World Academy of Science, Engineering and Technology
International Journal of Cognitive and Language Sciences
Vol:16, No:8, 2022

The Relationship between Representational Conflicts, Generalization, and Encoding Requirements in an Instance Memory Network

Mathew Wakefield, Matthew Mitchell, Lisa Wise, Christopher McCarthy

Open Science Index, Cognitive and Language Sciences Vol:16, No:8, 2022 publications.waset.org/10012645.pdf

Abstract—This paper aims to provide an interpretation of artificial neural networks (ANNs) and explore some of its implications. The interpretation views ANNs as a memory which encodes instances of experience. An experiment explores the behavior of encoding and retrieval of instances from memory. A localised representation ANN is created that allows control over encoding and retrieved memory sample size and is experimented with using the MNIST digits dataset. The relationship between input familiarity, conflict within retrieved samples, and error rates is described and demonstrated to be an effective driver for memory encoding. Results indicate that selective encoding and retrieval samples that allow detection of memory conflicts produce optimal performance, and that error rates are normally distributed with input familiarity and conflict. By using input familiarity and sample consistency to guide memory encoding, the number of encoding trials on the dataset were reduced to 18.33% of the training data while maintaining good recognition performance on the test data.

Keywords—Artificial Neural Networks, ANNs, representation, memory, conflict monitoring, confidence.

I. Introduction

ARTIFICIAL Neural Networks (ANNs) have revolutionized the application of Artificial Intelligence (AI) on various types of recognition problem. Yet, accounts of how ANNs solve these kinds of problems are often vague, reducing the perceived trustworthiness of AI solutions. Ideally, an explanation should expose key principles that offer guidance for development, application of AI, and be accessible to general lay audiences.

Perhaps one of the best explanations was provided by LeCun et al. [1]. Two ideas about how ANNs solve classification problems are touched on in that review. The first relates to fully connected ANNs and uses the idea of hyperplanes. In this paper this idea is called *Hierarchical Representations*. The second relates to convolutions and implies the idea of template matching. This idea is discussed here under the name *Hierarchical Composition*. While these ideas imply classical machine learning (ML) principles, the problem-solving logic of each is not fully explored in LeCun et al.'s review [1].

In this paper, hierarchical representations, hierarchical composition, and their implications are further articulated and explored. Both ideas are similar in that they denote a form of content addressable memory [2]. But it is noted that encodings of information in each differ in the degree to which they are distributed or localized to representational units (i.e., artificial neurons). Each also has different trade-offs in terms of computation. Critically, it is argued that to achieve data compression and efficient retrieval, hierarchical representations trade off the ability to guarantee the validity of representations. Representations may not reflect true experiences from training history, leaving room for anomalous behavior such as is observed in adversarial examples [3].

This paper notes that different types of encoding have varying latitude for the selection of conflicting representations. Conflict is where multiple alternatives with different meanings are activated to represent input data. Conflict detection has an important role to play in cognitive control [4], but while conflict is obvious in some encodings (i.e., localist), in other encodings (i.e., distributed) there is no direct means of identifying it. One possibility for the latter is to augment a distributed representation system with a memory for instances, as per Complementary Learning Systems theory [5], [6] and some trends in machine learning (e.g., [7]). This paper presents a localist ANN for instance memory which uses epistemic sampling to judge conflict. It is demonstrated that memory conflict has a relationship to encoding demands as well as generalization performance. The technique also allows for the judgement of familiarity, that when combined with conflict, can be formalized into a metric that can effectively separate classification responses into those with high and low error rates *a priori*. This provides an effective means of judging classification confidence which serves as a useful internal monitoring (or meta-cognitive) function. It is further demonstrated that utilising judgement confidence to guide memorization can further reduce encoding demand with minimal effect on performance.

II. Problem Solving Logic

A. Hierarchical Representations

Many interpretations of deep ANNs focus on the idea of hierarchical representations. For example, LeCun et al. [1] write, "Deep-learning methods are representation-learning methods with multiple levels of representation... that each transform the representation at one level (starting with the raw input) into a representation at a higher, slightly more abstract level" (p.436). Early work in [8] originally posited this idea as a 'superposition of memory traces' that are

M. Wakefield is with the Department of Computing Technologies, Swinburne University of Technology, Australia (e-mail mwakefield@swin.edu.au).

M. Mitchell and C. McCarthy are with the Department of Computing Technologies, Swinburne University of Technology.

L. Wise is with the Department of Psychological Sciences, Swinburne University of Technology.

distributed across multiple representational units (i.e., artificial neurons) that translate inputs to outputs. The authors asserted that the logic involved was a synthesis between exemplar and prototype models of classification. This implies encoding information about individual instances (i.e., exemplars) as well as determining common elements and their statistical variances (i.e., central tendencies or prototypes).

The foundational concepts of distributed representations convey advantages in terms of automatic generalization and content addressable memory [2]. Automatic generalization occurs through the capturing of ranges of variance. Content addressable memory occurs in that matches to common elements allow for effective memory lookup. From this perspective, training instances can be fit neatly into the large dimensional space afforded by multiple scalar representational units (i.e., artificial neurons). This insight is supported by observations that deep learning networks can effectively memorize their training data [9], [10]. Common elements (i.e., patterns) of instances are reused to compress the data and form prototypes during training [11]. Detection of common elements during classification point to the relevant parts of memory removing the need for search. From a distance-based classifier perspective (i.e., exemplar and prototype models) these are attractive properties for dealing with large volumes of instances. Such memory compression may be related to long-term memory found in the cortex as compared to more immediate to medium term memory mediated by midbrain structures such as the hippocampus, as argued by Complementary Learning Systems theory [5], [6].

The hierarchical element to distributed representation relates to the fact that stacking encoding (transformation) schemes allows better organization [1]. Instance data that may be highly overlapping or disjoint in attribute space can be sorted into neatly separable classes. This description appears to apply equally well to deep auto-encoders [12] and error backpropagation trained networks [13]. In both cases, learning is framed as an optimization problem on vector representations. The representation on each layer is adjusted to fit some objective function which captures some undesirable or desirable properties. For example, adjustments can be made to reduce error at the output in the case of error backpropagation learning [13], or to increase representational similarity for augmentations of the same instance along lines of common variance like cropping, mirror image, and brightness in contrastive learning [14].

The learning of hierarchical representations can be summarized as 'learning an encoding'. This can be interpreted as findings ways to partition higher-dimensional vector space [11]. It is in learning this encoding that hierarchical representations take on the properties of a content addressable memory. It is important to note that by hierarchically organising training instances into regions of a vector space of distributed units, this encoding scheme appears to trade-off veridical memory. Veridical memory is where the current state of representation can be validated as being within the distribution of previous experience. Deep learning networks have difficulty validating whether current input is inside or outside its training distribution, a problem that is not shared by instance memory approaches [15].

To illustrate the difference between encoding and veridical memory we can use a language metaphor. We consider a system for encoding concepts as text. Letters and morphology (i.e., common elements and variances) may be hierarchically stored. The encoding scheme means that minimal information is needed in memory to represent all words. But the encoding scheme itself is not a vocabulary of valid words in that encoding. For example, a nonsense word such as 'garbangle' is a valid representation in terms of common syntactic elements, but is not a valid word in English. While some additional layers can provide a vocabulary of valid words, the vector space in distributed representation systems presents a problem.

Boundaries and transformations of the vector space are somewhat arbitrary. As adversarial examples show, small dissimilarities in unfamiliar ways can cause a crossing of partition boundaries resulting in misclassification [3]. One way of interpreting this may be in imaging the representational space being folded to obscure unsampled areas in training (see Fig. 2). This removal of unsampled regions in representational space makes it difficult to detect out-of-distribution inputs [15] and is likely to contribute to problems with resolving the stability-plasticity dilemma [16] since space for new information is removed. In other words, the representation is highly specialized to the training data such that small perturbations can cause arbitrary boundary crossings. Thus, the compression and fast lookup of the distributed representational scheme trades off veridical memory. Other methods like probabilistic programming [17] and lazy distance-based methods [18] are unlikely to suffer this problem. Small and unfamiliar perturbations would not cause a boundary crossing. Unfamiliar input would be too low in probability or too dissimilar to memory to be considered valid.

The explanation of transformations between layers in hierarchical representations is often vague. While this notion conveys the idea that the representation is modified, it presents no principled insight into the logic of such transformations and their role in decision making. Given that learning algorithms typically pose a problem of optimising vector representations, the process of learning can be considered as largely empirical and atheoretical. All that guides learning is making the best of a potentially exhaustive list of adjustments. This not only makes learning slow, but it also largely precludes any general interpretation of learning because no specific logic is involved other than the search for what is optimal. So, while optimization learning makes ANNs universal learners [1] it comes at the cost of general interpretability and efficiency in the learning process.

The discussion thus far has focused on the idea of fully distributed representation. Nominally this is what standard ANNs use [2]. A fully distributed representation can be considered as a situation in which a single entity is represented by a unique pattern of activity across all representational units. For example, a layer in an ANN can be considered as a size n word where every unique state of activity across nodes represents a specific 'thing'. This excludes simultaneous activations of multiple candidates. The latitude for simultaneous activations has computational implications.

Primarily, it precludes the detection of representational conflicts, which are important signals for cognitive control [4], which itself may be important for memory encoding. As the experiments of this paper show (Section IV), conflict in activations has a relationship to classification performance, memorization demand, and can be used as a signal of the meta-cognitive judgment of confidence.

B. Hierarchical Composition

The hierarchical composition of features is an intuitive interpretation of object classification. It is often found in cognitive modelling [19] and is present in descriptions of convolutional filters of Convolutional Neural Networks [1]. The explanation of recognition is typically described in terms of feature engineering: the reduction of input data dimensionality into a more general description that increases the information available for making discriminative judgments [20]. A set of common but discriminative patterns or attributes is identified and selected from the input data to summarize it, and this process is repeated hierarchically on the summary representations. This process can be conceived of as template matching. LeCun et al. [1] describe this in similar terms, with templates for simple features being matched on one layer, these template activations being fed as input into the next set of templates for more complex features, and so on. Those authors thus describes learning as largely being a process of automatic feature engineering. This interpretation of recognition logic is intuitive and accessible, and can be visualized (e.g., [21]).

A key facet of the hierarchical composition idea is the selection of discrete entities (i.e., features) from a range of options. Selection of discrete entities is most notable when representation is localized, as it appears to be in the convolutional feature layers of Convolutional Neural Networks [22]. Localist representation in ANNs have routinely been criticized in the past for issues of inefficiency and overfitting [19]. Because localist representation can allow multiple, potentially conflicting memory activations, constraints are typically applied to force selection of a single entity in cognitive modelling [23]. The presence of conflict in selections is typically viewed as a signal of ambiguity or interference that needs to be resolved via the engagement of cognitive control [4]. For example, such resolution may be found by appealing to representations held in hierarchically superior locations of memory [24] or perhaps by goal directed rules that frame current attention [25]. As discussed in Section II-A, fully distributed representation also conceptually forces selection.

LeCun et al. [1] point out that automatic feature engineering of a hierarchical composition is a difficult problem. The authors note that convolutional architecture trained with error backpropagation learning is most remarkable for solving it. It can be conceived that the problem relates to finding ways to segment instances of input data into a set of meaningful, hierarchically ordered chunks automatically. In standard Convolutional Neural Networks the first few layers of convolutions appear to be general for the visual domain [27] and perhaps represent a common visual syntax of features. After these initial layers, encoded information begins to appear more idiosyncratic, much like fragments of instances. Some examples of information encoded into individual artificial neurons of a deep convolutional neural network are presented in Fig. 1. The figure presents synthetic images (left) that indicate a preferred input for maximum activation of an artificial neuron under some regularization constraints. The middle column shows highlighted pixels of the input images (right) that lead to high activation of the neuron. These examples show qualitative evidence of fragments of instances or prototypes encoded into individual neurons starting at early layers. Inspecting Fig. 1 it is difficult to imagine how the layer 3 feature example (left) could be associated with anything other than a person. The instance like quality of the encoding is notable especially in the top-left corner where there appears to be a tree in the background. The layer 4 example (left) also shows evidence of specific instances of vehicle chassis and windows. These examples contrast with the example feature on layer 5 that is associated with at least two classes (snakes and harps).

A cursory appraisal of visualizations (see Fig. 1) also suggests that many similar instance fragments may be encoded into individual representation units. This conflation of instances has the appearance of capturing a range of statistical variance around a prototype. However, while a prototype is a single template or entity, these conflations appear as many related instances clustered together. For example, in Fig. 1 the layer 4 example appears to show fragments from several distinct vehicles. The layer 5 example appears to show multiple snake like fragments, perhaps even in the same frame. The interpretation of a conflation of instance fragments into a single representation unit could be what [8] were suggesting when they described ANNs as a synthesis between exemplar and prototype models of classification. The key difference is that their suggestion was for a region of vector space, and here it is a single local unit.

Treating individual representational units as clusters into which to encode related entities shows how localization can achieve the same objectives as hierarchical representations (Section II-A). It also may act as a content addressable memory and support automatic generalization. However, being localized, it conceptually does not trade off veridical memory, or the detection of conflicts. This is because insofar as nodes represent specific features, there is hypothetically no folding of the representational space. Input that is outside of training distribution should lead to little to no activity in the encoded features that represent specific entities (e.g., faces, snakes etc). This unfamiliarity can at least in principle be detected and produce low confidence in classification judgement. In practice, hierarchical compositions of features in architectures such as Convolutional Neural Networks are often paired with distributed representations in a set of fully-connected layers just before the output. Such networks are vulnerable to adversarial and out-of-distribution examples and lack good means for determining judgement confidence [15].

III. Position

This paper argues that typical ANNs act somewhat in the capacity of distance-based classifiers on instances

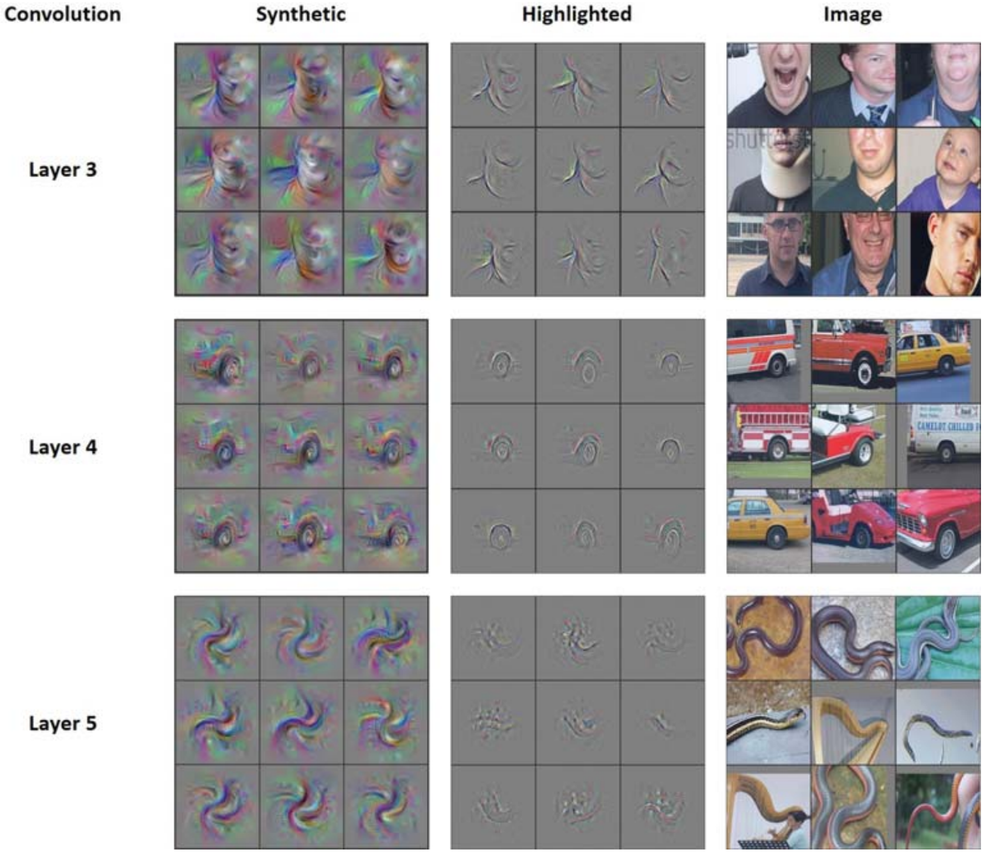

Fig. 1 Visualizations of individual artificial neurons in a deep convolutional neural network [Images selected with permission from [26] website]

and prototypes in the object recognition domain. The intention is to train a content-addressable memory that makes lookup and similarity computations efficient. Insofar as the representation captures statistical structures in the training data, automatic generalization can occur for novel inputs that fall into the right zones of the representational space. This perspective is generally supported by the observation that lazy, instance memory classifiers have comparable performance to autoencoders that focus on distributed representations on benchmarks like the MNIST digits; *k*NN L3 with no features achieves 2.83% error on the MNIST test set [28], while a two-layer Restricted Boltzmann Machine (RBM) achieves 2.49% error [12]. If distributed ANNs are simply encoding many instances into an efficient structure, then a similar performance between instance memory and representational encoding systems would be expected.

Content-addressable memory for efficient lookup and memory compression are both vitally important properties for systems that can scale to large problems. However, the type of representational encoding scheme selected conveys different trade offs. As argued through Section II, using fully distributed encoding may afford good data compression and efficient lookup but it presents an issue for veridical memory. Distortions in the representation space made by the encoding scheme may preclude judgements of familiarity and the lack of capability to represent multiple candidates prevents detection of conflicts. In contrast, a localized encoding scheme affords less data compression but still presents opportunities to make lookup efficient while retaining veridical memory. Localization allows familiarity to be judged by gauging representational activity (i.e., similarity) and the allowance for multiple competing activations allows for detection of conflict.

In line with the assertions of Complementary Learning Systems theory [5], [6], it is the position here that slowly tuned, long-term memory representation networks (i.e., neocortical) be supplemented with fast-acting instance memory. Representational transformations that emerge from fully-connected networks should be avoided as inputs to such instance memory because of their potential for distorting the representational space. Such networks perhaps are best thought of as representing automatic pathways for overlearned tasks. Compositional representations of localized features are better inputs for instance learning because they preserve the representational space, allowing for judgement of familiarity as well as preserving the capacity for solving the stability-plasticity dilemma by allowing new experiences to be encoded into free portions of memory [16].

In machine learning, typical implementations of instance memory are in the form of *k*NN (e.g., [29]). These approaches have the drawback of requiring the selection of a fixed sample size (i.e., the *k* parameter) as well as not being very biologically inspired. Localist artificial neural networks [19], such as Adaptive Resonance Theory (for an overview [30]) and SUSTAIN [31], provide biologically inspired connectionist networks that are suitable for instance memory. Unfortunately, these localist approaches tend to favour single selections for instances from memory (or representations) without explicitly monitoring for conflicts. Conflict monitoring, along with familiarity judgement are both meta-cognitive judgements that are important for the elicitation of cognitive control [4] as well an indication that memorization of new information is required.

In this paper an epistemic sampling approach within a connectionist framework is advocated for instance memory. Fig. 2 presents an illustration of the approach. Suns and stars on the figure indicate instances of different classes encoded into memory. The black dashed line indicates the true class boundary. The red dashed line represents a folding in space that could occur in distributed representation schemes due to a lack of sampling in that region of space that would remove the space between it and the true class boundary (this is an interpretation of how memory compression may occur within a distributed representation scheme). The black dots on Fig. 2 represent some new inputs to classify. The black circles around these dots represent the distance to the closest instance in memory. The blue circles around the black circles represent a additional sampling range for detecting conflicts (this sampling range is determined by a parameter labelled σ in the experiment that follows). Input number 1 demonstrates a well sampled region of space with low conflict. Input number 2 shows a mediumly well sampled region of space with high conflict. Finally, input number 3 shows an unsampled region of space with low conflict. These examples demonstrate how both conflict and familiarity (i.e., distance or sampling adequacy) are important for making reliable judgements.

The epistemic sampling approach in this paper is set within a connectionist framework. The sampling ranges are dynamically adjusted based on peak network activity (i.e., max familiarity or smallest distance), such that greater bias (smaller sampling) is elicited when input is familiar, and greater variance (larger sampling) is elicited when input is unfamiliar. This property is evident in Fig. 2 with input location 1 using a small sample due to high familiarity, and input location 3 using a large sample due to unfamiliarity. By using sampling, judgement of conflict can be performed by comparing the ratios of implications (i.e., class labels) within the selected sample. Such ratios have a conceptual link to Bayesian like (probability) judgements when the familiarity (i.e., conformity) of the sample is taken into account (e.g., see [32]). In this way the technique could present as a potential synthesis between Bayesian [33] and connectionist cognitive modelling techniques [34]. In the experiment that follows the effect of sampling range and familiarity are explored on a machine learning dataset. Conflict and familiarity are also combined into a judgement metric that is effectively utilized to judge decision confidence and guide memorization to reduce the amount of information to be stored.

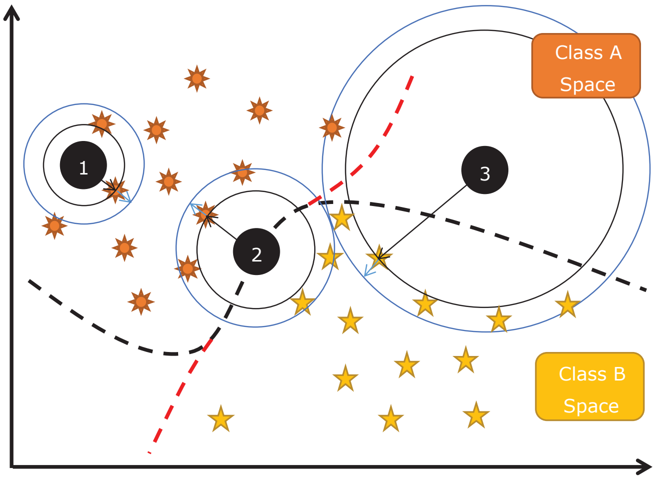

Fig. 2 Illustration of epistemic sampling in a hypothetical sample space

IV. Experiments

This section presents some experimental results demonstrating the usefulness of representational conflict on a classification task. First the effect of conflict on generalization performance and demand for encoding is established. The use of representational conflict as an indicator of judgment confidence is then demonstrated. This latter property is useful for signalling the need for supervised learning. This could present an opportunity for synergistic systems in which a classifier automatically handles clear instances and flags ambiguous instances for human intervention and labelling. This is demonstrated by using the judgement metric to request supervision only when the system is uncertain which reduces supervision requirements as well as lowering memorization demand.

A. Experimental Setup

The experiments presented adopted a distance-based classifier approach, setup as a localist ANN. The classifier was formulated in the logic of sets rather than vectors, with the focus on discrete entities and graph structure. Instances and categories were added as nodes along with all relevant links dynamically during learning. Nodes had an activation strength property that was used to determine selection.

The instance and category nodes were handled differently. Instance nodes had an activation value that was defined by a measure of similarity between the instance in memory and a new instance presented as input. This similarity function is inspired by the idea of resonance in Adaptive Resonance Theory [30], and includes the bottom-up and top-down interaction between the input and memory representation. The activation function involves the intersection between the set of data of the instance at the input, I_{input}, and the set of data for the instance held in memory, I_{memory}, proportional to the size of both sets,

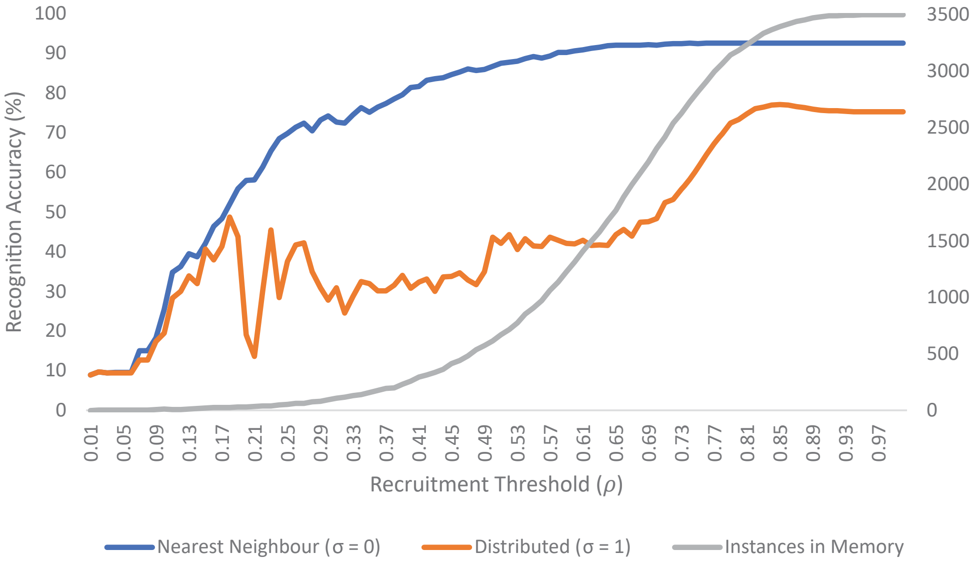

Fig. 3 Recognition accuracy (left axis) on the test set for the selective (blue) and aggregate similarity (orange) approaches: The number of instances memorized (right axis) is shown in grey

$$A_{instance} = \frac{|I_{input} \cap I_{memory}|}{|I_{input}|} \cdot \frac{|I_{input} \cap I_{memory}|}{|I_{memory}|} = \frac{|I_{input} \cap I_{memory}|^2}{|I_{input}| \cdot |I_{memory}|} \tag{1}$$

where $A_{instance}$ is the activation strength, $|I_{input} \cap I_{memory}|$ is the amount of shared data, $|I_{input}|$ is the size of the input set's data, and $|I_{memory}|$ is the size of the instance set's data. Values nearer 1.0 indicated a strong similarity, whereas values nearer 0.0 indicated little or no similarity. Similarity can be conceptualized as the distance between an input and instances in the representation space as shown in Fig. 2. Instances held links to category nodes, and selections were manipulated. A parameter, σ, determined a selection range of active instances, such that its value indicated a range from the maximum activation to be selected as shown in Fig. 2. For example, $\sigma = 0.0$ would only select instances at maximum activation, and $\sigma = 1.0$ would select all instances. Thus, the system was like *k*NN except that the number of neighbours was determined dynamically relative to the adequacy of sampling in memory. This is a simple, effective and online approach that is suitable for connectionist models. Statistical approaches to selecting sampling size are also possible (see [18]).

Category nodes were designed as hubs that aggregated the activation values of the sampled instances. In the experiments, the activation strength of a category node, $A_{category}$, was simply the sum of the activation strengths of the set of selected memory instances, $I_{selected} = \{I_1, I_2, ..., I_n\}$;

$$A_{category} = \sum_{n=1}^{I_{selected}} A_{instance} \tag{2}$$

The category node with the greatest activation was selected as the response.

The system was configured for stream learning. On every trial during training and testing, an input example was processed and compared to whatever was currently held in memory to generate a response. On training trials, when supervision was requested, feedback was provided after a response was made, allowing training trials to act as validation trials for estimates of model performance. On supervised trials, a parameter, ρ, was used to determine if the processed input should be recruited as a new node (i.e., memorized). The parameter ρ indicated a threshold of activation (i.e., similarity) in which an instance of the correct class already held in memory could be considered as equivalent to the currently processed input, and thus not requiring memorization. If no instance in memory was sufficiently active, the input was memorized along with its class label. The parameter ρ can be considered as similar to vigilance in Adaptive Resonance Theory [30], except that the system here does not adapt existing instances into prototypes.

The MNIST digits dataset was used for all experiments [28]. The benchmark performance specified on the website for *k*NN without features was 2.83% error on the test set and was considered the appropriate benchmark for these experiments when using the full training set of 60,000 instances. All nominal performances reported here were on the full test set of 10,000 instances, however for many of the demonstrations only a subset of the first 3,500 training instances were used. This was done to reduce processing time as models were run to get output under 100 different parameter values each (i.e., increments of 0.01 between 0.0 – 1.0 for the ρ and σ parameters). All instances in the dataset were discretised into coordinate point activations based on a cut-off value of 30 for each pixel. Thus, the input data and data committed to memory were a set of active coordinates within the 28×28 bounds of the original data. Inactive locations were not represented.

B. Familiarity & Sampling

In this experiment, 3,500 training instances were used with supervision on all trials, in order to explore the parameter space for ρ (recruitment/memorization) and σ (sampling range). The memory sampling range has a profound influence on encoding demand as well as generalization performance. This effect was demonstrated by manipulating the node recruitment parameter, ρ, for two models using extreme values of the sampling range ($\sigma = 0.0$, most similar only, and $\sigma = 1.0$, all instances). Fig. 3 illustrates that a selective approach produced substantially greater generalization performance for most of the ρ parameter range. The asymptotic curve indicates that near maximal performance could be achieved by encoding only around half the training set for this model. When all instances were selected, most of the training set needed to be encoded to reach its peak performance. Yet that performance was in the range of 16% more error than the selective model. Maximum performance for the two models was 7.48% error for the selective model with recruitment at $\rho = 0.77$, and 22.94% error at $\rho = 0.85$ for the aggregate model. This experimental result demonstrates how conflict in representation relates to encoding demands during learning, and to generalization performance during classification. The selection of many representations can be considered as generating conflict since the representations belong to many different classes. When there is more conflict in the system, more specific information needs to be encoded in order to overcome the effects of the competing representations at the output. By reducing conflict in the system, generalization improves and the need to encode information is also minimized.

The node recruitment value of $\rho = 0.85$ was used to investigate the full range of σ sample range values on a training set of 3,500. As depicted in Fig. 4, a small sample range produced the best generalization under these settings, before quickly dropping off with larger samples. This likely indicates a protective effect against outlier instances in memory when selecting a small range of options, as is common in *k*NN approaches. However, the overall trend indicates that reducing conflict in representations leads to better generalization performance.

A set of parameter values for selection, $\sigma = 0.09$, and recruitment, $\rho = 0.64$, were chosen as a good representative trade-off between generalization performance and encoding requirements. These were selected by comparing training set performances under different parameter settings. When run on the full training set of 60,000 examples it yielded a test set performance of 2.63% error with 24.12% (14,470 instances) of the training set encoded. Setting the contrast cut-off value to 70 (instead of 30) produces a test error of 2.38% with 34.47% encoded. These error rates are comparable to two-layer RBM (distributed representation network) performance of 2.49% [12], and the *k*NN benchmark performance of 2.83% error [28]. Given the crudeness of selection mechanisms for selecting information for encoding in the current scheme, such similarity in performance appears to justify the view that hierarchical representations in distributed representation networks are acting like a content-addressable memory.

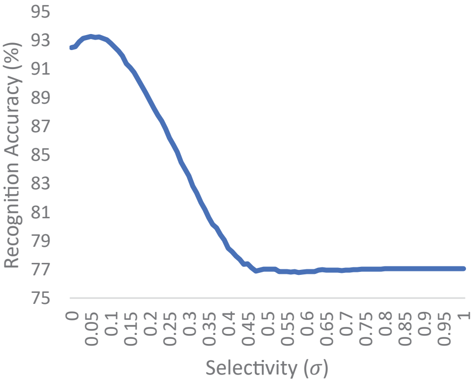

Fig. 4 The relationship between representational selectivity (σ) and recognition accuracy on the test set

C. Conflict & Confidence

The relationship between conflict and classification performance was explored descriptively. Using the balanced trade-off parameters of $\sigma = 0.09$ and $\rho = 0.64$ on a fully supervised training set of 60,000, it was found that significantly more instances were selected on incorrect trials ($M = 15.98 \pm 1.59$: 95% confidence interval) than correct trials ($M = 10.07 \pm 0.17$). More classes were represented in the selections of the incorrect trials ($M = 3.46 \pm 0.19$: 95% confidence interval) than on correct trials ($M = 1.61 \pm 0.02$), and the average activation value (i.e., similarity) was significantly weaker on incorrect trials ($M = 0.58 \pm 0.01$: 95% confidence interval) than on correct trials ($M = 0.65 \pm 0.001$). All these indicators can be combined to indicate judgement confidence. For example, a confidence metric, C, was tested using the following

$$C = A_{average} \cdot \frac{|S_{max}|}{|S|} \tag{3}$$

where $A_{average}$ is the average activation strength of the selected instances, $|S_{max}|$ is the maximum number of instances selected in a single class, and $|S|$ is the total number of instances selected across all classes. This metric indicates confidence when selected instances have high similarity to the input and are disproportionately within a single class in memory. By applying a confidence threshold, it can be seen in Fig. 5 that a low error rate of 0.60% could be obtained while responding to 90.54% of the test set. This result appears consistent with work that uses statistical confidence to pick the number of neighbours in *k*NN [18]. This metric requires validation on additional datasets.

The confidence metric can be used to judge certainty in a classification response, and thus also as a signal to request supervisory input. Using the full training set of 60,000 examples, the system was set to request supervision

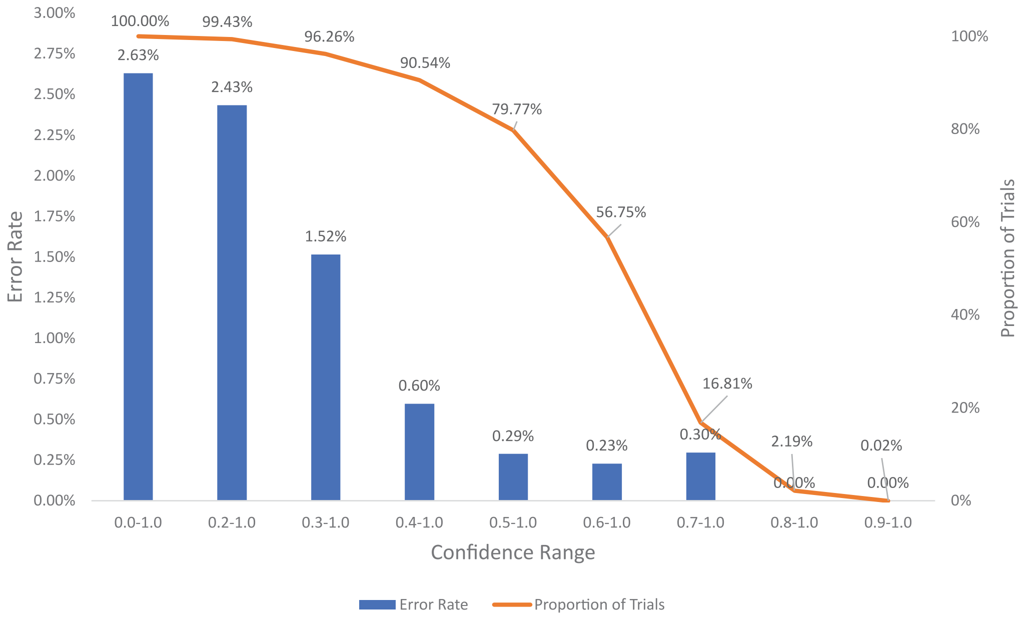

Fig. 5 Judgment confidence effect on test error (blue, left axis) and proportion of test set trials (orange, right axis): At a confidence threshold of 0.5, a low test-error of 0.60% can be obtained on a substantial proportion of trials (90.54%)

only if confidence was $C < 0.4$. The balanced parameters of $\sigma = 0.09$ and $\rho = 0.64$ were again used. In this configuration memorization only occurred when classification judgement lacked confidence and there was no suitably active instance in memory with the correct class label. The resulting system requested supervision for 10,999 examples (18.33%) of the 60,000 example training set. The system memorized a total of 7,548 examples or 52.16% of the 14,470 examples memorized when all trials were supervised. These reductions to supervisory input and memory demand came with only minimal detriment to classification performance. The system achieved an error rate of 2.78% on the test set, compared to 2.63% with full supervision. The system responded confidently on 84.89% of test trials with an error rate of only 0.48% on these trials. The remaining 15.11% of unconfident trials had an error rate of 15.22%. This result demonstrates that the confidence metric is effective in separating low and high error rate trials during performance.

V. Discussion

While high levels of conflict are generally detrimental to classification performance, allowing some level of conflict may be advantageous in terms of internal monitoring. Selecting a range of representations indicates input instances that are ambiguous when the representations have conflicting implications. The degree of conflict between selected representations can be used as a metric of judgement confidence, in line with suggestions from recent literature [15]. The inclusion of indicators of conflict and confidence in classification systems conveys some attractive properties. Firstly, it provides a mechanism for synergistic machine learning systems in applied settings. After some initial training, operationally sound confidence levels could be set, and clear cases of recognition could be automated. Ambiguous cases could then be flagged for human intervention, which in turn can act as supervised training data for improving the machine learning system.

The monitoring of conflict in representation is also important as it suggests developing conflict resolution mechanisms into ANNs. For example, hierarchical re-entrant architecture has long proven to be effective in using contextual information to resolve conflicts in cases of ambiguity such as visual occlusions [24]. Such architectures are popular in cognitive science for explaining vision [35] and tie into broader concepts of interactions between bottom-up and top-down sources of attention, such as Adaptive Resonance [30]. In a computer vision setting, it is seen as the interaction between discriminative and generative approaches [36].

Conflict signals can also be used to drive learning algorithms. Intuitively, contrasts between conflicting instances in memory can draw out discriminative features that either indicate inclusion or exclusion from a given class (or representational unit). Such contrasts are being pursued in supervised contrastive learning [37], however it is unclear to what extent difficult edge cases are captured using current approaches. Conflict and contrasts may also present opportunities to move away from optimization driven learning algorithms. Learning algorithms could instead focus on the principle of information encoding. Conceptually, this would be a more constructive process of finding free memory traces than optimization. This may reduce the need to process training sets multiple times, and open opportunities for extendable stream learning systems within an ANN framework.

VI. Conclusion

This paper provided an interpretation of ANN problem solving logic on classification problems. In a nutshell, input data are matched to fragments of instances that were encoded from the training set. Similar fragments are stored together in representational units or regions of vector space. This captures statistical variances and allows for data compression. Combining fragments across successive layers allows for neat separation into classes at the output. The networks thus act as a content-addressable memory for training instances and their mappings to classes. The degree of localization (i.e., convolutional filters) or distribution (i.e., fully connected layers) appears to affect the degree to which veridical memory can be supported.

The paper also presented results from experiments that showed a relationship between representational conflicts, generalization performance, and encoding requirements in an instance memory network. Selecting small samples from memory was found to improve generalization performance, and reduce memorization requirement. The experiment also showed how monitoring conflict could be used to measure confidence that could be used to control error rates in responses. By using judgement confidence to guide supervised learning, supervisory input was substantially reduced as were the number of instances requiring memorization with only minimal detrimental effects on classification performance.

Acknowledgment

This research was supported by an Australian Government Research Training Program Scholarship.

References

[1] Y. LeCun, Y. Bengio, and G. Hinton, "Deep learning," *Nature*, vol. 521, no. 7553, pp. 436–444, 2015.

[2] G. E. Hinton, D. E. Rumelhart, and J. L. McClelland, *Distributed Representations*. MITP, 1986, pp. 77–109.

[3] I. J. Goodfellow, J. Shlens, and C. Szegedy, "Explaining and harnessing adversarial examples," *arXiv preprint arXiv:1412.6572*, 2014.

[4] M. M. Botvinick, T. S. Braver, D. M. Barch, C. S. Carter, and J. D. Cohen, "Conflict monitoring and cognitive control," *Psychological Review*, vol. 108, no. 3, pp. 624–652, 2001.

[5] D. Kumaran, D. Hassabis, and J. L. McClelland, "What learning systems do intelligent agents need? complementary learning systems theory updated," *Trends in Cognitive Sciences*, vol. 20, no. 7, pp. 512–534, 2016.

[6] J. L. McClelland, B. L. McNaughton, and R. C. O'Reilly, "Why there are complementary learning systems in the hippocampus and neocortex: Insights from the successes and failures of connectionist models of learning and memory," *Psychological Review*, vol. 102, no. 3, pp. 419–457, 1995.

[7] M. M. Botvinick, S. Ritter, J. X. Wang, Z. Kurth-Nelson, C. Blundell, and D. Hassabis, "Reinforcement learning, fast and slow," *Trends in Cognitive Sciences*, vol. 23, no. 5, pp. 408–422, 2019.

[8] J. L. McClelland and D. E. Rumelhart, "Distributed memory and the representation of general and specific information," *Journal of Experimental Psychology: General*, vol. 114, no. 2, pp. 159–188, 1985.

[9] C. Zhang, S. Bengio, M. Hardt, B. Recht, and O. Vinyals, "Understanding deep learning requires rethinking generalization," *arXiv preprint arXiv:1611.03530*, 2016.

[10] ——, "Understanding deep learning (still) requires rethinking generalization," *Commun. ACM*, vol. 64, no. 3, p. 107–115, 2021.

[11] D. Arpit, S. Jastrzebski, N. Ballas, D. Krueger, E. Bengio, M. S. Kanwal, T. Maharaj, A. Fischer, A. Courville, and Y. Bengio, "A closer look at memorization in deep networks," in *International Conference on Machine Learning*. PMLR, Conference Proceedings, pp. 233–242.

[12] G. E. Hinton, "What kind of graphical model is the brain?" in *Proc. 19th International Joint Conference on Artificial intelligence*, vol. 5, 2005, Conference Proceedings, pp. 1765–1775.

[13] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, "Learning representations by back-propagating errors," *Nature*, vol. 323, no. 6088, pp. 533–536, 1986.

[14] T. Chen, S. Kornblith, M. Norouzi, and G. Hinton, "A simple framework for contrastive learning of visual representations," *arXiv preprint arXiv:2002.05709*, 2020.

[15] N. Papernot and P. McDaniel, "Deep k-nearest neighbors: Towards confident, interpretable and robust deep learning," *arXiv preprint arXiv:1803.04765*, 2018.

[16] S. Grossberg, "How does a brain build a cognitive code?" *Psychological Review*, vol. 87, no. 1, pp. 1–51, 1980.

[17] Z. Ghahramani, "Probabilistic machine learning and artificial intelligence," *Nature*, vol. 521, no. 7553, pp. 452–459, 2015.

[18] J. Wang, P. Neskovic, and L. N. Cooper, "Neighborhood size selection in the k-nearest-neighbor rule using statistical confidence," *Pattern Recognition*, vol. 39, no. 3, pp. 417–423, 2006.

[19] M. Page, "Connectionist modelling in psychology: A localist manifesto," *Behavioral and Brain Sciences*, vol. 23, no. 4, pp. 443–467, 2000.

[20] G. Dong and H. Liu, *Feature engineering for machine learning and data analytics*. CRC Press, 2018.

[21] J. Yosinski, J. Clune, A. Nguyen, T. Fuchs, and H. Lipson, "Understanding neural networks through deep visualization," *arXiv preprint arXiv:1506.06579*, 2015.

[22] J. S. Bowers, "Parallel distributed processing theory in the age of deep networks," *Trends in Cognitive Sciences*, vol. 21, no. 12, pp. 950–961, 2017.

[23] J. Grainger and A. M. Jacobs, *On localist connectionism and psychological science*. Mahwah, New Jersey: Lawrence Erlbaum, 1998, pp. 1–38.

[24] J. L. McClelland and D. E. Rumelhart, "An interactive activation model of context effects in letter perception: I. An account of basic findings," *Psychological review*, vol. 88, no. 5, pp. 375–407, 1981.

[25] D. A. Norman and T. Shallice, *Attention to Action: Willed and Automatic Control of Behavior*. Boston, MA: Springer US, 1986, pp. 1–18.

[26] J. Yosinski, "Understanding neural networks through deep visualization," 2015, accessed: 27-01-2021. [Online]. Available: http://yosinski.com/deepvis

[27] J. Yosinski, J. Clune, Y. Bengio, and H. Lipson, "How transferable are features in deep neural networks?" in *Advances in neural information processing systems 27 (NIPS 2014)*, Z. Ghahramani, M. Welling, C. Cortes, N. D. Lawrence, and K. Q. Weinberger, Eds. Curran Associates, 2014, Conference Proceedings, pp. 3320–3328.

[28] Y. LeCun, C. Cortes, and C. J. C. Burges, "The MNIST database," accessed: 03-06-2020. [Online]. Available: http://yann.lecun.com/exdb/mnist/

[29] A. Pritzel, B. Uria, S. Srinivasan, A. P. Badia, O. Vinyals, D. Hassabis, D. Wierstra, and C. Blundell, "Neural episodic control," in *Proceedings of the 34th International Conference on Machine Learning-Volume 70*. JMLR. org, Conference Proceedings, pp. 2827–2836.

[30] S. Grossberg, "Adaptive resonance theory: How a brain learns to consciously attend, learn, and recognize a changing world," *Neural Networks*, vol. 37, pp. 1–47, 2013.

[31] B. C. Love, D. L. Medin, and T. M. Gureckis, "Sustain: A network model of category learning," *Psychological Review*, vol. 111, no. 2, pp. 309–332, 2004.

[32] G. Shafer and V. Vovk, "A tutorial on conformal prediction," *Journal of Machine Learning Research*, vol. 9, no. 3, 2008.

[33] T. L. Griffiths, N. Chater, C. Kemp, A. Perfors, and J. B. Tenenbaum, "Probabilistic models of cognition: exploring representations and inductive biases," *Trends in Cognitive Sciences*, vol. 14, no. 8, pp. 357–364, 2010.

[34] J. L. McClelland, M. M. Botvinick, D. C. Noelle, D. C. Plaut, T. T. Rogers, M. S. Seidenberg, and L. B. Smith, "Letting structure emerge: connectionist and dynamical systems approaches to cognition," *Trends in Cognitive Sciences*, vol. 14, no. 8, pp. 348–356, 2010.

[35] V. Di Lollo, "The feature-binding problem is an ill-posed problem," *Trends in Cognitive Sciences*, vol. 16, no. 6, pp. 317–321, 2012.

[36] Z. Tu, X. Chen, A. L. Yuille, and S.-C. Zhu, "Image parsing: Unifying segmentation, detection, and recognition," *International Journal of computer vision*, vol. 63, no. 2, pp. 113–140, 2005.

[37] P. Khosla, P. Teterwak, C. Wang, A. Sarna, Y. Tian, P. Isola, A. Maschinot, C. Liu, and D. Krishnan, "Supervised contrastive learning," *arXiv preprint arXiv:2004.11362*, 2020.