

Reconstructing Abstract Concepts and their Blends Via Computational Cognitive Modeling

Rahul Sharma, Bernardete Ribeiro, Alexandre Miguel Pinto, and F. Amílcar Cardoso

Department of Informatics Engineering, University of Coimbra, Portugal.

E-mail:[rahul|brebeiro|ampinto|amilcar] [at] [dei] [dot] [uc] [dot] [pt]

Abstract—Concept Blending is one of the most prominent computational approaches to study and understand the underlying processes related to creativity. In this article, we show how to use the Regulated Activation Network (RAN) cognitive model to reconstruct abstract concepts and their blends. The MNIST dataset is used in this work to build a representation of abstract concepts. For the demonstration, three experiments were designed: first, shows how a high dimensional input image is encoded into a low dimension vector and further reconstructed back into an image; second, reconstruction of blends of abstract concepts that represent same digits; third, reconstructing blends of abstract concepts which represent different digits. The reconstructed images in all three experiments were visually analyzed. The best reconstructions were observed with the encoded image experiment obtaining Mean Squared Error of 0.00562 and an R-square score of 0.9193. The blends of similar abstract concepts also reconstructed the expected blend of a digit. The blends of dissimilar abstract concepts reconstructed the images by creating interesting symbols such as character x .

Index Terms—Computational Creativity, Concept Blending, Dimension Reduction, Concept Reconstruction, Computational Modeling

I. INTRODUCTION

Computational creativity can be defined as “*The philosophy, science, and engineering of computational systems which, by taking on particular responsibilities, exhibit behaviors that unbiased observers would deem to be creative*” [1]. Though computational creativity is a sub-field of Artificial Intelligence (AI), there is a significant difference between the traditional AI techniques and computational creativity with respect to the *problem* domain. For example, creating a new recipe is a problem hardly addressable with traditional AI approaches, but AI approaches are being used to realize computational creative objectives, like the Painting Fool project¹ makes use of machine learning to predict the similarity of two abstract images [2].

There has been noticeable progress in the field of computational creativity. MuzaCazUza [3] and IDyOM [3] are examples of music generator, whereas, COLIBRI [4] is a poetry generator, Copycat [5], Metacat [6], and Magnificat [7] are some examples of computationally creative models. There have been more than twenty creative systems been introduced until now capable of blending concepts, performing analogy [1] and so on. Chapter 2 of [8] provide a list of creative models and categorized them into four types of models:

Systems model, Evolutionary model, Domain-centered model, and Cognition-centered model. Evaluation of computationally creative systems is an ongoing issue, but with the involvement of human evaluation many systems have been validated, for example, theorems produced by the HR discovery system have been published in mathematical literature [9].

Combinatorial creativity is a category of creativity [10]. In this approach, the related (or distinct) concepts (or feature) are blended to produce a new concept. In essence, this strategy enables us to model creativity by searching for novelty in a space made up of probable combinations of concepts. In this article, we use a computational cognitive model named Regulated Activation Network (RAN) [11] to demonstrate the reconstruction of abstract concepts and their blends. The RAN is a hybrid modeling technique that dynamically builds a representation of abstract concepts both convex [12] and non-convex [13]. For this article’s demonstrations we use the convex abstract concept modeling with RAN [14] and MNIST [15] standard datasets. The reconstruction operation is performed by two alternating iterative processes, i.e., the upward and downward activation propagation mechanisms. The outcome of the reconstruction operation is evaluated using based upon the visual analogical reasoning of the reconstructed images. The reconstruction of encoded images is also evaluated using Mean Squared Error and R-square metrics.

The article is organized as follows: Section II puts forward the state of the art related to concept blending; Section III describes the convex abstract concept modeling and their process of reconstruction using MNIST dataset; Section IV reports the reconstruction experiments and discussions; the article ends with Section V providing conclusions and future work.

II. RELATED WORK

Conceptual Blending has a significant contribution to the field of computational creation of visuals and art. One of the works of conceptual blending is found as BLENDER [16], to blend images (logo made up of lines). The knowledge base of BLENDER was made manually and was capable of considering superficial knowledge, like identifying objects. To test BLENDER, it was subjected to combine concepts “boat” and “house”, all possible blends were generated exhaustively. Having its motivation from BLENDER and conceptual blending theory, a framework named DIVAGO [17] was proposed. Initially, DIVAGO was just a concept generator, generating

¹www.thepaintingfool.com

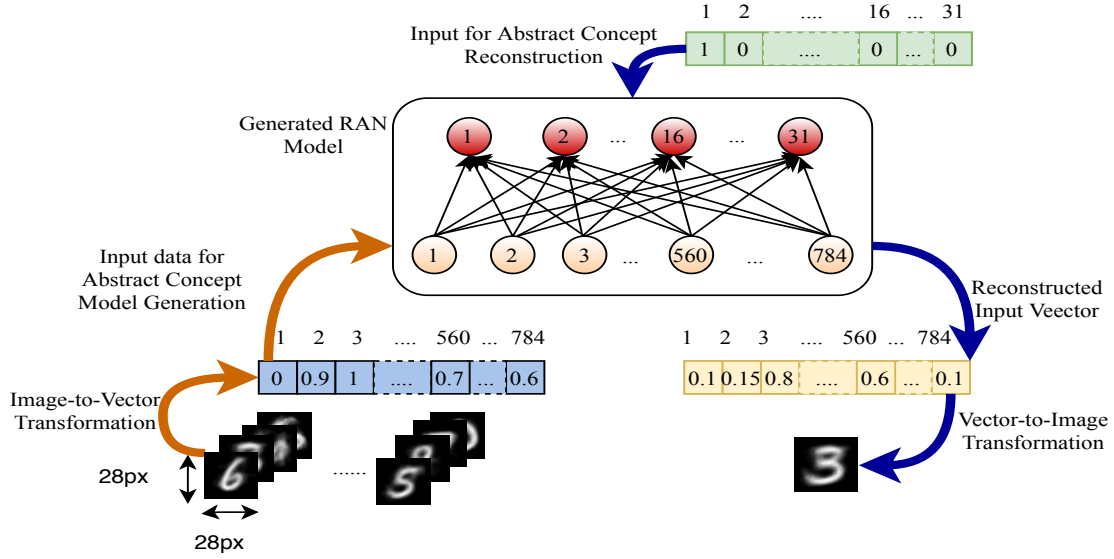


Fig. 1. Modeling and Reconstruction with RAN

concepts out of previous knowledge and later introduced an implementation of conceptual blending theory for both linguistic and visual domains.

A framework dedicated to formalizing the entire process of conceptual blending for the visual domain was introduced in [18]. The goal of the system was to generate images deemed to be creative. An interesting work tried to obtain visual blends of the images of *pig*, *angel*, and *cactus* [19]. A visual blending based system for Emogi generation combined the data from ConceptNet, EmogiNet, and Twitter Twemoji to generate visual blends of concepts [20]. In this article, we demonstrate how we can produce blends of abstract concepts using a computation cognitive model RAN. The blends are obtained by reconstruction operation of RAN's modeling where the geometrical associations among the abstract concepts are utilized to obtain their respective input level values.

III. THE RAN'S METHODOLOGY

In this section, we describe how convex abstract concepts are modeled using the RAN's methodology, further, we detail the RAN's abstract concept reconstruction process. Figure 1 shows an overview of RAN's modeling and reconstruction process. We used MNIST data in the demonstrations for visually aided analogical evaluation of the reconstructed images especially for the experiments described in Sections IV-B and IV-C. To build a model with RAN the data need to be pre-processed. In the MNIST dataset, each data point is in a greyscale i.e. $[0, 255]$ pixel values we first normalize the pixel values between $[0, 1]$ using min-max normalization. The original data is 28×28 images which were converted into a vector of size 784, as shown in Figure 1. The processed data is used to build a two-layered model using the model generation process described in Section III-A. The reconstruction operation is performed by using the generated RAN model

in a top-down fashion to propagate activation from a higher layer to the input layer. At last, once the activation vector is reconstructed at the input layer it is further transformed to a 28×28 image.

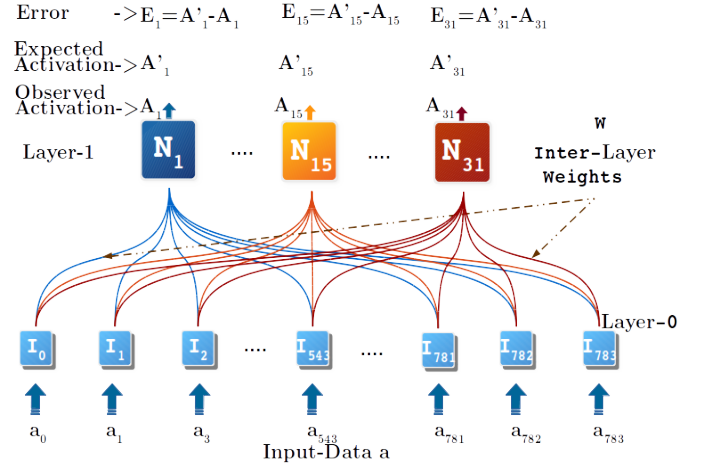


Fig. 2. Two layered RAN's model with MNIST dataset

A. Model Generation

Once the data is processed the model generation takes place, at first the input Layer-0 is created with 784 nodes as shown in Figure 2, then the following three steps are applied:

- **Step 1 Concept Identification:** In this process, we use the K-means clustering algorithm as a concept identifier initialize its K with 31 to identify 31 clusters in the data. In MNIST data there are images of handwritten digits between $[0, 9]$. By selecting $K=31$ we intend to identify one or more clusters representing each digit. The K-

means algorithm also returns the centroids of the cluster which are used as weights as described in Step 3.

- **Step 2 Concept Creation:** This process simulated the cognitive process of concept creation where a new Layer-1 is created having 31 nodes. These 31 nodes act as the abstract representative of the 31 clusters identified in the concept identification process (Step 1). In Figure 2 the Layer-1 shows the abstract concept layer with 31 nodes.
- **Step 3 Inter-Layer Learning:** In this step, we establish a relation between the input Layer-0 and abstract concept Layer-1. Since each abstract concept node at Layer-1 is representative all 31 clusters identified in Step-1, therefore, the centroids of the clusters are assigned as the inter-layer weight between the two layers. The Equation 1 show the generalized form of the equation where $m=784$ (size of Layer-0) and $n=31$ (size of Layer-1).

$$W = \begin{bmatrix} W_{1,1}, W_{1,2}, \dots, W_{1,m} \\ \dots \\ W_{i,1}, W_{i,2}, \dots, W_{i,m} \\ \dots \\ W_{n,1}, W_{n,2}, \dots, W_{n,m} \end{bmatrix} = \begin{bmatrix} C_1 \\ \dots \\ C_i \\ \dots \\ C_n \end{bmatrix} \quad (1)$$

Since the inter-layer weights W are one of the centroids in the input data, therefore, we re-transform these weights into their 28×28 images. Table I shows which weight W_n is transformed to what digit. Here we can see that the digit 0 is linking 4 nodes N_5, N_{10}, N_{20} and N_{23} therefore we can say that these 4 nodes are the symbolic abstract representatives of digit 0. Similarly, all the other nodes are attached to the symbolic representation of their respective digits.

B. Activation Propagation

After completing the 3 steps a two-layered model is obtained as shown by Figure 2. In RAN's modeling there two activation propagation mechanisms: first, to propagate activation from input Layer-0 to abstract concept Layer-1; second, to propagate activation downward from abstract concept Layer-1 to input Layer-0. The downward propagation operation is used for reconstruction operation.

Upward Activation Propagation: This is a process to propagate activation from Layer-0 to Layer-1. This is performed by calculating a Euclidean distance d_n between the input activation a_m and a weight vector in weight matrix W , as shown by Equation 2.
































$$d_n = \sqrt{\sum_{i=1}^m (W_{n,i} - a_i)^2} \quad (2)$$

The distance values obtained are then normalized using the Equation 3

$$D_n = \frac{d_n}{\sqrt{m}} \quad (3)$$

This normalized distance D_n is then passed to a non-linear function $f(x)$ (see Equation 4) to convert the distance

TABLE I
CLUSTER CENTERS REPRESENTING NODES IN LAYER-1

					
(1)	(2)	(3)	(4)	(5)	(6)
					
(7)	(8)	(9)	(10)	(11)	(12)
					
(13)	(14)	(15)	(16)	(17)	(18)
					
(19)	(20)	(21)	(22)	(23)	(24)
					
(25)	(26)	(27)	(28)	(29)	(30)
					
(31)					

values into similarity values which conforms to the following requirements: $f(0) = 1$; $f(1) = 0$; and $f(x)$ is continuous, monotonous, and differentiable in the $[0, 1]$ interval. Besides converting the distance values into similarity measure, Equation 4 also serves as a quashing function which steeply suppresses the activation towards zero with a small decrease in the activation value (for elaboration see Figure 5 in [12]).

$$f(x) = (1 - \sqrt[3]{x})^2 \quad (4)$$

Indeed, this upwards propagation is a geometric reasoning operation: a non-linear projection of a point in a m -dimensional space represented by layer l into a n -dimensional one, represented by layer $l + 1$ having n nodes. With MNIST data when an input data vector of size 784 is propagated upward we obtain a vector A_n of size 31 at Layer-1 and the node having the highest activation will be representing the input vector. As most data points will have a non-maximal distance to every centroid, all nodes in the upper layer will have some non-zero activation, although, for each input datum, the majority will have almost zero activation, except the node corresponding to the closest centroid.

Downward Activation Propagation: This operation is to used to obtain activation at input Layer-0 based upon the expected activation A'_n at Layer-1. The operation involves an iterative process of (1) upward activation operation; (2) error calculation; and (3) adjusting the input based upon the error. In the downward propagation, first, the input vector a_m is initialized with zeros. The expected activation vector A'_n is also initialized with the high activation value (usually 1) at one node and rest of the nodes are assigned lower activation values (usually 0 or less than 0.4).

The iteration begins by propagating the initialized input vector to Layer-1 to obtain the vector A_n at Layer-1. Then the error is calculated using the Equation 5 where a difference between the calculated between the expected activation vector A'_n and observed activation vector A_n .

$$E_n = A'_n - A_n \quad (5)$$

We use these individual errors to adjust the activation a_m of each node m in Layer-0 using $\Delta_{a_m} = \frac{\sum_1^n \Delta_{a_m, A_n}}{\#n}$ where $\Delta_{a_m, A_n} = (W_{n,m} - a_m) * (E_n)$. The overall impact of a_m on all A_n is summed together and normalized by dividing with maximum possible impact i.e. $\#n$. Finally, the error correction at node m of Layer-0 is calculated using Equation 6. The process is repeated for 500 iterations (t) and the activation values at input Layer-0 nodes a_m after the 500th iteration is considered as the reconstructed value of the expected activation A'_n .

$$a_m^{t+1} = \begin{cases} a_m^t + \Delta_{a_m} * (1 - a_m^t), & \text{if } \Delta_{a_m} > 0 \\ a_m^t + \Delta_{a_m} * (a_m^t), & \text{otherwise} \end{cases} \quad (6)$$

where the iteration $t = 1, \dots, 499$.

Usually, the values of Δ_{a_m} is in range $[-1, 1]$ and values of a_m^t is in range $[0, 1]$. When $\Delta_{a_m} > 0$ then the current activation a_m^t inversely impacts the update of activation. For e.g., no activation is added to a_m^t if a_m^t is already 1, and complete Δ_{a_m} is added to a_m^t if a_m^t is 0. When $\Delta_{a_m} \leq 0$ then the current activation a_m^t directly impacts the update of activation. For e.g., complete activation is added to a_m^t if a_m^t is 1, and no Δ_{a_m} is added to a_m^t if a_m^t is 0. Apart from upgrading the activations of nodes at Layer-0, the Equation 6 also helps to keeps a check on the upper and lower limits (i.e., 1 and 0 respectively) of the activation value at a node.

IV. EXPERIMENTS

In this article, we show three types of experiments using the two-layered model generated through RAN as depicted by Figure 2: reconstructing the encoded image inputs; reconstructing the blend of similar abstract concepts; reconstructing blends of dis-similar abstract concepts.

A. Reconstructing the encoded image inputs

In this experiment, we show how the reconstruction of an image is obtained when we propagated down the activation A_n vector observed by propagating upward an input image vector a_m . In Figure 2 we can observe that the size of input Layer-0 is 784 and the size of the abstract concept Layer-1 is 31. We can also say that we reduce the dimension of an input vector from 784 to 31, or encoding the a_m into A_n . Now, using the RAN's downward propagation operation we reconstruct this encoded A_n vector back to its input vector a_m .

In this experiment, first we used 10 centroid $[0, 1, 2, 3, 4, 6, 7, 8, 14, \text{ and } 15]$ images (see Table I) as input to the RAN's model. These centroids are chosen in such a way all ten digits are represented by the centroids. Since these are the

centroids it is expected to observe an activation value of 1 at a node that represents the centroid's cluster. As per expectation, we obtain the activation vectors as listed in Table II. Each A_n vector in Table II has at least one activation value equal to 1. These A_n vectors are the encoded version of the input vector a_m . We commence the reconstruction of the encoded activation by assigning the A_n as expected activation A'_n . We propagate downward each activation vector of Table II separately and obtain their respective input vectors as shown in Table III. We can see that the reconstructed images are almost identical therefore analogically we can conclude that the encoded images are reconstructed. Since we already have a reference input image, therefore, we also calculated the Mean Squared Error (MSE) and R-squared (R^2) score between the original input image and the reconstructed input image. We observed an average MSE of 0.00562 (ca.) and R^2 of 0.9193 (ca.) which is not only satisfactory but also commensurate with the analogical inferences made earlier.

B. Reconstructing Blends of Similar Abstract Concepts

In this experiment, we show how a blend of similar abstract concepts can be reconstructed using RAN's model of MNIST data. In Figure 2 similar abstract are those nodes in Layer-1 that represent same digit, for e.g., Nodes $N_5, N_{10}, N_{20}, N_{23}$ represent digit 0. To obtain the blend of similar abstract concept we create a binary expected activation vector A'_n of size 31 and assign activation value 1 to the nodes representing the same digit and 0 to other digits, e.g. $[0, 0, 0, 0, 1, 0, 0, 0, 0, 1, 0, 0, 0, 0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0]$ to blend zeros. The hypothesis of this experiment was if we try to blend abstract concepts of similar digit we are supposed to obtain an image of the digit that is represented by the abstract concepts being blended.

For these experiments, we created 10 binary expected activation vector A'_n for all 10 digits and performed the reconstruction operation for each vector at a time. The 10 reconstructed blends are listed in Table IV. By visually analyzing the images we conclude that the blends of the digits are indeed being reconstructed using the RAN's downward propagation operation. This also proves the hypothesis made for this experiment. The reconstructed image obtained in this experiment are recognizable but has a lot of noise. The best reconstruction is of digit 0 and the worst reconstruction of digit 9.

C. Reconstructing Blends of Dis-similar Abstract Concepts

In Figure 2 two abstract concepts are dis-similar if the abstract concept nodes are represent two or more different digits. This experiment is similar to the reconstruction of similar abstract concepts. The only difference is that, all the abstract concepts (digit representative nodes) that are being blended are initialized with 1 and rest are assigned 0 in the expected activation vector A'_n . For e.g., the A'_n to blend digit 2 and 5 is $[0, 0, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 1, 0, 0, 1, 0, 0, 0, 0, 0, 0, 1, 0, 0, 0]$.

TABLE II
OBSERVED ACTIVATION VALUES OF THE UPWARD PROPAGATED INPUT IMAGE

Tabel-I Image	Observed Activation (A_n)																														
4	0.31	0.32	0.31	0.24	1.00	0.27	0.24	0.28	0.27	0.39	0.24	0.28	0.28	0.24	0.23	0.28	0.25	0.26	0.33	0.30	0.32	0.24	0.38	0.23	0.29	0.33	0.28	0.28	0.25	0.30	0.23
14	0.25	0.34	0.33	0.38	0.23	0.30	0.38	0.38	0.45	0.27	0.41	0.38	0.32	0.33	1.00	0.36	0.31	0.33	0.32	0.29	0.33	0.32	0.26	0.39	0.44	0.34	0.35	0.38	0.39	0.38	0.51
15	0.39	0.37	0.33	0.39	0.28	0.29	0.37	0.34	0.36	0.31	0.35	0.37	0.33	0.32	0.36	1.00	0.40	0.32	0.37	0.29	0.36	0.38	0.27	0.34	0.44	0.33	0.39	0.42	0.32	0.36	0.35
2	0.32	0.29	1.00	0.36	0.31	0.30	0.35	0.45	0.41	0.38	0.32	0.47	0.32	0.31	0.33	0.33	0.33	0.32	0.38	0.35	0.46	0.33	0.33	0.31	0.39	0.47	0.33	0.34	0.32	0.44	0.31
1	0.39	1.00	0.29	0.33	0.32	0.33	0.34	0.30	0.32	0.32	0.33	0.31	0.45	0.38	0.34	0.37	0.37	0.34	0.38	0.30	0.32	0.35	0.30	0.30	0.34	0.34	0.43	0.35	0.34	0.34	0.32
7	0.28	0.30	0.45	0.47	0.28	0.37	0.43	1.00	0.48	0.33	0.44	0.45	0.37	0.41	0.38	0.34	0.35	0.37	0.42	0.42	0.35	0.40	0.36	0.39	0.40	0.42	0.35	0.36	0.42	0.45	0.38
0	1.00	0.39	0.32	0.33	0.31	0.27	0.31	0.28	0.29	0.32	0.27	0.29	0.34	0.32	0.25	0.39	0.43	0.28	0.33	0.27	0.35	0.37	0.27	0.27	0.32	0.32	0.35	0.30	0.27	0.32	0.25
3	0.33	0.33	0.36	1.00	0.24	0.32	0.40	0.47	0.42	0.30	0.42	0.40	0.35	0.37	0.38	0.39	0.40	0.34	0.39	0.36	0.34	0.50	0.29	0.39	0.40	0.39	0.36	0.37	0.36	0.39	0.39
8	0.29	0.32	0.41	0.42	0.27	0.35	0.41	0.48	1.00	0.32	0.43	0.40	0.36	0.43	0.45	0.36	0.34	0.34	0.37	0.33	0.35	0.37	0.30	0.38	0.43	0.39	0.33	0.39	0.42	0.48	0.40
6	0.31	0.34	0.35	0.40	0.24	0.42	1.00	0.43	0.41	0.30	0.42	0.41	0.44	0.47	0.38	0.37	0.39	0.50	0.32	0.30	0.37	0.37	0.27	0.42	0.37	0.36	0.43	0.30	0.43	0.45	0.40

TABLE III
OBSERVATIONS OF RECONSTRUCTION OF ENCODED CETROIDS

Table-I Image	Original Image	Reconstructed Image	MSE	R ²
4			0.00626	0.93990
14			0.00647	0.86235
15			0.00551	0.92837
2			0.00518	0.93690
1			0.00550	0.90858
7			0.00491	0.91735
0			0.00584	0.94753
3			0.00606	0.91568
8			0.00512	0.92725
6			0.00536	0.90914
Average			0.00562	0.91930

MSE: Means Squared Error; R²: R-squared Score

TABLE IV
RECONSTRUCTED IMAGES OF THE BLENDS OF ABSTRACT CONCEPTS

Blend Of	Reconstructed Image	Blend Of	Reconstructed Image	Blend Of	Reconstructed Image
0's		1's		2's	
3's		4's		5's	
6's		7's		8's	
9's					

reconstructed images of the blends of similar abstract concepts. The blend of 8's and 3's reconstructed an image more looking like digit 3's. The blend of 3's and 5's looked like a character in the Hindi language as shown in Figure 3. Blend of 2's and 5's looks like character x . The blend of 4's and 9's is some arbitrary reconstruction. With these experiments, we infer that we can produce blends of abstract concepts using RAN's downward propagation operation to recreate new combinations at the input layer. This reconstruction operation of blending unrelated abstract concepts can be deemed creative if the resultant image has some meaning to an evaluator as in the case of the blend of 2's and 5's where we obtain the character x .







Fig. 3. Hindi character

For this experiment, we created 4 A'_n vectors of size 31 representing the blends of digits 8's & 3's, 3's & 5's, 2's & 5's, and 4's & 9's. The hypothesis of this experiment was that the reconstructed images obtained with these experiments an arbitrary character (known or unknown). We conducted 4 reconstruction operations with 4 A'_n vector and the reconstructed images are displayed in Table V. The first observation was that the reconstructed images had less noise when compared to the

V. CONCLUSION

Concept blending is one of the creative cognitive processes. Concept blending [21] can be seen as a non-trivial fusion of two existing concepts giving rise to a new third one; it is more than a mere superposition. In this article, we show how a computational cognitive model, named Regulated Activation

TABLE V
RECONSTRUCTED IMAGES OF THE BLENDS OF DIS-SIMILAR BLENDS OF
ABSTRACT CONCEPTS

Blend Of	Reconstructed Image	Blend Of	Reconstructed Image
8's and 3's		3's and 5's	
2's and 5's		4's and 9's	

Networks (RAN), not only reconstructs the encoded input image but also reconstructs the blends of the abstract concepts. The RAN's modeling and MNIST data are used to demonstrate the reconstruction of abstract concepts and their blends. The 2 layered RAN's convex concept model was generated with 3000 MNIST images.

The first experiment showed how RAN's model can be used to encode an image vector of size 784 to a vector of size 31. In the experiment, we chose 10 images that were encoded into 10 vectors. Further, these encoded vectors were reconstructed using RAN's downward propagation mechanism. The reconstructed images were almost identical visually. The average Mean Squared Error between the original images and reconstructed was 0.00562 (ca.) and the average R-squared score was 0.9193. The observations indeed supported the high visual similarity between the original and reconstructed images.

The second experiment demonstrated how blends of two more similar abstract concepts can be reconstructed. In these experiments, we created 10 binary vectors of size 31 and assigned 1 to abstract nodes that represent the same digit and 0 to other nodes. This binary vector was used as an Expected Activation vector in the downward propagation operation. The reconstructed blends, in fact, were the digits whose abstract concepts were being blended. This visual evaluation supported the hypothesis of the experiment that the blend of the abstract concepts representing the same digit should reconstruct the same digit.

The third experiment was similar to the second experiment and demonstrated how the blends reconstruction takes place between two different abstract concepts representing different digits. In these experiments, we created four binary Expected Activation vectors of size 31 and initialized the digit representative nodes with 1 and rest with 0. The reconstructed blends were interesting: the blend of 8's and 3's looked like digit 3; the blend of 3's and 5's was like a Hindi language character; the blend of 2's and 5's was like character x ; and, the blend of 4's and 9's was some unknown symbol. With RAN's modeling, we were able to perform reconstruction of abstract concepts and their blends. In the second experiment, the reconstructed images contained a lot of noise. We are investigating what is causing the problem and in the future, we plan to improve the downward propagation mechanism such that it not only

remove noise but also obtain better images of the encoded images.

ACKNOWLEDGMENT

This work was supported by the project ConCreTe. The project ConCreTe acknowledges the financial support of the Future and Emerging Technologies (FET) programme within the Seventh Framework Programme for Research of the European Commission, under FET grant number 611733. This work was also partially supported by project NanoSen-AQM (SOE2/P1/E0569 (EU-INTERREG-SUDOE))

REFERENCES

- [1] S. Colton and G. A. Wiggins, "Computational creativity: The final frontier?" *Frontiers in Artificial Intelligence and Applications*, vol. 242, pp. 21–26, 2012.
- [2] S. Colton, "Experiments in constraint-based automated scene generation," *Proc. 5th Int. Joint Workshop on Comp. Creativity*, 2008.
- [3] M. T. Pearce and G. A. Wiggins, "Evaluating cognitive models of musical composition," *Proc. of the 4th International Joint Workshop on Computational Creativity*, 2012.
- [4] B. D. Agudo, P. G erv, and P. G. alez Calero, "Poetry generation in colibri," *Advances in Case-Based Reasoning*, vol. 2416, 2002.
- [5] M. Mitchell, "Analogy-making as perception: A computer model," *Cambridge, MA MIT press*, 1993.
- [6] D. Hofstadter, "Fluid concepts and creative analogies," *Computer Models of the Fundamental Mechanisms of Thought*. NY: Basic Books, 1993.
- [7] J. Rehling, "Letter spirit (part two): Modeling creativity in a visual domain. phd thesis," *Indiana University*, 2001.
- [8] C. P. Francisco, *Creativity and Artificial Intelligence, A Conceptual Blending Approach*. De Gruyter Mouton, 2008.
- [9] V. Sorge, S. Colton, R. McCasland, and A. Meier, "Classification results in quasigroup and loop theory via a combination of automated reasoning tools," *Commentationes Mathematicae Universitatis Carolinae*, vol. 49, no. 2, pp. 319–339, 2008.
- [10] M. A. Boden, *The creative mind: Myths and mechanisms*. Routledge, 2004.
- [11] A. M. Pinto and R. Sharma, "Concept learning, recall, and blending with regulated activation networks (abstract)," in *Proceedings of the 14th International Conference on Cognitive Modeling*, ser. ICCM. Penn State University, 2016, pp. 282–284.
- [12] R. Sharma, B. Ribeiro, A. Miguel Pinto, and F. A. Cardoso, "Exploring geometric feature hyper-space in data to learn representations of abstract concepts," *Applied Sciences*, vol. 10, no. 6, 2020.
- [13] R. Sharma, B. Ribeiro, A. M. Pinto, and F. A. Cardoso, "Learning non-convex abstract concepts with regulated activation networks," *Annals of Mathematics and Artificial Intelligence*, pp. 1–29, 2020.
- [14] —, "Modeling abstract concepts for internet of everything: A cognitive artificial system," in *2018 13th APCA International Conference on Automatic Control and Soft Computing (CONTROLO)*. IEEE, 2018, pp. 340–345.
- [15] Y. LeCun, C. Cortes, and C. J. Burges, "The mnist database of handwritten digits," 1999. [Online]. Available: <http://yann.lecun.com/exdb/mnist/>
- [16] F. C. Pereira and A. Cardoso, "The boat-house visual blending experience," in *Proceedings of the Symposium for Creativity in Arts and Science of AISB 2002*, 2002.
- [17] F. Pereira, "A computational model of creativity," Ph.D. dissertation, Ph. D. Dissertation, Universidade de Coimbra, 2005.
- [18] A. Steinbr uck, "Conceptual blending for the visual domain," Ph.D. dissertation, Masters thesis, University of Amsterdam, 2013.
- [19] J. M. Cunha, J. Gonalves, P. Martins, P. Machado, and A. Cardoso, "A Pig, an Angel and a Cactus Walk Into a Blender: A Descriptive Approach to Visual Blending," in *Proceedings of the Eighth International Conference on Computational Creativity*, 2017.
- [20] J. M. Cunha, P. Martins, and P. Machado, "How shell and horn make a unicorn: Experimenting with visual blending in emoji," in *International Conference on Computational Creativity*, 2018, pp. 145–152.
- [21] G. Fauconnier and M. Turner, *The Way We Think: Conceptual Blending and the Mind's Hidden Complexities*. Basic Books, May 2003. [Online]. Available: <http://www.worldcat.org/isbn/0465087868>