Machine Learning Engineer Nanodegree – Facial Keypoints Recognition

Capstone Proposal

Nicholas Low
December 31st, 2050

Proposal

Domain Background

Facial keypoints detection is a field of study that has been relevant for many years due to its potential in creating a non-invasive method of identifying points on a face that can be used to determine any number of details about a person. However, facial features can differ greatly on individuals due to many variations of conditions in gathering images of individuals; therefore, determining accurate information from these features can prove to be difficult.

A well-known beginning to facial recognition systems is due to Tuevo Kohonen, a Finnish academic, who explained that a neural network could perform facial recognition only on aligned and normalized face images utilizing eigenvectors eventually becoming known as eigenfaces seen in Figure 1 [1]. These eigenfaces are representations of what the average face may look like based on a large set of data and allow a computer to determine what parts of the face generally look like. In modern facial keypoints detection, eigenfaces still exist as a primary method of identifying parts of the face although the science has become much more advanced.

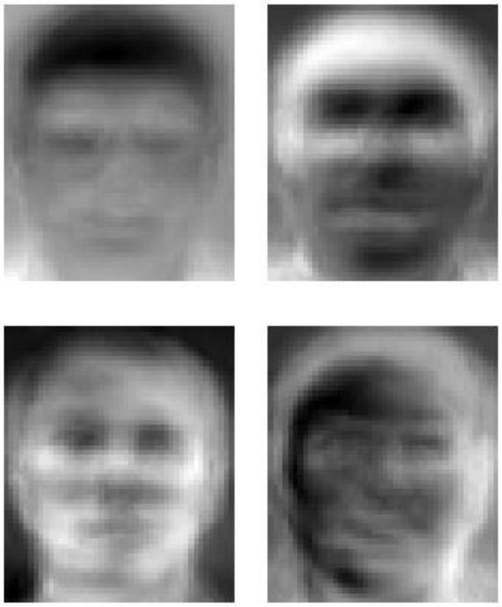

Figure 1: Example of Eigenfaces

By improving the method of facial keypoints detection systems, a number of solutions can occur. People may utilize these systems for lie detectors, medical diagnosis, biometrics, etc. On a bigger scale, every person may eventually have their own Sherlock Holmes, without the personality accompanying it (At least in the show "Sherlock"), to deduce the truth behind the many clues that exist within one's face.

Problem Statement

The problem to be solved is to predict keypoint positions/locations on face images. The goal is to predict the areas and parts where the mouth, eyes, ears, and nose are for all images with high accuracy. By determining the positions of these keypoints, a machine can gather information about the face. This will be a regression supervised learning problem as the data set includes labeled training data that uses continuous values in the form of pixels to learn from. A potential way to solve the facial keypoints detection problem is to utilize neural networks. The accuracy of the solution will be measured utilizing root mean squared error to determine the average of the squares of errors between a set of predicted outcomes and the actual outcomes.

Datasets and Inputs

The details of the dataset were taken from Kaggle's Facial Keypoints Detection competition [2]. The 15 keypoints being considered that represent elements in the face where left and right refer to the point of view of the subject are:

left_eye_center, right_eye_center, left_eye_inner_corner, left_eye_outer_corner, right_eye_inner_corner, right_eye_outer_corner, left_eyebrow_inner_end, left_eyebrow_outer_end, right_eyebrow_inner_end, right_eyebrow_outer_end, nose_tip, mouth_left_corner, mouth_right_corner, mouth_center_top_lip, mouth_center_bottom_lip.

Each data point for these elements is specified by an (x,y) real-valued pair in the space of pixel indices. Data points that are missing are left blank. The input image is displayed in the last field of the datasets consisting of a list of pixels (ordered by row), as integers between (0,255). There are 7049 images in the training set and 1783 images in the testing set and each of these images are 96x96 pixels.

Solution Statement

The goal of this project is to process all the training data to map major pixel coordinate areas to where the keypoints may be located which can possibly be determined by trying to use a neural network.

Benchmark Model

The benchmark model to be used to compare results will be a set of linear regression models for each type of keypoint to determine a floor in which a possible model will attempt to improve on. A linear regression is plausible as there should be an approximate relationship between keypoints in different images; therefore, someone's mouth won't be moving to a completely random location in different images. This benchmark will also be measured using room mean squared error to allow for a proper comparison to the solution.

Evaluation Metrics

The evaluation metric that will be used to quantify performance of both models is going to be the root mean squared error. The root mean square error is relevant for this problem because it works very well for a model whose main purpose is to predict. The root mean square error is the average distance of a data point from the fitted line which will help determine the generalization of the models to new data.

There will be 15 errors displayed for each model representing the spread in the values of the output compared to the model. The errors will be found by determining each of the predicted keypoints' deviation from their actual values, squaring these values, averaging the squared values, and then taking the square root. Mathematically, each of the errors will be determine by the equation:

$$RMSErrors = \sqrt{\frac{\sum_{i=1}^{n}(\hat{y}_i - y_i)^2}{n}}$$

Figure 2: Root Mean Square Error Equation

Project Design

Considering a training and test set have been provided through the Kaggle competition, a lot of it has probably been preprocessed. However, possible attempts at reducing the size of the data set will be to test for outliers by determining quartiles and filter unnecessary pixels in the image. After preprocessing, a theoretical workflow to approach a solution for the problem will be to start by applying the basic neural network (Fig. 3) and basic linear regression (Fig. 4) to the training data set and see the reliability of the predicted outcomes by testing them against the test set.

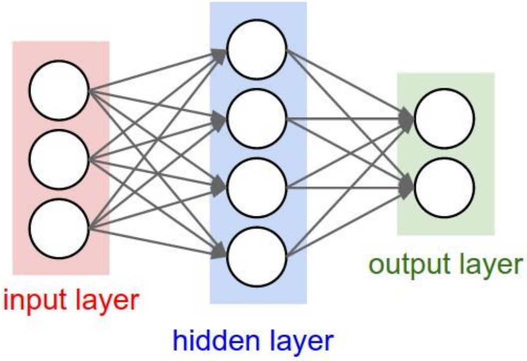

Figure 3: Neural Network Example

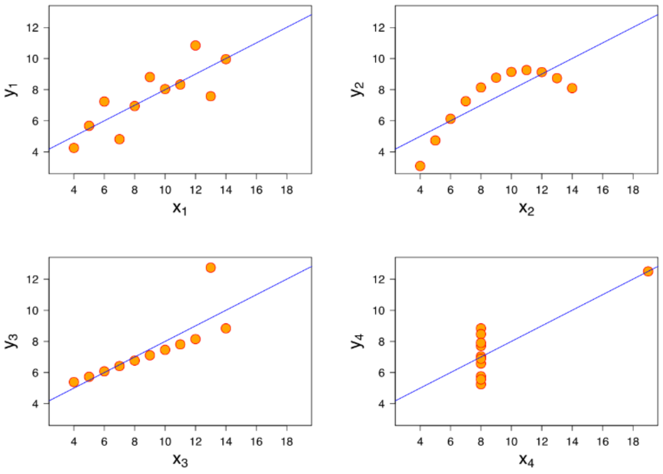

Figure 4: Linear Regression Examples

These error values will come out in sets of 15 root mean squared errors, determined by the equation in Figure 2, for each keypoint that the model is predicting a location for. The linear regression will be the benchmark model to compare the neural network to.

The next approach after determining errors is to look at the provided data sets and determine whether the split between training and test sets is optimal for utilizing neural networks. A strategy to approach this problem is to utilize k-fold and Grid Search Cross Validation to test out different sizes of training/test sets and parameters. K-fold cross validation will also help to ensure that the model generalizes to new data. These methods will also help to see if there are ways to improve on model.

Work Cited

(approx. 1 page)

[1] T. Choudhury, “History of face recognition,” 2000. [Online]. Available: http://vismod.media.mit.edu/tech-reports/TR-516/node7.html. Accessed: Jan. 10, 2017.

[2] Kaggle, “Data – Facial Keypoints Detection,” 2017. [Online]. Available: https://www.kaggle.com/c/facial-keypoints-detection/data. Accessed: Jan 11, 2017.