Matching Simulation and Data via Stochastically Quantized Neural Networks

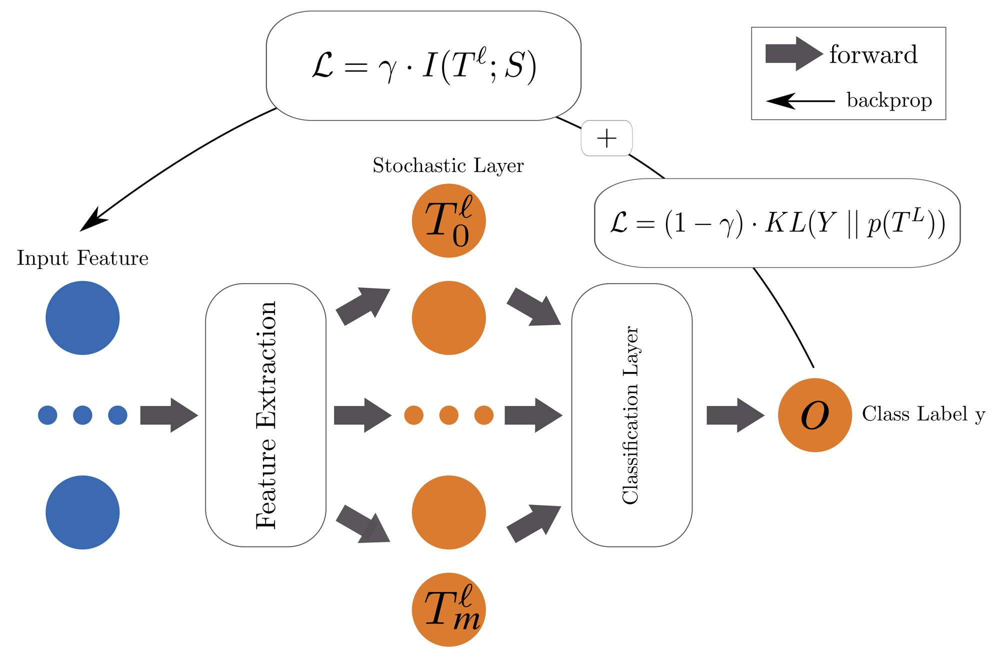
Mattia Cerrato[1], Marius Köppel[2]

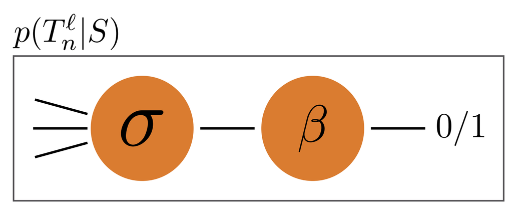
[1]Institut für Informatik, Johannes Gutenberg-Universität, Mainz
[2]Institut für Kernphysik, Johannes Gutenberg-Universität, Mainz

Abstract

Machine learning methodologies have increasingly been employed in high-energy physics research. The interest in ML for physics is due to the incredibly high amount of data which is produced by particle detectors, which makes it hard for researchers to analyze it in real time. Another issue with detector data is that it is not labeled, i.e. the distinction between signal and background is not available. Our contribution deals with learning invariant representations in a principled way by employing stochastically quantized neural networks. For this reason, machine learning models are first trained on simulated data for which the underlying data generation process is known. Furthermore, domain adaptation methods are used to transfer knowledge learned from the labeled simulated data to the unlabeled detector data, improving the performance of our contribution. To improve the use of such a model in high rate experiments, where the need of an online analysis of the data stream is paramount, we implement our model on Field Programmable Gate Arrays (FPGAs).

Stochastically Quantized Neural Network





Mutual Information Computation via Bernoulli activations

A feedforward neural network with L layers may be formalized by employing a sequence of "layer functions" ϕ^ℓ which computes the neural activations given an input $x \in \mathbb{R}^{d_0}$:

$$\phi^\ell(x) = \sigma(A^\ell \phi^{\ell-1}(x) + b^\ell), \qquad \ell = 1, \dots, L \tag{1}$$

$$\phi^0(x) = x, \tag{2}$$

We define neural representations as applications of ϕ to the random variable X which follows the empirical distribution of the x samples:

$$T^\ell := \phi^\ell(X), \qquad \ell = 1, \dots, L$$

We employ stochastic quantization of binary neurons, which we compute by sampling from $\mathcal{B}(\theta_i^l)$, where:

$$\theta_i^\ell = \sigma\left(\frac{1}{|\,T^{\ell-1}\,|}\sum_{i=1}^{|T^{\ell-1}|} w_i T_i^{\ell-1}\right), \tag{3}$$

where σ is the sigmoid function $\sigma(x) = \frac{1}{1+e^{-x}} \in [0,1]$. The entropy of a random variable following the Bernoulli distribution has a closed, analytical form:

$$H(T_i^\ell) = -(1-\theta_i^\ell) \cdot \log_2(1-\theta_i^\ell) - \theta_i^\ell \cdot \log_2(\theta_i^\ell). \tag{4}$$

We then consider the whole layer as a stochastic random vector $T^\ell = [T_1^\ell, T_2^\ell \dots T_m^\ell]$. If one assumes independence between the activations, it is then easy to compute the relevant mutual information measure for the whole layer. For instance, for the random vector $T^\ell = [T_1^\ell, T_2^\ell]$:

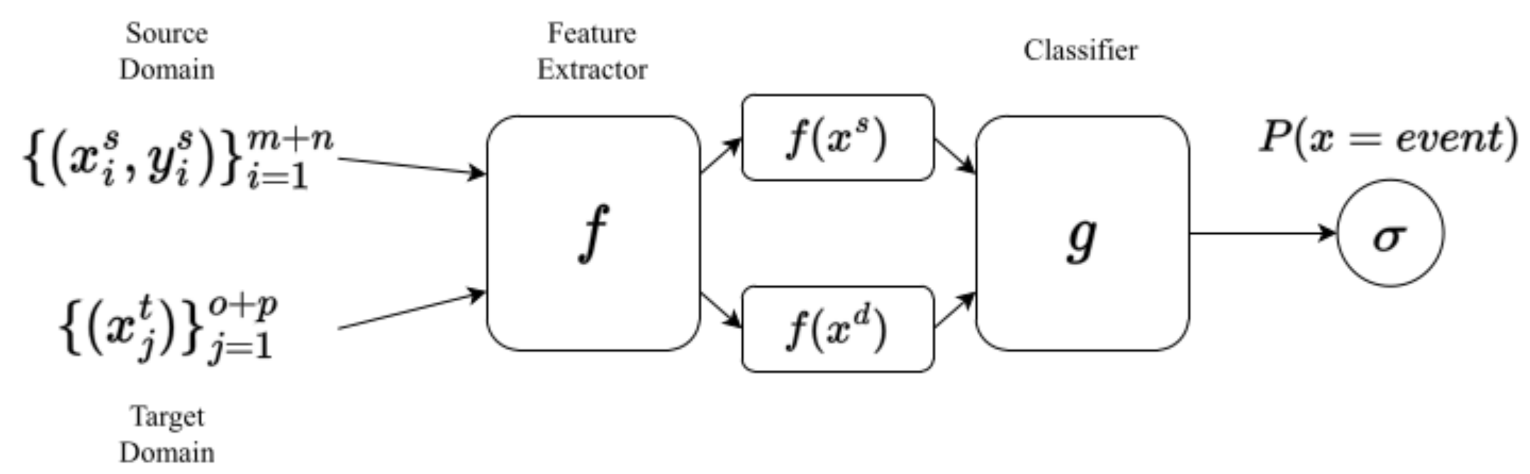
$$I(T^\ell; S) = H(T^\ell) - H(T^\ell \mid S) \tag{5}$$

$$- \left[\sum_{s \in S} H(T_1^\ell \mid S = s) + \sum_{s \in S} H(T_2^\ell \mid S = s)\right]. \tag{6}$$

Matching Simulation and Data

$\{(x_i^a, y_i^a)\}_{i=1}^m \quad y_i^a = 1 \;\forall i$ — Simulated signal
$\{(x_i^d, y_i^d)\}_{i=1}^n \quad y_i^d = 0 \;\forall i$ — Detector background
} $\{(x_i^s, y_i^s)\}_{i=1}^{m+n}$ — Source Domain

$\{(x_j^a\}_{j=1}^o$ — Simulated data (unlabeled)
$\{(x_j^d\}_{j=1}^p$ — Detector data (unlabeled)
} $\{(x_j^t)\}_{j=1}^{o+p}$ — Target Domain

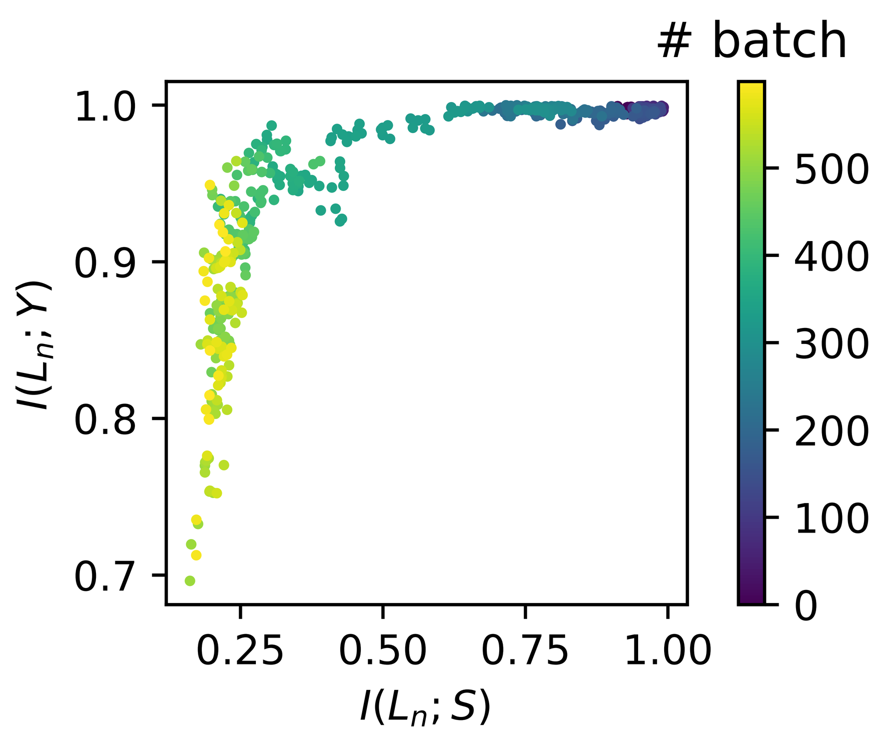


$$\mathcal{L} = \mathcal{D}(f(x^s), f(x^d)) + KL(g(f(x^s)) \,||\, y^s)$$

$$\mathcal{D} = I(f(x^s), S)$$

$$S \triangleq \mathbb{1}_{x_k \in \{(x_i^s, y_i^s)\}_{i=1}^{m+n}}$$

Evaluation on synthetic data



Conclusion & Future Work

- Resource efficient real time classification of unlabeled detector data to imporve readout applications
- Framework can be implemented on FPGAs to improve possible trigger applications
- First step to transport domain adaption on edge devices for high rate physics research

IEEE NPSS NUCLEAR & PLASMA SCIENCES SOCIETY

23rd IEEE Real Time Conference

References
[1] Tishby and Zaslavsky, Deep learning and the information bottleneck principle
[2] Courbariaux and Bengio, BinaryNet: Training Deep Neural Networks with Weights and Activations Constrained to +1 or -1