Improving Neural Predictivity in the Visual Cortex with Gated Recurrent Connections

Anonymous Author(s)

Abstract

Computational models of vision have traditionally been developed in a bottom-up fashion, by hierarchically composing a series of straightforward operations - i.e. convolution and pooling. The aim was to emulate simple and complex cells in the visual cortex and resulted in the introduction of deep convolutional neural networks (CNNs). Nevertheless, evidence obtained with recent neuronal recording techniques suggests that the nature of the computations carried out in the ventral visual stream is not completely captured by current deep CNN models. To fill the gap between the ventral visual stream and deep models, several benchmarks have been designed and organized into the Brain-Score platform, granting a way to perform multi-layer (V1, V2, V4, IT) and behavioral comparisons between the two counterparts. In our work, we shift the focus to architectures that take into account lateral recurrent connections, a ubiquitous feature of the ventral visual stream, to devise adaptive receptive fields. Through recurrent connections, the input’s long-range spatial dependencies can be captured in a local multi-step fashion and, as introduced with Gated Recurrent CNNs (GRCNN), the unbounded expansion of the neuron’s receptive fields can be modulated through the use of gates. To increase the robustness of our approach and the biological fidelity of the activations, we employ specific data augmentation techniques in line with several of the scoring benchmarks. We find that forcing some form of invariance, through heuristics, resulted in better neural predictivity.

1 Introduction

As suggested by abundant physiological evidence, recurrent circuits are ubiquitous in the visual cortex [1][2][3] and in several areas of the mammalian brain [4]. In visual layers, the effect of lateral recurrent connections is believed to contribute to receptive fields adaptation [5][6][7][8]. More generally, there are many indications of the capability of the brain to modulate the processing of visual signals on the basis of their context [9][10][11]. Classical CNNs, despite showing many similarities with the ventral visual stream, lack the ability of context modulation. Trying to incorporate recurrent connections in a CNN architecture poses a problem because standard Recurrent Neural Networks (RNNs) are designed to process time varying sequences of inputs, while computer vision models deal with static inputs. To both circumvent this limit and try to design a more biologically plausible architecture, the basic idea is to introduce a form of recursion across neurons of the same layer. In this way it is possible to avoid the problem of static inputs, but still give to each neuron in a layer information about the activity of the surrounding neurons, allowing them to receive information about a larger part of the image. Similar solutions have been proposed in different forms by many authors, often with the aim of better
simulating the visual system [12, 3, 13]. It is worth noticing that the computational graphs of these
models, when unfolded in time, look like pure feedforward hierarchies of operations enriched with a
number of skip connections. This results in architectures that are very similar to residual networks
[14]. To achieve even better biological adherence, it is useful to add gates in between the recursive
computations, as proposed by [15]. In this way the neurons’ receptive fields are explicitly modeled
and modulated by the gates.

2 Model

The introduction of gates is a distinctive feature of GRCNN and is motivated in [15]. The role of
gates can be intuitively and qualitatively understood as an extra layer of computation resembling
an attention mechanism [16]. Gates are designed to give an output between 0 and 1 that multiplies
(pointwise) the activations in the recurrent convolutional layer. This means that, during training, the
set of weights associated with each gate evolves in such a way as to give the network the capability to
notice which parts of the image are relevant. In more detail, a GRCNN is composed by a feedforward
sequence of blocks called gated recurrent convolutional layers (GRCL). Every GRCL block computes
a recursion on its inputs, and, between each recurrent operation, the gate system modulates the
effective amount of forwarded information - see Figure 1. The equations describing the computations inside the GRCL are the following:

\[
\begin{align*}
    x_0 &= A_0(u) \\
    g_t &= B_t(x_{t-1}) + C(u), \quad t = 1, 2, \ldots, T \\
    x_t &= x_{t-1} + \sigma(g_t) \odot A_t(x_{t-1}), \quad t = 1, 2, \ldots, T
\end{align*}
\]

where \( u \) is the input, \( A_t, B_t, C \) are convolutional operators with nonlinear activation functions
(ReLU) and batch normalization, \( \sigma \) is the logistic sigmoid function and \( \odot \) is the Hadamard
product (elementwise multiplication operator). We emphasize the variables \( x_t \) and \( g_t \) which represent
respectively the recurrent state variables and gate activations. The third equation clearly shows how
the absence of gates would lead GRCL to be a standard ResNet\( T \).

The complete model architecture is obtained by stacking two initial convolutional blocks, four GRCLs
with \( T = 3 \), and a final readout unit (see Figure 1). The architecture’s number of parameters is
comparable with ResNet50. For further technical details, we refer the reader to the original GRCNN
paper [15].

3 Augmentation & Regularization

One of the main limitations in devising a deep model that predicts the neural activity of the ventral
visual stream is the training procedure. Usually, vision models are trained on the ImageNet dataset
[17], which in its original formulation is related to a multiclass classification problem. On the other
side, Brain-Score’s benchmarks include several tasks [18, 19] which are not necessarily related to
classification. In our work, we try to alleviate this problem by employing and designing augmentation
and regularization strategies, inspired by some of the evaluation benchmarks. In particular, given
their importance in the overall scoring, we decided to focus on behavior - more details can be found
in [20], V1 and V2 tasks based on [21].

Given the spirit of the behavioral benchmark, we took advantage of CutMix [22], an augmentation
strategy that facilitates the recognition of different objects from partial views in a single image. More
specifically, in CutMix, patches are cut and pasted among training images while, at the same time,
the corresponding labels are linearly combined, resulting in: \( \tilde{x} = M \odot x_A + (1 - M) \odot x_B \) and
\( \tilde{y} = \lambda y_A + (1 - \lambda) y_B \), where \( M \in \{0, 1\}^{W \times H} \) denotes a binary mask indicating where to drop out
and fill in from two images, \( 1 \) is a binary mask filled with ones and \( \lambda \) is sampled from the uniform
distribution \( U(0, 1) \).
In parallel, to enhance the robustness of learned representations, we employed AugMix [23], which allows our model to explore the semantically equivalent input space around an image. Briefly, AugMix consists in combining simple augmentation operations - e.g. translation, rotation, shear - together with a consistency loss. Augmentations are sampled stochastically and concatenated while a consistent embedding around the input image is enforced by using the Jensen-Shannon divergence as a loss.

Lastly, we introduced a regularization term in a similar vein to [24], with the idea that behavioral traits cannot be fully described in terms of a scalar metric - e.g. accuracy - but need to be conceived in terms of higher-order descriptors such as reconstructing a confusion matrix, in order to force a network to fail in the same way as a primate would. We took as a reference human performances on 11 superclasses (i.e. groups of classes) of ImageNet [25], projected down the predictions of our model - 1000 dimensional vector - onto the 11 superclasses and imposed an additional cross entropy loss on this term, to quantify the distance between our model and human performances.

To improve scores on V1 and V2 benchmarks, we opted for an augmentation technique largely inspired by [21]. Given that V2 neurons are particularly sensible to textures while V1 neurons respond in a similar way to a texture and its phase randomized counterpart, we considered a texture dataset composed of images coming from the following [26, 27, 28], and we extended it by generating each texture’s phase randomized counterpart, named noise. After that, we fine-tuned our model’s first GRCL block - roughly corresponding to V1 - by freezing all the other layers and randomly blending input images separately with textures and noise. Later on, we fine-tuned the second GRCL block in the same way, but by only blending input images with textures. In this way we tried to emulate how specific layers in the visual cortex respond to different stimuli, and we got improvements in both V1 and V2 with respect to the baseline.

4 Results and conclusions

Introducing receptive fields modulation through a gated recurrent mechanism is beneficial in terms of neural predictivity in the visual cortex. The baseline GRCNN model shows promising results in several benchmarks. As shown in table 1 by introducing the previously mentioned regularization and augmentation techniques, scores improve on specific benchmarks, as well as on average.

In the current state, further work is needed to get a better sense of such preliminary results. In this regard, we have implemented several variants of the baseline model, including an integration of the VOOneBlock [29] as a substitute of the first convolutional block in the GRCNN. With more computational power we are planning to train this combined architecture from scratch. Ultimately, another interesting perspective for further exploration is related to how CNNs learn textures and not shapes [30], which is ultimately analogous to our augmentation strategy for V1 and V2.

5 Acknowledgment

All the authors wish to show their appreciation to MLJC for providing them with the essential computational power (Nvidia RTX 2080 Ti) for finetuning their models. S.A. would like to thank Arianna Di Bernardo and Sophie Bagur for useful and constructive discussions as well as COSYNE 2022 organizers for providing him with a Travel Grant to take part in the conference. S.P. would like to thank ISI Foundation for the hospitality while performing this work and for the computational resources, Giovanni Petri and Karolina Finc for the kind supervision, and Alan Perotti, Maxime Lucas, Antonio Leitao for the helpful discussions. S.P. would also like to thank the Polish National Science Foundation grant UMO-2016/20/W/NZ4/00354 for partially funding this work.

---

1 We open-sourced generated Textures and Noise dataset at https://github.com/sazio/TexturesNoiseDataset
2 We open-sourced the code for our experiments at https://github.com/sazio/brainscore2022
3 https://www.mljc.it/
4 https://www.isi.it
References


Zhe Li et al. “Learning from brains how to regularize machines”. In: Advances in neural information processing systems 32 (2019).


Gustaf Kylberg. The Kylberg Texture Dataset v. 1.0. External report (Blue series) 35. Centre for Image Analysis, Swedish University of Agricultural Sciences and Uppsala University, Uppsala, Sweden, Sept. 2011. URL: http://www.cb.uu.se/~gustaf/texture/.


Figure 1: **Top** GRCNN architecture trained on the ImageNet dataset. It is composed by 2 Convolutional layers, 4 GRCL blocks and a Fully Connected layer which is employed for classification. **Bottom** Schematic representation of a GRCL block. Note that if we unfold the computational graph through time, we end up with something that is very similar to a ResNet.

<table>
<thead>
<tr>
<th>Name</th>
<th>Average</th>
<th>V1</th>
<th>V2</th>
<th>V4</th>
<th>IT</th>
<th>Behavior</th>
</tr>
</thead>
<tbody>
<tr>
<td>GRCNN55 Behavior</td>
<td>.463</td>
<td>.509</td>
<td>.303</td>
<td>.482</td>
<td>.467</td>
<td>.554</td>
</tr>
<tr>
<td>GRCNN55</td>
<td>.462</td>
<td>.525</td>
<td>.306</td>
<td>.481</td>
<td>.479</td>
<td>.520</td>
</tr>
<tr>
<td>GRCNN109</td>
<td>.461</td>
<td>.520</td>
<td>.328</td>
<td>.475</td>
<td>.464</td>
<td>.521</td>
</tr>
<tr>
<td>GRCNN55 V1-V2</td>
<td>.458</td>
<td>.535</td>
<td>.314</td>
<td>.486</td>
<td>.481</td>
<td>.473</td>
</tr>
</tbody>
</table>

Table 1: Brainscore benchmarks results