Face recognition

Final Project

Machine Learning, Winter 2009

David Volquartz Lebech

March 20, 2009

Contents

1 Introduction

Artificial Neural Networks (ANNs) have shown to offer good performance for analysis of complex real world data. This includes applications such as speech recognition, hand writing and *face recognition*. This last example is the topic of this project.

In [1, section 4.7], a practical implementation of extracting facial features from image data (e.g. the direction in which a person is looking) is discussed and demonstrated. The source code for this is available online as well as the image data used for the neural network. This data is very well organized and thus a good starting point for further development.

This paper describes my implementation of a simple neural network for face recognition using the above mentioned image data. I have looked into different ways of training the data to recognize features that are recorded in the images such as mood (happy/sad), face posture (left/right/straight) and whether the person is wearing sunglasses or not. My implementation is built from the bottom and I have purposely not looked at the C implementation that is provided.

2 Task analysis

The target learning task for the system can be stated like this:

> Given a set of training images and an appropriate training function, the system should be able to correctly classify features of images that have not been seen before by the system.

Note that this specification does not actually mention the word *neural network*. It simply specifies the overall requirement from a user perspective. To further elaborate on this specification, I will describe the learning problem for the system [1, section 1]:

Task T: Face recognition of humans with different postures, moods and sunglasses preference.

Performance measure P: Percent of images correctly classified.

Training experience E: A set of training images from 20 different people.

Target function V: Determine if a person is wearing sunglasses.

As mentioned in the introduction, it is not only possible to classify whether a person is wearing sunglasses or not but this will be a good starting point because it is essentially a boolean target value. To create the target function V, I will train an *artificial neural network*.

3 Design

In the following sections, I will explain the design of my face recognition system. It should be noted that my fundamental neural network design is very similar to the design discussed in [1, section 4.7]. However, there is some very important differences, that I will discuss later in the paper.

3.1 Network structure

The structure of my neural network is outlined below and the choices will be further discussed in the text below:

- The neural network is a layered, feedforward network with one hidden layer and one output layer.
- The input layer has a number of units each corresponding to 1 pixel of the input images.
- The input images have a resolution of at least 30 X 32 pixels. There is thus at least 960 inputs.
- The input values range form 0 to 255 (black to white). These values are normalized to range between 0.0 and 1.0 to fit the output of the hidden units and output units.
- The hidden layer has 3 hidden units.
- The output layer has as many units as the number of possibilities of the target concept. For determining whether a person wears sunglasses or not, there will thus be 2 output units.
- All units in the hidden layer and output layer are sigmoid units.

Choosing one hidden layer is an obvious choice given that more hidden layers increase the complexity of the network, probably without adding more accuracy to the network. The discussion in [1, section 4.6.2] also hints this. However, I will not further investigate the claim and the one hidden layer will be a fixed design choice.

Having only three hidden units seems at first to be very few compared to the very big amount of input units. As we saw in one of the homework assignments during the course, the number of units determine the number of possible output values. The 3-unit hidden layer is e.g. able to represent $2^3 = 8$ different values if we interpret each unit being able to represent only true or false (or 1 and 0). For this project, the target output is a maximum of 4 values for mood and direction of face so I could have chosen only 2 hidden units. I will investigate the difference in my later discussion.

A note about the output units: Since I am using sigmoid units, it is difficult to output a 0 or 1. Instead, I define the value 0.9 to correspond to a 1 (or true) and the value 0.1 to correspond to 0 (or false). Since each output unit is one possible value of the target concept, a successful output is (in the sunglass example) a vector like $(1, 0)$ and not $(1, 1)$ or $(0, 0)$. This representation of outputs is called *1-of-n* output encoding [1, page 114].

One input unit for each pixel might seem to be overkill. One could argue that mean or median values over regions of the images will both be sufficient enough and drastically reduce the number of input units. However, this has in principle already been done when resizing the images from the original images to the very small 30 X 32 pixel size (although probably by a more complex procedure than just taking the median or mean). In a sense, each pixel thus represents a region of the original image.

3.2 The learning algorithm

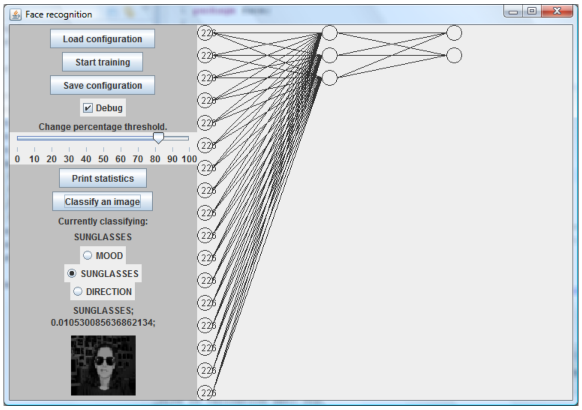
For training the neural network, I will use the BACKPROPAGATION algorithm, exactly as outlined in [1, table 4.2] with the exception that I add momentum to each weight update to possibly speed up convergence [1, section 4.5.2.1].

An important design choice that differs from the design seen in [1, section 4.7] is that I have chosen to use stochastic gradient descent instead of standard gradient descent. There are two reasons for doing this:

1. Training time decreases because the weight updates for each unit is based on the sum over all training examples.

2. Stochastic gradient descent can possibly avoid falling into local minima. [1, page 94]

Determining when to stop running the algorithm is another important choice and a matter that is dealt with in detail in [1, section 4.6.5]. Based on the image data, an evaluation set is randomly constructed that is separate from the training set. The amount of correctly classified instances of this



☑ Debug

Currently classifying:
SUNGLASSES

○ MOOD
● SUNGLASSES
○ DIRECTION

SUNGLASSES;
0.010530085636862134;

Figure 1: Screenshot from the program showing a (correct) classification of a person with sunglasses.

evaluation set will be used as the termination condition. This removes some of the possibilities of overfitting the training data but does not prevent it entirely.

3.3 Implementation

Rather than having a separate section, I will just briefly explain the actual implementation of my neural network. As operating platform, I have used Java SE version 6. I have tried to follow the good design rules of object oriented programming by, for example, having strong separation of concepts. I have constructed a basic framework for neural networks, including classes for e.g. Sigmoid units and layers, and I have then extended this framework to the more specific case of a two layered network with support for the Backpropagation algorithm.

Now I do want to sound pretentious so I should hurry and emphasize that the framework is not so general and comprehensive that it can be readily transferred to any other neural network setting. However, with a few changes it *can* work as the basis of later experiments that does not necessarily involve facial feature recognition.

Finally, for this specific project, I have designed a graphical user interface that connects to the network, mainly functioning as an observer and control center for ease of use (see figure 1). This part is not fail-safe in any way. The important part is the Backpropagation algorithm and what goes on beneath the surface. I will now turn to this aspect in the next section.

4 Results and discussion

To test the performance of my implementation, I have trained the neural network for classifying three different facial characteristics of the persons in the input data:

- Whether or not the person is wearing sunglasses
- Which direction the person is facing.
- Which mood the person expresses.

The classification has been carried out on both small pictures of size 30 X 32 and the same pictures with size 60 X 64. For all tests, I have used a learning rate of 0.1 and a momentum of 0.3. The Backpropagation algorithm was set to stop when reaching an accuracy over the validation set of 90%. There is 624 images in total and they are divided into two distinct sets of size 312 for training and validation, respectively. In all the following graphs, 1 iteration corresponds to 1 run through the entire training set of 312 images. For example, 10 iterations is thus 3120 weight updates.

4.1 Non-desired results and effects

As a first result, it should be noted, that if the validation set and the training set consist of the same data then high accuracy is naturally reached after very few iterations. I have not shown an example graph here but in fact, 80% accuracy or more can be reached within the first 5-10 iterations over all training examples. This approach is of course prone to overfitting as already discussed and I have not used this as the general method.

My first attempt with dividing the image data into a training and validation set was based on a simple half/half split down in the middle of the image data. This lead to the very strange but interesting classification seen in figure 2 where the correctly classified validation examples fluctuate between 50-60% and 0%. This effect is probably due to the fact that the image data consist of 20 persons where the training set and validation set thus each contained 10 different persons.

4.2 More pleasing results

When randomly choosing the validation set data and training data, I get better results. For the sunglasses classification, the desired 90% accuracy is reached for the small images (figure 3) while the larger images seem to get

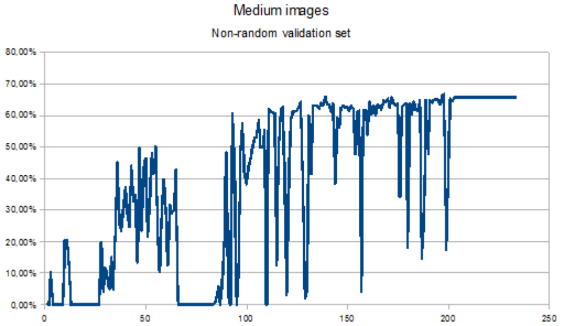


Figure 2: Strange classification behavior resulting from bad division of training and validation data.

stuck just below the 90% mark (figure 4). Subsequent individual classifications confirm the results. Not surprisingly, the initial classification correctness is already 50% because the target function only has 2 values.

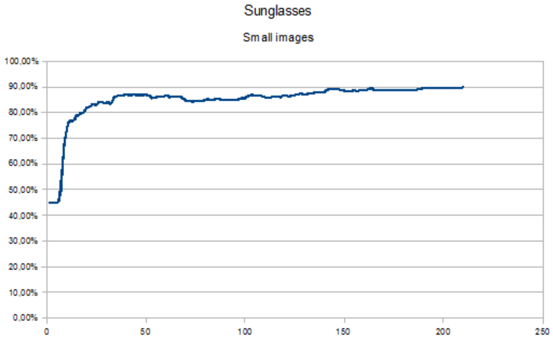
Perhaps more pleasing is the classification of direction which starts out with 0% accuracy for the first few iterations and then suddenly climbs very fast to get better accuracy. In this case, it is the small images that fail to get above 90% accuracy (figure 5) while the larger images converge successfully to the 90% goal (figure 6).

In all four of the above cases, it seems that once the accuracy starts increasing, it increases very fast. This is in touch with the examples shown in [1, figure 4.9] and could possibly also be ascribed to the momentum that helps “get the ball rolling” down the slope of the error surface [1, section 4.5.2.1].

4.3 Mood is difficult

It seems that interpreting a person’s mood is just as difficult on a computer as it can sometimes be in real life. While both direction and sunglasses show very good classification results, mood is a totally different story. For the small images, there is a steady increase in accuracy but it does not at any time go above 20% percent (figure 7). For the larger images, the results are the same and actually, some of the strange behavior seen in figure 2 repeat itself around iteration 300-350 where the accuracy suddenly drops to 0% (figure 8).

One of the explanations for this can be the very subtle ways in which the



6

Figure 3: The percentage of correct validations in the validation set for the target value sunglasses and 30 X 32 pixel images.

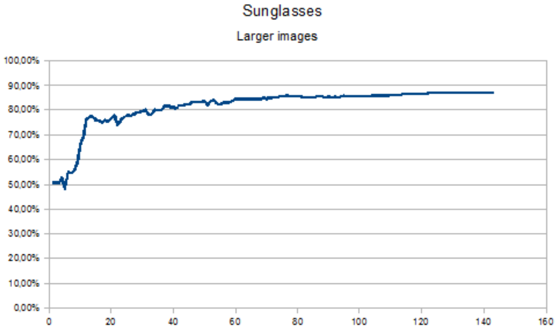


Figure 4: The percentage of correct validations in the validation set for the target value sunglasses and 60 X 64 pixel images.

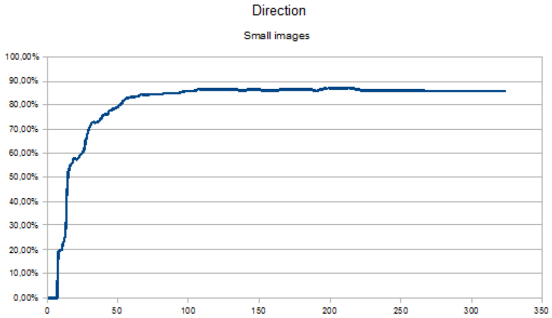


Figure 5: The percentage of correct validations in the validation set for the target value direction and 30 X 32 pixel images.

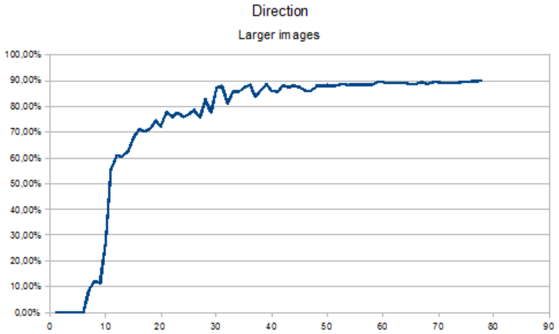


Figure 6: The percentage of correct validations in the validation set for the target value direction and 60 X 64 pixel images.

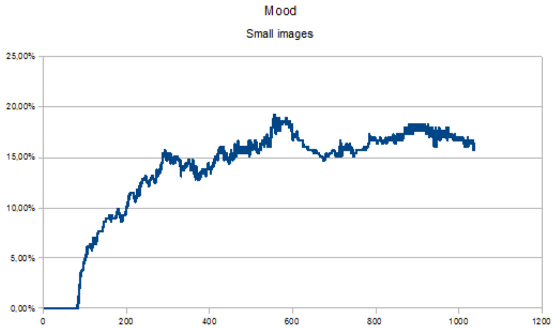


Figure 7: The percentage of correct validations in the validation set for the target value mood and 30 X 32 pixel images.

mood differs from image to image. Having a neutral face and a happy face look very similar and for all the images looking to the sides, it is very difficult to determine their facial expression in any way, while general contours like hair give away the direction the person is looking and the very marked black sunglasses help to determine this target function.

4.4 Changing algorithm parameters

In order to try and increase the accuracy of the mood classification, I tried adjusting the network structure to contain more hidden units to see if this representation would yield better results. First, I increased the number of hidden units to 4. This immediately gave a worse result than before which is seen in figure 9 where the accuracy is only 12-14%. Using only 2 hidden units however, produced a slightly better result of around 18-19%, as seen in figure 10. Since neither increasing or decreasing the number of hidden units helped produce better results, it seems that it can be concluded that the quality of the training data simply is not good enough for classifying mood.

When trying to classify sunglasses with 1, 2 or 4 hidden units (not shown here) I did not experience better or worse results. It thus seems that the 3 unit design choice is vindicated to the extent that it exactly covers the desired number of output values (2 or 4) but also leaves room for some extra degrees of freedom in the case where, e.g., we want to add another direction (down) to the possible directions a person can look.

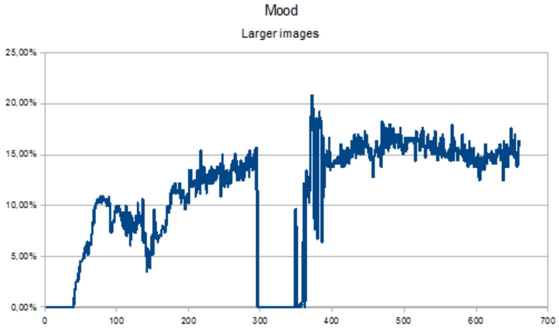


Figure 8: The percentage of correct validations in the validation set for the target value mood and 60 X 64 pixel images.

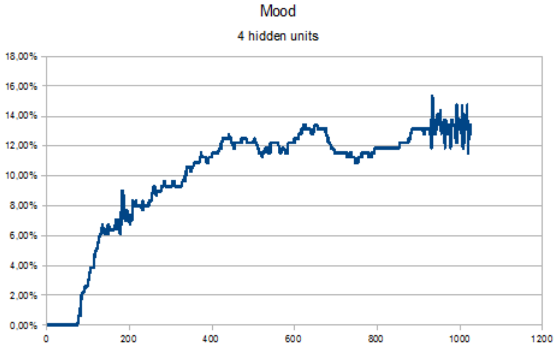


Figure 9: The percentage of correct validations in the validation set for the target value mood with 4 hidden units.

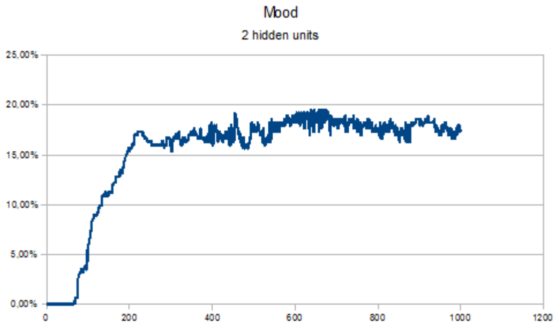


Figure 10: The percentage of correct validations in the validation set for the target value mood with 2 hidden units.

5 Conclusion

In this project, I have designed and implemented an artificial neural network for recognizing facial features in images. I have used a very basic version of the Backpropagation algorithm with stochastic gradient descent. With this algorithm I have successfully trained the network to classify whether or not a person is wearing sunglasses and in which direction the person is looking. In both cases, I received very satisfying accuracies of about 90% on independent validation data. Classifying the mood of a person turned out to be a more difficult task which I explain with the quality and low resolution of the image data.

There are several ways of extending the system. A simple improvement could be to carry out even more calibrations of the network to see which one is optimal. Another extension could be to implement different variations of the Backpropagation algorithm or even totally different learning algorithms. This would give a better overview of the limits and capabilities of the Backpropagation algorithm. For this paper, I would like to have included a survey of other approaches in the field which was indeed also my initial intention. In the end, however, I considered implementing the neural network and the learning process involved in this creation the most important task of the project which is the reason for my lack of references to other sources than the book and the implementation described there.

During this project, I have gained valuable insight into neural networks and how they work. I have also seen that having good training data is essential for learning specific target functions. One of the most important lessons learned for me though is that even a simple algorithm like Backpropagation actually produce decent results under certain circumstances. I am satisfied with the end result and look forward to further explore neural networks and machine learning.

References

[1] Tom Mitchell. *Machine Learning.* McGraw-Hill, 1st edition, 1997.