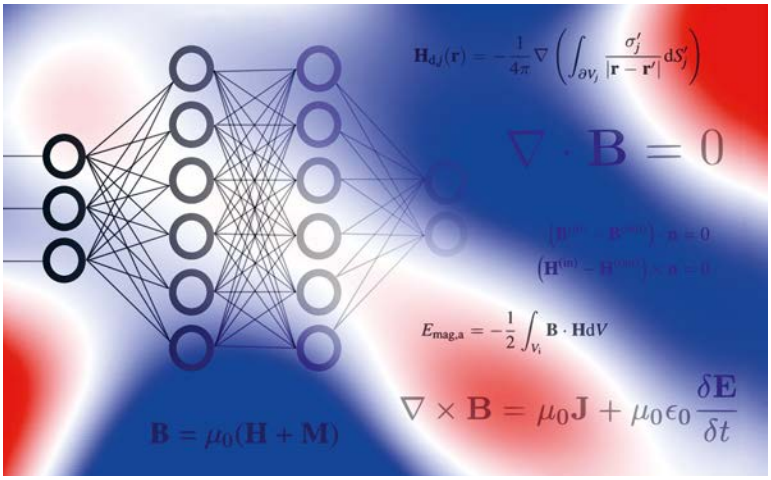


DEEP LEARNING FOR MAGNETISM

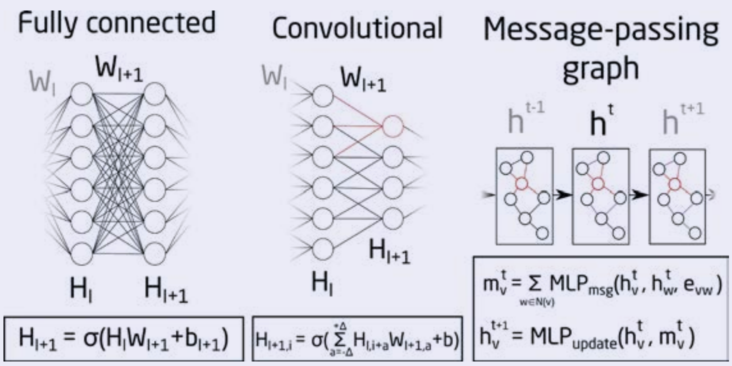
Stefan Pollok and Rasmus Bjørk –
DOI: https://doi.org/10.1051/epn/2022204
Department of Energy Conversion and Storage, Technical University of Denmark, 2800 Kgs. Lyngby, Denmark

In deep learning, neural networks consisting of trainable parameters are designed to model unknown functions based on available data. When the underlying physics of the system at hand are known, *e.g.*, Maxwell's equation in electromagnetism, then these can be embedded into the deep learning architecture to obtain better function approximations.

Magnetic fields are used in a multitude of applications, from MRI scanners to electric motors. These applications must have the magnetic field that ensures the highest application performance. Realising this field can involve everything from analytical or finite element modeling [1] to magnetic field measurements and device prototyping. To ease this process, whether it is reducing the computational load, assisting in characterising field measurements or suggesting new designs, machine learning is being increasingly used in the field of magnetism. This is what we explore in this work, with a special focus on deep learning (DL).

Deep learning - Function approximation with neural networks

Like other machine learning algorithms, DL is data-driven. The data can originate from measurements or simulations. Characteristic of DL is the mapping of input data to the output domain with a biology-inspired neural network (NN) structure. NNs consists of multiple layers of trainable parameters with the idea to extract progressively higher-level features. During training, NN parameters are updated to minimise the difference between the given outputs and the corresponding predictions based on a set of input data.

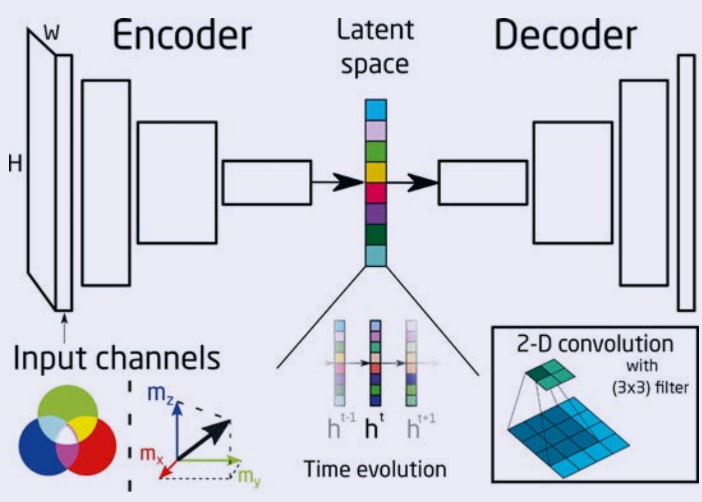


Layer setup of three different neural network architectures: Fully connected, convolutional, and message-passing graph. In the fully connected setup, each node of layer $\mathbf{H}_l$ is connected to the nodes of the next layer $\mathbf{H}_{l+1}$ by an individual weight. The values of these weights are updated during training to approximate a target function. After having multiplied the resulting weight matrix $\mathbf{W}_{l+1}$ with H_l and having added a bias term $\mathbf{b}_{l+1}$, a nonlinear activation function σ completes the mapping. In convolutional neural networks, the weight matrix between successive layers is sparse. As Δ is 1 in the presented case and the weights for all receiving nodes in the layer $\mathbf{H}_{l+1}$ are shared, the weight matrix contains only 3 parameters. In message-passing graph neural networks, the input data is embedded into nodes and edges. For each time step, a message $\mathbf{m}_v^t$ is calculated as the sum of incoming information from neighboring nodes $\mathbf{h}_v^t$ with edges $\mathbf{e}_{vw}$. The state of node v is then updated from $\mathbf{h}_v^t$ to $\mathbf{h}_v^{t+1}$ with $\mathbf{m}_v^t$. MLP stands for multilayer perceptron, which can be any feedforward neural network.

We consider three types of NN, which are visualised in Box 1:

- In the fully connected setup, each incoming node is connected by individual weights with all outgoing nodes. By stacking multiple layers followed by a nonlinear activation function, complex nonlinear mappings can be approximated.
- In convolutional neural networks (CNNs), there is a sparse connection between successive layers and the weights of a small filter matrix are shared, which makes it more data-efficient. The filter convolutes the neighbouring, incoming nodes to create local receptive fields in the next layer. By stacking multiple of these layers, high-level features can be learned.
- In graph neural networks (GNN), the NN nodes can be arbitrarily connected with other nodes. The next layer is created by updating each node with the sum of incoming messages from connected nodes. This can lead to powerful representations.

Although DL was established in the machine learning community some decades ago, it was not before residual connections [2] were introduced that DL was used in magnetism. Instead of modeling the unknown, underlying mapping between input-output pairs directly, DL architectures were built to model the residual between input and output. These reformulated mapping functions exhibit properties that are more easily learned during training.



Encoder-decoder structure realised with multiple 2-D convolutional layers. The input data has shape C×H×W, where C is the number of channels, H is the number of pixels in the height, and W is specifying the width of the input. The number of channels increases in the encoder part from layer to layer while the window size becomes smaller. By doing this, higher-level features can be extracted progressively while preserving information. A rich, low-dimensional representation of the input is obtained in the latent space. Optionally, time evolution or other operations can be performed computationally efficient in the latent space. In the decoder, the convolutional layers are translated to increase the window size back to a desired resolution $C_{out} \times H_{out} \times W_{out}$. In Kovacs *et al.* [5], the output of this structure shall resemble the magnetisation vectors of the input. This special case of an encoder-decoder setup is called an auto-encoder. Here, the main learning objective of the neural network weights of the 2-D convolution filter matrices is to find the rich latent space representation. In Khan *et al.* [4], the output of the decoder shall represent the magnetic field strength as target domain while feeding material information into the encoder. Originally, this DL architecture was used in computer vision to extract information of RGB images.

Motivation for deep learning for magnetism

We study the application of DL in magnetism because this is an ideal subset of physics for which to study if DL can extend our modeling capabilities, expand our physical understanding, and help realise more efficient technologies. The reason for this is that the physical laws governing magnetism has been understood since Maxwell, and the equations give rise to a predictable physical behaviour. Said in other words, magnetism does not contain complexities such as turbulence or chaotic behavior. This means that magnetism is an ideal proving ground for testing DL before moving on to more complex physical domains.

In physics, DL has been employed as a surrogate model for computationally expensive models, as a PDE solver without meshing, or in analysing tasks where data is available but the underlying physics is unknown. Within magnetism, DL has been employed in magnetostatics, micromagnetism, and electromagnetic (EM) setups. In magnetostatics, the calculation of magnetic fields is performed on the macroscale, *e.g.*, from low-frequency EM devices [3] or from permanent magnets [4]. In micromagnetism, the magnetisation inside a material is found and how this responds dynamically to, *e.g.*, the application of an external field [5]. EM setups include design optimisation of electric machines [6], controller design for electric drives [7], and finding the solution of EM field scattering [8].

Convolutional neural networks in magnetism

In 2019, DL in micromagnetism was first used by Kovacs *et al.* [5] to approximate the time evolution of the magnetisation in a thin film, as governed by the Landau-Lifshitz-Gilbert equation [9]:

$$\frac{\partial m}{\partial t} = -\gamma \mathbf{m} \times H_{eff} + \alpha \mathbf{m} \times \frac{\partial m}{\partial t},$$

where $\mathbf{m}$ is the local magnetisation, γ is the gyromagnetic factor, α the dampening constant, and H_{eff} the effective field, usually composed of four terms representing the exchange, demagnetisation, anisotropy and • • •

••• applied field. The authors employed a CNN in an encoder-decoder structure as depicted in Box 2. The idea behind the encoding pathway is to progressively extract higher-level features and to obtain a rich, low-dimensional representation of the data in a latent space without loss of information. In the autoencoder setup, the original data is reconstructed from latent space with a decoder. In this low-dimensional space, the time integration of the local magnetisation was learned with a second, fully connected NN based on previous states and the applied field. The method showed good agreement with a finite difference model.

Around the same time, Khan *et al.* [3] demonstrated that Maxwell's equations for low-frequency EM devices can be modeled by CNNs. In a similar encoder-decoder as shown in Box 2, the magnetic field is approximated given only design geometry, excitation, and material properties. The input material data was formulated similar to multichannel images with a resolution of H × W. This was done because DL was mainly developed within computer vision, where it extracts information from RGB images. Therefore, the magnetic problem was represented similarly, *i.e.*, by meshing the 2-D area around the EM device into a uniform grid. The material information *mat* at the center of each pixel was assigned to the channels previously used for the color components, resulting in a *mat* × H × W representation. The preprocessed input data was then convoluted with multiple stacked CNN layers with decreasing resolution but increasing number of channels. Thereafter, the compressed state was upsampled to the original resolution with stacked transposed convolutional layers. Instead of using it as an autoencoder, the norm of the magnetic field in each pixel center was learned directly. During training, the CNN parameters were updated to minimise the difference between predictions and the solution of a finite element model. For unseen EM configurations, this resulted in a normalised root mean square error of ~1%.

This early work in DL for magnetism inspired other researchers to switch the mapping direction between input and output of the DL network [4] as shown in Box 3. Now, based on a given magnetic field, the properties of a permanent magnet to generate that specific field can be inferred, allowing for inverse design of permanent magnet configurations.

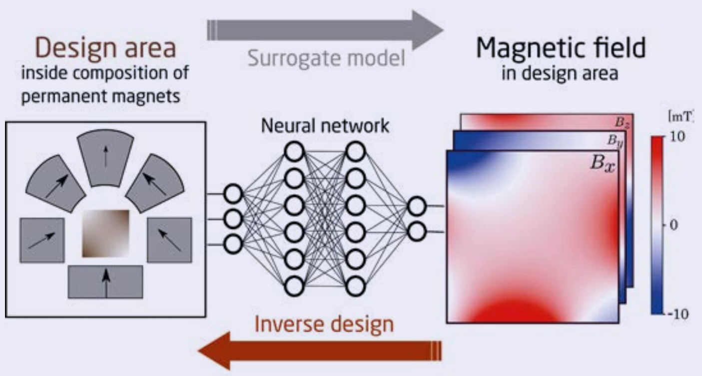
Application of more sophisticated deep learning architectures

So far, we have only described CNNs in an encoder-decoder fashion. Since the work of He *et al.* [2], more sophisticated DL architectures have emerged, which have been recently applied to magnetism.

For instance, generative adversarial networks are used for magnetic field prediction [10]. Given only a few points measurements in a 2-D area of interest, a trained generator network, consisting of multiple layers of CNNs, can predict missing magnetic field values with an error below 6 %. This is achieved by updating the generator parameters in such a way that a second NN, a so-called critic, cannot differentiate between real magnetic fields and generated magnetic fields. Additionally, reconstruction and physical constraints are included during training.

Such an embedding of the underlying physical laws into a DL architecture has been pursued further with so-called physics-informed neural networks [11]. Here, the underlying physical laws, *i.e.*, Maxwell's equations, along with surface and boundary conditions are directly integrated into the loss function of the DL architecture. Recently, this setup was extended to parametric magnetostatic problems by using a 10-dimensional parameter vector to allow for more flexibility [12].

Another interesting direction is the use of deep reinforcement learning in magnetism to perform controller design for magnetic technologies. Degrave *et al.* [7] recently showed that a trained NN can autonomously control the magnetic actuator coils to shape and maintain plasma inside tokamak vessel.



Possible pathways for the use of deep learning in magnetostatics. In many modern applications like magnetic resonance imaging or electric motors, permanent magnets of different shapes and magnetic properties are assembled to produce a magnetic field, whose field strength and homogeneity match the requirements for the given application in a design area. When finite element modeling becomes computationally expensive, a trained neural network can approximate Maxwell's equations and serve as a surrogate model during topology optimisation. As the mapping direction between input and output can easily be interchanged in a deep learning setup, a neural network can also be trained to take the desired magnetic field as input and to predict the structural properties of the composition of permanent magnets necessary to produce such a field.

Conclusion and Outlook

We have described the use of DL within magnetism and documented how DL has been used to calculate magnetic fields, to predict inverse magnet properties and to time evolve a micromagnetic model. Future directions of DL with magnetism can be to represent, *e.g.*, compositions of multiple permanent magnets as graphs. As graph structures do not rely on uniform grids in contrast to CNNs, magnetic fields can be predicted more efficiently. An instance of message-passing GNNs for predicting molecular properties was developed by Schütt *et al.* [13]. Recently, a general framework for modeling dynamics of physical systems with Transformers, a fully connected version of GNNs, was proposed by Geneva *et al.* [14]. This shows how exciting this area is and that it is likely to expand and evolve in the coming years. ■

About the Authors

Both authors are from the Technical University of Denmark, Department of Energy Conversion and Storage.
Stefan Pollok has been a PhD-student at the department since 2019 and is an expert on machine learning.

Rasmus Bjørk is professor and has worked with magnetism since 2007, both within modeling and experimental devices. They both develop the open-source modeling framework MagTense.

References

[1] R. Bjørk *et al.*, *J. Magn. Magn. Mater.* **535**, 168057 (2021)

[2] K. He *et al.*, Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR), 770 (2016)

[3] A. Khan, V. Ghorbanian, and D. Lowther, *IEEE Trans. Magn.* **55-6**, 7401305 (2019).

[4] S. Pollok, R. Bjørk, and P.S. Jørgensen, *IEEE Trans. Magn.* **57-7**, 2101604 (2021)

[5] A. Kovacs *et al.*, *J. Magn. Magn. Mater.* **491**, 165548 (2019)

[6] H. Sasaki, and H. Igarashi, *IEEE Trans. Magn.* **55-6**, 7401305 (2019)

[7] J. Degrave *et al.*, *Nature* **602**, 414 (2022)

[8] S. Qi *et al.*, *IEEE J. Multiscale Multiphys. Comput. Tech.* **5**, 83 (2019)

[9] T.L. Gilbert, *IEEE Trans. Magn. 40*-**6**, 3443 (2004)

[10] S. Pollok *et al.*, "Magnetic Field Prediction Using Generative Adversarial Networks" (2022), arXiv:2203.07897. [Online]. Available: https://arxiv.org/abs/2203.07897

[11] A. Kovacs *et al.*, *J. Magn. Magn. Mater.* **548**, 168951 (2022)

[12] A. Beltrán-Pulido, I. Bilionisb, and D. Aliprantisa, "Physics-informed neural networks for solving parametric magnetostatic problems" (2022), arXiv:2202.04041. [Online]. Available: http://arxiv.org/abs/2202.04041

[13] K.T. Schütt, O.T. Unke, and M. Gastegger, *Proc. Int. Conf. Mach. Learn. (PMLR)* **139**, 9377 (2021)

[14] N. Geneva, and N. Zabaras, *Neural Netw.* **146**, 272 (2022)