International Journal of Computer Assisted Radiology and Surgery (2018) 13:1311–1320
https://doi.org/10.1007/s11548-018-1797-4

ORIGINAL ARTICLE

# TernaryNet: faster deep model inference without GPUs for medical 3D segmentation using sparse and binary convolutions

Mattias P. Heinrich[1] · Max Blendowski[1] · Ozan Oktay[2]

Received: 27 January 2018 / Accepted: 21 May 2018 / Published online: 30 May 2018

## Abstract

**Purpose** Deep convolutional neural networks (DCNN) are currently ubiquitous in medical imaging. While their versatility and high-quality results for common image analysis tasks including segmentation, localisation and prediction is astonishing, the large representational power comes at the cost of highly demanding computational effort. This limits their practical applications for image-guided interventions and diagnostic (point-of-care) support using mobile devices without graphics processing units (GPU).

**Methods** We propose a new scheme that approximates both trainable weights and neural activations in deep networks by ternary values and tackles the open question of backpropagation when dealing with non-differentiable functions. Our solution enables the removal of the expensive floating-point matrix multiplications throughout any convolutional neural network and replaces them by energy- and time-preserving binary operators and population counts.

**Results** We evaluate our approach for the segmentation of the pancreas in CT. Here, our ternary approximation within a fully convolutional network leads to more than 90% memory reductions and high accuracy (without any post-processing) with a Dice overlap of 71.0% that comes close to the one obtained when using networks with high-precision weights and activations. We further provide a concept for sub-second inference without GPUs and demonstrate significant improvements in comparison with binary quantisation and without our proposed ternary hyperbolic tangent continuation.

**Conclusions** We present a key enabling technique for highly efficient DCNN inference without GPUs that will help to bring the advances of deep learning to practical clinical applications. It has also great promise for improving accuracies in large-scale medical data retrieval.

**Keywords** Deep learning · Pancreas · Segmentation · Sparsity · Model compression · Hamming distance

## Introduction

Deep convolutional neural networks (CNNs) have been shown to substantially improve common image analysis tasks in computer vision and (bio-)medical imaging. They have in particular advanced research in automatic segmentation and image classification. Dense prediction based on fully convolutional network (FCN) architectures [20] enables very accurate voxel-wise segmentation by a single forward pass of the input image through a trained CNN architecture [6]. However, FCNs also come with tremendous demand for memory and computational resources that can rarely be satisfied in clinical scenarios—in particular when envisioning a mobile application of computer-assisted diagnosis and interventions. Furthermore, the translation of deep learning into interactive clinical workflows will require processing times of few seconds, which up-to-date were only achievable using power-demanding GPUs. Surprisingly little research has been undertaken in deep learning for medical image analysis that attempts to limit model complexity. In this work, we address these challenges and present a new technique to advance state-of-the-art CNN and FCN approaches by intro-

This work was in part supported by the German Research Foundation (DFG) under Grant Number 320997906.

✉ Mattias P. Heinrich
heinrich@imi.uni-luebeck.de
http://mpheinrich.de

Max Blendowski
blendowski@imi.uni-luebeck.de

Ozan Oktay
o.oktay13@imperial.ac.uk

[1] Institute of Medical Informatics, University of Lübeck, Ratzeburger Allee 160, 23562 Lübeck, Germany

[2] Biomedical Image Analysis Group, Department of Computing, Imperial College London, London SW7 2AZ, UK

ducing the TernaryNet—a versatile end-to-end trainable deep learning architecture that drastically reduces computational and memory demand for inference. We achieve this goal by replacing floating-point matrix multiplications with ternary convolutions (based on sparse binary kernels), with both activations and weights restricted to values of $\{-1, 0, +1\}$. They can be calculated using a masked Hamming distance, a XOR/XNOR operation followed by a `popcount`, and reduce computational demand by up to a factor of 16. Our approach is not merely motivated by gains in computational performance, but also to explore the theoretical advantages of explicit sparsity promotion to reduce the risk of overfitting (as detailed in the following subsection) and learn more plausible neural network models. Our work extends recent approaches from computer vision that relied on binary convolutions [25], ternary weight networks [18], hashing by continuation [2] and our initial work on sparse binary convolutions [10]. The presented approach is to the best of our knowledge the first to use binary convolutions for semantic segmentation and the very first to propose ternary convolutions (and not only ternary weights since activations are also restricted) based on masked Hamming distances.

The TernaryNet can be employed for any given image analysis task, e.g. landmark regression or image-level classification, but we chose to demonstrate its applicability to medical imaging for the automatic voxel-accurate segmentation of the pancreas in CT scans, which is a particularly demanding task. Pancreas segmentation is very important for computer-assisted diagnosis of inflammation (pancreatitis) or cancer and furthermore to provide image-based navigational guidance for interventions, including endoscopy [6]. In the following, we will motivate the use of sparse binary kernels in deep convolutional networks and discuss related work for the use of quantisation in image analysis in particular in deep networks. Section 2 contains the detailed explanation of ternary quantisation and convolutions. Starting with a short discussion of current work on CT pancreas segmentation, we describe our experimental set-up in Sect. 3 and compare different strategies and choices for model complexity reduction. We discuss our results, potentials for further research and future implications of our novel ternary convolution concept in Sect. 4 and end with some concluding remarks.

*Motivation for sparse binary kernels:* Convolutional neuronal networks excel in image recognition tasks by mimicking the visual cortex of mammals. The visual information is detected by photoreceptor cells and transmitted and processed using multiple layers of neurons interconnected by synapses. Computational models have the capacity to replicate these mechanisms and can furthermore represent neural activations up to extremely high numerical precision (up to 8 decimal points). However, in nature the simple structure of neural cells and environmental influences severely limit the accuracy of subtle changes in activation and in addition the need to conserve energy may lead to a sparse as possible use of neural activity. Ohlshausen and Field [24] and Lee et al. [17] therefore established the idea of sparse coding for pattern recognition and neural networks. Those works demonstrate that powerful convolutional filters can be learned using few nonzero values by means of sparsity inducing L1 norms and a feature sign searching algorithm. Furthermore, we observe that the nonzero elements of these synthetic models of V1 cells tend to be close to values +1 and −1. Therefore, a ternary approximation of weights leads to only minor degradation of representational power (see Fig. 1).

*Related work:* Due to their computational efficiency, binary codes and their comparison using the Hamming distance (which counts the number of dissimilar bits in a long binary vector) are becoming increasingly popular for demanding image analysis tasks. They have been employed for hashing-based large-scale image retrieval [3,35], nearest-neighbour-based segmentation [7] and image registration [8]. In computer vision, binary descriptors are frequently used for real-time applications, e.g. tracking using BRIEF features [1]. There are, however, also cases where binarisation led to inadequate loss in representation quality as, e.g. reported for lung nodule classification in [5].

In our recent prior work [10], we proposed the use of sparse binary kernels with very large receptive fields inspired by BRIEF features and dilated convolutions [30,34] that enabled highly accurate segmentations without complex network architectures. Similarly and concurrently, [14] proposed local binary convolutions that are derived from local binary patterns. A key limitation of these works is, however, that their design does not allow us to automatically train nonzero elements within binary kernels. Instead, they have to be chosen once at random (with a similar manual design as proposed in [1]). We also did not realise binary or ternary activations thus the use of efficient computations without floating-point arithmetic was not possible. An alternative solution that has recently been proposed is the use of trained ternary filter weights [18,37]. In particular ternary weight networks [18] use a very simple, yet powerful, approximation and learning strategy based on the mild assumption of Gaussian statistics. They generalise the earlier ideas of [4,25] for learning binary weights and clearly demonstrate that ternarisation drastically reduces the accuracy gap to high-precision weights. Another related approach by Liu et al. [19] employs decomposition methods for sparsification of convolution filters and proposes a new implementation for fast sparse matrix multiplication.

While weight quantisation has quickly matured, another important aspect that has so far been only insufficiently addressed is the quantisation or sparsification of activations. Setting approximately half of the activations to zero using a rectifying linear unit (ReLU) is common practice in deep

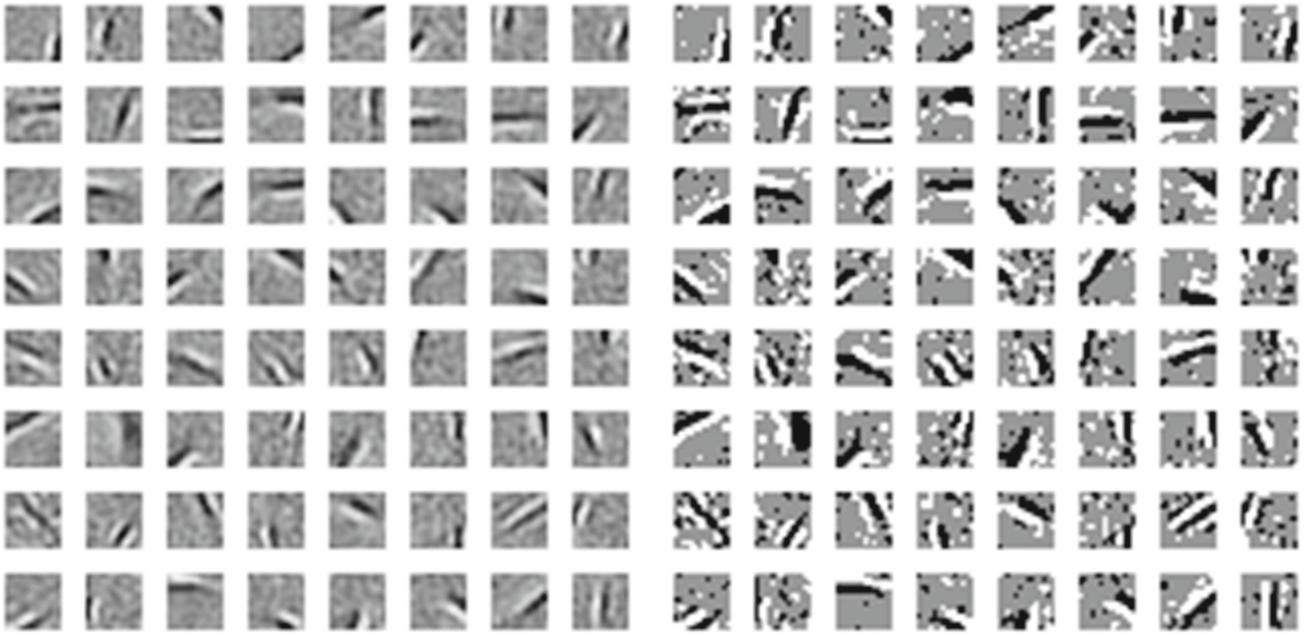

**Fig. 1** Left: Visual example of learned synthetic receptive fields (reproducing the results of [17]) using sparse coding techniques. Right: Ternarisation of weights demonstrates the low approximation error for these naturally inspired sparse filters

learning. Yet more drastic quantisation, e.g. using the sign function

$$\operatorname{sgn}(x) := (x \geq 0 \rightarrow 1) \wedge (x < 0 \rightarrow -1) \qquad (1)$$

as nonlinear activation leads to strong artefacts during forward passes and no gradient for backpropagation. Courbariaux et al. [4] therefore proposes an ad hoc solution that employs a rectangle (boxcar) function

$$\partial \operatorname{sgn} / \partial x \approx (|x| \leq 1 \rightarrow 1) \wedge (|x| > 1 \leq 0 \rightarrow 0) \qquad (2)$$

as a replacement, which was later also used in [25]. The downside of this approach is the fact that since two different functions are used during forward and backward propagation the training behaviour is ill-defined and potentially unstable. Cao et al. [2] propose a more justifiable approach based on the continuation of the hyperbolic tangent, which approaches the sign function with increasing slope $\beta$ in its limit:

$$\lim_{\beta \rightarrow \infty} \tanh(\beta x) = \operatorname{sgn}(x) \qquad (3)$$

They prove the convergence of this optimisation when employing a sequence of increasing values of $\beta$ during training. They limit the use of this function to the final layer within a framework for supervised hashing. In our work, we extend this concept to a ternary hyperbolic tangent as explained in detail in the following section and apply this function as nonlinearity throughout—for every activation—in our deep network models.

## Method

We aim to automatically segment the pancreas in regions of interest extracted from CT volumes. For this purpose, a fully convolutional U-Net architecture [26] is chosen. However, a V-Net [21] or multi-path network will most likely lead to similarly good segmentations and would also support our findings. The U-Net model can contain several million free parameters rendering it computationally demanding and prone to overfitting. Furthermore, as common for FCN architectures an efficient inference requires an unexpectedly large amount of memory due to the use of the `im2col` operations. They are necessary to perform multichannel convolutions of all elements in the feature maps in parallel using matrix multiplications between activations of preceding layers with a current filter bank [13]. We propose a ternary quantisation of weights and activations that is generic and therefore applicable to reduce complexity for any (convolutional) neural network architecture including FCNs.

*Ternary weights:* In order to limit the memory demand, reduce model complexity and enable inference of CNNs in practical clinical environments, it is desirable to reduce the precision of both activations and weights. Following the recent work of Li et al. [18], we aim to find the best approximation to the filter weights $\mathbf{W} \approx \alpha \tilde{\mathbf{W}}$ where $\alpha$ describes a (floating-point) scaling parameter and $\tilde{\mathbf{W}}$ consists of only ternary values $\{-1, 0, 1\}$. It is shown in [18] that the minimal quantisation error can be obtained by calculating:

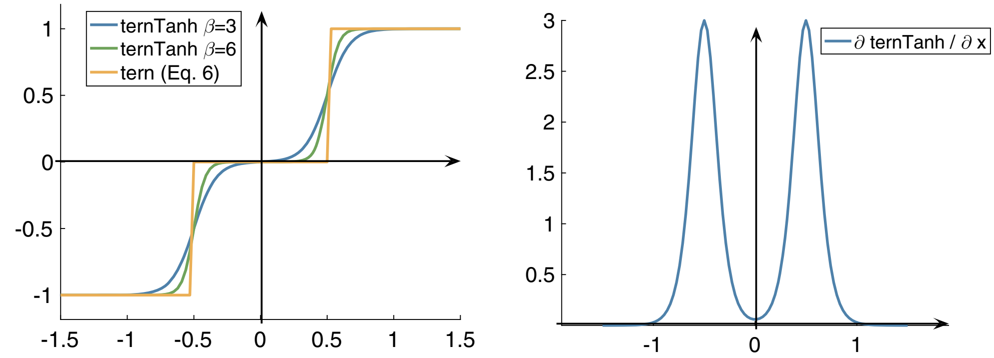

**Fig. 2** Visualisation of proposed ternary hyperbolic tangent as defined in Eq. 5 showing varying $\beta$ values for increasing steepness of slopes. The analytical derivative of our new nonlinearity is shown for $\beta = 3$ on the right

$$\tilde{\mathbf{W}}_i = \begin{cases} +1 & \text{if } W_i > \Delta \\ 0 & \text{if } |W_i| \le \Delta \\ -1 & \text{else} \end{cases} \quad \text{with } \Delta = \frac{0.7}{n}\sum_{i=1}^{n}|W_i| \tag{4}$$

and $\alpha = \frac{1}{n_\Delta}\sum_i |\tilde{W}_i||W_i|$ with $n_\Delta = \sum_i |\tilde{W}_i|$. When employing quantised weights during the training of a network using stochastic gradient descent with mini-batches (i.e. in virtually any case of deep learning), it is strongly advisable [4] to accumulate gradient updates with full precision (while using $\tilde{\mathbf{W}}$ for both forward and backward passes); otherwise, they would usually not exceed the threshold (according to Eq. 4) necessary to flip individual bits. This simple and straightforward ternary weight approximation already yields excellent accuracies for classification tasks (only 3.6% lower top-1 scores for ImageNet compared to full-precision networks [18]).

*Ternary activations:* The use of ternary weight approximations alone, however, cannot reduce the huge memory and computational demand required to store and process intermediate feature maps, since the resulting activations will still be full precision. The key contribution of our work is therefore the introduction of a new activation function that enables an accurate ternarisation of intermediate features in a neural network, which we coin ternary hyperbolic tangent. This proposed function ternTanh$(x)$ combines two hyperbolic tangents to form plateaus around zero and beyond +1 and -1:

$$\text{ternTanh}(x) = \frac{1}{2}\tanh(2\beta x - \beta) - \frac{1}{2}\tanh(-2\beta x - \beta) \tag{5}$$

In contrast to a sign function, the ternary hyberbolic tangent is fully differentiable and can therefore be used without custom changes to the learning procedure of deep networks. The parameter $\beta$ controls the slope and can be varied throughout the process of learning. In earlier iterations, it is beneficial to use smaller values for $\beta$ to enable sufficient gradient flow and avoid “dying” neurons. Eventually, we aim for a discrete step function tern$(x)$ that can be defined as:

$$\text{tern}(x) = \begin{cases} +1 & \text{if } x > 0.5 \\ 0 & \text{if } |x| \le 0.5 \\ -1 & \text{else} \end{cases} \tag{6}$$

Similar as above for the binary case covered in [2], the following continuation holds true (see Fig. 2 for visual example):

$$\lim_{\beta \to \infty} \text{ternTanh}(\beta x) = \text{tern}(x) \tag{7}$$

*Ternary convolutions and complexity analysis:* In combining both ternary weights and ternary activations, we can realise important avoidance of time-consuming floating-point multiplications, which were at the core of classical deep learning architectures. In [4,25], the idea of replacing full-precision inner products of an input tensor $\mathbf{I}$ and a filter bank $\mathbf{W}$ by Boolean operations and bit counting (population count) was explored for binary valued operands, i.e. $\mathbf{I}, \mathbf{W} \in \{-1, +1\}^c$, where $c$ denotes the size of a kernel (including both spatial extend and number of features). It is straightforward to show that a matrix multiplication and its inner products can be efficiently calculated in the Hamming space:

$$\mathbf{I}_i \mathbf{W}_j = c - 2\Xi\{\mathbf{I}_i \oplus \mathbf{W}_j\} \tag{8}$$

where $\oplus$ defines an exclusive OR (XOR) operator and $\Xi$ a bit-count over the $c$ bits in the rows of $\mathbf{I}$ and $\mathbf{W}$. Modern CPUs, FPGAs or embedded SoCs all contain instructions for efficiently calculating population counts of 64-bit strings in few cycles (using AVX extensions Intel CPUs achieve a throughput of 0.5 cycles [23]). This means that each bit-count replaces 64 floating-point multiplications and additions. Even when considering the highly optimised fused multiply addition (FMA) instructions on 256 bit wide registers (`mm256-fmadd-ps`), which are employed on modern

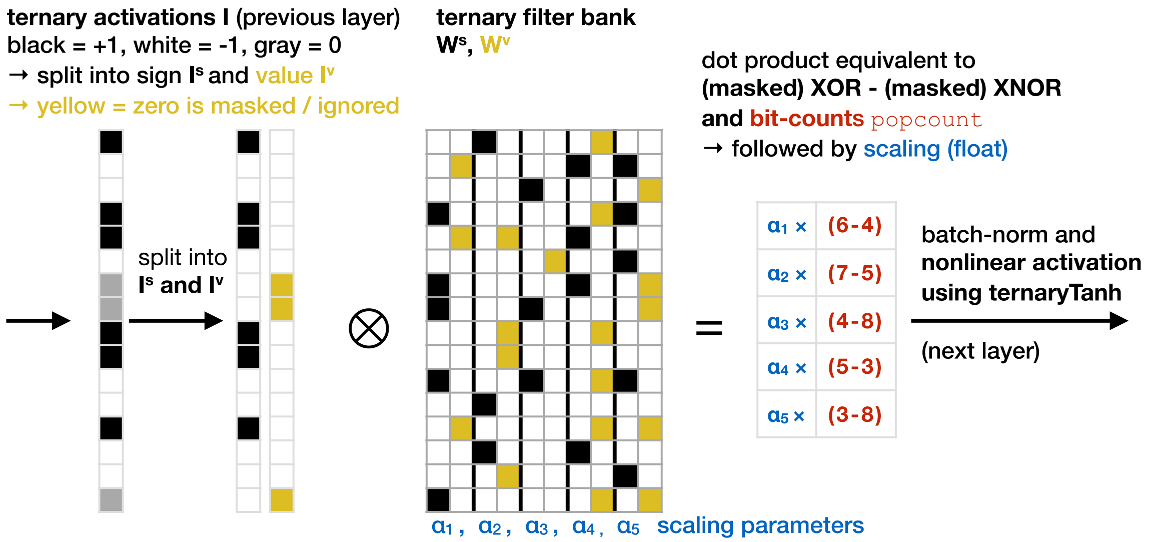
ternary activations I (previous layer)
black = +1, white = -1, gray = 0
→ split into sign Iˢ and value Iᵛ
→ yellow = zero is masked / ignored
split into
Iˢ and Iᵛ
ternary filter bank
Wˢ, Wᵛ
dot product equivalent to
(masked) XOR - (masked) XNOR
and bit-counts popcount
→ followed by scaling (float)
α1 × (6-4)
α2 × (7-5)
α3 × (4-8)
α4 × (5-3)
α5 × (3-8)
batch-norm and
nonlinear activation
using ternaryTanh
(next layer)
α1 , α2 , α3 , α4 , α5 scaling parameters

**Fig. 3** Visual example for the computation of ternary convolutions without floating-point operations. Ternary values are encoded by sign and value, i.e. +1 → (■,□), -1 → (□,□) and 0 → (□,🟨). The approximation for a ternary filter bank provides scaling parameters $\alpha$ see below Eq. 4. Ternary convolutions can be computed by masked XOR and XNOR operators followed by a bit-count according to Eq. 9. The output is batch normalised and passed on to the nonlinearity visualised in Fig. 2

Intel CPUs and that can process 8 packed FMAs in parallel in 0.5 cycles, we can gain a speed up of a factor of 8. When considering equal power consumption (floating-point operations require more complex logic), the improvements are even much higher.

Since previous work on binary quantisation of deep learning architectures [4,25] has led to severely reduced accuracy of 12–20% for image classification tasks, we aim to extend the concept of bit counting as replacement for matrix multiplications to ternary valued networks with $\mathbf{I}, \mathbf{W} \in \{-1, 0, +1\}^c$. As shown in Fig. 3, we can store ternary tensors using 2 bits per entry that encode the sign and value, respectively. We denote these two tensors as $\mathbf{I}^s, \mathbf{I}^v \in \{0, 1\}^c$ and $\mathbf{W}^s, \mathbf{W}^v \in \{0, 1\}^c$. The inner product calculation can then be realised using two bit-counts in Hamming space:

$$\mathbf{I}_i\mathbf{W}_j = \Xi\left\{\overline{(\mathbf{I}_i^s \oplus \mathbf{W}_j^s)}\&(\mathbf{I}_i^v + \mathbf{W}_j^v)\right\} - \Xi\left\{(\mathbf{I}_i^s \oplus \mathbf{W}_j^s)\&(\mathbf{I}_i^v + \mathbf{W}_j^v)\right\} \tag{9}$$

Here, & defines an AND operator, + the Boolean OR and $\overline{A \oplus B}$ the negated XOR. A more intuitive interpretation is that all operations involving a zero value are excluded and the first part of the equation calculates all positives elements of a dot product, i.e. $+1 \cdot +1$ and $-1 \cdot -1$, while the second part subtracts the number of times an opposing sign multiplication occurs. The complete concept of an individual building block for ternary convolutions in deep networks is shown in Fig. 3. In practice further speed-ups (halving the number of bit-counts) are possible when training the weight quantisation to follow a specified degree of sparsity, e.g. by replacing the rule derived in Eq. 4 and specify $\Delta$ so that in each kernel exactly 50% of entries are zero.

In summary, each module in our proposed TernaryNet architecture comprises a ternary approximation of filter weights together with a ternarisation of activations to enable low-power, high-speed ternary convolutions without floating-point operations. During training both weight updates for mini-batch optimisation and the activations using the new ternary hyperbolic tangent ternTanh are kept at full precision to enable gradient flow and precise learning. By extending the strategy of [2] to ternary activations and applying a continuously increasing slope $\beta$ during training, the network learns to cope with sparse and quantised activations, which is vital in order to avoid diverging objectives between training and testing. Batch-normalisation layers [12] are inserted between ternary convolutions and activations to accelerate the learning process and keep a zero mean of feature responses as well as an approximately unit normal distribution to ensure the nonlinearity is not easily saturated. A trained model can be stored using only 2 bits per weight and one (full-precision) scalar weighting value per feature channel—reducing the required memory by more than an order of magnitude. During model inference on unseen data, we employ the hard quantisation of Eq. 6 and thereby enable the use of Hamming distances for ternary convolutions. It is important to note that all common architectural design choices of modern deep networks, such as skip connections [26], dilated kernels [10,34] or dense feature concatenation [6,11] are useable with ternary convolutions.

## Experiments

To demonstrate the usefulness of TernaryNets for highly efficient medical image analysis, we explore the dense prediction (semantic segmentation) of the pancreas in CT. The extension of our model to multi-organ labelling is straightforward. Providing image guidance for interventional tasks relies on fast inference executed on common clinical workstations or even mobile devices. We therefore also analyse in detail the computational operations and memory requirements in our experiments. The highly variable shape and a relatively poor contrast of the pancreas as well as confusable neighbouring abdominal anatomies make this segmentation very difficult. Therefore, networks with large receptive fields are required to robustly capture sufficient regional context, while at the same time an automatic method should delineate local objects boundaries accurately and avoid oversegmentation of similar neighbouring structures within the field-of-view. Our experiments are based on the public NIH dataset that was described in [27]. It comprises 82 high-resolution CT scans along with accurate manual segmentations for training and validation.

*Comparison to state of the art:* Several approaches have been evaluated in the last few years on the NIH dataset and a similar corpus of abdominal CT scans (the BCV challenge data described in [32]). Accuracies for pancreas segmentation without CNNs are often relatively low, e.g. overlap scores of 40 and 49% have been reported for two different multi-atlas techniques in [31]. Employing discrete registration within multi-atlas label fusion [9] improved accuracies for pancreas segmentation to 74% Dice, ranking first at the MICCAI 2015 BCV challenge. The approach of [16] reached 60% overlap within the same challenge by combing registration-based localisation with deep CNNs. Roth et al. achieved a Dice score of 71% [27] on the NIH dataset when combining supervoxel-based deep region regression with CNN patch classification and could further improve their accuracy to 78% [28] using holistically nested networks together with random forest classifiers. Very recently, Zhou et al. [36] achieved an astonishing performance of 82% on the NIH data by training an iterative sequence of multiple (coarse-to-fine) deep CNNs. The use of densely connected layers within a V-Net architecture (Dense V-Net [6]) resulted in a Dice overlap of 66% (on both NIH and BCV datasets), which is also the only of the mentioned deep learning approaches that did not rely on a combination of classifiers or registration. In our own previous work [10], we reached 65% Dice for the BCV dataset using (untrained) sparse binary convolutions that enabled huge receptive fields but no binary (or ternary) convolutions.

*Baseline model*: Our aim is not necessarily to surpass current state-of-the-art accuracies, but to demonstrate and analyse the effects of network model quantisation. We therefore employ a four-level fully convolutional U-Net architecture [26] as an exemplary baseline. To fairly assess the influence of binarisation and ternarisation, we employ the same number of channels and convolution filters for all compared models and hyperbolic tangents (except for the final prediction layer) as baseline activation function. Table 1 lists the details of the chosen architecture, including the number of floating-point operations (FMAs) required per layer. The resulting receptive field of our networks is 36 voxels. Using floating-point precision, the network requires 2.6 million weights and thus 10.6 MBytes of storage for the model weights. During training, the model requires more than 5 GBytes of memory (using a mini-batch size of 10). For inference, this can be reduced to approximately 1 GByte.

*Compared models:* We have analysed in total seven variants of our baseline network to explore the effect of sparsity and quantisation to both activations and filter weights. Starting from the same baseline model, we define our TernaryNet by approximating weights using the ternary quantisation of Eq. 4 as proposed in [18]. The first layer always acts on continuous input and similar to previous work on binarisation [4,25] we performed no weight quantisation for it. As evident from the layer details in Table 1 the computational demand of this layer, with 1.76% of total MFlops and 0.17% of all weights, is negligible. During training we varied the value of $\beta$ in Eq. 7 linearly (and evenly with epochs) from 3.0 to 8.0 following the principal of continuation of [2]. The variant *no continuation* uses a fixed $\beta = 3$ for all epochs. To quantify whether our approach successfully reduces quantisation loss, we also compare a variant *without quantisation* that does not realise ternary convolutions. For binary convolutional networks (termed XNORnet [25], see Eq. 8), we explore the ad hoc gradient approximation according to the seminal work in [4]. As alternative, we adopt the continuation (see Eq. 3) for a classical tanh nonlinearity. Finally, the full-precision network is compared with ReLU activations for completeness.

*Data processing:* We resampled the original scans of the NIH dataset that had axial dimensions of 512 × 512 and 181–466 slices with thicknesses between 0.5 to 1.0 mm to isotropic voxel sizes of 1.0mm$^3$. We then performed a region-of-interest cropping with bounding boxes of dimensions 194 × 122 × 138 around the pancreas, yielding an approximate density of 2% for organ voxels (and 98% background). There exist several accurate algorithms that automatically predict bounding boxes and/or organ locations, e.g. [29,33], which could be employed for this task so it was considered out of scope for our study. Subsequently, we applied a zero mean unit variance transformation on the cropped CT volumes. Following related work on pancreas segmentation using CNNs [27,36], we employ only 2D convolutions, but provide a stack of several neighbouring slices (15 in our experiments) to each network. The output for each stack will be a probabilistic map of foreground and background proba-

**Table 1** Description of baseline U-Net model

| Layer | (Out)-Size | Kernel | # Channels | MFlops | Skip |
|---|---|---|---|---|---|
| Input | 236 × 172 × 15 | | | | |
| #1 Conv3D | 234 × 170 | 3 × 3 × 15 | 32 | 172 | |
| #2 Conv2D | 232 × 168 | 3 × 3 | 64 | 718 | →#13 |
| #3 Conv2D | 228 × 164 | 3 × 3 | 64 | 345 | |
| AvgPool2D | 114 × 82 | 2 × 2 | | | |
| #4 Conv2D | 112 × 80 | 3 × 3 | 128 | 661 | →#11 |
| #5 Conv2D | 108 × 76 | 3 × 3 | 128 | 303 | |
| AvgPool2D | 54 × 38 | 2 × 2 | | | |
| #6 Conv2D | 52 × 36 | 3 × 3 | 256 | 552 | →#9 |
| #7 Conv2D | 52 × 36 | 1 × 1 | 256 | 31 | |
| AvgPool2D | 26 × 18 | 2 × 2 | | | |
| #8 Conv2D | 26 × 18 | 1 × 1 | 256 | 31 | |
| Upsample2D | 52 × 38 | 2 × 2 | | | |
| #9 Conv2D | 50 × 34 | 3 × 3 | 256 | 2005 | #6 → |
| #10 Conv2D | 48 × 32 | 3 × 3 | 128 | 453 | |
| Upsample2D | 96 × 64 | 2 × 2 | | | |
| #11 Conv2D | 94 × 62 | 3 × 3 | 128 | 1719 | #4 → |
| #12 Conv2D | 92 × 60 | 3 × 3 | 64 | 407 | |
| Upsample2D | 184 × 118 | 2 × 2 | | | |
| #13 Conv2D | 180 × 116 | 3 × 3 | 64 | 1583 | #2 → |
| #14 Conv2D | 176 × 110 | 3 × 3 | 64 | 770 | |
| Prediction | 176 × 110 | 3 × 3 | 2 | 2 | |

Number of million fused multiply add (floating-point operations) is given as MFlops. To reduce the number of trainable parameters the convolutions in the lowest resolution level are 1 × 1. Outgoing and incoming skip connections are noted in the last column

bilities for the given central slice. No form of post-processing is employed, which could potentially further increase accuracy, but also influences the assessment of differences across methods.

*Training and implementation details:* We use a mini-batch size of 10 and Adam as optimiser with an initial learning rate of 0.0025. Each network is trained for 40 epochs with 150 iterations (1500 3D input stacks) without early stopping. The hyperparameters are not specifically optimised and kept same for all variants. Since we encountered a huge class imbalance, we use a weighted cross-entropy loss with 0.5 for background and 2.5 for organ pixels, but alternatively a Dice loss function [21] could automatically deal with it. We trained 5 separated folds of training and validation splits using 65–66 scans for training and 16–17 for testing. The derivatives of our ternary activation and the equivalent binary $\tanh(x)$ can be found analytically (using automatic differentiation), for the ad hoc approximation of binary activations in Eq. 2 we implemented a custom forward and backward pass. When approximating filter kernels, we keep a copy of the full-precision weights, perform the quantisation before forward pass and restore the original values after the backward pass and before calling the optimiser that performs a gradient step. To enable a reproduction of our results and further research, our pytorch implementation and pre-trained models will be made publicly available after submission at https://github.com/mattiaspaul/TernaryNet.

## Results and discussion

The performance of the seven compared models is evaluated quantitatively in terms of Dice overlap between automatic prediction (without further post-processing) and manual annotation. Average Dice values (and standard deviations) are compared in Table 2 alongside with statistical significance tests and memory usage for model parameters.

It can be seen that our proposed ternary convolutions perform on par with full-precision networks reaching an average Dice of 71.0%. This demonstrates the robustness and high accuracy of our proposed ternary quantisation scheme. The results are also comparable to a number of recent deep learning approaches that all relied on full precision and thus much larger and more complex models. When replacing the $\tanh(x)$ nonlinearity with a ReLU in the full-precision model, its accuracy can be further improved by 3.8%. However,

**Table 2** Dice overlap measures of pancreas for 82 CT scans (fivefold cross-validation)

| Architecture | **Avg. Dice** | stddev | *p* value | weight memory (MBytes) |
|---|---|---|---|---|
| Binary XNORnet | | | | |
| (continuation Eq. 3) | 48.4% | ±20.1 | $\ll$0.001 (–) | 0.33 |
| (adhoc gradients Eq. 2) | 66.9% | ±10.5 | 0.01(–) | 0.33 |
| TernaryNet | | | | |
| (using $\beta \rightarrow \infty$ in Eq. 5) | **71.0%** | **±9.5** | * | **0.66** |
| (without quantisation) | 71.8% | ±10.7 | 0.60 (o) | 0.66 |
| (no continuation in training) | 56.3% | ±19.3 | $\ll$0.001 (–) | 0.66 |
| Full-precision U-Net | 71.9% | ±10.2 | 0.54 (o) | 10.6 |
| ReLU instead of tanh | 75.7% | ±9.0 | 0.001 (+) | 10.6 |

Paired t tests are performed for significance analysis against TernaryNet, where (-) indicates that our method performed significantly better

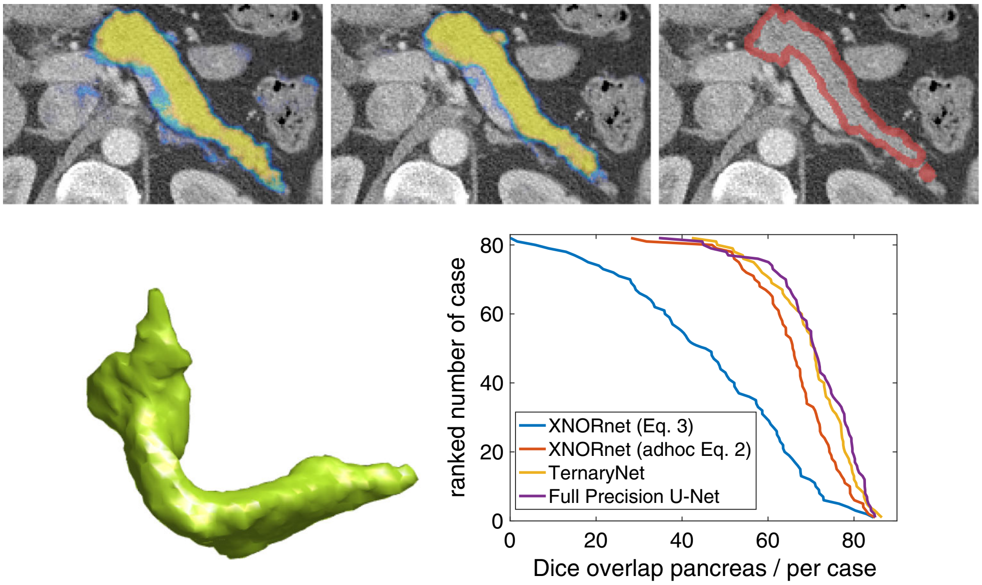

**Fig. 4** Top row: A visual comparison of case # 12 of the NIH demonstrates small but significant advantages of the ternary quantisation (middle) over the better performing ad hoc binary activation and quantisation, which oversegments a neighbouring structure (left). Our approach better matches the manual segmentation (right). Bottom row: 3D visualisation of our segmentation shows a very smooth surface (left). Ranked (sorted) Dice score compared across methods demonstrate that the full-precision model is not significantly better than our heavily quantised TernaryNet. Both Binary XNORnet variants perform inferior

the presumptions that symmetric activations are nowadays unsuitable to reach high accuracy has been refuted. Possibly, because the U-Net and similar architectures enable a very good backwards flow of gradients through their skip connections. The performance of binary quantisation is significantly lower than our approach. This is in particular evident for the variant that uses an analytically differentiable activation (see Fig. 4). We assert that this underlines the importance of sparse activations, which can contain a larger number of zero values—a key feature of our new nonlinearity. Sparse intermediate feature maps enable the network to adapt certain filter banks to specific subproblems while being unaffected by pertubations of unrelated data.

Training one entire model (within 40 epochs) requires about 15 min on an NVIDIA Titan Xp. Inference of the full-precision network on a CPU takes about 80 s. When employing a customised OpenCL implementation for Hamming distance calculation (used for ternary convolutions in Eq. 9), we estimated inference times of 5–7 s using a dual-core mobile CPU. This represents a more than 10× speed-up through our contributions. Further speed-ups can be gained by reducing the number of parameters in the expanding path and skipping every other slice in a 3D volume (and interpolating in between) or adjusting the ternary weight quantisation to increase sparsity and reduce the number of population counts.

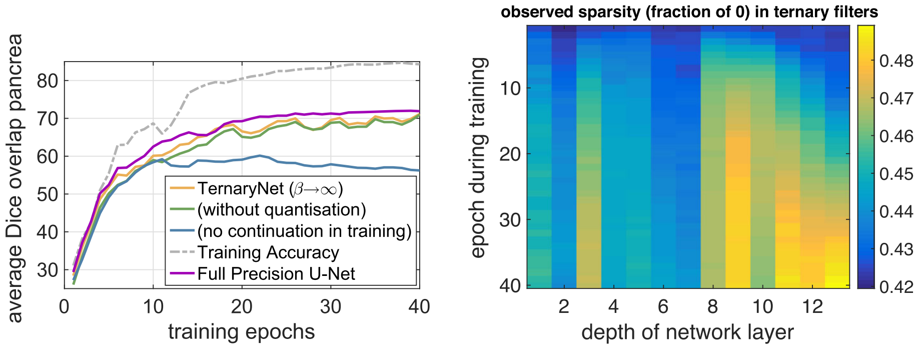

**Fig. 5** Left: By employing the continuation technique with increasing $\beta$ values during training epochs, we can significantly improve the outcome of our trained networks. The ternarised quantisation does thereby no longer affect segmentation quality measured in Dice overlap and approaches the quality of a full-precision U-Net. When comparing these validation curves with training accuracies, only a moderate gap is visible and no overfitting occurs. Right: The observed sparsity (fraction of zero values) in the trained ternary weights increases throughout the training process. This effect is more pronounced for deeper layers with high parameter counts

When analysing the sparsity of filter weights learned by our model across epochs, shown in Fig. 5, one can see a tendency to an increase in zero values in later layers and later epochs. These findings are supported by [22], which explore more sparsity in deeper layers together with increased accuracy. In comparison with the number of trainable weights in Table 1, it is notable that layers with increased sparsity at the end of training also contain most free parameters. This indicates that the model automatically avoids overfitting and sparsity acts as a regulariser. The importance of adapting the slope in our ternary hyperbolic tangent nonlinearity during training is clearly shown in Fig. 5, where the average Dice is plotted across training epochs. Note that the evaluation on validation cases always employs ternary convolutions and accordingly quantises activations using Eq. 6.

*Limitations and potential for further extensions:* While the results of a TernaryNet come close to a full-precision U-Net with hyperbolic tangent activation, there is a loss in accuracy of 3.9% to the more common ReLU variant. We empirically found that using a ReLU6 (which cannot exceed an output of 6) [15] performs as well (75.7% avg. Dice). Therefore, the performance gap could most likely be closed by increasing the expressiveness of the quantised activation.

## Conclusion

We have presented a pioneering approach for ternary convolutions in deep neural networks that relies on both ternarised activations and filter weights. Our work goes beyond previous efforts of binarisation that has often led to severe model degradation. In our experiments, we demonstrated that the TernaryNet maintains the high segmentation quality of the corresponding full-precision U-Net (around 71% Dice for pancreas CT with further potential for improvements), while realising 10× speed improvements and 15× lower memory requirements. This is in particular important when executing model inference for image-guided interventions on clinical or mobile computing hardware. We believe that the detailed description, publicly available implementation and convincing empirical findings along with the generality of our approach will help transfer the concept of ternary convolutions to other deep learning applications. We have seen a clear importance of designing a ternary activation that is analytically differentiable based on the underlying hyperbolic tangent nonlinearity as well as using a continuous adaption of its slope during training. This eases the complex training process and results in a high sparsity that is desirable for generalisation and supported by theoretical analysis in literature. When proven in other related fields of computer vision, we strongly believe that quantised networks will have an increasing impact and potentially lead to a wider adaptation of its underlying computational blocks (population counts) in mobile processors.

**Acknowledgements** We gratefully thank Nassim Bouteldja who assisted this work by pre-processing the CT scans and helpful discussions. We also acknowledge the great support of NVIDIA with the donation of a Titan Xp GPU used for this research.

## Compliance with ethical standards

**Conflict of interest** The authors declare that they have no relevant conflict of interest.

**Ethical approval** This article does not contain any studies with human participants or animals performed by any of the authors.

**Informed consent** Statement of informed consent was not applicable since the manuscript does not contain any participants' data.

## References

1. Calonder M, Lepetit V, Ozuysal M, Trzcinski T, Strecha C, Fua P (2012) BRIEF: computing a local binary descriptor very fast. PAMI 34(7):1281–1298
2. Cao Z, Long M, Wang J, Yu PS (2017) Hashnet: deep learning to hash by continuation. ICCV
3. Conjeti S, Roy AG, Katouzian A, Navab N (2017) Hashing with residual networks for image retrieval. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 541–549
4. Courbariaux M, Hubara I, Soudry D, El-Yaniv R, Bengio Y (2016) Binarized neural networks: training deep neural networks with weights and activations constrained to +1 or −1. NIPS
5. Farag A, Elhabian S, Graham J, Farag A, Falk R (2010) Toward precise pulmonary nodule descriptors for nodule type classification. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 626–633
6. Gibson E, Giganti F, Hu Y, Bonmati E, Bandula S, Gurusamy K, Davidson BR, Pereira SP, Clarkson MJ, Barratt DC (2017) Towards image-guided pancreas and biliary endoscopy: automatic multi-organ segmentation on abdominal CT with dense dilated networks. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 728–736
7. Heinrich MP, Blendowski M (2016) Multi-organ segmentation using vantage point forests and binary context features. In: Ourselin S, Joskowicz L, Sabuncu MR, Unal G, Wells W (eds) MICCAI 2016 LNCS. Springer, pp 598–606
8. Heinrich MP, Jenkinson M, Papież BW, Glesson FV, Brady M, Schnabel JA (2013) Edge-and detail-preserving sparse image representations for deformable registration of chest MRI and CT volumes. In: International conference on information processing in medical imaging. Springer, pp 463–474
9. Heinrich MP, Maier O, Handels H (2015) Multi-modal multi-atlas segmentation using discrete optimisation and self-similarities. In: VISCERAL challenge@ ISBI, pp 27–30
10. Heinrich MP, Oktay O (2017) BRIEFnet: deep pancreas segmentation using binary sparse convolutions. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 329–337
11. Huang G, Liu Z, van der Maaten L, Weinberger KQ (2017) Densely connected convolutional networks. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 2261–2269
12. Ioffe S, Szegedy C (2015) Batch normalization: accelerating deep network training by reducing internal covariate shift. In: International conference on machine learning, pp 448–456
13. Jia Y (2014) Learning semantic image representations at a large scale
14. Juefei-Xu F, Boddeti VN, Savvides M (2017) Local binary convolutional neural networks. In: IEEE conference on computer vision and pattern recognition (CVPR). IEEE, pp 4284–4293
15. Krizhevsky A, Hinton G (2010) Convolutional deep belief networks on cifar-10. Unpublished manuscript 40:7
16. Larsson M, Zhang Y, Kahl F (2017) Robust abdominal organ segmentation using regional convolutional neural networks. In: Scandinavian conference on image analysis. Springer, pp 41–52
17. Lee H, Battle A, Raina R, Ng AY (2007) Efficient sparse coding algorithms. In: Advances in neural information processing systems, pp 801–808
18. Li F, Zhang B, Liu B (2016) Ternary weight networks. In: NIPS workshop on efficient methods for deep neural networks (EMDNN). arXiv:1605.04711
19. Liu B, Wang M, Foroosh H, Tappen M, Pensky M (2015) Sparse convolutional neural networks. In: CVPR, pp 806–814
20. Long J, Shelhamer E, Darrell T (2015) Fully convolutional networks for semantic segmentation. In: CVPR, pp 3431–3440
21. Milletari F, Navab N, Ahmadi SA (2016) V-Net: fully convolutional neural networks for volumetric medical image segmentation. In: 3D vision, IEEE, pp 565–571
22. Molchanov D, Ashukha A, Vetrov D (2017) Variational dropout sparsifies deep neural networks. In: International conference on machine learning, pp 2498–2507
23. Muła W, Kurz N, Lemire D (2018) Faster population counts using AVX2 instructions. Comput J 61(1):111–120
24. Olshausen BA, Field DJ (1996) Emergence of simple-cell receptive field properties by learning a sparse code for natural images. Nature 381(6583):607
25. Rastegari M, Ordonez V, Redmon J, Farhadi A (2016) XNOR-Net: imagenet classification using binary convolutional neural networks. In: Leibe B, Matas J, Sebe N, Welling M (eds) ECCV. Springer, pp 525–542
26. Ronneberger O, Fischer P, Brox T (2015) U-net: convolutional networks for biomedical image segmentation. In: Navab N, Hornegger J, Wells WM, Frangi AF (eds) MICCAI 2015 LNCS. Springer, pp 234–241
27. Roth HR, Lu L, Farag A, Shin HC, Liu J, Turkbey EB, Summers RM (2015) DeepOrgan: multi-level deep convolutional networks for automated pancreas segmentation. In: Medical image computing and computer-assisted intervention. Springer, pp 556–564
28. Roth HR, Lu L, Farag A, Sohn A, Summers RM (2016) Spatial aggregation of holistically-nested networks for automated pancreas segmentation. In: Ourselin S, Joskowicz L, Sabuncu MR, Unal G, Wells W (eds.) MICCAI 2016 LNCS. Springer, pp 451–459
29. Urschler M, Ebner T, Štern D (2018) Integrating geometric configuration and appearance information into a unified framework for anatomical landmark localization. Med image anal 43:23–36
30. Wolterink JM, Leiner T, Viergever MA, Išgum I (2016) Dilated convolutional neural networks for cardiovascular MR segmentation in congenital heart disease. In: MICCAI RAMBO, pp 95–102
31. Xu Z, Burke RP, Lee CP, Baucom RB, Poulose BK, Abramson RG, Landman BA (2015) Efficient multi-atlas abdominal segmentation on clinically acquired CT with SIMPLE context learning. Med Image Anal 24(1):18–27
32. Xu Z, Lee C, Heinrich M, Modat M, Rueckert D, Ourselin S, Abramson R, Landman B (2016) Evaluation of six registration methods for the human abdomen on clinically acquired CT. IEEE Trans Biomed Eng pp 1–10
33. Xu Z, Panjwani SA, Lee CP, Burke RP, Baucom RB, Poulose BK, Abramson RG, Landman BA (2016) Evaluation of body-wise and organ-wise registrations for abdominal organs. In: SPIE medical imaging, p 97841
34. Yu F, Koltun V (2015) Multi-scale context aggregation by dilated convolutions. arXiv preprint arXiv:1511.07122
35. Zhang X, Liu W, Dundar M, Badve S, Zhang S (2015) Towards large-scale histopathological image analysis: hashing-based image retrieval. IEEE Trans Med Imag 34(2):496–506
36. Zhou Y, Xie L, Shen W, Wang Y, Fishman EK, Yuille AL (2017) A fixed-point model for pancreas segmentation in abdominal CT scans. In: International conference on medical image computing and computer-assisted intervention. Springer, pp 693–701
37. Zhu C, Han S, Mao H, Dally WJ (2017) Trained ternary quantization. ICLR conference