Visual Object Recognition & One-Shot Learning

Philippe M. Noël

Neuro140 Final Project, Spring 2019

Abstract

I trained multiple Convolutional Neural Networks (CNNs) to perform various image classification tasks. These tasks ranged from traditional image recognition on MNIST using a custom-made CNN, to the application of transfer learning using InceptionV3 and VGG16 to the MNIST and Omniglot data sets, and to an implementation of a Siamese Neural Network for one-shot learning on Omniglot. This work is motivated by an interest in one-shot learning and the "learning-to-learn" concept in making artificial intelligence more human-like. This project is an exploration along with implementation of all of these technologies and is designed to be accessible to anyone with intermediate Python programming skills and a conceptual understanding of supervised learning and convolutional neural networks. I demonstrate the phenomenon of one-shot learning through transfer learning with a classification accuracy on Omniglot of 99.9% through InceptionV3 and VGG16 models, confirming the exploration-exploitation hypothesis, and the phenomenon of one-shot learning through a Siamese Neural Network with a classification accuracy on 20-way Omniglot of 64%, demonstrating the features association/differentiation hypothesis. This project also includes a surface exploration of Memory-Matching Networks as an avenue for further developing one-shot learning.

1 Introduction

Intelligence is a fascinating concept. It is so much of what makes us humans, yet it is not fully understood to this day. Multiple elements are part of what we call human intelligence, and we have been forced to define them more clearly as artificial intelligence has quickly progressed in accomplishments and applications. There are five categories that are commonly cited in the cognitive science community as to elements that are required for an artificial intelligence (AI) to be said to have human-like intelligence. They are: intuitive physics, intuitive psychology, compositionality, learning-to-learn and causality [5].

In this paper, we interest ourselves at a specific one: learning-to-learn. While the exact definition of the concept varies, it centers around the idea that humans, when learning, learn a rich conceptual representation the thing they are learning about. In the case of simple objects or tasks, they are often able to learn and generalize their learning extremely rapidly, with just one or a few samples [5]. This is especially true with children, who engage in rapid mapping. There are even forms of learning that come from no prior exposure to any object, simply from generalizing prior knowledge. A child who was shown an elephant and is subsequently shown a parrot might not know what a parrot is, but already understands that a parrot is not an elephant, and this is without having ever been exposed to a parrot previously. On the contrary, current state-of-the-art deep neural networks trained through back-propagation typically require thousands, if not tens of thousands or hundreds of thousands of data points in order to build a generalization robust enough to provide good classification or regression accuracy on reasonably complex tasks. Multiple examples of this exist in computer vision, with AlexNet, along with various other fields, including statistical regression and natural language processing. Those systems develop extremely good pattern recognition, but have no understanding of the object they are recognizing, and thus are not said to be truly intelligent.

This is where this project comes in. My interest is in trying to incorporate a way for neural networks to learn more quickly, from fewer data points. This task is called *one-shot learning*, for learning from only one training point, or, two-shots learning for learning from two training points, and so on. In order for networks trained with back-propagation to learn in such few examples, they need to learn a somewhat richer representation or start from further along the learning iterative journey. This is what I am interested in mimicking. In order to experiment on this concept, I will be working on image recognition, which is a very well studied task in supervised learning and has a clear way of assessing the concept of one-shot learning, mainly can a network learn to recognize an object from only seeing it once. The state-of-the-art image recognition technologies use convolutional neural networks (CNNs) as a model for their impressive performances and relative similarity to the human visual system. In order to tackle this task, I formulate two potential hypotheses as to how such fast mapping can be accomplished in humans, and how to apply them to convolutional neural networks.

2 Hypotheses

In order to crystallize this learning-to-learn concept in a specific experiment, we formulate two hypotheses as to potential explanations for how humans learn from very few samples.

Hypothesis #1: Exploration-Exploitation

This hypothesis is based on the fact that humans inherit at birth an extremely complex visual system with already tuned sets of weights to recognize edges, colors, etc., and spend months and years learning to recognize various objects. These two elements together is the exploration phase, the phase in which an agent builds its toolkit to perform later. They can then use this trained visual system to more easily and quickly learn to recognize new objects with little training data. This is the exploitation phase, in which an agent makes use of what it learned during exploration.

Our hypothesis is that this exploration-exploitation leads to frequent one-shot or few-shots learning instances in humans due to the visual system and previous experience recognizing objects. In order to test this hypothesis, we will make use of deep CNNs trained on ImageNet, to mimic the building of a visual system over evolution and experience, and use transfer learning to replace the classification layer with a new one to try to recognize new objects are only a single epoch (only one image) of training on each image.

Hypothesis #2: Features / Memory Association

This hypothesis is based on the fact that humans, when learning to recognize an object, learn a conceptual understanding in addition to a visual representation. When asked to explain how they identified an object, humans will call to features of that object versus other objects and concepts attached to this object. "This is a panda because it is black and white, has fur and is very large.", or "This is a motorbike because it has two wheels and is on a highway in this picture.". If asked to put this object in context, humans can easily say that a motorbike can go fast and be used as a mode of transportation. Therefore, it seems humans, when learning, learn a richer representation than just a visual recognition, they build some association of the features of the object and understand how these pertain to the object and what they can be used for, and so for each object, they must associate elements to it that are more than just patterns recognition, but also features association and some form of long-term memory imprint to remember how those features map to the object.

Our hypothesis is that this features/memory association leads to frequent one-shot or few-shots learning instances in humans due to the conceptual understanding of the object that they build, which they can use to identify it. Rather than only focusing on the shapes or the color, humans can also use properties, applications and expected behaviors associated with items to more quickly narrow down which item is being presented to them. In order to test this hypothesis, we will explore Siamese Neural Networks, a type of neural network solving the task of identifying which of the objects in a pool of object is similar to the one being presented [13]. We could imagine this task as a traditional psychology experiment of placing multiple items of a table, and asking a child to pick a certain item they are shown. We will also explore Memory-Matching Networks, which build a richer feature space by making a decision based on the whole sample space, not only on the presented sample space [14, 15, 16].

These are the two hypotheses guiding our exploration and our experimentation. Having never experimented deeply with convolutional neural networks prior to this project, we defined a structure in layered steps for reaching the stage of testing our one-shot learning hypotheses on complicated networks. This project is segmented in 5 key implementation steps:

- Implementing a standard CNN from scratch on MNIST
- Performing transfer learning on MNIST from a large model trained on ImageNet
- Implementing a standard CNN from scratch on Omniglot
- Performing transfer learning on Omniglot from a large model trained on ImageNet
- Implementing a Siamese Neural Network on Omniglot

The structure of the project is so that I can explore and build understanding step-by-step from the ground up until reaching the point at which I can tackle one-shot learning. The first two steps are meant for exploration and learning, which I why they are performed on MNIST, which is a simpler dataset to work on. The third step is aimed at providing a benchmark for what is "ideal" on a network trained exclusively for Omniglot. Lastly, the last two steps are the attempts at tackling each of the two hypotheses for one-shot learning.

3 The Datasets

The first two steps of the project are performed on MNIST. I will not go into details on MNIST. It contains 60,000 training images and 10,000 test images of 28x28 pixels greyscale images of handwritten digits. Although the first part of the project utilizes MNIST in order to learn to build CNNs and work with image data more easily and with a standard benchmark in image recognition, the dataset we are interested in here is Omniglot. From Brenden Lake, one of the originators of the dataset, "The Omniglot data set is designed for developing more human-like learning algorithms. It contains 1,623 different handwritten characters from 50 different alphabets. Each of the 1,623 characters was drawn online via Amazon's Mechanical Turk by 20 different people" [6]. Each image is 105x105 pixels, greyscale.

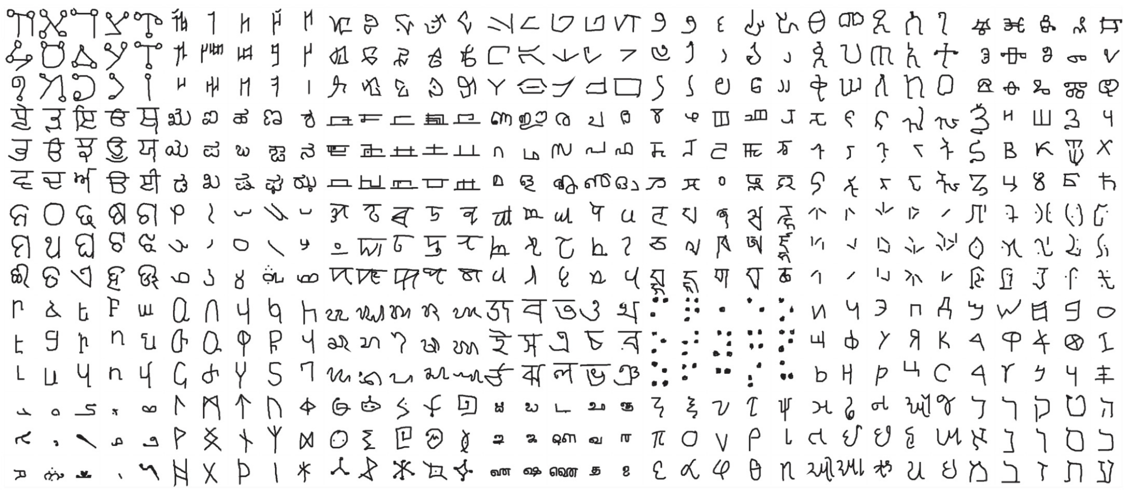

Figure 1: Part of the Omniglot dataset, from Brenden Lake.

This makes Omniglot the standard dataset in one-shot learning, the MNIST of one-shot learning, due to its large number of simple classes with very few samples each. It is often called the transpose of MNIST, since it has a lot of classes and few data points in each. It is reasonable to expect a human to be capable of recognizing a simple alphabetical character like these from only one image, which Lake et al. explain in their paper on Omniglot [7].

4 Pretrained Models

Transfer learning is based on the idea that a large network which learned a set of weights, if trained on a dataset diverse enough, should be able to generalize quite well to classifying other similarly complex images with little or no training. This idea rest on the concept of making use of off-the-shelf state-of-the-art convolutional neural networks trained on huge datasets for multiple days and replacing their dense classification layer without retraining the other layers [8]. In this section, before delving into our results, we outline the two off-the-self networks that we used for our transfer learning experimentation.

InceptionV3

The InceptionV3 model takes weeks to train on a gigantic computer with 8 Tesla K40 GPUs [8]. This makes it impossible to train on a single laptop or Google Colab GPU. It has nearly 25 million parameters and uses 5 billion multiply-add operations for classifying a single image. It was developed by Google as ReCeption and performed one of the best in ILSVRC (ImageNet Large Scale Visual Recognition Competition) in 2015 [10]. It is frequently used as an off-the-shelf model in transfer learning and can be easily imported from Keras.

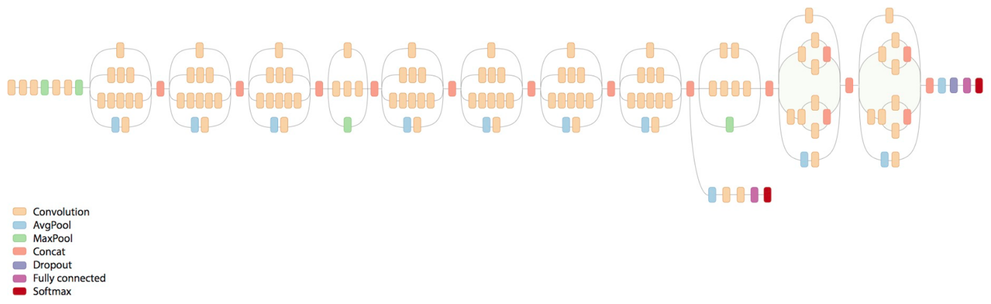

Figure 2: InceptionV3 Architecture, from Google AI Blog.

VGG16

VGG16 is a convolutional neural network model proposed by K. Simonyan and A. Zisserman in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The model achieves 92.7% top-5 test accuracy in ImageNet. This model was famously submitted to ILSVRC in 2014, improving over AlexNet by replacing large kernel-size filters in the first and second layers with multiple (3,3) kernels [2]. This type of kernels inspired us in choosing what kernels to use in building our models in the experimentation phase, seeing how VGG16 performs.

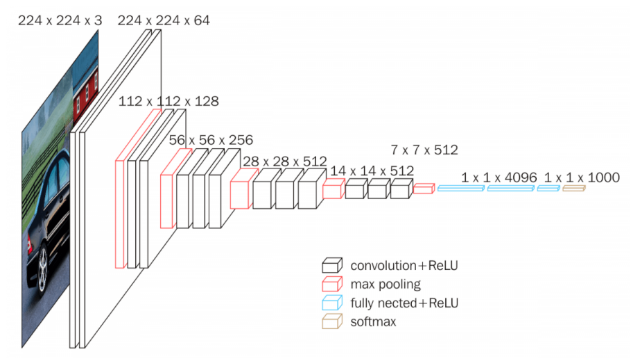

Figure 3: VGG16 Architecture, from Neurohive.

5 Results

Our experiment was separated in 5 stages. For each stage, one or two convolutional neural networks have been built and a lot of data pre-processing had to be done. We present here the results for all our stages, although we will focus our discussion on the last two stages, which are directly testing our two hypotheses.

Stage 1 - Experimenting with CNNs

In the first stage, I dabbled with building a CNN in Keras. This stage was meant as an introduction to the building of CNNs and to visual object recognition. The dataset used was MNIST, for its simplicity and ease of usage, along with its standard application as the introductory image recognition dataset. I built one model, consisting of two convolutional layers, two pooling layers and two normalization layer, then flattened and densely connected to one intermediate dense fully-connected layer connecting to the dense output layer 10-way softmax. Dropout layers were used, as they help reduce overfitting by setting some fraction of the weights randomly to zero [1].

The images arrays were not modified and the labels vectors were converted to one-hot encoded for working with the Keras backend. After only 5 epochs of training on the whole training set of 60,000 images, we achieved an evaluation performance on the test set of 99.82%. As we can see, the MNIST dataset is easily recognizable, as our network was quite shallow.

The network architecture was inspired from Krizhevsky et al. [1] and heuristics in the literature. Too many layers or neurons would likely lead to overfitting, so I preferred to keep the network reasonably shallow as the MNIST dataset is not very challenging. The kernel sizes for the convolutions were set to (3,3) as recommended in the literature for small images and hidden layers activation functions were set to ReLU as it generalizes better and trains faster [1]. The pooling layers kernels were set to (2,2) as well, to reduce the dimensionality rapidly and narrow down on specific features independently of the location in the image.

This stage was intended as exploratory and most of the work was spent in tuning the network and the hyperparameters to come up with the ones chosen. In this idea, I plotted some intermediate convolution outputs to explore how the CNN was picking up on patterns in the image, and was quite impressed by the results obtained [3]. CNNs rapidly and accurately learn to recognize lines and edges, as multiple guest speakers expressed in class.

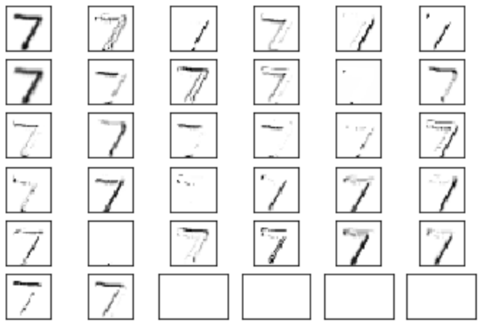

Figure 4: CNN Model 1 first convolution output on input ”7”.

Seeing the accuracy, I was extremely pleased with the results and the understanding developed and decided to move on to the next stage.

Stage 2 - Experimenting with Transfer Learning

In this section stage, I was interested in learning about the concept of ”Transfer Learning” and how to apply it, as I formulated it as one of my hypotheses for one-shot learning. The dataset used was once again MNIST, our dataset of choice for the learning phase, and the original pre-trained model that I decided to use was InceptionV3, which was built by Google and performed extremely well on challenging image classification contests. The concept behind transfer learning is to use a pre-trained large CNN and replace its final classification layer without retraining the hidden layers so the weights it learned are used to generalize to a different task. This can often lead to improved performances [8] for experiments with little computational resources available as was my case.

This stage was separated in two substages. In the first stage, I was interested in delving inside the network modified for transfer learning and truly understand how it works and how the data is represented. Since InceptionV3 is trained on ImageNet, it expects color images (3-channels images), while MNIST is made up of greyscale images (1-channel images). The MNIST dataset was therefore reshaped to be 3-channels and, following resources from the infamous TensorFlow tutorials from Hvass-Labs [8], I was able to obtain the transfer values from the InceptionV3 network. The transfer values are the values that are outputted at the end of the InceptionV3 network (the non-trained part) and fed into the new classification layer.

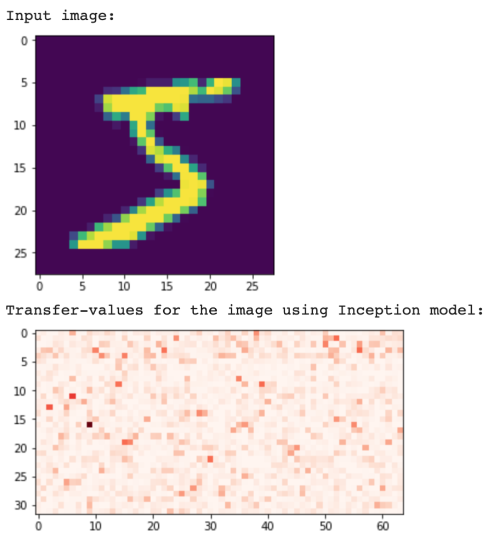

Figure 5: InceptionV3 transfer values on input 3-channels ”5”.

I faced multiple challenges in reshaping the data in this way and my dataset was merged into 20,000 images from the original 60,000, indicating that images had been combined together in the reshaping, which is not a valid transformation. Seeing the complications of working without Keras, I decided to move on to build my transfer learning experimentation in Keras now that I had a strong understanding of the data representation within the InceptionV3 network.

In this second substage, I used Keras to build the transfer learning model with InceptionV3 on MNIST [9]. The data was reshaped differently, leading to a representation without any merging of images, and the InceptionV3 model was added with a dense layer, a dropout layer and finally a 10-way softmax classification layer. Moreover, the InceptionV3 model requires images of at least 75x75 pixels, which forced us to reshape the MNIST images from 28x28 to 84x84 in order to preserve proportions.

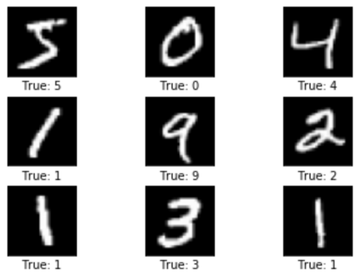

Figure 6: Reshaped MNIST.

After training one epoch over the entire training dataset, the accuracy converged at 90% on the training set and did not change, which led to the decision of not training any further. Since InceptionV3 is trained on much different images, it is likely that it is incapable of generalizing its weights that much, which is why we saw a rapid convergence in training loss. The evaluation accuracy we obtained was 81.65%, which shows that there was some overfitting, but that our transfer learning experiment was nonetheless extremely successful, as the random guess accuracy on a 10-way classification would have been hovering around 10%. We were satisfied with this accuracy and moved on to our next stage.

Stage 3 - Standard CNN Classification

This is the stage of the project at which we transition away from our toy dataset that is MNIST and with the real one-shot learning dataset that we are interested in: Omniglot. Most of the work of this stage was spent in applying Omniglot, designed for Siamese Neural Networks, to standard convolutional neural networks. The Omniglot dataset consists of 50 alphabets, each of which with multiple character classes which each contain 20 images of the specific character. I therefore split the dataset in a train and a test set, using a 80-20 rule, which meant 16 images of each character in the train set, for a total of 25,968 images, and 4 images of each character in the test set, for a total of 6,492 images.

We then built a standard CNN analogous to the one built in Stage 1, following a ”convolution layer” - ”pooling layer” - ”normalization layer” framework, but this time using 5 of those cycles instead of only two, to have a more complex network. This network was trained for a simple epoch, after which it reached a classification accuracy on the test set of 99.9%. It appears that the InceptionV3 is likely too powerful for working directly with Omniglot, which is not very surprising considering the discrepancies in image complexities, and therefore there likely might have been overfitting despite the high test set accuracy. As we were mainly interested in establishing a benchmark in this section, which clearly is an extremely high benchmark, we were satisfied with this result and moved on to the next stage.

Stage 4 - One-Shot Learning: Transfer Learning

This is the stage at which we start to delve more deeply in the hypotheses. My first hypothesis was that humans can perform one-shot or few-shots learning because they build on top of a highly developed visual recognition system that evolved over the course of human evolution and their own lifetime, while CNNs must learn their visual system from scratch for a specific task. In order to test this hypothesis, I combined Stage 2 and Stage 3 and used transfer learning on InceptionV3 and VGG16 to perform transfer learning on Omniglot.

The dataset was split following the same 80-20 rule as in Stage 3 and each image was augmented to be 3-channels to match with the input shape of our two pre-trained models. The two pre-trained models were then scraped from their classification layers and added a hidden dense layer and a new 1,623-way classification layer to fit with Omniglot. Two dropouts layers were also added with high dropouts to prevent overfitting, seeing how well a standard simple CNN performed in the previous stage.

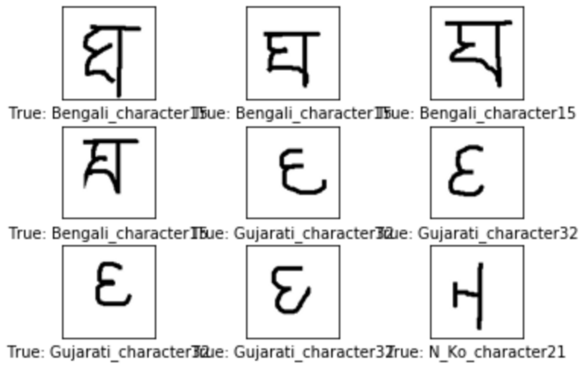
True: Bengali_character15
True: Bengali_character15
True: Bengali_character15
True: Bengali_character15
True: Gujarati_character32
True: Gujarati_character32
True: Gujarati_character32
True: Gujarati_character32
True: N_Ko_character21

Figure 7: Reshaped Omniglot.

The one-shot learning task then defined as follows. The Omniglot dataset was broken down so to store in an array of images a single image of each of the 1,623 classes along with their associated class label. These arrays, after having been reshaped, were thus of shape (1623, 105, 105, 3). The model was then trained on a single epoch of one of the randomly chosen array, so that the model would only see one image of each class. The model was then evaluated on the test set, which was split the same way, showing each of the 1,623 images only once again and seeing if the model can predict correctly. On InceptionV3, we reached a training accuracy of 99.94% and a test accuracy of 99.94%, expressing how powerful building from a strong visual recognition system can contribute to rapid mapping. On the VGG16 network, we reached a training accuracy of 99.92% and a test accuracy of 99.94%. We were extremely satisfied with these results, which appear to confirm our first hypothesis, and moved on to the final stage of the project. We will come back to those results in our discussion.

Stage 5 - One-Shot Learning: Siamese Neural Networks

In this final stage, we test our second hypothesis, mainly that humans engage in rapid mapping by building a richer representation of an object than simple pattern recognition. In order to test this hypothesis, we turn to a radically new architecture: Siamese Neural Networks. Siamese Neural Networks are a type of deep CNN combining two CNNs and computing the distance between their respective prediction on two different images in order to evaluate if the two images are from the same class or not. Clearly, this is a very different task than that of transfer learning, but also one that exhibits one-shot learning property as we could imagine a task in which a child is shown plenty of objects, and is then shown a new object and asked to go fetch the object that looks like one shown from the original list of objects. Each leg of the Siamese Neural Network is its own standard CNN learning to recognize features in the image. In this stage, we reproduced the implementation from Koch et al. [11] and thus tried to reproduce their results.

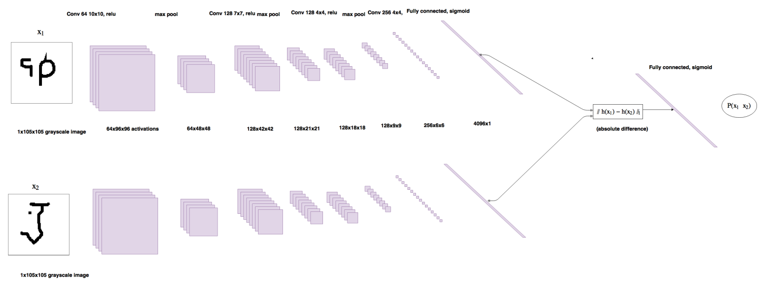

Figure 8: Siamese Neural Network Architecture, from Soren Bouma.

They experimented heavily with multiple parameters for each leg of their Siamese network and settled on the following architecture, which is the one we also used:

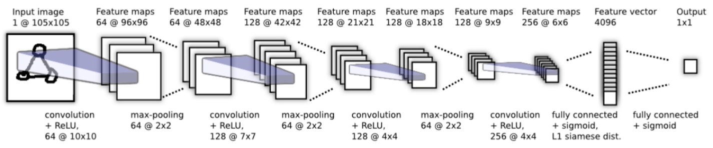

Figure 9: Siamese Neural Network Leg Architecture, Koch et al.

The task we built was analogous to the one from the Koch et al. paper. We defined a 20-way one-shot learning task on Omniglot, by sampling 20 random characters from different alphabets and one one of these character was the one to find in the list. The network is then evaluated, comparing the target image to each of the 20 sample images subsequently, only one of which is from the same class, and the similarity between the images is evaluated, the highest similarity (lowest distance) being the image predicted. This is an extremely powerful concept. The network is trained on the training alphabets, 30 alphabets, and is then evaluated on the testing alphabets, 20 alphabets. This signifies that once a Siamese Neural Network has been tuned, it can generalize not just to new data, but to entirely new classes from unknown distributions [11].

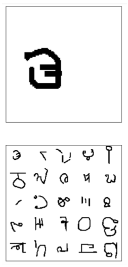

Figure 10: Example of a 25-way one-shot learning task.

We built a near exact representation of the Siamese Neural Network from the Koch et al. paper, only leaving out some fine-tuning features like layerwise momentum and layerwise learning rates, which would have little effects on the overall behavior that we aimed to observe, and trained the network on three different L2 regularizer values, 0.1, 0.01 and 0.001. We followed a great tutorial from Soren Bouma [12], adapting it and building on top of it. It was observed that the network performed best with a L2 regularizer of 0.01, reaching consistent 60% accuracy on the evaluation set and peaking at 64% accuracy. A regularizer value of 0.1 lead to a consistent accuracy of around 30%, showing how the regularization was too extreme for the network to truly perform well, while a regularizer value of 0.001 lead to drastically varying accuracies from iteration to iteration, ranging from as low as 2% to as a high as 74% with no clear converging scheme, showing how an unregularized network is incapable of performing well and is overfitting to the training data. This appears to confirm our second hypothesis in that humans learn features representation of objects that they can generalize to new objects.

6 Discussion

We can summarize the 5 stages of the project in two categories: exploration and exploitation, which is quite ironic considering this is my first hypothesis. In the first two stages of the project, I experimented with CNNs and transfer learning, having barely worked with CNNs before and never worked with transfer learning. It was a very challenging task and took a lot of work, but the results obtained are very satisfying. They show clear successes in applying those technologies and laid a solid foundation for moving forward to the one-shot learning part. The first element to discuss in Stage 3 is that of overfitting. Obtaining a result of 99.9% accuracy on a 1,623 classes classification task is impressively high for such little training, and this shows how the network is likely overfitting despite using multiple dropout layers and having a very high evaluation accuracy. Omniglot is a very simple dataset per CNNs standards and that is likely what explains such a high accuracy.

Transitioning to Stage 4 and our attempts at one-shot learning through transfer learning, we can see that it was a great success, performing over 99.9% accuracy again on Omniglot, similar to our previously established benchmark. Clearly, this confirms our first hypothesis. If there was no building on top of a previously strong visual system, the random guess accuracy would be 1 in 1,623, which is extremely low. Therefore, it is clear that this network performs extremely well by re-using its pre-learned generalizable set of weights. That being said, we must keep in mind the overfitting element in here. Our simple CNN performed exceedingly well and therefore it is no surprise that more complex networks, even though not trained on Omniglot, perform similarly well. The Omniglot 20 images for a specific class remain highly similar and the Omniglot dataset is not optimized for this type of standard networks. As such, we do consider our hypothesis to be confirmed, but want to put a mention of care on the actual accuracy, it likely would be lower on a more complex dataset.

Finally, transitioning to Stage 5, we can see that this was our lowest performance experiment and also the hardest one to build. We experimented with multiple values of L2 regularizer within the range defined by Koch et al. and observed difficulties in convergence. We observed the best results for a L2 regularizer of 0.01, showing how appropriate regularization can help convergence while too much prevents learning and not enough leads to overfitting. There is an open question with this type of architecture, as it is not fully generalizable one-shot learning, but rather one-shot recognition from a fixed sample. A major flaw of this approach is that it only compares the test image to every image in the N-way images sample (20 in our case), while it really should compare it to every image in the test set (thousands of images). The question is open as to how humans do this. When trying to recognize an elephant, do humans compare this image to all images they know, or only to a specific subset they know how to target with efficient search? Likely not to every image. This is an interesting question to build on and a natural question concerning Siamese Networks. As was pointed our during the midterm presentation by Colin, humans are actually exposed to multiple images for an extended period of time (watching an elephant for a few seconds in changing light), and so this task is not accounting for this fact, but is working with simpler data. An open question I therefore pose is, could we relax the constraints to 10, 20 or even 50-shots learning with very similar images, splitting a few seconds of video in its frame, and be able to perform Siamese classification on more complex images?That being said, we have confirmed our second hypothesis, as here there is clearly a fast-mapping associated with images only seen once before (the image to find in the sample), as an accuracy of 60% is much higher than the random guess accuracy of roughly 5%. This accuracy remains much below Koch et al.'s accuracy of 93% [11], but we trained our model much less, did not perform data augmentation and simplified the layerwise parameters slightly. That being said, we are extremely happy with the results, which demonstrate a clear case of one-shot learning.

7 Conclusion

In this project, I have experimented with standard convolutional neural networks on MNIST, achieving over 99% accuracy, and with transfer learning on InceptionV3 on MNIST, achieving over 80% accuracy. I then applied this knowledge to tackling the task of "learning-to-learn" in trying to make deep neural networks more human-like, tackling the problem of one-shot learning through two hypotheses, exploration-exploitation and features association. I first confirmed the one-shot learning hypothesis of building on top of already existing visual system through transfer learning, observing that transfer learning networks built from InceptionV3 and VGG16 perform over 99.9% accuracy on recognizing Omniglot characters after having seen only one of each characters. I then confirmed my second hypothesis, that there is a features association element to visual object recognition, obtaining over 64% accuracy on a Siamese Neural Network on a 20-way Omniglot one-shot task. The clear next step is to look further at Memory-Matching networks, which combine those two hypotheses in a single one by learning a full end-to-end classifier over images instead of over image pairs [14, 15, 16]. Other avenues include transfer learning with more challenging datasets and exploring new siamese architectures. I hope this project helps spur interest in meta-learning for deep learning beginners and that further work keeps being developed on the topic, as it seems a promising avenue for combining biological intelligence and artificial intelligence.

References

Convolutional Neural Networks (CNNs)

[1] A. Krizhevsky, I. Sutskever, G. E. Hinton. ImageNet Classification with Deep Convolutional Neural Networks. 2012. https://papers.nips.cc/paper/4824-imagenet-classification-with-deep-convolutional-neural-networks.pdf

[2] K. Simonyan, A. Zisserman. Very Deep Convolutional Networks for Large-Scale Image Recognition. 2015. https://arxiv.org/abs/1409.1556

[3] Hvass-Labs. GitHub - TensorFlow Tutorials - Convolutional Neural Network. 2018. https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/02_Convolutional_Neural_Network.ipynb

[4] Hvass-Labs. GitHub - TensorFlow Tutorials - Keras API. 2018. https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/03C_Keras_API.ipynb

Omniglot Dataset

[5] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum. Human-Level Concept Learning Through Probabilistic Program Induction. 2015. https://web.mit.edu/cocosci/Papers/Science-2015-Lake-1332-8.pdf

[6] B. M. Lake. GitHub - Omniglot Data Set For One-Shot Learning. 2019. https://github.com/brendenlake/omniglot

[7] B. M. Lake, R. Salakhutdinov, J. B. Tenenbaum. The Omniglot Challenge: A 3-Year Progress Report. 2019. https://arxiv.org/pdf/1902.03477.pdf

Transfer Learning

[8] Hvass-Labs. GitHub - TensorFlow Tutorials - Transfer Learning. 2018. https://github.com/Hvass-Labs/TensorFlow-Tutorials/blob/master/08_Transfer_Learning.ipynb

[9] C. Conwell. Bioai Colab Tutorial. 2019. https://colab.research.google.com/drive/149rTkGRw16OLC6LW6T9av23mPaehcysx

[10] S.-H. Tsang. Review: Inception-v3|1st Runner Up (Image Classification) in ILSVRC 2015. 2018. https://medium.com/@sh.tsang/review-inception-v3-1st-runner-up-image-classification-in-ilsvrc-2015-17915421f77c

Siamese Neural Networks

[11] G. Koch, R. Zemel, R. Salakhutdinov. Siamese Neural Networks for One-Shot Image Recognition. 2015. https://www.cs.cmu.edu/~rsalakhu/papers/oneshot1.pdf

[12] S. Bouma. One Shot Learning and Siamese Networks in Keras. 2017. https://sorenbouma.github.io/blog/oneshot/

[13] D. Karunakaran. One-Shot Learning Explained Using FaceNet. 2018. https://medium.com/intro-to-artificial-intelligence/one-shot-learning-explained-using-facenet-dff5ad52bd38

Memory-Matching Networks

[14] O. Vinyals, C. Blundell, T. Lillicrap, K. Kavukcuoglu, D. Wierstra. Matching Networks for One Shot Learning. 2017. https://arxiv.org/pdf/1606.04080.pdf

[15] Q. Cai, Y. Pan, T. Yao, C. Yan, T. Mei. Memory Matching Networks for One-Shot Image Recognition. 2018. https://arxiv.org/pdf/1804.08281.pdf

[16] A. Santoro, S. Bartunov, M. Botvinick, D. Wierstra, T. Lillicrap. One-shot Learning with Memory-Augmented Neural Networks. 2016. https://arxiv.org/pdf/1605.06065.pdf

Code Implementation

The following links to a GitHub repository containing the .ipynb of the project implementation, along wit the dataset and the Python helper file for renaming folders. The .ipynb notebook is fully self-contained and split into 7 sections, for each of the stages of implementation, and should be easily reproducible.

`https://github.com/Noep1997/one_shot_learning_project`

Lastly, here is the Google Colab link, if you would like to run the code directly for yourself.

`https://colab.research.google.com/drive/1vUMHqLKXff3WA-9D5cIUALiI7quP_Y_H`