

# LEVELS OF ABSTRACTION IN MACHINE LEARNING

Henry Blanchette

## 1 Introduction

The techniques of machine learning (ML) in the broader field of artificial intelligence (AI) has undeniably made huge headway recently. However there still seem to be some significant inadequacies with the state-of-the-art machine *intelligence* even for relatively narrow tasks. For example, modern ML in the form of a convolutional neural network (CNN) has achieved super-human ability in the classic computer game of Breakout, where success is measured by how quickly the algorithm can beat the game. The CNN achieved this level of skill in Breakout from “the ground up”, meaning that it was given no special information about the game other than control over input to the game (moving the paddle) and access to the output from the game (looking at the game screen).

A human reader recognizes that, in order to gain skill at Breakout, one needs to recognize or learn certain patterns about the physics of the game. One learns that the ball bounces off the paddle and this is the only way to keep the ball from going out of bounds, and that the ball bounces off of the bricks causing them to disappear and scores a point. Somewhere along the way, one learns that hollowing out a tunnel through all the brick layers lets the ball bounce around consecutively in the top area scoring lots of points.

But is this how the CNN learned to play? Are there analogous concepts of *ball*, *paddle*, *brick*, and *tunnel* represented within the neural network? One way to probe for answers is to change some feature of the game that requires the player to make use of one of these abstractions in order to readjust their strategy. And this is where the CNN exposes the fact that the knowledge that it has in the form of intricate weighted connections between nodes in its layers may not likely resemble anything like we’d expect from a human that is considered knowledgeable or skilled at breakout. If the game is modified so that the paddle is slightly higher on the vertical, a human player will have no trouble recognizing how to control this paddle and apply the same skills that they may have learned from the normal version of the game (e.g. tunneling). But the CNN utterly fails at this task. When the paddle is moved, it loses all apparently skill at the game. When it is trained on the new version of the game, starting from its previously-skilled configuration of weights, it basically scrambles its weights randomly before it re-achieves its previous level of skill. The CNN was unable to salvage any of its knowledge when it needed to account for even such a minor change.

The significance of this observation is that it suggests the CNN likely didn’t *understand* the game of Breakout in the same way we expect a human to understand the game, even though the CNN was super-human in performance. There are many other examples of this phenomenon, that Melanie brought up during her visit, such as how an image classifier CNN can become highly-proficient at recognizing images of, say, cows, yet when presented with an empty field of grass the CNN still sees cows. Or in some of the more extreme examples, one can engineer a *adversarial example* that is carefully designed to take advantage of an image classifier CNN’s trained structure to make it recognize whatever the designer wishes in an image while making the image appear relatively unchanged to a human viewers.

## 2 Levels of Abstraction

Recalling the Breakout case, the obvious question to ask of the CNN is “if the CNN didn’t learn even basic abstractions like *the paddle*, then what *did* the CNN learn that allowed it to play so amazing well?” While it is possible that CNN learned a few human-recognizable abstractions, the fact that it misses many that humans would find fundamental to understanding (e.g. the abstraction of *the paddle*) suggests that the CNN probably didn’t learn many. In some way the CNN must have found an extremely superficial way to understand Breakout, but still well enough to impress human players who do have a demonstrably deeper understanding.

A simple and likely explanation for the CNN’s behavior that accounts for its pattern-obliviousness is that it discovered a sort of “cheat” in the form of statistical correlations between certain inputs (pixels and game moves) that were easier to learn than the higher-order abstractions that humans find intuitive and typically expect from a good player (human or otherwise). The fact that the first significant attempts at designing intelligent systems was a symbolic approach, where human-intuitive patterns were *hard-coded* into the system. For example, early chess-player AIs needed heuristics such as “treat the queen as having a relative value of 9 when evaluating piece trade-offs” explicitly represented in their decision function).

A human reader should find it obvious that a human player attempting to learn Breakout can skip many of the early steps that were outlined, such as “the ball bounces off the paddle and this is the only way to keep the ball from going out of bounds.” A human can look at the Breakout game for less than a second and just know intuitively that the colored disk that is translated across the pixels of the game screen is an individual *object* and that it will *bounce* off of the boundaries of the screen and the bricks at the top in such a way that the angle of inflection will equal the angle of refraction of the collision. Humans are expected to have a sort of intuitive physics and concept of object-ness (among other concepts) that a CNN (at least, before training) is not expected to have. In this way, a continuation of the explanation for the CNN’s failure to handle a displaced paddle is that, as an example concept, the pattern of the paddle as a persistent object was never developed withing the CNN’s internal representation of Breakout.

Furthermore, it could be possible that such a concept of the paddle is not required for humans to play Breakout well either; the fact that humans apply the concept of *object* to the paddle could be completely accidental, negligible, or even inhibitory to the ability of a human to, in principal, excel at Breakout. As the CNN demonstrates, the ability to play the unmodified version of Breakout at an expert level clearly does not necessarily require such a conceptualization. I use the example of the *object* concept of the paddle because it seems an intuitively plausible explanation but I have not tested that hypothesis and am not aware of other’s having tested it either. My argument relies just on the fact that there is *some* concept that humans intuitively use when playing Breakout that they did not typically learn by playing Breakout — a concept which the CNN ends up not learning and is only manifested in bad play when the game is modified in a way that requires the concept for adaptation. The question of how humans developed the sorts of intuitions that have been invoked is a very interesting one, and will be addressed later.

So far it has been established that CNNs lack some higher-order intuitions about the tasks on which they are trained. The way that a CNN can still achieve prowess at its designated tasks have been referred to as “cheats,” in the form of superficial statistical and correlations between inputs, as opposed to true, deeper understanding. These “cheats” are examples of **first-order patterns**. A first-order pattern is one that relies exactly on the values of the first-order information about inputs i.e. the first layer of the CNN. Respectively, **higher-order patterns** are those that rely on

the exact values of higher-order inputs (in some hierarchy of abstract) i.e. the deeper layers of the CNN (so, there could be second-order, third-order, etc.<sup>1</sup>).

The first layer has a unique node for each feature that is fed to the CNN. For example, if the input is an  $(m \times n)$ -pixel image then each node of the first layer corresponds the red-green-blue (RGB) value of one of the  $mn$  pixels. Due to the structure of a CNN, most first-order patterns are easier to *learn* (train to optimize using the typical training procedures of CNNs) than most higher-order patterns. This is because the weights of the network are initialized randomly and raw inputs are given to the first layer.

Consider the following toy example of a first-order pattern (which is also used in an experiment layer) just to demonstrate the point. Suppose there is a CNN with an input layer,  $x$  internal fully-connected layers, and an output layer that consists of two nodes (the output nodes). The first output node indicates the relative probability that the inputted image is solid black, and the second output node indicates the relative probability that it is not solid black. If  $x$  was 0 i.e. there are no internal layers, then the CNN just needs to independently maximize each of the weights of connections going into the first output node and independently minimize each of the weights going into the second. This maximization is very easy for a gradient-descent training procedure (the typical training procedure) because it is exactly monotonic — there is no interaction between features (pixel values). However for each layer added into the network ( $x > 0$ ), the same kind of process happens but is just slowed down and buffered by the noise that the added layers introduce.

Observe that it is not, in principal, impossible for the CNN in this example to use the deeper layers to develop abstractions e.g. one node in the 10th layer sums the number of black pixels on the left half of the image and a different node in the 10th layer sums the number of white pixels on right half of the image. The key result is just that these sorts of abstractions are improbable, and more efficient weightings that do not develop such abstractions will be trained to first in the training process. Even in the case that there exist some more efficient higher-order abstractions, if training reaches a local maximum using low-order abstraction it is not always likely and often unlikely that the training will ever deviate towards the former.

This line of thinking points to a simple explanation for the *frame problem*. The frame problem is that of developing AI that is just as adept as current narrow AI but also able to apply its learned skill generally (at least, as generally as humans seem to do). The seeming paradox that superhuman Breakout-playing CNN can't even deal with a slight change of paddle height is resolved by the (previous) explanation that the human conception of the paddle is not likely very important for becoming Breakout-adept, and that there's no reasons to expect the CNN to learn such a thing given this (in addition to the fact that it never trained with this Breakout variant). In fact, I would go further to say that the fact that CNNs can so quickly become so good at Breakout without some concepts that humans consider fundamental and intuitive is evidence that such concepts are inhibitory for playing the game best. Considering beyond just breakout, this argument applies broadly to any task for which CNNs succeed on test on training scenarios but fail on generalized

---

<sup>1</sup>Though this hierarchy may seem well-defined in how I present it, I intentionally make the distinction just between first-order and higher-order. I suspect that the representation of higher-order patterns within a CNN (or a generic neural network) will have significant interplay between, say, the 10th and 11th layers. In this way, there is decreasing significance to the distinction between layers the deeper you go. Again, I focus mainly the distinction on that between the first layer and those beyond because this is where the distinction is the most significant (given this *decreasing returns on abstraction* model) and also relates most specifically to concrete explanation and experimentation; artificial intelligence researches still consider the human interpretation of the patterns that CNNs do learn to be a massive open problem.

scenarios. The presence of many general (often higher-order) patterns that humans intuitively pick up have little reason, on their own, to predict the (optimized to be lower-order) patterns that CNNs optimize.

This explanation, however, does not provide a clear path forward for how to solve the frame problem. The next section describes an experiment that aims to explore one possible solution that keeps the previous explanation of the frame problem in mind.

### **3 Experimental Design**

### **4 Results**

### **5 Conclusions**

### **References**