Image classification:
an attempt at surpassing the Convolutional Neural Network

Arnold Kokoroko
Computer Science and Operations Research
Université de Montréal
Montréal, Qc
arnold.kokoroko@umontreal.ca

Ilyas Amaniss
Computer Science and Operations Research
Université de Montréal
Montréal, Qc
ilyas.amaniss@umontreal.ca

Joshua Amelia
Computer Science and Operations Research
Université de Montréal
Montréal, Qc
joshua.amelia@umontreal.ca

1 Introduction

Convolutional neural networks (CNN) are known to be very accurate and efficient with image classification tasks. Therefore, we decided to compare the performance of a CNN with a Multilayer Perceptron model (MLP) and a Random Forest classifier. To explore this question, three image datasets were considered: Fashion MNIST, CIFAR-10 and HASYv2. Using those three datasets, the three models described above will be trained, their hyperparameters fine-tuned and their performance compared to determine if either the random forest classifier or the multilayer perceptron can surpass the performances of a convolutional neural network.

2 Classifiers

2.1 Multilayer Perceptron

The Multilayer perceptron is an artificial neural network, and one of the most generalized among them. It is composed of one or multiple fully connected hidden layers in addition to an input and output layer and uses backpropagation to train the network. We used the ReLU activation function in the hidden layers to avoid the vanishing gradient problem. Moreover, in this experiment the MLP and the CNN roughly have the same total amount of parameters to better compare the two models.

2.2 Convolutional Neural Network

CNNs are a class of deep neural networks. They apply a series of convolutions and pooling on their input allowing them to extract meaningful features from a high input space. This way, they require fewer parameters for the same input space than a fully connected network. A toned-down version of the VGG network architecture[1] was used for its proven feature extraction ability.

2.3 Random Forest

By utilizing the principles of bagging and being careful with overfitting to the training data, the random forest gains by taking a decision based on the predictions of multiple decision trees. Many implementations of this classifier exist, such as constructing decision trees with different set of features, or having some trees with bigger weight than others.

3 Datasets

3.1 Fashion MNIST

The Fashion-MNIST [2] is a standard replacement for the common MNIST handwritten digit dataset. It comprises of 28x28 greyscale images of articles of clothing divided into 10 classes.

3.2 CIFAR-10

The CIFAR-10 [3] is a collection of 32x32 color images of 10 various classes such as cats, airplanes and birds. It is a popular dataset in image classification and it provides more complexity than the Fashion-MNIST with its 3 input channels and the different spatial representation of its images.

3.3 HASYv2

The HASYv2 [4] comprises of 32x32 images of 369 handwritten symbol classes. It is a dataset with lower state-of-the-art results due to the variance of its content. This will increase the diversity of our results and it may give us additional insight on the behavior of our classifiers.

4 Feature Design

4.1 Histogram of Oriented Gradients

Histogram of Oriented Gradients (HOG) finds the orientation of gradients in cells of an image [5]. For our experiments we used 4x4 pixel cells and segmented the gradients into 8 orientations. This effectively becomes an edge finder for the image and reduces the input size to $\sim \frac{1}{2}$ for greyscale images and to $\sim \frac{1}{6}$ for colour images.

5 Experiments

5.1 Multilayer Perceptron

Classifier's Input

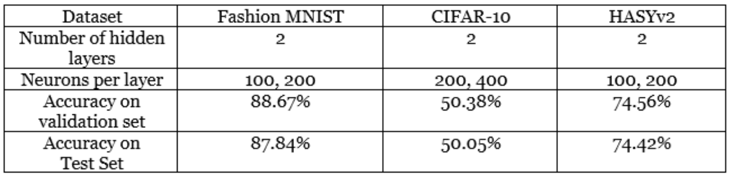
For each dataset three methods of inputting the image were tested for the MLP: directly inputting the pixels, converting to greyscale then inputting the pixels and calculating the HOG and inputting the HOG for the image. The HOG method performed significantly worse ($\sim 8\%$) on the validation for the different datasets so was not used with the MLP. Fashion-MNIST and HASYv2 performed the best with greyscale images. Fashion-MINST was already in greyscale and HASYv2 was black and white images with 3 channel pixels. CIFAR-10 performed the best with colour pixel input.

Training and fine-tuning

The model was trained using mini-batch gradient decent and we found the number of epochs that gave the best results on the validation dataset. We used 10% of the training dataset as validation data for fine-tuning. We tested with 1-3 hidden layers and 50-1000 neurons in each layer. Our best results were at 2 layers with 100 and 200 neurons for the Fashion-MNIST and HASYv2, and 2 layers with 200 and 400 neurons for the CIFAR-10. There were negligible increases in performance for large increases parameters when we tried with more layers or neurons per layer.

Results

The model with the optimal hyperparameters was tested on the test dataset. The results are summarized in figure 1. The MLP performed well (88% accuracy) on the Fasion-MNIST. This is likely due to the few classes and limited input size. The performance dropped on the HASYv2 (74% accuracy) and the MLP performed poorly on the CIFAR-10 (47% accuracy).



Dataset	Fashion MNIST	CIFAR-10	HASYv2
Number of hidden layers	2	2	2
Neurons per layer	100, 200	200, 400	100, 200
Accuracy on validation set	88.67%	50.38%	74.56%
Accuracy on Test Set	87.84%	50.05%	74.42%

Figure 1: Multilayer Perceptron accuracy with optimized hyperparameters

5.2 Convolutional Neural Network

Classifier's Input

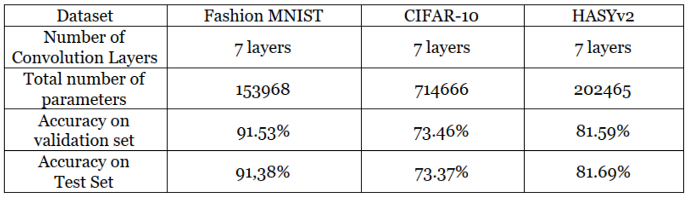
Unlike the MLP and the random forest, CNNs are designed to learn from high-dimensional inputs while using a reasonable amount of parameters. Therefore, the HOG method was not used for this classifier and the original input size of the images was kept.

Training and fine-tuning

The model was trained in a similar fashion than the MLP using mini-batch gradient decent. To avoid overfitting, the number of epochs was also found using the validation dataset. The model architecture replicated the VGG network architecture[1] using 3x3 convolutions with ReLU activation and 2x2 max-pooling layers. We then used a grid search on the different hyperparameters such as the number of kernels, learning rate, number of neurons and batch size to determine the best hyperparameters.

Results

The results of the CNN tested on the three datasets with its optimal hyperparameters are summarized in figure 2. As expected, it performed fairly well in all 3 datasets with its highest accuracy being on Fashion-MNIST (91%), followed by HASYv2 (81%) and CIFAR-10 (73%). It is important to note here that the focus was to compare the difference between the 3 classifiers as opposed to build a complex CNN. Therefore, methods used to obtain state-of-the-art results were avoided such as adaptive learning rate and momentum for optimization, and dropout or elastic net for regularization.



Dataset	Fashion MNIST	CIFAR-10	HASYv2
Number of Convolution Layers	7 layers	7 layers	7 layers
Total number of parameters	153968	714666	202465
Accuracy on validation set	91.53%	73.46%	81.59%
Accuracy on Test Set	91,38%	73.37%	81.69%

Figure 2: Convolutional Neural Network accuracy with optimized hyperparameters

5.3 Random Forest

Classifier's Input

For the random forest classifier, the first model was built simply by flattening the images and feeding them to the classifier for training. In order for the random forest to better classify the images, we implemented an additional model using the HOG features extraction with the hope of improving its performance compared to simply using the raw pixels of the images (refer to section 4).

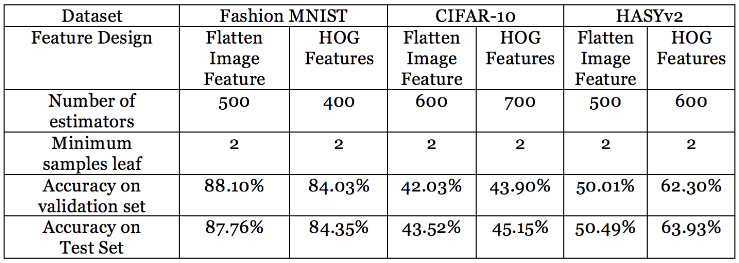
Training and fine-tuning

The two models were trained using 3-fold cross validation using the entire provided training set and tested on the provided test set. The 3-fold cross validation allowed to fine tune the following hyperparameters of the models using a grid search approach: the number of estimators used and the minimum number of samples required to be left from the left and right branches at a leaf node. The number of estimators varied from 400, 500, 600, 700 and the minimum samples leaf from 2, 3 or 4.

Then following the grid search approach with the different folds, the model with the highest accuracy on the validation set was selected for each dataset.

Results

Using the models with their optimized hyperparameters, each random forest classifier was then tested on their respective test set. The results are summarized in figure 3. The best performance were obtained on the Fashion MNIST dataset using the flatten images with an accuracy of 87.76%.



Dataset	Fashion MNIST		CIFAR-10		HASYv2	
Feature Design	Flatten Image Feature	HOG Features	Flatten Image Feature	HOG Features	Flatten Image Feature	HOG Features
Number of estimators	500	400	600	700	500	600
Minimum samples leaf	2	2	2	2	2	2
Accuracy on validation set	88.10%	84.03%	42.03%	43.90%	50.01%	62.30%
Accuracy on Test Set	87.76%	84.35%	43.52%	45.15%	50.49%	63.93%

Figure 3: Random Forest accuracy with optimized hyperparameters

An interesting observation is that the random forest classifier had better performances on the Fashion MNIST, and worst performances on the CIFAR-10. This difference in accuracy can be due to the format of the images processed. Indeed, the Fashion MNIST dataset is made of 10 classes of 28x28 greyscale images, which are easier to learn from compared to the 32x32 colored images of the CIFAR-10 or the 369 output classes of the HASYv2 dataset. Another interesting behavior was to see better performance using the flatten image instead of the HOG features.

6 Conclusion

6.1 Convolutional Neural Network is the clear winner

In the end, the CNN stayed on top of the image classification task with its ability to analyze visual imagery. Nonetheless, the MLP and Random Forest gave surprisingly good results when classifying the Fashion MNIST dataset with only 4% difference compared to the CNN. They are able to extract features and classify images almost as well when working with simpler inputs. However, with high dimensional inputs, the capacity of the MLP and Random Forest to learn features is restricted and they are more likely to memorize the data and overfit. They also have a limited sense of spatial representation compared to the CNN with its local connectivity and translation invariance attributes. However, we have learned that in some cases using a strong feature extraction technique can help solve this problem, and this technique should always be considered when optimizing classifiers.

6.2 Further improvements

Other feature extraction techniques could be used such as Scale Invariant Feature Transform (SIFT) or Speeded-Up Robust Features (SURF) which use the difference of Gaussians and its second derivative on the images to extract important keypoints. For the neural networks, optimization techniques such as a decaying learning rate, regularization techniques such as L1, L2 and dropout, and also data augmentation could have been applied before and during training to improve performances.

7 Acknowledgements

We certify that this report is our own work, based on our personal study and research, and that we have acknowledged all material and sources used.

Link to the Repository on Google Drive

https://drive.google.com/open?id=1mzdubqLkq8G5l32BRUJjbAzYGbtjrGrh

References

[1] K. Simonyan and A. Zisserman, "Very deep convolutional networks for large-scale image recognition," *arXiv 1409.1556*, 09 2014.

[2] Zalando, "Fashion-mnist." `https://github.com/zalandoresearch/fashion-mnist`. Accessed: 2019-12-14.

[3] U. of Toronto, "Cifar-10." `https://www.cs.toronto.edu/~kriz/cifar.html`. Accessed: 2019-12-14.

[4] M. Thoma, "The hasyv2 dataset." `https://arxiv.org/pdf/1701.08380.pdf`. abs 1701.08380, 01 2017.

[5] Wikipedia, "Histogram of oriented gradients." `https://en.wikipedia.org/wiki/Histogram_of_oriented_gradients`. Accessed: 2019-12-14.