

Redes Neurais Quantizadas para Filtragem Online de Elétrons em um Experimento de Altas Energias

Gabriel H. B. Lisboa*, Lucas B. A. Nunes*, Pedro H. G. Lima*, Clara A. Pacheco*, Nestor P. Neto**, Igo Amauri dos S. Luz**, Larissa F. B. da Silva*, Leandro Assis dos Santos*, João V. D. F. Pinto*, Edmar E. P. de Souza**, Fernanda D. V. R. Oliveira*, Eduardo F. Simas Filho**, José Manoel de Seixas*, Denis Oliveira Damazio***

Resumo— Com as futuras atualizações para a *Run-4* do *Large Hadron Collider*, o componente *Global Trigger* será implementado em *hardware* digital no sistema de *Trigger* do experimento ATLAS para selecionar partículas a 1MHz. Este trabalho propõe a implementação em FPGA do algoritmo *NeuralRinger* para seleção online de elétrons, comparando as técnicas de Quantização Pós-Treinamento (PTQ) e *Quantization-Aware Training* (QAT). Os resultados indicam que ambas as técnicas mantiveram o desempenho do modelo em software, chegando a 9.51% na taxa de falso alarme no ponto de operação menos restritivo e consolidando-se como uma opção para a operação do algoritmo em *hardware*.

Palavras-Chave— experimento ATLAS, Trigger, *Field Programmable Gate Array*, elétrons, quantização, *Quantization-Aware Training*.

Abstract— With future updates for the Large Hadron Collider Run-4, the Global Trigger component will be implemented in digital hardware within the ATLAS experiment Trigger system to select particles at 1MHz. This work proposes the FPGA implementation of the NeuralRinger algorithm for online electron selection, comparing Post-Training Quantization (PTQ) and Quantization-Aware Training (QAT) techniques. Results indicate that both techniques maintained the software model performance, reaching 9.51% of false alarm rate for the less restrictive operating point and establishing itself as an option for the algorithm's operation in hardware.

Keywords— ATLAS experiment, Trigger, Field Programmable Gate Array, electrons, quantization, Quantization-Aware Training.

I. INTRODUÇÃO

Experimentos de Física de Partículas dedicam-se à investigação dos constituintes elementares da matéria e suas interações fundamentais [1]. Um instrumento crucial para o avanço experimental desta área é o *Large Hadron Collider* (LHC), localizado no Centro Europeu para Pesquisa Nuclear (CERN), o maior e mais potente acelerador de partículas do mundo [2], que colide feixes de prótons a uma energia de 13,6 TeV¹, em

quatro pontos de interação [3]. Nesses pontos, experimentos de grande porte são responsáveis por registrar os produtos emergentes dessas interações. O experimento *A Toroidal LHC Apparatus* (ATLAS) [2], o maior do LHC, é um detector de propósito geral, capaz de estudar uma ampla variedade de processos físicos resultantes das colisões [4]. Caracterizado por sua geometria cilíndrica, possui uma estrutura composta por quatro subsistemas, dentre eles os calorímetros eletromagnéticos e hadrônicos, destinados à medição da energia depositada pelas partículas [5]. A disposição em camadas destes subsistemas permite uma reconstrução abrangente dos eventos de colisão.

A operação do ATLAS em condições de alta taxa de eventos apresenta desafios significativos. Identificar elétrons resultantes das colisões (e quaisquer decaimentos subsequentes) é uma tarefa complexa, pois sua faixa de energia é dominada por um ruído de fundo produzido por jatos hadrônicos (cone de partículas produzido pela interação hadrônica de quarks e glúons) [6], os quais geram um volume substancial de dados não pertinentes [7]. Com colisões ocorrendo a cada 25 nanossegundos, o volume de dados pode alcançar aproximadamente 70 TB/s, o que impede o armazenamento de todos os eventos [7]. Portanto, para armazenar apenas partículas de interesse, um sistema de filtragem *online* (*Trigger*) é empregado [2].

No programa de atualizações do LHC para o período de coleta de dados entre 2030 - 2034 (*Run-4*), está previsto o aumento do número médio de interações por colisão para 200, produzindo um alto empilhamento de sinais. Visando manter a eficiência de detecção em tais condições de operação [8], o *Trigger and Data Acquisition* (TDAQ), sistema de seleção e aquisição de dados do ATLAS, será aprimorado no *High Luminosity LHC* (HL-LHC). A principal inovação será a introdução do *Global Trigger*, cujo núcleo de processamento, o *Global Event Processor* (GEP), executará algoritmos de seleção de partículas diretamente em *hardware* digital, usando *Field Programmable Gate Array* (FPGA), operando com uma taxa de saída de 1 MHz.

Nesse contexto, esse trabalho propõe um estudo sobre os impactos da implementação em *hardware* programável de um método de classificação online de elétrons baseado em aprendizado de máquina, chamado *NeuralRinger* [9], no *Global Trigger*, explorando formas de quantizar o algoritmo.

*Laboratório de Processamento de Sinais (COPPE/Poli, UFRJ), Rio de Janeiro - RJ, e-mail: gabriel.lisboa@lps.ufrj.br, jodafons@lps.ufrj.br
Laboratório de Sistemas Digitais (PPGEEC/UFBA), Salvador - BA, e-mail: edmar.egidio@ufba.br *Brookhaven National Laboratory, NY, US. O presente trabalho foi realizado com apoio da Coordenação de Aperfeiçoamento de Pessoal de Nível Superior - Brasil (CAPES) - Código de Financiamento 001. Os autores desse trabalho também gostariam de agradecer às instituições de fomento nacionais CNPq, FAPERJ, FAPESB, RENAFAE e também ao CERN e à colaboração ATLAS pelo suporte e incentivo nesse desenvolvimento e aos futuros.

¹O (eV) é uma unidade definida como a quantidade de energia cinética ganha por um elétron quando submetido a uma diferença de potencial de 1V no vácuo.

II. ATUALIZAÇÃO DO TRIGGER DO ATLAS

O TDAQ será atualizado para as novas condições de operação da *Run-4* (Figura 1) com um novo nível zero (L0), desenvolvido em FPGA. Esse nível realizará a leitura dos sinais dos calorímetros para classificação dos eventos. A janela de tomada de decisão do *trigger* deverá operar em até 10 μ s. Em seguida, o resultado da identificação dos eventos e os dados do detector são transmitidos através do Sistema de Aquisição de Dados (DAQ) a 1 MHz para o *Event Filter* (EF) (composto por um conjunto de processadores), responsável por reduzir a taxa de eventos para 10 kHz. Os eventos selecionados são então armazenados em mídia permanente.

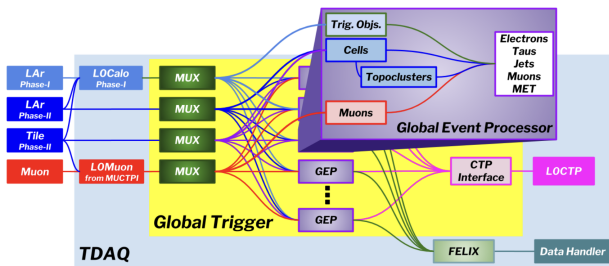


Fig. 1: Diagrama de blocos do novo sistema de *trigger* do ATLAS para a *Run-4* [10].

A. NeuralRinger

Para a identificação de partículas eletromagnéticas, como elétrons e fótons, no sistema de *Trigger* do ATLAS, emprega-se o algoritmo de classificação *NeuralRinger* [9]. A estratégia deste algoritmo reside na construção de anéis concêntricos que capturam a geometria da distribuição de energia de um chuveiro de partículas, que são, então, avaliados por uma rede neural para classificação. A Figura 2 ilustra o fluxo do *NeuralRinger* para a *Run 4* em duas etapas: *Feature Extraction* (FEX) e *Hypothesis* (HYPO).

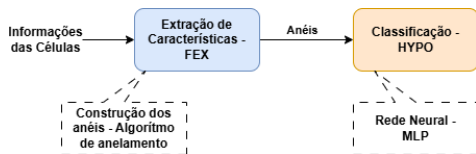


Fig. 2: Diagrama de Blocos do NeuralRinger.

Na primeira etapa (FEX) os anéis são construídos conforme a Figura 3. O retângulo vermelho representa o centro do chuveiro de partículas a ser classificado, e cada anel captura a energia depositada a certa distância do centro, o que permite captar a geometria da deposição energética. Essa geometria permite a diferenciação de classes, como mostra a Figura 4, que compara o perfil de jatos (ruído) e elétrons (sinal).

Com os anéis construídos, a segunda etapa realiza a classificação utilizando esses dados com um modelo baseado em redes neurais. Atualmente, o modelo consiste em um comitê de especialistas de *Multilayer Perceptron* (MLP) [12]. Cada membro do comitê é treinado para classificar amostras em uma certa região do detector em um intervalo da energia total de tal

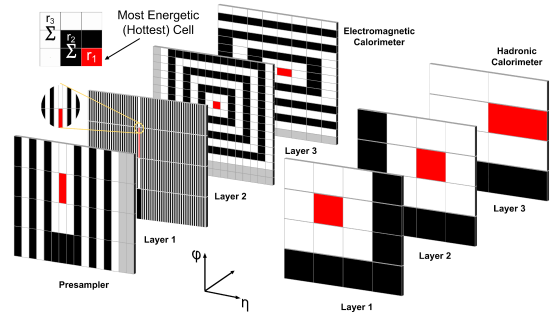


Fig. 3: Ilustração do processo de anelamento para os calorímetros eletromagnético e hadrônico. Fonte: [11].

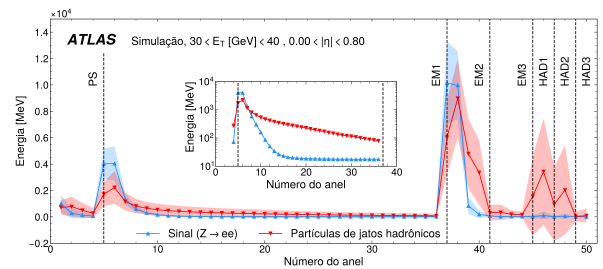


Fig. 4: Perfil médio dos sinais formatados em anéis, coletados pelos calorímetros.

forma que, quando juntos, cobrem todo o espectro de energia e a área do detector. Isso é feito pois ocorrem diferenças expressivas na dinâmica do evento em função da energia e na granularidade das células em função da posição.

B. Quantização e Quantization-Aware Training

Redes neurais são tipicamente treinadas em ponto flutuante de 32 bits (FP32), inviabilizando sua implantação direta em ambientes com altas restrições de recursos [13]. Para adequar essas arquiteturas, emprega-se a quantização, reduzindo a largura de bits dos pesos e das ativações da rede para formatos de menor resolução [13]. Uma possível abordagem é a Quantização Pós-Treinamento (PTQ), que reduz a resolução dos parâmetros sem necessidade de retreinamento [14], substituindo operações de ponto flutuante por aritmética de inteiros ou ponto fixo. Essa conversão é essencial em hardwares como as FPGAs [15], pois evita a monopolização de recursos lógicos (*Look-Up Table* (LUT) e *Digital Signal Processor* (DSP)) e reduz a complexidade computacional do sistema [16].

Apesar do PTQ apresentar resultados expressivos na redução do custo computacional, sua aplicação pode levar à degradação da eficiência do modelo, por ter sido otimizado sob o pressuposto de alta precisão [15]. A discretização de seus parâmetros introduz um ruído no mapeamento de entrada e saída, podendo afetar a propagação da informação nas camadas da rede. Para superar essa limitação, a literatura adota a abordagem do *Quantization Aware Training* (QAT) [17].

No QAT a rede neural é exposta aos efeitos da quantização durante a fase de treinamento. Os pesos, ativações e sinais de entrada são mapeados para intervalos discretos, simulando o comportamento que ocorreria no *hardware*. Dessa forma,

a saída da rede e o erro calculado pela função custo já contemplam as perdas de informação causadas pela redução de bits, e a rede é capaz de ajustar seus pesos para compensar os erros de arredondamento provenientes da quantização [17].

Para viabilizar a propagação de gradientes através de funções de quantização não deriváveis, o QAT utiliza o *Straight-Through Estimator* (STE) [18]. Ele atua no *backward pass* realizando o *passthrough* das operações discretas da quantização ao definir suas derivadas como a identidade. Embora o erro seja calculado sobre as previsões geradas pelos pesos quantizados, as atualizações são aplicadas a uma cópia dos parâmetros mantida em ponto flutuante. Esse mecanismo evita que os pesos fiquem estagnados em um mesmo nível de discretização (o que causaria o desaparecimento do gradiente) e permite que o modelo aprenda a contornar o ruído de quantização, adaptando sua distribuição de pesos para minimizar a degradação durante a inferência em *hardware* [17].

III. METODOLOGIA

Este trabalho foca na implementação da etapa *HYPO* do *NeuralRinger* em *hardware*, realizando um estudo de comparação entre dois modelos que utilizam métodos distintos de quantização (PTQ e QAT), utilizando como *baseline* um modelo em software (ponto flutuante) baseado no modelo desenvolvido por Ferreira *et al.* [19] para o *NeuralRinger*. A implementação da etapa *FEX*, junto com uma versão inicial da *HYPO* foram desenvolvidas em um trabalho anterior [20].

A. Arquitetura de hardware

Os especialistas do comitê de redes neurais compartilham a mesma arquitetura: uma camada oculta com cinco neurônios com função de ativação ReLU e um neurônio de saída para classificação binária (por meio dos anéis construídos na *FEX*) com função de ativação sigmóide.

Essa topologia de rede foi selecionada após uma série de estudos comparativos envolvendo diferentes arquiteturas e configurações [7]. O principal objetivo foi estabelecer um equilíbrio entre a eficiência na identificação de elétrons e a necessidade de execução rápida, uma vez que o algoritmo é operado no ambiente do *High-Level Trigger* (HLT) do ATLAS, que impõe severas restrições de latência. A configuração atual está em uso no sistema de *trigger* online desde 2017 [7].

A Figura 5 apresenta um diagrama da implementação da rede em FPGA, junto com um bloco responsável pela seleção e armazenamento dos parâmetros dos diferentes especialistas, implementado com um registrador com 261 posições.

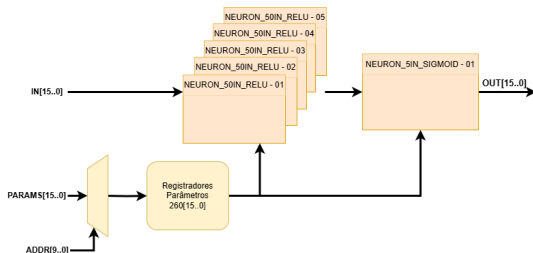


Fig. 5: Diagrama de blocos da *HYPO*.

Para reduzir o esforço de desenvolvimento e recursos computacionais na implementação da função sigmóide em *hardware*, adotou-se uma modelagem baseada em uma aproximação linear por partes da sigmóide, conforme a Figura 6.

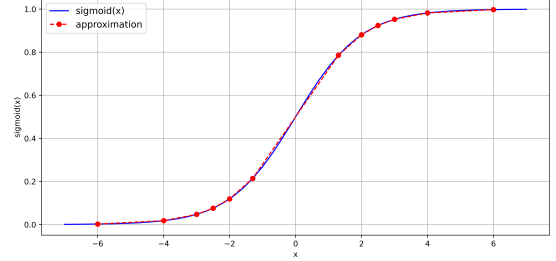


Fig. 6: Aproximação linear por partes da função sigmóide.

B. Modelo com Quantização Pós-Treino

Com o objetivo de reduzir o custo computacional do circuito implementado, a MLP desenvolvida neste estudo foi projetada utilizando quantização em ponto fixo no formato Qm.n (m bits inteiros, n bits decimais e 1 bit de sinal) gerando a aproximação de pesos da rede $\tilde{w}$ de acordo com a equação 2. As entradas foram codificadas em formato Q15 (16 bits) e os pesos em formato Q7.15 (23 bits). A maior largura de palavra dos pesos foi necessária para acomodar operações intermediárias com maior precisão nos neurônios. A escolha desses formatos permitiu o alinhamento eficiente dos operadores aritméticos, simplificando a lógica de implementação dos neurônios em Linguagem de Descrição de *Hardware* (HDL).

$$y = w \cdot x \quad (1)$$

$$\tilde{w} = \text{clip}(\text{int}(w \cdot 2^n), -2^m, 2^m - 2^{-n}) \quad (2)$$

$$\tilde{y} = \tilde{w} \cdot x \quad (3)$$

C. Modelo com Quantization-Aware Training

Para comparação com o método de PTQ, foi treinado um modelo alternativo utilizando a técnica de QAT com STE. Nesse modelo, também foi aplicada a quantização no formato Q7.15 nos parâmetros, e Q15 para os dados de entrada. Após o treinamento, os parâmetros foram transferidos para o circuito da FPGA para simulação da inferência em HDL.

D. Figuras de Mérito e Limiares de Decisão

Foi utilizada como figura de mérito principal para o treinamento o Índice Soma-Produto (SP) (Equação 4), que estima a relação entre a Probabilidade de Detecção (PD) e a Taxa de Falso Alarme (FA).

$$SP = \sqrt{\sqrt{PD \cdot (1 - FA)} \cdot \frac{PD + (1 - FA)}{2}} \quad (4)$$

A PD ($\frac{TP}{TP+FN}$) é um indicador de eficiência na classificação de elétrons, enquanto a FA ($\frac{FP}{FP+TN}$) representa a taxa de aceitação de jatos hadrônicos classificados como elétrons. TP são os verdadeiros positivos (elétrons corretamente classificados), FN são os falsos negativos (elétrons classificados como

jatos hadrônicos), FP são os falsos positivos (jatos hadrônicos classificados como elétrons) e TN são os verdadeiros negativos (jatos hadrônicos corretamente descartados).

Para os limiares de decisão das redes, são definidos quatro pontos de operação para os modelos, refletindo diferentes níveis de pureza de dados, do mais restritivo ao mais brando: *Tight*, *Medium*, *Loose* e *Very Loose*). Cada um possui uma taxa de detecção de elétrons pré definida pelo experimento e o FA é aquele obtido por consequência.

E. Descrição dos Dados e Modelos

A arquitetura do modelo tradicional [19] foi preservada. Assim, utilizamos modelos especializados em 5 intervalos específicos de E_T (energia transversa) e 5 de η (pseudorapidez), totalizando 25 regiões. São utilizados 50 anéis por evento, onde para a montagem dos anéis, foram consideradas apenas as células com energia acima de 2σ do ruído eletrônico e do empilhamento. As redes neurais foram treinadas e avaliadas com o mesmo conjunto de dados, utilizando a validação cruzada *K-Fold*, com 5 folds (4 para treino e 1 para validação). A função custo (*Loss Function*) utilizada foi a Entropia Cruzada Binária [12], e o critério de parada do treinamento foi o valor máximo do SP de validação, com paciência de 25 épocas.

IV. RESULTADOS

Primeiramente, foi realizada uma análise do comportamento do treinamento do modelo QAT. A Figura 7 mostra a curva de perda para o treinamento e validação do QAT e do *Baseline*. É possível notar que o método QAT exige mais épocas de treinamento, se comparado ao *baseline*. Isso pode ser explicado pelo ruído introduzido pelo STE, onde o gradiente aproximado pode causar oscilações que impedem o modelo de encontrar um mínimo tão estável quanto o *baseline*.

Para analisar a variação entre os resultados do treinamento do modelo QAT em relação ao *baseline* em ponto flutuante, foi calculado o erro entre as figuras de mérito (SP, PD e FA) desses dois modelos, para cada ponto de operação. Definimos os modelos para atingirem a mesma eficiência de sinal do modelo de referência do *Trigger* atual. Isso resulta em um valor de SP e FA específicos para cada modelo. A Tabela I mostra esses resultados para o ponto de operação *Medium*, o segundo mais restritivo. Observa-se que a perda apresentada pelo modelo quantizado é mínima, e em apenas algumas regiões o FA atinge diferenças acima de 1 pp em relação ao *baseline*. A maior diferença observada acontece em uma região onde o detector apresenta menor resolução e menor quantidade de sensores ($1.37 \leq \eta < 1.54$) devido a passagem de cabos de instrumentação. Os demais pontos de operação seguem o mesmo padrão.

Na Tabela II, estão listados os resultados da simulação da inferência em FPGA dos dois modelos quantizados (PTQ e QAT) e os resultados da inferência do *baseline* em software. A comparação foi feita por meio dos valores de PD e FA calculados em cada um dos 4 pontos de operação.

Observa-se que ambos os modelos obtiveram resultados semelhantes ao *baseline*, com o PTQ igualando o modelo em ponto flutuante. Logo, o erro introduzido pela quantização foi

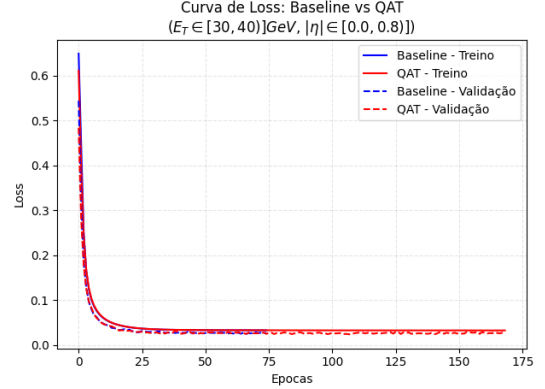


Fig. 7: Curvas de treino e validação para os modelos *Baseline* e QAT para a região de $E_T = [30, 40)$ GeV e $\eta = [0, 0.8)$.

TABELA I: Erro das figuras de mérito entre modelos *baseline* x QAT por faixa de E_T e η para o ponto de operação *Medium* (Valores de SP, PD e FA em pp).

E_T [GeV]	$ \eta $				
	[0.00, 0.80)	[0.80, 1.37)	[1.37, 1.54)	[1.54, 2.37)	[2.37, 2.50)
[15, 20)	SP: -0.39	SP: 0.11	SP: -0.39	SP: 0.23	SP: 0.00
	PD: 0.10	PD: 0.05	PD: 0.20	PD: 0.11	PD: 0.00
	FA: 0.87	FA: -0.17	FA: 0.94	FA: -0.34	FA: 0.00
[20, 30)	SP: -0.09	SP: -0.24	SP: -1.49	SP: -0.06	SP: -0.10
	PD: 0.10	PD: 0.10	PD: 0.10	PD: 0.10	PD: 0.17
	FA: 0.28	FA: 0.57	FA: 2.95	FA: 0.21	FA: 0.36
[30, 40)	SP: -0.02	SP: -0.12	SP: -0.55	SP: -0.14	SP: -0.15
	PD: 0.10	PD: 0.10	PD: 0.11	PD: 0.10	PD: 0.12
	FA: 0.13	FA: 0.34	FA: 1.17	FA: 0.37	FA: 0.42
[40, 50)	SP: -0.00	SP: -0.06	SP: -0.71	SP: -0.08	SP: -0.50
	PD: 0.10	PD: 0.10	PD: 0.10	PD: 0.10	PD: 0.11
	FA: 0.10	FA: 0.22	FA: 1.50	FA: 0.25	FA: 1.11
[50, ∞)	SP: 0.02	SP: -0.06	SP: -1.10	SP: -0.04	SP: 0.55
	PD: 0.10	PD: 0.11	PD: 0.11	PD: 0.10	PD: 0.11
	FA: 0.07	FA: 0.22	FA: 2.28	FA: 0.19	FA: -1.00

praticamente nulo, visto que o PTQ parte do modelo *baseline* até ser quantizado. Por outro lado, o QAT resultou em valores ligeiramente menores nas figuras de mérito, mas pequeno o suficiente para assumir que essa diferença é provocada pela incerteza relacionada ao retreinamento da rede. Uma análise mais detalhada pode ser vista na Figura 8, que mostra a distribuição da saída dos diferentes modelos. As saídas dos modelos PTQ e QAT seguem aproximadamente a mesma distribuição do *baseline*, e as razões das distribuições de saída dos modelos quantizados em relação ao ponto flutuante estão em torno de 1, demonstrando que nenhum dos modelos apresentou viés considerável em relação ao *baseline*. Os resultados encontrados podem ser justificados pela alta resolução de bits adotada no experimento, que tornou o erro de quantização baixo, preservando a fronteira de decisão original do *baseline*.

V. CONCLUSÕES

Este trabalho apresentou a implementação em FPGA do algoritmo *NeuralRinger* para a atualização do sistema de *Trigger* do ATLAS, comparando o impacto das técnicas de PTQ e QAT. Os resultados demonstraram que a quantização

TABELA II: Comparação de Eficiência *Baseline* vs PTQ vs QAT

Ponto Op.	PD %			FA %		
	<i>Baseline</i>	PTQ	QAT	<i>Baseline</i>	PTQ	QAT
<i>Very Loose</i>	99.80	99.80	99.80	9.62	9.62	9.51
<i>Loose</i>	99.70	99.70	99.70	7.71	7.71	7.45
<i>Medium</i>	99.60	99.60	99.60	6.65	6.65	6.54
<i>Tight</i>	99.50	99.50	99.50	5.94	5.94	5.87

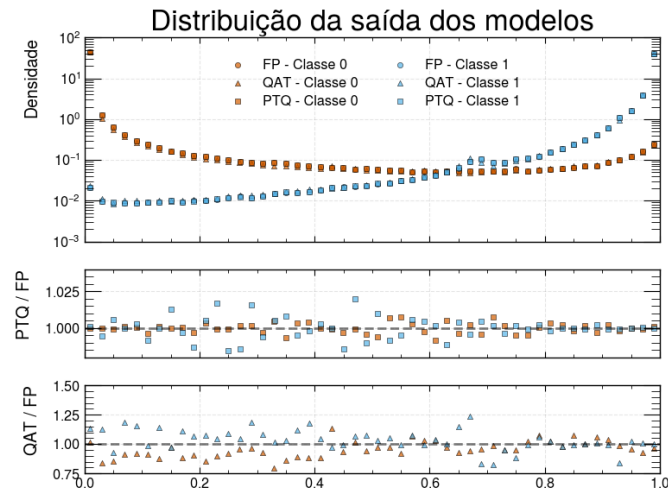


Fig. 8: Comparação da distribuição da saída dos modelos para as classes jato hadrônico e elétron, 0 (laranja) e 1 (azul) respectivamente. Os eixos inferiores mostram a razão dos intervalos dos histogramas dos modelos PTQ e QAT desenvolvidos em relação a referência em ponto flutuante FP.

em Q7.15 dos parâmetros da rede, e em Q15 para os anéis, é viável para a operação do algoritmo no *Global Trigger*.

Os resultados mostraram que, para a quantização utilizada neste trabalho, a técnica de PTQ e QAT apresentaram desempenhos equiparáveis ao *baseline* em todos os pontos de operação. Comparando exclusivamente a degradação de eficiência do QAT em relação ao *baseline*, apenas em algumas regiões do detector foi observada uma degradação considerável causada pela quantização, com o valor máximo de erro no FA sendo de 2.95%. Conclui-se que, para redes neurais de baixa complexidade como as MLPs do comitê, a quantização dos parâmetros de modelos com ambas as técnicas utilizadas preservam a fronteira de decisão do modelo, sendo abordagens promissoras para a implementação final na *Run-4* do LHC.

Os resultados obtidos com a alta precisão abrem caminho para quantizações menores do modelo. Visto que não foi observada degradação perceptível com a configuração utilizada, o próximo passo é investigar outras resoluções menores, cenário em que o erro de quantização torna-se consideravelmente maior e o QAT pode superar o PTQ por conseguir adaptar a topologia da rede a esse ruído.

REFERÊNCIAS

[1] W. N. Cottingham and D. A. Greenwood, *An Introduction to the Standard Model of Particle Physics*, 2nd ed. Cambridge University Press, Jul. 2023.

[2] ATLAS Collaboration, "The ATLAS Experiment at the Large Hadron Collider," CERN, Tech. Rep. ATLAS-31-07-2019, July 2019, Overview document from ATLAS Rome Group. [Online]. Available: <https://www.roma1.infn.it/exp/atlas/>

[3] "LHC Machine," *JINST*, vol. 3, p. S08001, 2008.

[4] A. Collaboration, "The ATLAS experiment at the CERN Large Hadron Collider: a description of the detector configuration for Run 3," *J. Instrum.*, vol. 19, no. 05, p. P05063, may 2024.

[5] A. Airapetian *et al.*, "ATLAS calorimeter performance technical design report," ATLAS Collaboration, Tech. Rep., 12 1996.

[6] "Review of particle physics," *Phys. Rev. D*, vol. 110, p. 030001, Aug 2024. [Online]. Available: <https://link.aps.org/doi/10.1103/PhysRevD.110.030001>

[7] J. V. D. F. Pinto, "Filtragem Online Segmentada Baseada em Redes Neurais operando na informação de um calorímetro de altas energias de fina granularidade," Ph.D. dissertation, Universidade Federal do Rio de Janeiro - UFRJ, Apr. 2022.

[8] V. A. Ferraz, "Detecção online eficiente de eventos raros utilizado detectores finamente segmentados," Ph.D. dissertation, COPPE/Universidade Federal do Rio de Janeiro (UFRJ), Rio de Janeiro, Brazil, 2018.

[9] J. M. Seixas, L. P. Calôba, M. N. Souza, A. L. Braga, and A. P. Rodrigues, "Neural second-level trigger system based on calorimetry," 1995. [Online]. Available: <https://cds.cern.ch/record/278995>

[10] "Technical Design Report for the Phase-II Upgrade of the ATLAS TDAQ System," CERN, Geneva, Tech. Rep., 2017. [Online]. Available: <https://cds.cern.ch/record/2285584>

[11] G. Aad *et al.*, "Performance of electron and photon triggers in ATLAS during LHC Run 2," *Eur. Phys. J. C*, vol. 80, no. 1, p. 47, 2020.

[12] C. Bishop, *Pattern recognition and machine learning*. Springer New York, 2006, vol. 4.

[13] B. Jacob, S. Kligys, B. Chen, M. Zhu, M. Tang, A. Howard, H. Adam, and D. Kalenichenko, "Quantization and training of neural networks for efficient integer-arithmetic-only inference," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, June 2018.

[14] M. Nagel, M. Fournarakis, R. A. Amjad, Y. Bondarenko, M. van Baalen, and T. Blankevoort, "A white paper on neural network quantization," *arXiv preprint arXiv:2106.08295*, 2021, qualcomm AI Research.

[15] H. Wu, P. Judd, X. Zhang, M. Isaev, and P. Micikevicius, "Integer quantization for deep learning inference: Principles and empirical evaluation," 2020. [Online]. Available: <https://arxiv.org/abs/2004.09602>

[16] F. Kordi, P. Fortier, and A. Miled, "Optimized fixed point mac unit for neural network on fpga," in *2024 International Conference on Microelectronics (ICM)*, Dec 2024, pp. 1–5.

[17] M. Kirtas, A. Oikonomou, N. Passalis, G. Mourgiaris-Alexandris, M. Moralis-Pegios, N. Pleros, and A. Tefas, "Quantization-aware training for low precision photonic neural networks," *Neural Networks*, vol. 155, pp. 561–573, 2022.

[18] C. Sun, Z. Que, T. Aarrestad, V. Loncar, J. Ngadiuba, W. Luk, and M. Spiropulu, "Hqg: High granularity quantization for real-time neural networks on fpgas," in *Proceedings of the 2026 ACM/SIGDA International Symposium on Field Programmable Gate Arrays*, ser. FPGA '26. New York, NY, USA: Association for Computing Machinery, 2026, p. 79–91. [Online]. Available: <https://doi.org/10.1145/3748173.3779200>

[19] I. S. Ferreira, J. M. de Seixas, E. E. P. D. Souza, and D. O. Damazio, "Conjunto de redes neurais para seleção online de elétrons em um calorímetro finamente segmentado e com elevada taxa de empilhamento de sinais," in *Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)*, F. G. aes, D. Ferreira, and G. Barreto, Eds. Belo Horizonte, MG: SBIC, 2025, pp. 1–8.

[20] I. A. D. S. Luz, G. H. B. Lisboa, N. P. Neto *et al.*, "Redes neurais embarcadas em fpga para classificação online de elétrons em um calorímetro de altas energias," in *Anais do XVII Congresso Brasileiro de Inteligência Computacional (CBIC 2025)*, F. G. aes, D. Ferreira, and G. Barreto, Eds. Belo Horizonte, MG: SBIC, 2025, pp. 1–8.