

Sensorimotor Prediction with Neural Networks on Continuous Spaces

Michaël Garcia Ortiz^(✉)

SoftBank Robotics Europe - AI Lab, 43, rue du Colonel Pierre Avia,
75015 Paris, France

mgarciaortiz@softbankrobotics.com

Abstract. In the context of Developmental Robotics, we propose to learn how the sensations of a robot are modified by its action. Many theories of Artificial Intelligence argue that sensorimotor prediction is a fundamental building block of cognition. In this paper, we learn the sensorimotor prediction on data captured by a mobile robot equipped with distance sensors. We show that Neural Networks can learn the sensorimotor regularities and perform sensorimotor prediction on continuous sensor and motor spaces.

1 Introduction

State-of-the-art algorithms in Machine Learning, and more recently in Deep Learning [6], provide very powerful modules for perception [7] and action [15] for robotic systems. End-to-end skill learning [8] also demonstrated the feasibility of fully trainable architectures. However, these approaches rely heavily on extensive labeling of datasets [14], and expert design and fine-tuning of the learning algorithms. Applied to robotics, these approaches allow a certain level of skills in perception and action, but don't allow a robot to learn new concepts. The challenge of genuine artificial intelligence for robots remains: how can a robot learn how to perceive the world and learn to act in it by itself?

Current applications of Machine Learning for robotics are very far from how humans and animals learn. Babies, for example, don't have access to labeled information. In order to recognize objects, environments, or other persons, children build their own model of the world by interacting with it. Developmental robotic [2] proposes to take inspiration from animals and human intelligence. The goal is to identify the mechanisms that endow cognitive development, in order to allow robot to develop autonomously. We place ourselves in this context, where an agent has to learn, without prior knowledge or labels provided by the engineer, how to perceive its environment and how to act in it.

The agent senses its environment through sensors, and acts on its environment by controlling its motors. This sensorimotor loop is the only information about the world accessible to the agent. By learning how actions modify sensor values, the agent builds a model of its interaction with the world. This model can be used to predict future sensory states depending on motor commands,

in order to avoid undesirable states or reach desirable ones. Many fundamental [3, 4, 12] as well as practical [17] theories of intelligence and cognition argue that sensorimotor prediction is an elementary building block of cognition.

In this paper, we consider a robotic agent, endowed with distance sensors, moving in a fixed environment. We compare neural network architectures that can be used to learn sensorimotor prediction in the case of continuous sensor and motor spaces.

2 Related Works

Our approach differs from existing approaches by not using explicit vector quantization for the input and output space, which means that we perform prediction as regression and not classification. Additionally, the robot learns to control its sensorimotor space autonomously, without relying on features implemented by the designer.

In [18], the authors use a mobile robot navigating in a fixed environment. They use sensorimotor prediction as a forward model to perform prediction for an arbitrary motor program, in order to choose a sequence of actions that prevents the robot from colliding with obstacles. The speed of the robot is constant, but its direction can change. Their architecture relies on vector quantization for sensors, and context units that serve as a temporal internal representation that disambiguate situation and help the learning.

In [11], a sensorimotor prediction approach is used for navigation in the case of a simulated agent. The architecture presented uses a lot of prior knowledge: the sensorimotor prediction is based on already pre-processed features, and a predicting algorithm for each constituent of the environment. This is not compatible with a developmental approach, where the constituents of the environment are not known in advance but should be discovered.

In [10], the authors propose a computational model for sensorimotor contingencies and an algorithm for predicting future states of the sensors, in the case of a robot equipped with distance sensors, moving in a rectangular empty environment. The sensor values are quantized in 3 values, and the robot translates on 2 axis (no rotation). This work doesn't provide technical solutions to learning sensorimotor prediction in more complex (possibly continuous) spaces.

In [5], sensorimotor prediction is used as a forward model in order to control a robotic arm. The authors consider continuous sensor and motor spaces, and perform sensorimotor prediction on the configurations of the joint of the arm. Even if they showed robust online learning, the notion of multiple environmental element (such as walls and corners in our case) influencing this prediction is missing.

Even if the presented approaches are related to our work, the simplification of motor and sensor space (through vector quantization) doesn't guarantee that they can be used in more complex environments or with more complex sensors.

3 Presentation of the Approach

The agent acts in its environment through motors M_i that can be controlled by sending a motor command m_i , and it senses its environment through the sensors S_i receiving sensory signals s_i . At each timestep t , the agent sends a motor commands $m_i(t)$ and receives sensor values $s_i(t)$. The agent experiences sequences of $\{s(t), m(t)\}$, which are collected in order to learn a sensorimotor predictive model. The sensorimotor prediction is learned offline, and online incremental learning is not considered in this work. We focus on whether learning sensorimotor prediction can be achieved using standard neural networks, however, we will need to prove in future works that an incremental approach can be used, for the results to be used in a developmental robotics setting.

3.1 Prediction as a Regression Problem

Learning the sensorimotor prediction can be approached as a regression problem. The task of the regression algorithm is to learn the mapping. $(s(t), m(t) \rightarrow \Delta s(t+1))$ where $\Delta s(t+1) = s(t+1) - s(t)$. Our prediction algorithm is trained using neural network. We will propose several architectures in order to learn the sensorimotor prediction.

We predict the change in sensory values $\Delta s(t+1)$ instead of learning to predict the future value of the sensory input $s(t+1)$. Fundamentally, what interests us is to learn how motors affect sensors, so it makes sense to predict this change. On a more pragmatic level, by only predicting the change and not the raw value, we optimize the capacity of the neural network, and we avoid representing redundant information in the network. This is in line with the predictive coding approach [3,4].

Each architecture (illustrated in Fig. 1) is trained using gradient descent. For each training example, the inputs $((s(t), m(t)))$ are used to formulate a prediction $\Delta s(t+1)_{pred}$ that we compare to the actual output $\Delta s(t+1)$. We compute the mean squared error as the average (over a batch of training samples) squared difference between the actual (desired) and the predicted output. We use this error signal to update the weights of the networks using gradient descent. Once the network is trained, we use it to perform prediction on a separate portion of the dataset and evaluate its prediction capabilities.

3.2 Neural Network Architectures for Sensorimotor Prediction

Feed Forward Neural Network. As a baseline for learning sensorimotor prediction, we propose to use a standard Feed-Forward Neural Network. The network takes as input a concatenation of $s(t)$ and $m(t)$. It is composed of several hidden layers of either sigmoid or rectifier linear units. The output layer is a linear layer connected to $s(t+1)$, and all the layers are fully connected.

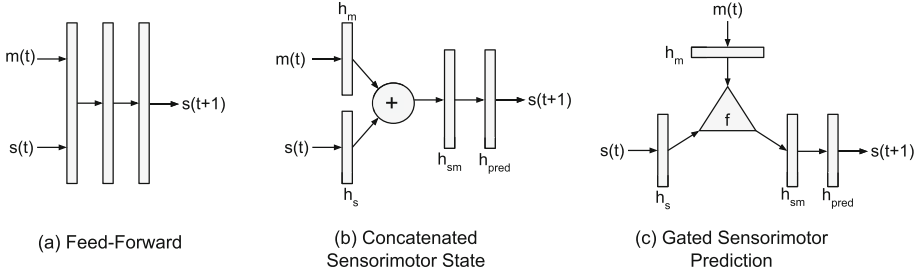


Fig. 1. Different architectures used for sensorimotor prediction

Concatenated Sensorimotor State. We want to compare the standard Feed-Forward architecture with an architecture where representations for sensors and motors are learned separately, and then concatenated to perform sensorimotor prediction. First, $s(t)$ is projected (fully connected) to a representation layer h_s , and similarly $m(t)$ is projected to a different representation layer h_m . h_s and h_m are concatenated and projected to a layer h_{sm} . This representation, supposed to represent the sensorimotor state of the robot, is then projected to a prediction layer h_{pred} , which in turn is used to predict the output $s(t+1)$.

Gated Sensorimotor Prediction. As suggested in [16], gated interactions can be used to learn sensorimotor prediction. We take inspiration from Gated Neural Networks to propose an architecture where motors are influencing sensors through multiplicative gating interactions using factors. $s(t)$ is projected to a representation layer h_s , and $m(t)$ is projected to a representation layer h_m . h_s and h_m are then each projected on the factors f of the gating neural network. The factor activations are multiplied and fully connected to a representation layer h_{sm} , in turn projected to a prediction layer h_{pred} used to predict the output $s(t+1)$.

3.3 Long-Term Prediction

An important property of sensorimotor prediction is the capacity to predict future sensory states by simulating motor sequences. In order to predict multiple timesteps into the future, we propose to use the result of the sensorimotor prediction ($\Delta s(t+1)$) at time $t+1$ to update the value of $s(t+1)$: $s(t+1) = s(t) + \Delta s(t+1)$. We can, in turn, perform sensorimotor prediction ($s(t+1), m(t+1) \rightarrow \Delta s(t+2)$).

This approach doesn't consider long-term dependencies, and we expect that it is not suitable to learn long-term predictions. However, we argue that it is sufficient to predict the immediate evolution of the sensor values depending on the motor sequence.

4 Experimental Setup

We use a Thymio-II robot [13] for our experiments. We use its 5 front distance sensors, and each sensor encodes a distance value as an integer in the range [1500, 5000], approximately corresponding to [13 cm, 0 cm]. Motors are controlled in speed by integer commands in the range $[-500, 500]$, approximately corresponding to a range of speeds of $[-15 \text{ cm/s}, 15 \text{ cm/s}]$. For our experiments, we limit the range of the motors to $[-200, 200]$. The sensor values are rescaled as floats in the range $[0.0, 1.0]$ and motor commands are rescaled as floats in the range $[-1.0, 1.0]$. We control the robot using the library Aseba [9]. The frame rate is superior to 10 Hz, and we interpolate the sensor readings at 5 Hz. The environment of the robot is a rectangular empty maze of size $60 \times 80 \text{ cm}$. Every 2 s, the robot picks randomly a new motor command (see Fig. 2(a)). We collected 40 sequences of 120 min each (around 1.4 million data points). We illustrated the sensations of the robot in Fig. 2(b).

The sensors are noisy, and the transfer function from sensor reading to actual distance is not linear. Additionally, these values depend on the reflective properties of the surface. The perception of what a wall or a corner is can't lie only in the values of the sensors, as these values change dramatically depending on their calibration, the surface, or their orientation. This highlights the pertinence of sensorimotor contingency theory [12], which states that the world imposes regularities on the way sensor values are changed by action, and that the mastering of these regularities is what constitutes perception.

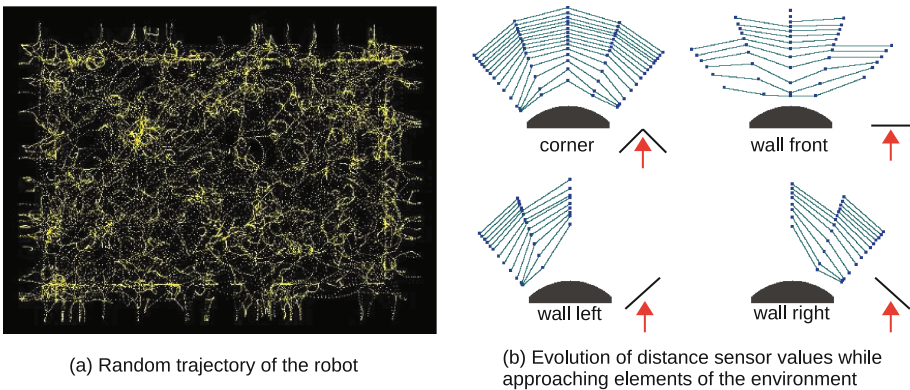


Fig. 2. (a) Top-down visualization of the trajectory of the robot during one recorded sequence. (b) Sensor values of the robots captured while approaching different elements of the environment (illustrated with a black line for the environment and red arrow for the direction of the robot). Each line corresponds to a sensor reading at a certain distance from the element of the environment. The displacement of the robot between two consecutive lines is 0.5 cm. (Color figure online)

5 Experiments and Results

We use Tensorflow [1] to program our Neural Networks. We train the different architectures for 1 million iterations, with a batch size of 32 and a learning rate of 0.01, and we compute the average mean squared error of prediction over 100000 random samples from a separate dataset. For clarity, we display the MSE of the prediction multiplied by a factor 100.

Prediction Error for the Baselines. We compare different configurations of the baseline architecture presented in Sect. 3. The results are presented in Table 1. Rectifier Linear units are performing better than Sigmoid units. We suppose that it is because of the continuous nature of the mapping it tries to learn. In the following experiments, we will use Rectifier Linear Units.

Comparison of Structured Networks. We compare the two architectures that learn separate sensory and motor representations. We found that fixing the representation size of h_m to 3 is sufficient. We fixed the number of factors to 256 for the Gated sensorimotor prediction, and experimented with multiple sizes of h_s and h_{sm} . As we can see from Table 2, splitting the learning of sensors and motors doesn't improve the results. For equivalent size of the network, a standard Feed-Forward network with Rectifier Linear Units performs better than the network with separate learning of sensory representations. It might mean that the network benefits from very early sensorimotor representations.

Illustration of the Long-Term Prediction. We use an already trained model (Feed-Forward with 3 layers and 128 units per layer) to predict the future values of the sensors over multiple timesteps, depending on different motor commands. We can generate predictions of the change in the sensor space, and reconstruct future values of the sensors. The evolution of sensor values depending on the motor commands is presented in Fig. 3. One line correspond to one prediction, and we plot one prediction every 3 timestep. As can be observed, the chaining of prediction can be used to successfully predict the future values of the sensors depending on the motor commands.

Table 1. MSE for the baseline Feed-Forward neural networks

	Layer size	1 layer	2 layers	3 layers
Rectifier linear units	32	0.202	0.174	0.159
	64	0.187	0.159	0.149
	128	0.179	0.152	0.144
Sigmoid units	32	0.211	0.200	0.290
	64	0.210	0.199	0.247
	128	0.211	0.194	0.242

Table 2. MSE for the structured neural networks

		Size of h_{sm}					
		Concat. sensorimotor state			Gated sensorimotor state		
		32	64	128	32	64	128
Size of h_s	32	0.166	0.157	0.151	0.161	0.153	0.151
	64	0.164	0.156	0.150	0.158	0.154	0.149
	128	0.163	0.155	0.150	0.155	0.152	0.148

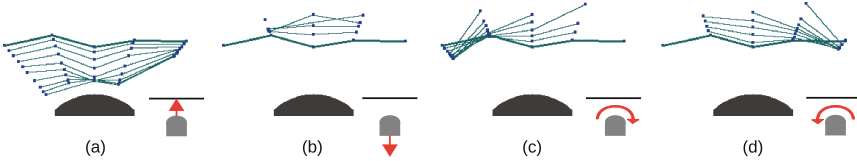


Fig. 3. Prediction across multiple timesteps, for different motor commands. The robot is facing a wall (bold line), and a trained model is used to predict the future sensory values depending on the motor commands. (a) corresponds to a movement forward, (b) to a movement backward, (c) to a rotation to the right and (d) to the left.

6 Conclusion

In this paper, we motivated the use of sensorimotor prediction in order for an autonomous robot to acquire knowledge about the regularities of interaction with its environment. We presented different neural architectures for this sensorimotor prediction, and showed that Feed-Forward Neural Networks with Rectifier Linear Units can be used to learn on continuous sensorimotor spaces. We also found that early sensorimotor representations might be beneficial to the overall learning, as learning sensor and motor separately appears to be detrimental to the overall quality of prediction. Finally, we showed that it was possible to chain the predictions in order to simulate future sensory values of the robot depending on its motor commands.

In future works, we want to investigate the use of predictive coding as a means to perform an efficient vector quantization on continuous sensory space. This predictive coding strategy will also be a way to transform our current framework into an incremental learning framework. Additionally, we want to propose a probabilistic approach based on generative models. Another direction is to use the sensorimotor representations learned in the context of continuous sensorimotor prediction, and use them to predict discrete events, such as collisions. Another axis of development is the use of latent variables to represent the context of the robot in its environment. Finally, we want to investigate the possibility of using genetic algorithm to learn an efficient structure for sensorimotor prediction.

References

1. Abadi, M., et al.: TensorFlow: Large-scale machine learning on heterogeneous systems (2015). software available from tensorflow.org
2. Cangelosi, A., Schlesinger, M.: *Developmental Robotics: From Babies to Robots*. The MIT Press, Cambridge (2014)
3. Clark, A.: Whatever next? predictive brains, situated agents, and the future of cognitive science. *Behav. Brain Sci.* **36**(3), 181–204 (2013)
4. Friston, K.: The free-energy principle: a unified brain theory? *Nat. Rev. Neurosci.* **11**(2), 127–138 (2010)
5. Ghadirzadeh, A., Btepage, J., Kragic, D., Bjrkmán, M.: Self-learning and adaptation in a sensorimotor framework. In: 2016 IEEE International Conference on Robotics and Automation (ICRA), pp. 551–558, May 2016
6. Goodfellow, I., Bengio, Y., Courville, A.: *Deep Learning*. MIT Press (2016)
7. Krizhevsky, A., Sutskever, I., Hinton, G.E.: Imagenet classification with deep convolutional neural networks. In: Pereira, F., Burges, C.J.C., Bottou, L., Weinberger, K.Q. (eds.) *Advances in Neural Information Processing Systems 25*, pp. 1097–1105. Curran Associates, Inc. (2012)
8. Levine, S., Finn, C., Darrell, T., Abbeel, P.: End-to-end training of deep visuomotor policies. *J. Mach. Learn. Res.* **17**(1), 1334–1373 (2016)
9. Magnenat, S., Rétornaz, P., Bonani, M., Longchamp, V., Mondada, F.: ASEBA: a modular architecture for event-based control of complex robots. *IEEE/ASME Trans. Mechatron.* **16**(2), 321–329 (2011)
10. Maye, A., Engel, A.K.: Extending sensorimotor contingency theory: prediction, planning, and action generation. *Adapt. Behav.* **21**(6), 423–436 (2013)
11. Möller, R., Schenck, W.: Bootstrapping cognition from behavior - a computerized thought experiment. *Cogn. Sci.* **32**(3), 504–542 (2008)
12. O'Regan, J., No, A.: A sensorimotor account of vision and visual consciousness. *Behav. Brain Sci.* **24**(05), 939–973 (2002)
13. Riedo, F., Chevalier, M.S.D., Magnenat, S., Mondada, F.: Thymio II, a robot that grows wiser with children. In: 2013 IEEE Workshop on Advanced Robotics and its Social Impacts (ARSO), pp. 187–193. IEEE (2013)
14. Russakovsky, O., Deng, J., Su, H., Krause, J., Satheesh, S., Ma, S., Huang, Z., Karpathy, A., Khosla, A., Bernstein, M., Berg, A.C., Fei-Fei, L.: ImageNet large scale visual recognition challenge. *Int. J. Comput. Vision (IJCV)* **115**(3), 211–252 (2015)
15. Schaal, S.: Dynamic movement primitives - a framework for motor control in humans and humanoid robotics. In: Kimura, H., Tsuchiya, K., Ishiguro, A., Witte, H. (eds.) *Adaptive Motion of Animals and Machines*. Springer, Tokyo (2006). doi:[10.1007/4-431-31381-8_23](https://doi.org/10.1007/4-431-31381-8_23)
16. Sigaud, O., Droniou, A.: Towards deep developmental learning. *IEEE Trans. Cogn. Dev. Syst.* **8**(2), 99–114 (2016)
17. Sutton, R.S., Barto, A.G.: *Introduction to Reinforcement Learning*, 1st edn. MIT Press, Cambridge (1998)
18. Tani, J.: Model-based learning for mobile robot navigation from the dynamical systems perspective. *IEEE Trans. Syst. Man Cybernetics Part B* **26**(3), 421–436 (1996)