{ DevFest }

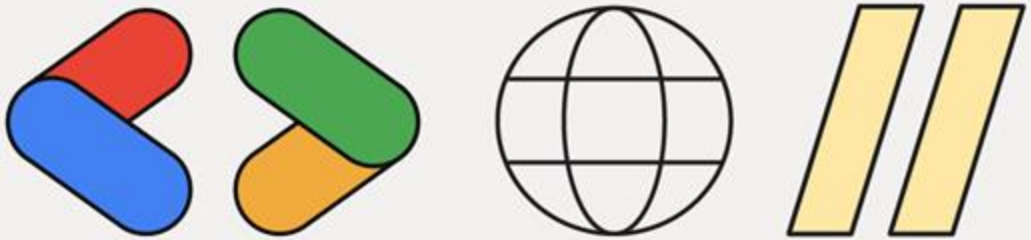
Dubai, UAE

Implementing eXplainable AI with Tensorflow & Keras

Rohan Mitra

ML Engineer @Bayut | Dubizzle
Research Assistant for ML @ CUD
Organizer & Speaker GDG Sharjah

Google Developer Groups



DevFest

Dubai, UAE

Code With Me!

Download template notebook & dataset
Upload both into Google Colab

Google Developer Groups

XAI is a way to get a glimpse into what complex AI models have learnt and how they reason

Google Developer Groups

AI @DevFest

White Box AI Models

- Linear Regression
  - Coefficients indicate importance & contribution to final output
- Decision Trees
  - Produce branched “if” statements
- Explainable, but limited performance

Black Box AI Models

- Neural Networks
  - Non-linear function approximator
- High performance, but poor visibility
- Works, but WHY??

XAI

- How does the model think?
- How the model reacts to changes
- Understand WHY this output for this input
- Retrospective

Integrated Gradients

- Linear Regression Equation:
- $y = c_1x_1 + c_2x_2 + c_3x_3 + \cdots c_nx_n$
- c_i is the importance of feature x_i
- c_i shows the overall contribution of x_i towards y
- c_i is the **gradient** of y w.r.t x_i

- Can we do this for neural networks?

Integrated Gradients

- Lets look at neural network gradients!
- $\frac{\partial(output)}{\partial(input)}$ is the gradient

- But how is the input changing?
- How do we measure this change?

Integrated Gradients – Intuition

- Walk from an “average” point in the dataset to the input we’re interested in
- Walk = adjust feature values to move a baseline point towards the input
- See how each step changes the output
  - Big change → important feature
- Take the average gradient across the path

Integrated Gradients – Inputs

- Linear path from baseline (b) to input (x)

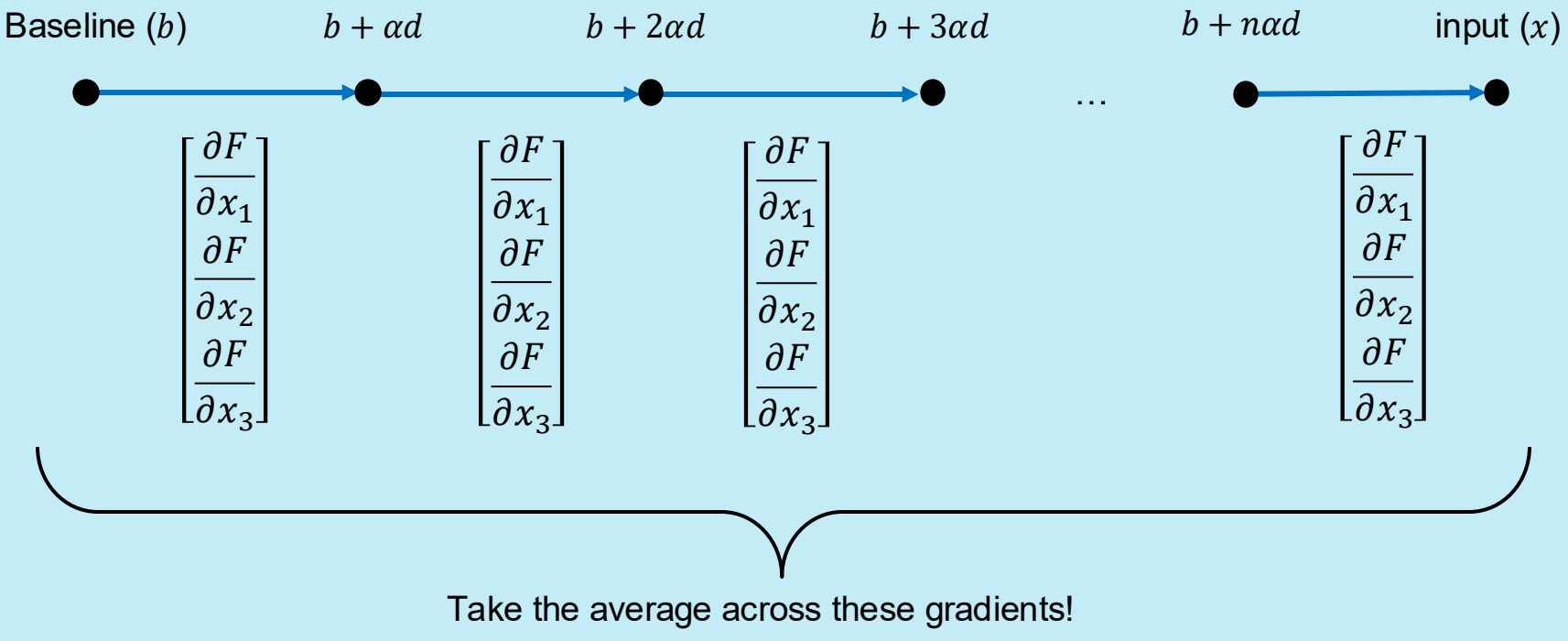
- The delta $d = (x - b)$

Integrated Gradients – Outputs

- As α varies, we can measure the output of the neural network (F)
- This gives us $\frac{\partial F(input)}{\partial (input)}$
- Attribution = $(input\ -baseline) * Gradient$
- [amount moved = distance traveled * speed]

$$(Attribution)_{input} = \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right) * \begin{bmatrix} \frac{\partial F}{\partial x_1} \\ \frac{\partial F}{\partial x_2} \\ \frac{\partial F}{\partial x_3} \end{bmatrix}$$

Integrated Gradients – Outputs



Integrated Gradients – Final Formula

- Since we vary the input, we have multiple gradient values
- We average the gradients for each feature

$$(Integrated\ Gradients)_{\ i} = (x - b) * \int_{\alpha=0}^{1} \frac{\partial F\big(b + \alpha(x - b)\big)}{x_i} d\alpha$$

- To implement:

$$(Integrated\ Gradients)_{\ i} = (x - b) * \frac{1}{m} \sum_{k=1}^{m} \frac{\partial F\big(b + \alpha(x - b)\big)}{x_i}$$

Integrated Gradients – Completeness

- Integrated Gradients is “complete”
- The total attributions is the difference between $F(baseline)$ and $F(input)$
- $y_{pred\ from\ input} - y_{pred\ from\ baseline} = \sum attributions$

$$\begin{bmatrix} att(x_1) \\ att(x_2) \\ att(x_3) \end{bmatrix} = \left(\begin{bmatrix} x_1 \\ x_2 \\ x_3 \end{bmatrix} - \begin{bmatrix} b_1 \\ b_2 \\ b_3 \end{bmatrix} \right) * \begin{bmatrix} Avg\ \frac{\partial F}{\partial x_1} \\ Avg\ \frac{\partial F}{\partial x_2} \\ Avg\ \frac{\partial F}{\partial x_3} \end{bmatrix}$$

Integrated Gradients – Result

- Found “coefficients” for the neural network like in Linear Regression
  - Like: $y = a_1 f_1 + a_2 f_2 + a_3 f_3 + \cdots a_n f_n$
- Large attributions show more important features
  - Bigger change in output because of small change in input
- Can account for the exact change in prediction based on the change in a feature
  - Just like in linear regression
- “Local” explanations – **coefficients only valid for that particular input & baseline**

DiCE – Diverse Counterfactual Explanations

- HOW to change the input to get desired output
- Wiggles the inputs until output is in desired range
- Focuses on minimal input changes first

Case Study – G-taxi

- You are an AI Engineer at G-taxi
- You need to build a price prediction model for each ride that the user can book
- You built a great model with <10% percentage difference
- But your manager and your customers want to know WHY the model has given certain prices
- XAI to the rescue!

DevFest

Dubai

Thank You!

https://www.linkedin.com/in/rohan-mitra14/

https://github.com/ro1406

scholar.google.com/citations?user=Jh0DuX4AAAAJ

Medium https://medium.com/@rohanmitra8

Google Developer Groups