Modeling visual numerosity with deep learning

1 Introduction

We investigate the illustration of 'visual number sense' in deep hierarchical models, by Zorzi et. al.. By building a multi-layer Deep Belief Network, the authors show that it is possible to extract features with a sense of basic numerosity in the range of integers 1 to 32. More specifically, using features in deeper network layers, it is possible to linearly classify integers below or above 8 (or 16) with fair accuracy. Further, the weber ratio calculated from representative numerosity neurons were estimated (w=0.15) to be similar to that of human adults. This results correlates well with cognitive studies of animals and humans [2], which indicate that numerical competence did not only emerge with linguistic or symbolic understanding, but rather a built-in biological function[1].
One goal of this study is to discover high-level induction with hierarchical neural networks. Experiments [reference required] suggests that children might learn larger numbers through induction. For instance, she may construct the idea of 'seven' which is out of the her basic biological function, by lining up 'four' with 'three' side-by-side. In this study, we attempt to discriminate numerosity on a finer scale with hierarchical connectionist models. Further, we build unsupervised temporal links and identify the model's ability to discover induction patterns in the visual numerosity input. Finally, depending on time and resource constraints, we would like to compare this with data collected from the same human cognition task.

2 Experiments

To develop a sense for numerosity using deep networks, we first attempt to replicate Zorzi et. al.'s results. I exactly replicated generation of training data in Zorzi et. al. and produced data in the form of square visual fields, as shown in Figure **??**.
Instead of Restricted Boltzmann Machines, we apply a more efficient technique of using an auto-encoder with pre-training then fine-tuning with quasi-newton optimization. The two-layer auto-encoder is pre-trained layer-wise, then optimized as a whole to reconstruct the original input.
As in Zorzi et. al., the architecture of the auto-encoder neural networks has 80 units on the first layer and 400 units on the second. The first layer neurons learn center-surround filters, as seen in Figure 1.

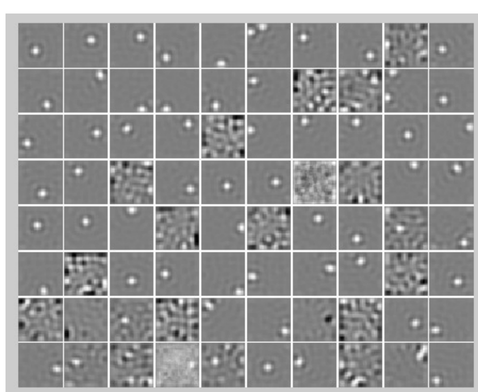

Figure 1: Layer one center-surround filters learned by the neural network

With a pre-trained (layer-wise) auto-encoder, a logistic classifier is trained on the second layer features to produce an accuracy of 82.7% accuracy, compared with the reported 93%. Looking at the reconstruction objectives, the two-layer network has yet large reconstruction errors (2.5e6 as compared to 1.5e6). After optimizing the two-layer network, reconstruction visualizations are much better, see Figure 2, and a linear classifier is able to produce 87.5% accuracy on discriminating numerosities below and above 16. Further I classified the same data with reference number 8, and the accuracy becomes 94%, Zorzi et. al. did not specify their 93% result was on reference number 8 or 16.

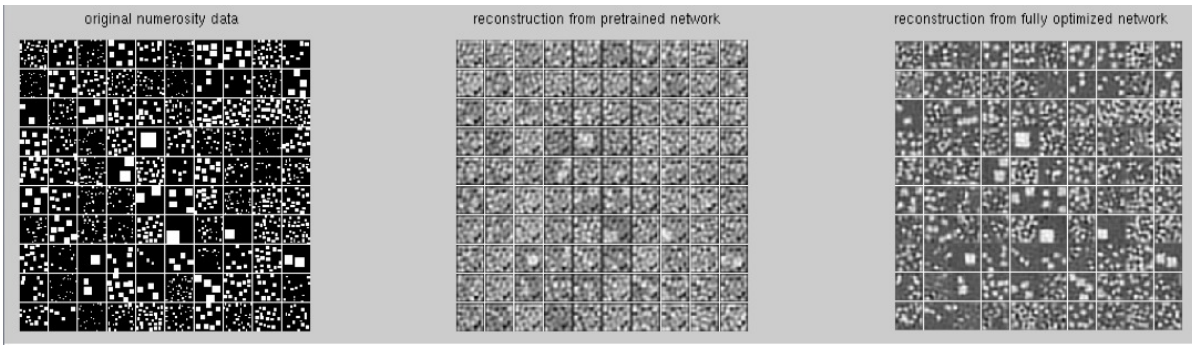

Figure 2: Reconstructions made by the deep auto-encoder

As a starting model, the deep auto-encoder seems to be able to nearly replicate Zorzi's model.

2.1 Denoising Auto-encoders and whitening

I did experiments adding noise to the training process, it turns out that unless the noise level is carefully selected on a very fine-scale (adding a tiny bit of noise), results did not improve (87.8% at best). I also tried whitening. After whitening, the first layer weights could not converge to center-surround filters, and classification accuracy dropped.

2.2 Weber ratio

To reproduce Weber ratio using a sigmoid fit, I trained the classifier and produced results for reference numbers 8, 16. The resulting plot is shown in Figure 3. It took some googling to find how to find the weber ratio, but it turns out we need to fit a error function (sigmoid-shaped) to the data.

2.3 Regression experiment

3 Developmental progression of weber ratio

There has been behavioral psychology evidence that humans progressively improve their abilities to discriminate numbers. In other words, weber ratio for telling apart numbers increase consistently as one grows up. Here we perform an experiment to simulate the development progression using the deep auto-encoder, and observe the changes in weber ratio.

The details of the experiment is as follows: we apply the same deep auto-encoder as in the past sections. To simulate development progress, we perform stochastic gradient descent on the network. More specifically, since the deep network should be learned progressively, we apply the following two experiment scenarios: [1] Using stochastic gradient descent, we train the first and second layers iteratively, and observe the changes in weber ratio as optimization progresses. By iteratively, we mean that once a gradient step is taken on layer one, it is kept fixed before a gradient step is taken on layer 2, and so on.

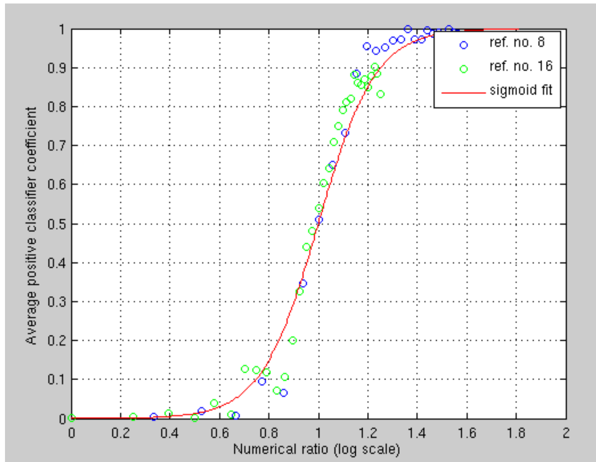

Figure 3: Numerosity comparison using deep auto-encoder

[2] Using stochastic gradient descent, we first train the first layer auto-encoder, and observe changes in the weber ratio as optimization progresses. Then, we train the second layer. While adding these features to the classifier, we observe changes in the weber ratio as optimization progresses.

References

[1] A. Nieder. Counting on neurons: the neurobiology of numerical competence. *Nature Reviews Neuroscience*, 2005.

[2] M. Piazza, V. Izard, P. Pinel, D. L. Bihan, and S. Dehaene. Tuning curves for approximate numerosity in the human intraparietal sulcus. *Neuron*, 2007.