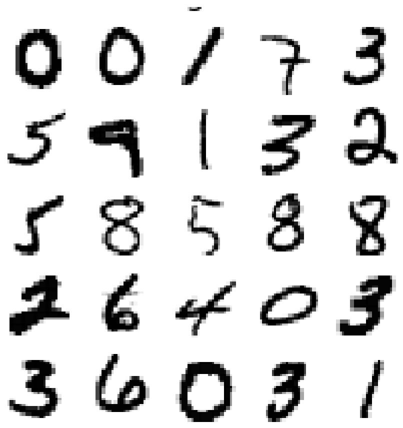
Statistical Machine Learning

Introduction to Neural Networks

Zipcode dataset: is it a "4" or "9"?



Zipcode (MNIST) dataset: have 16x16 pixel grayscale images of "4"s and "9"s

Goal: train a model to predict whether a 16x16 grayscale image is of a "4" or "9".

Logistic Regression Model

Additional Resources

- Not covered in the ISL textbook, so
- See chapter 11 (sections 3 and 4) in ESLII

Here's a quote from section 11.3 on page 392:

"There has been a great deal of *hype* surrounding neural networks, making them seem magical and mysterious. As we make clear in this section, they are just nonlinear statistical models…"

Also see "Hands on Machine Learning" via Safari Books on the library's website.

Origins of Network Networks

1957: Frank Rosenblatt invented the “perceptron” algorithm to mimic the behavior of biological neurons.

Neural Networks: the basics

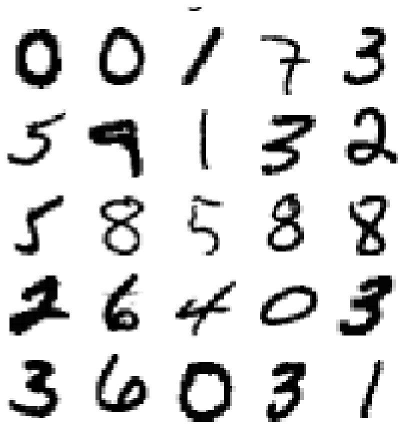
- Can be used to predict either a quantitative or a qualitative (k categories) response variable Y based on p predictor variables

- Works better than simpler methods when the p predictor variables have a complex relationship among themselves and with Y

- Can be computationally difficult to train
  - Google's TensorFlow can be used with R or Python

- Aside from a few cases, the resulting models are hard to interpret.

Zipcode dataset: is it a “4” or “9”?



Zipcode (MNIST) dataset: have 16x16 pixel grayscale images of “4”s and “9”s

Goal: train a model to predict whether a 16x16 grayscale image is of a “4” or “9”.

Network Diagram

Results: 91.7% accuracy

Using 16*16=256 inputs to classify a binary output with

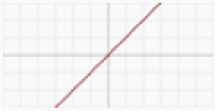
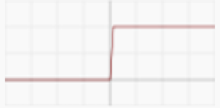
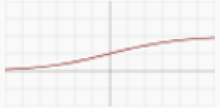
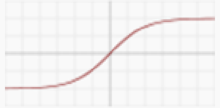
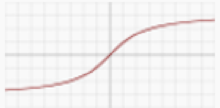
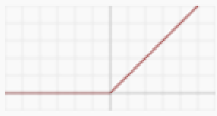
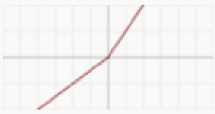
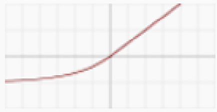
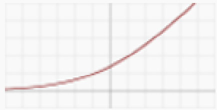
- 2 hidden layers of 10 and 20 neurons,
- Relu activation for the hidden layers,
- Logistic activation for the output layers,
- Training to maximize the binomial probability

Choices for a “Forward Feed” Neural Network

- Activation function for each layer
- Activation function for the output
- Number of hidden layers
- Number of “neurons” in a hidden layer
- Loss function to minimization (i.e. mean squared training error

Common Activation Functions

(from "Towards Data Science" website)

Name	Plot	Equation	Derivative
Identity		$f(x) = x$	$f'(x) = 1$
Binary step		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x \neq 0 \\ ? & \text{for } x = 0 \end{cases}$
Logistic (a.k.a Soft step)		$f(x) = \frac{1}{1+e^{-x}}$	$f'(x) = f(x)(1 - f(x))$
TanH		$f(x) = \tanh(x) = \frac{2}{1+e^{-2x}} - 1$	$f'(x) = 1 - f(x)^2$
ArcTan		$f(x) = \tan^{-1}(x)$	$f'(x) = \frac{1}{x^2+1}$
Rectified Linear Unit (ReLU)		$f(x) = \begin{cases} 0 & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} 0 & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Parameteric Rectified Linear Unit (PReLU)[2]		$f(x) = \begin{cases} \alpha x & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
Exponential Linear Unit (ELU)[3]		$f(x) = \begin{cases} \alpha(e^x - 1) & \text{for } x < 0 \\ x & \text{for } x \geq 0 \end{cases}$	$f'(x) = \begin{cases} f(x) + \alpha & \text{for } x < 0 \\ 1 & \text{for } x \geq 0 \end{cases}$
SoftPlus		$f(x) = \log_e(1 + e^x)$	$f'(x) = \frac{1}{1+e^{-x}}$

Training a neural network

Generally a numerical algorithm must be used to find the model "weights" corresponding to the minimum value training error.

Variations of gradient descent are a common choice.

Common Types of Neural Networks

- Forward Feed

Additional choice of which neurons are connected:

- Convolutional
- Recurrent

Unsupervised learning:

- Autoencoder