Computational Modelling of a Retinal Ganglion Cells' Receptive Field using Machine Learning Approaches

Computational Modelling of a Retinal Ganglion Cells' Receptive Field using Machine Learning Approaches

Gautham P. Das[a], Philip J. Vance[b,*], Dermot Kerr[b], Sonya A. Coleman[b],
Thomas M. McGinnity[c], Jian K. Liu[d]

[a] *Department of Mechanical Engineering, Amrita School of Engineering, Amritapuri, Amrita Vishwa Vidyapeetham, Amrita University. email: pdasgautham@gmail.com*
[b] *Intelligent Systems Research Centre, University of Ulster, Magee Campus, Northern Ireland. e-mail: {p.vance, d.kerr, sa.coleman}@ulster.ac.uk*
[c] *College of Science and Technology, Nottingham Trent University, UK. e-mail: martin.mcginnity@ntu.ac.uk*
[d] *Centre for Systems Neuroscience, Department of Neuroscience, Psychology and Behaviour, University of Leicester, Leicester, UK. e-mail: jian.liu@leicester.ac.uk*

Abstract

Artificial vision using computational models that can mimic biological vision is an area of ongoing research. One of the main themes within this research is the study of the retina and in particular, retinal ganglion cells which are responsible for encoding the visual stimuli. A common approach to modelling the internal processes of retinal ganglion cells is the use of a linear - non-linear cascade model, which models the cell's response using a linear filter followed by a static non-linearity. However, the resulting model is generally restrictive as it is often a poor estimator of the neuron's response. In this paper we present an alternative to the linear - non-linear model by modelling retinal ganglion cells using a number of machine learning techniques which have a proven track record for learning complex non-linearities in many different domains. A comparison of the model predicted spike rate shows that the machine learning models perform better than the standard linear - non-linear approach in the case of temporal white noise stimuli.

Keywords: Artificial Vision, Biological Vision, Machine Learning, Retinal Ganglion Cell, Receptive Field

[*] Corresponding Author

1. Introduction

Biological vision systems have been rigorously researched for more than a century with the early studies of the retina, published by Cajal [1] in 1892, considered to be a milestone in the birth of the field of neuroscience [2]. Some decades later it was discovered that the retina, which is located at the back of the eye, is responsible for converting observed visual stimuli into a form of temporal coding [3], known as spikes (or action potentials). These spikes are then propagated via the optic nerve to the visual cortex for higher level processing. Most of the major cell types in the vertebrate retina have been identified [4, 5] although the exact role of many of these cells and the specific encoding scheme used to transmit information remains unclear [5]. Development of artificial vision systems using computational models of the retina, particularly of retinal ganglion cells (RGCs) is an area of ongoing research. There is a number of difficulties associated with this modelling task, mainly related to three particular aspects; firstly, cells within the retina are grouped into five distinct layers (see Figure 1), with two of these layers responsible for complex lateral interconnections between the cell types (Horizontal and Amacrine cells); secondly, many of these cells have complex non-linearities and adaptation mechanisms that are yet to be understood; thirdly, the specific encoding scheme used to transmit the information is not fully understood. Due to these factors, the exact information abstraction that takes place in progressing from the absorption of photons in the photoreceptors to the encoding of spikes in the ganglion cells is still not fully understood [5, 6], and thus modelling RGCs is a challenging task.

An early approach to developing computational models of the retina was based on non-linear system identification techniques which are often used as way of reverse engineering complex systems [8]. Previously they has been used to understand the responses of auditory neurons in [8], retinal ganglion cells in [9] and in [10], the Wiener theory of non-linear system identification [11] was applied to study the underlying functional relationship between a cell membrane potential and the resulting spiking response from RGCs of a catfish. Following this study, the Wiener-Volterra method [12] has been extensively used to model non-linear biological systems [13, 14, 15, 16, 17, 18, 19]; however a major drawback of the Wiener-Volterra approach is the geometrically increasing computational complexity with the kernel order [20]. Moreover, the interpretation of higher order kernels is difficult.

Various block-structured [21] or modular models in the form of cascaded

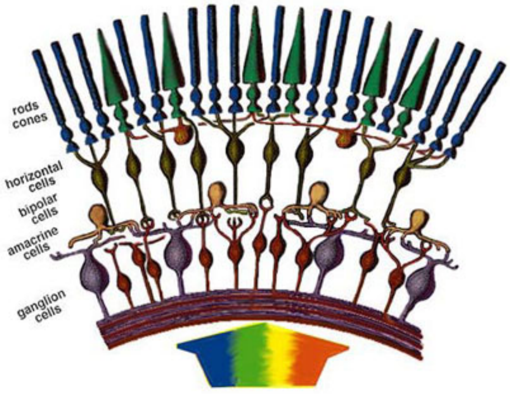

Figure 1: Diagram showing the various cells and structure of the retina [7]

or parallel configurations have been used to overcome the limitations of Wiener-Volterra models. Cascaded modular models may take various forms such as linear - non-linear (LN) [22], non-linear - linear (NL) [23], linear - non-linear - linear (LNL) [14], etc. The LN model usually consists of a linear filter module cascaded with a non-linear transformation module. Obtaining the model parameters of an LN model with a single linear filter, the shape of the filter and the static non-linearity can easily be achieved by a reverse correlation analysis with a Gaussian white noise stimulus [24], making the LN method a widely used approach for modelling RGCs. An important characteristic observed from this modelling process is that many models illustrate that there is significant non-linear processing taking place within a RGC and its connected cells.

An alternative to system identification and cascaded modular approaches is to use a machine learning based nonparametric regression algorithm. Previous research has shown that many machine learning algorithms can successfully learn the underlying complex non-linearities in the data while generating an opaque model of the system in other domains [25, 26, 27]. Artificial neural networks are examples of such regression algorithms used to model biological aspects of vision systems. For example [28] used a multi-layer feed-forward neural network with the backpropagation training algorithm to model the non-linear responses of individual neurons in the visual cortex of a cat generated by spatio-temporal random-bar stimuli. Similarly, [29] used a multi-layer feed-forward neural network to model the responses of neurons in the visual cortex of a cynomolgus monkey using various complex artificial visual stimuli. Responses of neurons in the primary visual cortex of a macaque monkey to natural image stimuli were modelled using a multi-layer feed-forward neural network in [30] and using a Self-Organising Fuzzy Neural Network (SOFNN), the behaviour of a salamander retinal ganglion cell was accurately modelled in [27].

In this work, the relative performance of different machine learning techniques in modelling RGCs is explored. Specifically, multilayer perceptron (MLP) network, Bayesian regularised neural network (BRNN), non-linear auto-regressive model with exogenous inputs (NARX) network, support vector regression (SVR) and k-nearest neighbour (kNN) regression are used to model the RGCs. The performance of the developed machine learning models are compared with models developed using the standard LN technique [24].

The outline of the paper is as follows. Section 2 introduces the different computational modelling techniques used. Details of the electro-physiological experiments and the different visual stimuli used in these experiments are given in Section 3. Section 4 presents the results from the modelling experiments along with a performance comparison of the developed models based on these results with Section 5 discussing the results from the modelling experiments and exploring future work.

2. Methods

2.1. Linear - non-linear(LN)

The standard approach for modelling an RGC is a cascaded linear - non-linear (LN) technique [24]. This method, as outlined in Figure 2, separates the model estimation into a cascaded linear filter and a non-linear function. Initially, the input stimuli to the model is convolved with the linear filter to get a linear estimate of the neurons response. The non-linear transformation maps the linear estimate to a spike rate estimate.

The LN model, as depicted by Figure 2, involves only one linear filter, although multiple linear filters are possible [31]. The linear filter is computed by examining the spike triggered average (STA) of the RGCs response to stimulus over a specific time window. It is defined as the average stimulus preceding each spike, otherwise explained as the average stimulus which contributed to the occurrence of a spike. This can be mathematically represented as:

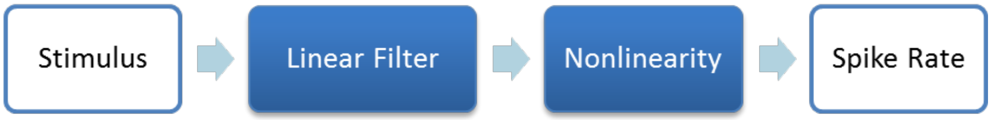

Figure 2: Block diagram depicting the LN technique with single linear filter: the stimulus is first passed through a linear filter to get a linear estimate, on which a static non-linear transformation is applied to get the estimated spike rate

$$STA = \frac{\sum_{t=\tau}^{T} x_t f_t}{\sum_{t=\tau}^{T} f_t} \tag{1}$$

where T is the total time period in which spikes are recorded, x_t is the sequence of stimulus intensity values within a temporal window of size τ preceding a spike, and f_t is the number of spikes recorded at time t. The size of the temporal window is determined by examining the duration of the average response and ascertaining when the stimulus intensity converges to zero [24]. Illustrations of STAs are shown in Figure 3 where three example STAs have been plotted for both an ON-type cell (Figure 3.a) and OFF-type cell (Figure 3.b) which show the inherent variability across cells.

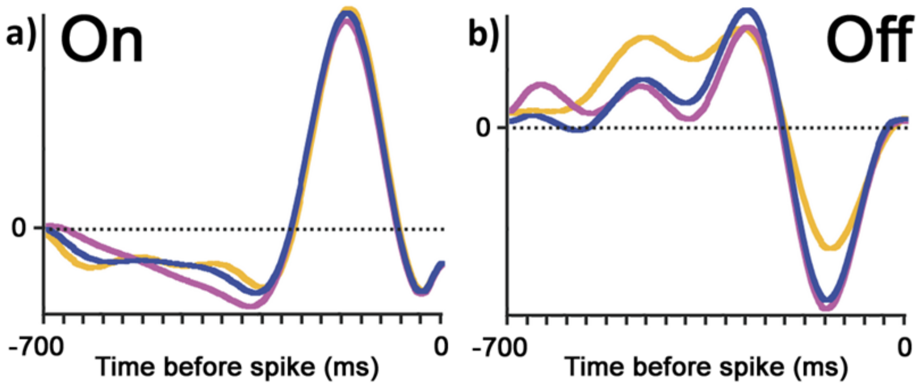

Figure 3: Illustration of computed STAs for a salamander RGC; a) ON-type cell and b) OFF-type cell.

In this work, τ was identified as 700ms which, when binned in equal time bins of $33\frac{1}{3}$ms (the stimulus update rate), equates to 21 time bins (samples).

After computing the linear filter, the next stage is the estimation of the non-linear function. By examining the average spike count for equal or nearly equal stimulus values [24] and plotting for each time point for the stimulus presentation, a function may be fitted that best represents the non-linearity. For RGCs, the static non-linearity is generally fitted well by a parameterised form of cumulative normal density function [24], though other solutions may also apply. It can be denoted by:

$$N(x) \approx \alpha C(\beta x + y) \tag{2}$$

where C is the cumulative normal density function, x is the linear estimate, α is the maximum firing rate of the neuron, β is the sensitivity of the non-linearity to the linear estimate and y is the maintained drive to the cell that determines the spike rate in the absence of net visual stimulation. In order to fit this non-linearity, appropriate values must be determined to fit the least squared error for each of the α, β and y parameters to the data within the graph of the average spike count per stimulus value.

2.2. Machine Learning Methods

As discussed in Section 2.1, each spike can be attributed to a set of stimuli preceding the spike which, when averaged over all the spikes, is known as the STA. This information is crucial in estimating the standard LN method as this forms the linear filter, and thus these stimulus values that precede each spike are also utilised within each of the machine learning methods. Figure 4 shows an overview of a generic machine learning method where $x(t)$ is the current value of the visual stimulus whilst $x(t-1)$ to $x(t-\tau)$ are the previous values in the time series up to the size of the window τ. These values act as the inputs to the machine learning method $f(x)$ which learns to predict the output $y(t)$ on the next time step.

Artificial neural networks (ANNs) offer a desirable solution for modelling complex input-output relationships without assuming a priori function forms of models. ANNs have been previously used to model biological aspects of the vision system [28, 29, 30]. ANNs have an advantage over system identification techniques and cascaded models as they have a fast and simple implementation. However, this advantage has to be balanced against the weakness that the obtained mapping is opaque, and not easily analysed. Hence, such an opaque ANN model of the neuron's stimulus-response relationship becomes less useful for understanding the underlying neuronal architecture and structure. Even so, we use such techniques to model neuronal behaviour with

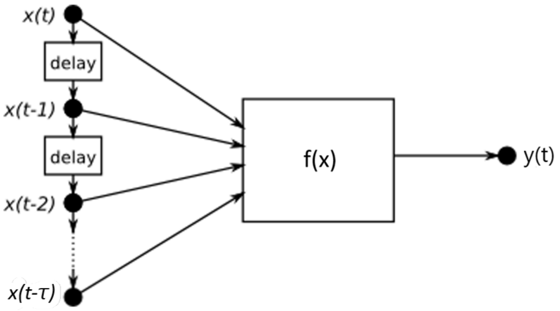

Figure 4: Overview of a generic machine learning method.

artificial visual scenes, and to represent the mapping between stimulus and response due to their fast and simple implementation. Different neural network techniques have been investigated in this work and successfully applied to generate models of RGCs.

Multi-Layer Perceptron (MLP)

A multi-layer perceptron (MLP) is a feed-forward neural network (FFNN) model that maps sets of input data onto a set of appropriate outputs and is the most commonly used neural network in prediction tasks [26], [32]. An MLP consists of multiple layers of nodes in a directed graph, with each layer fully connected to the next one. The MLP commonly utilises a supervised learning technique called backpropagation for training the network. Except for the input nodes, each node is a neuron (or processing element) with a activation function. Mathematically the MLP relates the output $y(t)$ to the input $x(t)$ by:

$$y(t) = w_0 + \sum_{j=1}^{\sigma} w_j g\Big(w_{0,j} + \sum_{i=1}^{\tau} w_{i,j} x_i(t)\Big) \tag{3}$$

where τ and σ are the number of input nodes and hidden nodes respectively [33]. The weights w_j and w_i are referred to as connection weights, and represent the tuneable parameters of the network. In this work, the activation function g of the neurons in MLP is represented with a sigmoid function.

Bayesian Regularised Neural Network (BRNN)

A Bayesian regularised neural network (BRNN) is an extension of the common MLP. A potential issue in an MLP is the high probability of overfitting and overtraining which leads to poor generalisation of the network. In a BRNN, in order to reduce the potential overfitting, a mathematical technique known as Bayesian regularisation is incorporated into the training process. Generally, the goal of each training step of an MLP is to minimise the sum of squared errors between the model output and the target output. Bayesian regularisation modifies this to:

$$F(\alpha, \beta) = \beta E_D + \alpha E_w \tag{4}$$

where F is the objective function, E_D is the sum of squared errors, E_w is the sum of square of network weights, and α and β are objective function parameters [34]. In a BRNN the network weights are considered to be random variables and thus their density function is written according to the Bayes' rule as:

$$P(w|D, \alpha, \beta, M) = \frac{P(D|w, \beta, M)P(w|\alpha, M)}{P(D|\alpha, \beta, M)} = \frac{1}{Z_{F(\alpha,\beta)}} exp(-F(\alpha, \beta)) \tag{5}$$

where w is the vector of network weights, D represents the data vector, M is the neural network model being used and $Z_{F(\alpha,\beta)}$ is a normalisation factor as described by [35].

In Bayesian regularized networks, overly complex models are penalised, as unnecessary linkage weights are effectively driven to zero. The network will calculate and train on the nontrivial weights, also known as the effective number of parameters, which will converge to a constant as the network grows [36].

NARX

The non-linear auto-regressive model with exogenous inputs (NARX) is a non-linear model estimation technique, which models a dynamic non-linear system that expresses the present value of the output as a non-linear combination of previous and present values of the input and previous values of the output. This model can be represented mathematically as:

$$\begin{aligned} y(t+1) = f[&y(t), .., y(t-\tau_y+1), x(t-k), \\ &x(t-k+1), ..., x(t-\tau_x-k+1)], \end{aligned} \tag{6}$$

where f is the non-linear mapping function, $x(t)$ and $y(t)$ respectively denote the inputs and outputs of the model at time t, while τ_x and τ_y are the order of input and output memories respectively and the parameter k is a delay term [37, 38]. The function f is a non-linear function, such as a neural network as used here or a polynomial or other network. In a NARX neural network, the mapping function f is approximated using a standard MLP. Thus a NARX network can be considered as a combination of a time delay neural network (TDNN) and a recurrent neural network (RNN).

Support Vector Regression (SVR)

Support vector regression (SVR) [39] is an extension of the popular support vector machine (SVM) classifier to regression problems. The SVR transforms the original training data to a higher dimensional feature space F, through some non-linear transformation function ϕ. A linear regression in the higher dimension space F corresponds to a non-linear regression in the low dimensional input space $\mathbb{R}^n$. This is expressed mathematically as:

$$\phi : \mathbb{R}^n \rightarrow F, w \in F, \tag{7}$$

$$f(x) = \langle w, \phi(x) \rangle + b, \tag{8}$$

where the dot product operator is denoted by $\langle ., . \rangle$, w represents the flatness of function f and b is a threshold. During the training process, a function $f(x)$ that results in a maximum deviation from the obtained targets y_i and minimum value of w simultaneously is found. This optimisation problem can be written as:

$$\textit{minimise} \qquad \frac{1}{2}||w||^2 + C \sum_{i=1}^{l} \xi_i + \xi_i^* \tag{9}$$

$$\textit{subject to} \qquad \begin{cases} y_i - \langle w, x_i \rangle - b \leq \epsilon + \xi_i, \\ \langle w, x_i \rangle + b - y_i \leq \epsilon + \xi_i^*, \\ \qquad\qquad \xi_i, \xi_i^* \geq 0, \end{cases} \tag{10}$$

where C represents the trade-off between flatness and the amount up to which deviations larger than ϵ are tolerated. The slack variables (ξ_i, ξ_i^*) are introduced to make the optimisation feasible. The dual of this problem is easily solved using saddle point conditions via a kernel function $K(x_i, x)$. The solution of the dual problem is given in [40] as:

$$w = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i \tag{11}$$

and

$$f(x) = \sum_{i=1}^{l} (\alpha_i - \alpha_i^*) x_i K(x_i, x) + b \tag{12}$$

where $(\alpha_i, \alpha_i^* \geq 0)$ are Lagrange multipliers. Using this kernel trick enables us to avoid explicitly calculating the original transformation function ϕ. In our experiments a radial basis kernel is used.

k-Nearest Neighbour (kNN)

The k-nearest neighbours algorithm (kNN) [41] is a commonly used algorithm for classification and regression based on a similarity measure (e.g. distance functions) in a feature space. The kNN algorithm learns through the localisation information of the training objects. In classification applications, the kNN algorithm classifies an input object into a class based on a majority vote by its k closest neighbours. When used for regression, the kNN algorithm predicts the value of the input, as the average of its nearest k neighbours. In our implementation which is based on regression, the value of k was determined using five-fold cross validation with the Euclidean distance as the distance measure.

3. Experimental Setup

In the work presented here we develop a number of computational ganglion cell models using the different techniques discussed above. Neuronal recordings obtained from isolated retinas are used to obtain the models. During the physiological experiments retinas from dark-adapted adult axolotl salamander (Ambystoma mexicanum) [42] were isolated and placed ganglion-cell-side-down on a planar multi-electrode array and submersed in a chemical solution to prolong extracellular recordings. The isolated retina was visually stimulated by projecting the screen of a gamma-corrected miniature organic light-emitting diode (OLED) monitor onto the photoreceptors. The stimulus was updated at a rate of 30Hz, meaning a new visual stimulus pattern was presented approximately every 33.33ms. Action potentials were recorded from the RGCs using a multi-electrode array and sampled at a frequency of 10kHz with spike times measured with respect to the beginning of the stimulus presentation.

3.1. Visual Stimulus

The visual stimuli used in these experiments were generated using Gaussian white noise which is stochastic and highly interleaved, and comprised of a wide range of visual inputs. Two types of artificial non-repeated visual stimulus sequences were generated. In the first sequence, new intensity values were drawn randomly from a Gaussian distribution at a rate of 30Hz and all pixels in the field were illuminated with that light intensity. This temporal stimulus sequence (Figure 5.a) is referred to as full-field flicker (FFF) in the remainder of this paper. In the second stimulus sequence, stimulus intensities were Gaussian distributed in both space and time and resulted in a sequence of binary checker board patterns. This spatio-temporal stimulus sequence (Figure 5.b) is referred to as checker-board flicker (CBF) in the remainder of this paper. The CBF has increased complexity compared with the FFF because of the spatial component.

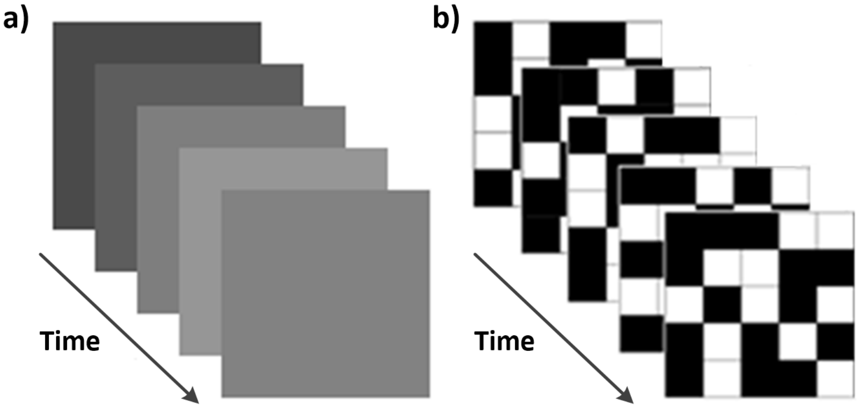

Figure 5: Gaussian white noise visual stimulus sequences: (a) full-field flicker (FFF) (b) checker-board flicker (CBF)

3.2. Data Pre-processing

The main aim of the data pre-processing stage is to manipulate the data to obtain a regression dataset, i.e., input-output dataset corresponding to the visual stimulus – neural response. The regression dataset can then be used to develop and evaluate computational models of the RGCs.

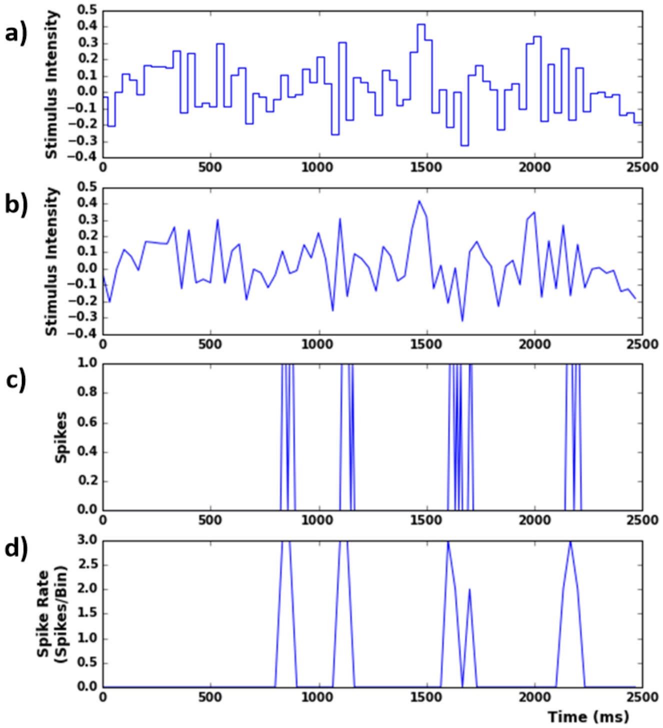

Figure 6: Data pre-processing of FFF dataset: (a) stimulus intensity, (b) sampled stimulus intensity, (c) recorded neuronal spikes and (d) spikes binned at 30Hz

The data from the electro-physiological experiments with FFF stimuli included 54000 samples of stimulus and spike recordings for a timespan of nearly 30 minutes. Each sample corresponds to a new full-field stimulus, initially recorded at 10 kHz and binned at 30Hz (i.e., every 33.33ms) to correspond to the update rate of the visual stimulus. As the stimulus has uniform intensity across all pixels, this intensity value was chosen to be the model input. Using this method, a continuous valued single input – single output dataset was obtained. Figure 6.a - Figure 6.b show examples of the original and sampled sequences of the stimulus intensities, and Figure 6.c

- Figure 6.d show the corresponding original and binned sequences of the neural spike recordings from an RGC for a 2500ms presentation of stimuli. Among the total number of samples used, 80% (43,200 samples) were used for constructing a training dataset and the remaining 20% (10,800 samples) of unseen data were used for constructing a testing dataset.

When using CBF stimuli, it is important to identify the stimulus values that contribute to the RGC eliciting a spike i.e. identify only those stimulus values that fall within the cells RF and contribute to that cell eliciting a spike rather than using all stimulus values. To determine the location and shape of the RF for each cell, reverse correlation [43] (also known as spike triggered averaging) is used; the RGC is stimulated with CBF stimuli, cell activations are recorded and subsequently used to calculate the STA. The spatial component of the RF is then isolated across time using Singular Value Decomposition and fitted with a 2D Gaussian function that approximates its size and shape [44].

Figure 7 shows the various stages of the data pre-processing after the RF has been defined for the various stimulus values to be extracted from the RF region. Pertinent CBF pixel intensities located either inside, or on the border of the cell's RF region were first identified as shown in Figure 7.b and extracted. The local stimulus for each cell was then weighted using a 2D Gaussian filter (Figure 7.c) to emulate the visual stimulus perceived by the ganglion cells [45]. The mean of the weighted pixel intensities (Figure 7.d) for each cell's RF region was calculated and used as input to the computational models. The data for the modelling experiments using CBF included 64500 samples of stimulus and spike recordings for a timespan of approximately 36 minutes. Each sample corresponds to a new checker-board stimulus, i.e. in every 33.33ms. Among these 80% (51,600 samples) was used for training and the remaining 20% (12,900 samples) was used for testing.

3.3. Overview of Modelling Experiments

In the most basic classification, a ganglion cell may be identified as an ON, OFF, OFF-ON or ON-OFF cell depending on its response to variations in light intensity [46]. Classifications which identify these types and subtypes are possible based on the shapes of the STA [47, 48] or on characteristic features detected in spike triggered covariance (STC) analysis [49]. In this work, the former case is utilised for both temporal and spatio-temporal stimuli. Four ganglion cells (Cell-A, Cell-B, Cell-C, and Cell-D), from a total of 36 cells from the same recording session, were selected as representatives

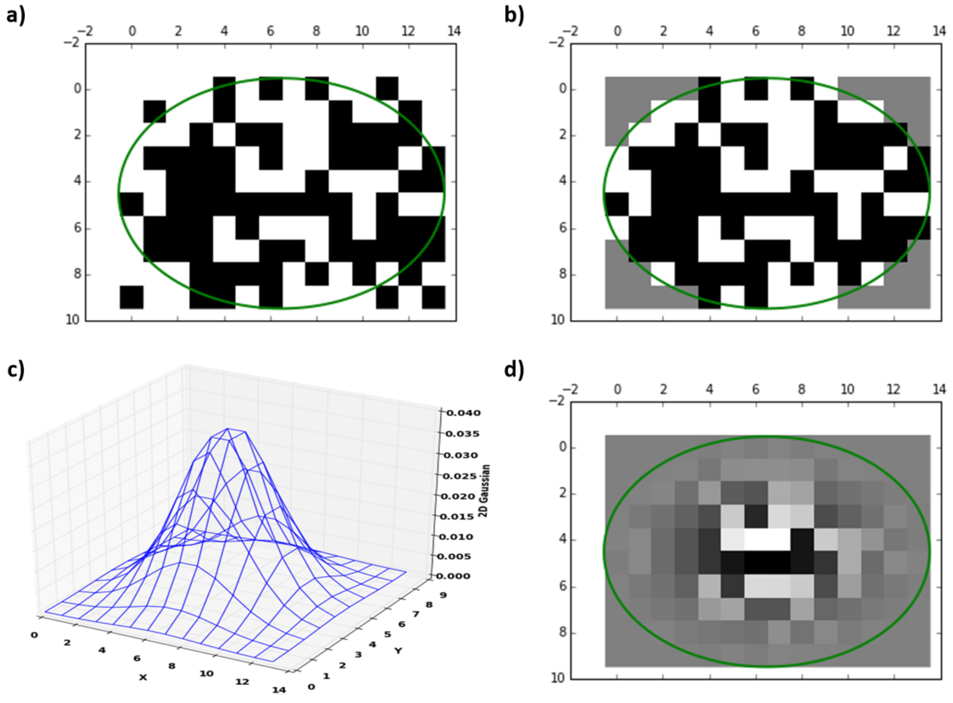

Figure 7: Pre-processing step which shows how the local stimulus pertaining to a cells receptive field is weighted with a 2D Gaussian filter: (a) binary checker-board pattern with an RGC's RF boundary marked, (b) pixel intensities within the cell's RF region are extracted, (c) example of 2D Gaussian used to weight the stimulus intensities and (d) weighted intensities of the local stimulus.

of different types of RGCs and used for the development of the computation models presented in this paper.

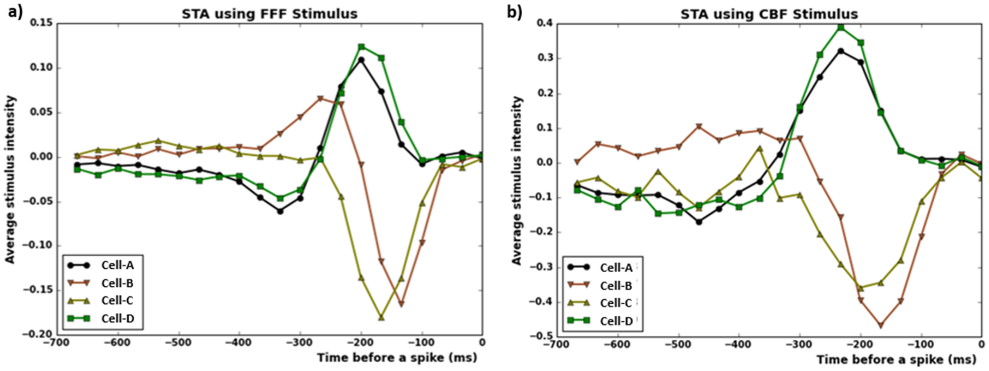

Figure 8: Spike triggered averages of the selected ganglion cells: (a) using the neural recordings with FFF stimulus and (b) using neural recordings with CBF stimulus.

Figure 8 shows the STAs of the selected RGCs obtained using the neuronal recordings with FFF and with CBF stimuli. Although alternative interpretations are possible from the shape of these STAs, here we refer to Cell-A and Cell-D as ON-type cells, and Cell-B and Cell-C as OFF-type cells. The STAs of the cells in Figure 8 have a span of 21 samples corresponding to 700ms. To utilise this information, the computational models were presented with 21 lagged values of the stimulus time series to correlate with the 21 samples in the STA. Computational models of RGCs were then developed using the LN and machine learning techniques with the input-output datasets obtained after the data pre-processing stage discussed in Section 3.2. Spike rate prediction performances of the developed models for the FFF and CBF datasets are discussed in Sections 4.1 and 4.2 respectively.

4. Results

All models were trained to predict the recorded cell's spike count, binned at the stimulus update rate. The models using the LN technique were developed with their STAs (calculated using FFF stimuli, Figure 8.a) as the linear filter and the non-linearity was approximated using a parameterised cumulative normal density function (2) between the linear estimate (output of the linear filter) and the target (i.e. actual) spike rate. An example of the non-linearity fitting between the linear estimate and the spike rate for Cell-B is shown in Figure 9 where the cell demonstrated a saturating non-linearity.

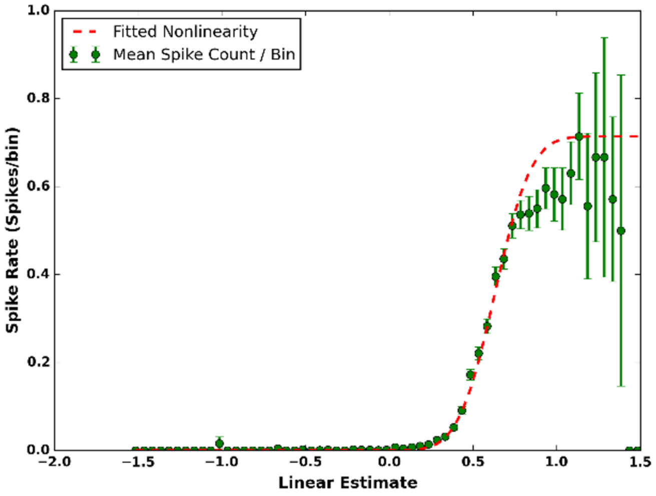

Figure 9: Non-linearity fitting between the linear estimate and the target spike count for Cell-B using FFF stimulus. The markers show the mean spike rate and the error bars show the standard deviation. The dashed line shows the non-linearity fitted with cumulative normal density function.

4.1. Modelling of RGCs stimulated with FFF

Prediction accuracy in terms of root mean square error (RMSE) between the actual cell's spike rate and the model predicted spike rate is used to measure and compare the performance of all the presented models. The RMSE values for the computational models of the ON-type cells are given in Table 1. For both the ON-type cells, the models developed using the SVR and BRNN algorithms outperformed the other techniques for training and testing respectively. In general, the SVR, BRNN and MLP algorithms outperform the standard LN approach for both the ON-type cells. The RMSE values of the models for the OFF-type cells are given in Table 2. Again, the models developed using the SVR and BRNN algorithms outperform the other techniques respectively for the training and testing samples of Cell-B, and the models developed using MLP and NARX algorithms outperform the other models for the training and testing samples of Cell-C. In general, the MLP and NARX algorithms perform better than the standard LN approach for both the OFF-type cells.

Table 1: RMSE values for ON-type cells using FFF stimuli

Method	Cell-A		Cell-D	
	Training	**Testing**	**Training**	**Testing**
LN	0.568	0.534	0.515	0.509
MLP	0.502	0.481	0.468	0.468
BRNN	0.461	**0.460**	0.439	**0.446**
NARX	0.700	0.687	0.855	0.886
SVR	**0.436**	0.526	**0.430**	0.520
kNN	0.545	0.536	0.507	0.521

Table 2: RMSE values for OFF-type cells using FFF stimuli

Method	Cell-B		Cell-C	
	Training	**Testing**	**Training**	**Testing**
LN	0.175	0.175	0.152	0.151
MLP	0.166	0.168	**0.149**	0.151
BRNN	0.159	**0.163**	0.154	0.155
NARX	0.170	0.174	0.151	**0.149**
SVR	**0.156**	0.201	0.159	0.181
kNN	0.182	0.194	0.168	0.176

Plots representing the BRNN model prediction and computed neuronal spike rate for Cell-A using a randomly selected time window of duration 30s (900 samples) are shown in Figure 10. The BRNN was selected to illustrate model fit as it was one of the best performing algorithms in terms of RMSE based on results from Table 1 and Table 2, and similarly Cell-A was selected as it elicited more spikes than the other modelled cells. The target and predicted spike rates for the training samples are shown in Figure 10.a, while the error histogram for the same period is shown in Figure 10.b. Similarly, the spike rates and error histograms for the testing samples are shown in Figure 10.c and Figure 10.d respectively. It can be seen from Figure 10.a and Figure 10.c that the BRNN model generally predicts the cell's temporal response and spike rates. The error histograms in Figure 10.b and Figure 10.d show that the majority of the prediction error falls within ±0.5 spikes/bin.

4.2. Modelling of RGCs stimulated with CBF

When using the CBF stimuli the recorded neuronal response illustrates that the ON-type cells produced more spikes (Cell-A: 3831, Cell-D: 3615) than the OFF-type cells (Cell-B: 2965, Cell-C: 1271) within the same recording period. All cells had a minimum spike rate of zero spikes/bin. Both the ON-type cells (Cell-A and Cell-D) had a maximum spike rate of five spikes per bin using CBF. Among the OFF-type cells, the maximum spike rate for Cell-B was one spike/bin and that for Cell-C was three spikes per bin.

The modelling process used for the CBF stimuli is similar to that used for the FFF stimuli except that the stimulus values within the cell's RF are extracted and averaged in the pre-processing stage (see Section 3.2). All models were trained to predict the cell's recorded spike counts, binned at the stimulus update frequency. The models developed using the LN technique were developed with their STAs (calculated using CBF stimuli, Figure 8.b) as the linear filter and the non-linearity was estimated using the cumulative normal density function (2) to transform the linear estimate (output of the linear filter) to the target spike rate. An example of the non-linearity for Cell-B is shown in Figure 11 where the cell demonstrates a threshold non-linearity.

The RMSE values comparing the models of the ON-type cells are given in Table 3 and those for the OFF-type cells are given in Table 4. In the case of the ON-type cells, the SVR and MLP algorithms perform better than the other algorithms for both training and testing respectively. Among the OFF-type cells, the SVR algorithm performs best considering the training

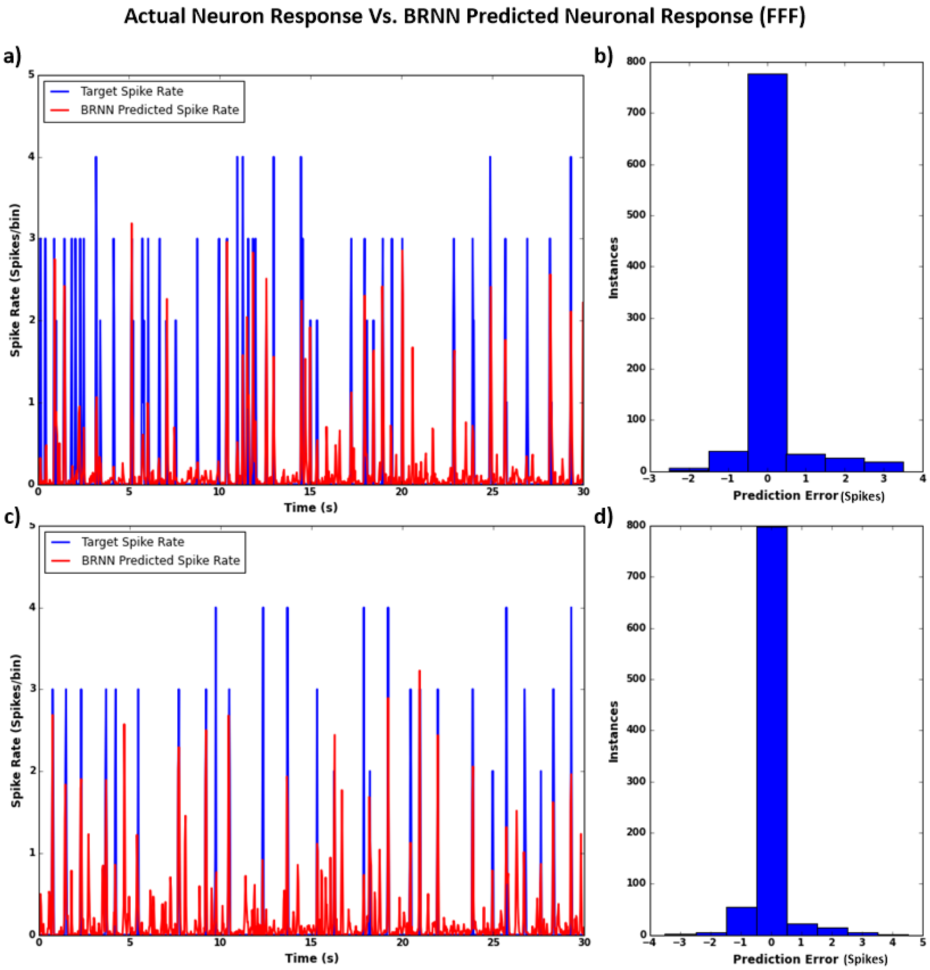

Figure 10: Computed neuronal spike rate responses of Cell-A (OFF-type cell) vs. model predicted responses of the fitted BRNN model for randomly selected 900 samples (corresponding to 30s time window) of both training and testing sets (FFF Stimulus). (a) Prediction results and (b) error histogram for the training samples. (c) Prediction results and (d) error histogram for the testing samples.

dataset and the LN, MLP, BRNN and NARX algorithms perform equally as well as the LN method for Cell-B, whilst the NARX algorithm performs better than the LN technique for Cell-C using the testing dataset.

Overall, considering the CBF stimuli, although the MLP and BRNN models perform marginally better than the LN approach across all four RGCs the performance improvement is not as apparent as that observed from the

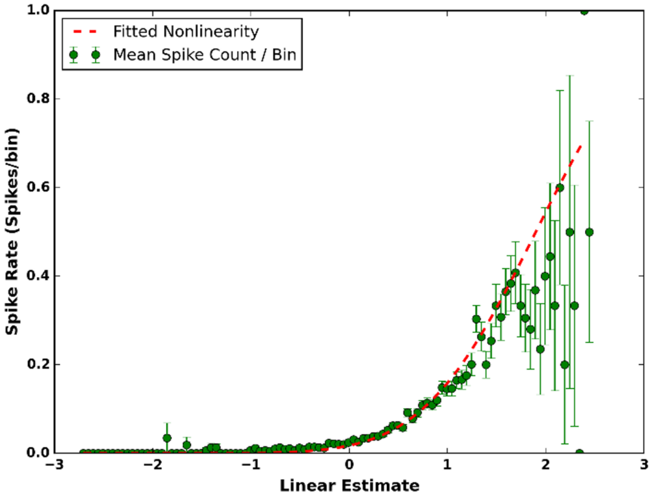

Figure 11: Non-linearity fitting between the linear estimate and the target spike count for Cell-B using CBF stimulus. The green markers show the mean spike rate and the error bars show the standard deviation. The red line shows the non-linearity fitted with cumulative normal distribution function.

Table 3: RMSE values for ON-type cells using CBF stimuli

Method	Cell-A		Cell-D	
	Training	**Testing**	**Training**	**Testing**
LN	0.331	0.336	0.416	0.376
MLP	0.329	**0.334**	0.413	**0.375**
BRNN	0.330	0.335	0.413	0.376
NARX	0.334	0.339	0.506	0.481
SVR	**0.292**	0.344	**0.377**	0.392
kNN	0.324	0.346	0.407	0.396

Table 4: RMSE values for OFF-type cells using CBF stimuli

Method	Cell-B		Cell-C	
	Training	**Testing**	**Training**	**Testing**
LN	0.203	**0.198**	0.150	0.145
MLP	0.202	**0.198**	0.146	0.144
BRNN	0.202	**0.198**	0.147	0.144
NARX	0.202	**0.198**	0.146	**0.143**
SVR	**0.173**	0.205	**0.134**	0.147
kNN	0.199	0.205	0.145	0.148

models developed for the FFF stimulus (see Table 1 and Table 2). Again, similar to the modelling results with the FFF stimulus, the difference in the range of RMSE values between the ON-type cells and OFF-type cells is due to the difference in the number of spikes elicited by the specific RGC type.

Plots representing the MLP model prediction and computed neuronal spike rate for Cell-A considering a randomly selected time window of duration 30s (900 samples) for both the training and testing samples are shown in Figure 12. The MLP was selected as it is one of the best performing algorithms for the testing samples and for visual representation of the model performance Cell-A is selected as it elicited more spikes than the other cells. Figure 10.a and Figure 10.c show both predicted and target spike rates for the training and testing samples respectively. Figure 10.b and Figure 10.d respectively show the prediction error histogram for the training and testing samples from the same period. The MLP model was able to estimate the temporal spike behaviour correctly and most of the prediction error falls within ±0.5 spikes/bin, however the actual number of spikes predicted fails to correctly match the binned spike rate for the cell in question. A comparison between Figure 10 and Figure 12 shows that the spike rate predictions are better approximated in the case of the FFF stimulus than those obtained for the CBF stimulus. One possible reason for this behaviour is the spatial variations within the CBF data. Compared with the FFF stimulus which contains uniform intensity, the data extracted from the RF region of the CBF is averaged as a representative value. This averaging process may not account for non-linear processing that occurs within subunits of the cell resulting in reduced model accuracy thereby explaining the different predicted neural response with different stimuli. Furthermore, due to the way the CBF

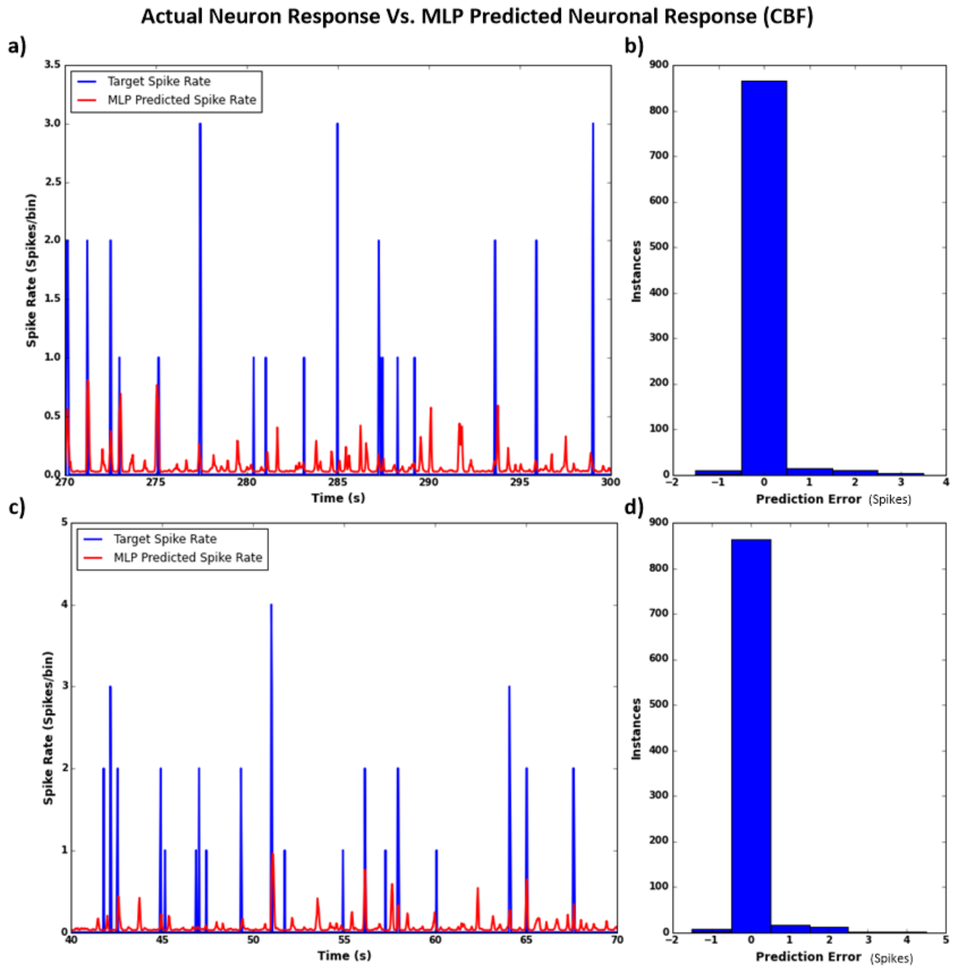

Figure 12: Computed neuronal spike rate responses of Cell-A (OFF-type cell) vs. model predicted responses of the fitted MLP model for randomly selected 900 samples (corresponding to 30s time window) of both training and testing sets (CBF Stimulus). (a) Prediction results and (b) error histogram for the training samples. (c) Prediction results and (d) error histogram for the testing samples.

data is processed, the computational models cannot learn any underlying relations within the spatial information. One approach would be to include all the information within the RF region rather than an averaged representative value. A major constraint to this however is the dimensionality of the stimulus data, which would increase exponentially and correspondingly increases the computational complexities.

5. Discussion and Future Work

Modelling biological visual systems is difficult due to insufficient knowledge about the internal components and their organisation, and the complexity of the interactions within the system. The standard LN approach derives computational models of visual neurons using physiological experimental data which are quantitatively fitted through an input-output analysis. Owing to the non-linear processing within RGCs in this paper we explore the use of machine learning techniques for modelling such cells, including the complex non-linearities. Specifically, we use a number of different machine learning based regression algorithms (MLP, BRNN, NARX, SVR and kNN) to develop computational stimulus-response models of different types of RGCs using different visual stimuli. For the purpose of performance comparison, the RGCs were also modelled using the standard LN technique. In order to observe how the model performance varies depending on the type of RGCs we present results from four ganglion cells – two ON-type cells and two OFF-type cells.

Many of the explored machine learning algorithms outperform the standard LN approach in predicting the RGCs response when stimulated with either FFF or CBF stimuli. The improved accuracy of the machine learning model's predicted spike rate when compared with the LN technique clearly demonstrated when considering the FFF stimuli. However, the machine learning models only performed on-par or marginally better than the LN technique when considering the CBF stimuli. This variation in performance was observed for both types of RGCs (ON-type cells and OFF-type cells) modelled in the experiments presented here. One possible reason for this reduction in performance when the RGCs are stimulated with CBF stimuli is that the models ignore the spatial variations present in the CBF. This is resultant of collating all pixel information with the RF region and taking the mean intensity as the input. This means that any importance based on the configuration of individual checkers within the RF is lost. To improve this result, we would need to consider some additional inputs to the model which account for spatial variance in addition to temporal variance. However, by considering all pixel intensities from the receptive field region of the image, the increased complexity would result in a huge computational processing cost thus a balance between accuracy and complexity must be sought which will be the next stages of this work.

Acknowledgement

The research leading to these results has received funding from the European Union Seventh Framework Programme [FP7-ICT-2011.9.11] under grant number [600954] ["VISUALISE"]. The experimental data contributing to this study have been supplied by the "Sensory Processing in the Retina" research group at the Department of Ophthalmology, University of Goettingen as part of the VISUALISE project.

References

[1] S. R. y. Cajal, La rétine des vertébrés, Cell. 9 (1893) 17–257.

[2] M. Piccolino, Cajal and the retina: a 100-year retrospective, Trends Neurosci. 11 (12) (1988) 521–525.

[3] R. Nelson, Visual responses of Ganglion cells, in: H. Kolb, E. Fernandez, R. Nelson (Eds.), Webvision Organ. Retin. Vis. Syst., Salt Lake City (UT): University of Utah Health Sciences Center, 2007, pp. 169–226.

[4] R. H. Masland, Neuronal diversity in the retina, Curr. Opin. Neurobiol. 11 (4) (2001) 431–436. doi:10.1016/S0959-4388(00)00230-0.

[5] R. H. Masland, The neuronal organization of the retina, Neuron 76 (2) (2012) 266–280. doi:10.1016/j.neuron.2012.10.002.

[6] T. Gollisch, M. Meister, Eye smarter than scientists believed: Neural computations in circuits of the retina, Neuron 65 (2) (2010) 150–164. arXiv:NIHMS150003, doi:10.1016/j.neuron.2009.12.009.

[7] A. R. Aqra, Human medical physiology: Ocular physiology (Physiology of the eye).
URL `http://hmphysiology.blogspot.com/p/occularphysiology.html`

[8] E. de Boer, P. Kuyper, Triggered Correlation, IEEE Trans. Biomed. Eng. 15 (3) (1968) 169–179.

[9] D. Kerr, M. McGinnity, S. Coleman, Modelling and analysis of retinal ganglion cells through system identification, in: Proc. Int. Conf. Neural Comput. Theory Appl., 2014, pp. 158–164. doi:10.5220/0005069701580164.

[10] P. Z. Marmarelis, K. I. Naka, White-noise analysis of a neuron chain: an application of the Wiener theory, Science (80-.). 175 (4027) (1972) 1276–1278.

[11] N. Wiener, Nonlinear problems in random theory, Wiley, New York, USA, 1958.

[12] V. Volterra, Theory of functionals and of integro and integro-differential equations, Dover Publications, New York, USA, 1959.

[13] P. Z. Marmarelis, V. Z. Marmarelis, Analysis of physiological systems: The white noise approach, Springer, 1978.

[14] M. J. Korenberg, H. M. Sakai, K. Naka, Dissection of the neuron network in the catfish inner retina: III. Interpretation of spike kernels, J. Neurophysiol. 61 (1989) 1110–1120.

[15] M. J. Korenberg, I. W. Hunter, The identification of nonlinear biological systems: Volterra kernel approaches, Ann. Biomed. Eng. 24 (2) (1996) 250–268.

[16] V. Z. Marmarelis, Identification of nonlinear biological systems using Laguerre expansions of kernels, Ann. Biomed. Eng. 21 (6) (1993) 573–589. doi:10.1007/BF02368639.

[17] J. D. Victor, R. M. Shapley, B. W. Knight, Nonlinear analysis of cat retinal ganglion cells in the frequency domain, Proc. Natl. Acad. Sci. USA 74 (7) (1977) 3068–3072.

[18] J. D. Victor, Nonlinear systems analysis: Comparison of white noise and sum of sinusoids in a biological system, Proc. Natl. Acad. Sci. USA 76 (2) (1979) 996–998. doi:10.1073/pnas.76.2.996.

[19] J. D. Victor, R. Shapley, A method of nonlinear analysis in the frequency domain, Biophys. J. 29 (3) (1980) 458–483. doi:10.1016/S0006-3495(80)85146-0.

[20] R. Herikstad, J. Baker, J.-P. Lachaux, C. M. Gray, S.-C. Yen, Natural movies evoke spike trains with low spike time variability in cat primary visual cortex, J. Neurosci. 31 (44) (2011) 15844–15860. doi:10.1523/JNEUROSCI.5153-10.2011.

[21] F. Giri, E.-W. Bai (Eds.), Block-oriented Nonlinear System Identification, Springer Lecture Notes in Control and Information Sciences, 2010.

[22] S. Ostojic, N. Brunel, From spiking neuron models to linear-nonlinear models, PLoS Comput. Biol. 7 (1). doi:10.1371/journal.pcbi.1001056.

[23] X. Cao, Encoding of natural images by retinal ganglion cells, Phd thesis, University of Southern California (2010).

[24] E. J. Chichilnisky, A simple white noise analysis of neuronal light responses, Netw. Comput. Neural Syst. 12 (2) (2001) 199–213.

[25] S. Bind, A. K. Tiwari, A. K. Sahani, A survey of machine learning based approaches for Parkinson disease prediction, Int. J. Comput. Sci. Inf. Technol. 6 (2) (2015) 1648–1655.

[26] M. Zhang, Artificial Higher Order Neural Networks for Modeling and Simulation, Information Science Reference (an imprint of IGI Global), Hershey, PA, 2013.

[27] S. McDonald, D. Kerr, S. Coleman, P. Vance, Modelling Retinal Ganglion Cells using Self-Organising Fuzzy Neural Networks, in: Proc. 2015 Int. Jt. Conf. Neural Networks, Killarney, 2015, pp. 1–8.

[28] B. Lau, G. B. Stanley, Y. Dan, Computational subunits of visual cortical neurons revealed by artificial neural networks., Proc. Natl. Acad. Sci. U. S. A. 99 (13) (2002) 8974–9. doi:10.1073/pnas.122173799.

[29] S. R. Lehky, T. J. Sejnowsky, R. Desimone, Predicting responses of nonlinear neurons in monkey striate cortex to complex patterns, J. Neurosci. 12 (9) (1992) 3568–3581.

[30] R. Prenger, M. C.-K. Wu, S. V. David, J. L. Gallant, Nonlinear V1 responses to natural scenes revealed by neural network analysis, Neural Networks 17 (5-6) (2004) 663–679. doi:10.1016/j.neunet.2004.03.008.

[31] O. Schwartz, J. W. Pillow, N. C. Rust, E. P. Simoncelli, Spike-triggered neural characterization, J. Vis. 6 (4) (2006) 484–507. doi:10.1167/6.4.13.

[32] S. Haykin, Neural Networks: A comprehensive Foundation, 2nd Edition, Prentice-Hall, 1999.

[33] M. Khashei, M. Bijari, A novel hybridization of artificial neural networks and ARIMA models for time series forecasting, Appl. Soft Comput. 11 (2) (2011) 2664–2675.

[34] D. J. C. Mackay, A practical Bayesian framework for backpropagation networks, Neural Comput. 4 (3) (1992) 448–472.

[35] F. D. Forsee, M. T. Hagan, Gauss-Newton approximation to Bayesian learning, in: Proc. 1997 IEEE Int. Conf. Neural Networks, 1997, pp. 1930–1935.

[36] F. R. Burden, D. A. Winkler, Robust QSAR models using Bayesian regularized neural networks, J. Med. Chem. 42 (16) (1999) 3183–3187.

[37] J. M. P. Menezes, G. a. Barreto, Long-term time series prediction with the NARX network: An empirical evaluation, Neurocomputing 71 (16-18) (2008) 3335–3343. doi:10.1016/j.neucom.2008.01.030.

[38] H. Xie, H. Tang, Y. H. Liao, Time series prediction based on NARX neural networks: An advanced approach, in: Proc. 2009 Int. Conf. Mach. Learn. Cybern., Vol. 3, 2009, pp. 1275–1279. doi:10.1109/ICMLC.2009.5212326.

[39] C. Cortes, V. Vapnik, Support-vector networks, Mach. Learn. 20 (3) (1995) 273–297.

[40] V. N. Vapnik, Statistical Learning Theory, Wiley New York, 1998.

[41] N. S. Altman, An introduction to kernel and nearest-neighbor nonparametric regression, Am. Stat. 46 (3) (1992) 175–185.

[42] J. K. Liu, T. Gollisch, Spike-triggered covariance analysis reveals phenomenological diversity of contrast adaptation in the retina, PLOS Comput. Biol. 11 (7) (2015) e1004425. doi:10.1371/journal.pcbi.1004425.

[43] D. Ringach, R. Shapley, Reverse correlation in neurophysiology, Cogn. Sci. 28 (2) (2004) 147–166. doi:10.1016/j.cogsci.2003.11.003.

[44] T. Gollisch, M. Meister, Rapid neural coding in the retina with relative spike latencies., Science 319 (5866) (2008) 1108–1111. doi:10.1126/science.1149639.

[45] G. W. Schwartz, H. Okawa, F. a. Dunn, J. L. Morgan, D. Kerschensteiner, R. O. Wong, F. Rieke, The spatial structure of a nonlinear receptive field, Nat. Neurosci. 15 (11) (2012) 1572–1580. doi:10.1038/nn.3225.

[46] H. K. Hartline, The response of single optic nerve fibers of the vertebrate eye to illumination of the retina, Am. J. Physiol. 121 (1938) 400–415.

[47] R. Segev, J. Puchalla, M. J. Berry II, Functional organization of ganglion cells in the salamander retina, J. Neurophysiol. 95 (2006) 2277–2292. doi:10.1152/jn.00928.2005.

[48] O. Marre, D. Amodei, N. Deshmukh, K. Sadeghi, F. Soo, T. E. Holy, M. J. Berry II, Mapping a complete neural population in the retina, J. Neurosci. 32 (43) (2012) 14859–14873. doi:10.1523/JNEUROSCI.0723-12.2012.

[49] A. L. Fairhall, C. A. Burlingame, R. Narasimhan, R. A. Harris, J. L. Puchalla, M. J. Berry II, Selectivity for multiple stimulus features in retinal ganglion cells, J. Neurophysiol. 96 (5) (2006) 2724–2738. doi:10.1152/jn.00995.2005.