EMOJI RECOGNITION SOFTWARE

TRAINING NEURAL NETS WITH BACKPROPAGATION vs. GENETIC ALGORITHM

Timothy Henry Charles Tamm, Khalid Tawil, Fatma Akcay
Harvard College, CS 182 - Artificial Intelligence

INTRODUCTION The central idea for our project focuses on software that can recognize drawings of emojis and tries to predict what kind of emoticon was drawn. We focused on four main emotions/ faces: "happy", "sad", "laughing", and "heart eyes". An example of each is included in [Figure 1].
Our software uses elements of machine learning to learn what a face is "supposed" to look like. We use artificial neural networks that go through a lot of training images similar to the below. After that we use a different approach consisting of genetic algorithms to train neural nets. We compared both approaches in terms of time and accuracy to determine what is better at recognizing our emoji.

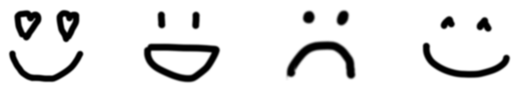

Figure 1 Example of input images for hearts, laugh, sad, and smile.

METHODS Our system depends on 3 main components:

1. Decision making system: Our project uses a Neural Net in order to make decisions. The Neural net has 25^2 inputs, 120 hidden neurons and 4 outputs. Each of the inputs corresponds to a pixel in the original picture and takes on a value of 1 for black or 0 for white. The outputs correspond to the four emojis the system is able to identify: Sad, happy, laughing, heart eyes.
2. Training system: We implemented 2 different algorithms for training our neural net: A genetic algorithm, which uses the weights of a neural net as the chromosomes and a Backpropagation algorithm.
3. Image conversion system: We used the Python image library to convert input and training images to 25x25 pixels black and white images and then convert that data into an array of 0s and 1s that we can use as inputs for our system.

For training data we each drew 15 of each emoji, resulting in 45 pictures of each emoticon. For testing we use 8 pictures (2 of each emoji) that are not included in the training data.
The program is interacted with through command line arguments and a terminal based user interface. The UI currently allows the user to save a trained net, calculate the squared average error of the net, run a specific image through the net, run our testing data through the net and continue training the net.

RESULTS On our test computer (Thinkpad T440s, i5 processor, 12gb RAM), it takes the backpropagation algorithm about 9 seconds to go through a 100 iterations. It takes about 70 seconds to go through one generation using the genetic algorithm.

- **Backpropagation**: The algorithm was able to produce passable results after 300 iterations, good results after 500 iterations and excellent ones after about 1000 iterations. Even though the squared average error of the net becomes very small after it's been trained well, it will still occasionally output a completely wrong answer because our training data is quite limited.
- **Genetic Algorithm**: The genetic algorithm seems to flatline after about 50 generations and gets stuck at an average squared error of 1.5. At such an error rate, the neural net is still completely off with its predictions.

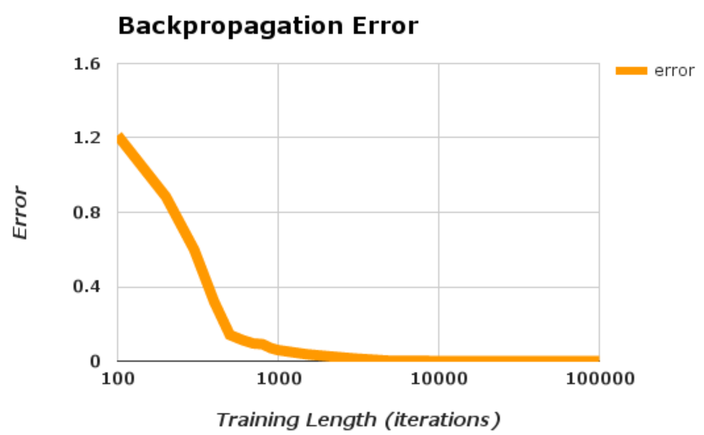

Figure 2 Error vs. Training length for Backpropagation

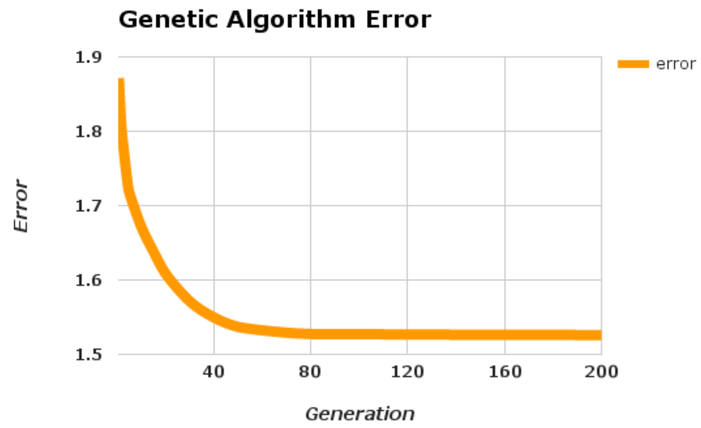

Figure 3 Error vs. Generation for Genetic Algorithm

CONCLUSIONS Overall, training the neural nets through the genetic algorithm was exponentially worse than backpropagation. Since each neural net has 75480 weights that need to be calibrated, taking a pseudo-random approach such as a genetic algorithm leads to bad results, because the runtime of the algorithm is too long for us to be able to go through a sufficient number of generations for the net to be trained well.

FUTURE WORK Our initial inspiration for this project was to be able to recognize emotions from human face inputs. This is still a possibility using our program. For the scope of this project, we decided to limit the input to a simple emoji image so the program could easily learn the emoticons. For human-face inputs, there would be greater variation and we would have to train it for longer. However it is feasible and definitely a possibility for future work. Such a program could be useful for helping people on the autism spectrum that have a difficult time recognizing a person's displayed emotions. Users would be able to use the program to recognize what emotions look like.

References

1. http://www.scientific.net/AMM.380-384.4057
2. https://escholarship.org/uc/item/65g4f175
3. http://link.springer.com.ezp-prod1.hul.harvard.edu/book/10.1007/978-3-642-40728-4
4. http://www.cell.com/current-biology/abstract/S0960-9822(14)01039-2?_returnURL=http%3A%2F%2Flinkinghub.elsevier.com%2Fretrieve%2Fpii%2FS0960982214010392%3Fshowall%3Dtrue

Acknowledgements We would like to thank the entire teaching staff of CS182 for their feedback and help in developing our project. Special thanks to Ben Green for advising this project.