

# Deconvolution and Checkerboard Artifacts

AUGUSTUS ODENA et al.

2016/10/17

## 1 Deconvolution & Overlap

低解像度から高解像度な画像を生成しようとする時、一般に私たちは Deconvolution 操作を用いる。しかし、Deconvolution はカーネルサイズがストライドで割り切れない時に uneven overlap を引き起こす。ネットワークは原理的には慎重に uneven overlap を防ぐように重みを学習することができるが、実際にはネットワークは完全にこれを回避することに苦しむ。また、overlap pattern は二次元の場合がもっとも顕著である。二次元の場合、2つの overlap pattern がかけ合わされ、unevenness が二乗されてしまうからである。

また、主に最近では多層の Deconvolution 層を用いて一連の低解像度の描写から、繰り返しより大きな画像を作る。これらの多層の Deconvolution 層は artifacts を打ち消すことが可能だが、しばしば artifacts は組み合わされ様々な大きさの artifact が作り出される。

ストライド 1 の Deconvolution はうまくいったモデルの最終層に使われているのをしばしば目にする。これは artifacts を減らすのにかなり有効である。They can remove artifacts of frequencies that divide their size, and reduce others artifacts of frequency less than their size. また、Deconvolution はより低い周波数の artifacts を生成してしまう。

これらの artifacts は普通でない（平均的でない、強い）色を出力するときに最も顕著である。ニューラルネットワークの層は主にバイアス項を持っているため、平均的な色を出力するのは容易である。明るい赤の様な平均からかけ離れている色ほど Deconvolution はより多くの貢献を必要とする。

## 2 Overlap & Learning

理論上はモデルは uneven overlap となる位置に出力が均一になるように注意深く書き込むことができるようになる。これは、特に相互作用している複数のチャンネルを持っている時に成功するような、扱いにくい釣り合わせのための動作である。artifact を避けることは非常にフィルターの可能性を制限してしまう。実際には、ニューラルネットワークはこれらのパターンを完全に避けようともがく。

それどころか、uneven overlap を持ち artifact を避けることができないモデルだけでなく、even overlap を持つモデルにおいても、しばしば似たような artifact を引き起こすカーネルを学習する。それは uneven overlap を持つモデルの通常の動作ではないが、even overlap はいとも簡単に artifact を引き起こす。

完全に artifact を避けることはフィルターに対して大きな制約をかけることになり、実際上いまだに artifact に対してマイルドなモデルでも artifact が現れる。これにはおそらく沢山の要因がある。例えば GAN において 1つの問題は DIsriminator とその勾配についてだが、大きな問題は Deconvolution にあると思われる。

最善の場合でも、簡単に artifact を作りだす関数を表現してしまうので deconvolution は脆い（不安定）である。最悪の場合、deconvolution は常に artifact を作り出してしまう。

### 3 Better Upsampling

上記のような artifacts を生成しない Upsampling の一つ目の手法はカーネルサイズがストライドで割り切れるようにし、uneven overlap を避ける方法である。これは最近提案された sub-pixel convolution [1] と等しい。しかしこの手法ではまだ容易に artifact を作り出してしまう。

もう一つの手法は高解像度への upsampling と特徴量を計算する convolution とを切り離すことである。例えば nearest-neighbor interpolation や bilinear interpolation を用いて画像を resize したのち convolution 層へ通す。これは自然な発想であり、超解像においてうまくいった手法とだいたい似ている。deconvolution とこの resize-convolution は共に線形操作であり、行列として解釈することができる。これは2つの違いを見るのに役立つ方法である。deconvolution が各 output window に対して特定の入力を持つ場合、resize-convolution は暗に high frequency artifacts を抑える方法で、重み共有をしていることになる。

私たちは nearest-neighbor interpolation で最良の結果を得て、bilinear interpolation はうまくいかなかった。しかしこれは単に、私たちのモデルにおいて nearest-neighbor interpolation が deconvolution のために最適化されたハイパーパラメーターで偶然うまくいったことを意味している。また、素朴に画像特徴の high-frequency に強い耐性のある bilinear interpolation を使用したことによって発生した問題であることをさしている。

### 4 Image Generation Results

私たちの実験では nearest-neighbor interpolation が幅広い状況でうまくいった。この手法が有効であった1つのケースは GAN である。単に deconvolution 層を、nearest-neighbor interpolation の後に convolution が続く層に切り替えただけで様々な周波数の artifact が消えた。実は、artifact に関する差は学習が行われる前に見られる。ランダムな重みで初期化された generator が作りだす画像に注目すると、私たちは artifact を見ることができる。

これはつまり、artifact は画像を生成する方法のせいであり、GAN の学習方法の問題ではないことを示している。

私たちは様々な種類のモデルで artifact を確認し、upsampling 方法を resize-convolution にり替えると artifact がなくなるとことがわかった。これらが先ほどの artifact が GAN 特有ではないと言える証拠である。確認に使用した様々なモデルは checkerboard に脆いことが判明した（特に、コスト関数が明示的に checkerboard に対して明示的に耐性がない場合）。しかし、resize-convolution を採用すると artifact が消えた。

### 5 Artifacts in Gradient

convolution 層の勾配を計算するときはいつでも、backward pass において deconvolution を行う。これは勾配において画像生成に deconvolution を使う時と同様に checkerboard patterns を引き起こすことができてしまう。

私たちはいくつかのケースで checkerboard patterns が起こるのを発見した。Generator が checkerboard patterns に対して毛嫌いまたは偏愛している、どちらの場合でもない時、Discriminator における strided convolution は checkerboard patterns を引き起こすことができる。

GAN においてこの artifacts in Gradient が影響しているのかどうかはいまいちわからなかった。影響しているように捉えることができそう。

## 6 Conclusion

## 参考文献

- [1] Wenzhe Shi, et al., Real-Time Single Image and Video Super-Resolution Using an Efficient Sub-Pixel Convolutional Neural Network  
<https://arxiv.org/abs/1609.05158>