CSCI E-80A (25387):

Introduction to Artificial Intelligence

Companion Course to E-100 Science of Intelligence

© Prof. Brian Subirana

Director, MIT Auto-ID Laboratory

77 Mass Ave, Cambridge MA 02139

Week 3: Coding Assignment

CENTER FOR
Brains
Minds+
Machines

Table of Contents

What is a Smart Machine?

The main concern on Artificial Intelligence discussed today is that deep learning is completely empiricist. This means that there are no underlying logic structures built around a specific problem. The performance of the system is a statistical exercise successful mostly only under very specific conditions.

The first of these conditions is what is known as supervised learning. Most deep learning networks nowadays require for the user to input training examples that are clearly labeled with their appropriate answer, so that the computer can compare its expectations with the actual result and train itself. This separates deep learning from any "smart" organism. When human babies are born for example, they learn to identify their parents or their toys without any direct feedback. How to mimic this organic unsupervised learning in machines is still a hot topic in research, and will inevitably mean an increase in the complexity behind deep learning computations.

The second condition for deep learning to perform is the shear amount of necessary training data. For a simple handwritten digit neural network to be properly trained, it's not unusual to use upwards of tens of thousands of samples, all properly labeled and in the same format. This becomes a problem when trying to solve issues that haven't been recorded properly enough over time. A very good example of this is the fluctuation in world economics. Only in the last 100 years we have been diligently calculating and recording the oscillations in world capital. In the United States of America only around 20 recessions have been recorded in that period of time; training a neural network to predict recessions is therefore extremely challenging because of the lack of germane data.

The definition of intelligence in the context of machine learning and A.I. is blurry and can sometimes be misleading. In 1950, Turing attempted to shed some light on this topic by defining the "Turing Test", where a machine should be able to fool a judge by pretending to be a human[1]. This idea became popular quickly, with most A.I. chatbot efforts being checked against this benchmark. Weizenbaum's ELIZA was the first program to pass the test consistently in 1966[2], which generated doubts on the validity of the test. Other similar tests arose because of this, such as the "Chinese room"[3], with its set of supporters[4] and opponents[5]. With the rise of Deep Learning as the flagship of A.I., intelligence in the context of machines is nowadays often confused with pattern recognition. Both industry and academia flock around concepts such as computer vision or speech recognition; although these are indeed essential in our understanding of human intelligence, its definition can be argued to go beyond these. Human intelligence includes more

---

[1] A.M Turing. Computing machinery and intelligence. October 1950

[2] Weizenbaum, Joseph. ELIZA – A Computer Program For the Study of Natural Language Communication Between Man And Machine, Communications of the ACM (January 1966), 9 (1): 36–45, doi:10.1145/365153.365168

[3] Searle, John. Minds, Brains and Programs. Behavioral and Brain Sciences, (1980) 3 (3): 417–457, doi:10.1017/S0140525X00005756

[4] M. Bishop & J. Preston (eds.) Essays on Searle's Chinese Room Argument. Oxford University Press. (2001)

[5] Hauser, Larry. "Searle's Chinese Box: Debunking the Chinese Room Argument". Minds and Machines (1997), 7 (2): 199–226,

complex cognitive tasks, such as modelling the world, imagining things, explaining and understanding what we see, solving problems and planning actions to make these real. Ultimately, not only are we able to model the world, but we can modify these models and build new ones as we learn[6].
Spelke defines a human "knowledge core" that englobes the tools and skills infants use to be able to learn from experience[7]. Skills such as object permanence[8] are present in infants very early on. Research suggests that a common-sense core provides humans a basic understanding of physical objects and substances[9], intentional agents and their interactions[10] – "intuitive theories", or abstract systems of knowledge about physics[11] (forces, masses…) and psychology[12] (desires, beliefs, plans…). Computationally defining this core is challenging, let alone understanding and mimicking how this core is trained or acquired by humans in the first place.

Approaching these problems with traditional supervised deep learning algorithms seems to be conceptually wrong, as this cognitive core appears to be hardcoded. Lerer et al attempted to do this for a physics engine, but were confronted with very low training efficiency (200k training samples to predict physics of 2 to 4 cubes)[13]. Battaglia et al took a different approach for the same problem, developing a probabilistic model based on an "intuitive physics engine" that allows for fast and robust inferences in complex natural scenes where crucial information is unobserved[14]. A similar approach was used for an "intuitive psychology engine", mimicking human understanding of subject beliefs and goals[15].

---

[6] Lake, Ullman, Tenenbaum & Gershman, "Building machines that learn and think like people", Behavioral and Brain Sciences (2017). Volume 40 2017 , e253

[7] Spelke, Elizabeth S. Core knowledge. American Psychologist, Vol 55(11), Nov 2000, 1233-1243

[8] Renée Baillargeon, Elizabeth S. Spelke, Stanley Wasserman. Object permanence in five-month-old infants. Cognition. Volume 20, Issue 3, 1985, Pages 191-208

[9] McCloskey M, Caramazza A, Green B (1980) Curvilinear motion in the absence of external forces: Naive beliefs about the motion of objects. Science 210(4474):1139–1141.

[10] Southgate V, Csibra G. Inferring the Outcome of an Ongoing Novel Action at 13 Months. Developmental Psychology 2009, Vol. 45, No. 6, 1794–1798.

[11] Ernő Téglás, Edward Vul, Vittorio Girotto, Michel Gonzalez, Joshua B. Tenenbaum, Luca L. Bonatti. Pure Reasoning in 12-Month-Old Infants as Probabilistic Inference. Science 27 May 2011: Vol. 332, Issue 6033, pp. 1054-1059 DOI: 10.1126/science.1196404

[12] Shari Liu, Tomer D. Ullman, Joshua B. Tenenbaum, Elizabeth S. Spelke. Ten-month-old infants infer the value of goals from the costs of actions. Science 24 Nov 2017: Vol. 358, Issue 6366, pp. 1038-1041 DOI: 10.1126/science.aag2132

[13] Adam Lerer, Sam Gross, Rob Fergus. Learning Physical Intuition of Block Towers by Example. ICML'16 Proceedings of the 33rd International Conference on International Conference on Machine Learning (2016)- Volume 48 Pages 430-438

[14] Peter W. Battaglia, Jessica B. Hamrick, and Joshua B. Tenenbaum. Simulation as an engine of physical scene understanding. Proceedings of the National Academy of Sciences (2013) 110(45).

[15] Julian Jara-Ettinger, Hyowon Gweon, Laura E. Schulz, Joshua B. Tenenbaum. The Naïve Utility Calculus: Computational Principles Underlying Commonsense Psychology. Trends in Cognitive Sciences, Volume 20, Issue 10, October 2016, Pages 785

It is clear that we are still very far from being able to mimick all human cognitive tasks in machines, we can nonetheless take inspiration on the way we think to make better algorithms.

MNIST Convolutional Keras Tutorial

Up to this point we have only talked about the most basic kind of layer in deep learning. Feedforward (or Dense) layers are very easy to use, relatively quick to train and can be quite performant for simple tasks. Convolutional layers on the other hand (sometimes refered to as CNNs or Convolutional Neural Networks), allow to repeatedly apply a set of filters over a whole image, which makes them very effective for computer vision. One of the keys for the success of convolutional layers in computer vision is that they do not require for the input to be flattened (which we had to do for our Dense network), mainitaining spatial correlations in its computations. In our example, essentially a CNN looks for specific traits in the image to determine whether it is or not a digit. Although this is not effectively how a trained CNN works, we could imagine how if the model spotted a "circle" in the image, the model would think the number might be a "0", or maybe a "9" or "6". If instead the model spotted a straight vertical line, it might think it's a "1", "7" or "4". In actuality, trained CNN filters can be rather cryptic, but conceptually the network is indeed looking for recognizable features. not cover this here you are free to read the article below to understand how this is done.

Convolutional Layers are not as popular for Natural Language Processing as text doesn't allow to leverage convolutions strength in revealing 2D and 3D spatial relations, so for the next assignment you will only need to test your convolutional network with the computer vision and speech recognition assignments.

We will not delve too deeply into how convolutional networks work, but you are encouraged to expand your knowledge by reading up on some of the resources below.

Further Reading (Optional)

A Comprehensive Guide to Convolutional Neural Networks — the ELI5 way. *By Sumit Saha.*
https://towardsdatascience.com/a-comprehensive-guide-to-convolutional-neural-networks-the-eli5-way-3bd2b1164a53
This article will give you a good in depth overview of how Convolutional Neural Networks work, as well as explain the main advantages of using them for Computer Vision. You will learn as well about Max Pooling layers, which you will need to implement in the next assignment.

Building a Convolutional Neural Network (CNN) in Keras. *By Eijaz Allibhai.*
https://towardsdatascience.com/building-a-convolutional-neural-network-cnn-in-keras-329fbbadc5f5
This article goes over how to build a CNN in Keras, you will find this article extremely helpful for the next assignment…

You will now be tasked with adding convolution to the `generic_vns_function` described earlier. Note that you will not be getting a lot of guidance here, which means you will need to do some research on your own.

MNIST Convolutional Layers Assignment

This assignment is based on the code explained in the "*Deep Learning with Keras*" section from assignment 2, make sure you review it before you get on with this assignment.

All the work you do in this section should be properly recorded and compiled into a final report, combined with the other exercises. You will find the guidelines for this report at the end of the assignment.

- Add a convolutional (CNN) layer, a max pool layer and a flattening layer to the `generic_vns_function`. You will have to change the shape of the input for this to work! (*Check the Keras docs at https://keras.io to learn more*)
- Train, test and compare performance with the NN from the previous Dense Layers Assignment for both handwritten digits and speech.
- Tinker with batch size, learning rate, the convolutional layer's hyperparameters (filter size, etc) and epochs to obtain better performance.