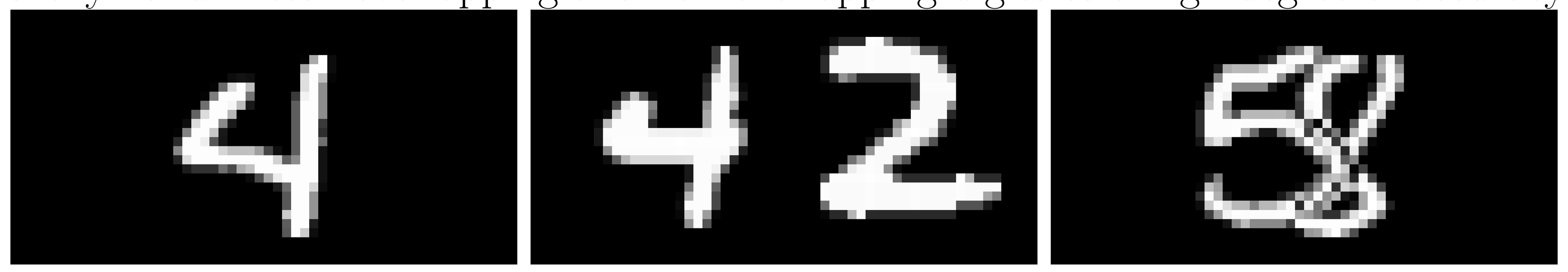
Recognizing Overlapping Handwritten Digits Using Neural Networks

Professor Mark Stankus*, Alexa White*, Cameron Fredrickson*, Grant Bernosky*, and Tim Royston*

Frost Research 2018

Our Goal

We wanted to use a machine learning concept called a neural network to learn how to identify handwritten overlapping and non overlapping digits to a high degree of accuracy.

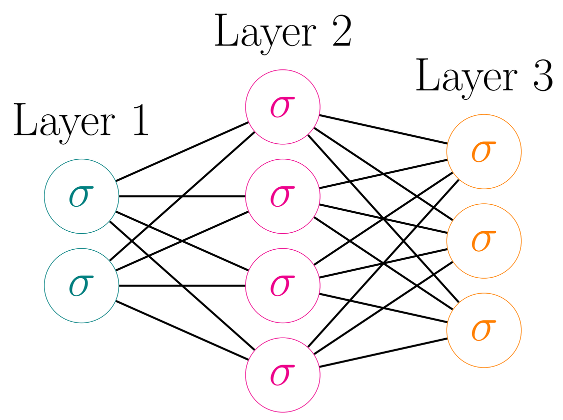


What is a neural network made of?

A neuron is a real-valued function which takes as input a vector $\mathbf{x}$, a vector of weights $\mathbf{w}$ and a bias b which outputs $\sigma(\mathbf{x} \bullet \mathbf{w} + b)$. The function σ is called an activation function whose output is usually within $[-1, 1]$ or $[0, 1]$. A commonly used activation function with a range of $[0, 1]$ is the sigmoid function:

$$\sigma(z) = \frac{1}{1 + e^{-z}}.$$

A densely-connected layer of neurons is a function which encapsulates a finite number of neurons, where each neuron in the layer is connected to every neuron in the next layer.



A layer is a vector-valued function which takes as input a vector $\mathbf{x}$, a set of weight vectors $\mathbf{w_1}, \ldots, \mathbf{w_m}$ and a set of biases $b_1, \ldots, b_m$ which outputs

$$\begin{bmatrix} \sigma(\mathbf{x} \bullet \mathbf{w_1} + b_1) \\ \vdots \\ \sigma(\mathbf{x} \bullet \mathbf{w_m} + b_m) \end{bmatrix}$$

Cost Function for Classifying Single Digits

- A neural network used to classify single digit images has an output vector $\hat{\mathbf{y}}_0$ with 10 entries. Each entry is a value between 0 and 1 and can be interpreted as a probability that the given image is of a particular digit. The 1st entry corresponds to a '0', the 2nd entry to a '1' and the 10th entry to a '9'.
- For example, we might have $\hat{\mathbf{y}}_0 = [0 \ \ 0 \ \ 0 \ \ 0.2 \ \ 0.5 \ \ 0.1 \ \ 0.1 \ \ 0 \ \ 0 \ \ 0]^T$.
- Suppose the weights and biases of our neural network are $\mathbf{w}_0$ and $\mathbf{b}_0$ and $\mathbf{x}_0$ is an image of a '4'. The label of $\mathbf{x}_0$ would be $\mathbf{y}_0 = [0 \ \ 0 \ \ 0 \ \ 0 \ \ 1 \ \ 0 \ \ 0 \ \ 0 \ \ 0 \ \ 0]^T$.
- The cost function C describes how close our prediction is to the label of the image and is defined by $C_{\mathbf{x}_0}(\mathbf{w}_0, \mathbf{b}_0) = \|\mathbf{y}_0 - \hat{\mathbf{y}}_0\|$.

How does a network learn?

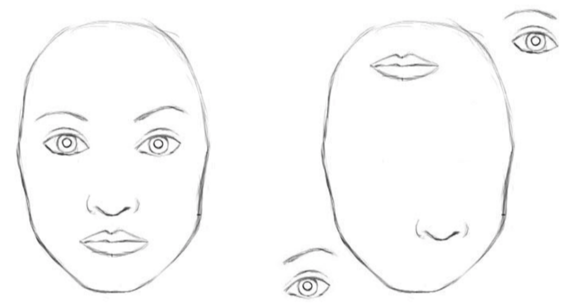
- For a fixed $\mathbf{x}$, the graph of the function $C_{\mathbf{x}}$ is a surface
- To train a neural network we want to go "downhill" along that surface by following the negative of the gradient of $C_{\mathbf{x}}$
- We want to find $\mathbf{w}_0$ and $\mathbf{b}_0$ so that $C_{\mathbf{x}}(\mathbf{w}_0, \mathbf{b}_0)$ is small for each training vector $\mathbf{x}_0$
- The neural network "has learned" if $C_{\mathbf{x}}(\mathbf{w}_0, \mathbf{b}_0)$ is also small for each test vector $\mathbf{x}$

Training a Neural Network

1. Convert black and white images to matrices of 0's and 1's, then flatten them into vectors
2. Separate theses flattened vectors into training, validataion, and testing data
3. Adjust the weights and biases as the training data is fed to the the neural network
4. Validate those adjustments by testing the network on the validation data
5. Train on several epochs, completing steps 3 and 4 is refered to as "1 epoch"
6. Feed in the test data to see how well the training generalizes to unseen data.
7. Save the neural network (the adjusted weights and biases)
8. Now we can use the trained network to classify future data

Image Recognition Solutions

Before 2017, the most common type of neural network used in image recognition was the convolutional neural network (cnn). A cnn classifies images by determining the existence of **certain features** in an image. It does not consider the relative positions and orientations of these features. A cnn might recognize a nose, an eye, or a mouth, but classify both of the following as faces.



[2]

A capsule network [3] classifies an image as a face if and only if positions and orientations of the the features relative to the location of the face are correct. In the picture below on the right, recognizing the mouth would lead the network to predict that the rest of the face would resemble the highest of the three faces shown and recognizing the eye would lead the network to predict that the rest of the face would resemble the lowest of the three faces shown. Since the face predictions do not agree, the image is not classified as a face.

[1]

Our Solution

Our Data:

- MNIST is a collection of single digit handwritten numbers.
- With Python code and MNIST we created a dataset of single digit and overlapping double digit images.

Our Neural Networks:

- We used TensorFlow an open source machine learning framework created by Google to code our neural networks.
- We created and trained a densely connected neural network called choice_net to classify each data sample in our randomly shuffled dataset as a single digit or double digit image
- The output of choice_net is fed to one of 2 capsule networks named single_caps and double_caps which were trained on single digit and double digit images respectively.

Results

All of our neural networks were trained on 2 epochs of 60,000 training samples and 5,000 validation samples. Our testing dataset contained 10,000 samples.

- choice_net achieved a test accuracy of 99.9+%
- The single_digit_caps achieved a test accuracy of 99.1987%
- The double_digit_caps achieved a test accuracy of 98.8381%

The code is publically available at `https://github.com/mstankus/Frost-Neural-Network-Research-2018`.

References

[1] Nick Bourdakos. *Understanding Capsule Networks - AI's Alluring New Architecture.* Feb. 2018. URL: `https://medium.freecodecamp.org/understanding-capsule-networks-ais-alluring-new-architecture-bdb228173ddc`.

[2] Max Pechyonkin. *Understanding Hinton's Capsule Networks. Part I: Intuition.* Nov. 2017. URL: `https://medium.com/ai%C2%B3-theory-practice-business/understanding-hintons-capsule-networks-part-i-intuition-b4b559d1159b`.

[3] Sara Sabour, Nicholas Frosst, and Geoffrey E Hinton. "Dynamic Routing Between Capsules". In: *Advances in Neural Information Processing Systems 30.* Ed. by I. Guyon et al. Curran Associates, Inc., 2017, pp. 3856–3866. URL: `http://papers.nips.cc/paper/6975-dynamic-routing-between-capsules.pdf`.

(This is a partial list containing the resources used to create the poster)

Acknowledgments

* Frost Research Fellows, funded by the Bill and Linda Frost Fund, Cal Poly, San Luis Obispo