IST Extended Abstract - Information Systems and Computer Engineering

Attention Network: steep learning curve in an invariant pattern recognition model

Luís Sá Couto

March 2018

Abstract Hubel and Wiesel's findings about the visual cortex inspired deep models for invariant pattern recognition. However, to achieve high performance, unlike the brain, such models require large training sets of labeled data. Such characteristic may be caused by the trade-off between invariance and discriminatory power that the current implementation of the complex cell by subsampling introduces. Therefore, we suggest a new view for the operation of complex cells which is based on quantization of a polar coordinates space. By introducing this view, we gain invariance capabilities, while minimizing the loss of discriminatory power. Furthermore, biological work from the same author suggests that the brain may learn in two distinct phases. Some of the aforementioned deep models follow this principle, but employ typical data hungry machine learning classifiers to implement the second phase. For that reason, we suggest the addition of a biologically inspired second-phase to complete the architecture. Such phase is based on searchlight selective attention that compares sets instead of vectors. The two-phased model is then able to learn from fewer representatives of each category without any optimization. This seems to be in accordance with the way humans learn visual patterns, since we do not need so many examples. The resulting model is tested with an invariant pattern recognition task in the MNIST and ETL-1 datasets. We verify that the model is able to achieve better accuracies with less training examples. More specifically, on the MNIST test set, the model achieves a 100% accuracy when trained with little more than 10% of the training set.

Department of Computer Science and Engineering, Instituto Superior Técnico
E-mail: luis.sa.couto@tecnico.ulisboa.pt
Supervisor: Professor Andreas Wichert

Fig. 1 Three variants of the same category.

Keywords Hubel Wiesel's Hypothesis · Selective Attention · Invariant Pattern Recognition · Deep Learning · Steep Learning

1 Introduction

To correctly recognize visual patterns, a model must be able to tolerate transformations in, for instance, shifts and sizes. Therefore, it is often said that a model needs to possess invariance capabilities to such transformations. Let us consider, as an example, the three handwritten digits presented in Figure 1. A good pattern recognizer, would, in theory, be invariant to the aforementioned transformations and classify all three images as belonging to the digit zero category.

In fact, as we look at the provided example, we become very aware of how easy it is for our brain to tolerate such changes. Actually, a biological study on mammals that was conducted by Hubel and Wiesel [1], showed that this tolerance is provided through a special kind of cells that exists inside the visual cortex. These are called complex cells and they react to specific visual stimuli regardless of their position. More specifically, a complex cell appears to be discriminant on the kind of visual feature, as it reacts differently to an horizontal line when compared to, for instance, a vertical

one. However it also shows shift invariance as it reacts equally to several shifts of the same feature.

The study of the visual cortex inspired several computational models for visual pattern recognition like Convolutional Networks (CNNs) [2], the Neocognitron [3], HMAX [4,5] or the Map Transformation Cascade (MTC) [10]. Despite the fact that such models achieved tremendous results, the need for many training examples raises the question of how the brain recognizes patterns with so few examples. In fact, even Hinton, a great pioneer of the field, raised the questioned in a recent interview[1].

So, with that in mind, we set out to try to get closer to answering the question by:

- Proposing a new look over the complex cell that allows the model to learn shifted variants of a category from fewer examples (justified in section 2.1 and proposed in section 3).
- Making a distinction between two learning phases where: first, like Hubel and Wiesel proposed, an unsupervised stage occurs where the model acquires the capability to see; second, the supervised learning of pattern categories, can occur from fewer examples (justified in section 2.2 and proposed in section 5).
- Switching from the typical vector comparison paradigm to a biologically inspired set comparison which increases invariance to distortion (proposed in section 4).
- Using the aforementioned contributions to create the possibility to achieve a steep learning curve, where a very simple one stage model can achieve high performances after learning from few examples (described in section 5 and validated in section 6).

2 Background

In this section, we further detail the problem that we are trying to address. More specifically, we describe two key points where current models can be improved through the integration of biological insights, like the ones that led to their formulation. In fact, those points may even justify the discrepancy between current computational models and the mammalian visual system in terms of training requirements.

2.1 The invariance trade-off

Throughout this subsection, we describe the trade-off that the currently used subsampling operation in the complex cell introduces. To achieve just that, we start by further detailing the classical hypothesis of Hubel and Wiesel, so as to understand the role of this cell in the process of visual pattern recognition.

[1] https://www.axios.com/artificial-intelligence-pioneer-says-we-need-to-start-over-1513305524-f619efbd-9db0-4947-a9b2-7a4c310a28fe.html

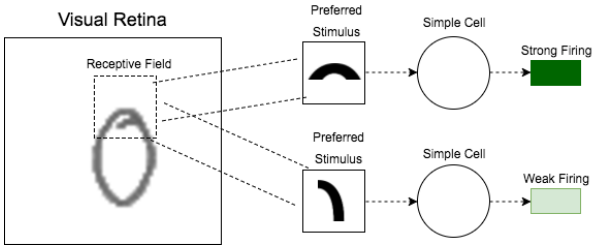

Fig. 2 An illustration of the functioning of a simple cell.

2.1.1 Hubel and Wiesel's view over the visual cortex

By presenting patterns to several mammals and measuring the reaction of an area of their brain called the visual cortex, Hubel and Wiesel formulated several findings on the mammalian visual system [1,6,7]. One of the most important finding was that, in the visual cortex, there are two types of cells.

Simple cells possess both an excitatory and an inhibitory region [7] so they can react to specific stimuli that they previously learned. As they receive as input local subpatterns in a specific position and orientation they will show some reaction. The strongest reaction happens when the cell receives as input the subpattern that it has learned. Figure 2 illustrates the functioning of a pair of cells of this kind for a specific region of an image in the visual retina. This corresponds to a simple local pattern matching to code an image into features and it is usually also referred to as a convolutional step [2].

Besides, simple cells, the classical hypothesis also refers the complex cell. These units have larger receptive fields that also react to stimuli [7]. However, they respond to a given subpattern in the same way, regardless of its position in the receptive field. So, they do not only integrate the subpatterns that were recognized by simple cells, but also allow for their shifting. Figure 3 presents a cell of this sort reacting invariantly to shifts of the same pattern. This operation is usually implemented through local subsampling, by taking a window and assigning to it only the strongest part of it, discarding information about the specific position inside the window where the reaction was strong [2].

2.1.2 The problem of subsampling in the complex cell

As was previously stated, with the intent to add invariance, complex layers tile the s-planes into windows and,

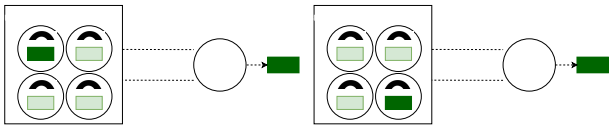

Fig. 3 A complex cell reacting invariantly to shifted occurrences of its preferred stimulus.

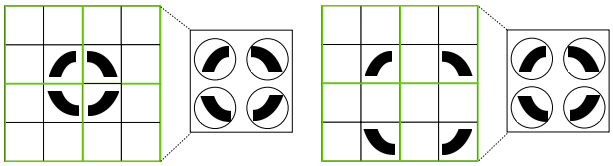

Fig. 4 An example where two shifts of the digit zero are recognized invariantly.

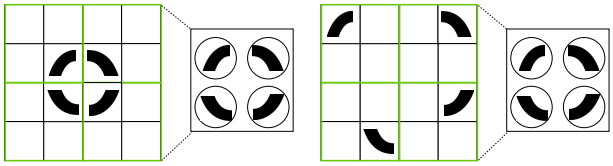

Fig. 5 An example where the model recognizes as the same an instance of the digit zero and a pattern which is not that digit.

for each of those windows, discard the specific information about the position of the feature inside it. By doing so, these models tolerate shifts that may occur inside a given window. To clarify, let us consider Figure 4 as an example where this mechanism allows a model to recognize two variants of the same category (i.e. the digit zero) as so.

However, the addition of invariance that comes from discarding information in the complex cells introduces a loss of discriminatory power [8]. More specifically, we can analyze Figure 5 and see how the invariance mechanism can also mislead a model.

The previously described trade-off suggests that current models are capable of tolerating some of the small shifts that can occur. Yet, it also shows that large shifts of the full pattern or even changes in size can become problematic. Therefore, to go around this issue, current models require large training sets that can accommodate examples of the biggest possible number of a category's variants.

2.2 Learning in two phases

Another key issue that we analyze is the learning phases that occur in current models when compared to the ones that may occur inside the brain. In fact, two different phases seem to happen before one can classify patterns. Biological experiments by Hubel and Wiesel show that a primary stage of learning occurs that enables vision [9]. Young infant cats with functional vision had one of their eyes sutured at the beginning of life. Without being capable of receiving images, the sutured eye was unable to develop and the cats became blind. We can see this process as the stage of learning that deep models of the visual cortex go through for cells to learn their preferred stimuli. Since such learning happens on every image that the eye processes during the first days of its life, it makes sense to implement this behavior as an unsupervised procedure with a relatively large training set.

With a functional eye, the process of learning categories (e.g. the digit zero) seems to be independent. Since a cat that goes through the first phase of learning can see, but until someone teaches it what is a playing ball he can see playing balls but it cannot recognize them as so. Furthermore, unlike the first phase of learning that requires days, a cat can learn what is a playing ball quickly and from very few examples.

In fact, this separation of two different phases of learning is not entirely new as some models describe their implementation of the classical hypothesis as a kind of preprocessing and then use an independent classifier to learn categories [10,5]. However, the use of typical machine learning classifiers, instead of a biologically inspired one, can be one of the reasons for these models' need for larger training sets.

Other models like Convolutional Networks [2] or the Neocognitron [3] have built-in classifiers that learn categories within the same process of preferred stimuli learning. Such property can be one of the reasons for the need for many training examples.

So, just like [10,5] we propose to look at Hubel and Wiesel's hypothesis as a first phase that performs a kind of preprocessing. However, afterwards, we do not intend to use, like some of the aforementioned models, a typical machine learning classifier. Instead, we propose to explore a different, biologically plausible hypothesis with less data requirements.

3 A different complex cell

In this section, we propose a different way of implementing invariance gain in the layer of complex cells. These units receive, as input, the output from the layer of simple cells. Said layer maps the image into visual features. Let us focus on Figure 6 which illustrates its operation. For the sake of the example, we consider as visual features four oriented lines with orientations of 0°, 45°, 90° and 135°.

Analyzing the resulting output we can clearly identify the need for invariance as two apparently very similar patterns are completely different for a vector-based

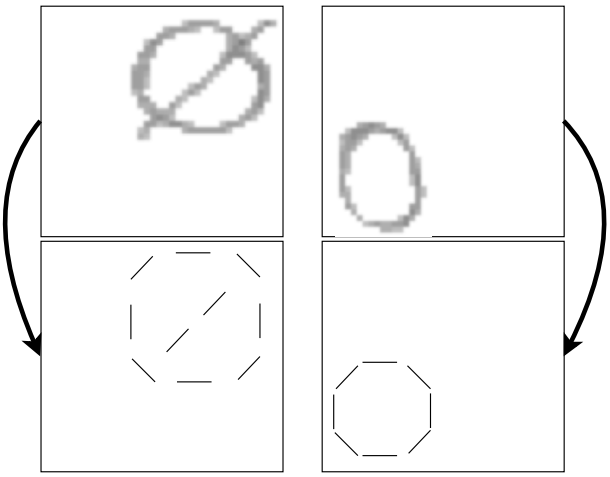

Fig. 6 Two patterns go through the layer of simple cells and, thus, are mapped into features.

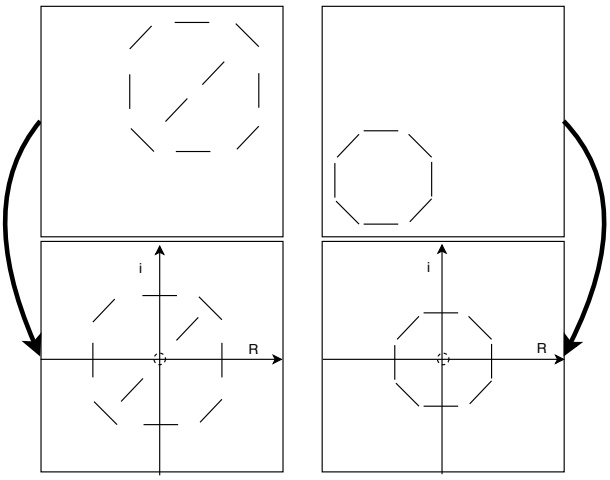

Fig. 7 Two patterns go through the first purpose of the complex cell as the positions of the features are coded in a polar referential which is centered in the pattern's center.

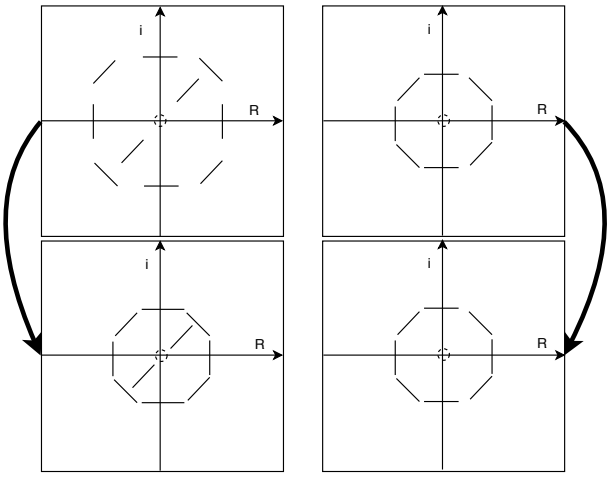

Fig. 8 Two patterns go through the second purpose of the complex cell as the positions of the features are normalized to the radius one circle.

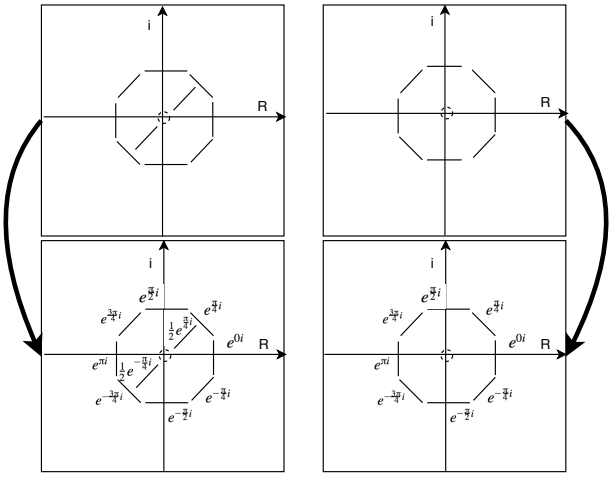

Fig. 9 The output of the complex layer for two patterns, where each feature's position is coded as a complex number.

comparison due to their global shifts. For that reason, the first purpose is to, as Figure 7 suggests, find the pattern's center and code the visual features' positions in a polar coordinates referential whose origin is placed in that same center.

As we look at the output we can see that patterns are now closer. However, there is still a difference introduced by the size variance caused by local shifts of each feature. To address this issue, the second purpose of the layer of complex cells would be to normalize the features' positions to the radius one circle as Figure 8 suggests.

So, the final output of the complex layer is an invariant representation of the pattern as a set of features, where each of which is identified by its type and by a complex number that corresponds to its polar coordinates in the aforementioned referential. For simplicity, we provide Figure 9 as an illustration.

With that said, one might question how is there biological feasibility in the previously described polar coordinates representation. However, if we consider that each feature corresponds to the output of a given complex cell, we can conclude two key points: first, the type of feature (i.e. the orientation of the line) corresponds to the cell's preferred stimuli; second, the complex number that codes the position can be described as the pair of phase and amplitude of the cell's firing. Furthermore, the idea itself of using polar coordinates has been widely referred to in biological work as a transformation that happens between the retina and the cortex inside the brain [11].

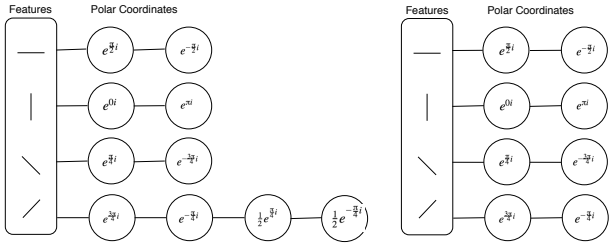

Fig. 10 Two patterns coded as sets where each feature is coded by its type and its position.

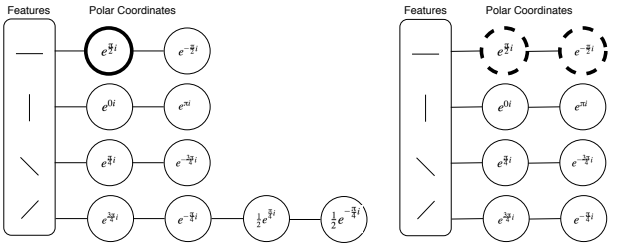

Fig. 11 A feature is selected for analysis and is compared with previously learned features of the same kind.

4 Comparing sets instead of vectors

After applying the previously described complex layer, a pattern is coded as a set of features, where each is described by its kind and its location [15,16]. Going back to our ongoing example, we provide Figure 10 as an illustration of the resulting sets.

Such representation can also be biologically interpreted. In fact, it was suggested [12] that after recognizing local features in a pattern, the visual system feeds them into a visual buffer. Inside that buffer, each feature is categorized by two mechanisms [13]: the first is referred to as the "what" pathway which is located in the temporal lobe and codes the kind of the feature; the second pathway codes, in the parietal lobe, the "where" of the feature, that is, its position inside the image. The two pathways are fulfilled and form a cognitive entity [14–16]. Such coding is in accordance with our set representation, where each can be seen as a set of cognitive entities in the visual buffer.

With the patterns represented as sets, a new problem emerges on how to compare two of them. First of all, we assume that, in a comparison, one of the sets is the input set (i.e. the one in the visual buffer) and the other is stored in memory (i.e. it was learned beforehand). So, the intuition behind our comparison is to find out how many features of the input set are in accordance with the memorized set. More specifically, two features are in accordance if they are of the same kind and their positions are close in the polar referential.

Let us go back to our ongoing example (see Figure 10) and define the leftmost set as input and the rightmost one as a learned representative of the digit zero category. To perform the comparison we must also define a threshold that states how close must two features be so that they are considered to be close. With that clarified, the set comparison can be performed by focusing on one of the input's features at a time. For the sake of the example, we will focus on the first feature of the first kind as Figure 11 suggests. Afterwards, the comparison occurs for a given threshold (we will assume $T = 0.2$) to decide if the feature under analysis is relevant or not. In our example, the operation described in Figure 12 occurs and, thus, the feature under analysis is defined as relevant to recognize the digit zero category.

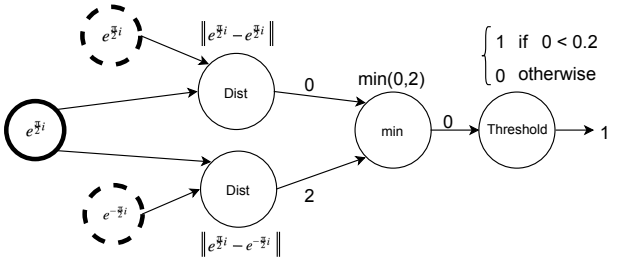

Fig. 12 A sketch of the analysis operation for a given feature.

After performing the aforementioned operation, we find that 8 out of the 10 features in the input are marked as relevant. Hence, we say that the similarity between the input set and the learned set is $\frac{8}{10} = 0.8$.

The first thing that one should question regards the advantages of using sets. In fact, patterns are often built by a different amount of features. Even in our example, the two patterns belong to the same category and do not share all the features. Furthermore, by getting away from the strict vector based hard comparison, this approach allows for the use of a threshold based soft comparison that allows for invariance to small distortions.

A second question that comes up, regards the biological feasibility of the aforementioned comparison process. A very similar idea was biologically described before as a searchlight mechanism that is controlled by the thalamus [17]. According to this theory [18], this mechanism focuses attention in a given entity analyzing it. At the completion of an analysis, the attention window is shifted throughout the buffer so as to find the next attention focus [13,19]. The purpose of the analysis, is to define if an entity is relevant enough to recognize a class. So, the focused object is compared to similar objects of previously learned class representatives and, if it is of interest to recognize a class, the object is marked to capture attention [19] for a final decision. Afterwards, the marked objects, where attention remains, are bounded as a whole for recognition [13,20, 21].

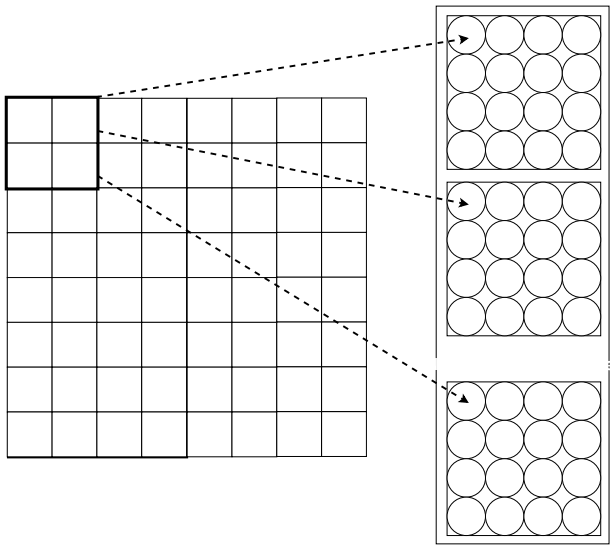

Fig. 13 A sketch of a connection between the input image and cells of several planes in the simple layer.

5 A two-phased architecture

Throughout this section we describe the details of the implementation of the previously described principles.

5.1 Visual cortex

The first phase of the model is inspired by Hubel and Wiesel's study of the visual cortex and as so it is built by simple and complex layers. As was stated before, this phase works as a kind of preprocessing step to map a pattern into invariant features before recognition.

5.1.1 A layer of simple cells

The s-layers in this model are very similar to the ones of related models [2,3,10]. First of all, cells are organized into planes, where each plane is tuned to recognize a given feature. Each cell can be identified by the plane $k \in K$ it belongs to and by its position $\mathbf{n}$ inside it.

The input image corresponds to the retina and has topologically ordered connections to the layer of simple cells. Thus, each cell receives, as input, a window of the image. We denote said window by $R^I_{\mathbf{n}}$ which refers to the receptive field of a cell with position $\mathbf{n}$, in any plane, that comes from the input image I. This receptive field is basically the set of positions $\mathbf{v} \in R^I_{\mathbf{n}}$ of the pixels that feed their input to the cell. The connections between a receptive field in the input image and cells of several planes are illustrated in Figure 13.

Each simple cell that belongs to cell plane k has a preferred stimulus w_k which is compared with the contents of the receptive field through a cosine similarity. In general, to define the operation of a simple cell, we use the following quantities:

- $O_I(\mathbf{n})$: weight stored by pixel with position $\mathbf{n}$ inside the input image.
- $w_k(\mathbf{v})$: weight stored by pixel with position $\mathbf{v}$ inside the preferred stimulus w_k.
- $R^I_{\mathbf{n}}$: set of positions inside the receptive field from input image I to cell with position $\mathbf{n}$, in any plane.

The comparison between input and preferred stimulus yields the net input net_S to a cell with position $\mathbf{n}$ in plane k as Equation 1 suggests.

$$net_S(\mathbf{n},k) = \frac{\sum_{\mathbf{v}\in R^I_{\mathbf{n}}} O_I(\mathbf{n}+\mathbf{v}) w_k(\mathbf{v})}{\sqrt{\sum_{\mathbf{v}\in R^I_{\mathbf{n}}} O_I(\mathbf{n}+\mathbf{v}))^2}\ \sqrt{\sum_{\mathbf{v}\in R^I_{\mathbf{n}}} w_k(\mathbf{v})^2}} \tag{1}$$

Simple cells in the proposed model implement the "winner-takes-all" principle. It was shown [10,22] that such mechanism exhibits more robustness to noisy images. So, for each window of the image, only the cell with the strongest firing passes its output to the next layer. Hence, we can write the output of such a cell using Equation 2.

$$O_S(\mathbf{n},k) = \begin{cases} 1 & \text{if } \arg\max_{l\in K} net_S(\mathbf{n},l) = k \\ 0 & \text{otherwise} \end{cases} \tag{2}$$

Overall, the purpose of this layer is to code the image into previously learned features (i.e. preferred stimuli). The learning process of these stimuli intends to, somehow, mimic the previously described idea of cats gaining the capability to see. So, as was done in MTC [10], we implement the very simple, yet effective unsupervised k-means clustering. More specifically, several images are shown to the layer which breaks them into windows for clustering. Each of the final centroids represents a stimulus and so it will correspond to a plane in the architecture.

5.1.2 A layer of complex cells

After the simple layer, complex cells are supposed to add shift invariance. In our proposed model, each of these cells receives input from all the simple cells in the previous layer. However the connections between the two layers are not topologically ordered, since the position information is coded in each s-cell signal. For that reason, the complex layer is organized differently.

Unlike the aforementioned biologically inspired models, we implement the principle that the mammalian

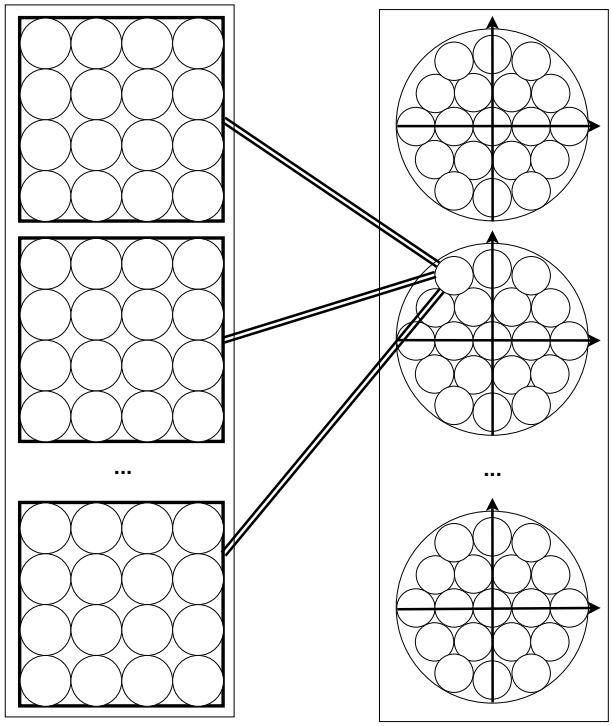

Fig. 14 A sketch of the connections between simple cells and complex cells where every s-cell is connected to every c-cell.

vision may leverage a polar coordinates system which differs from the retinal one [11]. So, complex cells are organized in planes and each plane is organized so as to sample the continuous space of the radius 1 polar circle. Such quantization is performed by the complex cells, such that each occupies a small circle that contains several positions. Figure 14 illustrates the described organization and its connection to the previous layer.

Throughout this section, we refer to the space covered by a c-cell with polar position $\mathbf{n}'$ as $B^C_{\mathbf{n}'}$. The radius of these circles depends on the number of c-cells in the layer and controls the loss of information that comes from the quantization process.

To perform the change of coordinate system, a complex cell transforms each signal $\mathbf{v}$ from each simple cell to a polar coordinates signal $\mathbf{v}'$ and fires if one of those transformed signals corresponds to a position inside the space that it samples. More specifically, if a c-cell from plane k that samples the radius r circle centered in $\mathbf{n}'$, will react if there is a simple cell from plane k with position $\mathbf{v}$ in the retinal coordinate system, such that when $\mathbf{v}$ is converted to the polar signal $\mathbf{v}'$ it belongs to that c-cell's circle, that is $\mathbf{v}' \in B^C_{\mathbf{n}'}$.

As was described beforehand, the first intuition of our proposed complex cell is to place a polar coordinates referential on the pattern's center $\mathbf{C}$. The center can be computed through the weighted average of all the positions that contain at least one active cell. If we define R^S_k as the set of positions in plane k of the simple layer, we can write Equation 3 to describe the center computation.

$$\mathbf{C} = \frac{\sum_{k \in K} \sum_{\mathbf{v} \in R^S_k} \mathbf{v}\ O_S(\mathbf{v}, k)}{\sum_{k \in K} \sum_{\mathbf{v} \in R^S_k} O_S(\mathbf{v}, k)} \tag{3}$$

The second purpose of the cell is to normalize the pattern's radius to the typical radius 1 circle. So, the pattern's current center must be computed as a normalizing term. Equation 4 describes that computation.

$$R = \max_{k \in K, \mathbf{v} \in R^S_k} ||\mathbf{v} - \mathbf{C}|| O_S(\mathbf{v}, k) \tag{4}$$

Using the aforementioned quantities, we can write Equation 5 to describe the mapping of a signal $\mathbf{v}$ in the retinal coordinate system to its corresponding $\mathbf{v}'$ in the polar system.

$$\mathbf{v}'(\mathbf{v}) = \frac{\mathbf{v} - \mathbf{C}}{R} \tag{5}$$

Finally, we can describe the net input to a c-cell from plane k that covers the radius r circle centered in position $\mathbf{n}'$ as the minimum distance between the center of such circle and a transformed signal that comes from the corresponding k-th s-plane. Equation 6 described this computation.

$$net_C(\mathbf{n}', k) = \min_{\mathbf{v} \in R^S_k} ||\mathbf{v}'(\mathbf{v}) - \mathbf{n}'|| \tag{6}$$

The net input is then passed through a threshold based activation function that determines if the closest signal belongs to the space it covers. More specifically, if the distance between the center of the circle and the closest feature is smaller or equal to the radius r, then the cell fires (see Equation 7).

$$O_C(\mathbf{n}', k) = \begin{cases} 1 & \text{if } net_C(\mathbf{n}', k) \leq r \\ 0 & \text{otherwise} \end{cases} \tag{7}$$

The information about the position of each active cell is passed through its signal. In particular, we code this position as a complex number described by the cell's phase and amplitude of firing. So, by the end of the c-layer, the pattern is: first, described into several kinds of features, one for each plane; second, each of its features is associated with an active c-cell which has an invariant position coded into a complex number in a polar coordinates system.

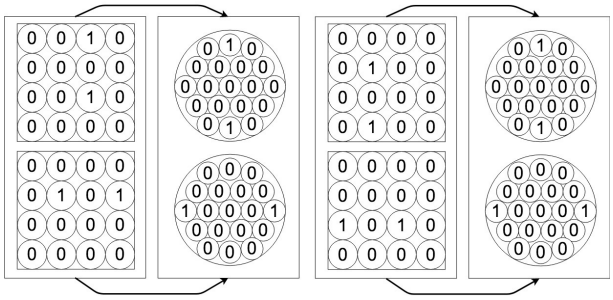

Fig. 15 Two shifted versions of the same pattern yielding the same activation of complex cells.

With the previously described c-cell operation, the model gains the invariance to shifts that Hubel and Wiesel discovered in their studies. In order to specifically exemplify in terms of cell activation the said gain we provide Figure 15, where two shifted versions of the same pattern cause different activation patterns in the s-layer but the same in the c-layer.

5.2 Attention based recognition

As was described before, after the processing of an image a description of its objects fills a visual buffer. Each cognitive entity in the buffer is described through two pathways. Afterwards, a searchlight mechanism is used to focus attention on specific entities and perform classification. This section covers the details on how we propose to implement such principles.

5.2.1 Searchlight "Mark" operation

While processing the visual buffer, a searchlight analyzes each entity and decides, for each category, if that entity is relevant (i.e. worthy of attention to recognize that category) or not [16].

To implement this behavior, let us assume that we have a trained memory where there are labeled sets of entities. While processing an object in the visual buffer, we compute, for each class, if it should be marked or not through a threshold function.

Consider the visual buffer and the example memory from Figure 16, where there are N_C class representatives and each is stored as a set of cognitive entities [16]. Each of these entities can belong to one of K kinds and has its position coded in polar coordinates. When the searchlight focuses feature (k, j), it computes, for each class representative c, Equation 8 that yields the distance of the focused feature, to the closest feature in D_k^c, which is the set of features of the same kind in that representative c. Such distance is then fed as net input to an activation function ϕ (see Equation 9) with

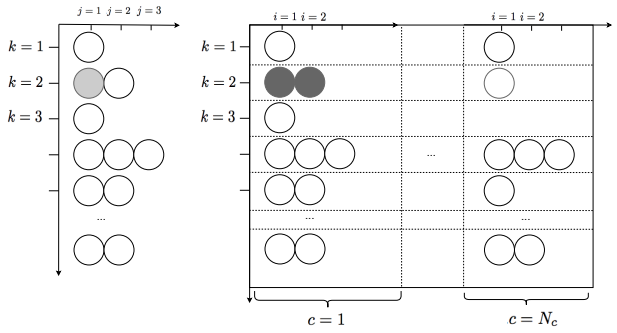

Fig. 16 An example memory where feature $j = 1$ of kind $k = 2$ in the visual buffer is under analysis.

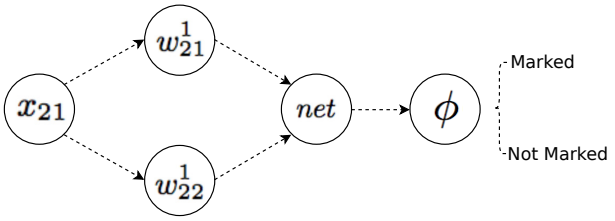

Fig. 17 An illustration of the mark operation between class representative $c = 1$ and feature $j = 1$ of kind $k = 2$ in the visual buffer.

threshold T that will decide if the feature should be marked or not (see Equation 10). Figure 17 exemplifies the mark operation for feature $(k, j) = (2, 1)$ and representative $c = 1$.

$$net_{kj}^c = \min_{i \in D_k^c} |x_{kj} - w_{ki}^c| \tag{8}$$

$$\phi(x, T) = \begin{cases} 1 & \text{if } x \geq T \\ 0 & \text{otherwise} \end{cases} \tag{9}$$

$$mark_{kj}^c = \phi(net_{kj}^c, T_k^c) \tag{10}$$

This process is repeated for every feature in the buffer and, so, in the end, for each class, there is a group of marked features.

5.2.2 Bound and compare sets

After marking, the model bounds all the active entities for a given class representative. We implement this bound operation as a simple normalized count. So, we define the similarity between the image inside the visual buffer and the memorized representative of a class as the percentage of marked units, in the buffer, for that representative. Equation 11 describes this operation.

$$sim(c) = \frac{\sum_{k=1}^{K} \sum_{j \in D_k^c} mark_{kj}^c}{\sum_{k=1}^{K} |D_k^c|} \tag{11}$$

The model computes the most similar of all memorized representatives as Equation 12 suggests. Afterwards, that representative's label can be seen as the classification output for the input image.

$$o = \underset{c \in \{1,\ldots,N_C\}}{\arg\max} \; (sim(c)) \tag{12}$$

6 Experiments

The purpose of this work is to demonstrate that, by integrating our contributions, a very simple model can achieve a very steep learning curve. In fact, it is this brain like ability to learn from few examples that we intend to gain. For that reason, we measure the accuracy of the model for several train sets with growing sizes, so as to plot a learning curve. This curve has the accuracy as the dependent variable and the size of the training set as the independent one.

Looking at the second phase of the proposed model, we made the assumption that we had a fulfilled memory with class representatives. Also, we made the assumption that the marking thresholds were defined for each feature kind of each representative inside the memory. So a question rises on how to find these parameters. Of course one could optimize classification by applying, for instance, a typical backpropagation algorithm. However, with the intent to demonstrate the advantage that comes from this new approach in isolation, we employed three principles: first, unlike other models we minimize preprocessing; second, we do not perform any optimization of the parameterizations; third, we apply a simple Instance-based learning where we store the training representatives and classify a test pattern with the label of the most similar of all the memorized ones.

6.1 Setting the experiments

Before starting the experiments the architecture's parameters must be defined. In particular, the layer of simple cells works with 4×4 pixel windows (or receptive fields) that come from the input image and are shifted by a unit. These cells are organized into 15 planes, where each is tuned to recognize a preferred stimulus which was determined by clustering.

The complex layer performs the space quantization of the radius 1 polar circle, through 15 planes with several c-cells, where each samples a small circle.

Regarding the second phase of the model, as was stated before, simple Instance-based learning is applied, so training patterns are stored for comparison. Such comparison depends on the choice of a threshold, which we define as 10% of the maximum possible distance between two features in the radius 1 circle, for all kinds of all representatives.

The comparison of two patterns through sets could be seen as, potentially, time consuming. However, in practice, this is not the case as each pattern is represented by 15 sets of, on average, 30 features. Thus, in the worst case, a comparison would take $15 \times 30 \times 30$ operations. Furthermore, to achieve a faster simulation, since we used a serial computer, instead of sampling the full space in the complex cell, we adaptively sample just the required space. More specifically, if we look at classification time, since training is fairly quick, the longest running time in a slow MATLAB interpreter running on a simple intel server was around a day.

6.2 The ETL-1 dataset

A validation task was performed on the widely used [3,10] ETL-1 database[2] of handwritten digits. In fact, Fukushima used this database to test his seminal Neocognitron model [3].

It is relevant to say that many models [10,23] used advanced preprocessing techniques like contrast and edge extraction before using this dataset. However, in this work, a simple threshold mechanism was used to remove some background noise.

ETL-1 does not have a predefined separation between training and test sets. For that reason, to achieve comparability with the optimized version of the Neocognitron [23], we also use 3000 patterns as the maximum size of a training set and as the size of the test sets. However, these sets cover less than half of the full dataset as it contains over 14000 patterns. For that reason, we designed the experiment a bit differently. First of all, we broke the dataset into four folds of 3000 patterns. Then, we applied a kind of cross validation by training and testing on all the possible pairs of these folds. The mean accuracies and standard deviations achieved by this experiment are presented in Figure 18 in the shape of a learning curve.

Analyzing the learning curve on Figure 18, we can see that, like a mammal, at the beginning, the model improves its accuracy with experience until it reaches a given optimal point. Afterwards, the accuracy is approximately constant with some small oscillations happening when very noisy patterns are introduced in the training set.

By looking at Table 1, which contains the comparable models' results, we denote that our model is able

[2] http://etlcdb.db.aist.go.jp/

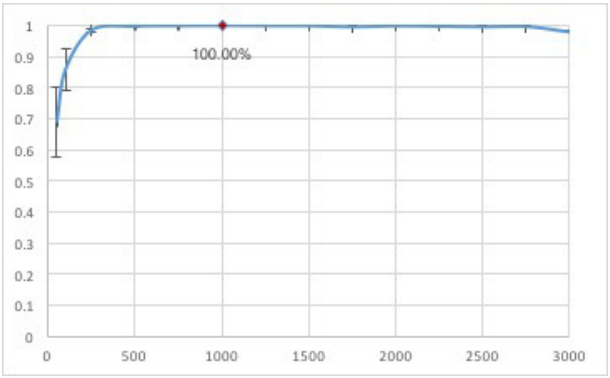

Fig. 18 Accuracy on the ETL-1 test set in terms of the size of the training set.

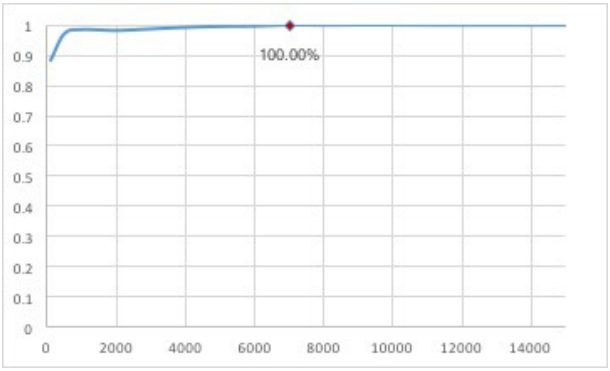

Fig. 19 Accuracy on the MNIST test set in terms of the size of the training set.

Table 1 Results of comparable biologically inspired models on ETL-1.

Model	Train Set	Test Set	Accuracy
MTC [10]	200	3000	93.33%
Attention Net1	200	3000	98.47%
Neocognitron [23]	3000	3000	98.6%
Attention Net2	1000	3000	100.00%

Table 2 Results of comparable biologically inspired models on the MNIST test set.

Model	Train Set	Test Set	Accuracy
CNN [25]	60000	10000	98.30%
MTC [22]	60000	10000	99.30%
Ensemble of CNNs [26]	60000	10000	99.70%
Capsule Networks [24]	60000	10000	99.75%
Attention Net	7000	10000	100.00%

to achieve similar or better accuracies with less training examples. Furthermore, by applying the aforementioned cross validation, our approach covers almost all the dataset so the results are even more robust.

6.3 The MNIST dataset

The famous MNIST dataset[3] has been widely used to test invariant pattern recognition models [22,2]. For that reason, we also validated our approach on it.

Unlike ETL-1, this dataset has a predefined separation between training and test set. The former possesses 60000 images, whereas the latter possesses 10000 images. Since such separation exists, in this case, we do not use cross validation. For that reason, the performed experiment was done so as to compute a learning curve where accuracy on the 10000 sized test set is the dependent variable and the number of training examples is the independent one.

Analyzing the learning curve on Figure 19, we can see that, like in the previous dataset, at the beginning, the model improves its accuracy with experience until it reaches a given optimal point. Afterwards, the accuracy is approximately constant with some small oscillations happening when very noisy patterns are introduced in the training set. This behavior is somewhat different from typical models, where accuracy increases with the training set's size.

[3] http://yann.lecun.com/exdb/mnist/

Table 2 summarizes the results obtained by other biological models like Convolutional Networks (CNNs) [2], the Map Transformation Cascade (MTC) [10,22] and Capsule Networks [24] a recently proposed variation of CNNs. Analyzing it we can see that the achieved accuracies are generally close to the ones achieved by previous state-of-the-art models. However, our approach is able to achieve them with far less examples. In fact, it even improves the best results so far [24] by achieving 100% accuracy using a bit over 10% of the training set.

7 Conclusion

Hubel and Wiesel based models achieved state-of-the-art results in invariant pattern recognition tasks. However, they exhibit some characteristics that can be worth improving: first, too many training examples are required due to the loss of information that occurs on the subsampling operation of complex cells; second, some of these models join preferred stimuli and class learning which is biologically unlikely; third, the ones that do not join the two phases, rely on data hungry, non-biological machine learning classifiers.

We proposed a different approach where we separate the process into two distinct learning phases. The first is a classical hypothesis inspired phase that works as a mechanism to code a pattern into invariant features. The posterior phase is a biologically inspired set based classifier.

The first phase starts by using a simple cosine similarity to implement a "winner-takes-all" simple cell that maps a pattern into features. Afterwards, a new implementation for the complex cell is used. By quantization of the radius 1 circle, the c-layer maps the features' positions to an invariant polar coordinates referential. Such operation, allows for toleration to shifts in the pattern. The output of the first phase fulfills a visual buffer that contains information about which features are present in the pattern and what are their positions.

The second phase uses a set based comparison that tolerates small distortions and is based on previous biological work. More specifically, each class has a group of memorized representatives that are stored as sets of features. To perform classification, a visual searchlight mechanism compares each feature in the visual buffer to those of the representatives that are stored in memory and marks, for each class, the relevant ones.

After the mark operation, to compute the similarity between the contents of the visual buffer and a given memorized representative, we bound all the marked entities by counting and normalizing. From all the memorized options, the most similar is chosen as the recognized output.

The model was tested with Instance-based learning in the typical task of handwritten digit recognition, in two popular datasets. The results achieved served the purpose as they demonstrate that by integrating biological insights we can achieve higher performance using fewer training examples.

References

1. D. H. Hubel and T. N. Wiesel. Receptive fields, binocular interaction and functional architecture in the cat's visual cortex. *The Journal of Physiology*, 160:106–154, 1962.
2. Yann LeCun, Yoshua Bengio, et al. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10):1995, 1995.
3. Kunihiko Fukushima. Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. *Biological Cybernetics*, 36:193–202, 1980.
4. Maximilian Riesenhuber and Tomaso Poggio. Hierarchical models of object recognition in cortex. *Nature*, 2:1019–25, 1999.
5. Thomas Serre, Lior Wolf, Stanley Bileschi, Maximilian Riesenhuber, and Tomaso Poggio. Robust object recognition with cortex-like mechanisms. *IEEE transactions on pattern analysis and machine intelligence*, 29(3):411–426, 2007.
6. D. H. Hubel and T. N. Wiesel. Receptive fields and functional architecture of monkey striate cortex. *The Journal of physiology*, 195:215–243, 1968.
7. David H Hubel. *Eye, brain, and vision*, volume 22. Scientific American Library New York, 1988.
8. Etienne Barnard and David Casasent. Shift invariance and the neocognitron. *Neural Networks*, 3(4):403–410, 1990.
9. David H Hubel and Torsten N Wiesel. Effects of monocular deprivation in kittens. *Naunyn-Schmiedebergs Archiv für Experimentelle Pathologie und Pharmakologie*, 248(6):492–497, 1964.
10. Ângelo Cardoso and Andreas Wichert. Neocognitron and the Map Transformation Cascade. *Neural Networks*, 23:74–88, 2010.
11. E. L. Schwartz. Anatomical and physiological correlates of visual computation from striate to infero-temporal cortex. *IEEE Transactions on Systems, Man, and Cybernetics*, SMC-14(2):257–271, 1984.
12. Charles G. Gross and Mortimer Mishkin. The neural basis of stimulus equivalence across retinal translation. In S. Harnad, R. Dorty, J. Jaynes, L. Goldstein, and G. Krauthamer, editors, *Lateralization in the Nervous System*, pages 109–122. Academic Press, New York, NY, USA, 1977.
13. Michael I. Posner and Marcus E. Raichle. *Images of mind*. Scientific American Library, New York, NY, USA, 1994.
14. James A. Anderson. *An introduction to neural networks*. The MIT press, Cambridge, MA, USA, 1995.
15. Andreas Wichert, Joao Dias Pereira, and Paulo Carreira. Visual search light model for mental problem solving. *Neurocomputing*, 71(13-15):2806–2822, 2008.
16. Andreas Wichert. The role of attention in the context of associative memory. *Cognitive Computation*, 3(1):311–320, 2011.
17. F. Crick. Function of the thalmic reticular complex: the searchlight hypothesis. In B.J. Baars, W.P. Banks, and J.B. Newman, editors, *Essential Sources in the Scientific Study of Consciousness*. The MIT Press, Cambridge, MA, USA, 2003.
18. Cathryn J. Downing and S. Oinker. The spatial structure of visual attention. In M.I. Posner and O.S.M. Marin, editors, *Mechanisms of attention: Attention and performance XI*, pages 171–187. Erlbaum, Hillsdale, NJ, USA, 1985.
19. Stephen M. Kosslyn. *Image and Brain: The Resolution of the Imagery Debate*. The MIT Press, Cambridge, MA, USA, 1994.
20. Antonio R. Damasio and Hanna Damasio. Cortical systems for retrieval of concrete knowledge: The convergence zone framework. In C. Koch and J.L. Davis, editors, *Large-Scale Neural Theories of the Brain*, pages 61?–74. The MIT Press, Cambridge, MA, USA, 1994.
21. Anne Treisman and Stephen Gormican. Feature analysis in early vision: evidence from search asymmetries. *Psychological Review*, 95(1):15–30, 1988.
22. Ângelo Cardoso and Andreas Wichert. Handwritten digit recognition using biologically inspired features. *Neurocomputing*, 99(Supplement C):575 – 580, 2013.
23. Kunihiko Fukushima. Neocognitron for handwritten digit recognition. *Neurocomputing*, 51:161–180, 2003.
24. Sara Sabour, Nicholas Frosst, and Geoffrey E. Hinton. Dynamic routing between capsules. In *Advances in Neural Information Processing Systems*, pages 3859–3869, 2017.
25. Yann LeCun, Léon Bottou, Yoshua Bengio, and Patrick Haffner. Gradient-based learning applied to document recognition. *Proceedings of the IEEE*, 86(11):2278–2323, 1998.

26. Dan Ciregan, Ueli Meier, and Jürgen Schmidhuber. Multi-column deep neural networks for image classification. In *Computer vision and pattern recognition (CVPR), 2012 IEEE conference on*, pages 3642–3649. IEEE, 2012.