Data Analytics Project

Handwritten digits recognition

Recognize handwritten digits and turning those images into numeric data.

Name: Yue Cheng
Chenggang Lou
Saira Rotondo
Sidi Wu

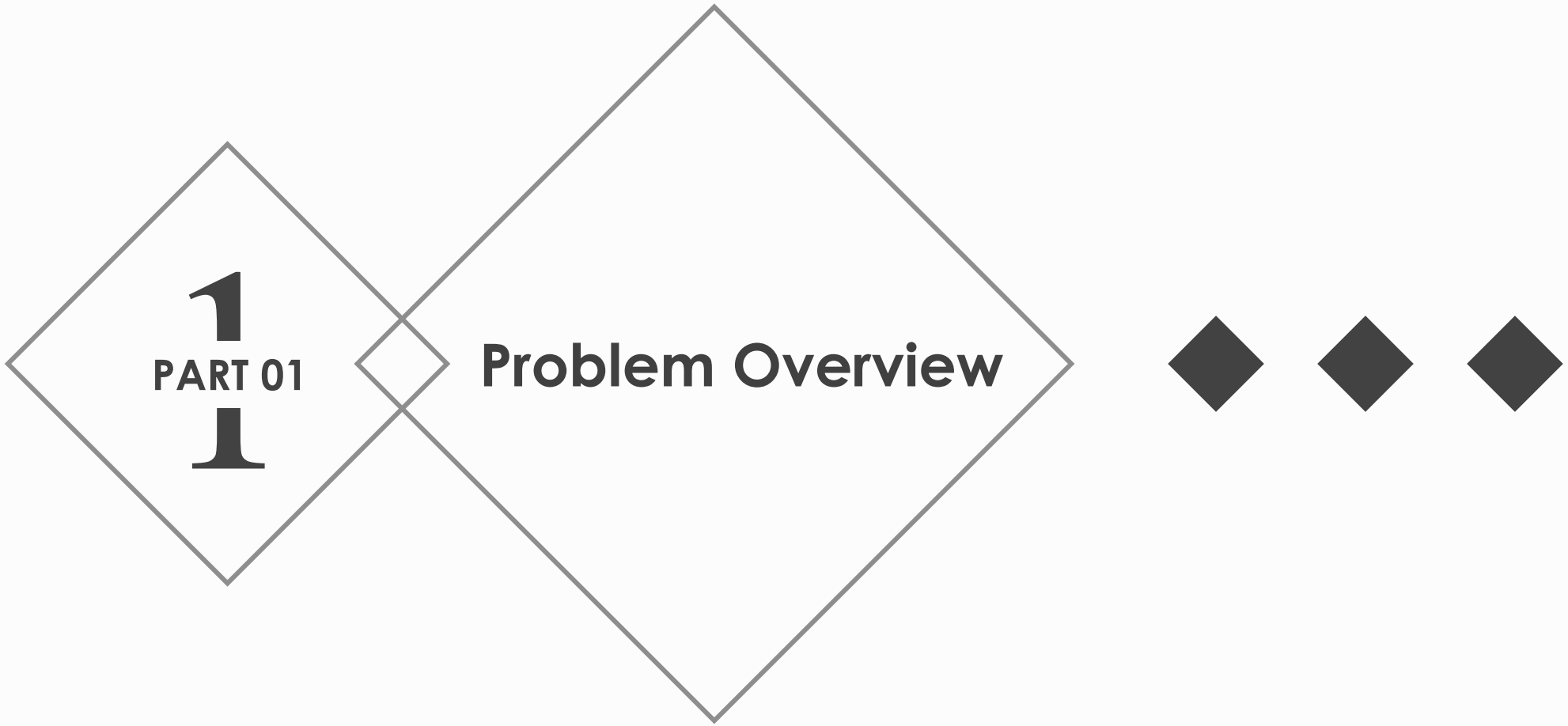
Handwritten digits recognition

1 Question Overview

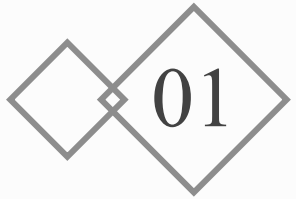
2 Data set

3 Model Selection

4 Prediction



Problem Overview



INTRODUCTION

problem overview

HYPOTHESIS: Devise model to recognize handwritten digits.

01 OVERVIEW

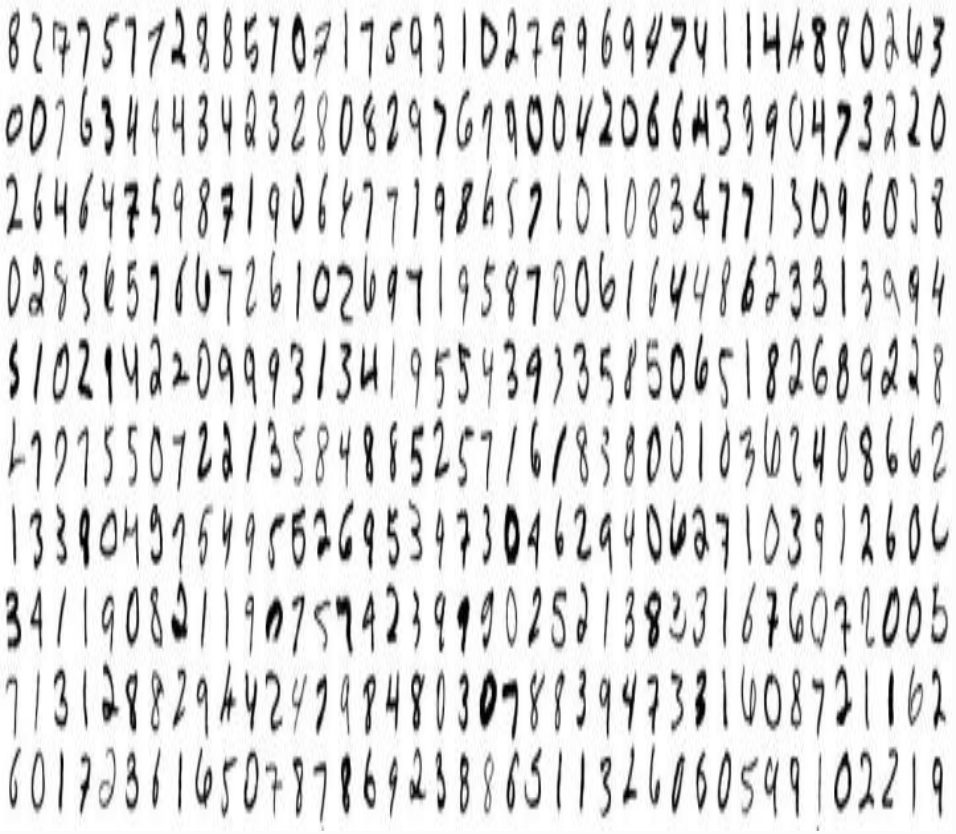
problem overview

Predictors (input variables): numeric data frame, gray scale, 60, 000 observations

Responses (output variables): The label (numerical value) digit 0-9

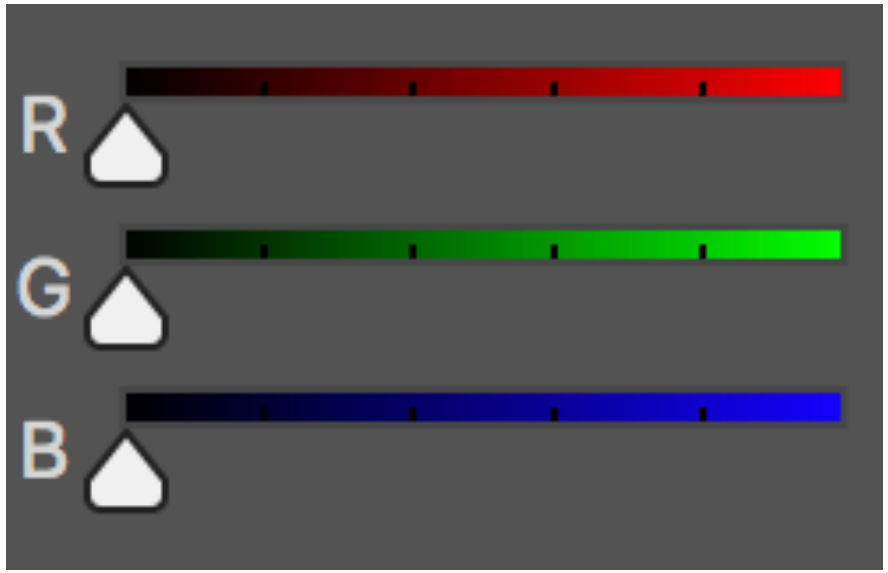
How was the data collected? We used a large data set collected from the MNIST database of handwritten digits

DATA SET URL: http://yann.lecum.com/exdb/mnist



01 Overview

problem overview



➢For simplicity instead of using a RGB color scale we used a gray scale

➢parameters for gray scale are variables between….

0=white, 255=black

Each pixel in the image data was represented numerically.

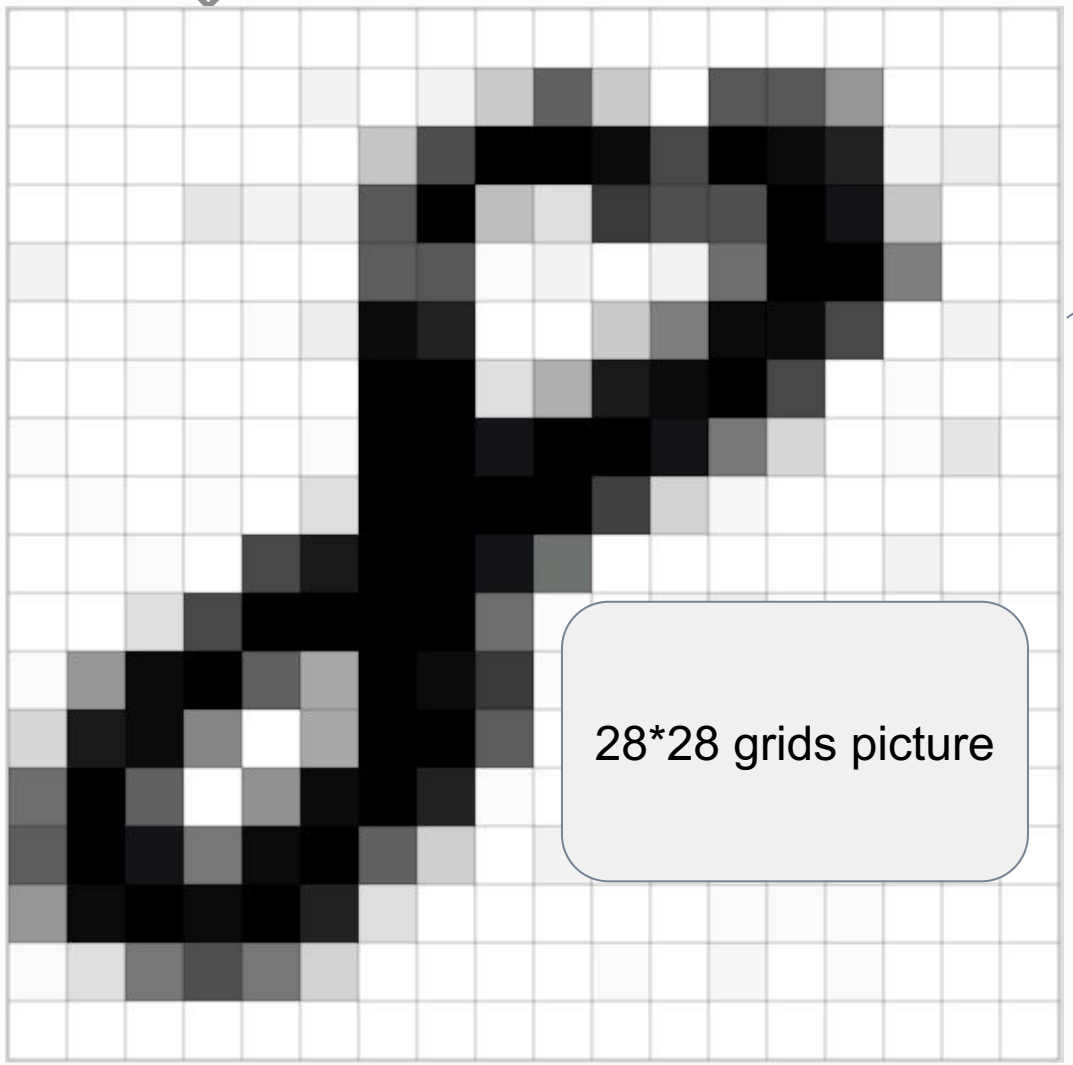
PART 02

Data Set

02

Input Variables

DATA SELECTION



28*28 grids picture

60,000 training images
10,000 testing images

Unroll into 784 dimensional vector

0	255
235	125

……

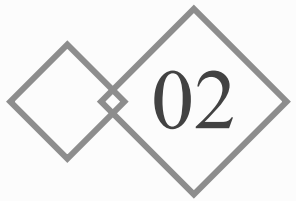
100	210
250	243

……

255	169
100	200

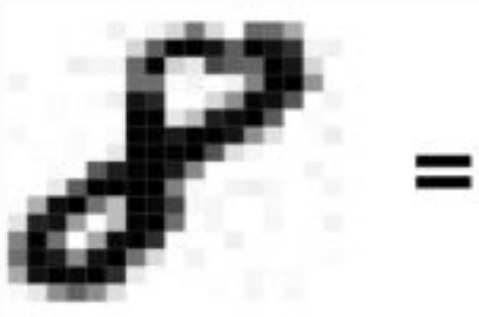
……

212	222
199	211



02 Normalized Input Variables

DATA SELECTION



=

[0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 12, 0, 11, 39, 137, 37, 0, 152, 147, 84, 0, 0, 0, 0,
0, 1, 0, 0, 0, 41, 160, 250, 255, 235, 162, 255, 238, 206, 11, 13, 0, 0, 0, 0, 16, 9, 9, 150, 251, 45, 21, 184, 159, 154, 2
55, 233, 40, 0, 0, 10, 0, 0, 0, 0, 0, 145, 146, 3, 10, 0, 11, 124, 253, 255, 107, 0, 0, 0, 0, 3, 0, 4, 15, 236, 216, 0, 0,
38, 109, 247, 240, 169, 0, 11, 0, 1, 0, 2, 0, 0, 0, 253, 253, 23, 62, 224, 241, 255, 164, 0, 5, 0, 0, 6, 0, 0, 4, 0, 3, 252
, 250, 228, 255, 255, 234, 112, 28, 0, 2, 17, 0, 0, 2, 1, 4, 0, 21, 255, 253, 251, 255, 172, 31, 8, 0, 1, 0, 0, 0, 0, 0, 4,
0, 163, 225, 251, 255, 229, 120, 0, 0, 0, 0, 0, 11, 0, 0, 0, 0, 21, 162, 255, 255, 254, 255, 126, 6, 0, 10, 14, 6, 0, 0, 9
, 0, 3, 79, 242, 255, 141, 66, 255, 245, 189, 7, 8, 0, 0, 5, 0, 0, 0, 0, 26, 221, 237, 98, 0, 67, 251, 255, 144, 0, 8, 0, 0
, 7, 0, 0, 11, 0, 125, 255, 141, 0, 87, 244, 255, 208, 3, 0, 0, 13, 0, 1, 0, 1, 0, 0, 145, 248, 228, 116, 235, 255, 141, 34
, 0, 11, 0, 1, 0, 0, 0, 1, 3, 0, 85, 237, 253, 246, 255, 210, 21, 1, 0, 1, 0, 0, 6, 2, 4, 0, 0, 0, 6, 23, 112, 157, 114, 32
, 0, 0, 0, 0, 2, 0, 8, 0, 7, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0]

784 dimensional vectors whose elements are from 0~255
Normalize the data by divided by 255
Normalized Input variables: between 0 and 1

02

Output Variables

DATA SELECTION

60,000 dimensional vectors whose elements are digit number 0~9

PART 03

Model Selection

03 Softmax Regression

Model Selection

Generalizes logistic regression

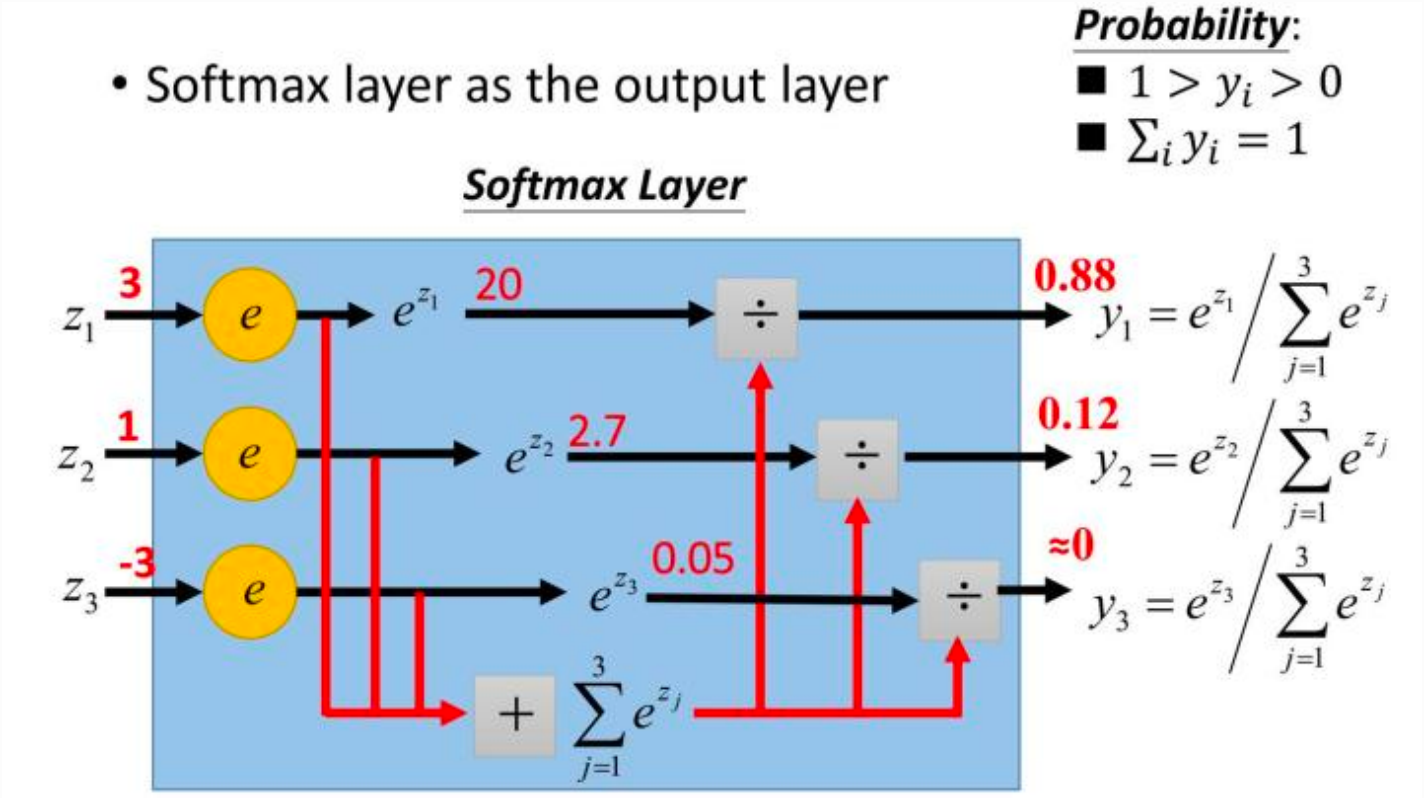
Supervised learning algorithm

Multi-class Classification

$$P(y=j|X)=\frac{e^{\theta_j^T X}}{\sum_{i=0}^{K} e^{\theta_i^T X}}$$

Loss(Cross-Entropy)

$$L_i=-log(\frac{e^{f_{y_i}}}{\sum_j e^j})$$



03

Logistic Regression VS Softmax Regression

Model Selection

Label

$$y^{(i)} \in \{0, 1\}$$

Hypothesis

$$h_\theta(x) = \frac{1}{1 + \exp(-\theta^T x)}$$

Cost function

$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m} y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log(1 - h_\theta(x^{(i)}))\right]$$

Label

$$y^{(i)} \in \{1, 2, \ldots, k\}$$

Hypothesis:

$$h_\theta(x^{(i)}) = \begin{bmatrix} p(y^{(i)} = 1 | x^{(i)}; \theta) \\ p(y^{(i)} = 2 | x^{(i)}; \theta) \\ \vdots \\ p(y^{(i)} = k | x^{(i)}; \theta) \end{bmatrix} = \frac{1}{\sum_{j=1}^{k} e^{\theta_j^T x^{(i)}}} \begin{bmatrix} e^{\theta_1^T x^{(i)}} \\ e^{\theta_2^T x^{(i)}} \\ \vdots \\ e^{\theta_k^T x^{(i)}} \end{bmatrix}$$

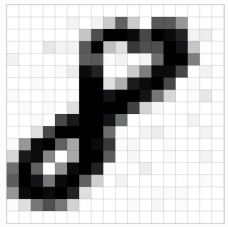
Cost function

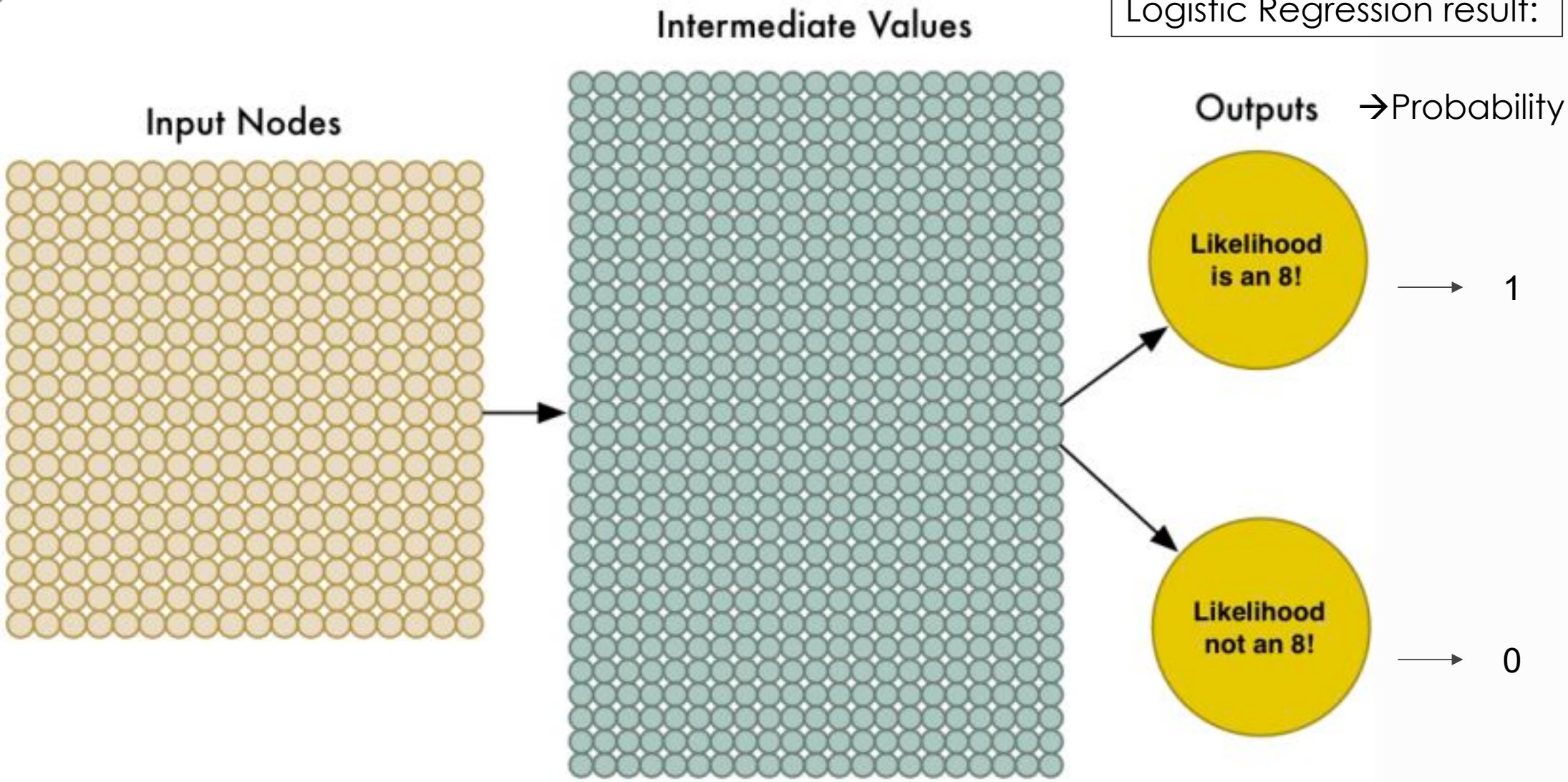
$$J(\theta) = -\frac{1}{m}\left[\sum_{i=1}^{m}\sum_{j=1}^{k} 1\left\{y^{(i)} = j\right\} \log \frac{e^{\theta_j^T x^{(i)}}}{\sum_{l=1}^{k} e^{\theta_l^T x^{(i)}}}\right]$$

03 Comparison: Logistic Regression

Model selection

Logistic Regression result:

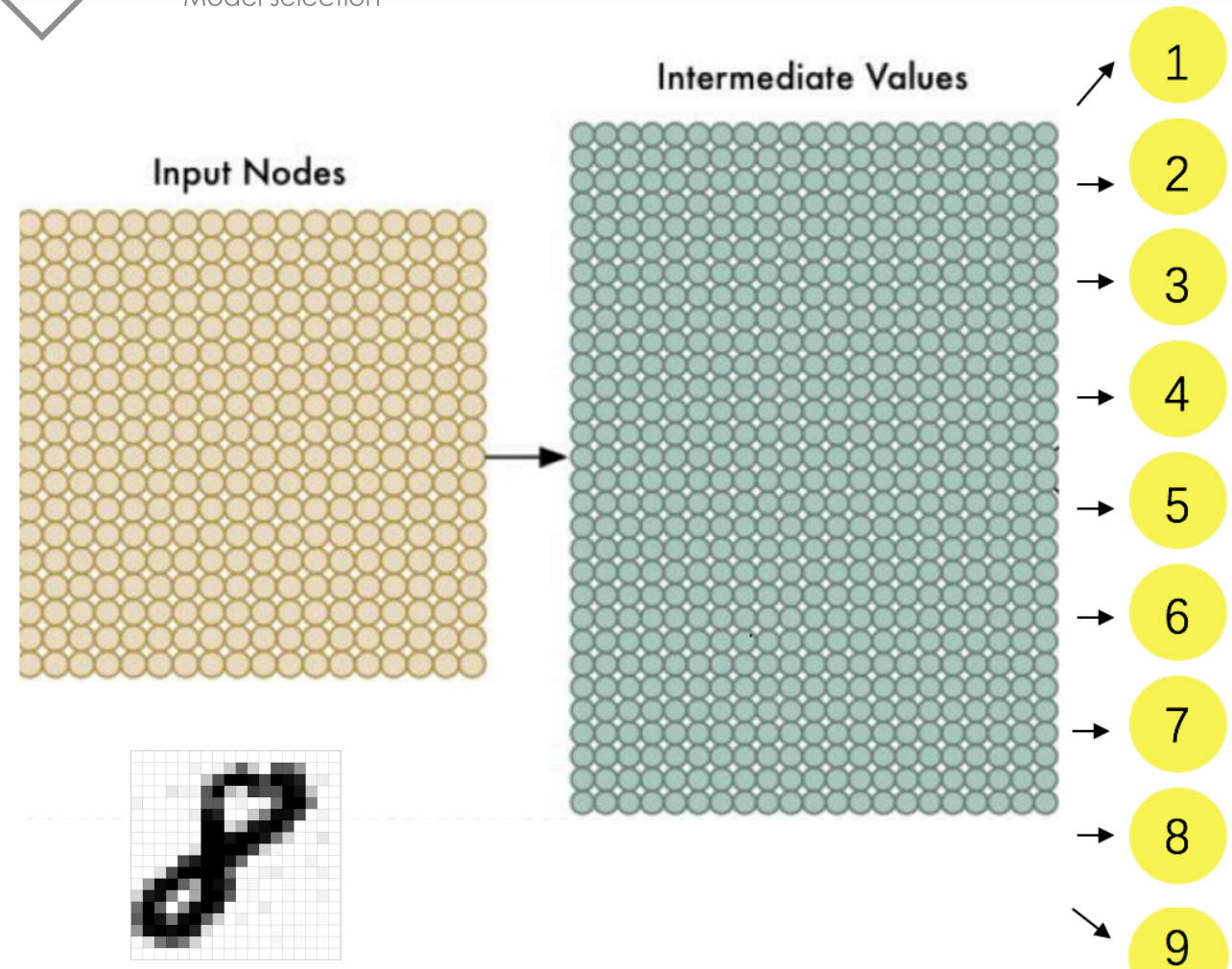




03 Comparison: Softmax Regression

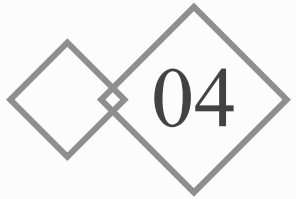
Model selection

Softmax Regression result:



PART 04

Regression & Prediction



Training Data

Regression & Prediction

```
model1= softmaxReg(x, y, hidden = c(), funName = 'sigmoid', maxit = 15, rang = 0.1,type = "class", algorithm = "sgd", rate = 0.01, batch = 1000)
model2= softmaxReg(x, y, hidden = c(), funName = 'sigmoid', maxit = 15, rang = 0.1,type = "class", algorithm = "adagrad", rate = 0.01, batch =1000)
model3= softmaxReg(x, y, hidden = c(), funName = 'sigmoid', maxit = 15, rang = 0.1,type = "class", algorithm = "rmsprop", rate = 0.01, batch =1000)
model4= softmaxReg(x, y, hidden = c(), funName = 'sigmoid', maxit = 15, rang = 0.1,type = "class", algorithm = "momentum", rate = 0.01, batch= 1000)
model5= softmaxReg(x, y, hidden = c(), funName = 'sigmoid', maxit = 15, rang = 0.1,type = "class", algorithm = "nag", rate = 0.01, batch = 1000)
```

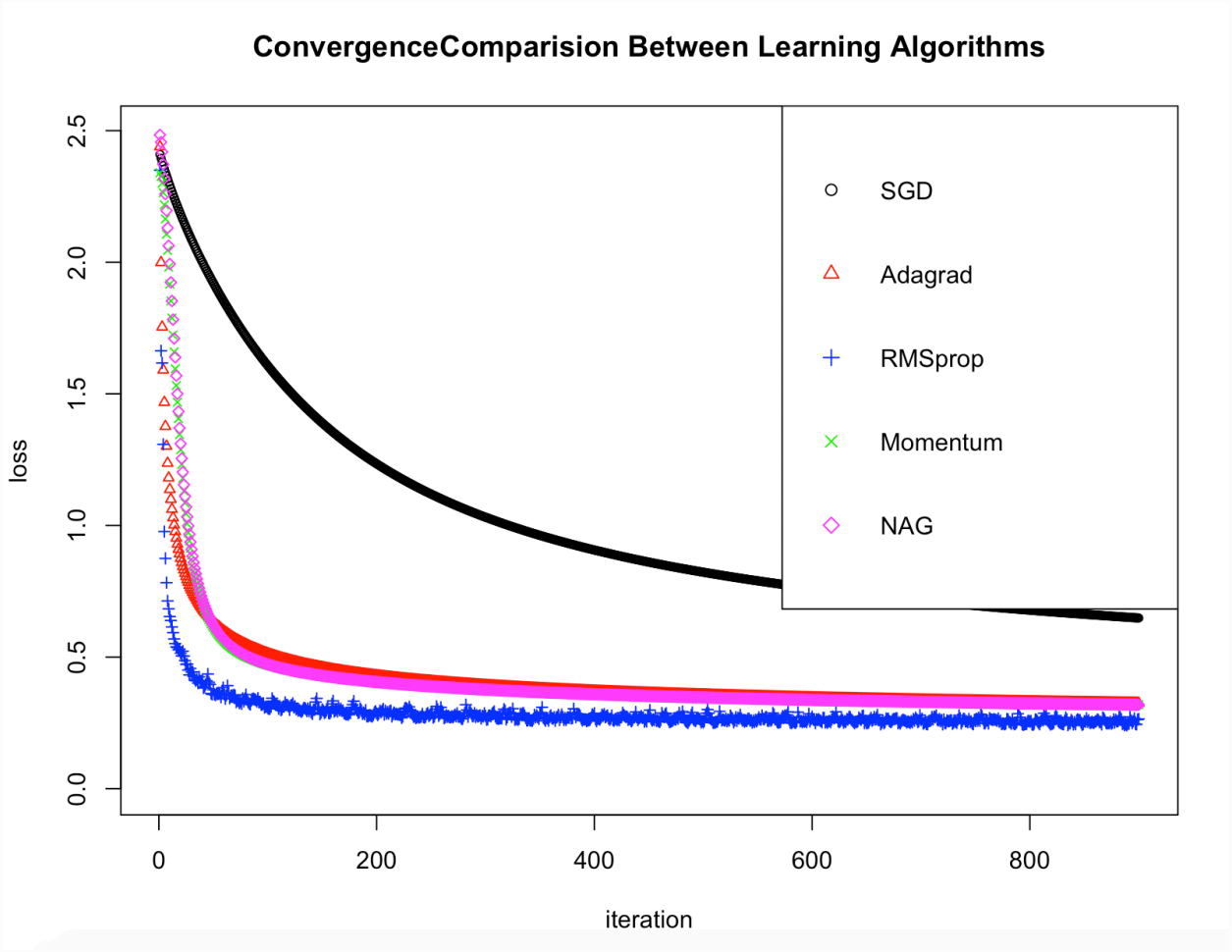
hidden	Numeric vector of integers specifying the number of hidden nodes in each layer, e.g. hidden = c(8,5,...). Default NULL
funName	Name of neural network activation function, including 'sigmoid', 'tanh', 'relu'. Default 'sigmoid'
maxit	Integer for maximum number of iterations. Default 3000
rang	Parameter for the range of initial random weights [-rang, rang]. Default 0.1
type	Parameter indicating the type of softmax task: 'class' denotes the softmax classfication model and the fitted values are factors; 'raw' denotes softmax regression model and the fitted values are raw probability or percentage data of each group. Default 'class'.
algorithm	Parameter indicating which gradient descenting learning algorithm to use, including 'sgd', 'adagrad', 'rmsprop', 'adadelta', 'momentum', 'nag'(Nesterov Momentum), etc. Default 'rmsprop'
rate	Parameter for the initial learning rate. Default 0.0
L2	Boolean variable indicating whether L2 regularization term is added to the loss function and gradient to prevent overfitting. Default FALSE.
batch	Parameter for mini-batch size. Default 50.

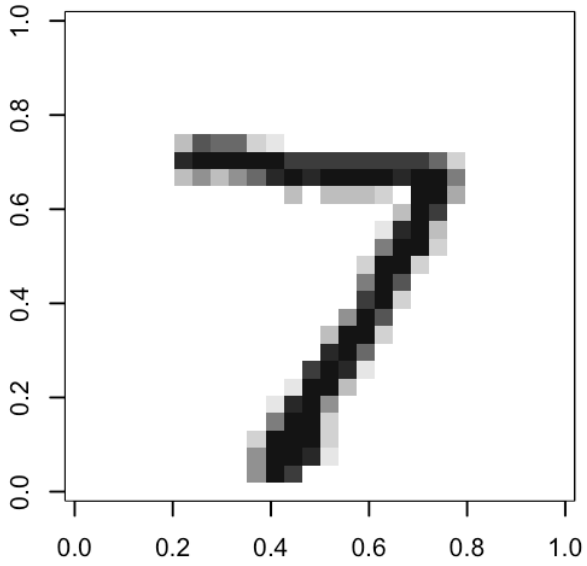
04

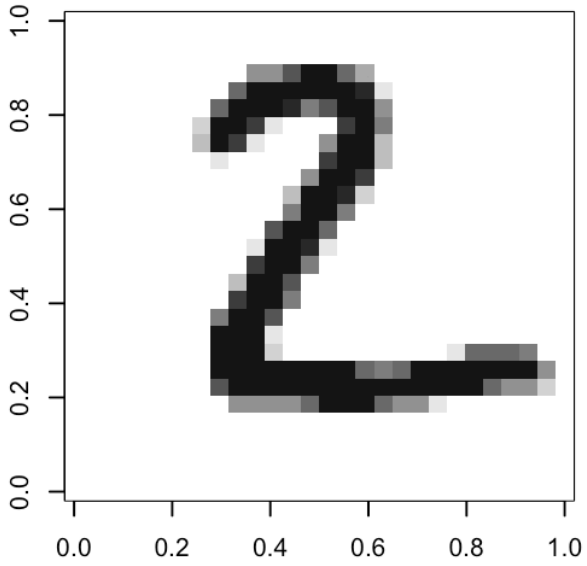
Prediction Results

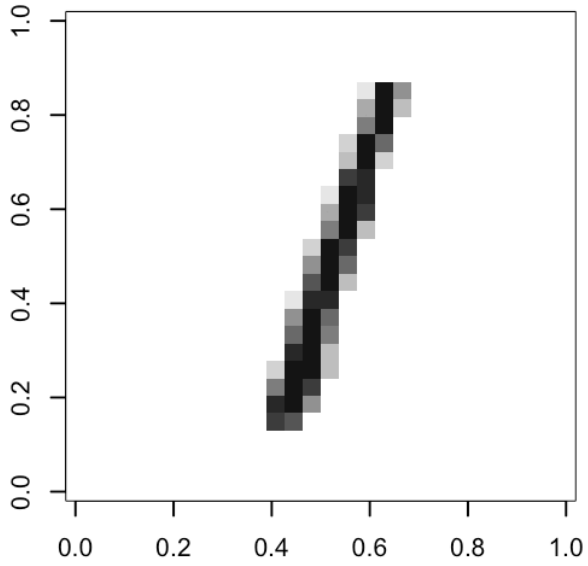
Regression & Prediction

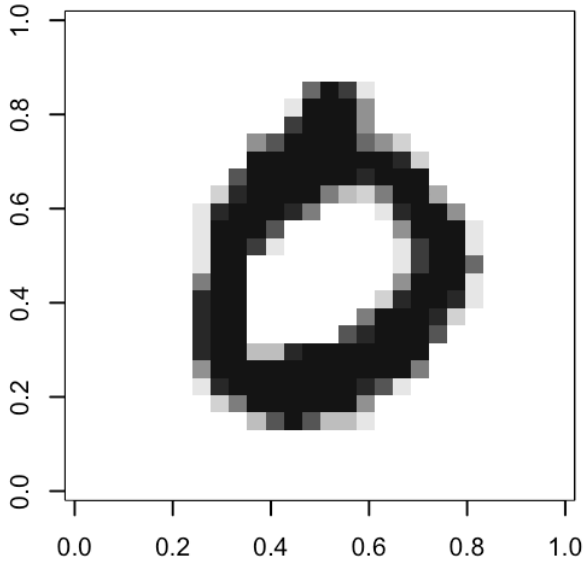
Model	Accuracy	Algorithm
1	85.76%	sgd
2	91.38%	adagrad
3	**92.76%**	**rmsprop**
4	91.54%	momentum
5	91.60%	NAG

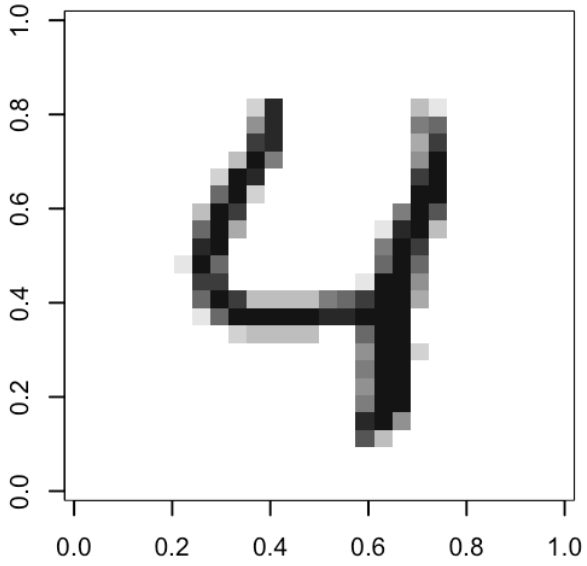


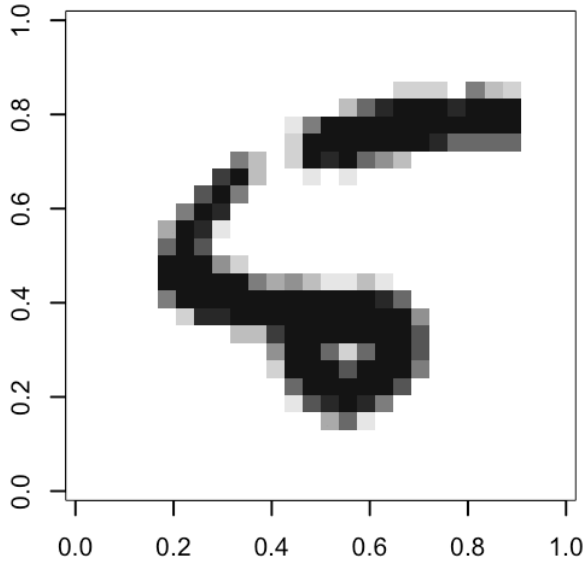

0.0
0.2
0.4
0.6
0.8
1.0
7

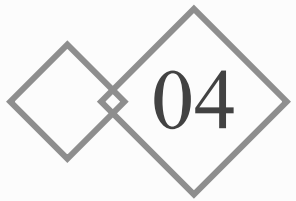

0.0
0.2
0.4
0.6
0.8
1.0
2


0.0
0.2
0.4
0.6
0.8
1.0
1


0.0
0.2
0.4
0.6
0.8
1.0
0


0.0
0.2
0.4
0.6
0.8
1.0
4

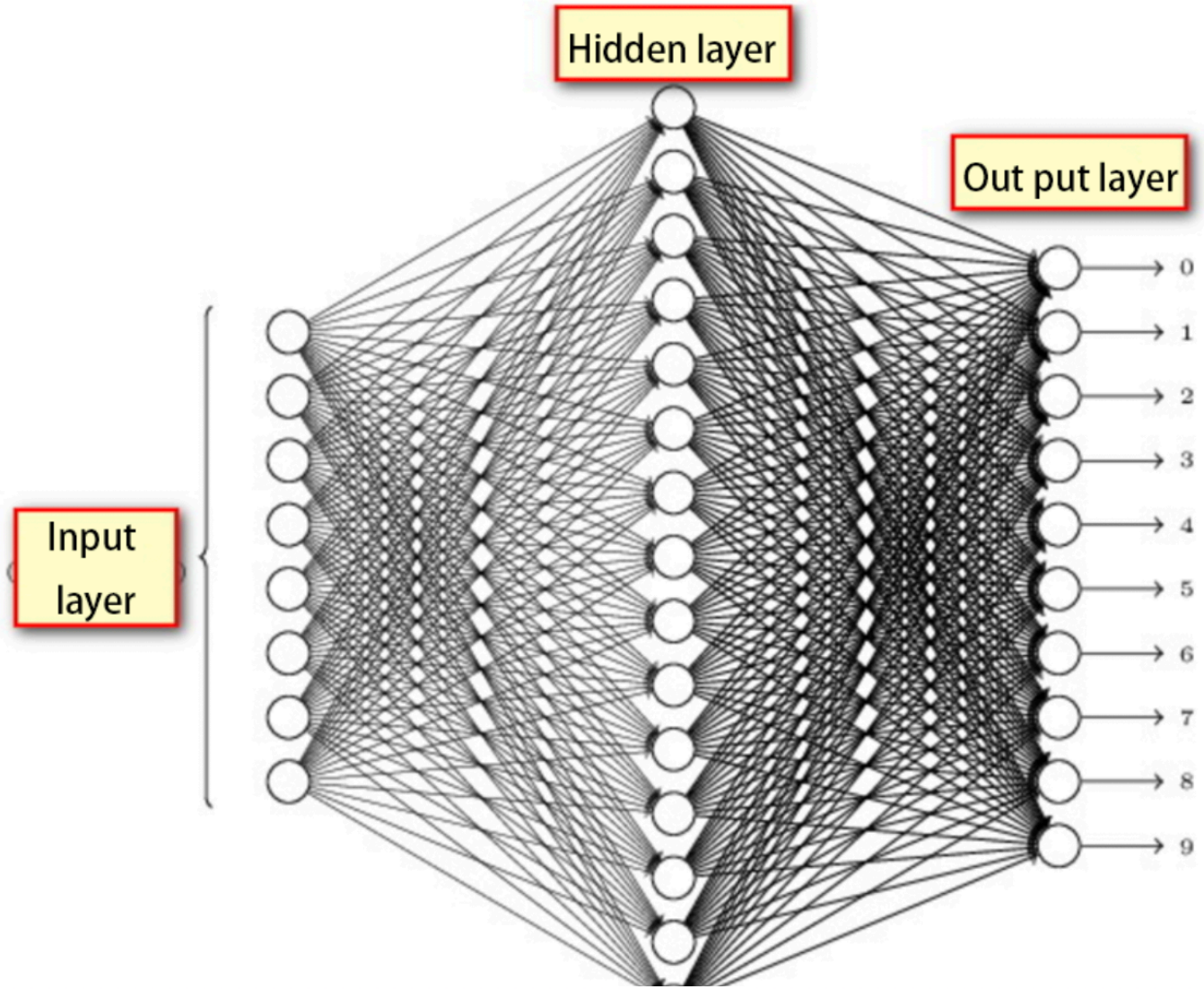

0.0
0.2
0.4
0.6
0.8
1.0
6



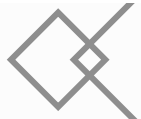
Neural Network

Regression & Prediction

Input layer: 784 nodes
Hidden layer 1: 500 hidden units
Hidden layer 2: 300 hidden units
Softmax output layer: 10 nodes




```
softmaxReg(x, y, hidden = c(500,300), funName = 'sigmoid', maxit = 10, rang = 0.1,type = "class", algorithm = "rmsprop", rate = 0.01, batch =1000)
```



03

Prediction Accuracy

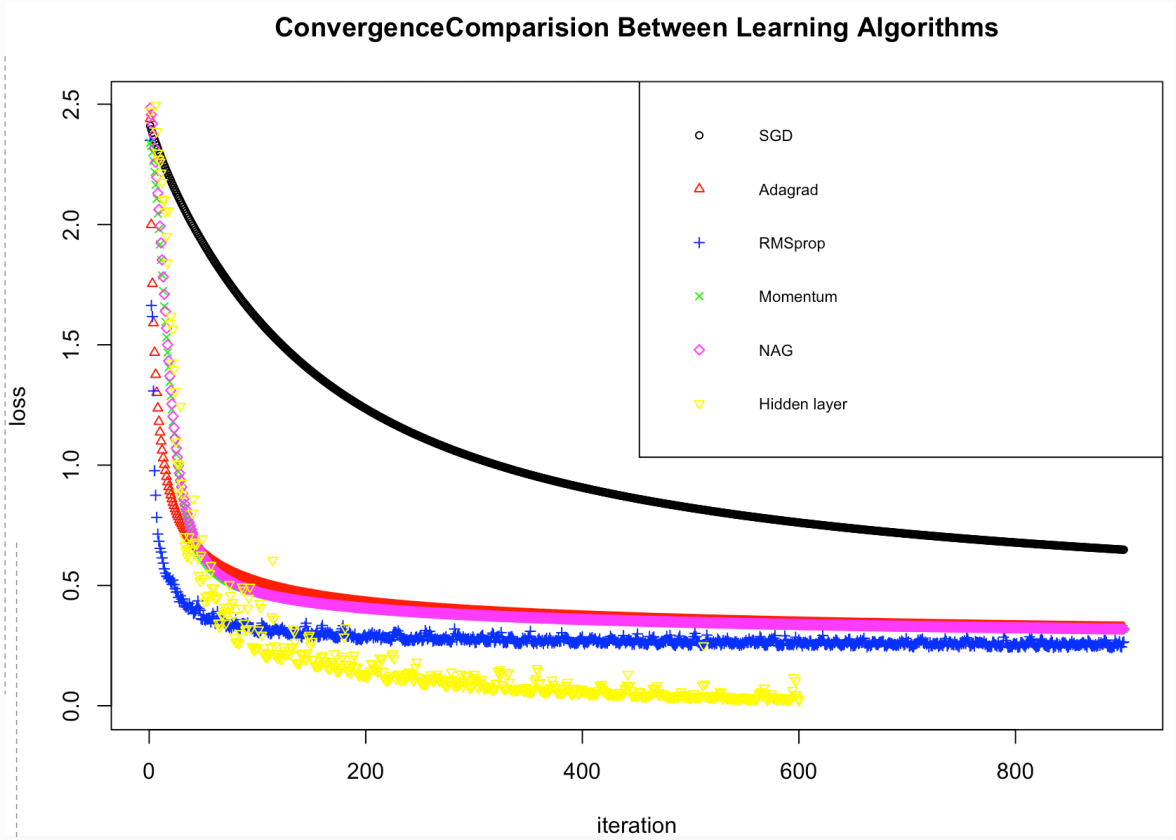
Regression & Prediction

Accuracy

Neural Network Model	**98.16%**
RMSprop Model	92.76%

Loss

Neural Network Model	0.021
RMSprop Model	0.265



Reference

Xichen Ding (2016), Package ‘softmaxreg’
https://cran.r-project.org/web/packages/softmaxreg/softmaxreg.pdf

Softmax Regression
http://ufldl.stanford.edu/wiki/index.php/Softmax_Regression#Softmax_Regression_vs._k_Binary_Classifiers

THE MNIST DATABASE of handwritten digits
http://yann.lecun.com/exdb/mnist/

Data

Handwritten digits recognition

Thanks & Questions