# Deep Network Explanation Methods

Anirudh Seth
aniset@kth.se

September 2020

## 1 Introduction

Deep Neural Networks have achieved state of the art performance in several domains like natural language processing [15], computer vision [8] , weather forecasting [4] etc. The high accuracy is often achieved by models which are quite complex and have millions of trainable parameters.This often comes at the cost of interpretability of the model.With the increased use of such networks in critical domains like law enforcement [2],medical diagnosis [12] ,autonomous vehicles [9] etc, there is a inherent need to understand how the various aspects of training data drive the decisions of the NN.The explainability of a model can instill trust in its predictions and also provide useful insights for further improvement.

This essay summaries three different approaches to understand the predictions made by a neural network. Koh et al.[5] rely on influence functions , a classic technique from robust statistics , Lundberg et al.[6] propose a unified framework called SHAP based on Shapley values, a concept from cooperative game theory and finally Selvaraju et al [11] introduce Grad-CAM that relies on gradients to generate a heat map indicating regions of importance of an image.

## 2 Methods

### 2.1 Understanding Black-box Predictions via Influence Functions [5]

One way to study the influence of a training example is to retrain the model without it ,but this approach is practically infeasible.The authors use influence functions to trace model's predictions through the learning algorithm back to the training data.The goal is to identify training points that are most responsible for the given prediction.

The proposed approach is to study the change in prediction if an additional weight $\epsilon$ is put on a training point $z = (x, y)$. Given n training points: $z_1, \ldots, z_n$ and loss: $L\left(z_i, \theta\right)$ , the empirical risk minimization involves finding parameter that minimizes the empirical risk i.e $\hat{\theta} \triangleq \arg\min_{\theta \in \Theta} \frac{1}{n} \sum_{i=1}^{n} L\left(z_i, \theta\right)$.The influence function as derived by Cook et al [3] is:

$$\mathcal{I}_{\text{upweight, params }}(z) \triangleq \left.\frac{d\hat{\theta}_{\epsilon, z}}{d\epsilon}\right|_{\epsilon=0} = -H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}) \tag{1}$$

where $\hat{\theta}_{\epsilon, z}$ are the parameters after upweighting a sample z and $H_{\hat{\theta}} = \frac{1}{n} \sum_{i=1}^{n} \nabla_\theta^2 L\left(z_i, \hat{\theta}\right)$ is the Hessian of the empirical risk. The authors in the paper mathematically derive , the effect of training point z on the loss of test point $z_{test}$ as

$$\mathcal{I}_{\text{upweight, loss }}\left(z, z_{\text{test }}\right) \triangleq \left.\frac{dL\left(z_{\text{test }}, \hat{\theta}_{\epsilon, z}\right)}{d\epsilon}\right|_{\epsilon=0} = -\nabla_\theta L\left(z_{\text{test }}, \hat{\theta}\right)^{\top} H_{\hat{\theta}}^{-1} \nabla_\theta L(z, \hat{\theta}) \tag{2}$$

For the analogous problem of perturbing a training input $z = (x, y)$, the authors define $z_\delta \stackrel{\text{def}}{=} (x + \delta, y)$ ,$\hat{\theta}_{z_\delta, -z}$ as the empirical risk minimizer on the training points $z_\delta$ in place of z.The influence function can be evaluated as

$$\mathcal{I}_{\text{perturb, loss }}\left(z, z_{\text{test }}\right) \triangleq \left.\nabla_\delta L\left(z_{\text{test }}, \hat{\theta}_{z_\delta, -z}\right)\right|_{\delta=0} = -\nabla_\theta L\left(z_{\text{test }}, \hat{\theta}\right)^{\top} H_{\hat{\theta}}^{-1} \nabla_x \nabla_\theta L(z, \hat{\theta}) \tag{3}$$

This tells us how the loss at the test point changes with the change in the value of z. The above expressions can be evaluated efficiently using some optimization techniques like conjugate-gradient methods [7] or stochastic Hessian inversion [1]. The authors present a computationally tractable method of evaluating the effect of training set perturbations on the model predictions, without the need of training the model again. The derivations for the influence function make the following assumptions :

- Loss is convex with respect to the model parameters
- Loss is twice differentiable with respect to the model parameters

The authors empirically show that influence functions can still show good results when the above assumptions are relaxed.

## 2.2 A Unified Approach to Interpreting Model Predictions [6]

The author introduce SHAP (SHapley Additive exPlanations) which belongs to the class of models called "additive feature attribution methods". Such methods have an explanational model $g\left(z'\right)$ of the original model $f(x)$ that can be expressed as a linear function of binary variables. Each feature $x_i$ is replaced by a binary variable $z_i'$ indicating if $x_i$ is present or not.

$$g\left(z'\right) = \phi_0 + \sum_{i=1}^{M} \phi_i z_i' \tag{4}$$

where $z' \in \{0,1\}^N, N$ is the number of simplified input features, and $\phi_i \in \mathbb{R}$x‘ and tells how much the presence of feature $i$ contributes to the final output, which helps in the interpretation of the original model. The intuition in SHAP is to compute $\phi_i$ from the Shapley value in game theory. These values calculate the importance of a feature by comparing what a model predicts with and without the feature.The shapley value of a feature i given a prediction p (from the origninal mode f(x)) can be computed as

$$\phi_i(p) = \sum_{S \subseteq N/i} \frac{|S|!(n-|S|-1)!}{n!} \left[f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right) - f_S\left(x_S\right)\right] \tag{5}$$

where $S$ is the subset of features from all features $N$ except for feature $i$, $\frac{|S|!(n-|S|-1)!}{n!}$ is a weighting factor that counts the number of permutations of the subset S , $f_S\left(x_S\right)$ is the expected output given the features subset $S$ that is similar to the marginal average on all other and $f_{S\cup\{i\}}\left(x_{S\cup\{i\}}\right)$ represents the difference made by feature $i$.

The proposed method has the desirable properties of 1.)Local Accuracy,2.)Missingness and 3.)Consistency. The exact computations in SHAP are challenging (especially the expectation term as features -N increase), the author proposes alternative model agnostic approximations which are faster

1. Kernel SHAP (Linear LIME [10] + Shapley values)
2. Deep SHAP (DeepLIFT [13] + Shapley values)
3. Linear SHAP
4. Low-Order SHAP
5. Max SHAP

The above methods use a sampling approximation of the classic Shapley value equations (Equation 5) yet obtain similar approximation accuracy.

## 2.3 Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization [11]

The authors propose an approach - Gradient-weighted Class Activation Mapping (Grad-CAM) for producing visual explanations of a CNN classification model.The method uses the gradients of any target class flowing to the last convolution layer to generate a coarse localization map highlighting the regions of high importance in an image. (regions that caused the NN to make the prediction) The method can be broken to the following three steps.

1. For any class $c$, we first compute the gradient of the score for class $c, y^c$ with respect to feature map activation's $A^k$ of the last convolutional layer (in theory any conv layer can be used).

$$\alpha_k^c = \overbrace{\frac{1}{Z}\sum_i\sum_j}^{\text{global average pooling}} \underbrace{\frac{\partial y^c}{\partial A_{ij}^k}}_{\text{gradients via backprop}} \tag{6}$$

2. The gradients are then global-average-pooled over the width and height dimensions (indexed by $i$ and $j$ respectively) to obtain the neuron importance weights $\alpha_k^c$.
3. The next step is to perform a weighted combination of forward activation maps.
4. Finally, a ReLU activation is applied to only include the features that have a positive influence on the class of interest. This results in a coarse heatmap which can be overlayed on the orignal image to highlight the regions of high importance.

$$L^c_{\text{Grad-CAM}} = \text{ReLU}\underbrace{\left(\sum_k \alpha_k^c A^k\right)}_{\text{linear combination}} \tag{7}$$

**Guided GradCAM**
Grad-CAM lacks the ability to show the fine grained importance details in the image. Visualization methods like Guided Backpropogation [14] capture the pixel space-gradient and hence capture finer details. The authors combine both these methods to get richer visualizations.

1. $L^c_{\text{Grad-CAM}}$ is upsampled to the resolution of the input image.
2. Element wise multiplication of both results after upsampling as specified above to obtain $L^c_{\text{Guided Grad-CAM}}$.

## 3 Comparison

The authors in [5] demonstrate how influence function are a much more general method of discovering influencing training examples in comparison to euclidean distance measure by implementing them on two different architectures a) an SVM with an RBF kernel b) Inception v3. Although the derivation assumes the loss to be convex and twice differentiable , the authors empirically show how the functions can be approximated when these assumption are violated. The results are corroborated by comparing accuracy of 1.)complete retraining of a network using hingeloss (which is not differentiable ) with 2.)influence function with smooth hingeloss. The influence function matched the actual change with a Pearson's R = 0.95. Hence this method can also be applied to a wide variety of networks and still produce good results.SHAP[6]has the advantage of being easily adapted to work with variety of networks - tree networks, deep neural network etc. The methods can also be incorporated with existing techniques like Lime and DeepLIFT for better results .The approximation for shapley value computation are straightforward. Grad-CAM's [11] performance in terms of localization ability, class-discriminativeness, trustworthiness and faithfulness is compared with Guided BackPropagation [14] , CAM [17] and c_MWP[16] on AlexNET, VGG16, GoogleNET on the imagenet dataset. The method achieves better classification and localization error % in many cases and has the additional advantage of being simple and its applicability on variety of CNN model-families. Grad-CAM is fast and easier in terms of implementation in comparison to SHAP[6] and Influence function [5] requiring no approximations,retraining or any change to the model architecture. The biggest drawback of Grad-CAM is its restriction to the family of Convolution neural networks.This method can only be applied to specific architectures.

Some advantages of Influence functions[5] are their ability to craft adversarial training images ,helping understand the bias in the dataset by studying the negative influence , handling domain mismatch and mislabelled training data identification. The authors in [6] compare the performance of SHAP with LIME, DeepLIFT on a user study. From the results measured in terms of 1.)sickness score and 2.) max allocation SHAP values were more consistent with human intuition in comparison to other models. Grad-CAM on the other hand has the advantage of visually highlighting the areas of image that played the major role in its prediction. This has the unique advantage of interpretability using heat map which is quite visually pleasing.Grad-CAM however cannot be used to generate adversarial samples or to study the effect of different features on predictions .

Influence functions rely heavily on approximations. The quality of the approximations play a significant role in the performance. There is no clear definition of how many samples are influential for eg-top 3 or top 10. The functions only take into account the up weighting/perturbation of a single training point at an instance and not several training points at once.The proposed modifications to SHAP like kernel SHAP etc can compute the estimates with fewer evaluations. However, the Shapley values still need to be computed for a lot of points N , making the approach quite slow .Also,kernel SHAP makes the assumption of features being independent which could result in poor results especially if features are correlated . Grad-CAM is unable to localize multiple occurrences of an object in an image.The algorithm involves frequent up sampling and down-sampling which could also results in loss of information. Some results from the original paper also highlight the methods inaccurate localisation of heatmap with reference to coverage of class region.

# References

[1] Naman Agarwal, Brian Bullins, and Elad Hazan. *Second-Order Stochastic Optimization for Machine Learning in Linear Time.* 2016. arXiv: `1602.03943 [stat.ML]`.

[2] Soon Chun et al. "Crime Prediction Model using Deep Neural Networks". In: June 2019, pp. 512–514. ISBN: 978-1-4503-7204-6. DOI: `10.1145/3325112.3328221`.

[3] Ralph Dennis (viaf)46836293 Cook and Sanford (viaf)73928000 Weisberg. *Residuals and influence in regression.* eng. New York (N.Y.) : Chapman and Hall, 1982. ISBN: 041224280X. URL: `http://lib.ugent.be/catalog/rug01:000065026`.

[4] D. N. Fente and D. Kumar Singh. "Weather Forecasting Using Artificial Neural Network". In: *2018 Second International Conference on Inventive Communication and Computational Technologies (ICICCT).* 2018, pp. 1757–1761.

[5] Pang Wei Koh and Percy Liang. *Understanding Black-box Predictions via Influence Functions.* 2017. arXiv: `1703.04730 [stat.ML]`.

[6] Scott Lundberg and Su-In Lee. *A Unified Approach to Interpreting Model Predictions.* 2017. arXiv: `1705.07874 [cs.AI]`.

[7] James Martens. "Deep Learning via Hessian-Free Optimization". In: *Proceedings of the 27th International Conference on International Conference on Machine Learning.* ICML'10. Haifa, Israel: Omnipress, 2010, pp. 735–742. ISBN: 9781605589077.

[8] E. Nishani and B. Çiço. "Computer vision approaches based on deep learning and neural networks: Deep neural networks for video analysis of human pose estimation". In: *2017 6th Mediterranean Conference on Embedded Computing (MECO).* 2017, pp. 1–4.

[9] Molly O'Brien et al. *Dependable Neural Networks for Safety Critical Tasks.* 2019. arXiv: `1912.09902 [cs.LG]`.

[10] Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. ""Why Should I Trust You?": Explaining the Predictions of Any Classifier". In: *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining.* KDD '16. San Francisco, California, USA: Association for Computing Machinery, 2016, pp. 1135–1144. ISBN: 9781450342322. DOI: `10.1145/2939672.2939778`. URL: `https://doi.org/10.1145/2939672.2939778`.

[11] Ramprasaath R. Selvaraju et al. "Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization". In: *International Journal of Computer Vision* 128.2 (Oct. 2019), pp. 336–359. ISSN: 1573-1405. DOI: `10.1007/s11263-019-01228-7`. URL: `http://dx.doi.org/10.1007/s11263-019-01228-7`.

[12] Qeethara Al-Shayea. "Artificial Neural Networks in Medical Diagnosis". In: *Int J Comput Sci Issues* 8 (Feb. 2011), pp. 150–154.

[13] Avanti Shrikumar, Peyton Greenside, and Anshul Kundaje. *Learning Important Features Through Propagating Activation Differences.* 2017. arXiv: `1704.02685 [cs.CV]`.

[14] Jost Tobias Springenberg et al. *Striving for Simplicity: The All Convolutional Net.* 2014. arXiv: `1412.6806 [cs.LG]`.

[15] Tom Young et al. *Recent Trends in Deep Learning Based Natural Language Processing.* 2017. arXiv: `1708.02709 [cs.CL]`.

[16] Jianming Zhang et al. *Top-down Neural Attention by Excitation Backprop.* 2016. arXiv: `1608.00507 [cs.CV]`.

[17] B. Zhou et al. "Learning Deep Features for Discriminative Localization". In: *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR).* 2016, pp. 2921–2929.