Université de Montréal

Modeling functional brain activity of human working memory using deep recurrent neural networks

par

Pravish Sainath

Département d'informatique et de recherche opérationnelle
Faculté des arts et des sciences

Mémoire présenté en vue de l'obtention du grade de
Maître ès sciences (M.Sc.)
en Informatique

Décembre 2020

Université de Montréal

Faculté des arts et des sciences

Ce mémoire intitulé

Modeling functional brain activity of human working memory using deep recurrent neural networks

présenté par

Pravish Sainath

a été évalué par un jury composé des personnes suivantes :

Liam Paull
(président-rapporteur)

Guillaume Lajoie
(directeur de recherche)

Pierre-Louis Bellec
(codirecteur)

Sarath Chandar Anbil Parthipan
(membre du jury)

Résumé

Dans les systèmes cognitifs, le rôle de la mémoire de travail est crucial pour le raisonnement visuel et la prise de décision. D'énormes progrès ont été réalisés dans la compréhension des mécanismes de la mémoire de travail humain/animal, ainsi que dans la formulation de différents cadres de réseaux de neurones artificiels à mémoire augmentée.

L'objectif global de notre projet est de former des modèles de réseaux de neurones artificiels capables de consolider la mémoire sur une courte période de temps pour résoudre une tâche de mémoire et les relier à l'activité cérébrale des humains qui ont résolu la même tâche. Le projet est de nature interdisciplinaire en essayant de relier les aspects de l'intelligence artificielle (apprentissage profond) et des neurosciences. La tâche cognitive utilisée est la tâche N-back, très populaire en neurosciences cognitives dans laquelle les sujets sont présentés avec une séquence d'images, dont chacune doit être identifiée pour savoir si elle a déjà été vue ou non. L'ensemble de données d'imagerie fonctionnelle (IRMf) utilisé a été collecté dans le cadre du projet Courtois Neurmod.

Nous étudions plusieurs variantes de modèles de réseaux neuronaux récurrents qui apprennent à résoudre la tâche de mémoire de travail N-back en les entraînant avec des séquences d'images. Ces réseaux de neurones entraînés optimisés pour la tâche de mémoire sont finalement utilisés pour générer des représentations de caractéristiques pour les images de stimuli vues par les sujets humains pendant leurs enregistrements tout en résolvant la tâche. Les représentations dérivées de ces réseaux de neurones servent ensuite à créer un modèle de codage pour prédire l'activité IRMf BOLD des sujets. On comprend alors la relation entre le modèle de réseau neuronal et l'activité cérébrale en analysant cette capacité prédictive du modèle dans différentes zones du cerveau impliquées dans la mémoire de travail.

Ce travail présente une manière d'utiliser des réseaux de neurones artificiels pour modéliser le comportement et le traitement de l'information de la mémoire de travail du cerveau et d'utiliser les données d'imagerie cérébrale capturées sur des sujets humains lors de la tâche N-back pour potentiellement comprendre certains mécanismes de mémoire du cerveau en relation avec ces modèles de réseaux de neurones artificiels.

Mots clés: réseau de neurones récurrents à mémoire court et long terme (LSTM), mémoire de travail, IRMf, modèle de codage, similarité de représentation

Abstract

In cognitive systems, the role of working memory is crucial for visual reasoning and decision making. Tremendous progress has been made in understanding the mechanisms of the human/animal working memory, as well as in formulating different frameworks of memory-augmented artificial neural networks.

The overall objective of our project is to train artificial neural network models that are capable of consolidating memory over a short period of time to solve a memory task and relate them to the brain activity of humans who solved the same task. The project is of interdisciplinary nature in trying to bridge aspects of Artificial Intelligence (deep learning) and Neuroscience. The cognitive task used is the N-back task, a very popular one in Cognitive Neuroscience in which the subjects are presented with a sequence of images, each of which needs to be identified as to whether it was already seen or not. The functional imaging (fMRI) dataset used has been collected as a part of the Courtois Neurmod Project.

We study multiple variants of recurrent neural network models that learn to remember input images across timesteps. These trained neural networks optimized for the memory task are ultimately used to generate feature representations for the stimuli images seen by the human subjects during their recordings while solving the task. The representations derived from these neural networks are then to create an encoding model to predict the fMRI BOLD activity of the subjects. We then understand the relationship between the neural network model and the brain activity by analyzing this predictive ability of the model in different areas of the brain that are involved in working memory.

This work presents a way of using artificial neural networks to model the behavior and information processing of the working memory of the brain and to use brain imaging data captured from human subjects during the N-back task to potentially understand some memory mechanisms of the brain in relation to these artificial neural network models.

Keywords: long-short term memory (LSTM) networks, working memory, fMRI, encoding model, representational similarity

Contents

List of tables

List of figures

List of acronyms and abbreviations

RNN	Recurrent Neural Network
LSTM	Long-short term Memory Network
SAB	Sparse Attentive Backtracking
CNN	Convolutional Neural Network
HRF	Hemodynamic Response Function
BCE	Binary Cross-entropy Error
fMRI	Functional Magnetic Resonance Imaging
BOLD	Blood Oxygen Level Dependent
TR	Repetition Time
WM	Working Memory
PFC	Prefrontal Cortex

Gratitude

I would like to thank my supervisors Dr. Pierre Bellec and Dr. Guillaume Lajoie for initiating me into this exploration of the field of Neuro-AI and facilitating my journey of learning through their continuous support and sharing of ideas. I appreciate all their help and patience with me.

I would also like to thank my parents for imbibing a scientific spirit in me and seeding and encouraging my interest in Science from a very young age. I express my deep gratitude to all the teachers who have taught me so far for helping me benefit from the light of knowledge.

Chapter 1

Introduction

The brain is a complex and uniquely intriguing system that is composed of billions of neurons that are interconnected. The computations of these neurons manifest themselves over several organized components that are responsible for specific processes and lead to all our thoughts and actions. Our notion of intelligence as humans has been primarily been through this entity. Yet, the question that has amazed us over centuries is how the brain makes this intelligent behavior possible. The 17th-century philosopher and scientist René Descartes famously wrote "I think, therefore I am" [1] [**26**]. From an era when these questions were merely philosophical pondering, we find ourselves in a time with sophisticated technologies to measure and record activity from the brain to help us better answer this question.

Scientific advancement over the years has empowered humans with the ability to take much closer and detailed look at the brain, although with several challenges. With the introduction of functional imaging of the brain [**88, 87**], it has been possible to *non-invasively* record data of human brain activity under various conditions. Using these data, there have been significant advances in our understanding of the brain as well as in methods of diagnosis of certain medical conditions. Over the years, this has been possible because of a systematic development of a whole suite of statistical methods to analyze brain functional imaging data and better understand the functional architecture and working of the brain.

On the other hand, Deep learning has dominated Artificial Intelligence (AI) in the past decade [**3**]. There was a transition from symbolic to more connectionist systems of artificial neural networks [**98, 42, 105**]. These networks work because of data-driven training methods that help them construct good representations of the input data through different computational hierarchies. These networks learn meaningful transformations of the inputs that are associated with the tasks that they are trained on. Thus, the applications of these networks can go beyond just making predictions for the trained task.

[1] Originally in French: *je pense, donc je suis* and later in Latin : *Cogito, ergo sum*

These developments provide a good opportunity and timing for synergy to explore possibilities of similarities and differences at various levels between these two systems : brains and artificial neural networks. Especially, the relationship of these systems at the computational level is a great research avenue for computer scientists. In both cases of brains and artificial networks, we know that the responses observed in these systems are a result of complex computations dependent on the input data and other conditions. A recent study [**17**] takes this view and argues for the use of deep neural networks as scientific models in the study of cognitive phenomena. The study emphasizes the different advantages of deep neural networks in this pursuit - predictive (practical applications, experimental substitute for the brain), explanatory (useful for teleological, mathematical and post-hoc explanations) and exploratory (generating new hypotheses, proof-of-principle demonstrations and determining the suitability of phenomena).

Deep learning methods have already started being employed in analysis of brain imaging data [**108, 110**]. There have been several studies such as [**120**] that have explored the detection of brain disorders. Research from recent times such as [**126, 52, 60**] have attempted to model the functioning of visual and auditory systems in the brain by successfully exploiting deep neural networks. Some of these studies have also led to a better understanding of the these perceptual systems [**117, 52**]. However, there still remains a huge potential to leverage deep learning to study the brain in terms of representations and further expand the toolbox of statistical methods for brain imaging data analysis. These models when used in conjunction with brain recordings can potentially help build hypotheses about the nature of the environment and the information processing in the brain.

Memory is a very important aspect of intelligence. Many neuroimaging studies have been carried out to understand the representation of memory in specific regions of the brain [**19, 14, 35**]. Deep learning has significantly advanced with respect to the deep neural networks that have memory capabilities [**51, 10, 101**]. But, there has not been much progress in utilising these modern deep neural networks to understand memory processing in the human brain. Our work is a step in this direction. Most current understanding has been achieved by analyzing fMRI data from the mirror perspectives of encoding and decoding. When analyzing data from the encoding perspective, one attempts to understand how activity varies when there is concurrent variation in the world.

In the study that we present in the thesis article, we model the functional Magnetic Resonance Imaging (fMRI) activity recorded from human solving a memory task. We achieve this by training architectural variants of deep neural networks, specifically deep recurrent neural networks such as Long-short Term Memory (LSTM) to solve a memory task. We use the network representations obtained for the stimuli to create predictive models to encode the fMRI data recorded from humans performing the memory task. We further establish the

relationship between memory representations in trained deep recurrent neural networks and the brain regions involved in memory processing.

We hope that our contribution provides value by reinforcing the idea of studying memory in the brain using artificial neural networks and serves as a prototype for further extending to other cognitive memory tasks. From an Artificial Intelligence perspective, it expands the use of learned representations of recurrent networks by exploring the relevance of them to memory in brain. From a Cognitive Neuroscience standpoint, it has a potential to help understand visual cognition and memory in the brain in addition to providing a new way of modeling brain activity during memory processing.

In the remainder of this document, we first present the initial chapters (Chapters 2, 3, 4 and 5) to familiarize the reader with the necessary background to understand our work. We then discuss our experimental and modeling methods and present the analysis done in our study in the form of a scientific article.

Chapter 2

Functional Magnetic Resonance Imaging

In this chapter, we introduce some preliminary ideas and terminology in functional magnetic resonance imaging (fMRI) to help understand the neuroimaging data that we use to model in our work.

2.1. Functional Neuroimaging

Functional neuroimaging is a class of methods that use neuroimaging to measure activity in some brain areas, in order to understand how different aspects of cognitive function arise from them. These technologies that record data from the neural activity in the brain form the core of methods used to further our knowledge about the brain.

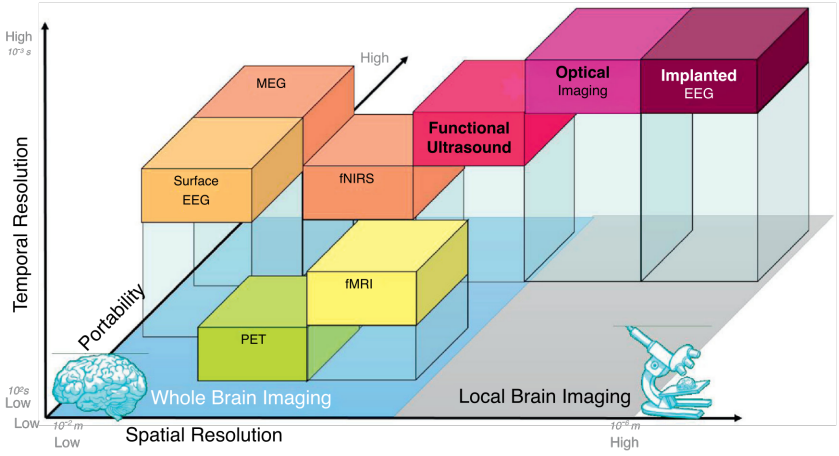


Fig. 2.1. An overview of various functional neuroimaging techiques depicted to indicate their spatial and temporal resolutions along with portability of the recording technique (from *Deffieux et al.* [**24**] - CC BY 4.0)

Some commonly used means of functional neuroimaging are functional magnetic resonance imaging (fMRI), electroencephalography (EEG) and magnetoencephalography (MEG). The dependence on quantitative methods for analysis of the recorded neuroimaging data using statistics, data science and other computational methods opens up ways for contributions by the community of computer scientists to makes advances.

Figure 2.1 shows placement of many functional neuroimaging methods with respect to their portability and resolutions in time and space. For example, MEG signals have a higher temporal resolution compared to fMRI and are portable, while fMRI has a higher spatial resolution and less portable. It is worth noting that many methods are useful in different contexts, despite their limitations.

fMRI has been widely used and it has dominated the study of the brain the past years, mainly because it is non-invasive in the sense that it does not harm or cause side effects to the people being and its ability to model the whole brain at a fair temporal resolution.

2.2. Magnetic Resonance Imaging (MRI)

Some nuclei that contain an odd number of protons (such as water) align along an external constant magnetic field. Under this condition, they posses a spin property that causes them to emit a characteristic electromagnetic signal when they are perturbed a weak magnetic field oscillating at a certain specific frequency characteristic of the nuclei. The protons in these kinds of nuclei are always spinning about their axis producing a net magnetic moment along the direction of the axis of the spins. This happens when this characteristic frequency matches that of the perturbing magnetic pulse, causing a resonance effect that is referred to as Nuclear Magnetic Resonance (NMR).

MRI uses this principle of NMR to measure the net magnetization of the all such nuclei in a given space. An MRI scanner such as the one shown in Figure. 2.2 consists of a very strong magnet that produces the magnetic field in the *bore* where the participant lies down on the table. Radio-frequency (RF) coils capable of producing the oscillating pulse at different frequencies are used to excite the nuclei at particular locations which results in the emission of a radio frequency signal (referred to as the echo pulse), which is received by another coil. The frequency of the received signal from each location is mapped to a corresponding intensity value to structure the image as arrays of pixels.

Time to Echo (TE) is a setting of the experiment indicating the the time taken between the transmission of the RF pulse and the receipt of the echo signal. Different types of tissues produce different types of contrasts. This is due to the fact that tissues that need to be images have different characteristic frequencies when the NMR can occur.

Fig. 2.2. A view of a MRI scanner showing the table extended out of the bore (attribution: Liz West from Boxborough, MA - CC BY 2.0)

2.3. Blood-oxygen-level-dependent (BOLD) Signal

The neurons in the brain depend on oxygen in the blood for the metabolic needs of their firing activity. On being stimulated with some condition, the neurons that actively respond to it fire at a faster rate increasing their energy demand. Through the blood vessels, the body supplies more oxygen to these active neurons compared to the inactive neurons through the phenomenon called as the *hemodynamic response*. This alters the relative proportions of oxyhemoglobin and deoxyhemoglobin (oxygenated and deoxygenated blood respectively) and thus it can be used a proxy to measure neuronal activity.

Seiji Ogawa and his colleagues identified the the difference in magnetic properties of oxygenated and deoxygenated hemoglobin [**87**] and successfully demonstrated [**88**] the detection of the signal variation in an MRI scanner. The NMR signal originating from this is referred to as the *Blood Oxygen Level Dependent*. The BOLD fMRI signal can be thought of as representing the proportion of the oxygenated and deoxygenated hemoglobin in the blood. The entire pipeline of measurement is illustrated in Figure 2.3. Thus fMRI BOLD signal is normally observed close to areas with active neurons and thus can be considered as stimuli-driven activations.

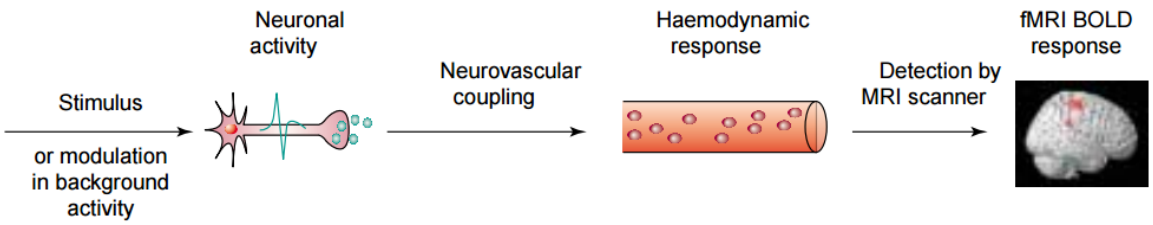


Fig. 2.3. A schematic pipeline summarizing the phenomena involved in task-triggered hemodynamic response and the eventual detection of BOLD signals during a fMRI recording (from *Arthurs et al.* [**4**] © 2002, used with permission from Elsevier)

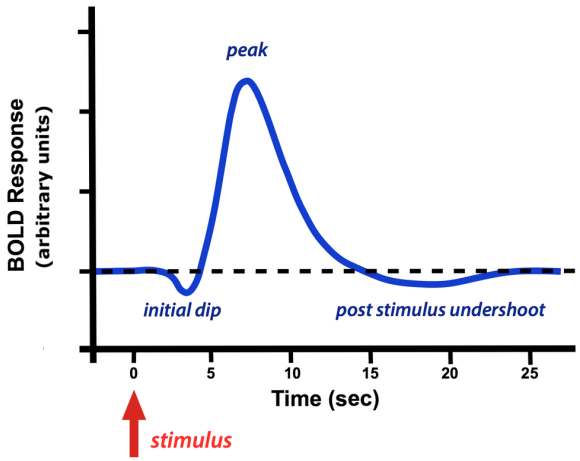


Fig. 2.4. The shape of a typical BOLD Response. After the stimulus onset, there is an initial dip, followed by a peak and a post stimulus undershoot seconds beyond which the signal goes back to its usual level indicated by the dashed line.

It has been empirically established that the BOLD response is indicative of an increase in neuronal activity [**85**]. The hemodynamics phenomena is characterized by a *hemodynamic response function (HRF)* as seen in Figure 2.4 where it has a slow activation peak and eventual fall. It can be seen that a stimulus that lasts for a brief period of few seconds acts as an impulse to produce the response lasting till 25 s. It can be viewed as a general model of the BOLD response to an impulse neural input (triggered by the stimulus).

2.4. fMRI Experiment

During scanning, the participant is in a lying position on a table inside the bore of an MRI scanner with their head placed in a head coil. In a task-based fMRI experiment, the participant takes part in a task paradigm and is well-equipped with the necessary tools (screen, buttons, etc.) required to solve these tasks.

A *session* is a single-stretch of recording on a given day. Each functional session is divided into a fixed number of *runs*. A *run* can be composed of many *blocks* of different task and rest periods. The participant can be presented with some *stimuli* and/or asked to perform some action with certain time intervals within each block.

After each fixed time interval called the repetition time (TR), a three-dimensional image of the BOLD activity across the entire volume of the brain is captured by the scanner. fMRI images are usually acquired as axial slices of certain thickness called the *slice thickness*. The *field of view (FOV)* indicates the vertical extent of the brain that is present inside the image. The matrix size is the number of grids in the axial slice images.

The individual units of recorded data are called the volume elements or *voxels* with their intensities representing the strength of the BOLD signal at that location. The slice thickness,

FOV and matrix size together determine the voxel dimensions (usually in mm). At the end of each fMRI data recording session, the output produced is a time-series of image volumes representing the intensities for each voxel that is part of the three-dimensional brain volume. This is illustrated in Figure 2.5.

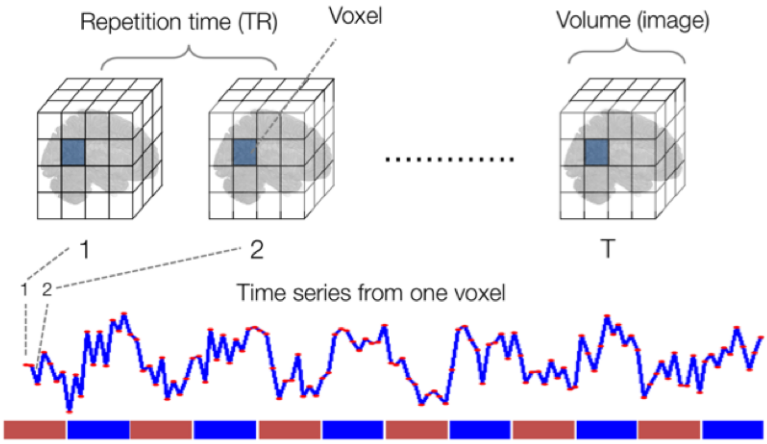


Fig. 2.5. An illustration of the BOLD time series corresponding to a single voxel from the recorded spatial volumes across multiple time steps (TRs). The colorbar indicates different task conditions denoted by red and blue. (*© 2015 Tor D. Wager and Martin A. Lindquist* [**73**])

The main purpose of recording the fMRI data during experiments is to analyze the data and explain brain function and behavior. One key goal of fMRI data analysis is the localization of brain regions that are active during a specific task, to understand their cognitive function.

2.5. Preprocessing

Recorded fMRI data is usually not directly usable for analysis. If the data does not plausibly satisfy some conditions, it can cause many statistical methods to be inapplicable. A sequence of appropriate preprocessing steps are required to be carried out to remove spurious artifacts, enforce certain statistical assumptions and standardize the spatial locations of brain regions. This is an essential step in assuring the quality of recorded data. A number of these steps are usually implemented using some software packages like fMRIPRep [**32, 31**] while few others need to be done programatically.

However, these steps need to be carefully chosen and used in the right way to make the use of the data. Not all steps might be necessary in all cases and some might need to be modified. While there is an exhaustive list of these steps, we briefly summarize some important ones below.

2.5.1. Slice-time Correction

It needs to be ensured that all the voxels contained in a particular brain volume were acquired at the exactly the same time. However, this is not the case in many cases where the activity is recorded as 2D axial slices of the brain volume with some delay between them. This is fixed by interpolating the values for a common time point based on the values from the entire time course [**31**]. However, this step is not needed in cases where we are able to simultaneously acquire multiple slices fast.

2.5.2. Motion Correction

It is assumed that time course points from a particular voxel contains signals from only its actual location. This is violated when there is head motion during and in between the scan session and thus it needs to be corrected to ensure the consistency of the values represented by the voxels. This is handled by applying a rigid body transformation (rotation and translation parameters) to each volume to align them to remove the effects of any motion. This is carried out based on a reference volume in the time course (usually the first) [**55**].

2.5.3. Susceptibility Distortion Correction

The magnetic field inside the scanner could not be very homogeneous in all cases. This causes a spatial distortion in the imaged data due to the errors in the conversion from the frequencies to spatial locations to map the value at each voxel. This method reduces the effect of these distortions by estimating the inhomogeneity map of the field and adjusting the mapping of locations by calculating the displacements of the voxels.

2.5.4. Co-registration

Registration of an image is the process of fitting it into another space. The fMRI BOLD image volumes recorded from the scanner might not be well aligned to the structural features of the brains. To address this, the recorded anatomical images are used as they have a clear distinction of boundaries. Thus, the functional volumes are fitted to the anatomical image and map the fMRI signal onto the surfaces generated in the anatomical images using a regression.

2.5.5. Spatial Normalization

The shapes and sizes of brains of individuals is different and a common space of fixed dimensions is very useful in analyses as they map to the same anatomical or functional structures. This is achieved using spatial normalization, in which the data of each subject is warped into a standard template space such as the ones [**37**] developed by the Montreal

Neurological Institute. This step is essentially a warping algorithm that uses a complicated nonlinear normalization with a large set of parameters. Nonlinear algorithms based on diffeomorphic registration, which is an invertible transform that maps from subject space to template space and back have been very successful. Advanced Normalization Tools (ANTs) [**5**] algorithm is one such example.

2.5.6. Confounds Removal

The BOLD signal typically has a very small amplitude and when it that is measured it can potentially be confounded by a variety of causes - physiological (breathing, cardiac activity, etc.), hardware (coil heating up, frame displacement etc.) or other sources that can contribute to the global noise signal. So, in order to address this, a set of confound time series are estimated as noise components by performing methods such as ICA or CompCor [**12**]. Finally these nuisance parameters are regressed out to remove their effects.

2.6. Spatial Orientations

fMRI data analysis often produces statistical maps of the brain that contain values in each spacial location. As we deal with brains in a three dimensional space and with complex structures present, having a common notion of orientations in space and names for different parts is a useful thing. In this section, we will highlight the basic terminology used for this purpose.

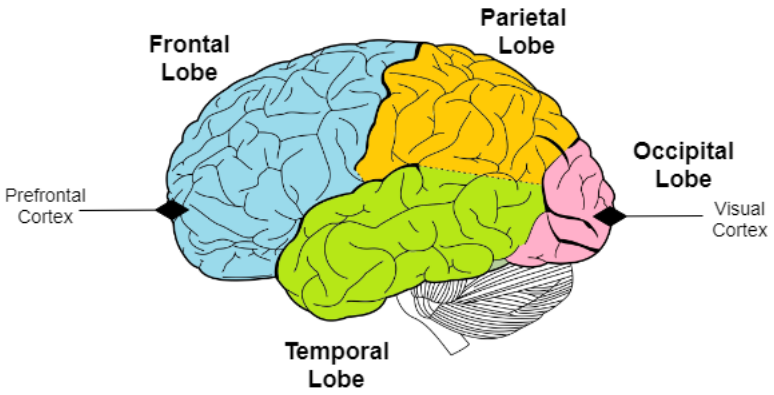


Fig. 2.6. A diagram of the human brain showing the different lobes and some important cortices. The 4 lobes of the brain are occipital (pink), parietal (orange), temporal (green) and frontal (blue). The two important cortices (among many other) that are highlighted are the prefrontal coretex located in the frontal lobe and the visual cortex in the occipital lobe (Colored and labelled on a sketch by Henry Vandyke Carter)

The brain surface is organized into high-level distinct structures called lobes. Figure 2.6 shows the names and locations of these lobes in the human brain. These are occipital, parietal, temporal and frontal lobes that contain specialized areas for different cognitive

functions. Figure 2.6 also shows some curved shape structures that are spread across the surface of the brain and these can be used to locate areas on the surface. Each outward folding is a *gyrus* and each inward fold is a *sulcus*.

The results in three dimensions consisting of values for each spatial location in the volume can only be displayed in two dimensions using cross-sectional planes along the three directions. Figure 7.3 provides the names and basic orientation of those slices and their spatial relation to the overall head and brain surface.

In the three dimensions of the space representing the brain, specific names are given to different directions (axes) and these planes. The X axis is the standard brain coordinate space that indicates the (*lateral* sides) left-to-right dimension (of the participant). The back-to-front dimension is the Y axis which spans from *posterior* at the back of the brain to *anterior* at the front. The Z axis is the bottom-to-top dimension extends from *inferior* (bottom) to *superior* locations (top). These locations are sometimes also referred to as *ventral* (top-to-bottom) and *dorsal* (bottom-to-top).

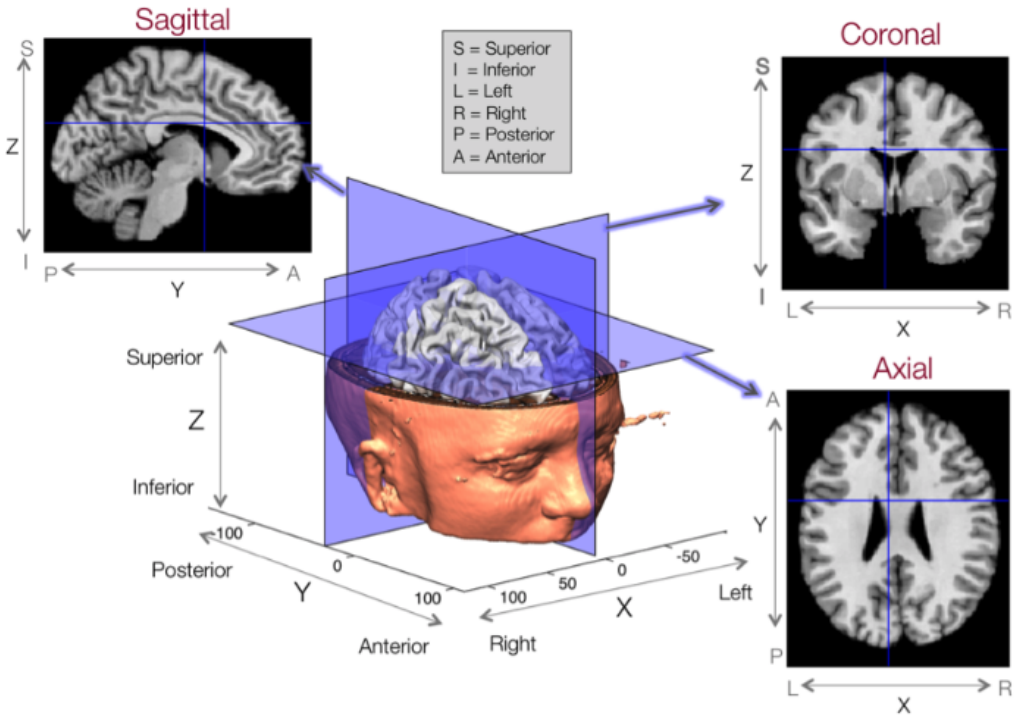


Fig. 2.7. A depiction of the three types of cross-sectional planes in a 3D brain volume, along with the associated axes indicating the various terms associated with the directions in brain anatomy (legend in gray box in the top). The images corresponding to the three cross-sectional planes - sagittal, coronal and axial are also shown. (*© 2015 Tor D. Wager and Martin A. Lindquist* [**73**])

In the subsequent chapters, to associate certain terms used to identify positions or directions in the brain, it might be useful to refer to Figure 2.7 and Figure 2.6.

Chapter 3

The N-back task and working memory

In this chapter, we outline the basic ideas of working memory and introduce the N-back task. We also summarize some existing understandings about the brain areas relevant to working memory, based on knowledge gained from many studies in Neuroscience.

3.1. Working memory (WM)

Working memory (WM) is an important resource for maintaining and manipulating small sets of information online for a brief period of time, a critical ability that supports general learning. The role WM acts as a link between perception, attention, and long-term memory processing [**78, 8**]. Memory researchers have traditionally classified human memory systems into three distinct types: sensory memory, short-term memory and long-term memory [**78**].

Sensory memory refers to the prolonged neural activations in the sensory areas as a result of the stimulus. One such example is the response evoked by a visual stimulus in the primary visual cortex V1. However, this is merely a buffer memory without directly being used in decision making or action. It is meant for using sensory information at higher levels of memory.

In short-term memory, the memory trace for information that is held decays quickly within seconds, but if reinforced by active rehearsal, this information may be transferred into long-term memory where it can be retained for much longer periods. It is not uncommon to use the terms short-term memory and working memory interchangeably. A common notion is to consider working memory as a mechanism for both maintenance and manipulation, whereas short-term memory as only for temporary maintenance.

Previous research has also focused on different sensory modalities in the short-term memory. For visual short-term memory, object representations are created/encoded rapidly, are maintained by means of an active mechanism and are terminated when active maintenance ends. The storage capacity of this ability is limited to just a few simple objects. This function

is achieved by using different components in the visual system and a higher-level executive control.

3.2. Models of working memory

Many theories about working memory have been proposed in several studies over the years. A good number of them models have been extensively tested and studied [**78**].

Although many cognitive models exist for WM, the Baddeley model is a sufficient one to understand the visual working memory. The original model, introduced by Alan Baddeley in 1986 [**9**], WM includes a central executive that monitors two modality-dependent independent subsystems- the visuospatial sketchpad, and the phonological loop. The central executive is an attentional control system that interacts with the remaining components. Later, this model was expanded [**6**] with an additional sub-system was called the "episodic buffer". Figure 3.1 shows the diagram of this model composed of these interacting components.

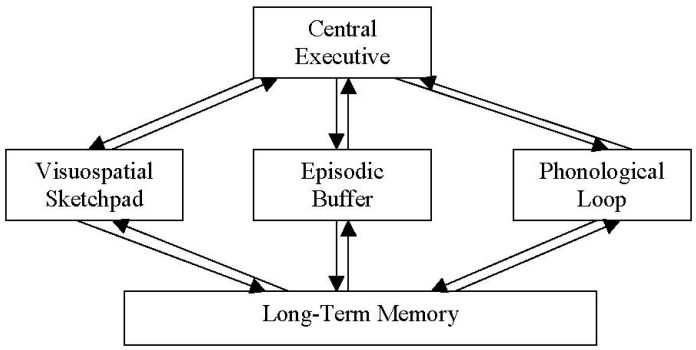


Fig. 3.1. A schematic diagram of Baddeley's multicomponent model of working memory (based on [**6**]). The 3 components in the middle - Visuospatial sketchpad, Phonlogical Loop and Episodic Buffer are controlled by the Central Executive and interact with the Long-Term Memory.

The visuospatial sketchpad processes visual and spatial information while the phonological loop takes care of verbal and auditory information through an articulatory control process. Both the visuospatial sketchpad and phonological loop are comprised of an active rehearsal and a passive storage component for their respective modalities of information. The episodic buffer stores information in a multidimensional code and its helps in integrating the information from the three other components - visuospatial sketchpad, phonological loop and long-term memory.

3.3. The N-back Task

The N-back task, first described by *Kirchner et al.* [**63**] in 1958, involves presenting of fast and continuously changing stimuli for measuring retention of information in this very short duration. It was originally conceived to investigate the effect of age in the performance among adults of different age groups. It is a standard technique used in several experimental studies to quantify the capacity and analyze the properties of the working memory.

In the N-back task, individuals are presented with a continuous sequence of stimuli and are required to recall if a stimulus was presented a specified number of steps back in the sequence (N represents how far back in the past sequence the participant needs to remember). The stimulus that needs to be remembered to solve the current trial is referred to as the *target*. Solving the task essentially involves determining if the current stimulus is a target or a non-target.

As shown in Figure 3.2, at an N of one, the target would be the stimulus that was presented immediately prior to the current stimulus. At an N of two, the correct response is to a repeat of the stimulus that was presented two prior to the current stimulus. The task difficulty or the load increases correspond with the value of N. This task is very pertinent in the study of WM due to the attentional and memory requirement where there is need to maintain the target stimulus and to continuously update the stimuli held in the memory.

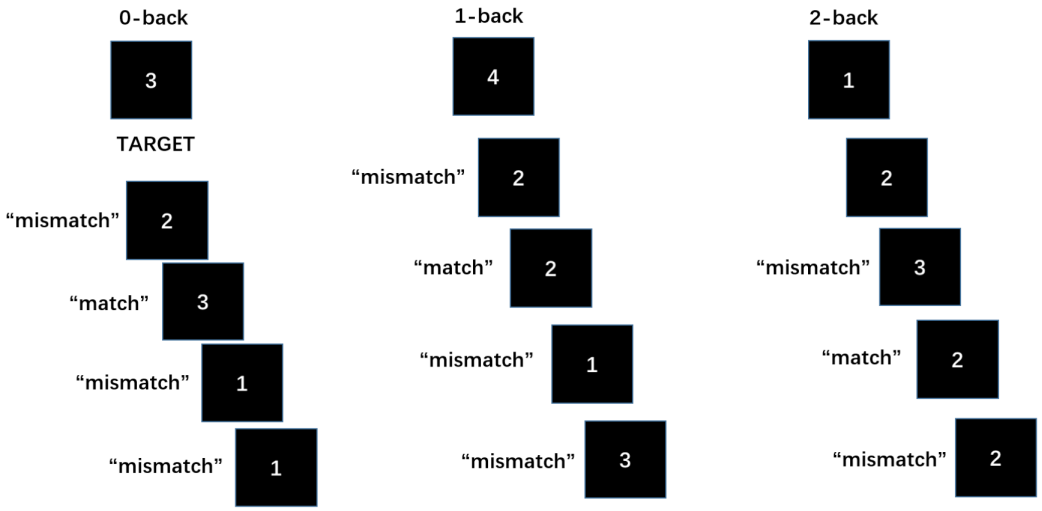


Fig. 3.2. Examples of a letter N-back task for N = 0, 1 and 2.
(Left) 0-back : The first letter in the sequence is the target and the subsequent letters are matched with this. The target is the only letter that needs to be remembered.
(Middle) 1-back : Each letter is matched with the previous letter which is the target.
(Right) 2-back : Each letter is matched with the letter two-steps back and it is the target.

There are many variants of this task based on various settings of the experiments: stimulus type (verbal, visual, spatial, etc.), the target stimulus feature (eg. identity or the location of stimulus), the length of the inter-stimulus delay, the amount of cognitive load (eg. value of N), and interval of retention or distraction. The tasks designed for humans typically use abstract objects, faces, letters, words, digits or common objects.

Although this task is seemingly simple, it involves multiple processes such as: perceiving the stimulus one at a time, encoding of each stimulus in the working memory, maintaining the representation of the target stimulus in memory, updating the representation at the next step, deciding if new stimulus matches the target and pressing a button to indicate the match or non-match.

Adult neuroimaging studies using the N-back task usually vary load between 1-back and 3-back, with 0-back typically serving as a control condition as done in [**89**].

3.4. Neural correlates of Visual Working Memory (VWM)

In this section, we summarize the insights gained from many fMRI studies about the brain areas that are activated during tasks involving working memory.

The most common method of modeling fMRI data is by fitting a General Linear Model (GLM) to the time series of each voxel using some possible explanatory variables. In the case of N-back tasks, these variables could be some settings of the experiment (duration, onset, etc.), characteristic of the input (eg. target or non-target) or any other factor that could explain the BOLD signal (eg. age). However, despite some success with this method of modeling, it has many limitations and it has been widely criticized [**81**].

Most of the brain areas identified to be related to WM are based on meta-analysis that combine the activation maps of many studies to identify the common ground. These studies mostly use a technique called Activation likelihood estimation (ALE) [**111, 69**], which aims at determining areas with significant probabilities of being activated across several experiments.

The broad results from the neuroimaging studies on adults about the neural basis of WM have dominantly been linked to the frontal and parietal cortices [**20**].

Specifically, the prefrontal cortex (PFC) has been found to be one of the key areas as it plays a crucial role in WM, by functioning as the central executive ([**19, 14**]).

Different parts of the frontal cortex have been explored in relation the types of stimuli in the WM task. The dorsolateral PFC (dlPFC) was found to be dominant in manipulation or updating of memory. [**36**] The ventrolateral PFC (vlPFC) is considered to be involved in encoding and maintenance of memory in the WM [**99, 28**].

The ventral and dorsal visual stream notion that is popular in Neuroscience associates the dorsal stream with the processing of object location and motion, and the ventral stream with object recognition. Some studies have confirmed this theory even for visual working memory by highlighting the role of dorsolateral PFC in processing object location memory [**21**] and associating the ventrolateral PFC with object identity memory [**104**]. Thus, the dorsal frontal areas are more sensitive to spatial information and the ventral frontal areas are related to visual non-verbal objects. Other studies also indicate the impact of the memory load on the activations in the PFC [**89**].

The studies specific to visual working memory have also highlighted the involvement of the temporal and occipital lobes. Areas in the occipital region were found to serve as the visuospatial sketchpad according to [**103**]. The study in [**115**] discusses activities in occipito-temporal areas for tasks involving longer sequences.

The work in [**15**] comprehensively analyzed the correspondence of areas in the brain with components of Baddeley's model seen in Sec. 3.2. Figure 3.3 summarizes these findings.

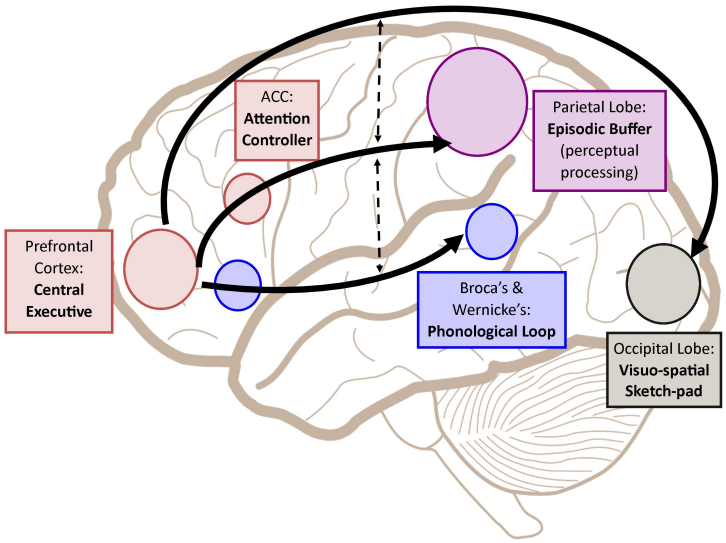


Fig. 3.3. A depiction of the components in Baddeley's multicomponent working memory model [**8**] mapped on to different brain regions. The control flow from the central executive and the information manipulation are indicated using the arrows. The ACC (Anterior cingulate cortex) acts as the attention controller (from *Jia Chai et al.* [**15**])

Other meta analyses such as [**92, 97, 89**] have determined specific areas in the lobes that are active in visual WM. Figure 3.4 highlights and lists these areas in the meta analysis of visual working memory tasks.

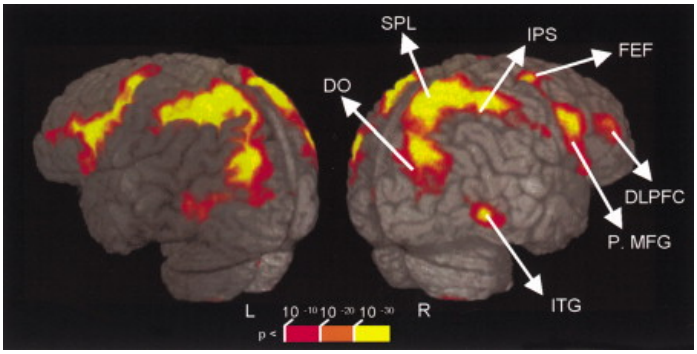


Fig. 3.4. A representation of the statistical map (group) of contrasts of functional activations observed in an fMRI visual working memory study [**92**].
The color bar shows uncorrected p values (yellow is better) and the labelled regions are : DLPFC - dorsolateral prefrontal cortex, DO - dorsal occipital, FEF - frontal eye field, IPS - intraparietal sulcus, ITG - inferior temporal gyrus, P. MFG - posterior middle frontal gyrus, SPL - superior parietal lobule

Thus, from all these studies, we remark that the visual working memory areas that are expected to activated are distributed over the frontal, temporal and occipital regions, forming a distributed pattern.

Chapter 4

Recurrent Neural Network Models

In this chapter, we describe the basic details about the various recurrent neural network models that we considered in our study and ways to use their representations.

Recurrent neural networks (RNNs) are different from feed forward neural networks in that they have additional lateral (self or temporal) connections in the hidden layers. These lateral connections with the same hidden layers are supposed to be from a previous*time step* and this gives rise to the phenomenon of *recurrence* in these networks. This feature of these networks enables them to model temporal dynamics of the inputs unlike the feed forward networks.

4.1. Vanilla Recurrent Neural Network (RNN)

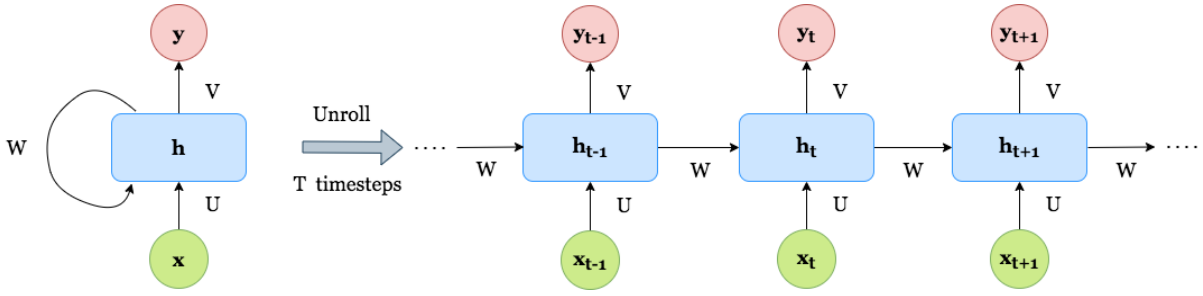


Fig. 4.1. A Simple RNN (with one hidden layer) shown along with an equivalent unfolded network highlighting the temporal connections. U, W and V are the input-to-hidden, hidden-to-hidden and hidden-to-output parametric matrices.

RNNs are very useful for sequential prediction and modeling tasks as they have a hidden state that allows them to integrate a good amount of information about the past in an efficient and scalable manner, updating it using non-linear dynamics in a deterministic way.

The weight sharing implemented in the lateral (temporal hidden-to-hidden) connections makes this possible and also makes the computation efficient.

Several hidden layers can be stacked together in a feed forward way with individual recurrent in units across each layer. Thus, in each layer except the first hidden layer, the inputs would be hidden states from the previous hidden layer. Such stacked networks are referred to as *Stacked RNNs* and they are able to improve training and give a superior performance while learning from inputs. The number of such layers can effectively function as a hyperparameter.

The dynamics in a stacked RNN operates by using the hidden unit activations from previous time steps and propagating it forward in time. It can be observed in Figure 4.1 that the parameters of the model are the input-hidden weight matrix W and hidden-hidden weight matrix U. The dynamics is characterized by the change in the hidden unit activations over time. This value is referred to as the *hidden state* or *context vector*. Each hidden state at time step t is computed using the current input x_t and the previous hidden state h_{t-1}.

For an activation function f, this can be mathematically described by :

$$h_t^{(1)} = f(W^{(1)}x_t + U^{(1)}h_{t-1}^{(1)})) \tag{4.1.1}$$

$$h_t^{(l)} = f(W^{(l)}h_t^{(l-1)} + U^{(l)}h_{t-1}^{(l)})) \quad \text{for} \quad l \geq 1 \tag{4.1.2}$$

An RNN that has been time-unrolled for finite time steps produces a network that can be considered as a feed forward network (as shown in Figure 4.1). The training of RNNs is done just like feed forward networks on their equivalent networks that unrolled across time [**50, 90**]. It is to be noted that the weight vector of the recurrent connections of the hidden layers across multiple timesteps remains the same (weight sharing). Thus, backpropagation happens through all these weights and as it is done across time (due to the presence of these lateral connections) and this process is called backpropagation through time (BPTT) [**121, 43**]. This unrolling is necessary in order to be able to train the network using backpropagation. The number of timesteps to unroll before training (T) is also a hyperparameter that can be chosen during training.

These basic types of RNNs, called as Simple RNNs or Vanilla RNNs are useful but are limited in capacity and are not very good at learning from input in the long-term past. This is because they suffer from the problem of exploding and vanishing gradients [**91**], making training difficult in scenarios requiring longer range memories from the past.

4.2. Long Short Term Memory (LSTM)

The LSTM (introduced in [**51**]) is a more complicated variant of an RNN with more powerful dynamics to consolidate information over longer timesteps. The "long term" memory is explicitly stored in a vector of memory cells referred to as *cell state* and denoted by c_t.

This is in addition to the hidden state h_t that is output. These two variables - cell state and hidden state constitute the overall state of each LSTM cell.

Although many LSTM architectures differ in their connectivity structure and activation functions, all LSTM architectures have explicit memory cells for storing information for long periods of time. The LSTM is able to overwrite the memory cell, retrieve it, or keep it for the next time step. This is achieved with the help of a mechanism called *gating* which uses additional parameters to operate *gates* to compute additional variables to control the flow of information in the network.

Figure 4.2 shows a typical LSTM cell with 3 different gates - forget gate that controls what part of information from the remembered past (previous cell state c_{t-1}) needs to be further remembered, input gate that controls what part of the current input (x_t) needs to be remembered and output gate that controls what part of the currently remembered information (cell state c_t) needs to be output as the hidden state h_t.

Each LSTM cell uses a separate set of input-hidden parameters (W) and hidden-hidden parameters (U), in addition to non-linearity such as a sigmoid function to compute the gate activation vectors (or states) f_t, i_t and o_t.

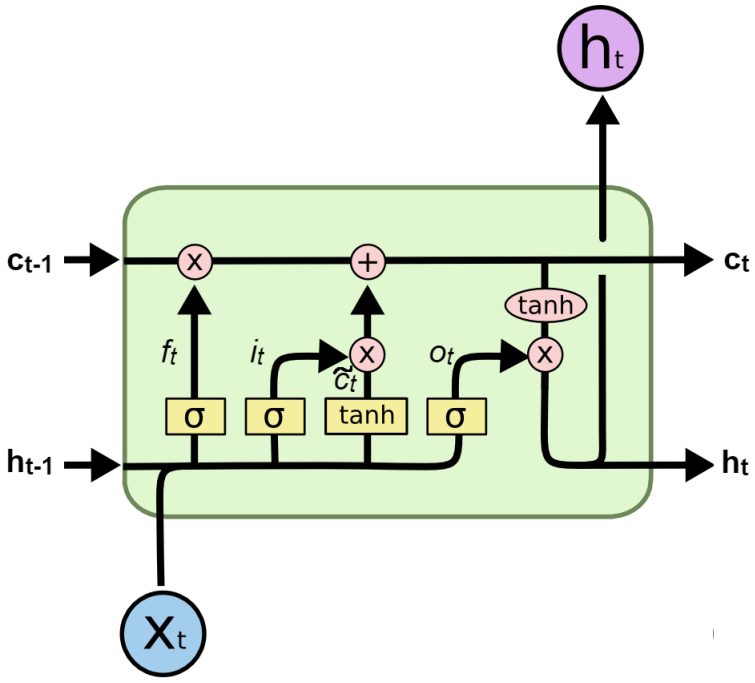


Fig. 4.2. Schematic diagram of a simple LSTM cell with a *tanh* activation function and gates - input, forget and output gates that use a sigmoid function (σ) to compute the respective gate activation vectors f_t, i_t and o_t; *tanh* for computing the cell activation vector $\tilde{c}_t$ using the current input x_t and previous hidden state h_{t-1}. The flow of information through these gates to compute the hidden state h_t and cell state c_t based on the previous cell state c_{t-1}, current cell activation vector c_{t-1} and the gate activations.

In addition, the cell has a set of parameters for computing a vector called the cell input activation $\tilde{c}_t$ from the current input x_t and the hidden state h_{t-1} from the previous timestep.

These equations below that describe the dynamics of a typical LSTM, where σ denotes the sigmoid function and both σ and *tanh* are element-wise operations. $\odot$ denotes the element-wise product operation (Hadamard product).

$$\begin{pmatrix} i_t \\ f_t \\ o_t \\ \tilde{c}_t \end{pmatrix} = \begin{pmatrix} \sigma \\ \sigma \\ \sigma \\ \tanh \end{pmatrix} \begin{pmatrix} U^{(l)} & W^{(l)} \end{pmatrix} \begin{pmatrix} h_t^{(l-1)} \\ h_{t-1}^{(l)} \end{pmatrix} \tag{4.2.1}$$

$$c_t^{(l)} = f_t \odot c_{t-1}^{(l)} + i_t \odot \tilde{c}_t \tag{4.2.2}$$

$$h_t^{(l)} = o_t \odot \tanh(c_t^{(l)}) \quad \text{for} \quad l \geq 1 \tag{4.2.3}$$

$$h_t^{(0)} = x_t \tag{4.2.4}$$

The parameters U and W can be considered to be the concatenated parameters of the input, forget, output gates and the cell input activation. The computed gate activation vectors are essentially act as mask to turn on or off different components of the cell state vector and are thus combined in different ways to compute the current cell c_t and hidden state h_t vectors.

As stated in Eq 4.2.2, the current cell state c_t is computed by combining a part of the previous cell state c_{t-1} (allowed to remember by the forget gate activation) and a part of the current cell input activation $\tilde{c}_t$ (allowed to pass by the input gate activation). Finally, Eq 4.2.3 uses the output gate activation to select the part of the current cell state c_t to be output from the cell as the current hidden state h_t.

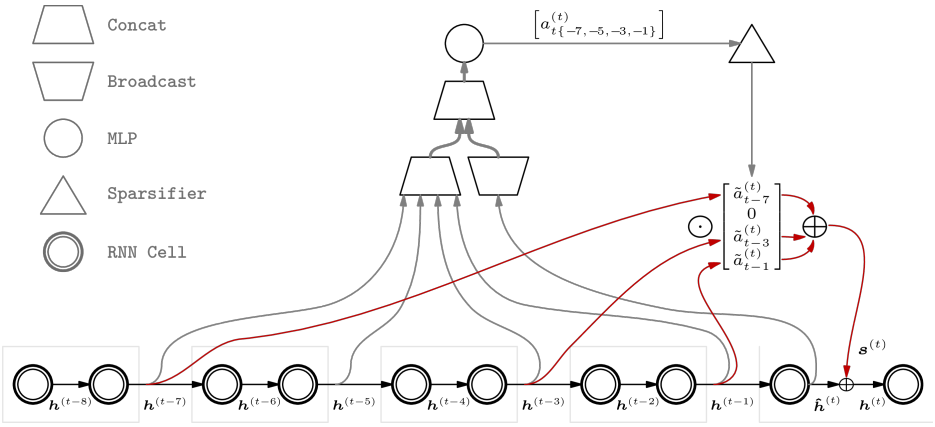
Thus, the gating mechanism allows for information from way back in the past to be stored in the memory cell and used when required.

4.3. Long Short Term Memory - Sparse Attentive Backtracking (LSTM-SAB)

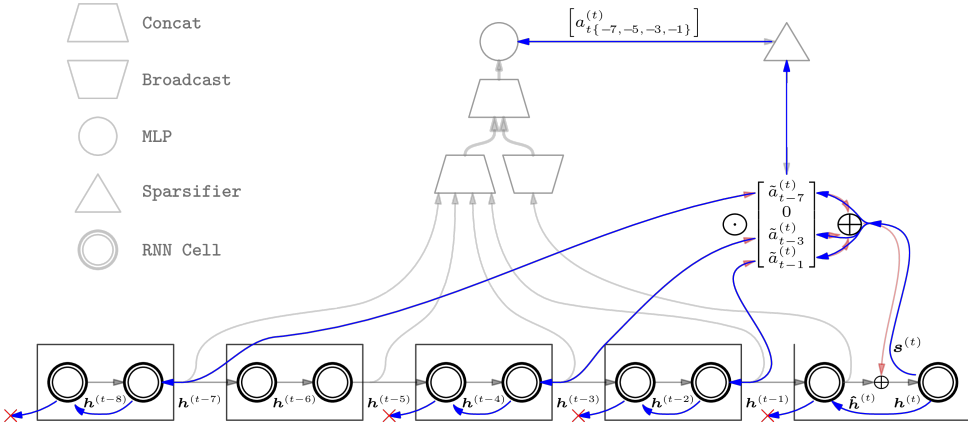
In several cases, all past states might not be relevant to make decisions during a given time step. When recalling memories, the brain is known to be reminded of only selected *relevant* memories from the past instead of all of them. A high-level approximation of this idea can be implemented to learn a set of sparse weights over the past states. This idea of weighing the past states by their level of relevance and combining them is referred to as attention and it can be learned [**10**]. *Sparse Attentive Backtracking* (SAB) is a mechanism

that is based on this idea and was introduced by *Ke et al.* [**58**] in 2018. It is an enhancement that can be used in recurrent neural networks to allow them focus on relevant past states.

LSTM-SAB is a version of the LSTM network augmented with the SAB mechanism. In terms of architecture, it is similar to LSTM but has an additional attention module which is a Multi Layer Perceptron (MLP) to compute the attention weights and a sparsifier. This type of network also maintains a macro-state or a set of memories M that are relevant.



(a) RNN-SAB Forward Pass : Sparse retrieval - The gray arrows show the computation of attention weight by the MLP through the broadcast and concatenation of the current provisional hidden state $\hat{h}^{(t)}$ with all the memories. The sparsifier selects and normalizes only the k_{top} greatest raw attention weights with some weights having on-zero values shown in red.



(b) RNN-SAB Backward Pass : Sparse replay - The gradients are passed to the hidden states selected in the forward pass and a local truncated backprop is performed around those hidden states. Gradients flow along the blue arrows and their truncation at the red crosses

Fig. 4.3. A schematic diagram depicting the operations of a SAB augmented RNN : (a) Forward pass of an RNN-SAB (b) Backward pass of an RNN-SAB; from [**58**]

As shown in Figure 4.3(a), during the forward pass the usual LSTM cell and the provisional hidden state $\hat{h}^{(t)}$ is computed. The MLP takes all the set of states in the stored

memories M and the current state as inputs and computes a vector of attention weights over all the past states. This attention weight vector is sparsified (a number of weights become zero) using a simple ReLU over the difference with the maximum attention weight among the k_{top} memories. The sparsified attention enables the retrieval of the most relevant memories. This sparse attention vector is weighted over the memory states in the set of stored memories M to compute another state s. The hidden state of the LSTM-SAB cell $\hat{h}^{(t)}$ is computed as the sum of the provisional hidden state and the state s. The output of the LSTM-SAB cell is computed with separate set of weights for the LSTM hidden state $\hat{h}$ and the state s. At every k_{att} time steps, the hidden state is added to the set of memories.

During the backward pass, the gradients are propagated along the paths to the subset of memories selected during the forward pass and their temporal predecessors situated k_{trunc} time steps before. This is indicated in Figure 4.3(b) where the sparse set of memories are said to be replayed for the backpropagation to happen.

Thus, the sparse retrieval and sparse replay of memories are used in the forward and backward passes respectively and these are the basic principles behind why the LSTM-SAB network performs efficiently.

4.4. Convolutional Long Short Term Memory Network (ConvLSTM)

The dynamics of the LSTM described above in Section 4.1 involve multiplication of the hidden and input vectors with the respective weight matrices to compute the different gates and cell states. However, *Shi et al.* [**101**] introduced a variant of LSTMs which replaced these product operations with convolutions. They demonstrate that spatiotemporal correlations are better captured in ConvLSTM network as observed with consistently better performance compared to equivalent fully-connected LSTM networks for prediction tasks.

This modification causes the weights in the layers to be convolutional filters which are able to localize on the hidden states. In addition, it also offers some computational advantages by reducing the overall number of array multiplication operations.

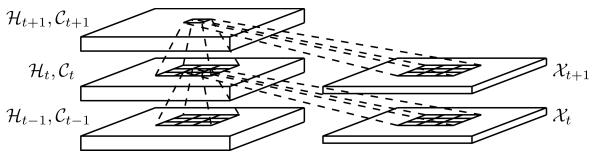


Fig. 4.4. A visual representation of the operation happening inside a ConvLSTM cell (from [**101**]). The convolution of the input at each timestep with separate kernels for the input and hidden states to compute the hidden and cell states is illustrated.

The following equations describe the computations happening in a ConvLSTM cell. Ws, Us and bs are the respective parameters. σ_g is the gate activation function which is usually the sigmoid function and σ_c is the cell activation function which is usually the $tanh$ function. $*$ denotes the convolution operation and $\odot$ denotes the element-wise product operation.

$$i_t = \sigma_g(W_i * x_t + U_i * h_{t-1} + b_i) \tag{4.4.1}$$

$$f_t = \sigma_g(W_f * x_t + U_f * h_{t-1} + b_f) \tag{4.4.2}$$

$$o_t = \sigma_g(W_o * x_t + U_o * h_{t-1} + b_o) \tag{4.4.3}$$

$$\tilde{c}_t = \sigma_c(W_c * x_t + U_c * h_{t-1} + b_c) \tag{4.4.4}$$

$$c_t = f_t \odot c_{t-1} + i_t \odot \tilde{c}_t \tag{4.4.5}$$

$$h_t = o_t \odot \sigma_h(c_t) \tag{4.4.6}$$

These networks are used to incorporate a spatial structure in learning the long-term dependencies in a set of sequential inputs. They have been proven to be useful in tracking objects in video data and forecasting and predicting complex multivariate time-series [**101, 106, 77**].

Chapter 5

Relating Representations of Models and Brains

In this chapter, we highlight about how representations from computational models such as deep neural networks can be useful. Then, we summarize some methods using which the representations in computational models and the brain can be related with each other and among themselves.

5.1. Learned Model Representations as features

Deep neural networks are parametric models that perform a sequence of computations to produce outputs. Trained deep neural are essentially composed of the set of optimized parameters and use them to compute some useful transformations of the data. The hidden layers in deep neural networks learn high-level representations from the data that can be used as *features*. These are referred to as features as they are meant to highlight some important aspects of the data that is transformed. It is this aspect of these networks that eliminates the need for explicit feature engineering. Thus, one useful advantage of trained deep neural networks and an interesting application is their usage to extract features from their inputs. These features can be used for analysis or other associated tasks [**127, 128**]. From a systems perspective, they are equivalently viewed as the *responses* produced by individual units or layers to the input data.

Neural networks that are trained using data samples can be used to understand the data that they were used to model with. The representations formed in convolutional neural networks have been analyzed in many studies [**72, 75, 11**]. In the case of images, these networks have been found to learn to process hierarchically organized visual representations such as edges, contours, textures, object parts, etc [**30, 129**]. These network-derived representations of images have also been found to be related to the processing of visual information in the brain [**66, 126**].

Recurrent neural networks are good choices over other options such as Hidden Markov Models for many sequential tasks also because of the rich representations that these networks use for learning and the useful features that can be derived from these representations. The representations capture statistical properties of the sequential data and can be analyzed further as done in [**76, 33**]. There have also been some attempts in interpreting the representations of RNNs [**49, 53, 122**] or using them as features for other tasks [**16**].

5.2. Comparing Model Representations

There have been studies that investigated the similarities between representations obtained from deep neural network representations and have made attempts to quantify them [**71**]. These consider data from two set of representations derived from models and use some analysis on them to identify some mathematical relationship and produce a similarity score.

Canonical Correlation Analysis (CCA)[**96**] is classic technique in statistics that computes a correlation metric by identifying a linear transformation that maximally correlates the two vectors spaces. The benefits and downsides of this method in the context of deep neural network representations have been studied [**82, 64**] and better variants such as Singular Vector CCA (SVCCA [**94**]) have been developed. The work in [**64**] surveys different approaches and introduces another method called Centered Kernel Alignment (CKA) to compute the similarity.

5.3. Comparing Model and Brain Representations

Similar to computational models, the brain can also be viewed as a system that produces responses as observed from its activity (electrical, magnetic, vascular, etc.) and behavior (actions/decisions in the real world) on being stimulated with some input conditions [**17**]. Thus, these activity patterns can be used to compare with representations from other deep neural networks (or computational models) or even from other patterns of brain activity. In order to understand the similarities between any two representations (computational models and brain activity) of the same or similar inputs, we need some standard ways to compare them. Especially, with brain representations, there are challenges in corresponding the patterns to one another. The methods listed in Sec. 5.2 may not always useful for this, owing to the complex and noisy nature of brain recordings.

While many approaches exist for comparing representations with those of brain activity, we briefly discuss two commonly used ways of comparing representations from neural networks (or any other computational models) and brains (response signals from recordings or behavior). These are based on some methods used in computational neuroscience [**27**].

5.3.1. Representational Similarity Analysis

Representational similarity analysis (RSA) as a general method was introduced in [**67**] to conceptualize a notion of representational distance between two sets of patterns by making use of the idea of second-level distances. For each set of activity patterns that need to be compared, a representational dissimilarity matrix (RDM) is constructed to model the distances between different subsets/samples in each set expressed by a dissimilarity metric. This matrix of first-level distances consists of values computed based on the distances between the patterns under each condition. The final similarity score is given by computing the distance between these two RDMs corresponding to the two sets. Thus, this method uses distances at two different levels to asses the similarity.

The distance (or dissimilarity) metrics at the two levels can be chosen according to the nature of the data. The most commonly used dissimilarity metrics are 1 - Pearson's correlation, 1 - Spearman's correlation and Euclidean distance. These RDMs capture the pattern of the activity distributed across the different conditions [**29**]. Another distance metric is chosen at the second level to compute using the two RDMs. Correlation-based metrics are oftenly used convenient choices. The choice of the distance metrics is very important in the application of this method and one needs to be very careful. The similarity needs to be extensively tested and validated using good statistical tests to ensure generalization.

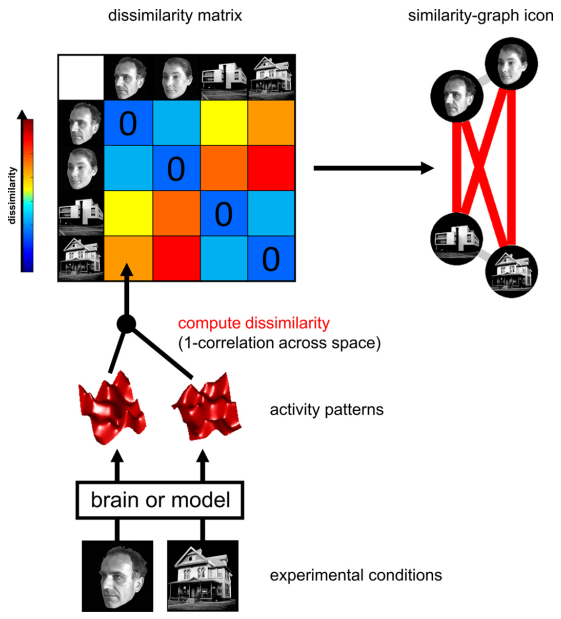


Fig. 5.1. A depiction of the construction of a representation dissimilarity matrix (RDM). A brain or model is used to compute a dissimilarity score between activity patterns (middle) obtained for all pairs of conditions (bottom) to feature in the respective matrix location (top). This corresponds to a certain similarity relationship of each of these conditions (right).

Many studies have successfully used this method to compare model and brain representations across many domains. Early work such as [**65, 68**] used RSA to compare representations between models, human and monkey brains in a visual task. Another study [**124**] used this method to establish the similarity between the visual areas in the brain of a monkey and convolutional neural network representations while in [**95**] this was extended to many different architectures. The study in [**18**] used convolutional neural networks and MEG recordings of object recognition to detect the hierarchy of visual areas in the brain. Whereas in [**80**] they were identified to be similar to both fMRI and MEG.

Thus, the benefits of RSA is its simplicity and the flexibility that it offers in the comparison of representations among and across different entities - brains, models, brain areas, model layers, individuals, species and modalities (eg. fMRI-MEG, EEG-fMRI).

The main disadvantage of RSA is that some feature dimensions in the representation that may not be relevant to the input could weaken the effect of the important dimensions in the similarity matrix and give an erroneous similarity. The remedy is to actually identify the relevant dimensions through other means. Also, this method is only able to quantify the representational similarity between two systems and it cannot be used to directly do predictive modeling of activity patterns. However, there have been some recent advances [**2, 34**] in attempting this.

5.3.2. Encoding Analysis

Encoding models have been long studied in Neuroscience [**83, 119**] and have resulted in some good understanding about the brain in terms of its behavior and function from this encoding perspective. This method is all about constructing an encoding model and analyzing it to estimate the similarity of representations.

Encoding models, in general predict the brain (or model) response patterns from input stimuli (or data). The space of all the possible input stimuli is considered as the *input space*. The objective of these models is to essentially learn to transform the input space into the representational space of the brain (or model), referred to as the *activity space*. Under this hypothesis, the features derived from models are considered to be situated in a *feature space* realized by the transformation of the input space by the model. So, the problem of encoding is about using the model features to learn a mapping to the target activity space.

The feature space is assumed to essentially represent the *code* of the inputs and similarly the brain activation patterns to contain a different code of the input stimuli. The basis of this approach relies on the idea that studying the mapping of the feature space to the activity space can reveal the similarity between these two codes. This similarity is measured by using metrics to measure the goodness of the prediction as a score between the actual and predicted activity patterns.

As the model is required to map one vector space to another by learning from data, this can be considered as a regression problem. Many statistical learning methods such as those discussed in [**107**], can be employed to model this regression. However, using a linear model to achieve this is very useful as it is a well-studied method [**83, 74, 123, 57**]. As a linear relationship is the simplest one that two vector spaces can share, it is very useful in identifying the similarities. Additionally, the lower capacity linear model offers an advantage in requiring lesser data for the fit. This approach assumes a *linearizing* feature space [**123**] as the non-linear mapping from the inputs linearizes the relationship between the features and the activity patterns. Under this method of modeling, the notion is that the non-linear transformed features of the inputs form the basis set of the activity subspace located inside the feature space. An example of this is illustrated in Figure 5.2.

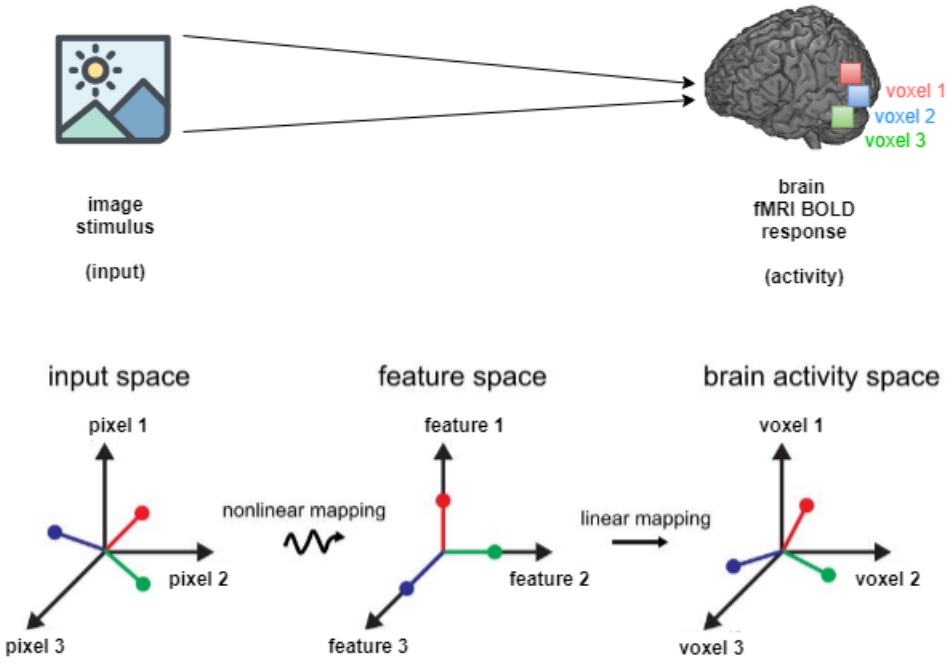


Fig. 5.2. An illustration of a linearizing encoding model that produces voxel responses from image pixels through a feature space. The input space of pixels (left) in the stimuli (left) are transformed using a nonlinear mapping to the features space by a computational model (middle). A linear mapping transforms this feature space into the brain activity space (right) of voxels.

In practice, regularized linear models have been found to be more advantageous for this linear encoding [**27**] and have been employed in several analyses [**54, 118, 60**].

After performing the regression, the next step is to use some metrics to quantify the generalization of the linear predictive model using validation strategies. Due to the limited

availability of brain activity data, cross-validation could be used. Some commonly used metrics are Pearson's correlation coefficient, coefficient of determination and mean absolute error. This score on the validation set is the measure of similarity between the representations.

Many studies have used convolutional networks to train on object detection and predict brain activity in visual regions of the brain activity for different modalities such as electrophysiological recordings [**125**] and fMRI [**48**]. Image visual features extracted from a large suite of visual tasks (segmentation, scene detection, etc.) were used to predict fMRI visual activity in [**118**], giving promising results. Representations of sounds (from neural networks) have been compared with the auditory cortex in [**86**] by predicting fMRI activity. Encoding analysis was also used in [**54**] with features derived from language models based on LSTMs to predict fMRI activity and analyze their similarities.

A good generalization in prediction is associated with the similarity in the receptive fields of both the model and the brain activity. This aspect can be used in identifying the computations happening in the brain areas that cause the changes in the activity (as in [**48, 61**]). Thus, the predictive power of the features at different sites in the brain can help reveal a computational hierarchy by matching different representations from the network to these sites. In [**48**], this method of comparing representations of CNNs has been used to detect the hierarchy of brain areas in the visual ventral stream. Similarly, in other studies, the hierarchy of the areas involved in speech [**23**] and audition [**61**] have also been detected.

In addition to enabling us to estimate the similarity between two sets of representations, this method helps us form hypotheses about the nature of these representations (neural code) by making predictions. It also facilitates in the tracing of hierarchical dependencies in the brain activity by corresponding brain areas with the elements in the network structure.

Article.

Recurrent neural network activations optimized for N-back task reveal functional activity of visual working memory

by

Pravish Sainath[1], Pierre Bellec[2], and Guillaume Lajoie[3]

([1]) Computer Science Dept., Université de Montréal
Mila – Quebec AI Institute

([2]) Psychology Dept., Université de Montréal
Centre de Recherche de l'institut Universitaire de Gériatrie de Montréal (CRIUGM)

([3]) Math & Stats Dept., Université de Montréal
Mila – Quebec AI Institute

This article will be submitted to Human Brain Mapping.

The majority of the work described in this article was done by Pravish Sainath. Dr. Pierre Bellec provided the initial idea for the project, the data in addition to mentoring and continued feedback. Dr. Guillaume Lajoie offered guidance and valuable comments throughout the project.

Résumé. Ces dernières années, de nombreuses études ont pu relier les réseaux de neurones artificiels au cerveau dans des tâches de perception, comme la vision. Avec le succès croissant des réseaux de neurones dans le traitement des séquences et des souvenirs, nous les utilisons pour étudier un type fondamental de mémoire appelé la mémoire de travail visuelle. Pour cela, nous utilisons la tâche cognitive populaire appelée tâche N-back, dans laquelle les sujets sont présentés avec une séquence d'images, dont chacune doit être identifiée pour savoir si elle a déjà été vue ou non. Nous proposons une procédure de modélisation qui consiste à entraîner des réseaux de neurones récurrents, en particulier des LSTM, pour résoudre la tâche N-back en les entraînant sur des stimuli d'image. Nous utilisons les représentations apprises de ces différents types de réseaux pour construire des modèles de codage de l'activité IRMf BOLD enregistrés à partir de sujets qui ont résolu la même tâche. Enfin, nous analysons la qualité de prédiction de l'activité cérébrale fonctionnelle dans ces modèles d'encodage pour évaluer leur relation avec diverses zones cérébrales à différents niveaux. Dans l'ensemble, les résultats suggèrent que les représentations apprises par les réseaux récurrents sont liées à l'activité fonctionnelle de la mémoire visuelle de travail dans le cerveau, et certaines propriétés architecturales de ces réseaux donnent des représentations plus alignées avec le cerveau.
Mots clés : réseau de neurones récurrents à mémoire court et long terme (LSTM), mémoire de travail, IRMf, modèle de codage, similarité de représentation

Abstract.

In recent years, many studies have been able to relate artificial neural networks with the brain in tasks of perception, such as vision. With the increasing success of neural networks in processing sequences and memories, we use them to study a fundamental type of memory called the visual working memory. For this, we use the popular cognitive task called the N-back task, in which the subjects are presented with a sequence of images, each of which needs to be identified as to whether it was already seen or not. We propose a modeling procedure that involves training recurrent neural networks, specifically LSTMs, to solve the N-back task by training them on image stimuli. We use the learned representations of these different types of networks to build encoding models of the fMRI BOLD activity recorded from subjects who solved the same task. Finally, we analyze the quality of prediction of brain activity in these encoding models to assess their relationship with various brain areas at different levels. Overall, the results suggest that the representations learned by the recurrent networks are related to the functional activity of the visual working memory in the brain, and certain architectural properties of these networks yield representations that are more aligned with the brain.
Keywords: long-short term memory (LSTM) networks, working memory, fMRI, encoding model, representational similarity

1. Introduction

Working memory is a system that enables an agent or an animal to maintain and manipulate information over a short period of time. It is a key aspect of intelligence and it is essential for almost all cognitive tasks like imagination, planning, action or decision making [**78**]. This type of memory has been extensively studied in Cognitive Neuroscience and Psychology as it is an important aspect for understanding the processing of thoughts [**8, 7**].

The mechanisms of complex computations behind the functioning of the working memory that enable its interfacing with other brain regions and its contribution to behavior are not fully understood. A good paradigm to investigate this question is analyzing the functional Magnetic Resonance Imaging (fMRI) activity of humans performing working memory (WM) tasks [**20, 89**]. Traditional fMRI studies of memory tasks have identified some regions associated with this function, based on group analyses [**19, 14**]. The modeling in these studies has been mostly based on the task conditions rather than representations of the stimuli [**116, 89, 97, 109, 109**]. There is a need for a computational model that explains the WM function based on the change in the stimuli.

On the other side, in modern Artificial Intelligence, neural networks with temporal (or lateral) connections are common architectures that have the capacity to operate with memory-oriented sequential tasks [**50**]. This architectural aspect of recurrence is based on the inductive biases from the brain. Lately, the rise of memory-augmented networks [**51, 10**] is state-of-the-art. This study aims to bridge the gap between the fMRI study of working memory and the development of deep neural networks that can inform on how the brain's WM works.

Visual Working Memory is used to refer to the short-term storage and manipulation of perceived visual information in memory. It allows us to hold a visual snapshot for a few seconds after we stopped seeing the stimuli. For a duration of a few seconds, a part of the stimuli is transferred into visual working memory. Many cognitive tasks test this aspect of the memory. The N-back task (introduced in [**63**]) is a very standard memory task where a sequence of stimuli are presented one after the other and the participants identify a match with the stimuli seen in the trial N steps back in time.

Among the variants of recurrent neural networks (RNNs) capable of remembering information over time, we choose the long short-term memory network (LSTM) for this study. The reason behind choosing LSTMs is because of the way they are parameterized and equipped with an explicit memory cell that accumulates state information with the help of multiple gates for efficiently carrying forward information over time by learning good representations of the stimuli. This representation that encodes information about the stimuli is what we attempt to leverage in the study.

Many studies such as [**60, 124, 79**] have explored human/animal visual perception and their similarity with artificial neural networks across various neuroimaging modalities. Studies such as [**62**] have highlighted the necessity of recurrent models for modeling vision tasks as they are able to capture well the correlations over time. [**47**] have extensively used deep neural networks to model fMRI voxel responses in the visual stream. Visual stimuli in the brain are relevant for functions beyond perceptual processing and the next step in this direction is memory. While the modeling of visual tasks in fMRI have seen advances in incorporating representations of the stimuli, for memory it has not been significant. This is where representations learned by recurrent neural networks could potentially offer an opportunity.

The principal objective of this study is to design and implement a pipeline to model the patterns of functional activity in the human brain during a working memory (image N-back) task using stimuli feature representations derived from various trained recurrent neural networks. The model uses representations from neural networks that are trained to solve the same tasks as the brain. We establish the better quality of prediction by this model in regions relevant to working memory. Our approach offers a better strategy compared to traditional modeling methods in fMRI [**109, 38**] by making use of features more relevant to the stimuli.

The predictability of the functional activity from the recurrent neural network-derived features is used to evaluate their alignment with the brain [**83**]. We demonstrate the superiority of these features over (memory less) convolutional neural network features and study their relationship with different brain regions. Overall, our study provides a recipe for a way of modeling fMRI memory tasks and highlights the similarity of recurrent neural network representations and brain activity for this (N-back) task.

2. Methods

In this section, we first present an overview of functional magnetic resonance imaging in Sec. 2.1, then outline the neuroimaging experimental setup of our study in Sec. 2.2, followed by the steps carried out to extract the BOLD time series data in Sec. 2.3.3. We then describe the models of the task network used to learn artificial neural network representations in Sec. 2.4 and finally the encoding model to relate these representations to the brain in Sec. 2.5.

2.1. Functional Magnetic Resonance Imaging

When the neurons in a certain region of the brain are active, there is change in the metabolic activity in this region due to the energy needed by these cells. The concentration of oxygenated blood increases in these active regions compared to the deoxygenated blood. These two types of blood possess different magnetic properties because of the oxygenated and deoxygenated hemoglobin that they carry respectively.

Magnetic Resonance Imaging (MRI) is a technique that images tissues by aligning the atomic nuclei to a magnetic field and measuring the electromagnetic signal emitted by them. An MRI scanner can distinguish the magnetic properties of the oxygenated and deoxygenated hemoglobin in the brain. Thus, with the help of an MRI scanner, it is possible to produce images of the brain with these differing contrasts of such cerebral *hemodynamic* activity. This method is called functional magnetic resonance imaging (fMRI). The signal that is measured is referred to as the blood-oxygen-level-dependent (BOLD).

The fMRI BOLD signal measures the metabolic activity of neurons instead of directly measuring the neuronal activity. Despite being an indirect measure of the actual neuronal activity, it is a good choice for studying the functional activity of the brain as these two types of activities are coupled. When compared to many other methods of functional imaging, fMRI is non-invasive and offers an acceptable spatial resolution to localize brain activity. It is particularly useful in our study as the visual working memory that we attempt to model is distributed over many regions of the brain and as it allows for recording the activity across the whole brain.

Typically, in studies of cognitive processes, participants are asked to perform certain carefully designed tasks inside the scanner. A single stretch of recording is referred to as a *session*. A *session* can be composed of many *blocks* of different task and rest periods. At specific time intervals within each block, the participants are asked to perform a *trial* in which they can be presented with some *stimuli* and/or be asked to do some action.

At every fixed time interval called the repetition time (TR), a three-dimensional image of the BOLD activity across the entire volume of the brain is captured by the scanner. The individual units of recorded data are called the volume elements or *voxels* with their intensities representing the strength of the BOLD signal at that location. Thus, a single session of fMRI data recording is a time-course of intensities for each voxel that is part of the three-dimensional brain volume. The time series of voxels that are clustered to represent some anatomical/functional structures called *parcels* can be combined to obtain the time series corresponding to those parcels.

2.2. Experimental Setup and Design

The dataset used in this study, referred to as the Human Connectome Project test-retest (hcptrt, 2020-alpha2 release[1]), was collected as a part of the Courtois Neuromod Project [**13**]. Each participant performed the functional localizer tasks developed by the Human Connectome Project [**114**] including the N-back task (working memory task) over 15 different sessions. The data corresponding to the cognitive tasks other than the working

[1] `https://docs.cneuromod.ca/en/2020-alpha2/DATASETS.html#hcptrt`

memory (WM) task were not considered in our study. Before each task, participants were given detailed instructions and examples, as well as a practice run.

In the WM task, the participants were presented sequences of images and were asked to make associations based on the nature of images seen in the recent past. For example, in the 2-back task they had to respond whether each image matched the one that was seen 2 trials back (referred to as the *target*). Whereas, in the 0-back task the very first image (cue in Fig. 0.1) shown in the sequence is *target* and they had to indicate to indicate if each image matched the target.

Each session was approximately 5 minutes long and was composed of two types of sub tasks: a category specific representation (0-back) and a working memory task (2-back). The participants were presented with task blocks of either places, tools, faces, and body parts. Within each run, all 4 types of stimuli were presented in block, with each block being labelled as a 2-back task (participants needed to indicate using button presses if they saw the same image two images back), or a version of a 0-back task (participants were shown a target at the start of the trial and they needed to indicate using button presses if the image that they were seeing matched the target). There were thus 8 different event types for the stimulus among : place, tools, face or body, and N-back type among 0-back and 2-back. Each stimulus image was presented for 2 seconds, followed by a 500 ms inter-stimulus interval. Each of the 2 runs included 8 event types with 10 trials per type, as well as 4 fixations blocks (15 secs).

The data was recorded on a 3T MRI Siemens Prisma scanner at the *Unité de Neuroimagerie Fonctionnelle* (UNF[2]). The scanning settings used a repetition time (TR) of 1.49 seconds, 60 slices, an acquisition matrix of 96x96 and voxels of size 2 mm x 2 mm x 2 mm. Figure 0.1 illustrates the schematic view of a recorded run. The protocol for the experiments was approved by the local ethics institutional review board, the *Comité d'éthique de la recherche vieillissement-neuroimagerie* and all participants provided informed consent to participate to the study.

2.3. Data Preprocessing

As in all fMRI studies, the collected raw data is not directly usable for analysis and required to be processed before use. It was preprocessed using a standard pipeline which yielded many derivatives and files. Results included in this manuscript come from preprocessing performed using fMRIPrep 20.1.1+38.g8480eabb ([**32, 31**]; RRID:SCR_016216), which is based on Nipype 1.5.0 ([**44, 45**]; RRID:SCR_002502).

[2] `https://unf-montreal.ca`

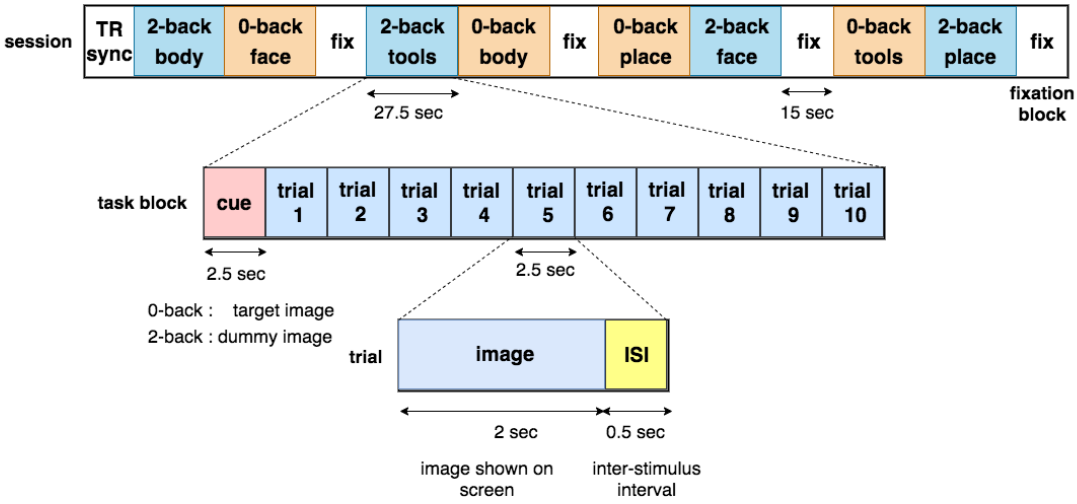


Fig. 0.1. Summary of the event structure in a session of the working memory (WM) task in the Courtois Neuromod HCP test-retest (hcptrt) dataset. An example is shown of a 2-back tools block composed of a cue and 10 trials in each of which an image is sown for 2 seconds with a gap of 0.5 seconds.

2.3.1. Anatomical data preprocessing

The T1-weighted (T1w) image was corrected for intensity non-uniformity (INU) with `N4BiasFieldCorrection` [**112**], distributed with ANTs 2.2.0 [[**5**], RRID:SCR_004757], and used as T1w-reference throughout the workflow. The T1w-reference was then skull-stripped with a Nipype implementation of the `antsBrainExtraction.sh` workflow (from ANTs), using OASIS30ANTs as target template. Brain tissue segmentation of cerebrospinal fluid (CSF), white-matter (WM) and gray-matter (GM) was performed on the brain-extracted T1w using `fast` [FSL 5.0.9, RRID:SCR_002823, [**130**]]. Volume-based spatial normalization to one standard space (MNI152NLin2009cAsym) was performed through nonlinear registration with `antsRegistration` (ANTs 2.2.0), using brain-extracted versions of both T1w reference and the T1w template. The following template was selected for spatial normalization: ICBM 152 Nonlinear Asymmetrical template version 2009c [[**37**], RRID:SCR_008796; TemplateFlow ID: MNI152NLin2009cAsym].

2.3.2. Functional data preprocessing

For each of the BOLD runs found per subject (across all tasks and sessions), the following preprocessing was performed. First, a reference volume and its skull-stripped version were generated using a custom methodology of fMRIPrep. A deformation field to correct for susceptibility distortions was estimated based on two echo-planar imaging (EPI) references with opposing phase-encoding directions, using `3dQwarp` [**22**] (AFNI 20160207). Based on the estimated susceptibility distortion, an unwarped BOLD reference was calculated for a

more accurate co-registration with the anatomical reference. The BOLD reference was then co-registered to the T1w reference using `flirt` [FSL 5.0.9, [**56**]] with the boundary-based registration [**46**] cost-function. Co-registration was configured with nine degrees of freedom to account for distortions remaining in the BOLD reference. Head-motion parameters with respect to the BOLD reference (transformation matrices, and six corresponding rotation and translation parameters) are estimated before any spatiotemporal filtering using `mcflirt` [FSL 5.0.9, [**55**]]. The BOLD time-series (including slice-timing correction when applied) were resampled onto their original, native space by applying a single, composite transform to correct for head-motion and susceptibility distortions. These resampled BOLD time-series will be referred to as preprocessed BOLD in original space, or just preprocessed BOLD. The BOLD time-series were resampled into standard space, generating a preprocessed BOLD run in ['MNI152NLin2009cAsym'] space. First, a reference volume and its skull-stripped version were generated using a custom methodology of fMRIPrep. Several confounding time-series were calculated based on the preprocessed BOLD: framewise displacement (FD), DVARS and three region-wise global signals. FD and DVARS are calculated for each functional run, both using their implementations in Nipype [following the definitions by [**93**]]. The three global signals are extracted within the CSF, the WM, and the whole-brain masks. Additionally, a set of physiological regressors were extracted to allow for component-based noise correction [CompCor, [**12**]]. Principal components are estimated after high-pass filtering the preprocessed BOLD time-series (using a discrete cosine filter with 128s cut-off) for the two CompCor variants: temporal (tCompCor) and anatomical (aCompCor). tCompCor components are then calculated from the top 5% variable voxels within a mask covering the subcortical regions. This subcortical mask is obtained by heavily eroding the brain mask, which ensures it does not include cortical GM regions. For aCompCor, components are calculated within the intersection of the aforementioned mask and the union of CSF and WM masks calculated in T1w space, after their projection to the native space of each functional run (using the inverse BOLD-to-T1w transformation). Components are also calculated separately within the WM and CSF masks. For each CompCor decomposition, the k components with the largest singular values are retained, such that the retained components' time series are sufficient to explain 50 percent of variance across the nuisance mask (CSF, WM, combined, or temporal). The remaining components are dropped from consideration. The head-motion estimates calculated in the correction step were also placed within the corresponding confounds file. The confound time series derived from head motion estimates and global signals were expanded with the inclusion of temporal derivatives and quadratic terms for each [**100**]. Frames that exceeded a threshold of 0.5 mm FD or 1.5 standardised DVARS were annotated as motion outliers. All resamplings can be performed with a single interpolation step by composing all the pertinent transformations (i.e. head-motion transform matrices, susceptibility distortion correction when available, and co-registrations

to anatomical and output spaces). Gridded (volumetric) resamplings were performed using `antsApplyTransforms` (ANTs), configured with Lanczos interpolation to minimize the smoothing effects of other kernels [**70**]. Non-gridded (surface) resamplings were performed using `mri_vol2surf` (FreeSurfer).

2.3.3. BOLD time series extraction

The preprocessed BOLD data is not directly used by machine learning algorithms for the purpose of modeling. Specific steps were carried out to further improve the signal-to-noise ratio and remove unnecessary artifacts from the data. This was done using the packages provided by Nilearn 0.6.2 [**1**]. These steps are very standard in fMRI data analysis and have been used in several previous studies [**116, 79, 47, 60, 54**]. They help satisfy some statistical assumptions about the data and seek to remove some aspects that are not useful, enhancing the quality of the data analysis. They are summarized below:

- *Masking* : The anatomical mask derived from 2.3.1 was used to remove the voxels from the non-brain regions (background and skull). The masked fMRI data that is 4-dimensional was converted into a data matrix composed of the BOLD time series of each brain voxel.
- *Detrending* : A consistent linear trend was eliminated in the BOLD time series by considering the residuals from the linear regression of the signal on its scan time.
- *Spatial smoothing* : A 3-dimensional Gaussian spatial smoothing filter was applied with a full-width half-maximum (fwhm) of 6 mm.
- *Normalization* : The BOLD signals were then normalized by Z-scoring the time series (subtracting each voxel value in the time series with the univariate mean and dividing by the univariate standard deviation over a session) ensuring that the variance of the time series is 1.
- *Confounds Removal* : The confounds of the 6 motion parameters (3 translation and 3 rotation generated by fMRIPrep) were used to regress on the signal and adjust them.
- *Block Extraction* : The BOLD time series of these voxels for the specific task blocks (2-back body, 0-back face, etc.) were extracted from the BOLD signals of each session based on the event onset timings provided as a part of the dataset.

The steps listed above preprocess and extract the time series of each voxel in the brain data for each task condition. We also required to perform analysis on parcels in the brain. So, we used Multiresolution Intrinsic Segmentation Template (MIST, developed by *Urchs et al.* [**113**]) which is a brain atlas of different parcellated regions across various cortical and subcortical regions labeled in a hierarchy. The MIST atlas was applied as a mask on the BOLD images to obtain the time series of each parcel in the brain volume for the task blocks. All the steps listed above were carried out on these parcels to obtain the parcel time series.

Thus, the final preprocessed data for each subject is essentially a large set of time series (of voxel or parcels) belonging to different task blocks corresponding to specific stimuli image sequences.

2.4. Task Network Model

The task network model is a neural network that is trained to perform the N-back task based on labelled sequences of images. It is composed of two blocks - recognition and memory modules, as shown in Figure 0.2.

The number of pixels is very high in the stimuli images, making it hard for many recurrent models to be trained directly on image sequences. The input images need to be converted to a different representational space for easier training. Therefore, this modular setup with two different blocks is necessary. The purpose of the recognition module is to extract necessary features from the input images and effectively transform them to a form that is easier for the memory module to use the input to solve the N-back task.

Different variants of recurrent network models, all with this general architecture, are used to elucidate the effect of distinct computational mechanisms: LSTM has explicit memory cell operations that are very relevant to a memory task like the N-back task, LSTM-SAB offers a sparse selection of memories from the past, and ConvLSTM uses more efficient convolution operations. In general, these networks differ only in the way in which they use the information from the input and the past to solve the task.

All models trained for the task network use images from the ImageNet [**25**] dataset. This dataset was chosen as it contained images close to the stimuli images used in the task from [**114**]. A good subset of the ImageNet dataset was selected to match the distribution of the stimuli dataset and achieve good generalization. Example images are shown in Sec. A.1.

The details about the training of the recognition module are outlined in Section 2.4.1 and about training the memory module are explained in subsequent Section 2.4.2. Additional details on the hyperparameters of the trained models are specified in Appendix A in Sec. A.2 for the recognition module and A.3 for the memory module.

2.4.1. Convolutional Neural Network

The network architecture used is popularly known as VGG-16 (adopted from [**102**]) and is a very common one in computer vision. It is a very deep feed-forward network that consists of convolutional and pooling layers stacked on top of each other in the fashion depicted in the recognition module in Figure 0.2, followed by 3 layers of fully-connected layers.

A VGG-16 model pretrained on performing object categorization on the ImageNet dataset (1000 categories) was taken initially, before fine tuning the network to a smaller set of labels involving the super-categories that are related to the task stimuli: people, scenes and

instrumentation. The 3 fully connected layers were truncated from this network before using them with the memory modules in sequential N-back task. The features extracted from the final layer activations ($\mathbf{q_t}$ in Figure 0.2) for the stimuli images are used as a baseline in the encoding models for comparing the effect of the memory module.

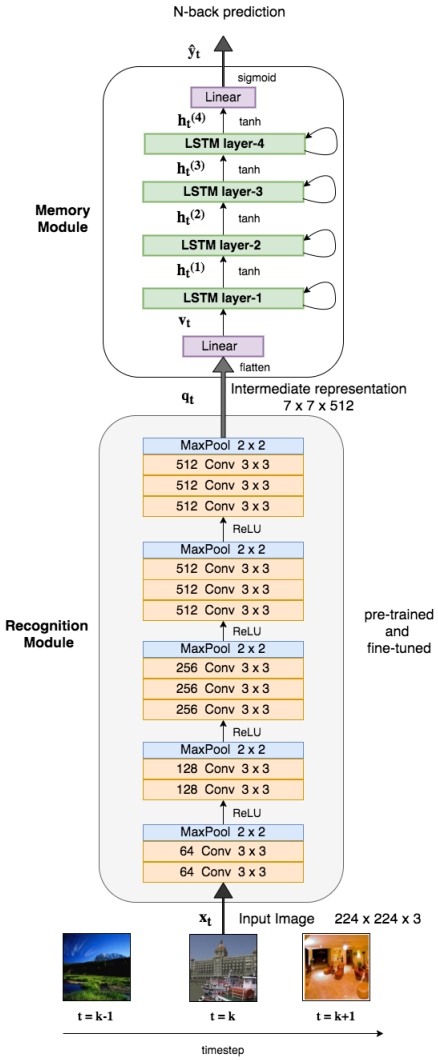


Fig. 0.2. The overall architecture of the task used in this study : It is composed of 2 components - (i) The Recognition Module composed of a truncated CNN that is pre-trained and fine-tuned for image recognition and (ii) The Memory Module with multiple stacked recurrent layers with a final linear layer (with sigmoid activation) for binary classification. The intermediate representation between these two modules is flattened and passed through a linear layer.

2.4.2. LSTM Recurrent Network

First introduced in [**51**], LSTMs are commonly used building blocks for using neural networks to learn from sequential data.

A 4-layered stacked LSTM with a hidden dimension of 512 between the layers was used as shown in Figure 0.2. The truncated CNN (recognition module) is connected to the memory module. For all the recurrent models, the training sequences were composed of the images ($\mathbf{x_t}$) sampled from the ImageNet dataset from the super-categories: people, animals, scenes and artefacts. These images were normalized with the mean and standard deviation of the subset and also randomly gray scaled as the task stimulus consisted some gray scale images. The labels were binary labels ($\mathbf{y_t}$) indicating 0 for no N-back match and 1 for N-back match computed through code.

The task network predicts a sigmoid value $\hat{\mathbf{y}}_\mathbf{t}$ at each time step indicating the probability that the output is 1. As it is a sequential binary classification task, the network is optimized to learn weights to minimize the binary cross-entropy (BCE) loss given by :

$$\mathcal{L} = \sum_{t=0}^{T} -\mathbf{y_t} \log(\hat{\mathbf{y}}_\mathbf{t}) - (1 - \mathbf{y_t})(1 - \log(\hat{\mathbf{y}}_\mathbf{t}))$$

T refers to the sequence length of time unrolled network which was set to 10 to match the N-back task condition. The initial values of hidden state vectors $\mathbf{h_0}^{(\mathbf{l})}$ were reset to all zeros at the start of every sequence of images to refresh the context as the context from the previous sequence is not relevant. The first 2 blocks of the recognition module were frozen during the LSTM training to reduce computational cost due to backpropagation and the all the remaining layers were trained.

2.4.3. LSTM-SAB Recurrent Network

This network is very similar to the LSTM but has an architectural enhancement that offers advantages in the way of using the past hidden states.

SAB stands for Sparse Attentive Backtracking, and it uses a differentiable and sparse attention mechanism to make selections from past states effectively by retrieving a minimum possible number of them. It employs a dedicated memory for storing past hidden states. An attention module made of a multilayer perceptron (MLP) that computes a set of attention weights from the past states stored in memory and the current state. These attention weights that are sparsified to select only a few memories (with non-zero attention weights). These selected memories are weighted by the attention weights and combined to compute another state vector that is added to the current hidden state. This is the mechanism used to retrieve memories sparsely. During the backward pass, these sparsified attention weights are used to identify the relevant past hidden states from memory store and propagate the gradients only in their locality instead of all past states. This is referred to as the sparse replay. Thus, in effect, this model strengthens the computational link between the hidden state with the most relevant past hidden states.

The additional parameters involved are the weights in the neural network corresponding ot the attention module. The additional algorithmic hyperparameters involved are:
k_{top} - the number of relevant memories to choose, k_{att} - the number of time steps to wait for adding the state into memory, and k_{trunc} - the number of previous states of the selected memories to propagate the gradient. These were chosen based on optimal validation performance.

This network also used a 4-layer stacked LSTM with the hidden dimension of 512 for each layer the network as shown in Figure 0.2. The data and loss function were exactly same as the LSTM described in Section 2.4.2. The detailed algorithms for training a LSTM-SAB are described in [**58**].

2.4.4. ConvLSTM Recurrent Network

ConvLSTM was initially proposed and used in [**101**] and popularly used to model the visual system in [**84**]. This type of LSTM is very similar to the original, vanilla variant. The difference is that the cell-level matrix product operations are replaced by convolution operations for hidden and input vectors. All the inputs, cell outputs, hidden states and gates of the ConvLSTM are 3D tensors with the first dimension indicating time and the last two dimensions as spatial dimensions. The computations in the ConvLSTM determine the current state of a certain cell in the grid of the input using the inputs and past states of its local neighbors. This type of network is particularly useful in tracking objects across sequences of images.

We use a ConvLSTM with a hidden vector of size 512 x 4 x 4 for the training of the model similar to the LSTM described in Section 2.4.2.

2.4.5. Random Network

This is the same network as shown in the Figure 0.2 but without any training. The weights present in this network are randomly initialized values. For each layer, the weights are independently initialized using a normal distribution with zero mean and a very small standard deviation chosen based on the dimensions of the current and previous layer according to Xavier's initialization [**40**]. Precisely, for a layer l this standard deviation used is the value given by $\frac{\sqrt{2}}{\sqrt{dim(\mathbf{h^{(l-1)}})+dim(\mathbf{h^{(l)}})}}$. The features extracted from the recurrent layer activations for the stimuli images in this network are also used as a baseline in the encoding models for understanding the effect of the trained network representations.

2.5. Encoding Model

The purpose of the encoding model is to use the trained task networks to derive features and map them to the brain responses in the neuroimaging data. As with many studies

using encoding analysis [**83, 54, 118**], this mapping is carried out using a regression model to regress the task network-derived features on to the fMRI BOLD activity. It is done to examine how these features are related to the BOLD activity (recorded during the WM task) in different regions of the brain. It is a key step to determine how well the representations learned by these artificial neural networks are able capture the dynamics of the brain activity.

Figure 0.3 shows the overall set of procedures carried out in this section. The end result is to obtain statistical brain maps that indicate the predictability of the brain activity from the task network-derived features correspondong to the stimuli used in the WM task.

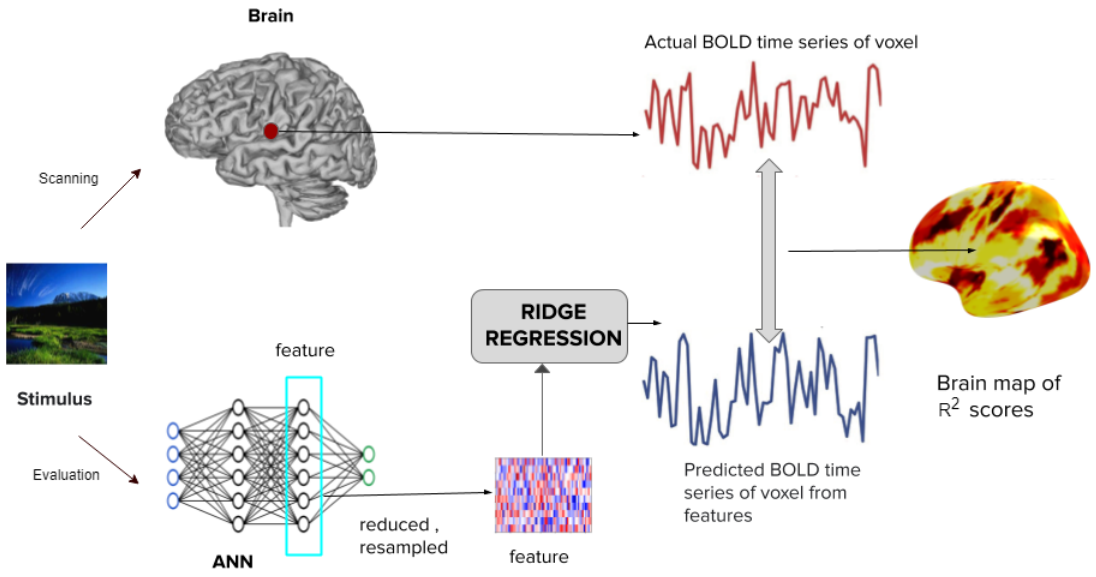


Fig. 0.3. Schematic diagram of the encoding model. The stimuli images used while scanning the human participant are evaluated using the trained ANNs (task networks) to extract features. These features are reduced and resampled before performing a ridge regression to predict BOLD signals. The R^2 score between the predicted and actual BOLD signals is computed for each voxel and mapped on to the brain volume to produce a brain map.

In Sec. 2.5.1, we discuss the extraction of features from the trained networks. However, it is not possible to use the features from the trained task network to readily in the encoding model (regression model). These extracted features are required to be reduced (Sec. 2.5.2) and resampled (Sec. 2.5.3) before regressing them. The details about the regression model are described in Sec. 2.5.4 and in Sec. 2.5.5, we highlight a way to account for the hemodynamic delay in the BOLD signal.

2.5.1. Feature Extraction

The stimuli images data from the fMRI task were used in the trained task network models to generate task-relevant features. The stimuli images used during the fMRI recording are

resized to the image size 224 x 224 that is acceptable by the recognition module of the task network.

The recurrent layer activations generated in the neural network are depicted as $h_t^{(1)}$, $h_t^{(2)}$, $h_t^{(3)}$ and $h_t^{(4)}$ in Figure 0.2. These hidden vectors can effectively serve as the *contextual embeddings* to represent the summary of the *task-relevant* stimulus sequence (context seen by the network) until that point t. The hidden vectors computed are transmitted to the next timestep of the recurrent cells, as well as to the higher feed-forward layers for computing the task output. Thus, the learning of weights in the recurrent network during backpropagation conditions these hidden vectors to be task-relevant while carrying the information from the past timesteps. This can be thought of as being similar to self-supervised word embeddings learned using autoregressive models of words from a text corpus, except that this task needs to be a supervised one.

2.5.2. Feature Reduction

The number of features dimensions (or components) in the contextual embeddings are much higher than the number of samples. When these are directly used in a regression model, it can affect stability of the regression. In addition, the contextual embeddings derived from the stimuli presented during the same block of a session are correlated with each other because of the dependence of each hidden vector in the recurrent neural network on that of the previous timestep. Thus, it is useful to both reduce the dimensionality and transform them to a different space oriented along the direction with the maximum variance. We reduce the feature dimensions by performing a Principal Components Analysis (PCA) decomposition on the set of embeddings from the same layer for all the data. We then choose the top principal components (PCs) that explain 95% of the variance in the original data (for example, about 127 components out of 512 feature dimensions in LSTM layer-3).

2.5.3. Resampling

The BOLD signal is a measure of vascular changes that are indirectly related to neural activity, and is traditionally modeled as a convolution operation with an one-dimensional kernel called the *hemodynamic response function* (HRF) [**41, 38**]. This function represents the impulse response of the BOLD signal for a stimulus.

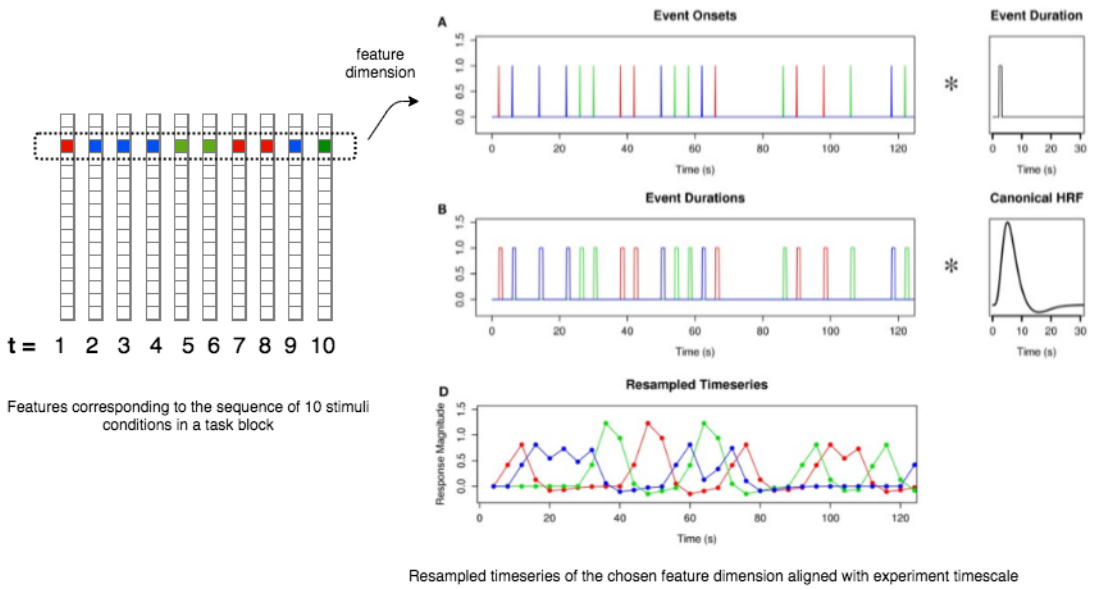


Fig. 0.4. Convolution of the features with the canonical hemodynamic response function (HRF) for upsampling them to ensure that the features match the timesteps of the BOLD data. The features corresponding to the sequence of images are extracted from the task network and each dimension are convolved with the HRF double gamma function aligned with the event onset timings of the stimuli. The final resampled timeseries after this operation corresponds one-on-one to the timecourse of the neuroimaging data.

The number of samples in the feature time series is lesser than the number of samples (TRs) in the task blocks of the BOLD series to be predicted. Therefore, in order to upsample the timescale of the extracted features with the timescale of the experiment, we convolve each component of the features with the canonical hemodynamic response function (HRF) which is a double gamma function expressed as a difference of two gamma functions [**41**]. Figure 0.4 depicts this operation. The series of features is now upsampled and the samples corresponding to the TR of the BOLD series is aligned and matched. Thus, the HRF convolution helps in upsampling the features and also in modeling the hemodynamics phenomena.

2.5.4. Ridge Regression Model : Mapping features to brain activity

The overall idea of an encoding model is to learn a parametric function f denoted by $f : S \rightarrow R$ which maps stimuli in the set S to the brain responses in the set R.

The deep neural network features obtained after performing the steps till Sec. 2.5.3 can be considered to be in a vector space referred to as the *feature space* as these are the features which are transformed versions of the actual inputs represented by the stimuli image sequences that can be located in the *input space*. This transformation involving the operations in the neural network, dimensionality reduction and resampling can be thought of a $\phi(\mathbf{x})$ that is applied on the each of the inputs $\mathbf{x}$ from the input space. The target for

this regression is the space of the functional brain activity given by the preprocessed fMRI BOLD signals and it is referred to as the *activity space* in this case [**83**].

The feature space learned from the neural networks needs to be mapped on to the activity space of the functional activity (fMRI BOLD) in order to build the encoding model. Technically, it is desirable to use linear regression methods to achieve this. This choice is motivated by a variety of strong reasons [**83**]. Firstly, neuroscientists view the brain as a system that takes the stimuli as input and uses high-level non-linear computations to form representations of the brain activity. Consequently, we would require all the complex non-linear transformations of the stimuli to be done by the powerful artificial neural networks if they are also like the brain. The transformation from the feature space to activity space could be any simple model in this case. In addition, the usage of this type of linearizing feature space (non-linear mapping from the input space to feature space and linear mapping from the feature space to activity space) is very common in computational neuroscience [**83, 123**] and it can be considered that the non-linearly transformed stimuli form the basis of the subspace that embeds the activities [**27**]. Most of the work done in fMRI encoding studies use this linear hypothesis [**83, 74, 123, 57**] and many theories have been developed around it. We also remark that the number of data samples available in most neuroimaging studies for the purpose of training the encoding model is limited. As most non-linear models (including deep neural networks) generally require a higher number of samples for training [**107**], they tend to underfit the data if the number of samples are low with respect to the capacity of the model. Also, an analysis of linear relationship between the features and the activity in brain regions is likely to inform how similar are the task network derived features and the brain activity as we have a good statistical framework about linear dependencies [**123**]. Only if there is a linear transformation from the feature space to the activity space, it makes sense to think that these two spaces are aligned as it is the simplest possible transformation.

Among the linear regression methods, regularized models of regression are preferred. Since the number of frames (TRs) in data recorded from humans is limited, the number of features are very high compared to the number of training samples in most cases. Thus, a sparse regression method such as ridge regression has been used in many similar studies such as [**54, 118, 60**]. This method uses the L2-norm to avoid the poor fits of the data and ensures that a good number the weights are zero (sparse). This is very useful in avoiding the ill-conditioning of the regression and for better interpretability.

Figure 0.3 illustrates the schematic setup of the encoding model. It is necessary that the features considered in the regression belong to the same feature space and thus they need to be derived from the same layer of the same neural network.

The encoding model is fitted individually per voxel (mass univariate model). The feature matrix $\mathbf{H}$ corresponding to the n number of frames (TRs) of p-dimensional stimulus vectors that are available is constructed by concatenating the reduced and resampled feature vectors

giving a dimension of $n \times p$. The voxel values per frame (TR) are taken as a vector $\mathbf{z^{(i)}}$. Therefore, for each voxel i, we fit models to compute the parameter matrix $\hat{\beta}^{(i)}_{ridge}$ such that :

$$\hat{\beta}^{(i)}_{ridge} = \underset{\beta \in \mathbb{R}^p}{\arg\min} \|\mathbf{z^{(i)}} - \mathbf{H}\beta\|^2{}_2 + \lambda \|\beta\|^2{}_2$$

The penalty λ is chosen using a 6-fold cross-validation using the best R^2 score. The coefficient of determination (R^2) between the actual and predicted BOLD responses gives the measure of how predictive the particular voxel is based on the regressed feature. This value for each voxel is plotted on a brain volume in the respective location. For a predicted signal $\hat{z}^{(i)}_{test}$ for the ground truth signal $z^{(i)}_{test}$ it can be expressed as:

$$R^{2^{(i)}} = 1 - \frac{\|z^{(i)}_{test} - \hat{z}^{(i)}_{test}\|^2}{\|z^{(i)}_{test} - mean(\hat{z}^{(i)}_{test})\|^2}$$

As a statistical check, permutation tests were carried out to quantify the significance of the encoding model predictions. This was done by shuffling the target responses 5000 times and obtaining the False Discovery Rate (FDR) corrected p-values for the encoding model results using the Benjamini-Hochberg procedure, similar to [**118**]. The voxels with the FDR corrected $p > 0.05$ in the fitted models were rejected and not considered as they are not significant.

Here, it is worth highlighting that under this mass univariate approach, the set of parameters for each voxel $\hat{\beta}^{(i)}_{ridge}$ are different for each voxel and the regularization penalty λ is chosen individually for each voxel. The regression uses the same inputs (features) for all voxels, and thus, it is these parameters that determine the mapping. The $\hat{\beta}^{(i)}_{ridge}$ weights taken together can be viewed as the linear transformation that transforms these features $\mathbf{h_t(l)}$ in the feature space to the activity space of the voxels $\mathbf{z^{(i)}}$.

2.5.5. Hemodynamic Delay

fMRI BOLD is a slow signal due to the delay in the hemodynamic response in the brain. So the timing of the response signal usually does not correspond to the stimulus onset timing as it could as well be a response to a previous stimulus condition. The HRF convolution described in Section 2.5.3 already models this hemodynamic delay with a fixed peak time. So, the encoding model is likely to be trained and evaluated on mismatched signal peaks. As we are interested in the quality of prediction, we handle this by fitting models on forward time-shifted BOLD signals and cross-validating to identify the optimal shift. The search for this shift is effectively to try and identify the peak of the signal.

The ridge regression model described in 2.5.4 was fitted on voxel time series shifted by number of frames (TRs) in [1,2,3,4] and 6-fold cross-validation was used to select the best delay hyperparameter for the data. This value was determined to be 2 in most cases in our analysis.

3. Results

In this section, we present the results of our experiments that we described in Sec. 2. First, Sec. 3.1 details the performance of the trained task networks. Next, Sec. 3.2 presents the analysis of the voxel-based encoding models using the R^2 brain maps obtained (as described in Sec. 2.5.4). Finally, Sec. 3.3 presents the R^2 brain maps by encoding the parcel signals and the analysis of performance of representations from different layers in the task networks.

3.1. Task network performance

We present the performances of the task networks that were trained as described in Sec. 3.1 on the N-back tasks (2-back and 0-back that were trained independently). In this section the results correspond only to the metrics obtained on validating the trained task networks and the fMRI data was not considered at this stage.

The results with the held-out validation accuracy and binary cross entropy error (BCE) for the task networks that were trained for the respective N-back task are given in Table 0.1 and 0.2. All trained networks (except the Random LSTM which is untrained), give a good validation performance of more than 95% with early stopping.

	2-back accuracy (validation)	2-back BCE (validation)
Random LSTM	49.32 %	0.869
LSTM	96.86 %	0.009
LSTM-SAB	98.29 %	0.006
ConvLSTM	99.16 %	0.002

Table 0.1. 2-back : Task network performance (classification accuracy and binary cross-entropy error) on held-out validation set of image sequences for the different models considered.

	0-back accuracy (validation)	0-back BCE (validation)
Random LSTM	52.65 %	0.794
LSTM	95.33 %	0.010
LSTM-SAB	98.90 %	0.008
ConvLSTM	97.36 %	0.012

Table 0.2. 0-back : Task network performance (classification accuracy % and binary cross-entropy error) on held-out validation set of image sequences for the different models considered.

3.2. Voxel-wise Encoding Analysis

The BOLD signals of the WM task (2-back), preprocessed as described in Sec. 2.3.3 represent time series for each voxel in the brain volume. These were used in the encoding models (Sec. 2.5.4) based on the features extracted from the networks discussed in Section 2.4 by using the respective stimuli.

Figure 0.5 shows the volumetric R^2 maps for the encoding models fitted for each voxel as described in Section 2.5.4. Each voxel represents the R^2 score which is the proportion of variance in the BOLD data in that voxel that can be explained by the task network layer-derived features used in the encoding. This value is represented by the color of the respective voxel in the statistical brain map.

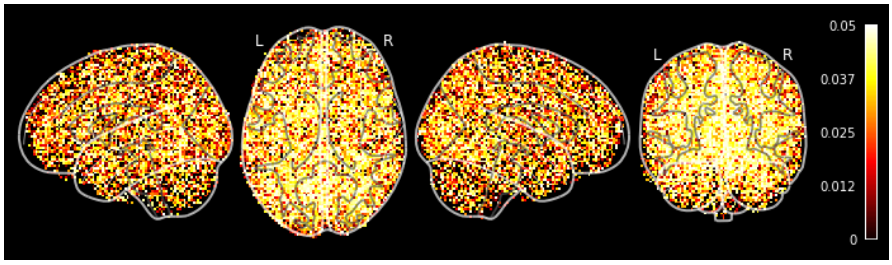
In Figure 0.5, we can note that, compared to the untrained Random LSTM layer activations, the other trained networks have given better performance as seen in the R^2 maps for the 2-back task. As we have a concrete difference in the predictability of the brain activity from the untrained and trained network activations, we can be more confident about the impact of the training of the task-network.

It can be clearly seen in Figure 0.5(b) that for the last layer (layer 7) of the CNN model, the concentration of the encoded voxels with a high variance explained are predominantly in the occipito-temporal regions. This is in conformance with prior studies such as [**59**].

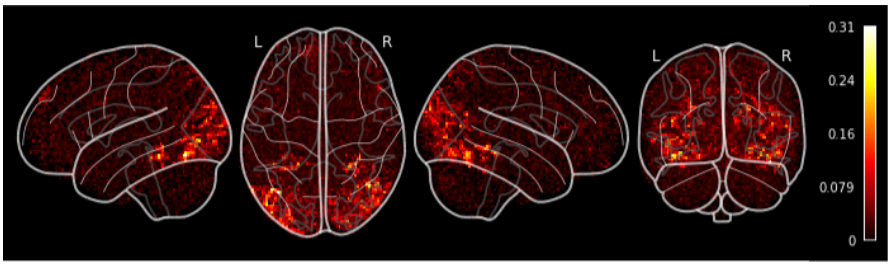
For the other R^2 maps of the recurrent models, there are more significant voxels in not just the occipital and temporal regions but also distributed across the fronto-parietal region.

As seen in Figure 0.5 (b) and (c), for the CNN derived brain map and the LSTM derived brain maps of encoding model performance (R^2 score), it can be noted that the CNN model has received less score compared to the LSTM in certain regions in the temporal, medial and fronto-parietal regions.

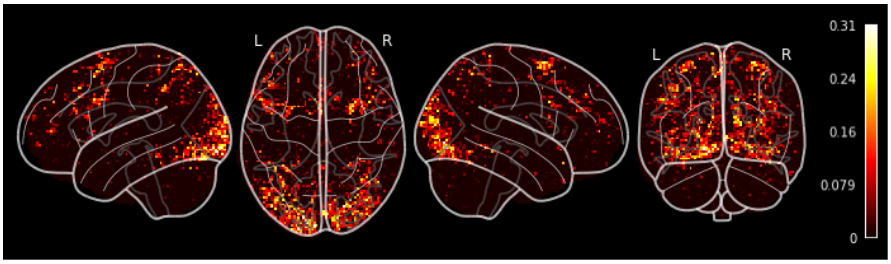
It is also interesting to observe that there are a few voxels in the occipital and temporal regions with a good score for both models (CNN and LSTM). These voxels are of interest to potentially interpret the effect of the recurrent computations on the CNN features in explaining the variance of the BOLD activity. We need to note that recurrent network-derived features are merely non-linear transformations of the same CNN features used.



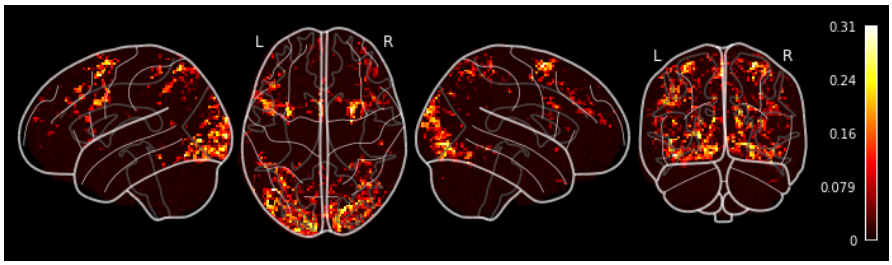
(a) Random LSTM



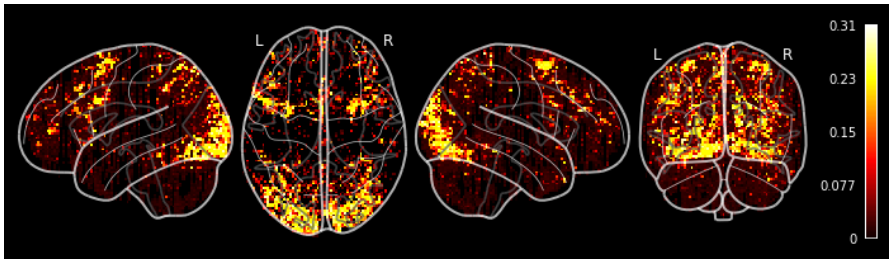
(b) CNN (layer 7)



(c) LSTM (layer 2)



(d) LSTM-SAB (layer 2)



(e) Conv LSTM (layer 2)

Fig. 0.5. sub-03 - 2-back Voxel-wise performance: The brain maps of the encoding performance for each voxel in terms of R^2 value between predicted and actual BOLD responses during the 2-back task for the features derived from the considered task-network models. Random features in (a) yield a random map. CNN features in (b) give high performance in areas involved in visual processing while the recurrent network features in (c), (d) and (e) extend into areas involved in visual working memory.

3.3. Parcel-wise Encoding Analysis

As described in Sec. 2.3.3, the BOLD signals of the working memory (WM) task blocks were parcellated using the MIST ROI atlas [**113**]. They were then used in parcel-wise encoding models (Sec. 2.5.4) based on features derived from the networks discussed in Section 2.4.

The results of fitting the parcel-wise encoding model on 2-back features are shown in Figure 0.6. The colours of the individual parcels correspond to different R^2 values which can be interpreted as the proportion of variance in these parcels explained by the considered features. We observe that in Figure 0.6(b), the CNN (layer 7) features show some level of variance in the parcels from the occipital and temporal areas with the highest R^2 in parcels in the occipital region. Figure 0.6(c), (d) and (e) show a high R^2 value in many areas of the brain. This indicates that the features derived from the LSTM variants seem to explain much more variance in occipital, temporal, frontal and parietal regions.

Based on the R^2 score of the individual parcels of the brain maps generated by the recurrent neural network features, 6 regions of interest (ROIs) have been chosen and listed in Table 3.3. These are regions that have been found to be active during working memory tasks as observed by many fMRI studies [**116, 89, 97, 109**].

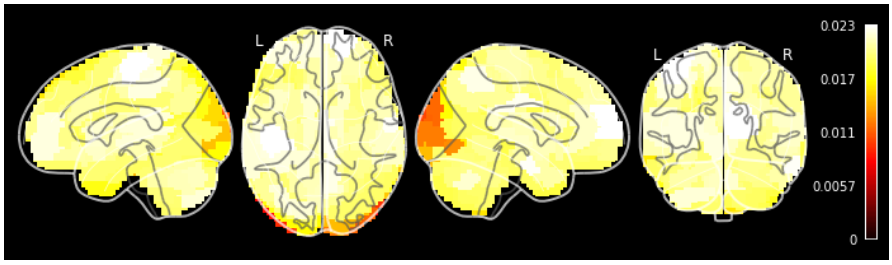
ROI label	Name
R_IPsul	right_INTRAPARIETAL_SULCUS
L_FEF	left_FRONTAL_EYE_FIELD
L_VLPFcor	left_VENTROLATERAL_PREFRONTAL_CORTEX
L_ITgyr	left_INFERIOR_TEMPORAL_GYRUS
L_OCCTgyr_l	left_OCCIPITOTEMPORAL_GYRUS_lateral
R_FUSgyr_vl	right_FUSIFORM_GYRUS_ventrolateral

Table 0.3. The ROIs from the MIST ROI atlas [**113**] selected for analysis based on the top encoding model performances in these parcels in the atlas. The label corresponding to these parcels are listed in the left column with their full names in the right column.

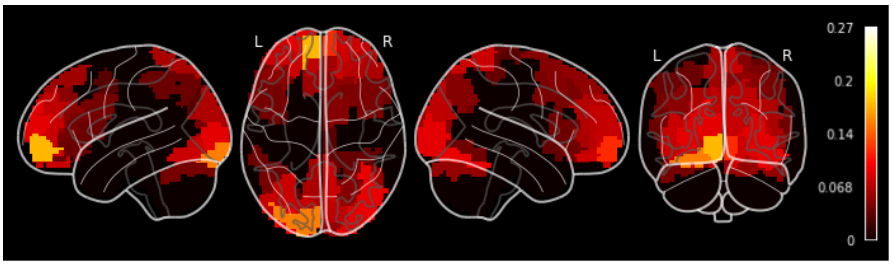
Figure 0.7 shows the encoding model performances R^2 at each of these ROIs plotted for different layers of the memory module in Figure 0.2 based on cross-validated performance. The error bars indicate the standard deviation of the averaged model performance scores during the cross-validation. The layer preference or selectivity is in favour of the layer whose feature space gives a better performance for the encoding model.

There seem to be not much difference between the LSTM and LSTM-SAB model for ROIs such as R_IPsul and L_VLPF. In many cases, the difference between the performance of two consecutive layers are very close to each other. This is expected as they are correlated with each other.

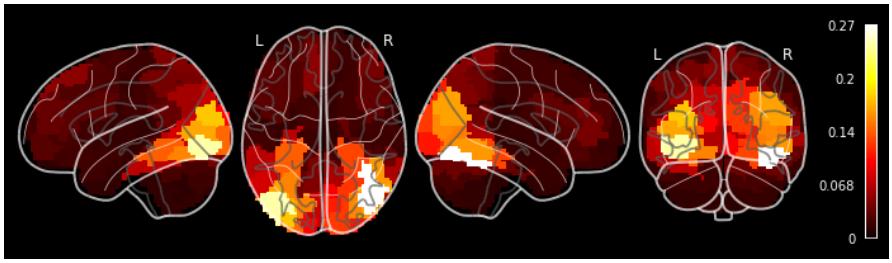
The layer preferences of the ConvLSTM seem to be different from the LSTMs in many cases. It can be possibly attributed to the difference in dimensions of their feature spaces.



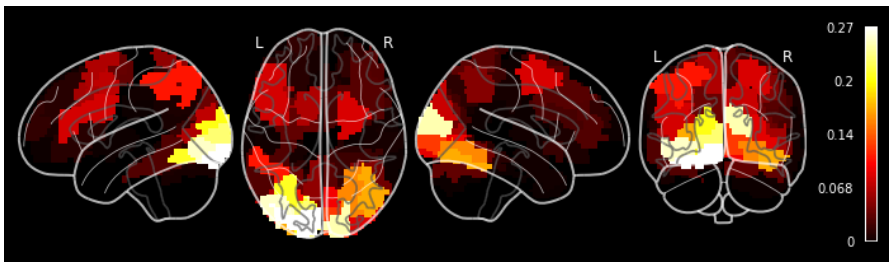
(a) Random LSTM



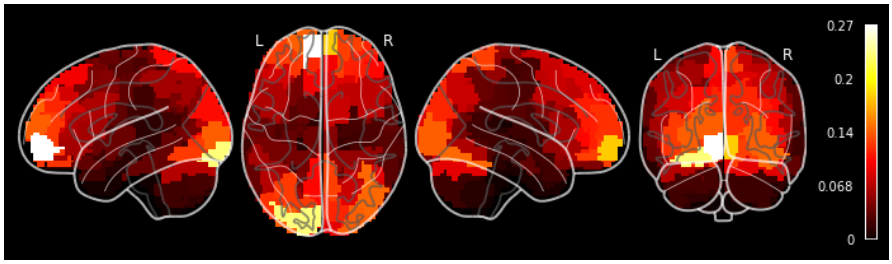
(b) CNN (layer 7)



(c) LSTM (layer 2)



(d) LSTM-SAB (layer 2)



(e) Conv LSTM (layer 2)

Fig. 0.6. sub-03 - 2-back Parcel-wise performance: The brain maps of the encoding performance for each MIST ROI parcel in terms of R^2 value between predicted and actual BOLD responses during the 2-back task for the features derived from the considered task-network models. Random features in (a) yield a random map. CNN features in (b) are able to give good performance only in parcels involved in visual processing, while the recurrent network features in (c), (d) and (e) include parcels involved in visual working memory.

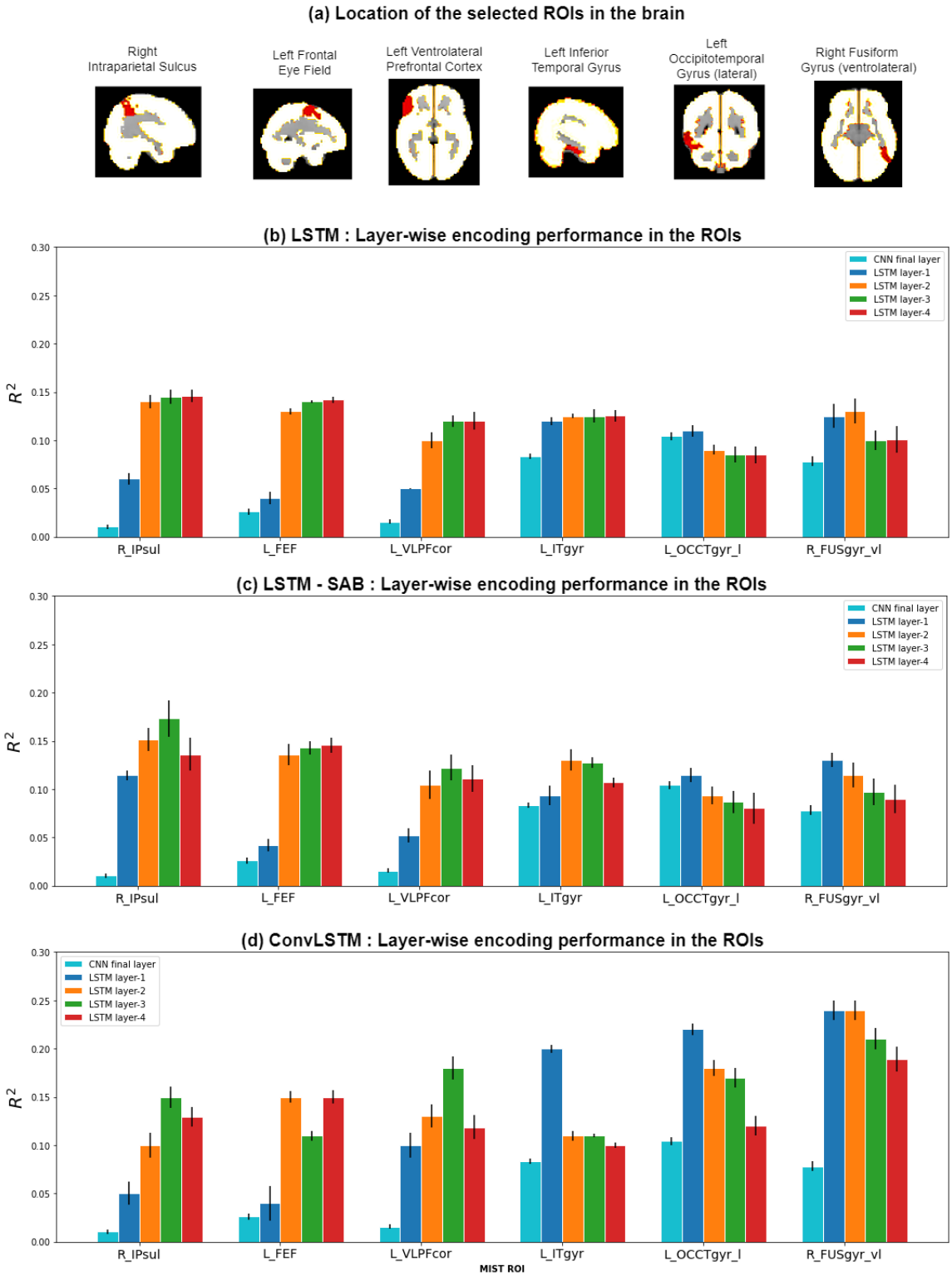


Fig. 0.7. sub-03 - 2-back layer-wise encoding performance across ROIs : Figure (a) highlights (in red) the location of the selected ROIs in the brain volume (Table 3.3), shown as a reference for the plots below. Figures (b),(c) and (d) depict the plots of R^2 value between predicted and actual BOLD responses in the 2-back task for the LSTM, LSTM-SAB and ConvLSTM features respectively. Each color corresponds to the features derived from the specific layers (final CNN layer and the 4 LSTM layers), plotted as a separate bar for each region of interest. The plots compare the average test performance of features from different layers in encoding fMRI BOLD activity.

4. Discussion

The computational mechanisms of activations in the visual working memory of the brain have been less understood. We trained different task networks with variants of LSTMs (from Section 2.4) in the memory module to peform the N-back task, namely - LSTM, LSTM-SAB and ConvLSTM. For each stimuli image in the 2-back sequence, features were extracted from these networks as well as those from a Random (untrained) LSTM and the final layer of the CNN (Recogniton Module). Each of these features were reduced, resampled and regressed on to the fMRI BOLD signals corresponding to the stimuli sequence to obtain the performance at voxel and parcel levels in terms of the variance explained R^2 as described in Section 2.5.

Firstly, we observe that the representations derived from trained networks are able to explain the variance better than that of untrained random networks.

We remark that the layer activations of recurrent models have performed better compared to the CNN across the brain. Especially, the fronto-parietal areas have shown significant improvements when compared against the CNN. It is in the these frontal areas that we find the regions like the Frontal Eye Field (FEF), Prefrontal Cortex (PFC), etc. with structures involved in executive control. Also, some voxels in the inferior temporal gyrus (ITG) have a significantly higher explained variance for the recurrent models than the CNN model. Various theories point to the prevalence of the visual working memory acting through the inferior temporal regions from a frontal executive [**116**].

Previous studies like [**47, 79, 62, 118**] that have modeled CNN-derived visual features to encode the brain responses found relationship of the initial layers of the CNN to the activity in early visual areas and the later layers to the higher visual areas. One important difference between our study and these previous works is the task that the network is optimized for: while most of these studies largely use the object recognition task, our networks have been optimized for the memory task. The LSTM-derived features that we employed were able to explain more variance than the higher layer feature of the CNN baseline in the inferior temporal and fusiform regions. While the predictions of the CNN are mostly in the early visual areas, the LSTM-derived features are able to explain the variance better in higher visual areas (ITG) and working memory-related areas (PFC, FEF). The CNN-derived features have not captured the variance in these higher areas sufficiently.

Among the LSTM and the LSTM-SAB models, the performance of the SAB models is slightly better for the considered runs. One plausible reason is the focus of the credit on the N-back target using the learned sparse attention increases the information captured in the feature. Thus, this additional capacity of the LSTM-SAB model could help learn better features.

The ConvLSTM features have marginally outperformed the LSTM-SAB possibly due to the strength of the convolution filters as some spatial attributes are likely to be preserved

through all the feature transformations happening in the network. The performance of these features as seen as the variance explained by them (R^2) on the visual areas in the temporal region are better than the other LSTM features.

Deeper layers are preferred by some parcels in frontal and parietal regions while earlier recurrent layers preferred inferior-temporal regions. This phenomenon is interesting because even in the brain the lower layers of the network are mostly involved in perception and the deeper layers are involved in higher cognitive functions. The temporal dynamics of the multi-layer LSTM is the reason behind this. The lower layers act more like a buffer of the visual features whereas the higher layer activations are more correlated with the (N-back) decision made. However, there are many parcels which have an equal preference for multiple layers. As the layer activations are correlated with one another as they are successively derived, this is not surprising.

We see that predicting 0-back BOLD responses from features derived from a task network optimized for the same task yields good results, but the impact of the recurrent models is not that significant. The BOLD activations of the frontal regions are lesser in 0-back compared to 2-back as there is only maintenance of the object in memory and there is no updating of the memory as in the case of 2-back.

We emphasize that the similarity of these representations and the strengths of certain type of representations of recurrent neural networks are due to the correlations gleaned from the stimuli by the network as well the correlated responses evoked in the brain. It needs to be further probed if there are computational similarities at different levels in addition to these representational similarities.

5. Conclusion

We were able to train various task networks and optimize them for solving simple memory tasks. Overall, we outlined and demonstrated a procedure for explaining the functional activity of working memory of the human brain from recurrent neural network representations that gave a fairly good performance compared to convolutional neural networks. Our approach offers significant advantages to methods to Computer Science as well as Cognitive Neuroscience by both demonstrating a way of computationally modeling brain activity during a memory task and highlighting the similarity among different memory representations (computational and brain activity).

Firstly, **we have shown a useful application of the features extracted from recurrent neural networks.** The task fMRI signals, as well as the representations of recurrent neural networks are not interpretable. However, our findings seem to indicate a plausible alignment between these two representations. This phenomenon can be potentially exploited using appropriate statistical methods to analyze how relevant information is encoded and

transformed in each of these domains (fMRI and recurrent networks) [**39**]. We observed that the variability of the functional activity in brain areas involved in working memory is related to the task-optimized recurrent neural network features. From a computational perspective, this has an advantage in modeling a complex multivariate signals such as the fMRI BOLD as it is better than the traditional methods [**38, 116, 89, 109**] as it considers as much information possible from the stimuli. Our brain evolved to process memories through some computations on the observed stimuli, but we have yet to fully understand these. The observation of its similarity with the task network features suggests a possible universality in the computational principles involved in consolidating short term memories.

Secondly, **we have demonstrated that feature maps derived from recurrent neural networks trained on the same working memory tasks as humans, display localized activity corresponding the human brain regions known to be involved in visual working memory**. Specifically, the key areas involved in working memory function : the occipito-temporal (visual processing), inferior-temporal (visual understanding, short term memory) and fronto-parietal (executive control) areas of the brain were highly correlated with the features. In addition, from our parcel-wise analysis, we remark that the representations from different layers of the task network uncover a hierarchy of parcels - late convolutional and early recurrent layers are associated with visual areas; deeper recurrent layers are related to more frontal and fronto-parietal areas. Thus, our models inform us about the functional organization and information processing in working memory.

Thirdly, **our experiments and analysis underscore the importance of the recurrent computations in modeling brain activity of a memory-related task**. This was observed by comparing the statistical brain maps of the control conditions that did not involve recurrence where the predicted activity was limited to the visual stream. Also, it can be argued that networks with richer and more useful representational power such as LSTMs and its variants are able to explain the brain activity better for a distributed system such as the working memory. The features obtained from these networks can be considered to represent correlations of inputs over multiple time scales that are computed using combinations of feed-forward and temporal (lateral) connections. The relevance of these features with the working memory of the brain can be used to test other hypotheses about this memory. Further, our study seems to indicate that the architectural elements of recurrent neural networks such as: gating, sparse attention, dedicated memory, spatial convolution, etc. could possibly drive the RNN representations to be more aligned with the representations of functional activity in the brain due to better capturing of temporal correlations by such models. Thus, such networks can be utilized for designing good features for modeling as we have shown the integration of specific computational models with the brain activity.

Finally, **our approach attempts to offer a proof-of-concept that deep neural networks can be used to model not just sensory processing but it also a promising**

way to model integrative cognition such as memory. This approach of fMRI modeling can potentially be applied to other cognitive tasks, more specifically memory tasks or other tasks which indirectly require a memory component. In the future, this coarse-grained relationship between these representations needs to be further extended for more complex recurrent architectures and training, for a solid theory to emerge.

Acknowledgements

This work was done as a part of the Courtois NeuroMod project that was made possible by a donation from the Courtois foundation. These funds are administered by the Fondation Institut Gériatrie Montréal (FIGM) at Centre de Recherche de l'institut Universitaire de Gériatrie de Montréal (CRIUGM), part of CIUSSS du Centre-Sud-de-l'île-de-Montréal, as well as Université de Montréal.

References

[1] Alexandre Abraham, Fabian Pedregosa, Michael Eickenberg, Philippe Gervais, Andreas Mueller, Jean Kossaifi, Alexandre Gramfort, Bertrand Thirion, and Gael Varoquaux. Machine learning for neuroimaging with scikit-learn. *Frontiers in Neuroinformatics*, 8, 2014.

[2] Andrew James Anderson, Benjamin D Zinszer, and Rajeev DS Raizada. Representational similarity encoding for fmri: Pattern-based synthesis to predict brain activity using stimulus-model-similarities. *NeuroImage*, 128:44–53, 2016.

[3] Itamar Arel, Derek C Rose, and Thomas P Karnowski. Deep machine learning-a new frontier in artificial intelligence research [research frontier]. *IEEE computational intelligence magazine*, 5(4):13–18, 2010.

[4] Owen J Arthurs and Simon Boniface. How well do we understand the neural origins of the fmri bold signal? *TRENDS in Neurosciences*, 25(1):27–31, 2002.

[5] B.B. Avants, C.L. Epstein, M. Grossman, and J.C. Gee. Symmetric diffeomorphic image registration with cross-correlation: Evaluating automated labeling of elderly and neurodegenerative brain. *Medical Image Analysis*, 12(1):26–41, 2008.

[6] Alan Baddeley. The episodic buffer: a new component of working memory? *Trends in cognitive sciences*, 4(11):417–423, 2000.

[7] Alan Baddeley. Working memory and language: An overview. *Journal of communication disorders*, 36(3):189–208, 2003.

[8] Alan Baddeley. *Working memory, thought, and action*, volume 45. OuP Oxford, 2007.

[9] Alan D Baddeley and Graham Hitch. Working memory. In *Psychology of learning and motivation*, volume 8, pages 47–89. Elsevier, 1974.

[10] Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

[11] David Bau, Bolei Zhou, Aditya Khosla, Aude Oliva, and Antonio Torralba. Network dissection: Quantifying interpretability of deep visual representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 6541–6549, 2017.

[12] Yashar Behzadi, Khaled Restom, Joy Liau, and Thomas T. Liu. A component based noise correction method (CompCor) for BOLD and perfusion based fmri. *NeuroImage*, 37(1):90–101, 2007.

[13] Julie Boyle, Basile Pinsard, Amal Boukhdhir, Sylvie Belleville, Simona Brambatti, Jen-I Chen, Julien Cohen-Adad, André Cyr, Adrian Fuente, Pierre Rainville, and Pierre Bellec. The courtois project on neuronal modelling. In *26th Annual Meeting of the Organization for Human Brain Mapping*, Neuroinformatics and Data Sharing, June 2020.

[14] Todd S Braver, Jonathan D Cohen, Leigh E Nystrom, John Jonides, Edward E Smith, and Douglas C Noll. A parametric study of prefrontal cortex involvement in human working memory. *Neuroimage*, 5(1):49–62, 1997.

[15] Wen Jia Chai, Aini Ismafairus Abd Hamid, and Jafri Malin Abdullah. Working memory from the psychological and neurosciences perspectives: a review. *Frontiers in psychology*, 9:401, 2018.

[16] Kyunghyun Cho, Bart Van Merriënboer, Caglar Gulcehre, Dzmitry Bahdanau, Fethi Bougares, Holger Schwenk, and Yoshua Bengio. Learning phrase representations using rnn encoder-decoder for statistical machine translation. *arXiv preprint arXiv:1406.1078*, 2014.

[17] Radoslaw M Cichy and Daniel Kaiser. Deep neural networks as scientific models. *Trends in cognitive sciences*, 23(4):305–317, 2019.

[18] Radoslaw Martin Cichy, Aditya Khosla, Dimitrios Pantazis, Antonio Torralba, and Aude Oliva. Comparison of deep neural networks to spatio-temporal cortical dynamics of human visual object recognition reveals hierarchical correspondence. *Scientific reports*, 6(1):1–13, 2016.

[19] Jonathan D Cohen, William M Perlstein, Todd S Braver, Leigh E Nystrom, Douglas C Noll, John Jonides, and Edward E Smith. Temporal dynamics of brain activation during a working memory task. *Nature*, 386(6625):604–608, 1997.

[20] Susan M Courtney, Leslie G Ungerleider, Katrina Keil, and James V Haxby. Object and spatial visual working memory activate separate neural systems in human cortex. *Cerebral cortex*, 6(1):39–49, 1996.

[21] Nelson Cowan, J Scott Saults, and Candice C Morey. Development of working memory for verbal–spatial associations. *Journal of memory and language*, 55(2):274–289, 2006.

[22] Robert W. Cox and James S. Hyde. Software tools for analysis and visualization of fmri data. *NMR in Biomedicine*, 10(4-5):171–178, 1997.

[23] Wendy A de Heer, Alexander G Huth, Thomas L Griffiths, Jack L Gallant, and Frédéric E Theunissen. The hierarchical cortical organization of human speech processing. *Journal of Neuroscience*, 37(27):6539–6557, 2017.

[24] Thomas Deffieux, Charlie Demene, Mathieu Pernot, and Mickael Tanter. Functional ultrasound neuroimaging: a review of the preclinical and clinical state of the art. *Current opinion in neurobiology*, 50:128–135, 2018.

[25] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. Fei-Fei. ImageNet: A Large-Scale Hierarchical Image Database. In *CVPR09*, 2009.

[26] René Descartes. *Discours de la méthode pour bien conduire sa raison et chercher la vérité dans les sciences*, volume 1. Hachette et cie, 1878.

[27] Jörn Diedrichsen and Nikolaus Kriegeskorte. Representational models: A common framework for understanding encoding, pattern-component, and representational-similarity analysis. *PLoS computational biology*, 13(4):e1005508, 2017.

[28] Mark D'Esposito, Bradley R Postle, and Bart Rypma. Prefrontal cortical contributions to working memory: evidence from event-related fmri studies. *Executive control and the frontal lobe: Current issues*, pages 3–11, 2000.

[29] Shimon Edelman. Representation is representation of similarities. *Behavioral and brain sciences*, 21(4):449–467, 1998.

[30] Dumitru Erhan, Yoshua Bengio, Aaron Courville, and Pascal Vincent. Visualizing higher-layer features of a deep network. *University of Montreal*, 1341(3):1, 2009.

[31] Oscar Esteban, Ross Blair, Christopher J. Markiewicz, Shoshana L. Berleant, Craig Moodie, Feilong Ma, Ayse Ilkay Isik, Asier Erramuzpe, Mathias Kent, James D. andGoncalves, Elizabeth DuPre, Kevin R. Sitek, Daniel E. P. Gomez, Daniel J. Lurie, Zhifang Ye, Russell A. Poldrack, and Krzysztof J. Gorgolewski. fmriprep. *Software*, 2018.

[32] Oscar Esteban, Christopher Markiewicz, Ross W Blair, Craig Moodie, Ayse Ilkay Isik, Asier Erramuzpe Aliaga, James Kent, Mathias Goncalves, Elizabeth DuPre, Madeleine Snyder, Hiroyuki Oya, Satrajit Ghosh, Jessey Wright, Joke Durnez, Russell Poldrack, and Krzysztof Jacek Gorgolewski. fMRIPrep: a robust preprocessing pipeline for functional MRI. *Nature Methods*, 2018.

[33] Matthew Farrell, Stefano Recanatesi, Timothy Moore, Guillaume Lajoie, and Eric Shea-Brown. Recurrent neural networks learn robust representations by dynamically balancing compression and expansion. *bioRxiv*, page 564476, 2019.

[34] Callie Federer, Haoyan Xu, Alona Fyshe, and Joel Zylberberg. Improved object recognition using neural networks trained to mimic the brain's statistical properties. *Neural Networks*, 131:103–114, 2020.

[35] Julie A Fiez, Elizabeth A Raife, David A Balota, Jacob P Schwarz, Marcus E Raichle, and Steven E Petersen. A positron emission tomography study of the short-term maintenance of verbal information. *Journal of Neuroscience*, 16(2):808–822, 1996.

[36] Paul C Fletcher and R Nx˙ A Henson. Frontal lobes and human memory: insights from functional neuroimaging. *Brain*, 124(5):849–881, 2001.

[37] VS Fonov, AC Evans, RC McKinstry, CR Almli, and DL Collins. Unbiased nonlinear average age-appropriate brain templates from birth to adulthood. *NeuroImage*, 47, Supplement 1:S102, 2009.

[38] Karl J Friston, Andrew P Holmes, JB Poline, PJ Grasby, SCR Williams, Richard SJ Frackowiak, and Robert Turner. Analysis of fmri time-series revisited. *Neuroimage*, 2(1):45–53, 1995.

[39] Alona Fyshe, Partha P. Talukdar, Brian Murphy, and Tom M. Mitchell. Interpretable semantic vectors from a joint model of brain- and text-based meaning. *Proceedings of the conference. Association for Computational Linguistics. Meeting*, 2014:489–499, Jun 2014. PMC4497373[pmcid].

[40] Xavier Glorot and Yoshua Bengio. Understanding the difficulty of training deep feedforward neural networks. In *Proceedings of the thirteenth international conference on artificial intelligence and statistics*, pages 249–256. JMLR Workshop and Conference Proceedings, 2010.

[41] GH Glover. Deconvolution of impulse response in event-related bold fmri. *NeuroImage*, 9(4):416—429, April 1999.

[42] Richard M Golden. A unified framework for connectionist systems. *Biological Cybernetics*, 59(2):109–120, 1988.

[43] Christoph Goller and Andreas Kuchler. Learning task-dependent distributed representations by backpropagation through structure. In *Proceedings of International Conference on Neural Networks (ICNN'96)*, volume 1, pages 347–352. IEEE, 1996.

[44] K. Gorgolewski, C. D. Burns, C. Madison, D. Clark, Y. O. Halchenko, M. L. Waskom, and S. Ghosh. Nipype: a flexible, lightweight and extensible neuroimaging data processing framework in python. *Frontiers in Neuroinformatics*, 5:13, 2011.

[45] Krzysztof J. Gorgolewski, Oscar Esteban, Christopher J. Markiewicz, Erik Ziegler, David Gage Ellis, Michael Philipp Notter, Dorota Jarecka, Hans Johnson, Christopher Burns, Alexandre Manhães-Savio, Carlo Hamalainen, Benjamin Yvernault, Taylor Salo, Kesshi Jordan, Mathias Goncalves, Michael Waskom, Daniel Clark, Jason Wong, Fred Loney, Marc Modat, Blake E Dewey, Cindee Madison, Matteo Visconti di Oleggio Castello, Michael G. Clark, Michael Dayan, Dav Clark, Anisha Keshavan, Basile Pinsard, Alexandre Gramfort, Shoshana Berleant, Dylan M. Nielson, Salma Bougacha, Gael Varoquaux, Ben Cipollini, Ross Markello, Ariel Rokem, Brendan Moloney, Yaroslav O. Halchenko, Demian Wassermann, Michael Hanke, Christian Horea, Jakub Kaczmarzyk, Gilles de Hollander, Elizabeth DuPre, Ashley Gillman, David Mordom, Colin Buchanan, Rosalia Tungaraza, Wolfgang M. Pauli,

Shariq Iqbal, Sharad Sikka, Matteo Mancini, Yannick Schwartz, Ian B. Malone, Mathieu Dubois, Caroline Frohlich, David Welch, Jessica Forbes, James Kent, Aimi Watanabe, Chad Cumba, Julia M. Huntenburg, Erik Kastman, B. Nolan Nichols, Arman Eshaghi, Daniel Ginsburg, Alexander Schaefer, Benjamin Acland, Steven Giavasis, Jens Kleesiek, Drew Erickson, René Küttner, Christian Haselgrove, Carlos Correa, Ali Ghayoor, Franz Liem, Jarrod Millman, Daniel Haehn, Jeff Lai, Dale Zhou, Ross Blair, Tristan Glatard, Mandy Renfro, Siqi Liu, Ari E. Kahn, Fernando Pérez-García, William Triplett, Leonie Lampe, Jörg Stadler, Xiang-Zhen Kong, Michael Hallquist, Andrey Chetverikov, John Salvatore, Anne Park, Russell Poldrack, R. Cameron Craddock, Souheil Inati, Oliver Hinds, Gavin Cooper, L. Nathan Perkins, Ana Marina, Aaron Mattfeld, Maxime Noel, Lukas Snoek, K Matsubara, Brian Cheung, Simon Rothmei, Sebastian Urchs, Joke Durnez, Fred Mertz, Daniel Geisler, Andrew Floren, Stephan Gerhard, Paul Sharp, Miguel Molina-Romero, Alejandro Weinstein, William Broderick, Victor Saase, Sami Kristian Andberg, Robbert Harms, Kai Schlamp, Jaime Arias, Dimitri Papadopoulos Orfanos, Claire Tarbert, Arielle Tambini, Alejandro De La Vega, Thomas Nickson, Matthew Brett, Marcel Falkiewicz, Kornelius Podranski, Janosch Linkersdörfer, Guillaume Flandin, Eduard Ort, Dmitry Shachnev, Daniel McNamee, Andrew Davison, Jan Varada, Isaac Schwabacher, John Pellman, Martin Perez-Guevara, Ranjeet Khanuja, Nicolas Pannetier, Conor McDermottroe, and Satrajit Ghosh. Nipype. *Software*, 2018.

[46] Douglas N Greve and Bruce Fischl. Accurate and robust brain image alignment using boundary-based registration. *NeuroImage*, 48(1):63–72, 2009.

[47] Umut Güçlü and Marcel A. J. van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.

[48] Umut Güçlü and Marcel AJ van Gerven. Deep neural networks reveal a gradient in the complexity of neural representations across the ventral stream. *Journal of Neuroscience*, 35(27):10005–10014, 2015.

[49] Tian Guo, Tao Lin, and Nino Antulov-Fantulin. Exploring interpretable lstm neural networks over multi-variable data. In *International Conference on Machine Learning*, pages 2494–2504. PMLR, 2019.

[50] Michiel Hermans and Benjamin Schrauwen. Training and analysing deep recurrent neural networks. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 190–198. Curran Associates, Inc., 2013.

[51] Sepp Hochreiter and Jürgen Schmidhuber. Long short-term memory. *Neural computation*, 9(8):1735–1780, 1997.

[52] Ha Hong, Daniel LK Yamins, Najib J Majaj, and James J DiCarlo. Explicit information for category-orthogonal object properties increases along the ventral stream. *Nature neuroscience*, 19(4):613, 2016.

[53] Bo-Jian Hou and Zhi-Hua Zhou. Learning with interpretable structure from gated rnn. *IEEE transactions on neural networks and learning systems*, 31(7):2267–2279, 2020.

[54] Shailee Jain and Alexander G. Huth. Incorporating context into language encoding models for fmri. In *Proceedings of the 32nd International Conference on Neural Information Processing Systems*, NIPS'18, page 6629–6638, Red Hook, NY, USA, 2018. Curran Associates Inc.

[55] Mark Jenkinson, Peter Bannister, Michael Brady, and Stephen Smith. Improved optimization for the robust and accurate linear registration and motion correction of brain images. *NeuroImage*, 17(2):825–841, 2002.

[56] Mark Jenkinson and Stephen Smith. A global optimisation method for robust affine registration of brain images. *Medical Image Analysis*, 5(2):143–156, 2001.

[57] Kendrick N Kay, Thomas Naselaris, Ryan J Prenger, and Jack L Gallant. Identifying natural images from human brain activity. *Nature*, 452(7185):352–355, 2008.

[58] Nan Rosemary Ke, Anirudh Goyal, Olexa Bilaniuk, Jonathan Binas, Michael C Mozer, Chris Pal, and Yoshua Bengio. Sparse attentive backtracking: Temporal credit assignment through reminding. In S. Bengio, H. Wallach, H. Larochelle, K. Grauman, N. Cesa-Bianchi, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 31, pages 7640–7651. Curran Associates, Inc., 2018.

[59] Alexander JE Kell and Josh H McDermott. Deep neural network models of sensory systems: windows onto the role of task constraints. *Current opinion in neurobiology*, 55:121–132, 2019.

[60] Alexander J.E. Kell, Daniel L.K. Yamins, Erica N. Shook, Sam V. Norman-Haignere, and Josh H. McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644.e16, May 2018.

[61] Alexander JE Kell, Daniel LK Yamins, Erica N Shook, Sam V Norman-Haignere, and Josh H McDermott. A task-optimized neural network replicates human auditory behavior, predicts brain responses, and reveals a cortical processing hierarchy. *Neuron*, 98(3):630–644, 2018.

[62] Tim C. Kietzmann, Courtney J. Spoerer, Lynn K. A. Sörensen, Radoslaw M. Cichy, Olaf Hauk, and Nikolaus Kriegeskorte. Recurrence is required to capture the representational dynamics of the human visual system. *Proceedings of the National Academy of Sciences*, 116(43):21854–21863, 2019.

[63] Wayne K Kirchner. Age differences in short-term retention of rapidly changing information. *Journal of experimental psychology*, 55(4):352, 1958.

[64] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International Conference on Machine Learning*, pages 3519–3529. PMLR, 2019.

[65] Nikolaus Kriegeskorte. Relating population-code representations between man, monkey, and computational models. *Frontiers in Neuroscience*, 3:35, 2009.

[66] Nikolaus Kriegeskorte. Deep neural networks: a new framework for modeling biological vision and brain information processing. *Annual review of vision science*, 1:417–446, 2015.

[67] Nikolaus Kriegeskorte, Marieke Mur, and Peter Bandettini. Representational similarity analysis - connecting the branches of systems neuroscience. *Frontiers in Systems Neuroscience*, 2:4, 2008.

[68] Nikolaus Kriegeskorte, Marieke Mur, Douglas A Ruff, Roozbeh Kiani, Jerzy Bodurka, Hossein Esteky, Keiji Tanaka, and Peter A Bandettini. Matching categorical object representations in inferior temporal cortex of man and monkey. *Neuron*, 60(6):1126–1141, 2008.

[69] Angela R Laird, P Mickle Fox, Cathy J Price, David C Glahn, Angela M Uecker, Jack L Lancaster, Peter E Turkeltaub, Peter Kochunov, and Peter T Fox. Ale meta-analysis: Controlling the false discovery rate and performing statistical contrasts. *Human brain mapping*, 25(1):155–164, 2005.

[70] C. Lanczos. Evaluation of noisy data. *Journal of the Society for Industrial and Applied Mathematics Series B Numerical Analysis*, 1(1):76–85, 1964.

[71] Yixuan Li, Jason Yosinski, Jeff Clune, Hod Lipson, and John E Hopcroft. Convergent learning: Do different neural networks learn the same representations? In *FE@ NIPS*, pages 196–212, 2015.

[72] Tsung-Yu Lin and Subhransu Maji. Visualizing and understanding deep texture representations. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 2791–2799, 2016.

[73] Martin A Lindquist and Tor D Wager. Principles of functional magnetic resonance imaging. *Handbook of neuroimaging data analysis*, pages 3–48, 2014.

[74] Christian K Machens, Michael S Wehr, and Anthony M Zador. Linearity of cortical receptive fields measured with natural sounds. *Journal of Neuroscience*, 24(5):1089–1100, 2004.

[75] Aravindh Mahendran and Andrea Vedaldi. Understanding deep image representations by inverting them. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 5188–5196, 2015.

[76] Niru Maheswaranathan, Alex H Williams, Matthew D Golub, Surya Ganguli, and David Sussillo. Universality and individuality in neural dynamics across large populations of recurrent networks. *Advances in neural information processing systems*, 2019:15629, 2019.

[77] Mahshid Majd and Reza Safabakhsh. A motion-aware convlstm network for action recognition. *Applied Intelligence*, 49(7):2515–2521, 2019.

[78] Akira Miyake and Priti Shah. *Models of working memory: Mechanisms of active maintenance and executive control*. Cambridge University Press, 1999.

[79] Yalda Mohsenzadeh, C. Mullin, D. Pantazis, and A. Oliva. Emergence of topographical correspondences between deep neural network and human ventral visual cortex. In *Conference on Cognitive Computational Neuroscience, 5-8 September 2018, Philadelphia, Pennsylvania*, 2018.

[80] Yalda Mohsenzadeh, Caitlin Mullin, Dimitrios Pantazis, and Aude Oliva. Emergence of topographical correspondences between deep neural network and human ventral visual cortex, 2019.

[81] Martin M Monti. Statistical analysis of fmri time-series: a critical review of the glm approach. *Frontiers in human neuroscience*, 5:28, 2011.

[82] Ari S Morcos, Maithra Raghu, and Samy Bengio. Insights on representational similarity in neural networks with canonical correlation. *arXiv preprint arXiv:1806.05759*, 2018.

[83] Thomas Naselaris, Kendrick N Kay, Shinji Nishimoto, and Jack L Gallant. Encoding and decoding in fmri. *Neuroimage*, 56(2):400–410, 2011.

[84] Aran Nayebi, D. Bear, J. Kubilius, Kohitij Kar, S. Ganguli, David Sussillo, J. DiCarlo, and Daniel Yamins. Task-driven convolutional recurrent models of the visual system. *ArXiv*, abs/1807.00053, 2018.

[85] Yuval Nir, Lior Fisch, Roy Mukamel, Hagar Gelbard-Sagiv, Amos Arieli, Itzhak Fried, and Rafael Malach. Coupling between neuronal firing rate, gamma lfp, and bold fmri is related to interneuronal correlations. *Current biology*, 17(15):1275–1285, 2007.

[86] Sam V Norman-Haignere and Josh H McDermott. Neural responses to natural and model-matched stimuli reveal distinct computations in primary and nonprimary auditory cortex. *PLoS biology*, 16(12):e2005127, 2018.

[87] Seiji Ogawa and Tso-Ming Lee. Magnetic resonance imaging of blood vessels at high fields: in vivo and in vitro measurements and image simulation. *Magnetic resonance in medicine*, 16(1):9–18, 1990.

[88] Seiji Ogawa, Tso-Ming Lee, Alan R Kay, and David W Tank. Brain magnetic resonance imaging with contrast dependent on blood oxygenation. *proceedings of the National Academy of Sciences*, 87(24):9868–9872, 1990.

[89] Adrian M Owen, Kathryn M McMillan, Angela R Laird, and Ed Bullmore. N-back working memory paradigm: A meta-analysis of normative functional neuroimaging studies. *Human brain mapping*, 25(1):46–59, 2005.

[90] Razvan Pascanu, Caglar Gulcehre, Kyunghyun Cho, and Yoshua Bengio. How to construct deep recurrent neural networks. *arXiv preprint arXiv:1312.6026*, 2013.

[91] Razvan Pascanu, Tomas Mikolov, and Yoshua Bengio. On the difficulty of training recurrent neural networks. In *International conference on machine learning*, pages 1310–1318. PMLR, 2013.

[92] Luiz Pessoa, Eva Gutierrez, Peter A Bandettini, and Leslie G Ungerleider. Neural correlates of visual working memory: fmri amplitude predicts task performance. *Neuron*, 35(5):975–987, 2002.

[93] Jonathan D. Power, Anish Mitra, Timothy O. Laumann, Abraham Z. Snyder, Bradley L. Schlaggar, and Steven E. Petersen. Methods to detect, characterize, and remove motion artifact in resting state fmri. *NeuroImage*, 84(Supplement C):320–341, 2014.

[94] Maithra Raghu, Justin Gilmer, Jason Yosinski, and Jascha Sohl-Dickstein. Svcca: Singular vector canonical correlation analysis for deep understanding and improvement. *network*, 200(200):200, 2017.

[95] Rishi Rajalingham, Elias B Issa, Pouya Bashivan, Kohitij Kar, Kailyn Schmidt, and James J DiCarlo. Large-scale, high-resolution comparison of the core visual object recognition behavior of humans, monkeys, and state-of-the-art deep artificial neural networks. *Journal of Neuroscience*, 38(33):7255–7269, 2018.

[96] JO Ramsay, Jos ten Berge, and GPH Styan. Matrix correlation. *Psychometrika*, 49(3):403–423, 1984.

[97] Claudia Rottschy, Robert Langner, Imis Dogan, Kathrin Reetz, Angela R Laird, Jörg B Schulz, Peter T Fox, and Simon B Eickhoff. Modelling neural correlates of working memory: a coordinate-based meta-analysis. *Neuroimage*, 60(1):830–846, 2012.

[98] David E Rumelhart, Peter M Todd, et al. Learning and connectionist representations. *Attention and performance XIV: Synergies in experimental psychology, artificial intelligence, and cognitive neuroscience*, 2:3–30, 1993.

[99] Joseph B Sala and Susan M Courtney. Binding of what and where during working memory maintenance. *Cortex*, 43(1):5–21, 2007.

[100] Theodore D. Satterthwaite, Mark A. Elliott, Raphael T. Gerraty, Kosha Ruparel, James Loughead, Monica E. Calkins, Simon B. Eickhoff, Hakon Hakonarson, Ruben C. Gur, Raquel E. Gur, and Daniel H. Wolf. An improved framework for confound regression and filtering for control of motion artifact in the preprocessing of resting-state functional connectivity data. *NeuroImage*, 64(1):240–256, 2013.

[101] Xingjian Shi, Zhourong Chen, Hao Wang, D. Yeung, W. Wong, and Wang chun Woo. Convolutional lstm network: A machine learning approach for precipitation nowcasting. In *NIPS*, 2015.

[102] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. *arXiv preprint arXiv:1409.1556*, 2014.

[103] Edward E Smith and John Jonides. Working memory: A view from neuroimaging. *Cognitive psychology*, 33(1):5–42, 1997.

[104] Edward E Smith and John Jonides. Storage and executive processes in the frontal lobes. *Science*, 283(5408):1657–1661, 1999.

[105] Paul Smolensky. Connectionist ai, symbolic ai, and the brain. *Artificial Intelligence Review*, 1(2):95–109, 1987.

[106] Hongmei Song, Wenguan Wang, Sanyuan Zhao, Jianbing Shen, and Kin-Man Lam. Pyramid dilated deeper convlstm for video salient object detection. In *Proceedings of the European conference on computer vision (ECCV)*, pages 715–731, 2018.

[107] Thomas C. Sprague, Geoffrey M. Boynton, and John T. Serences. The importance of considering model choices when interpreting results in computational neuroimaging. *eNeuro*, 6(6), 2019.

[108] Heung-Il Suk, Chong-Yaw Wee, Seong-Whan Lee, and Dinggang Shen. State-space model with deep learning for functional dynamics estimation in resting-state fmri. *NeuroImage*, 129:292–307, 2016.

[109] Maki Suzuki, Toshikazu Kawagoe, Shu Nishiguchi, Nobuhito Abe, Yuki Otsuka, Ryusuke Nakai, Kohei Asano, Minoru Yamada, Sakiko Yoshikawa, and Kaoru Sekiyama. Neural correlates of working memory maintenance in advanced aging: Evidence from fmri. *Frontiers in Aging Neuroscience*, 10:358, 2018.

[110] Amirhessam Tahmassebi, Amir H. Gandomi, Ian McCann, Mieke H. J. Schulte, Anna E. Goudriaan, and Anke Meyer-Baese. Deep learning in medical imaging: Fmri big data analysis via convolutional

neural networks. In *Proceedings of the Practice and Experience on Advanced Research Computing*, PEARC '18, New York, NY, USA, 2018. Association for Computing Machinery.

[111] Peter E Turkeltaub, Guinevere F Eden, Karen M Jones, and Thomas A Zeffiro. Meta-analysis of the functional neuroanatomy of single-word reading: method and validation. *Neuroimage*, 16(3):765–780, 2002.

[112] N. J. Tustison, B. B. Avants, P. A. Cook, Y. Zheng, A. Egan, P. A. Yushkevich, and J. C. Gee. N4itk: Improved n3 bias correction. *IEEE Transactions on Medical Imaging*, 29(6):1310–1320, 2010.

[113] S Urchs, J Armoza, and C et al. Moreau. Mist: A multi-resolution parcellation of functional brain networks [version 2; peer review: 4 approved]. In *MNI Open Res*, 2019.

[114] David C. Van Essen, Stephen M. Smith, Deanna M. Barch, Timothy E. J. Behrens, Essa Yacoub, Kamil Ugurbil, and WU-Minn HCP Consortium. The wu-minn human connectome project: an overview. *NeuroImage*, 80:62–79, Oct 2013. 23684880[pmid].

[115] Tor D Wager and Edward E Smith. Neuroimaging studies of working memory. *Cognitive, Affective, & Behavioral Neuroscience*, 3(4):255–274, 2003.

[116] Tor D. Wager and Edward E. Smith. Neuroimaging studies of working memory:. *Cognitive, Affective, & Behavioral Neuroscience*, 3(4):255–274, Dec 2003.

[117] EY Walker et al. Inception in visual cortex: in vivo-silico loops reveal most exciting images. biorxiv published online december 28, 2018, 2018.

[118] Aria Wang, Michael Tarr, and Leila Wehbe. Neural taskonomy: Inferring the similarity of task-derived representations from brain activity. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32, pages 15501–15511. Curran Associates, Inc., 2019.

[119] Panqu Wang, Vicente Malave, and Ben Cipollini. Encoding voxels with deep learning. *The Journal of neuroscience : the official journal of the Society for Neuroscience*, 35(48):15769–15771, Dec 2015. 26631460[pmid].

[120] Dong Wen, Zhenhao Wei, Yanhong Zhou, Guolin Li, Xu Zhang, and Wei Han. Deep learning methods to process fmri data and their application in the diagnosis of cognitive impairment: a brief overview and our opinion. *Frontiers in neuroinformatics*, 12:23, 2018.

[121] Paul J Werbos. Backpropagation through time: what it does and how to do it. *Proceedings of the IEEE*, 78(10):1550–1560, 1990.

[122] Scott Wisdom, Thomas Powers, James Pitton, and Les Atlas. Interpretable recurrent neural networks using sequential sparse recovery. *arXiv preprint arXiv:1611.07252*, 2016.

[123] Michael C-K Wu, Stephen V David, and Jack L Gallant. Complete functional characterization of sensory neurons by system identification. *Annu. Rev. Neurosci.*, 29:477–505, 2006.

[124] Daniel L Yamins, Ha Hong, Charles Cadieu, and James J DiCarlo. Hierarchical modular optimization of convolutional networks achieves representations similar to macaque it and human ventral stream. In C. J. C. Burges, L. Bottou, M. Welling, Z. Ghahramani, and K. Q. Weinberger, editors, *Advances in Neural Information Processing Systems*, volume 26, pages 3093–3101. Curran Associates, Inc., 2013.

[125] Daniel L. K. Yamins, Ha Hong, Charles F. Cadieu, Ethan A. Solomon, Darren Seibert, and James J. DiCarlo. Performance-optimized hierarchical models predict neural responses in higher visual cortex. *Proceedings of the National Academy of Sciences*, 111(23):8619–8624, 2014.

[126] Daniel LK Yamins and James J DiCarlo. Using goal-driven deep learning models to understand sensory cortex. *Nature neuroscience*, 19(3):356–365, 2016.

[127] Jason Yosinski, Jeff Clune, Yoshua Bengio, and Hod Lipson. How transferable are features in deep neural networks? *arXiv preprint arXiv:1411.1792*, 2014.

[128] Dong Yu and Michael L Seltzer. Improved bottleneck features using pretrained deep neural networks. In *Twelfth annual conference of the international speech communication association*, 2011.

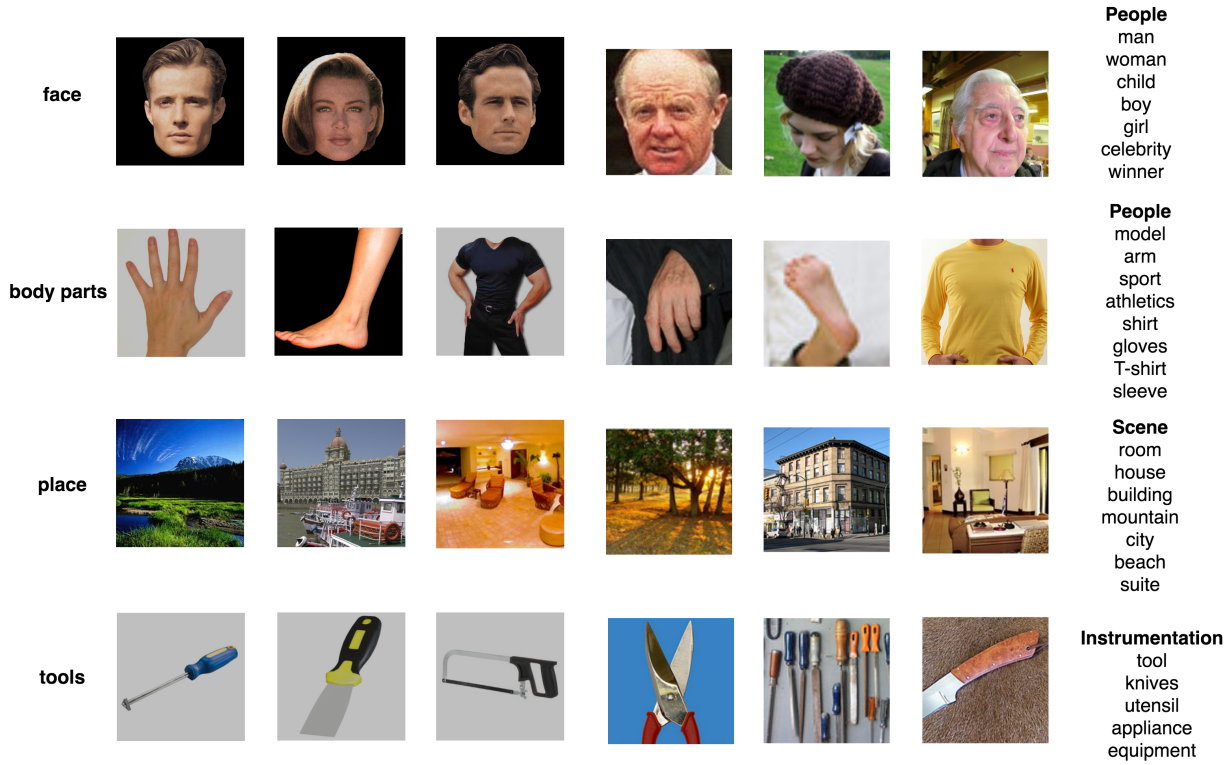
[129] Matthew D Zeiler and Rob Fergus. Visualizing and understanding convolutional networks. In *European conference on computer vision*, pages 818–833. Springer, 2014.

[130] Y. Zhang, M. Brady, and S. Smith. Segmentation of brain MR images through a hidden markov random field model and the expectation-maximization algorithm. *IEEE Transactions on Medical Imaging*, 20(1):45–57, 2001.

Appendix A

Task Network Models : Additional details

A.1. Training Dataset



(a) HCP stimulus images for the 4 categories indicated on the left

(b) ImageNet images with the category (synset) labels from the dataset indicated on the right

Fig. A.1. Sample images from the Human Connectome Project (HCP) [**114**] Working Memory (WM) task stimuli with their categories that are shown side-by-side with some ImageNet [**25**] images used in the training that are related to the stimuli image distribution

A.2. Recognition Module Training

As stated in Section 2.4.1, a Convolutional Neural Network (CNN) called the VGG-16 [**102**] that was pretrained on the ImageNet dataset was considered. The fully connected layers of this network were removed and replaced by a new set of linear layers as shown in Figure A.2.

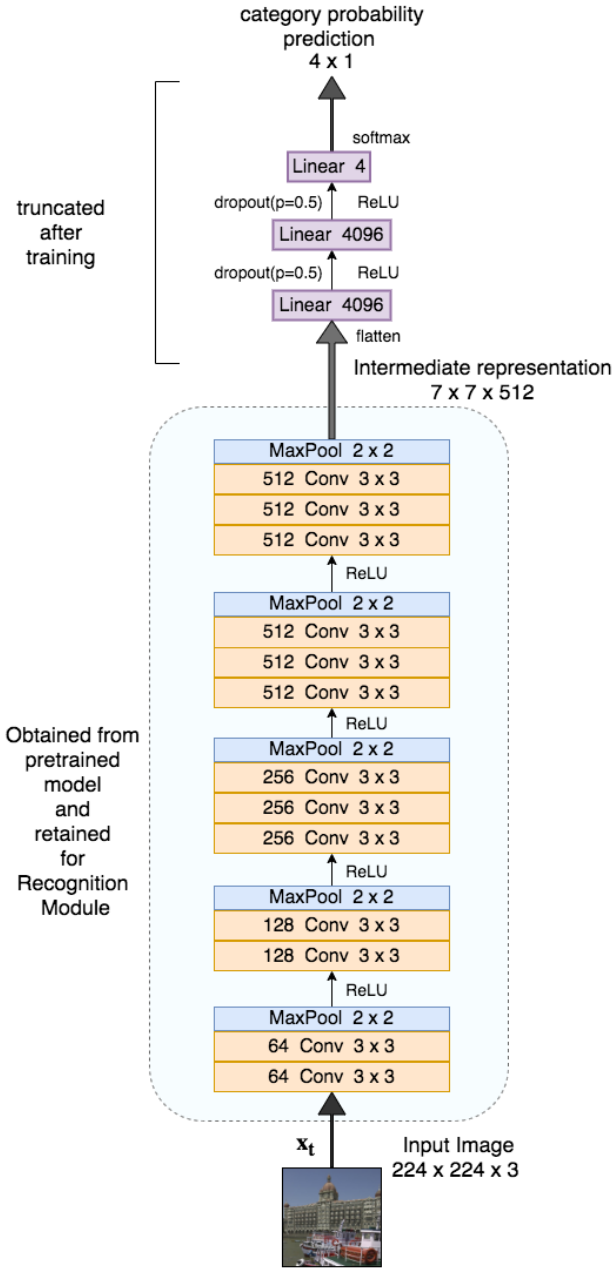


Fig. A.2. The entire Convolutional Neural Network (CNN) used for training the Recognition Module before truncation (Based on VGG-16)[**102**]. After training, the convolution and pooling units (in the bottom, inside the box with the dashed line) are retained while the linear units (in the top, shown in purple) that constitute the classifier are truncated.

The last layer in Figure A.2 shows a linear layer of 4 units with softmax activation to indicate the four categories in the task stimuli (face, body parts, place, tools). The actual category labels of the images in the sampled dataset are encoded as one-hot vectors. The predicted softmax values and actual labels are used to compute the cross-entropy loss for training.

Hyperparameter	**Value used in CNN Training**
Number of conv layers	13
Number of fully connected layers	3
Convolution Kernel Size, k (all conv layers)	3x3
Batch Size, b	64
Optimizer	Adam
Learning Rate, lr	0.003
Dropout probability in Fully Connected layers, p	0.5

Table A.1. The hyperparameter values used in training the Convolutional Neural Network before using it as the Recognition Module in the task network

Table A.1 lists the hyperparameters used in this network for the object categorization task. Dropout layers are used only in the linear layers to avoid overfitting to the training set.

A.3. Memory Module Training

A.3.1. 2-back

Hyperparameter	**LSTM**	**LSTM-SAB**	**ConvLSTM**
Number of recurrent layers, L	4	4	4
Hidden layer dimension, $\|h\|$	512	512	512 x 7 x 7
Sequence length, T	10	10	10
Batch size, b	32	32	16
Optimizer	RMSProp	RMSProp	RMSProp
Learning rate, lr	0.0001	0.0001	0.00022
Optimizer decay rate, α	0.99	0.99	0.95
k_{top}	-	3	-
k_{att}	-	1	-
k_{trunc}	-	3	-
Hidden layer kernel size	-	-	3x3
Hidden layer padding	-	-	1

Table A.2. 2-back task : The hyperparameter values used in training the task network which consists of both the Recognition and Memory Modules

In the task networks for the 2-back task, the hyperparameters used are shown in Table A.2. The batch size was determined by using a validation set to get the best performance. The sequence length T is set to 10 to account for the 10 timesteps in the task block when the stimuli are shown.

A.3.2. 0-back

The 0-back task networks were trained with the hyperparameters used shown in Table A.3.

Hyperparameter	**LSTM**	**LSTM-SAB**	**ConvLSTM**
Number of recurrent layers, L	4	4	4
Hidden layer dimension, $\|h\|$	512	512	512 x 7 x 7
Sequence length, T	11	11	11
Batch size, b	16	16	16
Optimizer	RMSProp	RMSProp	RMSProp
Learning rate, lr	0.00025	0.00020	0.0001
Optimizer decay rate, α	0.99	0.99	0.90
k_{top}	-	3	-
k_{att}	-	1	-
k_{trunc} (same as T above)	-	11	-
Hidden Layer kernel size	-	-	3x3
Hidden layer padding	-	-	1

Table A.3. 0-back task : The hyperparameter values used in training the task network which consists of both the Recognition and Memory Modules

The sequence length is 11 in the case of 0-back as opposed to 10 in 2-back is because of the extra cue stimulus image.