INTRODUCTION

Humans have an outstanding generalization power for visual categorization: we can recognize multiple instances of a class of objects even after seeing only one exemplar. This property constitutes the Holy Grail for researchers in the computer vision field, but also draws the attention of researchers in the AI community in general, as well as cognitive neuroscientists and psychologists, due to the modern mutualism between these areas.

In the past few years, the rise of deep learning has boosted the efforts to create the ultimate artificial vision system, with record-breaking performances on complex datasets such as ImageNet. However, these rich algorithms often require large amounts of high-quality data, with many variations of objects belonging to the same class, to be trained on. In the present work, we make a short demonstration of how a simple deep learning architecture can achieve good performance in a digit classification task, but also show how it cannot learn features that remain invariant to affine transformations of the digits. On the other hand, when exemplars belonging to the affine-transformed dataset are used for training, the network improves its performance significantly on the transformed test set.

NEURAL NETWORK ARCHITECTURE

The neural network implemented contained two hidden layers (500 and 150 units each), as well as input and output layers (Figure 1). All of the units had sigmoid non-linearities. Backpropagation and a mini-batch gradient descent implementation were employed to minimize the cross-entropy cost function with an L2 regularization term. All the network hyper-parameters were tuned using grid-search on Training Set 2, defined in Datasets.

DATASETS

The datasets used belong to affineMNIST [1]. Training Set 1 corresponds to the just-centered version of affineMNIST training-validation dataset (the original 60,000 28×28 images from MNIST [2] remapped to 40×40 images). Test Set 1 corresponds to 10,000 images from the just-centered version of affineMNIST test set (Figure 2). Training Set 2 corresponds to all 60,000 images from a batch of the transformed version of the affineMNIST training and validation dataset. Test Set 2 corresponds to 10,000 randomly sampled images from the transformed version of the affineMNIST test set (Figure 3).

RESULTS

For each training set, ten different instances of the network were initialized and trained. Mean performance and sem are reported for each test set on Table 1.

<table>
<thead>
<tr>
<th>Classification accuracy</th>
</tr>
</thead>
<tbody>
<tr>
<td>Training Set</td>
</tr>
<tr>
<td>1</td>
</tr>
<tr>
<td>2</td>
</tr>
</tbody>
</table>

Table 1. Mean (sem) network performances across ten instances.

DISCUSSION

- The long-known problem of learning mappings between inputs and outputs that remain invariant to certain transformations is easily observed in the results from Table 1: there is a large drop in accuracy when testing the network in digits that were affine - transformed when compared to digits that appeared centered within the image frame.
- If we consider that the full input space is composed by all the possible instances of every digit, poor generalization is not surprising, as the network is only exposed to a very small sub-space of this distribution and tunes its parameters to describe it as accurately as possible, while the transformed digits present a bigger amount of intra-class variability, which remains as a very unexplored region of input space (Figure 4). Thus, locality [3], a central assumption of many machine learning algorithms, is violated and performance drops.

CONCLUSIONS

- We have shown that naive feed-forward neural networks cannot make inferences in regions of input space they have not properly explored during training.
- The performance of a classification algorithm depends on architecture style and optimization, but most of all, on the quality of the dataset used for training.
- It is important to design architectures that build desired properties into the network, when these are difficult to directly learn from the dataset. Such is the case for Convolutional Neural Networks, whose design introduces translation invariance into the learned mappings [4].
- The simple example shown here illustrates the need for problems of Transfer Learning [4] or Domain Adaptation [5] to be properly addressed, as they could help improve algorithmic performance in situations where the data probability space is not extensively sampled.

REFERENCES