Biological Plausibility of Sparse Algorithms in Computer Vision

Stephan Grzelkowski[1] and Shirin Dora[1]

[1] *University of Amsterdam, Amsterdam, Netherlands*

Abstract

Neuroscience and artificial intelligence have a long history of contributing to each others progress. Early models in computer vision research were heavily influenced by our knowledge of early visual processing in the cortex. Sparseness quickly proved to be an effective constraint to produce powerful algorithms that were able to perform object classification. In this review, we present different methods for enforcing sparse populations in computer vision and how they affect network properties that relate to similarity with the sensory processing stream in the cortex. We show that different forms of regularization can be an effective way of promoting sparseness in neural network populations. L1 and L2 regularization, common ways of restraining the size of network parameters, create distributions of network activity that are comparable to those found in biological networks. In particular, L1 regularization can be compared to the evolutionary pressure on cortical processing to reduce energy consumption. Another form of regularization, dropout, randomly removes units in the network during training. This method also results in a sparse population of units during the testing phase. We argue that the introduction of dropout can be compared to the noisy conditions of natural sensory processing. The last spars algorithm we discuss relies on a specific form of activation function, the rectifier linear unit (ReLU). The ReLU resembles response characteristics of neurons in the early visual cortex and is therefore of particular interest for investigating the effects on network activity. In addition to producing sparse populations, all discussed methods affect the feature selectivity of units in the networks. As there is no consensus in the literature on the interaction between sparseness and feature selectivity we highlight the evidence sparse algorithms contributes to this discussion. This review offers insights into the interaction of computer vision and neuroscience and demonstrates gaps in our understanding of the connections between the two that need to be further investigated.

Introduction

Early machine learning models were heavily influenced by neuron-like architectures[1] as biology offers a successful example of a system optimized for tasks artificial intelligence aims to solve. Adaptation of our understanding of the brain's computational mechanisms to machine learning algorithms has brought rapid advances in artificial intelligence research.[2] In recent years, artificial intelligence has conversely brought insights to neuroscience through computational models that have enabled us to more closely investigate processing characteristics.[3–5] Experiments have shown that human psycho-physics can predict the performance of deep learning models for adversarial examples[6] and vice versa that the activity of specific layers of an artificial neural network can predict the activity of V4 neurons in the primate brain.[7] The shared structure between artificial intelligence and biology offers new ways to investigate cortical processing.

Computer vision and its biological analog are a prime example of this interaction.[8,9] Visual processing is one of the most thoroughly investigated biological sensory systems[10] and computer vision has been one of the most rapidly advancing technologies in the last decade.[8,11] A large factor for the success of models for visual processing is the promotion of sparse populations. Sparsity measures the proportion of units in a system that are active at a given point in time and is a prominent feature of the visual stream in the cortex. This review will investigate how sparseness constraints affect artificial neural network populations.

Hubel and Wiesel[12] discovered that neurons in the primary visual sensory areas of the brain respond to simple features like bars. In higher hierarchical areas, neurons become more selective to specific features, have larger receptive fields, and fewer neurons are active at the same time. Although feature selectivity and sparsity seem to be correlated in the visual stream, several papers have questioned a causal link between the two.[13,14] By observing the effects of enforcing sparse algorithms on unit properties, computer vision can offer insight into how these two characteristics are related.

Feature selectivity and sparsity have been associated with the brain's power to adapt to variable inputs: objects in our visual scenery are often partially occluded, can appear from different angles and under different environmental conditions. Despite these transformations, our brain is capable of recognizing these objects with ease. The brain's ability to be invariant to different image transformations is an essential aspect of its processing power that is difficult to reproduce for artificial neural networks. Deep neural networks rely on a large number of parameters and samples to achieve invariance to these transformations. However, a model with many parameters carries the risk of violating the principle of Occam's Razor: the solution for the problem can easily become overly complex.[15] In Machine Learning, this phenomenon is termed overfitting, which refers to the divergence of the error between the training data and testing data. An increase in the performance for the test set leads to poor performance on new samples. Artificial intelligence has used different methods to improve the generalization from training to testing data. Interestingly, implementation of these methods often causes the network to exhibit brain-like characteristics with regards to sparsity and feature selectivity.

In this review, we will summarize how different methods applied in computer vision compare to the characteristics of the visual processing stream in the cortex and in particular how they affect sparsity and feature selectivity. We will start by giving a short overview of the relevant attributes of the visual system. Next, we will compare different methods of promoting sparse populations in machine learning and their impact on sparsity and feature selection. We will conclude this review with a brief outlook into possible future collaboration between computer vision and neuroscience and suggest algorithms that might help us bridge the gap between them.

A biological basis of visual processing

The visual system is divided into two separate pathways, the ventral and dorsal stream. They are considered to serve two different functionalities. The dorsal stream is concerned with object location and the ventral stream processes object identification.[16–18] As we will be analyzing computer vision algorithms for object classification, we will give a short overview of the sensory processing in the ventral stream. It is composed of connections from the primary visual cortex (V1), over the extrastriate areas (V2, V4), to the inferotemporal cortex (IT) and recurrent projections throughout these areas. A characteristic of this pathway is the increase in feature complexity and selectivity throughout the processing stages of this hierarchy.[19,20] Neurons in V1 respond to simple object features such as orientation and direction of movement.[12] They project to V2 where neurons integrate the signal and encode more complex features.[21,22] V4 neurons respond to curved features that are composites of the shapes represented in V2.[23] In area IT, neurons are narrowly selective to highly complex shapes and objects, such as faces.[19,24–26] Along this hierarchical organization, neurons in later areas have larger receptive fields and are more invariant to small transformation in the input space.[25,27,28]

Neurons in higher-level regions are more selective and population responses grow sparser.[27,29,30] Throughout neuroscience literature, the term sparseness is used in two ways: (1) population sparseness measures the percentage of neurons active at a given point in time; (2) lifetime sparseness measures the response of a specific neuron to different stimuli.[13,31] Lifetime sparseness is closely related to the selectivity of a neuron, as the more selective a neuron is to a given stimulus feature the less frequent it will be active. Note that in computer vision literature sparseness generally refers to the population sparseness of the network. To avoid confusion, when we mention sparseness (or sparsity) in this review, we are referring to population sparseness, the proportion of active to inactive neurons at a given time.

Although it is poorly understood how the cortex enforces sparsity and selectivity in the sensory processing stream, sparse coding has multiple advantages that make it an attractive strategy for our brain. Research has shown that sparse coding increases memory capacity and reduces the number of units required to form a representation of a given stimulus.[30,32] A reduction in the proportion of active neurons reduces the amount of energy required to process the constant stream of information.[33] This suggests that sparse coding might be an evolutionary attractive strategy for the brain.

There has been some debate on the relation between sparseness and selectivity. The general intuition is, that when neurons in a network are more selective to specific features, fewer neurons are active at any given time. This would suggest a causal connection between sparseness and selectivity. When observing object selectivity and population sparseness in a set of neurons in the monkey IT area, Franco et al.[34] found strikingly similar values (77%) for both measures. Similarly, presenting natural images increased selectivity and population sparseness in Macaque V1 neurons.[35] On the contrary, when analyzing three different biologically inspired feature coding models, Wilmore and Tolhurst[31] proved that selectivity is not an informative measure of population sparseness, suggesting that the two measures are not causally linked. In support of Wilmore and Tolhurst findings, Berkes et al.[14] presented two primary visual cortex experiments that undermine the assumption that sparse coding is optimal for visual processing: (1) selectivity and sparseness decreased in ferret V1 in later developmental stages and for learned stimuli; (2) selectivity and sparseness increased in rats transitioning from awake to anesthetized state.[14] In both cases, the decrease in these two measures appeared in states that are expected to show less optimal sensory processing. However, there could be other reasons for these changes that are not covered by these models or confounded by cortical state transitions. Overall, the literature on the relation between selectivity and sparseness and its importance for object recognition is inconclusive. Large scale population recordings and computational models of visual processing could help us better understand these processes.

L2 minimization

An important aspect of successfully trained computer vision models is the ability to generalize well from training examples to new data points. When finding the solution to a system using a large set of parameters, such as in deep learning, minimizing the error on the training can lead to extreme parameter values. While they might fit the data with minimal error, this can be detrimental to the prediction of new samples. Restricting the parameters by introducing a regularization term to the error function is one way to address this problem.

L2 regularization finds widespread use in machine learning algorithms to improve generalization to testing data. Also known as Tikhonov regularization, this procedure has its origin in regression systems and has since been widely adopted in other machine learning applications.[11,36,37] L2 regularization adds the sum of the squared parameter values to the error function of the networks and thereby encourages lower values for the fitting parameters:

$$C^* = C + \lambda \sum_{i}^{N} w_i^2 \tag{1}$$

where C is the original cost function, λ is the hyperparameter that determines the impact of the regularizer and w are the parameters of the network. This results in a solution with smaller parameter values that better generalizes from training to testing data.[38,39] In overparameterized deep architectures the number of trainable parameters exceeds the number of samples in the input data. Here, L2 regularization encourages units to extract relevant features from the input. As training and testing data share features, this feature representation generalizes better to new samples.

Employing L2 regularization in neural networks creates units that have similarities with neurons in early visual areas. They respond selectively to simple features such as bars, gradients and bars[29] (Figure 1). Application in deep learning models resulted in later layers of the network encoding contour shapes comparable to those encoded by neurons in V2.[40] Restraining the parameter weights also affects the population sparseness of the units in the network. Before we can investigate how L2 regularization affects sparsity, we need to summarize different ways to quantify sparsity. Strictly speaking, population sparseness is the proportion of units in a system that are active at a given time. In artificial neural networks, the simplest way to calculate sparseness is to count the number of active and non-active units and take an average over all samples. Problematically, this approach to sparsity ignores the overall shape of the activity distribution, as it solely accounts for on or off states of neurons. However, a network with generally weak activity should arguably considered to be sparser than a network with generally strong activity. For an elaborate discussion on different methods for measuring sparsity in a population refer to Hurley et al.[41] L2 regularization results in a large proportion of units with activation values close to zero. As high parameter values are suppressed, the distribution of unit activation is much closer to 0.

In addition to creating a sparse network, L2 regularization is similar to training a network with noisy data to improve generalization.[42,43] Adding noise during training makes it difficult for the model to precisely fit the training data. In L2 regularization, the model must achieve a balance between optimally fitting the training samples while keeping its parameters small. Both approaches, push the model to generalize better to unlearned samples. It can be argued that L2 regularization creates more realistic training conditions that can be compared to natural object recognition and its noisy stimuli.

L1 regularization

In contrast to L2 regularization, L1 regularization (or LASSO) adds the absolute values of the weight parameters to the error function:

$$C^* = C + \lambda \sum_{i}^{N} |w_i| \tag{2}$$

where C is the original cost function, λ is the hyperparameter that determines the impact of the regularizer and $|w|$ are the absolute values of the parameters of the network. Similar to L2 regularization this suppresses high activation in the units of the model. In contrast to adding the squared parameters to the cost function, adding absolute values of the parameters equally punishes small values resulting in more non-active units and sparser populations. As fewer units are active to represent the relevant information in the stimuli, there is more pressure on the model to code for generally important features in the data.[44]

One of the effects of this is that L1 regularization is more efficient at classifying stimuli in a rotational invariance problem compared to L2 minimization. When increasing the number of

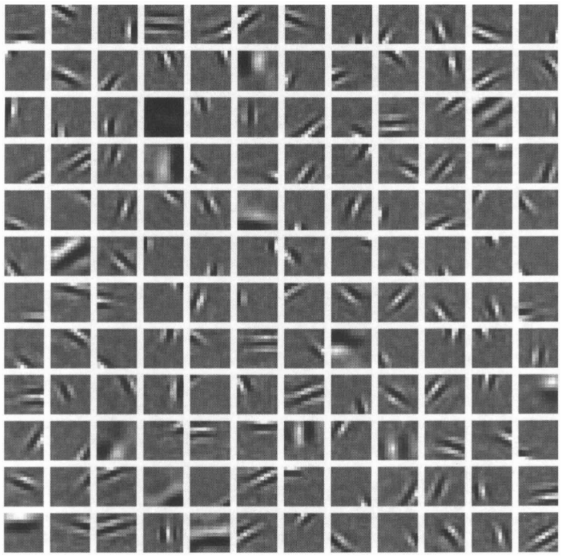

Figure 1: Visualization of the features learned by a sparse coding regime. The learned features are comparable to the preferences of V1 orientation selective neurons. Adapted from Olshausen et al.[29]

data points, the L2 regularized network required a linear increase in parameters to achieve comparable classification performance. In contrast, by employing L1 regularization in the same architecture, only a logarithmic increase of parameters was necessary.[45] The improvement in efficiency can be linked to the promotion of sparsity by the L1 regularizer. As a result, the network is more likely to only select the features that best explain the training data. These features will likely be shared with the testing data, therefore, improving the ability of the network to generalize.

However, the reliance on a small number of units to encode the information of a stimulus can lead to unstable representations. A noisy signal might be misinterpreted by the network because crucial units are not activated. L1 regularization discourages small activations more strongly than L2 regularization. As a result, there is a smaller quantity of units that carry partial information about the stimulus that might help to compensate for the errors of other units.

To address which of the regularizers, L1 or L2, makes more sense in a biological setting, we will investigate the plausibility of synapses in the cortex being restricted in similar ways and the effects on network characteristics. The weights of artificial neural nets can be compared to the connections between neurons in the cortex. Glutamate activity, the synaptic transmission of action potentials, is the major energy consumption of the brain with about 80%.[33,46,47] This acts as an evolutionary pressure on the brain to reduce the number and strength of connections (while maintaining performance). Both L1 and L2 regularization have comparable effects. The major difference between the two regularization methods expresses for very low and very large weight values (Figure 2). L1 regularization implements a linear relation between the strength of a connection and its costs for the network. The quadratic nature of L2 regularization less heavily punishes very weak connections compared to L1 regularization. Although weaker connections require less glutamate transmission, it is unlikely that weak connections require less than linearly lower costs (see Figure 2 x-range 0-1), as there are baseline costs for maintaining axon terminals between two neurons.[48] The cost of even weak connections under L1 regularization often leads to their extinction, as they contribute little to the computation. Removing these weak connections would result in lower maintenance costs for the cortex.[49] As the cost of connections in L2

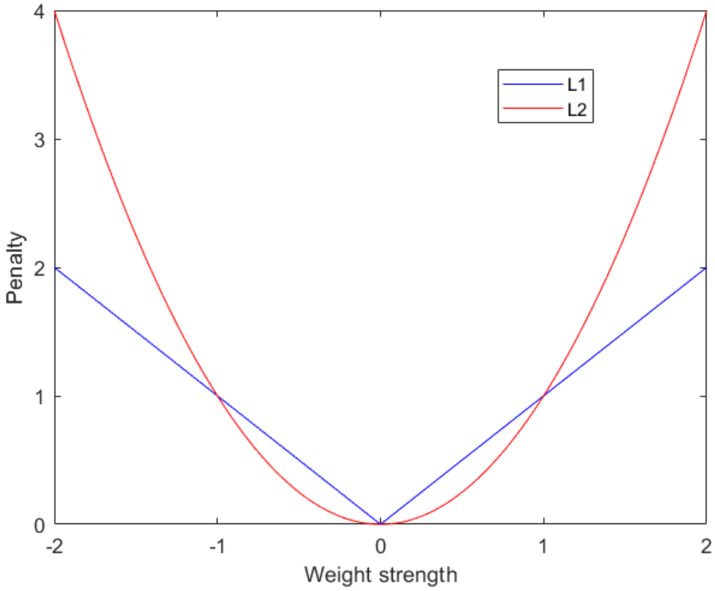

Figure 2: Visualization of the cost function of L1 and L2 regularizers. These function map the strength of a weight to the penalty added to the optimization algorithm. Blue: L1 regularization (equ. 2); Red: L2 regularization (equ. 1). The two function diverge for very low (0-1) and very high (¿1) weight strengths

constrained networks grows polynomial with their strength, strong connections are highly discouraged. Mapping the connectivity strength of pyramidal neurons in the rat visual cortex revealed a distribution with long tail.[50] This long tail is less plausible under the conditions of L2 regularization, as the weights in the long tail would carry large costs. Both examples suggest L1 regularization as the more likely candidate for producing comparable sparsity distributions.

However, there is a drawback to L1 regularizers that might be detrimental for object recognition in biological settings. As mentioned previously, L1 performs better in the rotational invariance problem as feature selection is more pronounced than under the L2 regime. Although this makes it easier to match a new input to an already trained object under a different context, such as different rotation or lightning, the algorithm is less stable to outliers and might struggle with new objects. The small weights in a L2 regularized network could help to find the closest matching pair under the trained set and thus be better suited for real-world circumstances with a constant stream of new objects and changes of visual conditions. However, our brain might be less troubled by this problem as it could potentially be solved by integrating slightly varying visual inputs over time and by the addition of contextual information. This idea has not been explored in artificial intelligence and might require networks with a temporal component to integrate information over time.

Dropout

Another way to prevent overfitting is to apply dropout. During training, neurons have a chance to be temporarily removed from the network along with all their connections.[51] After training, the parameters are normalized by the probability of dropout to retain the same total parameter size. Applied to the standard machine vision task like the MNIST dataset, the dropout network achieved lower training error than a comparable network with other regularization methods, such as L1 and L2.

To analyze effects on coding characteristics, Srivastava et al.[51] trained two networks with the same architecture, a single hidden layer with 256 units, to classify the MNIST dataset. While one network was trained conventionally, the other employed 50% dropout on the hidden units in the training phase. During the testing phase with dropout no longer in effect, the modified network showed sparser activation of its units than the conventional network (Figure 3).

The authors also compared the features that the units in these two networks encoded. The network without dropout showed no discernible patterns, whereas the units in the dropout network coded for image features such as ”edges strokes and spots in parts of the images”[51] (Figure 3 (C-D)). The authors argue that the randomization of the dropout forces units in the network to code for relevant features. When each unit has a random chance to be dropped, they cannot rely on the presence of each other and must perform independently to optimize performance. As a result, each unit must find general features of the input data to represent. This reduces co-dependencies of units and improves generalization to new samples.[52]

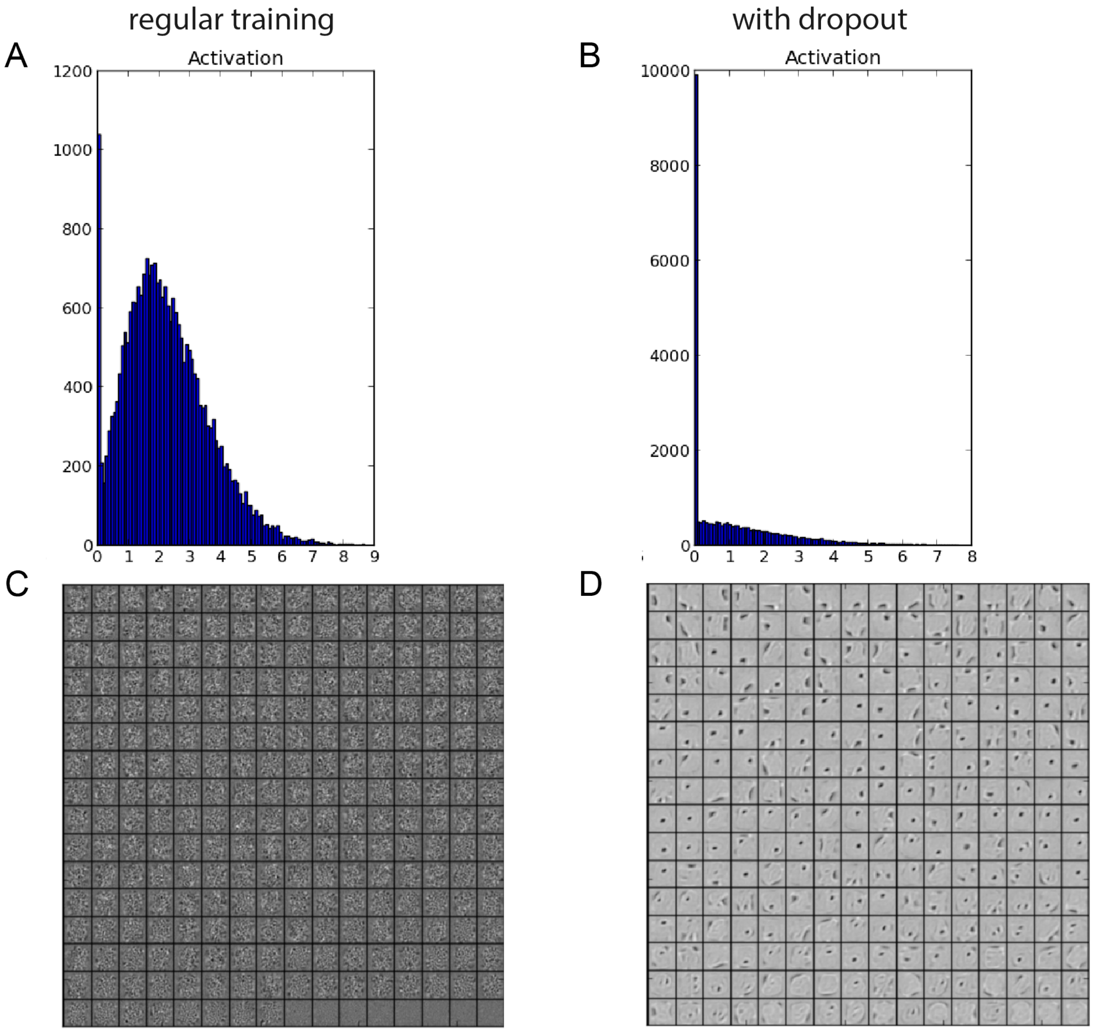

Figure 3: **Upper panel** Histogram of the unit activations of the same network without (A) and with (B) dropout applied. **Lower panel** Visualization of the features encoded by the 256 units in a single hidden layer without (C) and with (D) dropout applied. Adapted from Srivastava et al.[51]

This argument lends itself well to be compared to sensory processing in the cortex. Spiking activity has inherently random features as it follows Poisson processes.[53–55] Training with dropout can be compared to these noisy conditions. Similar to units in a dropout network, neurons in the brain cannot rely on the presence of another neuron to adjust for mistakes or co-dependently code for a feature. Dropout causes artificial neural networks to be closer to the biological noisy conditions of neuron population activity by introducing a random aspect beyond the random initialization of the weights. As this method improves generalization and promotes relevant feature selection it suggests that the semi-random behavior of neurons in the visual stream is an important component for its processing capabilities.

However, there is a caveat in comparing dropout networks with the cortical system. A network with dropout changes from training to testing. While some units are left out during training, the network can make full use of all connections in the testing phase. There is no evidence that suggests that the cortical structure changes similarly between the two phases. On the contrary, research into the developmental changes in the rat brain shows the opposite effect. The synaptic connectivity of neurons reaches peak levels at a juvenile stage, after which it gradually decreases until reaching a stable point in adulthood.[56,57] Accordingly, neurons are more interconnected during development, as opposed to the dropout regime where the training phase progresses with fewer neurons compared to the testing. This indicates that dropout does not lend itself well as a model for comparing developmental stages in the brain.

ReLU

Previously, we described different forms of regularizers that create sparse codes. In this next section, we will be analyzing how a particular type of activation function affects coding in artificial neural networks. The rectifier linear units (ReLU) finds widespread application in artificial intelligence and (Figure 4) replaced other activation functions such as the sigmoid or logistic. Hahnloser et al.[58] formulated the ReLU in the framework of spiking activity of neurons. It is based on two major characteristics of neurons in the visual processing stream, ”multistability” and ”analog response”. These neurons have active and silent states (multistability) while simultaneously being able to exhibit graded responses depending on their stimulus preference (analog response) (Figure 4 A).

Formally the ReLU is defined by a set of equations:

$$f(x) = \begin{cases} 0 & x <= 0 \\ x & x > 0 \end{cases} \tag{3}$$

where x is the input to the neuron.

A major advantage of the rectifier activation function is that it naturally develops sparsity in a network. With random, uniform initialization of the weights, around 50% of units have non-zero activation before training.[59] Thus, similar to the L1, L2, and dropout regularizers the ReLU promotes sparsity.

To prevent run-away weights Glorot et al.[59] added L1 regularization to the optimization algorithm. This combination resulted in a new minimum for the MNIST classification task with an error of 1.43% in a 3 hidden layer (n = 1000) network. Additionally, Glorot et al. analyzed the interaction between sparsity and the performance on image classification using MNIST. By testing different weights for the L1 penalty (λ in equ. 2), they showed that optimal sparsity (percentage of non-units) lied around 80% (Figure 4 B).[59] However, the network relied on L1 regularization to create optimal predictions of the testing data. This prevents us from drawing direct conclusions between the use of the ReLU and its effects on sparsity in the network. Additionally, there is a discrepancy between the sparsity level that showed optimal performance in the ReLU network (around 80%) and sparsity in early cortical areas (95-98% in V1).[59,60] However, the big structural differences between the two systems make a direct comparison difficult.

Regardless of these concerns, ReLUs offer an interesting point for further exploration as their development was built on the characteristics of neurons in the visual cortex. Multistability and the analog response of these neurons is directly related to their selectivity to particular stimulus fea-

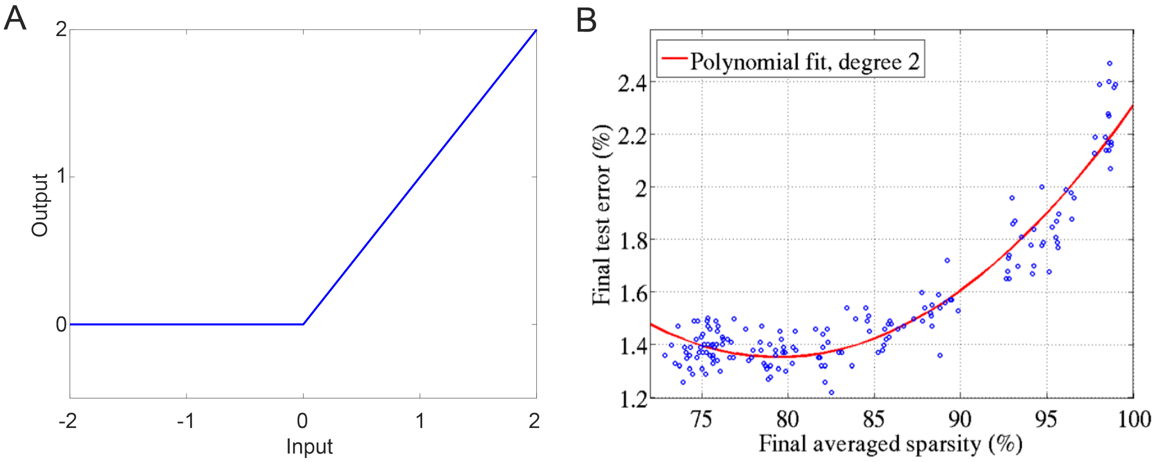

Figure 4: **A** Activation function of a ReLU. **B** Interaction between average sparsity of a 3 hidden layer network with ReLUs and the test error. Each dot represent a different training run of the same network architecture with varying L1 weight. Red line indicates 2nd degree polynomial fit. Optimal sparsity is around 80%. Adapted from Glorot et al. 2011.[59]

tures. This can best be explained with an example. Neurons in the primary visual cortex respond primarily to simple features such as bars and are not activated by other image components such as color (multistability). These neurons have a preferred orientation and the closer the stimulus is to that orientation the more strongly the neuron responds (graded response). The ReLU is a form of hard-coding these features into the response properties of a network. The fact that the implementation of this activation function results in sparser unit activations suggests that these features might be an important factor for creating sparse populations in the cortex. In line with this reasoning, Ukita[61] has found that the low-level features selectivity in these networks is crucial for generalization. In other work Spratling et al.[62] showed, that sparse algorithms are an important aspect for constructing feature selective units and are a strategy that higher order cortical regions could employ to perform classification. These findings provide evidence for a causal relation between sparsity and selectivity and their importance for the ability of a network to generalize from trained to new samples.

Discussion

The ability to generalize from training data to testing data is an important aspect of evaluating the performance of computer vision algorithms. Regularization is a powerful tool for preventing overfitting in neural networks and promoting generalization. L1- and L2-regularization employ similar strategies by adding constraints to the optimization algorithm that are related to parameter size. This method of regularization can be compared to the evolutionary pressure on the cortex to reduce energy consumption. The next form of regularization we analyzed, is dropout. A network employing dropout trains with randomly removing units and their connections during the learning phase. This adds noise to the processing, which creates conditions that can be compared to the noisy firing patterns of neurons in the cortex. ReLUs are not a form of regularization but have a specific activation function that change the processing characteristics of the network. In comparison to the earlier used sigmoid or logistic activation functions, the ReLU more closely mimics the tuning properties of early visual neurons. Similar to the discussed regularizers, the ReLU promotes sparse coding and creates feature selective units. The fact that all the presented methods similarly affect population sparseness and feature selectivity simultaneously, suggests that there exists a functional link between the two. This stands in contrast to research that showed that enforcing selectivity in an artificial neural network was not sufficient to evoke a sparse population. It remains challenging to understand how these two characteristics are related to each other and what functional role they play in the visual processing stream of the cortex.

Additionally, there are major differences between biological and artificial neural networks that make it difficult to transfer conclusions from one model to the other. For instance, the optimization algorithms in machine learning are difficult to consolidate with our understanding of learning in the brain. Current computer vision algorithms for object recognition employ back-propagation, which relies on symmetric weights between units and that the computed error value can travel from the output layer back through the network. These are not a plausible assumption in a biological network where synapses are one directional and the concept of a traveling error gradient is incomprehensible. Nonetheless, work by Scellier and Bengio[63, 64] has shown, that learning based on an energy function and prediction error, is conceptually similar to back-propagation. This might offer a possible explanation for bridging the gap between learning in computer vision and in cortical networks.

Another method of bringing neuroscience and computer vision closer together is the use of spiking neural networks (SNNs). Units in SNNs have discrete spike events, a big improvement in regard to mimicking biological neurons compared to other artificial neural networks. Training these networks is challenging because traditional optimization methods such as back-propagation and gradient descent are difficult to implement.[65] A possible learning mechanism for these networks is Spike Timing Dependent Plasticity (STDP) which modulates connection strength based on the relative time of spikes in two connected neurons. This method is the core mechanism of synaptic plasticity[66–69] and as such learning in the brain. Although struggling to compete against other computer vision models in performance and speed due to their complexity, there are specific problems that the SNNs excel at. Interestingly, in many cases, these are tasks at which humans outperform artificial neural networks, e.g. robust invariant object recognition.[70, 71] SNNs carry another major benefit that makes them an interesting subject for future research. They introduce a temporal component to computer vision algorithms. This opens new opportunities to investigate the temporal dynamics of visual processing. The closer resemblance with the V1-IT processing stream is of particular interest. Analysis into population sparsity and feature selectivity in SNNs might clarify how these two characteristics are connected and how they contribute to sensory processing. Additionally, by finding better and more efficient ways to train these networks, we might be able to develop more suitable algorithms for challenging tasks.

We have shown that sparseness is an important aspect of effective models in computer vision. Sparse networks are useful for a wide range of applications, such as autonomous cars[72] or biomedical image processing.[73] Yet, it is hard to object that human performance surpasses them in some aspects, such as adaptation to novel situations or the integration of contextual information. Analyses into which properties of the cortical processing stream lay the foundations for these functions might uncover mechanics that offer new strategies to solve these kinds of tasks. We expect that future collaboration between neuroscience and computer vision will further benefit both fields. Neuroscience can provide solutions to problems in artificial intelligence and analysis of artificial neural networks will be able to bring further insight into cortical processing.

References

1. Hopfield, J. J. Neural networks and physical systems with emergent collective computational abilities. *Proceedings of the national academy of sciences* **79**, 2554–2558 (1982).

2. Hassabis, D., Kumaran, D., Summerfield, C. & Botvinick, M. Review Neuroscience-Inspired Artificial Intelligence. *Neuron* **95** (2017).

3. Yamins, D. L. K. & DiCarlo, J. J. Using goal-driven deep learning models to understand sensory cortex. *Nature Neuroscience* **19**, 356–365 (2016).

4. Khaligh-Razavi, S.-M. & Kriegeskorte, N. Deep Supervised, but Not Unsupervised, Models May Explain IT Cortical Representation. *PLoS Computational Biology* **10**, e1003915 (2014).

5. Khaligh-Razavi, S.-M., Henriksson, L., Kay, K. & Kriegeskorte, N. Explaining the hierarchy of visual representational geometries by remixing of features from many computational vision models. *bioRxiv* 009936 (2014).

6. Zhou, Z. & Firestone, C. Humans can decipher adversarial images. *Nature Communications* **10**, 1334 (2019).

7. Bashivan, P., Kar, K. & DiCarlo, J. J. Neural population control via deep image synthesis. *Science* **364**, eaav9436 (2019).

8. Kriegeskorte, N. Deep Neural Networks: A New Framework for Modeling Biological Vision and Brain Information Processing. *Annual Review of Vision Science* **1**, 417–446 (2015).

9. Medathati, N. V. K., Neumann, H., Masson, G. S. & Kornprobst, P. Bio-inspired computer vision: Towards a synergistic approach of artificial and biological vision. *Computer Vision and Image Understanding* **150**, 1–30 (2016).

10. Chalupa, L. M., Werner, J. S. & Barnstable, C. *The visual neurosciences*, vol. 1 (MIT press Cambridge, MA, 2004).

11. LeCun, Y., Bengio, Y. & Hinton, G. Deep learning. *Nature* **521**, 436–444 (2015).

12. Hubel, D. H. & Wiesel, T. N. Receptive fields of single neurones in the cat's striate cortex. *The Journal of physiology* **148**, 574–591 (1959).

13. Willmore, B. D. & King, A. J. Auditory cortex: representation through sparsification? *Current Biology* **19**, R1123–R1125 (2009).

14. Berkes, P., White, B. & Fiser, J. No evidence for active sparsification in the visual cortex. In *Advances in neural information processing systems*, 108–116 (2009).

15. Hawkins, D. M. The problem of overfitting. *Journal of chemical information and computer sciences* **44**, 1–12 (2004).

16. Goodale, M. A. & Milner, A. D. Separate visual pathways for perception and action. *Trends in neurosciences* **15**, 20–25 (1992).

17. Mishkin, M., Ungerleider, L. G. & Macko, K. A. Object vision and spatial vision: two cortical pathways. *Trends in Neurosciences* **6**, 414–417 (1983).

18. Ungerleider, L. G. Two cortical visual systems. *Analysis of visual behavior* 549–586 (1982).

19. Tanaka, K. Inferotemporal Cortex and Object Vision. *Annual Review of Neuroscience* **19**, 109–139 (1996).

20. Tyler, L. K. *et al.* Objects and Categories: Feature Statistics and Object Processing in the Ventral Stream. *Journal of Cognitive Neuroscience* **25**, 1723–1735 (2013).

21. Hegdé, J. & Van Essen, D. C. Selectivity for complex shapes in primate visual area v2. *Journal of Neuroscience* **20**, RC61–RC61 (2000).

22. Anzai, A., Peng, X. & Van Essen, D. C. Neurons in monkey visual area V2 encode combinations of orientations. *Nature Neuroscience* **10**, 1313–1321 (2007).

23. Pasupathy, A. & Connor, C. E. Responses to Contour Features in Macaque Area V4. *Journal of Neurophysiology* **82**, 2490–2502 (1999).

24. Schwartz, E. L., Desimone, R., Albright, T. D. & Gross, C. G. Shape recognition and inferior temporal neurons. *Proceedings of the National Academy of Sciences of the United States of America* **80**, 5776–8 (1983).

25. Tanaka, K. Mechanisms of visual object recognition: monkey and human studies. *Current Opinion in Neurobiology* **7**, 523–529 (1997).

26. Tanaka, K., Saito, H., Fukada, Y. & Moriya, M. Coding visual images of objects in the inferotemporal cortex of the macaque monkey. *Journal of neurophysiology* **66**, 170–89 (1991).

27. Rust, N. C. & Dicarlo, J. J. Selectivity and tolerance (”invariance”) both increase as visual information propagates from cortical area V4 to IT. *The Journal of neuroscience* **30**, 12978–95 (2010).

28. Kovács, G., Vogels, R. & Orban, G. A. Selectivity of macaque inferior temporal neurons for partially occluded shapes. *The Journal of neuroscience : the official journal of the Society for Neuroscience* **15**, 1984–97 (1995).

29. Olshausen, B. A. & Field, D. J. Sparse coding with an overcomplete basis set: A strategy employed by v1? *Vision research* **37**, 3311–3325 (1997).

30. Olshausen, B. A. & Field, D. J. Sparse coding of sensory inputs. *Current Opinion in Neurobiology* **14**, 481–487 (2004).

31. Willmore, B. & Tolhurst, D. Characterizing the sparseness of neural codes. *Network: Computation in Neural Systems* **12**, 255–270 (2001).

32. Hinton, G. E., McClelland, J. L., Rumelhart, D. E. *et al. Distributed representations* (Carnegie-Mellon University Pittsburgh, PA, 1984).

33. Attwell, D. & Laughlin, S. B. An Energy Budget for Signaling in the Grey Matter of the Brain. *Journal of Cerebral Blood Flow & Metabolism* **21**, 1133–1145 (2001).

34. Franco, L., Rolls, E. T., Aggelopoulos, N. C. & Jerez, J. M. Neuronal selectivity, population sparseness, and ergodicity in the inferior temporal visual cortex. *Biological Cybernetics* **96**, 547–560 (2007).

35. Vinje, W. E. & Gallant, J. L. Sparse coding and decorrelation in primary visual cortex during natural vision. *Science* **287**, 1273–1276 (2000).

36. Tikhonov, A. N. & Arsenin, V. I. *Solutions of ill-posed problems*, vol. 14 (Winston, Washington, DC, 1977).

37. Golub, G. H., Hansen, P. C. & O'Leary, D. P. Tikhonov Regularization and Total Least Squares. *SIAM Journal on Matrix Analysis and Applications* **21**, 185–194 (1999).

38. Frank, L. E. & Friedman, J. H. A statistical view of some chemometrics regression tools. *Technometrics* **35**, 109–135 (1993).

39. Hoerl, A. E. & Kennard, R. W. Ridge Regression: Biased Estimation for Nonorthogonal Problems. *Technometrics* **12**, 55–67 (1970).

40. Lee, H., Ekanadham, C. & Ng, A. Y. Sparse deep belief net model for visual area v2. In *Advances in neural information processing systems*, 873–880 (2008).

41. Hurley, N. & Rickard, S. Comparing Measures of Sparsity. *IEEE Transactions on Information Theory* **55**, 4723–4741 (2009).

42. Webb, A. Functional approximation by feedforward networks: a least-squares approach to generalization. *IEEE Transactions on Neural Networks* **5**, 363–371 (1994).

43. Bishop, C. M. Training with Noise is Equivalent to Tikhonov Regularization. *Neural Computation* **7**, 108–116 (1995).

44. Tibshirani, R. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)* **58**, 267–288 (1996).

45. Ng, A. Y. Feature selection, l 1 vs. l 2 regularization, and rotational invariance. In *Proceedings of the twenty-first international conference on Machine learning*, 78 (ACM, 2004).

46. Raichle, M. E. & Gusnard, D. A. Appraising the brain's energy budget. *Proceedings of the National Academy of Sciences of the United States of America* **99**, 10237–9 (2002).

47. Sibson, N. R. *et al.* Stoichiometric coupling of brain glucose metabolism and glutamatergic neuronal activity. *Proceedings of the National Academy of Sciences* **95**, 316–321 (1998).

48. Sotelo, C. Purkinje Cell Ontogeny: Formation and Maintenance of Spines. *Progress in Brain Research* **48**, 149–170 (1978).

49. Rehn, M. & Sommer, F. T. A network that uses few active neurones to code visual input predicts the diverse shapes of cortical receptive fields. *Journal of computational neuroscience* **22**, 135–146 (2007).

50. Song, S., Sjöström, P. J., Reigl, M., Nelson, S. & Chklovskii, D. B. Highly Nonrandom Features of Synaptic Connectivity in Local Cortical Circuits. *PLoS Biology* **3**, e68 (2005).

51. Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting. *The journal of machine learning research* **15**, 1929–1958 (2014).

52. Hinton, G. E., Srivastava, N., Krizhevsky, A., Sutskever, I. & Salakhutdinov, R. R. Improving neural networks by preventing co-adaptation of feature detectors. *arXiv preprint arXiv:1207.0580* (2012).

53. Stein, R. Some Models of Neuronal Variability. *Biophysical Journal* **7**, 37–68 (1967).

54. Gerstein, G. L. & Mandelbrot, B. Random walk models for the spike activity of a single neuron. *Biophysical journal* **4**, 41–68 (1964).

55. Baddeley, R. *et al.* Responses of neurons in primary and inferior temporal visual cortices to natural scenes. *Proceedings of the Royal Society of London* **264**, 1775–1783 (1997).

56. Semple, B. D., Blomgren, K., Gimlin, K., Ferriero, D. M. & Noble-Haeusslein, L. J. Brain development in rodents and humans: Identifying benchmarks of maturation and vulnerability to injury across species. *Progress in neurobiology* **106-107**, 1–16 (2013).

57. Micheva, K. D. & Beaulieu, C. Quantitative aspects of synaptogenesis in the rat barrel field cortex with special reference to GABA circuitry. *The Journal of Comparative Neurology* **373**, 340–354 (1996).

58. Hahnloser, R. H. R., Sarpeshkar, R., Mahowald, M. A., Douglas, R. J. & Seung, H. S. Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit. *Nature* **405**, 947–951 (2000).

59. Glorot, X., Bordes, A. & Bengio, Y. Deep sparse rectifier neural networks. In *Proceedings of the fourteenth international conference on artificial intelligence and statistics*, 315–323 (2011).

60. Lennie, P. The physiology of color vision. *The science of color* **2**, 217–242 (2003).

61. Ukita, J. Causal importance of orientation selectivity for generalization in image recognition (2018).

62. Spratling, M. W. Classification using sparse representations: a biologically plausible approach. *Biological cybernetics* (2013).

63. Scellier, B. & Bengio, Y. Equilibrium Propagation: Bridging the Gap between Energy-Based Models and Backpropagation. *Frontiers in Computational Neuroscience* **11**, 24 (2017).

64. Scellier, B. & Bengio, Y. Equivalence of Equilibrium Propagation and Recurrent Backpropagation. *Neural Computation* **31**, 312–329 (2019).

65. Tavanaei, A., Ghodrati, M., Kheradpisheh, S. R., Masquelier, T. & Maida, A. Deep learning in spiking neural networks. *Neural Networks* **111**, 47–63 (2019).

66. Caporale, N. & Dan, Y. Spike Timing–Dependent Plasticity: A Hebbian Learning Rule. *Annual Review of Neuroscience* **31**, 25–46 (2008).

67. Dan, Y. & Poo, M.-M. Spike Timing-Dependent Plasticity: From Synapse to Perception. *Physiological Reviews* **86**, 1033–1048 (2006).

68. Dan, Y. & Poo, M.-m. Spike Timing-Dependent Plasticity of Neural Circuits. *Neuron* **44**, 23–30 (2004).

69. Song, S., Miller, K. D. & Abbott, L. F. Competitive Hebbian learning through spike-timing-dependent synaptic plasticity. *Nature Neuroscience* **3**, 919–926 (2000).

70. Kheradpisheh, S. R., Ganjtabesh, M. & Masquelier, T. Bio-inspired unsupervised learning of visual features leads to robust invariant object recognition. *Neurocomputing* **205**, 382–392 (2016).

71. Kheradpisheh, S. R., Ganjtabesh, M., Thorpe, S. J. & Masquelier, T. STDP-based spiking deep convolutional neural networks for object recognition. *Neural Networks* **99**, 56–67 (2018).

72. Hancock, P. A., Nourbakhsh, I. & Stewart, J. On the future of transportation in an era of automated and autonomous vehicles. *Proceedings of the National Academy of Sciences of the United States of America* **116**, 7684–7691 (2019).

73. De Fauw, J. *et al.* Clinically applicable deep learning for diagnosis and referral in retinal disease. *Nature medicine* **24**, 1342 (2018).