This homework contains some questions and problems marked with a star. These are intended for graduate students enrolled in the 598NSG section. Undergraduates enrolled in the 498NSU section are welcome to solve them for extra credit. The topics covered in this homework include: DNN complexity estimation in fixed-point, fixed-point inference in Python, back-propagation for non-linear systems, linear prediction in fixed-point, SGD-based linear prediction, SGD with delayed updates. This homework needs to be completed in two parts. Part 1: complete any 3 problems (2 problems for students in 498 section) by 5pm Friday 09/25 and Part 2: the rest (4 problems) by 5pm Friday 10/02. No extension will be provided, so make sure to get started as soon as possible

Problem 1: SQNR of a Dot Product due to Output Quantization

A N-dimensional dot product given by

$$y = \sum_{j=1}^{N} w_j x_j \tag{1}$$

where $x_j \in [-x_m, x_m]$ and $w_j \in [-w_m, w_m]$ are uncorrelated RVs with zero mean and variances of σ_x^2 and σ_w^2, respectively. Derive the analytical expression for the SQNR due to quantization of y given below:

$$\text{SQNR}_{q_y(\text{dB})} = 6B_y + 4.8 - [\zeta_{x(\text{dB})} + \zeta_{w(\text{dB})}] - 10\log_{10}(N) \tag{2}$$

where $\zeta_x = \frac{x_m}{\sigma_x}$ and $\zeta_w = \frac{w_m}{\sigma_w}$.

Problem 2: Neural Network Complexity Estimation in Fixed-Point

In this problem, you are asked to once again consider the ResNet-18 and VGG-11 networks discussed in HWK 1 Problem 3. You are always encouraged to write scripts to solve the problem. What are the representational and computational costs of both networks assuming a uniform precision assignment of 10 bits?

Problem 3: Fixed-Point Inference with Python

In this problem, we will perform fixed-point inference using the CIFAR-10 network we had prepared in Homework 1 Problem 4. In the newly provided zip file, you will find source code from the last homework along with additional files for dynamic ranges and precisions. These new files contain data that may be helpful to complete this problem.

1. Determine the minimum uniform precision requirement needed to maintain a test accuracy within 1% of the floating-point baseline. Note: in order to quantize weights and activations, you need to set their dynamic ranges, these can be found in the file provided on the course website. (HINT: activations are unsigned and weights are signed.)

2. Using the *precision offsets* from the provided file on the course website, determine the minimum per-layer precision requirements needed to maintain a test accuracy within 1% of the floating-point baseline.

Note: the per-layer precision assignments were obtained via the noise equalization method we have discussed in class.

3. Compare the corresponding representational and computational costs of precision assignments in Part 1 and 2.

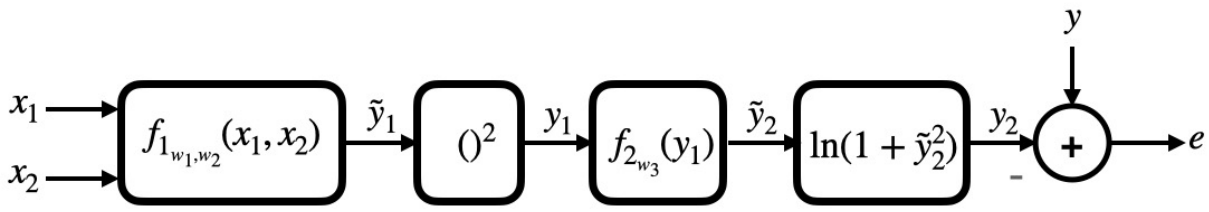

Figure 1: The non-linear system for Problem 4

Problem 4: Back-propagation for Non-linear System

Consider the non-linear system in Figure 1 where the goal is to predict the label y given an input (x_1, x_2). The system is parametrized by three weights (w_1, w_2, w_3). The following are the equations describing the feedforward computations of the systems:

$$\tilde{y}_1 = w_1 x_1 + w_2 x_2$$

$$y_1 = \tilde{y}_1^2$$

$$\tilde{y}_2 = w_3 y_1$$

$$y_2 = \ln(1 + \tilde{y_2}^2)$$

It is desired to minimize the mean-squared error

$$C = \mathbb{E}[e^2] = \mathbb{E}[(y - y_2)^2]$$

Assuming (x_1, x_2) are coordinates uniformly distributed over the unit circle, i.e. $x_1^2 + x_2^2 = 1$. The system is trying to predict its angle $y \in [-\pi, \pi]$ where

$$y = \begin{cases} \arctan(\frac{x_2}{x_1}), & x_1 > 0 \\ \pi + \arctan(\frac{x_2}{x_1}) & x_1 < 0 \text{ and } x_2 > 0 \\ -\pi + \arctan(\frac{x_2}{x_1}) & x_1 < 0 \text{ and } x_2 < 0 \\ \pi/2 & x_1 = 0 \text{ and } x_2 > 0 \\ -\pi/2 & x_1 = 0 \text{ and } x_2 < 0 \end{cases}$$

1. Derive expressions for the following three gradients: $\frac{\partial C}{\partial w_1}$, $\frac{\partial C}{\partial w_2}$ and $\frac{\partial C}{\partial w_3}$. Simplify as much as you can. (HINT: use the chain rule and LOTUS).

2. Derive closed form expressions for the following three instantaneous gradients: $\frac{\partial e^2}{\partial w_1}$, $\frac{\partial e^2}{\partial w_2}$ and $\frac{\partial e^2}{\partial w_3}$.
3. Using derivation in part 2, implement the SGD algorithm applied to this setup. Plot the convergence curves and the evolutions of the weights (w_1, w_2, w_3).
4. * Analytically determine the number of bits B_{w1} needed to quantize w_1 such that the MSE C increases by no more than 1% over its floating-point value reached in Part 3. Assume that the quantization noise appears in an additive form at the output. [Hint: use the expression for the gradient from Part 1 as the quantization noise gain for w_1.]
5. * Validate your answer in Part 4 by running the nonlinear system with w_1 quantized to B_{w1} bits.
6. This system is a non-convex system and therefore has many local minima. To see this, can you find a solution that is different from the one in Part 3?

Problem 5: Fixed-Point Linear Predictor *

Calculate the minimum precision of the input and coefficient for the 3-tap predictor designed in Homework 1 Problem 5 such that the fixed-point model achieves an SNR that is within $0.1dB$ of the floating-point SNR ($=10\log_{10}\frac{\sigma_x^2}{\sigma_e^2}$ where σ_x^2 are the variances of the input and error, respectively). Estimate the computational cost of this precision assignment. Validate your answer by simulating the fixed-point and floating-point predictors.

Problem 6: SGD-based Linear Predictor for Time Series

Code a 3-tap SGD-based linear predictor for predicting x_n from Homework 1 Problem 5. The coefficients of this online predictor should approach the optimal ones previously determined.

1. Calculate the maximum value of the learning rate μ_m (stability bounds) for which convergence is guaranteed.
2. Plot the convergence behavior of the MSE and coefficients for $\mu = 0.5\mu_m$. Remember to use ensemble average plots. Compares these with the MSE and coefficients obtained in Homework 1 Problem 5. Repeat this part for $\mu = 0.01\mu_m$ and $\mu = 50\mu_m$.
3. Sketch an architecture and calculate its critical path delay, latency, and throughput in terms of the adder delay (T_A) and the multiplier delay (T_M).

Problem 7: SGD with Delayed Updates *

The conventional SGD algorithm can be modified by delaying the weight vector update by M_1

iterations as follows:

$$\mathbf{w}_n \leftarrow \mathbf{w}_{n-1} + \mu \nabla_{\mathbf{w}} \left(e_{n-M_1}^2 \right)$$

1. Code a 3-tap SGD-based linear predictor for predicting x_n as in Homework 1 Problem 5. Assume $M_1 = 5$. Plot its convergence behavior of the coefficients.

2. Sweep the learning rate μ to see if you can find a case where the original SGD algorithm converges but the delayed one does not.

3. Determine the minimum value of M_1 such that the critical path delay of the pipelined architecture is equal to one multiplier delay, i.e. $T_{cp} = T_M$. Assume $T_M > T_A$. Draw the pipelined architecture.