BST 261: Data Science II
Lecture 14
Advanced Topics:
Generative Deep Learning

Heather Mattie
Harvard T.H. Chan School of Public Health
Spring 2 2019

Generative Deep Learning

- Generative deep learning methods using RNNs and CNNs have been around for awhile, but have recently been getting a lot of attention
  - 2002: Douglas Eck applied LSTM to music generation - he is now at Google Brain and started a research group called Magenta to use deep learning to create engaging music
  - 2013: Alex Graves applies recurrent mixture density networks to generate human-like handwriting
  - Many more, a few that we'll talk about today

- Many researchers in this field have said that "generating sequential data is the closest computers get to **dreaming**"

Text Generation with LSTM

Generative RNNs for Text

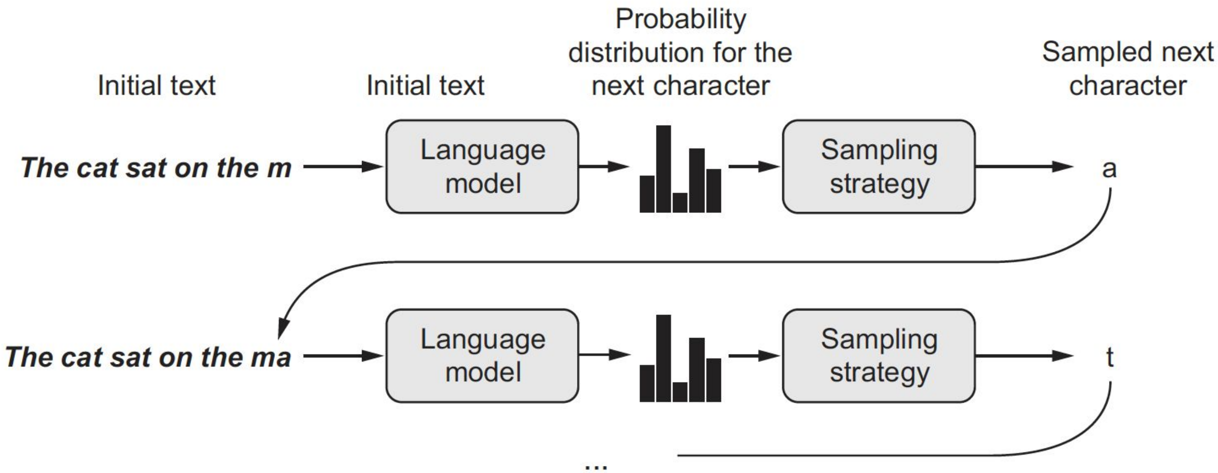
- RNNs have been successfully used for
  - Music generation
  - Dialogue generation
  - Image generation
  - Speech synthesis
  - Molecule design

- Main idea for text generation: train a model to predict the next token or next few tokens in a sequence

- **Language Model:** any network that can model the probability of the next token given the previous ones
  - Captures the latent space of language - its statistical structure
  - Once it is trained, you can sample from it to generate new sequences

Generative RNNs for Text

- Process
  - 1. Feed it an initial string of text (called conditioning data)
  - 2. Ask the model to generate the next character or word
  - 3. Add the generated output back to the input data
  - 4. Repeat many times



- Can generate sequences of arbitrary length
- The generated sequences will reflect the structure of the training data

Generative RNNs for Text

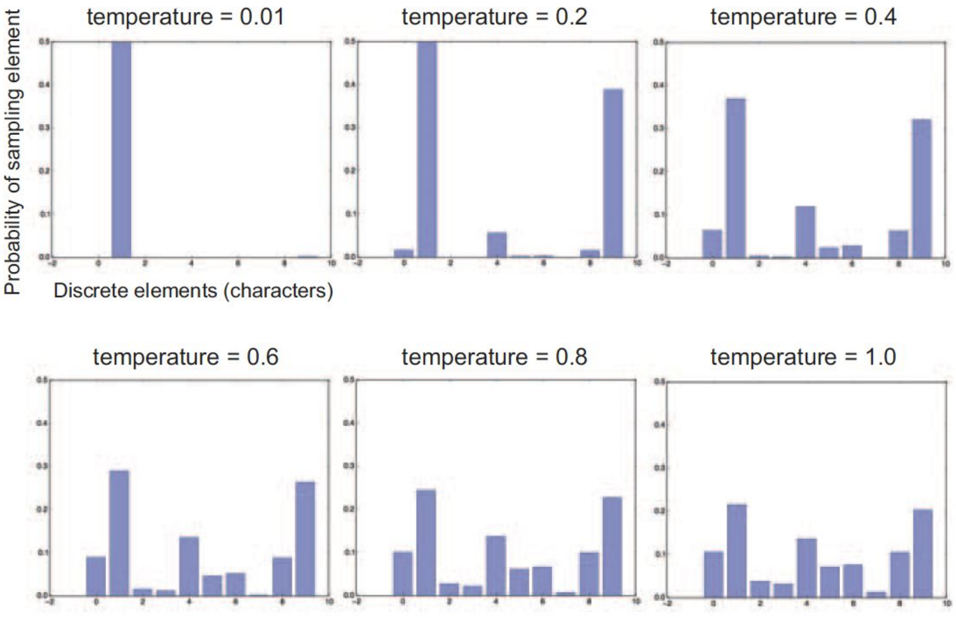
- We can choose the next character or word in different ways - some are better than others

- A naive approach is greedy sampling - always choosing the most likely next character or word
  - This results in repetitive, predictable strings and not very coherent language

- Better approach is stochastic sampling
  - Sample next characters or words with specific probability from a probability distribution
  - Allows even unlikely characters or words to be sampled at times, generating more interesting and creative sentences
  - Doesn’t offer a way of controlling the randomness in the sampling process

Generative RNNs for Text

- New parameter to tune: **softmax temperature**
  - Controls the amount of randomness
  - More randomness = similar probability for every character or word and results in more interesting output
  - Less randomness = higher probability for just one or a few characters or words and results in repetitive output
  - Can change the amount of randomness via the temperature value
    - Higher temperature = more randomness
    - Lower temperature = more deterministic



Character-level LSTM Generation

- Need a lot of data to train from

- Can choose from many sources, referred to as a **corpus**
  - Wikipedia
  - The Lord of the Rings
  - The writings of Nietzsche translated into English

- Let’s see an example with the writings of Nietzsche as our corpus

Character-level LSTM Generation

- Training the language model and sampling from it:
- Given a trained model and a seed text snippet, do the following repeatedly

  - 1. Draw from the model a probability distribution for the next character, given the generated text available so far
  - 2. Reweight the distribution to a certain temperature
  - 3. Sample the next character at random according to the reweighted distribution
  - 4. Add the new character at the end of the available text

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 20 with temperature = 0.2:

“new faculty, and the jubilation reached its climax when kant and such a man in the same time the spirit of the surely and the such the such as a man is the sunligh and subject the present to the superiority of the special pain the most man and strange the subjection of the special conscience the special and nature and such men the subjection of the special men, the most surely the subjection of the special intellect of the subjection of the same things and”

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 20 with temperature = 0.5:

“new faculty, and the jubilation reached its climax when kant in the eterned and such man as it’s also become himself the condition of the experience of off the basis the superiory and the special morty of the strength, in the langus, as which the same time life and "even who discless the mankind, with a subject and fact all you have to be the stand and lave no comes a troveration of the man and surely the conscience the superiority, and when one must be w "

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 20 with temperature = 1.0:

“new faculty, and the jubilation reached its climax when kant, as a periliting of manner to all definites and transpects it it so hicable and ont him artiar resull too such as if ever the proping to makes as cnecience. to been juden, all every could coldiciousnike hother aw passife, the plies like which might thiod was account, indifferent germin, that everythery certain destrution, intellect into the deteriorablen origin of moralian, and a lessority o"

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 60 (model has now fully converged) with temperature = 0.2:

“cheerfulness, friendliness and kindness of a heart are the sense of the spirit is a man with the sense of the sense of the world of the self-end and self-concerning the subjection of the strengthorixes–the subjection of the subjection of the subjection of the self-concerning the feelings in the superiority in the subjection of the subjection of the spirit isn’t to be a man of the sense of the subjection and said to the strength of the sense of the"

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 60 with temperature = 0.5:

“cheerfulness, friendliness and kindness of a heart are the part of the soul who have been the art of the philosophers, and which the one won’t say, which is it the higher the and with religion of the frences. the life of the spirit among the most continuess of the strengther of the sense the conscience of men of precisely before enough presumption, and can mankind, and something the conceptions, the subjection of the sense and suffering and the"

Character-level LSTM Generation

- Random seed:
  - “new faculty, and the jubilation reached its climax when kant"
- Output at epoch 60 with temperature = 1.0:

“cheerfulness, friendliness and kindness of a heart are spiritual by the ciuture for the entalled is, he astraged, or errors to our you idstood–and it needs, to think by spars to whole the amvives of the newoatly, prefectly raals! it was name, for example but voludd atu-especity"–or rank onee, or even all "solett increessic of the world and implussional tragedy experience, transf, or insiderar,–must hast if desires of the strubction is be stronges"

Character-level LSTM Generation

- Low temperature results in repetitive and predictable text, but local structure is highly realistic

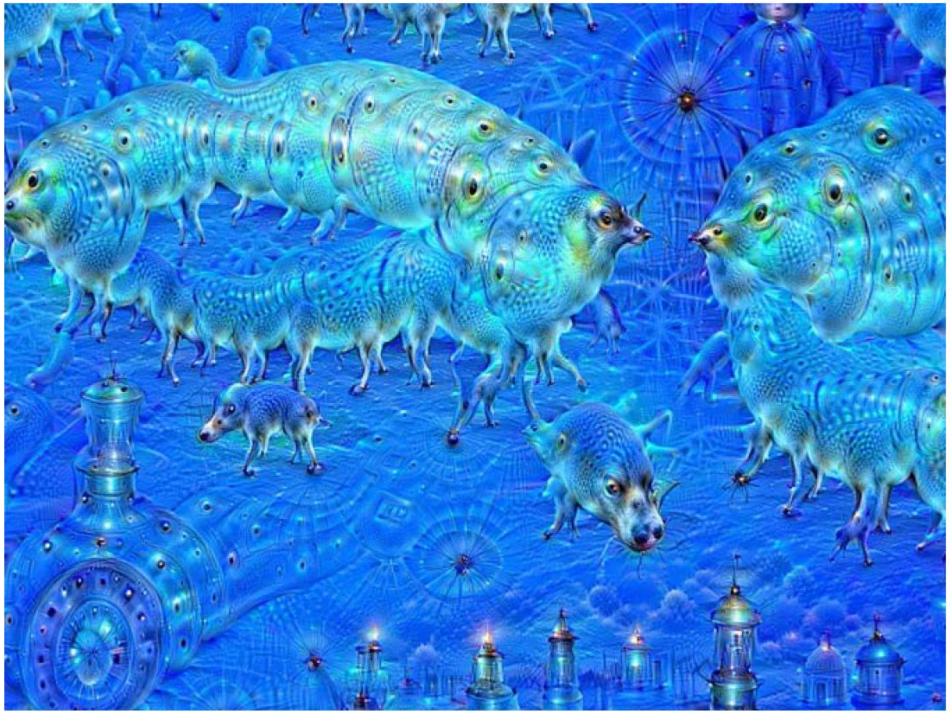
- Higher temperatures result in more interesting, surprising and creative text, sometimes creating new words - but the local structure breaks down and most words are strings of random characters

- Generally, somewhere in the middle (around 0.5) creates the most interesting text - but this depends on the corpus and the human reading the results

DeepDream

DeepDream

- DeepDream is an artistic image-modification technique that uses the representations learned by convolutional neural networks
  - Released by Google in the summer of 2015
  - Trained on ImageNet

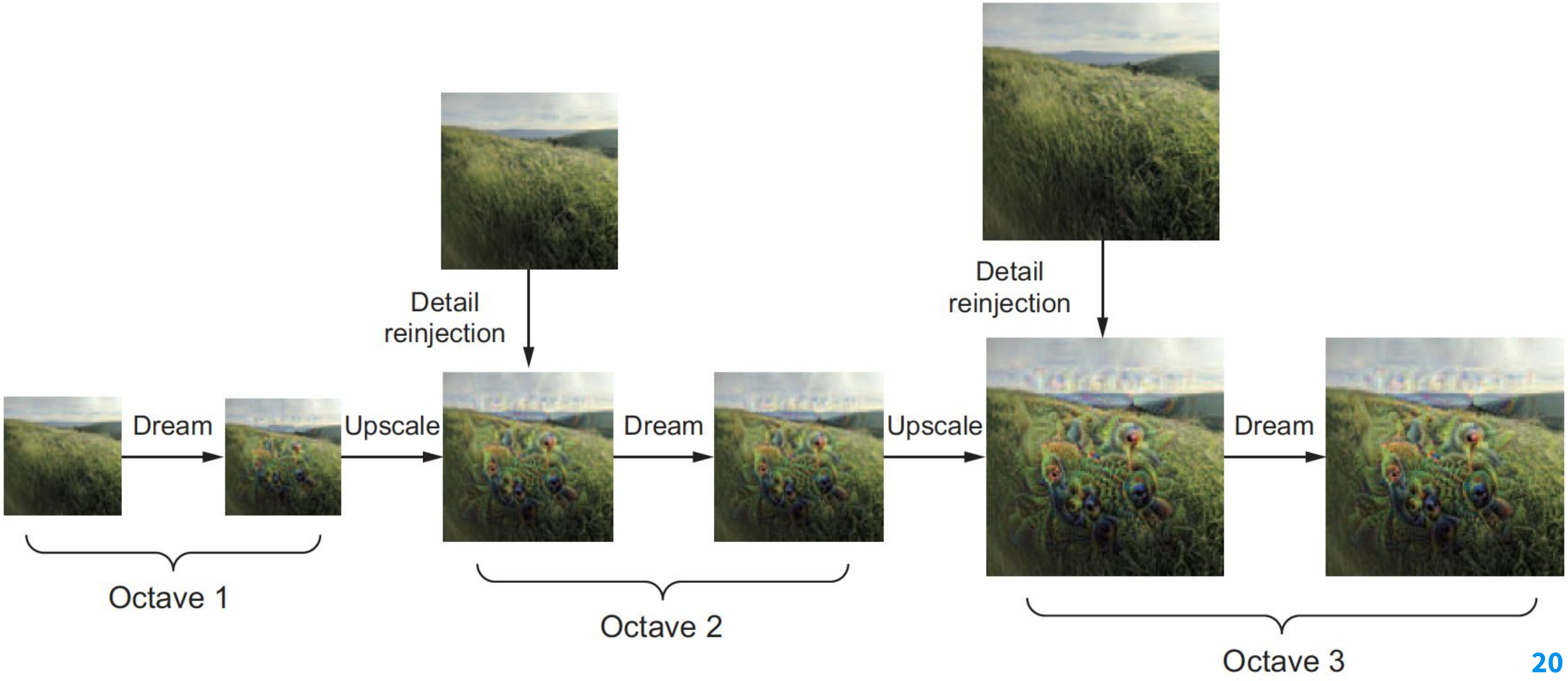


DeepDream

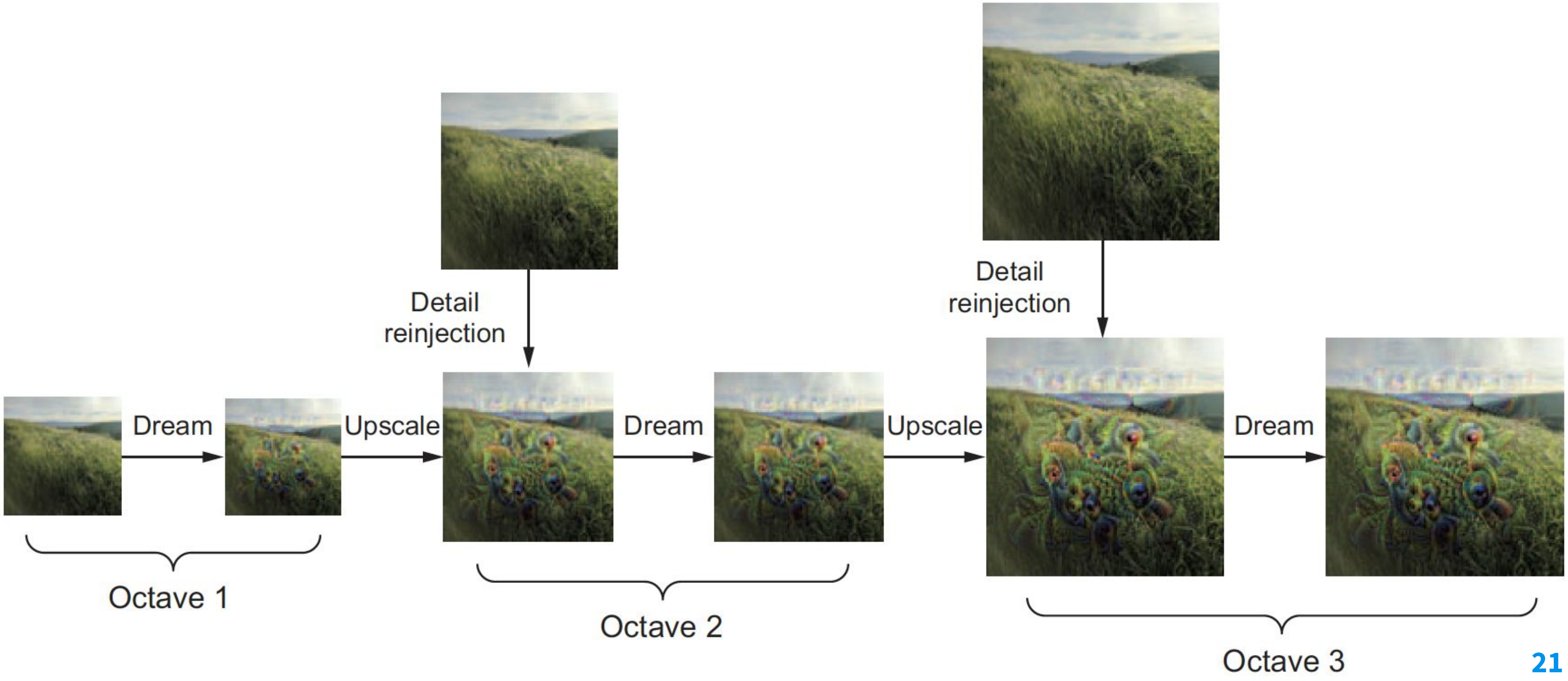
- The DeepDream algorithm is almost identical to the convnet filter-visualization technique introduced in lecture 7, consisting of running a convnet in reverse: doing gradient ascent on the input to the convnet in order to maximize the activation of a specific filter in an upper layer of the convnet

- DeepDream uses this same idea, with a few simple differences
  - You try to maximize the activation of entire layers rather than that of a specific filter, thus mixing together visualizations of large numbers of features at once
  - You start not from blank, slightly noisy input, but rather from an existing image—thus the resulting effects latch on to preexisting visual patterns, distorting elements of the image in a somewhat artistic fashion
  - The input images are processed at different scales (called octaves), which improves the quality of the visualizations

- First, define a list of scales (also called **octaves**) at which to process the images
- Each successive scale is larger than the previous one by a factor of 1.4 (it's 40% larger): you start by processing a small image and then increasingly scale it up
- For each successive scale, from the smallest to the largest, you run gradient ascent to maximize the loss you previously defined, at that scale. After each gradient ascent run, you upscale the resulting image by 40%



- To avoid losing a lot of image detail after each successive scale-up (resulting in increasingly blurry or pixelated images), you can use a simple trick: after each scaleup, you'll reinject the lost details back into the image, which is possible because you know what the original image should look like at the larger scale



DeepDream

DeepDream

- Layers that are lower in the network contain more-local, less-abstract representations and lead to dream patterns that look more geometric

- Layers that are higher up lead to more-recognizable visual patterns based on the most common objects found in ImageNet, such as dog eyes, bird feathers, and so on

Neural Style Transfer

Neural Style Transfer

- Another major development in deep-learning-driven image modification
- Introduced by [Leon Gatys et al.] in 2015
- Variations have been introduced, some even as smartphone apps
- Neural style transfer consists of **applying the style** of a reference image to a target image **while conserving the content** of the target image

Neural Style Transfer

- Style
  - textures, colors, and visual patterns in the image, at various spatial scales
- Content
  - higher-level macrostructure of the image

- Like other neural nets, we need to define and minimize a loss function
  - We want to conserve the content of the original image, while adopting the style of the reference image
  - If we can mathematically define content and style, our loss function would be:

Loss = distance(style(ref_image) - style(generated_image)) +
distance(content(original_image) - content(generated_image))

Neural Style Transfer

- Deep CNNs offer a way to define this loss function mathematically
- Recall:
  - Activations from earlier layers in a network contain local information
  - about the image
  - Activations from higher layers contain increasingly global,
  - abstract information

- Content loss
  - The content of an image is more global and abstract and should be captured by the representations of later layers
  - Loss is the L2 norm between the activations of an upper layer in a pretrained convnet, computed over the target image, and the activations of the same layer computed over the generated image
  - Ensures the generated image will look similar to the original target image

Neural Style Transfer

- Style loss
  - Uses multiple layers of the CNN
  - Try to capture the appearance of the style reference image at all spatial scales extracted by the convnet, not just a single scale
  - Use the Gram matrix of a layer's activations: the inner product of the feature maps of a given layer
  - This inner product can be understood as representing a map of the correlations between the layer's features
  - These feature correlations capture the statistics of the patterns of a particular spatial scale, which empirically correspond to the appearance of the textures found at this scale
  - Aims to preserve similar internal correlations within the activations of different layers, across the style-reference image and the generated image
  - Guarantees that the textures found at different spatial scales look similar across the style-reference image and the generated image

Neural Style Transfer

- Summary
  - Preserve content by maintaining similar high-level layer activations between the target content image and the generated image. The convnet should “see” both the target image and the generated image as containing the same things
  - Preserve style by maintaining similar correlations within activations for both low-level layers and high-level layers. Feature correlations capture textures: the generated image and the style-reference image should share the same textures at different spatial scales

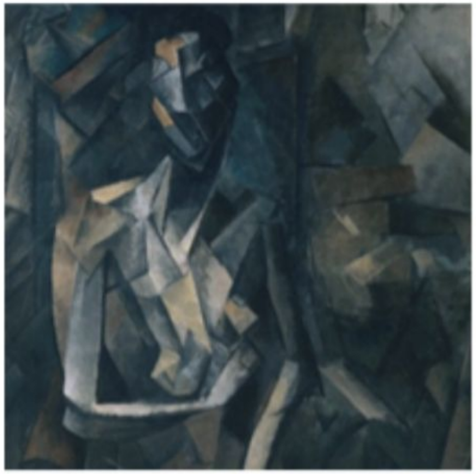
Neural Style Transfer

Awesome blog post

Content C

+

Style S



=

Generated Image G

Picasso Dancer

$$J(G) = \alpha J_{\text{content}}(C, G) + \beta J_{\text{style}}(S, G)$$

Neural Style Transfer

[Image Style Transfer Using CNNs]()

“ A Neural Algorithm of Artistic Style that can separate and recombine the image content and style of natural images. The algorithm allows us to produce new images of high perceptual quality that combine the content of an arbitrary photograph with the appearance of numerous well-known artworks”