Understanding of Deep Neural Networks and Their Applications in Real-World Challenges

Student Researcher: Ian D. Stebbins

Advisor: Dr. Salimeh Yasaei Sekeh

Bowdoin College / University of Maine
School of Computing and Information Science

Abstract

Deep neural networks (DNNs) are a subset of machine learning, a field in which an architecture of computer algorithms can draw inferences from sets of data, "learn", and furthermore make predictions. The most modern and powerful machine learning algorithms are driven by DNNs. Typically, a multi-layer architecture of interconnected neurons or nodes, DNNs can accurately classify and predict, based on features of given data. For this project, I spent my summer with the Sekeh Lab, a machine learning and artificial intelligence (AI) lab based at The University of Maine. This project was dedicated to learning concepts such as classification, model architecture, optimizers, accuracy, efficiency, compression, and pruning. A parallel focus was the utilization of programming skills to implement and create DNNs in python. The result is a formal set of classification experiments, a comparison of the accuracy of three different deep-learning models on the MNIST and CIFAR-10 datasets.

Learning Objectives and Methodology

The main type of problem focused on was supervised classification, a problem in which a model is trained on a set of predictor features with assigned class labels. The resulting trained model is then used to predict and assign class labels for values in which the predictor features are known, but the class labels are unknown or not given [1]. A basic instance of this type of problem can be simulated using the multi-layer perceptron (MLP) model, Iris flower dataset, and the Python package Scikit-learn. More complex color image datasets, however, often lead to a decrease in the accuracy of the neural network.

Convolutional neural networks or CNNs work using multidimensional data and significantly outmatch MLPs at image classification tasks. Benchmark CNN architectures such as AlexNet, VGG-16, and ResNet-50 have revolutionized the field of computer vision and artificial intelligence [3][4][5]. These types of models can be recreated and utilized using python packages such as Keras and TensorFlow, however, can be inefficient in comparison to smaller models. Large CNNs can have millions of trainable parameters resulting in computationally expensive models with high memory requirements. Such size is limiting to large-scale deployment, mobile use, and many other applications of deep CNNs [6].

The answer to oversized models is Compression, a broad term to describe the slimming or size reduction of neural networks. This is especially pertinent today as models get increasingly accurate, but increasingly complex, large, and incompatible with many systems. Pruning is one type of compression technique that involves removing some of the learned weights from a particular model, reducing its size and expense. Many approaches to pruning can be taken, each trying to preserve the accuracy of the network while reducing its size as much as possible [7]. Novel pruning methods can make a drastic and immediate impact and are the ongoing focus of a project started during the summer with a fellow Sekeh Lab member.

Experiment Objectives and Methodology

The main objective of the classification experiment was to create a formal comparison of basic and complex deep learning models on datasets of differing complexity. Additionally, the model development process and testing layout can serve as a baseline for future pruning experiments. The experiment utilizes three basic deep learning models, a small multi-layer perceptron (MLP) model, a 2D convolutional neural network (CNN), and a transfer learning model. All the models were coded in python using Keras and TensorFlow with assistance from NumPy for data preprocessing.

The MLP architecture implements a dense layer of size 128 with a rectified linear unit (ReLU) activation function and a second dense output layer of size 10 with a SoftMax activation. The model uses categorical cross entropy as a loss function and Adam as an optimizer. The CNN architecture implements three 2D convolutional layers each with ReLU activation, two 2x2 max-pooling layers, a dense layer of size 128 with ReLU activation, a 25% dropout rate layer, and a dense output layer of size 10 with a SoftMax activation. It utilizes categorical cross-entropy loss and RMSprop activation. The transfer learning model implemented the VGG-16 architecture with pre-trained weights from ImageNet, while having added trainable layers on top. The added trainable architecture implemented dense layers of size 256 and 128 with ReLU activation and a dense output layer of size 10 with a SoftMax activation. It utilizes categorical cross-entropy loss and RMSprop activation.

Data used for this experiment was MNIST, a dataset of 60,000 greyscale images of handwritten digits, and CIFAR-10, a dataset of 60,000 color images of 10 different classes for training and testing [8][9]. Each model was trained for 10 epochs with an 80/10/10 ratio for training/validation/testing data. The Matplotlib and Tabulate packages were used to generate the figures for visual aid and comparison.

While no large-scale network compression was done via pruning, future implementations will attempt to tackle this problem.

Results Obtained

As seen in figure 1, all models performed relatively well at classifying the MNIST dataset. The testing data shown in figure 3 exhibit greater than 95% accuracy for all models. Notably, the transfer learning model was the worst performer on MNIST. In comparison, model accuracy on the CIFAR-10 dataset shows a drop in the performance of all models. The MLP was the worst performer on this color image set with a performance of 41.5% accuracy (figure 4). As seen in figure 2, the transfer model initially performed better than the CNN, however over 10 training epochs the CNN improves much more to an accuracy of about 74.8%.

Significance and Interpretation of Results

Although the transfer model worked well, it did not match or better the performance of the custom CNN. One issue may have been errors upscaling the 32x32 CIFAR-10 images to the input VGG size of 224x224. Given this, a fully customized CNN trained on one dataset, however, can be more accurate than a generalized one. While the MLP performed well initially on the greyscale handwritten digits, just one small step toward more complex image classification led to a tremendous reduction in accuracy. This exhibits the importance of larger models and CNNs in image classification and computer vision. The fields of hyperspectral imaging (HSI), satellite image classification, and remote sensing especially have vast potential to utilize CNNs [10]. To name a few, glacier mapping, remote sensing of sea ice, landslide detection and prediction, wildfire detection and prediction, flood risk and detection, remote sensing of water quality, agricultural crop mapping, and natural disaster mapping and response, can all be revolutionized using large DNNs. Evidently, network compression is extremely important. Developing accurate models, but also making them a usable and efficient size will aid in solving a vast array of problems. Future work will be focused on network pruning and applying this to remote sensing problems.

Figures/Charts

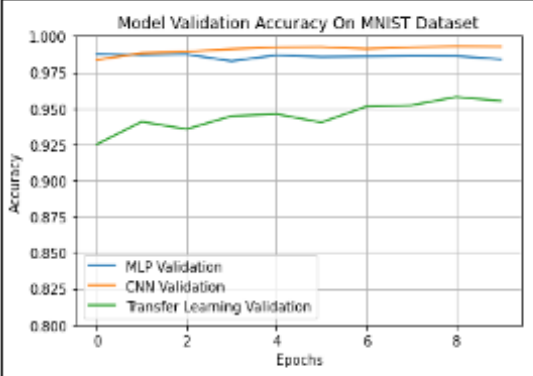


Fig 1. Comparing Model Accuracies on MNIST Validation Data over Training Epochs

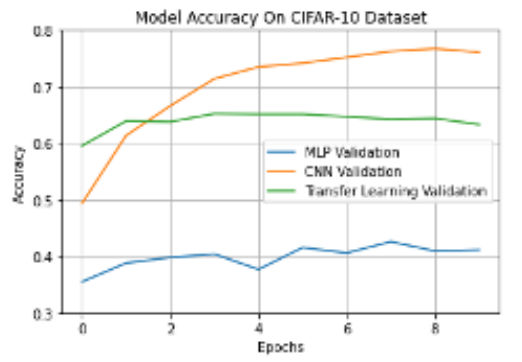
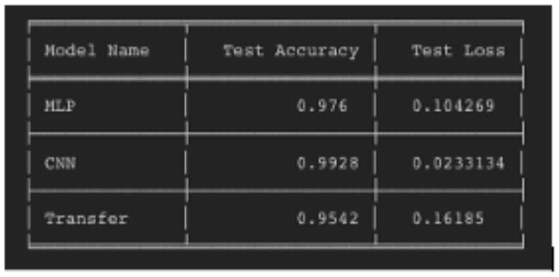
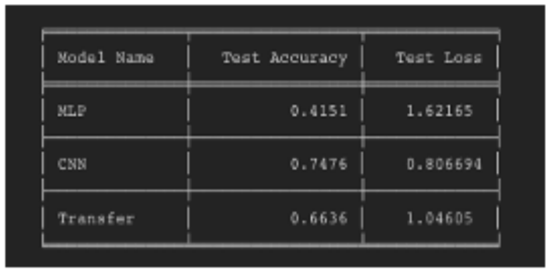


Fig 2. Comparing Model Accuracies on CIFAR-10 Validation Data over Training Epochs



```
| Model Name | Test Accuracy | Test Loss |
|------------+---------------+-----------|
| MLP        |         0.976 | 0.104269  |
|------------+---------------+-----------|
| CNN        |        0.9928 | 0.0233134 |
|------------+---------------+-----------|
| Transfer   |        0.9542 | 0.16185   |
```

Fig 3. Comparing Accuracy on MNIST Testing Data



```
| Model Name | Test Accuracy | Test Loss |
|------------+---------------+-----------|
| MLP        |        0.4151 |   1.62165 |
|------------+---------------+-----------|
| CNN        |        0.7476 |  0.806694 |
|------------+---------------+-----------|
| Transfer   |        0.6636 |   1.04605 |
```

Fig 4. Comparing Accuracy on CIFAR-10 Testing Data

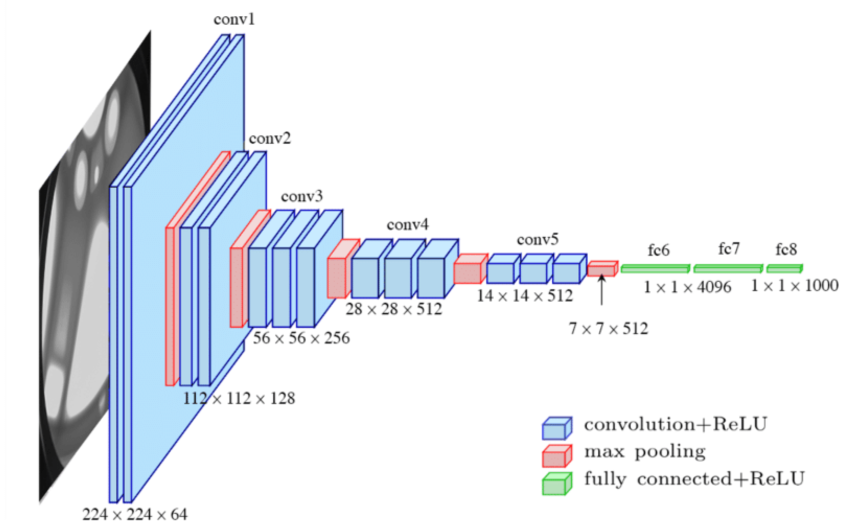


Fig 5. VGG-16 Architecture. Borrowed from Ferguson, Max & ak, Ronay & Lee, Yung-Tsun & Law, Kincho. (2017) [11]

Acknowledgments and References

I want to thank the Maine Space Grant Consortium for giving me the opportunity to be a part of their Institutional Collaboration Pilot Program Research Experience. Without the MSGC I would not have been able to connect with the Sekeh Lab and open doors into the world of machine learning and AI. I would also like to give a tremendous thanks to my advisor, Dr. Salimeh Yasaei Sekeh for welcoming me into her lab, and her continued instruction and support throughout the summer.

[1] Kotsiantis, S. B., Zaharakis, I. D., & Pintelas, P. E. (2006). Machine learning: a review of classification and combining techniques. *Artificial Intelligence Review*, *26*(3), 159-190.

[2] UCI Machine Learning Repository: Iris data set. (n.d.). Retrieved 2022, from https://archive.ics.uci.edu/ml/datasets/iris

[3] Krizhevsky, A., Sutskever, I., & Hinton, G. E. (2012). Imagenet classification with deep convolutional neural networks. *Advances in neural information processing systems*, *25*

[4] Simonyan, K., & Zisserman, A. (2014). Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*.

[5] He, K., Zhang, X., Ren, S., & Sun, J. (2016). Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition* (pp. 770-778).

[6] Ganesh, M. R., Corso, J. J., & Sekeh, S. Y. (2021, January). Mint: Deep network compression via mutual information-based neuron trimming. In *2020 25th International Conference on Pattern Recognition (ICPR)* (pp. 8251-8258). IEEE.

[7] Molchanov, P., Mallya, A., Tyree, S., Frosio, I., & Kautz, J. (2019). Importance estimation for neural network pruning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition* (pp. 11264-11272).

[8] *The mnist database*. MNIST handwritten digit database, Yann LeCun, Corinna Cortes and Chris Burges. (n.d.). Retrieved 2022, from http://yann.lecun.com/exdb/mnist/

[9] CIFAR-10 and CIFAR-100 datasets. (n.d.). Retrieved 2022, from https://www.cs.toronto.edu/~kriz/cifar.html

[10] Soucy, N., & Sekeh, S. Y. (2022). CEU-Net: Ensemble Semantic Segmentation of Hyperspectral Images Using Clustering. *arXiv preprint arXiv:2203.04873*.

[11] Ferguson, M., Ak, R., Lee, Y. T. T., & Law, K. H. (2017, December). Automatic localization of casting defects with convolutional neural networks. In *2017 IEEE international conference on big data (big data)* (pp. 1726-1735). IEEE.