deepsense.io

BIG DATA SCIENCE

How to have fun with deep learning

Jakub Czakon

Deep Learning

“Real learning, attentive, real learning, deep learning, is playful and frustrating and joyful and discouraging and exciting and sociable and private all the time, which is what makes it great.”

Eleanor Duckworth

Deep Learning

*"Real learning, attentive, real learning, deep learning, is playful and **frustrating** and joyful and **discouraging** and exciting and sociable and private all the time, which is what makes it great."*

Eleanor Duckworth

Deep Learning

*“Real learning, attentive, real learning, deep learning, is playful and **frustrating** and joyful and **discouraging** and exciting and sociable and private all the time, which is what makes it great.”*

Eleanor Duckworth

“Deep learning doesn't shine.”

Marie von Ebner-Eschenbach

Deep Learning

*"Real learning, attentive, real learning, deep learning, is playful and frustrating and **joyful** and discouraging and **exciting** and sociable and private all the time, which is what makes it **great**."*

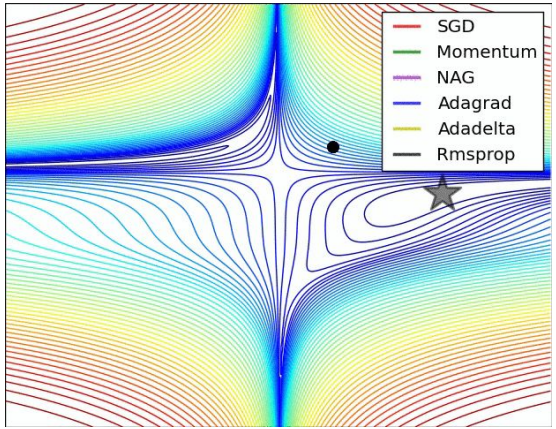
Eleanor Duckworth

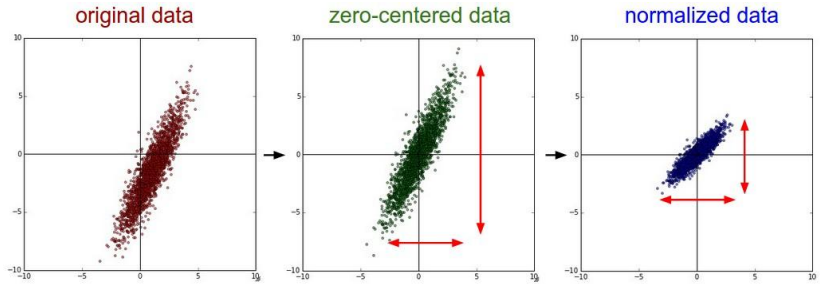
Deep Learning

Deep Learning

- SGD Algorithms
- Machine Learning
- Linear Algebra
- Calculus
- Image processing
- GPU computing
- Hyperparameter Optimization

$$w(k+1)=\left(1-\frac{1}{2\left(1+\left\|\frac{\partial\hat{y}(k)}{\partial w(k)}\right\|^2\right)}\right)w(k)+\frac{1}{2\left(1+\left\|\frac{\partial\hat{y}(k)}{\partial w(k)}\right\|^2\right)}e(k)\frac{\partial\hat{y}(k)}{\partial w(k)}$$





Deep Learning

- Go top level
- Use pretrained models
- Build on existing solutions
- Be creative

Pick Your Spots!

JOEL GRUS

is sort of a famous author

Fizz Buzz in Tensorflow

interviewer: Welcome, can I get you coffee or anything? Do you need a break?

me: No, I've probably had too much coffee already!

interviewer: Great, great. And are you OK with writing code on the whiteboard?

me: It's the only way I code!

interviewer: ...

me: That was a joke.

interviewer: OK, so are you familiar with "fizz buzz"?

me: ...

interviewer: Is that a yes or a no?

me: It's more of a "I can't believe you're asking me that."

interviewer: OK, so I need you to print the numbers from 1 to 100, except that if the number is divisible by 3 print "fizz", if it's divisible by 5 print "buzz", and if it's divisible by 15 print "fizzbuzz".

me: I'm familiar with it.

interviewer: Great, we find that candidates who can't get this right don't do well here.

me: ...

interviewer: Here's a marker and an eraser.

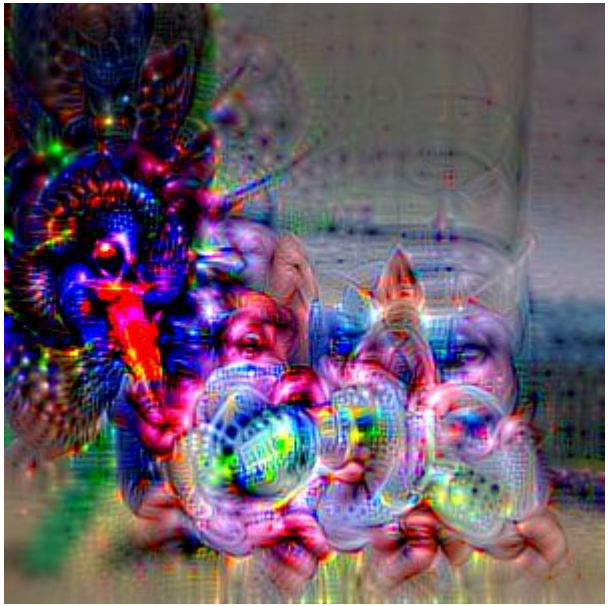
me: [thinks for a couple of minutes]

interviewer: Do you need help getting started?

me: No, no, I'm good. So let's start with some standard imports:

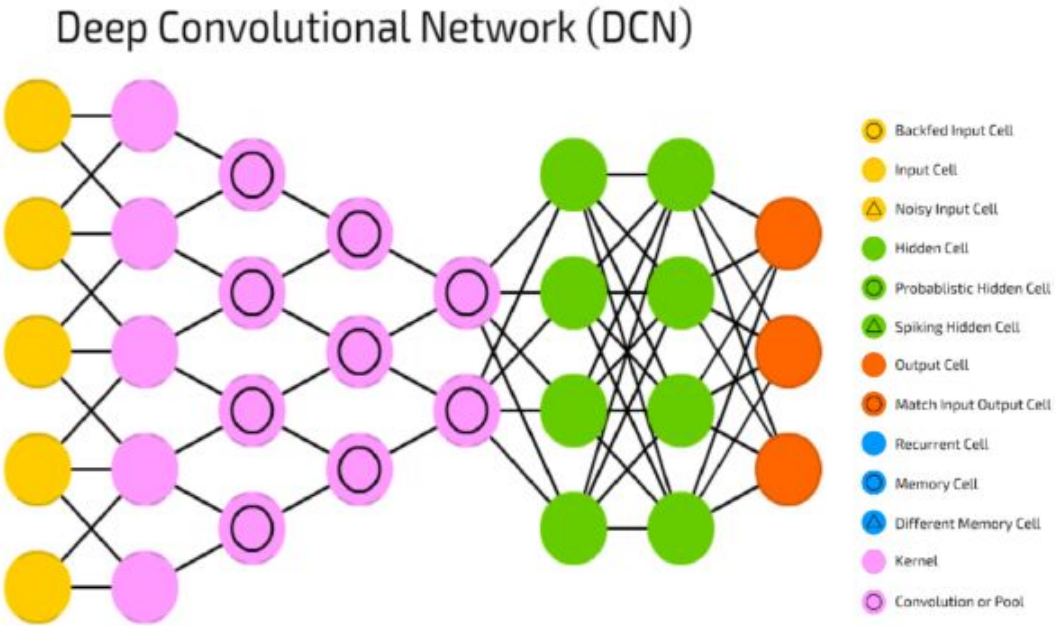
```
import numpy as np
import tensorflow as tf
```

Deep Dream



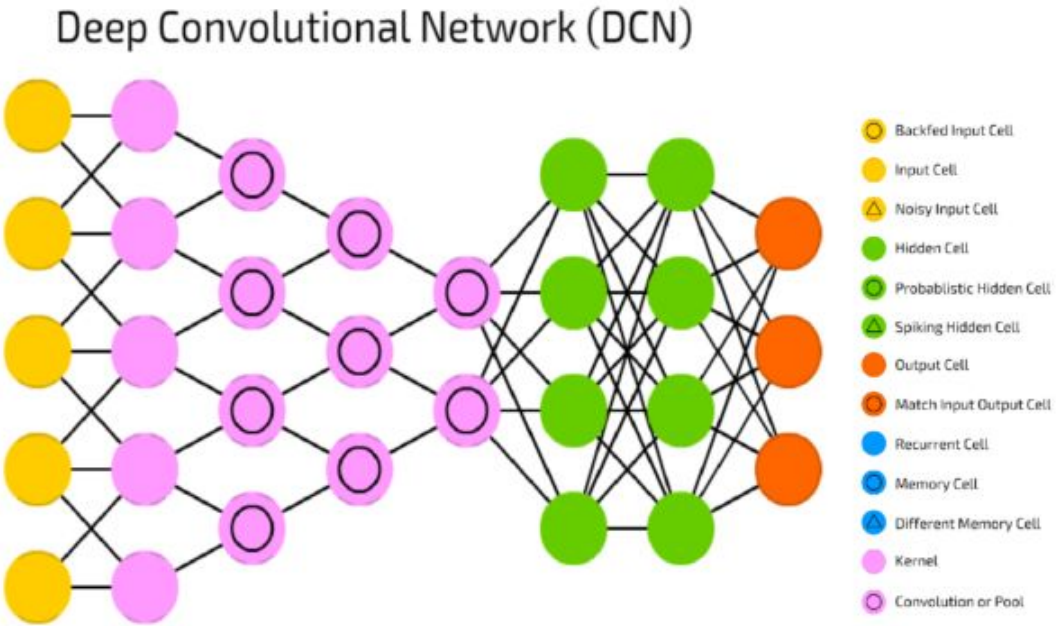
https://research.googleblog.com/2015/06/inceptionism-going-deeper-into-neural.html

Deep Dream



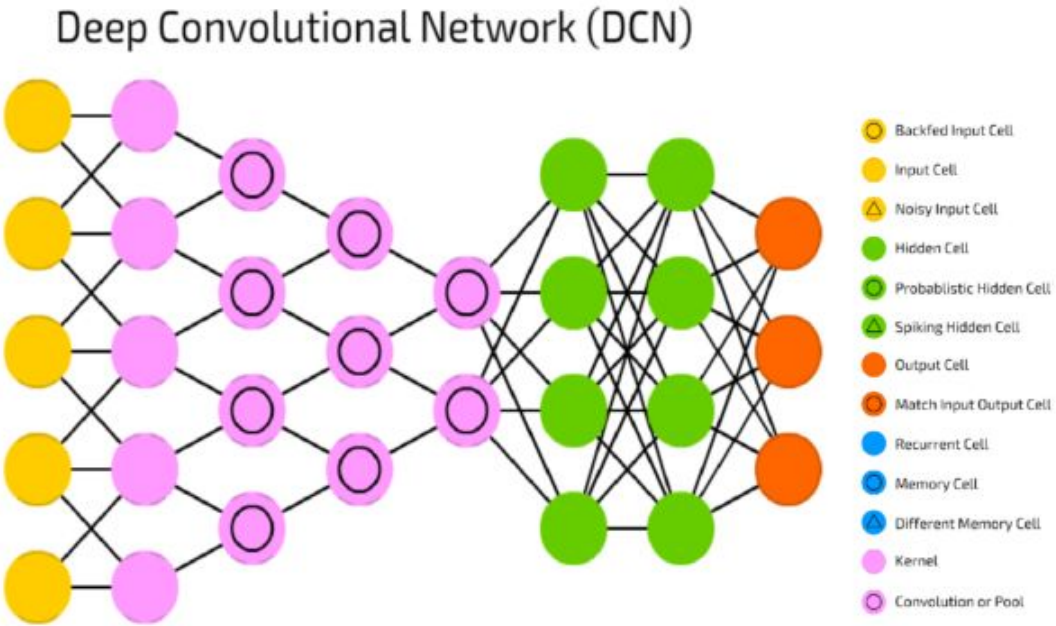
http://www.cognitivetoday.com/2016/10/deep-learning-neural-networks-simplified.html

Deep Dream



- VGG
- Inception
- Resnet

Deep Dream

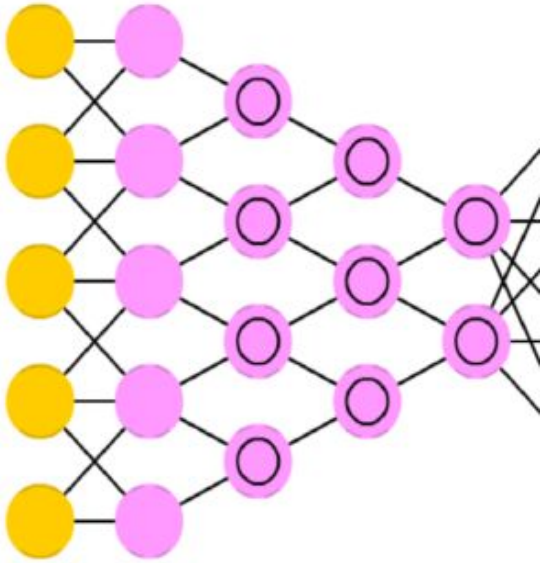


- VGG
- Inception
- Resnet

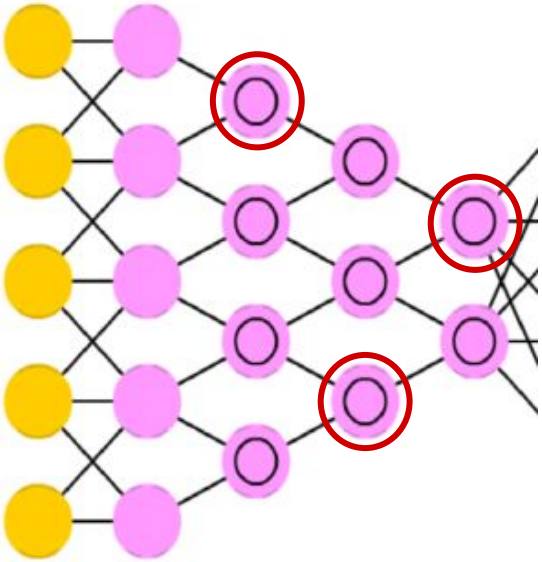
```
from keras.applications import vgg16,inception_v3,resnet50

vgg = vgg16.VGG16()
```

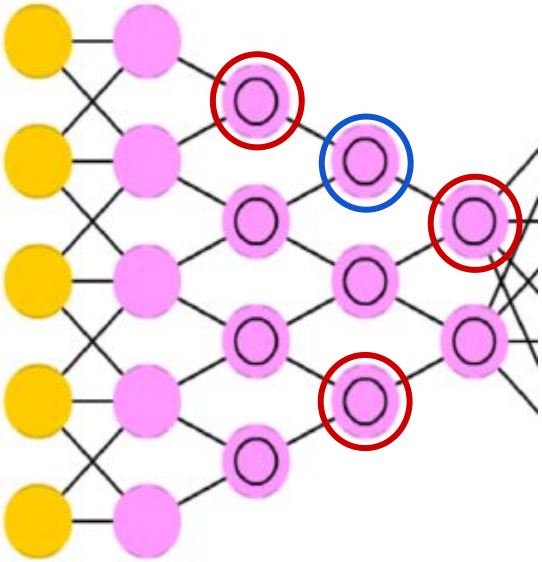
Deep Dream



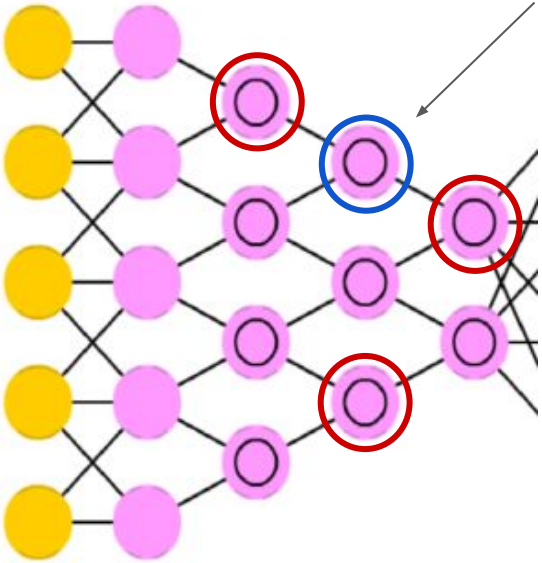
Deep Dream



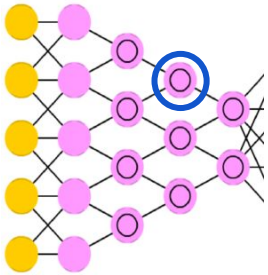
Deep Dream



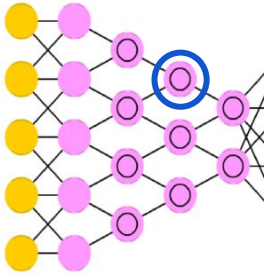
Deep Dream



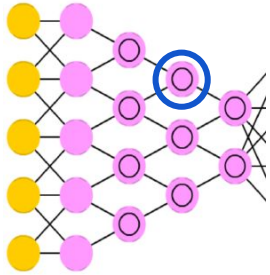
Deep Dream



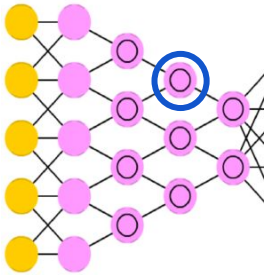
Calculate Cost



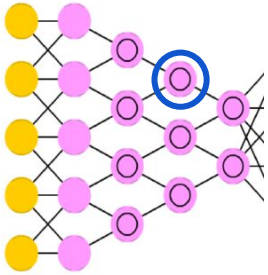
Upgrade Image



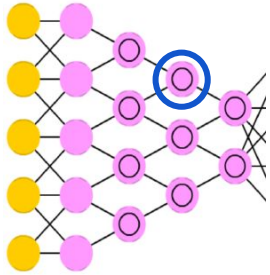
Deep Dream



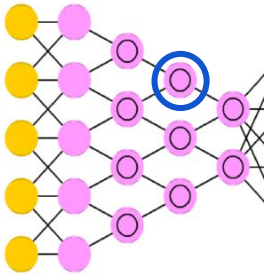
Calculate Cost



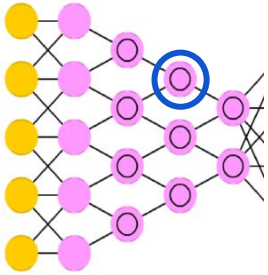
Upgrade Image



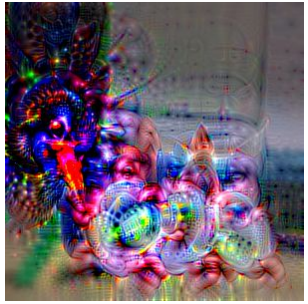
Deep Dream

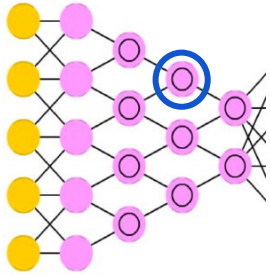


Calculate Cost

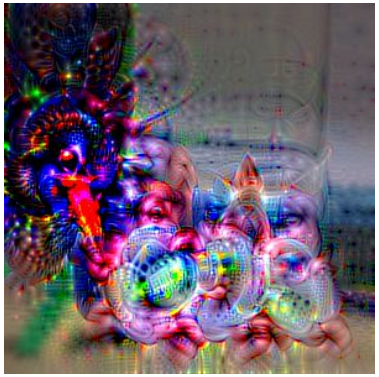


Upgrade Image





Deep Dream



```
settings = {'features': {'block1_conv1': 0.05,
                         'block2_conv2': 0.05,
                         'block3_conv3': 0.05},
            'continuity': 0.1,
            'dream_l2': 0.3,
            'jitter': 5}

# define the loss
loss = K.variable(0.)
for layer_name in settings['features']:
    # add the L2 norm of the features of a layer to the loss
    assert layer_name in layer_dict.keys(), 'Layer ' + layer_name + ' not found in model.'
    coeff = settings['features'][layer_name]
    x = layer_dict[layer_name].output
    shape = layer_dict[layer_name].output_shape
    # we avoid border artifacts by only involving non-border pixels in the loss
    loss -= coeff * K.sum(K.square(x[:, 2: shape[1] - 2, 2: shape[2] - 2, :])) / np.prod(shape[1:])

# add continuity loss (gives image local coherence, can result in an artful blur)
loss += settings['continuity'] * continuity_loss(dream) / np.prod(img_size)
# add image L2 norm to loss (prevents pixels from taking very high values, makes image darker)
loss += settings['dream_l2'] * K.sum(K.square(dream)) / np.prod(img_size)

# compute the gradients of the dream wrt the loss
grads = K.gradients(loss, dream)

outputs = [loss]
if type(grads) in {list, tuple}:
    outputs += grads
else:
    outputs.append(grads)

f_outputs = K.function([dream], outputs)
```

Job properties Size 23.66 MB Jakub Czakon

Deep Dream training

S Succeeded after a minute

Dashboard | Channels | Charts | Parameters | Actions

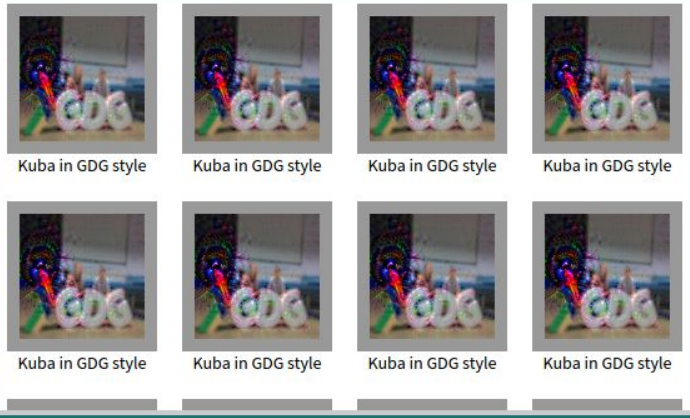
Sort by order number

numeric (1) text (1) image (1)

Search

1 logging_chan…	2 training loss	3 result image
Iteration 99 com…	-4181.880859375	*image*

3 result image



Job properties Size 23.66 MB Jakub Czakon

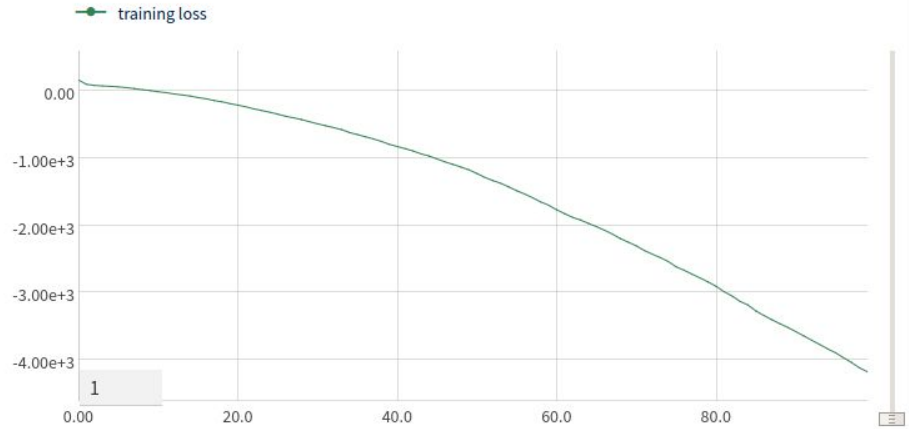
Deep Dream training

S Succeeded after a minute

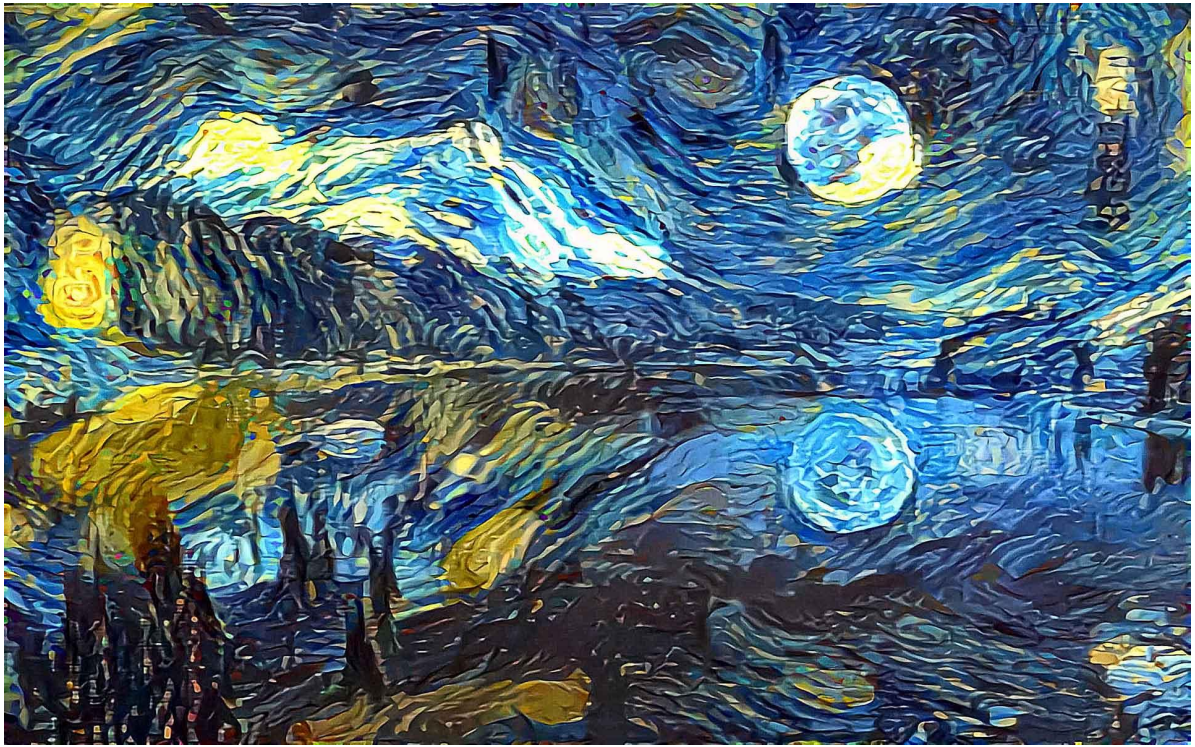
Dashboard | Channels | Charts | Parameters | Actions

create new chart

training loss delete edit

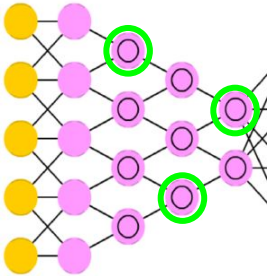


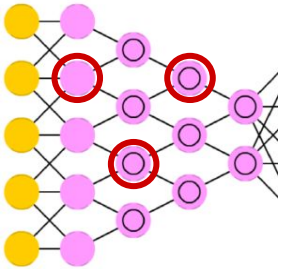
Style Transfer



https://github.com/titu1994/Neural-Style-Transfer

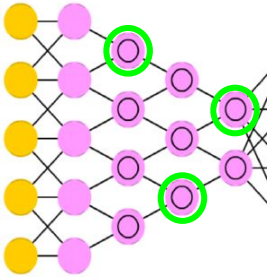
Style Transfer

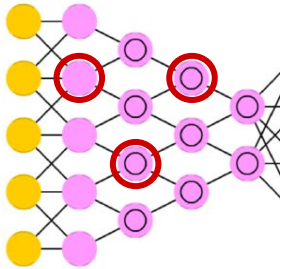


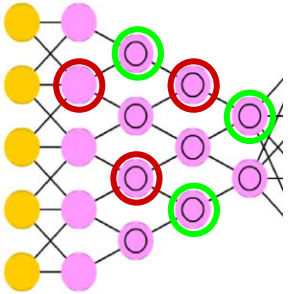


https://arxiv.org/abs/1508.06576

Style Transfer

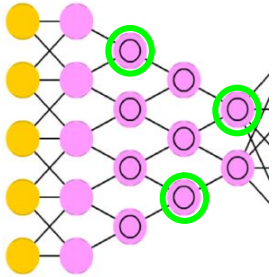




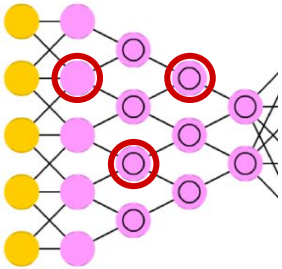


https://arxiv.org/abs/1508.06576

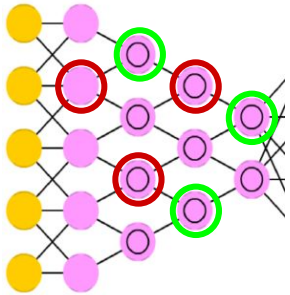
Style Transfer



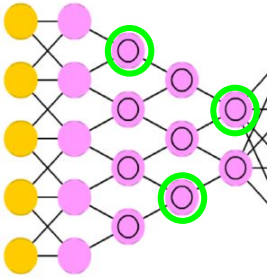
Calculate Cost



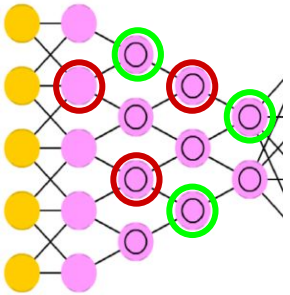
Upgrade Image



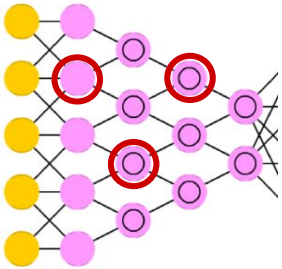
Style Transfer



Upgrade Image



Calculate Cost



Style Transfer

```
loss = K.variable(0.)
layer_features = outputs_dict['block4_conv2']
base_image_features = layer_features[0, :, :, :]
combination_features = layer_features[2, :, :, :]
loss += content_weight * content_loss(base_image_features,
                                      combination_features)

feature_layers = ['block1_conv1', 'block2_conv1',
                  'block3_conv1', 'block4_conv1',
                  'block5_conv1']
for layer_name in feature_layers:
    layer_features = outputs_dict[layer_name]
    style_reference_features = layer_features[1, :, :, :]
    combination_features = layer_features[2, :, :, :]
    sl = style_loss(style_reference_features, combination_features)
    loss += (style_weight / len(feature_layers)) * sl
loss += total_variation_weight * total_variation_loss(combination_image)

# get the gradients of the generated image wrt the loss
grads = K.gradients(loss, combination_image)

outputs = [loss]
if type(grads) in {list, tuple}:
    outputs += grads
else:
    outputs.append(grads)

f_outputs = K.function([combination_image], outputs)
```

Job properties Size 0.00 kB Jakub Czakon

Style transfer training

R Running for a minute abort

Dashboard | Channels | Charts | Parameters | Actions

Sort by order number

numeric (1) text (1) image (1)

Search

1 logging_chan...
Iteration: 57

2 training loss
791396352

3 result image
image

3 result image

Kuba in GDG style | Kuba in GDG style | Kuba in GDG style | Kuba in GDG style

Kuba in GDG style | Kuba in GDG style | Kuba in GDG style | Kuba in GDG style

Job properties Size 0.00 kB Jakub Czakon

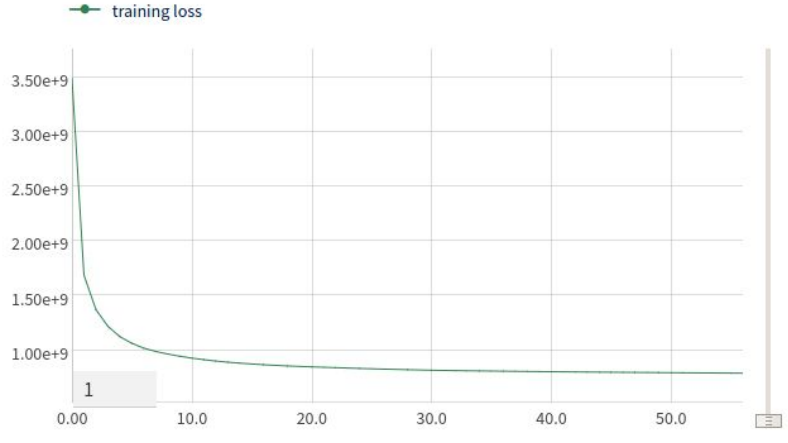
Style transfer training

R Running for 2 minutes abort

Dashboard | Channels | Charts | Parameters | Actions

create new chart

training loss delete edit



Character Recurrent Neural Network

“Real learning, attentive, real learning, deep learning, is playful and frustrating and joyful and discouraging and exciting and sociable and private all the time, which is what makes it great.”

Character Recurrent Neural Network

“Real learning, attentive, real learning, deep learning, is playful and frustrating and joyful and discouraging and exciting and sociable and private all the time, which is what makes it great.”

```
X: ['R', 'e', 'a', 'l', ' ', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n']
y: ['t']

X: ['l', ' ', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v']
y: ['e']

X: ['e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v', 'e', ',', ' ']
y: ['r']

X: ['n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v', 'e', ',', ' ', 'r', 'e', 'a']
y: ['l']
```

Character Recurrent Neural Network

“Real learning, attentive, real learning, deep learning, is playful and frustrating and joyful and discouraging and exciting and sociable and private all the time, which is what makes it great.”

```
X: ['R', 'e', 'a', 'l', ' ', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n']
y: ['t']

X: ['l', ' ', 'l', 'e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v']
y: ['e']

X: ['e', 'a', 'r', 'n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v', 'e', ',', ' ']
y: ['r']

X: ['n', 'i', 'n', 'g', ',', ' ', 'a', 't', 't', 'e', 'n', 't', 'i', 'v', 'e', ',', ' ', 'r', 'e', 'a']
y: ['l']
```

```
X = Input(shape=(max_len, nr_classes))
rnn = LSTM(256, return_sequences=True)(X)
rnn = Dropout(0.2)(rnn)
rnn= LSTM(256, return_sequences=False)(rnn)
rnn= Dropout(0.2)(rnn)
text_output = Dense(nr_classes,activation='softmax')(rnn)

char_rnn_model = Model(X,y)
char_rnn_model.compile(loss='categorical_crossentropy', optimizer='adam')
```

Character Recurrent Neural Network

"Deep learning is fun ... "

https://github.com/alvations/Quotables

https://github.com/sherjilozair/char-rnn-tensorflow

Character Recurrent Neural Network

*"**Deep learning is fun** … "*

"Deep learning is funding."

"Deep learning is funny. It sits in the future. I made my theory, friend."

"Deep learning is fun to love."

"Deep learning is funding of head that then became personal."

"Deep learning is fundamental ideas that dazzle you."

https://github.com/alvations/Quotables

https://github.com/sherjilozair/char-rnn-tensorflow

Magic the Gathering

http://www.mtgsalvation.com/forums/creativity/custom-card-creation/612057-generating-magic-cards-using-deep-recurrent-neural?page=98

Magic the Gathering

http://www.mtgsalvation.com/forums/creativity/custom-card-creation/612057-generating-magic-cards-using-deep-recurrent-neural?page=98

deepjazz.io

Star 1,319

deepjazz

Using Keras & Theano for deep learning driven jazz generation

LISTEN GITHUB Download .zip Download .tar.gz

Using Keras & Theano for deep learning driven jazz generation

I built *deepjazz* in 36 hours at a hackathon. It uses Keras & Theano, two deep learning libraries, to generate jazz music. Specifically, it builds a two-layer LSTM, learning from the given MIDI file. It uses deep learning, the AI tech that powers Google's AlphaGo and IBM's Watson, **to make music -- something that's considered as deeply human**.

deepjazz has been featured in The Guardian, Aeon Magazine, Inverse, Data Skeptic, the front page of HackerNews, and GitHub's trending showcase (1200+ stars). It has led to the most popular "AI" artist on SoundCloud with 172,000+ listens. Currently, deepjazz is being used as reference material for the course "Interactive Intelligent Devices" at the University of Perugia.

Want to listen?

jakubczakon / devfest

Watch 0 | Star 0 | Fork 0

Code | Issues 0 | Pull requests 0 | Projects 0 | Wiki | Pulse | Graphs | Settings

Branch: master | devfest / README.md | Find file | Copy path

jakubczakon Update README.md — b976fef just now

2 contributors

30 lines (15 sloc) | 1.2 KB | Raw | Blame | History

Google Developers Group Devfest Warsaw 2016

GDG DevFest Season 2016

Jakub Czakon

"How to have fun with deep learning"

https://deepsense.io

deepsense.io BIG DATA SCIENCE

www.codilime.com/

codilime® CREATING VALUE

Find out more

- jakub.czakon@codilime.com
- We are waiting for your questions!
- Join our team!
  jobs@deepsense.io

www deepsense.io

www codilime.com