Deep Learning for ECE EECE-580G

Convolutions - From the brain to the GPU

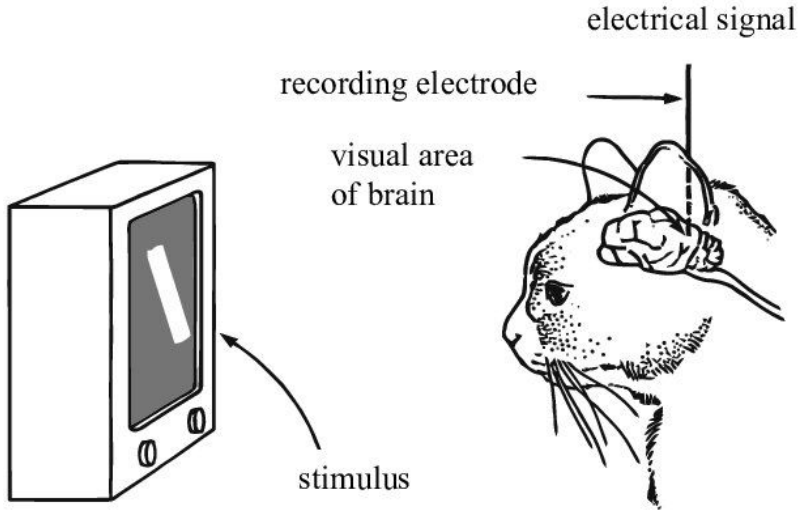
Hubel & Wiesel experiments

David Hubel and Torsten Wiesel

Source: Harvard Brain Tour

Hubel & Wiesel experiments

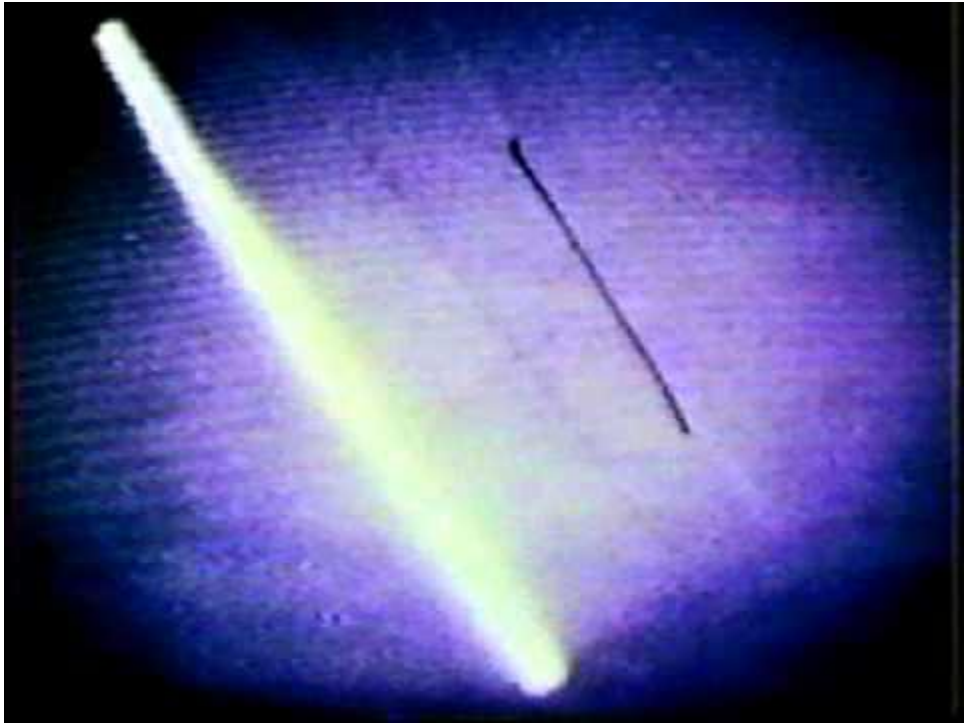
- 1959 - 1962 papers
- Slide projector showing specific patterns and mouvements
- Single neuron activity recording
- Used the the Tungsten electrode (Hubel 1957) to recorde single neuron activity
- Nobel prize winners (1981) for their work on receptive field in the visual cortex
- The Tungsten electrode is still a common tool to record electrical activity of cells. cc Neuralink :)



Source: Nguyen, Anh, Jason Yosinski, and Jeff Clune. "Understanding neural networks via feature visualization: A survey." Explainable AI: Interpreting, Explaining and Visualizing Deep Learning. Springer, Cham, 2019.

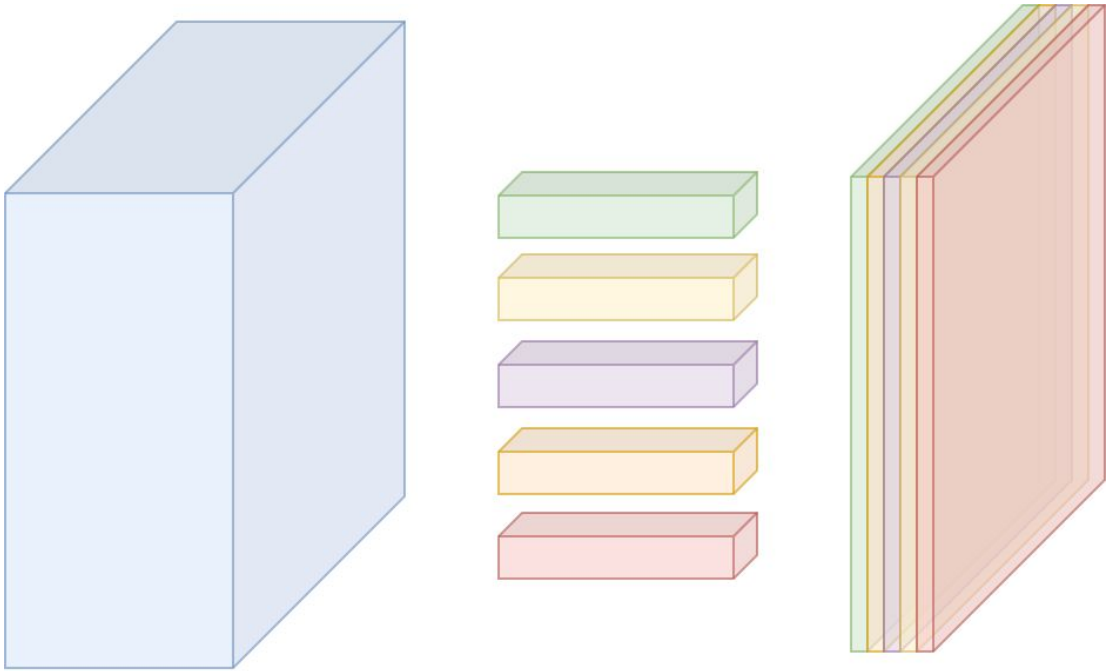
Ethics of animal experimentation: Animal Welfare Act

Hubel & Wiesel experiments



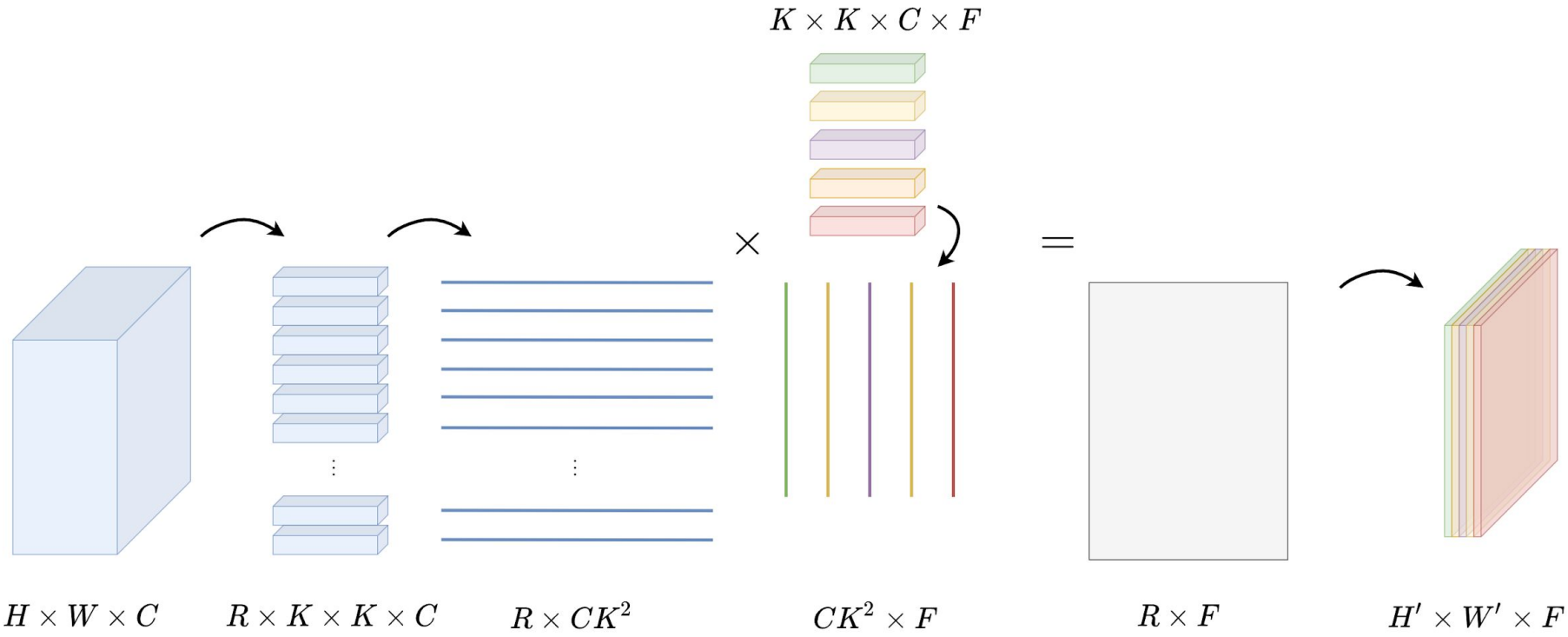
Fast Convolutions in GPU

Convolutions on volumes



$A^{[i-1]}$ $K^{[i]}$ $Z^{[i]}$

GEMM (im2col)


$K \times K \times C \times F$
$\times$
$=$
$H \times W \times C$
$R \times K \times K \times C$
$R \times CK^2$
$CK^2 \times F$
$R \times F$
$H' \times W' \times F$

FFT

- Convolution in spatial domain is equivalent to multiplication (elementwise) in the Fourier domain

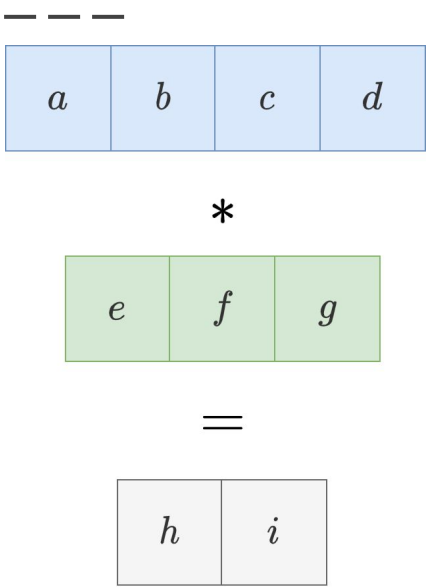
$$Z^{[i]} = \mathcal{F}^{-1}(\mathcal{F}(A^{[i-1]}) \odot \mathcal{F}(k^{[i]}))$$

- Can easily be extended to an arbitrary number of channels and filter
- Can be generalized to different padding strategies
- Speed-ups only for large kernels (usually bad for 3x3 or 5x5 kernels)

FFT

Vasilache, Nicolas, et al. "Fast convolutional nets with fbfft: A GPU performance evaluation." arXiv preprint arXiv:1412.7580 (2014).

Winograd - minimal example

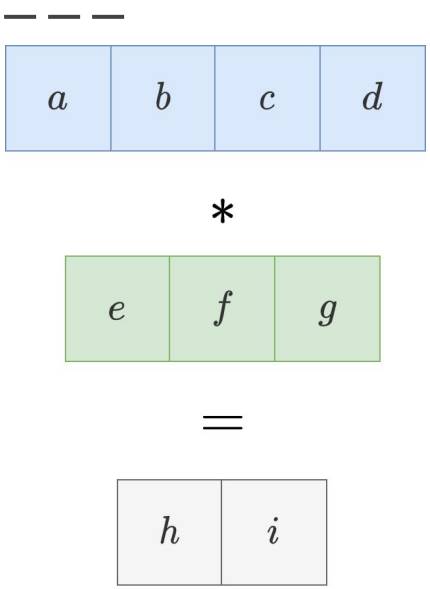


h = ae+bf+cg
i = be+cf+dg

Output: 6 mult + 4 add

Lavin, Andrew, and Scott Gray. "Fast algorithms for convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

Winograd - minimal example



h = j+k+l
i = k-l-m

j = (a-c)e
k = (b+c)(e+f+g)/2
l = (c-b)(e-f+g)/2
m = (b-d)g

Data: 4 add
Kernel: 2 mult + 3 add
Output: 4 mult + 4 add

Lavin, Andrew, and Scott Gray. "Fast algorithms for convolutional neural networks." Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition. 2016.

Other implementations

```
bool CudnnSupport::GetConvolveAlgorithms(
    std::vector<dnn::AlgorithmType>* out_algorithms) {
  out_algorithms->assign({
      // clang-format off
      CUDNN_CONVOLUTION_FWD_ALGO_IMPLICIT_GEMM,
      CUDNN_CONVOLUTION_FWD_ALGO_IMPLICIT_PRECOMP_GEMM,
      CUDNN_CONVOLUTION_FWD_ALGO_GEMM,
      CUDNN_CONVOLUTION_FWD_ALGO_DIRECT,
      CUDNN_CONVOLUTION_FWD_ALGO_FFT,
      CUDNN_CONVOLUTION_FWD_ALGO_FFT_TILING,
#if CUDNN_VERSION >= 5000
      CUDNN_CONVOLUTION_FWD_ALGO_WINOGRAD,
#endif
      // clang-format on
  });
```

End