Interpreting your deep learning model by SHAP

towardsdatascience.com/interpreting-your-deep-learning-model-by-shap-e69be2b47893

Edward Ma

Aug 18, 2018

As mentioned in previous article, model interpretation is very important. This article continues this topic but sharing another famous library which is SHapley Additive exPlantions (SHAP)[1]. It is introduced by Lundberg et al. who proposed a unified approach to interpreting model predictions.

After reading this article, you will understand:

- What is Shapley Value
- SHapley Additive exPlantions (SHAP)
- Use Case
- Takeaway

Shapley Value

Before introducing SHAP, let take a look on Shapley value which is a solution concept in cooperative game theory.

$$\phi_i(v) = \frac{1}{|N|!} \sum_{R} \left[v(P_i^R \cup \{i\}) - v(P_i^R) \right]$$

Φ: Shapley value
N: Number of player (feature)
P_i^R: Set of player with order
$V(P_i^R)$: Contribution of set of player with order
$V(P_i^R \cup \{i\})$: Contribution of set of player with order and player i

Let take a development team as an example. Our target is going to deliver a deep learning model which needs to finish 100 line of codes while we have 3 data scientists (L, M, N). 3 of them must work together in order to deliver the project. Given that:

Contribution among coalition

V(X)	Line of codes
L	10
M	30
N	5
L, M	50
L, N	40
M, N	35
L, M, N	100

Order	L Contribution	M Contribution	N Contribution
L, M, N	V(L) = 10	V(L,M) – V(L) = 50 – 10 = 40	V(L,M,N) – V(L,M) = 100 – 50 = 50
L, N, M	V(L) = 10	V(L,M,N) – V(L, N) = 100 – 40 = 60	V(L,N) – V(L) = 40 – 10 = 30
M, L, N	V(L,M) – V(M) = 50 – 30 = 20	V(M) = 30	V(L,M,N) – V(L,M) = 100 – 50 = 50
M, N, L	V(L,M,N) – V(M,N) = 100 – 35 = 65	V(M) = 30	V(M,N) – V(M) = 35 – 30 = 5
N, L, M	V(L,N) – V(L) = 40 – 5 = 35	V(L,M,N) – V(L,N) = 100 – 40 = 60	V(N) = 5
N, M, L	V(L,M,N) – V(M,N) = 100 – 35 = 65	V(M,N) – V(N) = 35 – 5 = 30	V(N) = 5

Marginal Contribution by different orders

We have 3 player so the total combination is 3! which is 6. The above tables show the contribution according to different order of coalition.

Contributor	Shapley Calculation	Shapley Value
L	1/6(10+10+20+65+35+65)	34.17
M	1/6(40+60+30+30+60+30)	41.7
N	1/6(50+30+50+5+5+5)	24.17

According to the Sherley value formula, we have the above tables. Although the capability of M is 6 times greater than N (30 vs 5), M should get 41.7% of reward while N should get 24.17% reward.

SHapley Additive exPlantions (SHAP)

The idea is using game theory to interpret target model. All features are “contributor” and trying to predict the task which is “game” and the “reward” is actual prediction minus the result from explanation model.

In SHAP, feature importance is assigned to every feature which is equivalent to mentioned contribution. Let take auto loan (car loan) as an example. We have “New Driver”, “Has Children”, “4 Door” and “Age”.

Theoretically, number of combination is 2^n, where n is number of feature. Given that we want to know the Shapley value of the “Age”. We will predict all of the following combination with and without “Age” feature. There is some optimization mentioned

Combination	New Driver	Has Children	4 Door
1	No	No	No
2	Yes	No	No
3	No	Yes	No
4	No	No	Yes
5	Yes	Yes	No
6	Yes	No	Yes
7	No	Yes	Yes
8	Yes	Yes	Yes

All possible combinations

By using the the Shapley formula, SHAP will compute all above scenario and returning the average contribution and . In other word, it is **not talking about the difference when the particular feature missed**.

Use Case

SHAP provides multiple explainers for different kind of models.

- TreeExplainer: Support XGBoost, LightGBM, CatBoost and scikit-learn models by Tree SHAP.
- DeepExplainer (DEEP SHAP): Support TensorFlow and Keras models by using DeepLIFT and Shapley values.
- GradientExplainer: Support TensorFlow and Keras models.
- KernelExplainer (Kernel SHAP): Applying to any models by using LIME and Shapley values.

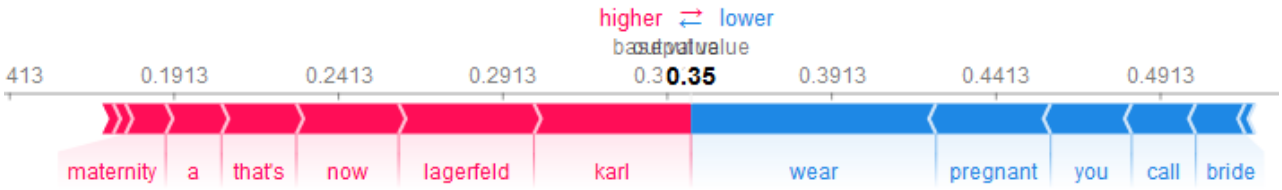
The following sample code will show how we can use DeepExplainer and KernelExplainer to explain text classification problem.

DeepExplainer

```
explainer = shap.DeepExplainer(pipeline.model, encoded_x_train[:10])
shap_values = explainer.shap_values(encoded_x_test[:1])

x_test_words = prepare_explanation_words(pipeline, encoded_x_test)
y_pred = pipeline.predict(x_test[:1])
print('Actual Category: %s, Predict Category: %s' % (y_test[0], y_pred[0]))

shap.force_plot(explainer.expected_value[0], shap_values[0][0], x_test_words[0])
```

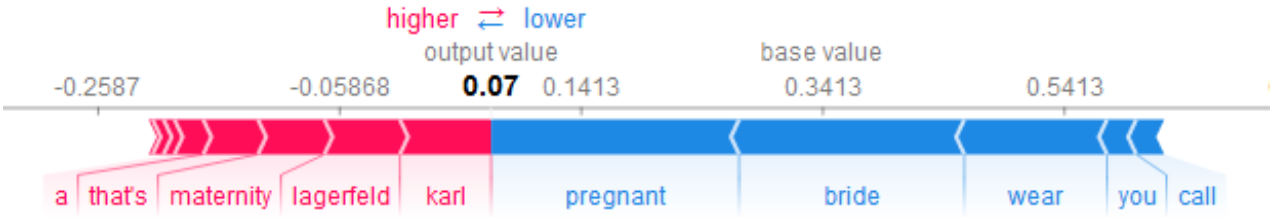


KernelExplainer

```
kernel_explainer = shap.KernelExplainer(pipeline.model.predict, encoded_x_train[:10])
kernel_shap_values = kernel_explainer.shap_values(encoded_x_test[:1])
```

```
x_test_words = prepare_explanation_words(pipeline, encoded_x_test)
y_pred = pipeline.predict(x_test[:1])
print('Actual Category: %s, Predict Category: %s' % (y_test[0], y_pred[0]))

shap.force_plot(kernel_explainer.expected_value[0], kernel_shap_values[0][0], x_test_words[0])
```



Takeaway

To access all code, you can visit my github repo.

- Shapley value is the average contribution of features which are predicting in different situation. In other word, it is **not talking about the difference when the particular feature missed**.
- SHAP includes **multiple algorithms**. You may check out paper for more detail on LIME, DeepLIFT, Sharpley value calculations.
- It is possible that DeepExplainer and KernelExplainer **introduce different results**.

About Me

I am Data Scientist in Bay Area. Focusing on state-of-the-art in Data Science, Artificial Intelligence , especially in NLP and platform related. You can reach me from Medium or Github.

Reference

[1] Lundberg S. M., Lee Su-In. A Unified Approach to Interpreting Model Predictions. 2017. http://papers.nips.cc/paper/7062-a-unified-approach-to-interpreting-model-predictions.pdf

[2] Lundberg S. M., Erion G. G., Lee Su-In. Consistent Individualized Feature Attribution for Tree Ensembles. 2017. https://arxiv.org/pdf/1802.03888.pdf