Layered relevance propagation

July 28, 2017

1 Introduction

Assumption

Before getting started, lets assume the following set up:
A neural network is defined as a function $f(x)$ which has multiple layers and each layer has a set of neurons. These neurons do some element wise computations on pixels(other neurons depending on the complexity of the neural network) which is x followed by non linear activation functions. The final output is represented as x_f. Now, the input to neural network is an image of p pixels. This is represented as $\{x_p\}$.

There are two main techniques for analyzing neural network prediction.

1.1 Sensitivity

It measures the local effect of the neurons for the given output. It is discrete and shows how much impact a neuron makes in deciding the output.

$$\Sigma_p(\frac{\delta f}{\delta x_p})^2 = ||\Delta_x f(x)||^2$$

1.2 Decomposition

It measures the global effect for the given ouput. It is contiuous.

$$\Sigma_p[f(x)]_p = f(x)$$

In this blog, we will look into the decomposition analysis of neural networks.

2 Deep Taylor decomposition method

2.1 Taylor decomposition

A decomposition method based on taylor expansion of a differentiable function $f(x)$ at a root point $\tilde{x}$. The $\tilde{x}$ is chosen such that $f(\tilde{x}) = 0$. The first order taylor expansion is given by:

$$f(x) = \left(f(\tilde{x}) + \frac{\delta f}{\delta x}|_{x=\tilde{x}} \right)^T .(x - \tilde{x}) + \varepsilon$$

In relevance to neural network,

$$= 0 + \sum_p \frac{\delta f}{\delta x}|_{x=\tilde{x}}.(x_p - \tilde{x}_p) + \varepsilon \tag{1}$$

The sum was over all pixels p in the image and the $\tilde{x}_p$ are the pixel values for root point. The terms inside the sum $\sum_p$ are the relevances assigned to pixels. Therefore, relevance is defined as :

$$R_p(x) = \frac{\delta f}{\delta x}|_{x=\tilde{x}}.(x_p - \tilde{x}_p) \tag{2}$$

In large deep networks the root point $\tilde{x}$ is hard to perceive for the data point x.

This is addressed by **Deep Taylor decomposition method** by decomposing the already learned function $f(x)$ into a set a sub functions which applies locally to sub set of pixels directly or on abstracted version of them depending on the layer at which they are located in the network.

2.2 Relevance Backpropogation

Let $f(x)$ yield an output x_f. The output is then decomposed on to the neurons present in the previous layers. So, lets take x_j to be a neuron present at a layer and its relevance to be R_j. We need to decompose/redistribute R_j on to the set of neurons in the lower layer which connects it and this set is represented as x_i. Therefore, the relevance(R_j) in terms of x_i is can be derived by the taylor decomposition method explained above, and we define a root point for this decomposition function as $\{\tilde{x}_i\}^{(j)}$.

$$\begin{aligned} R_j &= \left(\frac{\delta R_j}{\delta \{x_i\}}|_{\{\tilde{x_i}\}^{(j)}} \right)^T .(\{x_i\} - \{\tilde{x_i}^{(j)}\}) + \varepsilon_j \\ &= \sum_i \underbrace{\frac{\delta R_j}{\delta \{x_i\}}|_{\{\tilde{x_i}^{(j)}\}}.(\{x_i\} - \{\tilde{x_i}\}^{(j)})}_{R_{ij}} + \varepsilon_j \end{aligned} \tag{3}$$

Now, to determine the total relevance of a neuron x_i you need to sum all the relevances contributing to the neuron x_i.

$$\begin{aligned} R_i &= \sum_j R_{ij} \\ &= \sum_j \frac{\delta R_j}{\delta \{x_i\}}|_{\{\tilde{x_i}^{(j)}\}}.(\{x_i\} - \{\tilde{x_i}\}^{(j)}) \end{aligned} \tag{4}$$

We can see that there are two features to be noted from the above equations:

- The relevance is conserved during redistribution.
- If the values of all neurons are positive or equal to zero then according to taylor decomposition, the relevance of the same is also positive.

We take the relevance of the neuron in upper layer and redistribute it to the neurons connected to the lower layer. Similarly, we compute the relevance of the current neuron by summing up the redistributed relevancies of the upper layer neuron which is connected to it. This process is analogous to brack propagation and hence the term **relevance back propagation**. The below picture would give you a better understanding.

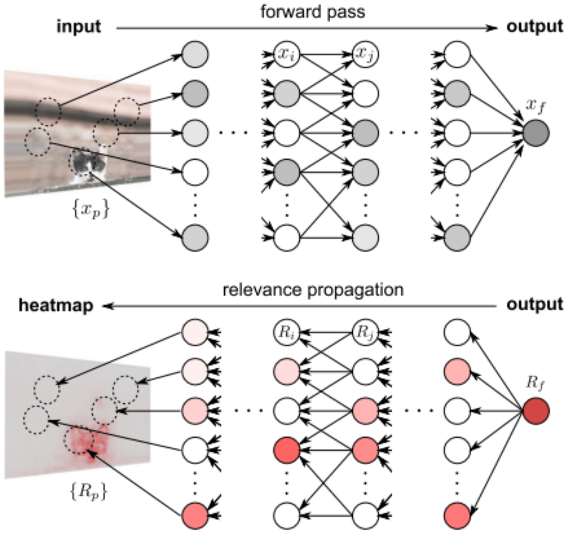

Figure 1: Relevance back propagation

2.3 Choosing a root point

It was always said that we would choose a root point in such a way that $f(\tilde{x}) = 0$ but the ways to satisfy that condition was never explained and this section shall discuss that.

2.3.1 Unconstrained input space

This is the simple and ideal scenario where the input belongs to any real value $X = R^d$. In this case, we can choose the root point $\{\tilde{x}_i\}^{(j)}$ that is closest in terms of euclidean distance to $\{x_i\}$. Therefore for a positive relevance say R_j the root point in such cases would be intersection of $f(\tilde{x_i}) = 0$ and vector of maximum descent(search direction) is given by $\tilde{x_i}^{(j)} = \{x_i\} + t.\mathbf{w}_j$. $\mathbf{w}_j$ is the weight parameter connecting the neuron x_j.

Search direction is defined as,

$$\tilde{x_i}^{(j)} = \{x_i\} + t.\mathbf{w}_j \tag{5}$$

According to root point definiton,

$$f(\tilde{x_i}) = 0$$

$$\sum_i \tilde{x_i}^{(j)}.w_{ij} + b_j = 0 \tag{6}$$

Substituting eq (5) in (6)

$$t = \frac{-1}{\sum_i w_{ij}^2}.\sum_i w_{ij} + b_j$$

Injecting it back to eq (5)

$$\{\tilde{x_i}\}^{(j)} = \{x_i - \frac{w_{ij}}{\sum_i w_{ij}^2}.\sum_i w_{ij} + b_j\} \tag{7}$$

Therefore substituting (7) in equation (4) and simplifying it,

$$R_i = \sum_j (\frac{w_{ij}^2}{\sum_{i'} w_{i'j}^2}) R_j \tag{8}$$

This is the redistributed relevance

2.3.2 Constrained input space

When the input domain $X \subset R^d$ then the nearest root might not be in the same set, hence we restrict our search space according to our feature space domain. Multiple cases of such a setting are explained below.

Application of Rectified Linear Units

For ReLU, the search domain is restricted to only the positive part of w_{ij} and hence the redistributed relevance is given by:

$$R_i = \sum_j \frac{z_{ij}^+}{\sum_{i'} z_{i'j}^+} R_j \tag{9}$$

where $z_{ij}^+ = x_i w_{ij}^+$ and w_{ij}^+ denotes positive part of w_{ij}.

Application for Images as inputs

Similarly in the case of image classification tasks, we would consider only those feature spaces that fall under the domain of admissible pixel values in each dimension. The input space can be written as,

$$B = \{\{x_i\} : \forall_{i=1}^d l_i \leq x_i \leq h_i\}$$

where the input has d dimensions and $l_i \leq 0$ & $h_i \geq 0$ are the smallest and largest admissible pixel for each dimension. Then, the redistributed relevance is:

$$R_i = \sum_j \frac{z_{ij} - l_i w_{ij}^+ - h_i w_{ij}^-}{\sum_{i'} z_{i'j} - l_i w_{i'j}^+ - h_i w_{i'j}^-} \tag{10}$$