Why Retraining Can Be Harder Than Training

A neural network perspective on learning, unlearning and relearning

Photo by Mary Blackwey on Unsplash

In a rapidly changing world, humans are required to quickly adapt to a new environment. Neural networks show why this is easier said than done. Our article uses a perceptron to demonstrate why unlearning and relearning can be costlier than learning from scratch.

Introduction

One of the positive side effects of artificial intelligence (AI) is that it can help us to better understand our own human intelligence. Ironically, AI is also one of the technologies seriously challenging our cognitive abilities. Together with other innovations, it transforms modern society at a breathtaking speed. In his book “Think Again”, Adam Grant points out that in a volatile environment rethinking and unlearning may be more important than thinking and learning [1].

Especially for aging societies this can be a challenge. In Germany, there is a saying “Was Hänschen nicht lernt, lernt Hans nimmermehr.” English equivalents are: “A tree must be bent while it is young,” or less charmingly: “You can’t teach an old dog new tricks.” In essence, all

these sayings suggest that younger people learn more easily than older persons. But is this really true, and if so, what are the reasons behind it?

Obviously, the brain structure of young people is different to that of older persons from a physiological standpoint. At an individual level, however, these differences vary considerably [2]. According to Creasy and Rapoport, the "overall functions [of the brain] can be maintained at high and effective levels" even in an older age [3]. Aside from physiology, motivation and emotion seem to play vital roles in the learning process [4][5]. A study by Kim and Marriam at a retirement institution shows that cognitive interest and social interaction are strong learning motivators [6].

Our article discusses the question from the perspective of mathematics and computer science. Inspired by Hinton and Sejnowski [7], we conduct an experiment with an artificial neural network (ANN). Our test shows why retraining can be harder than training from scratch in a changing environment. The reason is that a network must first unlearn previously learned concepts before it can adapt to new training data. Assuming that AI has similarities with human intelligence, we can draw some interesting conclusions from this insight.

Artificial neural networks

Artificial neural networks resemble the structure and behavior of the nerve cells of our brain, known as neurons. Typically, an ANN consists of input cells that receive signals from the outside world. By processing these signals, the network is able to make a decision in response to the received input. A perceptron is a simple variant of an ANN [8]. It was introduced in 1958 by Rosenblatt [9]. Figure 1 outlines the basic structure of a perceptron. In recent decades, more advanced types of ANNs have been developed. Yet for our experiment, a perceptron is well suited as it is easy to explain and interpret.

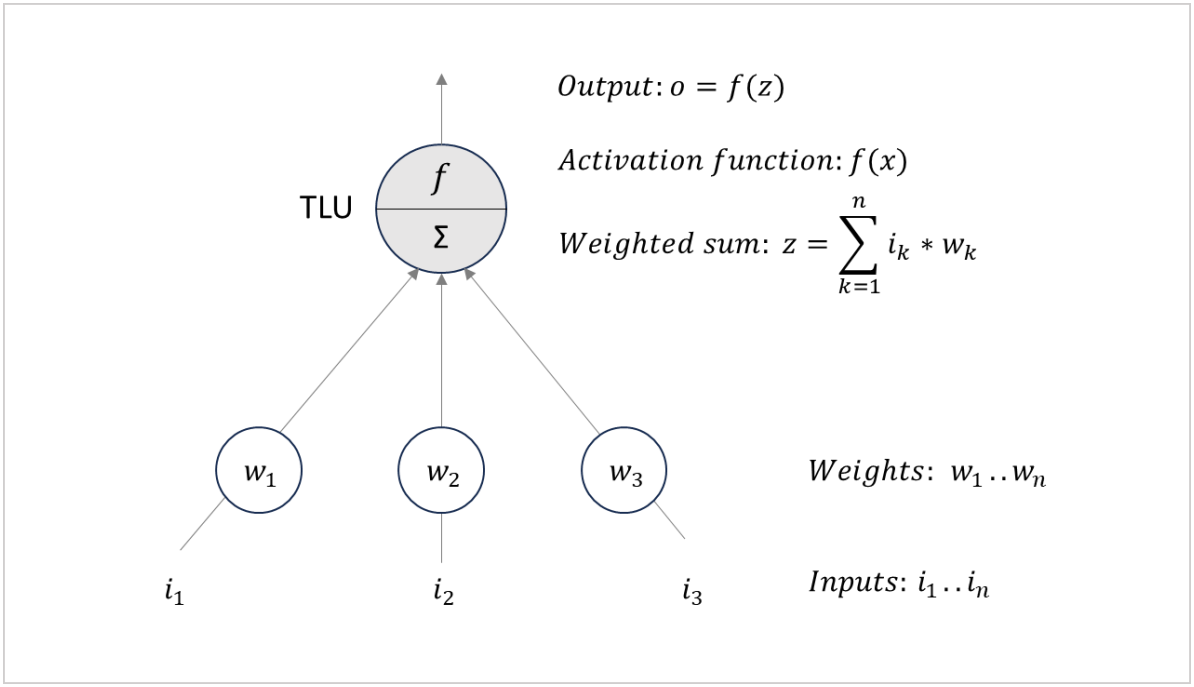

Figure 1: Structure of a single-layer perceptron. Own representation based on [8, p. 284].

Figure 1 shows the architecture of a single-layer perceptron. As input, the network receives *n* numbers ($i_1..i_n$). Together with learned weights ($w_1..w_n$), the inputs are transmitted to a threshold logic unit (TLU). This TLU calculates a weighted sum (z) by multiplying the inputs (i) and the weights (w). In the next step, an activation function (f) determines the output (o) based on the weighted sum (z). Finally, the output (o) allows the network to make a decision as a response to the received input. Rosenblatt has shown that this simple form of ANN can solve a variety of problems.

Perceptrons can use different activation functions to determine their output (o). Common functions are the *binary step function* and the *sign* function, presented in Figure 2. As the name indicates, the binary function generates a binary output $\{0,1\}$ that can be used to make yes/no decisions. For this purpose, the binary function checks whether the weighted sum (z) of a given input is less or equal to zero. If this is the case, the output (o) is zero, otherwise one. In comparison, the sign function distinguishes between three different output values $\{-1,0,+1\}$.

$$binary(z) = \begin{cases} 0 \; if \; z \leq 0 \\ 1 \; if \; z > 0 \end{cases} \qquad sgn(z) = \begin{cases} -1 \; if \; z < 0 \\ 0 \; if \; z = 0 \\ +1 \; if \; z > 0 \end{cases}$$

Figure 2: Examples of activation functions. Own representation based on [8, p. 285].

To train a perceptron based on a given dataset, we need to provide a sample that includes input signals (features) linked to the desired output (target). During the training process, an algorithm repeatedly processes the input to learn the best fitting weights to generate the output. The number of iterations required for training is a measure of the learning effort. For our experiment, we train a perceptron to decide whether a customer will buy a certain mobile phone. The source code is available on GitHub [10]. For the implementation, we used Python v3.10 and scikit-learn v1.2.2.

Learning customer preferences

Our experiment is inspired by a well-known case of (failed) relearning. Let us imagine we work for a mobile phone manufacturer in the year 2000. Our goal is to train a perceptron that learns whether customers will buy a certain phone model. In 2000, touchscreens are still an immature technology. Therefore, clients prefer devices with a keypad instead. Moreover, customers pay attention to the price and opt for low-priced models compared to more expensive phones. Features like these made the Nokia 3310 the world's best-selling mobile phone in 2000 [11].

Figure 3: Nokia 3310, Image by LucaLuca, CC BY-SA 3.0, Wikimedia Commons

For the training of the perceptron, we use the hypothetical dataset shown in Table 1. Each row represents a specific phone model and the columns “keypad,” “touch” and “low_price” its features. For the sake of simplicity, we use binary variables. Whether a customer will buy a device is defined in the column “sale.” As described above, clients will buy phones with keypads and a low price (*keypad=1* and *low_price=1*). In contrast, they will reject high-priced models (*low_price=0*) and phones with touchscreens (*touch=1*).

ID	keypad	touch	low_price	sale
0	1	0	1	1
1	1	0	0	0
2	0	1	0	0
3	0	1	1	0

Table 1: Hypothetical phone sales dataset from 2000

In order to train the perceptron, we feed the above dataset several times. In terms of scikit-learn, we repeatedly call the function `partial_fit` (source code see here). In each iteration, an algorithm tries to gradually adjust the weights of the network to minimize the error in predicting the variable “sale.” Figure 4 illustrates the training process over the first ten iterations.

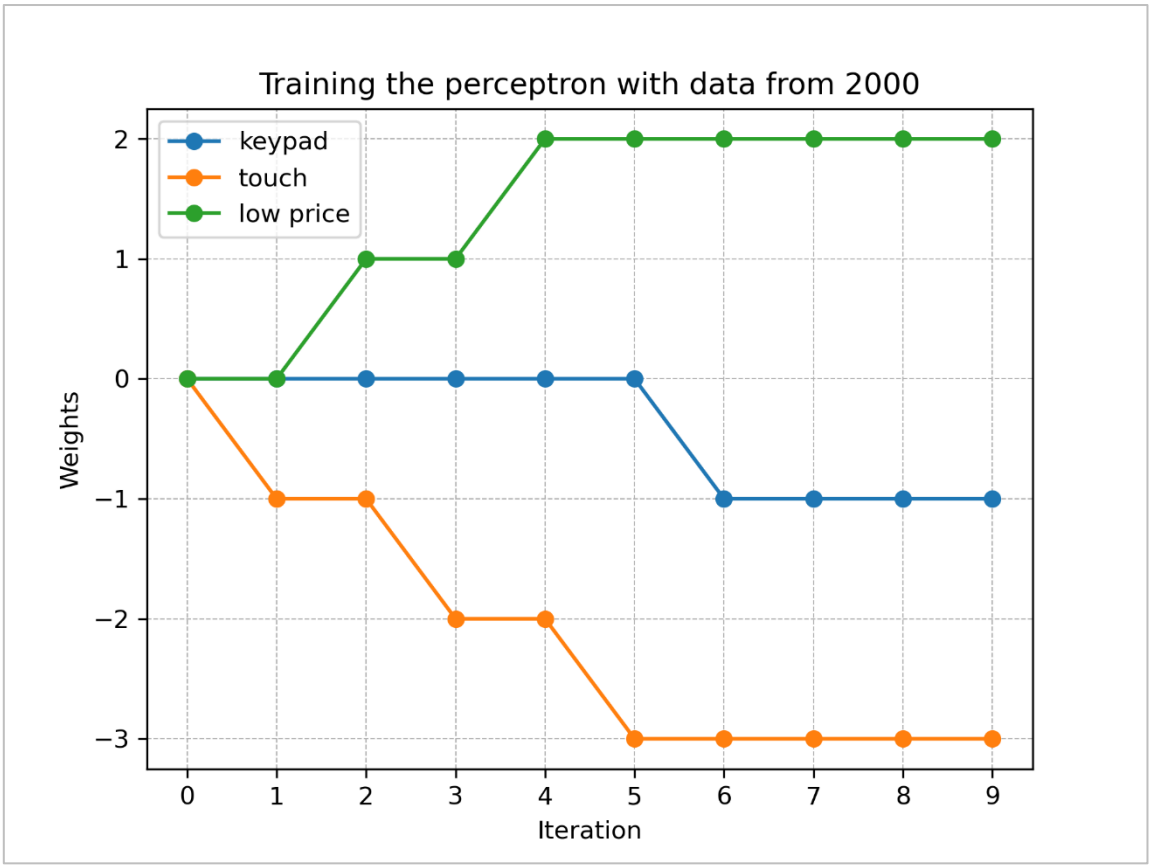

Figure 4: Training the phone sales perceptron with data from 2000

As the above diagram shows, the weights of the perceptron are gradually optimized to fit the dataset. In the sixth iteration, the network learns the best fitting weights, subsequently the numbers remain stable. Figure 5 visualizes the perceptron after the learning process.

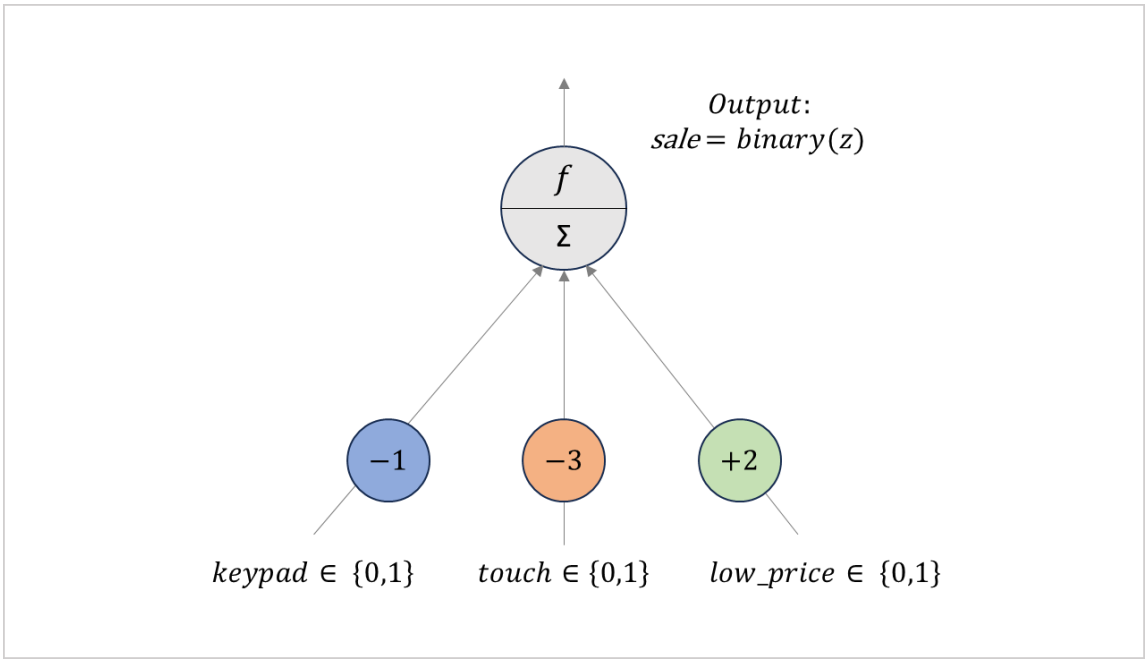

Figure 5: Phone sales perceptron trained with data from 2000

Let us consider some examples based on the trained perceptron. A low-priced phone with a keypad leads to a weighted sum of *z=-1*1–3*0+2*1=1*. Applying the binary step function generates the output *sale=1*. Consequently, the network predicts clients to buy the phone. In contrast, a high-priced device with a keypad leads to the sum *z=-1*1–3*0+2*0=1=-1*. This time, the network predicts customers to reject the device. The same is true, for a phone having a touchscreen. (In our experiment, we ignore the case where a device has neither a keypad nor a touchscreen, as customers have to operate it somehow.)

Retraining with changed preferences

Let us now imagine that customer preferences have changed over time. In 2007, technological progress has made touchscreens much more user-friendly. As a result, clients now prefer touchscreens instead of keypads. Customers are also willing to pay higher prices as mobile phones have become status symbols. These new preferences are reflected in the hypothetical dataset shown in Table 2.

ID	keypad	touch	low_price	sale
0	1	0	1	0
1	1	0	0	0
2	0	1	0	1
3	0	1	1	1

Table 2: Hypothetical phone sales dataset from 2007

According to Table 2, clients will buy a phone with a touchscreen (*touch=1*) and do not pay attention to the price. Instead, they refuse to buy devices with keypads. In reality, Apple entered the mobile phone market in 2007 with its iPhone. Providing a high-quality touchscreen, it challenged established brands. By 2014, the iPhone eventually became the best-selling mobile phone, pushing Nokia out of the market [11].

Figure 6: iPhone 1st generation, Carl Berkeley — CC BY-SA 2.0, Wikimedia Commons

In order to adjust the previously trained perceptron to the new customer preferences, we have to retrain it with the 2007 dataset. Figure 7 illustrates the retraining process over the first ten iterations. As Figure 7 shows, the retraining requires three iterations. Then, the best fitting weights are found and the network has learned the new customer preferences of 2007. Figure 8 illustrates the network after relearning.

Figure 7: Retraining the phones sales perceptron with data from 2007

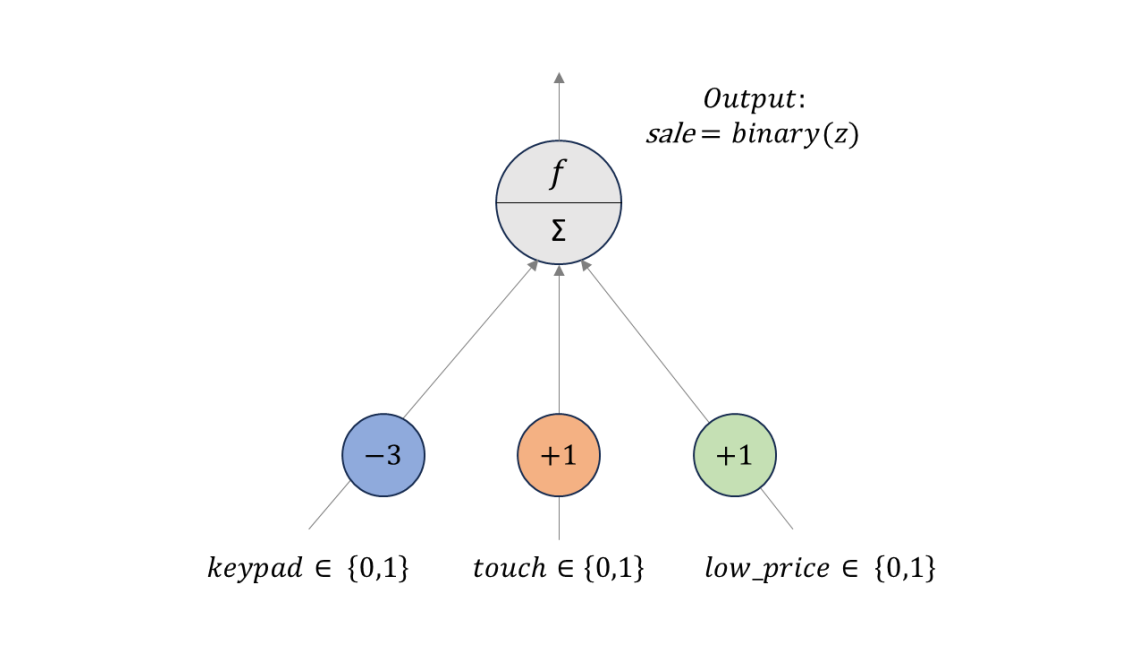

Figure 8: Phone sales perceptron after retraining with data from 2007

Let us consider some examples based on the retrained perceptron. A phone with a touchscreen (*touch=1*) and a low price (*low_price=1*) now leads to the weighted sum *z=-3*0+1*1+1*1=2*. Accordingly, the network predicts customers to buy a phone with these features. The same applies to a device having a touchscreen (*touch=1*) and a high price (*low_price=0*). In contrast, the network now predicts that customers will reject devices with keypads.

From Figure 7, we can see that the retraining with the 2007 data requires three iterations. But what if we train a new perceptron from scratch instead? Figure 9 compares the retraining of the old network with training a completely new perceptron on basis of the 2007 dataset.

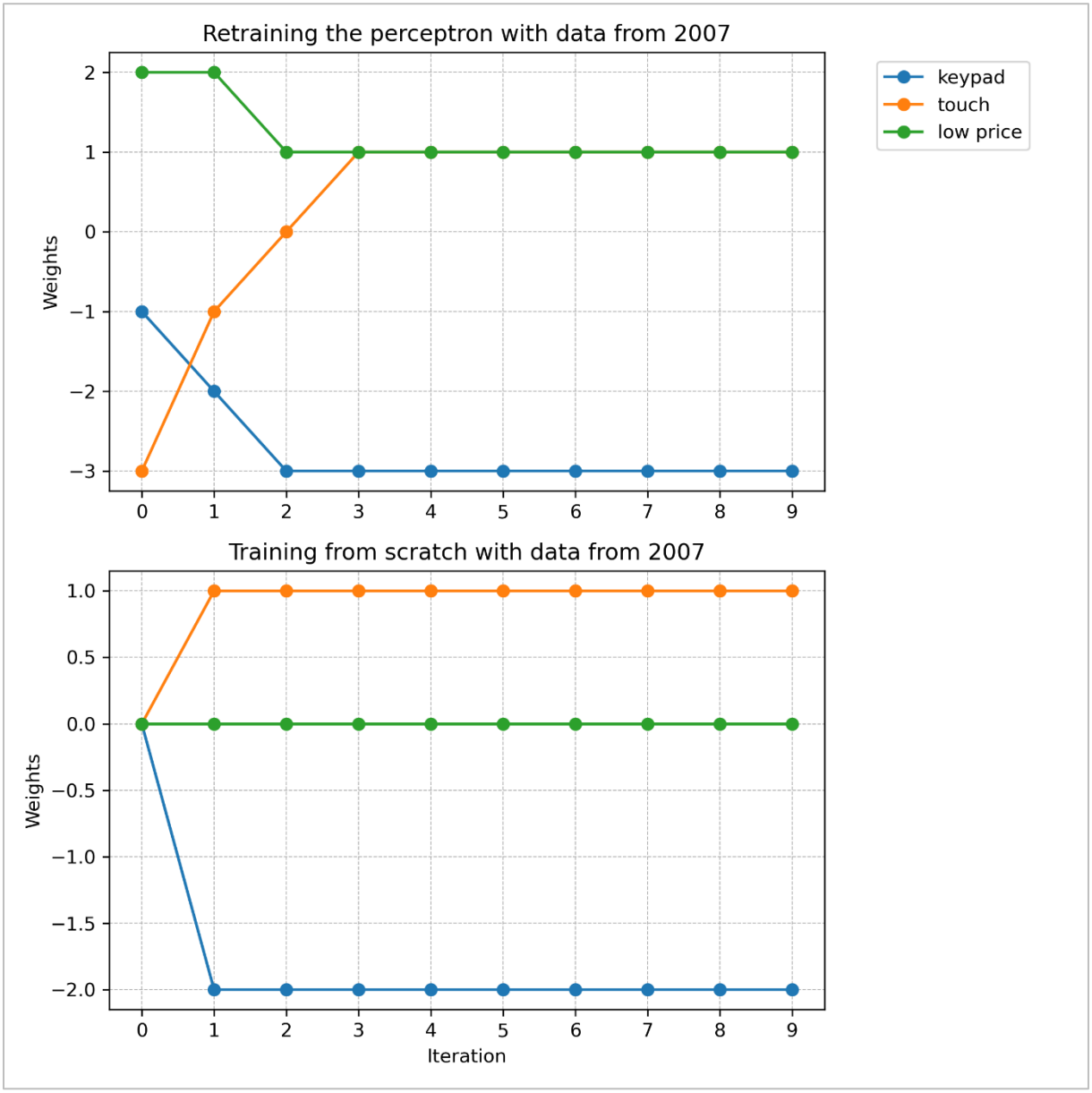

Figure 9: Retraining vs training from scratch with data from 2007

In our example, training a new perceptron from scratch is much more efficient than retraining the old network. According to Figure 9, training requires only one iteration, while retraining takes three times as many steps. Reason for this is that the old perceptron must first unlearn previously learned weights from the year 2000. Only then is it able to adjust to the new training data from 2007. Consider, for example, the weight of the feature “touch.” The old network must

adjust it from -3 to +1. Instead, the new perceptron can start from scratch and increase the weight directly from 0 to +1. As a result, the new network learns faster and arrives at a slightly different setting.

Discussion of results

Our experiment shows from a mathematical perspective why retraining an ANN can be more costly than training a new network from scratch. When data has changed, old weights must be unlearned before new weights can be learned. If we assume that this also applies to the structure of the human brain, we can transfer this insight to some real-world problems.

In his book "The Innovator's Dilemma", Christensen studies why companies that once were innovators in their sector failed to adapt to new technologies [12]. He underpins his research with examples from the hard disk and the excavator market. In several cases, market leaders struggled to adjust to radical changes and were outperformed by market entrants. According to Christensen, new companies entering a market could adapt faster and more successfully to the transformed environment. As primary causes for this he identifies economic factors. Our experiment suggests that there may also be mathematical reasons. From an ANN perspective, market entrants have the advantage of learning from scratch, while established providers must first unlearn their traditional views. Especially in the case of disruptive innovations, this can be a major drawback for incumbent firms.

Radical change is not only a challenge for businesses, but also for society as a whole. In their book "The Second Machine Age," Brynjolfsson and McAfee point out that disruptive technologies can trigger painful social adjustment processes [13]. The authors compare the digital age of our time with the industrial revolution of the 18th and 19th centuries. Back then, radical innovations like the steam engine and electricity led to a deep transformation of society. Movements such as the Luddites tried to resist this evolution by force. Their struggle to adapt may not only be a matter of will, but also of ability. As we have seen above, unlearning and relearning can require a considerable effort compared to learning from scratch.

Conclusion

Clearly, our experiment builds on a simplified model of reality. Biological neural networks are more complicated than perceptrons. The same is true for customer preferences in the mobile phone market. Nokia's rise and fall has many reasons aside from the features included in our dataset. As we have only discussed one specific scenario, another interesting research question is in which cases retraining is actually harder than training. Authors like Hinton and Sejnowski [7] as well as Chen et. al [14] offer a differentiated view of the topic. Hopefully our article provides a starting point to these more technical publications.

Acknowledging the limitations of our work, we can draw some key lessons from it. When people fail to adapt to a changing environment, it is not necessarily due to a lack of intellect or motivation. We should keep this in mind when it comes to the digital transformation. Unlike digital natives, the older generation must first unlearn "analog" concepts. This requires effort and time. Putting too much pressure on them can lead to an attitude of denial, which translates into conspiracy theories and calls for strong leaders to stop progress. Instead, we should develop concepts for successful unlearning and relearning. Teaching technology is at least as important as developing it. Otherwise, we leave the society behind that we aim to support.

All images unless otherwise noted are by the authors.

About the authors

Christian Koch is an Enterprise Lead Architect at BWI GmbH and Lecturer at the Nuremberg Institute of Technology Georg Simon Ohm.

Markus Stadi is a Senior Cloud Data Engineer at Dehn SE working in the field of Data Engineering, Data Science and Data Analytics for many years.

References

1. Grant, A. (2023). *Think again: The power of knowing what you don't know*. Penguin.
2. Reuter-Lorenz, P. A., & Lustig, C. (2005). Brain aging: reorganizing discoveries about the aging mind. *Current opinion in neurobiology*, *15*(2), 245–251.
3. Creasey, H., & Rapoport, S. I. (1985). The aging human brain. *Annals of Neurology: Official Journal of the American Neurological Association and the Child Neurology Society*, *17*(1), 2–10.
4. Welford AT. Motivation, Capacity, Learning and Age. *The International Journal of Aging and Human Development*. 1976;7(3):189–199.
5. Carstensen, L. L., Mikels, J. A., & Mather, M. (2006). Aging and the intersection of cognition, motivation, and emotion. In *Handbook of the psychology of aging* (pp. 343–362). Academic Press.
6. Kim, A., & Merriam, S. B. (2004). Motivations for learning among older adults in a learning in retirement institute. *Educational gerontology*, *30*(6), 441–455.
7. Hinton, G. E., & Sejnowski, T. J. (1986). Learning and relearning in Boltzmann machines. *Parallel distributed processing: Explorations in the microstructure of cognition*, *1*(282–317), 2.
8. Géron, A. (2022). *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O'Reilly Media, Inc.
9. Rosenblatt, F. (1958). The perceptron: a probabilistic model for information storage and organization in the brain. *Psychological review*, *65*(6), 386.
10. Koch, C. (2024). Retrain Python Project. URL: https://github.com/c4ristian/retrain. Accessed 11 January 2024.
11. Wikipedia. List of best-selling mobile phones. URL: https://en.wikipedia.org/wiki/List_of_best-selling_mobile_phones. Accessed 11 January 2024.
12. Christensen, C. M. (2013). *The innovator's dilemma: when new technologies cause great firms to fail*. Harvard Business Review Press.
13. Brynjolfsson, E., & McAfee, A. (2014). *The second machine age: Work, progress, and prosperity in a time of brilliant technologies*. WW Norton & Company.
14. Chen, M., Zhang, Z., Wang, T., Backes, M., Humbert, M., & Zhang, Y. (2022, November). Graph unlearning. In *Proceedings of the 2022 ACM SIGSAC Conference on Computer and Communications Security* (pp. 499–513).