

# Feature visualization

C Olah, A Mordvintsev, L Schubert  
Distill

Presenter: Jason Yao

# Overview

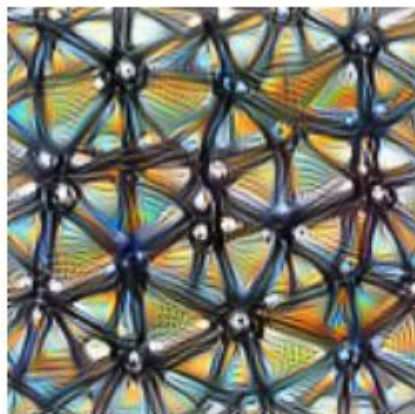
- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion

# Motivation

- Interpreting neural networks via feature visualization
- Examine the major issues and explore common approaches to solving them
- Introduce a few tricks for exploring variation in what neurons react to, how they interact, and how to improve the optimization process

# Background

- Two major threads in neural network interpretability
  - Feature visualization
  - Attribution



**Feature visualization** answers questions about what a network — or parts of a network — are looking for by generating examples.



**Attribution**<sup>1</sup> studies what part of an example is responsible for the network activating a particular way.

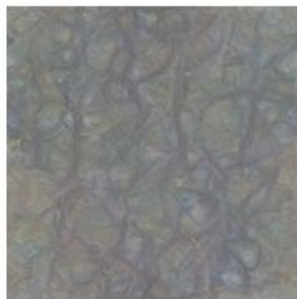
# Feature Visualization by Optimization

- Optimization: use derivative to iteratively tweak the input towards a certain goal — whether activate a neuron or have a specific final output

Starting from random noise, we optimize an image to activate a particular neuron (layer mixed4a, unit 11).



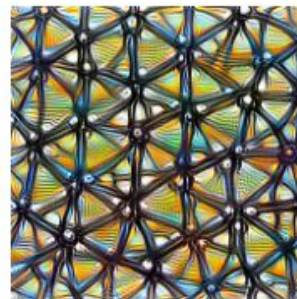
Step 0



Step 4



Step 48



Step 2048

# Optimization Objectives

- Objective: the part of network that we want to understand its pattern

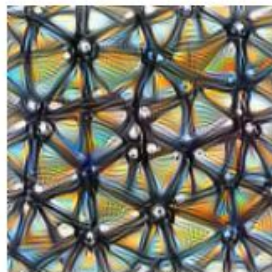
Different **optimization objectives** show what different parts of a network are looking for.

**n** layer index  
**x,y** spatial position  
**z** channel index  
**k** class index



**Neuron**

$\text{layer}_n[x,y,z]$



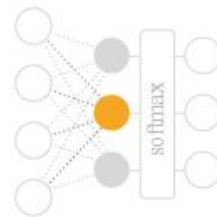
**Channel**

$\text{layer}_n[:, :, z]$



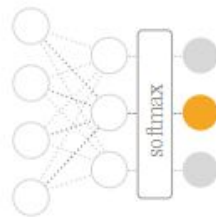
**Layer/DeepDream**

$\text{layer}_n[:, :, :]^2$



**Class Logits**

$\text{pre\_softmax}[k]$



**Class Probability**

$\text{softmax}[k]$

# Optimization Objectives

- For individual feature: search for neuron, or an entire channel.
  - Channel is more suitable because naturally each channel represent one feature in CNN
  - This paper mostly create image from Channel
- Layer: usually use DeepDream objective
- Class Logits: the evidence for each class, produce better visual quality than class probability
- Class probability: easy to tweak when make the alternatives unlikely
  
- More to explore!!!

# Why visualize by optimization?

- Why generating a example is better than use the example already have in dataset?
  - it separates the things causing behavior from things that merely correlate with the causes

**Dataset Examples** show us what neurons respond to in practice



**Optimization** isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



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*



# Why visualize by optimization?

- Pros

- Separates correlated example, find most related example
- Flexibility
  - We can choose any target we want, without worry about how a dataset example match the target
  - We can choose iteration step and visualize how features evolve as the network trains

- Cons

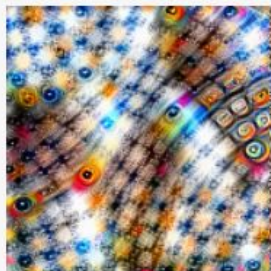
- Only show one facet examples
- See the section **“The Enemy of Feature Visualization”**

# Overview

- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion

# Diversity

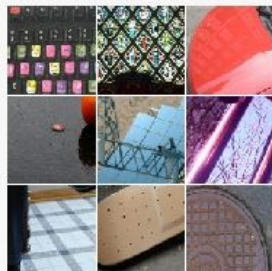
- Generally optimization may only show us one “facet” of features
- Comparison with looking through dataset examples:



**Negative** optimized



**Minimum** activation examples



Slightly negative activation examples

0



Slightly positive activation examples



**Maximum** activation examples



**Positive** optimized

Layer mixed 4a, unit 6

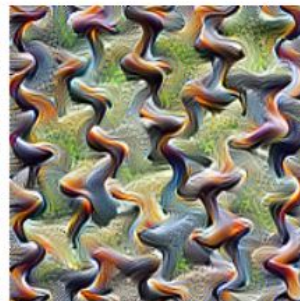
REPRODUCE IN A  NOTEBOOK

# Achieving Diversity with Optimization

- Simple way to achieve **diversity**: adding diversity term to the objective function
  - One possibility is to penalize the cosine similarity of different examples
  - Another is to use ideas from style transfer to force the feature to be displayed in different styles
- For lower level neurons, a diversity term can reveal the different facets a feature represents



Simple Optimization



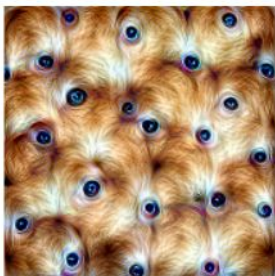
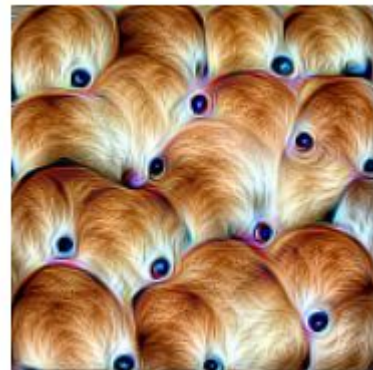
Optimization with diversity reveals four different, curvy facets. *Layer mixed4a, Unit 97*



Dataset examples

# Achieving Diversity with Optimization

- Diversity generated examples help us to verify our previous hypothesis about the feature of the neuron
  - From one example, one may infer that this neuron activate on the top of dog heads
  - But looking at the other example, we will see this neuron actually activate on the texture and color similar to dog fur



Dataset examples



# Achieving Diversity with Optimization

- For high level neuron, diverse example even show different objects that stimulate the pattern



Simple Optimization



Optimization with diversity reveals multiple types of balls. Layer *mixed5a*, Unit 9



Dataset examples

- Shortcoming of the simple method
  - The pressure of creating different examples may cause **unrelated artifacts**
  - Make examples different in an **unnatural** way

# Achieving Diversity with Optimization

- Shortcoming of the simple method
  - The pressure of creating different examples may cause unrelated artifacts
  - Make examples different in an unnatural way
  - Mixture of ideas

Cat and Fox



Simple Optimization



Optimization with diversity show cats, foxes, but also cars. *Layer mixed4e, Unit 55*



Dataset examples

- Conclusion: neurons are not necessarily the right semantic units for understanding neural nets

# Overview

- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion



# Interaction between Neurons

- In real model, combination of neurons works together to represent images
- Geometrically thinking the process of combination:
  - Define all possible combination of activation vectors as the activation space
  - Each neuron is a basis vector of the space
  - Combination is also just a vector in the space
  -
- What is the relationship between direction of vector and interpretability?
  - Szegedy *et al.* found that random directions seem just as meaningful as the directions of the basis vectors
  - Bau, Zhou *et al.* found the directions of the basis vectors to be interpretable more often than random directions.
  - The paper's experiment and conclusion: random directions often seem interpretable, but at a lower rate than basis directions

# Direction and interpretability

Dataset examples and optimized examples of **random directions** in activation space. The directions shown here were hand-picked for interpretability.

REPRODUCE IN A  
CO NOTEBOOK



*mixed3a, random direction*



*mixed4c, random direction*



*mixed4d, random direction*



*mixed5a, random direction*

# Interpolate between Neuron

- similar to interpolating in the latent space of generative models

REPRODUCE IN A  
CO NOTEBOOK



Layer 4c, Unit 369

Layer 4a, Unit 476

# Interaction Have More To Explore

- How to select meaningful directions
- Whether there even exist particularly meaningful directions.
- How directions interact — for example, interpolation can show us how a small number of directions interact
- People have no clue about all this questions

# Overview

- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion



# The Enemy of Feature Visualization

- A common result of optimization without constraint:
  - An image full of noise and nonsensical high-frequency patterns that the network responds strongly to

Even if you carefully tune learning rate, you'll get noise.

Optimization results are enlarged to show detail and artifacts.

REPRODUCE IN A  
 NOTEBOOK



Learning Rate (0.05)



Step 1



Step 32



Step 128



Step 256

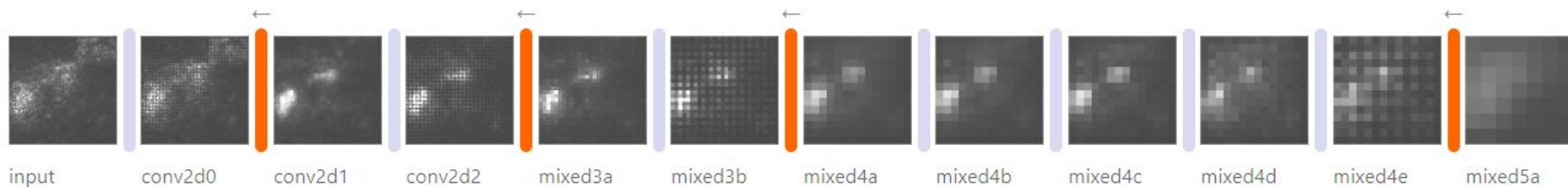


Step 2048

# The Enemy of Feature Visualization

- A common result of optimization without constraint:
  - People don't understand where this high frequency pattern come from,
  - One guess is strided convolutions and pooling operations, create high-frequency pattern in gradient

Each **strided convolution or pooling** creates checkerboard patterns in the gradient magnitudes when we backprop through it.



- Conclusion: while we are enjoying the benefit of freedom from optimization, we still need some constraint on optimization for our needs

# The Spectrum of Regularization







- Most notable papers on feature visualization have a point on how to add regularization
- On one extreme, if we don't regularize at all, we end up with adversarial examples.
- On the opposite end, we look through our dataset and run into limitations discuss before.
- In the middle, we have three main families of regularization options



# The Spectrum of Regularization

**Weak Regularization** avoids misleading correlations, but is less connected to real use.

**Strong Regularization** gives more realistic examples at risk of misleading correlations.

		Unregularized	Frequency Penalization	Transformation Robustness	Learned Prior	Dataset Examples
	<b>Erhan, et al., 2009</b> [3] Introduced core idea. Minimal regularization.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<b>Szegedy, et al., 2013</b> [11] Adversarial examples. Visualizes with dataset examples.	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>
	<b>Mahendran &amp; Vedaldi, 2015</b> [7] Introduces total variation regularizer. Reconstructs input from representation.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<b>Nguyen, et al., 2015</b> [14] Explores counterexamples. Introduces image blurring.	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
	<b>Mordvintsev, et al., 2015</b> [4] Introduced jitter & multi-scale. Explored GMM priors for classes.	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>
	<b>Øygard, et al., 2015</b> [15] Introduces gradient blurring. (Also uses jitter.)	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

# The Spectrum of Regularization



**Tyka, et al., 2016** [16]

Regularizes with bilateral filters.  
(Also uses jitter.)



**Mordvintsev, et al., 2016** [17]

Normalizes gradient frequencies.  
(Also uses jitter.)



**Nguyen, et al., 2016** [18]

Paramaterizes images with GAN generator.



**Nguyen, et al., 2016** [10]

Uses denoising autoencoder prior to make  
a generative model.



# Three Families of Regularization

- Frequency penalization: penalize high frequency noise
  - Discourage legitimate high-frequency features like edges
  - Can be slightly improved by using a bilateral filter, which preserves edges, instead of blurring
- 

Frequency penalization  
directly targets high  
frequency noise

REPRODUCE IN A  
 NOTEBOOK



$L_1$  (-0.05)



Total Variation (-0.25)



Blur (-1)



Step 1



Step 32



Step 128



Step 256



Step 2048

# Three Families of Regularization

- Transformation robustness: find image still activate the optimization target even under some transformation
  -

Stochastically transforming the image before applying the optimization step suppresses noise

REPRODUCE IN A  
 NOTEBOOK



Jitter (1px)

Rotate (5°)

Scale (1.1×)



Step 1



Step 32



Step 128



Step 256



Step 2048

# Three Families of Regularization

- Learned priors: learn a model of real data and keep examples reasonable
  - Produces the most photorealistic visualizations
  - Unclear what came from the model being visualized and what came from the prior
- One approach is to learn a generator that maps points in a latent space to examples of your data, such as a GAN or VAE, and optimize within that latent space
- An alternative approach is to learn a prior that gives you access to the gradient of probability; which allow jointly optimize for the prior along with your objective
- One recovers a generative model of the data conditioned on that particular class
- Wei *et al.* approximate a generative model prior, at least for the color distribution, by penalizing distance between patches of the output and the nearest patches retrieved from a database of image patches collected from the training data.

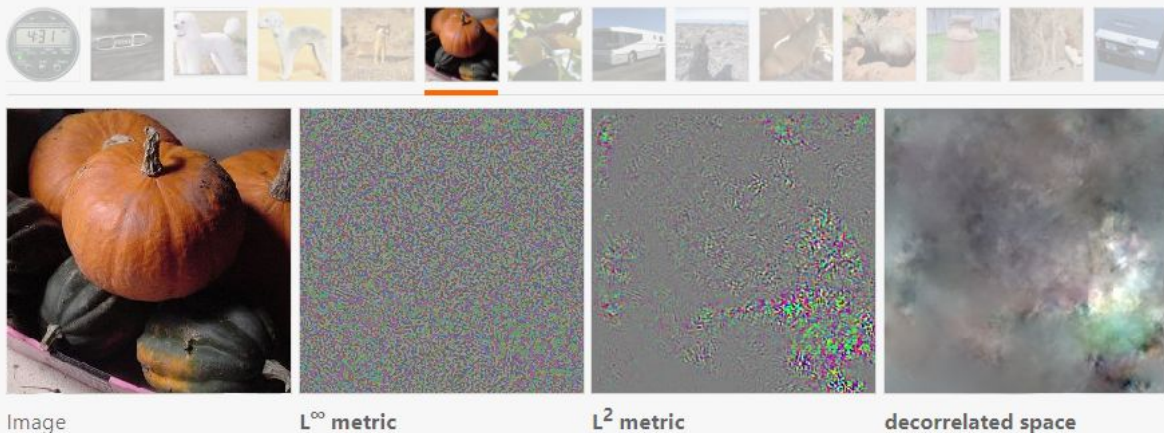
# Overview

- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion

# Preconditioning and Parameterization

- Preconditioning: Transforming the gradient.
  - Doing the gradient descent in a way that is most straight, or steepest, in another parameterization of the space or under a different notion of distance
  - Using the right preconditioner can make an optimization problem radically easier
- What is a good preconditioner: decorrelated and whitened the original image
- Usage of different distance metric:

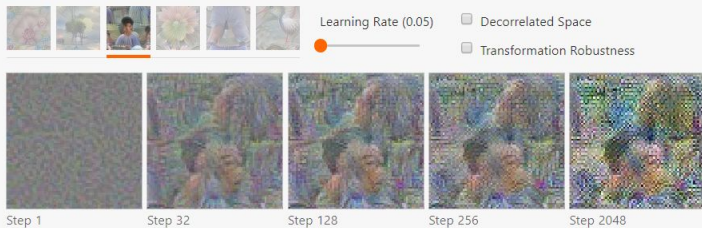
Three directions of steepest descent under different notions of distance



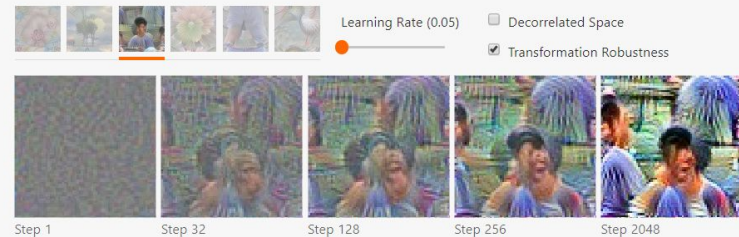


# Preconditioning and Parameterization

Combining the preconditioning and transformation robustness improves quality even further



Combining the preconditioning and transformation robustness improves quality even further



Combining the preconditioning and transformation robustness improves quality even further



Combining the preconditioning and transformation robustness improves quality even further





# Preconditioning and Parameterization

- Doubts
  - Is the preconditioner merely accelerating descent, which show us the result we will see in normal gradient descent?
  - Or is it also regularizing, changing which local minima we get attracted to?
- People have no concrete conclusion

# Overview

- Motivation and Background
- Diversity
- Interaction between Neurons
- The Enemy of Feature Visualization
- Preconditioning and Parameterization
- Conclusion

# Conclusion

- In neural networks interpretability, feature visualization stands out as one of the most promising and developed research directions
- Feature visualization will never give a completely satisfactory understanding
- There remains still a lot of important work to be done in improving feature visualization, Notable ones include
  - understanding neuron interaction
  - finding which units are most meaningful for understanding neural net activations
  - giving a holistic view of the facets of a feature