al. (2018) emphasize the fact that for fair comparison of IL model accuracies in experiments, the same ordering of data needs to be maintained on different models.

3. Approach

Our overarching goal is to find a heuristic for class ordering that consistently yields greater or equivalent average classification accuracy relative to random class orderings. We tested three classes of heuristics, each with two versions engineered towards the opposing learning strategies mentioned above:

1. Introducing most visually similar classes will yield best results
2. Introducing most visually distinct classes will yield best results.

In total, we have six different heuristics, each with their own minor variations depending on distance measurements and image type (greyscale vs. RGB).

3.1 Distance Learning

The first heuristic that we call distance learning (DL), is a greedy heuristic. DL computes the exemplar for each class, or the pixel wise average of all the images belonging to the same class. Subsequently, the distances separating each exemplar from one another are computed and stored in an adjacency matrix. The particular distance measure depends on...
the number of channels in the input images: if the images are greyscale, the pixel-wise euclidean distance is used and we leave it at that. However, if the image has RGB channels, two scenarios exist. We first tested the flat method, wherein we take the mean of the three channels at each pixel for each image in the given class, and then perform regular euclidean distance between class exemplars. This is more or less what we take the mean of the three channels at each pixel for each image in the given class. Moreover, instead of the traditional euclidean distance, we take the sum of the pixel-wise values (layer). And then the min and max keywords represent whether the ordering of classes is based on the minimum or maximum distance separating class exemplars, which represents the two intuitive learning strategies mentioned earlier.

3.2 Cluster Learning

The second and third heuristics we present are less conventional—although one could argue convention isn’t so much the problem in such a sparse field—as we introduce heuristics that create tasks with variable amounts of classes per increment. Similar to the first heuristic, the second and third heuristics also compute the exemplars for each class, which are then passed through a feature extraction routine to obtain their vectorized feature representations. Subsequently, the feature vectors of the exemplars are clustered, in our case using mini-batch kmeans (Sculley 2010), and the centroids of each cluster are evaluated. Afterwards, the cluster-wise distances are computed and stored in an adjacency matrix.

The second heuristic, termed distance clustering, is similar to DL but with clusters. We define the initial set of classes for training as those found in one of the two clusters with the greatest distance separating them, found as the maximum entry in the adjacency matrix. The classes in the cluster that was not picked as the initial set then become the set of classes comprising the second increment. Finally, we keep referring to the adjacency matrix to find the next closest cluster to the last one picked in the ordering, and define the following increment as the set of classes composing that cluster.

In the third heuristic, termed scattered (or scattered distance clustering), we also compute the maximum distances separating the centroids of each cluster, but we don’t define the clusters themselves as the classes in a single cluster. Instead, the first increment is composed of one randomly picked class from each cluster. Therefore, the first increment contains as many classes as there are clusters defined in the algorithm. The second increment contains randomly picked classes from all non empty clusters left. We repeat the process until all clusters are empty. It is worth noting that the number of classes contained in consecutive increments either stays constant or decreases using this heuristic.
4. Implementation

4.1 Datasets

Testing was performed on common baseline datasets for easy comparison to previous work and as well future comparison in this field. Further, we chose datasets of varying scales to put the scalability of our technique to the test. We chose two greyscale image datasets, the MNIST digit (Le-Cun ) and MNIST fashion datasets (Xiao, Rasul, and Vollgraf 2017), and two RGB image datasets, CIFAR10 and CIFAR100 (Krizhevsky, Hinton, and others 2009). All datasets have 10 classes, except for CIFAR100 which has 100.

4.2 Models

We experimented with three main models: one for digit MNIST, another for fashion MNIST, and finally a VGG16 model (Simonyan and Zisserman 2014) for CIFAR10&CIFAR100. The first model for MNIST is a simple neural network taken from the Keras “Simple MNIST convnet” and we used the Adam (Kingma and Ba 2014) optimizer with categorical cross-entropy loss. For the fashion MNIST model, we use a model with 3 convolutional layers, each with 256 units, Relu activation, 3x3 kernels, and “HeUniform” kernel initializers (He et al. 2015). These layers are followed by a max-pooling layer, a flattening layer, a dense layer with 75 units, and a softmax classifier. This model has the same optimizer and loss function as the one above.

The VGG16 model is loaded from the Keras package with Imagenet weights and followed up by a global max-pooling layer, a dropout layer, and a dense layer with 2048 units, and a softmax classifier. For all results presented in this paper, we train the models for 50 epochs at each increment without freezing any weights (for VGG16). We also used an EarlyStopping callback with a patience of 15 and minimum delta of 0.001.

5. Results

In figures 1 and 2, we present experimental data from both CIFAR10 and CIFAR100. For both datasets, each data point represents values averaged out over 4-6 replicate experiments. In the case of CIFAR10, each increment, including the first one, saw the addition of one extra class at each time point. For CIFAR100, the model was trained on 10 initial classes, and each increment saw the addition of 5 new classes. Different combinations of numbers of classes introduced at the first and all subsequent increments were experimented with, and yielded similar results.

In Figure 1, the blue bars show the average accuracy over all increments for all the distance heuristics presented in this paper. We see that for CIFAR10 (Figure 2a), the accuracy of the random heuristic is inferior only to the semantic and flat-max heuristics by 2.43% and 3.25% respectively. Similarly, for CIFAR100 (Figure 2b) the random heuristic has a lower accuracy than the semantic, flat-max, and layer-max heuristics by 0.05%, 5.24%, and 3.44% respectively. Thus we conclude that flat-max is the only heuristic so far that consistently gives the network higher accuracy than the random heuristic. Further, from data on the two MNIST datasets and

---

```
Algorithm 3: Min_Distance_Clustering

Inputs: dataset D with set of classes \( \mathcal{C} = \{C_1, C_2, ..., C_n\} \)
\( k \) =number of clusters

1. exemplars ← Compute_Exemplars(C)
2. feature_vectors ← retrieve feature vector for each exemplar from model’s penultimate layer output
3. predictions ← output predictions of clustering algorithm on feature_vectors with \( k \) clusters
4. exemplars_per_cluster ← array containing at each index, all the exemplars \( \in i^{th} \) cluster
5. classes_per_cluster ← array containing, at each index, all the classes \( \in i^{th} \) cluster
6. centroids ← array containing all the centroids computed for each cluster
7. for \( i, j = 0 \) to \( k \) do
8.     distance_adjacency_matrix[i][j] ← euclidean_distance(centroids[i], centroids[j])
9. end
10. first_cluster_index ← one of the two indices for the min distance entry in distance_adjacency_matrix
11. cluster_order[0] ← first_cluster_index
12. while cluster_order.size(!= k) do
13.     append cluster\( _y \) to cluster\( _o \), where cluster\( _y \) not in cluster\( _o \) & cluster\( _y \) nearest to cluster\( _o \)[−1]
14. end
15. if scattered then
16.     while classes_per_cluster not empty do
17.         increments[i] ← remove one class from each non empty cluster in classes_per_cluster
18.     end
19. else
20.     increments[i] ← ith cluster in cluster_order
```
the two CIFAR datasets, we observe that the more complex the problem becomes (number of channels per image, number of images per class, number of classes, etc.) the greater the margin between the random heuristic and the flat-max heuristic grows.

In Figure 2, the plots show the accuracy of the six heuristics at each incremental step of the IL experiment. Firstly, one of the more drastic observations to make is by how much the minimum distance heuristics under-perform in comparison to the other heuristics, and, moreover, the consistency of this drop in accuracy from the very first increments. Secondly, the inconsistency of the accuracies in figure 2b, even for flat-max, shows that there is still room for much improvement when it comes to designing better heuristics, as those initial increments are the ones that provide the largest increases to the per approach mean accuracy.

6. Discussion

In this paper, we qualify and quantify the relationship between the ordering of classes and the accuracy of IL algorithms while finding a heuristic that lets us obtain accuracy boosting class orderings that generalize across datasets, models, and hyperparameters. In the results section above, we train the aforementioned models using an IL mechanism described as joint training in (Li and Hoiem 2017). Thus, it is important to raise the issue that the effects of these ordering heuristics may not translate equivalently to actual IL algorithms where data from previous classes is scarce or completely absent. Nevertheless, as joint training is often used as an “upper bound” on the performance that IL algorithms can potentially reach, it is quite promising to show that our heuristic works on this baseline.

On a different note, the accuracy trends of the heuristics tested on CIFAR100 reveal volatile behaviors, even for our most efficient heuristic. This alone serves as motivation to seek improved heuristics that will yield even greater boosts in accuracy by smoothing the curve, and also begs the question about the scalability of ordering heuristics as only the min-distance heuristics had relatively constant accuracies throughout. Finally, our findings put into question the way data orderings have traditionally been taken for granted as negligible factors in the experimental IL setup. We want to bring attention to this problem by showing that there does indeed exist ordering heuristics that grant an edge over others. Therefore, we would like to invite the community to revisit previous works where ordering was overlooked in comparison studies and even to boost peak accuracy.

7. Future Work

First and foremost, our future works includes a statistical analysis to investigate the significance of the accuracy boosts and decreases provided by the ordering heuristics. Likewise, as mentioned in an earlier section, our study still lacks rigor in comparing the effect of data orderings for various incremental learning strategies as carried out in (Masana, Twardowski, and Van de Weijer 2020). We seek to test the effect of all the heuristics presented in this paper on the following three algorithms: Lwf by (Li and Hoiem 2017), iCaRL by (Rebuffi et al. 2017), and LUCIR by (Hou et al. 2019). Additionally, we would like to test the scalability of the flat-max heuristics and others on heavier duty data sets like tiny Imagenet and Imagenet1000. Incidentally, these experiments would also give us the opportunity to investigate if any heuristics based on the semantics of image datasets such as the inherited hierarchical relationships from the WordNet dataset, have any meaningful impact on IL algorithm accuracy. Finally, following our finding that minimum distance heuristics actually significantly reduce model accuracy, we would like to investigate how class ordering may be used as an alternate strategy to carry out adversarial attacks.

8. Acknowledgements

The work reported in this paper is supported by the National Science Foundation under Grant No. 2050919. Any opinions, findings and conclusions or recommendations expressed in this work are those of the author(s) and do not necessarily reflect the views of the National Science Foundation.

References

Figure 2: Comparing mean accuracies obtained at each increment for the different class ordering strategies.

(a) CIFAR10

(b) CIFAR100