DEF: Differential Encoding of Featuremaps for Low Power Convolutional Neural Network Accelerators

Alexander Montgomerie-Corcoran
Imperial College London
London, UK
alexander.montgomerie-corcoran15@imperial.ac.uk

Christos Savvas-Bouganis
Imperial College London
London, UK
christos-savvas.bouganis@imperial.ac.uk

ABSTRACT
As the need for the deployment of Deep Learning applications on edge-based devices becomes ever increasingly prominent, power consumption starts to become a limiting factor on the performance that can be achieved by the computational platforms. A significant source of power consumption for these edge-based machine learning accelerators is off-chip memory transactions. In the case of Convolutional Neural Network (CNN) workloads, a predominant workload in deep learning applications, those memory transactions are typically attributed to the store and recall of feature-maps. There is therefore a need to explicitly reduce the power dissipation of these transactions whilst minimising any overheads needed to do so. In this work, a Differential Encoding of Feature-maps (DEF) scheme is proposed, which aims at minimising activity on the memory data bus, specifically for CNN workloads. The coding scheme uses domain-specific knowledge, exploiting statistics of feature-maps alongside knowledge of the data types commonly used in machine learning accelerators as a means of reducing power consumption. DEF is able to out-perform recent state-of-the-art coding schemes, with significantly less overhead, achieving up to 50% reduction of activity across a number of modern CNNs.

KEYWORDS
Power Optimisation, Activity Coding, Neural Networks

1 INTRODUCTION
Power-efficiency is a sought after characteristic in many CNN accelerators. As the use-case for these accelerators moves closer to edge-based applications, the constraints of the platform they are deployed on become more and more prominent. In many applications the power-budget of the system is the main limiting-factor, which often places a bottleneck on the computational performance of the CNN accelerator. Previous work identified that accessing data stored in DRAM is a significant source of power consumption for many embedded systems [5, 9]. The power consumption of the memory subsystem is due to both the DRAM chips themselves as well as due to the bus-lines connecting off-chip memory to the device. These bus-lines have high capacitance compared to on-chip routing, and therefore consume a considerable proportion of the total dynamic power of the system. Dynamic power consumption for a bus is modelled by Equation (1):

$$P_{\text{dynamic}} = n \cdot V_{dd}^2 \cdot f_{clk} \cdot C \cdot a$$  \hspace{1cm} (1)

where $n$ is the number of bus lines, $V_{dd}$ is the driving voltage, $f_{clk}$ is the clock frequency of the bus, $C$ is the capacitance of the bus lines and $a$ the the activity on the bus. The values $n$, $V_{dd}$, $f_{clk}$ and $C$ are system parameters that are fixed for a specific platform. The activity, $a$, however is highly dependant on the workload and how it is encoded for transmission.

As the activity is dependant on the memory transactions, a common technique to address high activity is to introduce a coding scheme for reducing the bus activity, both for address lines and data lines. Existing coding schemes introduce either spatial [8, 17] or temporal [16] redundancy in order to carry information about the coding scheme. Temporal redundancy introduces extra words into the data stream, impacting performance. Spatial redundancy introduces extra bus-lines, impacting the available resources. Moreover, coding schemes can be categorised as either adaptive [8, 16, 17] or static [11]. Static coding schemes utilise fixed properties of the bus activity to design their coding scheme, resulting in approaches with the need of limited resources, whereas adaptive schemes use online methods.

The coding scheme proposed in this work is a static scheme that utilises knowledge of the CNN workload in order to efficiently reduce the activity along the off-chip memory bus. By using statistical knowledge of the feature-maps transferred between the memory and the accelerator during execution of a CNN workload, the coding scheme is able to achieve improved reduction in activity compared to recent state-of-the-art coding schemes. This is achieved without introducing any temporal or spatial redundancy.

The paper is structured as follows: Section 2 covers background work on CNN accelerators and their workloads. Related state-of-the-art coding schemes are also introduced. Section 3 introduces DEF \(^1\), the proposed coding scheme. Finally, the proposed coding scheme is evaluated against other coding schemes and results obtained from prototyping on an FPGA device are reported in Section 4.

2 BACKGROUND & RELATED WORK

2.1 Activity Minimisation Schemes
There have been many coding schemes proposed in order to reduce activity on off-chip busses. The main activity reduction schemes of interest are Bus-Invert (BI) coding [17], Probability-Based Mapping (PBM) [7, 11], Adaptive Bus Encoding (ABE) [16] and Adaptive Word Reordering (AWR) [8]. BI minimises the maximum number of transitions. The inverse of the incoming signal is sent if the hamming distance with the previous signal is greater than $\frac{1}{2}$. An extra bus line is needed to indicate this transition. PBM combines an entropy encoder with a bitwise decorrelator[18]. This ensures that the most frequent words create the least transitions. ABE finds clusters of highly correlated bus-lines within a window and performs an XOR operation with the basis line and the cluster. An extra bus line and word are required to propagate the basis

\(^1\)The code can be found at https://github.com/AlexMontgomerie/def
information over the channel. AWR reorders the incoming words such that there are fewer transitions between neighbouring words, using a nearest neighbour algorithm. Extra bus-lines are required to recover the original ordering.

2.2 CNN Accelerators

A number of edge-based accelerators have been proposed for CNNs [21]. Typically, these accelerators are either streaming-architectures [1, 20] or single computation engines [2, 13]. The constraints of edge-devices has made it difficult to faithfully implement CNN networks on them due to the large number of parameters and costly implementation of floating point arithmetic. As such, research has been focused at reducing the size and computational complexity of CNN networks. Commonly used techniques are based on the quantisation of feature-maps/weights as well as in the pruning of the network’s parameters [4, 22]. Quantization schemes are employed to reduce the number of bits used to represent weights and feature-maps, as well as reducing the computational cost of the CNN, with fixed-point becoming the de-facto representation in edge-tailored CNN accelerators [1, 2, 20]. The bit-width of these representations generally range between 1 to 16 bits.

Moreover, a number of proposed CNN accelerators identify off-chip memory as a significant source of power dissipation, with representative examples being the EYERISS [2] and eCNN [6] accelerators. The EYERISS accelerator employs Run Length Encoding (RLE) between off-chip memory and the accelerator hardware. It makes use of the abundance of zeros found in feature-maps, allowing for significant compression to be achieved, reducing the number of memory transactions. In eCNN, a Huffman coding scheme for parameters is introduced in order to reduce the memory transactions. Additionally, power-awareness has been explored in the context of CNN accelerators in [9], where power consumption was modelled for the fpgaConvNet [20] accelerator, which also encapsulated power consumption of off-chip memory. By using this model, a design space exploration tool was able to discover power-efficient designs for given performance targets.

The proposed methodology either departs from the problem setting of the above approaches (in the case of activity minimisation schemes) or provides improved trade-offs between power-consumption reduction and necessary resources for its implementation. This is achieved by providing a domain-specific activity reduction approach, capitalising on the statistical properties and access patterns that are commonly seen in CNN workloads.

3 PROPOSED CODING SCHEME

The proposed coding scheme aims at reducing the dynamic power consumption by reducing the activity along the data bus connected to off-chip memory. The average activity, denoted by $a$, is given in Equation (2) for a bus width of $n$ bits and transmitting a stream, $x$, of $m$ words.

$$a(x) = \frac{1}{m \cdot n} \sum_{i=1}^{m} \sum_{j=0}^{n} x_{i,j} \oplus x_{i-1,j}$$

(2)

where $x_{i,j}$ is the $j^{th}$ bit of the $i^{th}$ word in the stream.

Following a similar approach to many state-of-the art coding schemes [7, 11, 12], a decorrelator is introduced into the encoding path of the scheme. The decorrelating function is described in Equation (3) for the $i^{th}$ word in the stream.

$$d(x_i) = x_i \oplus d(x_{i-1})$$

(3)

Introducing the decorrelator into the encoding path means that the activity of a decorrelated stream can now be viewed as described in Equation (4).

$$\hat{a}(x) = a(d(x)) = \frac{1}{m \cdot n} \sum_{i=0}^{m} \sum_{j=0}^{n} x_{i,j}$$

(4)

This view of activity reveals a far simpler objective of the coding scheme: rather than minimising the number of transitions between two consecutive bits, the popcount (number of ‘1’ bits) per word needs to be minimised.

To achieve a reduction in total ‘1’ bits, first the representation of data being sent along the bus is considered. As quantization of feature-maps is becoming increasingly more common for CNN accelerators, the proposed scheme makes the assumption that the data transmitted across the bus is an $n$-bit fixed point integer. Twos complement (TC) is a common integer representation, however the redundancy of repeated sign bits adds significantly to the activity. Therefore a sign-magnitude (SM) integer representation is adopted [10]. It is worth noting that the full range of the TC value can be preserved in the SM representation, by keeping the value of $-2^{n-1}$ the same in both representations. With the SM representation, the popcount for each integer value is distributed more symmetrically around 0.

This point is illustrated in Figure 1, where the symmetry can clearly be seen.

![Figure 1: Popcount for all 16-bit signed integer values for both TC and SM representation.](image)

Using this representation allows for certain properties regarding the activity to be guaranteed, namely the following:

**Theorem 1.** Let $x$ be a stream of $n$-bit two's complement integers with $m$ elements. Let $y$ be the respective sign-magnitude integer representation of $x$. If, $|x_i| < 2^k \forall i \in [0, m]$, and $k < n$, then the activity of $y$ after it is decorrelated is bounded such that $\hat{a}(y) \leq \frac{k+1}{n}$.

**Proof.** For a sign-magnitude integer, $y_i$, if the absolute value is below a threshold, that is $|y_i| < 2^k$, then the popcount is such that $\sum_{j=0}^{n} y_{i,j} \leq k + 1$. If this bound is held for all words in a stream, $y$ of sign-magnitude integers, then the activity of the stream is bounded such that, $\hat{a}(y) = \frac{1}{m \cdot n} \sum_{i=0}^{m} \sum_{j=0}^{n} y_{i,j} \leq \frac{1}{m \cdot n} \sum_{i=0}^{m} \sum_{j=0}^{n} (k+1) \leq \frac{k+1}{n}$.

As can be seen, a bound on the absolute of the sign-magnitude stream guarantees a bound on the activity also. As such, by decreasing the magnitude of each word in the stream, the activity is also...
The encoder and decoder can be introduced into a CNN accelerator needed for the diff encoder/decoder redundancy. The most significant resource overhead is the buffering scheme is lossless, and requires no additional spatial or temporal redundancy. This complete coding scheme is commonly used across accelerators.

For both streaming architectures [1, 20] and single computation engines [3], the data-access pattern of the feature-map from the off-chip memory, depends on the words must be stored. The proposed coding scheme exploits pixel similarity by taking the difference between an activation and its neighbour of the same channel. This value, \( x_i \), is chosen to be the number of channels the feature-map has, as channel-first streaming, which is illustrated in Figure 2a, is commonly used across accelerators.

**Figure 2: Feature maps in CNN accelerators.**

In this figure, the distance between the middle pixel and all other pixels is taken, and averaged across a sample of images from the Imagenet dataset [14]. Only a single channel of the input feature-map is shown. This figure highlights the fact that neighbouring activations of the same channel of a feature-map on average have a small absolute difference. This property is carried throughout the CNN, although becomes less prominent deeper into the network, due to the translational invariance properties of CNNs.

The proposed coding scheme exploits pixel similarity by taking the difference between an activation and its neighbour of the same channel, as described in Equations (5a) and (5b), for the \( i^{th} \) word in the stream:

\[
\text{(diff encoder)} \quad \hat{x}_i = x_i - x_{i-k} \quad (5a)
\]

\[
\text{(diff decoder)} \quad x_i = \hat{x}_i + x_{i-k} \quad (5b)
\]

where \( k \) indicates the distance in the stream from the neighbouring activation of the same channel. This value, \( k \), depends on the data-access pattern of the feature-map from the off-chip memory. For both streaming architectures [1, 20] and single computation engines [3], \( k \) is chosen to be the number of channels the feature-map has, as channel-first streaming, which is illustrated in Figure 2a, is commonly used across accelerators.

**Figure 3: Overview of a system utilising DEF.**

The complete system for the DEF coding scheme can be seen in Figure 4 for both the encoder and decoder. This complete coding scheme is lossless, and requires no additional spatial or temporal redundancy. The most significant resource overhead is the buffering needed for the diff encoder/decoder, where \( k \) words must be stored. The encoder and decoder can be introduced into a CNN accelerator as shown in Figure 3. They are placed in the compute pipeline such that only encoded feature-maps are sent/retrieved to/from DRAM.

![DEF encoder and decoder](image)

**Figure 4: Block diagrams for DEF coding scheme.**

4 EVALUATION

The proposed coding scheme, DEF, is evaluated against existing state-of-the-art coding schemes. These coding schemes are split into activity minimisation schemes (\( ABE, AWR, PBM \) and BI) and compression schemes (Huffman and RLE). First, the impact of all the coding schemes on switching activity will be evaluated and compared. The impact on power dissipation of off-chip memory will then be compared for an FPGA platform. Stimulus for the coding schemes is generated using the Neural Network Distiller framework [22]. This framework is used to generate quantised feature-maps that are of a signed, asymmetric fixed-point integer type. A random sample of 128 images from the ImageNet dataset [14] are used as inputs for the quantised networks. A channel-first streaming order is assumed. The metrics of interest for this evaluation are the following:

- **Transitions Ratio**: The ratio of bit transitions ("0" to "1" and "1" to "0") of the encoded stream to the unencoded stream, as given in Equation (6). The transitions relate to the dynamic energy consumed on the bus.

\[
tratio = \frac{\text{Total Transitions encoded}}{\text{Total Transitions unencoded}} \quad (6)
\]

- **Average Activity**: The average number of transitions on the bus per word per bus-line, as described in Equation (7). This metric relates to the average dynamic power consumed by the bus.

\[
a_{avg} = \frac{\text{Total Transitions}}{\text{Bus Width} \times \text{Total Words}} \quad (7)
\]

4.1 Activity Minimisation Schemes

A comparison is performed between the mentioned activity reduction schemes for different quantisation levels for the widely used state-of-the-art network MobileNetv2 [15]. This can be seen in Figure 5, where a comparison is made for average activity throughout the MobileNetv2 network. The BI and ABE coding schemes show lower reduction as the bitwidth is increased, whereas PBM improves with increased bitwidth, as well as DEF to some extent. DEF has the lowest activity across most bitwidths (apart from a bitwidth of 2). This figure covers the typical range of bitwidths, however 4-bit, 8-bit and 16-bit are the most common throughout accelerators, and so will be the subject of the rest of the evaluation.
The activity minimisation schemes are further compared in Table 1 for MobileNetv2. The activity and transitions ratio are compared alongside memory usage and bus-width. The memory usage indicates the amount of bytes of storage needed for the implementation of the coding scheme. This table highlights the overheads for each coding scheme and the impact they have on the activity as well as on the number of transitions. ABE, BI and AWR\(^2\) all require extra bus-lines in order to propagate coding information across the bus. It is worth noting that even a single extra bus-line can become prohibitive for certain platforms due to physical bus-width limits. The impact of extra bus-lines on transitions can also be seen. Both ABE and AWR generally increase the number of transitions despite a reduction in activity, suggesting that the use of extra bus-lines may have a negative impact on the energy consumed. This table highlights the improvements DEF brings, where it can be seen to have the lowest transition ratio and activity. This is achieved with no extra bus-lines and minimal impact on resource overheads. Compared to PBM, the only other coding scheme which does not require extra bus-lines, DEF outperforms on both metrics with comparable or less memory overhead.

\(\text{Table 1: Comparison of metrics and usage of activity minimisation schemes for MobileNetv2.}\)

*Figure 6: Activity reduction across all layers of MobileNetv2 for bitwidths of 4, 8 and 16.*

that all coding schemes have a significant amount of variance at a bitwidth of 4. DEF generally has significant reduction for the first few layers, capturing the large pixel similarity observed early on. There is also an increase in activity reduction towards the end, due to the large number of zeros observed in later layers, which the coding scheme is able to exploit.

The activity reduction of the coding schemes is further investigated across seven state-of-the-art CNNs for bitwidths 4, 8 and 16 in Figure 7, where a weighted average is taken across the layers of each network. It can be seen that the reduction is not uniform across CNNs for any given coding scheme or bitwidth. The depth and structure of different CNNs lead to variations in the resulting activity. It can be seen that DEF achieves the highest reduction across all networks for bitwidths of 8 and 16, achieving a reduction between 20% and 50%.

### 4.2 Compression Schemes

The proposed scheme is also compared against popular compression schemes within the ML community. The aim of the comparison is to quantify the extent of impact compression actually has on dynamic energy consumption, and how this compares to a coding scheme which only targets activity reduction. The compression schemes compared are RLE and Huffman. Furthermore, the proposed scheme, DEF, is also combined with RLE to evaluate the impact of the combination of the two schemes on power and energy reduction.

---

\(^2\)The AWR coding scheme is implemented with a window size of 4.
An in-depth comparison is performed between the mentioned approaches using the GoogleNet [19] network, and the results are presented in Table 2, where the compression ratio is also introduced as a metric of interest. The results show that compression has an impact of increasing both activity as well as the number of transitions, as observed in Huffman and RLE approaches. However, by introducing an activity minimisation scheme into the compression scheme, activity and transition reduction can be achieved, as shown by the performance of the DEF+RLE scheme.

Table 2: Comparison of metrics and usage of compression schemes for GoogleNet.

<table>
<thead>
<tr>
<th>Encoding Scheme</th>
<th>$t_{ratio}$</th>
<th>$a_{avg}$</th>
<th>Compression Ratio</th>
<th>Memory (B)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unencoded)</td>
<td>-</td>
<td>-</td>
<td>1.00</td>
<td>0.0</td>
</tr>
<tr>
<td>DEF</td>
<td>0.6212</td>
<td>0.2470</td>
<td>0.2555</td>
<td>1.00</td>
</tr>
<tr>
<td>RLE</td>
<td>2.0020</td>
<td>0.4465</td>
<td>0.3667</td>
<td>1.27</td>
</tr>
<tr>
<td>DEF+RLE</td>
<td>1.0401</td>
<td>0.2787</td>
<td>0.1914</td>
<td>1.42</td>
</tr>
<tr>
<td>Huffman</td>
<td>1.3077</td>
<td>0.4509</td>
<td>0.4956</td>
<td>2.00</td>
</tr>
</tbody>
</table>

To further evaluate the compression schemes and their impact on activity, a new metric referred to as normalised activity is introduced, and given in Equation (8):

$$\bar{a}_{avg} = \frac{a_{avg}}{\text{Compression Ratio}}$$

This models the activity of the bus if the compressed stream was stretched to an uncompressed length. The activity and normalised activity are compared for the compression schemes as well as DEF in Figure 8, for the various bitwidths on GoogleNet. In this figure, the green line represents the baseline activity of the unencoded stream. The results show that DEF achieves the lowest activity. The compressed and normalised activity for Huffman vary significantly, with the normalised activity placing Huffman close to or below the baseline activity. The combination of RLE and DEF does not improve on the activity of DEF on it’s own for either compressed or normalised activity, despite the compression ratio it can achieve.

4.3 FPGA Prototyping

To assess the impact of the proposed coding scheme within a complete CNN accelerator, the power consumption of the memory subsystem supporting a CNN workload is measured on an FPGA platform. A Xilinx ZC702 board is used with a design frequency of 200MHz. The power consumption of the VCC1V5 rail that drives the IO between the FPGA and DRAM as well as the three DRAM chips themselves is measured. In the experiment, encoded streams are sent to and from the FPGA, emulating a typical CNN accelerator design which utilises two of the DRAM chips independently.

<table>
<thead>
<tr>
<th>Encoding Scheme</th>
<th>Average Power (mW)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(unencoded)</td>
<td>1556.2</td>
</tr>
<tr>
<td>ABE</td>
<td>1522.9</td>
</tr>
<tr>
<td>BI</td>
<td>1509.8</td>
</tr>
<tr>
<td>PBM</td>
<td>1546.2</td>
</tr>
<tr>
<td>AWR</td>
<td>1549.7</td>
</tr>
<tr>
<td>DEF</td>
<td>1458.5</td>
</tr>
</tbody>
</table>

Table 3: Power consumption in MobileNetv2 for activity minimisation coding schemes on a XC7020.

The power consumption of memory transactions is given for each activity minimisation scheme in Table 3, when a workload from MobileNetv2 is used. It is worth noting that the idle power consumption of the memory sub-system is around 630mW. The results indicate that DEF is able to reduce the memory subsystem power by 100mW.
The power reduction of the coding schemes is also compared in Figure 9. DEF is able to achieve a 6% reduction in power consumption, which is 2x greater than the next comparable coding scheme, BI. Please note that reduction in reported power compared to reported activity is lower, as the active power of the DRAM chips are also taken into account. The coding schemes mostly impact IO power, and their gains are weighted accordingly to their contribution to the overall active power of the memory subsystem.

The compression coding schemes are also evaluated on the FPGA platform. Power, energy and execution time are given for the first layer of AlexNet in Table 4. The table highlights the significance of the relative power consumption between IO and the DRAM chips, as although DEF is able to reduce the power to some extent, ultimately Huffman is able to transfer the feature-maps more energy-efficiently as the DRAM is active for less time.

<table>
<thead>
<tr>
<th>Encoding Scheme</th>
<th>Power (mW)</th>
<th>Time (ms)</th>
<th>Energy (µJ)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(uncoded)</td>
<td>1534.2</td>
<td>5.519</td>
<td>8466.7</td>
</tr>
<tr>
<td>Huffman</td>
<td>1556.8</td>
<td>1.383</td>
<td>2152.9</td>
</tr>
<tr>
<td>DEF</td>
<td>1397.7</td>
<td>5.526</td>
<td>7723.8</td>
</tr>
<tr>
<td>DEF+RLE</td>
<td>1465.0</td>
<td>2.519</td>
<td>3690.7</td>
</tr>
</tbody>
</table>

Table 4: Power, Execution Time and Energy of compression schemes for AlexNet on a XC7020.

Finally, the overheads of implementing DEF on a XC7020 FPGA are evaluated. The maximum resources needed for MobileNetv2 for 8-bit feature-maps are as follows,

| FF: 77, LUT: 344, BRAM: 0, DSP: 0 |

In total, the feature-map with the greatest number of channels (1280) of MobileNetv2’s workload will still only require less than 1% of the total board resources. This means that the power overhead for the implementation will be insignificant also.

5 CONCLUSION

The paper proposed a novel encoding scheme for activity reduction, tailored to CNN workloads when communication with off-chip memory is required. The coding scheme is able to achieve up to 50% reduction in activity for feature-maps of CNNs, proving its suitability for this specific domain. The impact of activity reduction is demonstrated on an FPGA platform, and the coding scheme is able to reduce the power consumption of a memory subsystem by 6%, with minimal overhead.