Feature Visualization

Johannes Vogt

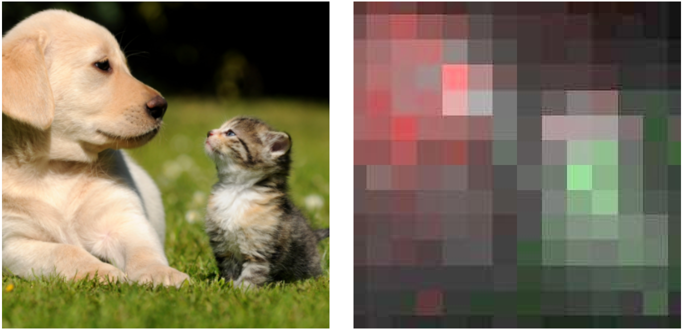
21.06.2018

Feature Visualization

How neural networks build up their understanding of images

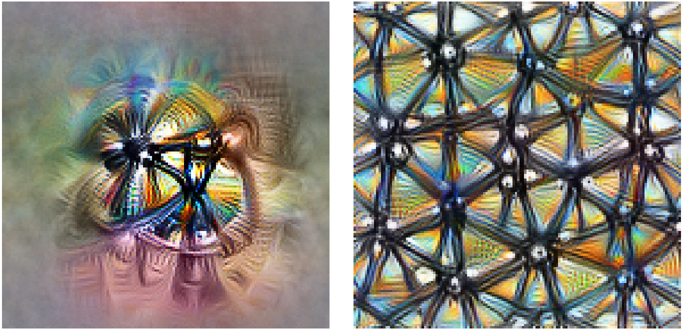
Olah, et al., "Feature Visualization", Distill, 2017. [3]

Attribution



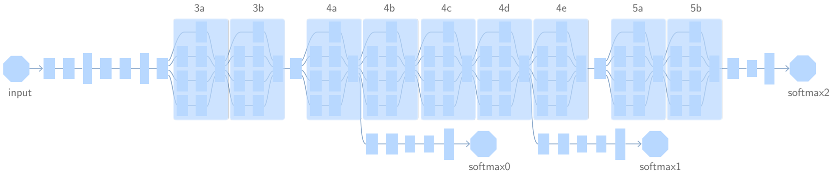
What parts are responsible for the activation?

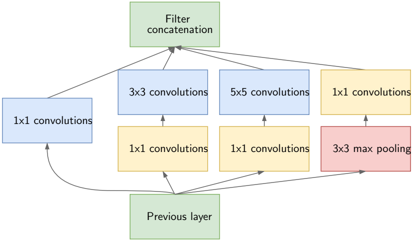
Feature Visualization



What kind of image does the neuron/layer activate for?

Network Architecture (GoogLeNet)



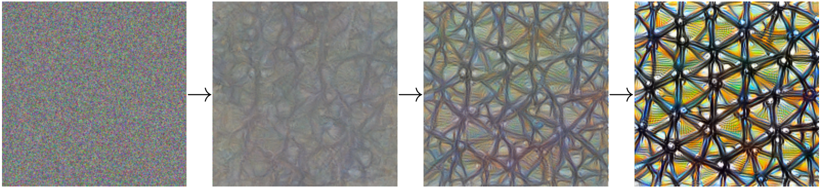


Inception module [5]

Feature Visualization by Optimization

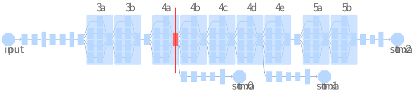
- Freeze parameters of trained model
- Update image X using the gradients $\nabla_X \mathcal{L}$ to optimize an objective $\mathcal{L}$:

$$X \mathrel{+}= \lambda \nabla_X \mathcal{L}$$

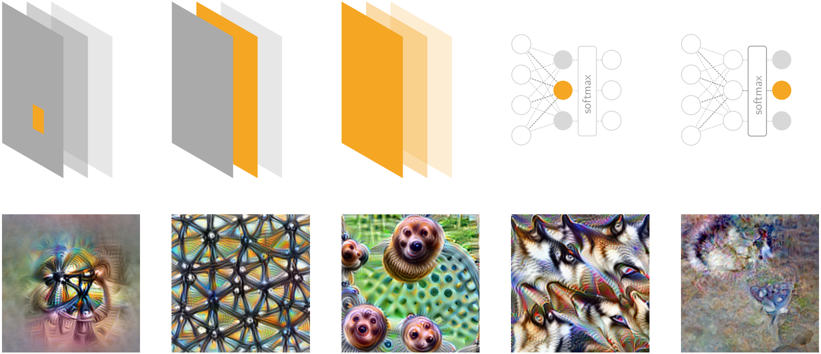


Step 0 → Step 4 → Step 48 → Step 2048

layer *mixed4a*
channel *11*



What to visualize

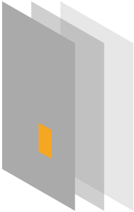


Neuron **Channel** **Layer** **Class Logits** **Class Probability**

There are a lot of different parts of a network that we might want to interpret, for which we need different objective functions.

Optimization Objective

For a neuron



The idea is to maximize the activation z_{lcxy} of a given neuron:

$$\mathcal{L}_{lcxy} = z_{lcxy}$$

l: layer index
c: channel index
x,y: spatial position

For a negative optimization, the output of the pre-activation $\hat{z}_{lcxy}$ can be taken for the gradient to be non-zero in case of a ReLU.

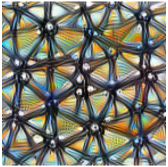
Optimization Objective

For a channel

Extending the neuron objective to a channel, following objective can be used:

$$\mathcal{L}_{lc} = \frac{1}{w * h} \sum_{x,y} z_{lcxy}$$

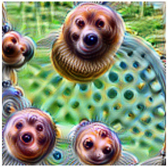
w: layer width
h: layer height



This objective leads to a repetitive occurance of the pattern.

Optimization Objective

For a layer



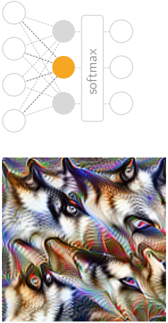
A layer captures many patterns, so it's difficult to find a good objective.

One approach to this is the Deep Dream objective which aims to maximize what the layer deems *'interesting'*:

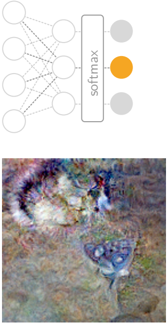
$$\mathcal{L}_l = \|z_l\|^2$$

Optimization Objective

For a class label



pre-softmax[k]



post-softmax[k]

For the class labels, there are two possibilities:
Optimizing the *pre-softmax* activation (evidence of the class k):

$$\mathcal{L}_L = \hat{z}_{Lk}$$

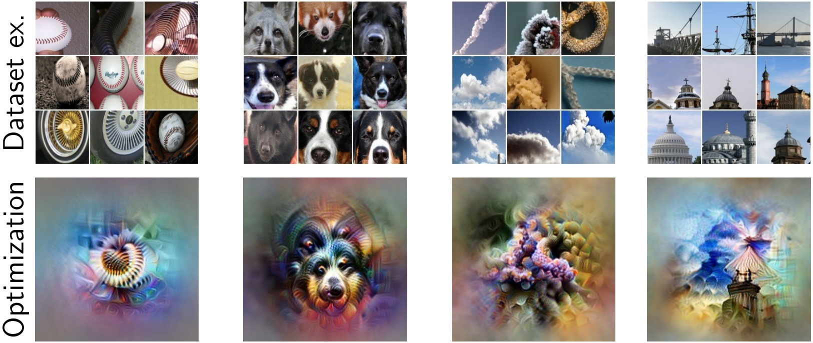
k: class index
L: output layer

Or optimizing the *post-softmax* activation (probability of the class k, given the evidence):

$$\mathcal{L}_L = z_{Lk}$$

Why Optimization?

Dataset examples vs. optimization



Baseball — or stripes? *mixed4a, Unit 6*

Animal faces — or snouts? *mixed4a, Unit 240*

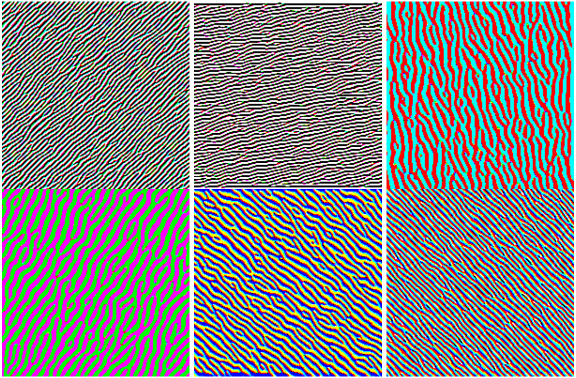
Clouds — or fluffiness? *mixed4a, Unit 453*

Buildings — or sky? *mixed4a, Unit 492*

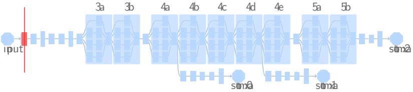
Although dataset examples give a good intuition about what a neuron activates for, it might not show the full picture.

Examples

Edges

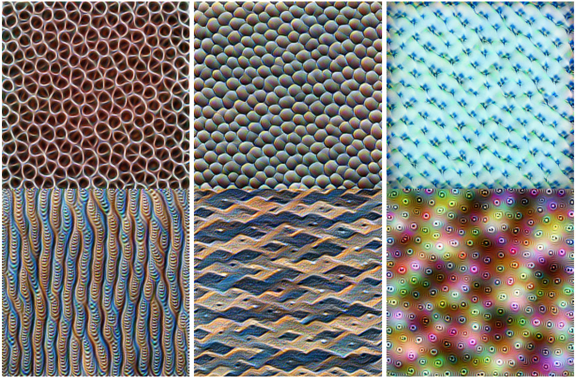


layer *conv2d0*

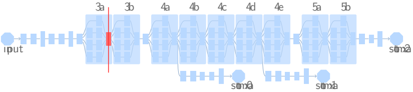


Examples

Textures

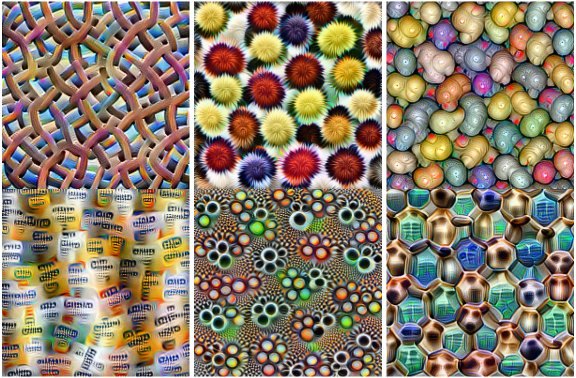


layer *mixed3a*

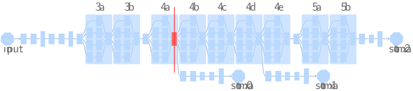


Examples

Patterns

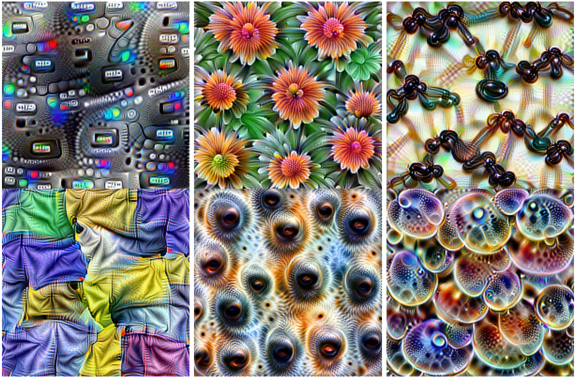


layer *mixed4a*

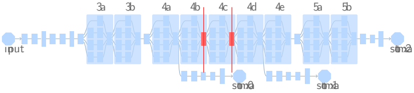


Examples

Parts

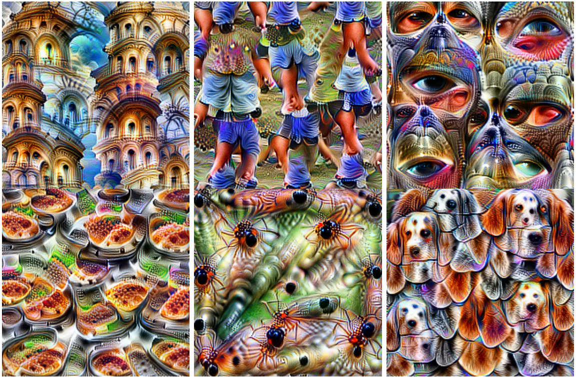


layer *mixed4b and mixed4c*

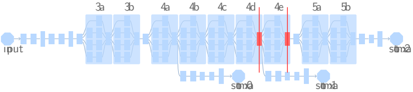


Examples

Objects

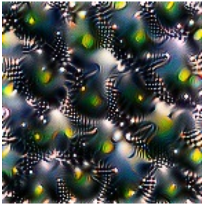


layer *mixed4d and mixed4e*

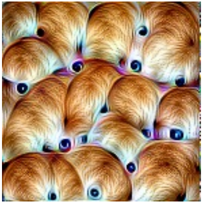


Diversity

Motivation

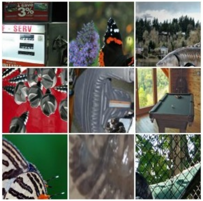


Negative optimized



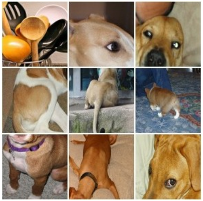
Positive optimized

Minimum activation examples



Slightly negative activation examples

Slightly positive activation examples



Maximum activation examples

Does the optimized image show the ”facets” of activation?

Diversity

Approaches

- "Intra-class" diversity, optimizing for the cluster centroids
- Use diverse dataset examples as starting point
- Generative model → pick diverse samples
- Optimize with diversity term

Diversity

Optimization with diversity term

Gram matrix G:

$$G_{i,j} = layer_n[:, :, i] \cdot layer_n[:, :, j]$$

Diversity term as negative pairwise cosine similarity:

$$C_{diversity} = -\sum_{a} \sum_{a \neq b} \frac{vec(G_a) \cdot vec(G_b)}{\|vec(G_a)\| \, \|vec(G_b)\|}$$

Optimize jointly for optimization objective and diversity term:

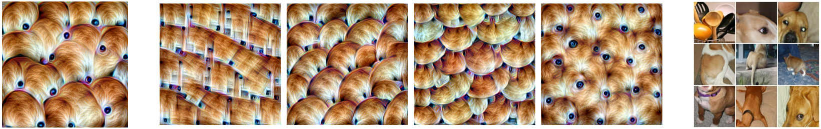
$$\mathcal{L} \mathrel{+}= C_{diversity}$$

Diversity Examples

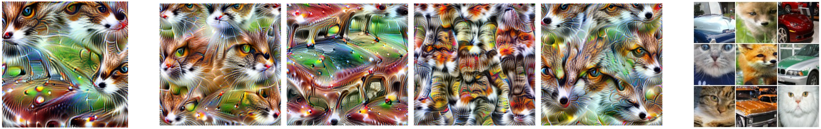
Simple Optimization

Optimization with diversity

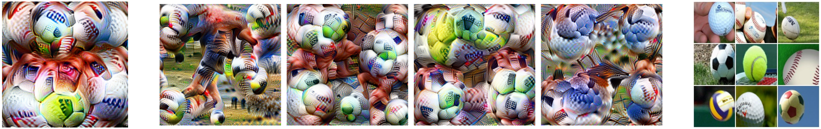
Dataset examples



Layer mixed4a, Unit 143



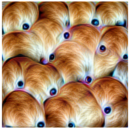
Layer mixed4e, Unit 55



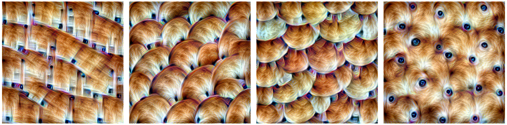
Layer mixed5a, Unit 9

Diversity Examples

Simple Optimization

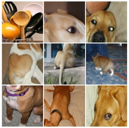


Optimization with diversity

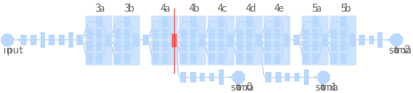


Layer mixed4a, Unit 143

Dataset examples

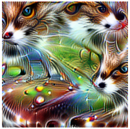


In contrast to the simple optimization which suggests the channel activation on top of a dogs head through the curved fur and eyes, the diverse examples show that the channel also reacts to just the brown fur texture.

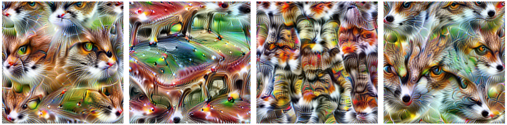


Diversity Examples

Simple Optimization

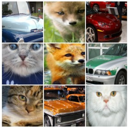


Optimization with diversity

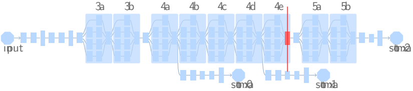


Layer mixed4e, Unit 55

Dataset examples

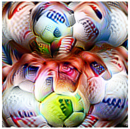


This channel activating for cats, foxes and cars shows that for a better understanding we may also need to examine combinations of neurons.

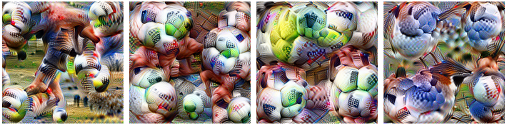


Diversity Examples

Simple Optimization

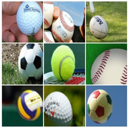


Optimization with diversity



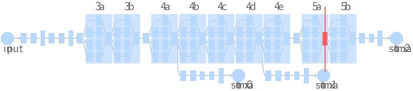
Layer mixed5a, Unit 9

Dataset examples



The expectation of optimization with diversity might be images with different kinds of balls, like in the dataset examples.

This example shows that the diversity term can also be misleading, because it pushes images to be different from each other, introducing features that are not relevant to the objective.



Interaction between Neurons

If single channels are the basis directions of the activation space of a layer, this can be extended to the activation direction

$$d_l = \begin{bmatrix} z_{l0} & z_{l1} & \dots & z_{lC} \end{bmatrix}^T$$

The objective for a given layer l and direction d_l is

$$\mathcal{L}_l = \frac{1}{whC} \sum_{x,y,c} z_{lcxy} d_{lc}$$

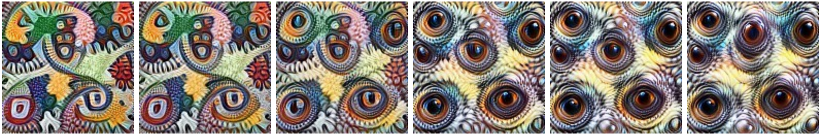
Random directions have been found to seem as interpretable as basis directions. [6]
Basic directions have been found to be interpretable more often than random directions. [1]

Interaction between Neurons

Interpolation

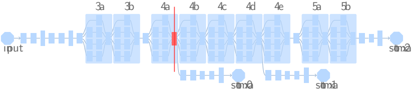
Given two channels, their base directions can be interpolated by

$$d_{l,t} = (1-t)d_{lc_1} + td_{lc_2}$$



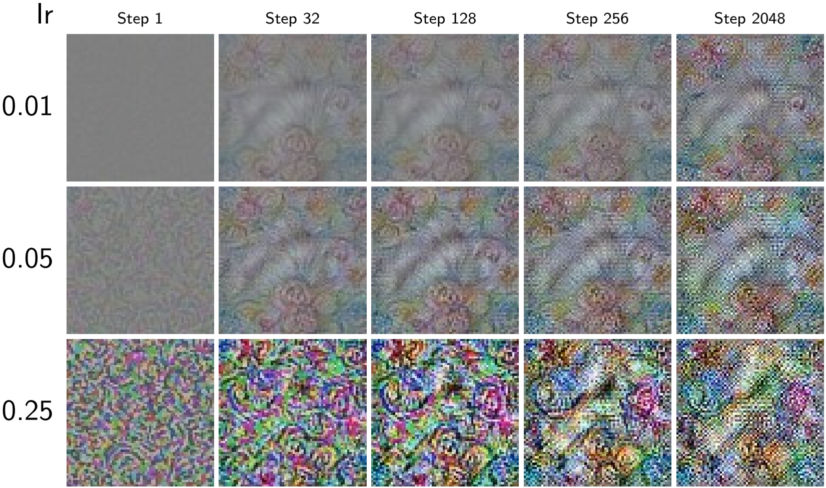
channel 476

channel 460



The Enemy of Feature Visualization

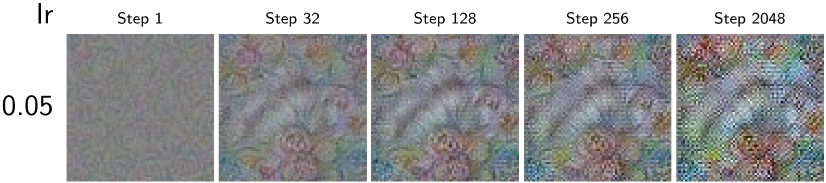
How to make visualizations look good



Feature visualization without regularization

The Enemy of Feature Visualization

How to make visualizations look good



Feature visualization without regularization

When optimized without regularization, high frequency patterns emerge to activate the neuron.
While it is not fully understood, the patterns seem to be caused by strided convolutions and pooling operations, creating a high frequency grid pattern in the gradient.

The Spectrum of Regularization

To make generated images more natural looking, a range of regularization techniques can be applied:

Weak Regularization

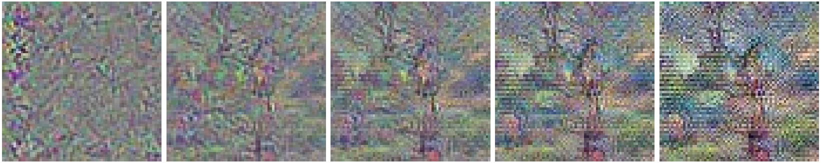
- Frequency penalization
- Transformation robustness

Strong Regularization

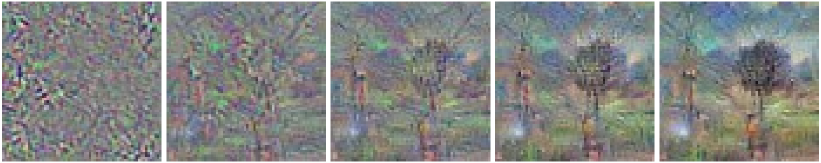
- Learned priors
- Dataset examples

Regularization

Frequency penalization



No frequency penalization



L_1: -0.05, total variation: -0.25, blur: -0.1

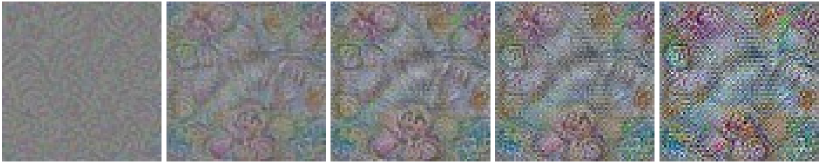
Total variance penalizes variance between neighboring pixels.

Blurring implicitely penalizes high-frequency noise.

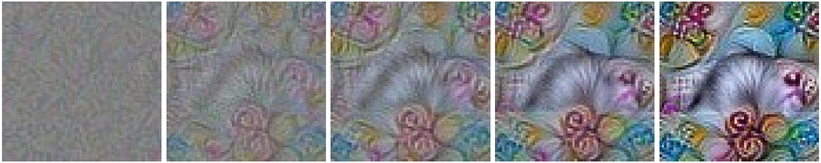
These approaches also discourage legitimate high-frequency patterns like edges.

Regularization

Transformation robustness



No transformation robustness



Jitter: $1px$, Rotate: 5°, Scale: $1.1x$

Using transformation robustness as regularizer leads to images that still activate the target even if they're slightly transformed.

Regularization

Learned prior

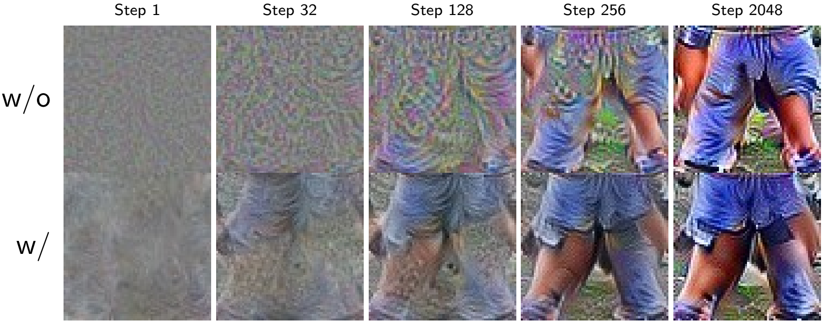
Another step to creating natural looking images is to train a model of the real data and try to enforce it.

This approach can produce photorealistic visualizations [2], but it's not necessarily clear what came from the visualization objective and what came from the prior.

Preconditioning and Parameterization

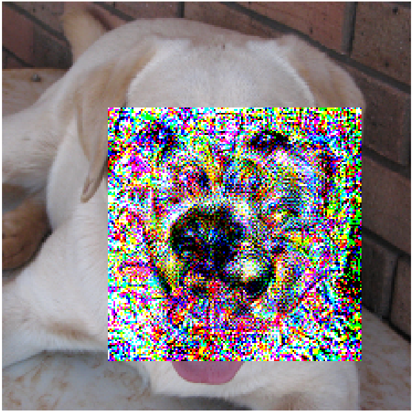
Another way to improve visual quality is to "precondition" the image.
The preconditioner chosen here is a transformation g to the fourier basis and a color decorrelation h using a Cholesky decomposition from the training set.
Instead of using the image X directly, the transformed image $\tilde{X} = h(g(X))$ is used as input for the network.

Preconditioning and Parameterization

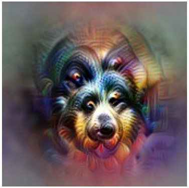


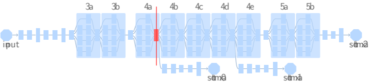
Resulting visualizations **in decorrelated space** seem to have better visual quality and develop faster.
(lr $= 0.05$; with transformation robustness)

Optimization initialized with example



Visualization of mixed4a, channel 240, initialized with a picture fitting to the neuron. The snout stays the same during optimization while the non related parts get more high frequency patterns. (no regularization used)





Conclusion

- Basics of feature visualization and optimization objectives
- How do we get diversity in feature visualization?
- Improving the visual quality of generated examples

→ There's a lot of room for more research.

Follow-up paper: Olah, et al., "The Building Blocks of Interpretability", Distill, 2018.[4]

[1] D. Bau, B. Zhou, A. Khosla, A. Oliva, and A. Torralba. Network dissection: Quantifying interpretability of deep visual representations. 2017.

[2] A. Nguyen, A. Dosovitskiy, T. Yosinski, Jason band Brox, and J. Clune. Synthesizing the preferred inputs for neurons in neural networks via deep generator networks. *NIPS 29*, 2016.

[3] C. Olah, A. Mordvintsev, and L. Schubert. Feature visualization. *Distill*, 2017. https://distill.pub/2017/feature-visualization.

[4] C. Olah, A. Satyanarayan, I. Johnson, S. Carter, L. Schubert, K. Ye, and A. Mordvintsev. The building blocks of interpretability. *Distill*, 2018. https://distill.pub/2018/building-blocks.

[5] C. Szegedy, W. Liu, Y. Jia, P. Sermanet, S. Reed, D. Anguelov, D. Erhan, V. Vanhoucke, and A. Rabinovich. Going deeper with convolutions. *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 1–9, 2015. arXiv:1409.4842.

[6] C. Szegedy, W. Zaremba, I. Sutskever, J. Bruna, D. Erhan, I. Goodfellow, and R. Fergus. Intriguing properties of neural networks. *arXiv preprint*, 2013. arXiv:1312.6199.

Thank you!