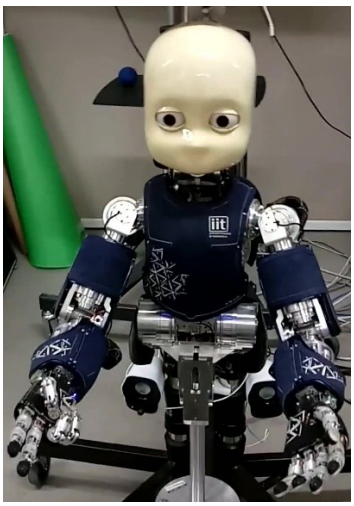
Developing the Knowledge of Number Digits in a child like Robot

DI NUOVO, Alessandro <http://orcid.org/0000-0001-5308-7961> and MCCLELLAND, James L.

Developing the Knowledge of Number Digits in a child-like Robot

Supplementary Information

iCub Robot Finger configurations



Supplementary Figure 1. The robotic platform and finger representations.

At the top, the iCub robot. At the bottom, the number representations with the iCub right-hand fingers: one, two, three, four and five. Numbers from six to nine are represented with two hands, with the right hand fully open, e.g. 6 equals 5 on the right and 1 on the left. In practice, the embodied representation is formed by the joint angles from the fingers' motor encoders. For the full sequence, a supplementary video of the iCub counting from 1 to 10 is also included.

Supplementary Comments and Results

Scenario 1: Learning to process spoken digits while acquiring counting principles

The simulations in this scenario may be extended by merging the recurrent network model presented in[1] with the deep convolutional model proposed in this article, it would also be possible to capture children's learning of the ordering of the digits as well, thus capturing the stable order principle underlying the natural numbers.

For this experiment, we can also report that the simple baseline was significantly slower and less efficient than all the others. In the case of the two smaller sample sizes, the baseline is significantly the worst, and it is not able to achieve the same accuracy even after an additional 25 epochs of training. Only in the case of the larger sample size, the simple baseline achieved an accuracy better than the others (+0.27%), but only after 22 (median) additional epochs of training, i.e. after a total 47 epochs (median).

Scenario 2: A Longitudinal study of spoken digits recognition in embodied artificial agents

Interestingly, in scenario 2, the control model was significantly more efficient with the medium group, where we noted that the control model reached its highest accuracy on the training set (highest average was 0.974 with 2788). On the contrary, the control model accuracy decreased on larger sets (average accuracy on the training set was around 0.96 with 27884) despite the mean squared error (MSE) of the predicted random representations continued improved for both training and testing. Indeed, the control model achieved a lower performance against the simple baseline on the full-sized training set (27884) when the average MSE on the testing set was 0.007. This suggests an inverse relationship between the two targets, which indicates that generic random representations successfully augmented the training only when their prediction was not fully accurate (average testing MSE in the range [0.012, 0.026]). This may explain also the quicker learning of the control model with 5576 examples because the MSE is in the range [0.012, 0.017] during the first 16 epochs. We didn't note a similar effect with the embodied models, which have a stable accuracy on the training set from 2788 onward and still achieved higher results than the baseline.

Supplementary Table 1. Student t-test *p* values for the longitudinal experiment.
Comparisons are between the model in the first and the second of the header rows. p values smaller than 0.05 are in bold. Black is a positive difference, red is negative.

Training examples (pre/full)	Random values			Cardinal Numerosity		Robot fingers
	Cardinal Numerosity	Robot fingers	Baseline	Robot fingers	Baseline	Baseline
After Epoch 25 (average of testing after epochs with lowest training loss)						
32/128	**0.0166**	**0.0055**	0.5837	0.6894	**0.0009**	**0.0001**
128/512	**<0.0001**	**<0.0001**	0.0003	0.6038	**<0.0001**	**<0.0001**
256/1024	**<0.0001**	**0.0004**	**<0.0001**	0.7223	**<0.0001**	**<0.0001**
697/2788	0.3637	0.8279	**<0.0001**	0.4512	**<0.0001**	**<0.0001**
1394/5576	**0.0051**	**0.0065**	**<0.0001**	0.8645	**<0.0001**	**<0.0001**
3485/13942	**0.0073**	**0.0015**	0.7466	0.3488	**0.0058**	**0.0010**
6971/27884	**0.0013**	**0.0098**	0.0647	0.5153	0.0843	0.3079
Final (average of testing after epochs with lowest training loss)						
32/128	0.1315	0.1289	0.1845	0.7927	**0.0042**	**0.0070**
128/512	**0.0466**	**0.0461**	**<0.0001**	0.7388	**<0.0001**	**<0.0001**
256/1024	0.9239	0.6442	**<0.0001**	0.6890	**<0.0001**	**0.0002**
697/2788	0.6120	0.9784	**0.0001**	0.6020	**0.0001**	**<0.0001**
1394/5576	0.4495	0.2707	**0.0003**	0.6918	**<0.0001**	**<0.0001**
3485/13942	**0.0116**	0.1973	0.5593	0.0805	**0.0025**	0.0574
6971/27884	0.0899	0.6294	0.6451	0.2282	0.2201	0.9827

Supplementary Table 2. Comparison of control conditions.

Average accuracy rates (Acc) on the test set, with Standard Deviations (SD) and Cohen's d. Accuracy rates are highlighted in green when significantly (p<0.05) better than baseline, in bold when significantly (p<0.05) better (black) or worse (red) than the pre-trained baseline. The final rows of this table show the median epochs when test accuracy was greater than 99% of the baseline's final average accuracy.

Training examples (pre/full)	Pre-trained baseline		Control Model with Random Values				Baseline			
	Acc	SD	Acc	SD	*d*	*p*	Acc	SD	*d*	*p*
After Epoch 1										
32/128	0.1485	0.0231	0.1481	0.0237	-0.015	0.9507	0.1406	0.016	-0.373	0.1195
128/512	0.2291	0.0331	0.2382	0.0255	0.292	0.2209	**0.1901**	0.024	-1.267	<0.0001
256/1024	0.3037	0.0336	0.3011	0.0437	-0.061	0.7960	**0.2224**	0.026	-2.563	<0.0001
697/2788	0.5758	0.0468	0.5802	0.0631	0.074	0.7553	**0.2967**	0.030	-6.727	<0.0001
1394/5576	0.8293	0.0384	0.8292	0.0213	-0.003	0.9895	**0.4159**	0.052	-8.569	<0.0001
3485/13942	0.9176	0.0156	0.9210	0.0094	0.254	0.2859	**0.6784**	0.036	-8.046	<0.0001
6971/27884	0.9420	0.0083	0.9384	0.0143	-0.289	0.2270	**0.8279**	0.028	-5.289	<0.0001
Final (average of testing after epochs with lowest training loss)										
32/128	0.4054	0.0248	0.4093	0.0268	0.142	0.5599	0.4000	0.028	-0.195	0.4236
128/512	0.7140	0.0171	0.7213	0.0181	0.392	0.1075	**0.6960**	0.025	-0.803	0.0014
256/1024	0.8368	0.0114	**0.8487**	0.0107	1.015	0.0001	0.8340	0.013	-0.221	0.3649
697/2788	0.9211	0.0078	0.9230	0.0064	0.257	0.2924	**0.9166**	0.006	-0.618	0.0123
1394/5576	0.9452	0.0037	0.9463	0.0039	0.273	0.2590	**0.9419**	0.005	-0.707	0.0045
3485/13942	0.9632	0.0024	0.9630	0.0030	-0.073	0.7619	0.9625	0.003	-0.227	0.3496
6971/27884	0.9712	0.0017	0.9714	0.0017	0.144	0.5590	0.9716	0.002	0.257	0.2996
Median Epoch when testing accuracy was greater than 99% of the Baseline's Final Average Accuracy										
32/128	49		44				43			
128/512	48		36				45			
256/1024	39		31				46			
697/2788	22		17				34			
1394/5576	17		16				22			
3485/13942	13		13				14			
6971/27884	9		12				14			

The pre-trained baseline is comparable to control model, both significantly increased initial accuracy, however, this did not always result in a significant improvement with respect to the baseline of the final accuracy, where the pre-trained baseline was almost in all cases less accurate than the control model, but there was a statistically significant difference only once. In general, the pre-trained baseline was also slower; indeed, it reached 99% of the baseline final accuracy usually several epochs later than the control model.

To provide additional information for understanding the meaning of embodied representation quantitatively, Supplementary Table 3 presents the correlation matrices between spoken digits and the three output representations used in our experiments. In the case of the spoken digits, the correlation was calculated between the average spectrograms. In the case of cardinal numerosity, the correlation was calculated adding a zero to the end of each vector to avoid a division by zero. Otherwise, nine has a standard deviation equal to zero because it is represented by a vector of ones only.

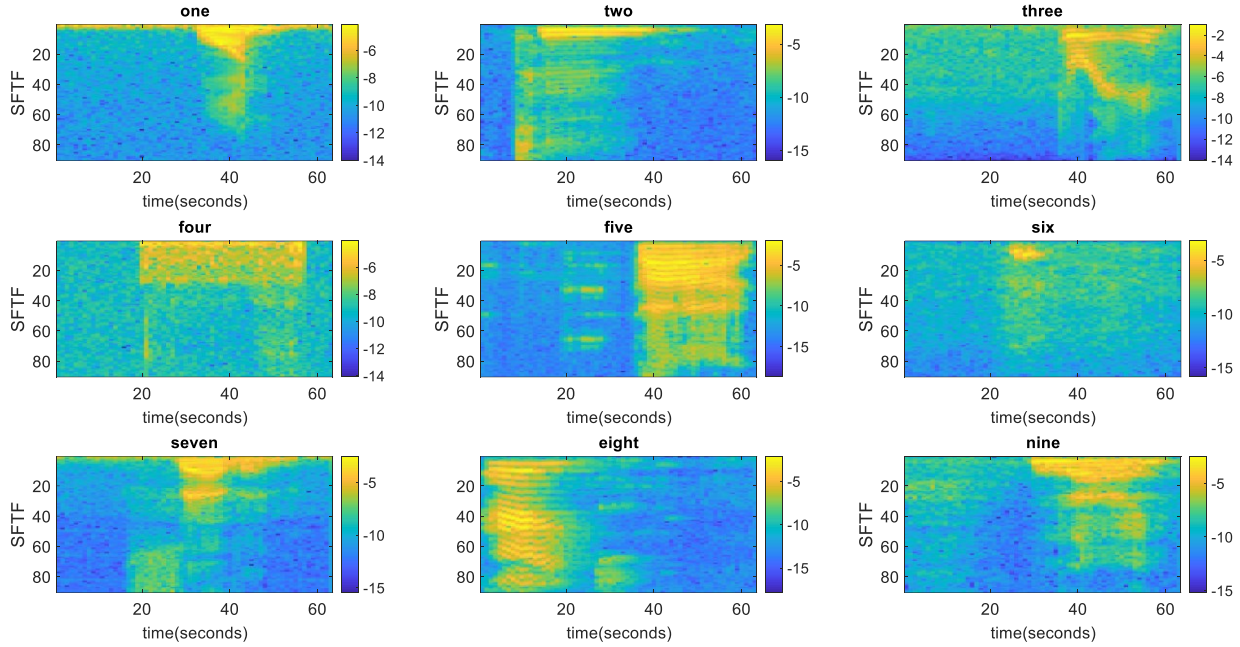
Analysing the correlation coefficients, it could be noted that there are several moderate correlations between the spoke digit spectrograms: 1 with 3,7 and 9; 3 with 5 and 9; 5 with 8 and 9. These correlations make the learning slower because the network will require longer training to find the best parameters to differentiate between these digits. The one-hot representation will not provide any meaningful feedback, because it represents all the digits in the same way with equal correlations among all of them. In contrast, the cardinal numerosity and the robot fingers have a very similar correlation profile, where the only

exception are the two pairs 3-4 and 8-9, with only the latter being significantly different: a moderate positive correlation versus a weak negative correlation. However, these two pairs of spoken digits have a weak correlation (especially 8-9), therefore, they are not particularly difficult to classify. This can explain the similar performance shown by the cardinal numerosity and the robot fingers in our experiments. Furthermore, stronger correlations between the spectrograms were found for non-adjacent digits, while the cardinal numerosity and the robot fingers correlations linearly decrease with their difference. For instance, while the spectrograms of 1 and 3 have a moderate correlation with 9, the cardinal numerosity and the robot fingers have weak correlations among these digits. This property can help the network to better differentiate among these numbers as we have found in our experimental results, e.g. see Table 2 and the comments for Scenario 1.

Supplementary Table 3. Correlation matrices.
Correlation coefficients are between the average spectrograms of spoken digits, the two artificial representations (cardinal numerosity and one-hot) and the iCub robot finger positions. Correlations are highlighted with different shades (light = moderate, dark = strong) of green (positive) or red (negative). Weak correlations are in yellow (positive) or orange (negative).

	1	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*	*1*	*2*	*3*	*4*	*5*	*6*	*7*	*8*	*9*
	Spoken digits (average spectrogram)									**cardinal numerosity (thermometer)**								
1	1.000	0.176	0.563	0.389	0.408	0.308	0.545	-0.149	0.617	1.000	0.667	0.509	0.408	0.333	0.272	0.218	0.167	0.111
2	0.176	1.000	0.077	0.250	-0.268	0.179	0.355	0.454	0.029	0.667	1.000	0.764	0.612	0.500	0.408	0.327	0.250	0.167
3	0.563	0.077	1.000	0.421	0.555	0.497	0.338	-0.188	0.664	0.509	0.764	1.000	0.802	0.655	0.535	0.429	0.327	0.218
4	0.389	0.250	0.421	1.000	0.337	0.383	0.462	-0.173	0.423	0.408	0.612	0.802	1.000	0.817	0.667	0.535	0.408	0.272
5	0.408	-0.268	0.555	0.337	1.000	0.318	0.142	-0.578	0.570	0.333	0.500	0.655	0.817	1.000	0.817	0.655	0.500	0.333
6	0.308	0.179	0.497	0.383	0.318	1.000	0.442	-0.200	0.330	0.272	0.408	0.535	0.667	0.817	1.000	0.802	0.612	0.408
7	0.545	0.355	0.338	0.462	0.142	0.442	1.000	-0.118	0.301	0.218	0.327	0.429	0.535	0.655	0.802	1.000	0.764	0.509
8	-0.149	0.454	-0.188	-0.173	-0.578	-0.200	-0.118	1.000	-0.181	0.167	0.250	0.327	0.408	0.500	0.612	0.764	1.000	0.667
9	0.617	0.029	0.664	0.423	0.570	0.330	0.301	-0.181	1.000	0.111	0.167	0.218	0.272	0.333	0.408	0.509	0.667	1.000
	one-hot									**robot fingers**								
1	1.000	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	1.000	0.660	0.489	0.475	0.378	0.295	0.220	0.144	0.145
2	-0.125	1.000	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	0.660	1.000	0.746	0.726	0.578	0.450	0.336	0.219	0.221
3	-0.125	-0.125	1.000	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	0.489	0.746	1.000	0.571	0.774	0.604	0.450	0.293	0.296
4	-0.125	-0.125	-0.125	1.000	-0.125	-0.125	-0.125	-0.125	-0.125	0.475	0.726	0.571	1.000	0.860	0.670	0.501	0.326	0.330
5	-0.125	-0.125	-0.125	-0.125	1.000	-0.125	-0.125	-0.125	-0.125	0.378	0.578	0.774	0.860	1.000	0.779	0.582	0.379	0.383
6	-0.125	-0.125	-0.125	-0.125	-0.125	1.000	-0.125	-0.125	-0.125	0.295	0.450	0.604	0.670	0.779	1.000	0.751	0.489	0.494
7	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	1.000	-0.125	-0.125	0.220	0.336	0.450	0.501	0.582	0.751	1.000	0.657	0.664
8	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	1.000	-0.125	0.144	0.219	0.293	0.326	0.379	0.489	0.657	1.000	-0.128
9	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	-0.125	1.000	0.145	0.221	0.296	0.330	0.383	0.494	0.664	-0.128	1.000

Database



Supplementary Figure 2. Examples of Short-Time Fourier Transforms Spectrograms for digits from 1 to 9.

Statistical Analysis

The performance measure is the accuracy rate, which is the number of correctly classified digits divided by the total number of spoken digits in the test set, and it was calculated on the test set after each training epoch. Tables report the average accuracy rate and the standard deviation of 32 independent runs as described in the pseudo-code below. In Figures, the graphs present the performance or average differences between models with varying epochs.
The pseudo-code below describes the procedure to calculate the average accuracy, including the best accuracy for the minimum loss.

For each k ∈ {128,512,1024,2788,13942, 27884}
For each i ∈ [1,32]

$$best\_epoch_i = \min\left(classifier\_loss_{k,i}\right)$$

For each epoch

$$accuracy(k, epoch) = average_{i\in[1,32]}(accuracy_i(k, epoch))$$

$$final\_accuracy_{(k)} = average_{i\in[1,32]}(accuracy_i(k, best\_epoch_i))$$

The best performance in the training history is calculated as the maximum of the average accuracies per epoch. The significance of differences was verified with the Student t-test, where we considered the differences statistically significant when $p<0.05$.
To evaluate the effect size of the differences, we calculated Cohen's d^2, which is defined as the difference between two means divided by a standard deviation for the data:

$$d = \frac{\mu_1 - \mu_2}{std_p}$$

where $\mu_1 - \mu_2$ is the difference between the two averages (μ) and std_p is the pooled standard deviation:

$$std_p = \sqrt{\frac{std_1^2 + std_2^2}{2}}$$

This means that if d is 1, the two groups' averages differ by one standard deviation; a d of .5 means that the two groups' averages differ by half a standard deviation; and so on. Cohen suggested that d=0.2 be considered a “small” (trivial) effect size, 0.5 represents a “medium” effect size and 0.8 a “large” effect size [2].

References

1. De La Cruz, V. M., Di Nuovo, A., Di Nuovo, S. & Cangelosi, A. Making fingers and words count in a cognitive robot. *Front. Behav. Neurosci.* **8,** 13 (2014).

2. Cohen, J. *Statistical Power Analysis for the Behavioural Science (2nd Edition)*. *Statistical Power Analysis for the Behavioural Science* (Routledge, 1988).