Simulating Human Learning Effects in Convolutional Neural Networks

Anisha Bhogale (abhogale6@gatech.edu)
Georgia Institute of Technology

Josh Cabral (jcabral9@gatech.edu)
Georgia Institute of Technology

Tianyi Liu (tliu422@gatech.edu)
Georgia Institute of Technology

Naum Markenzon (nmarkenzon3@gatech.edu)
Georgia Institute of Technology

Abstract

The ordering of material during human learning impacts how well that material is learned. This has been shown in the cognitive science literature, where many studies have demonstrated that spaced practice leads to more durable learning than massed practice (Rohrer & Taylor, 2007). Similarly, expanded practice leads to better learning than massed practice (Landauer & Bjork, 1978). During a testing period post-learning, spaced and expanded learners performed significantly better than massed learners. We investigate whether this finding holds in a Convolutional Neural Network (CNN). For the model, we will orchestrate the training examples so that the network is either trained in a massed, spaced, or expanded sequence. We expect the model to show better accuracy when trained on the spaced or expanded sequences versus the massed sequence. A positive result would have major implications in both the human learning and machine learning fields. Future research might explore a dynamically organized training set such that examples that have high error in earlier epochs are oversampled in later epochs.

Keywords: human learning; machine learning; learning effects; massed; spaced; expanded; sequencing; dataset; convolution; neural networks; cnn; deep learning;

Introduction

Psychology Review

The studying of learning effects in humans is not new; Ebbinghaus first researched the phenomenon in 1885 with his seminal, self-conducted study that displayed that distributed learning was more beneficial than learning in one session (Ebbinghaus, 2013). Since then, many other studies have shown the same effect: spaced learning results in better performance than massed learning (Landauer & Bjork, 1978, Grote, 1995, Rohrer & Taylor, 2007). To clarify the difference between these practices, we refer to Rohrer & Taylor's study. Their work focused on the learning of math concepts. They refer to massed learning as "overlearning" and mimic the practice by telling subjects to cycle through a list of words five or ten times. They find cycling ten times is more helpful in a more immediate test, but after four weeks, the retention rate is abysmal for both conditions. For spaced learning, the amount of spacing was varied to find the perfect balance, which had similar results to the previous studies: spaced learners performed significantly better than massed learners in a test one week after learning (Rohrer & Taylor, 2007). We also explore the "expanded" learning effect. This is an extension of spaced learning in which each item of a target problem is repeated in longer intervals. Expanded practice also improves test performance. In fact, it performed better than both massed and spaced practice (Landauer & Bjork, 1978). Figure 1 shows the difference between the three types of practices we will be investigating.

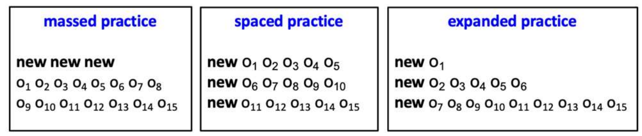

Figure 1: A visualization of the difference between massed, spaced, and expanded practice.

Machine Learning Review

The first artificial neural neurons were created in 1948 by McCulloch & Pitts in which a mathematical model they established was able to imitate the functions of biological neurons (McCulloch & Pitts, 1948). Neural networks formed from combining neurons together to form layers, and the fully connected layer was created (every input neuron is connected to every output neuron through a weighted connection and a bias). These neural networks paved the road for more complex networks such as convolutional neural networks (LeCun, 1998), which used different types of layers such as convolutional and average pooling layers. Convolutional layers are layers which have very little parameters compared to fully connected layers and only have parameters for the kernel/s which is an m x n filter. The kernel slides across the image, and calculates a convolution/cross-correlation with the image which creates an output feature map. These convolutional neural networks (CNNs) were excellent at image recognition, and the first CNN was LeNet (LeCun, 1998). LeNet was able to differentiate between different handwritten digits and determine what digits they are. However, since LeNet, computational power became much more cheap and accessible, so the size and complexity of these CNNs grew with it, and networks have been created with dozens of layers. As a result, CNNs are able to solve many problems in AI such as object detection, image segmentation, semantic segmentation etc.

Our Study

While there are many studies showing these learning effects in people, there is a lack of literature modeling the effects through neural networks. In the rest of this paper, we explore whether these learning effects can be modeled using Convolutional Neural Networks, or CNNs. We predict that the CNN will have the best accuracy for expanded practice, with spaced and massed practice following.

Method

Dataset

CIFAR-10 is a popular image dataset that is widely used as an introductory or benchmark dataset for modern computer vision systems. We decided to use CIFAR-10 for a number of different reasons. 1) CIFAR-10 contains 10 different classes. This helps us because it makes it much easier to create "filler" content for spaced, massed, and expanded learning while we focus on a singular target class. The additional classes to learn helps us to more realistically simulate a human learning experience. 2) All the classes are mutually exclusive meaning that there is no overlap between them. 3) CIFAR-10 has 5,000 training examples for each class and 1,000 test images, which provides us with an abundant amount of data at our disposal to potentially work with. Additionally, Keras (our machine learning framework) has a convenient, readily available download for CIFAR-10 so we could use the images directly from their library, which contributed to the ease of use when developing our model. The images are relatively simple with 3 color channels and a height of 32 and a width of 32. Even though this is smaller than the size of the original training, our model is able to be adjusted to a different sized input. Example images of this dataset can be seen in Figure 2 below.

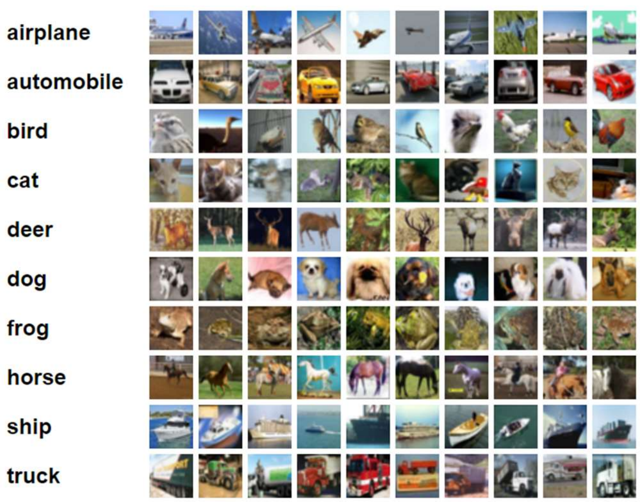

Figure 2: Sample images from CIFAR-10, an image dataset consisting of ten classes of everyday things.

Model

The model that we are focused on is the VGG-16 model which takes our (32, 32, 3) image and gives a prediction of 10 different classes. This model consists of 13 convolutional layers, 5 max pooling layers, and 3 dense layers. VGG-16 can take images of 256x256 but it can be adjusted to take images of any size and in our specific use case, images of 32x32 which is the size of the images in CIFAR-10. Training convolutional layers for large convolutional neural networks is a very computationally laborious process which requires weeks of training. In a normal computer vision CNN model, all the filter layers and dense layers would need to be trained from scratch.

Instead, we decided to use a pre-trained convolutional neural network, more specifically VGG-16. We take the outputs from the convolutional part of the neural network, after all convolution and max-pool layers as shown in Figure 3, and use them as features to train another smaller neural network that will classify the output. There are three main important reasons to use a pre-trained convolutional neural network: 1) The main motivational intuition here is that using a pre-trained CNN allows us to simulate human sight by directly working with extracted, high-level image features that could abstractly represent concepts such as eyes, legs, and wheels. 2) Training the lower layers of the CNN network manually is computationally expensive and also heavily increases the dimensionality of the resulting model, due to the number weights in the filter layers that would need to be trained. By using a pre-trained network we can reduce the dimensionality from almost 15 million in the full VGG-16 network to just the weights in our dense layer (around 33,000), a difference of a factor of 440x. 3) Training the earlier layers of the CNN would take multiple epochs to train properly and therefore would expose the model to the dataset many times. In order to best simulate spaced, massed, and expanded learning, it is preferable to not conduct an excessive amount of unnecessary training by instead working directly with valuable image features as explained in 1). This would also allow us to train our classifier at a reasonably fast rate.

As a result, we will reuse the convolutional layers in VGG-16 which will allow us to train the model significantly faster than training the model from scratch. In our training, we freeze the weights in the convolutional layers but we remove all of the dense layers and replace them with two untrained dense layers. Our dense layers consisted of a fully-connected neural network with one hidden layer of size 64, followed by a final softmax layer of size 10, which corresponds to each of our ten classes. We then use our various learning strategies to train these dense layers without changing the frozen convolutional layers. As stated before, we are using the Keras machine learning framework for python. We also standardized our learning rate hyperparameter lr = 0.001.

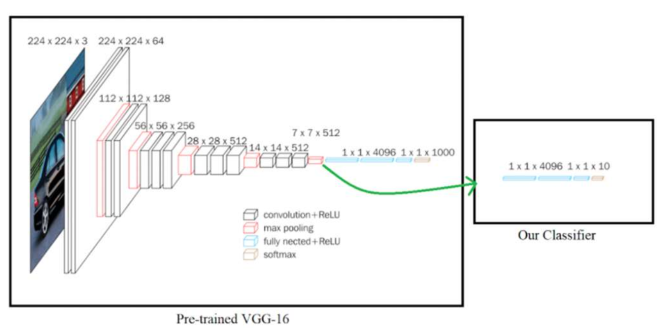

Figure 3: A visualization of the VGG16-based architecture that we used for our experiment. Our classifier is trained using the image features taken from the pre-trained VGG-16.

Training Strategies

In order to simulate the specified human learning effects in the convolutional neural network, we created 4 different sequences of the VGG data set and a random baseline. These sequences are Massed, Spaced, Expanded, and Expanded with variable weights, which we created based off of the literature. For each training strategy, we generated a dataset with approximately 50,000 randomly sampled images from the CIFAR-10 dataset. The distribution of these datasets depends on the sequencing technique being used. For each strategy, we generated a separate dataset for each class. This was to ensure that every class had a chance of being the class that we focused on. For each class, we trained a new model for a single epoch. This is because having multiple epochs would invalidate the sequencing approach that we are using and make it difficult to train. Because we are only training with one epoch, we do not expect the performance accuracy for the model to be anywhere near that of top-performing CNN models, but we decided that it would be sufficient to observe any contrast between the different learning method models. We also began by using a batch size of 1 which made it very difficult to train because it would prevent the dataset from fitting inside of the available ram as well as it trained approximately 32x slower than training with a batch size of 64.

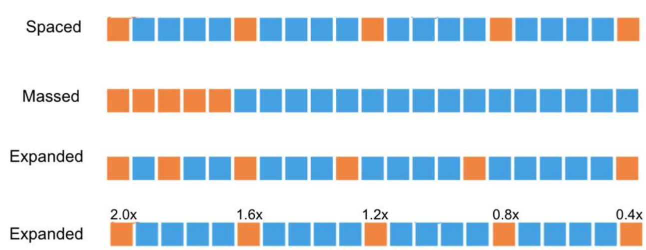

Figure 4: These are the 4 sequencing strategies that we are exploring in this project, with orange representing our target class and blue representing a non-target class. This image is meant as a visual aid, so exact numbers, sequences, and weight values from the aid are not the ones necessarily used in our models.

Baseline

The Baseline model is a "control" model in a sense. This model had the same parameters as the other models but used a completely unsequenced training and testing set. This was effectively the CNN run on the entire CIFAR-10 dataset.

Massed

The Massed sequencing of the dataset simulates the learning of all target examples first and then the rest of the examples. To sequence the dataset like so, we first separated all the training examples of the target class. Next, we randomly shuffled the examples of the nine remaining classes. We combined all the data again into a new dataset containing the 5,000 target examples and then 45,000 random other examples in sequence, such that the 5,000 target examples were all used to train our model first, followed by using the 45,000 random other examples.

Spaced

The Spaced sequencing of the dataset simulates learning by spacing the examples of the target class with interspersed randomized subsets of the other classes. In order to do so, we separated all the training examples of the target class. Next, we randomly shuffled the examples of the nine remaining classes. We combined all the data again into a new dataset containing the 5,000 target examples (each spaced 10 examples away from each other) and then 45,000 random other examples in between the examples of the target class. Our CNN was fed this new dataset for every target class.

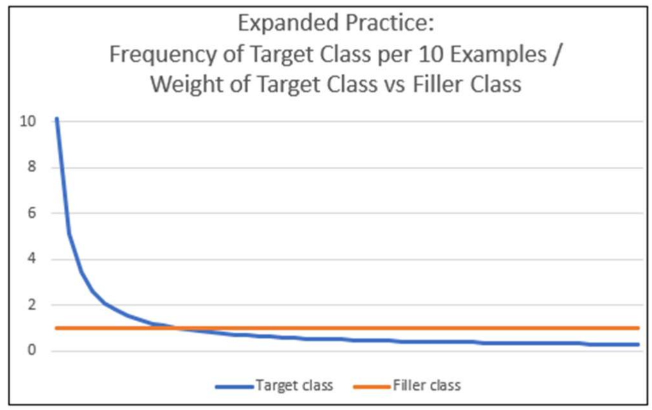

Figure 5: A visual representation of the order of sequencing in our expanded practice implementation. The x-axis represents the frequency of the target class in relation to the training series, and the y-axis represents both the frequency of the target class per 10 examples, in our expanded learning model, and the weight of the target class, in the expanded learning with weights model.

Expanded

The Expanded sequencing of the dataset begins by showing the model several instances of the class that we are focusing on. Then, as the model sees more data, we reduce the rate by which the model sees the specific class that we are focusing on. The goal is to replicate how humans learn by Expanded Practice in the Neural Network model. In order to create this sequencing of data, we use a double for loop which enables us to control how the distribution of the target class changes over time. The outer loop controls the number of spaces between the target class and the other class. In our implementation, we use a range from 0 to 50. This means that for our sequencing, we start off with a space of 0 between each target class instance and end with a space of 50 between the target class instances. The resultant frequency for the target class is shown by the blue line in figure 5. For each group of spacing, we have 1024 data points distributed as described above. We use 1024 because of the ease of use with batching which is extremely important when considering the training of a convolutional neural network.

Expanded with Weighted Sampling

In addition to expanded sequencing, we also explored an alternative way to represent expanded practice. The key intuition behind expanded learning in humans is that the target example is exercised when the ability to successfully retrieve that example is at its most vulnerable. In addition to representing expanded practice by sequencing the training data (as in the section above), we also attempt to represent expanded practice by adjusting the weights for each individual target class training example. In machine learning literature, training weights are commonly adjusted in underrepresented classes in order to emphasize their importance. We use training weights in a different way. In our model, the training examples are sequenced exactly like that for our spaced learning model, but the sample weight for each individual of the target class is steadily decreased hyperbolically over time. The intuition is that the decreasing individual weight on the target represents an increasing difficulty in retrieval. For example, for the target class, the weights for each individual example start at 10 times that of the non-target class weight, and slowly diminishes until 0.2 times that of the non-target class. This also mimics the expanded learning effect from the previous expanded learning model because a weight of 0.2 represents the 1-in-50 frequency of training examples at the end of the previous expanded learning model. A constant weight of 0.1 was added to all the weights for the target class in this model in order to equalize the combined average training weight to be consistent with those of our previous models. This is done to ensure that each class received an equal amount of weight to its training process so that no model received an inherent advantage to its training by taking advantage of additional training weights. In contrast, the weight for the non-target class remains at a constant 1x for all examples. Figure 5 gives a visual representation of the weight put onto each training example of the target class. The blue line and orange line represent the training weights for the target class and filler class, respectively.

Testing Strategies

We reason that some classes in CIFAR-10 may potentially be easier to learn than other classes. In order to account for this potential disparity, we trained a model once per class, for a total of ten separate models for each of our learning methods, which gave us a more comprehensive analysis to work with. We split the testing data into each class and then individually ran the model on each class. This generated the 10x10 heatmap charts as shown in the Results section below in Figures 6,7,8, and 9. All of our results are generated by applying our trained models to the test dataset, which consists entirely of examples that were not used in training, which is a standard practice in machine learning literature. From a psychological perspective, this is analogous to working on sample practice problems before taking an exam. From a machine learning perspective, we use a separate testing dataset to avoid our models overfitting towards the test dataset.

Results

We use heatmaps to represent our results. In each of the heatmaps, the columns represent the target class for a particular training model, where the rows represent how the model performed for each specific class. The heatmap values represent validation accuracy for a classification of a particular class. The values along the diagonal therefore represent the effect of the learning methods that our system attempts to model, and so they are the values of particular interest in our analysis.

Results for Baseline

As this was simply the entire dataset run on the model, the result was the accuracy of the entire test set, which came out to be 0.598.

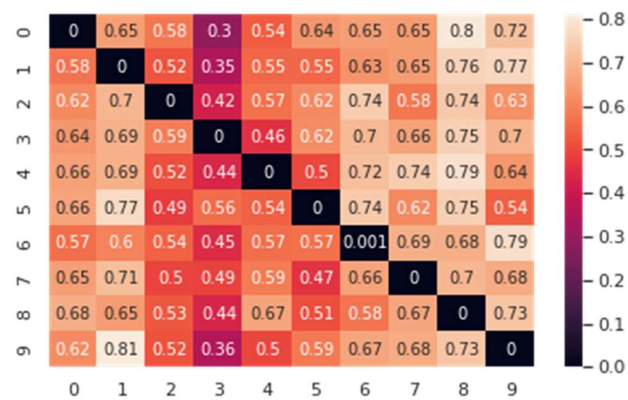

Figure 6: A heatmap of the results from the model that was trained using massed sequencing of data.

Results for Massed Learning

The massed practice condition shows a clear 0.0 accuracy down the diagonal, which indicates that the model could not accurately classify the target class. Since the model did not see any instances of the target class in the last 45,000 (out of 50,000) training examples, we hypothesize that it “forgot” the target class. This is consistent with the finding that massed practice would result in poor testing performance.

Results for Spaced Learning

For the spaced practice condition, we did not notice a significant advantage for the accuracy of the target class (numbers on the diagonal). However, unlike the massed practice condition, we observe that the network does not “forget” the target class (accuracy of 0 for the target class). Thus, this finding is consistent with our research hypothesis that spaced practice is more beneficial than massed practice.

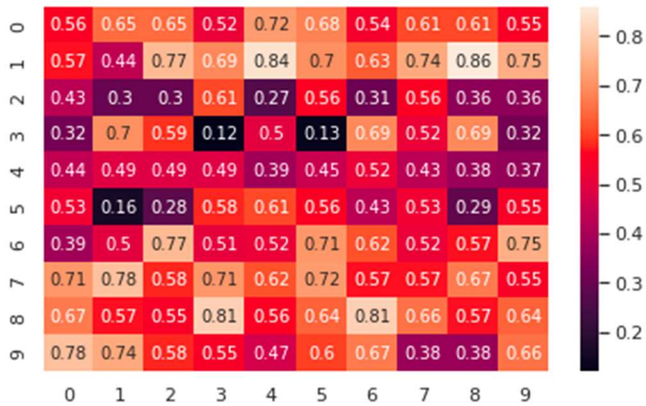

Figure 7: A heatmap of the results from the model that was trained using spaced sequencing of data.

Results for Expanded Learning

We can see that along the diagonal representing the target class, the accuracy score is generally lower than the rest of the heatmap. The clear dark diagonal means the expanded sequencing of data performed worse than the other classes that were randomly sequenced in the rest of the data. This shows us that expanded practice is not a reliable approach for training convolutional neural networks.

Figure 8: A heatmap of the results from the model that was trained using expanded practice sequencing of data.

Results for Expanded Learning with Weighted Sampling

In expanded practice with weighted sampling tests, we notice that, like the other expanded practice tests, the performance for the target class is almost always one of the worst performing classes, although not by much in most of the cases. This achieved notably better results than the previous expanded practice representation, but still not as good of results as spaced learning.

Figure 9: This is a heatmap of the results from the model that was trained using our alternative representation of expanded practice using weights.

Discussion and Future Direction

From our results, we note that there is no improvement in our machine learning model performance by using a spaced, massed, or expanded sequencing of data over a baseline model. We can see that the order of performance for each of our models is: Spaced > Expanded with Weights > Expanded > Massed. From a psychology perspective, this seems to make some sort of sense in that spaced and expanded learning both perform better than massed learning, although it is not consistent with our hypothesis that expanded learning should have the best performance on the target class. However, from a machine learning perspective, we observe that the systems that learned the best had the most frequent and consistent training of the target class towards the end of the training sequence. This effect is most pronounced in the massed and spaced sequences. In the massed sequencing model, all of the target class examples were at the beginning of the training sequence, and so by the end the model had not seen any examples of the target class for 45,000 examples, and thus performed extremely poorly. In contrast, for the spaced sequencing model, the training frequency for the target class is consistent throughout the training process, giving the spaced sequencing model similar results to the baseline model. The expanded model performed somewhere in between, with the frequency of target class training examples greatly reduced as it neared the end of training. With the intuition of the performance disparity analyzed with a machine learning explanation in this way, we cannot conclusively determine if our convolutional neural network can be used as a reasonable model of spaced, massed, and expanded learning.

This effect can potentially be addressed in future directions in a number of different ways. One way to address this effect might be to conduct the experiment with a much simpler dataset and model such that a model could reasonably learn something significant while only using a significantly smaller amount of training. Even though we used a pre-trained CNN to reduce the dimensionality of our problem, our small neural network still contained over 33,000 weights, which is in the same ballpark as our number of training examples. For example, a logic gate dataset using a small, simple, neural network could be a logical future direction for this line of research.

Another unrelated potential future direction might be to explore a dynamically organized training set such that examples that have been previously misclassified are resampled in order to help improve and reinforce the classification performance of the target class.

Conclusion

In our research, we attempted to explore and model the documented effects of spaced, massed, and expanded learning by leveraging convolutional neural networks and target class data sequencing to classify simple images. Our results show that spaced learning performed the best, followed by expanded learning and then massed learning, although none of the sequences outperformed a baseline machine learning model. We provide a simple machine learning explanation of why the models performed the way they did, and thus we cannot conclusively determine if our CNN is a reasonable model for spaced, massed and expanded learning. We also provide potential future avenues in order to improve upon our research.

References

Ebbinghaus, H. (2013). Memory: A contribution to experimental psychology. Annals of Neurosciences, 20(4). https://doi.org/10.5214/ans.0972.7531.200408

Grote, M.G. (1995). The Effect of Massed Versus Spaced Practice on Retention and Problem-Solving in High School Physics.

Landauer, T. K., & Bjork, R. A. (1978). Optimum rehearsal patterns and name learning. Practical Aspects of Memory, 1.

LeCun, Y., Bottou L., Bengio Y., and Haffner P. (1998). Gradient-based learning applied to document recognition. Proceedings of the IEEE.

McCulloch, W. S., & Pitts, W. (1948). The Statistical Organization of Nervous activity. Biometrics, 4(2), 91. https://doi.org/10.2307/3001453

Rohrer, D., & Taylor, K. (2007). The shuffling of mathematics problems improves learning. Instructional Science, 35(6), 481–498. https://doi.org/10.1007/s11251-007-9015-8.