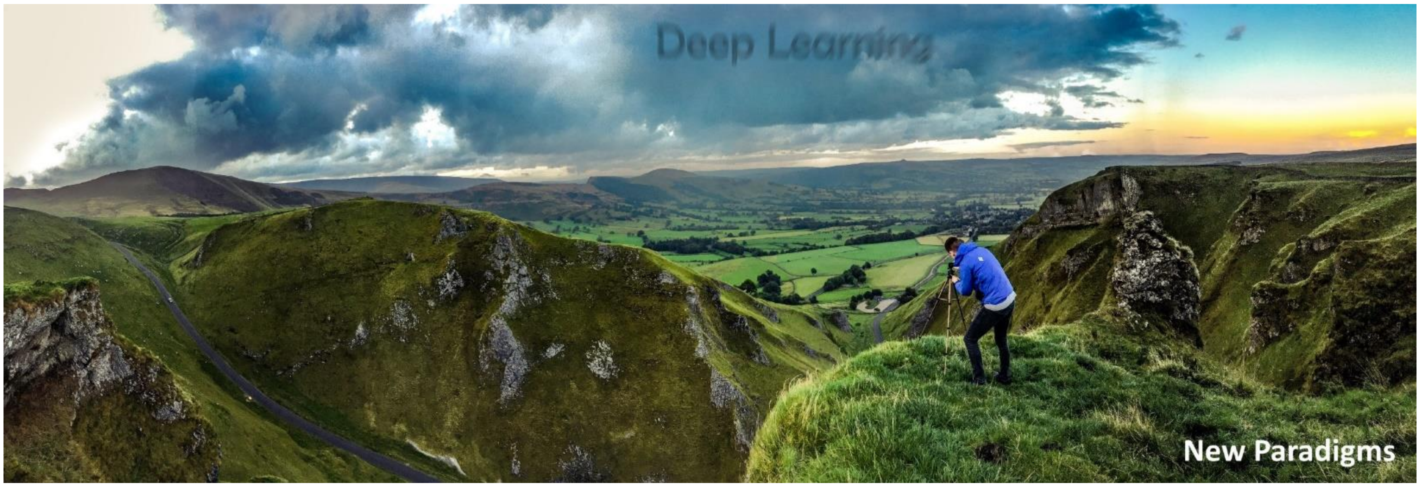
How Managers Should Prepare for Deep Learning: New Paradigms



Exploring the unique challenges for managing analytic systems enabled by Deep Learning *[photo by Charlie Wild on Unsplash]*

By Richard Hackathorn, June 27, 2018
https://towardsdatascience.com/how-managers-should-prepare-for-deep-learning-new-paradigms-28de63054ea6

> This article is the first in a series about the **Managerial Perspectives on Deep Learning**, which are targeted toward managers who are involve with or responsible for analytical systems enabled by **Deep Learning** (DL) using artificial neural network technology. They faced confusing concepts and unique challenges when dealing with these systems. This article focuses on the **New Paradigms** that guide the thinking of these managers about the basic nature of DL. [1]

I have spent my professional career crawling around large information technology (IT) systems as a university professor, software entrepreneur, and industry analyst. Over the last five years, that focus has sharpened on business analytics and especially on deep learning (DL), as driven by neural network technology. I recently completed the Coursera specialization in Deep Learning and reflected upon its implications. [2]

> *Note: I dislike the terms* deep learning *and especially* artificial intelligence *for these articles because of confusing implications and distracting baggage. The proper technical term is* artificial neural networks *(ANN). So, imagine DL = ANN below.*

DL has unique differences with conventional machine learning, especially in the context of enterprise systems. Further, DL has the potential to enhance those systems in ways that are unimaginable today, both for good and not-so good.

IT-DL Disconnect

A major obstacle to the proper adoption of DL within enterprise systems is the conceptual disconnect between IT professionals and DL practitioners. The two groups think differently about the fundamentals—problem statement, nature of data, delivery of results, performance accuracy, data governance, and value to the business.

In the middle of this *IT-DL disconnect* are the managers (and executives) who are involved with analytical systems enabled by DL within large IT ecosystems. These managers are savvy about the organization's priorities and IT resources, often interacting cross-functional with IT and data science groups. They are also

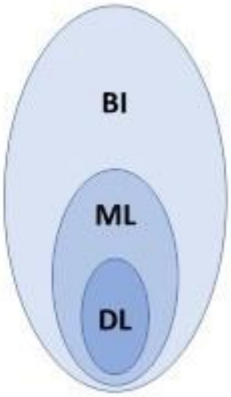
© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)

aware of the remarkable accomplishments of DL, courtesy of Google. They are amazed by emerging applications for facial recognition, autonomous driving, textual analysis, and conversational chat-bots. Further, they are concerned by the mystical hype filling the media about the wonderful benefits and tragic implications of artificial intelligence (AI) as delivered via DL.

The audience for these articles are these managers. They will be the ones responsible for guiding analytic applications using DL. They will take credit when these apps are successful and held accountable when #FAIL. They have a difficult role dealing with DL issues, which are evolving rapidly. Also, they must deal with *the IT-DL disconnect* and need to bridge this gap.

The Question

The question for this article series is…



What is new and different about DL that is relevant to managers responsible for DL-enabled systems?

Through this series, this question will take several forms. This article will discuss the **New Paradigms** that managers need to understand the unique nature of DL, as compare with business intelligence (BI) in general and conventional machine learning (ML) in particular. This understand is essential to enable the fluid conversations with DL practitioners and collaboration with IT groups so that DL projects are successful.

The following sections examines these new paradigms, contrasting them with the older (but still relevant) IT paradigms. For each, the take-away lessons from this paradigm shift are noted.

TL;DR—Choose a few that look interesting. Read others later.

Generalizing Beyond Known Data

The old IT paradigm for analytics is to generate insights that will allow managers at various levels to make better decisions based on data. This paradigm emerged in the 1970's with *decision support systems* by Peter Keen and others. Since then, a key role of IT has been to provide organized data in the form of reports and dashboards, initially on heavy stacks of paper to CRT screens to mobile devices. This trend has caused a boom in the visual analytics area, with flexible dashboard products like Tableau and Qlik. Visually describing known data to managers is (and always will be) essential.

The new DL paradigm for analytics is to support methods for *generalizing beyond known data*. The competitive differentiator for organizations will be to infer (smartly guess) the dynamics of complex business systems. This implies predicting future events, as in predictive analytics. More importantly, this implies the ability to understand why past events unfolded in unique and surprising ways within these complex systems. The models generated by DL have the potential to distill and capture these complexities beyond what is humanly possible. This is both comforting and disconcerting.

Take-Away: DL is all about… Taking specific known facts about your organization and creating generalized models of its behavior. Be aware of business situations in which a decision requires generalizing beyond the data available …which is almost every decision! Was the decision made subjectively (intuitively?) based on a person's long experience? Or, was it based on specific data? If so, what was the data, and how was this data analyzed? Questioning the basis for decisions becomes more critical when managing DL-driven systems.

Cultivating a Farm

The old IT paradigm for software projects is like building a house. First, design it; gather the resources; build it; move on to the next house, along with some maintenance of previous houses. Over the decades, software development methodologies have improved greatly. Agile methods currently permit more flexibility in design and implementation. However, remaining is the notion of completing the project so that the resources can be applied to new projects.

The new DL paradigm is more like a farmer cultivating a field. It is an ongoing process. A farmer does not seed a field and then leave it. There is continual effort through the harvest and into planning for the coming season. A DL project is not a *project* in the normal sense! It is a continuing process of maturing the DL model to better support the use case.

"A farmer in a tractor, clearing wheat during sunset in Lincoln"
by Noah Buscher on Unsplash

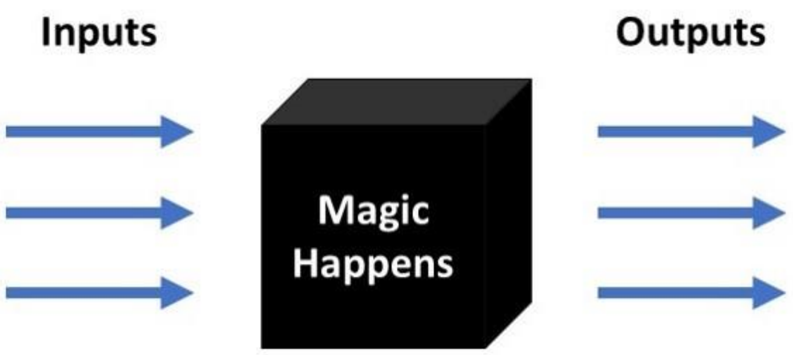
Take-Away: Invest slowly and carefully in DL 'projects' for the long-term. Evaluate each step whether the business benefits exceed the business costs and risks. Do not expect big or quick returns from "low-lying fruit". Expect only moderate and consistent gains. On the other hand, do not hesitate to kill DL projects early and often. Avoid thoughts that a 'project' is finished and can continue operation unattended.

Coping with the Blackbox

The IT paradigm is that application systems should be (and can be) clearly specified and thus easily maintained and governed. The fact is that the new DL paradigm is one wicked blackbox!

DL models consist of a set of matrices, propagated through layers of a neural network to yield accurate predictions. Within the thousands of numbers of those matrices, the results are humanly impossible to interpret and explain. For example, answering the questions about why a specific loan application was rejected by the DL model. This is a **huge** issue of interpretability with DL results, which must be resolved for an operational DL systems.

© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)



The DL Blackbox where the Magic Happens

However, there are two sides to this coin. Managers should not get sucked into the gory details, becoming the deer-in-the-headlight person in the room. Cope with the blackbox as a manager! The figure below illustrates the situation. The magic is the responsibility of the data science team.

Take-Away: You are in a technical discussion with techies about new DL-enabled system, slowly descending into the darkness. Go to whiteboard and draw a simple box. Ask about outputs; then about inputs. Question whether the outputs promote specific business goals. Question whether we able to curate data properly as the input for training examples. Question how the results affecting customers will be interpreted from the model and explained. Don't leave the room until everyone (especially you) understands and agrees! Manage to the blackbox.

Teaching by Example

The IT paradigm is to program the predefined procedure into hard code. First, one must specify the procedure, then code it, and finally debug it.

The DL paradigm has fundamental differences of DL with typical IT software projects of the past. First difference is that you program by training the DL model with examples that are labeled with the expected result for each example. This is called *supervised learning*.

I dislike the terms 'learning' or even 'training' in this context. The first implies that it is the responsibility of the model to learn. The second implies that the model should obey a rigid protocol to perform the task. Both imply that the responsibility lies with the DL model. In fact, the real responsibility lies with the manager, along with the data science team, to **teach** the DL model by clear **examples** labeled as 'good' or 'bad'.

The analogy for the manager is like a first-grade teacher instructing the class about the zoo, for which no one in your class has seen. Your responsibility is like prepare them for their zoo trip next week. So, you prepare a set of examples of the animals that they will see, describing them in detail.

Technically, this process in DL is referred to as 'training the model' but it is more like teaching or even mentoring the model. Like first graders, the model knows very little at the beginning. Unlike first graders, the model will only learn via your examples. Expect bad results from the first graders and from your model if your examples are poorly chosen or poorly labeled. Is there a sufficient number? Are they diverse enough to cover future situations? And so on.

Take-Away: Take responsibility for curating the data as labeled examples to *teach* the DL model. This will be a major part of your discussion about inputs for the blackbox. You will find that this responsibility will be heavily

© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)

dependent on your existing IT ecosystem to storage and integrate data across the data warehouse and possible data lakes. Be thankful for this ecosystem legacy. If no such ecosystem, find another job.

Preserve the Information

The IT paradigm for data curation is to clean and organize data for human consumption, pleasant to the human eye. In contrast, the DL paradigm is to preserve information within the data (a la Shannon information entropy) for machine consumption, raw at its finest granularity. DL models love ugly bits!

The analogy is curation as in an art museum. The curator preserves the art works, while enhancing its essence with context and history. Seems that the data lake folks can get back on track by internalizing this analogy. Plus, data warehouses play a critical role by providing that context (i.e., supplemental features to the example data).

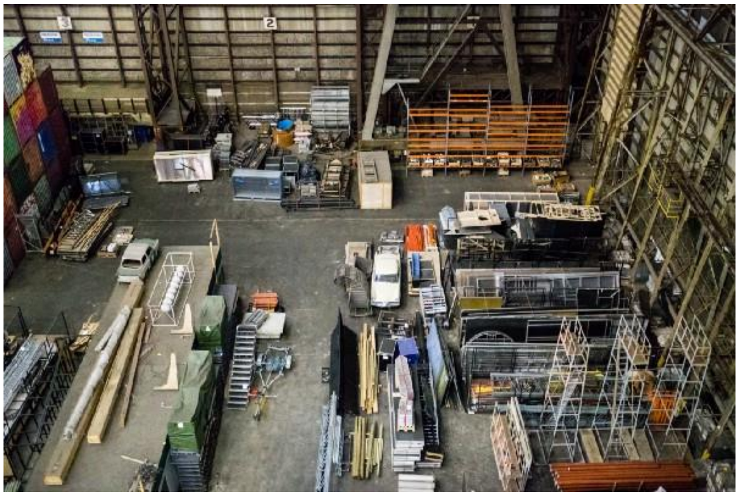
Many DL projects start as apps that are isolated from the data warehouse, only to realize that doing anything useful for the organization must be carefully linked with the DW. Save yourself the extra effort, start with the data warehouse and build outward. Leveraging as repository of official structured data, containing the context and relationships, data warehouses reflect the information that is perceived to be important to the operation of the organization.

Raw data (messy for a human) is good, if it accurately conveys accurate information about the organization and its environment. Hence, data lake efforts are good if designed properly to assess validity and correct accuracy/validity.

Take-Away: Power of DL begins and ends with the data. So, do not trash the bits to make them look pretty! Ugly is the true face of reality.

Treating Data as Reality Photos

The IT paradigm visualizes data as a tabular spreadsheet of rows and columns. This is the paradigm shift that affected me personally as I was learning about DL. I would frequency fall back on my old tabular thinking, as if this was the ground-truth. However, DL became much easier when I let loose of this crutch and started thinking matrices, and then tensors (higher dimensional matrices), as the ground-truth. In a subtle way, thinking tabular (rows-columns, instances-features) **overly** simplifies reality. Our eyes and ears do not capture minute-to-minute reality in a spreadsheet!



The DL paradigm shifts to images and other *unstructured* data formats in two ways. First, DL image processing for object detection and identification have been highly successful, implying

© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)

that the embedded spatial information in a photo is leveraged. Second, a new DL trick is to treat tabular data as pixels in an actual (but messy) image and then utilize the same effective DL techniques. The early results using this trick have been amazing. [3]

Take-Away: Stop thinking about data as tabular! Instead think about data as photos of business reality. Further, think of business behaviors as videos of that reality. The resulting DL models may perform better.

Exceeding Human-Level Performance

The IT paradigm is to computerize organizational tasks by programming the procedure for these tasks into hard code. The motivation is to perform the tasks cheaper and faster as compared to the equivalent human performance. For example, automated manufacturing assembly line (with a few humans at critical points) have replaced the human equivalent because of this cheaper-faster objective.

The new paradigm for DL is teach DL models to perform these same tasks with the expectation of exceeding *human-level performance* (HLP). Why?

We humans have created tools smart enough to out-smart ourselves!

This statement seems vindicated by news of computers beating world champions in the games of chess and go, along with Jeopardy by IBM Watson in 2011. However, many dismiss these achievements since these examples are merely games, not real use cases in chaotic business situations.

Maybe not so… Andrew Ng has often stated that DL technology can out-performed humans at any intellectual task that takes a human a few seconds of thought to perform. And, he has started a venture fund to proof his point. [4]

Every month, DL evolves to increase that 5 seconds to more thoughtful tasks. This is especially apparent in image recognition tasks, from manufacturing defects on semiconductors to CT scans for cancer. The scope of intellectual tasks is gradually increasing, invading areas previously reserved for white-collar college-educated workers.

For managers, the significance of this DL accomplishment is that it enables humans to create algorithms that exceed HLP in challenging intellectual tasks. Further, it exposes the nature about thinking and learning of these tasks so that consistent incremental improvements beyond HLP can be achieved.

This is NOT the usual sci-fi story where super-intelligent robots decide to eliminate humans because of our inferiority. The real problem is the unintended consequences of embedded DL applications in larger systems, like Facebook, Amazon, Google, and other companies who impact many lives.

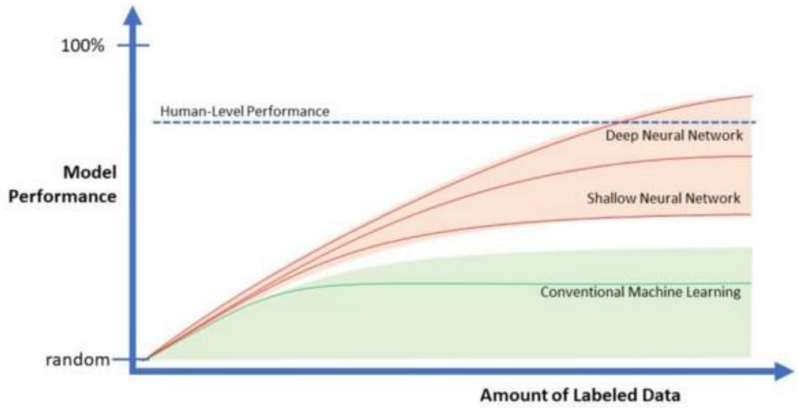
Take-Away: Think boldly (and responsibly) about applying DL to significant use cases. It is now possible to DL-enable tasks that were unimaginable, given age-old constraints on HLP. DL bestows upon us a great power, along with a great responsibility, to be used wisely and for the good of humanity. DL has been compared to other major inventions of mankind, like nuclear energy. If valid, this comparison is a sobering thought.

© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)

Data Drives Performance

The IT paradigm is that, with more data, one is able to generate more relationships within the data and more cross-functional comparisons, thus more useful insights can emerge. This hypothesis has driven the evolution of data warehousing and business intelligence over last decades.

The new DL paradigm is similar and synergistic. With more data, DL models are able to perform better in terms of their accuracy of generalizations (i.e., predictions). Below is a simple, yet profound, slide of Andrew Ng in his Coursera DL Specialization is titled, “Scale Drives DL Progress”.



For any predictive algorithm, the accuracy of the model prediction is influenced by the amount of curated labeled example data. Toward the left side, these amounts may be a few thousand examples, less than a gigabyte. Toward the right side, these amounts may be millions or billions of examples, exceeding hundreds of terabytes. The lower left signifies making a prediction based on no data and no prior experience, like flipping a coin.

Here are several points to note…

- At low amounts, any predictive algorithm is as good as any other. DL has no advantage; hence, use conventional machine learning algorithms (like random forest) since they are easier, quicker, and offer better interpretability.
- At higher amounts, DL models with more data always win over those with less data. However, the DL models must become more complex in their structure (i.e., more hidden layers with more weights per layer) to adsorb additional information within the data. And, this structural complexity implies increased effort to train and interpret the model.
- The upper right is the sweet-spot where the machine algorithms is able to exceed HLP. Realize that not all use cases requires HLP so that less data and simpler algorithms may be sufficient.

Take-Away: Simply put, whoever owns the data wins. The company that controls the most data eventually wins over all other companies! Again, the power of DL begins and ends with the data, regardless of how fancy and complex is the model structure. Labelled training datasets are the ‘Achilles heel’ of DL.

Trained Models as Packaged Apps

The usual IT paradigm for delivering software tools that solve specific use cases is via packaged applications, usually by proprietary software vendors that specialized in that use case. For example, Salesforce.com markets a customer relationship management package that tracks the complete sales activity, from initial contract through purchase, for large companies.

The new DL paradigm is Software 2.0… Andrej Karpathy, Director of AI for Testa, nailed this concept in his recent article and then tweeted to all of this programmer friends, “Gradient descent can write code better than you. I’m sorry.” [5] Ouch!

Can you visual it… Millions of programmers losing their jobs to DL, descending on Washington to demand legislation to limit DL utilization and ban GPU-enabled equipment. And, the same may happen for packaged software vendors!

For managers of DL systems, the implication is that the DL models that you have been carefully maturing over past years… that's your custom packaged app! The next-gen of packaged app vendors will be selling pre-trained DL models on extensive example datasets. Via transfer learning, these pre-trained models could enable your DL projects to save months of effort to attain required performance levels.

Take-Away: Treat your trained DL models with great respect. These are your corporate IT jewels, like a mature data warehouses have been. Be savvy about transfer learning, both within your industry and across other industries with similar informational objects.

Augment, Not Automate!

The IT paradigm is to computerize to reduce routine tasks and increase consistency in performing task.

The DL paradigm is more of an ethical assertion containing a good dose of common sense… **Use DL to augment human tasks, not to automate those tasks.**

The challenge here is to creatively reinsert the human back into the workflow. The term *Human-In-The-Loop* (HITL) has been used for decades in the interface design of *fast* systems (where decision timing requires sub-second responses), like piloting fighter jets. Typical modes for HITL are: monitoring for unusual situations, approving cases of high uncertainty, investigating situations where predictions were false, vetoing cases that do not feel right, and the like.

Francois Chollet, a noted DL researcher, argues that DL tools should not be used to manipulate people. Instead, DL should give people the control over those tools to pursue their own goals and passions. [6]

The DL augmentation challenge is not time constrained; it is thought (or informational) constrained. Within the DL models, knowledge is contained within millions of numbers, which makes total sense for tensor-smashing but little sense to a human brain. We need a new generation of *immersive analytics* (like the next-gen of visual analytics) to create virtual spaces representing the full complexity of those numbers, using all human senses. I think that humans are up to this task and will put them on equal footing with their computationally superior artificial colleagues. [7]

Take-Away: When designing a DL system, where are your HITL critical points? How will augment, enlarge, and humanize the task for persons? How will human judgment be applied in murky situations? How will those persons be properly equipped to perform those HITL tasks?

Own The Responsibility!

In conclusion, DL has the potential within enterprise systems for much good and much harm. At all levels, managers who are responsible for DL-enabled systems must own the responsibility for monitoring and balancing benefits and costs for the organization and for society.

This article describes ten paradigm shifts that these manager should understand and convey to their colleagues, lessening the IT-DL disconnect problems and enabling successful utilization of DL.

Appreciation… to the DL study group at Data Detectives of Boulder for our discussions over the past months, which helped clarify these thoughts.

Finally… If you benefit from these articles, please support my Patreon to create and mentor small peer groups of managers to explore key management issues of analytical systems enabled by Deep Learning. If this program might be of interest to a colleague, please share a link or tweet. Thanks, Richard

Notes

[1] Overview of the article series. Surprised about how this question has exploded into dozens of topics. This should keep me busy for awhile!
https://medium.com/@Hackathorn/series-on-how-managers-should-prepare-for-deep-learning-f5b795b36148

[2] My reflections on DL after completing the Coursera Specialization in Deep Learning. First part comments on logistics of the course, while the second part delves deeper into the issues arising from creating tools that can out-smart ourselves. Conclusion: *With great power comes great responsibility*.
https://towardsdatascience.com/deep-issues-lurking-within-deep-learning-f923a96564c7

[3] Using pixels to represent tabular data. An emerging DL trick that has deeper implications for *Teaching by Examples*. Related is the article by Rutger Ruizendaal that highlights *entity embedding* for categorical features, building upon de Brébisson et al. paper in 2015. [Updated 7/4/2018. Still have not found that reference for the tabular-to-image trick! Help?]
https://towardsdatascience.com/deep-learning-structured-data-8d6a278f3088

[4] Five-year old article that still summarizes the impending impacts of Deep Learning, along the background of Andrew Ng. Often cited.
https://www.wired.com/2013/05/neuro-artificial-intelligence/

[5] Article on *Software 2.0* by Karpathy. Excellent description of how DL software differs from our conventional thoughts about software over past decades. His cute tweet summarizes his points!
https://medium.com/@karpathy/software-2-0-a64152b37c35

[6] Personal views by François Chollet on his social and ethical concerns about AI. Clear and thoughtful. Reinforces the *Augment, Not Automate* point.
https://medium.com/@francois.chollet/what-worries-me-about-ai-ed9df072b704

[7] Immersive Analytics (IA), a long-term passion on mine! Don't get me started! However, I now see a tight coupling between DL and IA via immersive virtual worlds (massive and collaborative) driven by clusters of DL models, all focused on a specific problem domain. Interested? More details at…
https://www.immersiveanalytics.com/

© Bolder Technology, Inc., 2019 – Attribution 4.0 International (CC BY 4.0)