MLP Coursework 4: Project Final Report
Transfer learning and dataset size
Group G74

s1700260, s1457374, s1784849

Abstract

Write This Last.

1. Introduction

Large-scale neural networks have become increasingly prevalent in artificial intelligence [INSERT REFERENCE HERE]. Such architectures effectively solve computer vision problems that were previously considered difficult to address using traditional symbolic manipulation [INSERT REFERENCE HERE]. One concern with modern neural networks is that their performance scales with the quantity of data used to train them, as demonstrated in our previous piece of research [INSERT REFERENCE HERE]. This concern is a problem for individuals or institutions with small amounts of data that want to implement these architectures as solutions, or for research purposes.

Data driven techniques exist, such as data augmentation [INSERT REFERENCE HERE], which help solve this problem [footnote: see our previous paper for a further explanation]. However, to draw from our previous research [INSERT REFERENCE HERE] we investigate alternative, machine learning based approaches, specifically transfer learning [INSERT REFERENCE HERE] as a method to improve the performance of neural network architectures that utilise small amounts of data. These techniques allow us to use the knowledge that a given (pre-trained) network already has and adapt the architecture to suit our requirements [INSERT REFERENCE HERE].

The objective of transfer learning is to use the knowledge acquired by an existing network that was built to solve a similar task. The main idea is to try to adapt that previous knowledge to the new problem to be solved. One of the advantages of this technique is that the solution for the new problem does not have to start from scratch. Prior work can be re-utilised which is important when the resources are limited. This situation is specially true for convolutional neural networks which, depending on the size of the architecture, can require an important amount of computational power.

There exist a number of pre-trained neural networks to solve classification tasks in the field of computer vision. One of them is the Visual Geometry Group network (VGG net)[INSERT REFERENCE HERE]. This network implements a simple yet deep architecture with several convolutional layers. There are several configurations that vary the number of convolutional layers among 11, 13, 16, and 19. In every configuration the architecture includes maximum pooling layers of dimensions 2x2 and stride 1. The convolutions are performed using 3x3 filters with stride and pad of 1. Finally, ReLU activations after each convolution are utilised.

VGG net was firs used in the ImageNet Challenge 2014[INSERT REFERENCE HERE], which means that is was originally trained using the Imagenet dataset [INSERT REFERENCE HERE]. However, the network has also been trained in other datasets like CIFAR-10 [INSERT REFERENCE HERE] and CIFAR-100[INSERT REFERENCE HERE]. It is important to make this distinction since each dataset has been developed for a different purpose, therefore, they have specific characteristics. The knowledge learnt from a network depends on the dataset used to train it. Thus, it is critical to know the characteristics of the dataset utilised to train the network since it will play and important role when implementing transfer learning.

ImageNet is a large-scale hierarchical image database [INSERT REFERENCE HERE]. The structure of this dataset contains 12 main groups or sub-trees. Each sub-tree contains a variable number of categories. The total number of categories is 5247, each one containing on average 600 images; thus, the total number of images in the dataset is 3.2 million. As described, ImageNet is a large and diverse dataset. However, the original version of VGG net was trained on a subset of Imagenet that contains only 1000 categories using 1.3 million images for training, 50K images for validation and 100K images for testing.

1.1. Objectives

The research presented within this report pertains to two fundamental objectives. The first of which concerns itself with improving the performance of image classification tasks using transfer learning and deep feature extraction with limited amounts of data. The second investigates the generalisability of the methods as mentioned above to two distinctly different datasets [REF: APPENDIX: PARAGRAPH ON DATASETS]. Furthermore, we intend to propose and utilise a framework for measuring the visual similarity between datasets using a modified siamese neural network [INSERT REFERENCE HERE] architecture.

1.2. Interim Report Findings

The research questions offered within this report have been drawn from our initial findings, as documented in the interim report [INSERT REFERENCE HERE]. To aid in orientating the reader, we provide a summary of these findings here.

It was possible to demonstrate that a correlation exists between the amount of available data used to train, test and validate a simple convolutional neural network and the generalised accuracy of the model. This correlation shows that as the amount of data reduces, so does the validation accuracy.

Furthermore, we were able to prove that shallower networks have a higher versatility when the perceived visual similarity between classes within a dataset is high [footnote: to clarify, shallow networks perform better on data where classes are distinctly different from one another. We compare our clothes dataset to our expressions dataset in order to draw this conclusion, as expressions are inherently more difficult to identify - a reference to psychological literature goes here]. We suggest that this is because shallow networks do not have the opportunity to extract the necessary features in order to distinctly identify subtle differences between classes, as a larger depth provides a higher number of learnable abstractions in representation.

Finally, we demonstrated that data augmentation effectively improves the performance of a simple convolutional neural network, but it appears there is a threshold where the benefits plateau. This provides further motivation for our investigation.

2. Research Questions

Following from the conclusions drawn from our interim report, our research questions are listed below.

1. How can transfer learning be applied to a convolutional neural network in order to improve the accuracy of the model?

2. How does the application of transfer learning affect two distinctly different datasets? [footnote: the first dataset (clothes) will have low visual similarity between classes, whereas the second dataset (expressions) will have high visual similarity between classes.]

[INSERT RESEARCH QUESTIONS ABOUT ONE SHOT LEARNING]

2.1. Hypotheses

H.1 Transfer learning will provide a performance benefit with respect to the generalisability (measured through validation accuracy) of a simple convolutional neural network architecture.

H.2 Transfer learning will provide a larger performance boost when the size of the dataset used to initially train the model is small rather than large.

H.3 Classification tasks where perceived visual similarity between classes is high will outperform their proposed counterpart tasks where perceived visual similarity is low.

Within the remainder of this report a section that outlines the methodologies utilised to conduct our research [INSERT SECTION REFERENCE] is initially given. Thereafter, all experimental results are presented [INSERT SECTION REFERENCE] and a review of related work is provided [INSERT SECTION REFERENCE]. Finally, a set of conclusions are drawn [INSERT SECTION REFERENCE].

3. Methodology

A discussion of the methodologies used to conduct our experiments is provided herein. These methodologies include all selected data structures and algorithms.

3.1. Transfer Learning

3.1.1. Chosen Transfer Model (VGG16)

We decided to select the VGG16 network [INSERT REFERENCE HERE] to transfer from in order to conduct our experiments. This was primarily due to the similarity between our and their datasets and tasks. To elaborate, we required a pre-trained network that had been trained on image data in order to solve a classification task, VGG16 met these requirements. In addition, there are two main flavours of VGG16, the first uses the ImageNet dataset [INSERT REFERENCE HERE - get from VGG16 paper] and the second utilises CIFAR-100 [INSERT REFERENCE HERE] for training. We decided to use the ImageNet flavour of VGG16 because it has a higher number of classes than CIFAR-100. It follows that we may assume it embodies more knowledge and can provide better results for our particular datasets. Finally, it should be noted that VGG16 has 19 layers (16 convolutional layers and 3 fully connected ones).

3.1.2. Data Structure and Learning Algorithm

To conduct our experiments, a modification of the fully connected layers is made to adapt the model to our needs. This pipeline has a bottleneck between the convolutional layers and the fully connected ones. To reduce the impact of this bottleneck, we divide the process into two components.

Firstly, the fully connected layers are used to convert the raw images from the dataset to another representation. This representation is the output of the last convolutional layer. In some sense, this can be seen as an encoding process, where the network transitions from images that provide information about intensity, to images that provide other kinds of information. Simply put, we are converting the features from the raw images (pixels) to another feature space.

The aforementioned process is time consuming, however, one of the benefits it provides is the dimensionality reduction. Input images are converted from 64x64 (4096) dimensional pixel vectors (representative of features) to a 2048 dimensional feature space that cannot be obviously identified or described. Once we have finished this process, we use the output to feed the fully connected layers.

Secondly, an adaption to the fully connected layers is required, we define three alternatives. The main difference each alternative is the number of fully connected layers utilised within the architecture. In the first option, we use one fully connected layer, which provides 205,607 parameters. The second alternative is to add another fully connected layer, leaving 210,307 parameters. The final alternative uses three fully connected layers, providing 211,407 parameters.

For the first fully connected layer we use the L2 regularisation strategy [INSERT REFERENCE HERE]. The main objective here is to avoid overfitting, which is especially important for small data. Each layer uses a uniform parameter initialisation strategy. [[This kind of initialisation allows to have the activation done in the linear part of a sigmoid function.]] For the output layer, a softmax activation with a categorical cross entropy error loss [INSERT REFERENCE HERE] is implemented. The implemented learning rule is Adam [INSERT REFERENCE HERE].

3.2. One Shot Learning

3.2.1. Siamese Neural Network

[[PLACEHOLDER]]

3.2.2. Algorithm

[[PLACEHOLDER]]

4. Experiments

4.1. Transfer Learning

4.1.1. Motivation

[[TODO: Add additional objective]]

Within our set of experiments associated with transfer learning using the proposed methodology [INSERT SECTION REFERENCE HERE], we intend to measure the performance benefit that transfer learning provides. As it has already been established in our previous research that there is a reduction in model performance when the size of the dataset is reduced, our main focus is to measure the variability of benefit that transfer learning provides when using small data. Furthermore, we intend to conduct a manual sensitivity analysis in order to extract the optimal hyperparameters from a pool of proposed values.

The aim of these experiments is to aid in providing a practical method of performing image classification tasks when there is limited available data to train, test and validate a model similar to our own.

4.1.2. Description

[[TODO: Tidy This Up]]

[[TODO: Mention The Seed Used and add a footnote to use the attached codebase to reproduce results.]]

[[TODO: Mention the fact that we're using two bloody datasets.]]

Following from our previous research, we concluded that a reduction in available data correlated to a reduction in model performance. As a result, we are concentrating on experiments using small data. Therefore, each experiment that was conducted used the following dataset sample sizes: 100%, 10%, 1%, 0.1%.

In order to search for a practical set of system parameters, we decided to also vary the activation function within the fully connected layers. Each experiment was initially performed using a standard sigmoid function [INSERT REFERENCE HERE] due to our initialisation strategy. However, we also ran each experiment using exponential linear units (ELU) [INSERT REFERENCE HERE] because this activation function enables non-linearities between layers. Moreover, ELU has a gradient that is similar to the natural gradient (smooth).

Finally, we also performed a manual sensitivity analysis by varying the learning rate values from the follow pool: 0.01, 0.001, 0.0001.

Based on the description of the components in the methodology for the transfer learning approach, we configured the experiment using three fully connected layers, two different types of activation function, four dataset subsamples and a pool of three different learning rates.

Combinations of the above result in 72 different experiments. Due to the amount of experiments and the limited computational resources, we set a fixed number of epochs for every experiment (20), and a fixed size for the mini-batches used for the stochastic gradient descent calculation (50).

After running the aforementioned experiments, we analysed the results based on the validation accuracy. Then, we selected the configurations that get the highest values for validation accuracy for dataset sizes of 1% and 0.1%. For further investigation we then perform new experiments with higher number of epochs to analyse the behaviour of the fully connected part of the architecture.

4.1.3. Results

[[TODO: Tidy This Up]]

In order to compare our results to baselines obtained in the previous report, we base our analysis of the transfer learning experiments on the validation accuracy.Overall, there is still an evident reduction of the performance of the architecture relate to the size of the datasets, as is shown in [figure X].

[INSERT FIGURE X HERE]

For each dataset, the configuration that provides the highest accuracy for the smallest dataset size are quite different, as seen in 1. It is specially important to note the activation functions. It is expected that the Sigmoid activation function gives better results since the uniform initialisation strategy is meant to benefit this activation function. However, in the case of expressions dataset, the best result is obtained with ELU, but the best accuracy using Sigmoid activations is not that far 0.30.

The proposed approach has provided more benefits to the clothes dataset than to the expressions one, as seen in 1 and 2. This gives us some clues about the domain of the images in every dataset. Based on these results, we can say that the "distance" between the clothes domain and ImageNet domain is smaller than the "distance" between the expressions domain and ImageNet domain.

In comparison with the baseline system, it is clearly evident that the performance of the proposed system is better. For the clothes dataset reaches a performance similar to the one obtained using data augmentation. However, for the expressions dataset, the performance surpasses the one obtained with data augmentation. This might indicate that the conversion from the original feature space (pixel intensity) to the VGG16 feature space is more beneficial for the expressions than for the clothes.

4.1.4. Interpretation and Discussion

[[TODO: Tidy This Up]]

Observing the behaviour of the fully connected layers for the smallest sizes of both datasets, it is evident that the system is more stable for the clothes than for the expressions as seen in figure behaviour.png In the case of the clothes, for 1% and 0.1% of the original size, the system reaches a point where it starts to overfit. However, it is noticeable how the configuration of the system is robust enough to try to minimise this situation. The history is different for the expressions. Here, the behaviour is completely different not just compared to the clothes, but between the sizes of the dataset. In the case of the 1% of the size, the configuration of the system is not robust enough to minimise the effect of overfitting. Once the overfitting starts, the system cannot make too much about it. However, for the case of 0.1%, the configuration of the system is robust, not just to minimise the effects of overfitting, but to recover from it. This is not something immediate and it takes some time.

4.2. Siamese Neural Network

4.2.1. Motivation

[[PLACEHOLDER]]

4.2.2. Description

[[PLACEHOLDER]]

4.2.3. Results

[[PLACEHOLDER]]

4.2.4. Interpretation and Discussion

[[PLACEHOLDER]]

5. Related Work

[[PLACEHOLDER]]

6. Conclusions

[[PLACEHOLDER]]

[[REFERENCES GO HERE]]

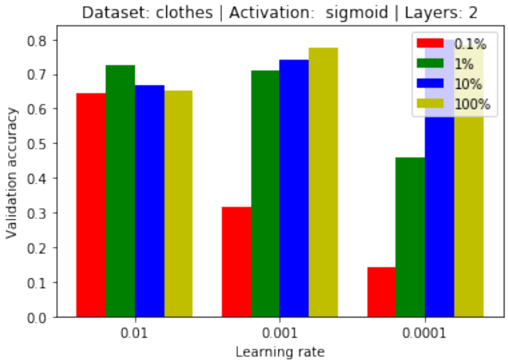

Figure 1. [[INSERT CAPTION HERE]]

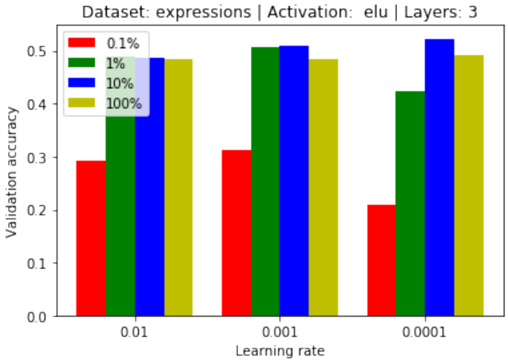

Figure 2. [[INSERT CAPTION HERE]]

Dataset	**Validation accuracy**	**Fully connected layers**	**Activation**	**Learning rates**
Clothes	0.64	2	Sigmoid	0.01
Expressions	0.31	3	ELU	0.001

Table 1. Configuration that allows the highest accuracy for size of 0.1% for clothes and expressions datasets

Dataset	**Baseline**	**Data Augmentation**	**Transfer learning**
Clothes	0.50	0.76	0.75
Expressions	0.32	0.44	0.50

Table 2. Validation accuracy comparison between baseline system and approaches using data augmentation and transfer learning for size of 1% for clothes and expressions datasets