Implementing Prediction Model using Artificial Intelligence and Deep Learning

[1] Sathe Amol Vaijinath, Assistant Professor Department of Mechanical Engineering, MMIT Pune
[2] Aditi Jadhav, Assistant Professor Department of IT MGM University, Aurangabad.

Abstract

Profound learning models represent another learning worldview in man-made consciousness (man-made intelligence) and AI. Ongoing advancement brings about picture examination and discourse acknowledgment have created a gigantic interest in this field on the grounds that likewise applications in numerous different spaces giving huge information appear to be conceivable. On a disadvantage, the numerical and computational procedure hidden profound learning models is exceptionally difficult, particularly for interdisciplinary researchers. Consequently, we present in this paper a starting audit of profound learning approaches including Profound Feedforward Brain Organizations (D-FFNN), Convolutional Brain Organizations (CNNs), Profound Conviction Organizations (DBNs), Autoencoders (AEs), and Long Transient Memory (LSTM) organizations. These models structure the significant center designs of profound learning models right now utilized and ought to have a place in any information researcher's tool kit. Significantly, those center structural structure blocks can be formed deftly — in a nearly Lego-like way — to fabricate new application-explicit organization models. Thus, an essential comprehension of these organization models is vital to be ready for future improvements in computer based intelligence.

Keywords: deep learning, artificial intelligence, machine learning, neural networks, prediction models, data science

1. INTRODUCTION

We are residing in the large information period where all areas of science and industry create gigantic measures of information. This goes up against us with phenomenal difficulties in regards to their examination and understanding. Thus, there is a critical requirement for novel AI and computerized reasoning techniques that can help in using these information. Profound learning (DL) is a particularly original philosophy right now getting a lot of consideration (Hinton et al., 2006). DL portrays a group of learning calculations as opposed to a solitary strategy that can be utilized to learn complex expectation models, e.g., multi-facet brain networks with many secret units (LeCun et al., 2015). Critically, profound learning has been effectively applied to a few application issues. For example, a profound learning strategy set the standard for the order of manually written digits of the MNIST informational collection with a blunder pace of 0.21% (Wan et al., 2013).

Further application regions incorporate picture acknowledgment (Krizhevsky et al., 2012a; LeCun et al., 2015), discourse acknowledgment (Graves et al., 2013), normal language getting it (Sarikaya et al., 2014), acoustic displaying (Mohamed et al., 2011) and computational science (Leung et al., 2014; Alipanahi et al., 2015; Zhang S. et al., 2015; Smolander et al., 2019a,b). Models of fake brain networks have been utilized since about the 1950s (Rosenblatt, 1957); in any case, the ongoing rush of profound learning brain networks began around 2006 (Hinton et al., 2006). A typical quality of the numerous varieties of managed and unaided profound learning models is that these models have many layers of stowed away neurons learned, e.g., by a Confined Boltzmann Machine (RBM) in mix with Backpropagation and mistake slopes of the Stochastic Inclination Plunge (Riedmiller and Braun, 1993). Because of the heterogeneity of profound learning approaches a complete conversation is extremely difficult, and hence, past surveys focused on committed sub-points. For example, a 10,000 foot perspective without itemized clarifications can be found in LeCun et al. (2015), a notable synopsis with many itemized references in Schmidhuber(2015) and surveys about application spaces, e.g., picture examination (Rawat and Wang, 2017; Shen et al., 2017), discourse acknowledgment (Yu and Li, 2017), normal language handling (Youthful et al., 2018), and biomedicine (Cao et al., 2018).

Interestingly, our audit focuses on a middle level, giving likewise specialized subtleties normally precluded. Given the interdisciplinary interest in profound realizing, which is important for information science (Emmert-Streib and Dehmer, 2019a), this makes it more straightforward for individuals new to the field to begin. The subjects we chose are centered around the center procedure of profound learning approaches including Profound Feedforward Brain Organizations (DFFNN), Convolutional Brain Organizations (CNNs), Profound Conviction Organizations (DBNs), Autoencoders (AEs), and Long Transient Memory (LSTM) organizations. Further organization designs which we examine help in understanding these center methodologies.

The historical backdrop of brain networks is long, and many individuals have contributed toward their improvement throughout the long term. Given the new blast of interest in profound learning, it isn't is actually to be expected that the task of credit for key advancements isn't uncontroversial. In the accompanying, we were focusing on a fair show featuring just the most recognized commitments. In 1943, the principal numerical model of a neuron was made by McCulloch and Pitts (1943). This model pointed toward giving a theoretical plan to the working of a neuron without impersonating the biophysical component of a genuine TABLE 1 | An outline of regularly involved enactment capabilities for neuron models. Initiation capability ϕ(x) ϕ ′ (x) Values Exaggerated digression tanh(x) = e x−e −x e x+e−x 1 − ϕ(x) 2 (−1, 1) Sigmoid S(x) = 1 1+e−x ϕ(x)(1 − ϕ(x)) (0, 1) ReLu R(x) = (0 for x < 0 x for x ≥ 0 (0 for x < 0 1 for x ≥ 0 [0,∞) Heaviside capability H(x) = (0 for x < 0 1 for x ≥ 0 δ(x) [0, 1] Signum capability sgn(x) = ? ??? ??? −1 for x < 0 0 for x = 0 1 for x > 0 2δ(x) [−1, 1] Softmax yi = e xi Pn j e xj ∂yi ∂j = yi δij − yj (0, 1) organic neuron. It is intriguing to take note of that this model didn't think about learning. In 1949, the first thought regarding naturally spurred learning in quite a while was presented by Hebb (1949). Hebbian learning is a type of unaided learning of brain organizations.

In 1957, the Perceptron was presented by Rosenblatt (1957). The Perceptron is a solitary layer brain network filling in as a straight double classifier. In the cutting edge language of ANNs, a Perceptron involves the Heaviside capability as an enactment capability.

In 1960, the Delta Learning rule for learning a Perceptron was presented by Widrow and Hoff (1960). The Delta Learning rule, otherwise called Widrow and Hoff Learning rule or the Most un-Mean Square rule, is an inclination plummet learning rule for refreshing the loads of the neurons. It is a unique instance of the backpropagation calculation.

In 1968, a strategy called Gathering Technique for Information Taking care of (GMDH) for preparing brain networks was presented by Ivakhnenko (1968). These organizations are broadly viewed as the principal profound learning organizations of the Feedforward Multi-facet Perceptron type. For example, the paper (Ivakhnenko, 1971) utilized a profound GMDH network with 8 layers. Strangely, the quantities of layers and units per layer could be gained and were not fixed all along.

In 1969, a significant paper by Minsky and Papert (1969) was distributed which demonstrated the way that the XOR issue can't be advanced by a Perceptron in light of the fact that it isn't straightly distinguishable. This set off a respite stage for brain networks called the "Computer based intelligence winter."

In 1974, blunder backpropagation (BP) has been recommended to use in brain organizations (Werbos, 1974) for learning the weighted in a managed way and applied in Werbos (1981). Be that as it may, the actual strategy is more seasoned (see e.g., Linnainmaa, 1976).

In 1980, a various leveled complex brain network for visual example acknowledgment called Neocognitron was presented by Fukushima (1980). After the profound GMDH organizations (see over), the Neocognitron is viewed as the second counterfeit NN that merited the characteristic profound. It presented convolutional NNs (today called CNNs). The Neocognitron is basically the same as the engineering of current, managed, profound Feedforward Brain Organizations (D-FFNN) (Fukushima, 2013).

In 1982, Hopfield presented a substance addressable memory brain organization, these days called Hopfield Organization (Hopfield, 1982). Hopfield Organizations are a model for intermittent brain organizations.

In 1986, backpropagation returned in a paper by Rumelhart et al. (1986). They showed tentatively that this learning calculation can produce valuable interior portrayals and, thus, be useful for general brain network learning undertakings.

In 1987, Terry Sejnowski presented the NETtalk calculation (Sejnowski and Rosenberg, 1987). The program figured out how to articulate English words and had the option to work on over the long haul.

In 1989, a Convolutional Brain Organization was prepared with the backpropagation calculation to learn written by hand digits (LeCun et al., 1989). A comparative framework was subsequently used to peruse transcribed checks and postal divisions, handling traded looks at the US in the last part of the 90s and mid 2000s. Note: During the 1980s, the second rush of brain network research arose by and large by means of a development called connectionism (Fodor and Pylyshyn, 1988). This wave went on until the mid 1990s.

In 1991, Hochreiter concentrated on a key issue of any profound learning organization, which connects with the issue of not being teachable with the back engendering calculation (Hochreiter, 1991). His review uncovered that the sign spread by back engendering either diminishes or increments without limits. In the event of a rot, this is corresponding to the profundity of the organization. This is currently known as the evaporating or detonating slope issue.

In 1992, a first halfway solution for this issue has been proposed by Schmidhuber (1992). The thought was to pre-train a RNN in an unaided manner to speed up resulting directed learning. The concentrated on network had in excess of 1,000 layers in the repetitive brain organization.

In 1995, oscillatory brain networks have been presented in Wang and Terman (1995). They have been utilized in different applications like picture and discourse division and creating complex time series (Wang and Terman, 1997; Hoppensteadt and Izhikevich, 1999; Wang and Brown, 1999; Soman et al., 2018).

In 1997, the principal managed model for learning RNN was presented by Hochreiter and Schmidhuber (1997), which was called Long Transient Memory (LSTM). A LSTM forestalls the rotting mistake signal issue between layers by making the LSTM organizations "recollect" data for a more drawn out timeframe.

In 1998, the Stochastic Slope Drop calculation (gradientbased learning) was joined with the backpropagation calculation for further developing learning in CNN (LeCun et al., 1989). Thus, LeNet-5, a 7-level convolutional network, was presented for grouping transcribed numbers on checks.

In 2006, is broadly viewed as a leading edge year on the grounds that in Hinton et al. (2006) it was shown that brain networks called Profound Conviction Organizations can be proficiently prepared by utilizing a methodology called eager layer-wise pre-preparing. This started the third flood of brain networks that utilized the term profound learning famous.

In 2012, Alex Krizhevsky won the ImageNet Huge Scope Visual Acknowledgment Challenge by utilizing AlexNet, a Convolutional Brain Organization using a GPU and enhanced LeNet5 (see above) (LeCun et al., 1989). This achievement began a convolutional brain network renaissance in the profound learning local area (see Neocognitron).

In 2014, generative antagonistic organizations were presented in Goodfellow et al. (2014). The thought is that two brain networks contend with one another in a game-like way. Generally speaking, this lays out a generative model that can create new information. This has been classified "the coolest thought in AI over the most recent 20 years" by Yann LeCun.

In 2019, Yoshua Bengio, Geoffrey Hinton, and Yann LeCun were granted the Turing Grant for calculated and designing leap forwards that have made profound brain networks a basic part of figuring.

Architectures Of Neural Networks

Artificial Neural Networks (ANNs) are are numerical models that have been spurred by the working of the cerebrum. In any case, the models we examine in the accompanying don't target giving organically reasonable models. All things being equal, the reason for these models is to dissect information.

Model of an Artificial Neuron

A model of a neuron is the fundamental component of any neural network. A neuron model's fundamental tenet is that an input, x, together with a bias, b, are weighted by, w, and then combined.

Feed forward Neural Networks

The neurons must be linked to one another in order to create neural networks (NNs). The most basic shallow and deep architecture for a NN is a feed forward structure. In general, a network's depth refers to the amount of nonlinear transformations that occur between its separating layers, whereas a hidden layer's breadth refers to the dimensionality of its hidden neurons.

Recurrent Neural Networks

Two subclasses of the Recurrent Neural Network (RNN) model family may be distinguished based on how they handle signals. IERNs (infinite impulse recurrent networks) are used in the first and second, respectively (IIRNs). The difference is that a FRN is given by a directed acyclic graph (DAG), which may be unrolled over time and replaced with a feedforward neural network, as opposed to an IIRN, which is a directed cyclic graph (DCG) for which such unrolling is not practical.

Deep Feedforward Neural Networks

It is shown that a feedforward neural network with one hidden layer and a small number of neurons may approximate every continuous function on a compact subset of R n. (Hornik, 1991). This is supported by the universal approximation theorem. The learning of such a network proved to be quite difficult and is not covered by the universal approximation theorem. This calls for the usage of an FFNN with several hidden layers. Another issue that makes learning such networks difficult is that their widths could increase rapidly. It's noteworthy to note that FFNNs with several hidden layers and a limited number of hidden neurons may also prove the universal approximation theorem (Lu et al., 2017). DFFNNs are thus favoured over (shallow) FFNNs in practical applications due to their superior learnability.

Convolutional Neural Networks

A convolutional neural network (CNN) is a special kind of feedforward neural network that uses convolution, ReLU, and pooling layers. A typical CNN is composed of layers from the Feedforward Neural Network family, including convolution, pooling, and fully connected layers. Each connection between neurons in one layer and neurons in the layer above it often acts as a network parameter in conventional ANNs. This might lead to a very high number of parameters. Instead of using entirely linked layers, a CNN uses local connections between neurons, which means that a neuron is only connected to neighbouring neurons in the next layer. As a result, the total number of parameters in the network may be significantly reduced. Also, each link between local receptive fields and neurons makes use of a set of weights. This group of weights is referred to as a kernel. All of the other neurons that connect to their local receptive fields will exchange the results of these computations between the local receptive fields and neurons using the same kernel, which will be stored in a matrix known as an activation map. The sharing attribute is known as CNN weight sharing (Le Cun, 1989). As a consequence, different kernels will generate distinct activation maps, and the number of kernels may be altered using hyper-parameters. As a result, regardless of the total number of connections between the neurons, the size of the local receptive field, or the kernel, determines the overall number of weights in a network.

Table 1 List of popular deep learning models, available learning algorithms (unsupervised, supervised)

Model	Unsupervised	Supervised	Software
Autoencoder	Yes		Keras (Chollet, 2015), R: dimRed (Kraemer et al., 2018), h2o (Candel et al., 2015), RcppDL (Kou and Sugomori, 2014)
Convolutional Deep Belief Network (CDBN)	Yes	Yes	R & python: TensorFlow (Abadi et al., 2016), Keras (Chollet, 2015), h2o (Candel et al., 2015)
Convolutional Neural Network (CNN)	Yes	Yes	R & python: Keras (Chollet, 2015) MXNet (Chen et al., 2015), Tensorflow (Abadi et al., 2016), h2O (Candel et al., 2015), fastai (python) (Howard and Gugger, 2018)
Deep Belief Network (DBN)	Yes	Yes	RcppDL (R) (Kou and Sugomori, 2014), python: Caffee (Jia et al., 2014), Theano (Theano Development Team, 2016), Pytorch (Paszke et al., 2017), R & python: TensorFlow (Abadi et al., 2016), h2O (Candel et al., 2015)
Deep Boltzmann Machine (DBM)		Yes	python: boltzmann-machines (Bondarenko, 2017), pydbm (Chimera, 2019)
Denoising Autoencoder (dA)	Yes		python: boltzmann-machines (Bondarenko, 2017), pydbm (Chimera, 2019)
Long short-term memory (LSTM)		Yes	rnn (R) (Quast, 2016), OSTSC (R) (Dixon et al., 2017), Keras (R and python) (Chollet, 2015), Lasagne (python) (Dieleman et al.,

			2015), BigDL (python) (Dai et al., 2018), Caffe (python) (Jia et al., 2014)
Multilayer Perceptron (MLP)		Yes	SparkR (R) (Venkataraman et al., 2016), RSNNS (R) (Bergmeir and Benítez, 2012), keras (R and python) (Chollet, 2015), sklearn (python) (Pedregosa et al., 2011), tensorflow (R and python) (Abadi et al., 2016)
Recurrent Neural Network (RNN)		Yes	RSNNS (R) (Bergmeir and Benítez, 2012), rnn (R) (Quast, 2016), keras (R and python) (Chollet, 2015)
Restricted Boltzmann Machine (RBM)	Yes	Yes	RcppDL (R) (Kou and Sugomori, 2014), deepnet (R) (Rong, 2014), pydbm (python) (Chimera, 2019), sklearn (python) (Chimera, 2019), Pylearn2 (Goodfellow et al., 2013), TheanoLM (Enarvi and Kurimo, 2016)

Fully-Connected Layer

The fundamental hidden layer unit in FFNN is a fully-connected layer. Fascinatingly, to better simulate the non-linear relationships of the input features, a fully connected layer is frequently placed between the penultimate layer and the output layer for standard CNN designs as well. Due to the numerous factors it introduces, which might result in overfitting, the value of this has lately been questioned (Hinton, 2014). In order to replace the function of linear layers, more and more researchers have begun to build CNN architectures without requiring such a fully connected layer using different methods such as max-over-time pooling (Lin et al., 2013; Kim, 2014).

Important Variants of CNN

The first study to examine how the depth of the network affects a CNN's performance was VGGNet (Hinton, 2014). The Visual Geometry Group and Google DeepMind proposed VGGNet, and it was used to study buildings in depth to 19 levels (e.g., compared to 11 for AlexNet Krizhevsky et al., 2012b). By adding 11 extra convolution layers, VGG19 increased the network's eight weight layers (AlexNet's suggested structure) to 19 weight layers.

The overall number of parameters rose from 61 million to 144 million, but most of them are used by the fully linked layer. According to their reported findings, the top-1 val error (percentage of times the classifier did not give the correct class with the highest score) on the ILSVRC dataset decreased from 29.6 to 25.5, and the top-5 val error (percentage of times the classifier did not include the correct class among its top 5) on the ILSVRC dataset in ILSVRC2014 decreased from 10.4 to 8.0.

GoogLeNet With Inception

Adding extra layers and layer parameters is the most logical technique to increase a Convolutional Neural Network's performance (Hinton, 2014). Two significant issues will arise as a result, though. One is that overfitting will result from having too many parameters, and the other is that the model would be difficult to train.

Google introduced GoogLeNet (Szegedy et al., 2015). Traditional state-of-the-art CNN designs, prior to the invention of inception, largely concentrated on expanding the size and depth of the neural network, which also raised the computing cost of the network. In contrast, GoogleLeNet unveiled a design that combines a lightweight network topology with state-of-the-art performance.

ResNet

In theory, CNNs with deeper structures outperform those with shallower ones (Hinton, 2014). Deeper networks should be more accurate predictors since they can represent high-level information from the input more effectively (Donahue et al., 2014). But, adding further layers is not possible. The authors of the article (He et al., 2016) noted the phenomenon that adding more layers might actually degrade performance. In their experiment, networks A and B had N layers each, whereas the initial N layers had the same structure. Network B had N + M layers. Surprisingly, network B had a larger training error than network B when trained on the CIFAR-10 and ImageNet datasets.

The addition of an additional M layers should, in principle, improve performance, but instead, they received larger errors that cannot be attributed to overfitting. The cause of this is that, unlike the vanishing gradient phenomenon, the loss is being optimised to local minima. The deterioration problem is what we're talking about here (He et al., 2016).

In order to address CNNs' degrading issue and maximise CNN depth, ResNet (He et al., 2016) was developed. The authors of (He et al., 2016) developed a unique CNN structure that, in principle, could be extended to an unlimited depth without compromising accuracy.

Conclusion

We provided a fundamental overview of profound learning models, such as Deep Feed Forward Networks (D-FFNN), Convolutional Networks (CNNs), Deep Conviction Networks (DBNs), Auto Encoders (AE), and Long Short-Term Memory Networks (LSTMs). These models may be seen as the central structures that dominate profound learning at the moment. Also, we looked at related concepts like hard back proliferation and confined Boltzmann machines that are necessary for a specialist to understand these models. The components of the centre compositional structure blocks explored in this study can be used to construct an infinite number of brain network models due to the versatility of organisation designs allowing a "Lego-like" generation of new models. So, having a fundamental understanding of these elements is vital if you want to be ready for future advancements in artificial intelligence.

References

[1] Alipanahi, B., Delong, A., Weirauch, M. T., and Frey, B. J. (2015). Predicting the sequence specificities of DNA-and RNA-binding proteins by deep learning. Nat. Biotechnol. 33, 831–838. doi: 10.1038/n bt.3300

[2] An, J., and Cho, S. (2015). Variational Autoencoder Based Anomaly Detection Using Reconstruction Probability. Special Lecture on IE 2.

[3] Arulkumaran, K., Deisenroth, M. P., Brundage, M., and Bharath, A. A. (2017). Deep reinforcement learning: a brief survey. IEEE Signal Process. Mag. 34, 26–38. doi: 10.1109/MSP.2017.2743240

[4] Bergmeir, C., and Benítez, J. M. (2012). Neural networks in R using the stuttgart neural network simulator: RSNNS. J. Stat. Softw. 46, 1–26. doi: 10.18637/jss.v046.i07

[5] Biran, O., and Cotton, C. (2017). “Explanation and justification in machine learning: a survey,” in IJCAI-17 Workshop on Explainable AI (XAI). Vol. 8, 1.

[6] Bottou, L. (2010). “Large-scale machine learning with stochastic gradient descent,” in Proceedings of COMPSTAT’2010 (Springer), 177–186.

[7] Breiman, L. (2001). Statistical modeling: the two cultures (with comments and a rejoinder by the author). Stat. Sci. 16, 199–231. doi: 10.1214/ss/ 1009213726

[8] Cao, C., Liu, F., Tan, H., Song, D., Shu, W., Li, W., et al. (2018). Deep learning and its applications in biomedicine. Genomics Proteomics Bioinform. 16, 17–32. doi: 10.1016/j.gpb.2017.07.003

[9] Cao, S., Lu, W., and Xu, Q. (2016). “Deep neural networks for learning graph representations,” in Thirtieth AAAI Conference on Artificial Intelligence.

[10] Carreira-Perpinan, M. A., and Hinton, G. E. (2005). “On contrastive divergence learning,” in Proceedings of the Tenth International Workshop on Artificial Intelligence and Statistics (Citeseer), 33–40.

[11] Charles, A. S., Olshausen, B. A., and Rozell, C. J. (2011). Learning sparse codes for hyperspectral imagery. IEEE J. Select. Top. Signal Process. 5, 963–978. doi: 10.1109/JSTSP.2011. 2149497

[12] Chen, T., Li, M., Li, Y., Lin, M., Wang, N., Wang, M., et al. (2015). Mxnet: a flexible and efficient machine learning library for heterogeneous distributed systems. Chimera (2019). Pydbm. arXiv:1512.01274.

[13] Cho, K., Van Merriënboer, B., Gulcehre, C., Bahdanau, D., Bougares, F., Schwenk, H., et al. (2014). Learning phrase representations using rnn encoderdecoder for statistical machine translation. arXiv [Preprint]. arXiv:1406.1078. doi: 10.3115/v1/D14-1179

[14] Chollet, F. (2015). Keras. Available online at: https://github.com/fchollet/keras

[15] Cohen, G., Afshar, S., Tapson, J., and van Schaik, A. (2017). Emnist: an extension of mnist to handwritten letters. arXiv[Preprint]. arXiv:1702.05373. doi: 10.1109/IJCNN.2017.7966217

[16] Dai, J., Wang, Y., Qiu, X., Ding, D., Zhang, Y., Wang, Y., et al. (2018). BigDL: a distributed deep learning framework for big data. arXiv:1804.05839

[17] Deng, J., Zhang, Z., Marchi, E., and Schuller, B. (2013). “Sparse autoencoder-based feature transfer learning for speech emotion recognition,” in 2013 Humaine Association Conference on Affective Computing and Intelligent Interaction (IEEE), 511–516.

[18] Dixon, M., Klabjan, D., and Wei, L. (2017). Ostsc: over sampling for time series classification in R.

[19] Doersch, C. (2016). Tutorial on variational autoencoders. arXiv [Preprint]. arXiv:1606.05908.

[20]Donahue, J., Jia, Y., Vinyals, O., Hoffman, J., Zhang, N., Tzeng, E., et al. (2014). “Decaf: a deep convolutional activation feature for generic visual recognition,” in International Conference on Machine Learning, 647–655.

[21]Doshi-Velez, F., and Kim, B. (2017). Towards a rigorous science of interpretable machine learning. arXiv [Preprint]. arXiv:1702.08608.

[22] Duda, R. O., Hart, P. E., and Stork, D. G. (2000). Pattern Classification. 2nd Edn. Wiley.

[23]Elsayed, N., Maida, A. S., and Bayoumi, M. (2018). Reduced-gate convolutional LSTM using predictive coding for spatiotemporal prediction. arXiv [Preprint]. arXiv:1810.07251.

[24]Emmert-Streib, F. (2006). A heterosynaptic learning rule for neural networks. Int. J. Mod. Phys. C 17, 1501–1520. doi: 10.1142/S0129183106 009916

[25]Emmert-Streib, F., and Dehmer, M. (2019a). Defining data science by a data-driven quantification of the community. Mach. Learn. Knowl. Extract. 1, 235–251. doi: 10.3390/make1010015

[26]Emmert-Streib, F., and Dehmer, M. (2019b). Evaluation of regression models: model assessment, model selection and generalization error. Mach. Learn. Knowl. Extract. 1, 521–551. doi: 10.3390/make10 10032

[27]Enarvi, S., and Kurimo, M. (2016). TheanoLM–an extensible toolkit for neural network language modeling. Proc. Interspeech 3052-3056 doi: 10.21437/Interspeech.2016-618

[28]Esteva, A., Kuprel, B., Novoa, R. A., Ko, J., Swetter, S. M., Blau, H. M., et al. (2017). Dermatologist-level classification of skin cancer with deep neural networks. Nature 542:115. doi: 10.1038/nature21056

[29]Fischer, A., and Igel, C. (2012). “An introduction to restricted boltzmann machines,” in Progress in Pattern Recognition, Image Analysis, Computer Vision, and Applications (Springer), 14–36.

[30]Fodor, J. A., and Pylyshyn, Z. W. (1988). Connectionism and cognitive architecture: a critical analysis. Cognition 28, 3–71.

[31]Fukushima, K. (1980). Neocognitron: A self-organizing neural network model for a mechanism of pattern recognition unaffected by shift in position. Biol. Cybernet. 36, 193–202.

[32]Fukushima, K. (2013). Training multi-layered neural network neocognitron. Neural Netw. 40, 18–31. doi: 10.1016/j.neunet.2013.01.001

[33]Gers, F. A., and Schmidhuber, J. (2000). “Recurrent nets that time and count,” in Proceedings of the IEEE-INNS-ENNS International Joint Conference on Neural Networks. IJCNN 2000. Neural Computing: New Challenges and Perspectives for the New Millennium (IEEE), Vol. 3, 189–194.

[34]Gers, F. A., Schmidhuber, J., and Cummins, F. (1999). Learning to forget: continual prediction with LSTM. Neural Comput. 12, 2451–2471. doi: 10.1162/089976600300015015.