Kohonen maps on hand-written digits

Unsupervised and reinforcment learning in neural networks

Nicolas Casademont · Teo Stocco

Abstract

Using MNIST dataset, this project shows how Kohonen maps can be adapted to classify 4 selected hand-written digits. It suggests a method to assign digits to Kohonen prototypes and analyze the variation of parameters: network size, neighborhood function width. The best parameters achieve an error rate of 30 % over $4 \cdot 500 = 2000$ samples.

Learning rate

Starting with default parameters (36 neurons, unit distance, $\sigma = 3$), the learning rate has been empirically chosen based on the number of iteration. $\eta = 0.005$ seems to converge fast enough with accurate precision over 5000 iterations.

The convergence is determined using $L2$ norm over all samples (slow): once the gradient of the norm stablizes around 0 given a threshold tolerance (e.g. 0.0025 works well with a learning rate of $\eta = 0.005$). Splitting the data into train and test datasets (naive 10 % cut) have been tried but no overfitting illness appeared. Thus in order to keep the algorithm efficient, a 10-elements sliding window over the train sample is used to approximate the full norm.

Prototypes

visualization and description of the learnt prototypes

Visualize your prototypes. Describe your results.

Digit assignment

a description and visualization of your method of assigning the digit that is repre- sented by a prototype

Using the information in (labels.txt), find a way to automatically assign one digit to each prototype that is represented best by it. Visualize and interpret your results.

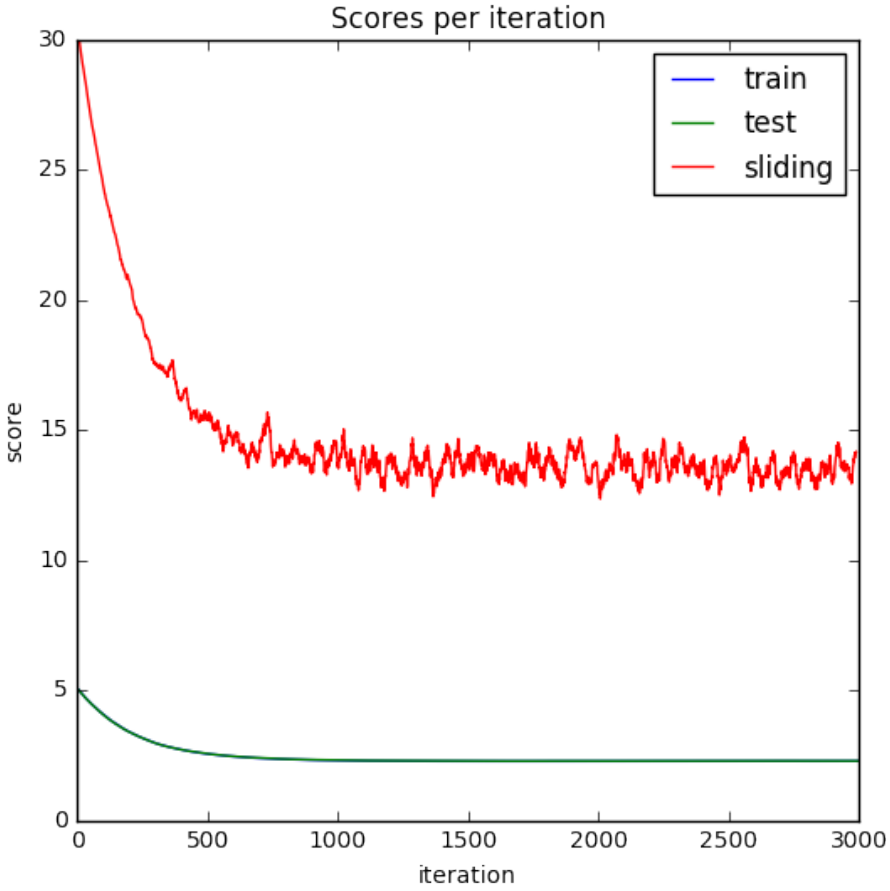
Scores per iteration
train
test
sliding
score
iteration
0
5
10
15
20
25
30
0
500
1000
1500
2000
2500
3000

Figure 1:

3. Pour ça en gros j'ai pris le prototype le plus proche de chaque data et ensuite compté les labels par prototype (tu as le code dans le dernier push) Je t'ajoute l'image avec les confidences.

Exploration

Network size

Explore different sizes of the Kohonen map (try at least 3 different sizes, not less than 36 units).

1. Network size : plus c'est grand plus tu as de façon d'écrire chaque nombre. (quand j'aurais fini j'ajouterais une image grande avec la confidence)

Neighborhood function width

Explore different widths of the neighborhood function (try at least $\sigma = 1$, 3, and 5) Does the optimal width depend on the size of the Kohonen map?

2. width neigh : plus c'est grand plus tu t'approche d'une moyenne (c'est logique tu les ramènes tous à chaque item que tu apprends)

Dynamic neighborhood function width

your results of the exploration of different network sizes and widths of the neighborhood function with a discussion of the results, and your results and discussion of varying the width of the neighborhood function over time.

Until now, the width of the neighborhood function has been constant. Now, start with a large σ and decrease it over the runtime of the algorithm. Does it improve your result?

1. c'est un moyen de représenter l'éloignement des noeuds
2. ça permet de stabiliser plus proprement les prototype
3. Petit truc je te fournirais une image de la fonction qui réduit sigma avec les différents coefficients.

Conclusion

Références