EmoNet: Deep Learning for Gesture Recognition

Jordan Duprey
jduprey@stanford.edu

Abstract

In this paper I explore using audio and video features from the Acted Facial Expressions in the Wild dataset to improve upon classification accuracy of 7 emotions using a deep learning approach to emotion recognition. Using a combined model of a small ResNet feeding into an 3 layer LSTM to process individual video frames and a 3 layer neural net to process audio features that have been projected into a smaller subspace using LDA, I trained a rather large classifier that ultimately did not converge due to limited time and computational resources.

1. Introduction

Humans are complicated beings. Since the dawn of time, the way people interact and display their emotions in social settings have evolved into extremely complex and subtle combinations of audio and visual cues. Despite the technological innovations in communication technology, miscommunications due to misinterpretation of information is still as prevelant as ever. However, with the rise in popularity of artificial intelligence, this all could change. If current trends in social networking continue, and people continue to depend on online resources to serve as a platform for modern social networks, emotion recognition technology has the potential to truly augment these interactions like never before.

This paper seeks to explore a deep learning approach to emotion recognition that incorporates audio and video inputs from people in a wide variety of situations who are all experiencing similar emotions. By tackling the issues with intra-class variance for each emotion, the algorithm developed in this paper could potentially change the way we interact with other people online. Services such as Snapchat, Skype, and Facetime would be able to dynamically adjust users' experiences based on their mood. In fact, the implications of this technology spread far beyond the domain of social networking. Static sources of audio visual content such as Youtube, Instagram, and Facebook would be able to sort and search data based on people's emotions and reactions. Human computer interaction as we know it would be completely changed once the problem of emotion recognition is truly solved.

2. Previous Work

The main source of inspiration for this project was the Emotion Recognition in the Wild Challenge. Each year competitors work with two datasets, Acted Facial Expressions in the Wild (AFEW) and Static Facial Expression in the Wild (SFEW), to try to classify a hand labelled set of audio and video clips from various movies. Each of these clips feature an actor displaying one of seven emotions (Happy, Sad, Angry, Fear, Disgust, Surprise, Neutral).

In 2013, the first year this competition was held, Kahou, Bouthillier, and Lamblin [1] won the competition using a combination of deep learning techniques. In their paper EmoNets: Multimodal deep learning approaches for emotion recognition in video, they described how their algorithm focuses on building several models consisting of "a convolutional neural network, focusing on capturing visual information in detected faces, a deep belief net focusing on the representation of the audio stream, a K-Means based bag-of-mouths model, which extracts visual features around the mouth region and a relational autoencoder, which addresses spatio-temporal aspects of videos." Ultimately this approach resulted in a test set accuracy of .476 on the AFEW dataset.

Two years later, Yao, Shao, Ma and Chen of Intel Labs China won the competition with a different approach focusing on building a feature set based on expression-specific facial features [2]. They submitted many models, but the winning approach combined 3 linear SVMs trained with facial feature relations, an Audio model and a CNN model to achieve a test accuracy of .538 on the AFEW dataset.

For this experiment, I will attempt to improve on these approaches by using a state of the art CNN to process the input video before combining it with the audio features in a neural net. Additionally, I will be using the entire frame instead of just the faces since body language is important in expressing emotion as well. My hope is that by better capturing the subtle features from each frame of the video, I will be able to more accurately classify the emotions while using the same audio features.

Figure 1. Single frame from video clips displaying happy, sad and angry actors respectively

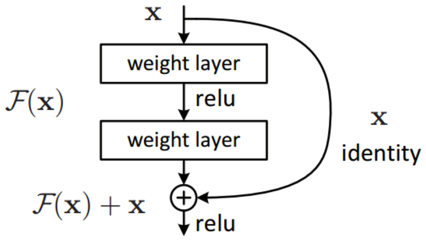
x
weight layer
$\mathcal{F}(\mathbf{x})$
relu
weight layer
x
identity
$\mathcal{F}(\mathbf{x}) + \mathbf{x}$
relu

Figure 2. This diagram demonstrates the main components of a ”residual block” in a ResNet

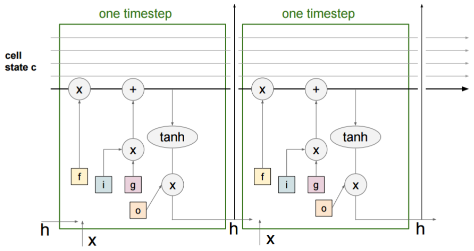

Figure 3. Visual representation of how the hidden components of the the LSTM model are calculated using lstm

3. Solution

This solution makes use of the same dataset, Acted Facial Expressions in the Wild, as the previous experiments did for the EmotiW challenge. For each of the seven categories there are roughly 100 short video clips provided for training, validation and testing. After splitting the data accordingly, I then trained one large neural net that incorporated all of the features.

3.1. Video

For the first portion of the model, I processed frames from the video clips in AFEW. Using the Unix command line tool ffmpeg, I sampled 25 frames per second of video and resized them into 256x256 images. Once the frames from the video were in a common format, I looked for a pre-trained convolutional neural network to process the individual images.

For that portion of the model I ended up using a model based off of ResNet developed by Microsoft for the ImageNet Large Scale Visual Recognition Challenge (ILSVRC) in 2015. This annual contest involves training and classifying CNN’s on two datasets CIFAR-10 and CIFAR-100, which have 10 and 100 image categories respectively. The fundamental building block of this architecture, the residual block, helped Microsoft’s 100 layer and 1000 layer models win the ILSVRC in 2015 with the record low error rate of .0357. In the interest of reducing training time, I chose to use a 20 layer model that was trained on CIFAR-10 which would perform slightly worse than the above models, but would still leverage this cutting edge technology to hopefully outperform the baseline established in previous year’s EmotiW submissions.

However, ResNet was only designed to process static images. For this model, I needed to process variable length sequences of related images. To preserve all this spatio-temporal information, I needed to connect the ResNet to a recurrent neural network. To do this, I removed the softmax, dense and average pooling layers at the end of the pretrained resnet and fed the output into a RNN. Unfortunately, vanilla RNN’s are subject to the issue of vanishing gradients so instead I built a 3 layer model using the LSTM algorithm as shown in Figure 3 [6].

3.2. Audio

For the audio portion of this model, I used the precalculated audio features that came with the dataset. This 1500 dimensional vector consisted of various features sampled from different points in the audio signal associated with each video clip. The majority of these features were related to the Mel-frequency cepstral coefficients (MFCC), a set of audio processing descriptors frequently used in state of the art voice recognition and music information retrieval systems [4]

Once I had loaded the vector of audio features, I looked

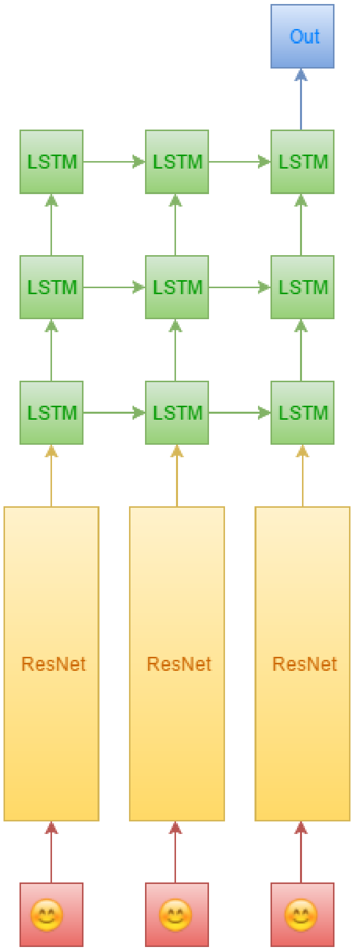

Figure 4. The video portion of the neural network

up a few different ways to reduce the dimension of the input to lower the runtime. I ultimately chose Linear Discriminant Analysis as the dimension reduction algorithm to experiment with. LDA builds upon the idea of maximizing variance in the projected subspace that is used in other dimension reduction algorithms like Principal Component Analysis. In the case of classification, once the data has been projected into a lower dimensional subspace it would be useful to maximize the variance between data points of different classes while minimizing the variance of data points within the same class. Ronald Fisher formally defined how LDA achieves this by defining the between class and within class scatter matrices, S_W and S_B respectively, as follows[3]:

$$S_W = \sum_{i=1}^{c} \sum_{x_k \in X} (x_k - \mu_i)(x_k - \mu_i)^T \tag{1}$$

$$S_B = \sum_{i=1}^{c} n_i(\mu_i - \mu)(\mu_i - \mu)^T \tag{2}$$

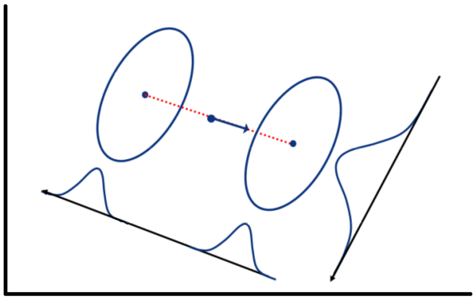

Figure 5. Projection from a 2-dimensional subspace into a 1-dimensional subspace along the vectors shown, where the one on the bottom maximizes the between class scatter, assuming the two ellipses represent two distinct classes

where c is the number of classes, X is the training set, and μ_i is the vector of mean values of class i. LDA attempts to find the projection matrix W_{opt} that maximizes the ratio between S_B and S_W. This can be solved by making the columns of W_{opt} the first k eigenvectors corresponding the largest k eigenvalues in the solution to the generalized eigenvalue problem[3].

$$S_B w_i = \lambda S_W w_i \tag{3}$$

For the purposes of this project I chose to use the values generated from projecting our data into a 512 dimensional subspace. Once these features had been loaded and projected into a smaller subspace, I passed it through a 3 layer neural net and added it to the existing model to be combined with the video features.

3.3. Putting it all together

Once the audio and video features had been calculated, I concatenated the resulting feature vectors. I then ran this combined feature vector through three more sequences of dense layers followed by rectifiable linear activations before adding a softmax activation at the very end. Additionally, I utilized a dropout rate of .5 after every relu layer in the model to avoid overfitting during training.

Once the softmax activation layer had been added, I optimized this model using 20 epochs of stochastic gradient descent using batch sizes of 16. This optimization involved calculating and minimizing the cross-entropy loss during each iteration to converge on an optimal model.

4. Experiments

Once I had designed the architecture for the project, I began prototyping the model in Keras using Python. I found a pretrained ResNet to expedite the training process and began my hyperparameter search. I started by keeping all

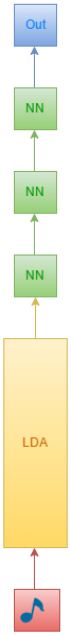

Figure 6. The audio portion of the neural network

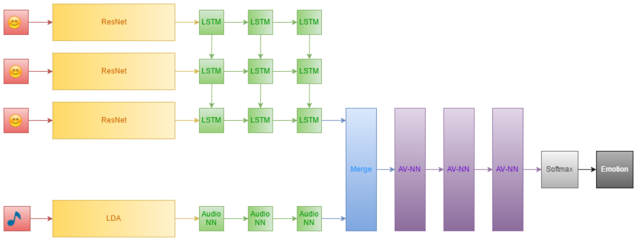

Figure 7. The full model incorporating both the audio and video features

other hyperparameters constant and only varying the learning rate. but this didn't prove to be enough. After a dozen epochs using different learning rates sampled from a uniform distribution between 1e-6 to 1e-8, the loss function was not showing any sign of converging properly. Increasing and decreasing this range didn't seem to dramatically affect the results either.

I then opted to use a different optimization function. Instead of stochastic gradient descent I chose to use the Adam algorithm [7]. This method of optimization combines the benefits of two separate optimization techniques, momentum and RMSProp optimization into one efficient method of locating local minima using a moving average of squared gradients. However, this also failed to properly minimize the loss function leading to an unusable model. Once again varying the learning rate proved to be insufficient and I ultimately ran out of time to train this rather deep network.

5. Conclusions and Moving forward

Ultimately the project did not produce a working model that was able to classify the video and audio features into the 7 categories outlined by the AFEW dataset. While this project was slightly too large in scope for the timeframe allotted for this project, current trends do suggest that deeprer neural networks with larger capacities are the way forward to truly improve upon classification accuracy as exemplified by the winning submission from the past 4 years at ILSVRC. Thus, with more time and computational resources, I plan to continue the hyperparameter search and optimize this model to get a set of weights that give more meaningful results.

References

[1] Kahou, Bouthillier, Lamblin, et. al *EmoNets: Multimodal deep learning approaches for emotion recognition in video*

[2] Yao, Shao, Ma, Chen *Capturing AU-Aware Facial Features and Their Latent Relations for Emotion Recognition in the Wild*

[3] Ekenel, Hazim Kemal and Rainer Stiefelhagen *Two-class Linear Discriminant Analysis for Face Recognition.*

[4] Muller, Meinard *Information Retrieval for Music and Motion.*

[5] He, Zhang, Ren, Sun *Deep Residual Learning for Image Recognition*

[6] Sepp Hochreiter, Juurgen Schmidhuber *Long Short-term Memory*

[7] Kingma, Ba *Adam: A Method for Stochastic Optimization*