An embodied model for handwritten digits recognition in a cognitive robot

DI NUOVO, Alessandro <http://orcid.org/0000-0003-2677-2650>

An Embodied Model for Handwritten Digits Recognition in a Cognitive Robot

Alessandro Di Nuovo
Sheffield Robotics, Department of Computing
Sheffield Hallam University
Sheffield, S1 2NU, United Kingdom
Email: a.dinuovo@shu.ac.uk

***Abstract*— This paper presents an embodied model for recognition of handwritten digits in a cognitive developmental robot scenario. Inspired by neuro-psychological data, the model integrates three modules: a stacked auto-encoder network to process the visual information, a feedforward neural controller for the fingers, and a generalized regression network that associates number digits to finger configurations.**

Results from developmental learning experiments show an improvement in the digits' recognition rate thanks to the inclusion of the robot fingers in the training especially in its early stages (epochs) or with a low number of examples. This behaviour can be linked to that observed in psychological studies with children, who seem to benefit of finger counting only in the initial stage of mathematical learning.

These results suggest the potential of the embodied approach to favour the creation of a psychologically plausible developmental model for mathematical cognition in robots and to support the creation of more complex models of human-like behaviours.

***Index Terms*—Number cognition, Handwritten digits' recognition, finger counting, modular cognitive architecture, symbol grounding.**

I. Introduction

A baby develops many cognitive skills by interacting with other human beings and the environment using his limbs and senses, and consequently, the form of the human body largely influences the development of his intelligence [1].

Mathematical knowledge is believed to be one of the skills that can be extended from a rather limited set of inborn abilities through bodily experiences to an ever-growing network of conceptual domains [2].

Several behavioural studies have shown that gestures have significant a role in the early development of mathematical cognition has been widely studied in children and they suggest that there is an embodied component in learning mathematical concepts [3]. Various embodied strategies, such as finger counting (for a recent special issue on the topic see [4]) and pointing gestures (e.g. [5]), can facilitate the acquisition of number cognition and predict mathematical achievement in children [6], [7]. Importantly, several studies suggest that finger processing may play a role in setting up the biological neural networks on which more advanced mathematical computations are built [8]. However, it also been observed that children use finger counting to support their early mathematical learning and this correlates with better performance, but they do not show gestures in later stages, after they have successfully learned the basic concepts [9].

A recent neuroscientific research shows that adult humans activate the motor cortex while processing digits and number words, even if motor actions are inhibited [10]. It has been shown that the motor cortex activation for small numbers (1-5) is contralateral to the hand used to start the finger counting, therefore relating the finger configurations used to represent numbers and their cardinal meaning. The authors of [10] hypothesize this is the result of an Hebbian association in the early stages of number learning when finger configurations are used by both teachers and children to represent numbers while explain mathematical concepts [11].

A recent research approach known as Cognitive Developmental Robotics (CDR) is naturally suited to study the embodied basis of mathematical learning, where the use of robots, able to interact with the environment and perform gestures such as finger counting, offers the natural tool to model the symbols grounding in sensorimotor knowledge and experience [12]. The CDR approach can also be used to study cognitive dysfunctions and test possible rehabilitation procedures, e.g. [13]. In fact, CDR is defined as the "*interdisciplinary approach to the autonomous design of behavioural and cognitive capabilities in artificial agents (robots) that takes direct inspiration from the developmental principles and mechanisms observed in natural cognitive systems (children)*" [14].

This paper introduces and experimentally tests a neural network model that incorporates the embodied contribution observed in [10], created with the aim to support the effectiveness of the embodied approach in the early mathematical cognition.

The architecture presented here integrates three modules implementing different capabilities: one for finger counting, derived from the previous studies of the author [15]–[17], a module for processing visual inputs (handwritten digits) [18], and an associative module that is designed with an inexpensive generative approach to recreate the motor input

from the visual input. The purpose of this design is to improve the performance in the recognition of handwritten digits to provide a proof-of-concept of the embodied contribution to mathematical learning in robots.

II. Related Work

Among the few attempts to study mathematical cognition via the CDR approach, Ruciński et al. [19] showed that pointing gestures allowed the iCub robot to significantly improve the counting accuracy. Recently, Di Nuovo et al. ([15], [16], [20]) investigated artificial models for the learning of associations between (motor) finger counting, (visual) object counting and (auditory) number words and sequence learning, to explore whether finger counting and the association of number words or digits to each finger could serve to bootstrap the representation of number. The results obtained in the various modelling experiments show that learning the number word sequences together with finger sequencing helps the fast building of the initial representation of numbers in the robot. The neural network's internal representations for these two counting conditions result in qualitatively different patterns of the similarity between numbers. In fact, the internal representations of the finger configurations themselves can be a basis for the building of an embodied number representation in the robot, something in line with embodied and grounded cognition approaches to the study of mathematical cognitive processes. Just as has been found with young children, through the use of finger counting and verbal counting strategies, such a robotic model develops finger and word representations that subsequently sustain the robot's learning the basic arithmetic operation of addition [15]. Finally, using the deep learning approach, Di Nuovo et al. [17] presented an advanced model with superior learning efficiency. The new model was validated in a simulation of the embodied learning behaviour of bi-cultural children, using different finger counting habits to support their number learning.

Aspects of numerical cognition have also been investigated using deep learning architectures and training methods, e.g. restricted-Boltzmann machines and the Contrastive Divergence Learning (e.g. [21], [22]). The deep learning approach is inspired by the complex layered organization of the cerebral cortex. Deep layered processing is thought to be a fundamental characteristic of cortical computation, making it a key feature in the study of human cognition. Deep learning approaches have recently been applied to the modelling of language and cognitive processing, showing how structured and abstract representations can emerge in an unsupervised way from sensory data, through generative learning in deep neural networks (for an overview see [23]). Deep learning architectures and algorithms are becoming popular among connectionist modellers as they represent a new efficient approach to building many layers of information processing stages in deep architectures for pattern classification and for feature or representation learning [24].

Some attempts of using deep learning strategies to model other developmental learning tasks can be found in the literature, for a recent survey the reader can refer to [25]. For instance, an unsupervised deep learning model has been proposed to approach the multimodal learning for autonomous robots [26].

III. Materials and Methods

A. Handwritten Digits Data Set

This example uses synthetic data built for training and testing the artificial neural network. The synthetic images have been generated by applying random affine transformations to digit images created using different fonts [27].
Each digit image is 28-by-28 (784) pixels and a total of 9,000 examples were generated. Figure 2 gives some examples.

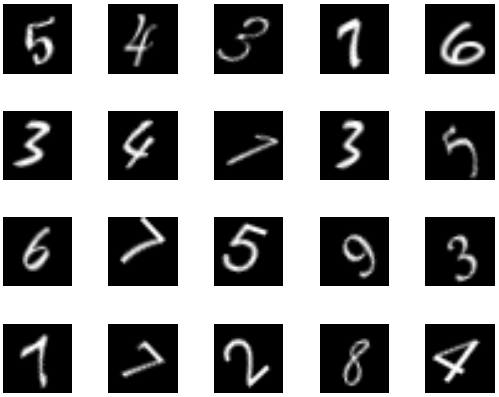

Figure 1. Examples of the handwritten digits in the dataset.

Half of the examples were randomly chosen for the training, while the other half was used for the testing. The distribution of the examples was uniform in both the training and the testing datasets, which counted a total of 4,500 items each, 500 for each digit considered. The proportion was the same for the training and the test set.
Only digits from 1 to 9 were considered in this study, as in the one we want to replicate had this constraint too [10]. In fact, in our case, the zero has no direct fingers activation that can be associated and therefore we decided to leave it out.

B. The iCub robot

The cognitive robotic platform used for the experiments presented here is the simulation of the iCub humanoid robot.

The iCub (Figure 3 on the right) is a popular open source platform designed for developmental robotics research, based on a child-like morphology, with 53 degrees of freedom, adopted by more than 20 laboratories worldwide. iCub is an open-source humanoid robot platform designed to facilitate cognitive developmental robotics research as detailed in [28].

Figure 2. The iCub humanoid robot platform: The realistic simulator (left); The real platform (right).

The iCub provides motor proprioception (joint angles), force/torque sensors tactile information on the fingers, 2 standard cameras in biomimetic DOF (pan, tilt, vergence) setup for vision, inertial sensors. One of the most advanced

parts of the iCub is the hand, that comprises 9 DoF, for a total of 18 DoF, and it is the result of a design that optimized the level of integration of the hand in the overall robot to meet the child-like project specifications in terms of dimensions, dexterity, and sensorization.

The implementation used for the experiments presented here is a simulation of the iCub humanoid robot (Figure 2 on the left). The simulator, which was developed with the aim to accurately reproduce the physics and the dynamics of the physical iCub [29], allows the creation of realistic physical scenarios in which the robot can interact with a virtual environment. Physical constraints and interactions that occur between the environment and the robot are simulated using a software library that provides an accurate simulation of rigid body dynamics and collisions.

In this work, we control the fingers only, which have 7 DoF for each hand, distributed as follows: 2 degrees of freedom for thumb, index, and middle fingers, but only one for controlling the ring and pinky fingers, that are "glued" together. Because of the limitation with the last two fingers the finger configurations are not sequential as represented in Figure 4. To balance the input, we duplicated the contribution of the motors that control two fingers, therefore we have 16 inputs for the motor module. Numbers from six to ten are represented by adding left-hand fingers with all the right-hand fingers open (e.g. six is five on the right hand plus one on the left hand).

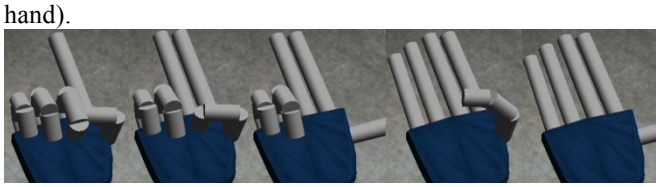

Figure 3. Number representation with the right-hand fingers of the iCub. From left to right: one, two, three, four and five. Numbers from six to nine are represented with two hands.

C. A Modular Neural Network model for Handwritten Digits' Recognition in an Embodied Robot

This work aims to investigate the use of a modular (expandable) cognitive architecture, based on the recent deep learning strategies and components, for showing the advantages and applications of the embodied principles to mathematical cognition. Indeed, the design and implementation process has been inspired by a control engineering methodology with the deep-layered organization of the human brain, in which "*there are parts* (modules) *that control other parts* (modules)" [30].

The architecture created for this work makes use of recent deep learning strategies for designing an expandable deep-layered architecture, in which modules with lower level functions (e.g. finger control, visual inputs) can be connected to higher level cognitive capabilities (e.g. number cognition) and between themselves through several intermediate layers that integrate the contributions (output results) from lower modules to produce inputs for higher modules. Single modules can be independent and implemented with appropriate multilayer neural networks architectures that are specialized in realizing specific tasks.

The main component is the Auto-encoder network [18], which is trained to replicate its input at its output. It consists of two parts: an encoder that creates a hidden representation from the inputs; a decoder that attempts to map this representation back to the original input. Therefore, the size of its input will be the same as the size of its output. However, the number of neurons in the hidden layer is less than the size of the input/output, thus, the purpose of the auto-encoder is to learn a compressed representation of the input that can be used to extract the salient features (e.g. of an image) for further processing.

The training of an Auto-encoder is considered unsupervised in the sense that no labelled data is needed. Once the Auto-encoder network is trained, the encoder part can be used to initialize weights and biases of deep neural networks, in order to improve the performance of the full network [18].

In this work, the implementation of the auto-encoder neural network is from Mathworks MATLAB Deep Learning Toolbox. The architecture presented here (Figure 1) is created by merging three blocks (modules):

1. A three-layer classifier for finger counting (motor module) [17].
2. A stacked auto encoder neural network for handwritten digits recognition (visual module) [18].
3. A Generalised Regression Network for Visuo-motor association [31].

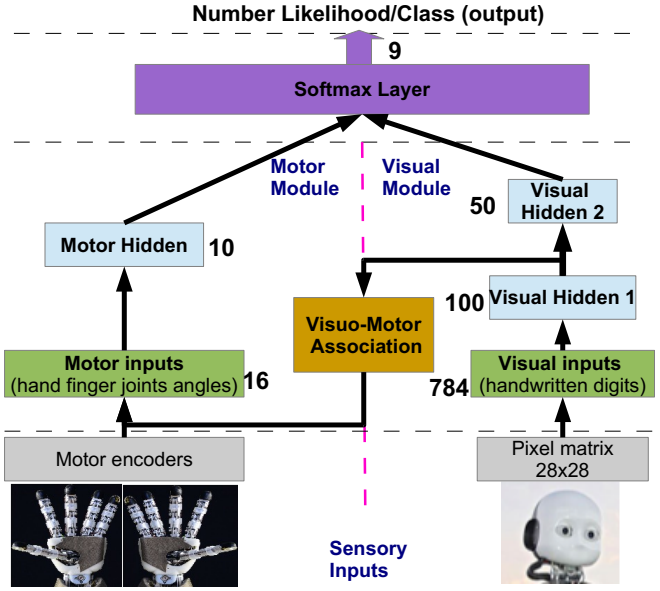

Figure 4. A block representation of the embodied architecture for handwritten digits' recognition.

The blocks are independently pre-trained before merging, and then the full network is trained via backpropagation.

A block is a multi-layer network. Number classes are the outputs of each block. The motor block is on the left, the visual block is on the right. In the middle, the Visuo-Motor Association block (yellow) is trained to predict the motor activations of finger configurations for the corresponding number digits from the hidden outputs of Visual Autoencoder (Hidden 1). This is to allow the classification when only images are provided as an input and replicate the motor cortex activation shown in [10].

1) The motor module

A network acting as finger controller was included to model motor activation when processing numbers, as shown by [10].

This module is formed of three layers: one responsible for connecting the inputs, one hidden layer and the final output layer, whose units has a *softmax* transfer function.

The hidden layer of the motor module is pre-trained using the encoder part of an Auto-encoder network that learns to recreate the finger configurations. Then, the output layer is trained to classify the hidden output in the number classes (1-9) using a supervised backpropagation algorithm with labelled examples.

2) The visual module

The visual module is realized stacking one or two Auto-encoders and a *softmax* layer to build classifier of handwritten digits to number classes. In the stacking process, the decoder part of the Auto-encoders is removed and the encoders are connected to form a single multi-layered network.

The original vectors in the training data had 784 dimensions. After passing them through the first encoder, this was reduced to 100 dimensions. After using the second encoder, this was reduced again to 50 dimensions. The final *softmax* layer uses the 50-dimensional vectors representing the salient features to classify into different digit classes.

The 100-dimensional output from the hidden layer of the first auto-encoder is a compressed version of the input, which summarizes its response to the features of the handwritten digit images. After training the first auto-encoder, the second auto-encoder is trained in a similar way. The main difference is that it makes use of the features that were generated from the first auto-encoder as the training data so that the encoder in the second auto-encoder learns an even smaller representation of the input data.

3) The visuo-motor association module

This block is a generalized regression neural network (GRNN) that has a hidden layer with radial basis transfer functions and a special output layer that uses a linear transfer function without bias to match the targets.

The role of the GRNN is to learn the association between the two modules. The network is trained to predict the finger configurations (motor joints' angles) from the outputs of the visual hidden 1. This network is responsible for providing the necessary input to the network in the testing phase when there is no actual movement of the robot's fingers.

The GRNN is designed using a generative approach [32] in which a radial basis unit is added to the hidden layer for each input presented to the network during training. Therefore, the first layer weights are set using input values, and the first layer biases are all set to 0.8326/spread. For our experiments, the spread of radial basis functions is 0.1.

The generative approach is particularly beneficial for our experiments because it is very quick to create the network and, moreover, the outputs will always be a valid finger configuration.

D. Training and Testing

The training is divided into two phases:

1. First, the modules are trained separately. The motor and visual modules are trained independently with their own *softmax* layer, using the procedure described in [18].
   The hidden layers are pre-trained for 100 epochs using their own inputs as outputs:
   - The motor Auto-encoder is pre-trained with the finger configurations. Inputs/outputs are the values read from the fingers' motor encoders.
   - The Auto-encoder networks are pre-trained to reproduce their inputs. Auto-encoder 1 reproduces the digits' images, while the Auto-encoder 2 is trained to reproduce the output of the hidden units of Auto-encoder 1.

   Then, a *softmax* layer is trained to classify the numbers in a supervised fashion with the following inputs (separately):
   - activation of the motor hidden units.
   - activation of visual hidden 1 or 2 for the vision module (1 hidden or 2 hidden layers).

   To conclude phase 1, the Visual-Motor association module is trained to predict the fingers' motor joint configurations from the hidden outputs of the Autoencoder 1. This enables the classification when the only inputs are images of digits.
2. In the second phase, the entire network (Figure 1) is re-trained to classify the numbers using the digit images as the only input. The fingers' configurations are predicted by the visual-motor association and the predictions are used as input for the motor module.

In the experimental testing, the motor input is inhibited and substituted by the output of the Visuo-motor associative network that replicates the finger's motor positions. Therefore, predicted motor positions are processed by the motor module that is still actively contributing to the classification. In fact, the model imitates the motor cortex activation shown in [10].

IV. Experimental Results and Discussion

In our experimental test, the number of examples provided in the training phase was varied to better assess the developmental performance of the model. For the same reason, it has been also evaluated the effect of a varying number of training epochs, i.e. the number of times the input/output examples are presented.

We performed a total of eight experiments, where the number of examples provided for training was 45, 90, 225, 450, 900, 1125, 2250, 4500. We also recorded the performance at four steps: after the initialisation phase (INIT), 50, 100 and 1000 epochs. For a correct comparison of the results, the experiments were repeated in such a way that the entire training set was always used. For instance, if 45 examples were provided, the experiment was repeated 100 times, with the different set of examples, and the mean value was used in the analysis.

Each experiment was repeated 11 times. The classification accuracy percentage (or recognition rate) shown in tables is the median value of the 11 runs for each experiment. The Student t-test has been used to statistically verify the significance of the differences between the different approaches.

The performance of the architecture is summarized in Figure 4, where the classification accuracy of the standard visual auto-encoder module is compared with the embedded model. The columns represent the performance for a varying number of handwritten digit examples, while the rows the training epochs.

TABLE I. EMBEDDED VS STANDARD MODELS: RECOGNITION ACCURACY IN PERCENTAGE ON THE TESTING SET WITH VARYING EXAMPLES AND EPOCHS

Epoch	Stacked Auto-Encoder (Visual only, 1 hidden layer)							
N	45	90	225	450	900	1125	2250	4500
init	*26.7*	*37.2*	*47.2*	*48.4*	*49.2*	*48.9*	*48.1*	*47.0*
50	*28.2*	*38.3*	*59.9*	*73.6*	*85.1*	*91.8*	*95.2*	*98.0*
100	*28.2*	*38.3*	*59.9*	*73.9*	*85.8*	*93.0*	*96.3*	*99.1*
1000	*28.2*	*38.3*	*59.9*	*73.9*	*85.8*	*93.0*	*96.3*	*99.1*
	Stacked Auto-Encoder (Visual only, 2 hidden layers)							
N	45	90	225	450	900	1125	2250	4500
init	*23.4*	*30.7*	*28.7*	*26.8*	*26.0*	*25.8*	*22.7*	*23.5*
50	*24.4*	*39.2*	*56.4*	*67.0*	*75.2*	*75.2*	*69.6*	*72.4*
100	*24.4*	*39.3*	*57.5*	*69.5*	*81.4*	*89.0*	*92.0*	*96.4*
1000	*24.4*	*39.3*	*57.5*	*69.9*	*82.5*	*90.7*	*94.7*	*98.2*
	Embedded Model (1 hidden layer in Visual module)							
N	45	90	225	450	900	1125	2250	4500
init	*30.0*	*41.3*	*55.6*	*47.8*	*41.4*	*42.8*	*42.8*	*45.0*
50	***33.7***	*43.4*	*59.3*	*74.5*	*87.4*	*94.5*	*97.2*	*99.7*
100	***33.7***	*43.4*	*59.8*	*75.2*	***87.9***	*94.7*	*97.4*	***99.8***
1000	***33.7***	***43.6***	***60.7***	***75.3***	***87.9***	*94.7*	*97.4*	***99.8***
	Embedded Model (2 hidden layers in Visual module)							
N	45	90	225	450	900	1125	2250	4500
init	*29.3*	*41.2*	*55.4*	*47.5*	*41.0*	*42.9*	*43.0*	*45.0*
50	*31.5*	*40.9*	*57.3*	*74.4*	*87.5*	***94.9***	***97.6***	***99.8***
100	*31.6*	*41.1*	*57.8*	*74.4*	*87.5*	***94.9***	***97.6***	***99.8***
1000	*31.8*	*41.2*	*58.5*	*74.4*	*87.5*	***94.9***	***97.6***	***99.8***

init = *initial performance before full backpropagation learning.*

From Table I, it can be seen there is a clear advantage in integrating the finger counting in the handwritten digit recognition. In fact, for the lowest number of examples (45) the embedded models show a significantly better recognition performance (5.5% and 7.4% over the visual ones with the same number of hidden layers) with $p<0.05$. The improvement in the recognition accuracy is confirmed for all the other test cases with statistical significance $p<0.05$. It can also be noted that the performance improvement is higher when the examples are limited (45, 90) and after fewer training epochs.

It can also be noted an advantage with lower examples of having a single hidden layer in the vision module have, while the deeper embedded network (motor and two hidden layers in the vision module) is performing best with the greater number of examples (≥1125).

Figure 5 presents the ratio between the sum of motor weights and the sum of visual weights of the neural links between the last hidden layer units and the *softmax* layer. The decrease of the ratio implies a weaker contribution of the motor module in the final classification.

Table II presents the comparison of the embodied model with and without the pre-train phase and, also, other number coding as input instead of the finger configuration. The results show a performance improvement with the pre-training that is statistically significant ($p<0.05$), while small differences are recorded with other number codings, but these are not statistically significant ($p>0.05$).

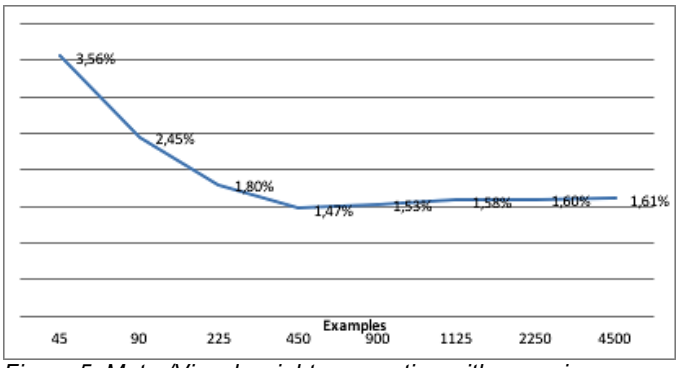

Figure 5. Motor/Visual weights proportion with a varying number of examples. The higher the ration the stronger the contribution of motor module.

TABLE II. EMBEDDED MODEL RECOGNITION ACCURACY (PERCENTAGE): COMPARISON WITH AND WITHOUT PRE-TRAINING AND DIFFERENT NUMBER CODINGS

Epoch	With pre-training (Robot motor joints angles)							
N	45	90	225	450	900	1125	2250	4500
init	*30.0*	*41.3*	*55.6*	*47.8*	*41.4*	*42.8*	*42.8*	*45.0*
50	*33.7*	*43.4*	*59.3*	*74.5*	*87.4*	*94.5*	*97.2*	*99.7*
100	*33.7*	*43.4*	*59.8*	*75.2*	*87.9*	***94.7***	*97.4*	***99.8***
1000	*33.7*	*43.6*	*60.7*	*75.3*	*87.9*	***94.7***	*97.4*	***99.8***
	Without pre-training (Robot motor joint angles)							
N	45	90	225	450	900	1125	2250	4500
init	*11.1*	*11.1*	*11.1*	*11.1*	*11.1*	*11.1*	*11.1*	*11.1*
50	*30.1*	*39.2*	*55.8*	*70.4*	*84.0*	*92.2*	*95.1*	*98.1*
100	*30.8*	*39.9*	*56.9*	*70.9*	*84.5*	*92.6*	*96.3*	*99.5*
1000	*31.4*	*40.5*	*56.8*	*71.2*	*84.8*	*92.7*	*96.5*	*99.6*
	Numerosity magnitude							
N	45	90	225	450	900	1125	2250	4500
init	*30.5*	*42.2*	*56.0*	*48.1*	*41.4*	*42.9*	*42.7*	*44.9*
50	*33.8*	*43.8*	*57.8*	*68.3*	*87.4*	*76.6*	*95.3*	*97.3*
100	*33.8*	*43.9*	*59.4*	*74.0*	***88.1***	*91.8*	*97.0*	*99.7*
1000	*33.8*	*44.0*	***60.9***	***76.2***	***88.1***	***94.7***	***97.6***	***99.8***
	Single neuron activation							
N	45	90	225	450	900	1125	2250	4500
init	*32.3*	*41.3*	*57.4*	*72.0*	*85.4*	*93.1*	*96.6*	*99.6*
50	***33.9***	***44.1***	*60.2*	*74.0*	*86.2*	*93.4*	*96.7*	*99.6*
100	***33.9***	*43.9*	*60.0*	*73.6*	*86.0*	*93.4*	*96.6*	*99.6*
1000	***33.9***	*43.9*	*60.0*	*73.5*	*86.0*	*93.4*	*96.6*	*99.6*

V. CONCLUSION AND FUTURE WORK

This paper presented a first attempt to design a modular cognitive architecture for mathematical processing in robots.

Inspired from the association between digits and fingers observed in [10], we created a cognitive robotics model and performed a series of developmental learning experiments in which we simulate the learning of small number digits (1-9) without the association of robot's finger configurations.

In comparisons with a standard non-embodied approach (Table I), the embodied learning model show an improvement of up *7.4%* in recognition rate thanks to the inclusion of the finger configurations as an input during the training. In fact,

analysing the weights of the network (Figure 5), we see a reduction of motor network contribution

These results can be related to the observation that children benefit at beginning of using finger counting in their learning but do not show gestures in later stages of mathematical learning [9]. Indeed, given the performance improvement, we can hypothesise that they associate fingers and number to facilitate their initial learning, but they abandon this habit when they are more experienced because the advantage is not sufficiently high to justify the extra time needed for opening and closing the fingers. This is coherent with the time pressure proposed by the embodied cognition theory [33].

In Table II, we see that the pre-train led to better results in terms of quicker learning and higher recognition rate. The pre-train can be considered as an exposure of the robot to finger counting by external subjects (e.g. a teacher) before the actual learning. This is consistent with the evidence that exposure to gesturing while talking about numbers can promote children numerical understandings [3].

Finally, as future work, we will apply the same embodied approach to both number words and handwritten digits, using well know datasets, e.g. the MNIST, with the aim to compare the performance of the embodied network with standard machine learning approaches.

ACKNOWLEDGMENTS

The author wishes to thank prof. James McClelland for the valuable comments and discussion about this work.

This work was supported in part by the EPSRC grant EP/P030033/1 (NUMBERS) and by the European Commission under Grant n. 703489 (CARER-AID).

The author is grateful to the NVIDIA corporation for the donation of a Tesla K40 and a GeForce TITAN X that have been used for the experiments.

REFERENCES

[1] R. Pfeifer, J. Bongard, and S. Grand, *How the body shapes the way we think: a new view of intelligence*. MIT press, 2007.

[2] G. Lakoff and R. Nuñez, *Where Mathematics Comes From: How the Embodied Mind Brings Mathematics into Being*. Basic Books, 2001.

[3] S. Goldin-Meadow, S. C. Levine, and S. Jacobs, "Gesture's role in learning arithmetic," L. D. Edwards, F. Ferrara, and D. Moore-Russo, Eds. Information Age Publishing, 2014.

[4] F. Domahs, L. Kaufmann, and M. H. Fischer, *Handy numbers: Finger counting and numerical cognition*. Frontiers E-books, 2014.

[5] M. W. Alibali and A. A. DiRusso, "The function of gesture in learning to count: More than keeping track," *Cogn. Dev.*, vol. 14, no. 1, pp. 37–56, 1999.

[6] S. D. Newman, "Does finger sense predict addition performance?," *Cogn. Process.*, vol. 17, no. 2, pp. 139–146, 2016.

[7] I. Long, S. A. Malone, A. Tolan, K. Burgoyne, M. Heron-Delaney, K. Witteveen, and C. Hulme, "The cognitive foundations of early arithmetic skills: It is counting and number judgment, but not finger gnosis, that count," *J. Exp. Child Psychol.*, vol. 152, pp. 327–334, 2016.

[8] K. Moeller, L. Martignon, S. Wessolowski, J. Engel, and H.-C. Nuerk, "Effects of finger counting on numerical development - the opposing views of neurocognition and mathematics education.," *Front. Psychol.*, vol. 2, no. November, p. 328, Jan. 2011.

[9] N. C. Jordan, D. Kaplan, C. Ramineni, and M. N. Locuniak, "Development of number combination skill in the early school years: When do fingers help?," *Dev. Sci.*, vol. 11, no. 5, pp. 662–668, 2008.

[10] N. Tschentscher, O. Hauk, M. H. Fischer, and F. Pulvermüller, "You can count on the motor cortex: finger counting habits modulate motor cortex activation evoked by numbers.," *Neuroimage*, vol. 59, no. 4, pp. 3139–48, Feb. 2012.

[11] M. W. Alibali and M. J. Nathan, "Embodiment in mathematics teaching and learning: Evidence from learners' and teachers' gestures," *J. Learn. Sci.*, vol. 21, no. 2, pp. 247–286, 2012.

[12] A. Cangelosi, A. Morse, A. Di Nuovo, M. Rucinski, F. Stramandinoli, D. Marocco, V. De La Cruz, and K. Fischer, "Embodied language and number learning in developmental robots," in *Conceptual and Interactive Embodiment: Foundations of Embodied Cognition*, vol. 2, Routledge, 2016, pp. 275–293.

[13] D. Conti, S. Di Nuovo, A. Cangelosi, and A. Di Nuovo, "Lateral specialization in unilateral spatial neglect: a cognitive robotics model," *Cogn. Process.*, vol. 17, no. 3, pp. 321–328, 2016.

[14] A. Cangelosi and M. Schlesinger, *Developmental robotics: From babies to robots*. MIT Press, 2015.

[15] V. M. De La Cruz, A. Di Nuovo, S. Di Nuovo, and A. Cangelosi, "Making fingers and words count in a cognitive robot.," *Front. Behav. Neurosci.*, vol. 8, no. February, p. 13, 2014.

[16] A. Di Nuovo, V. M. De La Cruz, A. Cangelosi, and S. Di Nuovo, "The iCub learns numbers: An embodied cognition study," in *International Joint Conference on Neural Networks (IJCNN 2014)*, 2014, pp. 692–699.

[17] A. Di Nuovo, V. M. De La Cruz, and A. Cangelosi, "A Deep Learning Neural Network for Number Cognition: A bi-cultural study with the iCub," in *IEEE International Conference on Development and Learning and Epigenetic Robotics (ICDL-EpiRob) 2015*, 2015, pp. 320–325.

[18] G. E. Hinton and R. R. Salakhutdinov, "Reducing the dimensionality of data with neural networks," *Science (80-.).*, vol. 313, no. 5786, pp. 504–507, 2006.

[19] T. Rucinski, M., Cangelosi, A., and Belpaeme, "Robotic model of the contribution of gesture to learning to count.," in *Proceedings of of IEEE ICDL-EpiRob 2012*, 2012.

[20] A. Di Nuovo, V. M. De La Cruz, and A. Cangelosi, "Grounding fingers, words and numbers in a cognitive developmental robot," in *IEEE Symposium on Cognitive Algorithms, Mind, and Brain (CCMB)*, 2014, pp. 9–15.

[21] I. Stoianov and M. Zorzi, "Emergence of a'visual number sense'in hierarchical generative models," *Nat. Neurosci.*, vol. 15, no. 2, pp. 194–196, 2012.

[22] M. Zorzi, I. Stoianov, and C. Umiltà, "Computational Modeling of Numerical Cognition," in *Handbook of mathematical cognition*, vol. 19, 2005, pp. 67–84.

[23] M. Zorzi, A. Testolin, and I. P. Stoianov, "Modeling language and cognition with deep unsupervised learning: A tutorial overview," *Front. Psychol.*, vol. 4, 2013.

[24] Y. Bengio, *Learning deep architectures for AI*, vol. 2, no. 1. Now Publishers Inc., 2009.

[25] O. Sigaud and A. Droniou, "Towards deep developmental learning," *IEEE Trans. Cogn. Dev. Syst.*, vol. 8, no. 2, pp. 99–114, 2016.

[26] A. Droniou, S. Ivaldi, and O. Sigaud, "Deep unsupervised network for multimodal perception, representation and classification," *Rob. Auton. Syst.*, vol. 71, pp. 83–98, 2015.

[27] Mathworks, "Train Stacked Autoencoders for Image Classification," 2017. [Online]. Available: https://uk.mathworks.com/help/nnet/examples/training-a-deep-neural-network-for-digit-classification.html. [Accessed: 02-Apr-2017].

[28] G. Metta, L. Natale, F. Nori, G. Sandini, D. Vernon, L. Fadiga, C. von Hofsten, K. Rosander, M. Lopes, J. Santos-Victor, A. Bernardino, and L. Montesano, "The iCub humanoid robot: An open-systems platform for research in cognitive development," *Neural Networks*, vol. 23, pp. 1125–1134, 2010.

[29] V. Tikhanoff, A. Cangelosi, P. Fitzpatrick, G. Metta, L. Natale, and F. Nori, "An open-source simulator for cognitive robotics research: the prototype of the iCub humanoid robot simulator," in *PerMIS '08: Proceedings of the 8th Workshop on Performance Metrics for Intelligent Systems*, 2008, pp. 57–61.

[30] A. Roy, "Connectionism, controllers, and a brain theory," *IEEE Trans. Syst. Man, Cybern. A Syst. Humans*, vol. 38, no. 6, pp. 1434–1441, 2008.

[31] T. Hastie and R. Tibshirani, *Generalized additive models*. Wiley Online Library, 1990.

[32] P. D. Wasserman, *Advanced Methods in Neural Computing*. New York: Van Nostrand Reinhold, 1993.

[33] M. Wilson, "Six Views of Embodied Cognition," *Psychon. Bull. Rev.*, vol. 9, no. 4, pp. 625–636, 2002.