Deep learning for classifying molecules by their magnetic ordering

1) The Project Scope and the Research Gap.

The project scope concerns scientific machine learning, particularly the classification of chemical compounds based on the property called magnetic order. The data is obtained from the "Material Project" database, which tabulates chemical formulas and magnetic orders.

The input "x" will be the molecular formula which will be decoded before being fed into the model. At the same time, the output "y" will be the magnetic order which is one of four classes that are mutually exclusive making the SoftMax activation function the choice for the output layer. Feeding the chemical formula to the network is done by first extracting an array of numerical features like molecular mass, average molecular mass, number of atoms, etc. Another way to encode and feed the molecule's information is by representing it as a graph where each node is an atom and each edge is a bond. Those graphs that encode the molecules are called graph neural networks or GNNs, and they are fed into graph convolution networks or GCNs that will classify them.

The model that could be used will depend on the type of encoding of the molecule, if an array of numerical values was used then the dense, fully connected, and conventional neural networks will be used. While if molecules were decoded by GNNs then graph convolutional networks or GCNs will be used. The model with the best performance as well as the best efficiency will be used. The type of input encoding and model selection will be decided by implementing and testing each idea.

2) The Type of the project and literature review:

The project type will be "Beating the Classics", meaning using a deep learning

approach in domains and applications where non-DL methods were used. The following papers may show the current methods used to classify molecules by their magnetic order.

The first is from(Jang et al., 2023) paper where electronic structure was used as an input to estimate the molecule's magnetic order using decision trees. The second paper (Khatri & Kashyap, 2023) is using a random forest algorithm to predict the magnetic order of Magnesium-based molecules. So far deep neural networks have not been used in this application, hence this project will introduce an opportunity to fill this research gap.

3) The Dataset type and the proposed split of the dataset:

The dataset is obtained from the "Material Project" database. The data is tabulated as two columns of the chemical formula of each molecule and the magnetic order respectively as shown in **Table 1**. We can restrict the data to be experimentally observed limiting them to around 48,000 data points, yet with high quality. If we relax this requirement we can have access to all data points whether experimental or from numerical simulation, and this will give us around 150,000 data points.

Table 1: A sample of the dataset where the formula is the input "x" and the magnetic ordering is the output "y"

Formula	Magnetic Ordering
$Co(IN_3)_2$	AFM
VC_3	NM
$Mg(BrN_3)_2$	FM
$KAlN_4$	FM
$Cu(BrN_3)_2$	FM
$Ru(IN_3)_2$	NM
SiO_2	NM

The split for the dataset also shown in **Table 2** will be the following:

1. The Training Set: this will be used to train the model and it will be the distribution that includes both experimental and simulation data because the training process requires lots of data,

2. The Dev Set: The dev set or the development set (also called the validation set) is used to test the generalization error or the model to reduce the variance. The dev set is also used to make decisions like changing the hyperparameters like regularization or the architecture of the model. It will be from the distribution of the data that is empirically observed. The reason to choose this distribution is to make the quality of the prediction match that of empirical predictions. This will make the distribution of the training set and the dev/test sets different which could introduce a data-mismatch error

which will be solved by adding the Training-dev set discussed ahead.

3. The Test Set: The test set is like the dev set but it is not used to tune any hyperparameter and it is used to test the generalization of the model as a final test. It will be from the same distribution as the dev set and the same size as well.

4. The Training-Dev Set: The Training-Dev set is introduced to solve the issue of using two different distributions in training and validation/testing. So the Training-Dev set will be from the same distribution as the Training set and it will tell us if there is a variance issue when comparing the training error with the training-dev error. If the training-dev error and the dev-error are different this will tell us that the issue may not just be variance but also a data-mismatch problem which can be handled differently from variance or bias problems. Hence the importance of this newly added training-dev set. This idea is mentioned by the researcher Andrew Ng in his course "Deep Learning Specialization".

Table 2: The dataset distribution and percentage

Dataset	Distribution	Percentage
Training set	Distribution 1: Empirical + Simulation	90.00%
Training-Dev set	Distribution 1: Empirical + Simulation	3.33%
Dev set	Distribution2: Empirical only	3.33%
Test set	Distribution 2: Empirical only	3.33%

The error analysis used for fine-tuning the hyperparameters and for architecture selection will shed light on the type of problems presented as well as the strategies to solve them. The principle of “orthogonalization” will be used which states that trying to implement a set of strategies that solve a single problem at a time like for high bias or high variance and so on, without using a strategy that may attempt to solve both simultaneously to make our analysis easier and more effective. **Table 3** summarizes the strategies used for each problem and error type.

Table 3: The Error Analysis Process

Error Type	Problem	Strategy
High Training Error	High Bias	A more complex model, more epochs, and a better optimization algorithm
High Training-Dev Error	High Variance	More data, higher regularization term
High Dev Error	Data-Mismatch	Make the Two distributions more similar (synthetic data)
High Test Error	Generalization problem, overfitting the dev set	Bigger Dev Set
High real-world performance error	The model Target is not met	Change the dev/test sets to make them more similar to real-world data

4) The work-breakdown structure:

The work will be split into the six stages below:

1. Extracting and preprocessing the data.
2. Decoding the information of each molecule either using an array of numbers or a GNN.
3. Building a NN model fast and doing error analysis (bias, variance, data-mismatch,

and manual error analysis for a random sample of the errors)

4. Reiterate step 3, based on the error like if we have a high bias problem then make the model more complex or if we have a high variance problem then increase regularization or if possible increase the number of data like new data or data augmentation and so on with other error types)
5. After iterating and building a model with good performance, build an application or an API to be used by other users.
6. Documentation: Report writing and presentation preparation.

Table 4: work-breakdown structure

No.	Start Date	End Date	Task	Duration in days
1	26.10.24	01.11.24	Extracting and preprocessing the data.	7
2	02.11.24	15.11.24	Decoding the molecular fingerprints	14
3	16.11.24	29.11.24	Building a model quickly with error analysis	14
4	30.11.24	17.12.24	Iterating and fine-tuning	18
5	18.12.24	31.12.24	Shipping and API building	14
6	01.01.25	21.01.25	Documentation	21

5) The tools used, software and hardware:

The hardware will be my PC as well as some free cloud computing platforms if needed like Google Colab, Paperspace, and Azure Machine Learning. Google Drive and Github will be used to store the data and the code.

The software and tools will be Jupyter Notebook and Python libraries and frameworks like Tensorflow, and Keras.

The textbook reference will be “Machine Learning in Chemistry” by Cartwright, H. M. for the implementation of decoding the chemical formula.

References:

Cartwright, H. M. (2020). Machine learning in chemistry: The impact of artificial intelligence. Royal Society of Chemistry.

Jang, Y., Kim, C. H., & Go, A. (2023). Classification of magnetic order from electronic structure by using machine learning. Scientific Reports, 13(1), 12445. https://doi.org/10.1038/s41598-023-38863-7

Khatri, Y., & Kashyap, A. (2023). Advancing magnetic material discovery through machine learning: Unveiling new manganese-based materials. APL Machine Learning, 1(4). https://doi.org/10.1063/5.0171320

Taoreed O. Owolabia, b., ∗, Kabiru O. Akandec, , S. O. O., Abdullah Alqahtanid, & Aldhafferid, a. N. (2017). Incorporation of GSA in SBLLMbased neural network for enhanced estimation of magnetic ordering temperature of manganite. Journal of Intelligent & Fuzzy Systems.

Thomas N. Kipf, M. W. (2016). Semi-Supervised Classification with Graph Convolutional Networks. arXiv.