# Notes on the series of articles on the topic Deconstructing Neural Networks using Circuits

compiled by D.Gueorguiev, 5/18/2024

## Introductory notes

Given the popularity of the Neural Networks for Computer Vision it is clear that the scientific community needs *mechanistic interpretability* for those neural network architectures.

Speculative claims about Neural Networks operations and explainability of the latter

**Features**: *Features* are the fundamental unit of Neural Networks. They correspond to *Directions*. By *Directions* the authors mean linear combination of neurons in a layer. One can think of this as a direction vector in the vector space of activations of neurons in a given layer. Often individual neurons will be discussed and analyzed but there are cases where analyzing combinations of neuronal output is the best way to understand the functioning of the neural net. This becomes even more important when we will be dealing with polysemantic neurons.

**Feature Visualization by Optimization**:

Neural networks are differentiable with respect to their inputs. This means that

//TODO: finish the section on Feature Visualization by Optimization

**Polysemantic Neurons**: Neural networks often contain polysemantic neurons that respond to multiple unrelated inputs. For example, [InceptionV1](https://distill.pub/2020/circuits/early-vision/) contains one neuron that responds to cat faces, fronts of cars and cat legs.

//TODO: finish the section on Polysemantic Neurons

## Discussion on *Early Vision* of InceptionV1

*Early Vision* denotes the first five layers of the neural net InceptionV1 which is shown below:

A screenshot of a computer

Description automatically generated

Figure 1: The InceptionV1 network with the first five layers shown in orange.

The first convolution layer is denoted as conv2d0 and is comprised of two kinds of features: color-contrast detectors and Gabor filters.

//TODO: finish the section on the EarlyVision part of InceptionV1

## Appendix

### Gabor filter

The Gabor filter is a linear filter used for texture analysis. This filter analyzes whether there is any frequency content in the image in specific directions in a localized region around the point or region of analysis.

In spatial domain a 2D Gabor filter is a Gaussian kernel function modulated by a sinusoidal plane wave similar to Gabor transform.

//TODO: finish the section on Gabor filter

### Gabor transform

The Gabor transform is a special case of the short-time Fourier transform. It is used to determine the sinusoidal frequency and phase content of local sections of a signal as it changes over time. The function to be transformed is first multiplied by a Gaussian which can be regarded as a window function, and the resulting function is then transformed with a Fourier transform to derive the time-frequency analysis. Using Gaussian as a window function means that the signal near the time being analyzed will have higher weight. The Gabor transform of a signal is defined as

The Gabor transform is [over-complete](https://en.wikipedia.org/wiki/Overcompleteness).

The Inverse Gabor transform

The inverse Gabor transform exists and because the forward Gabor transform is over-complete the original function can be recovered in a variety of ways. For example , the “unwindowing” approach can be used for any :

Alternatively, all of the time components can be combined:

Properties of the Gabor transform

Linearity

Shifting

Modulation

Power Integration

Energy sum

Power decay

if for

if for

Recovery

Example:

An input signal with 1 Hz frequency component when and 2 Hz frequency component when

Applying Gabor transform yields the following beautiful picture in frequency domain which identifies the occupied frequencies by the signal. Note that the shorter in time is the data sample of the input signal the fuzzier will be the boundaries of the frequency distribution due to the time-frequency uncertainty governing the transform.

A graph of a waveform

Description automatically generated with medium confidence

Discrete Gabor transform

A discrete version of Gabor representation

## References

[Zoom In: An Introduction to Circuits, Chris Olah et al, OpenAI, 2020](https://distill.pub/2020/circuits/zoom-in)

[An Overview of Early Vision in InceptionV1, Chris Olah et al, OpenAI, 2020](https://distill.pub/2020/circuits/early-vision/)

[Curve Detectors, Nick Cammarata et al, OpenAI, 2020](https://distill.pub/2020/circuits/curve-detectors/)

[Naturally Occurring Equivariance in Neural Networks, Chris Olah et al, OpenAI, 2020](https://distill.pub/2020/circuits/equivariance/)

[High-Low Frequency Detectors, Ludwig Schubert et al, OpenAI, 2021](https://distill.pub/2020/circuits/frequency-edges/)

[Curve Circuits, Nick Cammarata et al, OpenAI, 2021](https://distill.pub/2020/circuits/curve-circuits/)

[Visualizing Weights, Chelsea Voss et al, OpenAI, 2021](https://distill.pub/2020/circuits/visualizing-weights/)

[Branch Specialization, Chelsea Voss et al, OpenAI, 2021](https://distill.pub/2020/circuits/branch-specialization/)

[Weight Banding, Michael Petrov et al, OpenAI, 2021](https://distill.pub/2020/circuits/weight-banding/)

[Visualizing Higher Level Features of a Deep Network, Dumitru Erhan et al, Universite de Montreal, 2009](https://github.com/dimitarpg13/deep_learning_for_image_processing/blob/main/literature/articles/interpretability/Visualizing_Higher-Layer_Features_of_a_Deep_Network_Erhan_2009.pdf)

[Gabor Filter, Wikipedia](https://en.wikipedia.org/wiki/Gabor_filter)

[Gabor Transform, Wikipedia](https://en.wikipedia.org/wiki/Gabor_transform)