Gsoc 2022 Project: FASEROH

Harrison B. Prosper[1] *and Sergei V. Gleyzer*[2]
[1]Florida State University, [2]University of Alabama

Abstract
We describe a project to build a seq2seq model that maps histograms to empirical symbolic representations.

1 Introduction

State-of-the-art sequence to sequence models (seq2seq) have yielded spectacular advances in neural machine translation (NMT) (see, for example, Ref. [1]). Recently, these models have been successfully applied to symbolic mathematics by conceptualizing the latter as translation from one sequence of symbols to another [2]. It is easy to imagine numerous tasks that can be construed as translations. In the proposed Gsoc project the goal is to create a tool that automatically provides an accurate symbolic representation of a histogram by construing the problem as one of translation from a histogram to a symbolic function. We call the project Fast Accurate Symbolic Empirical Representation Of Histograms (FASEROH).

2 Project Description

The goal of the project is to use available seq2seq models to create the mapping,

$$g : H(N_1, \cdots, N_K) \to f(x, \theta)\,;\, \hat{\theta},$$

where g is a trained seq2seq model, H is a histogram, that is, a sequence of K integers, $N_1, \cdots, N_K$, $f(x, \theta)$ is a symbolic function for which $\int_0^1 f(x, \theta)\, dx = 1$, and $\hat{\theta}$ are the best-fit values of the parameters θ. Since this project will be a proof of principle, the seq2seq task will be limited to 1-dimensional histograms defined on the unit interval $\{U : 0 \leq x \leq 1\}$.

2.1 Data Generation

Initially, we shall avoid the overhead of actually fitting proposed functions by generating histograms from the generated functions as described in Algorithm 1.

Algorithm 1 Data generation

initialize data set: $\mathbb{T} \leftarrow \{\}$
initialize range of summed histogram count N: $N_{\min}, N_{\max}$
initialize range of the number histogram bins K: $K_{\min}, K_{\max}$
while $k \in [1, \cdots T]$ **do**
  generate function: $f(x, \theta)$, $x \in U$, $\quad \int_0^1 f(x, \theta)\, dx = 1$
  generate best-fit parameters: $\hat{\theta}$
  generate summed histogram count: $N \leftarrow \text{randint}(N_{\min}, N_{\max})$
  generate number of histogram bins: $K \leftarrow \text{randint}(K_{\min}, K_{\max})$
  generate histogram: $H \leftarrow N f(x, \hat{\theta})$
  update data set: $\mathbb{T} \leftarrow \mathbb{T} \cup \{H, f, \hat{\theta}\}$
end while

If time permits, the covariance matrix, Σ, associated with $\hat{\theta}$ can be estimated using `iminuit`[1].

[1]https://pypi.org/project/iminuit/

2.2 Function Generation

The key to FASEROH is generating a sufficiently diverse set of functions that span the space of functions used in particle physics and cosmology. Therefore, the first step is to create an inventory of the functions most commonly used in these fields, noting their typical parameterizations. We call this set the set of *base functions*, $\mathbb{B}$. A tentative algorithm for generating functions is given in Algorithm 2.

Algorithm 2 Function generation

initialize function combination operators: $\mathbb{O} \leftarrow \{+, /, *\}$
select number of functions to combine: $m \leftarrow \text{randint}(1, 3)$
randomly select function: $f \leftarrow b(x, \phi) \in \mathbb{B}$
while $k \in [1, \cdots m-1]$ **do**
  randomly select base function: $b(x, \phi) \in \mathbb{B}$
  randomly select combination operator: $o \in \mathbb{O}$
  if o == + **then**
    combine functions: $f \leftarrow f + o + \omega_k\, b$
  else
    combine functions: $f \leftarrow f + o + b$
  end if
end while
simplify: $f \leftarrow \text{simplify}(f)$
integrate function: $I \leftarrow \text{integrate}(f, (x, 0, 1))$
if I **then**
  normalize function: $f \leftarrow f \,/\, I$
else
  discard f.
end if

In Algorithm 2 we discard functions that cannot be integrated. Note also, we introduce scale parameters ω_k when base functions are summed. The parameters θ comprise those of the base functions and any scale parameters ω_k that may have been introduced.

2.3 Testing

The simplest way to test the quality of the mapping is to perform a goodness-of-fit (gof) test, the simplest of which for Poisson data is to compute the quadratic form

$$X = \sum_{k=1}^{K} \frac{(N_k - n_k)^2}{n_k},$$

where

$$n_k = \int_{\text{bin}_k} f(x, \theta)\, dx,$$

is the mean count in histogram bin k. For a good fit one expects $X/\text{ndf} \approx 1$, where ndf (the number of degrees of freedom) is given by $K - P$ with P the number of free parameters θ. It is usually a good idea to exclude counts N_k smaller than about 5 in which case the ndf must be adjusted accordingly.

Bibliography

[1] Felix Stahlberg. Neural machine translation: A review and survey, 2020.

[2] Guillaume Lample and François Charton. Deep learning for symbolic mathematics, 2019.