JAUES

Journal of Al-Azhar University Engineering Sector

Vol.15, No. 54, October 2020, 1146-1156

PRECISION ON DEMAND: A NOVEL LOSSLES MIXED-PRECISION COMPUTATION TECHNIQUE

A. Al-Marakeby
Systems and Computers Engineering Dept. , Faculty of Engineering , Al-Azhar Univ.
Cairo, Egypt.
E-mail: a.marakeby@azhar.edu.eg

ABSTRACT:
Nowadays, there are wide range of computing-intensive applications that require a powerful computing platform. This computational complexity can be reduced significantly using lower precision, but certainly the accuracy will be affected. In this work, a novel lossless mixed-precision computation technique is used to reduce the computational complexity for such problems while keeping the same accuracy of higher precision. The "precision on demand" technique depends on iterative computation method, which utilizes discarding stages. The discarding stages are commonly used in many applications where some intermediate results are discarded, and they do not contribute to the final results. Max, min, threshold, and ReLU -operations are examples of such discarding stages. Lower precision is used to compute all intermediate results, then non-discarded values are recomputed using higher precisions. This technique enhances performance, improves power-consumption, reduces memory requirements, and allows implementing heavy computational systems on low resources and low-cost platforms. This work presents the decision on demand concept, and discuss many details related to hardware architecture implementations and optimizations. CNN inference is used as a case study, and speed is improved by a factor of 1.6x- 3.5x.

KEYWORDS: Mixed Precision, Floating Point, Convolutional Neural Networks, Computational Complexity , Field Programmable Gate Arrays, Image Classification.

الدقة عندالاحتياج: طريقة حسابية مبتكرة باستخدام الدقة المختلطة دون فقد

أشرف المراكبي
قسم هندسة النظم والحاسبات، كلية الهندسة ،جامعة الأزهر، القاهرة ، مصر.
البريد الإلكتروني: E-mail: a.marakeby@azhar.edu.eg

الملخص:
يوجد في هذه الأيام نطاق عريض من التطبيقات التي تحتاج إلى قوة حاسوبية كبيرة. هذه التعقيدات الحسابية يمكن تقليلها بشدة عند استخدام الدقة المنخفضة ولكن بالطبع ستتأثر دقة المخرجات. في هذا البحث تم استخدام طريقة مبتكرة لا ستخدام الدقة المختلطة مع المحافظة على نفس المخرجات التي تم الحصول عليها باستخدام الدقة المرتفعة. طريقة الدقة عند الاحتياج تعتمد على طريقة حسابية تكرارية تستفيد من المراحل التجاهلية. تستخدم المراحل التجاهلية بكثرة في تطبيقات كثيرة حيث يتم تجاهل بعض النتائج الوسيطة ولا يتم استخدامها في حساب النتائج النهائية . القيمة العظمى والقيمة الصغرى وحساب التقويم الخطي والتقويم الحدي يمثل بعض أمثلة على المراحل التجاهلية. في هذه الطريقة يتم حساب جميع القيم الوسيطة بدقة أقل وبعد ذلك يتم استخدام الدقة الأعلى في إعادة حساب القيم التي لن يتم تجاهلها. هذه الطريقة تحسن الأداء وتحسن استهلاك الطاقة وتقلل الذاكرة المطلوبة وتتيح تنفيذ عمليات حسابية معقدة على أجهزة حاسوبية محدودة الإمكانيات

ورخيصة الثمن . في هذا البحث سيتم تقديم مفهوم الدقة عند الاحتياج كما سيتم مناقشة تفاصيل تتعلق بمعمارية العتاد المستخدم في تنفيذ هذه الفكرة وطرق تحسينه. الشبكات العصبية الالتفافية تم استخدامها كحالة للدراسة وتم تحقيق تحسين في السرعة من من ١.٥ إلى ٣.٦ أضعاف السرعة بالطرق التقليدية.

الكلمات المفتاحية: الدقة المختلطة ، الأرقام العشرية ، الشبكات العصبية الالتفافية، التعقيدات الحسابية، مصفوفات بوابات المجال المبرمجة، تصنيف الصور.

1. INTRODUCTION

There are significant improvements in computers speed, memory, architecture, and parallel cores. A tiny smart phone can be more powerful than Deep Blue supercomputer which beat Kasparov in 1997. Special hardware devices and accelerators are available with huge number of cores and large memory size such as GPUs, TPUs, and FPGA. Although this significant progress, there are many applications that are very heavy and requires more and more computation power. Deep learning, AI , Big Data , computer vision , video processing and many other applications needs tens of GFLOPs to hundreds of TFLOPs to accomplish the computation of simple tasks. This represents a challenge and add many difficulties for running these systems, especially when using embedded systems for real time performance. Vision systems for self-driving car needs very high computational power to analyze video frames from several cameras within few milli seconds. Precision represents a crucial parameter and plays a main rule in reducing or increasing the complexity, time, and power consumption[4][5]. Lower precision helps in performing more operations per second and, reduces memory bandwidth requirements. Higher precision arithmetic increases the computation accuracy, but with the cost of lower performance and higher memory bandwidth requirement. The influence of precision can be small in some cases, while it can be great and important in other cases. Using High level languages such as C++, the developer can choose single precision instead of double precision floating point representations, which can increase the computation speed twice. In the case of designing an FPGA accelerator, a special hardware architecture can be built for arbitrary precision, formats, or bit widths. Floating point, fixed point, 32 bit, 16 bit , and even 4 bit representations can be used if this can satisfy the given requirements. The speed, area and power consumption in these cases can be improved by a factor of 10 x or more[7][11]. However, the accuracy represents the main problem of lower precision computation. Mixed-precision algorithms are used to achieve the accuracy of high precision computation with the efficiency close to low precision computation[1][2][3]. Most of mixed-precision techniques required careful error analysis to estimate the effects of lower precision. Also design efforts are needed to determine which stages can be computed using lower precision, and to determine the appropriate precision level. Usually, analysis and design for a specific problem cannot be generalized. In this work, a new mixed-precision technique based on discarding stages is presented. The “Precision on Demand” (PoD) method combines different levels of precision without losing the accuracy of higher precision. No need for error analysis, where there is no precision errors or differences compared to the high precision computations. The precision on demand depends on iterative computation method, which exploits discarding stages. The discarding stages are commonly used in many applications where some intermediate results are discarded, and they do not contribute to the final results. For example, the max-pooling in CNN passes one value from 4 values and discard others. It is not required to waste time and resources in computing the discarded values using high precision. This technique focusses on utilizing these discarding operations to apply mixed-precision computation. The main contribution of this work can be summarized as follow:

1- This mixed-precision technique is lossless, and the error between high precision and mixed-precision computations is zero.
2- The PoD technique depends on the discarding stages which are commonly found in many applications.
3- This technique supports different precision levels and can be implemented on different architectures.

.

2. BACKGROUND

There are many standards and custom data formats used to represent numbers. Floating point , fixed point, and integer representations are wieldy used with different bit widths. Choosing the appropriate format and precision depends on the problem requirements. The IEEE-754 standard for floating-point arithmetic (revised in 2008) [14] uses a triplet to represent a FP value *x: (s,e,m),* where $x = (-1)^s 2^e m$. IEEE 32 bit FP is called single precision , and 64 bit is called double precision. A floating-point variable can represent a wider range of numbers than a fixed-point variable of the same bit width but with a lower precision. Working precision ε_w is the precision used to store the input data ($\varepsilon_w = 2^{-24}$ for 32 bit FP, $\varepsilon_w = 2^{-53}$ for 64 bit FP). While these errors are very small, but they can be magnified or accumulated when a sequence of calculations is applied on the initial inputs. Many systems are developed which combine single precision FP with double precision FP in a mixed precision computation model. Buttari et al. used mixed precision iterative refinement technique for the solution of dense linear systems [3]. They showed that, the performance of many dense and sparse linear algebra algorithms can be significantly enhanced while maintaining the 64-bit accuracy of the resulting solution. Their approach can be applied not only to conventional processors but also to other technologies such as FPGA and GPUs. Lam et al. presented a framework that uses binary modifications to build mixed precision modifications of existing binaries that were originally developed to use only double precision [8]. They achieved a speed up of 2x in some cases. Other work of combining single precision FP with double precision can be found in [4][10]. More improvements can be achieved using lower precision if possible, such as16 bits, 8 bit and 4 bits(floating point and fixed point). Sun et al. used mixed precision linear solver on FPGA. They proposed an innovative architecture for a configurable computing [1]. They used custom formats and compared frequencies and resource utilization for different bit widths. Dynamically adjustable fixed point formats are widely used in many applications such as CNN and deep neural networks. Yufei et al. optimized the convolution operation to accelerate deep neural networks on FPGA[11]. They used fixed-point data representation, and both pixels and weights are 16-bit. The decimal points are dynamically adjusted according to the ranges of pixel values in different layers to fully utilize the existing data width. The top-1 and top-5 ImageNet classification accuracy degradation is within 2% compared with software floating point implementation. Dong et al. developed a Hessian aware quantization of neural networks with mixed-precision[5]. Their work allowed for the automatic selection of the relative quantization precision of each layer, based on the layer's Hessian spectrum. They quantized the SqueezeNext model to uniform 8-bit precision, with 0.04% top-1 accuracy drop. Reconfigurable architectures are important to exploit efficiently the mixed precision computation. Jaiswal et al. developed a unified architecture for double/two-parallel single precision FP adder[7]. Liang et al. designed an ALU architecture that support dynamic precision operation on the fly[9]. An n-bit operand is partitioned into k sub-blocks with block size of n/k bits . Adjacent sub-blocks can be combined dynamically to form a super-block according to the required precision[9]. Tan proposed a 64-bit multiply accumulator that can compute one 64x64, two 32x32, four 16x16, or eight 8x8 unsigned/signed multiply-accumulations using shared segmentation [13]. Most of existing methods of mixed precision reduce the accuracy and needs a careful study analysis of the errors effects. In this work, the proposed technique is different. The PoD computation model does not give any errors and the mixed precision computations are exactly the same as higher precision. This model can be easily generalized to different applications and different platforms.

3. PRECISION ON DEMAND CONCEPT

The precision on demand technique depends on iterative computation with different precisions starting with lower precision and then higher precisions as required. The main condition for the success of this concept is the existence of discarding process for some calculated values. At these discarding stages, some calculated values contribute to the next computations stages while some other calculated values are completely discarded and have no effects for the next computations. For example, the max function for finding the maximum value of a list *N* values are considered as a discarding stage. Only the maximum value will be used for the next computations while the other values are discarded. These discarding processes are massively

found in many applications, and they represent the base of the precision on demand PoD technique. Fig .1 shows the discarding process for max function where all values are discarded except for the maximum value.

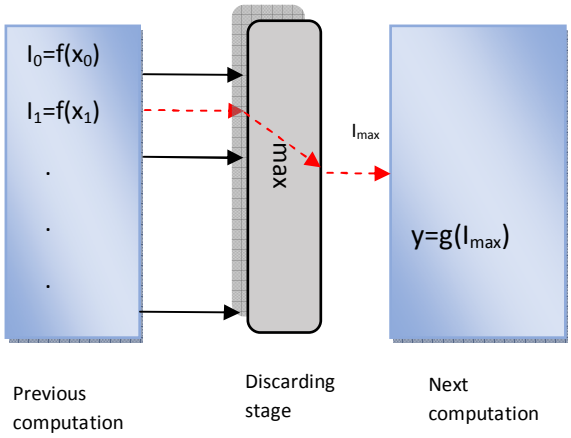


Fig.1 Discarding process for max function

The function $I=f(x)$ can be computed with a lower precision ε_L for all values of I. After finding the max value, the computation of $I_{max}=f(x)$ is repeated with a higher precision ε_H . Hence, the function $y=g(I_{max})$ is computed using the same accuracy of higher precision without any losses, but with a reduced computational cost. A simplified speedup model is given in eq. 1.

$$S = T_h / T_{mix}$$
$$T_{mix} = NT_{SL} + T_{SH}$$
$$T_h = NT_{SH}$$
$$S = NT_{SH} / (NT_{SL} + T_{SH}) \qquad (1)$$

Where N is the size of the list, T_h is the total time required to calculate all variables using higher precision , and T_{mix} is total time using mixed precision. The average time for single variable calculation is denoted as T_{SH} and T_{SL} for higher precision and lower precision respectively. Assuming N=100, and $T_{SH}=2\ T_{SL}$, the Speedup S=1.96 . Assuming β is the ratio between T_{SH} and T_{SL} , eq. 1 can be written as

$$S = \beta N / (N + \beta) \qquad (2)$$

The value of β for small bit widths will be increased. If 16 bit floating point is used with 64 bit floating point , β can be within the range of *4* ,and the speed up is improved to $S=3.8$.
As stated before, mixed precision not only improve speed, but also power , area utilization , memory size , memory bandwidth,….etc. Assuming R is the number of recalculations and D is the number of discarded values, it is required to increase D and decrease R to improve the performance. In the previous example, R =1 for max function, but it's not for other problems. Also, R can be increased for the max function if there is an overlap in the error bounds. If the two max values of the list I are: $I_1=4.8\pm0.2$, $I_2=4.7\pm0.2$, it is required to recalculate both of I_1 and I_2 . In this case, R=2 and there are two candidates for the max value and the lower precision calculations cannot determine exactly the max value. Fig.2 shows the repeated values and discarded values based on the overlap between error bounds.

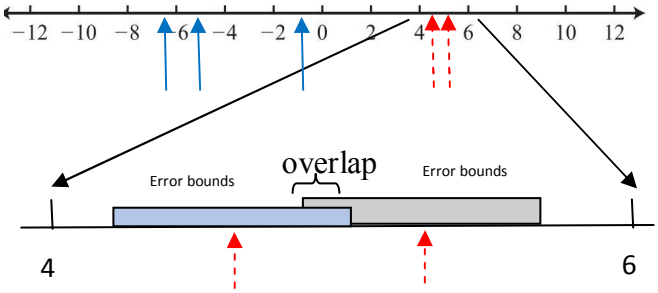


Fig.2 recalculation of the max function (Dashed arrows represent recalculation with higher precision-solid arrows represent discarded values)

To include the number of recalculations in eq.2 the speedup is expressed as

$$S = \beta N / (N + \beta R) \qquad (3)$$

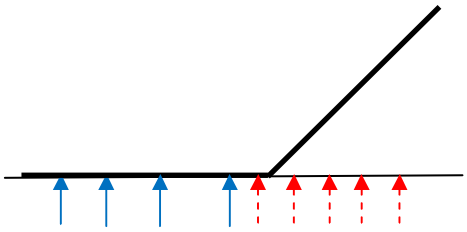
The same concept is applied to different discarding functions such as min , thresholding, ReLU , ..etc. in fig.3 the range of values which requires recalculations with higher precision is illustrated for different functions. The R value (number of recalculations) is large for these functions. If R is very large or tends to N , all values will be computed twice using different precisions and the mixed precision performance will be less than higher precision performance. The value of R in these cases depends on the input data and profiling is required to ensure the improvements.

Multi levels mixed precision

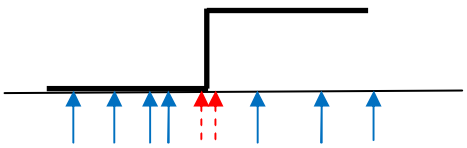
Three or more different precisions can be used to give improvements more than using only two levels of mixed precision. The main problem of multi-levels mixed precisions is the repetition of the same computation several times. If this model is not used carefully, it can degrade the performance instead of improving it. Assuming we have three levels of precisions with single variable computation times T_{S1},T_{S2}, and T_{S3}, R_2, and R_3 is the number of variable computed again using precision 2 and 3. The speedup is given by

$$S = N T_{S3} / (N T_{S1} + R_2 T_{S2} + R_3 T_{S3}) \quad (4)$$

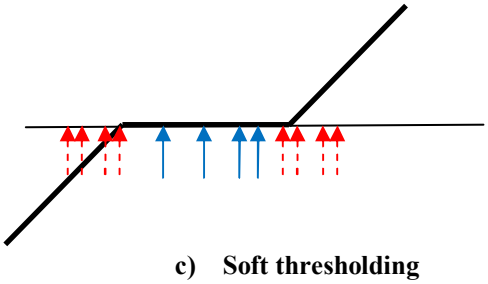
From this equation, multi -level mixed precision can improve performance using only very low precisions with fast computation time.



a) ReLU



b) Step Thresholding



c) Soft thresholding

Fig.3 recalculation of different discarding functions

Error bounds

Lower precision representations and calculations have problems in approximation errors, overflow , and underflow. Error bounds calculation is important for obtaining accurate results using precision on demand technique. Error bounds depends on both the round off errors and

the accumulation of errors due to the sequence of calculations. For the example shown in fig. 1, let $f(x) = 3x^2 - 2x + 5$, and $x_j = 1.43453$. Using 16 bit Fixed point Q8.8, x will be represented approximately as 1.43 with rounding error $= 0.00453$. But after multiplication and addition the final error will be -0.0298689. The final computation error is denoted as $\boldsymbol{\delta}$, and $f(x) = f_{low\_prec}(x) \pm \boldsymbol{\delta}$. Hence, if $abs(I_{max} - I_k) > \boldsymbol{\delta}$, then it is required to repeat the computation of I_k with higher precision to check if $I_k > I_{max}$. The value of δ should be computed on the worst case based on the used precision and the sequence of calculations.

Incremental Precision Computation

It is assumed that, higher precision computations don't depend on lower precision computations nor exploit their results. Dedicated architectures can be implemented to exploit the lower precision results in higher precision computation for the same variable. This can save time and resources significantly. For example, the 32 bit fixed point addition consists of the 16 bit LSB addition combined with the lower precision result. Not all operations nor architectures support this incremental precision computations, but performance can be improved using this technique. The precision on demand technique can be summarized in the following steps illustrated in fig.4.

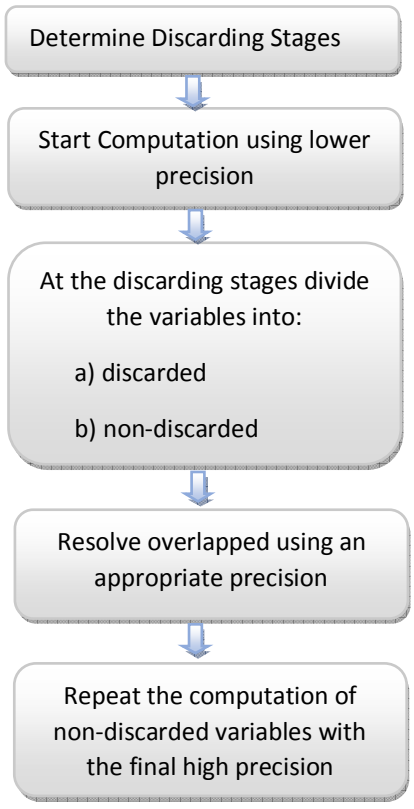


Fig. 4 Precision on Demand steps

4. CASE STUDY: CNN INFERENCE

Convolutional Neural Network (CNN) has achieved great success in a large number of applications and became among the most powerful and widely used techniques in computer vision[12]. However, CNN models are computational-intensive and they are difficult to be integrated into embedded systems and real time applications such as smart phones, , robots ,and self driving cars. A lot of research can be found for solving these computation problems by finding new lite models such as MobileNet[16] or optimizing existing heavy models[2][10]. The Precision on Demand (PoD) technique is suitable for CNN inference due

to the existing of many discarding stages. Applying PoD for CNN inference does not prevent other optimizations and the combination of PoD with other techniques improves performance more and more. The main difference between this technique and others is the lossless computation accuracy. Many research in this field achieve zero system accuracy losses, but the computations contains many errors. Some mixed precision and reduced precision CNN inference systems achieve the TOP-1, and TOP-5 accuracy as same as the higher precision, but with completely different values. Hence, there is no certainty that, the reduced precision model version is typically like the original high precision model. The behavior of reduced precision models can be completely different in many non-tested cases. However, PoD technique gives the same system accuracy and the same computation values. Hence, the mixed precision version is the same as the higher precision version in all cases. Fig.5 shows the structure of CNN model.

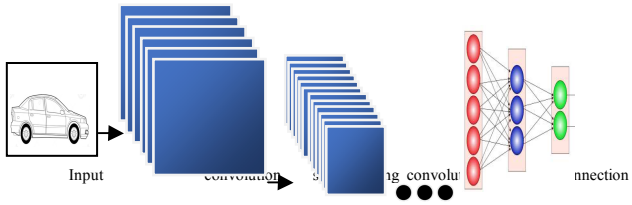


Fig.5 Convolutional Neural Network (CNN) model

The Rectified Linear Units (ReLUs) are widely used in CNN models, and they represent good discarding stages for PoD. The function $f(x) = max(0,x)$ can be computed using lower precision to discard easily the values which are far from zero. The non-discarded values will be computed again using the higher precision. When a convolution layer is followed by a ReLU as shown in fig.6 , the mixed precision computation can save time , power and resources. For 3x3 kernels, there are 9 multiplications and 9 additions (or 9 multiply and accumulate operations), then the discarding stage is applied to the result of convolution. Very small bit widths can be used initially to discard many values with small computational cost. Overflow represents a challenge for small bit widths, but dynamic range can be used to overcome this problem.

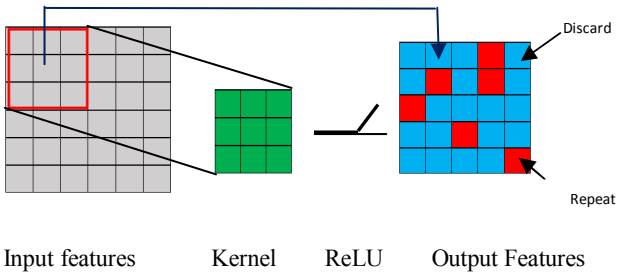


Fig.6 Convolution + ReLU layer

Max-pooling is an important stage in CNN models and usually 3 values are discarded from each 4 values. Fig.7 shows the max-pooling stage and the discarded values. More than one value may require recalculation if there is an overlap between error bounds.

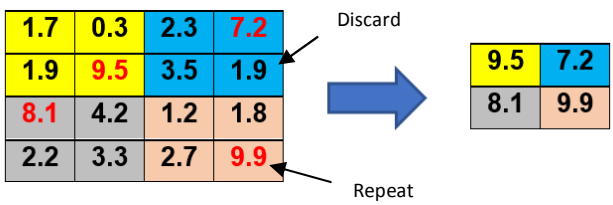


Fig.7 sub-sampling (max-pooling) discarding stage

5. HARDWARE ARCHITECTURES

Most CPUs and GPUs use standard fixed bit widths arithmetic units. Algorithms which exploit custom bit widths and custom data format such as PoD, are not efficient on CPUs and GPUs. Dedicated hardware gives very high flexibility to implement such algorithms and the performance can be improved significantly using these custom architectures. In the case of CNN inference, the operations are restricted almost in multiplications and additions (or MACs) . Arithmetic units can be reconfigurable to work with different precisions as required. Mixed precision adders, multipliers , and MACs are designed with different bit widths and different formats. They can be configured on fly using control signals to work as a single high precision unit or multiple low precision units[7][9]. Fig.8 (a) shows a simple ALU architecture which can be used as a single 16 bit or double 8 bit ALUs. This technique saves resources, but it is not applicable to many cases. Many operations and data formats are not suitable for implementation as reconfigurable mixed precision units . Another solution is to add different separate units for each precision. Special arithmetic units are implemented for low precision computation and another arithmetic units are implemented for high precision computation.

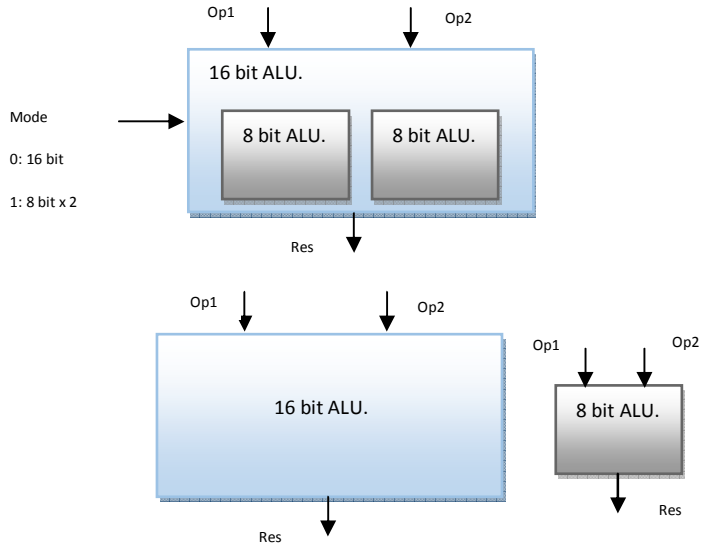


Fig.8 Mixed precision ALU

a)Reconfigurable mixed units. **b)Separate units**

Separate mixed precision units looks as a bad choice for working with mixed precision, but actually, it is a practical solution in many cases . In some cases, the same work done by 32-bit floating point multiplier can be done by 8-bit multiplier. The IP core of 32 bit floating point multiplier requires 7 x (DSP 9-bit multiplier) +111 LUT + 464 reg, while the 8-bit multiplier requires only a single DSP multiplier unit (or about 90 LUT). This means that, load can be reduced from high precision units, with a very small cost. Also, power consumption and time of 8-bit multiplier is very small compared to 32-bit floating point multiplier. For parallel structures, a simple reduction in the number of high precision cores may allow the addition of a huge number of low precision cores which can attract a big computation load.

6. EXPERIMENTAL RESULTS

Precision on Demand (PoD) Technique can be used in different application and can be implemented on different platforms. To test this technique, CNN inference is chosen as a case study.

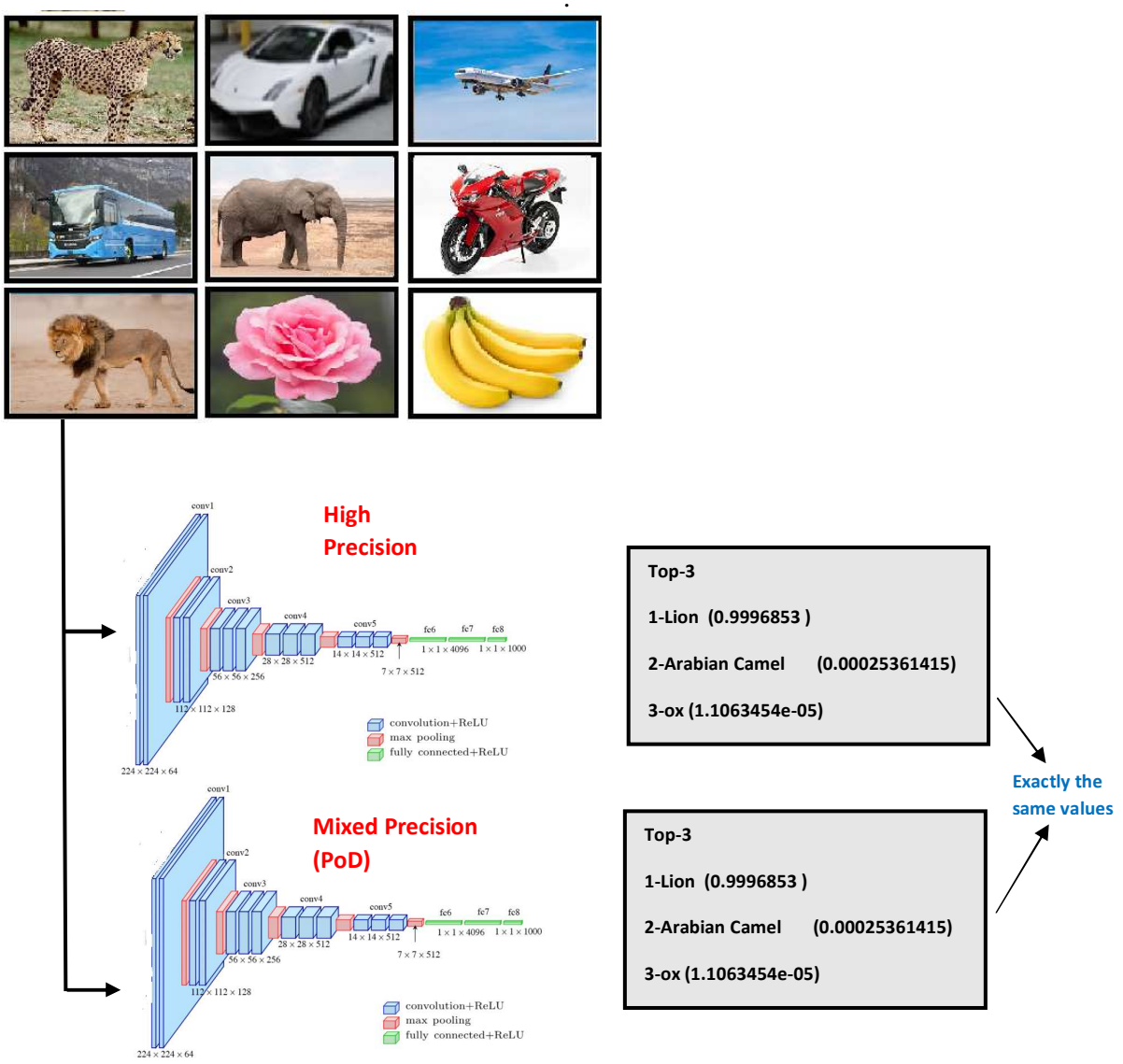


Fig.9 VGG-16 mixed precision (PoD) image classification

General purpose CPUs are chosen as the testing platform, but also other platforms expected performance are evaluated using simulation. Different parameters such as processing time , precision , data formats are used to estimate the expected speedup of PoD technique using other platforms. VGG-16 model with imagenet weights is used to classify images and the convolution layers are computed using PoD technique. The number of operations in the convolution layers is about 90% of the total operations. The convolution layers outputs are computed using lower precision, then ReLU and max-pooling stages determine discarded and non-discarded results. Only non-discarded results are recomputed again using the higher precision. All computation using lower precision are not propagated, and they do not contribute to the final results. They either discarded or recomputed again. The results of mixed precision system are exactly the same as higher precision system as shown in fig.9.
The input image is 224 x 224 x 3 and the total number of computations and re-computations for each layer is given in table 1. These results in table.1 represents the number of operations for classifying a single image using 32-bit /16-bit floating point representations. The error between higher precision computations and mixed precision is zero for all values.

.

Table.1 mixed precision (32-bit /16-bit) floating point PoD for VGG-16

layer	# Computations(N)	# re-computations(R)	Re-computation ration	# operations/ computation	Error
Conv1	3.2M	1.42M	44.3%	27	0.00
Conv2 +Pooling	3.2M	0.9M	28.1%	576	0.00
Conv3	1.6M	1.1M	68.75 %	576	0.00
Conv4 +Pooling	1.6M	0.298M	18.6%	896	0.00
Conv5	0.8M	0.45M	56.25%	896	0.00
Conv5	0.8M	0.42M	52.5%	1792	0.00
Conv7 +Pooling	0.8M	0.09M	11.2%	1792	0.00
Conv8	0.4M	0.15M	37.5%	1792	0.00
Conv9	0.4M	0.12M	30%	4608	0.00
Conv10 +Pooling	0.4M	0.025M	6.25%	4608	0.00
Conv11	0.1M	0.024M	24%	4608	0.00
Conv12	0.1M	0.019M	19%	4608	0.00
Conv13 +Pooling	0.1M	0.004M	4%	4608	0.00
Total Operations =13.6 GOPs Re-Computed Operations =3.9 GOPs Re-Compute Ratio=28.6%					

For each layer the number of computations N (number of pixels in output features) are given and the number of re-computations R. The number of operations required to compute a single pixel in output features is different for each layer. The ratio of re-computations is very small for convolution layers followed by a max-pooling. This is due the existence of two discarding stages (ReLU and max-pooling) . Usually, max-pooling re-compute one value from each 4 values, but in some cases more than one value can be re-computed due to error bounds overlapping. These results depends on the input data (image), but different images are tested and the results are within these ranges. Equation 3 can be used to estimate the speedup for different hardware architectures. Assuming the ratio between 32-bit FP and 16-bit FP time (β) is 3, then the speedup s=1.6 . This evaluated speedup depends only on the improvement in arithmetic processing time. Including many factors such as memory bandwidth, and number of processing elements will increase the speedup significantly. In some cases, one high precision processing elements (PEs), can be replaced with 10 or more low precision PEs with the same area. Using Fixed point representation has the advantage of fast processing and low resources, but the range is small. The same results are obtained using 16-bit fixed point, but with a step increase in re-computation ratio due to the overflow. Lower bit widths can be used also, but error bounds increase, causing more re-computations. An estimated speedup using FPGA/ASIC dedicated hardware is s=3.5. This technique can be combined with other optimization techniques to give more improvements such as zero-skipping, and depthwise separable convolution[1][2]. While the mixed precision behaviour is typically the same as high precision without any losses, hence the results will be the same for all situations and for tested and non-tested images. The same concept can be applied to other computing-intensive applications specially for systems which are error-sensitive.

.

7. CONCLUSION

Mixed precision algorithms improve performance, power consumption, and resource utilization, but with a cost of low accuracy. Precision on Demand (PoD) technique is a mixed precision technique which achieves the same accuracy of high precision computation. PoD technique works only with computational models containing discarding stages such as max, min and thresholding. PoD is more efficient if dedicated hardware is used which allow custom bit widths and data formats. PoD can be combined with other optimization techniques to achieve more improvements.

REFERENCES

1. Aimar, A., Mostafa, H., Calabrese, E., Rios-Navarro, A., Tapiador-Morales, R., Lungu, I. A., ... & Delbruck, T. (2018). Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps. IEEE transactions on neural networks and learning systems, 30(3), 644-656.
2. Aimar, Alessandro, et al. "Nullhop: A flexible convolutional neural network accelerator based on sparse representations of feature maps." IEEE transactions on neural networks and learning systems 30.3 (2018): 644-656.
3. Buttari, A., Dongarra, J., Langou, J., Langou, J., Luszczek, P., & Kurzak, J. (2007). Mixed precision iterative refinement techniques for the solution of dense linear systems. The International Journal of High Performance Computing Applications, 21(4), 457-466
4. Demmel, J., Hida, Y., Kahan, W., Li, X. S., Mukherjee, S., & Riedy, E. J. (2006). Error bounds from extra-precise iterative refinement. ACM Transactions on Mathematical Software (TOMS), 32(2), 325-351.
5. Dong, Z., Yao, Z., Gholami, A., Mahoney, M. W., & Keutzer, K. (2019). Hawq: Hessian aware quantization of neural networks with mixed-precision. In Proceedings of the IEEE International Conference on Computer Vision (pp. 293-302).
6. Howard, A. G., Zhu, M., Chen, B., Kalenichenko, D., Wang, W., Weyand, T., ... & Adam, H. (2017). Mobilenets: Efficient convolutional neural networks for mobile vision applications. arXiv preprint arXiv:1704.04861.
7. Jaiswal, M. K., Cheung, R. C., Balakrishnan, M., & Paul, K. (2014). Unified architecture for double/two-parallel single precision floating point adder. IEEE Transactions on Circuits and Systems II: Express Briefs, 61(7), 521-525
8. Lam, M. O., Hollingsworth, J. K., de Supinski, B. R., & LeGendre, M. P. (2013, June). Automatically adapting programs for mixed-precision floating-point computation. In Proceedings of the 27th international ACM conference on International conference on supercomputing (pp. 369-378).
9. Liang, G., Lee, J., & Peterson, G. D. (2012, July). ALU Architecture with Dynamic Precision Support. In 2012 Symposium on Application Accelerators in High Performance Computing (pp. 26-33). IEEE.
10. Qiu, Jiantao, et al. "Going deeper with embedded FPGA platform for convolutional neural network." Proceedings of the 2016 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays. 2016.
11. Ma, Y., Cao, Y., Vrudhula, S., & Seo, J. S. (2018). Optimizing the convolution operation to accelerate deep neural networks on FPGA. IEEE Transactions on Very Large Scale Integration (VLSI) Systems, 26(7), 1354-1367.
12. Sun, J., Peterson, G. D., & Storaasli, O. O. (2008). High-performance mixed-precision linear solver for FPGAs. IEEE Transactions on Computers, 57(12), 1614-1623.
13. Tan, D., Danysh, A., & Liebelt, M. (2003, June). Multiple-precision fixed-point vector multiply-accumulator using shared segmentation. In Proceedings 2003 16th IEEE Symposium on Computer Arithmetic (pp. 12-19). IEEE.
14. Zuras, D., Cowlishaw, M., Aiken, A., Applegate, M., Bailey, D., Bass, S., ... & Canon, S. (2008). IEEE standard for floating-point arithmetic. IEEE Std, 754(2008), 1-70