Explaining Deep Learning Models using Causal Inference

Tanmayee Narendra, Anush Sankaran, Deepak Vijaykeerthy, Senthil Mani

IBM Research

{tanarend, anusank, deepakvij, sentmani}@in.ibm.com

Abstract

Although deep learning models have been successfully applied to a variety of tasks, due to the millions of parameters, they are becoming increasingly opaque and complex. In order to establish trust for their widespread commercial use, it is important to formalize a principled framework to reason over these models. In this work, we use ideas from causal inference to describe a general framework to reason over CNN models. Specifically, we build a Structural Causal Model (SCM) as an abstraction over a specific aspect of the CNN. We also formulate a method to quantitatively rank the filters of a convolution layer according to their counterfactual importance. We illustrate our approach with popular CNN architectures such as LeNet5, VGG19, and ResNet32.

Introduction

Machine learning has come a long way in the past decade with the dramatic development of Deep Neural Networks (DNNs), and cheap availability of high-end general-purpose GPUs (GPGPUs) for high-speed computation (Chen et al. 2014), (Raina, Madhavan, and Ng 2009). These advancements have considerably improved the state-of-the-art in speech recognition, computer vision, natural language processing, etc. This success comes at a cost - DNNs derive their expressive power by learning a complex higher dimensional representation of the input data. In turn, the task of understanding intricacies of the intermediate representations and feature spaces of DNNs has become notoriously difficult. In spite of their opacity, DNNs are being used to power several end-user facing applications such as image recognition, machine translation as well as sensitive applications such as autonomous navigation (Pomerleau 1991), malware detection (Dahl et al. 2013), radiology (Shin et al. 2016), etc. and they influence our day to day decisions. Due to these factors, it has become paramount that we develop a framework to understand the inner workings of these models.

The need for frameworks for interpreting and explaining the inner working of these complex models hasn't gone unnoticed. Several approaches have been proposed in the recent past to understand the predictions output by DNNs. In particular, there has been significant progress in explaining the predictions of a Convolution Neural Network (CNN)

---

in the image domain. These techniques work by generating saliency maps which indicate the relevance of pixels in the image to the output of CNN. (Selvaraju et al. 2017) (Zhou et al. 2016)

Even though the existing methods have significantly improved the interpretability of CNNs, they suffer from a critical shortcoming. These techniques don't help humans to reason how and why perturbations to the model's structure (for example removing a filter from the n-th layer of the model) can impact the model's predictions.

In this work, we borrow ideas from causal inference (Pearl 2009) to provide a general abstraction of a DNN which allows for arbitrary causal interventions and queries and in turn offers an invaluable tool to quantitatively explain the importance and impact of the constituent components of a DNN on its performance. The ability to perform arbitrary causal interventions allows us to seamlessly capture the chain of causal effects from the input, to the filters (in turn the complex intermediate representations), to the DNN outputs. In addition to the above, the proposed abstraction will also allow us to answer a richer set of questions which the existing frameworks can't such as counterfactual queries. For instance, an end user would be able to ask "What is the impact of the n-th filter on the m-th layer on the model's predictions?". Such an abstraction provides a powerful tool through which an end user can debug, test and analyze the properties of a deployed DNN.

In this paper, we focus on image-based models and study the significance of its components (for instance, filters) based on a pre-decided metric, to illustrate the utility of the proposed framework. A standard way to measure the importance of model elements is through ablation testing. For a fixed data set, this involves removing specific component(s) from the CNN model, retraining the model on the same data set and testing the performance. The difference in the model performance is an indicator of the importance of the model components under study. Although ablation testing is a simple and intuitive way of measuring the influence of model components, it is computationally expensive, and may not be feasible for complex models.

One of the key advantages of the proposed framework is that, as opposed to the above technique, our approach only requires a one-time construction of the causal abstraction of a CNN and does not require any re-training of the model to

identify a filter's influence on the model's predictions.

The following are the research contributions of this work - First, we describe a general framework to build a causal model as an abstraction to reason over a specific aspect of a DNN. Second, we provide a simple approach to validate the causal model learnt, to check if the causal model abstraction adequately captures the behaviour of the DNN model. Last, we describe a method to quantitatively rank DNN model components according to their importance. We illustrate our framework with well known image classification models such as VGG-19 (Simonyan and Zisserman 2014), ResNet-32 (He et al. 2016) and LeNet-5 (LeCun et al. 1998) with CIFAR10 (Krizhevsky 2009).

Background

In this section, we will give a brief introduction to causality. A reader well versed in causal inference may skip this section.

Causal Theory

All explanations are arguably and inherently causal. Questions of the sort *What if*, and *Why* require an understanding of the underlying mechanisms of the system and the theory of causal inference is one method by which such questions might be answered.

Interpretability methods such as saliency maps (Zhou et al. 2016) (Selvaraju et al. 2017) only establish a correlation - while it is possible to say that a particular image region is responsible for it to be correctly classified, it cannot say what would happen if a certain portion of the image was masked out (This is a *What If* question.)

In the statistical paradigm, the objective is to approximate the joint distribution from the data at hand, in order to answer relevant questions about new data. Statistical models allow us to answer questions related to prediction. A causal model, on the other hand, subsumes a statistical model, and contains additional structure that helps in answering questions about interventions and distribution changes. (Peters, Janzing, and Schölkopf 2017)

Causal Models help in answering *what if* kinds of questions. Consider the deep learning model as the system under study. Causal Models contain additional structure about the system that helps in answering questions of the kind - *For this image, what if I black out this filter response to zero? Would my prediction still be correct?* and *If I blacked out this filter completely, how much would my accuracy change?*. These kinds of what-if questions, which involve making changes to the system under study, can be answered through causal models.

Any causal model must be expressive enough to answer *prediction*, *intervention* and *counterfactual* questions, which are in increasing order of difficulty. A *prediction* question is essentially an association - it involves asking what would happen to a variable of interest, if the values of other variables were observed to be a given value. Prediction questions have been very well studied in Statistics and Machine Learning, and there are several successful methods for answering observational questions. (Pearl 2009) (Peters, Janzing, and Schölkopf 2017) An *intervention* question on the other hand, deals with answering what would happen to the variable of interest, if a subset of variables were *set* to a particular value. A *counterfactual* questions deals with answering *What If* kinds of retrospective questions. (Pearl 2009)

Causality has a long history, and there are several formalisms such as Granger causality, Causal Bayesian Networks and Structural Causal Models. (Lattimore and Ong 2018). In this work, we consider the Structural Causal Model paradigm, which allows us to specify causal mechanisms by specifying a set of equations.

Structural Causal Models (SCM) Consider a set of random variables $\mathbb{X} = \{X_1, X_2, ..., X_n\}$. The Structural Causal Model for $\mathbb{X}$ is defined as the set of assignments

$$X_i \coloneqq f_i(PA_i, N_i)$$

where f_i is a function, $PA_i \subset \mathbb{X}, (X_i \notin \mathbb{X})$ and $N = \{N_i\}$ are jointly independent. PA_i denotes the parents of X_i, and N denotes the set of noise variables. Structural Causal Models are also called as Structural Equation Models. (Peters, Janzing, and Schölkopf 2017)

Every SCM is associated with a directed graph, where each edge can be colloquially interpreted as going from cause to effect. Using the equations of the SCM, the graph G associated with the SCM is constructed as follows - construct a vertex for every $X_i \in \mathbb{X}$, and draw an edge from every vertex in PA_i into X_i. In this work, we only consider Structural Causal Models whose graph G is a Directed Acyclic Graph (DAG). (Peters, Janzing, and Schölkopf 2017)

It is important to note that these equations are to be interpreted in as assignments, and are not bi-directional.

Estimating Interventions An intervention of setting a subset of variables $X \subset \mathbb{X}$ to a particular value is usually denoted as $do(X = x)$.

The causal effect of X on Y can be estimated with $P(Y = y|do(X = x)) = \sum_z P(Y = y|X = x, PA = z)P(PA = z)$ where PA denotes the parents of X in the associated graph, and z ranges over the set of values taken by PA. (Pearl 2009)

In this work however, we do not use this formulation because the high connectedness and the large number of variables in our causal graph makes estimating the marginal probabilities computationally expensive. Instead, for a fixed dataset D, we actually perform the intervention, and then estimate the expected value of Y over D. This is explained in detail in Section 5.

Existing Work

There have been several interesting applications where building a causal model of a system can provide a principled approach to reasoning about it. In (Bottou et al. 2013), a causal model is built of a computational advertisement system. Here, the main quantity of interest is the advertisement clickthrough rate. The causal model was used to provide a principled approach to model counfounding, and a framework to inexpensively reason about the effect of interventions such as changing the ad-placement algorithm without conducting randomized trials.

Causal models have been used to isolate noise from the signals from the Kepler space telescope, and this has helped in the discovery of several previosuly undiscovered exoplanets. (Schölkopf et al. 2016)

Modelling bias and fairness as a counterfactual is a new way to quanitify the bias of models. (Kusner et al. 2017)

Ideas from causal inference have also been used in providing human understandable explanations in deep learning models. In (Harradon, Druce, and Ruttenberg 2018), a method is developed to build a causal model over human understandable image based abstractions of the model, which helps in asking counterfactual questions. In (Lu et al. 2018), a causal model is used to remove gender bias in NLP models.

Thus, causal inference has been applied to several domains with considerable success. To the best of our knowledge, this is the first work that uses causal inference to reason over DNN model components.

Proposed Approach

In this section, we explain how to build a causal model to reason over a deep learning model. To build a SCM of the DNN, first we build the DAG structure from the DNN. Second, we apply a suitable transformation, which captures the aspect of the DNN that we want to reason over. Last, we estimate the structural equations of the SCM. (See Figure 1) These steps are described in detail below.

Building the Structural Causal Model

Let the deep learning model under study be M_D, and its corresponding structural causal model be M_C. We need to estimate M_C from M_D.

DAG Structure   DNNs have an inherent DAG structure, which we exploit to construct the skeleton DAG for the Structural Causal Model (SCM) M_C. This DAG structure can be visualized by considering every node in a neural network as a vertex in a graph, and every forward connection as a directed edge. This DAG in the SCM can be constructed at the granularity level to which the DNN model is to be studied. For example, suppose we wish to study a CNN at the granularity level of its filters. The SCM DAG is constructed such that for a given filter in the ith convolution layer, there is an incoming directed edge from from every filter in the $(i-1)^{th}$ layer. On the other hand, if we wanted to study a DNN at a layer level, the SCM DAG is constructed such that there is a directed edge from the ith layer to the $(i+1)^{th}$ layer.

In this work, we will restrict ourselves to studying the CNN model at the level of its filters.

Applying a Suitable Transformation   The causal abstraction of the DNN is built to encapsulate some aspect of the underlying DNN model, over which reasoning, and *what if* questions might be answered. To this effect, an appropriate transformation function ϕ must be selected, keeping in mind the nature of intervention questions that need to be answered from the causal model. For example, if it was required to answer questions about the variance of filter responses in convolution layers, a transformation that encodes variance must

be formulated. The choice of ϕ is not restricted, and transformation functions can be arbitrarily complex.

However, there is one practical consideration - In a CNN, every filter response from a convolution layer is a matrix. We will need to convert this matrix to a real number in order to estimate the equations in the SCM. Hence, in this work, we consider only those transformation functions whose range is a real number.

Let $\phi : \mathbb{R}^{p \times q} \to \mathbb{R}$ be a suitable transformation function, that captures that aspect of the filter that we wish to model in the SCM. If $M_j^i \in$ is the jth filter response of the ith layer, $\phi(M_j^i) = r_j^i$ is its transformation to a real number. (See Figure 1)

Learning the Equations of the SCM   Now, we need to estimate the structural assignments, or structural equations of the causal model.

For a Structural Causal Model whose causal graph is a DAG, the joint distribution can also be estimated with Causal Bayesian Networks. (Pearl 2009) However, if we were to estimate the joint distribution as a set of conditional probability tables, these tables become prohibitively large owing to the high connectedness of the DAG constructed from the CNN model. To illustrate this point, consider the VGG 19 model. The last pooling layer of the third block has 256 filters, which are connected to 512 filters in the first convolutional layer of the fourth block. The size of the conditional probability table for one filter in the fourth block is $2^{256} \approx 10^{77}$, which is prohibitively large. Hence, it is not possible to estimate the joint distribution with conditional probability tables.

Instead, we learn a function f such that

$$r = f(PA_r)$$

where r is node in the causal DAG, and PA_r denotes the parents of r. The function f is arbitrary. This reduces to the standard regression problem.

Let us see how this works with the SCM constructed from a CNN. Let the dataset be D. Consider the ith convolution layer of of the CNN, and let us assume that it consists of k filters. For a given image $D_i \in D$, let the set of filter responses from i^{th} convolution layer be $M = \{M_1^i, M_2^i, ..., M_k^i\}$.

Let ϕ be the transformation function, then we have

$$\begin{bmatrix} \phi(M_1^i) \\ \phi(M_2^i) \\ ... \\ \phi(M_k^i) \end{bmatrix} = \begin{bmatrix} r_1^i \\ r_2^i \\ ... \\ r_k^i \end{bmatrix} = \vec{R}^i$$

Now, for the given layer i, we need to approximate the set of functions $F_i = \{f_1^i, f_2^i, ..., f_k^i\}$. Recollect that the causal DAG was constructed in such a way that for every filter in layer i of the CNN, its parents are the set of all filters in the layer $i-1$. Hence, for the jth filter in the ith layer of the CNN, function in f_j^i can be approximated as follows -

$$r_j^i = f_j^i(\vec{R}^{i-1})$$

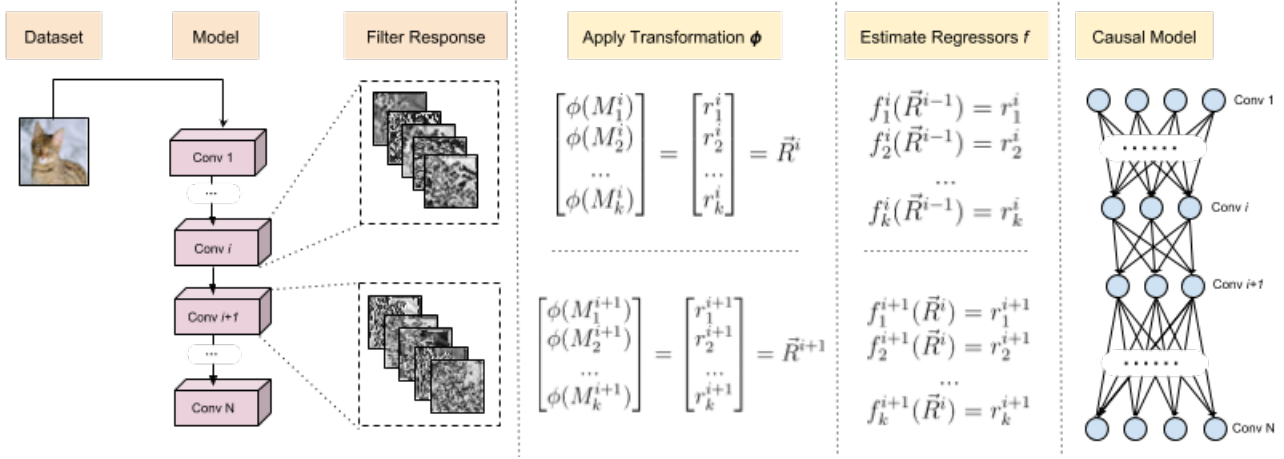The set of all functions f_j^i, together with the causal DAG gives us the Structural Causal Model.

Figure 1: Approach to build a SCM from a CNN

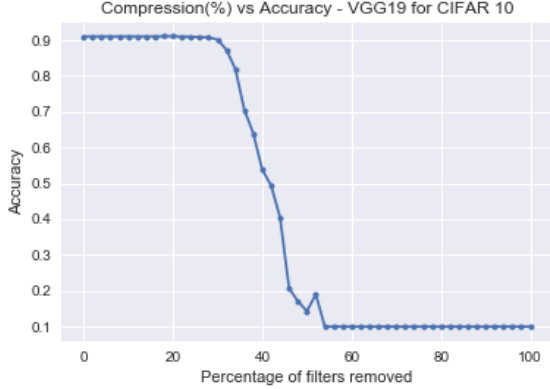| Dataset	Model	Accuracy
	LeNet 5	0.706
CIFAR10	VGG19	0.91
	ResNet 32	0.924

Table 1: Performance of trained models

Figure 2: Accuracy, when filters are removed in decreasing order of variance (VGG19 over CIFAR10)

Experimental Set Up

In this section, we describe the details of the experiments conducted and summarize the results. In this work, we consider only Convolutional Neural Networks (CNN).

Implementation Details

We apply the framework described previously to CNN models, VGG19 (Simonyan and Zisserman 2014), ResNet32 (He et al. 2016) and LeNet5 (LeCun et al. 1998), over CIFAR10 data set(Netzer et al. 2011). All models were trained using Keras with Tensorflow backend.

Table 1 shows the performance of pre-trained models.

Building the Causal Model

The causal DAG structure is inferred from the CNN model, and it remains fixed for a particular CNN model.

There is some flexibility in the choice of the transformation function ϕ and the method of estimating the structural equations F.

Transformation ϕ To build the SCM, we need to decide on an appropriate transformation ϕ , which captures some aspect of the filter that we want to reason about, and that can represent each filter of every CNN layer as a real number.

Filters with low variance have been observed to contain more information than those with higher variance (Golub, Lemieux, and Lis 2018). The simplest representation is to convert every filter into a binary number, indicating whether or not each filter has high variance or not. An SCM constructed with this transformation would allow us to reason about the importance of filters, assuming that high/ low variance is a good measure of filter importance.

To this effect, we first consider the transformation

$$\phi(M_j^i) = \begin{cases} 1 & \left\| M_j^i \right\| < \mu_j^i + \sigma_j^i \\ 0 & otherwise \end{cases}$$

where μ_j^i is the mean and σ_j^i is the variance of the jth filter response from the ith layer M_j^i over the dataset D. It is important to note that there is significant information loss with this transformation because, essentially we are converting a matrix to a bit. (If M_j^i is of dimension $p \times q$ then $\log_2(pq) \to \log_2 2$)

However, variance appears to be a good metric to reason about filter importance, as can be seen from Figure 2, 3, 4.

We consider another transformation, which takes the Frobenius norm of M_j^i.

$$\phi(M_j^i) = \left\| M_j^i \right\| = \sqrt{\sum_p \sum_q m_{pq}^2}$$

The information loss is lesser in this transformation. This transformation allows us to ask questions about the effect
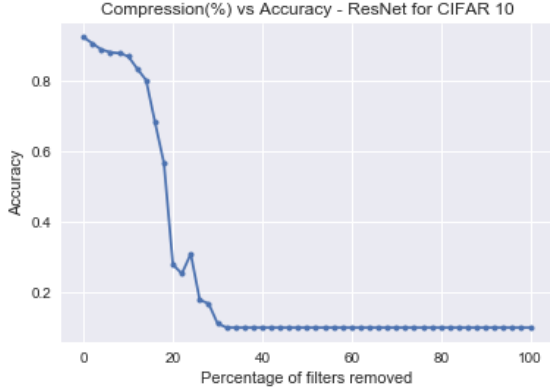
Figure 3: Accuracy, when filters are removed in decreasing order of variance (ResNet 32 over CIFAR10)
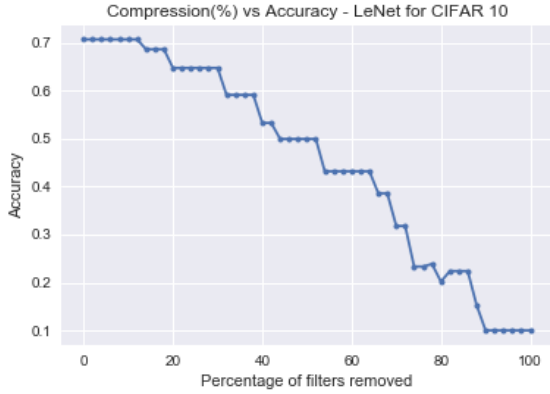

Figure 4: Accuracy, when filters are removed in decreasing order of variance (LeNet 5 over CIFAR10)

on the accuracy when intervening on a filter response and setting it to zero.

Before fitting regression models on the transformed dataset, we augment the dataset by making random interventions within the model and observing the effect. In other words, for each set of filter responses from a particular layer, we randomly zero out 10% of the filters, and apply the transformation on the rest of the filter responses in the downstream layers of the model. This is to ensure that the SCM learns from intervention data as well.

Model	Avg Accuracy	Min Accuracy
Logistic Regression	0.926	0.526
SVM	0.848	0.659
Random Forest	**0.987**	**0.722**

Table 2: Accuracy Metrics for SCM built over VGG19 (CIFAR10), using binary transformation

Model	Avg MSE	Max MSE
Linear	**0.361**	**12.9**
Ridge	0.365	13.4
LARS	73.1	2028.7

Table 3: Accuracy Metrics for SCM built over VGG19 32 (CIFAR10), Frobenius norm transformation

Model	Avg MSE	Max MSE
Linear	3.44	307.3
Ridge	**3.43**	298.3
LASSO	3.56	**296.4**

Table 4: Accuracy Metrics for SCM built over ResNet 32 (CIFAR10), Frobenius norm transformation

Learning the Structural Equations Now, we estimate a set of regressions F, which gives us the SCM. For the binary transformation function, we used popular classification algorithms like Logistic Regression and Random Forest to learn the structural equations. See Table 2 for more details. The average accuracy metric is 98% and is satisfactory.

For the Frobenius norm transformation, we used linear models to estimate the Structural Equations. Table 3, Table 4 and Table 5 shows the average and maximum MSE for VGG-19, ResNet 32 and LeNet 5 architectures.

Evaluation

In this section, we describe a method by which we can check if the causal model built previously is representative enough of the CNN and adequately captures the information of each filter. This is done primarily to ensure that the transformation chosen adequately captures the behaviour of the CNN model.

Sanity Check of the Causal Model

In order to check whether ϕ is an appropriate transformation, and whether the set of functions in the SCM $F = \{f_i^j\}$ adequately captures the model's performance, we run the dataset D through F, and check if it gives a *reasonable* accuracy. If the transformation function ϕ that was selected previously is not representative enough of the CNN, then the accuracy will be unsatisfactory.

Inference on the Causal Model

Inference using the causal model can be done at two levels - we can answer questions relating to a particular sample from the dataset, and we can answer questions about the entire dataset.

Model	Avg MSE	Max MSE
Linear	22.2	**375.9**
Ridge	**22.1**	376.9
LASSO	22.2	381.9

Table 5: Accuracy Metrics for SCM built over LeNet5 (CIFAR10), Frobenius norm transformation

Model	SCM Accuracy	Model Accuracy
VGG 19	0.902	0.91
LeNet 5	0.830	0.706
ResNet 32	0.727	0.924

Table 6: SCM Accuracy for different models learnt for CIFAR10

Let the sample be $D_i \in D$, let the subset of filters that we want to intervene on be denoted by X, and their values by x. The intervention of setting $X = x$ is denoted by $do(X = x)$. Let the filter whose behaviour we want to observe after the intervention be denoted by Y. If we were interested in measuring the impact of an intervention on the model's performance, Y would be chosen to be a filter from the last dense, classification layer of the CNN. It is important that $Y \notin X$.

To estimate the effect of an intervention on Y, in the Structural Equations of the SCM, we *set* $X = x$ and estimate the value of Y. (Pearl and others 2009)

To estimate the effect of an intervention on the dataset, we do the following -

$$\mathbb{E}[Y|do(X = x)] = \frac{\sum_{D, M_C^{do(X=x)}} Y_i}{|D|}$$

In other words, for every sample $D_i \in D$, we estimate the value of Y and take its average.

Results

Based on the evaluation methodology, we present our results in this section.

Sanity Check of the Causal Model

Several useful questions that we want to answer about the deep learning model involve estimating the effect of an intervention on the model's performance. In order to check whether this is captured *well enough* by the SCM, we measure the performance of the model over the dataset D when just the SCM is used for prediction.

In other words, every layer i with k filters in the CNN model is now associated with a set of Structural Equations $F^i = \{f_1^i, f_2^i, ..., f_k^i\}$ which we estimated by fitting regressions. Similar to a forward propagation through the CNN model, we pass the dataset D through the set of Structural Equations and check if the accuracy is satisfactory.

Table 6 shows the accuracy of the SCM over CIFAR10 dataset for VGG19, LeNet 5 and ResNet 32 models. This accuracy is reasonable and satisfactory.

We do not report accuracy metrics for the binary transformation, as the SCM model accuracy achieved is less than random choice. We hypothesize that although the average linear model accuracy is satisfactory, the information loss in the binary transformation is too large to give a satisfactory abstraction of the deep learning model.

Inference on the Causal Model

There are several kinds of intervention and counterfactual questions that might be answered from a well specified

Layer	Least Important	Most Important
Conv2D 1	8, 55, 10	15, 11, 20
Conv 2D 3	8, 39, 93	56, 116, 1
Conv 2D 5	226, 76, 34	211, 13, 88
Conv 2D 6	152, 164, 205	71, 233, 175
Conv 2D 7	84, 73, 229	177, 240, 16
Conv 2D 2	48, 4, 27	23, 1, 49
Conv 2D 4	117, 115, 79	50, 22, 39
Conv 2D 8	172, 102, 46	81, 2, 64
Conv 2D 9	101, 317, 441	309, 162, 373

Table 7: Most Important and Least Important Filters for VGG 19

Structural Causal Model. However, in this preliminary work, we restrict ourselves to answering questions related to the importance of a feature for correct classification.

We only consider the Frobenius norm transformation here, as the binary transformation fails to provide a satisfactory abstraction of the model. The intervention we seek to answer is as follows - what would be the effect on CNN model accuracy, if we set the Frobenius norm of a particular filter response to zero? This is the same as asking the effect on the model's performance, when one of the filters is set to zero. (It can be easily verified that the Frobenius norm of a matrix is zero, only when the individual elements of the matrix are also zero.)

We use the accuracy drop of the model to provide an estimate of the most important filters in the VGG 19 model. Table 7 shows the least and most important filters for different convolution layers. The importance of filters of a particular layer are ranked in increasing order of the SCM accuracy, that is, a filter is deemed to be less important if an intervention of setting it to zero, gives a lesser drop in SCM accuracy.

Discussion

The framework described in this work is generic, and has the potential to be used for a wide variety of applications.

1. **Model Compression**: For compressing deep learning models, typically certain nodes or layers are removed and the model is retrained to verify lossy nature of the model compression. However, using the SCM the model's performance post compression could be predicted without the need for retraining.

2. **Transfer Learning of Models:** It is a common practice to consider a model pre-trained using a large scale dataset such as ImageNet and then finetune the weights for the target task in hand. Using the learnt SCM model, we could potentially predict the performance of the original model on the target task, without the need for finetuning.

3. **Model Accuracy Prediction:** For different datasets and tasks, typically an extensive hyperparameter search is performed. For every combination of hyperparemeter the deep learning model has to be retrained to study the effect on performance. However, using the learnt SCM model,

the accuracy of a modified hyperparamter configuration could be predicted without the need for retraining.

Conclusion and Future Work

In this preliminary work, we provide a general framework for understanding a deep learning model by building an abstraction of a specific aspect of the DNN model using causal inference. In specific, we describe how a Structural Causal Model can be built, verified and used for inference. We illustrate the effectiveness of this method by using it to provide a ranking of filters of different layers in a CNN model.

There are several avenues for improving the strength of the causal models learned. Instead of approximating the functions of the Structural Equations with linear models, it might be worthwhile to explore of using a model class that has more expressive power (capacity) than linear models, which might allow us to answer complex queries. A more representative causal model would potentially allow one to answer complex intervention and counterfactual questions. Also, in this work, we have considered two basic transformation methods to build the causal model. There are potentially infinite possibilities of transformations to consider, which may prove to be more representative of the deep learning model. Lastly, the current formulation largely assumes that the dataset is fixed. This framework can be potentially extended to generalize over datasets.

References

[Bottou et al. 2013] Bottou, L.; Peters, J.; Quiñonero-Candela, J.; Charles, D. X.; Chickering, D. M.; Portugaly, E.; Ray, D.; Simard, P.; and Snelson, E. 2013. Counterfactual reasoning and learning systems: The example of computational advertising. *The Journal of Machine Learning Research* 14(1):3207–3260.

[Chen et al. 2014] Chen, Z.; Wang, J.; He, H.; and Huang, X. 2014. A fast deep learning system using gpu. In *Circuits and Systems (ISCAS), 2014 IEEE International Symposium on*, 1552–1555. IEEE.

[Dahl et al. 2013] Dahl, G. E.; Stokes, J. W.; Deng, L.; and Yu, D. 2013. Large-scale malware classification using random projections and neural networks. In *ICASSP*, 3422–3426. IEEE.

[Golub, Lemieux, and Lis 2018] Golub, M.; Lemieux, G.; and Lis, M. 2018. Dropback: Continuous pruning during training. *arXiv preprint arXiv:1806.06949*.

[Harradon, Druce, and Ruttenberg 2018] Harradon, M.; Druce, J.; and Ruttenberg, B. 2018. Causal learning and explanation of deep neural networks via autoencoded activations. *arXiv preprint arXiv:1802.00541*.

[He et al. 2016] He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

[Krizhevsky 2009] Krizhevsky, A. 2009. Learning multiple layers of features from tiny images. Technical report.

[Kusner et al. 2017] Kusner, M. J.; Loftus, J.; Russell, C.; and Silva, R. 2017. Counterfactual fairness. In *Advances in Neural Information Processing Systems*, 4066–4076.

[Lattimore and Ong 2018] Lattimore, F., and Ong, C. S. 2018. A primer on causal analysis. *arXiv preprint arXiv:1806.01488*.

[LeCun et al. 1998] LeCun, Y.; Bottou, L.; Bengio, Y.; and Haffner, P. 1998. Gradient-based learning applied to document recognition. *Proceedings of the IEEE* 86(11):2278–2324.

[Lu et al. 2018] Lu, K.; Mardziel, P.; Wu, F.; Amancharla, P.; and Datta, A. 2018. Gender bias in neural natural language processing. *arXiv preprint arXiv:1807.11714*.

[Netzer et al. 2011] Netzer, Y.; Wang, T.; Coates, A.; Bissacco, A.; Wu, B.; and Ng, A. Y. 2011. Reading digits in natural images with unsupervised feature learning.

[Pearl and others 2009] Pearl, J., et al. 2009. Causal inference in statistics: An overview. *Statistics surveys* 3:96–146.

[Pearl 2009] Pearl, J. 2009. *Causality*. Cambridge university press.

[Peters, Janzing, and Schölkopf 2017] Peters, J.; Janzing, D.; and Schölkopf, B. 2017. *Elements of causal inference: foundations and learning algorithms*. MIT press.

[Pomerleau 1991] Pomerleau, D. A. 1991. Efficient training of artificial neural networks for autonomous navigation. *Neural Computation* 3:97.

[Raina, Madhavan, and Ng 2009] Raina, R.; Madhavan, A.; and Ng, A. Y. 2009. Large-scale deep unsupervised learning using graphics processors. In *Proceedings of the 26th annual international conference on machine learning*, 873–880. ACM.

[Schölkopf et al. 2016] Schölkopf, B.; Hogg, D. W.; Wang, D.; Foreman-Mackey, D.; Janzing, D.; Simon-Gabriel, C.-J.; and Peters, J. 2016. Modeling confounding by half-sibling regression. *Proceedings of the National Academy of Sciences* 113(27):7391–7398.

[Selvaraju et al. 2017] Selvaraju, R. R.; Cogswell, M.; Das, A.; Vedantam, R.; Parikh, D.; and Batra, D. 2017. Grad-cam: Visual explanations from deep networks via gradient-based localization. In *ICCV*, 618–626.

[Shin et al. 2016] Shin, H.; Roth, H. R.; Gao, M.; Lu, L.; Xu, Z.; Nogues, I.; Yao, J.; Mollura, D. J.; and Summers, R. M. 2016. Deep convolutional neural networks for computer-aided detection: CNN architectures, dataset characteristics and transfer learning. *IEEE Trans. Med. Imaging* 35(5):1285–1298.

[Simonyan and Zisserman 2014] Simonyan, K., and Zisserman, A. 2014. Very deep convolutional networks for large-scale image recognition. *CoRR* abs/1409.1556.

[Zhou et al. 2016] Zhou, B.; Khosla, A.; Lapedriza, A.; Oliva, A.; and Torralba, A. 2016. Learning deep features for discriminative localization. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2921–2929.