ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

# Spatial Chirp-Z Transformer Networks

Jonas Degrave, Sander Dieleman, Joni Dambre and Francis wyffels *

Electric and Information Systems (ELIS)
Ghent University
Sint-Pietersnieuwstraat 41, 9000 Gent - Belgium

**Abstract**. Convolutional Neural Networks are often used for computer vision solutions, because of their inherent modeling of the translation invariance in images. In this paper, we propose a new module to model rotation and scaling invariances in images. To do this, we rely on the chirp-Z transform to perform the desired translation, rotation and scaling in the frequency domain. This approach has the benefit that it scales well and that it is differentiable because of the computationally cheap sinc-interpolation.

## 1 Introduction

As a general principle in machine learning, models will often perform better when you include more a priori knowledge. One form of a priori knowledge often available on datasets, is whether there are data manipulations under which the required output of the model stays the same. For instance, you could think of small translations of the images for computer vision applications, or small delays in audio for speech recognition. A common approach to incorporate this knowledge in the model, is by performing data augmentation using these known invariances [1], which can be done both during training and evaluation of the model.

Another way of doing so, is by making the model inherently insensitive to known invariances in the data. An example of this approach are convolutional neural networks with max pooling. Because of the inherent properties of this model, the classification will be robust against small translations in the image. This property is one of the main reasons for their effectiveness in image classification [2].

Therefore, when we competed in Kaggle's National Data Science Bowl 2014, where the goal was to classify images of plankton, we wanted to improve our performance by including these known invariances. In the case of the dataset of the competition, there was full rotational and scale invariance. We tried various approaches to incorporate this invariance into our model, one of which is discussed in this paper. Earlier preliminary results were posted in March 2015 when publishing our winning solution online[1].

*The research leading to these results has received funding from the European Commission (EC) Human Brain Project under grant agreement No 604102, and from the Agency for Innovation by Science and Technology in Flanders (IWT). The Tesla K40 used for this research was donated by the NVIDIA Corporation.

[1] http://benanne.github.io/2015/03/17/plankton.html

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

### 1.1 Related work

Since then, various papers have already been published exploring the idea of using affine transforms as module in a neural network [3, 4]. Currently, these methods rely on using a bilinear transform for performing the interpolation step in the image transform. In this paper, we present the original approach we developed before these publications, which uses the chirp-Z transform to perform both the coordinate transform and interpolate the image.

## 2 The Chirp-Z Transform

The chirp-Z transform (CZT) is a generalization of the more known discrete Fourier transform (DFT). Seen from the Z-transform point of view, you could say that while the DFT samples the Z-plane at uniformly-spaced points on the unit circle, the chirp-Z transform samples along spiral arcs in the Z-plane. Or alternatively from the Laplace transform point of view, while the DFT samples along the imaginary axis in the S-plane, the CZT samples along straight lines in the S-plane [5].

Concretely, the chirp-Z transform is defined by the following equation:

$$\mathrm{CZT}(x_n) = \sum_{n=0}^{N-1} x_n A^{-n} W^{nk}.$$

Here, $A$ is the starting point of the sampling, and $W$ is a complex scalar describing the complex ratio between points on the sampling contour. When $A = 1$ and $W = e^{\frac{-i2\pi}{N}}$ this reduces to the standard DFT.

The 2-dimensional chirp-Z transform is an extension of this idea onto 2 dimensional images in exactly the same way the DFT is extended. In the case of the DFT, this allows for a fast algorithm to perform convolutions [6]. Similarly, the chirp-Z transform has some interesting properties as well. It can for instance be used to perform translations, scaling and even rotations on images [7]. The resulting images are perfect interpolations of the input image, so if the input image is bandwith-limited, the resulting image will be a perfect reconstruction. This makes this technique interesting when applied on images which have been reconstructed from the frequency domain, which is for instance the case in MRI-imaging [7]. Also, perfect interpolation implies no information is lost, which allows for repeated manipulation of the same image without blurring [8].

Additionally, since the chirp-Z transform is linear, it can be part of a gradient descent method and can be evaluated fast in both the forward and the backward phase. This makes it a good candidate for use in a deep neural network.

## 3 Transform of an Image Using the Chirp-Z Transform

As described in the paper by Myagotin [8], we want to resample our image on the points $(p\Delta, q\Delta)$ where $p$ and $q$ are the discrete indexes of the pixel, and $\Delta$ is the distance between the neighbouring pixels.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8.
Available from http://www.i6doc.com/en/.

Then, the location $(x_{pq}, y_{pq})$ of the sampling point on the original image are given by:

$$x_{pq} = x_0 + \cos\theta(p\Delta - x_0) - \sin\theta(q\Delta - y_0)$$
$$y_{pq} = y_0 + \sin\theta(p\Delta - x_0) + \cos\theta(q\Delta - y_0)$$

And correspondingly, now we want to sample our original image in the point $g_{pq}$ by reconstructing from the chirp-Z domain. If we follow the definition for the 2 dimensional chirp-Z transform from, namely

$$Z_{pq}(h, \alpha, \beta) = \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} h_{lm} e^{-2\pi i\alpha(lp+mq)} e^{-2\pi i\beta(mp-lq)} \tag{1}$$

the value of the reconstructed point $g_{pq}$ is given by [5]:

$$g_{pq} = e^{-\pi i((\cos\theta+\sin\theta)(p\Delta-x_0)+(\cos\theta-\sin\theta)(q\Delta-y_0))}\ Z_{pq}(h, \frac{-\Delta\cos\theta}{N}, \frac{-\Delta\sin\theta}{N}).$$

Here $h$ is the DFT of the input image shifted such that the center of rotation is at the origin of the image coordinate system.

This equation is evaluated efficiently using only DFT's and multiplications [8]. To see how this works, we substitute the exponents in equation 1 as follows:

$$(lp + mq) = -(q - l)(p - m) + lm + pq$$
$$2(mp - lq) = (q - l)^2 - (p - m)^2 + (m^2 - l^2) + (p^2 - q^2)$$

and introduce the following three matrices with $\alpha = \frac{-\Delta\cos\theta}{N}$ and $\beta = \frac{-\Delta\sin\theta}{N}$

$$A_{lm} = e^{-\pi i(2\alpha lm + \beta(m^2 - l^2))}$$
$$B_{lm} = e^{\pi i(2\alpha lm + \beta(m^2 - l^2))}$$
$$C_{pq} = e^{-\pi i(2\alpha pq - \beta(p^2 - q^2))}$$

then, it follows that

$$Z_{pq} = C_{pq} \sum_{l=0}^{N-1} \sum_{m=0}^{N-1} h_{lm} A_{lm} B_{(q-l)(p-m)}.$$

From which we can find the $g_{pq}$ we are looking for. If we use the circular convolution operator '$*$' and the elementwise Hadamard product '$\circ$', this can be rewritten to

$$Z = C \circ ((h \circ A) * B).$$

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

**Algorithm 1** Transform image $I$ around $(x_0, y_0)$ with angle $\theta$ and scale $\Delta$

$a \leftarrow \Delta \cos\theta$
$b \leftarrow \Delta \sin\theta$
$p, q \leftarrow [N/2, ..., N-1, 0, ..., N/2-1]$
$r, s \leftarrow [0, ..., N-1]$
$P_{jk} \leftarrow \exp(\pi i(2p_j x_0/N + 2q_k y_0/N - 2ap_j q_k - b(p_j^2 - q_k^2))))$
$B_{jk} \leftarrow \exp(\pi i(2ar_j s_k + b(r_j^2 - s_k^2))$
$D \leftarrow \text{IFFT}(\text{FFT}(\text{FFT}(I) \circ P) \circ \text{FFT}(B))$
**return** $|D|/N^2$

## 4 The Algorithm

To implement the algorithm, we assume an efficient implementation of the DFT is available, namely the Fast Fourier Transform (FFT) [9]. We assume the result of the FFT-method is available in the most common 'not shifted' form, namely with the DC component on the location $(0, 0)$. The forward pass of this algorithm can therefore be written as described in Algorithm 1.

In this algorithm, all operations are differentiable, and therefore the transform of the image is as well. Sampling the image in a lower resolution can be done by removing the higher frequencies of $\text{FFT}(I)$ before transforming. Since the goal is often to crop the image and selecting only the important part, the loss of superfluous information is often beneficial.

This algorithm is as fast as the FFT-transform. This is true both in the forward and in the backward pass, since the derivative of the FFT-transform to its input is the IFFT-transform, which is the same as the FFT up to a coefficent. Therefore the complexity of this transform is $\mathcal{O}(n^2 \log n)$.

## 5 Experiments

To evaluate this approach, we used the same cluttered MNIST-dataset as was used to test comparable spatial transform methods [4]. The dataset is created by placing 3 MNIST digits on a square canvas with a width of 100 pixels. The first digit is placed by randomly sampling a vertical $y$ position on the canvas. The horizontal $x$ positions were randomly sampled such that the entire sequence fits on the canvas and the digits do not overlap. Digits are placed following a slope sampled from $\pm 45°$ and cluttered by placing 8 patches of 9 by 9 pixels sampled from the original MNIST digits. The trainset has 60 000 samples for training, 10 000 for validation and 10 000 for testing.

For evaluation, we made use of 2 different types of networks, which were implemented using Theano [10] and Lasagne [11]. We used a forward network approach [3] and a recurrent neural network approach [4]. The setup of these neural networks are described in Table 1. In these two models, we test four different approaches.

1. We test the models using the original bilinear interpolation method. With

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

| FFN-SPN model | RNN-SPN |
|---|---|
| $2 \times 2$ maxpool | $2 \times 2$ maxpool |
| $3 \times 3$ convolution (20 filters) | $3 \times 3$ convolution (20 filters) |
| $2 \times 2$ maxpool | $2 \times 2$ maxpool |
| $3 \times 3$ convolution (20 filters) | $3 \times 3$ convolution (20 filters) |
| $2 \times 2$ maxpool | $2 \times 2$ maxpool |
| $3 \times 3$ convolution (20 filters) | $3 \times 3$ convolution (20 filters) |
| Denselayer (200 units) | GRU (256 units) |
| Denselayer (4 or 6 units) + linear | Denselayer (4 or 6 units) + linear |
| Spatial Transform Layer | Spatial Transform Layer |
| $3 \times 3$ convolution (96 filters) | $3 \times 3$ convolution (32 filters) |
| $2 \times 2$ maxpool | $2 \times 2$ maxpool |
| Dropout | Dropout |
| $3 \times 3$ convolution (96 filters) | $3 \times 3$ convolution (32 filters) |
| $2 \times 2$ maxpool | $2 \times 2$ maxpool |
| Dropout | Dropout |
| $3 \times 3$ convolution (96 filters) | $3 \times 3$ convolution (32 filters) |
| Dropout | Dropout |
| Denselayer (400 units) | Denselayer (400 units) |
| Denselayer (3 units) + softmax | Denselayer (3 units) + softmax |

Table 1: The two models used to test our spatial transform layer

this method, there are 6 parameters defining the sampling grid. This allows all affine transforms.

2. We test the model using a bilinear interpolation method, where no skew is allowed. Therefore, only rotation, scaling and translation is available. This means the images are transformed with 4 degrees of freedom.

3. We test using the chirp-Z method explained before.

4. We test these models when no transform or downsampling takes place.

As you may find in Table 2, we found that the use of spatial transformer networks significantly improves the achieved accuracy on the cluttered MNIST dataset compared to standard neural networks. Also, we find that our chirp-Z approach performs similarly to the bilinear approach without skew, being able to achieve a 1.8% error rate.

## 6 Conclusion

In this paper, we show it is possible to transform images in a way derivatives can be calculated to the original images and the parameters. We have shown that this approach to transforming images works similarly well as the now common bilinear transform implementation and that they outperform standard convolutional neural networks.

ESANN 2016 proceedings, European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning. Bruges (Belgium), 27-29 April 2016, i6doc.com publ., ISBN 978-287587027-8. Available from http://www.i6doc.com/en/.

| | **Cluttered MNIST Sequences** | | | |
|---|---|---|---|---|
| | bilinear | bilinear no skew | chirp-Z | no spatial |
| *Model* | *Err. (%)* | *Err. (%)* | *Err. (%)* | *Err. (%)* |
| FFN-SPN $d$=1 | 4.4 | 4.5 | 5.0 | 7.8 |
| FFN-SPN $d$=2 | 2.0 | 5.3 | 3.3 | ” |
| FFN-SPN $d$=3 | 2.9 | 3.6 | 4.8 | ” |
| RNN-SPN $d$=1 | 1.8 | 4.1 | 4.1 | ” |
| RNN-SPN $d$=2 | 1.5 | 1.7 | 1.8 | ” |
| RNN-SPN $d$=3 | 1.8 | 1.5 | 2.8 | ” |

Table 2: Per digit error test scores on the cluttered MNIST sequence, $d$ is the down-sampling factor.

We have shown that using spatial transform layers can considerably improve performance on problems where the data is found in a part of the image, because another neural network can learn to find this relevant part autonomously. This further lowers the requirement for pre-processing in convolutional neural networks.

## References

[1] Patrice Y Simard, Dave Steinkraus, and John C Platt. Best practices for convolutional neural networks applied to visual document analysis. *International Conference on Document Analysis and Recogntion (ICDAR)*, page 958, 2003.

[2] Yann LeCun and Yoshua Bengio. Convolutional networks for images, speech, and time series. *The handbook of brain theory and neural networks*, 3361(10), 1995.

[3] Max Jaderberg, Karen Simonyan, Andrew Zisserman, and Koray Kavukcuoglu. Spatial transformer networks. *arXiv preprint arXiv:1506.02025*, 2015.

[4] Søren Kaae Sønderby, Casper Kaae Sønderby, Lars Maaløe, and Ole Winther. Recurrent spatial transformer networks. *arXiv preprint arXiv:1509.05329*, 2015.

[5] Lawrence R Rabiner, Ronald W Schafer, and Charles M Rader. The chirp z-transform algorithm and its application. *Bell System Technical Journal*, 48(5):1249–1292, 1969.

[6] CSS Burrus and Thomas W Parks. *DFT/FFT and Convolution Algorithms: theory and Implementation*. John Wiley & Sons, Inc., 1991.

[7] Raoqiong Tong and Robert W Cox. Rotation of NMR images using the 2D chirp-Z transform. *Magnetic Resonance in Medicine*, 41(2):253–256, 1999.

[8] AV Myagotin and EV Vlasov. Efficient implementation of the image rotation method using chirp-Z transform. *Pattern recognition and image analysis*, 24(1):57–62, 2014.

[9] William T Cochran, James W Cooley, David L Favin, Howard D Helms, Reg Kaenel, William W Lang, George C Maling Jr, David E Nelson, Charles M Rader, Peter D Welch, et al. What is the fast fourier transform? *Proceedings of the IEEE*, 55(10):1664–1674, 1967.

[10] Frédéric Bastien, Pascal Lamblin, Razvan Pascanu, James Bergstra, Ian Goodfellow, Arnaud Bergeron, Nicolas Bouchard, David Warde-Farley, and Yoshua Bengio. Theano: new features and speed improvements. *arXiv preprint arXiv:1211.5590*, 2012.

[11] Sander Dieleman, Jan Schlüter, Colin Raffel, Eben Olson, Søren Kaae Sønderby, Daniel Nouri, Daniel Maturana, Martin Thoma, Eric Battenberg, Jack Kelly, Jeffrey De Fauw, Michael Heilman, Brian McFee, Hendrik Weideman, Kashif Rasul, and Jonas Degrave. Lasagne: First release, August 2015.