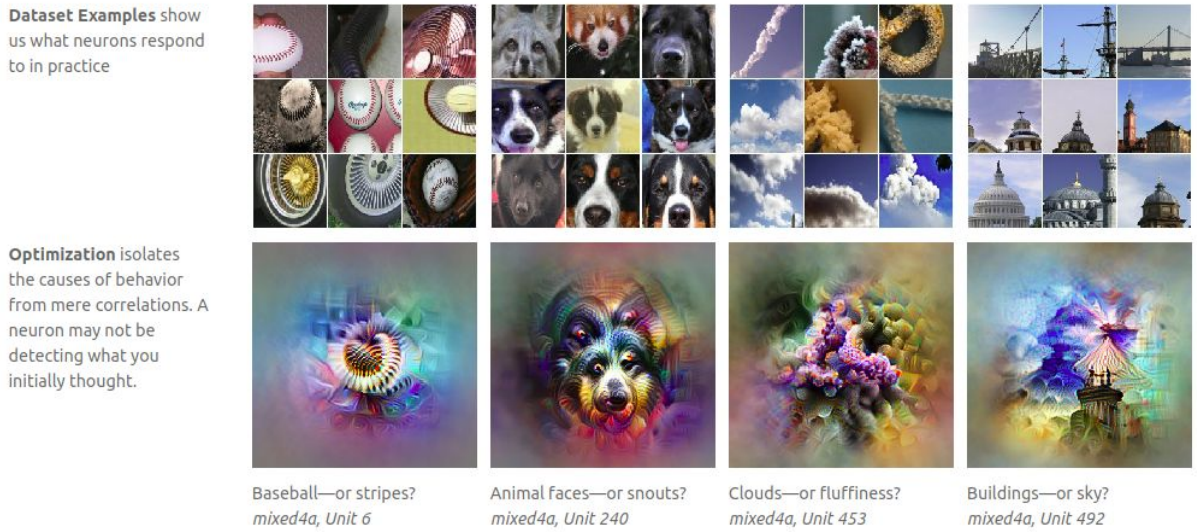
Visualizing the Image Features

(Notes while reading literature)

Aim : To increase the probability softmax gives to a class, it's easier to make the other outputs less likely rather then working towards making that class more likely.

Optimization can give us an example input that causes the desired behavior — but why bother with that? Couldn't we just look through the dataset for examples that cause the desired behavior?
It's because optimization approach enables us to know the features which actually excite a neuron. In dataset looking approach we may be misguided by things which mearley just correlate to the main cause.
Have a look at the following cases :



One generally adds a optimization term to the diversity objective to have an overall view and not make overfitting assumptions from one result.

To visualize features, we can't just go about optimizing the image. It generally leads to an image full of noise and nonsensical high frequency components. Mainly these high frequency components come from strided convolutions and pooling operations.

Regularization
To overcome this high frequency noise in the optimized image we do regularization.

With image classification, the neural network learns a set of visual abstractions and thus images are the most natural symbols to represent them. Were we working with audio, the more natural symbols would most likely be audio clips.

Style transfer
https://arxiv.org/pdf/1508.06576.pdf

The system uses neural representations to separate and recombine content and style of arbitrary images, providing a neural algorithm for the creation of artistic images
On top of the original CNN representations we built a new feature space that captures the style of an input image. The style representation computes correlations between the different features in different layers of the CNN.

****The paper states that the content and style of an image are separable using CNN.**
That is , we can manipulate both content and style independently to create perceptually meaningful images
To obtain the style of an image, we use a feature space originally designed to capture the texture of the image. This feature space is built on top of filter responses of each layer. It consists of the correlations between the different filter responses over the spatial extent of the feature maps
*Another important observation is that if u combine the style features extracted from the higher layers, the visual perception of the new image is better. This is because style extraction layers higher up in the model focus on local style features.

Though it is not possible to completely disentangle the style and content of an image, but while creating an image which matches content and style of 2 separate images, we have 2 separate loss terms for style and content and thus can smoothly regulate the emphasis on each of them.
The style representations simply compute the correlation between different neurons in the network. Extracting correlations between neurons is a biologically plausible computation that is, for example, implemented by so-called complex cells in the primary visual system.

Object Recognition network neglects those variations in the image which don't change the identity of the object. An another network built upon these features can

easily factorize upon these parameters. Our style transfer network is built over the outputs from the different layers of the network.

Finally to create an image which has style and content from 2 separate images, you do optimization over a white image. You try to optimize it so that the new images content and style map the original.

Method
To visualise the image information that is encoded at different layers of the hierarchy) we perform gradient descent on a white noise image to find another image that matches the feature responses of the original image.
We take the squared loss over the features of the new image and the original image.

You multiply a relative weight to the loss terms for style and the loss term for the content. And then do the grad descent.