Research Paper Analysis: Gaussian Error Linear Units (GELUs)

Alex Shrestha

CS545 Machine Learning

1 Introduction

For this research paper analysis, I decided to focus on activation functions due to their substantial importance in neural networks. Activations functions introduce the much needed non-linearity and thus are fundamental to neural networks. Without non-linear activations, a very deep layered neural network could be represented by a single layer. The paper I will analyze is Gaussian Error Linear Units (GELUs) by Dan Hendrycks and Kevin Gimpel. The paper's goal is to introduce a new activation function called the Gaussian Error Linear Unit or GELU for short. The paper emprically evalutates how the GELU performs against standard non-linear activations ReLU and ELU by running experiments on the MNIST dataset, Twitter POS tagging dataset, TIMIT dataset, and the CIFAR-10/100 dataset. One of the core reasons that led the authors to developing a new activation function is that they wanted to address the existing limitations of existing activations function like the sigmoid and ReLU. The sigmoid activations were useful in the early days of networks but as networks became deeper, the performance of the sigmoid diminished. Thus ReLU came about and it's ”non-smooth gating” provided faster and better convergence. One issue with ReLU was its inability to output negative values and its less probabilistic nature. Thus the GELU was born which incorporates a probabilistic approach which will explore next.

2 Analysis

The GELU activation function is $x\Phi(x)$ where $\Phi(x)$ is the standard Gaussian cumulative distribution function. The GELU nonlinearity weights inputs by their value, rather than gating inputs by their sign as ReLUs do. The goal is still to have a form of dropout involved. It is important to note that ReLU and dropout accomplish similar things (yielding a neuron's output) but in different ways: ReLU does this deterministically using the sign of an input whereas dropout does this randomly.

GELU multiplies the input by zero or one randomly by multiplying the neuron input x by $m \sim \text{Bernoulli}(\Phi(x)) = P(X \leq x),\ X \sim \mathcal{N}(0,1)$. We can notice that as x decreases, the probability of the input being dropped inversely increases. However, the goal is to have a "deterministic decision". What this means is that an input should not have multiple different outputs, leading to more stability since slight changes in the input will not cause a drastic change. To achieve this goal, the expected value of the transformation on an input x is taken: we have $\Phi(x) \times 1x + (1 - \Phi(x)) \times 0x = x\Phi(x)$. This equation can be approximated leading to:

$$\text{GELU}(x) = xP(X \leq x) = x\Phi(x) = x \cdot \frac{1}{2}\left[1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right]$$

where the error function is used to compute the cumulative distribution function of the Gaussian.

3 Conclusion

4 References