MODELLING RETINAL FEATURE DETECTION WITH DEEP BELIEF NETWORKS IN A SIMULATED ENVIRONMENT

Diana Turcsany and Andrzej Bargiela
School of Computer Science
The University of Nottingham
NG81BB, Nottingham, United Kingdom
Email: {dxt, abb}@cs.nott.ac.uk

Tomas Maul
School of Computer Science
The University of Nottingham Malaysia Campus
43500, Semenyih, Malaysia
Email: Tomas.Maul@nottingham.edu.my

KEYWORDS
Retina model; Neural learning; Deep belief network; Probabilistic modelling; Simulated environment

ABSTRACT
Recent research has demonstrated the great capability of deep belief networks for solving a variety of visual recognition tasks. However, primary focus has been on modelling higher level visual features and later stages of visual processing found in the brain. Lower level processes such as those found in the retina have gone ignored. In this paper, we address this issue and demonstrate how the retina’s inherent multi-layered structure lends itself naturally for modelling with deep networks. We introduce a method for simulating the retinal photoreceptor input and show the efficacy of deep networks in learning feature detectors similar to retinal ganglion cells. We thereby demonstrate the potential of deep belief networks for modelling the earliest stages of visual processing.

INTRODUCTION
Vision systems of humans and other biological systems have long been within the centre of interest in neuroscience and biology. The main focus of this work is the first stage of visual information processing in mammals, implemented by the retina.

Extending our knowledge of visual information processing mechanisms within biological systems has key importance in finding suitable learning algorithms for artificial retinæ. Such algorithms are much sought after, as building better computational models of the retina is not only crucial for advancing the field of retinal implant engineering, but could also improve image processing and computer vision algorithms.

Despite discoveries regarding the anatomy and physiology of neural structures, understanding biological vision systems remains an open problem. Due to the high number, diverse functionality and complex connection patterns of neurons, our knowledge regarding the roles of cells and circuits of the visual pathway is still incomplete. Even well studied processing units, such as the retina are not fully understood and recent studies have revealed a lot more complexity in the functionality implemented by retinal cells than traditionally believed (Golisch and Meister, 2010).

As a consequence of this incomplete knowledge, currently the design of computational retina models involves dealing with a great amount of uncertainty and only partially known information. In such an environment our aim is to develop computational models which exhibit substantial fidelity to the retina’s currently known neural structure, while also being highly adaptable. We anticipate that flexible probabilistic models, such as deep belief networks (Hinton et al., 2006) studied here, can provide the required adaptive power.

Modelling The Retinal Network
Mammalian retinæ contain 6 main cell types: rods, cones, and horizontal, bipolar, amacrine, and ganglion cells, which are organised into consecutive layers as illustrated in figure 1a. The first layer contains photoreceptors: rods and cones. Rods are essential for sensing light in darker environments, whereas cones provide the first processing step in colour vision. The retinæ of humans and trichromatic monkeys contain 3 types of cones: blue cones sense short, green cones medium and red cones long wavelength light. Colour vision is possible by comparing the response of different wavelength cones.
The synaptic connections between the photoreceptor layer and the horizontal and bipolar cells form the outer plexiform layer. The main role of horizontal cells is in local gain control, which guarantees the input signal reaching the further processing units is kept within the appropriate range. Bipolar cells receive input from photoreceptors and in the inner plexiform layer they connect onto ganglion and amacrine cells. Bipolar cells have centre-surround receptive field organisation and can either connect to the ON or the OFF pathway, which are respectively responsible for detecting light objects on dark background and vice versa.

Amacrine cells serve important roles in various tasks, e.g. object motion detection (Olveczky et al., 2003). The highest level of information processing within the retina is implemented by ganglion cells. These cells receive input from a number of bipolar cells and occasionally from amacrine cells. Different ganglion cell types detect specific visual features and transmit the extracted information through the optic nerve towards higher processing areas located in the visual cortex. The most well-known ganglion cells are the ON/OFF local edge detectors, exhibiting centre-surround antagonism. ON-centre cells receive excitatory signals when light appears in their receptive field centre and inhibitory signals resulting from light in the surround, while the opposite is true for OFF-centre ganglion cells. The response patterns of these cells are most often modelled as Difference-of-Gaussians (DoG) filters (see figure 2).

For more in depth descriptions of the retinal network, please refer to review papers by Kolb (2003), Wässle (2004) and Masland (2012).

Figure 2: Difference-of-Gaussians (DoG) filters: common models of ON (left) and OFF (right) type ganglion cells.

Neural computation units in the retina have been modelled at different abstraction levels, ranging from detailed models of single neurons (Kameneva et al., 2011; Usui et al., 1996; Velte and Miller, 1997), to populations of neurons (Pillow et al., 2008; Shlens et al., 2008) and networks of interacting retinal cell types (Gaudiano, 1992; Kien et al., 2012; Maul et al., 2011; Teeters et al., 1997). Our work focuses on higher level description of retinal circuits and assesses large-scale network models.

**Multi-layer Networks**

To learn a multiple layer generative model of the data where each new layer corresponds to a higher level representation of information, Hinton et al. (2006) train a deep belief network (DBN) by “greedy” layer-by-layer learning using the unsupervised restricted Boltzmann machine (RBM) algorithm. The network parameters obtained from the unsupervised learning phase are subsequently fine-tuned using backpropagation. The potential of DBNs for learning meaningful features is demonstrated on visual recognition tasks and text retrieval (Hinton and Salakhutdinov, 2006). Since Hinton et al. (2006)’s efficient training method for deep networks was introduced, there has been increasing research in this direction. Successful applications of deep belief networks and other deep architectures (Lee et al., 2009a; Salakhutdinov and Hinton, 2009) have been presented on visual tasks (Eslinger et al., 2012; Kavukcuoglu et al., 2010; Krizhevsky et al., 2012; Le et al., 2012) and on a number of machine learning problems (Collobert and Weston, 2008; Larochelle et al., 2007; Lee et al., 2009b; Salakhutdinov and Hinton, 2007).

**A Multi-layer Retina Model**

As described in the Introduction the retina possesses a multi-layer structure where each layer contains different cell types with specific functions. This distinctive structure can be captured best through a model which exhibits a similar deep architecture and utilises multiple abstraction levels. Deep belief networks are highly suitable for extracting successive layers of representation, where features on each consecutive layer are of increasing complexity. We, therefore, promote the use of multi-layer deep networks for retina modelling purposes owing to their ability to extract a hierarchy of distinctive features from data and provide the required flexibility for modelling in an uncertain environment.

Our experiments investigate a deep belief network based model of early visual processing incorporating both the outer and inner plexiform layers of the retina. The weights of the network are learnt using the training algorithm of Hinton et al. (2006).

**Simulated Environment**

The retinal network implements the first stages of object recognition and visual information processing in general, by performing a variety of image enhancement and feature extraction routines. As discussed before, the majority of ganglion cells with their centre-surround receptive fields implement edge detection mechanisms similar to DoG filters. Here we show that deep networks, even when trained fully unsupervised on a simulated photoreceptor input succeed in learning DoG type feature detectors exhibiting centre-surround receptive fields. Examples of learnt features are shown in figure 4.

We model the input that reaches a tiny area of the retina as circular spots of different size and colour on a uniform background. To approximate different wavelength light input, our dataset contains images with various background colours showing circular spots which can overlap each other. We use RGB images in order to keep similarity with the photoreceptor input of trichromats. The images can be categorised into positive and negative classes.
based on whether they contain a circle centred at the middle of the image. Examples are shown in the first rows of figure 3a and 3b. Our network model is first trained unsupervised to extract key features from this data. These features are subsequently utilised during the supervised training phase for the classification of images into two classes. This blob detection task mimics how ganglion cells learn to signal when differing light patterns reach the centre and surround of their receptive fields. We show in order to solve this task the network develops feature detectors similar to those implemented in the earliest stages of visual processing, including analogues of retinal ganglion cells with centre-surround receptive fields.

Figure 1b provides a schematic diagram of our DBN model trained on simulated photoreceptor input. As opposed to camera generated natural images or electrophysiological data from experiments, this simulated input and the corresponding class labels can be obtained with no cost and provides the advantage of having good control over the quality of data.

**Training Deep Belief Networks**

The key steps of the deep belief network training algorithm correspond to the unsupervised pretraining phase whereby the multi-layer representation is learnt one layer at a time using an RBM on each layer, followed by fine-tuning using backpropagation or a variant of the “wakesleep” algorithm (Hinton et al., 2006). The latter is illustrated in figure 1b. The restricted Boltzmann machine probabilistic graphical model is based on the Boltzmann machine model which contains 2-layers of nodes: visible and hidden nodes and has no constraints on which nodes can be connected. RBMs on the other hand do not contain links between nodes on the same layer, hence the “restriction”. They are therefore bi-partite graphs and are significantly quicker to train due to the conditional independence of hidden nodes given the visible nodes, and vice versa. In our visual recognition task visible nodes correspond to image coordinates and hidden nodes represent image features. See figure 1b for illustration.

The probability of a configuration (state) of the visible and hidden nodes $(v, h)$ can be calculated from the energy function of the RBM, which takes the form:

$$E(v, h) = -a^T v - b^T h - h^T W v,$$

(1)

where $W$ is the weight matrix describing the connections between visible and hidden nodes, while $a$ and $b$ are the biases of the visible and hidden nodes respectively. The probability of a configuration is then given by:

$$p(v, h) = \frac{e^{-E(v, h)}}{\sum_{h'} e^{-E(v, h')}}.$$

(2)

During the training phase, the probability of a given training example can be increased (the energy reduced) by altering the weights and biases. The following learning rule can be applied to maximise the log probability of the training data:

$$\Delta w_{ij} = \epsilon (v_i h_j > \text{data} - v_i h_j > \text{model}),$$

(3)

where $\epsilon$ is the learning rate and $<.>_{\phi}$ is used to denote expectations under the distribution $\phi$.

Due to the conditional independence properties, sampling from $< v_i h_j >_{\text{data}}$ is easy. In the case of RBMs with binary nodes, the probability of a hidden node $h_j$ being 1 given a randomly chosen training image $v$ is:

$$p(h_j = 1 | v) = \sigma(b_j + \sum_i v_i w_{ij}),$$

(4)

where $\sigma(x) = \frac{1}{1 + e^{-x}}$ is the logistic sigmoid function. An example of an unbiased sample is then given by $v_i h_j$. Sampling for the visible nodes is similarly easy: the probability of a visible node $v_i$ being 1 given the states of the hidden nodes is:

$$p(v_i = 1 | h) = \sigma(a_i + \sum_j h_j w_{ij}).$$

(5)

On the other hand, sampling from $< v_i h_j >_{\text{model}}$ is difficult and therefore approximations are normally applied. An efficient training method for RBMs which only broadly approximates the gradient of the log probability of the training data, is the single-step version (CD) of the Contrastive Divergence (Hinton, 2002) algorithm (CD). Each step of the algorithm corresponds to one step of alternating Gibbs sampling, where the states of the visible nodes are first set to a training example. Based on the states of the visible variables, binary states for the hidden nodes are sampled according to equation 4. This configuration of the hidden variables is subsequently used in the reconstruction phase where states for the visible nodes are sampled according to equation 5. The learning rule for the weights when using CD is then given by:

$$\Delta w_{ij} = \epsilon (v_i h_j > \text{data} - v_i h_j > \text{reconst})$$

(6)

and a similar rule can be applied for learning the biases.

In order to obtain an improved model, the sampling stage in each step can be continued for more iterations resulting in the general form of the CD algorithm: CD$_n$, where $n$ is the number of alternating Gibbs sampling iterations. RBMs and stacked RBMs (DBNs) trained efficiently using CD are very powerful tools for learning generative models of visual or other types of complex data. If the data is continuous valued, such as in our case, visible nodes other than binary can provide better models. With this in mind we construct our networks using Gaussian visible nodes which are suitable for modelling our data. In the case of Gaussian visible nodes the energy function becomes:

$$E(v, h) = \sum_i \frac{(v_i - a_i)^2}{2\sigma_i^2} - \sum_j b_j h_j - \sum_{i,j} \frac{v_i}{\sigma_i} h_j w_{ij},$$

(7)

where $\sigma_i$ is the standard deviation corresponding to visible node $i$. For more details on training DBNs see (Hinton, 2012).
Figure 3: Figures illustrating how randomly selected examples of the test set are reconstructed by a deep network with 5 hidden layers. (a) and (b) show the reconstruction of positive (images with circle centred in the middle) and negative examples (images without a circle centred in the middle) respectively. The first row in both (a) and (b) corresponds to the test examples while the second row shows their reconstructions.

EXPERIMENTS
As outlined in the previous section, we model the earliest stages of visual processing using DBNs and test our model on a circular spot detection task using simulated photoreceptor input.

Dataset
Our dataset is divided into a training and a test set, containing 10 000 and 5 000 64x64x3 RGB images respectively. Ten different background colours can be found in the training set and five in the test set. The background colours of the training and the test set are different in order to ensure substantial dissimilarity between training and test samples. Examples are shown in the first rows of figure 3a and 3b.

Training
All experiments were performed with a variety of different settings, including different numbers of hidden nodes per layer and different learning rate values. From these settings only the top performing ones are discussed here. On each layer an RBM with Gaussian visible nodes and binary hidden nodes was used for pretraining. The pretraining consisted of 10 000 epochs and in each epoch all examples of the training data was used for updating the RBM parameters. Subsequently, for the classification task 200 epochs of backpropagation was used for fine-tuning. Hidden node numbers of 100, 500 and 2 000 was examined and the network depth was ranging between 1 to 5 hidden layers.

Testing
We investigate the effectiveness of the unsupervised training phase by visualising the features learnt by RBMs on each layer of the network. As the weights of the first layer hidden nodes correspond to image locations, the visualisation of first layer features can be obtained by displaying the weight vector of the hidden node in the shape of the original image. Visualising the consecutive layers is not straightforward due to the non-linearities and only approximate features can be shown. We chose to show the common visualisation method of higher level features obtained by linear combination of features from the previous layers which makes it easy to see receptive field areas.

We also examine reconstructions of test examples calculated by the generative model. Reconstructions are obtained by feeding in an example to the network and subsequently calculating the top-down activations.

After the pretraining phase the network is trained by backpropagation to classify the input into 2 classes based on the existence of a circle in the centre of the image. Testing of the classifier is conducted on the test set. The change in performance of the classifier during the epochs of the backpropagation is measured by calculating the precision $P$, recall $R$ and F-measure scores $F = 2PR/(P + R)$.

EVALUATION
Reconstruction
Positive and negative class samples of the test set are shown in figure 3 together with their reconstructions generated by a 5 hidden layer deep network model after the unsupervised training phase. As can be seen the DBN generative model learns to encode the data with a limited
Figure 5: Figure showing how features from a lower layer are combined to form features on the subsequent layer. In (a) and (b) the first image in each row shows the feature detector learnt by a randomly selected node on the higher layer. The consecutive images in the row represent features from the lower layer which have the highest corresponding weights to the higher layer feature. These features are sorted according to the strength of their connections, decreasing from left to right. The corresponding weight is shown above each image. The visualisation of high level features is given by a linear combination of lower level features with their corresponding weights. (a) shows how second layer features are built up of first layer features, while (b) illustrates how third layer features are composed using second layer features. Note the RBMs often group similar lower level features to form a higher level feature.

Figure 6: Visualisation of the output layer and its connections to the highest level features in a network with 3 hidden layers trained using layer-by-layer pretraining followed by backpropagation. The first image in each row corresponds to an output layer node. The consecutive images in the row, similarly to figure 5, show features from the previous layer that have the strongest connections to the given output node. The corresponding weight is shown above each image. The first row visualises the strongest features and their weights for the positive class label (images with circle centred in the middle), while the second row corresponds to the negative class label.

number of hidden layers. These measures are plotted against the number of backpropagation epochs. One can see that the two networks trained using backpropagation without pretraining perform much worse than the pretrained networks. We noticed the features learnt by networks without pretraining were indistinct and noisy. The results also show multi-layered (2-4 hidden layers) networks perform superior to shallow networks, with 4 hidden layers being the best.

When trained in an adequate manner, each new layer of a DBN improves the generative model (Hinton et al., 2006). The strength of the multi-layered generative models is revealed when examining the graphs in figure 7: deep networks with at least 2 hidden layers can achieve high accuracy after only a few backpropagation epochs. This shows the unsupervised pretraining phase initialises the weights of the network to a favourable range and therefore less epochs of backpropagation is enough to guarantee good classification performance.

Classifiers
The precision, recall and the F-measure values achieved on the test set is shown in figure 7, for different numbers of hidden layers. These measures are plotted against the number of backpropagation epochs. One can see that the two networks trained using backpropagation without pretraining perform much worse than the pretrained networks. We noticed the features learnt by networks without pretraining were indistinct and noisy. The results also show multi-layered (2-4 hidden layers) networks perform superior to shallow networks, with 4 hidden layers being the best.

When trained in an adequate manner, each new layer of a DBN improves the generative model (Hinton et al., 2006). The strength of the multi-layered generative models is revealed when examining the graphs in figure 7: deep networks with at least 2 hidden layers can achieve high accuracy after only a few backpropagation epochs. This shows the unsupervised pretraining phase initialises the weights of the network to a favourable range and therefore less epochs of backpropagation is enough to guarantee good classification performance.

Features
Figure 4 shows randomly sampled features learnt by RBMs on each layer of a 5 hidden layer network. Figure 5 and 6 show features after backpropagation in a 3 hidden layer network. For each randomly chosen higher level feature, those features from the previous layer are displayed which have the strongest connections to the higher level feature. The feature visualisation reveals the network succeeded in learning DoG detectors and a number of features typical to early visual processing in the retina and visual cortex. Similarity to ON/OFF ganglion cells with centre-surround receptive fields are clearly noticeable in many of the learnt feature detectors.
This result could be anticipated as our input data and classification task of detecting spots in front of different coloured background, mimics functionality implemented in the early visual system. However, it is not obvious a DoG filter should emerge, as circles could be detected using different types of features. The fact that the network has learnt DoG filters in order to solve this task is therefore an important discovery.

Although the DBNs have been shown to have the ability to implement similar functionality to retinal ganglion cells, the underlying mechanisms, such as the network structure, is likely to be different. This is due to the fact that specific connections between retinal cells are not hard-coded into the algorithm, but are learnt in a primarily unsupervised fashion.

CONCLUSIONS

We have proposed a flexible probabilistic model of the retina and early stages of visual processing based on the state-of-the-art deep belief networks. We highlighted the resemblance between the inherently multi-layered retinal network and the deep network model. We trained our model on simulated photoreceptor input and evaluated the performance on a spot detection task, resembling functionality implemented in the early visual system. Supervised training of the networks with backpropagation was preceded by an unsupervised pretraining routine using RBMs. The multi-layer models achieved good classification results and, among other features of early visual processing, the networks learnt DoG feature detectors with centre-surround receptive fields resembling retinal ganglion cells.

In future work we will experiment with the deep network retina model using different types of input, e.g. natural image data, and different tasks corresponding to retinal ganglion cell and early visual processing functionalities. Furthermore, we will incorporate our retinal feature detectors into a convolutional network framework.

REFERENCES


**AUTHOR BIOGRAPHIES**

**DIANA TURCSANY** is a PhD student at The School of Computer Science at The University of Nottingham since 2012. She obtained her MSc degrees from Eötvös Loránd University, Hungary in 2011 and from Cranfield University, UK in 2010. Her research interests lie within the areas of machine learning, neural computation and computer vision. Her current research focuses on developing computational models of the retina. Her email is dxt@cs.nott.ac.uk and her webpage is at http://www.cs.nott.ac.uk/~dxt.

**ANDRZEJ BARGIELA** is a Professor at The University of Nottingham. Since 1978 he has pursued research on processing of uncertainty in the context of modelling and simulation of physical and engineering systems. His current research centres around computational intelligence and involves mathematical modelling, information abstraction, parallel computing, artificial intelligence, fuzzy sets and neurocomputing. His email is abb@cs.nott.ac.uk and his webpage is at http://www.bargiela.com/.

**TOMAS MAUL** is an Associate Professor at The University of Nottingham Malaysia Campus. Prior to that he worked for two years at MIMOS Bhd. as a Senior Researcher in the fields of pattern recognition and computer vision. He obtained his PhD from the University of Malaya in Computational Neuroscience. Currently, he conducts research in the areas of neural computation, optimisation and computer vision. His email is Tomas.Maul@nottingham.edu.my and his webpage can be found at http://kcztm.jupiter.nottingham.edu.my.