TRIED TPC01 Report

Carl Robinson - 11th Dec 2017

Part 2: Recognition of Handwritten Digits

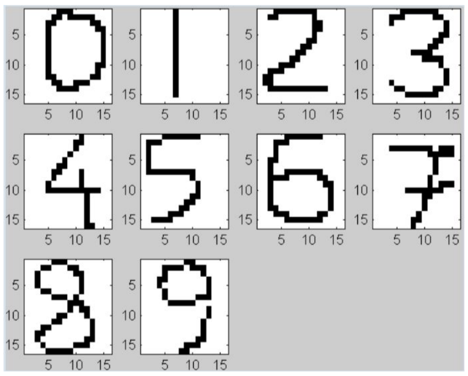
Objective, Data and Method

Objective & Method

The aim of this research is to investigate the implementation of self organising maps (SOM) for classification of handwritten digits, and to discover the optimal data coding and learning parameters (map size, iterations, temperatures). We use hexagonal SOMs, with a rectangular Gaussian neighborhood function, randomly initialized. Additionally we use two stages of batch SOM training; stage 1 uses a higher initial temperature to consider a wider neighborhood and stage 2 uses smaller temperatures to refine the vector quantization.

Dataset and Codings

The dataset contains 480 digits coded in binary (± 1), in an array of 256 rows x 480 columns. Each 16x16 binary image is represented by a 256-dimensional column vector in the file.



The following codings are used:

- HX / HY: Histogram of projections on the x-axis or y-axis (16 components)
- PG (Left profile) / PD - (Right Profile): Coordinate of the first white/black transition from the left/right (16 components)

We combine these codings, and normalise the data in the interval [-1, 1], to form 3 learning files:

1. HX, HY coding (32 components). Input file: hx_hy.txt
2. PG, PD coding (32 components). Input file: pg_pd.txt
3. HX, HY, PG, PD coding; (64 components). Input file: hx_hy_pg_pd.txt

Results & Conclusions

1)

340 of 480 digits used for training, the remaining 140 used for test; Map dimensions 12*12
Stage 1 parameters: iterations (100); temperature range [20.0 to 2.00];
Stage 2 parameters: iterations (100); temperature range [2.00 to 0.10];

Train and Test performances for each of the 4 codings

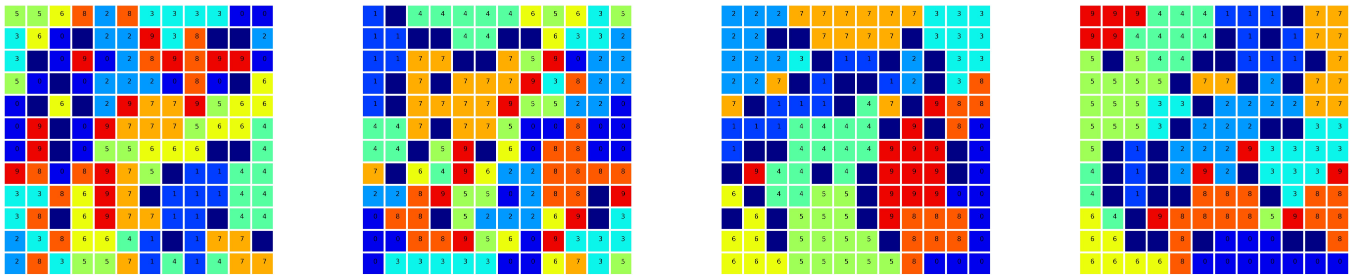
Coding	Training Set Performance	Test Set Performance
x.txt (no coding)	0.794	0.550
hx_hy.txt	0.826	0.743
pg_pd.txt	0.976	0.950
hx_hy_pg_pd.txt	0.994	0.936

- Iterations made little difference to the results; there was a slight improvement with a higher number.
- The best performance was obtained by starting with a large temperature of 20.0, and bringing this down to a very small temperature of 0.10.

- This combination of parameters allowed the referents to both fully spread across the data space and refine their positioning within the finite iterations allowed.
- All codings performed significantly better than the raw data, and significantly reduced the dimensionality of the data from 256 to only 32 or 64 components.
- The PG / PD coding consistently produced the lowest test classification error. This is because it exploited the characteristics of the handwritten characters the best, differentiating between similar digits such as 5 and 8 more effectively than a simple histogram representation could.

Majority vote referent classifications for each of the 4 codings

x (no coding) **hx_hy** **pg_pd** **hx_hy_pg_pd**

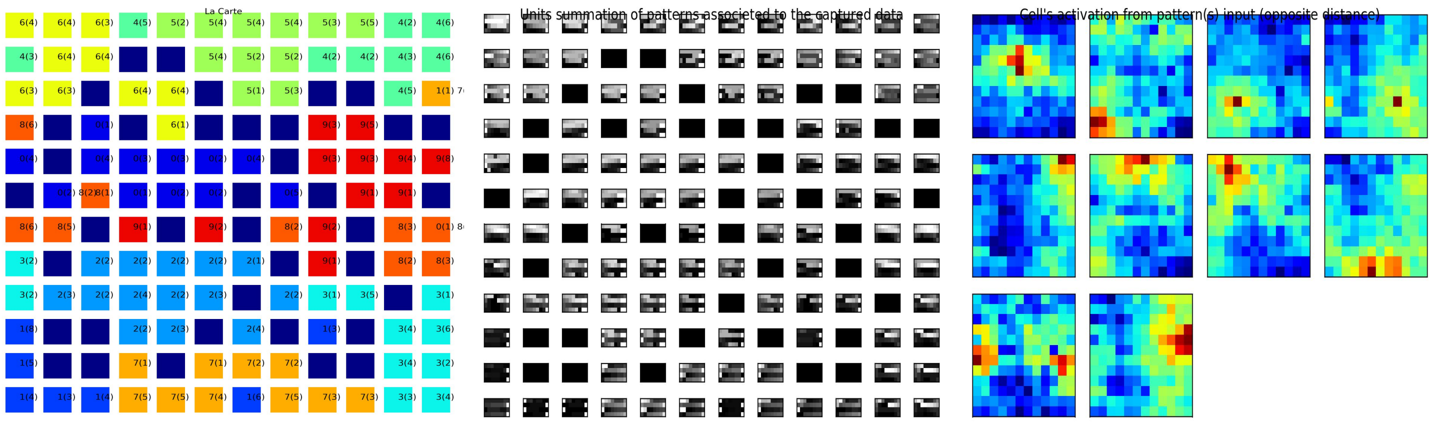


- The pg_pd coding groups all similarly-classed referents together, and does not split apart groups like hx_hy_pg_pd does. Also, the dark blue referents that have not been classified lie on the decision boundaries between classes. This shows that the order of the map is preserved, as neighbouring referents are assigned to the same class. We find that *pg_pd is the optimal coding,* so we will use this coding only from now on.

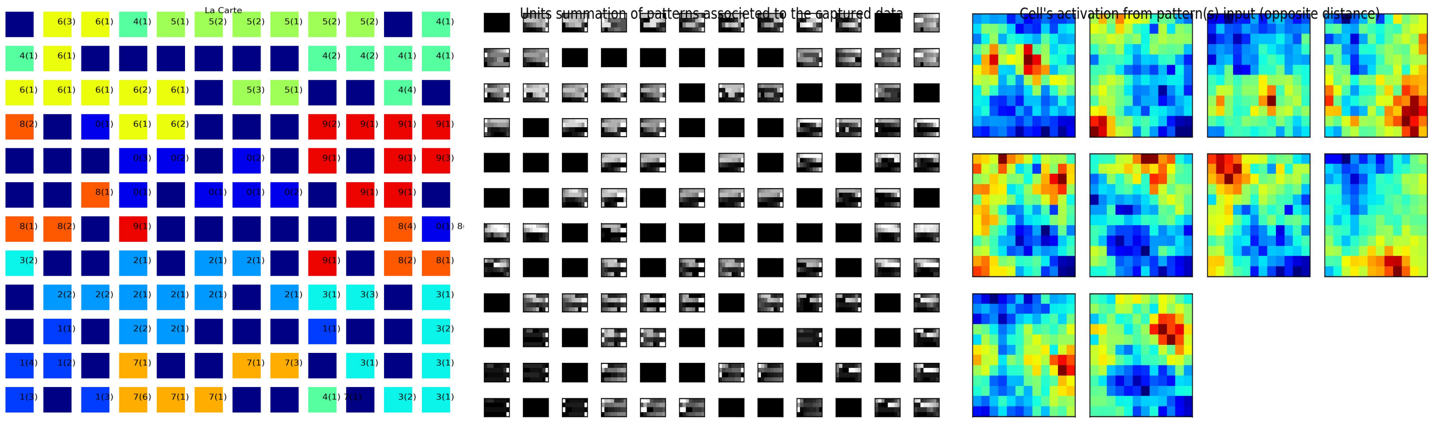
2)

Referent Classification Map (left), Referent Capture Summation (middle), and Neuron Activations (right)

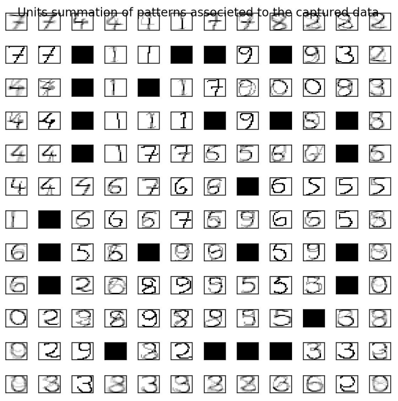
Training Set: (performance = 98.8%)



Test Set: (performance = 92.1%)



- On the classification map we print the class decided by majority vote, plus the number of examples of that class that contributed to the vote. This highlights the majority vote mechanism, and also shows us that neurons on the decision boundaries between classes are more likely to be contested by roughly equal numbers of examples belonging to two different classes.
- From the referent capture summations, we see that a similar pattern leads to similar classification results on the classification map. This shows that referents (neurons) close to each other on the map capture similar data, and are therefore associated with similar training/test examples.
- We see a generally higher level of neuronal activation for each of the first 10 digits of the test set than for the first 10 digits of the training set.
- If we pass the raw x.txt training data through the model and display the referent capture information, we see that each referent is a combination of all observations assigned to it. The clearer the representation, the more homogenous its member observations are, so we see that most neurons represent a single digit. The more blurred representations are referents with a mix of classes assigned to them. The black squares represent neurons with no observations assigned to them.



3)

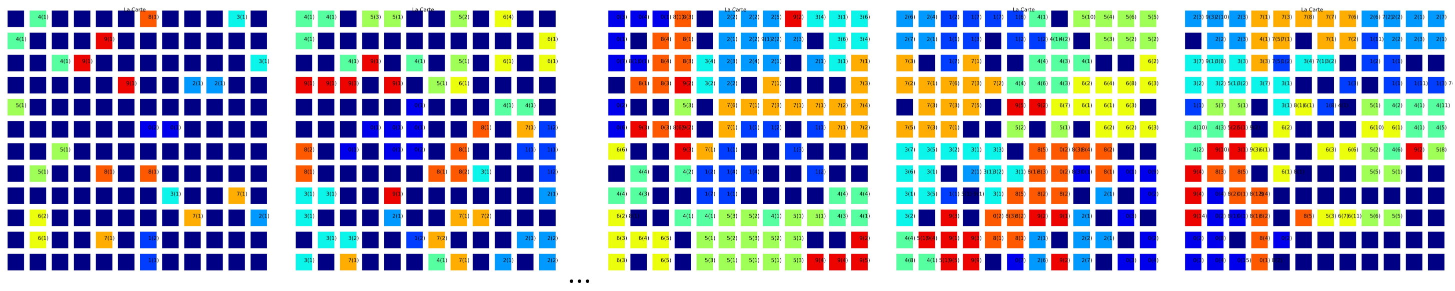
Training set and test set classification performance for varying training set sizes

Training Set Size (out of 480 total)	Training Set Performance	Test Set Performance
50	0.800	0.516
100	0.990	0.661
150	0.973	0.821
200	0.975	0.871
250	0.992	0.913
300	0.980	0.917
350	0.986	0.962
400	0.998	0.963
450	0.998	1.000

- As expected, the larger the training set, the higher the train & test classification performance of the model.

Test set classification of referents

Total examples = 480. Trained on N examples = {50, 100, ..., 300, 400, 450}



- When N is low, very few referents can be classified as there just isn't enough data to maximise the log likelihood and enable the referents to find a good generalised solution.
- When N is very high (N=450), the decision boundaries formed by unclassified referents grow larger, indicating more clearly defined separation between groups of referents, and therefore a more precise and certain classification decision.
- Also, very few examples remain for the test set; classifying a smaller test set is likely to lead to a higher performance as there are simply fewer to classify, which helps to explain the perfect classification score of 100%.