Adapting Neural Networks to Fixed-Point and Integer Arithmetic

Hanane BENMAGHNIA

hanane.benmaghnia@etudiant.univ-perp.fr

Supervisors: Matthieu MARTEL and Yasmine SELADJI

LAboratory of Mathematics and Physics (LAMPS)

University of Perpignan Via Domitia

June 20, 2019


Summary

Plan

Introduction

Difficulty to run tasks in real time

Embedded systems have often low memory

Difficulty maintaining efficiency with limited resources

New Problem:

Artificial Intelligence and especially Neural Networks are increasingly used in Embedded Systems !!!

Contribution

Input: A trained NN, working at some precision (Floating-Point Format)

Output: Code generation (the new NN) to "simulate" the original NN using integers and fixed-point arithmetic

Correctness: The new NN with smaller data types behaves almost like the original NN

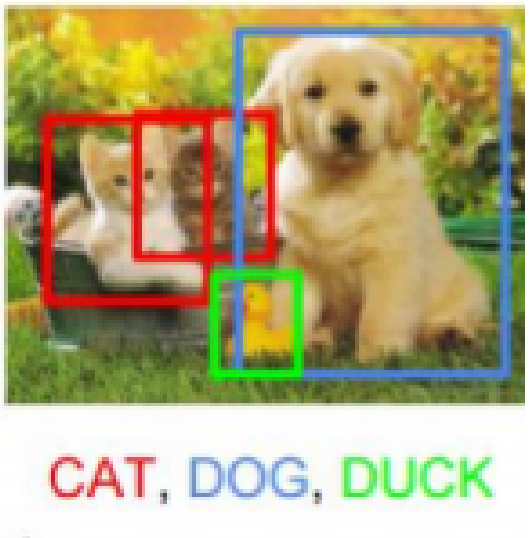
- Function Approximation: results stay in a desired hull
- Classification : only x% decisions differ from original NN

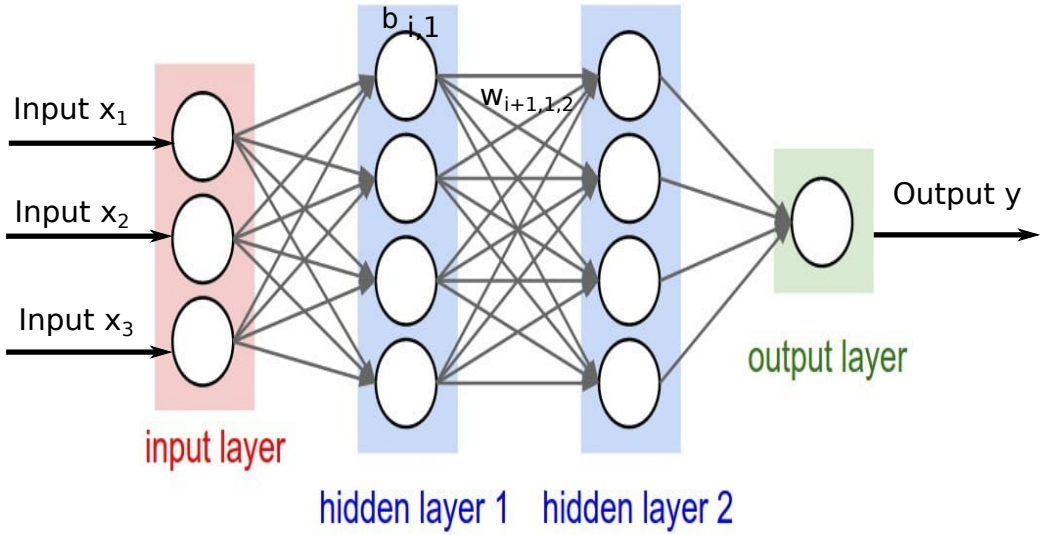
"Simulate": **Same behavior as the original network with x% error**

Plan

Neural Networks (NN)

- The conception is inspired by the biological functioning of neurons
- NN create fast classifications and generalizations
- NN are used on a variety of tasks (object recognition, speech and handwriting recognition, machine translation, medical diagnosis...)





Plan

Fixed-Point Format (FPF)

Float numbers are represented by **integers** in a format <m,l> or <I,F>

Definition

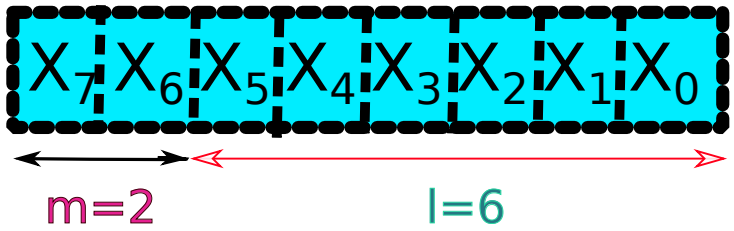
For $X = (x_{n-1}x_{n-2}...x_1x_0)_\beta$ and $x = X.\beta^{-l}$, we have: $p = m + l + 1$.

x : Fixed-Point number with implicit scale factor β^{-l},

X: Integer representation, p : Word **length**, m : The **most** significant bit,

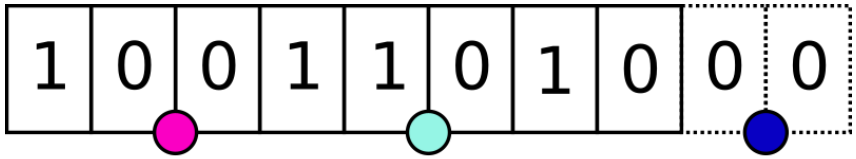
l : The **least** significant bit

Representation of X in a format <m,l>=<2,6>



⚠ **No exponent, the programmer must take care of the formats !**

Different interpretations of the same integer representation



A Fixed-Point number in different format

Scaling Factor Symbol	Format	Value of X	Value of x
purple	< 2, 6 >	$(10011010)_2 = (154)_{10}$	$(10.011010)_2 = (2.40625)_{10}$
cyan	< 5, 3 >	$(10011010)_2 = (154)_{10}$	$(10011.010)_2 = (19.25)_{10}$
blue	< 9, -1>	$(10011010)_2 = (154)_{10}$	$(100110100)_2 = (308)_{10}$

Representation of 154 in different formats

Plan

Plan

Addition

Let : x_1 and x_2 FP numbers with FPF <i, f> :

$$x_1 + x_2 = x_3$$

$$< i, f > \; + \; < i, f > \; = \; < i, \; f > \tag{1}$$

$$\begin{cases} 0 < i < p \\ 0 < i + f < p \\ f = p - i \end{cases} \tag{2}$$

Example :

$$\begin{array}{rl} & 000000.010 \quad <6, 3> \\ + & \\ & 001000.000 \quad <6, 3> \\ \hline & 001000.010 \quad <6, 3> \end{array}$$

Multiplication

Let : x_1 and x_2 FP numbers with FPF <i, f> :

$$x_1 \times x_2 = x_3$$

$$< i, f > \times < i, f > = < i - f, 2f > \tag{3}$$

$$\begin{cases} 0 < i < p \\ 0 < i + f < p \\ f = p - i \end{cases} \tag{4}$$

Shift Right

$$x_1 >> k = x_2$$

$$< i, f > \; >> \; k \; = \; < i + k, \; f - k > \tag{5}$$

$$\begin{cases} 0 < i < p \\ 0 < i + f < p \\ f = p - i \end{cases} \tag{6}$$

Shift Left

$$x_1 << k = x_2$$

$$< i, f > \; << \; k \; = \; < i - k, \; f + k > \tag{7}$$

$$\begin{cases} 0 < i < p \\ 0 < i + f < p \\ f = p - i \end{cases} \tag{8}$$

AND,OR, XOR

$$x_1 \ \ AND \ \ x_2 = x_3$$

$$< i, f > \ AND \ < i, f > \ = \ < i, \ f > \tag{9}$$

$$\begin{cases} 0 < i < p \\ 0 < i + f < p \\ f = p - i \end{cases} \tag{10}$$

Same format for OR and XOR !

Plan

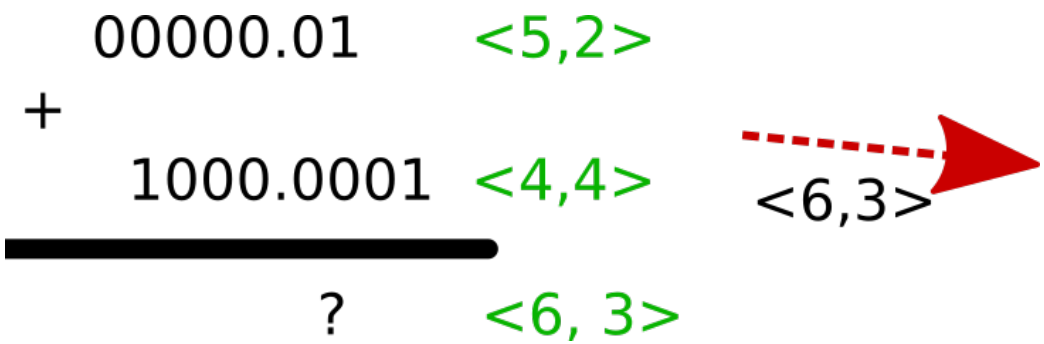
Addition

$$x_1 + x_2 = x_3$$

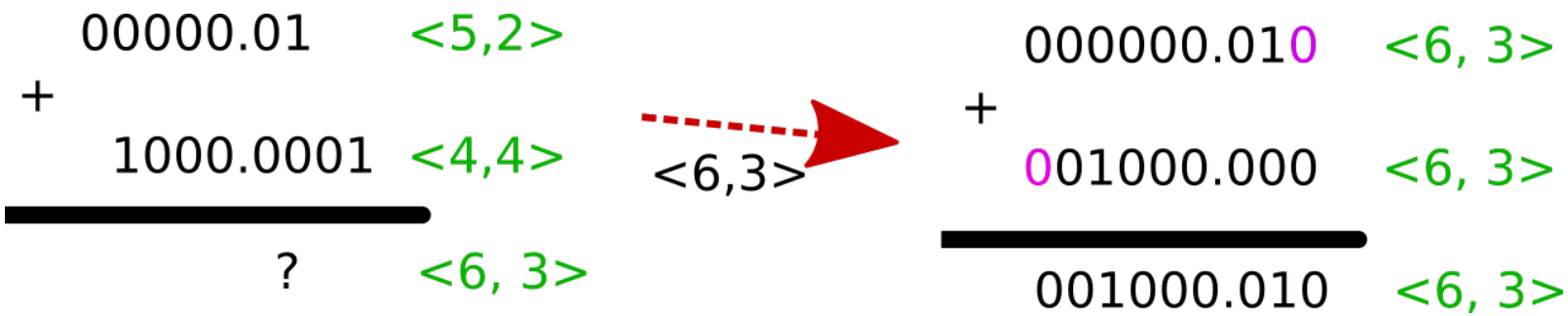
$$< i_1, f_1 > \ + \ < i_2, f_2 > \ = \ < i_3, \ f_3 > \tag{11}$$

$$\begin{cases} 0 < i_3 < p \\ 0 < i_3 + f_3 < p \\ i_3 = max(i_1, i_2) \\ f_3 = p - i_3 \end{cases} \tag{12}$$

Example 1: fixed format

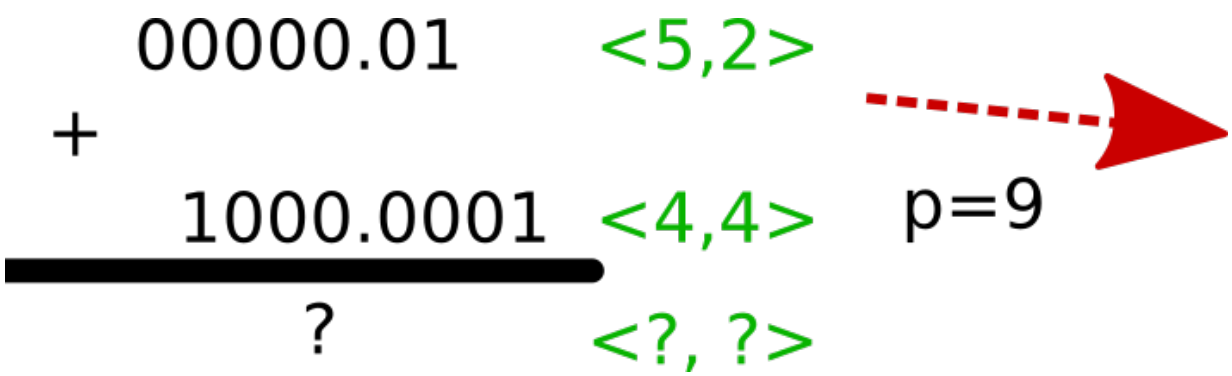


Example 1

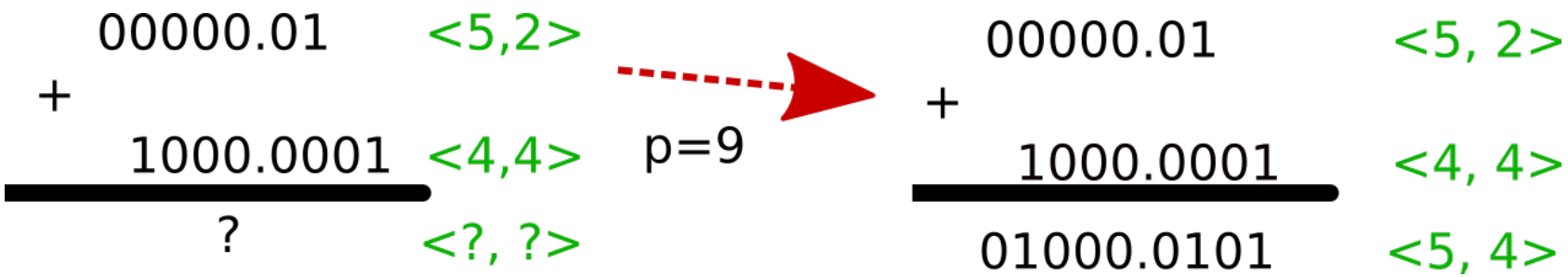

00000.01 <5,2>
+
1000.0001 <4,4>
?
<6, 3>
<6,3>
000000.010 <6, 3>
+
001000.000 <6, 3>
001000.010 <6, 3>

Example 2

Example 2: fixed precision



Example 2


00000.01 <5,2>
+
1000.0001 <4,4>
?
<?, ?>
p=9
00000.01
<5, 2>
+
1000.0001
<4, 4>
01000.0101
<5, 4>

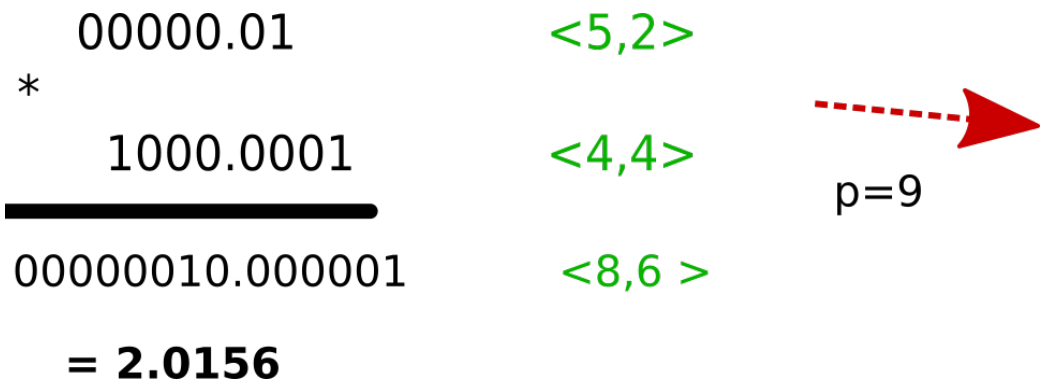
Multiplication

$$x_1 \times x_2 = x_3$$

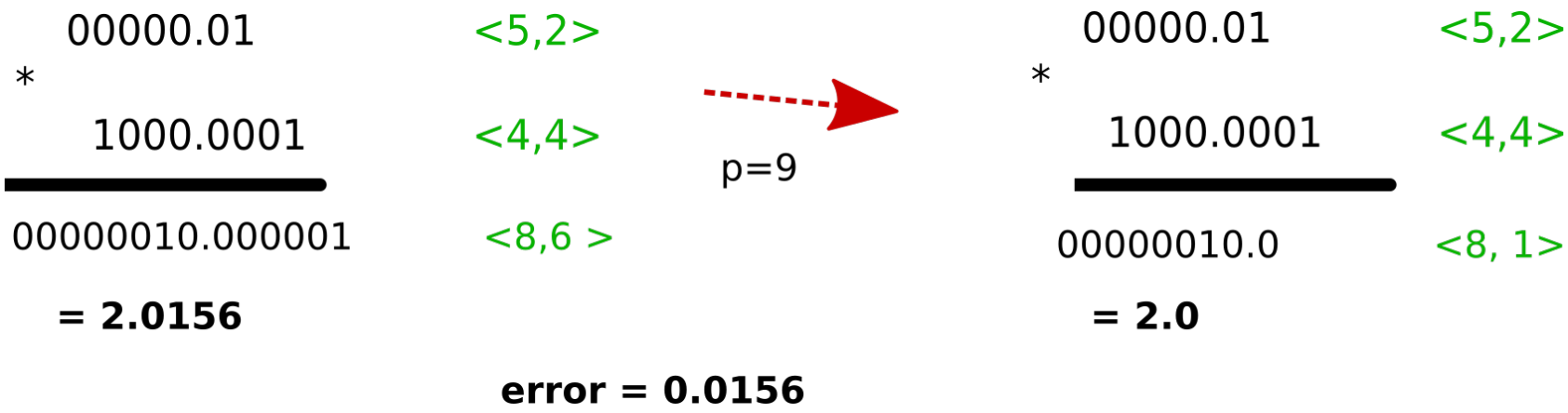
$$< i_1, f_1 > \times < i_2, f_2 > = < i_3, f_3 > \tag{13}$$

$$\begin{cases} 0 < i_3 < p \\ 0 < i_3 + f_3 < p \\ i_3 = i_1 + i_2 - 1 \\ f_3 = p - i_3 \end{cases} \tag{14}$$

Example: fixed precision



Example


00000.01
<5,2>
*
1000.0001
<4,4>
00000010.000001
<8,6 >
= 2.0156
p=9
00000.01
<5,2>
*
1000.0001
<4,4>
00000010.0
<8, 1>
= 2.0
error = 0.0156

AND, OR, XOR

$$x_1 \ \ AND \ \ x_2 = x_3$$

$$< i_1, f_1 > \ AND \ < i_2, f_2 > \ = \ < i_3, \ f_3 > \tag{15}$$

$$\begin{cases} 0 < i_3 < p \\ 0 < i_3 + f_3 < p \\ i_3 = max(i_1, i_2) \\ f_3 = p - i_3 \end{cases} \tag{16}$$

Same format for OR and XOR !

Plan

Format of weighted sum

Let :

$$f(\hat{x}) = W\hat{x} + b$$

$$f(\hat{x}) = \sum_{j=0}^{n} w_{ij}\,\hat{x}_j + b_i$$

$$f(\hat{x}) = \underbrace{\underbrace{\underbrace{w_{i1}\,\hat{x}_1}_{<i_2,f_2> \times <i_1,f_1>} + \underbrace{w_{i2}\,\hat{x}_2}_{<i_2,f_2> \times <i_1,f_1>} + \ldots + \underbrace{w_{in}\,\hat{x}_n}_{<i_2,f_2> \times <i_1,f_1>}}_{<i_1+i_2-1,f_1+f_2> \; + \; <i_1+i_2-1,f_1+f_2> \; + \; <i_1+i_2-1,f_1+f_2>} + \underbrace{b_i}_{<i_3,f_3>}}_{<i_4,f_4> \; = \; <i_1+i_2-1,f_1+f_2>}$$

$$<i_5,f_5> \; = \; <max(i_4,i_3),p-max(i_4,i_3)> \; = \; <max((i_1+i_2-1),i_3),p-max((i_1+i_2-1)>$$

$$\boxed{<\mathbf{i_5},\mathbf{f_5}> \; = \; <\mathbf{max}((\mathbf{i_1}+\mathbf{i_2}-\mathbf{1}),\mathbf{i_3}),\mathbf{p}-\mathbf{max}((\mathbf{i_1}+\mathbf{i_2}-\mathbf{1})>} \tag{17}$$

W : matrix of weights, $\hat{\mathbf{x}}$: inputs and **b** : bias

$<\mathbf{i_1},\mathbf{f_1}>$: representation format of the input vector $\hat{x}$

$<\mathbf{i_2},\mathbf{f_2}>$: representation format of W

$<\mathbf{i_3},\mathbf{f_3}>$: representation format of bias b

$<\mathbf{i_5},\mathbf{f_5}>$: representation format of the result $f(\hat{x})$

Plan

Conclusion & Perspectives

- Reducing the size and execution time of large NN with this approach
- Implementing a code transformation tool
- Experiments on large NN
- Handling classifiers and interpolators.

Related Work

Dutta S. Jha S., Sankaranarayanan S. Tiwari A. (2018). "Output Range Analysis for Deep Feedforward Neural Networks". In: *NFM*.

Gehr T. Mirman M., Drachsler-Cohen D. Tsankov P. Chaudhurri S. Vechev M.T. (2018). "AI^2: Safety and Robustness Certification of Neural Networks with Abstract Interpretation". In: *SP*.

Martel, Matthieu (2017). "Floating-Point Format Inference in Mixed-Precision". In: *NFM*.

Singh G. Gehr T., Puschel M. Vechev M.T. (2019). "An Abstract Domain for Certifying Neural Networks". In: *POPL*.

- Gehr T. 2018 Singh G. 2019

- Gehr T. 2018 Singh G. 2019
- Martel 2017
- Dutta S. 2018