ML Model Optimization Strategies for Image Datasets

Alexander Peplowski, Azfar Khoja, Daniel Wang
Montreal Institute for Learning Algorithms
Université de Montréal
Montréal, Québec

1 Introduction

Classification is one of several common tasks undertaken by machine learning. This paper provides an overview of some machine learning architectures (Support Vector Machines, Multi-Layer Perceptron, Convolutional Neural Networks) and evaluates their suitability for classification within the domain of computer vision. Then, we proceed to evaluate the performance of such architectures under varying hyperparameters and other conditions, such as neural network shape, regularization, and augmented data, to inquire into methods for optimizing model performance in respect of image classification.

1.1 Objectives

We will examine five optimization areas and their impact on some measure of performance:

Network Depth vs Performance How does changing the number of hidden layers used in MLP and CNN models affect model performance?

Data Availability vs Performance What is the impact of varying training set size and performing data augmentation on the performance of MLP and CNN models?

Network Node Count vs Performance What effect does the quantity of neurons have on performance? How should the number of NN nodes be selected?

Target Model vs Convergence Time How does selecting a particular ML model affect performance (test set accuracy) early in the training? How does performance change as the model trains?

Regularization vs Performance What is the effect of tuning a L2 regularizer on validation loss?

1.2 Datasets

The datasets below will be used to evaluate the performance of the models that we will examine.

F-MNIST The fashion-MNIST (6) dataset was created to be backward compatible with the MNIST dataset. It has the same image dimensions, number of training and test examples, and the same number of classes as the original MNIST digit set. This dataset contains 60,000 labelled training examples and 10,000 labelled test examples. The images are grey scale and have dimension 28x28 pixels. There are 10 different image classes.

CIFAR-10 The CIFAR-10 (2) dataset is a subset of the 80-million tiny images (5) data set It contains 60,000 labelled color images with dimensions 32x32 pixels and is subdivided into 50,000 training images and 10,000 test images. It has 10 image class labels that are balanced in the training/test sets.

32nd Conference on Neural Information Processing Systems (NeurIPS 2018), Montréal, Canada.

1.3 Baseline Models

To provide better control over the experiments, three baseline models were created. The experiments that we will perform will use these baseline models with the selected datasets while varying the target hyperparameter or dataset property of interest.

Convolutional Neural Network Our baseline CNN implementation is based off of the implementation by J. Lee Wei En (3), which was optimized for the CIFAR-10 dataset. For our CNN implementation, we removed dropout regularization since it would be difficult to approximate its effects across the model shapes and model types on which we would like to experiment. Instead, we used batch normalization as a form of regularization. Additionally, the dense layer size was reduced from 512 nodes to 256 nodes as to reduce the number of parameters in the network.

Aside from these changes, our baseline CNN model is the same. It has 4 convolutional layers that each use a (3x3) kernel. The max pooling layers, the feature map depths and the optimizer are the same as in the original implementation.

Multilayer Perceptron For our MLP baseline, a 3-layer model was arbitrarily chosen. After this arbitrary choice, we aim to force its properties to be as similar as possible to the CNN model. The number of nodes in the network was chosen such that the number of parameters in the MLP and CNN baselines are as similar as possible; in this case 490 nodes was chosen to fulfill this objective. The number of parameters are calculated to be: 871,018 trainable parameters for MLP and 871,530 trainable parameters for CNN. With a very similar number of trainable parameters, it will be easier to compare the MLP and CNN baseline models.

The other properties of the MLP are the same as the CNN baseline model: Batch Normalization is used after each layer and the Adam optimizer is used with default parameters.

Support Vector Machine We selected a SVM with linear kernel for our baseline model since the dimensionality of our datasets is large. The regularization term (alpha=0.01) was chosen via hyperparameter search on the F-MNIST dataset. The stochastic gradient descent optimizer was chosen since it is commonly used and is well-understood.

2 Experimental Procedure

In each experiment we will use both datasets to train each baseline model unless otherwise specified. Refer to each experiment for implementation details. The MLP and CNN models were created using Keras whereas the SVM model was created using scikit-learn (4).

Note that when we refer to "validation performance", we refer to the loss from evaluating the model on a random 10% split of the training set. When we refer to "test accuracy", we refer to the classification accuracy evaluated on the test set defined by each dataset.

2.1 Network Shape vs Performance

To compare the network shape we will create eight MLP and CNN implementations derived from the baseline MLP and CNN models. We would like to have the only variable in this experiment to be the number of hidden layers, not the number of parameters. Therefore, given the parameter count of each of the baseline models, we will generate new models which have approximately the same number of parameters given a fixed number of hidden layers. We will use from 1 to 8 layers in our experiment. The parameter count of each generated MLP and CNN model is summarized in table 2 and table 3

Given the set of models with approximate equal parameter counts, we are ready to begin the experiment. We will train the MLP model using the F-MNIST dataset and we will train the MLP models using the CIFAR-10 dataset and measure their validation loss.

2.2 Data Availability vs Performance

To understand the the effect of training set size on baseline models, we train on smaller stratified sets from CIFAR-10 and Fashion-MNIST and evaluate test set accuracy. In addition, we also investigate

the effect of data augmentation by mirroring all images horizontally and training on twice the data. We run experiments multiple times to account for variation and keep hyper-parameters constant.

2.3 Network Node Count vs Performance

To evaluate the node count impact on the performance of our NN models employed for image classification, we created new models whose number of nodes vary with respect to a scalar multiple of that of the baseline model node count (we use multiples 0.5x, 2x, 4x, 8x), in addition to the baseline models and we compared their performance on the two datasets, CIFAR-10 and Fashion MNIST.

2.4 Target Model vs Convergence Time

For this experiment, we would like to measure the performance of each model on the test set as a function of the number of training epochs. To do so, we will use the three baseline models each configured with a stochastic gradient descent (SGD) optimizer. We choose the same optimizer across the three models since the optimizer choice might have an impact on the training time. Also, the SGD learning rates are set to the default learning rates for each model (alpha=0.0001 for SVM, and alpha=0.01 for CNN, MLP). Note that only the F-MNIST dataset is used since it is computationally expensive to train a SVM using the CIFAR-10 dataset.

2.5 Regularization vs Performance

Here we explore the effect of L2 regularization on baseline models across datasets. We run a grid-search over L2 parameter values between $(10^{-4}, 0.3)$ and capture its effect on the model validation loss. We keep the other hyper-parameters constant and run the experiment multiple times to account for variances across each run.

3 Results

3.1 Network Shape vs Performance

When controlling for the number of parameters, we observe that when training on the F-MNIST dataset, the validation losses for all eight MLP models converge to approximately the same loss value. Refer to figure 5.

When controlling for the number of parameters and when training using the CIFAR-10 data set, all CNN models converge to a similar loss value, but some models have better performance than others. Refer to figure 4. In this experiment, the 4-layer model has the best performance.

3.2 Data Availability vs Performance

In the case of CIFAR-10, we see a gradual increase in test set accuracy as we increase the train set size for both CNN and MLP. Refer to figure 1. Whereas for F-MNIST, we see test accuracy saturate beyond using 60% of train set for CNN. Having augmented data helps achieve the best generalization across all models and datasets.

3.3 Network Node Count vs Performance

We observe that as the number of nodes increases, the validation loss decreases when training the CNN with four times the number of baseline nodes (see figure 2). However, the difference in the classification accuracy given by experimentation on the test data-set remain marginal for Fashion MNIST across the two neural networks, where the differences are measured to be approximately three percent. On the other hand, there were considerable improvements of upwards of fifteen percent in the test accuracy for the classification of CIFAR-10 data by the neural networks (both CNN and MLP) when increasing the number of nodes. However, it is worth noting that performance begins to worsen after a certain threshold of the number of the neurons is exceeded (refer to to table 1 in the appendix), which signals that the neural network has begun to over-fit.

3.4 Target Model vs Convergence Time

After just one epoch of training, we observe in figure 7 that the SVM has good performance on the test set at approximately 80% classification accuracy. It is followed by the MLP at 60% and the CNN at 10% accuracy. The relative performance of the models changes as we increase the number of training epochs. When we stop training after 15 epochs, the CNN provides the best performance, followed by the MLP model and then the SVM model.

3.5 Regularization vs Performance

There is no general trend to help with the selection of robust values for L2 regularization across datasets and models. Refer to figure 6. However as the graph illustrates their choice can make or break model performance.

4 Discussion and Conclusion

4.1 Network Shape vs Performance

When controlling for number of parameters, the number of hidden layers of the MLP model has little impact on performance for the F-MNIST dataset.

When controlling for number of parameters, the number of hidden convolutional layers is best at depth 4 for the CIFAR-10 dataset.

4.2 Data Availability vs Performance

With the right choice of hyper-parameters, using more data will improve model performance and augmenting the available data can be a good method for achieving better generalizability.

4.3 Network Node Count vs Performance

The results support the hypothesis that the capacity for a neural network to fit an arbitrary function rises when more nodes are incorporated to its layers. Moreover, our experiment provides evidence that validates some aspects of the universal approximation theorem (1) which states that a feed-forward neural network with one hidden layer may approximate any continuous function on compact subsets of $\mathbf{R}$. That is, given any continuous functions in compact subsets of $\mathbf{R}$, there exists a feed-forward neural network containing one hidden layer capable of approximating or fitting such a function. Although this idea encourages the addition of neurons to improve performance, excessive model capacity leads to over-fitting. Therefore, one should remain mindful of the capacity of the model and make an attempt to detect and prevent over-fitting.

We recommend using a method for hyperparameter tuning for finding an optimal number of neurons for some neural network to be implemented so that the best performance may be obtained. Some suggestions include, but are not limited to, Bayesian optimization and random search.

4.4 Target Model vs Convergence Time

SVMs with linear kernels require the fewest number of training epochs for good performance, but CNNs have the best performance on the F-MNIST dataset. Note that the SVM result might not generalize to higher-dimensional images since the computational cost of training a SVM algorithm gets large as the number of features (pixels and colors) increases.

4.5 Regularization vs Performance

High capacity models like CNN and MLP do not follow traditional validation curves used in machine learning. Hence an extensive L2 regularization parameter search might be expensive but rewarding experience.

Acknowledgments

Alexander Peplowski: Evaluation of network shape and performance, evaluation of algorithm convergence times and performance.

Azfar Khoja: Evaluation of performance using subsets of the train data and evaluation of validation loss on l2 regularizer parameter.

Daniel Wang: Design and implementation of neural networks (CNN and MLP) of various sizes.

References

[1] HORNIK, K.: Approximation capabilities of multilayer feedforward networks `http://www.vision.jhu.edu/teaching/learning/deeplearning18/assets/Hornik-91.pdf`

[2] Krizhevsky, A.: Learning multiple layers of features from tiny images `https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf`

[3] Lee, J.: Coding companion for intuitive deep learning part 2, `https://github.com/josephlee94/intuitive-deep-learning`

[4] Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., Duchesnay, E.: Scikit-learn: Machine learning in Python. Journal of Machine Learning Research **12**, 2825–2830 (2011)

[5] Torralba, A., Fergus, R., Freeman, W.T.: 80 million tiny images: A large data set for nonparametric object and scene recognition. IEEE Transactions on Pattern Analysis and Machine Intelligence **30**(11), 1958–1970 (Nov 2008)

[6] Xiao, H., Rasul, K., Vollgraf, R.: Fashion-mnist: a novel image dataset for benchmarking machine learning algorithms (2017)

Appendix

Table 1: Neural Network Performance based on the Number of Neurons

Test Accuracy based on Number of Neurons(N) (Scalar Multiple of the Baseline Number)					
Model and Dataset	0.5N	2N	4N	8N	16N
CNN CIFAR-10	0.6742	0.7503	0.8	0.8211	Insufficient RAM
CNN Fashion-MNIST	0.899	0.9263	0.9245	0.9258	Insufficient RAM
MLP CIFAR-10	0.3123	0.348	0.3784	0.4022	0.1808
MLP Fashion-MNIST	0.83	0.8621	0.8693	0.8057	0.4546

Table 2: Number of Parameters for MLP Models

Depth	Nodes per Layer	Number of Parameters
1	1096	871330
2	617	871831
3	490	870740
4	422	871018
5	377	869749
6	345	871135
7	320	870730
8	300	870610

Table 3: Number of Parameters from CNN Convolutional Layers Only

Depth	Kernel Size	Feature Map Depth	Nb Parameters	Error rel. Baseline
1	18x18	64	62272	-5.0%
2	4x4	64	68736	4.8%
3	4x4	39	66286	1.1%
4	3x3	32	65568	Baseline
5	3x3	27	64804	-1.2%
6	3x3	23	65236	-0.5%
7	3x3	21	65632	0.1%
8	3x3	19	64808	-1.2%

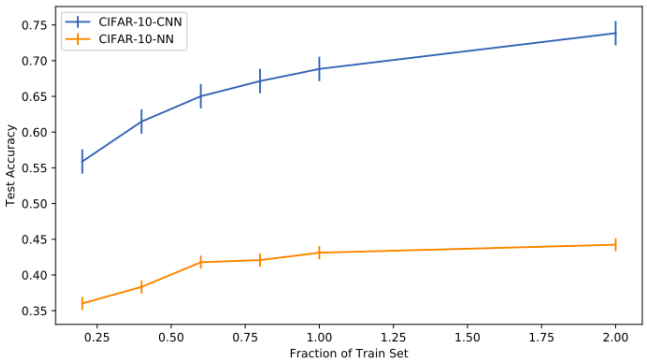

Figure 1: Test set accuracy using different fractions of the train set

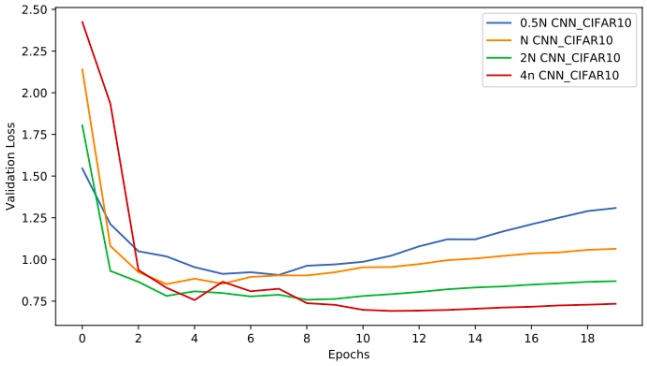

Figure 2: Validation Losses of CNNs of different sizes

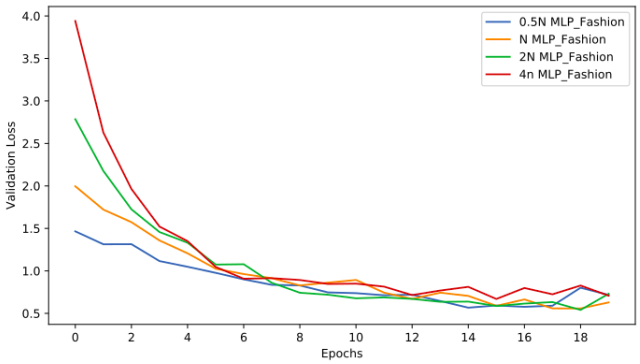

Figure 3: Validation Losses of MLPs of different sizes

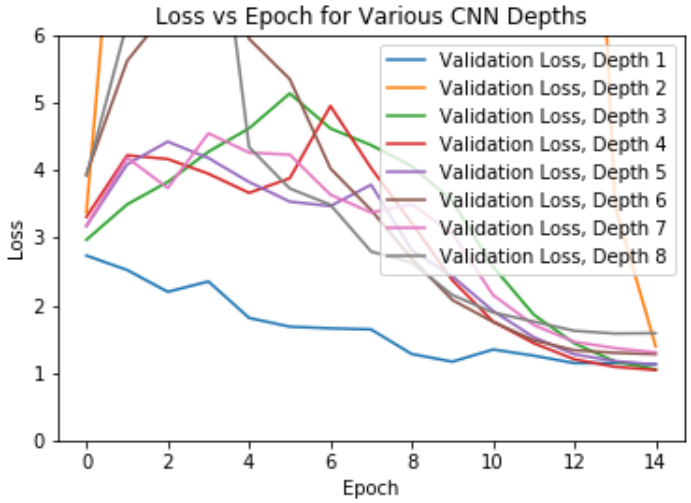

Figure 4: Comparing Loss Values for Various CNN Depths

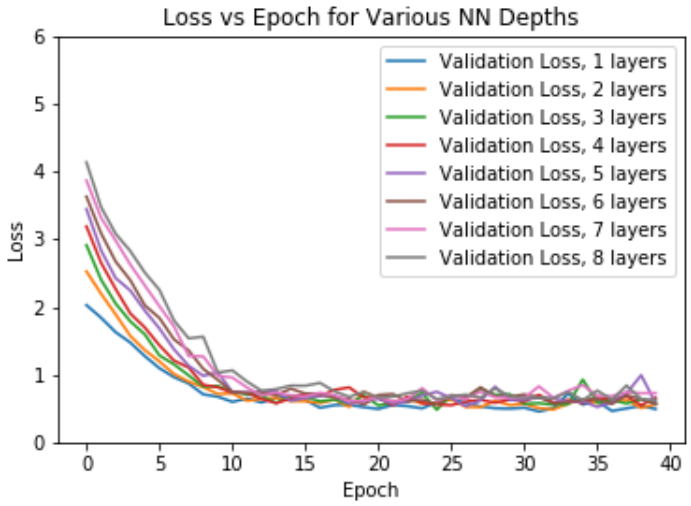

Figure 5: Comparing Loss Values for Various MLP Depths

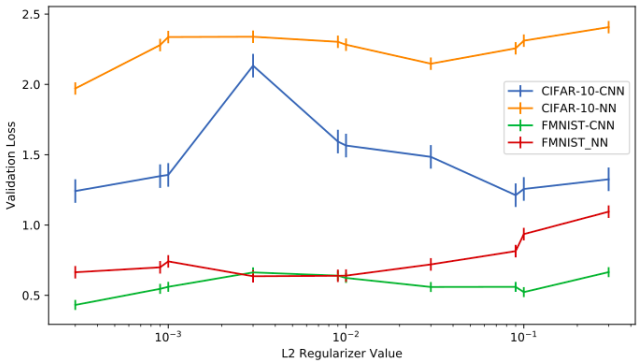

Figure 6: Impact of L2 regularizer parameter on validation loss

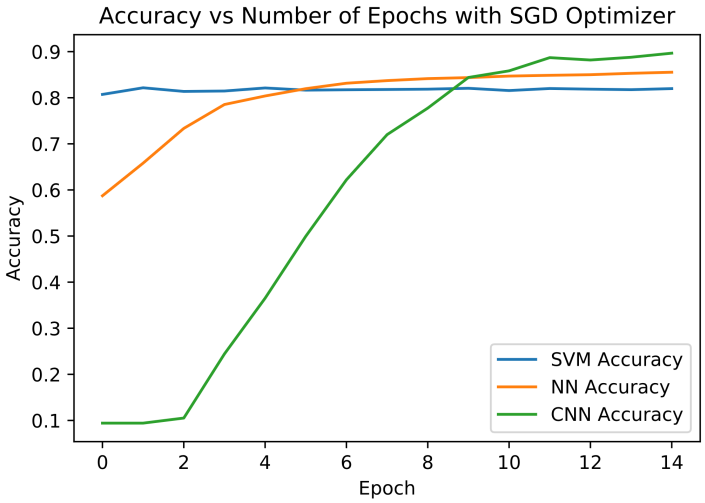

Figure 7: Test Set Accuracy of Various Models over Epochs Trained