The Science of AI: Probing Activations in Deep Networks

Vivek Mohan [1] **Gloria Sun** [2 3] **Viktor Schlegel** [1 4] **Tianhong Dai** [5] **Samyakh Tukra** [5] **Anil A Bharath** [1 3]

1. Introduction

The internal workings of trained deep neural networks (DNNs) are considered opaque. But these networks are only black-boxes if we do not try to comprehend them. We describe techniques, borrowed from neuroscience, that can be applied to probe the behaviours of deep neural architectures. Unlike traditional approaches to interpretability, we focus on probing the process of *encoding* states of the input observations, and empirical estimates or proxies of $p(\mathbf{x}, \mathbf{a}|\phi_w; \mathcal{H})$, where $\mathbf{a}$ represents a collection of joint levels of output from a collection of neurons within the network, ϕ_w represents the state of the world that gives rise to network observations, $\mathbf{x}$ and $\mathcal{H}$ denotes a top-level hypothesis concerning the environment in which a deep neural network operates.

The joint probability, $p(\mathbf{x}, \mathbf{a}|\phi_w; \mathcal{H})$ – together with the priors on the world state, should provide a reasonably complete description of the way in which the network represents the state of the environment in which it is placed. This joint estimation is admittedly hard to perform across all neurons and all possible observations states, and so we focus on relatively small numbers of neurons and find ways to summarise the distributions of their activations. Of key interest from the perspective of representation and control is how continuous variables are encoded by the activity of one or more neurons.

2. Interpreting Simple Networks

The work of this presentation falls into the category of *interpretability*; in particular, we move away from probing networks for categorisation, or class separability and focus instead on encoding for inference about continuous states of the real world.

We illustrate the core of this idea with an example taken from computer vision, in which a pair of early-layer convolutional units responds to an input visual field consisting of a circular shape (see Figure 1). Given outputs from the two units, inference of the tangent line to the boundary, given some location on the circle rim, can be obtained by simple nonlinearities applied across the channels corresponding to `conv` Layer 1 outputs. Here, the state of the world corresponds to a local orientation within a spatial field.

There are several tools that aim to support the interpretability of trained deep neural networks underlying modern AI systems; see [1] for a recent survey, focusing on explainability, and with emphasis on elucidating rules that may be implicitly learned and applied. Instead, we focus on *encoding* of world states as our key objective, particularly with regard to properties that can be expressed through real numbers, such as position and velocity.

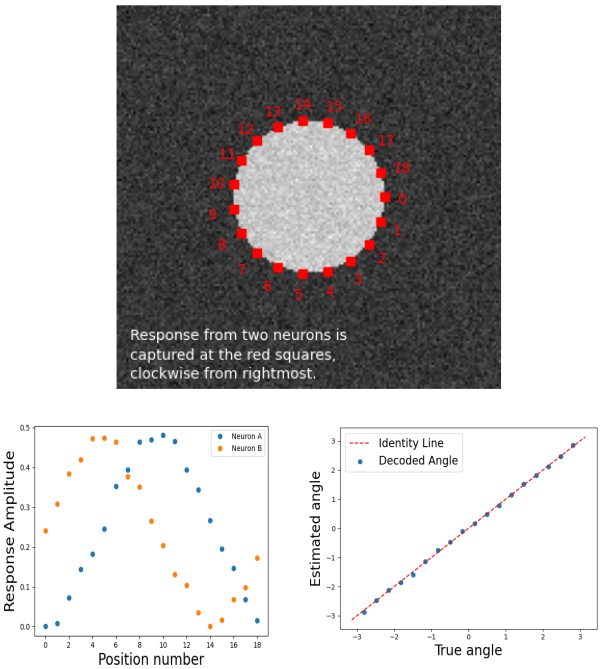

Fig. 1: A simple explanation of encoding a property of the "world" from the outputs of artificial neurons. (Top) An input visual field with specific locations indicated; (Bottom, Left) the responses from a pair of convolutional units at indicated locations from the top figure; (Bottom, Right) Decoded world state (ϕ_w). See text for details.

3. Methods I

We train networks to perform distinct tasks in which aspects of the world state have a physical meaning (orientation, position, velocity, see Fig. 2); we then use combinations of Response Weighted Noise Averaging (RWNA), correlation and covariance based estimates, and phase-weighted analysis to probe how neurons individually and jointly encode continuous information. In all cases, the input field consists of random noise fields, and we collect the ensemble of responses *and* noise patterns to effect our analysis.

The environments and tasks in which the networks were trained included classification, regression, and control tasks. Regression and control tasks were created within custom environments constructed within OpenAI Gym [2].

[1]*Imperial College London, Imperial Global Singapore* [2]*University of Cambridge, Cambridge, UK* [3]*Imperial College London, Department of Bioengineering, United Kingdom* [4]*University of Manchester, Department of Computer Science, United Kingdom* [5]*Kashmir Intelligence, United Kingdom*. Correspondence to: Anil A Bharath a.bharath@imperial.ac.uk.

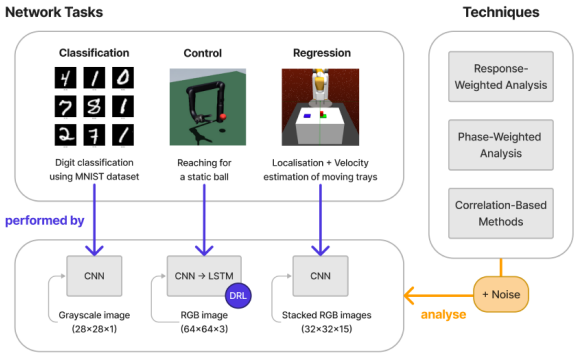

Fig. 2: Neurons in layers associated with visual encoding were probed by driving the network with white Gaussian noise. A series of techniques were applied to uncover associations between input stimuli and the responses of individual neurons, both before and after non-linearities.

4. Methods II

Response Weighted Noise Averaging is defined as follows: Let a noise field, generated on the input to a deep neural network be denoted as $\boldsymbol{\nu}_k(\cdot)$, and assume there are finite K noise realisations, $k = 0, 1, 2, \ldots K-1$. Then we can define the response-weighted noise average of neuron j in layer ℓ of a deep neural network by:

$$\mathcal{R}_{\ell,j}(\cdot) = \frac{1}{K} \sum_{k=0}^{K-1} R_{\ell,j;k} \boldsymbol{\nu}_k(\cdot) \tag{1}$$

This calculation is analogous to Spike-Triggered Averaging, a powerful idea derived from neuroscience [3], [4]. The equivalence is based on establishing a rough correspondence between the rate of firing of a biological (or spiking) neuron, and the strength of activation of a non-spiking artificial neuron.

In addition to this idea, we also considered a number of variants – Response Weighted Correlation, Pearson Correlation, Phase-Weighted Response Averaging and Mutual Information – to capture aspects of the relationship between pixels of the input noise field and the response of individual neurons.

The latter two of these techniques allow or make use of the joint distribution of pixel intensities and value a, output by one neuron. We take the following approach: any generalisation, or evolution of known techniques for characterising the behaviour of neurons should be consistent with existing understanding of how neurons' responses relate to inputs. Thus, we validate and develop our techniques by considering the behaviour of early-stage visual neurons. For a given neural layer, activities can be calculated directly or inferred from the layers and weights preceding and including the current layer.

5. Results

5.1 Receptive Field Mapping

A The quality of receptive field patterns (spatial fields that induce strong responses in artificial neurons) appeared to be better elucidated by using the Pearson Correlation between pixels of the input visual field and the activity of the neuron than by using response-weighted noise averaging (RWNA). However, the patterns uncovered were qualitatively very similar between RWNA and Pearson Correlation.

B Evidence was uncovered of inhibitory counterparts of excitatory neurons. These were found by eigenanalysis of the correlation matrices computed between pixels, and activations of neurons.

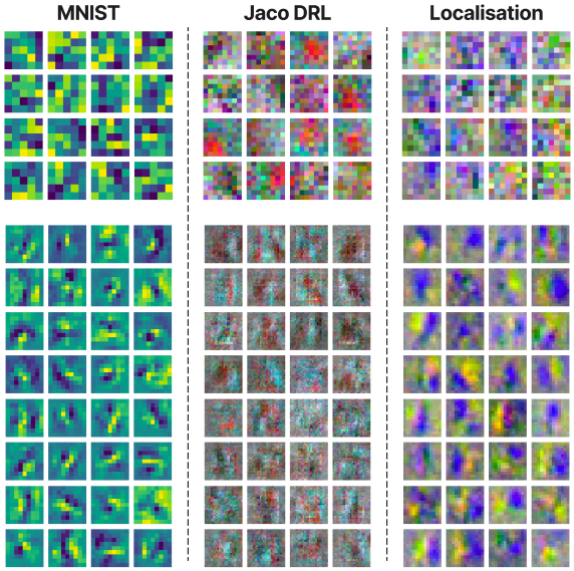

Fig. 3: Receptive fields derived from DNNs trained on different tasks. L1 and L2 conv layer responses for classification, control and positional estimation are all shown from left to right; from RWC (see Appendix). Best viewed in colour.

5.2 Findings: Joint Encoding

We found rich information in the eigenvalues of the covariance structures, both between the input spatial field values and the responses of artificial neurons. We will also present recent findings on joint encoding by pairs of neurons.

6. Conclusions

Interpretability tools for DNNs can be improved, and our current and future work explores approaches that move away from simply determining the input stimuli that induce response maxima and instead focuses on how artificial neural networks encode both states of their environment and actions.

Noise stimuli can be applied to all forms of neural input to discover encoding properties. We found that the sample efficiency of applying noise stimuli decreased with further penetration into the network, away from the output. This is likely because of the equivalent of filtering performed by the earlier layers. It suggests that some form of adaptivity is needed in noise generation to preserve samples efficiently as one penetrates further into deeper layers of the network.

Acknowledgments

AAB, VS and VM wish to acknowledge the support of the NRF CREATE Programme. Tianhong Dai was supported by Samsung Global Research during parts of this work. Gloria Sun is now with Cambridge University.

References

[1] Yu Zhang, Peter Tiňo, Aleš Leonardis, and Ke Tang. A survey on neural network interpretability. *IEEE Transactions on Emerging Topics in Computational Intelligence*, 5(5):726–742, 2021.

[2] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba. Openai gym. *arXiv preprint arXiv:1606.01540*, 2016.

[3] Dario Ringach and Robert Shapley. Reverse correlation in neurophysiology. *Cognitive Science*, 28(2):147–166, 2004.

[4] Peter Dayan and Laurence F Abbott. *Theoretical neuroscience: computational and mathematical modeling of neural systems*. MIT press, 2005.

[5] Tianhong Dai, Kai Arulkumaran, Tamara Gerbert, Samyakh Tukra, Feryal Behbahani, and Anil Anthony Bharath. Analysing deep reinforcement learning agents trained with domain randomisation. *Neurocomputing*, 493:143–165, 2022.

[6] Jonathan W Pillow and Eero P Simoncelli. Dimensionality reduction in neural models: an information-theoretic generalization of spike-triggered average and covariance analysis. *Journal of vision*, 6(4):9–9, 2006.

Appendix A. Training environments

In an earlier work, we trained networks to explore implicit encoding for control problems [5]. In that work, multiple agents were trained to learn to complete a reaching task by controlling a 7DoF arm (Jaco robot) described within the MuJoCo framework. Although the model for the robot included proprioceptive inputs, we trained some versions of the agent *without* the aid of proprioception, finding significant differences in the encoding of the early layers of the agentic network.

This has motivated research into finding out how *position of target*, a state of the world, might be encoded in the network.

To gain further insight, we subsequently created a new agent to directly encode position of targets, doing so i) in the Fetch environment, so that the environment is more widely reproducible by other researchers, and ii) doing so with multiple targets, and iii) moving on to moving targets.

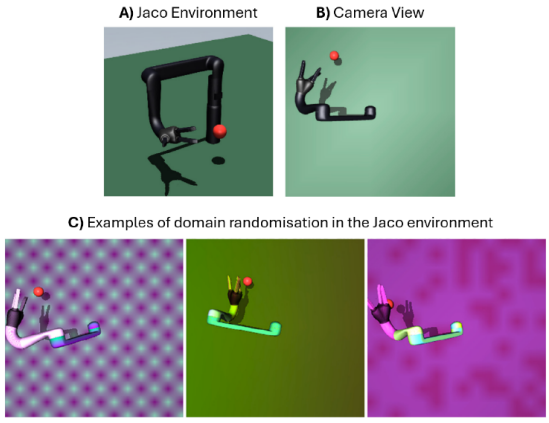

Fig. A1: The rendered MuJoCo-based model of the Jaco robot. The visual domain randomisation (changing visual appearance of the same physical scene) was found in earlier work to improve performance through modifying the early visual stages of the AI agent [5]; we develop techniques to further understand the way in which the environmental state is encoded by the outputs of neurons.

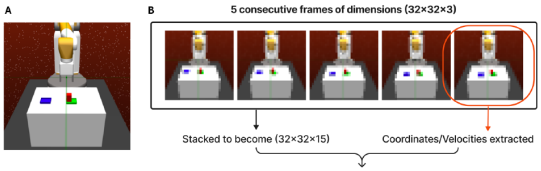

Fig. A2: The Fetch environment in OpenAI is modified so that a more complex visual task is presented to an agent seeking to fulfil a pick and place task. Two trays respectively hold and receive the object to be picked and placed, and both trays move in space. From the trajectories traced out by the moving trays, we train two networks to respectively estimate position and velocity of the objects.

1.1 Fetch Environment

We modified the Fetch environment, based on the standard moving target environment of a "pick and place" task, so that the tasks to be (respectively) picked and placed were moving. We illustrate the environment, with examples of frames as fed into the velocity and position estimating networks as shown in Figure A2.

Appendix B. New findings

2.1 Phase-Weighted Response Averaging

Using Phase-Weighted Response Averaging, a technique that groups responses into different amplitude windows, we are able to uncover more information about a neuron's behaviour, pre-threshold. In this case, response weighted averaging (RWA) collapses to an unstructured pattern unless we consider

the activity of the neuron in different ranges, and capture the pattern that induces values within those ranges. The usefulness of this in interpreting neural coding of stimuli depends on the *specific* value of the neuron's bias term, and the type of non-linearity used, but it enhances our understanding of the behaviour of artificial neurons in encoding continuous states.

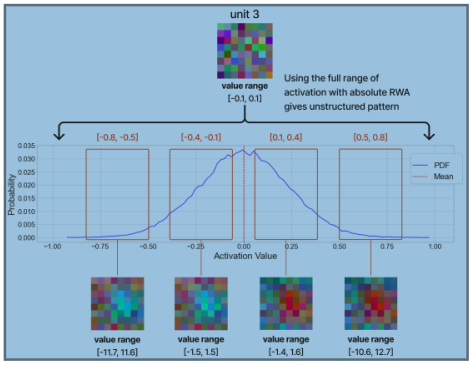

Fig. A3: Response Weighted Averaging, like Spike Triggered Averaging [4], can miss non-linear behaviours in neurons, because positive and negative responses (with respect to a mean response level) can cancel; this depends on the non-linearity applied to the neuron. By probing joint encoding instead (through PWNA), we can uncover these responses.

2.2 Response Weighted Covariance

We found that eigenanalysis of response weighted covariance also yielded insight into neural coding. Response Weighted Covariance (RWC) is to spike-triggered covariance [6] as response-weighted averaging is to spike-triggered averaging [4]. By sorting the eigenvalues, different spatial sub-fields emerged from some of the analyses.

RWC is obtained by replacing the terms of RWNA with products of deviations (from respective means) between pairs of locations in the input noise field, weighted by response amplitude, a, from a neuron. Eigenanalysis then follows, to determine the main components of response variation.