

9, (Optional) A Closer Look at Transposed Convolutions

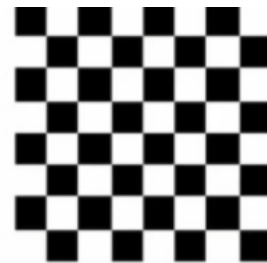
□ Intro

Now that you have an idea of what **transposed convolutions** are (also commonly referred to as "deconvolutions") and the checkerboard pattern problem that comes with using them, let's take a closer look! This interactive paper demonstrates the checkerboard pattern problem and how they are not exclusive to GANs, but any neural network that employs them.

Odena, et al., "Deconvolution and Checkerboard Artifacts", Distill, 2016. <http://doi.org/10.23915/distill.00003>

▼ problem of Checkerboard Artifacts

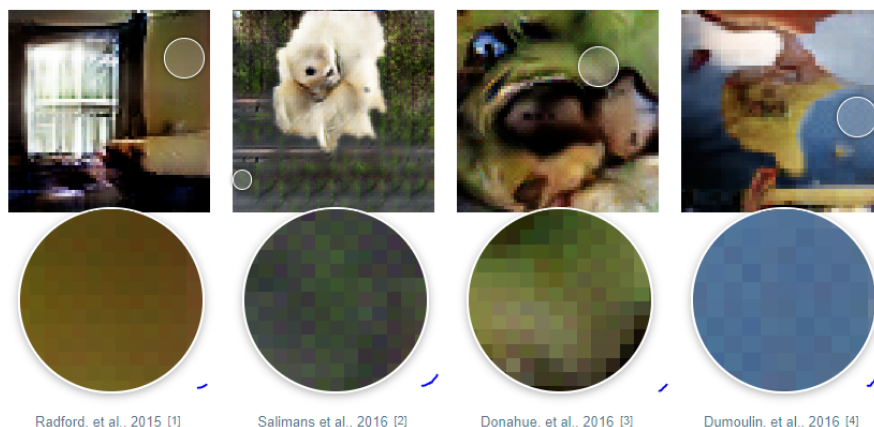
Problem: results have a checkerboard pattern



Deconvolution and Checkerboard Artifacts

AUGUSTUS ODENA	VINCENT DUMOULIN	CHRIS OLAH	Oct. 17	Citation:
Google Brain	Université de Montréal	Google Brain	2016	Odena, et al., 2016

When we look very closely at images generated by neural networks, we often see a strange checkerboard pattern of artifacts. It's more obvious in some cases than others, but a large fraction of recent models exhibit this behavior.



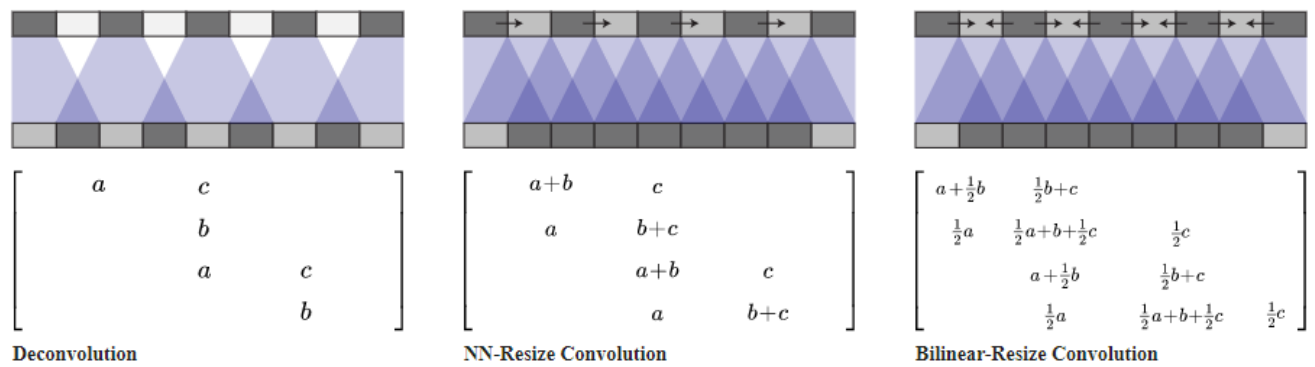
Radford, et al., 2015 [1]

Salimans et al., 2016 [2]

Donahue, et al., 2016 [3]

Dumoulin, et al., 2016 [4]

▼ reason behind of this checkerboard problem



▼ solution as for checkerboard artifacts caused by transposed convolutions

▼ 1, sub-pixel convolution (linear operations)

One approach is to make sure you use a kernel size that is divided by your stride, avoiding the overlap issue. This is equivalent to "**sub-pixel convolution**," a technique which has recently had success in image super-resolution [8]. However, while this approach helps, it is still easy for deconvolution to fall into creating artifacts.

▼ 2, separate out upsampling to a higher resolution (linear operations)

>> **nearest-neighbor interpolation OR bilinear interpolation**

Another approach is to separate out upsampling to a higher resolution from convolution to compute features. For example, you might resize the image (using **nearest-neighbor interpolation or bilinear interpolation**) and then do a convolutional layer. This seems like a natural approach, and roughly similar methods have worked well in image super-resolution (eg. [9]).

We don't necessarily think that either approach is the final solution to upsampling, but they do fix the checkerboard artifacts.

>> use upsampling instead of transposed convolution

▼ code

Resize-convolution layers can be easily implemented in TensorFlow using `tf.image.resize_images()`. For best results, use `tf.pad()` before doing convolution with `tf.nn.conv2d()` to avoid boundary artifacts.

□ Notes

□ Vocab

□ QOTD

□ Summary

