A Survey of Deep Learning for Scientific Discovery

by Maithra Raghu, Eric Schmidt @ Google

> In this survey, we focus on addressing this central issue, **providing an overview of many widely used deep learning models, spanning visual, sequential and graph structured data, associated tasks and different training methods, along with techniques to use deep learning with less data and better interpret these complex models —** two central considerations for many scientific use cases
>
> We also **include overviews of the full design process**, implementation tips, and links to a plethora of tutorials, research summaries and open-sourced deep learning pipelines and pretrained models, developed by the community

---

7 Interpretability, Model Inspection and Representation Analysis

> **Identifying underlying mechanisms** giving rise to **observed patterns in the data**. When **applying deep learning in scientific settings**, we can use these observed phenomena as prediction targets, but the **ultimate goal** remains **to understand what attributes give rise to these observations**

1. In the **Scientific field**, **Deep Learning** is often used to **UNDERSTAND a certain phenomenon**
    - For example
        - Input : **Amino acid**, output : **the predicted properties of the protein**
        - Understanding how that amino acid sequence resulted in the observed protein function
    - Interpretability techniques
        - **Fully understandable**, **step-by-step explanation** of the model's decision process
        - From **feature attributions** to **model inspection**
            - feature attr : Determining **what input features matter** the most
            - model ins. : Determining **what causes** neurons in the network to fire
            - These two types provide a rough split in the type of interpretability

2. **Two Distinctive methods**
    1. **Feature Attribution** (Per Example Interpretability)
        1. Concentrates on **taking a specific input** along **with a trained deep neural network**
        2. Determining **what features of the input** are most important
    2. **Model Inspection** (Model Inspection and Representational Analysis)
        1. **Revealing important, hidden patterns in the data** that the **model has implicitly learned** through being trained on the predictive task
        2. Example, **Machine translation => representation analysis techniques** used to illustrate latent linguistic structure learned by the model

7.1 Feature Attribution and Per Example Interpretability

Feature attr. @ a per example level => **Answering questions** such as **which parts of an input image** are **most important** for a particular model prediction

7.1.1 Saliency Maps and Input Masks

1. Saliency maps
    - Take the **Gradient of the output prediction with respect to the input**
    - Gives **mask** over the input => Highlighting **which regions have large gradients** (most important for the prediction)
2. Example => These **inspections are not perfect** though
    - Grad-CAM
    - SmoothGrad
    - IntGrad
    - https://github.com/PAIR-code/saliency
    - " The building blocks of interpretability "
        - Provides **the ability** to inspect **the kinds of features causing neurons across diff. hidden layers to fire**
        - https://distill.pub/2018/building-blocks/
        - https://github.com/tensorflow/lucid
    - Using Deconvolutional layers

7.1.2 Feature Ablations and Perturbations

Isolate the crucial features of the input either **by performing** ***feature ablations*** or **computing** ***perturbations*** **of the input** and **using these perturbations** along with the original input **to inform** the **importance of different features**.

1. Feature Ablations
    1. The notion of ***Shapely value***
        1. Estimates **the importance of a particular feature x_0** in the input **by computing the predictive power of a subset of input features**

        2. And averaging over all possible subsets
2. Utilizing perturbations
    1. LIME
        - Uses **multiple local perturbations** to enable learning an interpretable local model
    2. DEEPLIFT
        - Uses a reference input to compare activation differences

7.2 Model Inspection and Representation Analysis

Gaining insight not at a single input example level, but **using a set of examples to understand the salient properties of the data**

7.2.1 Probing and Activating Hidden Neurons

(i) Understanding what kinds of inputs it activates for

(ii) Directly optimizing the input to activate a hidden neuron

1. Network Dissection
    - **Hidden neurons** are categorized by **the kinds of features** they respond to
    - http://netdissect.csail.mit.edu/
2. Take a **NN**, **fix its params** and **optimize the input** to find the kinds of features that makes some hidden neuron activate

7.2.2 Dimensionality Reduction on Neural Network Hidden Representations

1. Dimensionality Reduction in standard scientific settings

    Useful in **revealing important factors of variation** and **critical differences in the data subpopulations**

    1. PCA (Principal component Analysis)
    2. t-SNE
    3. UMAP
2. The **NN may implicitly learn these important data attrs**. in its hidden representations **which can then be extracted through dimensionality reduction methods**.

7.2.3 Representational Comparisons and Similarity

1. A line of work has studied **comparing hidden representations across different NN models**
    1. Matching algorithms
    2. Canonical Correlation analysis
        1. Used to identify and understand many representational properties in NLP applications
        2. Modelling the mouse visual cortex as an ANN
    3. **Kernel based approach** to perform **similarity comparisons**

1. "Similarity of Neural Network Representations Revisited" with code implementations