Multilayer Neural Networks with Receptive Fields as a Model for the Neuron Reconstruction Problem

Wojciech Czarnecki

Department of Computer Linguistics and Artificial Intelligence,
Faculty of Mathematics and Computer Science,
Adam Mickiewicz University, Poznan, Poland

Abstract. The developed model consists of a multilayer neural network with receptive fields used to estimate the local direction of the neuron on a fragment of microscopy image. It can be used in a wide range of classical neuron reconstruction methods (manual, semi-automatic, local automatic or global automatic), some of which are also outlined in this paper. The model is trained on an automatically generated training set extracted from a provided example image stack and corresponding reconstruction file. During the experiments the model was tested in simple statistical tests and in real applications, and achieved good results. The main advantage of the proposed approach is its simplicity for the end-user, one who might have little or no mathematical/computer science background, as it does not require any manual configuration of constants.

Keywords: artificial neural network, receptive fields, computer vision, neuron reconstruction

1 Introduction

Understanding how the brain works is one of the greatest challenges of modern science [1]. The development of computational methods for the study of neural anatomy is of particular importance in this field [2]. The neuron reconstruction problem, i.e. retrieving a three-dimensional spatial structure graph of the neural cells imaged using various types of light microscopy, has been investigated since the 1980s [3]. A number of studies have been carried out, e.g. some concerning local automatic tracing methods, where a structure template (e.g. cylinders) is matched to the image data during the iterative greedy process [4]. Others represent a global approach to the problem, where after the seeding step, a minimum spanning tree search with some custom metric is executed [5]. Evers et al. [6] also proposed an approach based on the geodesic active contour model. There are also many semi-automatic algorithms, such as the efficient 3D tracing and editing method by Peng et al. [7]. To our knowledge, there have been no studies regarding the possible applications of fully machine learning-based models.

2 Motivation

Artificial intelligence methods are being successfully used in many fields of biology, from gene detection [8], through protein secondary structure prediction [9], to the analysis of microarrays [10]. They are also becoming more common in general computer vision as well as in biologically related applications [11] [12]. The choice of a neural network with receptive fields as the underlying model for the neuron reconstruction problem is motivated by the successful application of similar methods to image segmentation problems [13] [15], the fact that it is a simplified model of the human visual system, and that for a neurobiologist, i.e. the end-user of this kind of system, such an approach means that there is no required mathematical or computer science knowledge needed to use it, as the whole configuration of the model is executed automatically.

3 Model

The general idea is to create a feedforward neural network based model which, for a given two-dimensional matrix of voxel intensities (being its range of sight, located in the three-dimensional space, rotated in some direction $d \in \mathbb{R}^3$), can estimate if the neuron in this point is rotated in d. For a better analogy to actual sight, this *vision field* is also shifted in the direction $-d$ (so it sees more of what is 'ahead' than 'behind' it, as this data is more important for making a correct estimation).

3.1 Vision field

Let us denote for some fixed $p \in \mathbb{N}$ (called the *size of the vision field*):

$$V_{XY} = \{-p, -p+1, ..., 0, ..., p-1, p\} \times \{-p, -p+1, ..., 0, ..., p-1, p\} \times \{0\} \quad (1)$$

where XY corresponds to an XY slice of the 3D intensity array. Similarly, one can define V_{XZ}, V_{YZ} and any other type of field. The vision field of type t, centered in point $q \in \mathbb{R}^3$, rotated in the direction d can now be defined as the translation and rotation of V_f (for some fixed $v \in \mathbb{R}$ - *size of the shift*).

$$F_{t,q,d} = \left\{ q + x' + \frac{d}{\|d\|_2} v : x' \in R_d(V_t) \right\} \quad (2)$$

where $R_d(V_t)$ is a set of points from V_t, rotated in the spherical coordinate system of angles between vector d and $(1, 0, 0)$. Naturally, there is an infinite number of possible rotations of the two-dimensional plane in the direction of a given vector. Selecting this particular one is motivated by the fact that it ensures the biggest projection on the XY plane, which has the best resolution in the input data (and as such is the most reliable source of information). The input data is a three-dimensional array I of voxel intensities, and most of the points of the vision fields defined in Equation 2 have non-integer coordinates, so some interpolation

of the image function is required. The simplest possibility is to define f as a weighted sum of the values of the closest points with integer coordinates.

$$f(x) = \frac{\sum_{y \in BB(x)} I(y) \cdot max(1 - \|y - x\|_2, 0)}{\sum_{y \in BB(x)} max(1 - \|y - x\|_2, 0)} \tag{3}$$

where $BB(x)$ is a set of eight points with integer coordinates forming a unit bounding box of x. More advanced methods are also possible, but as this function is called many times during an algorithm's execution, it should be computationally easy. Figure 1 shows sample f function values for some vision fields.

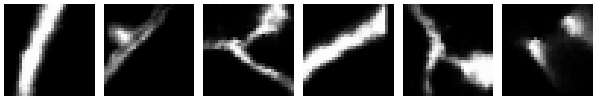

Fig. 1. Sample f function values of some XY vision fields, visualized as 31x31 px 8bit images ($p = 15$, $v = 5$)

.

3.2 Neural Network Estimator

The neural network estimator is composed of artificial neural networks (ANNs) that accept as their input a matrix of voxel intensity and output a value in the $[0, 1]$ range. This value shows if the image from a vision field, placed in point $x \in \mathbb{R}^3$, rotated in the direction d, represents a correctly rotated part of the neuron. The vision fields were defined as a set of points for simplicity but, naturally, we can treat them as two-dimensional matrices. The main element of this model is a four-layer feedforward neural network (Fig. 2), which is trained on the automatically generated training set and extracted from the provided example image stack and corresponding reconstruction file (Sec. 3.3). The weight sharing technique is not used, so that the network can develop different feature detectors in each receptive field (similar to the Phung et al. [13] approach). Every neuron, except for the output one, uses the hyperbolic tangent activation function, so it can provide both inhibitory and excitatory signals [14], while the output node activation is a simple sigmoid function. Bias neurons are also used in all layers.

The neural network estimator can be composed of many actual ANNs trained on various orientations of the vision fields. This helps to easily build more accurate models. Let us denote:

$$NNE(x, d) = \frac{\sum_{t \in T} NN_t(F_{t,x,d})}{|T|} \tag{4}$$

where T is a set of all the used field types (e.g. $T = \{XY, XZ\}$), and $NN_t(F_{t,x,d})$ is the output of the neural network indexed with t on the vision field $F_{t,x,d}$ values of function f (Eq. 3).

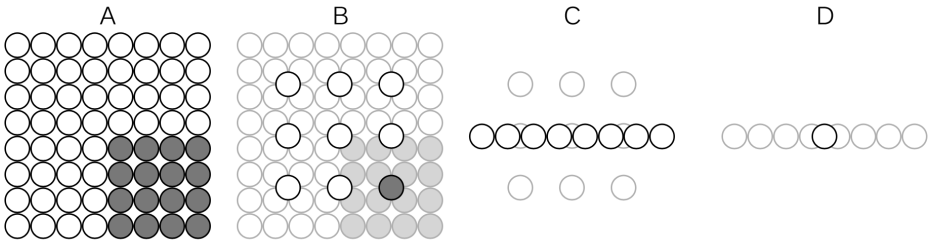

Fig. 2. Neural network topology. A - input layer, with each neuron connected to the corresponding point in the vision field, B - hidden receptive layer, where each neuron is connected only to its square shaped receptive field (gray neurons) in the input layer, C - hidden layer, fully connected with the previous one, D - output layer, also fully connected.

3.3 Learning

To train this model, only an image stack and corresponding manual reconstruction are needed. The depth first search algorithm executed on the reconstruction tree can extract (on each point of this graph) all the required vision fields. The only problem then is to find the expected output for this input according to the given data. A good heuristic for this task should:

- be absolute, i.e. its results have to be comparable whichever tracing position is considered
- prefer locally best fitted directions
- not use information from the image stack
- consider difficulties deriving from the bifurcation points
- take into account the diameter of the neuron in the current location
- be bounded, as it will be used as the desired output of the neural network.

For this purpose, the heuristic S defined in Equations 5, 6 and 7 is given. First, $dist_q(d,x)$ is defined, which measures the distance of point x from the half-line between the origin and direction d (with respect to the neuron diameter $diam_q$ in point q).

$$dist_q(d,x) = \begin{cases} 0, & \text{for} \quad \|x\|_2 \leq diam_q \\ \left\|x - d(x^T d)\right\|_2, & \text{for} \quad \|x\|_2 > diam_q \text{ and } x^T d \geq 0 \\ \|x\|_2, & \text{for} \quad \|x\|_2 > diam_q \text{ and } x^T d < 0 \end{cases} \tag{5}$$

To prefer locally best fitted directions, the *small score function* is defined as a weighted mean of *dist* values.

$$s_q(d, \{x_i\}_{i\in\{1,..,k\}}) = 1 - \frac{2}{k(k+1)} \sum_{i=1}^{k} (k-i+1) \frac{min\{dist_q(d, x_i - q), maxdist\}}{maxdist}, \tag{6}$$

where *maxdist* is the maximum value of *dist* worth considering in our applications. As it is a simple cut-off, its exact value is not crucial and can be set in the (1,p] range.

S is defined as a search through all possible paths starting in the considered point for the best (in the sense of s value) fit.

$$S(d,q) = max_{X \in T_q^p} \frac{max_{x \in X} max\{0, (x-q)^T d\}}{p} s_q(d, X) \tag{7}$$

where T_q^p is a set of paths in the reconstruction tree T from q to some leaf, with all nodes removed that are more distant from q than the size of the vision field - p. Sample S function values with corresponding f function values of some vision fields extracted during the learning phase are given in Figure 3.

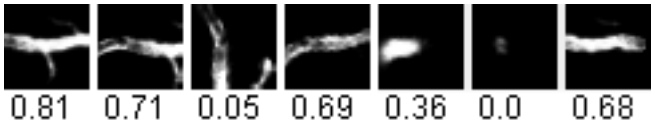

Fig. 3. Example f function values of vision fields with corresponding S values

4 Experiments and results

After simple tests the model parameters were set to 20 hidden neurons, 31x31 vision fields, and 4x4 receptive fields with 2 pixel overlapping. The network was trained using the backpropagation algorithm (learning rate set to 0.02, momentum set to 0.7). The early stopping method was used to find the best training stop moment [16]. Confocal microscopy images of the olfactory bulb of Drosophila, retrieved from the DIADEM Challenge [17] competition, were used during the experiments. One image (with a corresponding gold standard reconstruction) was used to construct the training set (3000 samples) and validation set (750 samples), and the rest of the images were used for testing. Tests were also run on images from the Stanford FlyBrain Database containing images from the same type of microscopy and brain region as the training data (unfortunately without corresponding reconstructions).

In most cases the DIADEM metric [18] was used to compare the results of the algorithm to the correct, manual reconstructions. During the statistical tests some other metrics were also used:

- SE - mean amount of incorrectly ordered triples (max, mid, low) by NNE, generated during validation of set extraction,
- ME - mean squared error on the validating set,
- MA - mean angle (in radians) between the correct direction (according to the heuristic S) and the one with the highest response of NNE among all possible directions (tested using one whole reconstruction file).

4.1 Learning phase tests

The first step of the experiments was to test the learning phase of the model and to find some of the remaining parameters. As mentioned before in Section 3.2, NNE can use many different vision fields and neural networks for them. In Table 1 one can find the sample results of testing three different settings. After

id	trained	used	SE	ME	MA
1.	XY	XY	0.0048	0.0096	0.289
2.	XY	XY, XZ	0.0032	0.0077	0.258
3.	XY, XZ	XY, XZ	0.0048	0.0104	0.290

Table 1. The 'trained' column contains vision fields on which the neural network(s) was/were trained. In the second experiment both vision fields were evaluated using a network trained on the XY field.

this phase all of the following experiments were performed using NNE composed of two vision fields (XY, XZ) trained on a single (XY) field, as this achieved the best results (mostly because of the low resolution among the Z-axis, which caused a low quality of the XZ training vectors). These were all executed on the testing set of images (different than those used for the training).

4.2 Evaluation tests

The simple greedy local tracing algorithm was implemented to test the model's usefulness in local automatic approaches. At each step the algorithm sets NNE in every direction (from the precomputed set of possible directions) and records its output. It moves in the direction d_{max} of the highest value.

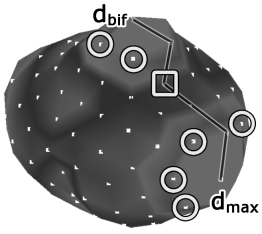

Fig. 4. A visualization of the NNE values (color and distance from the 'sphere' center is proportional to the network output) of all possible directions (white squares) - the directions in white circles are the 'candidates' for the bifurcation point and the direction in the white square is the one with the lowest value among all the shortest paths between d_{max} and the candidates.

It bifurcates if and only if there is a non-empty set of other directions (candidates) with a value of at least 0.7 of the d_{max} value. It chooses the direction

d_{bif} from this set by searching through the shortest paths on the unit sphere between those directions and d_{max} and selecting the point which has the lowest value on its path (see Figure 4 for more details).

	NeuronStudio	**NNE**	**Eugene Myers Team**
score	0.80	0.82	0.97

Table 2. Results of greedy local tracing in comparison to free NeuronStudio software and the DIADEM Challenge finalists' algorithm [19]

The best achieved result during the tests was 0.82 (Fig. 5), which is comparable to the NeuronStudio results, but far behind one of the DIADEM challenge finalists' algorithms - that of the Eugene Myers Team (Table 2). The results for the rest of the testing set were between 0.4 and 0.7 (mostly because of the weak bifurcation condition, but when it was user manually helped, by running it from the bifurcation points, the score increased to 0.7-0.9).

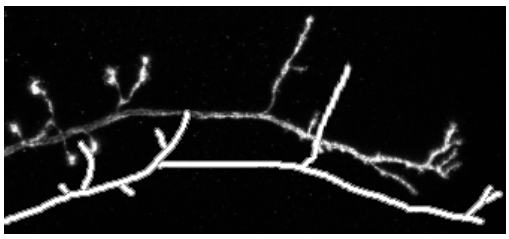

Fig. 5. Fragment of the XY projection of the best reconstruction achieved using the local greedy algorithm. The reconstruction is shifted for better visualization.

In applications where the user traces a neuron by continuously selecting some (possibly distant) points of the cell, the trained NNE can trace the dendrite using Dijkstra's algorithm with the distance between two spatially adjacent voxels a and b defined as follows:

$$d(a,b) = \left(1 - \frac{NNE(a, b-a) + NNE(b, a-b)}{2}\right) \|a-b\|_2 \tag{8}$$

Tests of this type of approach (Fig. 6) resulted in reconstructions of the DIADEM metric between 0.8 and 0.95, with a mean score of 0.89.

A similar approach can be used in global automatic methods, where after the seeding part (preprocessing of the image to find voxels that are definitely parts of the neuron), the minimum spanning tree algorithm (e.g. Kruskal's or Prim's algorithm) is executed according to the distance function defined in Equation 8.

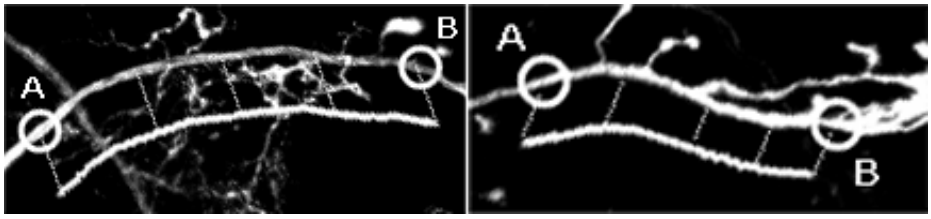

Fig. 6. Fragment of the XY projection of semi-automated tracing between point A and B selected by the user. Reconstruction is shifted for better visualization.

5 Conclusion

The model proposed in this paper is the first approach to using machine learning methods directly to the neuron reconstruction problem. The achieved results, wide range of possible applications (from manual reconstruction systems, where it can serve as a reconstruction assistant, suggesting the best tracing direction or simply visualizing the network output for all possible directions, to local and global automatic systems) and intuitive configuration (as every neurobiologist dealing with the reconstruction problem already has some manual reconstructions that can be used to train this model), all show that it can be a valuable alternative to the models currently being used.

Future plans include tuning the model's structure (e.g. introducing both inhibitory and excitatory receptive fields, similarly to those in Fernandes et al. [20]), improving the heuristic used for learning set extraction and a more complex method of network training (e.g. reinforcement learning with the agent moving along the neuron structure instead of simple batch learning).

References

1. Roysam, B., Shain W., Ascoli G.A.: The central role of neuroinformatics in the national academy of engineering's grandest challenge: reverse engineer the brain. Neuroinformatics, vol. 7, pp. 1–5 (2009)
2. Ascoli, G.A.: Computational Neuroanatomy: Principles and Methods. New Jersey: Humana Press (2002)
3. Capowski, J.J.: An automatic neuron reconstruction system. Journal of Neuroscience Methods, vol. 8, pp. 353–364 (1983)
4. Al-Kofahi, K.A., Lasek, S., Szarowski, D.H., Pace, C.J., Nagy, G., Turner, J.N., Roysam, B.: Rapid automated three-dimensional tracing of neurons from confocal image stacks. IEEE Trans Informat Technol Biomed, vol. 6, pp. 171–187 (2002)
5. Peng, H., Ruan, Z., Atasoy, D., Sternson, S.: Automatic reconstruction of 3D neuron structures using a graph-augmented deformable model. Bioinformatics, vol. 26, pp. 138–146 (2010)
6. Evers, J.F., Schmitt, S., Sibila, M., Duch, C.: Progress in functional neuroanatomy: Precise automatic geometric reconstruction of neuronal morphology from confocal image stacks, Journal of Neurophysiology, vol. 93, pp. 2331–2342 (2005)

7. Peng, H., Ruan, Z., Long, F., Simpson, J. H., Myers, E. W.: V3D enables real-time 3D visualization and quantitative analysis of large-scale biological image data sets, Nature Biotechnology, vol. 28(4), pp. 348–353 (2010)
8. Burge, C.B.: Modeling dependencies in pre-mRNA splicing signals, Computational Methods in Molecular Biology, pp. 127–163 (1998)
9. Jones, D.T.: Protein secondary structure prediction based on position-specific scoring matrices, Journal of Molecular Biology, vol. 292(2), pp. 195–202 (1999)
10. Brown, M.P.S., Grundy, W.N., Lin, D., Cristianini, N., Sugnet, C.W., Furey, T.S., Ares, M., Haussler, D.: Knowledge-based analysis of microarray gene expression data by using support vector machines, Proceedings of the National Academy of Sciences of the United States of America, vol. 97(1), pp. 262–267 (2000)
11. Turaga, S.C., Murray, J.F., Jain, V., Roth, F., Helmstaedter, M., Briggman, K., Denk, W., Seung, H.S.: Convolutional networks can learn to generate affinity graphs for image segmentation, Neural Computation, vol. 22(2), pp. 511–538 (2010)
12. Kreshuk, A., Straehle, C.N., Sommer, C., Koethe, U., Cantoni, M., Knott, G., Hamprecht, F.A.: Automated detection and segmentation of synaptic contacts in nearly isotropic serial electron microscopy images, PLoS ONE. 6 (2011)
13. Phung, S.L., Bouzerdoum, A.: A pyramidal neural network for visual pattern recognition. IEEE Transactions on Neural Networks, vol. 18(2), pp. 329–343 (2007)
14. Haykin, S.: Neural Networks and Learning Machines, Prentice Hall (2008)
15. Egmont-Petersen, M.: Image processing with neural networks - a review, Pattern Recognition, vol. 35, pp. 2279–2301 (2002)
16. Sarle, W.S.: Stopped Training and Other Remedies for Overfitting, Proceedings of the 27th Symposium on the Interface of Computing Science and Statistics, pp. 352–360 (1995)
17. Brown, K.M., Barrionuevo, G., Canty, A.J., De Paola, V., Hirsch, J.A., Jefferis, G.S.X.E., Lu, J., Snippe, M., Sugihara, I., Ascoli, G.A.: The diadem data sets: Representative light microscopy images of neuronal morphology to advance automation of digital reconstructions, Neuroinformatics, vol. 9(2-3), pp. 143–157 (2011)
18. Gillette, T., Brown, K., Ascoli, G.A.: The diadem metric: Comparing multiple reconstructions of the same neuron, Neuroinformatics, vol. 9, pp. 233–245 (2011)
19. Zhao, T., Xie, J., Amat, F., Clack, N., Ahammad, P., Peng, H., Long, F., et al.: Automated Reconstruction of Neuronal Morphology Based on Local Geometrical and Global Structural Models, Neuroinformatics, vol. 9(2-3), pp. 247–261 (2011)
20. Fernandes B.J.T., Cavalcanti G.D.C., Ren T.I.: Nonclassical receptive field inhibition applied to image segmentation, Neural Network World, vol. 19, pp. 21–36 (2009)