

# Deep Learning: Confidence Through Interpretation

**Project Report:** CS230: Deep Learning

**Team member:** Bhaskar Chattaraj ([bchattar@stanford.edu](mailto:bchattar@stanford.edu))

**Introduction:** Deep Learning has been incredibly successful in recent times especially in tasks that involve images and texts such as image classification and language translation. Neural Networks uses many layers of multiplication with learned weights and through non-linear activations. This gives it the power to discover any complicated relationship (universal approximation theorems) and learn features.

It is this same power that makes a deep learning framework a black box. End users want to understand why the model is working and use those to change features to get better results. Lack of understanding can also translate to lack of trust in the deep learning models.

We would like to know the following:

- Features that the Neural Net has learned
- Contribution of each input towards a particular prediction
- Ability to approximate certain input-output domain of the Neural Net with a simpler easily interpretable model

**Methodologies:** Surrogate Models are simpler interpretable models that are trained to approximate the predictions of a Neural Net. We will mostly be working with local surrogate models in this project. Local methods are designed to explain prediction around a neighborhood of specified inputs.

On searching for methodologies, the following two seem promising and simple:

## **LIME: Local Interpretable Model-agnostic Explanations**

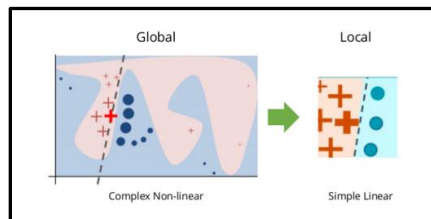
The key intuition behind LIME is that it is much easier to approximate a black-box model by a simple model *locally* (in the neighborhood of the prediction we want to explain). Instead of training a global surrogate model, LIME focuses on training local surrogate models to explain individual predictions.

Mathematically, locally surrogate model can be expressed as:

$$\text{Explanation}(x) = \text{ARGMIN}_{g \in G} \{L(f, g, \pi_x) + \Omega(g)\}$$

The explanation model for instance  $x$  is the model  $g$  (e.g. regression model) that minimizes loss  $L$  (e.g. mean squared error), which measures how close the explanation is to the prediction of the original model  $f$  (e.g. a Neural Net model), while the model complexity  $\Omega(g)$  is kept low.  $G$  is the family of possible explanations, for example all possible generalized linear models. The proximity measure  $\pi_x$  defines how large the neighborhood around instance  $x$  is that we consider for the explanation. In practice, LIME only optimizes the loss part, and the user has to decide on the model complexity class. One of the disadvantages of the surrogate models is that we observe **conclusions about the model and not about the data** since the surrogate model never sees the actual data or predictions.

When LIME wants to explain a prediction at data point  $X$ , it takes the following steps (see figure below):



1. Sample points around  $X$
2. Use complex model to calculate predictions for the sample
3. Sample weights are assigned proportional to the distance from  $X$

4. Fit new model with weighted samples
5. Use simple model to explain

**SHAP (SHapley Additive exPlanations)**

SHAP is a game theoretic approach to explain the output of any machine learning model. It connects optimal credit allocation with local explanations using the classic Shapley values from game theory and their related extensions.

SHAP goal is to explain the prediction of a given instance x by computing the contribution of each feature to the prediction. The feature values of a data instance act as players in a coalitional game theory. SHAP prediction output is a fair distribution of all the feature Shapley values. Shapely value is a distribution, it is an average of model contribution made by each player(features) over all permutation of player(features). The baseline for Shapley values is the average of all predictions.

The next steps are to explain Neural Net Binary Classification predictions using LIME and SHAP on the following data scenarios:

1. Synthetic Tabular Data
2. Tabular Data
3. Image Data
4. Text Data

**Synthetic Tabular Data:** This case is considered to give us some confidence in the interpretation we are making using LIME and SHAP

We considered the following simple binary classification model.

Output:  $Y = \text{Indicator} [2 * X1^2 - X2^2 + \epsilon \geq 1]$

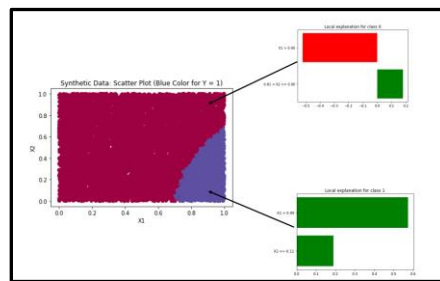
Following is the data generation process for 10K samples (9K training and 1K test):

- $X1, X2 \sim \text{Uniform}(0,1)$
- $\epsilon \sim \text{Normal}(0, 0.01)$

Models Fit with Y as Output and X1, X2 as Inputs:

- Neural Net with 3 hidden layers (20,20,20), ReLU activation and ADAM solver
- Training and Test Accuracy of 99.6% (with 70-30 split)

**LIME Analysis:**



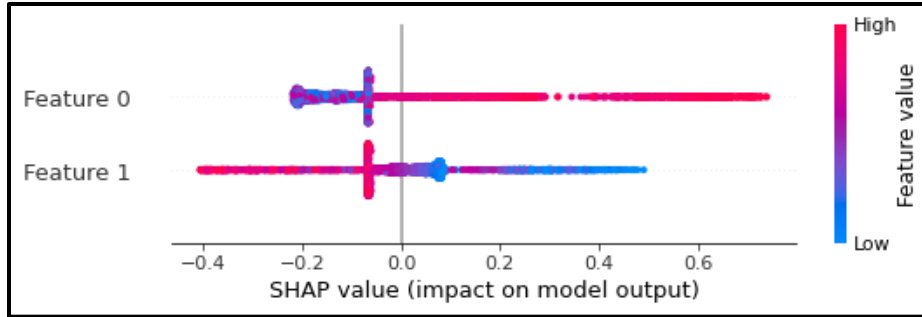
Above are results of a LIME local results around two points:

1. (0.92, 0.85): The prediction is class 0. X1 (red bar) has a negative probability contribution towards result and X2 positive.
2. (0.95, 0.05): Both variables have a positive contribution with X1 effect dominating towards a prediction of 1.

In both cases it makes sense given the structure of the data generation process.

**SHAP Analysis:**

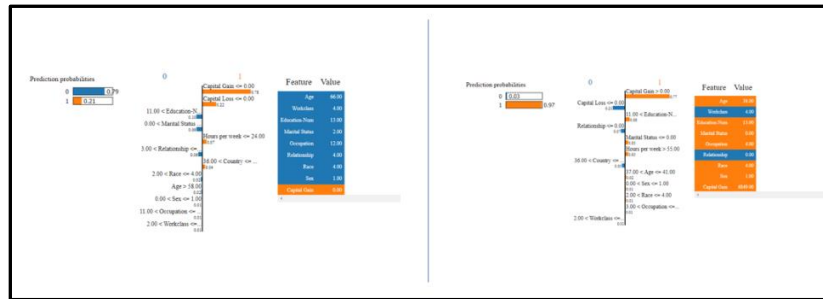
The figure below gives the SHAP plot for the variables (X1 is Feature 0 and X2 Feature 1). As it indicates higher value of X1 has a positive impact on a prediction of 1 and it is opposite for high values of X2. Another aspect that is explained is that the effect of X1 is more than X2 towards a prediction of 1 (rightly so given the data generation process).



**Tabular Data:** The data used is a census income data (<https://archive.ics.uci.edu/ml/datasets/census+income>) with 32,561 data points with the following salient points:

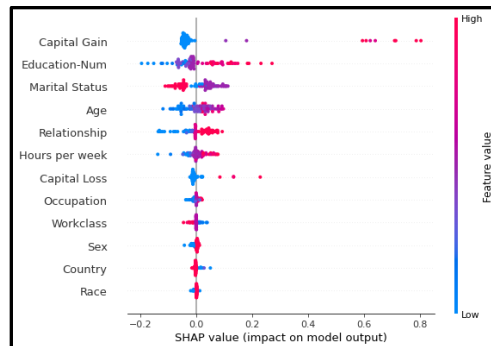
- 12 explanatory variables such as Age, Working Class, Education, Capital Gains, etc.
- Prediction task is to determine whether a person makes over 50K a year (>50K is 1 and 0 otherwise)
- Neural Net with 3 hidden layers (50,40,30), ReLU activation and ADAM solver
- Training Accuracy of 84.5% and Test Accuracy of 83.6% (with 80-20 split)

**LIME Analysis:**



The output above has a data point with prediction of 0 (Income <=50K) on the left and with a one on the right. As we see factors like higher Capital Gain, more Education and more Hours per week drive the results towards higher income class.

**SHAP Analysis:**



We used the SHAP Kernel Explainer above using 1K random points from training set and the above graph is based on 100 test data points. Kernel SHAP is a model agnostic method to approximate SHAP values using ideas from LIME and Shapley values. We see that Capital Gain, Higher Education and Marital Status drive the higher income class prediction.

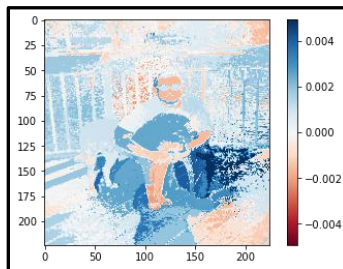
## Image Data:



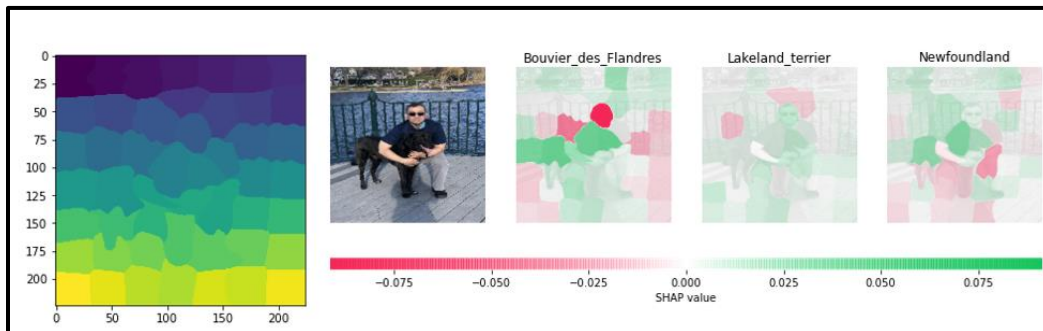
For this exercise we used VGG16, which is a convolutional neural network model proposed by K. Simonyan and A. Zisserman from the University of Oxford in the paper "Very Deep Convolutional Networks for Large-Scale Image Recognition". The picture we used as the input is one of me with my dog (mix of Lab, Shepherd and another breed) shown above.

The top 3 predictions from VGG16 is Bouvier de Flandres, Lakeland Terrier and New Foundland. If we look at the pictures the explanations are quite good (Bouvier de Flandres is a black furry dog breed similar in size to a Lab).

**LIME Analysis:** Superpixels are generated using a quickshift segmentation algorithm. Superpixels are interconnected pixels with similar colors and can be turned off by replacing each pixel with a user-defined color such as gray. For the case of image explanations in Lime, perturbations are generated by turning off and on some of the superpixels in the image. The picture below shows the heatmap for the first prediction. The dark blue ones (head and back of the dog) drives the results, whereas the hand around the dog does not! Appendix A has another analysis with the same picture using Google's Inceptionv3.



**SHAP Analysis:** In SHAP we segment the image into slices (50 slices for our image – see leftmost picture in the figure below). Here the SHAP algorithm removes a slice from the picture and gets the effect on probability for the top 3 picks. In the final results we see the importance of the effect on these probabilities (SHAP value) in the 3 pictures on the right of the figure below. The green areas (higher SHAP values) generally cover the head and back of my dog but ignores me (especially my hand) around it.



## Text Data:

We use the Sci-kit learn 20 newsgroups text dataset ([https://scikit-learn.org/0.19/datasets/twenty\\_newsgroups.html](https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html)). The 20 newsgroups dataset comprises around 18000 newsgroups posts on 20 topics split in two subsets: one for training (or development) and the other one for testing (or for performance evaluation). The split between the train and test set is based upon a messages posted before and after a specific date. Unfortunately, the SHAP Deep Explainer does not work with newer versions of Tensorflow or MLP Classifier with Text vectors. So, in this case we had to restrict to LIME analysis only.

- Prediction task is to determine whether article labeled as **talk.politics.guns** (=1) or **talk.politics.misc** (=0)
- Neural Net with 3 hidden layers (30,20,10), ReLU activation and ADAM solver
- Training Accuracy of 100.0% and Test Accuracy of 82.0%% (with 67-33 split)
- Model may be overtrained

**LIME Analysis:** LIME for text differs from LIME for tabular data. Variations of the data are generated differently: Starting from the original text, new texts are created by randomly removing words from the original text. The dataset is represented with binary features for each word. A feature is 1 if the corresponding word is included and 0 if it has been removed.

We looked at a few cases and present here a case where politics guns class is predicted. As you see in the figure below, presence of texts like BATF, guns and FBI triggered this choice and looks completely credible.




We did another case with Atheism versus Christianity and below is a LIME output. The model would have tagged the message as connected to Atheism (higher probability). One text that drives the result is **unm**, which is part of an email address. Maybe lot of atheism related articles came through this email address. This is an example of data leakage where an unintentional leakage of signal into the data that can wrongly increase accuracy.



## Summary:

- In almost all the cases above, LIME and SHAP showed that the models were working as intended. In the last case of using email as a feature, it can be used to improve the features to include in the model. Maybe in this case, remove emails from the text list.
- Local surrogate models, with LIME as a concrete implementation, are very promising. But the method is still in development phase and many problems like picking the right perturbation neighborhood need to be solved before it can be safely applied.
- SHAP connects LIME and Shapley values. This is very useful to better understand both methods. It also helps to unify the field of interpretable machine learning. Though SHAP (Kernel) is slow and ignores feature dependence.
- Both these methods can be extremely useful when evaluating Deep Learning output and can help building trust in the accuracy of the models.
- One of the methods of checking results as explained in the lectures was going through the error examples manually. In that step, these methods can help us shed more light on the predictions.
- This can be used in conjunction with the model interpretability methods taught in the course to enhance understanding and convince the decision makers on the actual accuracy of the models.

## References:

- "Why Should I Trust You?" Explaining the Predictions of Any Classifier – 2016 (Ribeiro, Singh and Guestrin)
- Grad-CAM: Visual Explanations from Deep Networks via Gradient-based Localization – 2019 (Selvaraju, Cogswell, et al)
- A Unified Approach to Interpreting Model Predictions – 2017 (Lundberg and Lee)
- Neural Network Attributions: A Causal Perspective – 2019 (Chattopadhyay, Manupriya, et al)
- Interpretable Machine Learning (Molnar)
- Doshi-Velez, Finale, and Been Kim. "Towards a rigorous science of interpretable machine learning," no. MI: 1–13. <http://arxiv.org/abs/1702.08608> ( 2017). 
- Alvarez-Melis, David, and Tommi S. Jaakkola. "On the robustness of interpretability methods." arXiv preprint arXiv:1806.08049 (2018)
- 

## Weblinks:

- <https://github.com/slundberg/shap>
- <https://marcotcr.github.io/lime/>
- <https://christophm.github.io/interpretable-ml-book/>
- <https://www.oreilly.com/content/introduction-to-local-interpretable-model-agnostic-explanations-lime/>
- <https://archive.ics.uci.edu/ml/datasets/census+income>
- [https://scikit-learn.org/0.19/datasets/twenty\\_newsgroups.html](https://scikit-learn.org/0.19/datasets/twenty_newsgroups.html)

## Appendix A



Inceptionv3 is a convolutional neural network for assisting in image analysis and object detection, and got its start as a module for GoogLeNet. It is the third edition of Google's Inception Convolutional Neural Network, originally introduced during the ImageNet Recognition Challenge. Just as ImageNet can be thought of as a database of classified visual objects, Inception helps classification of objects in the world of computer vision. The picture we used as the input is one of me with my dog (mix of Lab, Shepherd and another breed) shown above.

The top 4 predictions from are given below and almost looks magical.

*('n02099267', 'flat-coated\_retriever', 0.3687284)*

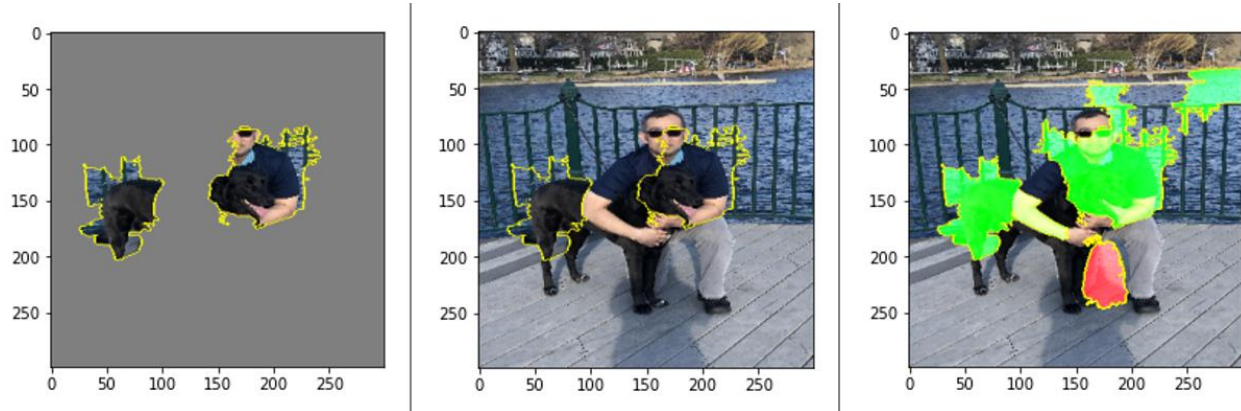
*('n02099429', 'curly-coated\_retriever', 0.27770984)*

*('n02099712', 'Labrador\_retriever', 0.13314323)*

*('n02105056', 'groenendael', 0.05306436)*

*('n02097130', 'giant\_schnauzer', 0.039444372)*

**LIME Analysis:** Superpixels are generated using a quickshift segmentation algorithm. Superpixels are interconnected pixels with similar colors and can be turned off by replacing each pixel with a user-defined color such as gray. For the case of this image, here are the superpixels.



The areas marked green or the superpixels on the left picture are the ones driving the highest probability choice.

The picture below shows the heatmap for the first prediction. The dark blue ones (head and back of the dog) drives the results, whereas my hand around the dog does not!

