Intelligent Manufacturing Systems

Assignment 3

Due Date: Dec. 3, 2020

Please hand in the "*Microsoft Word*" and "*Code*" files with zipped file name: **IMS_Assignment3_NAME_ID.zip** to *Moodle* by due.

Backpropagation Network (50%)

1. Please use Python to build up backpropagation network (BPN) for "**handwritten digit recognition**". Data set (DRtraining.xlsx) is collected from the Semeion Research Center of Sciences of Communication. It is available in text form and contains 1593 handwritten digits from 80 persons. The images are 16 × 16 pixels square box and in black and white format. (teacher took 15 samples for validation and you only see 1578 samples)

   Each record represents a handwritten digit, originally scanned with a resolution of 256 grays scale. Each pixel of the each original scanned image was first stretched, and after scaled between 0 and 1 (setting to 0 every pixel whose value was under a fixed threshold value 127 of the grey scale (127 included), and setting to 1 each pixel whose original value in the grey scale was over 127).

   We name variable "Pixel001" to "Pixel256". After the array of pixels there is information what digit the image depicts, i.e., "Target0" to "Target9". We use 256 nodes in input layer representing "Pixel" binary variable, and 10 nodes for output layer representing "Target" binary variable, i.e., 0 to 9.

   The pixel format is shown as following figure.

Pixel															
1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16
17	18	19	20	21	22	23	24	25	26	27	28	29	30	31	32
33	34	35	36	37	38	39	40	41	42	43	44	45	46	47	48
49	50	51	52	53	54	55	56	57	58	59	60	61	62	63	64
65	66	67	68	69	70	71	72	73	74	75	76	77	78	79	80
81	82	83	84	85	86	87	88	89	90	91	92	93	94	95	96
97	98	99	100	101	102	103	104	105	106	107	108	109	110	111	112
113	114	115	116	117	118	119	120	121	122	123	124	125	126	127	128
129	130	131	132	133	134	135	136	137	138	139	140	141	142	143	144
145	146	147	148	149	150	151	152	153	154	155	156	157	158	159	160
161	162	163	164	165	166	167	168	169	170	171	172	173	174	175	176
177	178	179	180	181	182	183	184	185	186	187	188	189	190	191	192
193	194	195	196	197	198	199	200	201	202	203	204	205	206	207	208
209	210	211	212	213	214	215	216	217	218	219	220	221	222	223	224
225	226	227	228	229	230	231	232	233	234	235	236	237	238	239	240
241	242	243	244	245	246	247	248	249	250	251	252	253	254	255	256

The digit "3", for example, is shown as following figure.

1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0
1	1	1	1	1	0	1	1	1	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	0	0	0	1	1	0	0	0
0	0	0	0	0	0	0	0	0	1	1	1	1	0	0	0
0	0	0	0	0	0	0	0	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	1	1	1	0	0	0	0	0	0
0	0	0	0	1	1	1	1	1	0	0	0	0	0	0	0
0	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0
0	0	0	0	0	0	0	1	1	1	1	1	0	0	0	0
0	0	0	0	0	0	0	0	0	0	1	1	1	1	0	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	0
0	0	0	0	0	0	0	0	0	0	0	0	1	1	1	1
0	0	0	0	0	0	0	0	0	0	0	0	0	1	1	1
0	0	1	1	1	0	0	0	0	0	0	0	1	1	1	1
0	0	1	1	1	1	0	0	0	1	1	1	1	1	1	0
0	0	0	1	1	1	1	1	1	1	1	1	0	0	0	0

Any question, you can google it (keyword: digit recognition) or refer to the following linkage.
https://machinelearningmastery.com/implement-backpropagation-algorithm-scratch-python/
If you would like to use Tensorflow, Keras, numpy, and pillow, you may refer to
https://medium.com/analytics-vidhya/deep-learning-project-handwritten-digit-recognition-using-python-26da7ed11d1c

(1-a, 40%) For BPN, try to investigate the effects of changing "**PARAMETERS**" such as learning rates, momentum, # of hidden units, # of hidden layers, etc. Show the numerical results and "**DIAGRAM**" from different perspectives (eg., MSE, convergence time, error of training data, error of testing data, etc.). Please show all your work in detail, in particular, you "MAY" need to design your **experiments with different parameters** systematically.

(1-b, 10%) Please predict the digit No.1579 to No.1593 (data source: DRpredict.xlsx) using your **best** established BPN model in (1-a) and fill out the following table.

No.	Digit Number (0-9) you predict
1579	?
1580	?
1581	?
1582	?
1583	?
1584	?
1585	?
1586	?
1587	?

1588	?
1589	?
1590	?
1591	?
1592	?
1593	?

Self-Organizing Map (50%)

2. Please use Python to build up Self-Organizing Map (SOM) model for clustering the "**nutritive value of foods**". Data set (SOMdata.xlsx) is collected from the USDA Food Composition Data. It is available in text form and contains 961 food items with 9 attributes (water content, energy, protein, fat, etc.). The dataset also provide you the "label" for the validation of the clustering results.

(2-a, 40%) For SOM, try to investigate the effects of changing "**PARAMETERS**" such as learning rate, topology type, neighborhood function, radius, etc. Show the numerical results and "**VISUALIZE**" the SOM results via topology map. Please show all your work in detail, in particular, you "MAY" need to design your **experiments with different parameters** systematically.

For the SOM package, you may refer to following linkage.
https://medium.com/@dalicodes/overview-of-self-organizing-maps-som-with-its-python-implementation-in-determining-safe-airlines-db8f6018a2b
For Tenserflow, you may refer to
https://rubikscode.net/2018/08/27/implementing-self-organizing-maps-with-python-and-tensorflow/

Note1: Sometimes we need to normalize data before feeding SOM algorithm. That is, normalize the data so that each variable has mean 0 and variance 1.
Note2: The dataset includes missing values and **impute them with "0"**. How to address it? (Hint: you may use the "label" to assist for missing value imputation).
Note3: For SOM visualization, you may tune parameters for "coloring". A little colouring may help you with the colors for the SOM grid and this aids visualization greatly.

(2-b, 10%) Since the dataset also provide you the "label" for the validation of the SOM clustering results, please compare the SOM results to the label and give a short discussion. For example, if the clustering results is poor, why? what's the cause you guess?

Note

1. Show all your work in detail. **Innovative** idea is encouraged.
2. If your answer refers to any external source, please "must" give an academic citation. Any

"plagiarism" is not allowed.

3. For BPN, reference of data source (Semeion Handwritten Digit)

Abstract: 1593 handwritten digits from around 80 persons were scanned, stretched in a rectangular box 16x16 in a gray scale of 256 values.

Data Set Characteristics: Multivariate
Number of Instances: 1593
Area: Computer
Attribute Characteristics: Integer
Number of Attributes: 256
Date Donated: 2008-11-11
Associated Tasks: Classification
Missing Values? N/A

Source:
The dataset was created by Tactile Srl, Brescia, Italy (http://www.tattile.it/) and donated in 1994 to Semeion Research Center of Sciences of Communication, Rome, Italy (http://www.semeion.it/), for machine learning research.
For any questions, e-mail Massimo Buscema (m.buscema '@' semeion.it) or Stefano Terzi (s.terzi '@' semeion.it)

4. For SOM, reference of data source (USDA Food Composition Data)

Title: Nutritive value of foods
Source Information:
-- Source: USDA Food Composition Data
http://www.nal.usda.gov/fnic/foodcomp/Data/
-- Date: 1999
Relevant Information:
This data set consists of values for food items, which are described in terms of household measures for the food and are provided for the weight of the household measure. The objective is to find dependencies between the food properties and then cluster the data and analyse the clusters (i.e. the food groups).
Number of instances: 961
Number of Attributes: 9

- Number of Labels: 2
- There are some missing values, all are unknown information.

Attribute Information:
All atrributes are numeric and continuous
N. Attribute

1	water	(water content (%))
2	energy	(food energy (kcal))
3	protein	(protein content (g))
4	fat	(fat contents (g))
5	cholesterol	(cholesterol content (mg))
6	carbohydrate	(carbohydrate content (g))
7	vitamin a (iu)	(vitamin A content (IU))
8	vitamin a (re)	(vitamin A content (RE))
9	ascorbic acid	(Ascorbic acid content (mg))

N. Label

1	saturated fat	(does the food contain saturated fat)
2	food description	(description of food)