Algorithm 1 Low-Latency Logistic Regression with AdamW

Precompute: `sigmoidTable` ▷ 32 KB LUT in L2 cache
Initialize: $\mathbf{w}_0, \mathbf{x}_0$
$\hat{\mathbf{w}}_0 \leftarrow \texttt{quantize8.8}(\mathbf{w}_0)$
$\hat{\mathbf{x}}_0 \leftarrow \texttt{quantize8.8}(\mathbf{x}_0)$
$\hat{y}_0 \leftarrow \texttt{sigmoidApprox}(\hat{\mathbf{w}}_0 \cdot \hat{\mathbf{x}}_0 \gg 8)$
$\Delta_0 \leftarrow \texttt{getDelta}(\hat{y}_0, \mathbf{w}_0, \mathbf{x}_0)$
for $i = 1, 2, \ldots$ **do**
    Load $\mathbf{x}_i$
    $\mathbf{w}_i \leftarrow \delta \mathbf{w}_{i-1} - \Delta_{i-1}$
    $\hat{\mathbf{w}}_i \leftarrow \texttt{quantize8.8}(\mathbf{w}_i)$
    $\hat{\mathbf{x}}_i \leftarrow \texttt{quantize8.8}(\mathbf{x}_i)$
    $\hat{y}_i \leftarrow \texttt{sigmoidApprox}(\hat{\mathbf{w}}_i \cdot \hat{\mathbf{x}}_i \gg 8)$
    $\Delta_i \leftarrow \texttt{getDelta}(\hat{y}_i, \mathbf{w}_i, \mathbf{x}_i)$
end for
procedure QUANTIZE8.8($\mathbf{v}$)
    for $v_i \in \mathbf{v}$ **do** ▷ AVX2 Vectorized
        Load $\mathbf{v}[i : i + 32]$
        $v_i \leftarrow \texttt{clamp}(v_i, -128, 127.994)$ ▷ Branchless Clamping
        $v_i \leftarrow v_i \ll 8$ ▷ Convert to 8.8 fixed-point
    end for
    return $\mathbf{v}$
end procedure
procedure SIGMOIDAPPROX(n)
    $i \leftarrow \texttt{clamp}(n, -1024, 1024)$
    $s \leftarrow \texttt{sigmoidTable}[i]$ ▷ Precomputed Lookup
    return s
end procedure
procedure GETDELTA($\hat{y}_i, \mathbf{w}_i, \mathbf{x}_i$)
    Load y_i
    $\mathbf{g}_i \leftarrow \hat{y}_i \mathbf{x}_i - y_i \mathbf{x}_i$
    Load $\mathbf{m}_{i-1}, \mathbf{v}_{i-1}$
    $\mathbf{m}_i \leftarrow \beta_1 \mathbf{m}_{i-1} + (1 - \beta_1)\mathbf{g}_i$
    $\mathbf{v}_i \leftarrow \beta_2 \mathbf{v}_{i-1} + (1 - \beta_2)\mathbf{g}_i^2$
    $\widehat{\mathbf{m}}_i \leftarrow \mathbf{m}_{i-1}/(1 - \beta_1^i) \cdot (i < 54) + \mathbf{m}_{i-1} \cdot (i \geq 54)$ ▷ Branchless Masking
    $\widehat{\mathbf{v}}_i \leftarrow \mathbf{v}_{i-1}/(1 - \beta_2^i) \cdot (i < 5544) + \mathbf{v}_{i-1} \cdot (i \geq 5544)$
    $\beta_1^{i+1} \leftarrow \beta_1^i \times \beta_1$
    $\beta_2^{i+1} \leftarrow \beta_2^i \times \beta_2$
    return $\gamma \widehat{\mathbf{m}}_i \cdot \texttt{rsqrt}(\widehat{\mathbf{v}}_i + \epsilon)$ ▷ AVX InvSqrt approximation
end procedure