Constructing energy-efficient mixed-precision neural networks through principal component analysis for edge intelligence

Indranil Chakraborty*, Deboleena Roy, Isha Garg, Aayush Ankit and Kaushik Roy

The ‘Internet of Things’ has brought increased demand for artificial intelligence-based edge computing in applications ranging from healthcare monitoring systems to autonomous vehicles. Quantization is a powerful tool to address the growing computational cost of such applications and yields significant compression over full-precision networks. However, quantization can result in substantial loss of performance for complex image classification tasks. To address this, we propose a principal component analysis (PCA)-driven methodology to identify the important layers of a binary network, and design mixed-precision networks. The proposed Hybrid-Net achieves a more than 10% improvement in classification accuracy over binary networks such as XNOR-Net for ResNet and VGG architectures on CIFAR-100 and ImageNet datasets, while still achieving up to 94% of the energy efficiency of XNOR-Nets. This work advances the feasibility of using highly compressed neural networks for energy-efficient neural computing in edge devices.

One approach to compressing neural network models is to modify the network architecture itself, so that it has fewer parameters. One such example is SqueezeNet10. Another method of compression is pruning, which aims to reduce redundancies in over-parameterized networks. To that effect, researchers have investigated several network pruning techniques, both during training11,12 and inference13,14. A different technique of model compression is to represent weights and activations with reduced precision. Quantized networks15 help achieve a reduction in energy consumption as well as improve memory compression compared to full-precision networks. Binary neural networks16 are an extreme case of quantization where the activations and weights are reduced to binary representations. These networks drastically reduce the energy consumption by replacing the expensive multiply and accumulate (MAC) operations with simple add or subtract operations. These massive reductions in memory usage and computational cost make them particularly suitable for edge computing. However, despite these benefits, the networks suffer from performance and scalability issues, especially for complex pattern recognition tasks. Several training algorithms17 have been proposed to optimize network performance to achieve state-of-the-art accuracy in extremely quantized neural networks. Although such training methodologies recover the performance hit caused by binarization weights alone, they fail to completely counter the degradation caused by binarizing both weights and activations.

In this work, we present Hybrid-Net, a mixed-precision network topology fashioned by the combination of binary and high-precision inputs and activations in different layers of a network. We use principal component analysis (PCA) to determine the significance of layers based on the ability of a layer to expand data into higher-dimensional space, with the ultimate aim of linear separability. Viewing a neural network as an iterative projection of the input onto a successively higher-dimensional manifold at each layer, until the data are eventually linearly separable, allows us to identify layers that contribute relevant transformations. Following the algorithm in ref. 18, we find the ‘significant dimensions’ in a layer as the number of dimensions that cumulatively explain 99% of the total variance of the layer. To that effect, we identify the ‘important layers’ in a network and design a mixed-precision neural network that combines binary weights and activations with high-precision quantization in select layers. This work contributes to the feasibility of using highly compressed neural networks for energy-efficient neural computing in edge devices.
variance of the output activation map generated by that layer. Because we want the data to be expanded into higher dimensions at each layer, we deem the layers at which significant dimensions increase from the previous layer as significant. Following the identification of significant layers, we increase the bit precision of the inputs and weights of those layers, keeping the rest of the layers entirely binary. Traditionally, PCA has been used primarily as a dimensionality reduction technique. It was also recently used to identify redundancies in different layers of a neural network and prune out the redundant features\(^5\). We propose a methodology where we use PCA in a reverse manner, that is, to increase the precision of the important layers. Hybrid-Net remarkably improves the performance of extremely quantized neural networks, while keeping the activations and weights of most of the layers binary. This ensures low energy consumption and high memory compression of extremely quantized neural networks, while achieving significantly enhanced classification performance compared to binary networks such as XNOR networks. This work not only achieves significant progress in the challenge of quantizing neural networks to binary representations, but also paves way for optimized yet highly accurate quantized networks suitable for enabling intelligence at the edge.

Related work
Various techniques have been proposed to improve the performance of quantized networks. Fully binary networks\(^6,7\) are constructed by replacing the activations with their sign. However, these networks usually suffer from significant degradation in accuracy, especially for larger datasets such as CIFAR-100 and ImageNet. One intuitive way of recovering quantization errors is using wider networks\(^8\), but this comes at the cost of increased energy consumption. There have been efforts focusing on gradient calculations for approximated sign functions to ameliorate the effect of binarization\(^9\). More general quantization schemes have also been explored for weights and activations\(^10,11\). Although weight quantization can be compensated by training the network with quantized weights\(^12\), it has been observed that input quantization poses a serious challenge to classification performance for precisions lower than 4 bits. One approach that addresses this challenge involves clipping the activations by setting an upper bound. Although this approach seems to be heuristic, recent efforts have focused on using trainable quantization that can be dynamically manipulated\(^13,14\). One such approach involves parameterized clipping, where the clipping level is dynamically adjusted through gradient descent\(^15\). Another approach proposes the use of batch-normalization layers after rectified linear unit activations to bound the activation values for effective quantization\(^16\). Note that most of these works focus on optimizing the activations when the quantization precision is 2 bits or more. Binary networks with both 1 bit activations and weights, despite offering the most benefits in terms of computation cost and memory compression, still suffer from significant degradation in performance compared to full-precision networks.

An alternative path towards improving the accuracy of binary neural networks focuses on network design techniques. To that effect, improved input representations through shortcut connections in deep networks can significantly improve the performance of binary neural networks without any increase in computation cost\(^17\). This is because shortcut connections are usually identity in nature and do not comprise expensive MAC operations. Combinations of different kinds of input precision have also been explored across different layers to circumvent the significant decrease in classification accuracy of such binarized networks\(^18\). There has been considerable effort in making the search for optimum neural architecture more sophisticated through efficient design space exploration\(^19\). A theoretical approach to predicting layer-wise precision requirements has been also explored\(^20\). Our work differs from most of the current efforts in quantized neural networks as it lies in the realm of hybrid network design for more optimal performance of neural networks where most of the layers still have 1 bit weights and activations. This motivated us to propose an algorithm to identify important layers and judiciously reinforce those particular layers with higher bit precision representation. To follow such a motivation, it is necessary to understand the significance of layers, which we explain in the next section.

**PCA-driven Hybrid-Net design**
A Hybrid-Net is a neural network that employs two different bit precisions for its weights and activations. The base network is of low precision, for example 1 bit, and certain layers are selected and set to a higher bit precision. To select the layers, we use PCA on the output feature maps of each of the layers. The input to any layer is binarized and convolved with the weight filters. We perform PCA on the resulting output tensor. This is performed individually on the output tensor of every layer. Given any set of correlated variables, such as the feature maps, PCA does an orthogonal transformation to map them to uncorrelated variables called principal components (PCs), which also form the orthogonal basis set for these tensors. Each of these resulting basis vectors identify directions of varying variance in the data, and are ordered in decreasing manner, with the first vector in the direction of highest variance. We perform such PCA on each convolutional layer in a standalone fashion based on the transformation that it applies on its input. It is applied only on the linear portion of the network before nonlinear activation. The aim of PCA is to identify redundancies and how many filters in that layer are necessary to give us a near perfect reconstruction of the output. Based on the redundancy data obtained from the PCA, we define our own significance metric to obtain significant layers from the network.

In a neural network, each layer applies a transformation on its input and projects it to a new feature space of ideally higher or the same dimension, with the objective of achieving linear separability. PCA provides the ability to study the directions of maximum variance in the input data. The pre-rectified linear unit activation map generated by a filter is considered to be composed of many instances of that particular filter. Performing PCA and finding the number of filters needed to explain a pre-defined cumulative percentage of variance identifies the number of significant dimensions for each layer. Higher number of principal components, needed to preserve a significant percentage (say \(T\)) of the total variance in the input, correspond to lesser redundant information carried by those tensors. This implies that those tensors have higher significant dimensionality. Ideally, we want the number of PCs, required to explain \(T\%\) of the total variance of the feature space, to increase as we move deeper into the network. This is to ensure extraction of more uncorrelated, unique features from the data, and projection into a higher-dimensional space that will eventually lead to linear separability at the classifier layer. Thus, the layers for which the number of PCs explaining variance in the output data is more than that in the input data contribute to significant transformations on the input data. Note that the significant dimensionality of layers does not always increase monotonically. However, regardless of the trend, it provides us with a way of judging the pliability of a layer to binarization. In this section, we propose a methodology to identify these significant layers and subsequently design mixed-precision networks by increasing the bit precision of those layers.

**Algorithm 1.** Hybrid-Net design methodology

| Function \(PCA(\text{activations}, \text{layer}, T)\) |
| 1. \([M,H,W,O]\) ← size(activations[\text{layer}]); |
| 2. \(\text{act\_fl}\) ← flatten(activations[\text{layer}],\(M\times H\times W\times O\)); |
| 3. run PCA on \(\text{act\_fl}\); |
| 4. \(\text{tot\_var}\) ← total variance; |
| 5. \(\text{cum\_var}\) ← cumulative sum of variance; |
PCA-driven identification of significant layers. We perform our analysis on activations of each layer, which provide a notion of activity of each filter in that layer. Let us consider the activation matrix of the $l$th layer, $X_l$. Layer $L$ has $O$ filter banks, each containing $F$ filters of size $k \times k$. $I$ and $O$ are the number of input and output channels. The first element of the $i$th output map of $X_l$ is the result of convolution of the first $k \times k \times I$-sized input patch with the $i$th filter bank. The rest of the elements of any particular output map of $X_l$ can be obtained by sliding over the entire input. Thus, if we consider $M$ as the size of a minibatch, $X_l$ is a four-dimensional (4D) tensor of size $M \times H \times W \times O$ where $H$ and $W$ are the height and width of the output. If we flatten the 4D data $X_l \in \mathbb{R}^{M \times H \times W \times O}$ into a 2D matrix $Y_l \in \mathbb{R}^{M \times F \times H \times W \times O}$, we would obtain $M \times H \times W$ samples, each containing $O$ elements equivalent to the number of filter banks. This process is shown in Fig. 1a.

When PCA is performed over the aforementioned 2D matrix $Y_l$, the singular value decomposition of the mean normalized, symmetric matrix $Y_l^T Y_l$ generates $O$ eigenvectors $\nu_i$ and eigenvalues $\lambda_i$. The total variance in the data is given by sum of the variances of individual parameters:

$$\text{Var} = \sum_{i=1}^{O} (\sigma_i^2) = \text{Tr}(Y_l^T Y_l)$$

(1)

The contribution of any component, $\lambda_i$, towards the total variance can be expressed as $\lambda_i / \text{Var}$. To calculate the number of significant components, we set a threshold value $T$, which is the amount of variance the first $k$ significant components are able to explain. This can be expressed as

$$\sum_{i=1}^{k} (\lambda_i^2) / \sum_{i=1}^{O} (\lambda_i^2) = T$$

(2)

An example of a typical curve of the cumulative sum of variance for different filter numbers, obtained by PCA, is shown in Fig. 1a (right). As the PCA analysis produces the $k$ most significant components to explain the $T$ fraction of the total variance, we proceed to identify the significant layers. We define a significant layer as a layer that transforms the input data such that the number of significant components to explain the $T$ fraction of the variance increases from that required for the output of the previous layer. Let $k_i$ be the number of significant components corresponding to the $i$th layer. It can then be said that layer $i$ contributes a relevant transformation on the input data if $k_i > k_{i-1}$. This means that the layer requires more significant components to explain the variance in the data at its output than the previous layer. However, for better control on deciding the important layers, we check the condition $k_i - k_{i-1} > \Delta$ to determine if the $i$th layer is significant. This is explained in Fig. 1b (middle), where the red symbols denote the significant layers where $k_i - k_{i-1} > \Delta$.

**Hybrid-Net design.** The PCA analysis helps us identify a set of important layers in an $N$-layer network. We have designed a hybrid network, ‘Hybrid-Net’, where we set the bit precision of weights and inputs of the important layers to a higher value, $k_b = 2, 4$, than the other layers, which have binary weights and inputs. This is shown in Fig. 1b (right). The weights and inputs of the first and final layers of an $N$-layer network are kept full-precision, according to standard practice\cite{12,22,23}. The quantization algorithm for any $k_b$-bit quantized layer can be readily derived from XNOR-Net\cite{11}, where the quantized levels are

$$q_{k_b}(x) = 2 \left(\frac{(2^b-1)(x+1)/2 - 1}{2^b - 1} \right)$$

(3)

In a layer with $k_b$-bit weights and activations, $q_{k_b}(x)$ is used instead of the sign function in layers with binary weights and activations. We use a slightly modified version of quantized networks, proposed in ref. \cite{1}, where the weights have a scaling factor $\alpha$ instead of just being quantized. The convolution operation between inputs $X$ and weights $W$ in such a network is approximated as

$$X^W \approx (\text{sign}(X)^* \text{sign}(W)) \odot \alpha$$

(4)

$$X^W \approx (q_{k_b}(X)^* q_{k_b}(W)) \odot \alpha$$

(5)

Here, $\alpha$ is the L1-norm of $W$ and acts as a scaling factor for the binary weights. In binary layers, the activation gradients are clipped such that they lie between $-1$ and $1$. In the $k_b$-bit layers, we get rid of the activation gradient clipping for better representation. Each layer of an $N$-layer Hybrid-Net has either binary or $k_b$-bit weight kernels, and the activations after each convolution are again quantized before passing to the next layer.

Hybrid-Net is expected to have a higher computation cost than a binary network. The parameter $\Delta$ decides the number of important layers to consider and hence a penalty is incurred due to the increase in bit precision. We can estimate the penalty in computation cost incurred due to the increase in bit precision in a network with $L_0$ significant layers as

$$\text{Penalty} = \sum_{i=1}^{N} B_i + \sum_{i \notin \text{SigLayer}} B_i \times p$$

(6)

Here $B_i$ is the computation cost of a binary layer and $p$ is the overhead of $k_b$-bit computation over binary computation. We will present a detailed analysis of energy consumption and memory usage later in this Article.

For residual networks, ResNets, we include another design feature in addition to the PCA-driven Hybrid-Net, improving input representations through residual connections. This has been
alluded to in ref. 20, where adding identity shortcut connections at every layer improves the representational capability of binary networks. In standard residual networks, such identity connections are added to address the vanishing gradient problem in deep neural networks. However, in the case of binary networks, these connections serve to provide an improved representation by carrying floating-point information from the previous layer. As a result, the Hybrid-Net design also considers the effect of adding such highway connections at every layer. Note that, in the case of convolution layers, which induce a change in size of each feature map, the

Fig. 1 | Illustration to show PCA analysis and subsequent Hybrid-Net design. a, The PCA function in Algorithm 1 for a particular layer, showing flattening of an instance of a 4D tensor and subsequent PCA analysis, yielding the plot showing how the cumulative explained variance accrues with the number of filters in a particular layer. The number of significant filters \( k \) is defined as the number of filters at which the cumulative sum reaches a threshold (say 0.99). b, The Main() function in Algorithm 1 is explained as we take a standard binary network (leftmost, with first and final layers having full-precision weights and activations) and perform the aforementioned PCA function on each binary layer (shown in white). The resulting plot (middle) shows the layer-wise variation in \( k \) in a ResNet-18 on ImageNet, for example. FC denotes fully connected layer. A layer is considered significant (red symbols) when \( k \) increases by at least \( \Delta \). The new Hybrid-Net (rightmost) is designed by increasing the precision of weights and activations for the significant layers (shown in blue).
shortcut connections consist of 1×1 convolution weight layers to account for the change in size.

Experiments, results and discussion
Experiments. We evaluated the performance of all the networks described in this section in PyTorch. We performed image classification on the datasets CIFAR-100 and ImageNet. The CIFAR-100 dataset has 50,000 training images and 10,000 testing images of size 32×32 in 100 classes. For the CIFAR-100 dataset, we explored the proposed Hybrid-Net design methodology on standard network architectures, ResNet-20, ResNet-32 and VGG-15 (VGG, Visual Geometry Group), where the training algorithm for the quantized layers was adopted from ref. 16. We extended our analysis to the ImageNet dataset, which is the most challenging dataset pertaining to image classification tasks. It consists of 1.2 million training images and 50,000 validation images divided into 1,000 categories. For simplicity, we considered ResNet-18 for our ImageNet evaluation. To compare against the proposed Hybrid-Net designs, we explored different network configurations as baselines for ResNet (Fig. 2a) and VGG architectures (Fig. 2b). XNOR-Net is the base skeleton network on which we designed other networks. Binary-Shortcut is the same as XNOR-Net, except that it has residual connections in every layer (similar to ref. 16). Hybrid-Comp A is formed by inter-layer sectioning, that is, dividing the network into two parts (N−k binary and k k-bit layers), where N is the number of layers between the first and last layer. The widths of the network architectures, shown in Fig. 2a,b, are for the CIFAR-100 dataset. For ImageNet, we used a wider network architecture, which we describe in Table 1. We have also compared our proposed Hybrid-Net designs against state-of-the-art quantized networks (for example, refs. 21,23,25) for ImageNet. We performed simulations for five different random initializations for all networks on the CIFAR-100 dataset. For simplicity, we performed simulations for three different random initializations on four selected networks and one initialization for the rest of the networks on the ImageNet dataset. The accuracy reported for both datasets is the top–1 accuracy, and the mean±s.d. accuracy provides the mean and standard deviation of accuracies obtained for different random initializations.

Energy efficiency and memory compression. We have briefly alluded to the possible penalty incurred due to increasing the bit precision.
of certain layers in a network. To identify its effect with respect to
the entire network metrics and further illustrate the benefits of the
proposed Hybrid-Nets, we performed a storage and computation
analysis to calculate the energy efficiency and memory compression
of the proposed networks. For any two networks A and B, the
energy efficiency and memory compression of network A with
respect to network B can be defined as

\[
\text{Energy efficiency (EE)} = \frac{E_A}{E_B}
\]
\[
\text{Memory compression (MC)} = \frac{M_A}{M_B}
\]

where \(E_A\) and \(E_B\) refer to the energy consumed by network A and
network B, respectively, and \(M_A\) and \(M_B\) are the memory used for
storing the weights of network A and network B, respectively. We
estimate the energy efficiency (EE) and memory compression
(MC) with respect to a full-precision network and normalize it
with respect to an XNOR-Net network, which is an entirely binary
network except for the first and final layers. Thus, the normalized
energy efficiency (EE\textsubscript{norm}) and normalized memory compression
(MC\textsubscript{norm}) of any network A can be written as

\[
\text{EE}(A) = \frac{\sum E_i(FP)}{\sum E_i(A)}
\]
\[
\text{MC}(A) = \frac{\sum M_i(FP)}{\sum M_i(A)}
\]
\[
\text{EE}_{\text{Norm}}(A) = \frac{\text{EE}(A)}{\text{EE}(\text{XNOR})}
\]
\[
\text{MC}_{\text{Norm}}(A) = \frac{\text{MC}(A)}{\text{MC}(\text{XNOR})}
\]

Here, \(E_i(FP)\) and \(M_i(FP)\) are the energy and memory, respectively,
consumed by the \(i\)th layer of a network with full-precision weights and activations, while \(E_i(A)\) and \(M_i(A)\) are the energy and memory
consumed by the \(i\)th layer of any network A under consideration.

Results of PCA. We performed PCA analysis on the activations of
each convolutional layer and extracted the number of PCs required
to explain a \(T\) fraction of variance in the data. Design parameters
such as \(T\) and subsequently \(\Delta\) are heuristic. For all analyses we fix
\(T=99\%\), as this makes the increases in significant components \(k\)
across various layers clearly distinguishable. The \(\Delta\) values are
chosen based on the variation in \(k\) across layers. A higher \(\Delta\) value yields a
smaller number of significant layers. For clarity, we perform our
analysis for various \(\Delta\) values.

ResNet architecture CIFAR-100. For ResNet architectures, we per-
formed PCA on a plain version of a binary network devoid of any
residual connections. We decided to do this to isolate the effect of
the convolution layers on the activations, instead of having residual
connections. This is done because we focus on the quantization of
the filters of the layers and the residual additions may distort the out-
put feature space and hence the information we seek from it. Figure
3a,b shows the variation in the number of filters required to explain
\(T=99\%\) with different layers for ResNet-20 and ResNet-32 archite-
cuctures, respectively. As expected, the maximum change in \(k\) occurs
when the number of output channels increases. However, we observe
a trend in both networks where the layers just after the output chan-
nels increasing from, say, 16 to 32 or 32 to 64, account for the maxi-
mum change in the number of significant filters. Based on our criteria
for significant layers, discussed in 'PCA-driven identification of sig-
nificant layers,’ we fix \(\Delta = 1\) for ResNet-20 and \(\Delta = 4\) for ResNet-32.
to identify the layers where the number of significant components undergoes a change of more than Δ. Figure 3a also shows those layers, marked by red symbols. Note that, by varying Δ, more or less layers can be considered significant. After performing this analysis on a plain version of the ResNet architecture, we perform network simulations on the standard version with residual connections.

**VGG architecture CIFAR-100.** For the VGG architecture, we performed PCA on a binary network that has binary weights and activations for all layers except the first and last. Figure 3c presents a plot showing how the number of filters required to explain T = 99% of the variance changes with different layers for a VGG-15 architecture. We observe that the number of significant filters mostly increases when the number of filter banks increases at a particular layer. For the rest of the layers, it remains fairly constant. As the PCA plot shows very little variation across layers, we consider a relatively lower Δ = 3 with respect to the number of filters. Significant layers are shown as red symbols in Fig. 3c. Table 2 lists the different combinations of significant layers obtained from ResNet and VGG architectures through the PCA analysis for different Δ values for the CIFAR-100 dataset. Note that we did not choose a lower Δ for ResNet-32, as this would have included many layers and increase the computation cost without a significant benefit in accuracy.

**ResNet architecture ImageNet.** We also performed PCA analysis on the ResNet-18 architecture for the ImageNet dataset. Figure 3d presents a plot showing how the number of filters required to explain T = 99% of the variance changes with different layers for ResNet-18 with Δ = 10. The significant layers identified by our proposed methodology are marked by red symbols. We observe a similar trend as for CIFAR-100, where the maximum increase in the number of significant filters, k, occurs in the first few layers after every change in filter size. We performed the PCA analysis for Δ = 10, 20, 30 to identify the significant layers, as listed in Table 2.

**Image classification results for CIFAR-100.** The classification results for CIFAR-100. The ResNet-N architecture consists of N−1 convolution layers and a fully connected classifier. As discussed before, the first convolution layer and the classifier have full-precision inputs and weights. For the CIFAR-100 dataset, we consider N = 20 and N = 32. We also consider a slightly modified version of ResNet, where we add identity shortcut connections at every layer instead of every two layers for better input representation, as discussed earlier. We increase the bit precision of the weights and inputs of the layers obtained from PCA analysis to bit-precisions k_w = 2 and k_i = 4 to form Hybrid-Net (k_w, k_i). The rest of the layers have binary representations for weights and inputs. We also compare the proposed Hybrid-Net with Hybrid-Comp A (k_w, k_i) (k), which is formed by splitting the entire network into N−k binary and k_k-bit sections. Table 3 shows the accuracy, energy efficiency and memory compression of the proposed Hybrid-Net based on ResNet-20 and ResNet-32 in comparison to XNOR-Net and the other kinds of hybrid network discussed in Fig. 2.

We observe that the proposed Hybrid-Net achieves a much superior tradeoff between accuracy, energy efficiency and memory compression than other kinds of hybridization technique. Moreover, in the case of both ResNet-20 and ResNet-32, Hybrid-Net increases the classification accuracy by 10–11% compared to an XNOR-Net, with minimal degradation in efficiency and compression. For example, Hybrid-Net (2,2) (Δ = 1) based on ResNet-20 can be expected to have only 36 ± 0.03% of the accuracy degradation of an XNOR network with respect to a full-precision network, while only costing 13% extra energy. Although quantizing the entire network to 2-bit inputs and weights (Quantize (2,2)) achieves a slightly higher accuracy, we show that our principle of increasing the bit precision of a few significant layers captures most of the increase in accuracy from an XNOR-Net to a 2-bit network. Hybrid-Net thus consumes 14% less energy and 12% less memory for ResNet-20 than a 2-bit network, with a performance within 2% of the latter. For ResNet-32, the benefits of Hybrid-Net are pronounced, as it consumes 24% less energy and 26% less memory than a 2-bit network with only 4% degradation in accuracy in comparison to the latter. Hybrid-Net thus ensures a significant improvement in accuracy over a binary network, without making the entire network 2-bit. We also show that Hybrid-Net achieves a higher accuracy than Hybrid-Comp A networks while consuming less energy for both ResNet-20 and ResNet-32, thus demonstrating the effectiveness of the design methodology.

**Image classification results for ImageNet.** We evaluate the proposed Hybrid-Net design in Table 4. The XNOR network is seen to suffer a significant degradation in accuracy from a full-precision network. Even the Binary-Shortcut 1 network, with residual connections at every layer, fails to recover the classification accuracy. With improved quantization schemes and weight update mechanisms, Bi-Real Net shows a slightly higher accuracy. Compared to these binary networks, we observe that the proposed Hybrid-Net, considering both 2- and 4-bit weights and activations, achieves up to and over 10% higher accuracy than the corresponding XNOR network. In particular, Hybrid-Net (2,2) (Δ = 20) can be expected to have only 38.6 ± 0.002% of the accuracy degradation of an XNOR network with respect to a full-precision network, while only costing 7% extra energy. Quantizing the activations and weights of the

| Network arch | Significant layers |
|--------------|--------------------|
| CIFAR-100    |                    |
| ResNet-20 (Δ = 1) | 8, 9, 10, 14, 15, 16, 18 |
| ResNet-20 (Δ = 2) | 8, 9, 14, 15 |
| ResNet-32 (Δ = 4) | 12, 13, 22, 23, 24 |
| VGG-15 (Δ = 3) | 3, 5, 8, 11, 12 |
| VGG-15 (Δ = 10) | 3, 5, 8 |
| ImageNet     |                    |
| ResNet-18 (Δ = 30) | 6, 10, 14, 15 |
| ResNet-18 (Δ = 20) | 6, 10, 11, 14, 15, 16 |
| ResNet-18 (Δ = 10) | 6, 7, 10, 11, 14, 15, 16 |
entire network to 2 bits can further increase the accuracy by 1–2%, but at the cost of a 15–20% increase in energy consumption than Hybrid-Net. Note that we have provided results as baselines for different input quantization algorithms, such as DoReFA-Net\textsuperscript{21}, LQ-Net\textsuperscript{23} and PACT\textsuperscript{25}, for the Quantize (2,2) network, although we have used the XNOR quantization (described in section ‘Hybrid-Net design’) in this work. We also show that Hybrid-Net (2,2) achieves up to 7.4% higher accuracy with respect to other methods of hybridization\textsuperscript{27}. This work shows that increasing the bit precision of a few significant layers can remarkably boost the performance of binary neural networks without making the entire network higher precision. Note that using improved quantization schemes\textsuperscript{26,27} can potentially further increase the accuracies of the proposed Hybrid-Nets.

**Statistical analysis.** We mentioned earlier that we have performed simulations for various random initializations. To understand the amount of variations in the results, we performed simulations for two cases:

1. Fixed optimal solutions: we define a binary base network with any random initialization. We run a PCA analysis and obtain an optimal Hybrid-Net. Next, we train the Hybrid-Net with various random initializations.
2. Varying optimal solutions: we train the base binary network with different random initializations and run PCA analysis on each case. This provides various optimal solutions of Hybrid-Net, with different combinations of significant layers. We then perform accuracy simulations for corresponding random initializations.

### Table 3 | Comparison of different networks on CIFAR-100

| ResNet-20 | FP accuracy 69.49% | EE(XNOR) 16.35, MC(XNOR) 17.26 |
|-----------|---------------------|----------------------------------|
| Network type | Best accuracy (%) | Mean ± s.d. accuracy (%) | EE\textsubscript{norm} | MC\textsubscript{norm} |
| XNOR | 50.50 | 50.23 ± 0.21 | 1 | 1 |
| Binary-Shortcut 1 | 54.16 | 53.92 ± 0.21 | 0.99 | 1 |
| Hybrid-Net (2,2) (\(\Delta = 1\)) | 62.84 | 62.5 ± 0.24 | 0.87 | 0.77 |
| Hybrid-Net (2,2) (\(\Delta = 2\)) | 60.93 | 60.53 ± 0.39 | 0.93 | 0.88 |
| Hybrid-Net (4,4) (\(\Delta = 1\)) | 63.88 | 63.38 ± 0.49 | 0.7 | 0.53 |
| Hybrid-Net (4,4) (\(\Delta = 2\)) | 61.62 | 61.53 ± 0.1 | 0.82 | 0.7 |
| Quantize (2,2) | 65.81 | 65.19 ± 0.49 | 0.73 | 0.65 |
| Hybrid-Comp A (2,2)(\(k = 6\)) | 62.36 | 61.79 ± 0.34 | 0.88 | 0.71 |
| XNOR2x | 63.03 | 62.81 ± 0.14 | 0.39 | 0.33 |

| ResNet-32 | FP accuracy 70.62% | EE(XNOR) 18.42, MC\textsubscript{Norm} 20.44 |
|-----------|---------------------|----------------------------------|
| Network type | Best accuracy (%) | Mean ± s.d. accuracy (%) | EE\textsubscript{norm} | MC\textsubscript{norm} |
| XNOR | 53.89 | 53.48 ± 0.27 | 1 | 1 |
| Binary-Shortcut 1 | 58.98 | 58.23 ± 0.61 | 0.99 | 1 |
| Hybrid-Net (2,2) (\(\Delta = 4\)) | 64.34 | 63.75 ± 0.39 | 0.94 | 0.87 |
| Hybrid-Net (4,4) (\(\Delta = 4\)) | 64.45 | 64.28 ± 0.18 | 0.84 | 0.69 |
| Quantize (2,2) | 68.04 | 67.73 ± 0.21 | 0.7 | 0.61 |
| Hybrid-Comp A (2,2) | 62.41 | 62.15 ± 0.2 | 0.91 | 0.76 |
| XNOR2x | 65.20 | 65.11 ± 0.07 | 0.38 | 0.31 |

| VGG-15 | FP accuracy 68.31% | EE(XNOR) 21.77, MC\textsubscript{Norm} 26.24 |
|--------|---------------------|----------------------------------|
| Network type | Best accuracy (%) | Mean ± s.d. accuracy (%) | EE\textsubscript{norm} | MC\textsubscript{norm} |
| XNOR | 54.30 | 54.23 ± 0.1 | 1 | 1 |
| Hybrid-Net (2,2) (\(\Delta = 3\)) | 61.81 | 61.67 ± 0.08 | 0.84 | 0.75 |
| Hybrid-Net (2,2) (\(\Delta = 10\)) | 60.13 | 59.87 ± 0.25 | 0.93 | 0.92 |
| Hybrid-Net (4,4) (\(\Delta = 3\)) | 63.38 | 63.12 ± 0.15 | 0.64 | 0.5 |
| Hybrid-Net (4,4) (\(\Delta = 10\)) | 60.37 | 60.06 ± 0.18 | 0.81 | 0.8 |
| Quantize (2,2) | 68.90 | 68.63 ± 0.28 | 0.65 | 0.55 |
| Hybrid-Comp A (2,2)(\(k = 3\)) | 58.01 | 57.46 ± 0.38 | 0.85 | 0.72 |
| XNOR2x | 58.24 | 57.35 ± 0.54 | 0.29 | 0.3 |
Articles

Fixed optimal solutions. We performed simulations for all cases explained in this Article for five different random initializations. For simplicity, we considered three random initializations for four networks for ImageNet. Note that the energy and memory costs depend on the network architecture and are fixed for a particular architecture. The results in Tables 3 and 4 show that the variations in accuracies are within 2% of the best accuracies.

Varying optimal solutions. We trained the base binary network for ResNet-20 on CIFAR-100 for different initializations and performed PCA analysis on each of them to obtain varying optimal solutions for different $\Delta$ values. We observe that the optimal solutions overlap quite significantly. The resulting energy efficiency and memory compression metrics also do not vary remarkably.

We performed accuracy analysis for each optimal solution for the corresponding random initialization to observe variations in the results. The results are presented in Table 5. Based on the analysis, we can expect Hybrid-Net (2,2) ($\Delta = 1$) to have only 37.7 ± 0.007% of the accuracy degradation of an XNOR network with respect to a full-precision network, with only 12 ± 1.22% extra energy cost. Similarly, Hybrid-Net (2,2) ($\Delta = 2$) is expected to have only 41 ± 0.01% of the accuracy degradation of a XNOR network with respect to a full-precision network, with only 9 ± 1.4% extra energy cost.

Table 4 | Comparison of different networks on ImageNet

| Network type       | Best accuracy (%) | Mean ± s.d. accuracy (%) | EE_Norm | MC_Norm |
|--------------------|-------------------|--------------------------|---------|---------|
| XNOR               | 50.33             | -                        | 1       | 1       |
| Binary-Shortcut 1  | 54.36             | 54.15 ± 0.15             | 1       | 1       |
| Bi-Real Net        | 56.9              | -                        | 1       | 1       |
| Hybrid-Net (2,2) ($\Delta = 30$) | 60.38 | 59.75 ± 0.44 | 0.96     | 0.87    |
| Hybrid-Net (2,2) ($\Delta = 20$) | 61.95 | 61.89 ± 0.04 | 0.93     | 0.8     |
| Hybrid-Net (2,2) ($\Delta = 10$) | 62.73 | - | 0.92 | 0.8 |
| Hybrid-Net (4,4) ($\Delta = 30$) | 61.70 | 60.54 ± 0.84 | 0.89 | 0.7 |
| Quantize (2,2)     |                   |                          |         |         |
|                    |                   |                          |         |         |
| Table 5 | Analysis of the effect of random initialization on varying optimal Hybrid-Net architecture (ResNet-20 on CIFAR-100)

| $\Delta$ | Initialization | Significant layers | Best accuracy (%) | Mean ± s.d. accuracy (%) | EE_Norm | MC_Norm |
|----------|----------------|--------------------|-------------------|--------------------------|---------|---------|
| 1        | 1              | 8, 9, 10, 14, 15, 16, 18 | 62.84 | 62.46 ± 0.23 | 0.88 | 0.77 |
| 2        | 2              | 5, 8, 9, 14, 15, 16 | 62.14 | 61.59 ± 0.29 | 0.89 | 0.82 |
| 3        | 3              | 7, 8, 9, 14, 15, 16 | 62.17 | 61.85 ± 0.19 | 0.89 | 0.82 |
| 4        | 4              | 2, 8, 9, 10, 14, 15, 16, 18 | 63.4 | 63.15 ± 0.13 | 0.86 | 0.77 |
| 2        | 2              | 8, 9, 14, 15 | 61.49 | 60.72 ± 0.52 | 0.93 | 0.88 |
| 3        | 3              | 8, 9, 14, 15, 16 | 61.29 | 61.48 ± 0.29 | 0.91 | 0.83 |
| 4        | 4              | 8, 9, 14, 15, 16 | 61.70 | 61.82 ± 0.53 | 0.89 | 0.82 |

Table 6 | Network configurations with randomly chosen layers with $k_b$-bit precision

| Network index | Network configurations | Best accuracy (%) | Mean ± s.d. accuracy (%) | Energy |
|---------------|------------------------|-------------------|--------------------------|--------|
| N1            | Hybrid-Net (2,2) ($\Delta = 4$) | 64.34 | 63.75 ± 0.39 | 0.94 |
| N2            | Hybrid-Net (25, 26, 27, 28, 29, 30, 31) | 62.70 | 62.62 ± 0.09 | 0.90 |
| N3            | Hybrid-Net (2, 11, 12, 20, 21, 23, 29) | 63.43 | 63.00 ± 0.37 | 0.91 |
| N4            | Hybrid-Net (12, 17, 18, 20, 24, 25, 26) | 63.81 | 63.46 ± 0.21 | 0.91 |
| N5            | Hybrid-Net (2, 5, 6, 7, 20, 28) | 61.26 | 61.04 ± 0.24 | 0.92 |
| N6            | Hybrid-Net (2, 17, 22, 25, 28, 30) | 63.57 | 63.43 ± 0.12 | 0.93 |
that random initializations lead to varying optimal solutions, which would mean that the PCA step needs to be performed for every network initialization and dataset. For edge applications, the network can be tuned for a target application during the initial stages of deployment.

**Optimality studies.** The optimality of the proposed Hybrid-Net configurations can be understood through two visualizations. First, we compare the network designed through the proposed PCA-driven methodology with arbitrarily defined network architectures with randomly chosen layers as $k$-bit precision. We performed this analysis on ResNet-32 for the CIFAR-100 dataset. For identical comparisons to the proposed network Hybrid-Net(2,2) ($\Delta = 1$), we defined networks with six or seven randomly chosen layers with $k$-bit precision. Table 6 describes the networks. The numbers within parentheses indicate the layers with $k$-bit precision. The rest of the layers in this network are binary. Figure 4a shows the resulting plot. We observe that hybrid network configurations with similar energy efficiencies lead to different accuracies. This leads to the intuition that some layers might be more significant than others. This is the premise of our technique, which is a methodology to identify those significant layers. We note that the PCA-driven methodology used in designing Hybrid-Net achieves a better energy–accuracy tradeoff than networks with randomly chosen layers and $k$-bit precision.

Another visualization of optimality would be provided by a comparison of the networks with varying $\Delta$ and other baselines considered in this work. Here, we plot the best accuracies obtained for different configurations of ResNet-20 on CIFAR-100 and ResNet-18 on ImageNet, as described in Tables 3 and 4 (and plots in Fig. 4b).

Note that it is difficult to comment on the Pareto frontier without doing an exhaustive search of networks, which would be quite time expensive. In this work, we have focused on improving the accuracy of extremely quantized neural networks without a significant degradation in energy efficiency. Our technique proposes a methodology to design hybrid networks through a PCA-driven significance analysis, which achieves 10% higher accuracy with a less than 6–10% increase in energy consumption. Of the network configurations considered, the Hybrid-Net (2,2) configurations show that the proposed design principle can lead us to optimality for both CIFAR-100 and ImageNet. However, the absolute Pareto optimality is difficult to gauge without performing an exhaustive search.

**Discussion.** The proposed Hybrid-Net design uses PCA-driven hybridization of extremely quantized neural networks, resulting in significant improvements and observations. One key contribution of the proposed methodology is that we can design hybrid networks without any iterations. It does not require an iterative design space exploration to identify optimal networks. Moreover, this methodology shows that increasing the bit precision of only the significant layers in a binary network achieves performance close to that of a network that is entirely composed of layers with higher bit precision weights and activations. Intuitively, a 2-bit network performs much better than a binary network. However, our analysis shows that it is not necessary to make the weights and activations of the entire

---

**Table 7 | Analysis of quantization of the first and last layers in Hybrid-Net (2,2) ($\Delta = 4$) on ResNet-32 for CIFAR-100**

| Last layer configuration | Best accuracy (%) | Mean ± s.d. accuracy (%) | Energy consumption | Memory |
|--------------------------|------------------|--------------------------|--------------------|--------|
| Full-precision           | 64.34            | 63.75 ± 0.39             | 1                  | 1      |
| Binary weights and activations | 56.93 | 56.83 ± 0.09 | 0.98 | 0.75 |
| Binary weights and full-precision activations | 61.07 | 60.36 ± 0.44 | 0.98 | 0.75 |
| 2-bit weights and activations | 62.41 | 62.35 ± 0.05 | 0.98 | 0.76 |

| First layer configuration | Best accuracy (%) | Mean ± s.d. accuracy (%) | Energy consumption | Memory |
|---------------------------|------------------|--------------------------|--------------------|--------|
| Binary weights and activations | 44.79 | 44.16 ± 0.74 | 0.85 | 0.98 |
| Binary weights and full-precision activations | 59.94 | 59.29 ± 0.57 | 0.93 | 0.98 |
| 2-bit weights and activations | 60.87 | 60.46 ± 0.25 | 0.85 | 0.98 |

---

**Fig. 4 | Illustration of the energy–accuracy optimality of Hybrid-Net.**

a. Accuracy versus energy efficiency plot showing that the PCA-driven hybrid network design achieves more optimal tradeoffs than randomly chosen layers. b. Normalized energy efficiency versus accuracy plot showing the optimality of considered network configurations. This shows that Hybrid-Net (2,2) networks lie to the right of the line joining the other networks considered. Note, that PACT involves a more advanced quantization algorithm than the other networks.
network 2-bit. Hybrid-Net achieves more than ~10% improvement over an XNOR network (which is a fully binary network except for the first and final layers), by increasing the bit precision of less than half of the entire network. In fact, for deeper networks like ResNet-34, this improvement is achieved with only an ~6% increase in energy consumption over XNOR-Net. Hybrid-Net thus goes a long way towards reaching close to high-precision accuracies with networks that are mostly binary as well as attaining energy efficiency and memory compression values comparable to binary networks such as XNOR-Net and BNN. Moreover, this methodology can be extended to any network where making significant layers of the network $k_{10}$-bit while keeping the rest of the network $k_{10}$-bit ($k_{10} > k_0$) can potentially produce comparable performance (with enhanced energy efficiency) to an entirely $k_{10}$-bit network.

Second, the performance of Hybrid-Net is subject to the nature of the plots obtained from PCA on the binary version of the networks. For example, for ResNet architectures (Fig. 3a–c for CIFAR-100 and Fig. 3d for ImageNet), we observe that the number of significant components increases for layers that are adjacent to the ones where the number of output channels increases, and then decreases for the later layers, which have the same number of output channels. It can be said that the later layers are not adding to the linear separability of the data and binarizing them preserves the accuracy, as observed. In other words, the significant layers identified using our proposed methodology contribute remarkably more to the linear separability than the other layers. This is reflected in the results, where we show that the performance difference between Hybrid-Net and a 2-bit network is less than 4%. However, for VGG architectures, we observe that the PCA plot remains fairly flat, which means that the identified significant layers are not remarkably different in their contribution to the linear separability of the data in comparison to the other layers. This is reflected in the performance difference (>6%) of Hybrid-Net from a 2-bit network for a VGG-15 network.

Third, we observe that increasing the bit precision of the weights and activations of the significant layers to 4 bits while keeping the rest of the layers binary is not the most energy-efficient way of improving the accuracy of a network. An entire 2-bit network proves to be more energy-efficient while performing better. This may be because the loss due to binarization cannot be significantly recovered by increasing the bit precision much higher than binary, while keeping most of the layers binary. Thus, the proposed methodology performs best when the precision of the significant layers is close to the base precision of the network (binary in our case).

Finally, the precision of the first and final layers is of utmost importance for extremely quantized neural networks. To study the influence of quantization on these layers, we performed experiments on our Hybrid-Net designs by making the weights of the final layer binary while having full-precision inputs, as suggested in ref. [7]. The results for ResNet-32 for CIFAR-100 for the configuration Hybrid-Net (2,2) (4 = 4) are listed in Table 7.

The energy consumption reduction after quantizing the last layer of the neural network is only 2% for CIFAR-100. On the other hand, quantizing the last layer leads to at least 2% degradation in accuracy (when the last layer is 2-bit precision). The first layer consumes more energy due to the larger output map size. We performed this analysis by quantizing the first layer as well. Although the energy consumption reduces by 15%, the accuracy also degrades by 3.5%.

The memory requirement of the last layer is a significant aspect. We observe that quantizing the last layer can lead to close to 25% lower memory. In the case of ImageNet, this will be even more significant with up to two times memory reduction. However, fully binarizing the last layer leads to significant accuracy degradation. Thus, keeping higher precision (2-bit or 4-bit) will lead to better accuracy–memory tradeoffs.

In this work, we have considered the quantization scheme, explored in ref. [7]. A plethora of works have focused on improving quantization for both inputs and weights[10–15]. Hybrid-Net focuses on improving the performance of binary neural networks through mixed-precision network design and we believe the improved quantization schemes should further increase the accuracy of both Hybrid-Nets and entirely 2- or 4-bit networks.

The enormous computing power and memory requirements of deep networks stand in the way of ubiquitous use of AI for performing on-chip analytics in low-power edge devices. The significant energy efficiency offered by compressed hybrid networks increases the viability of using AI, powered by deep neural networks, in edge devices. With the proliferation of connected devices in the IoT environment, AI-enabled edge computing can reduce the communication overhead of cloud computing and augment the functionalities of devices beyond primitive tasks such as sensing, transmission and reception in situ processing.

**Conclusion**

Binary neural networks offer significant energy efficiency and memory compression compared to full-precision networks. In this work, we propose a one-shot methodology for designing mixed-precision hybrid networks with binary and higher bit precision inputs and weights to improve the performance of extremely quantized neural networks in terms of classification accuracy, while still achieving significant energy efficiency and memory compression. The proposed methodology uses PCA to identify significant layers in a binary network that transform the input data such that the output feature space requires more significant dimensions to explain variance in data. PCA is usually exploited to perform layer-wise dimensional reduction. We use PCA in an opposite manner to determine which layers cause the number of significant dimensions to increase across input and output. Next, we increase the bit precision of the weights and activations of the significant layers and keep those of the other layers binary. The proposed Hybrid-Net achieves more than ~10% improvement over XNOR networks for ResNet and VGG network architectures on CIFAR-100 and ImageNet, with only ~6–10% increase in energy consumption, thus ensuring more than 15–20 times reduction in energy consumption and memory.
compression from full-precision networks. Memory compression, along with the close match to high-precision accuracies offered by the proposed mixed-precision network design using layer-wise information, allows us to explore interesting possibilities in the realm of hardware–software co-design. This work thus proposes an effective, one-shot methodology for designing hybrid, compressed neural networks and potentially paves the way to using energy-efficient hybrid networks for AI-based on-chip analytics in low-power edge devices with accuracy close to full-precision networks.

Methods

Energy efficiency and memory calculations. Energy efficiency. The primary model-dependent metrics that affect the energy consumption of the classification task are the energies consumed by the computations (MAC operations) and memory access in our calculations for energy efficiency. We exclude energy consumed due to data flow and instruction flow in the architecture. For a convolutional layer, there are \( I \) input channels and \( O \) output channels. Let the size of the input be \( N \times N \), size of the kernel be \( k \times k \) and size of the output be \( M \times M \). In Table 8 we present the number of memory accesses \( N_{\text{mem}} \) and computations \( N_{\text{comp}} \) for standard full-precision (FP) networks.

The number of binary memory accesses \( (N_{\text{mem}}) \) and computations \( (N_{\text{comp}}) \) in a binary layer is the same as the corresponding number in a full-precision layer of equivalent dimensions. As explained in equation (1), we consider additional full-precision memory accesses and computations for parameter \( a \), where \( a \) is the scaling factor for each filter bank in a convolutional layer. The number of accesses for \( a \) is equal to the number of output maps, \( O \). The number of full-precision computations is \( M^2 \times O \). Table 8 lists the number of \( k \)-bit and full-precision memory accesses and computations of any layer. We calculated the energy consumption from projections for 45 nm CMOS technology\(^{1,11}\). Considering the \( 32 \)-bit representation as full-precision, the energy consumptions for both binary and \( 32 \)-bit memory accesses and computations are shown in Table 8.

The energy consumed by any layer with \( k \)-bit weights and activations is given by

\[
E_k = N_{\text{mem}} \cdot k + N_{\text{comp}} \cdot kE \quad \text{(9)}
\]

Note that this calculation is a rather conservative estimate and does not take into account other hardware architectural aspects such as input-sharing or weight-sharing. However, our approach is concerned with modifications of network architecture and we compare the ratios of energy consumption. These aspects of the hardware architecture affect all the networks equally and hence can be taken out of consideration. Furthermore, FP MAC operations can be optimized for lower energy consumption. In our calculations, we have bluntly taken it as the sum of \( 32 \)-bit FP multiply and \( 32 \)-bit FP add operations. These optimizations are catered towards FP networks and reduce the FP energy consumption. This, in turn, will reduce the energy efficiency of the binary and hybrid networks. In this work, we are focused on comparing different kinds of binary and hybrid network, so this assumption of FP MAC energy will not affect the analysis.

Memory compression. The memory requirement for any network is given as the product of the total number of weights in the network multiplied by the precision of the weights. The number of weights in any layer is given by

\[
N_{\text{w,comp}} = I \times O \times k^2 \quad \text{(10)}
\]

considering the usual notations described already. Thus, the total memory requirements can be simply written as \( M = \sum N_{\text{w,comp}} + k_{\text{mem}} \) where \( k_{\text{mem}} \) is the precision of weights in the \( i \)-th layer. We can estimate \( M \) with respect to a full-precision network and normalize it with respect to an XNOR-Net network, which is an entirely binary network except for the first and final layers.

Note that the assumption for the energy and storage calculations for binary layers holds for custom hardware capable of handling fixed-point binary representations of data, thus leveraging the benefits offered by quantized networks.

Reporting Summary. Further information on research design is available in the Nature Research Reporting Summary linked to this article.

Data availability

All datasets used in this work are publicly available: CIFAR-100\(^{10}\) and ImageNet\(^{11}\).

Code availability

The publicly available tools Python and PyTorch were used to perform the experiments. Custom codes for the work are available at https://github.com/ichakrav2/pca-hybrid.

Received: 24 May 2019; Accepted: 5 December 2019; Published online: 17 January 2020

References

1. Gubbi, J., Buyya, R., Marusic, S. & Palaniswami, M. Internet of things (IoT): a vision, architectural elements and future directions. Future Gener. Comput. Syst. 29, 1645–1660 (2013).
2. Yao, S., Hu, S., Zhao, Y., Zeng, A. & Abdelzaher, T. DeepSense. In Proceedings of the 26th International Conference on World Wide Web - WWW 17 351–360 (ACM Press, 2017).
3. Krizhevsky, A. et al. ImageNet classification with deep convolutional neural networks. In Proceedings of Advances in Neural Information Processing Systems 25 1097–1105 (NIPS, 2012).
4. Szegedy, C. et al. Going deeper with convolutions. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 1–9 (IEEE, 2015).
5. He, K. & Zhang, X. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition 770–778 (IEEE, 2016).
6. Girshick, R. B. Fast R-CNN. In Proceedings of the IEEE International Conference on Computer Vision 1440–1448 (IEEE, 2015).
7. Kaufman, L. M. Data security in the world of cloud computing. IEEE Secur. Priv. 7, 61–64 (2009).
8. Gonzalez, N. et al. A quantitative analysis of current security concerns and solutions for cloud computing. J. Cloud Comput. Adv. Syst. Appl. 1, 1 (2012).
9. Li, D., Salondis, T., Desai, N. V. & Chua, M. C. DeepCham: collaborative edge-mediated adaptive deep learning for mobile object recognition. In Proceedings of 2016 IEEE/ACM Symposium on Edge Computing (SEC) 64–76 (IEEE, 2016).
10. Rendola, F. N., Han, S., Moskewicz, M. W. & Ashraf, K. SqueezeNet: AlexNet-level accuracy with 50x fewer parameters and <0.5MB model size. Preprint at https://arxiv.org/pdf/1602.07360.pdf (2016).
11. Alvarez, J. M. & Salzmann, M. Compression-aware training of deep networks. In Proceedings of Advances in Neural Information Processing Systems 856–867 (NIPS, 2017).
12. Weigend, A. S., Rumelhart, D. E. & Huberman, B. A. Generalization by weight-elimination with application to forecasting. In Proceedings of Advances in Neural Information Processing Systems 875–882 (NIPS, 1991).
13. Han, S., Pool, J., Tran, J. & Dally, W. Learning both weights and connections for efficient neural network. In Proceedings of Advances in Neural Information Processing Systems 1135–1143 (NIPS, 2015).
14. Ulrich, K., Meeds, E. & Welling, M. Soft weight-sharing for neural network compression. In International Conference on Learning Representations (ICLR, 2017).
15. Hubara, I., Courbariaux, M., Soudry, D., El-Yaniv, R. & Bengio, Y. Quantized neural networks: training neural networks with low precision weights and activations. J. Mach. Learn. Res. 18, 6869–6898 (2017).
16. Courbariaux, M., Hubara, I., Soudry, D., El-Yaniv, R. & Bengio, Y. Binary neural networks: training deep neural networks with weights and activations constrained to +1 or –1. Preprint at https://arxiv.org/abs/1602.02830 (2016).
17. Raguram, M., Ordonez, V., Redmon, J. & Farhadi, A. XNOR-Net: ImageNet classification using binary convolutional neural networks. In Proceedings of the European Conference on Computer Vision 525–542 (Springer, 2016).
18. Garg, I., Panda, P. & Roy, K. A low effort approach to structured CNN design using PCA. Preprint at https://arxiv.org/abs/1812.06224 (2018).
19. Mishra, A., Narayntadi, E., Cook, I. J. & Marr, D. WRPNc: wide reduced-precision networks. Preprint at https://arxiv.org/abs/1709.01134 (2017).
20. Liu, Z. et al. Bi-RealNet: enhancing the performance of 1-bit CNNs with improved representational capability and advanced training algorithm. In Proceedings of the European Conference on Computer Vision (ECCV) 722–737 (Springer, 2018).
21. Zhou, S. et al. DoLeFa-Net: training low bitwidth convolutional neural networks with low bitwidth gradients. Preprint at https://arxiv.org/abs/1606.06160 (2016).
22. Zhou, S.-C., Wang, Y.-Z., Wen, H., He, Q.-Y. & Zou, Y.-H. Balanced quantization: an effective and efficient approach to quantized neural networks. J. Comput. Sci. Technol. 32, 667–682 (2017).
23. Zhang, D., Yang, J., Ye, D. & Hua, G. LQ-Nets: learned quantization for highly accurate and compact deep neural networks. In Proceedings of the European Conference on Computer Vision (ECCV) 365–382 (Springer, 2018).
24. Jung, S. et al. Learning to quantize deep networks by optimizing quantization intervals with task loss. Preprint at https://arxiv.org/abs/1808.05779 (2018).
25. Choi, J. et al. PACT: parameterized clipping activation for quantized neural networks. Preprint at https://arxiv.org/pdf/1805.06085.pdf (2018).
26. Graham, B. Low-precision batch-normalized activations. Preprint at https://arxiv.org/abs/1702.08231 (2017).
27. Prabhu, A. et al. Hybrid binary networks: optimizing for accuracy, efficiency and memory. In Proceedings of 2018 IEEE Winter Conference on Applications of Computer Vision (WACV) 821–829 (IEEE, 2018).
28. Wu, B. et al. Mixed precision quantization of ConvNets via differentiable neural architecture search. Preprint at https://arxiv.org/pdf/1812.00090.pdf (2018).
Author contributions
I.C. and K.R. conceived the idea. I.C., D.R. and I.G. developed the simulation framework. I.C. carried out all experiments. I.C. and A.A. developed the energy and memory analysis framework. I.C., D.R., I.G. and K.R. analysed the results. I.C., D.R. and I.G. wrote the paper.

Competing interests
The authors declare no competing interests.

Additional information
Supplementary information is available for this paper at https://doi.org/10.1038/s42256-019-0134-0.

Acknowledgements
This work was supported in part by the Center for Brain-inspired Computing Enabling Autonomous Intelligence (C-BRIC), one of six centres in JUMP, a Semiconductor Research Corporation (SRC) programme sponsored by DARPA, in part by the National Science Foundation, in part by Intel, in part by the ONR-MURI programme and in part by the Vannevar Bush Faculty Fellowship.

NATURE MACHINE INTELLIGENCE

© The Author(s), under exclusive licence to Springer Nature Limited 2020
Reporting Summary

Nature Research wishes to improve the reproducibility of the work that we publish. This form provides structure for consistency and transparency in reporting. For further information on Nature Research policies, see Authors & Referees and the Editorial Policy Checklist.

Statistics

For all statistical analyses, confirm that the following items are present in the figure legend, table legend, main text, or Methods section.

- n/a
- The exact sample size (n) for each experimental group/condition, given as a discrete number and unit of measurement
- A statement on whether measurements were taken from distinct samples or whether the same sample was measured repeatedly
- The statistical test(s) used AND whether they are one- or two-sided
- Only common tests should be described solely by name; describe more complex techniques in the Methods section.
- A description of all covariates tested
- A description of any assumptions or corrections, such as tests of normality and adjustment for multiple comparisons
- A full description of the statistical parameters including central tendency (e.g. means) or other basic estimates (e.g. regression coefficient) AND variation (e.g. standard deviation) or associated estimates of uncertainty (e.g. confidence intervals)
- For null hypothesis testing, the test statistic (e.g. F, t, r) with confidence intervals, effect sizes, degrees of freedom and P value noted give P values as exact values whenever suitable.
- For Bayesian analysis, information on the choice of priors and Markov chain Monte Carlo settings
- For hierarchical and complex designs, identification of the appropriate level for tests and full reporting of outcomes
- Estimates of effect sizes (e.g. Cohen’s d, Pearson’s r), indicating how they were calculated

Our web collection on statistics for biologists contains articles on many of the points above.

Software and code

Policy information about availability of computer code

Data collection: N/A

Data analysis: Publicly available tools, Python and PyTorch, were used to perform the experiments. The custom codes for the work are available at: https://github.com/ichakra2/pca-hybrid

For manuscripts utilizing custom algorithms or software that are central to the research but not yet described in published literature, software must be made available to editors/reviewers. We strongly encourage code deposition in a community repository (e.g. GitHub). See the Nature Research guidelines for submitting code & software for further information.

Data

Policy information about availability of data

All manuscripts must include a data availability statement. This statement should provide the following information, where applicable:
- Accession codes, unique identifiers, or web links for publicly available datasets
- A list of figures that have associated raw data
- A description of any restrictions on data availability

All datasets used in the work are publicly available: CIFAR-100 (https://www.cs.toronto.edu/~kriz/cifar.html) and ImageNet (http://image-net.org/challenges/LSVRC/)
Field-specific reporting

Please select the one below that is the best fit for your research. If you are not sure, read the appropriate sections before making your selection.

- Life sciences
- Behavioural & social sciences
- Ecological, evolutionary & environmental sciences

For a reference copy of the document with all sections, see nature.com/documents/nr-reporting-summary-flat.pdf

Life sciences study design

All studies must disclose on these points even when the disclosure is negative.

- **Sample size**: Describe how sample size was determined, detailing any statistical methods used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.

- **Data exclusions**: Describe any data exclusions. If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.

- **Replication**: Describe the measures taken to verify the reproducibility of the experimental findings. If all attempts at replication were successful, confirm this OR if there are any findings that were not replicated or cannot be reproduced, note this and describe why.

- **Randomization**: Describe how samples/organisms/participants were allocated into experimental groups. If allocation was not random, describe how covariates were controlled OR if this is not relevant to your study, explain why.

- **Blinding**: Describe whether the investigators were blinded to group allocation during data collection and/or analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.

Behavioural & social sciences study design

All studies must disclose on these points even when the disclosure is negative.

- **Study description**: Briefly describe the study type including whether data are quantitative, qualitative, or mixed-methods (e.g. qualitative cross-sectional, quantitative experimental, mixed-methods case study).

- **Research sample**: State the research sample (e.g. Harvard university undergraduates, villagers in rural India) and provide relevant demographic information (e.g. age, sex) and indicate whether the sample is representative. Provide a rationale for the study sample chosen. For studies involving existing datasets, please describe the dataset and source.

- **Sampling strategy**: Describe the sampling procedure (e.g. random, snowball, stratified, convenience). Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient. For qualitative data, please indicate whether data saturation was considered, and what criteria were used to decide that no further sampling was needed.

- **Data collection**: Provide details about the data collection procedure, including the instruments or devices used to record the data (e.g. pen and paper, computer, eye tracker, video or audio equipment) whether anyone was present besides the participant(s) and the researcher, and whether the researcher was blind to experimental condition and/or the study hypothesis during data collection.

- **Timing**: Indicate the start and stop dates of data collection. If there is a gap between collection periods, state the dates for each sample cohort.

- **Data exclusions**: If no data were excluded from the analyses, state so OR if data were excluded, provide the exact number of exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.

- **Non-participation**: State how many participants dropped out/declined participation and the reason(s) given OR provide response rate OR state that no participants dropped out/declined participation.

- **Randomization**: If participants were not allocated into experimental groups, state so OR describe how participants were allocated to groups, and if allocation was not random, describe how covariates were controlled.

Ecological, evolutionary & environmental sciences study design

All studies must disclose on these points even when the disclosure is negative.

- **Study description**: Briefly describe the study. For quantitative data include treatment factors and interactions, design structure (e.g. factorial, nested, hierarchical), nature and number of experimental units and replicates.

- **Research sample**: Describe the research sample (e.g. a group of tagged Passer domesticus, all Stenocereus thurberi within Organ Pipe Cactus National
**Research sample**

Monument, and provide a rationale for the sample choice. When relevant, describe the organism taxa, source, sex, age range and any manipulations. State what population the sample is meant to represent when applicable. For studies involving existing datasets, describe the data and its source.

**Sampling strategy**

Note the sampling procedure. Describe the statistical methods that were used to predetermine sample size OR if no sample-size calculation was performed, describe how sample sizes were chosen and provide a rationale for why these sample sizes are sufficient.

**Data collection**

Describe the data collection procedure, including who recorded the data and how.

**Timing and spatial scale**

Indicate the start and stop dates of data collection, noting the frequency and periodicity of sampling and providing a rationale for these choices. If there is a gap between collection periods, state the dates for each sample cohort. Specify the spatial scale from which the data are taken.

**Data exclusions**

If no data were excluded from the analyses, state so OR if data were excluded, describe the exclusions and the rationale behind them, indicating whether exclusion criteria were pre-established.

**Reproducibility**

Describe the measures taken to verify the reproducibility of experimental findings. For each experiment, note whether any attempts to repeat the experiment failed OR state that all attempts to repeat the experiment were successful.

**Randomization**

Describe how samples/organisms/participants were allocated into groups. If allocation was not random, describe how covariates were controlled. If this is not relevant to your study, explain why.

**Blinding**

Describe the extent of blinding used during data acquisition and analysis. If blinding was not possible, describe why OR explain why blinding was not relevant to your study.

**Field work, collection and transport**

| Field conditions | Describe the study conditions for field work, providing relevant parameters (e.g. temperature, rainfall). |
|------------------|--------------------------------------------------------------------------------------------------|
| Location         | State the location of the sampling or experiment, providing relevant parameters (e.g. latitude and longitude, elevation, water depth). |
| Access and import/export | Describe the efforts you have made to access habitats and to collect and import/export your samples in a responsible manner and in compliance with local, national and international laws, noting any permits that were obtained (give the name of the issuing authority, the date of issue, and any identifying information). |
| Disturbance      | Describe any disturbance caused by the study and how it was minimized. |

**Reporting for specific materials, systems and methods**

We require information from authors about some types of materials, experimental systems and methods used in many studies. Here, indicate whether each material, system or method listed is relevant to your study. If you are not sure if a list item applies to your research, read the appropriate section before selecting a response.

### Materials & experimental systems

| n/a | Involved in the study |
|-----|-----------------------|
| [ ] | Antibodies            |
| [ ] | Eukaryotic cell lines |
| [ ] | Palaeontology         |
| [ ] | Animals and other organisms |
| [ ] | Human research participants |
| [ ] | Clinical data         |

### Methods

| n/a | Involved in the study |
|-----|-----------------------|
| [ ] | ChIP-seq              |
| [ ] | Flow cytometry        |
| [ ] | MRI-based neuroimaging |

**Antibodies**

| Antibodies used | Describe all antibodies used in the study; as applicable, provide supplier name, catalog number, clone name, and lot number. |
|-------|------------------------------------------------------------------|
| Validation | Describe the validation of each primary antibody for the species and application, noting any validation statements on the manufacturer’s website, relevant citations, antibody profiles in online databases, or data provided in the manuscript. |
Eukaryotic cell lines

Policy information about cell lines

Cell line source(s)
State the source of each cell line used.

Authentication
Describe the authentication procedures for each cell line used OR declare that none of the cell lines used were authenticated.

Mycoplasma contamination
Confirm that all cell lines tested negative for mycoplasma contamination OR describe the results of the testing for mycoplasma contamination OR declare that the cell lines were not tested for mycoplasma contamination.

Commonly misidentified lines
(See ICLAC register)
None any commonly misidentified cell lines used in the study and provide a rationale for their use.

Palaeontology

Specimen provenance
Provide provenance information for specimens and describe permits that were obtained for the work (including the name of the issuing authority, the date of issue, and any identifying information).

Specimen deposition
Indicate where the specimens have been deposited to permit free access by other researchers.

Dating methods
If new dates are provided, describe how they were obtained (e.g. collection, storage, sample pretreatment and measurement), where they were obtained (i.e. lab name), the calibration program and the protocol for quality assurance OR state that no new dates are provided.

☐ Tick this box to confirm that the raw and calibrated dates are available in the paper or in Supplementary Information.

Animals and other organisms

Policy information about studies involving animals, ARRIVE guidelines, recommended for reporting animal research

Laboratory animals
For laboratory animals, report species, strain, sex and age OR state that the study did not involve laboratory animals.

Wild animals
Provide details on animals observed in or captured in the field; report species, sex and age where possible. Describe how animals were caught and transported and what happened to captive animals after the study (if killed, explain why and describe method; if released, say where and when) OR state that the study did not involve wild animals.

Field-collected samples
For laboratory work with field-collected samples, describe all relevant parameters such as housing, maintenance, temperature, photoperiod and end-of-experiment protocol OR state that the study did not involve samples collected from the field.

Ethics oversight
Identify the organization(s) that approved or provided guidance on the study protocol, OR state that no ethical approval or guidance was required and explain why not.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Human research participants

Policy information about studies involving human research participants

Population characteristics
Describe the covariate-relevant population characteristics of the human research participants (e.g. age, gender, genotypic information, past and current diagnosis and treatment categories). If you filled out the behavioural & social sciences study design questions and have nothing to add here, write “See above.”

Recruitment
Describe how participants were recruited. Outline any potential self-selection bias or other biases that may be present and how these are likely to impact results.

Ethics oversight
Identify the organization(s) that approved the study protocol.

Note that full information on the approval of the study protocol must also be provided in the manuscript.

Clinical data

Policy information about clinical studies

All manuscripts should comply with the ICMJE guidelines for publication of clinical research and a completed CONSORT checklist must be included with all submissions.

Clinical trial registration
Provide the trial registration number from ClinicalTrials.gov or an equivalent agency.

Study protocol
Note where the full trial protocol can be accessed OR if not available, explain why.

Data collection
Describe the settings and locales of data collection, noting the time periods of recruitment and data collection.
Outcomes

Describe how you pre-defined primary and secondary outcome measures and how you assessed these measures.

ChIP-seq

Data deposition

- Confirm that both raw and final processed data have been deposited in a public database such as GEO.
- Confirm that you have deposited or provided access to graph files (e.g. BED files) for the called peaks.

Data access links

For "Initial submission" or "Revised version" documents, provide reviewer access links. For your "Final submission" document, provide a link to the deposited data.

Files in database submission

Provide a list of all files available in the database submission.

Genome browser session

Provide a link to an anonymized genome browser session for "Initial submission" and "Revised version" documents only, to enable peer review. Write "no longer applicable" for "Final submission" documents.

Methodology

Replicates

Describe the experimental replicates, specifying number, type and replicate agreement.

Sequencing depth

Describe the sequencing depth for each experiment, providing the total number of reads, uniquely mapped reads, length of reads and whether they were paired- or single-end.

Antibodies

Describe the antibodies used for the ChIP-seq experiments; as applicable, provide supplier name, catalog number, clone name, and lot number.

Peak calling parameters

Specify the command line program and parameters used for read mapping and peak calling, including the ChIP, control and index files used.

Data quality

Describe the methods used to ensure data quality in full detail, including how many peaks are at FDR 5% and above 5-fold enrichment.

Software

Describe the software used to collect and analyze the ChIP-seq data. For custom code that has been deposited into a community repository, provide accession details.

Flow Cytometry

Plots

- The axis labels state the marker and fluorochrome used (e.g. CD4-FITC).
- The axis scales are clearly visible. Include numbers along axes only for bottom left plot of group (a 'group' is an analysis of identical markers).
- All plots are contour plots with outliers or pseudocolor plots.
- A numerical value for number of cells or percentage (with statistics) is provided.

Methodology

Sample preparation

Describe the sample preparation, detailing the biological source of the cells and any tissue processing steps used.

Instrument

Identify the instrument used for data collection, specifying make and model number.

Software

Describe the software used to collect and analyze the flow cytometry data. For custom code that has been deposited into a community repository, provide accession details.

Cell population abundance

Describe the abundance of the relevant cell populations within post-sort fractions, providing details on the purity of the samples and how it was determined.

Gating strategy

Describe the gating strategy used for all relevant experiments, specifying the preliminary FSC/SSC gates of the starting cell population, indicating where boundaries between "positive" and "negative" staining cell populations are defined.

- Tick this box to confirm that a figure exemplifying the gating strategy is provided in the Supplementary Information.
**Magnetic resonance imaging**

### Experimental design

| Design type | Indicate task or resting state; event-related or block design. |
|-------------|----------------------------------------------------------------|
| Design specifications | Specify the number of blocks, trials or experimental units per session and/or subject, and specify the length of each trial or block (if trials are blocked) and interval between trials. |
| Behavioral performance measures | State number and/or type of variables recorded (e.g. correct button press, response time) and what statistics were used to establish that the subjects were performing the task as expected (e.g. mean, range, and/or standard deviation across subjects). |

### Acquisition

| Imaging type(s) | Specify: functional, structural, diffusion, perfusion. |
|-----------------|--------------------------------------------------------|
| Field strength | Specify in Tesla |
| Sequence & imaging parameters | Specify the pulse sequence type (gradient echo, spin echo, etc.), imaging type (EPI, spiral, etc.), field of view, matrix size, slice thickness, orientation and TE/TR/flip angle. |
| Area of acquisition | State whether a whole brain scan was used OR define the area of acquisition, describing how the region was determined. |
| Diffusion MRI | □ Used □ Not used |

### Preprocessing

| Preprocessing software | Provide detail on software version and revision number and on specific parameters (model/functions, brain extraction, segmentation, smoothing kernel size, etc.). |
|------------------------|--------------------------------------------------|
| Normalization | If data were normalized/standardized, describe the approach(es): specify linear or non-linear and define image types used for transformation OR indicate that data were not normalized and explain rationale for lack of normalization. |
| Normalization template | Describe the template used for normalization/transformation, specifying subject space or group standardized space (e.g. original Talairach, MNI305, ICBM152) OR indicate that the data were not normalized. |
| Noise and artifact removal | Describe your procedure(s) for artifact and structured noise removal, specifying motion parameters, tissue signals and physiological signals (heart rate, respiration). |
| Volume censoring | Define your software and/or method and criteria for volume censoring, and state the extent of such censoring. |

### Statistical modeling & inference

| Model type and settings | Specify type (mass univariate, multivariate, RSA, predictive, etc.) and describe essential details of the model at the first and second levels (e.g. fixed, random or mixed effects; drift or auto-correlation). |
|-------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| Effect(s) tested | Define precise effect in terms of the task or stimulus conditions instead of psychological concepts and indicate whether ANOVA or factorial designs were used. |
| Specify type of analysis: | □ Whole brain □ ROI-based □ Both |
| Statistic type for inference | (See Eklund et al. 2016) Specify voxel-wise or cluster-wise and report all relevant parameters for cluster-wise methods. |
| Correction | Describe the type of correction and how it is obtained for multiple comparisons (e.g. FWE, FDR, permutation or Monte Carlo). |

### Models & analysis

| n/a | Involved in the study |
|-----|-----------------------|
|     | □ Functional and/or effective connectivity |
|     | □ Graph analysis |
|     | □ Multivariate modeling or predictive analysis |

**Functional and/or effective connectivity**

Report the measures of dependence used and the model details (e.g. Pearson correlation, partial correlation, mutual information).

**Graph analysis**

Report the dependent variable and connectivity measure, specifying weighted graph or binarized graph, subject- or group-level, and the global and/or node summaries used (e.g. clustering coefficient, efficiency, etc.).
Multivariate modeling and predictive analysis

Specify independent variables, features extraction and dimension reduction, model, training and evaluation metrics.