Enforced Fairness for Machine Learning in Insurance

CPSC 471 Project Proposal

Tristan Brigham

February 2024

Initial Brainstorming Questions: Sciences

1. **Data Modalities and Deep Learning Applications**

   (a) *Examples of deep learning with this modality:* In the sciences modality, there is a seemingly infinite number of papers and machine learning that is being done all of the time. Some of the most impressive examples that I have seen include using physics-informed neural networks to simulate fluid dynamics and other dynamical systems

   (b) *Tasks applied using deep learning for science:* Starting with generative models, we know that there are many models that are used to simulate physics and movement using generative models. These ensure that we can approximate physics and show us how physical machines that we build might work in the real world.

   Additionally, we can predict the likelihood that some event happens in the future that can serve dividends to risk modeling and event simulation. This crosses the boundary between generative (by informing how the world state evolves for a model environment) and discriminative (by quantifying the risk or likelihood that some event happens).

   On the discriminative side, we can understand and predict the likelihood that things like radioactive decay happens at any timestep. This allows us to train a function that is able to effectively simulate things like the decomposition of an atom.

2. **Personal Interests and Projects in Deep Learning:** I do – one project that I have been particularly interested in for a while (and is the reason that I am taking this class) is machine learning's application to fields such as finance and insurance. However, something that is gaining importance in the industries is explainability and transparency of the networks that are being implemented in the world.

   I have thought about investigating how to create a neural network that is verifiable and efficient for industries such as insurance for a while now, and believe that this project would be the best time to start a project like that!

   This project would use tabular data from the insurance industry to try to create models that produce equitable insurance predctions with high fidelity and low variance. I believe that this is something that is legitimately valuable and a problem for the industry, and I like to think that I am solving problems instead of doing things just because.

3. **Improving the Field with Key Techniques**

   Impact of explainability, adversarial robustness, privacy, fairness, and efficiency: I think that explainability is the largest issue for the industry. If I get rejected for a loan or get an absorbitantly high insurance premium, then I would like to understand why I am being rejected or accepted. Additionally, with valid explainability outputs from the model, then it is easy for practitioners who are actually applying the models to understand and evaluate the model.

   However, all of the metrics that are mentioned above are very important for the industry (potentially to a higher extent).

   First, privacy is another important one for the models. It is crucial that we are not able to reverse engineer the public-facing pricing models for insurance so that the personal data from people can be backed out and stolen from the model. For instance, model inversion attacks should not be possible on the external facing models.

   In the same vein, to ensure that the insurance company that is using the model is making money and making positive expected value bets, it is important that the models that they have are adversarially robust. People should not be able to easily figrue out how to trick the model to give them better loan terms, approve them for insurance, or something else.

Finally, fairness is crucial because comapnies that do not have fair models are likely to be sued or fined by regulators. It is also a PR issue where they are making sure that the company is not dragged through mud for having unfair models.

4. **Exploration of Hugging Face**

   (a) *Top models and datasets on Hugging Face:*
   Top Models: stabilityai/stable-cascade (Text-to-Image), google/gemma-7b and google/gemma-7b-it (Text Generation), CohereForAI/aya-101 (Text2Text Generation).

   Top Datasets: HuggingFaceTB/cosmopedia, BioMistral/BioMistral-7B, CohereForAI/aya_dataset.

   Models and datasets on Hugging Face by their popularity (measured through downloads), update frequency, and user ratings (likes or stars). The models that I mentioned above used simple cross-entropy and NLP-specific loss functions which are optimized for fine tuning in language task settings.

   Users might encounter trust issues with the models when they believe that the data is biased or there are other fundamental flaws with the model architectures. A common criticism that has been leveraged recently is that the large language models that have been trained on book-specific data are more liberal than other models because of the general political leanings of books, while website-trained large language models veer more right and conservative due to the construction of internet data.

   Something that I noticed off of the bat is that most of the data and models that are on the website relate to text generation and large language models. This could be a function of the recent interest in things like large language models for everything due to ChatGPT. However, I find that most of the work that is being done with LLM's not not extremely interesting to me.
   I could use some of the datasets on the page, but I was also able to find some kaggle competitions where I could find other models that could represent my baseline model that I am trying to beat with my project.
   I would rather find interesting insights for things in the realm of tabular data. I thought that this dataset that I found that has information on the co-expression of genes was highly interesting. Additionally, this Boston housing prices dataset is another dataset that I would be interested in investigating.
   Another dataset that is more relatable to the work that I am interested in doing is the Prudential Life Insurance dataset which allows me to predict things like life outcomes and insurance premiums using machine learning.

5. **Trustworthiness Improvement Methods**

   (a) *Core ideas from literature or Arxiv for enhancing trustworthiness:*
   I investigated a few of the papers in greater detail below in order to understand the observability methods that are being employed.

      i. The first paper that I decided to investigate further was the LIME and text classification paper. Although this is a more general model and therefore may not be the most pertinent to the tasks that I am trying to accomplish with this process since it is not specific enough, I thought that the paper was highly interesting to read for the context that it provided beyond the basics that were provided in class.
      First, I found it interesting that this resource used the CAPTUM library as well. This seems to be a relatively common framework that I had never heard of before this class despite trying to be as involved as I could in the world of transparent machine learning.

One of the things that I appreciated was the use of the simple embedding-bag classifier which made the classification and scoring processes rather easy to process in the grand scheme of things. This helped me understand exactly how using the local model helps to explain the predictions that are being made.
This process can clearly be helpful in a wide variety of applications. I can imagine that it would be very helpful for something like tabular data as well where if I generate spurious or auxillary data, then I may be able to run a classifier on the data and understand exactly what the driving factors are behind such a prediction.

6. **Metrics for Trustworthiness**

   (a) *Analysis of existing metrics:*
   While I think that a lot of the methods for explainability and trust are rather good right now (most of them involve some sort of removing an input from the model or gradient calculation), I believe that there are more ways that this could be done. I start with a current investigation of the current methods below.
   First, the gradient based methods are the most mathematically rigorous attempts at explainability that I have seen. I think that the calculation of these gradient vectors shows a deliberate move towards more rigorous and clear explainability methods whereas before a lot of the work that was being done involved a lot of methods such as leaving inputs out which were not as rigorous.
   On the other hand, we still do not know whether these gradient based explainability methods actually do what they say that they are doing. For instance, we can have seemingly random parts of the image have rather large gradients for no good reason at all or overfitting, and by the time that we discover that it is often too late.
   On the other hand, leaving some percentage of the inputs out of the input of the model seems to make sense, but there is no way to entirely remove these inputs because of the way that the matrices are constructed because of the understanding that even doing nothing is doing something. The biases can clearly change the flow of all of the information through the network which is no good.
   To change this, I think that there should be more of a focus on two things: 1) outside methods that are not using the gradient or 2) actual manipulation of the weights.
   Some ideas that could exist in the first category are a more rigorous understanding and evaluation of things such as untraining methods for neural networks. On the other hand, actually manipulating the matrices to force explainability through something more surgical and strategic than regularization would be another good method to ensure that we only get the inputs that we want as importatn features of the model and we have observable models.

7. **Model Architecture and Trustworthiness**

   (a) *Ideas for model architecture adjustments:* Although this is not necessarily a question, I will provide a response anyways. One of the methods that I would like to implement as my project is a custom loss function, which, by the layer number, changes the number of nodes that are going to be regularized along with to what extent. This means that we are able to better understand which notes are the important notes throughout the course of the model.
   Additionally, I would like to implement a strategy where different inputs should activate different notes, depending on the input elements that they reflect. This means that we're better able to understand how different elements and attributes to different inputs affect the overall computation path and course of the model.

8. **Data Transformation for Improved Attributes**

   (a) *Examples of effective data augments:* There are lots of methods that have been employed in order to make sure that the model that we get as an output after training is explainable and helpful.

One of the methods that I have found that is most effective is higher order, variations, and combinations of input variables. Off times, these non-linear variables, when they are put into regressions, are able to be more accurate and act as linear variables. When we have this larger amount of variables that the regression can consider, this allows for more generalizable, as well as accurate models as outputs.

Another method that I have found to be particularly helpful is adding noise to the inputs. If we want to preserve privacy, a good way to do, this is to add noise to the inputs. That way, if we do have a model inversion attack deployed on our machine, learning model, the data that we lose is actually noisy. However, over the course of training the entire model the average should still be very close to the mean that we would've gotten anyways. This is because we would ad zero-mean noise to the data.

Problem Definition: Observable AI in Insurance

This is an early iteration of my project proposal that will likely undergo changes before the next submission.

The tradeoff between explainability and efficiency in current machine learning models is extreme. I seek to provide the framework and example implementations of how fairness and explain ability can be directly integrated into the training process for machine learning model.

The challenge of crafting machine learning models that are efficient and explainable in highly sensitive topics such as healthcare and insurance is a pertinent and highly investigated topic today. Machine learning has the opportunity to drastically transform the efficiency and workflows of such industries much like every other line of work that it has touched so far, but the sensitivity of the topics being investigated and manipulated in insurance and health necessitate more explainable yet efficient models before such models can be initiated in related institutions.

This problem, however, is not only one in theory. The problem of non-explainable machine learning models being used in such sensitive applications presents the potential for highly unfair or biased machine learning models to propagate inequities and injustices in society. And, as we continue to put more trust in the statistical models because of the fallacious belief that such algorithms and mathematical processes equates with justice and fairness, we only continue to increase the risk of a disastrous black swan event happening and eliminating years if not decades of trust that have been developed and allowed machine learning models to flourish.

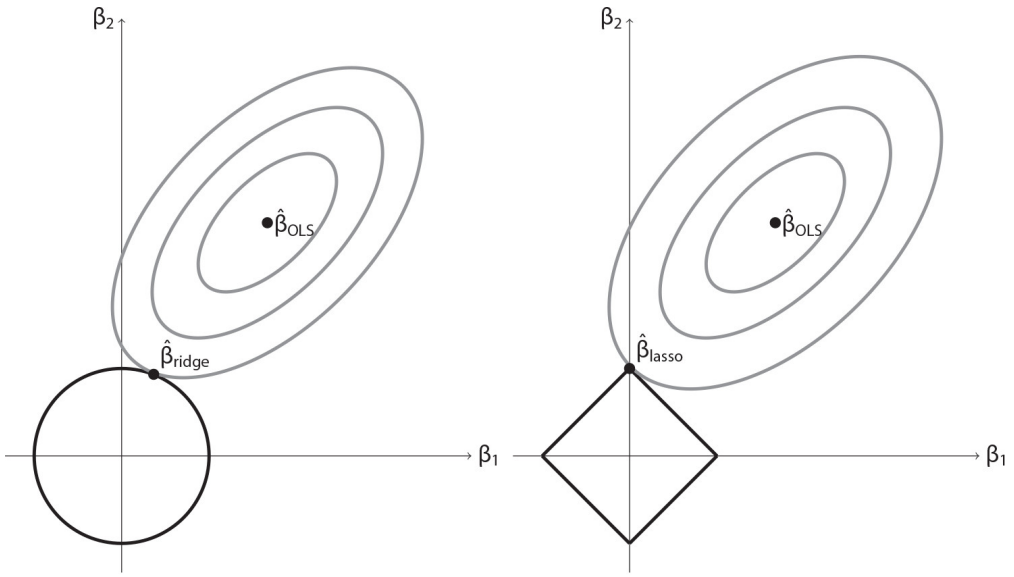
However, even if these problems are not found in the near term and the problems persist, allowing biased and invalid models to provide predictions and guidance for practitioners in society ensures and guarantees that systemic biases that have existed in the past continue to propagate (and potentially expand) their impact. Achieving a balance between efficiency and equity in machine learning models and other emergent technology in sensitive spaces is not only an intangible problem that I have made up – it is real and can have serious effects if it is not handled effectively and with care.

I propose a process which seeks to bring explain ability and fairness without compromising on efficiency into the insurance space using tabular data and mod modified statistical methods. This is merely the first step towards explainable and ethical AI, and will provide the groundwork for further advances in the future. There are several current methods that exist that can be used in order to enforce some degree of explainability. We investigate them below.

1. **Simple Regularization:** Current methods that exist for regularization include L_0, L_1, L_2, and L_∞ amongst others. While each of these regularizations serves a different purpose in the grand scheme of explainability, L_1 is frequently used because of its linear and symmetric structure in terms of the unregularized objective function. If we investigate the graph below showing the "equally good" values for the different regularizations, we can see that L_1 (lasso) regularization strikes a healthy balance between forcing many parameters to zero, and only performing regularization on a select subset of parameters. However, it may be optimal in some cases to use L_2 regularization (ridge regularization) to enforce a smoother equally good boundary and stronger regularization on outliers to the data. Regularization function such as L_0 gives us a plus-shaped "equally-good" graph, while regularization functions with large parameters such as L_∞ give us square-like "equally-good" graphs. [**Ahrens2020Lassopack**]

2. **Adversarial Loss:** This method consists of training a generator and a discriminator in order to try to generate data representations from the latent space input that closely mimic some target outputs. A discriminator can be programmed in a series of different ways such that it is evaluating different metrics for the generated data. One way that the discriminator can be trained is to discern whether some output is logically sound or not. Given enough time, this can lead to explainable and observable outputs being generated by the initial generator model.

   Given the keen and heightened targeting that we can do with this generative-discriminative method, we can ensure that the outputs mimic real life and are therefore somewhat explainable. Additionally,



(a) Ridge (b) Lasso

Figure 1: Equally Good Visualizations for Regularizations [**Ahrens2020Lassopack**]

we can map the throughput of a neural network to understand what kind and aspects of an input influence portions of the output. [**Tung2019SimilarityPreservingKD**]

3. **Quantile Loss:** This method works by allowing a more nuanced understanding of the models predictions. The distribution of the response variable is changed through the model in order to get us a better perspective into the inner workings of the matrix inside of the neural network.

   This provides us with enhanced observability into tail risk of machine learning models by allowing us to predict certain quantities instead of the mean output of a neural network. Effectively, this is allegorical to red-teaming a machine learning model and allows us to understand the variability of the outputs.

   Additionally, this means that the model is more robust and versatile which can often lead to more explainable neural network weights as redundancy is forcibly built into the model. [**Kang2021DataFreeKD**]

4. **Ranking Losses:** this can provide us intuition into the fairness of the model by highlighting overweighted and underweighted classes in the training data. We can understand by the consistency of the rankings of different labels whether the model is likely to prioritize or discriminate against certain output classes. Additionally, this helps us gain insight into the thinking of the model when we apply additional characteristics and outputs, such as probabilities of the final class outputs. If a model is highly confident in a consistently distributed number of output classes, then we might believe that the model is more fair and making decisions more based on the inputs. On the other hand, if the model is highly confident in a select few number of class outputs, then we might believe that the model is biased or skewed towards those classes regardless of the input. Additionally, the distribution of the ranked outputs gives us insight into the other potential classes that we could encounter with slight perturbations in the data.

   In essence, this helps us better understand where the model excels and struggles which offers us a clearer view of its internal workings and "thinking". [**Yang2023BalancedKD**]

5. **Contrastive Loss:** This version of loss effectively helps us understand the key differences between different inputs in our data. Additionally, it extends the models of buses to ensure that it understands

the subtle differences that exist between data points that we are attempting to classify using the machine learning model.

This can be crucial for helping with feature extraction and model precision as we can allow relatively similar though fundamentally different examples to represent contrasting classes when we are training the machine learning model.[**Wang2020Understanding**]

6. **Knowledge Distillation Loss:** This allows us to fine-tune what portions of the data and features our model should be looking at. By teaching the model to only consider certain parts of our extensive knowledge base that we can potentially feed into the model, we ensure that the outputs are applicable and valid for the context of our machine learning model. Part of the benefit of distilling or knowledge to a more simplistic representation and latent space is the potential for higher fidelity in our model weights and in explaining outputs due to the smaller number of features that it must represent and process in an input.

   This allows us to better understand the factors that influence an output, and can potentially open the possibility of cross-validating the entire model such that we get the influence of certain factors on the input and output of the model. [**Gou2020KnowledgeDS**]

However, the problem continues to persist with machine learning. Several empirical studies have shown that the methods deployed above to deal with the problem of explanation come with detrimental effects on the accuracy of the treated models.

We aim to provide the first end-to-end method for explainable neural networks with relation to tabular data that can be used in a high-intensity and fidelity environment such as an insurance agency or hospital. The model will prioritize explainability so that human evaluators can decide whether to trust the outputs and logic or not. The process will be done through advanced gradients and loss functions, integrated observability functions, and low-impact model checks and hashing to ensure that there is no tampering with the model.

The final output of the entire program will be a compressed model with a hash that can be used to check whether the model has been tampered with since the original training. This ensures that the model is not only explainable and fair after training, but also that it can continue to be veritably fair into the future.

Relevance to Trustworthy Aspects

Implementing explainable decision modelling for neural networks increases the amount of trust and fidelity exhibited in and by high-importance decision neural networks by improving the model standing in terms of our four principles for trustworthy machine learning: robustness, explainability, privacy, and fairness.

We elucidate how a more transparent method of explaining the decisions made in such a neural network influence each of the verticals below:

1. **Robustness:** Using the gradient and neural network note based fairness and observability metrics, we were able to better understand where the outlier data inputs drive or outputs. If we are able to understand this, we're able to employ methods such as model, knowledge distillation, gradient clipping, as well as random data generation and interpretation in order to train the model on more extreme inputs. Performing this process on the system that we create would allow us to ensure that the model does not act erratically with marginal data inputs.

   By ensuring that we understand how the model thinks, we're better able to protect against erratic inputs nullifying our previous training schema of the model, and will increase end users' trust that the model will generate sound outputs.

2. **Explainability:** The entire point of this framework and problem is to ensure that the outputs that we are getting from our neural network are explainable and justifiable. If we end up with a model that

is not explainable or results in outputs that are not accompanied by sound reasoning, then users with either over-trust or under-trust the model. They will rarely treat the model as a helpful suggestion which should be critiqued given unexplainable outputs in high-sensitivity situations and will either stop using the model entirely or delegate everything to the model.

The explainability aspect of this model ensures that the explanations are valid and reflect the true "thinking" of the neural network. This allows administrators and those that interpret the model to deploy skepticism where the model output warrants it while taking advantage of the efficiency and veritability gains that can be gleaned from explainable and observable AI.

3. **Privacy:** By gaining a better understanding of how the model does its thinking, we are better able to protect users against having their personal information leaked through the model. While this would be a marginal benefit that will likely not be implemented given the time constraints, better observability into the weights and processes of the machine learning model to understand what information is being stored could lead to un-training schema in the future that improve upon current frameworks.

   On the topic of un-training a model: if I am able to get a tabular data model that is able to accurately and legitimately emulate a model that would result in removing some set of features or training points, then this would constitute a large breakthrough in the world of neural networks which are, by definition, highly complex and do not frequently converge in a nice and deterministic model depending on the random seed that is being initialized for the model.

   More observability into the weights and gradients in general can help us better implement schema such as differential privacy to ensure that the adjustments being made in the neural network accurately reflect population characteristics and do not lead to personal information being leaked through the model constraints.

4. **Fairness:** This is another key point of the framework. We are attempting to create customized metrics that ensure that the model outputs are representative and valid without compromising seriously on the effectiveness or efficiency of the model. We must ensure that the model that we end up creating is not entirely irrelevant due to making poor decisions in the name of equity, while also ensuring that the model does not simply exist to continue to advance racial, gender, or other inequities that already exist in the world.

Methods

I will try advanced loss functions, custom batch-normalization, and a series of other methods to enforce fairness and explainability in my machine learning model. Through this process, I will try 3-main methods along with others that I can think of through the brainstorming process in order to generate explainable and valid machine learning models. A collection of the methods that I have devised so far exist below:

1. **Advanced Batch Normalization:** During the training process, I want to sample inputs so that the inputs are representative of the actual data distribution that we will be seeing throughout the course of the model.

   Ontop of that, as the inputs are propogated through the course of the model, I will continually update parameters for the inputs to each of the layers so that I can understand which inputs are being increased the most and attempt to correlate that with the inputs that are being considered as the most important in the input to the model

2. **Node-Explainability Loss Functions:** Over the course of training, it is important that we coax the model towards representing more explainable and tangible concepts for the user. In order to do this, I will attempt to implement a loss function that scales the coefficients of the regularization and

sparsity of results over the course of the model progressively higher as the gradients are backpropogated closer to the front of the model. This means that in the first layer, we perform minimal categorization and pruning. But, towards the later models, we are attempting to make the number of nodes that are activated for any one prediction as small as possible while making the divergence of the resulting latent-space vector representations as high as possible.

One equation that we can use for the loss function in order to complete this task is below. It changes dependingon the layer that we are considering:

$$L(i) = L_{base} + \lambda \times e^{\alpha i} \times L_{layer-wise}$$

Where the following information describes each of the variables:

- $L(i)$ represents the total loss corresponding to the i^{th} layer.
- L_{base} stands for the model's basic training loss over all the data.
- λ is a practical factor to modify the total level of the area of the corresponding system's box.
- $e^{\alpha i}$ is an exponential factor that helps to reasonably spread the sense of the specific space, based on the map of the class at the base and the final task of the clear class.
- α is a true fact that gives the idea of the specific same part, changing with the time of the new change for the road of the school, making it a kind of mix and great way.
- i is the index of the viewed layer, making a road from the starting to the end.

This is a generalized version of a loss function that I can use such that we increase the number of nodes that are being regularized over the course of the execution of the algorithm percentage-wise. For example, we might increase the number of nodes that are being regularized and manipulated at each layer by 5% per layer until a max of 50% of the nodes have been regularized in the final layers.

This ensures that the model is training so that we get some sort of explainability in terms of inputs at the nodes. To take this one step further, I would be interested in implementing some loss function so that similar outputs light up similar nodes in the neural network later in the network. On the other hand, highly similar inputs with different labels should be forced to light up progressively more unique combinations of neurons later in the network as compared to inputs with different labels. An example of such a loss function would look something like:

$$L(i) = L_{\text{base}} + \lambda \times e^{\alpha i} \times L_{\text{layer-wise}} + \beta \times L_{\text{contrastive}}(i) \tag{1}$$

where:

- $L_{\text{contrastive}}(i)$ represents the contrastive part of the loss that operates on the latent space representations. It encourages similar inputs to activate similar nodes and dissimilar inputs produce outputs even early in the network that are markedly different from different inputs' propogated outputs. This could be implemented using something like the cosine similarity or Euclidean distance between the feature representations pairs of inputs.
- β is a weighting factor that controls the influence of the contrastive loss component. It is a hyperparameter that should be tuned in the same fashion as any other hyperparameter.

Additionally, I will compute the importance of each node by multiplying the gradient that it outputs with the importance that it has with respect to the input to get the most important nodes of the neural network.

3. **Influence-Limiting Functions:** Another way that we can ensure that the model is removing the biased parameters is by limiting the influence that any confluence of input factors or individual inputs can have. Of course, this has been done before. So, we provide a derivation which we expect to provide markedly better results for the end user.

   There are several methods for removing the influence of a node without removing it from the entire neural network or forcing retraining. This is a derivative of the re-training that I have mentioned earlier, but is substantially less efficient than actual re-training and incremental learning methods that have been proposed so far.

   With that being said, this has the potential to make the discovery of patterns and information in the machine learning space substantially more efficient. With influence-limiting functions, we can ensure that problematic inputs are not considered in the final decision of the neural network while more important ones are. This can force compliance in a way that would not have been possible before.

   A potential method is shown below, but is not discussed in serious detail due to the fresh nature of the idea and iteration to ensue. Assume we have a vector of elements that multiply with some inputs in an element-wise manner to produce the output of the neural network layer $a_1, a_2, \ldots, a_n$, and we want to modify each element a_j $(j \neq i')$ by multiplying the element by $(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k})\%$ in order to step-up the influence of these nodes, and set $a_{i'}$ to zero. This ensures that the impact of $a_{i'}$ on this layer in the neural network which we construct is zero. The operation for each a_j $(j \neq i')$ can be expressed as follows:

$$a'_j = a_j \cdot \left(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k}\right)$$

   where for $a_{i'}$, we simply set:

$$a'_{i'} = 0$$

   Combining these into a single equation to represent the transformation of the entire sequence, we get:

$$a'_j = \begin{cases} 0 & \text{if } j = i' \\ a_j \cdot \left(1 + \frac{\|a_{i'}\|}{\sum_{k=1}^{n} a_k}\right) & \text{otherwise} \end{cases}$$

   This equation completes the operation where every element except for $a_{i'}$ is increased to maintain the relative performance of the network, and $a_{i'}$ itself is turned into zero. This is the simplest version of influence-limiting where the actual weights of the elements are reduced towards zero. But, this method requires subsequent fine tuning and retraining.

Additionally, one point that I made above that has not been addressed so far is the potential introduction of un-learning schema. This is another discipline that I would like to explore as a potential backup to or supplement for the machine learning model that I work on for observable tabular data. I believe that I could implement something like this using KL-Divergence as part of a loss function to penalize the model from going too far from the original inputs while enforcing a greater, error-increasing loss on any inputs that have a mahalanobis distance close to the inputs that we are trying to erase in order to push the gradient away from optimizing for said class or inputs while maintaining model accuracy. The KL-Divergence is defined as follows:

$$D_{\mathrm{KL}}(P \parallel Q) = \int_{-\infty}^{\infty} p(x) \log\left(\frac{p(x)}{q(x)}\right) dx$$

Therefore, an effective loss function might look something like the following:

$$\text{Hybrid Loss} = \text{Loss}_{\text{General}} + \lambda \cdot 1\{\text{Dist}_{\text{Mahalanobis}} \leq \text{Closeness}_{\text{Threshold}}\} \cdot \text{Loss}_{\text{Error Increment}} + \mu \cdot D_{\text{KL}}(P \parallel Q)$$

This ensures that the inputs that are similar to the one that we are untraining on receive positive error while other inputs ensure that we are keeping the model at a somewhat steady state.

Conclusion

Given the methods above, I seek to provide further transparency on tabular-data-focused machine learning models and the thought processes driving them without compromising on accuracy. The ultimate goal is to enforce a policy of trust such that users are not over-confident in the abilities of the models while continuing to find the model outputs valid and useful in measured and sensitive environments.

Works Cited

References

[1] A. Ahrens, C. B. Hansen, and M. E. Schaffer, "lassopack: Model selection and prediction with regularized regression in Stata," *The Stata Journal: Promoting communications on statistics and Stata*, vol. 20, no. 1, pp. 95–115, 2020.

[2] F. Tung and G. Mori, "Similarity-Preserving Knowledge Distillation," in *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, 2019, pp. 1365–1374.

[3] M. Kang and S. Kang, "Data-free knowledge distillation in neural networks for regression," *Expert Syst. Appl.*, vol. 175, 2021, Art. no. 114813.

[4] Y. Yang, S. He, Y. Qiao, W. Xie, and T. Yang, "Balanced Knowledge Distillation with Contrastive Learning for Document Re-ranking," in *Proceedings of the 2023 ACM SIGIR International Conference on Theory of Information Retrieval*, 2023.

[5] F. Wang and H. Liu, "Understanding the Behaviour of Contrastive Loss," in *2021 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, 2020, pp. 2495–2504.

[6] J. Gou, B. Yu, S. J. Maybank, and D. Tao, "Knowledge Distillation: A Survey," *International Journal of Computer Vision*, vol. 129, pp. 1789–1819, 2020.

Works Referenced

References

[1] D. Baehrens, T. Schroeter, S. Harmeling, M. Kawanabe, K. Hansen, and K.-R. Müller, "How to Explain Individual Classification Decisions," *arXiv preprint arXiv:0912.1128*, 2009.

[2] Anonymous, "Learning Global Additive Explanations for Neural Nets Using Model Distillation," in *International Conference on Learning Representations*, 2019, *Under review.*

[3] G. Montavon, S. Bach, A. Binder, W. Samek, and K.-R. Müller, "Explaining NonLinear Classification Decisions with Deep Taylor Decomposition," *arXiv preprint arXiv:1512.02479*, 2015.

[4] M. T. Ribeiro, S. Singh, and C. Guestrin, "Why Should I Trust You? Explaining the Predictions of Any Classifier," *arXiv preprint arXiv:1602.04938*, 2016.

[5] M. T. Ribeiro, S. Singh, and C. Guestrin, "Nothing Else Matters: Model-Agnostic Explanations By Identifying Prediction Invariance," *arXiv preprint arXiv:1611.05817*, 2016.

[6] S. M. Lundberg and S.-I. Lee, "A Unified Approach to Interpreting Model Predictions," in *Advances in Neural Information Processing Systems 30 (NIPS 2017)*, pp. 4765–4774, 2017.

[7] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in neural information processing systems*, pp. 5998–6008, 2017.

[8] H. Liu, Q. Yin, and W. Y. Wang, "Towards Explainable NLP: A Generative Explanation Framework for Text Classification," *arXiv preprint arXiv:1811.00196*, 2019.

[9] M. Hind, D. Wei, M. Campbell, N. C. F. Codella, A. Dhurandhar, A. Mojsilović, K. N. Ramamurthy, and K. R. Varshney, "TED: Teaching AI to Explain its Decisions," in *AAAI/ACM Conference on AI, Ethics, and Society (AIES '19)*, 2019.

[10] Z. Wu, S. Pan, F. Chen, G. Long, C. Zhang, and P. S. Yu, "A Comprehensive Survey on Graph Neural Networks," *arXiv preprint arXiv:1901.00596*, 2019.

[11] M. T. Ribeiro, S. Singh, and C. Guestrin, "Anchors: High-Precision Model-Agnostic Explanations," *arXiv preprint arXiv:*, 2020.

[12] Y. Bahri, J. Kadmon, J. Pennington, S. S. Schoenholz, J. Sohl-Dickstein, and S. Ganguli, "Statistical Mechanics of Deep Learning," *Annual Review of Condensed Matter Physics*, vol. 11, pp. 501–528, 2020.

[13] J. Devlin, M.-W. Chang, K. Lee, and K. Toutanova, "BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding," *arXiv preprint arXiv:1810.04805*, 2019.

[14] J. Kim, A. Rohrbach, T. Darrell, J. Canny, and Z. Akata, "Textual Explanations for Self-Driving Vehicles," *arXiv preprint arXiv:*, 2020.

[15] A. Radford, K. Narasimhan, T. Salimans, and I. Sutskever, "Improving Language Understanding by Generative Pre-Training," *OpenAI Blog*, 2018.