Lip Reading

Héctor Otero Mediero
hoterome@uci.edu

Nikita Samarin
nsamarin@uci.edu

March 13, 2017

1 Introduction

Why is Lip Reading important?

2 Description of the Problem

Describe in detail going from video to what output.

3 Previous Work

Talk about papers in the table.

4 Dataset

Description of OuluVS and excuses of why don't we have a larger dataset. Videos, frames, information about box coordinates, and sound. How many frames in total, dimensions of the images. Extra images from other datasets.

5 Hardware

Macbook Pro: 2.4GHz dual-core Intel Core i5 processor, 8GB DDR3L, 256GB

6 Technical Approach

6.1 Using video frames

Our first attemps at tackling the problem followed a pure lip reading approach, this is, using only a sequence of still images (video frames) from a subject to predict the sentence that's said in the video. Previous works in the field used 3D Convolutional Neural Networks to process the sequence of images with succesful results, but our hardware didn't fit well this tasks as the computation requirements is large and a single epoch for processing 500 videos took around 2 hours. Due to the impossibility of following this path, we flattened the input to be able to process each image as a 1D vector and be able to process it with a Recurrent Neural Network or a 2D convolutional Network (after forming a matrix with all the different frames). The results were not good as the dimensions of the individual images was too big in comparison with the amount of examples yielding networks with too many parameters that overfitted the data.

Lip Segmentation

In order to reduce the dimensionality of the data, we decided to limit the section of the photo used as input. As it's obvious, the totality of the information stored in an image comes from the mouth of the subject and since the space it occupies is rather small in comparison with the image size it does a great job at condensing the information and reducing the amount of parameters in the network.

Since solving image segmentation problems using Neural Networks has been thoroughly studied, we designed our own. The problem at hand was finding the coordinates of the top left corner and the width and height of a bounding box that surrounded the subject's mouth. The dataset we were working with included labeled data of each one of the frames extracted from the videos with this information.

As it can be seen in Figure 1 the architecture used is composed by alternating 2D Convolutional layers with MaxPooling2D layers and a final Dense layer that completes the regression task by predicting 4 values from each image. The intuition behind this architecture is that, as in Visual Computing techniques, the convolutional layers will be used to extract features at different levels of detail and the pooling layers help at reducing further the dimensionality of the image. Finally, the dense layer with a linear activation produces an unbounded value, ideal for the regression task.

The results obtained are really accurate. After 25 training epochs and using 80% of the data for training and the rest for test, we obtain a Mean Squared Error of 95. The semantics of this error for our case are that, on average, the predictions for the coordinates of the box and the labeled data differ in only 9 pixels, which for a 576x720 images represents less than 2% of the width

and height.

Some of the reasons behind this accurate prediction are the availability of a large set of images to work with and the fact that the mouths are centered along the X-axis.

RNN Architecture

Using the previous information to crop the images, we tried feeding them to a recurrent neural network due to the fact that there's a relation among the images that can be expressed as a sequence. Since we're in front of a classification task, we use a Dense layer at the top with a softmax activation function that returns values for the different classes that can be interpreted as the probability of an example of belonging to a certain class.

Both types of RNNs available in Keras, LSTMs and SimpleRNNs, were used to test whether there was a need for establishing a relation between values distant in the video sequence. The architecture that yielded the best results used the latter and is shown below. The layer configuration responds to the scarcity of the data (just 1000 videos), which forced us to constrain our network to a small amount of parameters, and the subsequent need to avoid overfitting, which led us to include dropout layers after each of the recurrent ones and try different types of regularization (L2 with a value of 0.01 providing the best performance).

The results obtained barely improved a random classification of the data with a 16% accuracy using categorical crossentropy as loss function for the model. We think that a solution could be found using this method due to the semantic relation between the images as explained previously but recurrent networks that can process this kind of data need a lot of parameters and, in our case, the relation with the amount of input data was far from ideal. 2D Convolutional networks were also tested but they produced worse results.

6.2 Using video sound

Due to the poor results obtained using just images, we decided to try solving the same problem (going from a video to the sentence said) but using the sound instead since in most cases where lip reading can be used, sound is available too (even if it's present with noise). Intuitively, the same information is stored both in the video and in the sound, the second one being more dense and simpler than the first.

Data Preprocessing & Augmentation

In order to reduce the problems found with the previous approach, we opted to increase the amount of training examples. The audio present in the videos was noisy so we tried different approaches to lessen the effects that they could provoke in our network, generating new audios by applying a rolling-mean filter on the audio with a window size of 10 making reducing the noise effectively.

Apart from the noise, we tested different representations of the data, integer and floating point, and a different number of channels, estereo or mono, choosing in both cases the second option as it produced better results.

CNN Architecture

Facing again a vectorial representation of our input data (in this case amplitude values for the audio) we could again choose between a recurrent or a convolutional approach. By representing the data we saw that audios only differed one from the other on the amount of words pronounced and that in terms of amplitude they were hardly separable. Because of this we chose convolutional layers as they treat the complete vector at once and could obtain they aforementioned features better than a recurrent network.

Using a layer structure (Figure X) similar to the one built to predict the bounding box but this time using 1D Convolution we obtained our best results at predicting the phrase said in a video, a 35% accuracy. The activation function for the hidden layers was RELU as it showed a faster convergence than the rest and the Adam (Adaptive Moment Estimation) optimizer homogenized the reduction of the loss in comparison to other optimizers that didn't stop the loss from increasing and decreasing greatly in the validation set.

6.3 Conclusions & Future Work