Vassar College
Digital Window @ Vassar

Senior Capstone Projects

2019

Investigation of optimal receptive field size for maximizing mutual information and increasing learning efficiency

Thomas J. Possidente

Recommended Citation
Possidente, Thomas J., "Investigation of optimal receptive field size for maximizing mutual information and increasing learning efficiency" (2019). *Senior Capstone Projects*. 848.
https://digitalwindow.vassar.edu/senior_capstone/848

Investigation of Optimal Receptive Field Size for Maximizing Mutual Information and Increasing Learning Efficiency

Thomas J. Possidente

Senior COGS Thesis

Abstract

As a parameter within an information processing system, receptive field (RF) size should be tuned to maximize information transmission per unit of energy expended. We hypothesize that tuning RF size to maximize the amount of mutual information per RF would result in more efficient statistical learning due to the increased predictability of patterns in inputs. We argue that convolutional neural networks (CNNs) perform statistical learning and are information processing systems governed by the principles of information theory and are thus adequate models of statistical learning in systems like the brain. In this experiment we generate sets of inputs with high mutual information in different spatial resolutions, then use supervised CNNs with various receptive field sizes to classify these inputs. Our results show that RF sizes that increase mutual information per RF generally result in more efficient statistical learning. In light of these results, we contend that RF sizes that increase mutual information per RF could also be advantageous for neural circuits in the brain that perform statistical learning, and thus maximizing mutual information could have been a factor in the evolution of receptive field size.

Introduction

The brain is fundamentally an information processing system. Its primary task is to make sense of sensory input data within the context of other data[1] to produce relevant actions and form relevant associations. As with any information processing system, the brain must operate within a set of constraints dictated by the laws of information theory. This means it should work to maximize information encoding efficiency. But as a part of a biological system, the brain is also subject to constraints dictated by the laws of evolution, meaning it should work to decrease the metabolic cost of its operations. Therefore, natural selection should favor a brain that transmits a maximized amount of information at a high signal-to-noise ratio[2] in a way that expends the least amount of energy possible.

There is abundant evidence to suggest that the brain does indeed work to maximize the amount of information gained from each Joule of energy expended. This idea is termed the Efficient Metabolic Principle by Stone (2018). Stone outlines several physiological parameters in the visual system that are tuned to transmit the most information at a low metabolic cost. These parameters include the coding of red, green, and blue cone photorecptor outputs in retinal ganglion cells (Buchsbaum & Gottchalk, 1983; Mather, 2006), the spacing of ganglion cell receptive fields (Borghuis et. al., 2008; Balasubramanian & Sterling, 2009; Vincent et. al., 2003), temporal receptive field structures (Srinivasan, Laughlin, & Dubs, 1982;), and the on/off concentric circle spatial structure of receptive fields (Srinivasan et. al., 1982; Hosoya, Baccus, & Meister, 2005). Each of these parameters has been tuned in such a way as to increase information encoding per unit of energy expended.

[1] Such as previous sensory input data and pre-existing associations.

[2] Signal-to-Noise Ratio compares the level of the desired signal to the level of background noise. A whisper to a friend in a noisy crowd has a low signal-to-noise ratio, while a shout to a friend in a silent room has a high signal-to-noise ratio.

The Efficient Metabolic Principle suggests that when physiological constraints allow it, more information per Joule of energy expended is always preferable. Stone (2018) also adds an important qualifier - that coding efficiency seems to be somewhat secondary to metabolic efficiency (i.e., information is maximized for a given metabolic cost, but metabolic efficiency is not often sacrificed for information coding efficiency).

Another physiological parameter of the visual system (and other systems) in the nervous system is the size of receptive fields (RFs). It is unclear exactly why RFs are the size that they are but given the evidence above it seems likely that their size is governed by metabolic and information-based constraints. Thus, this experiment seeks to investigate whether the size of RFs are optimized to increase mutual information (MI) per RF and thus increase metabolic efficiency and aid in statistical learning. We will use convolutional artificial neural networks (CNNs), a type of artificial neural network (ANN), as a model for statistical learning and information processing in general in the brain.

Information Theory and Mutual Information

Information theory is the mathematical representation of concepts, parameters, and rules governing the transmission of messages through communication systems (Martignon, 2001). In a more abstract sense, concepts from information theory can be applied to any system in which a message is transmitted through encoding and decoding. The message can be anything and the communication system through which it passes can be realized in many forms (ex. telephone wires, computers, the brain). Formally conceptualized by Claude Shannon (1948), Information Theory has since been expanded and applied to diverse fields of study, especially in cognitive science where it has been applied extensively to areas such as the study of consciousness

(Tononi, 2017), cognitive control (Fan, 2014), and cognitive linguistics (Hale, 2003). In this investigation we will use the concept of MI from Information Theory to conceptualize statistical learning as it relates to RF size in the processing of sensory information.

Information Theory dictates that the standard unit of measurement of information is a bit. Essentially, one bit of information is equal to the uncertainty resolved from a binary decision with equally likely outcomes, regardless of the system. This means that in a binary branching path diagram (Figure 1) it takes one bit of information to specify a path from one node to another, as long as the probability of branching left and right are equal. In order to express directions to one specific leaf node from the most interior node in Figure 1, it would require 4 binary decisions and therefore 4 bits of information. By standardizing the measurement of information in this way, any system that transmits information can be compared quantitatively. For example, the result of flipping a fair coin carries less information than the result of picking a number out of a hat containing numbers 1-8. This is because the number of yes/no questions (i.e. binary decisions) needed to specify a number between 1 and 8 (inclusive) is greater than the number needed to specify a binary. In this case, it would take 3 bits to specify the number[3], and only 1 to specify the result of the coin flip.

[3] 1) Is the number less than or equal to 4? – No. 2) Is the number less than or equal to 6? – Yes. 3) Is the number 5? – No. As a result of these 3 yes/no questions the number must be 6.

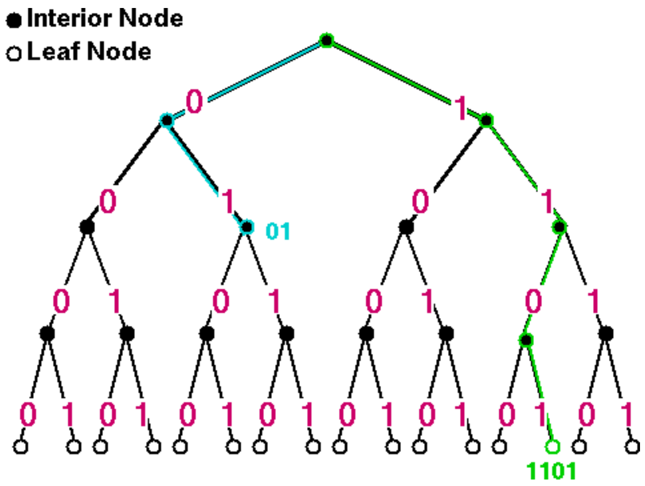


Figure 1 shows a binary branching path diagram in which each binary path choice is represented by a 0 or a 1. The blue and green paths show the binary directions required to get a specific node. The blue path indicates 2 bits of information while the green path requires 4.

In most practical applications, the probabilities of outcomes are not perfectly equal. In these situations, information is more easily measured using the formula:

$$H = -\sum_{i=1}^{n} p_i \cdot \log_2(p_i)$$

where H is information in bits, n is the number of different event possibilities, and p is the probability of event i occurring. This formula allows for the calculation of the information from many events with varying probabilities. One example of this would be the information gained from finding out the outcome of the total of the numbers on two fair dice. Because there are 11 possible outcomes and some values have different probabilities of occurring than others (e.g., 2 will be rolled about 3% of the time, while 7 will be rolled about 17% of the time), calculating the information gained from rolling the dice is not as easy as in Figure 1. Using the formula above (where p_i is the probability of rolling a certain number) we find that about 3.31 bits of information are gained from rolling 2 dice. Notice that if the probabilities of each outcome were equal, the information content would be greater (3.46). This agrees with our intuition that if events are equally likely, we resolve more uncertainty (and thus gain more information) through finding the outcome of the event compared with a case in which there is a high probability of one event occurring and all other events are low probability. In the latter case, we already have a good idea of what the outcome of the event will be, while in the former case we are completely uncertain, thus more information is gained from finding out the outcome. The ability to formally measure information through this formula was a breakthrough that spurred progress in a diverse range of fields, such as cryptography (Shannon, 1949) and data compression (Ziv & Lempel, 1977).

MI is a concept in Information Theory that deals with the relationship between two random variables that are sampled simultaneously. Ideas that are closely related to this concept are co-occurrence and correlation between two variables. The mathematical formulation of MI for two discrete variables[4] is:

$$I(X;Y) = \sum_{y \in Y} \sum_{x \in X} p(x,y) \log \left(\frac{p(x,y)}{p(x)\,p(y)} \right)$$

Where I(X;Y) is the MI between discrete variables X and Y, p(x,y) is the joint probability[5] of X and Y, p(x) and p(y) are the marginal probability[6] distribution functions of X and Y respectively, and the log function is base 2.

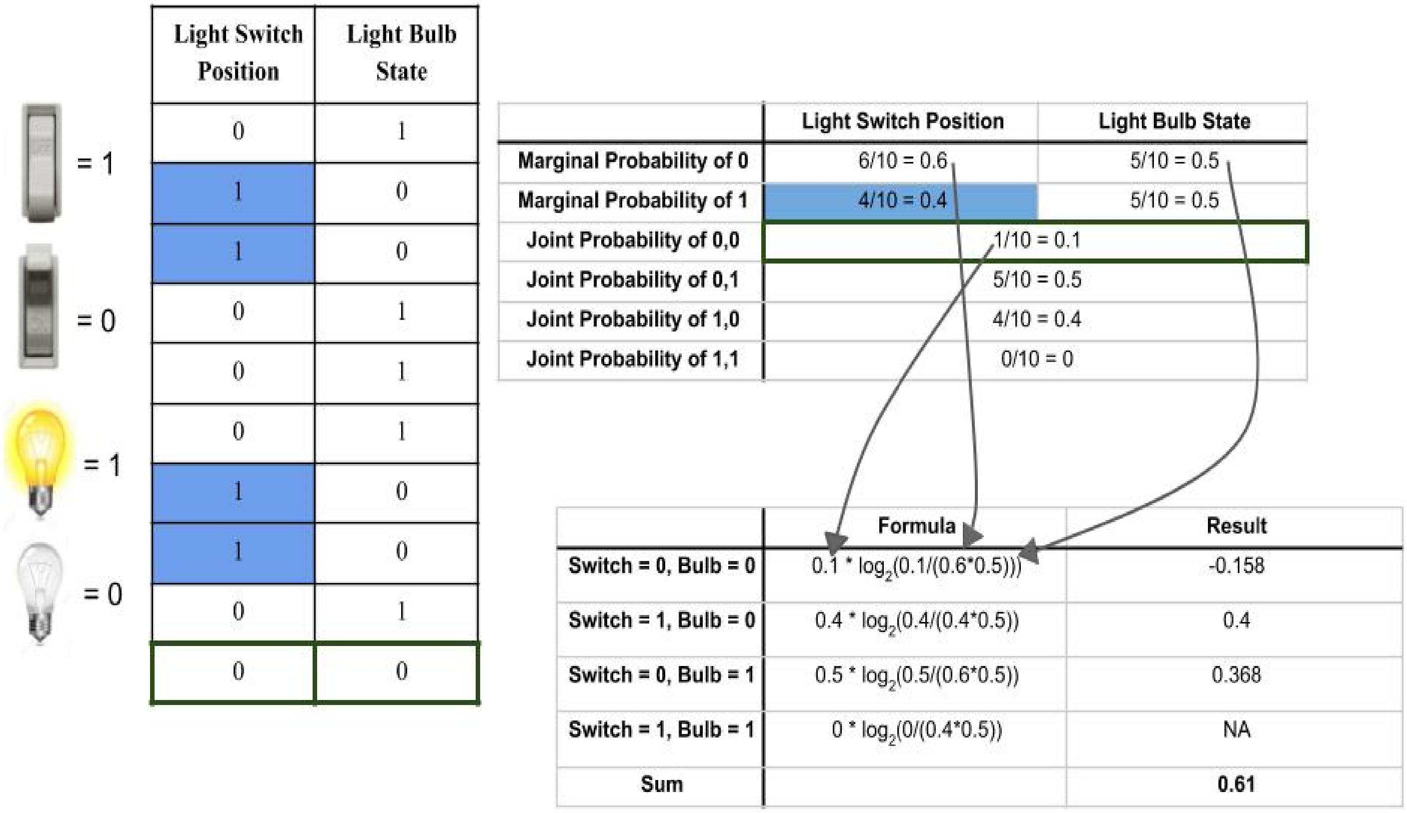
One simple example of two discrete variables with high MI is the position of a light switch and whether the light bulb it's connected to is producing light. Clearly, there is usually

[4] MI is measured differently for discrete and continuous variables, but for the purposes of this investigation only discrete variables will be relevant.

[5] Frequency that a specific combination of X and Y occurs in the set of sampled combinations.

[6] Frequency that a specific X value occurs in the set of sampled X values.

high MI between these variables because one variable reliably indicates the other and vice-versa. Figure 2 takes us through calculating the MI between these two variables. Clearly, the predictive relationship is not perfect, as can be seen in the 10th row of the leftmost table. In the 10th row we observe the light switch in the up position, but the light is not on. Perhaps the bulb has burned out. This row of data does not conform to the expected pattern formed by the other data, and thus it makes us less sure of the stable relationship between the two variables. Because of this, these variables do not achieve maximal MI. The MI between these two variables based on the observed samples is calculated in Figure 2 by determining the relevant marginal probabilities and joint probabilities, then plugging these values into the MI formula.



Light Switch Position	Light Bulb State
0	1
1	0
1	0
0	1
0	1
0	1
1	0
1	0
0	1
0	0

	Light Switch Position	Light Bulb State
Marginal Probability of 0	6/10 = 0.6	5/10 = 0.5
Marginal Probability of 1	4/10 = 0.4	5/10 = 0.5
Joint Probability of 0,0	1/10 = 0.1	
Joint Probability of 0,1	5/10 = 0.5	
Joint Probability of 1,0	4/10 = 0.4	
Joint Probability of 1,1	0/10 = 0	

	Formula	Result
Switch = 0, Bulb = 0	$0.1 * \log_2(0.1/(0.6*0.5)))$	-0.158
Switch = 1, Bulb = 0	$0.4 * \log_2(0.4/(0.4*0.5))$	0.4
Switch = 0, Bulb = 1	$0.5 * \log_2(0.5/(0.6*0.5))$	0.368
Switch = 1, Bulb = 1	$0 * \log_2(0/(0.4*0.5))$	NA
Sum		**0.61**

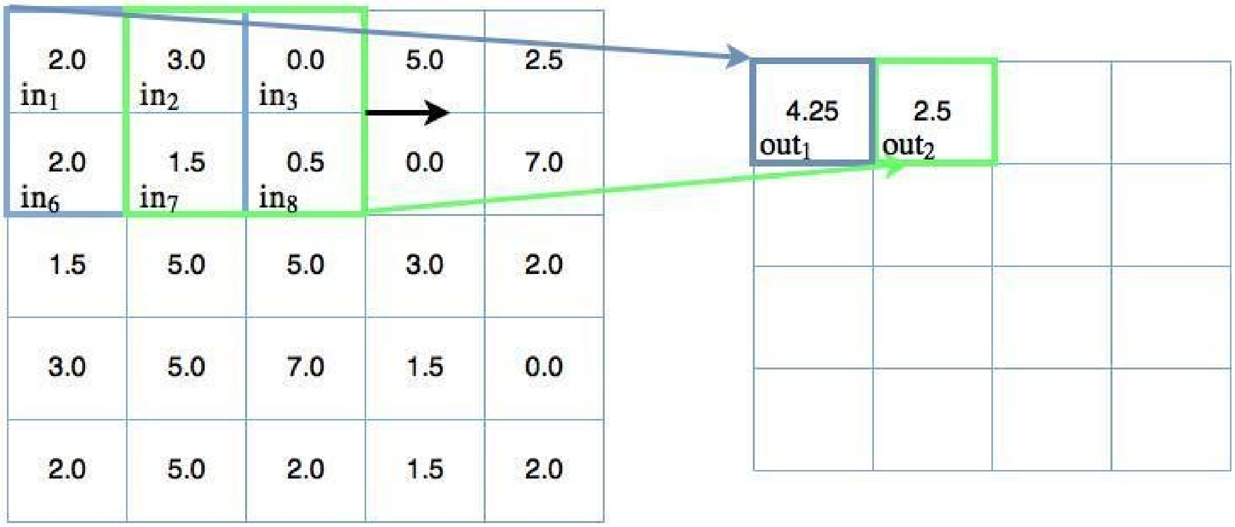
Figure 2 The table on the upper right shows the marginal and joint probabilities of the two variables examined in the table on the left. The blue cells demonstrate how the marginal probability of light switch position 1 was calculated (4 instances / 10 total observations = 0.4). The green highlighted cells show how the joint probability of light switch position 0 and light bulb state 0 was calculated (1 instance / 10

total observations = 0.1). The table on the bottom right shows the relevant probabilities plugged into the MI formula for each possible combination of the two variables in question. It also shows the result of each MI calculation and their sum, which is the total amount of MI between the two variables based on the observed data. The gray arrows demonstrate how the probabilities are plugged into the formula for one possible combination of the two variables (switch = 0, bulb = 0).

A final MI of 0.61 in Figure 2 means that if you know the state of one variable, you have gained 0.61 bits of information about the state of the other variable. If the light switch was in the down position in row 10, the predictive relationship would be perfect, and the MI would be equal to 1 bit. Recall that 1 bit is the amount of information required to disambiguate between two equally likely variables. Therefore, 0.61 bits is not enough information to be sure of the state of the other variable, although it does tell us something about what the other variable is likely to be.

A Brief Introduction to Convolutional Neural Networks

In this study we use CNNs as a model for statistical learning. CNNs are a type of ANN that are most commonly used for image classification and object recognition. They work by taking multiple “filters”, or groups of connection weights, and moving these filters sequentially through the pixels of the image all the while summing the product of the connection weights and pixel values to get an activation value that is passed on to the next layer in the network (Figure 3). The number of filters used, the size of the filters, and how many pixels the filter moves for each new calculation of activation are parameters in the network. The size of the filters is called the receptive field, while the number of pixels the filters slide over for each calculation is called the stride. Once each filter (each with different connection weights) has passed through the whole image, all the output matrices (known as feature maps) get passed on to be the inputs to the next layer of the network.



2.0 in_1	3.0 in_2	0.0 in_3	5.0	2.5
2.0 in_6	1.5 in_7	0.5 in_8	0.0	7.0
1.5	5.0	5.0	3.0	2.0
3.0	5.0	7.0	1.5	0.0
2.0	5.0	2.0	1.5	2.0

4.25 out_1	2.5 out_2		

Figure 3 The matrix on the left represents the input image pixel values. The blue box outlines the receptive field size (2x2) as it performs the first convolution. The connection weights (not shown) are all 0.5 and are multiplied by each pixel, then summed to get the output result shown in the matrix on the right. The green box outlines the receptive field after it has shifted over the image to perform its second convolution (stride size is 1). After the filter has moved over the whole image, another filter with different connection weights will restart the process.

Generally, after some number of convolutional layers, the feature maps are flattened into one long 1 dimensional array of values and fed into dense layers (Figure 4), the last of which classifies the image. These layers are what common ANNs are generally made out of and consist of some number of nodes, each of which takes into account each input value to produce an output value. The output of each node is determined by summing the products of each input value and its corresponding connection weight. Then, this value is put through a function (called an activation function) to determine the output of the node.

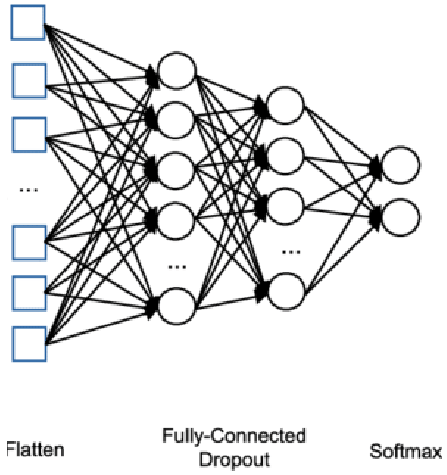


Figure 4 The flattened array of values from the feature maps (left) are fed into fully connected dense layers then to a dense layer using a softmax activation function for final classification.

In order to train a CNN, all connection weights start at random values between 0 and 1 (most commonly), then image input values are run through the layers of the network one at a time. For each image, the network's classification guess is compared to the true class of the image and some error metric is calculated (this metric is called loss). After some number of image inputs (called the batch size), a calculation called backpropagation is applied to change the connection weights in all layers of the network so as to try to decrease the error (loss). In a successful network, loss will decrease throughout training and the accuracy of classification will increase. A more comprehensive overview of CNNs, as well as more information about different activation functions and loss functions can be found in Rawat (2017).

Parallels Between Artificial Neural Networks and Biological Neural Circuits

Statistical Learning

Humans are thought to learn (at least in part) through internalizing statistical regularities in their environment in a process called statistical learning. If things (objects, object-properties, sounds, words, syllables, emotions, etc.) occur frequently in close spatial or temporal proximity, they become associated with each other. After reliable associations are formed, we often use them to predict things about our environment, such as the fact that if you see part of a fox poking out from behind a tree (Figure 3), you are likely to predict that the rest of the fox is behind the tree. This is because in a large number of previous exposures to foxes and other animals, heads tend to co-occur spatially with necks, bodies, legs, etc. This process of internalizing statistical regularities is thought to be a domain general mechanism of learning, likely occurring in the context of language learning (Abla, Katahira, & Okanoya, 2008; Cunillera et. al., 2006, McNealy, Mazziotta, & Dapretto, 2006), visual learning (Turk-Browne, Scholl, Chun & Johnson, 2009; Roser et. al., 2011), as well as auditory learning (McNealy, Mazziotta, & Dapretto, 2006).

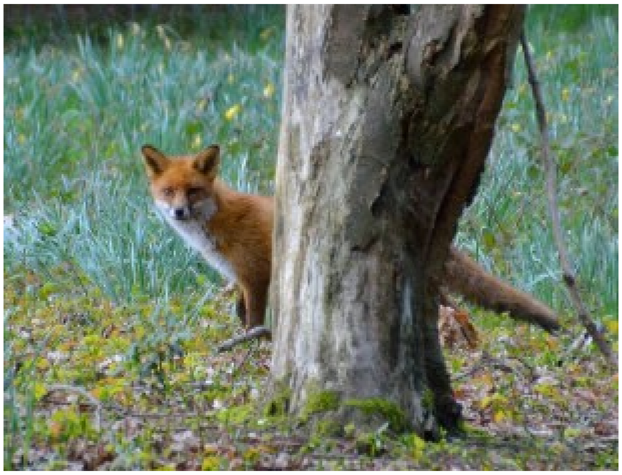


Figure 3 Shows a fox behind a tree. Although part of it is occluded by the tree, previous spatial associations between animals' heads and bodies cause us to predict that the fox's body extends behind the tree even though we cannot see it.

One of the most basic functions of artificial neural networks (ANNs) is to perform statistical learning – that is, to extract statistical regularities in input data (Bishop, 1999;

Ghahramani, 2015; Liu, 2018). In supervised learning[7], the task of an ANN is most often to find the patterns of features that consistently co-occur in inputs that are in the same categories. CNNs function similarly – throughout training each filter (which can be conceptualized as a set of connection weights) gradually adjusts to become attuned to a commonly occurring spatial pattern in the image set. Then, throughout training inputs that consistently express similar patterns of these features are labeled as a different category than other inputs that express a different pattern of features. ANNs, including CNNs, are simply iteratively adjusting connection weights according to the properties of the inputs presented to them, thereby learning through extracting statistical regularities. So, although they may do it differently at the mechanistic level, both brains and ANNs perform statistical learning.

Biological and Information Theoretic Comparison

ANNs and human neural circuits are both fundamentally information processing systems and thus follow the same information theoretic principles. Because a primary task of the brain and its circuitry is to encode and decode information about the world, it can be described in information theoretic terms. For example, sensory systems measure changes in energy in the world then encode them as patterned action potentials in sensory neurons. Then this information is decoded (usually in the brain) by extracting features in the changing energy patterns. In this way, information about the world is encoded and decoded by the human nervous system.

ANNs, and more specifically CNNs, can be thought of in similar terms. ANNs receive inputs in their input layer, encode these inputs into activation patterns across nodes in the hidden

[7] Supervised learning refers to iteratively adjusting parameters (connection weights) in an ANN based on the difference between the network's output and the correct output. Therefore, in supervised learning, having correctly labeled input data is necessary to train the network.

layer(s), then the output layer decodes these activation patterns into some meaningful representation of the inputs. Similarly, CNNs[8] receive two dimensional inputs and encode these inputs into 2D activation patterns by multiplying groups of input values with many different filters. These 2D activations are then flattened into a 1D vector which is fed into a standard ANN which then decodes the activation patterns into some meaningful representation of the inputs (ex. An output vector of [1,0] may represent that the input image contains a cat while [0,1] represents that it does not). Thus, both ANNs (and CNNs by extension) and biological neural circuits function by encoding and decoding information.

It has been made clear in previous sections that the brain operates according to principles of information theory, but what about ANNs? Here we seek to show that efficient information processing principles are fundamental in ANNs. We do this by attempting to show that 1) ANNs explicitly optimized for maximal information transference function efficiently in their tasks and 2) ANNs that are not explicitly optimized for maximal information transference by using information theory inspired learning rules are nevertheless successful and are actually indirectly optimized to process information efficiently.

Firstly, backpropagation methods that maximize MI between target output and ANN output result in successful and efficient learning (Santos, Alexandre, & de Sá, 2004; Silva, de Sá, & Alexandre, 2005). Additionally, learning rules that maximize the information transference in a network have been shown to perform redundancy reduction, effectively separating statistically independent components of their inputs and thus increasing information transference (Bell & Sejnowski, 1995). Similarly, Bichsel & Seitz (1989) show that setting connection weights in an

[8] Here and throughout we will restrict our scope to two-dimensional CNNs (i.e. CNNs in which the inputs as well as the filters that process the inputs are two dimensional). This is because we are working with binarized pattern matrices which are two dimensional.

ANN based on the minimal entropy (i.e., maximum information) principle optimizes network performance in small multilayer ANNs. These results indicate that ANNs that are optimized for explicitly maximizing information transference by use of principles from information theory are efficient and effective.

To address claim 2, we need only to state a few facts about successful ANNs.

1) Generally, successful classification ANNs have high MI between their outputs and the "ground truth" categories of the inputs being categorized.

The obvious goal of a classifying ANN is to have the ANN's classifications match the inputs' true classes. Essentially, any ANN that successfully does this is by definition maximizing MI between ANN output and the "ground truth" categories of the inputs. Thus, an inherent property of a categorizing ANN is to increase MI between inputs and outputs as much as possible.

2) A feature of an efficient ANN is that it reduces redundancy as much as possible, and thus performs efficient coding.

The goal of most ANNs is to learn to reliably map a relatively high number of input activations to a relatively low number of output neurons. A computationally efficient ANN uses the fewest number of nodes possible to do this. Therefore, ANNs simply work to maximize redundancy reduction in inputs, thus increasing the efficiency with which the information is processed. Of course, the amount of redundancy reduction possible depends on the categorization problem. For example, a CNN that works to determine whether a human face is present in a set of images containing chairs and faces will be able to perform more redundancy reduction than a CNN that works to categorize male human faces and female human faces. This is because the former CNN presumably needs only a relatively small amount of information from the pixels to decide whether the image is a chair or a face (whether a mouth shape occurs might even be enough – this could be accomplished by a single filter). But in the latter CNN, a larger amount of information would likely be needed to differentiate between human male and human female faces (whether a mouth shape occurs will not be enough anymore). Thus, the latter ANN may need to retain more of the redundant information to classify correctly. For example, it may need to use a combination of many filters that detect features such as face width, lip fullness, shape of hair, etc. And it may even be the case that very little redundancy reduction is possible for some complex classification problems. But generally, ANNs solve classification problems by reducing redundancy in the inputs by extracting only the information that is relevant for classification. Thus, in general, successful and efficient ANNs inherently follow principles of efficient information processing by maximizing MI between outputs and ground truth categories and by reducing redundancy.

In addition to these points, CNNs are used widely as models of biological visual system (Kriegeskorte, 2015). One replicated and generalized finding is that CNNs that perform better at object recognition tend to have representational spaces that are more similar to those found in the inferior temporal cortex in humans and some non-human primates (Yamins et. al., 2014; Khaligh-Razavi & Kriegeskorte, 2014). These results suggest that although CNNs are nowhere near equivalent to brains, they can act as good abstract models of some of the key processes involved in sensory information processing and statistical learning.

Receptive Fields

Biological receptive field size is a parameter in the sensory system that is of interest to us because it is a prime candidate for optimization based not only on metabolic constraints, but also based on information coding efficiency. We hypothesize that adjusting the size of a RF by maximizing the MI within each RF would in theory increase information processing efficiency and thus benefit statistical learning mechanisms. An additional (non-biological) parallel application of this idea is that CNN learning could be made more efficient by choosing a RF size that maximizes MI per RF. This would offer a systematic and theoretically grounded method of choosing a RF size when training CNNs, whereas currently RF sizes are generally picked using unprincipled hyperparameter tuning or some general rules of thumb and common practices ("Convolutional Neural Networks (CNNs / ConvNets)"; Rao, 2018).

The RF of a neuron is considered to be the region of sensory space (in the visual field, on the body, in auditory space) in which changes in the environment influence firing of the sensory neuron. Neurons in the visual, auditory, and somatosensory systems are known to have RFs. The concept of a RF also applies to neurons further along in sensory processing that synapse with groups of sensory neurons. For example, ganglion cells in the retina have RFs that encompass input from a population of photoreceptors. In turn, cells in the primary visual cortex (V1) have RFs that encompass input from a population of ganglion cells. RF sizes vary between and within stages of processing and in general RF size increases from lower to higher levels of processing (Amano, Wandell, & Dumoulin, 2009; Smith, et. al., 2001).

There are approximately 126 million photoreceptors in each human eye. These photoreceptors collect a huge amount of visual information about changes in light in the world. Much of this information is redundant and can be compressed while still maintaining its usefulness. Additionally, much of this information is not very useful to us and can be discarded. This lossy compression[9] of information is an example of an instance in which metabolic

[9] Lossy compression refers to compression of information in which some information is lost and not recovered.

efficiency can be closely tied to information coding efficiency. One way the brain works to compress this information while still maintaining its usefulness is likely through RFs. For example, the redundancy of information contained in red and green photoreceptor cone outputs[10] (and blue to a lesser extent) is compressed through adding and subtracting the RGB channels in such a way that creates 3 new channels whose results are less correlated, and therefore more efficient at transmitting information (a process called sum-difference encoding). Through efficient coding, redundancies in inputs can be eliminated, thus decreasing the amount of metabolic energy that needs to be expended carrying information.

We argue that the separation of incoming sensory information into smaller separate streams (via RFs) makes the processing of information more efficient in terms of metabolic cost and coding efficiency. By separating information into smaller streams whose size maximizes the amount of MI within the stream, the information processing system can more efficiently compress the information in each stream. To clarify this argument we can imagine a world in which the only things that exist are two-dimensional circles and squares that are all approximately the same size (let’s say they all fit inside a 5” by 5” box but not a 4” by 4” box). These shapes are placed in such a way that any 5” by 5” space is equally likely to contain a circle, square, or nothing. Suppose that for the beings in this world identifying circles and squares is the only thing that is important for survival and reproduction. Thus, statistical regularities in this world that are relevant to these beings would occur only at spatial scales smaller than 5” by 5” and it would be inefficient to analyze visual information for regularities at a spatial scale greater than 5” by 5”. Furthermore, we can find the spatial area that maximizes

[10] Neighboring red and green cones tend to have similar outputs because:
1) Neighboring points in the visual field tend to have similar colors
2) Red and green cones have similar tuning curves (they respond to light of similar frequencies)

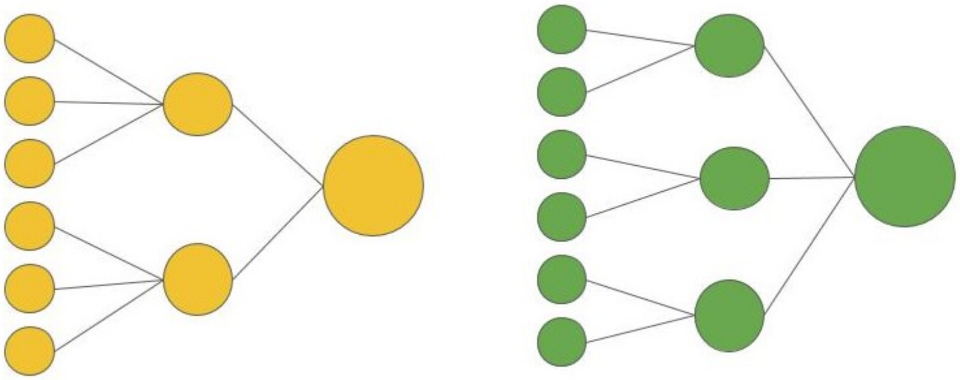
statistical regularities by finding the area in which MI is the greatest. By processing the visual world using MI-optimal RFs, these beings would ideally be able to compress all the information in a RF into just 1.58[11] bits of information, indicating whether the shape is a circle, a square, or neither. So, by processing the visual world through RFs that maximize MI, these beings would be able to perform the most information compression, thus processing information most efficiently. In the world we live in things are much more complicated because statistical regularities occur at many scales. But it stands to reason that there are some scales at which more regularities occur than others. Additionally, we are limited to collecting information at the range of scales determined by our sensors. For example, although statistical regularities occur at the microscopic level, these regularities cannot be detected by our visual system.

Evidence that supports the claim that separating information into smaller streams is more efficient can be found in the brain and in ANNs. Visual processing pathways in the human cortex exemplify the separation of information into smaller streams at both macro and micro levels. For example, at the macro level it is well known that the human visual system splits into two separate streams after processing in the occipital lobe - the ventral stream dedicated to object processing (sometimes called the “what” pathway) and the dorsal stream dedicated to processing the spatial relationship between agent and object (sometimes called the “how” or “where” pathway) (Goodale & Milner, 1992; Ungerleider & Haxby, 1994). One possible explanation for this division of information is that splitting the processing into two streams makes learning easier because separating out spatial information allows object information acquired in one spatial context to transfer to other spatial contexts (Rueckl, Cave, & Kosslyn, 1989). A similar kind of

[11] The formula for calculating information is $-\sum_i P_i * log_2(P_i)$ where P_i is the probability of outcome i. In this case there are 3 equally likely outcomes (square, circle, neither) so i = 3 and P_i = 1/3. This yields 1.58 bits.

information split is evident at the micro level, with initial visual processing relying on smaller RFs than later visual processing (Amano, Wandell, & Dumoulin, 2009; Smith, et. al., 2001). This too, may allow for more efficient learning. Jacobs and Kosslyn (1994) found that filtering inputs through small, non-overlapping Gaussian filters (conceptualized as RFs) before using them as inputs in a neural network led to better classification learning than with large overlapping Gaussian filters[12]. These results provide some possible clues as to why the visual system is structured the way it is. Dividing input into separate streams of processing likely makes information processing more efficient.

Our proposal is that the optimal RF size for capturing information in a set of stimuli can be estimated through finding the RF size that maximizes the MI available within each RF, and that this maximization would lead to more efficient statistical learning. But, if dividing the processing stream is beneficial as we have suggested, why is it that all RFs are not arbitrarily small? One reason is that this would likely increase the metabolic cost of information collection dramatically (Figure 4).



[12] The reverse was true for learning to identify single exemplars, but in this investigation, we are primarily interested in learning through statistical regularities between inputs (and therefore not single instances of an input).

Figure 4 In the yellow diagram, 6 sensory neurons output to 2 intermediate neurons which output to one neuron. The two neurons in the intermediate layer can be said to have a RF of size 3 (because they receive outputs from 3 sensory neurons). The green diagram is the same except there are 3 hidden neurons, each with an RF of 2. There are fewer neurons (9 < 10) and connections between neurons (8 < 9) in the yellow diagram, making it the less metabolically expensive choice.

A secondary reason we suggest here is that the information relevant to humans is generally structured at a particular resolution (one that is likely not arbitrarily small) and RF size should match this resolution. For example, there is an optimal RF size for capturing information about lines and edges in the world that are relevant to humans and there is a (different) RF size that is optimal for capturing information about the sound frequencies that are most relevant to humans.

But why should RF size match the resolution of relevant information in the world? The basic intuition is that if RF size is optimized based on maximization of MI per RF (i.e. matching RF size to the resolution of relevant information), this will increase the predictability of values within each RF, thus making processing more efficient and allowing for faster learning. And crucially, more MI per RF means that there is more redundancy within that RF, making for the possibility of greater compression of information. This means that in order to most efficiently gather and learn about information in the world, the structure of our sensory system should match the structure of information in the world.

To this end, the current study aims to investigate whether optimizing RF size based on maximizing the mutual MI per RF will lead to increased statistical learning in CNNs. If this is the case, MI maximization could be one potential factor contributing to the evolutionary process by which human RF size is influenced.

Methods and Results

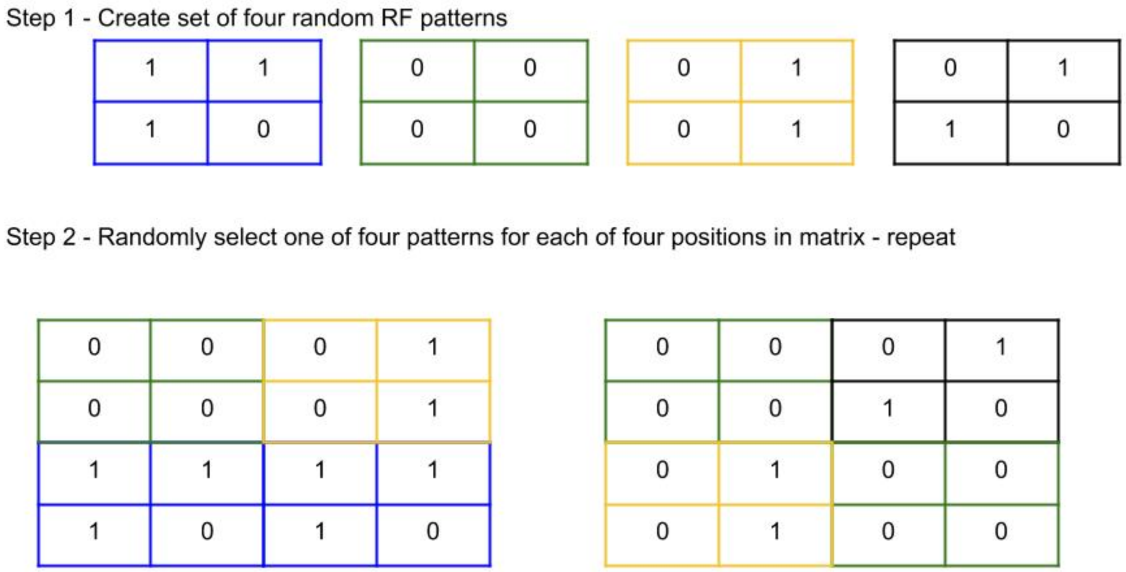
Overview

In this investigation we begin by creating sets of spatial patterns (matrices of 1s and 0s), then calculating the RF size that maximizes the average MI per RF. Then, we use CNNs to categorize these patterns. We do this for many different RF sizes and record which RFs result in the most efficient learning. Our hypothesis is that the RF size that results in the quickest learning will be the RF size that was determined to maximize average MI per RF. This would give some initial evidence that determining RF size by maximizing average MI per RF would be beneficial to biological agents in the context of statistical learning.

All code for input generation, data, MI analyses, and CNN analyses can be found on Github at https://github.com/thpossidente/Int-Seg-Model.

Input Generation

In order to generate the inputs for this investigation, we created matrices with varying degrees of MI for different RF sizes. We used non-overlapping RFs exclusively, meaning the RF size and stride size were the same for input generation and analysis. All matrices used only zeros and ones. Figure 5 shows the basic process behind creating a set of matrices with high MI for a specific RF size. In this example we will create a simple set of 4*4 matrices with a RF size of 2*2 that have high MI per RF.



Step 1 - Create set of four random RF patterns

1	1
1	0

0	0
0	0

0	1
0	1

0	1
1	0

Step 2 - Randomly select one of four patterns for each of four positions in matrix - repeat

0	0	0	1
0	0	0	1
1	1	1	1
1	0	1	0

0	0	0	1
0	0	1	0
0	1	0	0
0	1	0	0

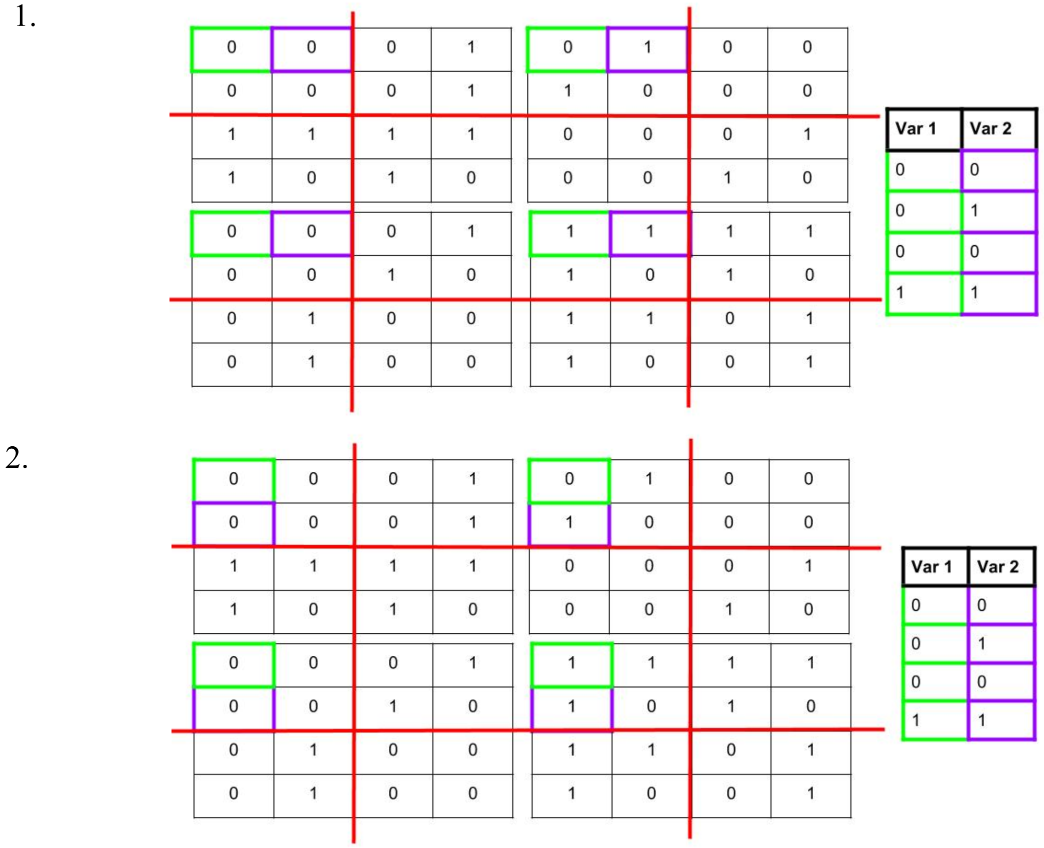
Figure 5 Shows the process of input generation. In step one, four unique random patterns of zeros and ones are arranged in a submatrix with dimensions specified by the desired RF size (2*2 in this case). These patterns will be the only ones used to construct the set of matrices. In step two, for each of the four 2*2 submatrices in the 4*4 matrix, one of the four RF patterns will be selected with replacement. Step two is repeated as many times as desired to create a set of matrices. In this case, 256 unique 4*4 matrices are possible, but only two are shown here.

Once 256 unique matrices were generated for a set, we simple duplicated the 256 matrices 20 times to acquire a set of 5120 matrices. This was so that we had 256 classes of matrices to classify using the CNNs. This relatively high number of classes was chosen because we did not want the classification task to be too easy, in which case we may have experienced a ceiling effect in CNN learning efficiency.

After the 5120 matrices were generated, the desired amount of noise was introduced to the set. Random noise was introduced by “flipping” some percentage of cells from zeros to ones and ones to zeros. Because noise is separately applied to each matrix, the MI per RF will decrease as the percentage of cells increases, until 50% of the cells are flipped. This is because flipping 50% - 100% of the pixels actually linearly decreases noise from 50% to 0% because there are only 2 possible values for each pixel. To generate a set of matrices with no (or very little) MI, matrices of a specified size are created and filled randomly with zeros and ones.

Calculating MI

In this study, average MI per RF was measured by breaking down each matrix into RF sized matrices and grouping these RF sized matrices based on their position in the larger matrix. Then pixel-wise MI was measured for each group of RF sized matrices separately, and the results were averaged. The setup for calculating the average MI per RF for a set of four 4*4 matrices with 2*2 RFs is shown in Figure 6.



1.

0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	0	1	0	0	0	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1

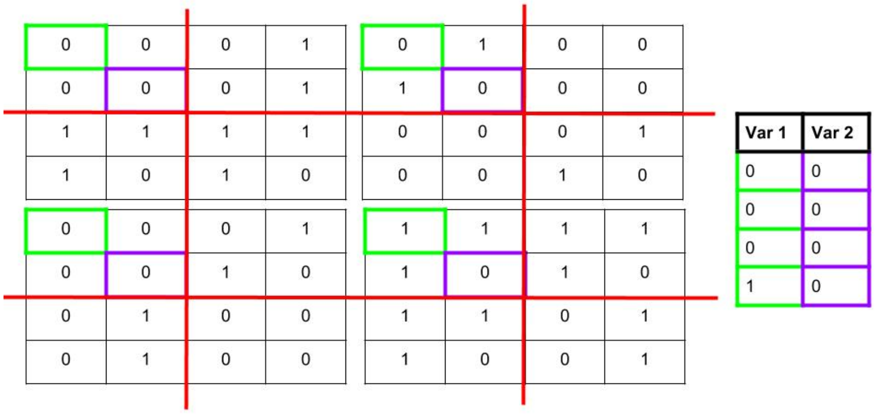
Var 1	Var 2
0	0
0	1
0	0
1	1

2.

0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	0	1	0	0	0	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1

Var 1	Var 2
0	0
0	1
0	0
1	1

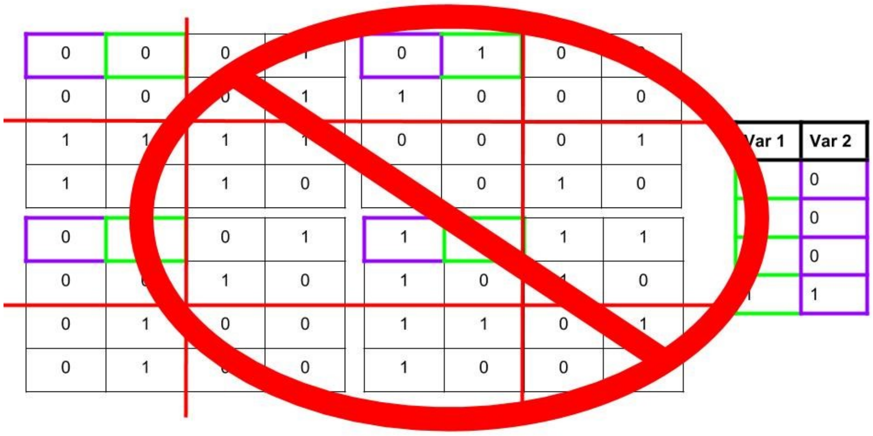
3.



0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	0	1	0	0	0	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1

Var 1	Var 2
0	0
0	0
0	0
1	0

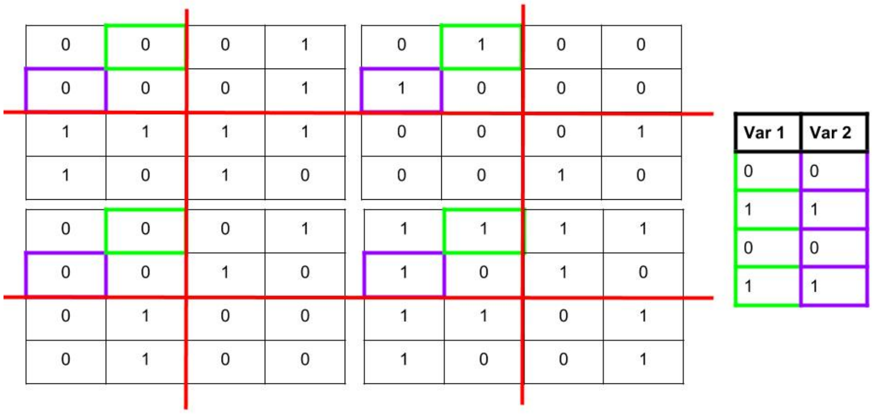
4.



0	0	0	1	0	1	0	[illegible]
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	[illegible]	1	0	[illegible]	0	1	0
0	[illegible]	0	1	1	[illegible]	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	[illegible]

Var 1	Var 2
[illegible]	0
[illegible]	0
[illegible]	0
1	1

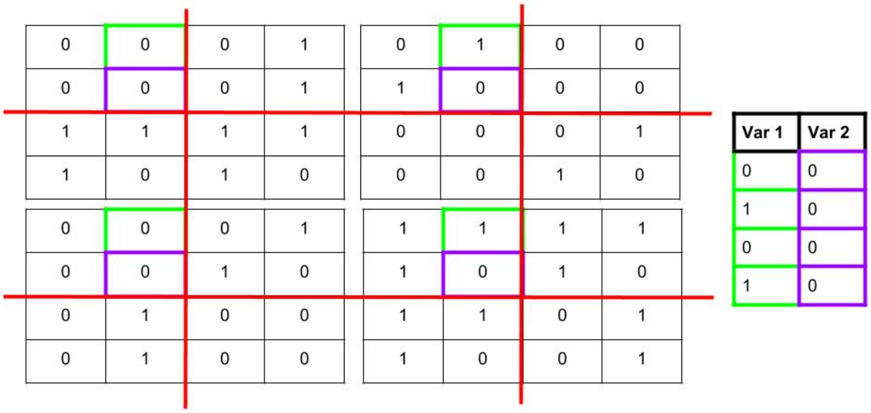
5.



0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	0	1	0	0	0	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1

Var 1	Var 2
0	0
1	1
0	0
1	1

6.



0	0	0	1	0	1	0	0
0	0	0	1	1	0	0	0
1	1	1	1	0	0	0	1
1	0	1	0	0	0	1	0
0	0	0	1	1	1	1	1
0	0	1	0	1	0	1	0
0	1	0	0	1	1	0	1
0	1	0	0	1	0	0	1

Var 1	Var 2
0	0
1	0
0	0
1	0

Figure 6 Shows a few iterations of the process of MI calculation for a set of four matrices. Each picture shows all four of the matrices, each divided into four RFs by red lines. The green and purple boxes highlight the cells whose values are being examined. The table on the right of each picture shows how the outlined cells are organized in a pairwise fashion for MI calculation. As you can see in pictures 1-3, pixel position (1,1) is compared to each other pixel position in the RF (except for its own position), Then in pictures 4-6, pixel position (1,2) is compared to each other pixel position in the RF (except for itself). This process will continue until every pixel position comparison within one RF has been made once. Then MI between all those pixel pairs are measured and the value is saved. The process then repeats for the next RF (the top right RF in the above diagrams). Once MI has been measured for each RF, the results are averaged to get a final average MI per RF. Note that we do not record the observations from picture 4 because the comparison between (1,1) and (1,2) has already been made and MI calculations are symmetric. This avoids double counting. Also note that using this process, the maximum MI of any RF size will be $\sum_{i=1}^{n}(n-i)$ where n is the number of cells in the RF. Thus, for the 2*2 RF example here, the maximum MI will be (4-1) + (4-2) + (4-3) + (4-4) = 6 bits.

MI is calculated between “Var 1” and “Var 2” in each table created from the process shown in Figure 6. This pixelwise MI is added together for all comparisons in a single RF, then the process starts over for the next RF. For the example shown in Figure 6, we would end up with MI sums for each of the four RFs. By averaging these four numbers we obtain the average MI per RF for the whole set of matrices.

Experiment 1 Methods - MI Calculations with Various RF Sizes

This experiment seeks to validate the hypothesis that RF sizes that match the structure of relevant information in the world (in this case, that match the input submatrix pattern size) will lead to an increase in MI per RF compared to other RF sizes. In this experiment, we created test pattern matrix sets with varying sizes of submatrix patterns (through the process shown in Figure 5) then calculated the average MI per RF for different RF sizes. We expected to find that MI would be greatest when calculating MI for a RF size that matched the submatrix pattern size. For example, if the test pattern matrix set was created by combining four 8*8 submatrices for each 16*16 matrix in the set, we would expect that average MI would be highest when MI per RF was based on a RF of size 8*8. This goes back to the argument that our sensory system should take advantage of the fact that information in the world that is relevant to us tends to be structured in a certain way, and thus there is a specific RF size that maximizes MI per RF. We hypothesize that this optimization would be useful to us because statistical learning mechanisms would benefit from increased predictability within each RF.

Fifteen pattern sets were generated with 16*16 pattern matrices. Submatrix pattern size possibilities were 2*2, 4*4, and 8*8 while possible noise levels were 0%, 10%, 20%, 30%, and 40% (3 submatrix size possibilities and 5 noise level possibilities = 15 sets). Noise was added by flipping X% of pixels in the matrix. Average MI per RF was calculated for each pattern matrix set using RFs of 2*2, 4*4, and 8*8. Additionally, a control set of 16*16 pattern matrices filled randomly with 0s and 1s was tested using the same RF sizes. We expect this set to have no (or very low) MI. Each pattern set in this experiment contained 5120 matrices. The same matrices created in this experiment were used for the second experiment.

Experiment 1 Results - MI Calculations on Various RF Sizes

The results of this analysis (shown in Figure 7) show that RF sizes that are the same as the submatrix pattern size used to create the pattern matrices in the set generally have higher MI per RF than most other RF sizes. There were 4 cases out of 15 that did not agree with our hypothesiss. Two of these cases were in the 4*4 submatrix pattern size condition and 2 of these cases were in the 8*8 submatrix pattern size condition. In each of the four nonconforming cases, lower RF sizes had greater percent MI per RF than the hypothesized RF size. Although some of the comparisons did not agree with our hypothesis, it is clear that when RF size matches pattern size it at least brings percent MI per RF levels up to the same level as the lower RF sizes, and in most cases exceeds them.

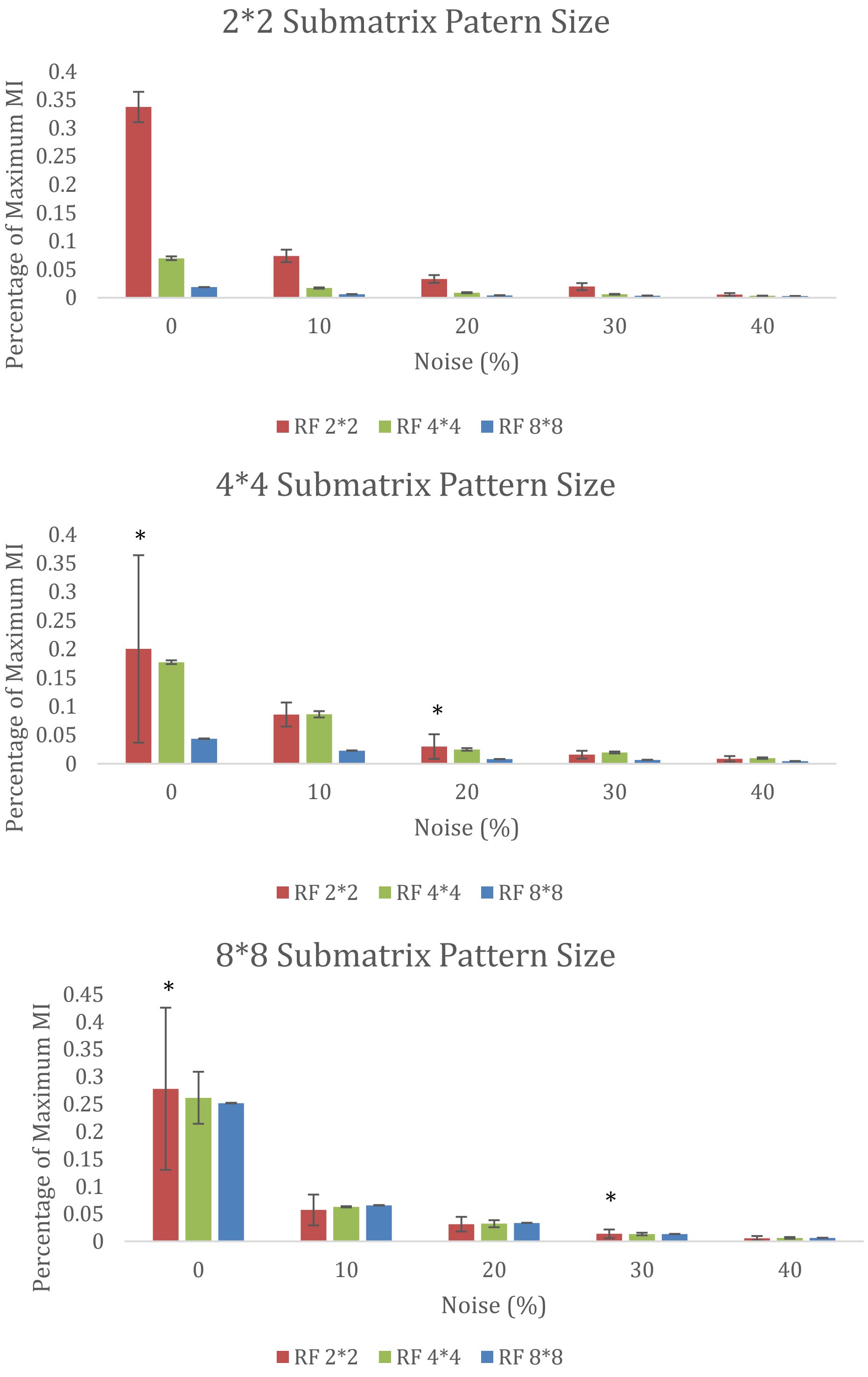


Figure 7 This figure shows the results for the 16*16 sets. In each graph, the average MI per RF for each RF size in each of the submatrix pattern size conditions for one noise level is depicted. The Y-axis shows the percentage of the maximum possible MI while the X-axis shows the size of the input pattern used to generate the input matrices. Each color bar represents a different RF size used to analyze MI. The asterisk indicates non-conforming comparisons.

For the control sets with randomly placed 1s and 0s, there were no instances in which MI was greater than any of the trials shown above. Ideally, it should be the case that there is zero MI per RF field in the random matrix set, but there is some MI that occurs by chance in sets with a relatively small number of pattern matrices. It was found that as the number of pattern matrices in the random set increased, MI approached zero (Figure 8).

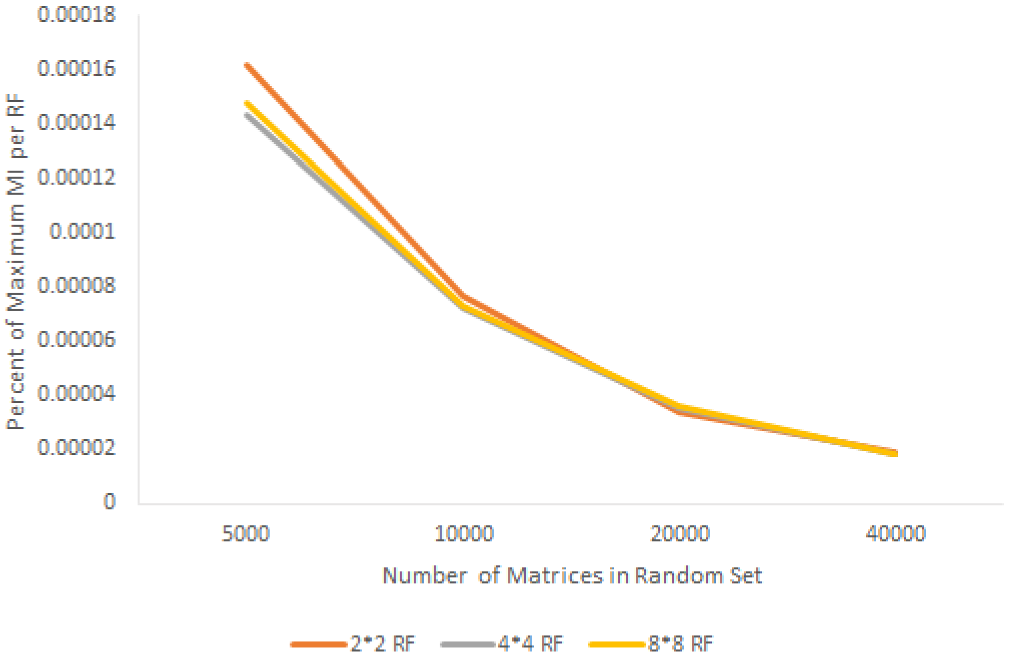


Figure 8 As the number of matrices in the set of random matrices increases, the percent of the maximum MI per RF for each RF size approaches zero.

Experiment 2 Methods – CNN Learning on Various RF Sizes

This experiment seeks to address the hypothesis that RF sizes that maximize MI are advantageous in terms of statistical learning. In this experiment we used the same sets of patterned matrices as in experiment 1 so that we would be able to directly compare MI per RF to CNN performance. Labels for the sets were created by simply taking the first 256 unique pattern matrices generated and labeling them 1-256, then repeating those matrices to create 5120 total pattern matrices. Noise was added only after labeling. Each of these sets was analyzed by a CNN with the following RF sizes: 2*2, 4*4, 8*8. For each RF size the sets were analyzed with 0%, 10%, 20%, 30%, 40%, and 50% noise. The metrics used to evaluate learning were loss at 5 epochs and loss at 25 epochs. Loss is the sum of the error for each sample calculated using categorical cross-entropy. Each combination of submatrix pattern size, RF size, and noise level was run 10 times. In summary, 3 sets of different submatrix pattern sizes were analyzed with 3 different RF sizes with 6 different noise levels.

The architecture of the CNN was very simple, consisting of one 16*16 2D convolutional layer with 4 filters (one for each possible submatrix pattern [see Figure 5]), one dense layer of 512 nodes, and one dense layer of 256 nodes (one for each label). The convolutional layer and the first dense layer used a Rectified Linear Unit (ReLU) activation function, while the last dense layer used a softmax activation function for classification. The stride of the convolutional layer was always the same as the RF size. The loss function used was categorical cross-entropy and the optimizer used as Adam with a learning rate of 0.001. Batch size was 100 and training was run for 100 epochs. 75% of the pattern matrices (3840) were used for training and the remaining (1160) were used for validation.

In order to quantitatively assess whether the pattern of CNN efficiency metrics matched up with the MI per RF calculations from experiment 1 we also made 2 Bayesian linear regression models. For the first, the dependent variable (DV) was loss at 5 epochs while the covariates were noise and percentage of maximum MI per RF. The second regression was identical except loss at 25 epochs was used as the DV instead of loss at 5 epochs. From these analyses we hoped to investigate how much of the variance in loss was explainable by percentage of maximum MI per

RF, and how much was explained simply by the noise level. We expected noise level explain a lot of the variation in loss, but percentage of maximum MI per RF should add additional explanatory power if increased MI per RF increases CNN learning efficiency.

Lastly, in order to get at the idea of metabolic cost for efficient information processing we also calculated an efficiency metric using average loss (over 10 runs) and number of trainable parameters in the network. To obtain this metric we simply multiply the number of parameters by the loss[13]. The lower the value of this efficiency metric, the lower the "metabolic" cost of the network.

Experiment 2 Results – CNN Learning on Various RF Sizes

A summary of the results of this experiment are shown in Figures 9 and 10 which show loss at epochs 5 and 25 (respectively) for each condition. In these graphs, the loss values shown are the average of the loss values from each of the 10 runs of the CNN for each specific condition. At 50% noise, accuracy was nearly equal to expected accuracy if classifying via random guessing which shows that, as expected, the 50% noise condition yielded no learning for any RF size in any submatrix pattern condition. This is because noise was maximized and there were no patterned regularities to distinguish one label from another. Because of this, the 50% noise condition was removed from Figures 9 and 10. In 23 of the 30 comparisons made, the RF size that resulted in the best performance (as measured by either loss at 5 or 25 losses) was the

[13] Although an initially intuitive metabolic cost metric would be loss per parameter, this metric would favor networks with high numbers of parameters, as they would have lower loss/parameter simply by virtue of having many parameters.

same RF size that resulted in maximum MI per RF in experiment 1. In Figures 9 and 10 asterisks mark the comparisons in which this was not true.

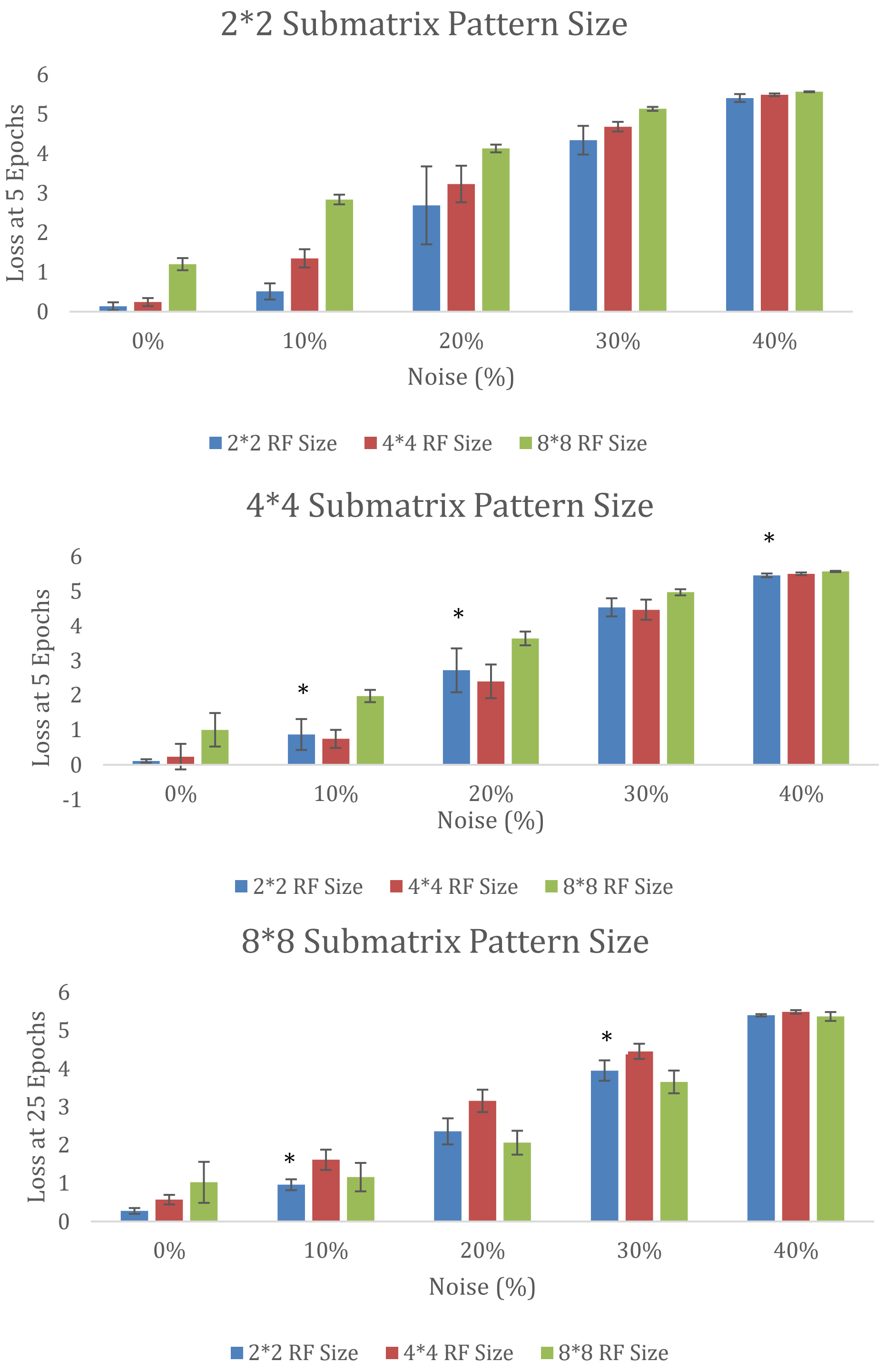


Figure 9 This figure shows the loss at 5 epochs (averaged over ten 25 epoch trials) for each condition tested. Each graph shows one of the three submatrix pattern size conditions. Noise level is on the X-axis, loss is on the Y-axis and the color of the bars represent the different RF sizes tested. The asterisk indicates non-conforming comparisons.

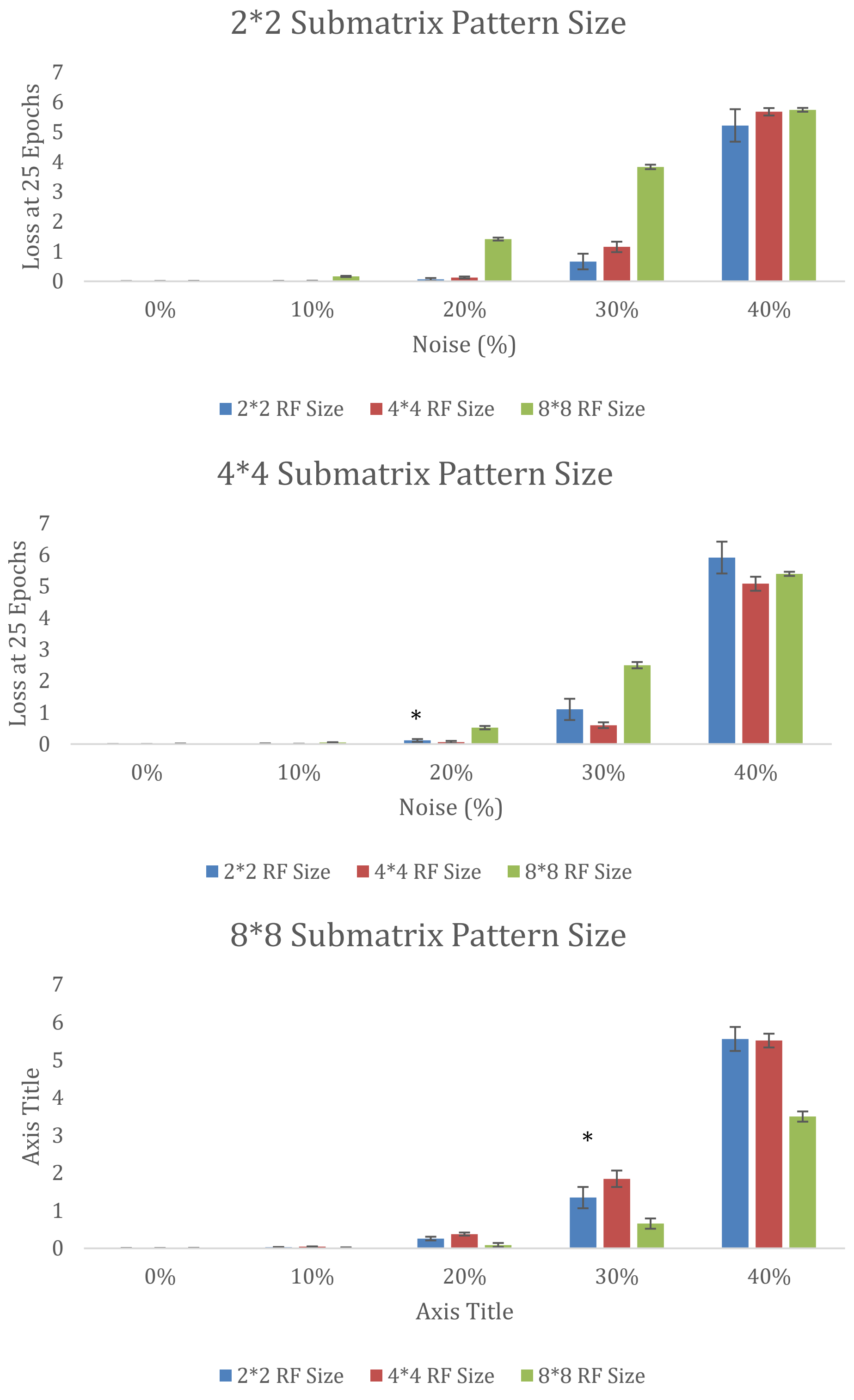


Figure 10 This figure shows the loss at 25epochs (averaged over ten 25 epoch trials) for each condition tested. Each graph shows one of the three submatrix pattern size conditions. Noise level is on the X-axis, loss is on the Y-axis, and the color of the bars represent the different RF sizes tested. The asterisk indicates non-conforming comparisons.

To see if these results matched the pattern of MI per RF for each condition, we created 2 Bayesian linear regression models. The loss from each of the 10 runs per condition were counted as separate data points for this analysis, yielding 540 data points for both loss at 5 epochs and loss at 25 epochs. For both models, noise + percentage of maximum MI per RF explained the most variance in loss. With loss at 5 as the DV, the data are $5.966*10^{7}$ times more likely under the noise + percentage of maximum MI per RF model than under the next best model (noise only) and $4.811*10^{271}$ times more likely than under the null model (Table 1). With loss at 25 epochs as the DV, the data are $1.319*10^{22}$ more likely under the noise + percentage of maximum MI per RF model than the next best model (noise only) and $6.746*10^{203}$ times more likely than the null model (Table 2). This supports our hypothesis that high MI per RF results in more efficient CNN learning.

Loss at 5 Epochs Model Comparison

Models	P(M)	P(M|data)	BF_M	BF_{01}	R^2
Percent of Maximum MI per RF + Noise	0.250	1.000	1.790e +8	1.000	0.906
Noise	0.250	1.676e -8	5.028e -8	5.966e +7	0.898
Percent of Maximum MI per RF	0.250	1.186e -198	3.559e -198	8.430e +197	0.476
Null model	0.250	2.079e -272	6.236e -272	4.811e +271	0.000

Table 1 This table compares each model to the best model (Percent of Maximum MI per RF + Noise). P(M) is our prior odds for each model being the best predictor of loss at 5 epochs. P(M|data) is the posterior odds of each model given the data. BF_M quantifies the change from prior odds to posterior odds. BF_{01} quantifies the likelihood of the data occurring under the best model divided by the likelihood of the data occurring under a given model. In other words, it is how many times more likely the best model is given the data than a given model. R^2 is the amount of variability in loss at 5 epochs explainable by the model.

Loss at 25 Epochs Model Comparison

Models	P(M)	P(M|data)	BF_{M}	BF_{01}	R^2
Percent of Maximum MI per RF + Noise	0.250	1.000	∞	1.000	0.831
Noise	0.250	7.580e -23	2.274e -22	1.319e +22	0.793
Percent of Maximum MI per RF	0.250	3.855e -181	1.156e -180	2.594e +180	0.192
Null model	0.250	1.482e -204	4.447e -204	6.746e +203	0.000

Table 2 This table shows the same thing as Table 1, but with loss at 25 epochs as the DV.

To better visualize these results, we created 2 scatterplots of the data used in the above analyses. These scatterplots show that as percentage of maximum MI per RF increases, loss at both 5 and 25 epochs generally decreases (indicating greater CNN efficiency) for each given noise level (Figure 11). From the plot showing the data for loss at 25 epochs it is clear that there was somewhat of a floor effect, as many models reached very low loss levels by 25 epochs.

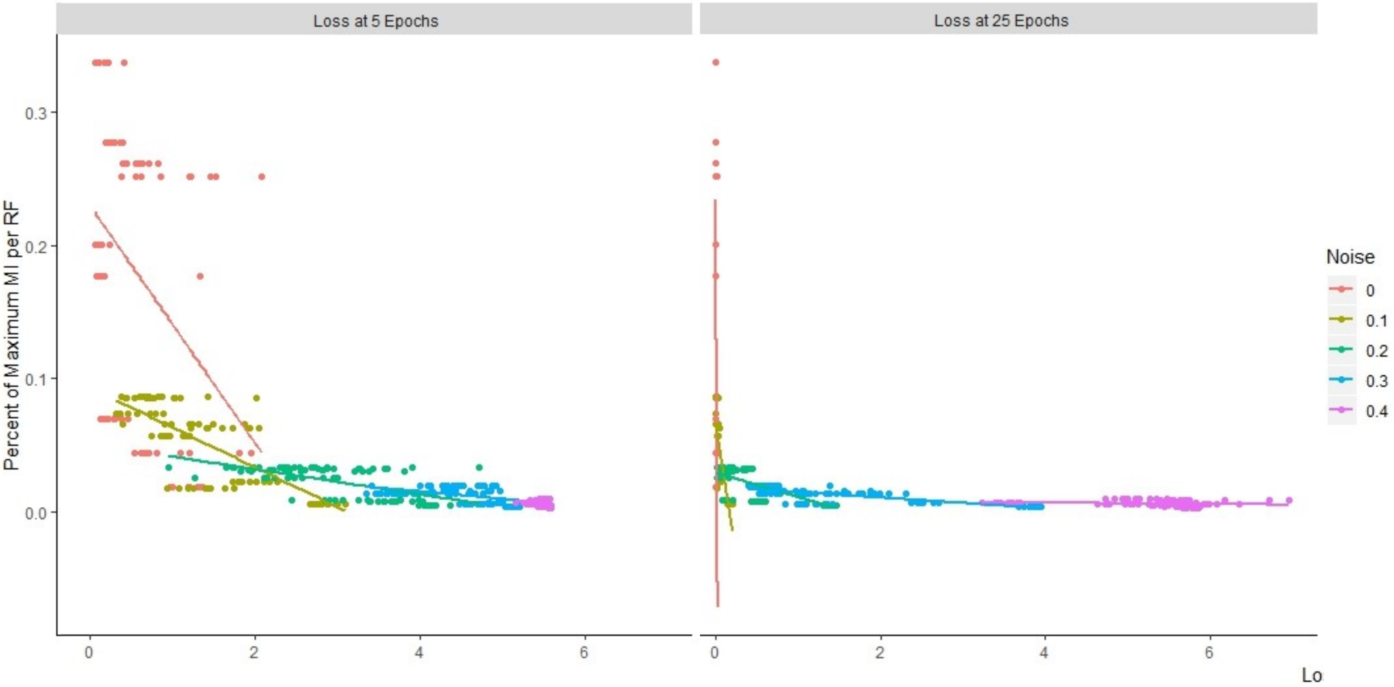


Figure 11 This figure shows percent of maximum MI per RF plotted against loss for loss at 25 epochs (left) and 5 epochs (right) for each noise condition.

Lastly, Figures 12 and 13 show the metabolic efficiency metric (loss multiplied by the number of trainable parameters in the network) for loss at 5 and 25 epochs respectively. For all networks using an RF size of 2*2 there were 292,032 trainable parameters, 4*4 RFs yielded 164,676 parameters, and 8*8 RFs resulted in 131,328 parameters. Clearly, the best performing networks are not always the most "metabolically" efficient, and smaller numbers of parameters tend to be more "metabolically" efficient at higher noise levels, perhaps because loss tends to be more similar across RF sizes in those conditions.

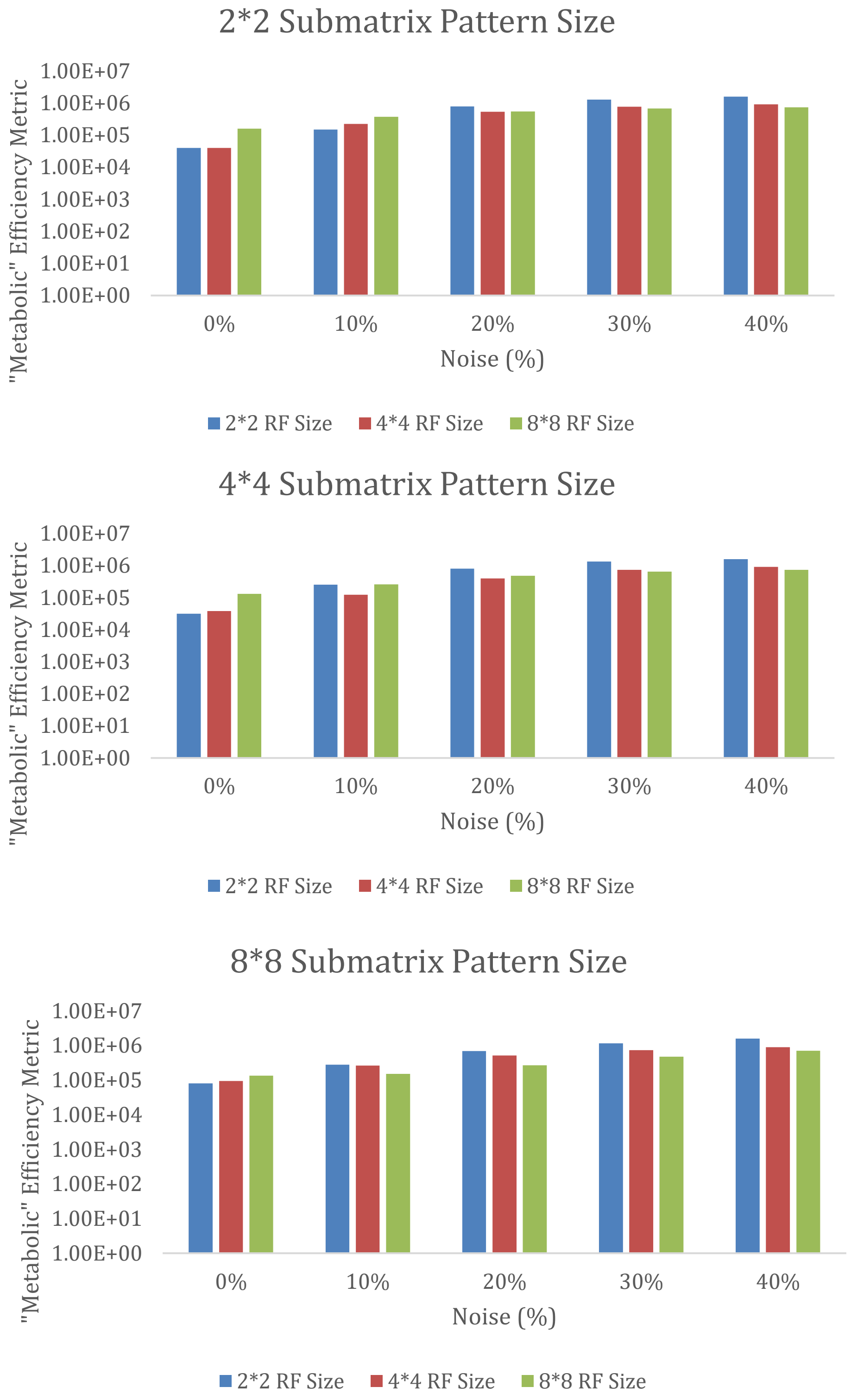


Figure 12 The graphs above show the “metabolic” efficiency metric at 5 epochs for each condition tested. Note that the Y axis is logarithmic (base 10) so that smaller values have visible bars.

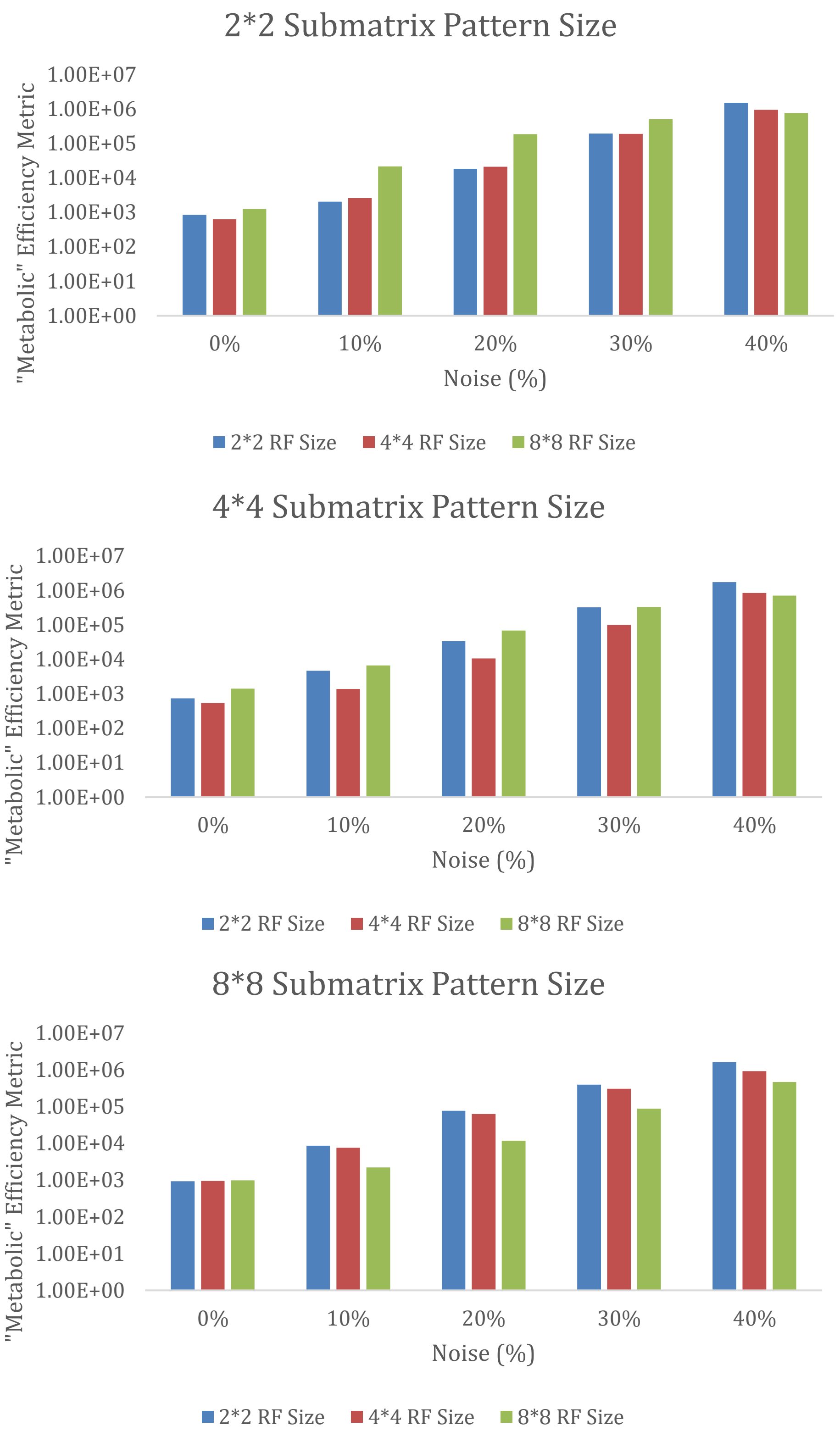


Figure 13 The graphs above show the "metabolic" efficiency metric at 25 epochs for each condition tested. Note that the Y axis is logarithmic (base 10) so that smaller values have visible bars.

Discussion

In Experiment 1 it was found that for RF sizes that were equal to the submatrix pattern sizes used to create the pattern matrix sets, MI per RF was generally highest. In the four non-conforming trials it is possible that because only 4 submatrix patterns are generated for each set, the patterns generated at higher submatrix pattern sizes happened to contain high MI at lower submatrix pattern sizes as well. Overall, the results from this experiment show that we are able to vary the RF size that maximizes MI through this technique of matrix pattern set generation. Additionally, as expected, the MI for the control (random) sets approached zero as the number of pattern matrices in the set was increased. This gives evidence that our implementation of calculating MI is reliable.

Now we turn to Experiment 2, whose aim is to show that higher MI per RF is indeed desirable in terms of statistical learning. In most cases, the RF size that had the highest percent of maximum MI per RF also resulted in the lowest loss when used as the RF for CNN classification. Additionally, the Bayesian linear regression models showed that at both 5 epochs and 25 epochs, including percent of maximum MI per RF added significantly to the model's ability to explain the variation in loss. Taken together, these results give evidence that in general, an RF size that results in higher MI per RF will increase the efficiency of learning in terms of loss in our CNNs. Because both CNNs and biological neural circuits operate according to the constraints of Information Theory and both engage in statistical learning, this validates the possibility that biological RF sizes could be partially evolved to maximize MI per RF in order to aid in information processing in the form of statistical learning.

Here it is important to discuss how the number of trainable parameters in each model may affect the results obtained. As the RF size increases, the number of trainable parameters in the CNN decreases. This means that CNNs with smaller RF sizes may potentially take longer to reach their minimum loss but may be able to reach a lower minimum. Conversely, CNNs with larger RF sizes may potentially take a shorter time to reach their minimum loss but may not reach as low of a loss. But this is not what we saw in our results, seeing as there were many cases in which CNNs with large RFs outperformed CNNs with small RFs in the long term (25 epochs) and cases in which CNNs with small RFs outperformed CNNs with large RFs in the short term (5 epochs). This led us to believe that any potential effects caused by the differential number of parameters in the CNNs were overshadowed by the effect of the MI per RF. It's important to note that in larger CNNs with many millions of parameters, the difference in parameters between small RFs and large RFs would be greater, leading to the possibility of larger effects on performance.

In experiment 2 we tried to take into account number of parameters by creating a metric of "metabolic" efficiency. As one might expect, when CNNs had relatively small differences in loss, the "metabolic" efficiency metric favored larger RFs (with lower numbers of parameters). But in cases where CNNs with low RFs (and thus more parameters) had lower loss than the other conditions by a large margin, they were still more "metabolically" efficient than the other low-parameter conditions. And because we have shown that in general loss varies inversely with percentage of maximum MI per RF, this suggests that when two or more RFs have sufficiently similar percentage of maximum MI per RF, and we care about the "metabolic" efficiency of the system, we should select the larger RF. This idea fits well with the Metabolic Efficiency Principle suggested by Stone (2018).

Another important point to consider is that the scale of information that is relevant changes for each organism and each sensory system. For example, the scale at which touch information is useful to a ladybug is likely different than the scale of touch information useful to a human. And the scale of touch information useful to a human may be different than the scale of auditory information useful for a human. In order to simplify this complex problem in this experiment we have assumed that the scale of information with the most MI is relevant to the "organism", but this may not always be the case. For example, as mentioned previously, it could be that the highest amount of MI possible occurs at a visual RF size that is microscopic. This RF size would not be very useful for many organisms and would be metabolically inefficient to implement due to the large number of photoreceptors that would be necessary[14]. So, we contend that within the scale of information that is 1) relevant to each organism and sensory system, 2) within the metabolic constraints of the organism, and 3) detectable by the organism's sensors, RF size should be tuned in order to maximize MI per RF in order to increase learning efficiency. This idea is similar to finding a local minimum in a certain range of values. The range of values is determined by what spatial scale of information is useful to the organism, what scale metabolic constraints will allow, and what scale the organism's sensors can detect, while the local minimum is the RF size that results in the highest MI per RF within this restricted range of values.

[14] How the scope of information relevant to the organism is determined is beyond the scope of this study, but presumably over time selection pressures push the organism's sensory systems to look for information at a scale that is most useful for that organism's survival and reproduction.

Conclusion

These results give evidence that CNNs with a RF size that results maximal MI per RF learn more effectively than CNNs with RFs of other sizes. In light of these results, we contend that because both CNNs (and ANNs more generally) and neural circuits in the brain perform statistical learning and are subject to the constraints imposed by information theory, RF sizes that increase MI in their inputs could also be advantageous for neural circuits in the brain that perform statistical learning. This keeps open the possibility that RFs in biological organisms may have evolved to be the size that they are in part to increase MI per RF and thus match the scale at which relevant information is structured in their world.

A parallel application of these results is to the training of CNNs in general. Currently, the most common method for determining RF size for a CNN is to perform hyperparameter tuning. This can be a slow and laborious method. As of now, we know of no commonly used systematic and theoretically grounded method for selecting a RF size in CNNs. Using the method of MI calculation described here, one could potentially select a RF size that maximizes MI per RF and therefore increases learning efficiency in CNNs.

Further Research

A Different Method of Calculating MI per RF

In this study, average MI per RF was measured by breaking down each matrix into RF-sized matrices, then grouping these RF-sized matrices based on their position in the larger matrix. Next, pixel-wise MI was measured for each group of RF-sized matrices separately, and the results were averaged. An alternative to this approach would be to break down each matrix into RF-sized matrices, not group them by position in the larger matrix, then calculate pixel-wise

MI separately for each unique pixel-pair combination. There are two advantages to this approach. One is that it would be blind to the position of the RF-sized matrix, and therefore spatial patterns across matrices in different positions would contribute to the MI total instead of these patterns having to occur in the same position for there to be MI as is the case in this study. Secondly, it would preserve the spatial information present in different pixel-pair combinations within each RF. For these reasons we hypothesize that this way of measuring MI would result in the ability to capture more of the MI present in the set than the present method. This method of measuring MI per RF may or may not yield different results than those presented here, and therefore it is crucial to attempt to replicate these findings using this alternative method of measurement.

Comparing MI-Optimal RF Size in Binarized Datasets

A natural next step in this investigation would be to move on from pattern matrix sets and use existing binarized image datasets such as the CalTech 101 Silhouettes Data Set. The idea here would be to find two or more binarized image datasets and perform MI analysis to find out what RF size results in maximization of MI per RF, then see if that size RF results in the best CNN performance compared to other RF sizes. Ideally, each dataset would have a different MI-optimal RF size so we could test this for many different MI-optimal RF sizes. This extension would seek to strengthen the results found here by using datasets that were not created for the purpose of having high or low MI at certain spatial scales.

Using MI Calculations to Estimate an Optimal RF Size for Training CNNs

As previously mentioned, a larger scale investigation of the feasibility of using MI calculations to choose an effective RF size for training CNNs could be fruitful. Although these MI calculations would be computationally intensive for large datasets of large RGB images (as opposed to the small datasets of small binary matrices used here), it may still be more efficient than trial-and-error methods of selecting an optimal RF size that are commonly employed today. Additionally, these computations may be sped up by estimating MI via sampling rather than an exhaustive computation approach. One could imagine an algorithm that samples pairs of pixels from an image set until a stable MI estimation is obtained for many different RF sizes. But still, in order to calculate MI from RGB images (whose pixel values are 1-256) one would need to use the continuous version of the MI formula which is more computationally expensive and more difficult to implement. So, it could be the case that even MI estimation by sampling is too computationally intensive for regular use in big data analysis, thus further investigation of this method is necessary. However, if successful even in some cases for some types of data, this method offers the possibility of a more systematic and theoretically grounded approach to determining RF size for CNNs in general.

Analyzing Natural Image Datasets to Estimate MI-optimal Biological RF Size

Another potential avenue to explore would be to perform the same methods used here (MI calculations and ANN analysis) on a large natural image dataset. This would accomplish two things. It would 1) attempt to replicate the findings here – namely that statistical learning occurs more efficiently if the RF size being analyzed maximize MI per RF. It would also 2) attempt to extend the findings here by investigating whether the optimal RF size for natural images is similar to the RF size found in retinal ganglion cells in the human visual system. We hypothesize that because both the brain and CNNs perform statistical learning and are subject to the constraints of information theory, the same size RF should be optimal (in terms of maximal MI) for the same inputs (or an approximation of the same inputs, i.e. a natural image dataset). Using the RF size of the retinal ganglion cell as a comparison makes the most sense here because these cells carry the inputs into the main visual information processing areas of the brain (although there is some processing that occurs before the photoreceptor outputs reach the ganglion cells). If successful this experiment could give evidence that 1) CNNs are a good model for information processing and statistical learning in the brain, and 2) the human visual system's RF size is influenced by the optimization of MI in each RF in order to increase efficiency in statistical learning.

One problem here could be the fact that RF size in humans is likely influenced by metabolic constraints and constraints imposed by what scale of information is relevant for the organism (as discussed previously). These same constraints do not apply to CNNs and therefore, the resulting optimal RF size could differ significantly from human retinal ganglion cells even if RF size is optimized based on MI in both systems. One potential solution would be to try to simulate these metabolic and relevance constraints in the CNNs, but penalties for metabolically inefficient numbers of parameters and relevance constraints would be difficult to determine in a theoretically motivated way. Thus, more research and careful consideration is required before this experiment could be carried out.

Unsupervised ANNs as a More Biologically Plausible Alternative to CNNs

Approaching the same problem with unsupervised ANNs instead of CNNs could make the connection between ANN models and human neural circuitry stronger. This is because unsupervised ANNs have much more in common with the mechanisms of statistical learning found in the brain. Besides exhibiting statistical learning and operating according to information theoretic principles, unsupervised ANNs could use winner-takes-all activation and Hebbian updating which is a key principle in statistical learning in the brain.

Hebb's postulate states that if neuron A continually takes part in causing neuron B to fire, the strength of connection between the two neurons will increase over time (Hebb, 1949) (i.e. nerves that fire together wire together). This idea has been expanded into a model of how associative learning occurs in mammals via long-term potentiation (LTP) and long-term depression (LTD), processes that rely on Hebbian updating (Jaffe & Johnston, 1990; Debanne, Gähwiler, & Thompson, 1994; Tsien, 2000; Shimizu, Tang, Rampon, & Tsien, 2000). Hebbian learning via LTP and LDP are still widely accepted as a strong model for learning in mammals (Maffei, 2018). Additionally, it is presumed that Hebbian updating (in the form of LTP and LTD) is a key mechanism driving statistical learning (Munakata & Pfaffly, 2004). The simple updating rule behind Hebb's postulate could be realized in unsupervised ANNs by increasing the connection weight between nodes A and B by the activation value of A minus the connection weight between A and B multiplied by a learning rate parameter.

Additionally, it would be plausible to implement a winner-take-all activation scheme in unsupervised ANNs. This means that, as in biological neurons, a node would either be activated or not with no intermediate activation possibilities. Lastly, ANNs that make use of statistical learning through Hebb's rule have been found to yield topographical maps of activation where neighboring nodes learn to activate in concert to similar stimuli (Konen, Maurer & Von Der Malsburg, 1994), a feature characteristic of the brain. Thus, unsupervised ANNs using Hebbian updating could to be a better model for statistical learning, a process that seems to occur abundantly across multiple domains in the brain.

It should be mentioned that Hebbian learning does not account for all of associative learning. For example, when learning to associate two events/stimuli that are significantly temporally divorced, learning likely occurs through a different mechanism, possibly one involving some sort of supervisory signal (Suvrathan, 2019). But nevertheless, it is abundantly clear that Hebbian learning is the major player in associative learning. Additionally, we do not mean to indicate that ANNs using Hebbian updating are equivalent to neural circuits in the brain. Neural circuits in the brain are much more complex and are modulated not only by different neurotransmitters, but also by glial cells and top-down regulation from other circuits. But a model is not an exact reproduction and needs only to possess features that are believed to be critical to the phenomena under investigation. The features relevant to learning about co-occurrences in stimuli are believed to be statistical learning and Hebbian updating, both of which can be convincingly modeled in an ANN using Hebbian updating.

A more detailed potential implementation of these Unsupervised ANNs is contained in the Appendix.

References

Abla, D., Katahira, K., & Okanoya, K. (2008). On-line assessment of statistical learning by event-related potentials. Journal of Cognitive Neuroscience, 20(6), 952-964.

Amano, K., Wandell, B. A., & Dumoulin, S. O. (2009). Visual field maps, population receptive field sizes, and visual field coverage in the human MT+ complex. *Journal of neurophysiology*.

Bell, A. J., & Sejnowski, T. J. (1995). An information-maximization approach to blind separation and blind deconvolution. Neural computation, 7(6), 1129-1159.

Bichsel, M., & Seitz, P. (1989). Minimum class entropy: A maximum information approach to layered networks. Neural Networks, 2(2), 133-141.

Bishop, C. M. (1999). Pattern recognition and feed-forward networks. In The MIT encyclopedia of the cognitive sciences (Vol. 13, No. 2). MIT Press.

Cunillera, T., Toro, J. M., Sebastián-Gallés, N., & Rodríguez-Fornells, A. (2006). The effects of stress and statistical cues on continuous speech segmentation: an event-related brain potential study. Brain research, 1123(1), 168-178.

Convolutional Neural Networks (CNNs / ConvNets). (n.d.). Retrieved May 9, 2019, from http://cs231n.github.io/convolutional-networks/#conv

Debanne, D., Gähwiler, B. H., & Thompson, S. M. (1994). Asynchronous pre-and postsynaptic activity induces associative long-term depression in area CA1 of the rat hippocampus in vitro. Proceedings of the National Academy of Sciences, 91(3), 1148-1152.

Fan, J. (2014). An information theory account of cognitive control. Frontiers in human neuroscience, 8, 680.

Ghahramani, Z. (2015). Probabilistic machine learning and artificial intelligence. Nature, 521(7553), 452.

Goodale, M. A., & Milner, A. D. (1992). Separate visual pathways for perception and action. Trends in neurosciences, 15(1), 20-25.

Hale, J. (2003). The information conveyed by words in sentences. Journal of Psycholinguistic Research, 32(2), 101-123.

Hebb, Donald O. "The organization of behavior: A neurophysiological approach." (1949).

Jacobs, R. A., & Kosslyn, S. M. (1994). Encoding shape and spatial relations: The role of receptive field size in coordinating complementary representations. Cognitive science, 18(3), 361-386.

Jaffe, D., & Johnston, D. (1990). Induction of long-term potentiation at hippocampal mossy-fiber synapses follows a Hebbian rule. Journal of Neurophysiology, 64(3), 948-960.

Kass, R. E., & Raftery, A. E. (1995). Bayes factors. Journal of the american statistical association, 90(430), 773-795.

Khaligh-Razavi, S. M., & Kriegeskorte, N. (2014). Deep supervised, but not unsupervised, models may explain IT cortical representation. PLoS computational biology, 10(11), e1003915.

Konen, W. K., Maurer, T., & Von Der Malsburg, C. (1994). A fast dynamic link matching algorithm for invariant pattern recognition. Neural networks, 7(6-7), 1019-1030.

Kriegeskorte, N. (2015). Deep neural networks: a new framework for modeling biological vision and brain information processing. Annual review of vision science, 1, 417-446.

Liu, Y. H. (2018). Feature Extraction and Image Recognition with Convolutional Neural Networks. In Journal of Physics: Conference Series (Vol. 1087, No. 6, p. 062032). IOP Publishing.

Maffei, A. (2018). Long-Term Potentiation and Long-Term Depression. In Oxford Research

Encyclopedia of Neuroscience.

Martignon L. (2001), Information Theory. In N.J. Smelser & P.B. Baltes, *International Encyclopedia of the Social & Behavioral Sciences*. Pergamon

McNealy, K., Mazziotta, J. C., & Dapretto, M. (2006). Cracking the language code: neural mechanisms underlying speech parsing. Journal of Neuroscience, 26(29), 7629-7639.

Munakata, Y., & Pfaffly, J. (2004). Hebbian learning and development. Developmental Science, 7(2), 141-148.

Rao, A. (2018, November 27). Convolutional Neural Network (CNN) Tutorial In Python Using TensorFlow. Retrieved from https://www.edureka.co/blog/convolutional-neural-network/

Rawat, W., & Wang, Z. (2017). Deep convolutional neural networks for image classification: A comprehensive review. Neural computation, 29(9), 2352-2449.

Roser, M. E., Fiser, J., Aslin, R. N., & Gazzaniga, M. S. (2011). Right hemisphere dominance in visual statistical learning. Journal of cognitive neuroscience, 23(5), 1088-1099.

Rueckl, J. G., Cave, K. R., & Kosslyn, S. M. (1989). Why are “what” and “where” processed by separate cortical visual systems? A computational investigation. Journal of cognitive neuroscience, 1(2), 171-186.

Santos, J. M., Alexandre, L. A., & de Sá, J. M. (2004, December). The error entropy minimization algorithm for neural network classification. In int. conf. on recent advances in soft computing (pp. 92-97).

Shannon, C. E. (1948). A mathematical theory of communication. Bell system technical journal, 27(3), 379-423.

Shannon, C. E. (1949). Communication theory of secrecy systems. Bell system technical journal, 28(4), 656-715.

Sharma, Avanish (2017, March 30). Understanding Activation Functions in Neural Networks. Retrieved from https://medium.com/the-theory-of-everything/understanding-activation-functions-in-neural-networks-9491262884e0

Shimizu, E., Tang, Y. P., Rampon, C., & Tsien, J. Z. (2000). NMDA receptor-dependent synaptic reinforcement as a crucial process for memory consolidation. Science, 290(5494), 1170-1174.

Silva, L. M., de Sá, J. M., & Alexandre, L. A. (2005, April). Neural network classification using Shannon's entropy. In ESANN (pp. 217-222).

Smith, A. T., Singh, K. D., Williams, A. L., & Greenlee, M. W. (2001). Estimating receptive field size from fMRI data in human striate and extrastriate visual cortex. *Cerebral cortex*, *11*(12), 1182-1190.

Stone, J. V. (2018). Principles of neural information theory: Computational neuroscience and metabolic efficiency. Place of publication not identified: Sebtel Press.

Suvrathan, A. (2019). Beyond STDP—towards diverse and functionally relevant plasticity rules. Current opinion in neurobiology, 54, 12-19.

Tononi, G. (2017). The Integrated Information Theory of Consciousness: An Outline. The Blackwell Companion to Consciousness.

Tsien, J. Z. (2000). Linking Hebb's coincidence-detection to memory formation. Current opinion in neurobiology, 10(2), 266-273.

Turk-Browne, N. B., Scholl, B. J., Chun, M. M., & Johnson, M. K. (2009). Neural evidence of statistical learning: Efficient detection of visual regularities without awareness. Journal of cognitive neuroscience, 21(10), 1934-1945.

Ungerleider, L. G., & Haxby, J. V. (1994). 'What' and 'where' in the human brain. Current opinion in neurobiology, 4(2), 157-165.

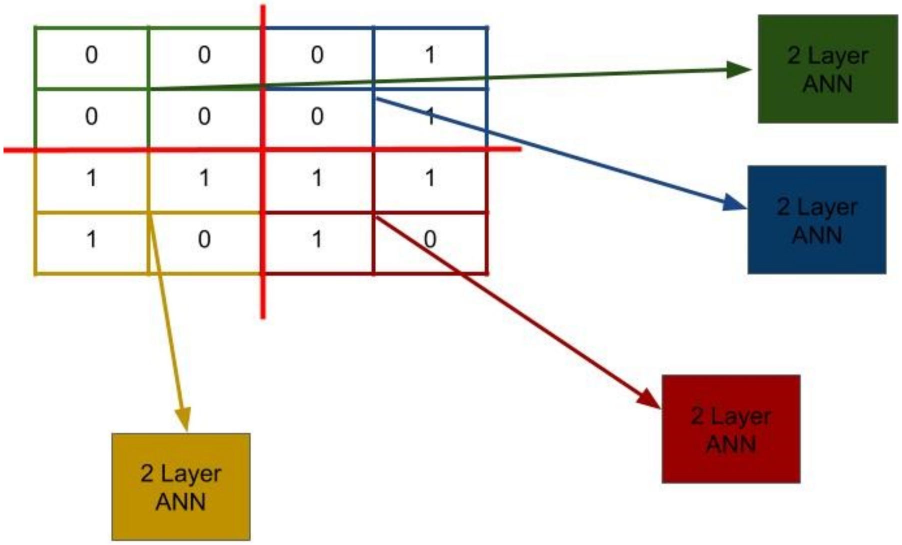
Yamins, D. L., Hong, H., Cadieu, C. F., Solomon, E. A., Seibert, D., & DiCarlo, J. J. (2014). Performance-optimized hierarchical models predict neural responses in higher visual cortex. Proceedings of the National Academy of Sciences, 111(23), 8619-8624.

Ziv, J., & Lempel, A. (1977). A universal algorithm for sequential data compression. IEEE Transactions on information theory, 23(3), 337-343.

Appendix

Potential Unsupervised ANN Implementation

There will be one ANN dedicated to processing each RF sized section of the input matrix (Figure 14). The inputs to each ANN would be vectors of 1s and 0s contained in the RF field that that ANN is dedicated to. Each of these ANNs will then output to a single output layer (Figure 15).



0	0	0	1
0	0	0	1
1	1	1	1
1	0	1	0

Figure 14 Shows how each RF contains the inputs to one unsupervised ANN. The green cells represent one 2*2 RF whose values are fed into the green ANN. The same applies to each of the other colors.

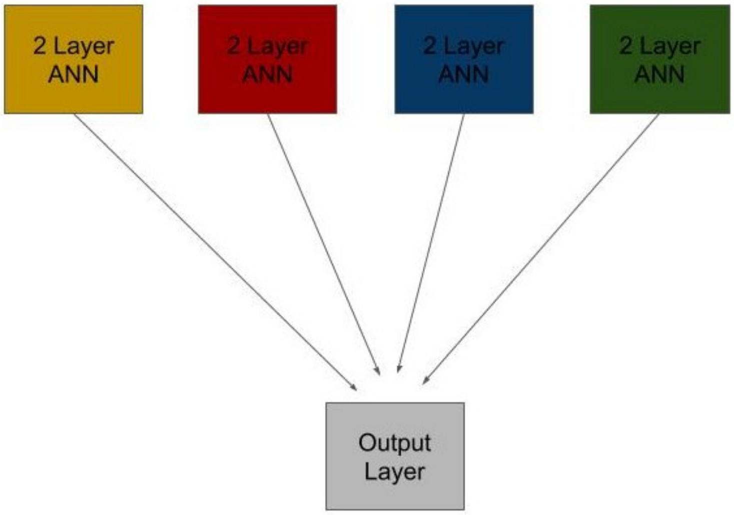


Figure 15 Shows how each ANN outputs to a single output layer.

The number of nodes in the input layer of each RF's ANN would vary according to the number of elements in the RF. For example, the input layer for an 8*8 RF would have 64 nodes. The number of nodes in the second layer of each RF's ANN would have 4 nodes, one for each possible input pattern (all sets of input matrices would consist of blocks of 4 unique patterns -- see Figure 5) The number of nodes in the final output layer would vary according to how many possible combinations of the four RF patterns there are. For example, for a set containing 8*8 matrices, each with four 4*4 RFs, with each RF having 4 possible input patterns, there are (4*4*4*4) 256 possible combinations of input patterns (assuming submatrix patterns can be used more than once in the same pattern matrix). If the system learns perfectly, each of the 256 output nodes would correspond to the presentation of a unique matrix input pattern.

The activation calculations for these ANNs would be calculated simply by multiplying incoming inputs by the corresponding connection weight for each input to a single node, then summing all these values to obtain the node's activation value. All activation values in a layer would then be compared and the highest value would be set to an activation of 1 while all the rest would be set to 0, thus implementing winner-takes-all activation.

Weight adjustment for these ANNs could be implemented using simple Hebbian updating:

$$W_{(i,j)} = W_{(i,j)} + LR * (x - W_{(i,j)})$$

where $W_{(i,j)}$ is the connection weight from node i to node j, LR is the learning rate parameter, and x is the input from node i to node j. In essence, the weight from node i to j is increased when there are high activation values being passed from node i to j. Hebbian updating can result in "run-away weights", meaning weights that iteratively increase to extremely high values over time. To avoid this problem, when calculating the update, the weight's current value is subtracted from the input activation. This would guarantee that the weight does not increase past the value of 1 (the maximum value of x is 1 due to winner-take-all activation).