Significance of feedforward architectural differences between the ventral visual stream and DenseNet

Anonymous Author(s)
Affiliation
Address
email

Abstract

There are many differences between convolutional networks and the ventral visual streams of primates. For example, standard convolutional networks lack recurrent and lateral connections, cell dynamics, etc. However, their feedforward architectures are somewhat similar to the ventral stream, and warrant a more detailed comparison. A recent study found that the feedforward architecture of the visual cortex could be closely approximated as a convolutional network, but the resulting architecture differed from widely used deep networks in several ways. The same study also found, somewhat surprisingly, that training the ventral stream of this network for object recognition resulted in poor performance. We examined the performance of this network in more detail. In particular, we made a number of changes to the ventral-stream-based architecture, to make it more like a DenseNet, and tested performance at each step. We chose DenseNet because is has a high BrainScore, and because is has some cortex-like architectural features such as large in-degrees and long skip connections. Most of the changes (which made the cortex-like network more like DenseNet) improved performance. Further work is needed to better understand these results. One possibility is that details of the ventral-stream architecture may be ill-suited to feedforward computation, simple processing units, and/or backpropagation, which could suggest differences between the way high-performance deep networks and the brain approach core object recognition.

1 Introduction

Deep convolutional networks trained for object recognition have a number of things in common with the primate ventral stream, including architectural similarities (e.g. spatially localized connections), comparable performance, and closely related representations [1]. There are also many differences, notably related to cell and network dynamics and development of representations over time. However, feedforward convolutional networks appear to be promising abstract models of core object recognition. Can convolutional networks be used to model the signals and transformations of core object recognition in detail? A recent study developed a data-driven convolutional network that was consistent with many architectural details of primate visual cortex [2]. However, that study reported performance far below state of the art when the ventral stream was trained on CIFAR-10. Here we confirm this result, and investigate whether there is a specific feature of the cortex-like architecture that limits performance. We find rather that multiple architectural details are responsible for the performance gap. This may suggest that the feedforward architecture of the ventral stream is not particularly well suited to core object recognition in the idealized context of standard deep-network training.

Submitted to 33rd Conference on Neural Information Processing Systems (NeurIPS 2019). Do not distribute.

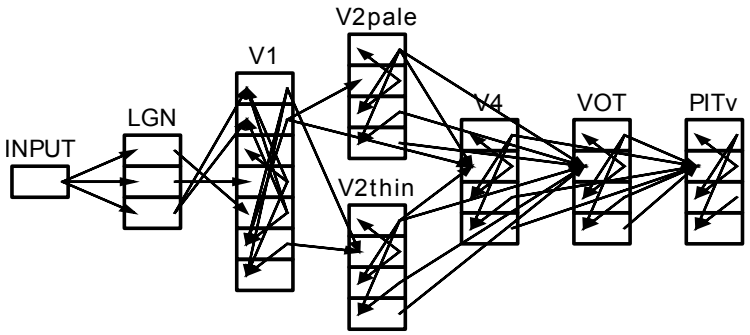

Figure 1: Structure of the ventral-stream-like network. Except where mentioned, parameters were taken from [2]. The boxes within each area indicate sub-populations with distinct connections. For most cortical areas, these are layers L2/3, L4, L5, and L6. For V1, these are L2/3(blob), L2/3(interblob), L4B, L4Cα, L4Cβ. For LGN, they are parvo, magno, and koniocellular divisions.

2 Ventral-Stream Network

We experimented with a simplified sub-network of the macaque single-hemisphere visual cortex architecture of [2]. This sub-network spans much of the ventral stream (Figure 1, including V1, V2 (thin and pale stripes), V4, VOT, and PITv. We used the retrograde-tracer measure, fraction labelled neurons extrinsic to the injection site (FLNe), to identify and omit the weakest inter-area connections (FLNe<0.02). To make training tractable, the number of neurons in each population was reduced by a factor of ten. The full visual resolution was taken to be 500x500 pixels, and smaller images from ImageNet and CIFAR-10 were taken to activate the central part of the visual field. L2/3 of PITv was connected to a final convolutional layer with 64 channels, and then either two 512-unit fully-connected layers (for CIFAR-10) or one 2048-unit fully-connected layer (for ImageNet), followed by a softmax classifier layer. Each linear layer in the model was followed by batch norm and ReLU layers. The code will be released on acceptance.

3 Results

[2] reported validation accuracy of 79% on CIFAR-10, for a similar ventral-stream sub-network that omitted connections with FLNe<0.15. Some training details were not reported. We trained our variation for 300 epochs, using SGD with momentum 0.9, starting with learning rate 0.1 and reducing it by 10x every 100 epochs. This resulted in validation accuracy of 84.59%. We also trained a standard DenseNet (github.com/kuangliu/pytorch-cifar) using the same procedure, resulting in validation accuracy of 95.36%. To understand the basis of this performance gap, we created hybrid networks, with features of both the ventral-stream network (VSN) and DenseNet. The VSN has a wide range of kernel sizes, optimized to fill realistic receptive field sizes. In our first hybrid (H1), we set all kernel sizes of the VSN to 3x3. The VSN also has a wide range of sparsity, with some connections consisting mostly of zeros. In the second hybrid network (H2), in addition to using 3x3 kernels, we eliminated pixel-wise sparsity, and limited channel-wise sparsity so that at least half of the input channels were used in each connection. Thirdly (H3), we replaced each layer with a two-layer bottleneck module, specifically a 1x1-kernel layer followed by 3x3 layer with four times fewer channels. The number of channels in the second layer was 0.3x the original number of channels. Fourthly (H4), we replaced the LGN layers with a DenseNet-like 24-channel input stage, eliminated remaining sparsity, and organized the remaining layers into three blocks with DenseNet-like transition layers between them. Each block grouped layers with the same spatial resolution. Connections that spanned multiple blocks were replaced by connections with the transition layers. Fifthly (H5), we set the numbers of channels in each module within a given block equal to the block-wise average, emulating the block-wise constant "growth rate" parameter of DenseNet. Finally (H6), we added any missing connections within each block, introducing many connections that do not exist in the brain.

Most of these modifications improved performance (Table 1). Making the growth rate uniform resulted in a slight performance drop. Replacing simple layers with bottleneck modules sharply impaired performance, including on training data (83% correct), although performance recovered

Table 1: CIFAR-10 performance in hybrid ventral-stream/DenseNet architectures (details in text).

Architecture Variation	Validation Accuracy
Baseline ventral-stream network	84.59%
Modifications from baseline (cumulative)	
H1 (3x3 kernels)	85.33%
H2 (dense kernels)	88.09%
H3 (bottlenecks)	76.79%
H4 (single-resolution blocks)	93.15%
H5 (uniform growth rate)	92.5%
H6 (dense connections)	93.89%
DenseNet	95.36%

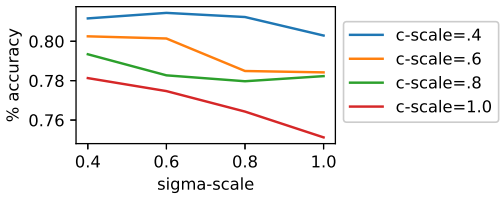

Figure 2: Effect of sparsity parameters c (channel-wise) and σ (element-wise) on performance. Different lines show CIFAR-10 performance of different ventral-stream networks with log-c multiplied by various scale factors. Larger c-scale means more sparsity. The horizontal axis indicates log-scaling of σ (Networks trained for 50 epochs with Adam; learning rate 0.001.)

in the next variation with transition layers. The bottleneck experiment was repeated with slight variations (i.e. a different training schedule, growth rate 0.2x rather than 0.3x the original number of channels, and a DenseNet-like input stage), and similar performance impairment was found. With these exceptions, hybrid networks performed better the more they had in common with DenseNet.

As discussed in [2], connection sparsity in the VSN is qualitatively distinct from various forms of sparsity in deep learning. We further explored the effect of sparsity on performance. Each connection in the VSN has two sparsity parameters, σ (pixel-wise probability of non-zero weight), and c (channel-wise probability of non-zero weights); the probability that a weight is non-zero is σc. We chose non-zero channels at random, and non-zero weights using SNIP [3]. The sparsity parameters vary over several orders of magnitude in different connections. We trained variations of the VSN in which we scaled the logarithms of these parameters (e.g. $\sigma = .0001$ with a scale factor of $.5$ would become $.01$). Unsurprisingly, reductions in sparsity tended to improve performance (Figure 2). So, physiologically realistic connection sparsity seems to be counter-productive in these deep-network models of core object recognition.

Representations in DenseNet are closely related to those in primate ventral stream [1]. However, we wondered whether, despite poorer performance of the VSN, it might have even more realistic representations due to its architecture. To explore this issue, we performed approximations of several experiments from the primate literature, with both a DenseNet-161 model pretrained on ImageNet (from torchvision.models), and a VSN trained on ImageNet, with c-scale=σ-scale=0.5. The VSN was trained with SGD for 90 epochs, with learning rate 0.1, decreasing by 10x every 30 epochs; best validation accuracy 63.14%(top-1)/84.97%(top-5). We recorded from the second-last ReLU layer that followed a convolutional layer in each network, and compared with published recordings from macaque inferotemporal cortex. The two networks had somewhat different representations, but neither was much more realistic (Figure 3).

4 Discussion

Architectural details of the ventral-stream model tended to impair core object recognition in our deep-learning setting. This may indicate inaccuracies in the architecture of [2]. Alternatively, it

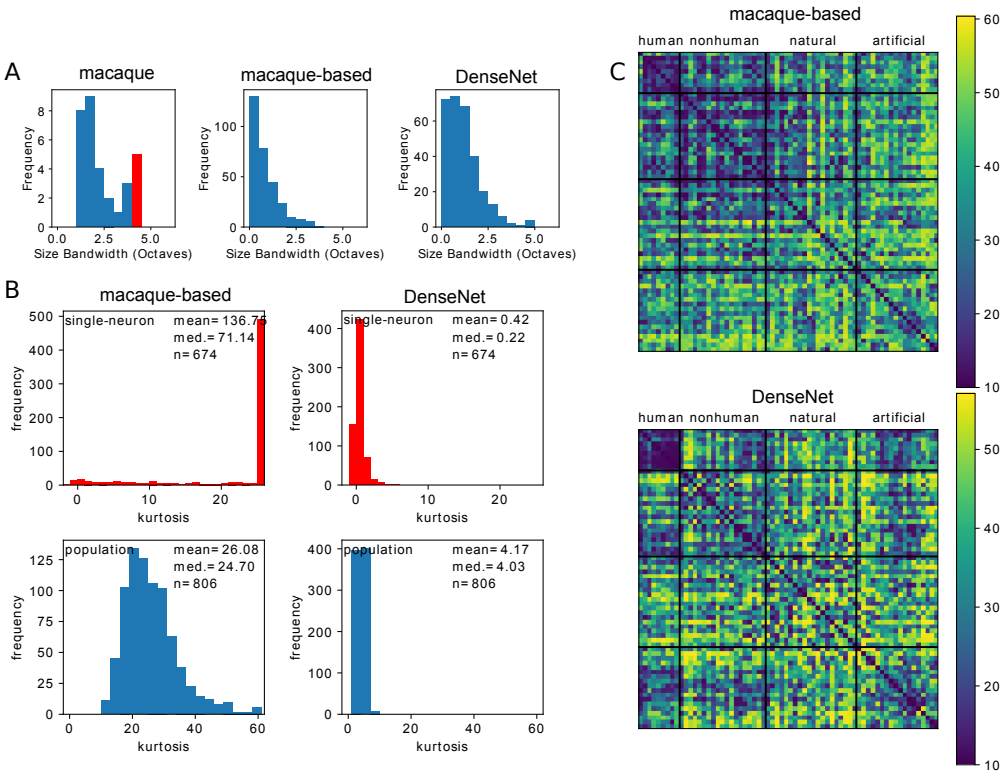

Figure 3: Examination of higher-level representations in the ventral-stream network and DenseNet. A, size-tuning bandwith. Macaque data replotted from [4] (red means ≥ 4). Both networks seem to have somewhat narrower size tolerances than monkeys. B, Distributions of single-unit selectivity (top) and population sparseness (bottom), with stimuli from [5]. In the monkey inferotemporal cortex data, mean selectivity is 3.5, mean sparseness is 12.51. The ventral-stream model has much higher means, and DenseNet has much lower means. C. Representational similarity, using a subset of images from [6]. Monkey cell data shows relatively low values throughout the lower-right quarter of this matrix (spanning non-animal natural and artificial images) [6], but neither of the deep networks does.

may suggest that the feedforward ventral-stream architecture is not well suited for such idealized settings, or even that it is not optimized specifically for core object recognition. The ventral stream architecture might be better suited for recurrent processing, unsupervised learning, and/or schemes such as predictive coding which may use sparse feedforward connections more efficiently.

References

[1] Martin Schrimpf, Jonas Kubilius, et al. Brain-score: which artificial neural network for object recognition is most brain-like? *BioRxiv*, page 407007, 2018.

[2] Bryan Tripp. Approximating the architecture of visual cortex in a convolutional network. *Neural Computation*, 31(8):1551–1591, 2019.

[3] Namhoon Lee, Thalaiyasingam Ajanthan, and Philip H S Torr. SNIP: Single-shot network pruning based on connection sensitivity. In *ICLR*, 2019.

[4] Minami Ito, Hiroshi Tamura, Ichiro Fujita, and Keiji Tanaka. Size and position invariance of neuronal responses in monkey inferotemporal cortex. *J Neurophysiol*, 73(1):218–226, 1995.

[5] Sidney R Lehky, Roozbeh Kiani, Hossein Esteky, and Keiji Tanaka. Statistics of visual responses in primate inferotemporal cortex to object stimuli. *J Neurophysiol*, 106(3):1097–117, sep 2011.

[6] Nikolaus Kriegeskorte. Relating population-code representations between man, monkey, and computational models. *Frontiers in Neuroscience*, 3(DEC):363–373, 2009.