Convolutional Neural Network Based Classification of Human Facial Expressions

Final Project for IS622 Machine Learning

Aadi Kalloo

Due December 18, 2016

Contents

Abstract

A convolutional neural network was developed for the purposes of classifying human facial expressions. Experiments were carried out to compare the performance of different models under the following conditions: without use of adversarial images, with use of adversarial images, and each of the two with use of edge-detection filters. It was found that using the original images with adversarial augmentation produced the highest performing model of the variations tested. The results of the best model produced here effected the achievement of 6th place in the associated Kaggle competition, with a score of 0.65 on the test dataset.

Keywords: facial expressions, classification, multi-class, adversarial

Introduction

Recognition of human communications, expressions, and interactions – especially through machine learning applications – has been an active field of research for over 10 years [Bartlett et al., 2004; Pantic and Bartlett, 2007]. However, given the wide array of emotions that humans can use to interact with one another, the ability of computer systems to recognize and classify emotion as communicated through facial expressions remains a challenging problem. The challenge in this task comes from three main sources: 1) very minute changes in facial muscle tension or position can result in a change of emotion class; 2) the lack of readily-available sources of training data means that prepared data sets may suffer issues of class balance, and lack of generalizability to arbitrary new data; and 3) human inter-rater reliability can be poor for more ambiguous / less "extreme" variants of the same emotion. Certain studies have shown that even healthy subjects can have a difficult time identifying facial expressions and, as a result, can have poor agreement when it comes to labeling certain emotions or expressions [Elfenbein et al., 2002; Guaita et al., 2009]. The culmination of these factors can result in a problem quite difficult for computers to solve.

In contrast to facial recognition (where the position and size of a face is detected), facial expression recognition attempts to classify the emotion or expression being shown on the face given facial cues and features. While many expressions are universal, the specific muscles used to create a particular expression can vary slightly across individuals. In this project, seven of the basic emotions were used: happiness, sadness, neutral, fear, anger, surprise, disgust. These emotions were chosen as they have been shown to be recognizable across cultures, and this ubiquity allows for a generalizable machine learning task. Research in this area aims to bring applications to the field ranging from robotics to medicine [Schmidhuber, 2015; Li et al., 2014; Prasoon et al., 2013].

The approach presented here uses a version of Artificial Neural Networks (ANN), called Convolutional Neural Networks (CNN), which are a special kind of ANN and have been shown to work well toward analyzing images through extraction of image features. Two main experiments were designed in order to find the best performing CNN model; each experiment compared the use of the Adagrad, Adadelta, and combination Adagrad+Adadelta optimizers.

Methodology

I Preprocessing

Each of the 13,719 images in the image set was scaled down to 48x48 pixels. All images were converted to grayscale with only a single channel and normalized to contain values between 0 and 255. Fifty percent of the data set were selected for generation of adversarial images. The data set was then split into training (80%) and validation (20%) sets.

II Network Architecture

For this project, a classical feed-forward CNN was developed using Keras for Python with the Theano backend. The network used consisted of six convolutional layers with 48 filters used for the first two, and 36 filters used thereafter. These employed kernel sizes of 3x3 followed by ReLU (Rectified Linear Unit) activation layers. A max pooling layer was placed after each convolutional layer. These convolutional layers were then followed by fully connected layers with two hidden layers of 200 nodes and 100 nodes, respectively. Finally, a softmax layer was used for conversion to designated classes.

Since the data set was not well-balanced (contained 41.8% happy, 49.7% neutral, 1.7% anger, 2.1% sad, 0.2% fear, 2.6% surprise, 1.7% disgust), it was of significant importance to compensate for this. As a means towards mending this gap, an optimizer was chosen that can adapt the learning rate independently for the parameters. Given its aptitude for dealing with sparse data, the Adagrad algorithm was chosen as the network's optimizer function. Given that the Adadelta optimizer was developed to be an improvement over Adagrad, it was chosen for comparison. When a non-adaptive algorithm was used, such as Stochastic Gradient Descent, validation accuracy did not move past 40% for any epoch of the training process. In addition, the 'class_weight' option was utilized in Keras, allowing the model to focus more on under-represented samples during training [Model Class API, n.d.]. Training was performed multiple times for a given set of parameters and a given training/validation set to ensure stability of results.

III Experimental Design

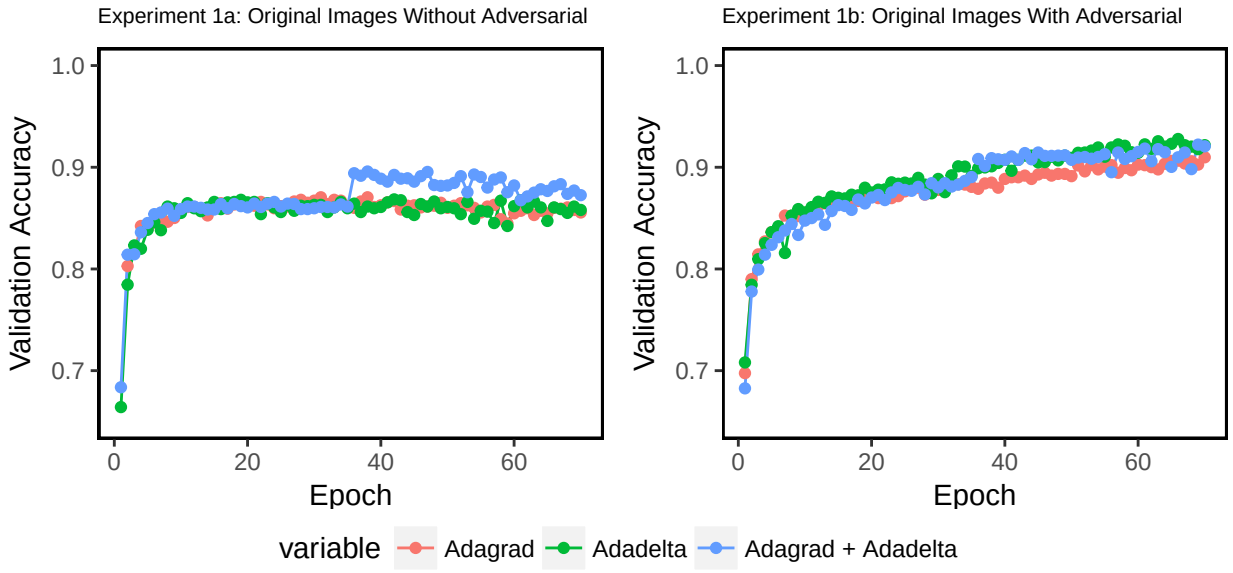
As an effort toward maximizing model performance, two main experiments were performed examining the performance of three optimizers: Adagrad, Adadelta, and Adagrad + Adadelta. For each of Adagrad and Adadelta training runs, 70 epochs were performed whereas Adagrad + Adadelta used 35 epochs each for Adagrad and Adadelta, respectively for a total of 70 epochs. The first experiment compared the three optimizers both with and without use of adversarial image generation, while the second experiment compared the three optimizers using the Canny edge detection filter. It was hypothesized that 1) the combination of Adagrad and Adadelta would perform better than either algorithm alone; 2) the use of adversarial image generation would produce better performance than without use of adversarial images; and 3) model performance could be enhanced through use of the Canny edge detection filter as this filter will highlight only features most salient in facial expression classification.

Below are examples of the image types used in Experiments 1a, 1b and 2, respectively:

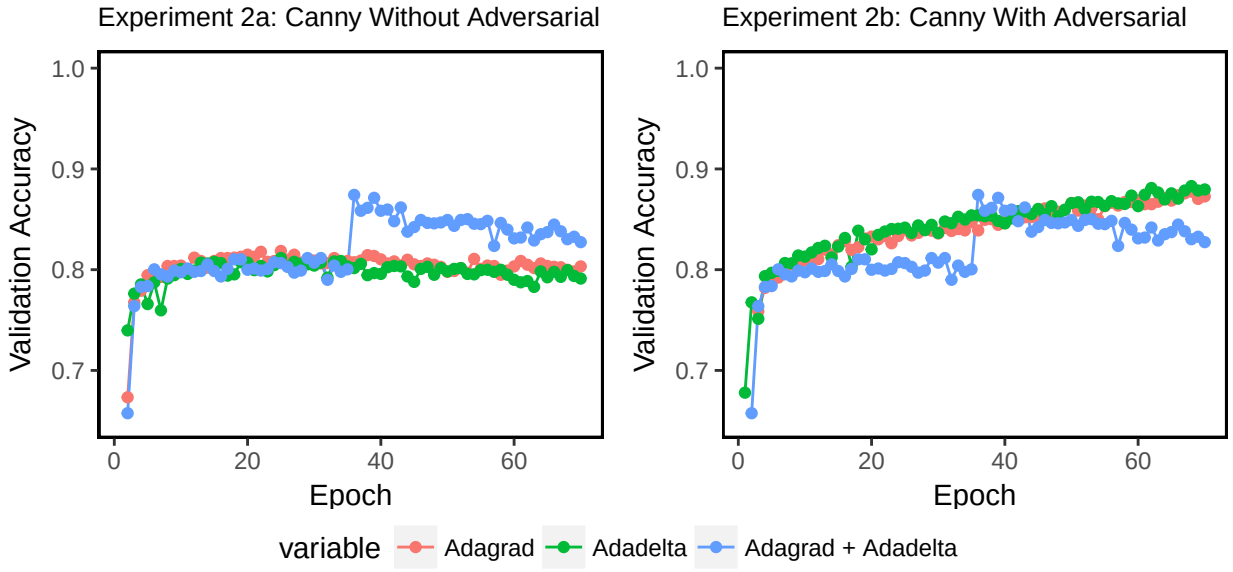
Figure 1: Example images used in Experiment 2. Left: Unaltered image. Center: Adversarial image generated using the following command in python – 0.01*np.sign(np.random.randint(-1, 1, (48, 48))). Right: Image generated using the Canny filter tool from the sklearn package.

Results

For original images (Experiment 1), training with adversarial images resulted in a model producing a validation accuracy of around 93%, while not using adversarial images resulted in a model that began overfitting and reached a peak validation accuracy of only 86%.



The graphs shown below highlight the differences in performance for use of Canny-filtered images when compared to the above classification models:



It was observed that the Adagrad + Adadelta training scheme significantly increased validation accuracy over Adagrad alone in Experiment 1a. However, in Experiment 1b, all three models performed similarly when adversarial images were used in the training process. Experiment 2 showed that the addition of Adadelta epochs to Adagrad can improve performance when using the Canny edge detection filter, but overall use of Canny images resulted in models inferior to those for the original images.

Discussion

The Facial Expressions Classification Project provided a thorough and challenging introduction to computer vision applications. The two experiments resulted in the development of several models, many of which disproved the initial hypotheses. In this project, it was found that models trained using the original non-Canny-filtered images but with adversarial image augmentation produced the highest validation accuracies.

Topics of particular investigation during this project, in addition to use of adversarial images, included filter count, network architecture, choice of optimizer. Filter count directly affected the amount of feature information fed into the fully-connected layers of the network. As filter count was varied through experimentation, it was found that its relationship with validation accuracy was analogous to the shape of the Shannon Entropy curve, as described by equation (1):

$$H(X) = -\sum_{i=1}^{n} P(x_i) log_b P(x_i)$$

If the number of features extracted was too little or too many, validation loss/accuracy either did not change for all epochs, or increased gradually but soon became "stuck" in local minima. This became evident through the halting of validation accuracy increases, usually around 40-60%. The same effect was seen in experimenting with network architecture – networks that were too large or too small resulted in decreased performance with respect to validation accuracy.

The observation that the addition of the Adadelta optimizer aided performance only for the non-adversarial data set can likely be attributed to the need for parameter-wise updates in learning rate. Since the data set without adversarial images was both smaller and less 'diverse', the monotonically decreasing learning rates of the Adagrad algorithm stalled increases in validation accuracy. As the Adadelta optimizer aims to address that very issue in the Adagrad algorithm, the performance of the smaller and adversarially-deficit dataset was enhanced. This effect was not seen in Experiment 1b likely due to the augmented availability of data within each class.

Future work on this project will include analysis using additional data sets, and examination of more advanced two-stage training methods such as the layer-transfer method outlined by Ding et al. (2014). More advanced CNN methods and larger data sets have the potential to produce high performing and well-generalizable networks capable of distinguishing more minute differences in facial expression variation.

References

Bartlett, M. S., Littlewort, G., Lainscsek, C., Fasel, I., & Movellan, J. (2004, October). Machine learning methods for fully automatic recognition of facial expressions and facial actions. In Systems, Man and Cybernetics, 2004 IEEE International Conference on (Vol. 1, pp. 592-597). IEEE.

Elfenbein, H. A., & Ambady, N. (2002). On the universality and cultural specificity of emotion recognition: a meta-analysis. Psychological bulletin, 128(2), 203.

Guaita, A., Malnati, M., Vaccaro, R., Pezzati, R., Marcionetti, J., Vitali, S. F., & Colombo, M. (2009). Impaired facial emotion recognition and preserved reactivity to facial expressions in people with severe dementia. Archives of gerontology and geriatrics, 49, 135-146.

Li, Q., Cai, W., Wang, X., Zhou, Y., Feng, D. D., & Chen, M. (2014, December). Medical image classification with convolutional neural network. In Control Automation Robotics & Vision (ICARCV), 2014 13th International Conference on (pp. 844-848). IEEE.

Model class API. (n.d.). Retrieved December 17, 2016, from https://keras.io/models/model/

Pantic, M., & Bartlett, M. S. (2007). Machine analysis of facial expressions (pp. 377-416). I-Tech Education and Publishing.

Prasoon, A., Petersen, K., Igel, C., Lauze, F., Dam, E., & Nielsen, M. (2013, September). Deep feature learning for knee cartilage segmentation using a triplanar convolutional neural network. In International Conference on Medical Image Computing and Computer-Assisted Intervention (pp. 246-253). Springer Berlin Heidelberg.

Schmidhuber, J. (2015). Deep learning in neural networks: An overview. Neural Networks, 61, 85-117.