Unblackboxing the black box

27/09/2017

Agenda

- Training
- Image
- Text
- **Pizza, chill and Q&A**

Why unblackboxing

- We need to debug/improve
- Clients want to understand
- Users deserve to know why

Let’s train something

- Facial recognition model

- Tweet sentiment model

```
['sat with jo ',
 "@YoungQ Awww I'm sorry Rob. If I was there, I'd give them a piece of my mind for ya. ",
 '@stinggoddess its an awesome movie. i fail to see how that is a bad thing. and yes they were fine ',
 'Hi guys !! i just seen the new moon trail ^^ its the best  http://bit.ly/LL8dN    &lt;3']


['                    is so sad for my APL friend.............',
 '                  I missed the New Moon trailer...',
 "           .. Omgaga. Im sooo  im gunna CRy. I've been at this dentist since 11.. I was suposed 2 just get a cr
own put on (30mins)...",
 '         i think mi bf is cheating on me!!!       T_T']
```

Let’s train something
Neptune

- Monitor
- Investigate
- Interact
- Archive
- …

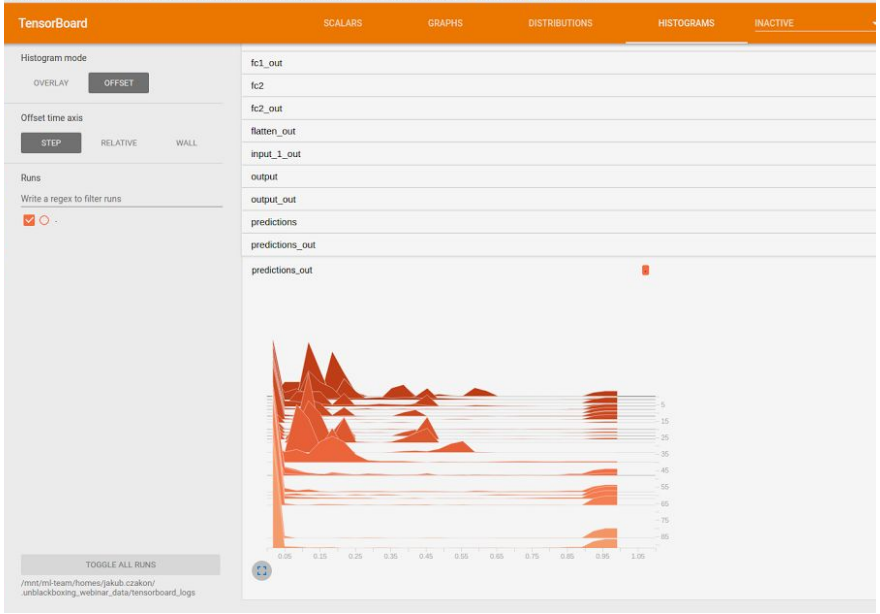
DEMO

https://neptune.ml/

Let’s train something
Tensorboard

- Graph
- Gradients
- Activations
- …

DEMO



Let unblackboxing begin

Image - Output Activations

```
layer_output = model.get_layer('interesting_layer').output
ouptut_extracotr = Model(inputs=base_model.input, outputs=layer_output)

output_extractor.predict(X)
```

DEMO

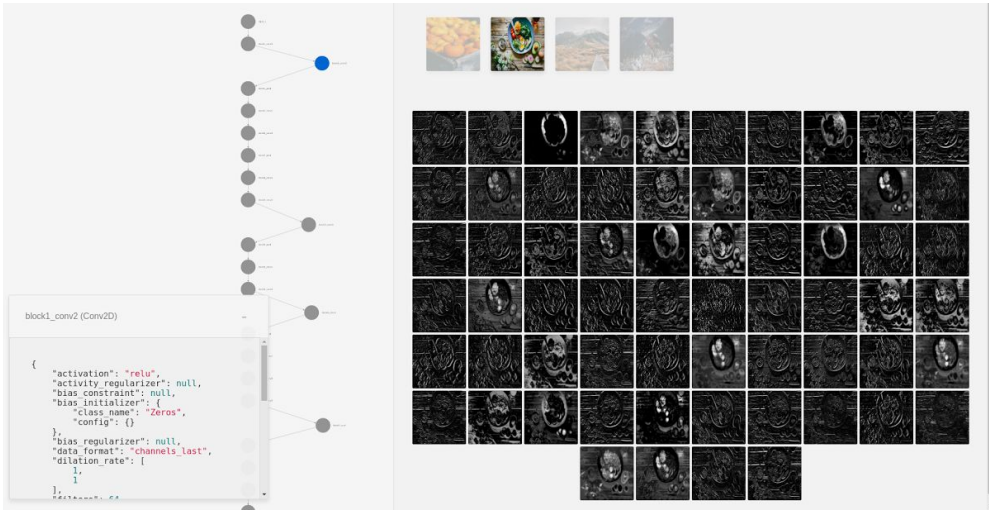
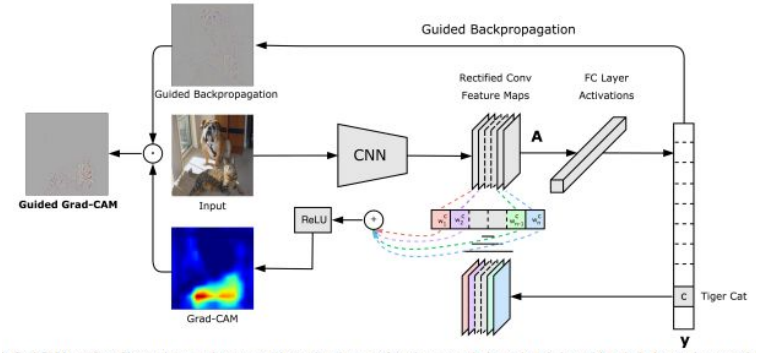


Image - CAM



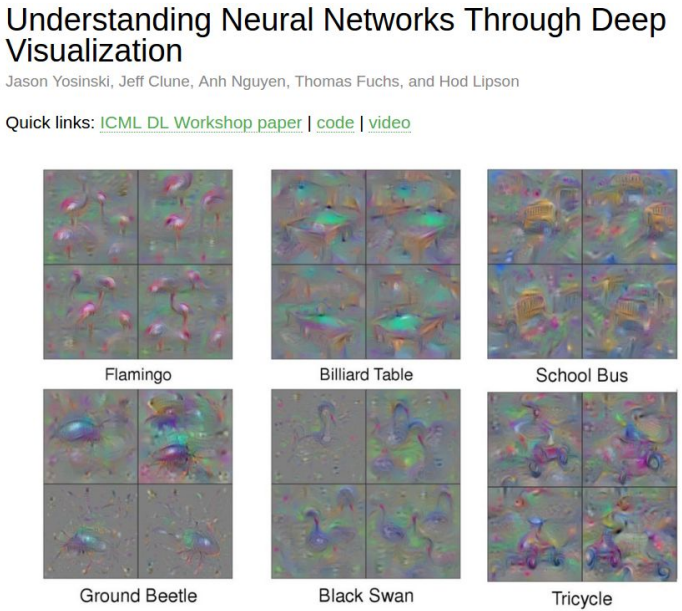

```
target_layer = lambda x: target_category_loss(x, category_index, nb_classes)
model.add(Lambda(target_layer,
                 output_shape = target_category_loss_output_shape))

loss = K.sum(model.layers[-1].output)
conv_output =  [l for l in model.layers[0].layers if l.name is layer_name][0].output
grads = normalize(K.gradients(loss, conv_output)[0])
gradient_function = K.function([model.layers[0].input], [conv_output, grads])

output, grads_val = gradient_function([image])
output, grads_val = output[0, :], grads_val[0, :, :, :]

weights = np.mean(grads_val, axis = (0, 1))
```

https://arxiv.org/pdf/1610.02391v1.pdf
https://github.com/jacobgil/keras-grad-cam/blob/master/grad-cam.py

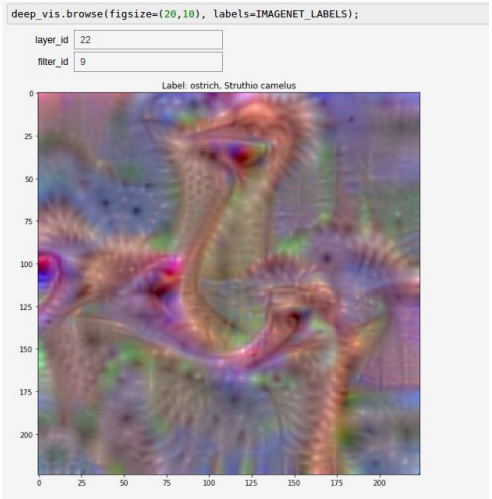
DEMO

Image - MAI

Max Activation Image (Deep Dream for Class)



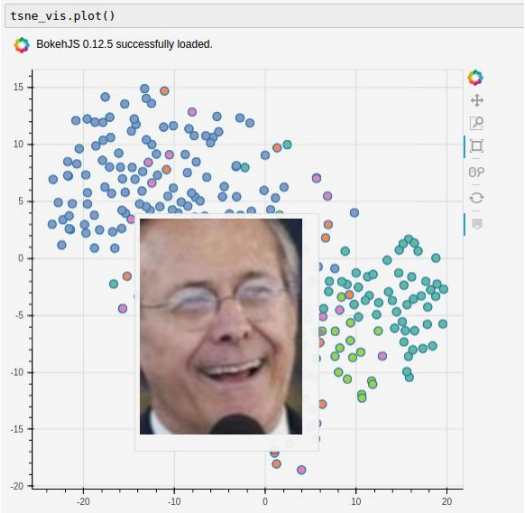
http://yosinski.com/deepvis



DEMO

Image - TSNE

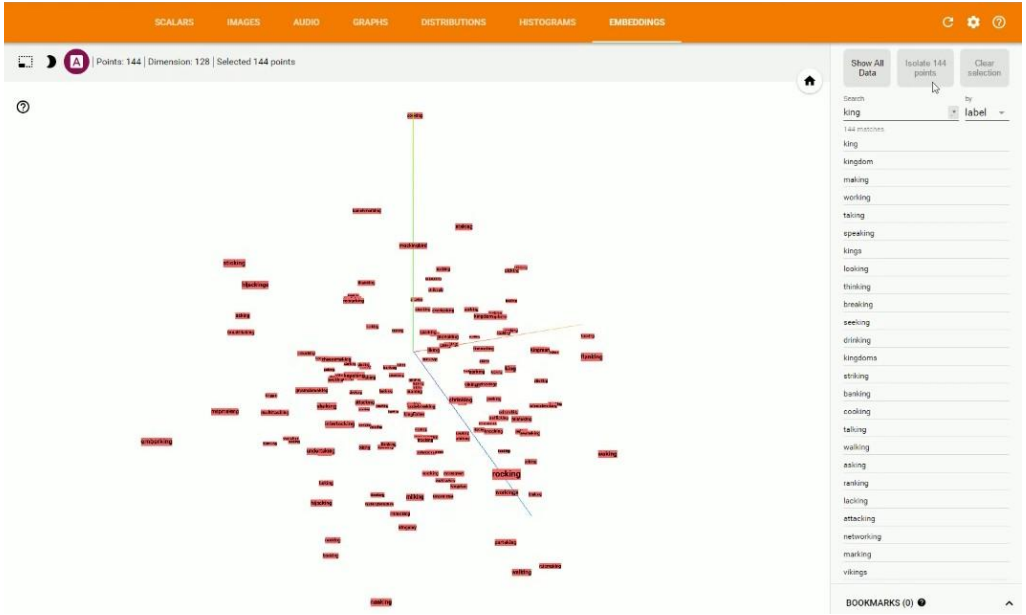
- Investigate the latent space
- More on distill https://distill.pub/2016/misread-tsne/



DEMO

Text - embeddings

- word2vec, doc2vec, vaguelycreativestuff2vec
- fastText *https://github.com/facebookresearch/fastText*



DEMO

https://www.tensorflow.org/programmers_guide/embedding

Text - LIME

Local Interpretable Model Agnostic Explanations

"Why Should I Trust You?"
Explaining the Predictions of Any Classifier

Marco Tulio Ribeiro
University of Washington
Seattle, WA 98105, USA
marcotcr@cs.uw.edu

Sameer Singh
University of Washington
Seattle, WA 98105, USA
sameer@cs.uw.edu

Carlos Guestrin
University of Washington
Seattle, WA 98105, USA
guestrin@cs.uw.edu

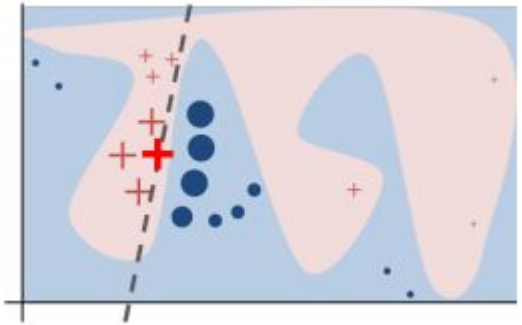
ABSTRACT

Despite widespread adoption, machine learning models remain mostly black boxes. Understanding the reasons behind predictions is, however, quite important in assessing *trust*, which is fundamental if one plans to take action based on a prediction, or when choosing whether to deploy a new model. Such understanding also provides insights into the model, which can be used to transform an untrustworthy model or prediction into a trustworthy one.

how much the human understands a model's behaviour, as opposed to seeing it as a black box.

Determining trust in individual predictions is an important problem when the model is used for decision making. When using machine learning for medical diagnosis [6] or terrorism detection, for example, predictions cannot be acted upon on blind faith, as the consequences may be catastrophic.

Apart from trusting individual predictions, there is also a need to evaluate the model as a whole before deploying it "in the wild". To make this decision, users need to be confident



https://arxiv.org/pdf/1602.04938.pdf

Text - LIME
Quick Explanation

- Idea:
  - Populate local subspace
  - Score observations in that subspace
  - Fit regression
    - Linear Regression! We want to reproduce the score value
    - Weighted by distance to original observation predictions
  - Interpret simple linear model

Text - LIME

Local Interpretable Model Agnostic Explanations

```
text = 'I absolutely love this tech talk but'
tweet_predictor.predict_proba([text])
```

```
array([[ 0.16043527,  0.83956468]], dtype=float32)
```

```
te = TextExplainer()
te.fit(text, tweet_predictor.predict_proba);
```

```
te.show_prediction(target_names=['negative','positive'])
```

y=positive (probability **0.870**, score **1.897**) top features

Contribution?	Feature
+1.712	Highlighted in text (sum)
+0.185	<BIAS>

i absolutely love this tech talk but

DEMO

Text - Investigate Activations

- Visualize activations for text models
- Interact with it!

https://medium.com/@plusepsilon/visualizations-of-recurrent-neural-networks-c18f07779d56

DEMO

Use it wisely

- Github repo https://github.com/deepsense-ai/unblackboxing_webinar
- Neptune https://neptune.ml/

Thanks