Neural network models of the tactile system develop first-order units with spatially complex receptive fields

Charlie W. Zhao
The University of Western Ontario

Mark J. Daley
The University of Western Ontario, mdaley2@uwo.ca

J. Andrew Pruszynski
The University of Western Ontario

Follow this and additional works at: https://ir.lib.uwo.ca/brainpub

Citation of this paper:
Zhao, Charlie W.; Daley, Mark J.; and Andrew Pruszynski, J., "Neural network models of the tactile system develop first-order units with spatially complex receptive fields" (2018). Brain and Mind Institute Researchers' Publications. 741.
https://ir.lib.uwo.ca/brainpub/741
Neural network models of the tactile system develop first-order units with spatially complex receptive fields

Charlie W. Zhao¹,², Mark J. Daley¹,³,⁴,⁶, J. Andrew Pruszynski¹,⁶,⁷,⁸,⁹,¹⁰*

¹ Dept. of Computer Science, Western University, London, Ontario, Canada, ² School of Medicine, Yale University, New Haven, Connecticut, United States of America, ³ Dept. of Biology, Western University, London, Ontario, Canada, ⁴ Dept. of Actuarial Sciences and Statistics, Western University, London, Ontario, Canada, ⁵ Vector Institute, Toronto, Ontario, Canada, ⁶ Brain and Mind Institute, Western University, London, Ontario, Canada, ⁷ Dept. of Physiology and Pharmacology, Western University, London, Ontario, Canada, ⁸ Dept. of Psychology, Western University, London, Ontario, Canada, ⁹ Robarts Research Institute, Western University, London, Ontario, Canada, ¹⁰ Dept. of Integrative Medical Biology, Umeå University, Umeå, Sweden

* andrew.pruszynski@uwo.ca

Abstract

First-order tactile neurons have spatially complex receptive fields. Here we use machine-learning tools to show that such complexity arises for a wide range of training sets and network architectures. Moreover, we demonstrate that this complexity benefits network performance, especially on more complex discrimination tasks and in the presence of noise. Our work suggests that spatially complex receptive fields are normatively good given the biological constraints of the tactile periphery.

Results

First-order tactile neurons in the hairless skin of the human hand have distal axons that branch in the skin and form many transduction sites [1–3], yielding spatially complex receptive fields with many highly sensitive zones [4,5] (Fig 1A). We have recently shown that this arrangement permits first-order tactile neurons to signal high-level features of touched objects such as the orientation of a touched edge [4,6,7], a capacity previously considered a hallmark of processing in the somatosensory cortex [8–10]. Here we leverage machine learning tools to investigate why complex receptive fields arise and what computational benefits they yield. We show that complex receptive fields arise under a wide range of training sets and biologically realistic network constraints. We also show that complex receptive fields benefit network performance, especially on more complex discrimination tasks and in the presence of noise.

We abstracted the tactile processing pathway with a four-layer feedforward neural network (Fig 1B and 1C). The input layer of our network consisted of 784 units, representing mechano-receptors distributed over a small patch of skin. In this arrangement, the weight matrix between the input and first hidden layer—which we call $W^{(1)}$—represents the receptive fields of first-order tactile neurons. Our network was trained on a range of stimuli including single...
C. Our four-layer feedforward neural network. The first layer models a small patch of skin, $W^{(1)}$ represents receptive fields, and the second layer models first order neurons. Layers 3 and 4 are a functional abstraction of the central nervous system. The relative sizes of each layer are shown but not to scale. Arrows represent fully connected feedforward weights between subsequent layers. End organs and first order neurons in (B) are colour matched with the layers that represent them in the model. (D) Examples of training data used to represent tactile stimuli. Each stimulus is shown on a 28 x 28 step grid. Stimuli were passed through a Gaussian filter and randomly rotated and translated. Points data were also randomly scaled.

Fig 1. Theoretical and analytical setup. (A) Examples of receptive fields from human first-order tactile neurons terminating in the fingertip acquired via microneurography. Color indicates the relative firing rate of the neuron when stimulated with a small punctate stimulus. For full details, see Pruszynski and Johansson (2014). (B) Graphic representation of a cross-section through the human glabrous skin. Note how a single afferent neurons branches and innervates multiple mechanoreceptive end organs. (C) Our four-layer feedforward neural network. The first layer models a small patch of skin, $W^{(1)}$ represents receptive fields, and the second layer models first order neurons. Layers 3 and 4 are a functional abstraction of the central nervous system. The relative sizes of each layer are shown but not to scale. Arrows represent fully connected feedforward weights between subsequent layers. End organs and first order neurons in (B) are colour matched with the layers that represent them in the model. (D) Examples of training data used to represent tactile stimuli. Each stimulus is shown on a 28 x 28 step grid. Stimuli were passed through a Gaussian filter and randomly rotated and translated. Points data were also randomly scaled.
We next asked how the degree of convergence between the input and first hidden layers influenced receptive fields. That is, how physical constraints placed on the number of first-order tactile neuron axons traveling within the peripheral nerve should affect connectivity to mechanoreceptors in the skin. We reasoned that increasing convergence would increase receptive field complexity, since this smaller set of units must still encode the same set of inputs. We tested this idea by decreasing the size of the first hidden layer from 81 to 36 units, closer to the lower limit of biologically relevant convergence [1–3], and training the network on the same four training sets described above. Increasing convergence did result in more complex receptive fields for alphabet and mixed networks (Fig 2B). On average, the 36-unit alphabet network had 3.0 more peaks than the 81-unit alphabet network (t(38) = 46.39, P < 0.01), and the 36-unit mixed network had 4.0 more peaks than the 81-unit mixed network (t(38) = 56.93, P < 0.01). Interestingly, however, the one point and the one and two point networks (our simplest training sets) did not show increased complexity with increased convergence (Fig 2B). In fact, the 36-unit one point network had 0.3 fewer peaks than the 81-unit one point network (t(38) = -8.55, P < 0.01), and the 36-unit one and two point network had 0.5 fewer peaks than the 81-unit one point two point network (t(38) = -10.00, P < 0.01).

At this point we further abstracted our network constraints to examine how they influenced the learned receptive fields. First, we trained our network on the mixed stimulus set without non-negative regularization in $W^{(1)}$ and found qualitative changes in receptive field morphology such that they no longer had structural similarities to our previously documented empirical receptive fields [4] (Fig 3A). Second, we trained our network on the mixed stimulus set with extreme convergence (4 units in the first hidden layer) and, again, found the resulting receptive fields did not resemble our empirical receptive fields (Fig 3B). Last, we trained our network on each of the four stimulus sets without convergence (i.e. 784 units in the first hidden layer). We reasoned that such a network may not develop complex receptive fields because it did not need to compress the input space, especially for the single dot training set given its simple spatial statistics. However, receptive fields with multiple highly sensitive zones emerged for all training sets to varying degrees (Fig 3C).

Given that our networks developed complex receptive fields under all network constraints and training sets, we investigated the functional consequences that such an arrangement had.
Fig 3. Alternative architectures. Same format as Fig 2 but showing exemplar receptive fields learned by three alternative networks featuring architectures with relaxed constraints.

https://doi.org/10.1371/journal.pone.0199196.g003
on sensory processing. In these analyses, we trained the network on unlabelled Mixed stimuli, then fixed $W^{(1)}$ and trained the remaining layers as a classifier using labelled Mixed stimuli. In our approach, the unsupervised training phase represents the encoding function of the tactile processing pathway, while the supervised training phase abstracts the more interpretive functions of the central nervous system. We compared this learned network against a network engineered to have single-peaked Gaussian receptive fields in $W^{(1)}$ on discrimination and identification tasks. For the engineered network, we selected the width of the Gaussian receptive field (SD = 3.0 steps) that resulted in best performance.

We first asked whether complex receptive fields benefit spatial accuracy. We had the network perform two-point discrimination, a task central to many studies of tactile acuity [13,18,19]. Specifically, we used a two-alternative forced choice paradigm and defined the difference limen as the separation distance between stimuli at which the network classified 75% of the stimuli correctly. The learned network had a mean difference limen of 6.94 (SD = 1.36) steps on our input space, which corresponds to a modelled distance of ~1–3 mm, depending on assumptions about mechanoreceptor innervation density. Overall, performance of learned and engineered networks were not significantly different with 81 units in the first hidden layer ($t(45) = -1.85, P = 0.071$; Fig 4A). Moreover, changing the degree of convergence from 81 to 36 units did not cause a statistically significant change in performance for either the learned or the engineered network ($F(1, 82) = 0.31, P = 0.58$; Fig 4A).

We then asked whether complex receptive fields benefit network performance in a more difficult identification task. We assessed each networks ability to correctly classify new instances of characters from the Roman alphabet not previously seen by the network during the training phase (see Methods), as has been previously done with human participants [14]. In this case, engineering $W^{(1)}$ to have single-peaked Gaussian receptive fields and increasing convergence both decreased network accuracy ($F(1,79) = 103.78, P < 0.01$, $F(1, 39) = 107.23, P < 0.01$, respectively), and the interaction between these factors was also significant ($F(1, 79) = 7.05, P = 0.0096$). That is, both learned and engineered networks performed well, but the learned networks outperformed engineered networks for both levels of convergence and the benefit of complex receptive fields increased with increased convergence (Fig 4B).

Finally, we asked whether complex receptive fields benefit network performance in the presence of noise. We introduced varying levels of normally distributed additive and

![Fig 4. Model performance. Performance of 81- and 36-hidden unit models either trained on mixed stimuli or engineered with fixed Gaussian receptive fields on the (A) two-point discrimination and (B) alphabet classification tasks. (A) Data points show the difference limen, defined as the separation distance at which the model classifies 75% of 2000 test points correctly. (B) Data points show the overall classification accuracy of 7800 tested Roman letters. Grey violin plots show the frequency distribution of difference limens and accuracy across model iterations. Performance is reported at varying levels of multiplicative or additive noise (see Methods). Groups may have different numbers of data points as some networks failed to converge and were not considered for testing.](https://doi.org/10.1371/journal.pone.0199196.g004)
multiplicative noise to the training data during both unsupervised and supervised training phases and then tested the network’s performance on a noiseless dataset. The effect of training noise on the network’s ability to classify characters from the Roman alphabet was substantial (Fig 4B). The learned network had an accuracy of 87.7% (SD = 1.1) with low levels of additive noise (see Methods) compared to 75.1% (SD = 2.5) for the fixed network with the same amount of noise, a statistically significant performance gap (t(41) = 20.65, P < 0.01). Convergence also significantly influenced classification accuracy under the different noise levels (F(6, 555) = 12.36, P < 0.01). The performance of the 36-unit network decreased by 1.4% compared to the 81-unit learned network with low levels of additive noise (t(38) = 4.25, P = 0.00013). In contrast, the performance of the 36-unit network with engineered Gaussian receptive fields decreased by 6.1% compared to the 81-unit engineered network (t(41) = 9.59, P < 0.01). The performance gap grew between learned and engineered networks with additional additive noise (Fig 4B). For all networks, multiplicative noise had a similar effect but much smaller effect size (Fig 4B).

Discussion

A core feature of the tactile processing pathway is that there are many more mechanoreceptors in the skin of the hand than there are first order tactile neurons in the median and ulnar nerves. It is not surprising, therefore, that first order tactile neurons branch [1–3] since this is the only way they can innervate all the available mechanoreceptors. What may be surprising is the spatial complexity and apparent heterogeneity of the innervation pattern [4,5], a feature which has been overlooked or ignored in previous models of the tactile processing pathway [13,20–22]. Our work here leverages simple machine learning tools to provide two fundamental insights in this respect. First, we show that spatially complex receptive fields are a normatively good and, perhaps, biologically parsimonious, arising under a wide range of training sets and network architectures. Second, we show that spatially complex receptive fields benefit network performance, especially in relatively difficult tasks and in the presence of noise.

Heterogeneously sampling the input space is a good thing for the nervous system to do because the input space of sensory stimuli is inherently sparse. Neural networks like the one we use here implicitly learn the statistical regularities (and thus sparsity) of the stimuli to which they are exposed. Indeed, such a machine learning approach has been shown to reproduce biological receptive field properties of neurons at various levels of the visual processing pathway [12,23]. Another suggestion for a mechanism to exploit sparsity comes from the field of compressed sensing, which shows that randomly sampling the input space can, under reasonable assumptions, allow a system to fully reconstruct a sparse input signal with fewer measurements than that prescribed by the Shannon-Nyquist theorem [24–27]. Given an input with sparsity S (at most S non-zero terms), in many situations the input signal can be fully reconstructed by randomly sampling at a frequency greater than 2S with no noise or multiplicative noise, or 4S with additive noise [24,26], consistent with our observation that networks with more spatially complex receptive fields are particularly immune to additive noise. Fig 5 illustrates a cartoon compressed sensing scenario in our experimental setting, showing that a network with fully randomized weights in the first hidden layer can perform strikingly well on the alphabet discrimination task relative to the learned and fixed networks we described above. That is, the random network performs only slightly worse than the learned network and equivalent to the fixed network with no noise and, as expected, is able to better maintain its performance as the amount of additive noise is increased. This is not to say that the heterogeneity of how first-order tactile neurons innervate mechanoreceptors is random—indeed random connectivity yields receptive fields that are qualitatively distinct from those we record.
from humans (Fig 5B)–but, rather, that even random sampling can outperform pixel-like sampling with Gaussian receptive fields.

**Methods**

**Feedforward neural network architecture**

We designed a four layer feedforward network model with layers $L_1$ to $L_2$ containing $s_1$ to $s_4$ units respectively. $s_1 = 784$, $s_2 = 81$ or $36$, $s_3 = 784$, and $s_4 = 26$ or $2$ depending on if the network is trained to perform alphabet classification or two-point discrimination. The general form of feedforward computation was as follows:

$$z_{l+1}^{(l+1)} = W^{(l)} a^{(l)}$$

$$a_{l+1}^{(l+1)} = f(z_{l+1}^{(l+1)})$$

![Diagram](https://doi.org/10.1371/journal.pone.0199196.g005)
We generated all training inputs \( \mathbf{X} \) such that \( x_{ij} \in \mathbb{R}^{28 \times 28} \). We generated Gaussian-points stimuli by initializing one or two peaks \( |x_{ij}| = 10 \) where \( i, j \) are integers chosen independently from distribution \( \mathcal{U}(0, 27) \), then passed through a two-dimensional Gaussian filter with width \( \sigma = 3.0 \).

We generated Roman letters stimuli as Helvetica characters normalized to 17 steps in height. We used similar height scaling for Braille characters. The filled portions of characters were initiated as \( |x_{ij}| = 1 \). We subjected each character to a random rotational angle drawn

Two-phased training and non-negativity constraint

We randomly initiated weights by drawing from distribution \( \mathcal{N}(0, 0.01) \). The general learning algorithm was mini-batch gradient descent with mini-batches of size 256. We trained the network in two phases. In the unsupervised learning phase, we trained \( L_1 \) to \( L_3 \), we used a rectified linear function \( f(x) = \max(0, x) \) for \( W^{(1)} \) and a softmax function for \( W^{(2)} \). For supervised training (\( L_1 \) to \( L_4 \), we used a rectifier for \( W^{(3)} \) and \( W^{(4)} \) and softmax for \( W^{(5)} \).

where \( W^{(0)} \) denotes the weights from layers \( L_1 \) to \( L_{l-1} \), \( \mathbf{z}^{(l+1)} \) is the weighted sum of outputs from layer \( L_0 \) and \( q^{(l)} \) is the output of layer \( L_0 \) after the activation function \( f \). For unsupervised training (\( L_1 \) to \( L_3 \)), we used a rectified linear function \( f(x) = \max(0, x) \) for \( W^{(1)} \) and a softmax function for \( W^{(2)} \). For supervised training (\( L_1 \) to \( L_4 \), we used a rectifier for \( W^{(3)} \) and \( W^{(4)} \) and softmax for \( W^{(5)} \).

We incorporated the asymmetric regularization term\([28]\), \( R(W^{(1)}) \), where

\[
R(W^{(1)}) = cW^{(1)}_j \text{if } W^{(1)}_j < 0; \ 0 \text{ otherwise}
\]

for each unit \( j \) of \( L_3 \) and unit \( i \) of \( L_2 \). \( c \) denotes an arbitrarily large constant, which we picked as 1000, that harshly penalized the network for learning negative weights in \( W^{(1)} \).

In the supervised phase, we froze \( W^{(1)} \) and trained \( L_1 \) to \( L_3 \) as an autoencoder. We reinitiated \( W^{(2)} \) between the two training phases. Depending on the discrimination task to be performed, the network may operate as a binary (for two-point discrimination) or multiclass (for alphabet) classifier. Gradient descent minimized the cross-entropy cost:

\[
J(W; x)_{\text{supervised}} = - \sum_x p(x) \log(q(x))
\]

The learning rule in this phase was:

\[
\Delta W^{(l)} := \Delta W^{(l)} + \nabla_{w^{(l)}} J(W; x) \text{ for } l = 2, 3, \text{ and } \Delta W^{(1)} = 0.
\]

Network hyperparameters used during training varied among different network architectures and training sets. Networks that did not reach convergence in the number of iterations were removed from testing.

Training stimuli

We generated all training inputs \( \mathbf{X} \) such that \( x_{ij} \in \mathbb{R}^{28 \times 28} \). We generated Gaussian-points stimuli by initializing one or two peaks \( |x_{ij}| = 10 \) where \( i, j \) are integers chosen independently from distribution \( \mathcal{U}(0, 27) \), then passed through a two-dimensional Gaussian filter with width \( \sigma = 3.0 \).

We generated Roman letters stimuli as Helvetica characters normalized to 17 steps in height. We used similar height scaling for Braille characters. The filled portions of characters were initiated as \( |x_{ij}| = 1 \). We subjected each character to a random rotational angle drawn
from distribution $N(0, 20)$ in degrees, followed by random horizontal and vertical translation drawn from distribution $N(0, 5)$ in steps.

We generated 60,000 training stimuli of each class. For Roman letters and Braille characters, there was approximately equal proportion of each character. Gaussian-points were evenly split between one and two points (i.e. 30,000 of each). We used standard one-hot encoding for labelling in supervised training.

Receptive field complexity

We bootstrapped 1000 receptive fields from each network. First, we designed a peak counting algorithm that calculated the number of significant local maxima contained in each receptive field. For each receptive field $R$, we define $r_{ij}$ as a peak if 1) it is a local maximum 2) $|r_{ij}| > (\max_k r_k)/2$, that is, the value of $r_{ij}$ is greater than half of the global maximum, and 3) $r_{ij}$ is at least 5 steps away from the next closest local maximum. These criteria prevent low amplitude noise from being counted as peaks. Second, we analyzed receptive fields in the frequency domain by performing discrete two-dimensional Fourier transformation using the Fast Fourier Transform algorithm. We performed Fourier transformation after normalizing sampled RFs by their peak values such that $\max_k r_k = 1.0$. Last, to compare information shared by each pair of networks, we used mutual information between pairs of bootstrapped RFs normalized by their respective entropies, such that 1.0 means perfect correlation and 0 means no mutual information. We binned weights into 10,000 bins before calculating mutual information so that the control group (learned versus learned) RFs has normalized mutual information of close to 1.0.

Model performance

We assessed network accuracy in two-point discrimination and alphabet classification. We implemented two-point discrimination using a two-alternative forced choice paradigm. We generated 2000 new stimuli (not used to train the network) of one and two Gaussian-points in equal proportions. Two Gaussian-points were spaced symmetrically about the center of the input space at distances 0 to 22 steps apart with increments of 2 steps. We subjected two Gaussian-points to a random integer rotational angle drawn from distribution $U(0, 90)$ in degrees. We defined the difference limen, or just-noticeable difference, for two-point discrimination as the distance at which the network correctly classified 75% of test stimuli. We estimated difference limen using cubic spline interpolation on the full accuracy plot.

We assessed the network on alphabet classification by testing it on 7800 new characters (not used to train the network, as above) with 300 instances of each letter, subjected to rotational and translational variability as described above.

To assess robustness against noise, we trained the networks with noise before testing them on noiseless data. We implemented multiplicative noise on input $X$ as $e_{ij} = c \cdot u \cdot x_{ij}$ for each coordinate $i, j$ in $X$, where $u$ was randomly drawn from distribution $N(0, 0.01)$. We implemented additive noise as $\theta_{ij} = c \cdot v \cdot \max_k x_k$, where $v$ was randomly drawn from distribution $N(0, 0.01)$. We designated $c = 1.0$ as low-level noise and $c = 3.0$ as high-level noise. Noise was re-instantiated at the beginning of each training epoch.

Acknowledgments

This work was supported by the Canadian Institutes of Health Research (Foundation Grant to JAP: 3531979) and the Natural Science and Engineering Research Council of Canada (Discovery Grant to MJD). JAP received a salary award from the Canada Research Chairs Program.
Author Contributions
Conceptualization: Charlie W. Zhao, Mark J. Daley, J. Andrew Pruszynski.
Formal analysis: Charlie W. Zhao.
Investigation: Charlie W. Zhao, J. Andrew Pruszynski.
Methodology: Charlie W. Zhao, Mark J. Daley, J. Andrew Pruszynski.
Project administration: J. Andrew Pruszynski.
Resources: J. Andrew Pruszynski.
Software: Charlie W. Zhao.
Supervision: Mark J. Daley.
Validation: Charlie W. Zhao.
Visualization: Charlie W. Zhao.
Writing – original draft: Charlie W. Zhao, J. Andrew Pruszynski.
Writing – review & editing: Charlie W. Zhao, Mark J. Daley, J. Andrew Pruszynski.

References


28. Lemme A, Reinhart RF, Stell JJ. Efficient online learning of a non-negative sparse