

Assignment 2

Dr. Wojtek Kowalczyk

wojtek@liacs.nl

Objectives of A2

1. Master 3 types of networks and learn some of their applications:
 - *Convolutional*
 - *Recurrent*
 - *Autoencoders*
2. Master modern tools for training deep networks on GPUs (*Python + Tensorflow/Keras*).
3. Learn how to apply “Deep Learning” to some interesting problems.

A2: step-by-step

1. Get acquainted with TensorFlow and Keras by studying and running various programs/tutorials that are available on the internet
2. Select, for each type of networks (CNNs, RNNs, AutoEncoders), an interesting problem and produce a “proof-of-understanding” by:
 - *Describing the problem and the network(s) used in your experiments*
 - *Describing the data, the experiments, and the results of your experiments*

This step shouldn't cost you much time: just browse the internet, re-run some experiments, modifying data or network parameters and describe your findings in the report (up to 3 pages in total).

A2: the last step

3. The Challenge:

Find an original problem and/or a data set and apply “Deep Learning” to it. You are free to use any type of network – also those that were not covered during the course.

Some ad-hoc ideas:

- *“from ASCII to UNICODE” (e.g., French, Polish, German)*
- *“automatic extractions of equations from pdfs”*
- *“images of equations into Latex”*
- *“restoration of images”*

A2: the report

1. Spend most time and effort on the “challenge” part of A2. This part of the report shouldn't exceed **7 pages**. Only this part of your work will be evaluated.
Don't forget to attach your code!
2. Produce **3 pages** (one per network type), providing “a proof of understanding” – it is a compulsory part (not graded) and the reviewers will only check if you actually produced such a “proof”.
3. Keep in mind that the grade for A2 will be based on the “challenge part” of your report.
4. *If needed: get help/feedback from instructors.*

Software

Keras

<https://github.com/keras-team/keras/tree/master/examples>

<https://keras.io/>

TensorFlow

<https://www.tensorflow.org/>

<https://colab.research.google.com/>

Hardware

- A GPU-servers: duranium.liacs.nl, tritanium.liacs.nl (reachable from LIACS network or via ssh gateway gold.liacs.nl)
- For small data sets you may use “normal computers”! TensorFlow can be installed on Linux, Windows, Mac
- Any PC with a modern NVIDIA GPU (with DDR5 RAM) would be much faster that a plain PC.
- On Wed. 20/03, 11:00: *[intro to DSLab/duranium, etc.](#)*

Resources

1. *Lots of examples implemented in KERAS/Tensorflow:*
<https://github.com/keras-team/keras/tree/master/examples>
2. https://github.com/Yaffa1607/Tensorflow_HVASS
3. Popular data sets used in Deep Learning:
<http://deeplearning.net/datasets/>
4. *Tutorials:*
<http://deeplearning.net/reading-list/tutorials/>
5. Papers:
<http://deeplearning.net/reading-list/>

Examples

- Sequence to Sequence Learning
- “Reading Comprehension”
- Visualization of Kernels of Convolutional NNs
- “Google Deep Dream”
- OCR (images of short words, distorted fonts)
- Sentiment/Text analysis (IMDB data sets)
- www.kaggle.com/deepmatrix/imdb-5000-movie-dataset
- Text Generation
- Adversarial Examples
- ...

Sequence To Sequence Learning

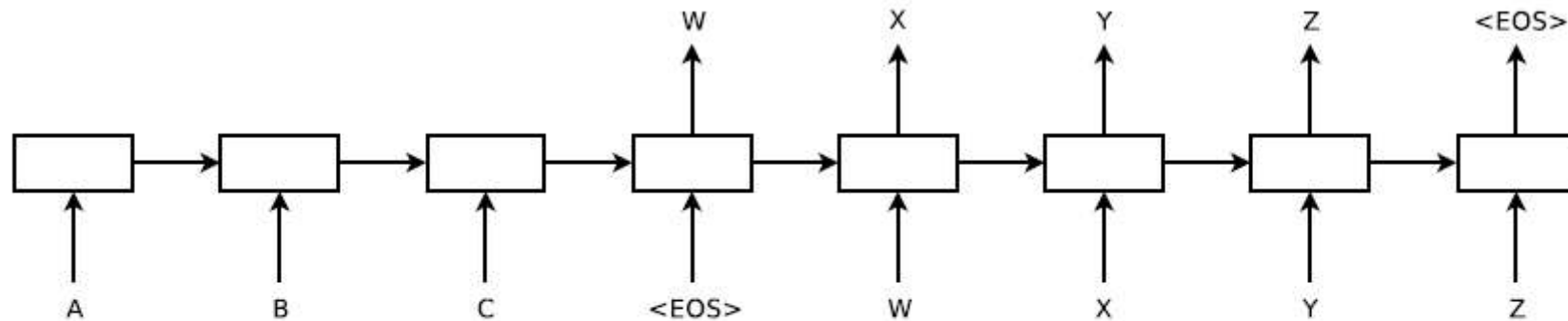


Figure 1: Our model reads an input sentence “ABC” and produces “WXYZ” as the output sentence. The model stops making predictions after outputting the end-of-sentence token. Note that the LSTM reads the input sentence in reverse, because doing so introduces many short term dependencies in the data that make the optimization problem much easier.

- English to French
- ‘3 + 5’ to ‘8’

Reading Comprehension

Develop a NN to pass the “reading comprehension task”

Task 1: Single Supporting Fact

Mary went to the bathroom.
John moved to the hallway.
Mary travelled to the office.
Where is Mary? A:office

Task 2: Two Supporting Facts

John is in the playground.
John picked up the football.
Bob went to the kitchen.
Where is the football? A:playground

Task 3: Three Supporting Facts

John picked up the apple.
John went to the office.
John went to the kitchen.
John dropped the apple.
Where was the apple before the kitchen? A:office

Task 4: Two Argument Relations

The office is north of the bedroom.
The bedroom is north of the bathroom.
The kitchen is west of the garden.
What is north of the bedroom? A: office
What is the bedroom north of? A: bathroom

Task 5: Three Argument Relations

Mary gave the cake to Fred.
Fred gave the cake to Bill.
Jeff was given the milk by Bill.
Who gave the cake to Fred? A: Mary
Who did Fred give the cake to? A: Bill

Task 6: Yes/No Questions

John moved to the playground.
Daniel went to the bathroom.
John went back to the hallway.
Is John in the playground? A:no
Is Daniel in the bathroom? A:yes

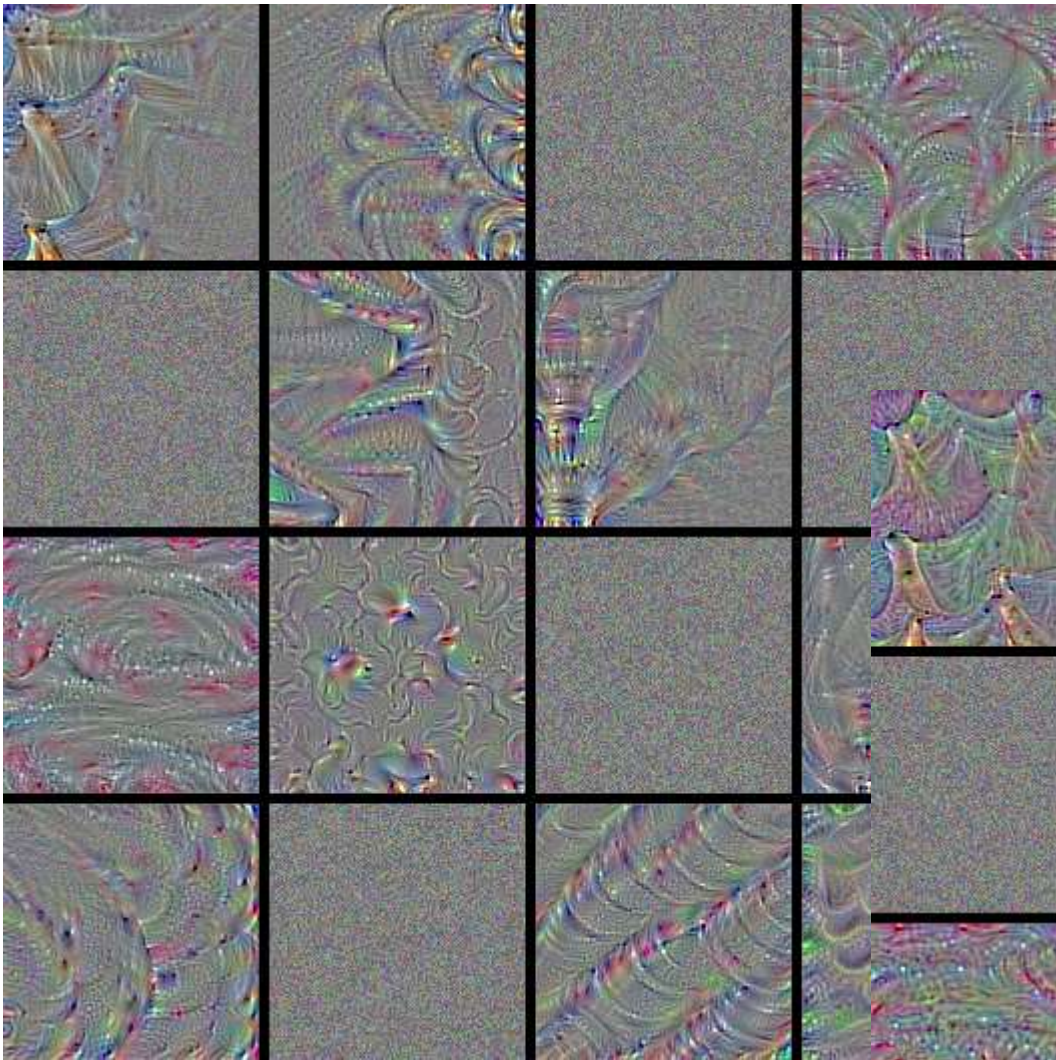
Reading Comprehension

TASK	Weakly Supervised		Uses External Resources	Strong Supervision (using supporting facts)						
	N-gram Classifier	LSTM	Structured SVM <i>COREF+ SRL features</i>	MemNN <i>Reshan et al. (2014)</i>	MemNN <i>ADAPTIVE MEMORY</i>	MemNN <i>AM + N-GRAMS</i>	MemNN <i>AM + NONLINEAR</i>	MemNN <i>AM + NG + NL</i>	No. of ex. req. ≥ 95	MultiTask Training
1 - Single Supporting Fact	36	50	99	100	100	100	100	100	250 ex.	100
2 - Two Supporting Facts	2	20	74	100	100	100	100	100	500 ex.	100
3 - Three Supporting Facts	7	20	17	20	100	99	100	100	500 ex.	98
4 - Two Arg. Relations	50	61	98	71	69	100	73	100	500 ex.	80
5 - Three Arg. Relations	20	70	83	83	83	86	86	98	1000 ex.	99
6 - Yes/No Questions	49	48	99	47	52	53	100	100	500 ex.	100
7 - Counting	52	49	69	68	78	86	83	85	FAIL	86
8 - Lists/Sets	40	45	70	77	90	88	94	91	FAIL	93
9 - Simple Negation	62	64	100	65	71	63	100	100	500 ex.	100
10 - Indefinite Knowledge	45	44	99	59	57	54	97	98	1000 ex.	98
11 - Basic Coreference	29	72	100	100	100	100	100	100	250 ex.	100
12 - Conjunction	9	74	96	100	100	100	100	100	250 ex.	100
13 - Compound Coref.	26	94	99	100	100	100	100	100	250 ex.	100
14 - Time Reasoning	19	27	99	99	100	99	100	99	500 ex.	99
15 - Basic Deduction	20	21	96	74	73	100	77	100	100 ex.	100
16 - Basic Induction	43	23	24	27	100	100	100	100	100 ex.	94
17 - Positional Reasoning	46	51	61	54	46	49	57	65	FAIL	72
18 - Size Reasoning	52	52	62	57	50	74	54	95	1000 ex.	93
19 - Path Finding	0	8	49	0	9	3	15	36	FAIL	19
20 - Agent's Motivations	76	91	95	100	100	100	100	100	250 ex.	100
Mean Performance	34	49	79	75	79	83	87	93		92

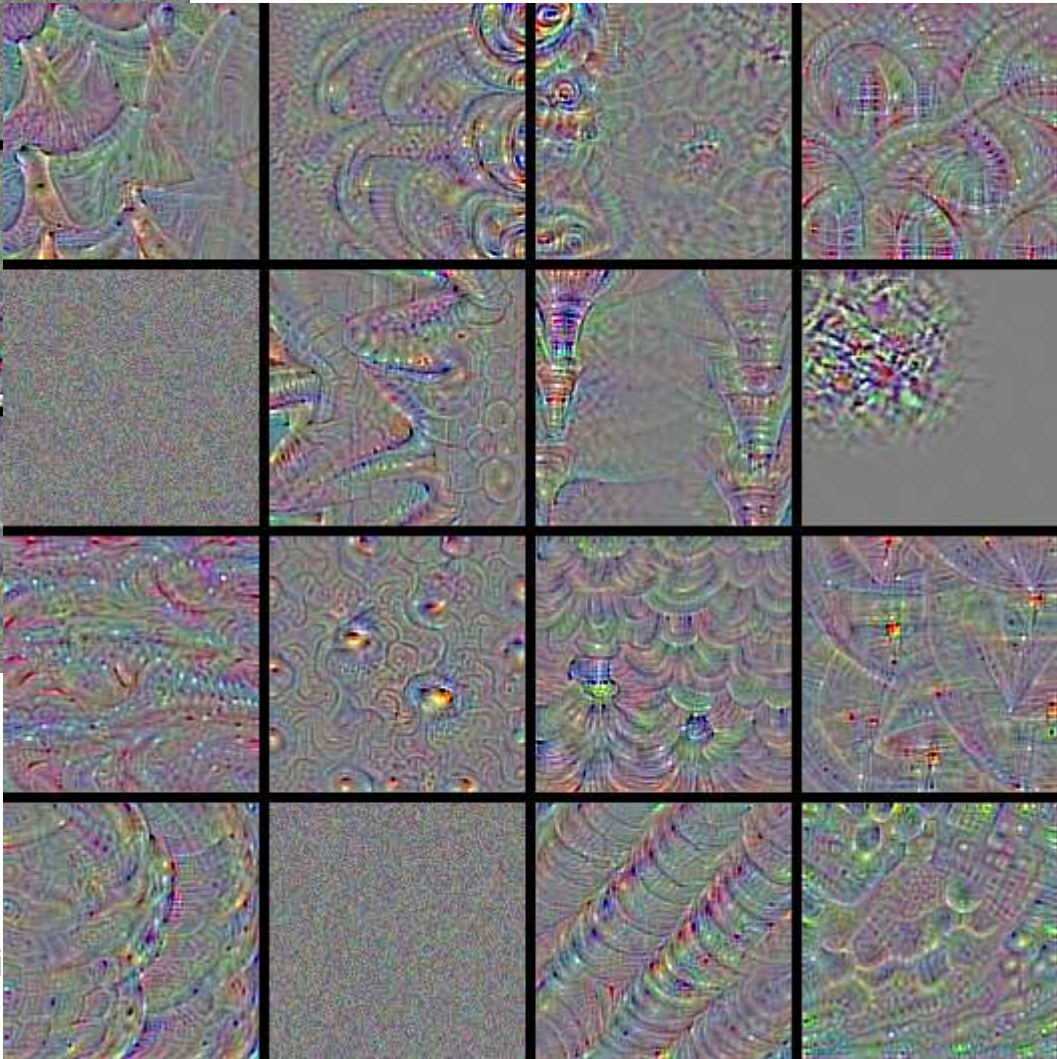
Visualization of Kernels of CNNs

- Train a CNNs (*or download it from the internet!*)
- Select a kernel you want to “understand”
- Search for an input image that strongly activates the selected kernel
=> optimization problem!
- Apply “inverted backpropagation”: input nodes play a role of weights; weights play a role of a “fixed input”

<https://jacobgil.github.io/deeplearning/filter-visualizations>



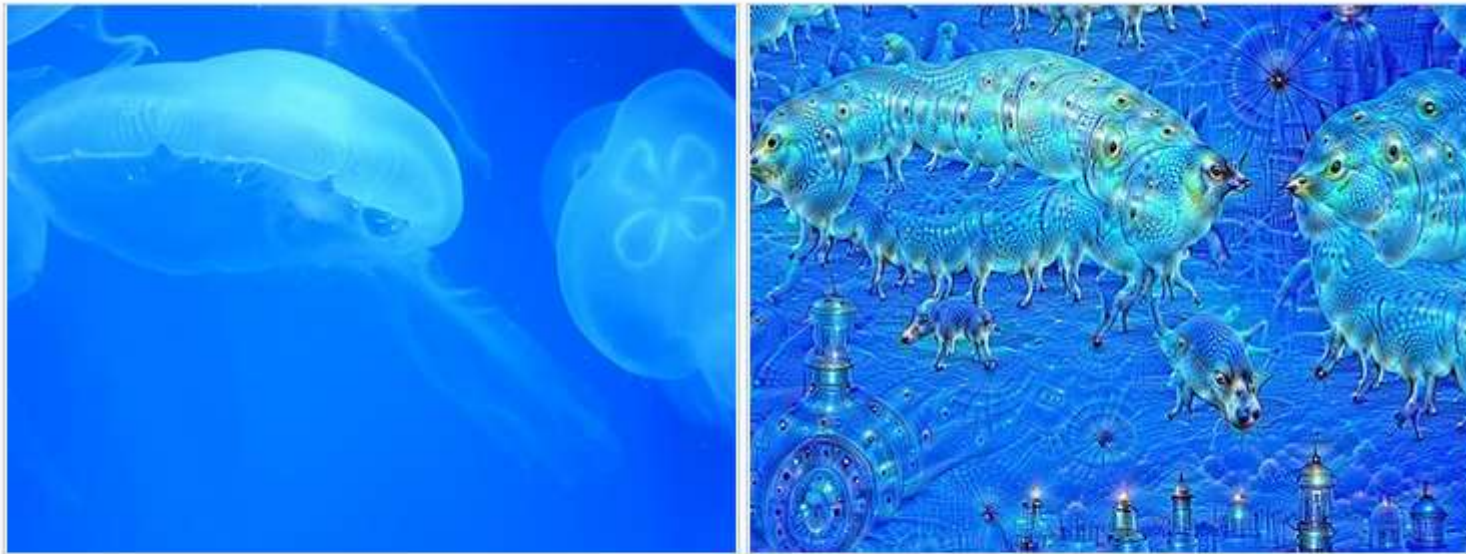
Neural Networks



Instructions

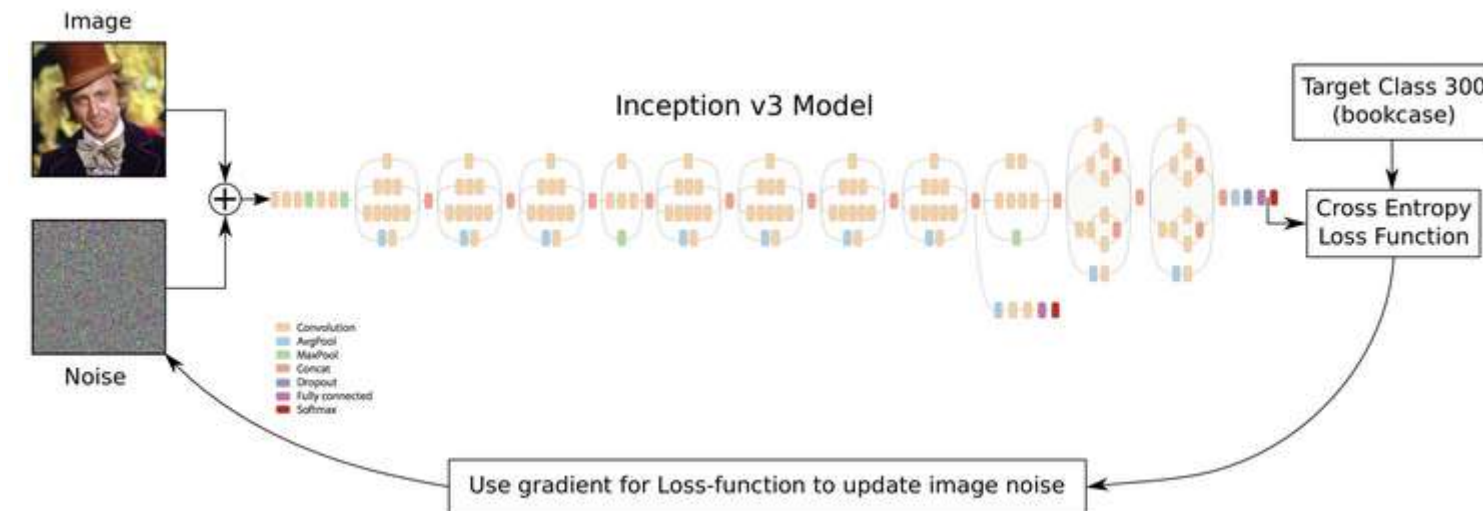
Deep Dream (check Wikipedia)

- Take a trained ImageNet (or something similar)
- Provide an input image (e.g., “a castle”)
- Smoothly change the input so it could be classified as a “fish”
- optimization problem => SGD
- Spectacular effects !



The same image before (left) and after (right) applying ten iterations of DeepDream

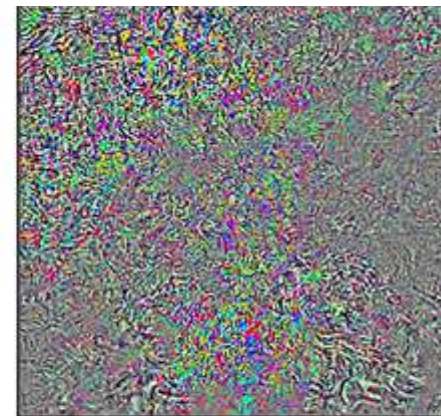
Adversarial Examples



Original Image:
bow tie (97.22%)



Image + Noise:
bow tie (0.00%)
bookcase (99.72%)



Amplified Noise

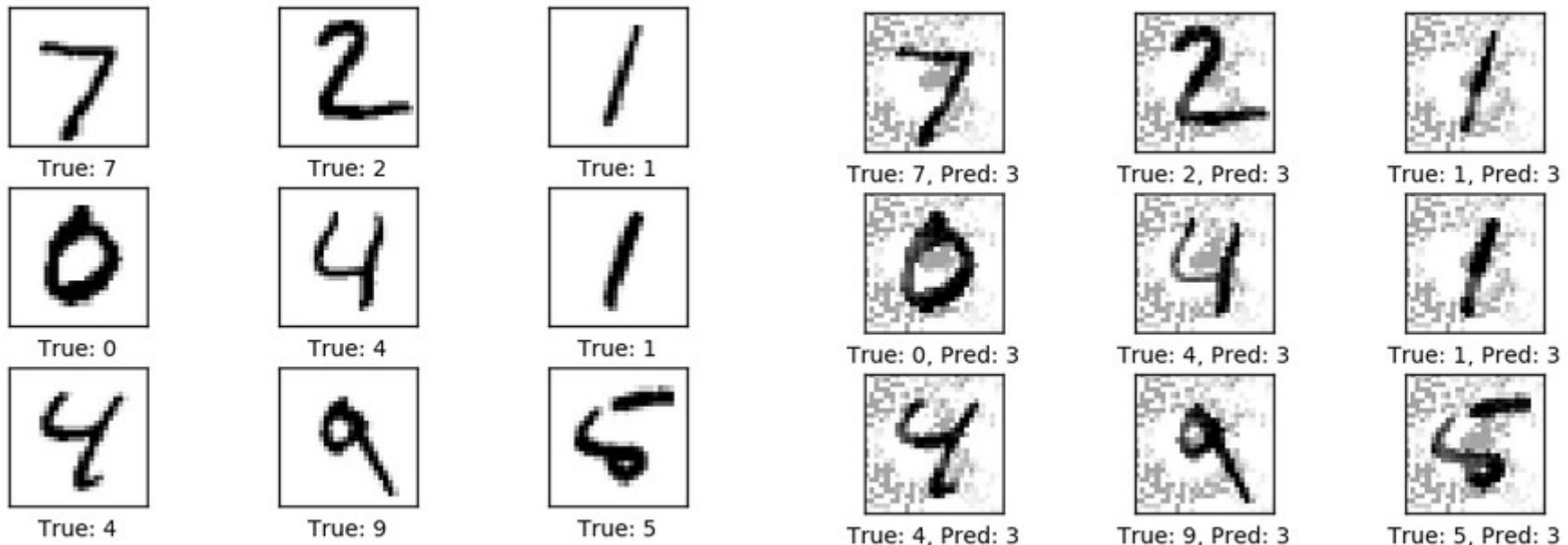
Noise min: -3.000, max: 3.000, mean: -0.000, std: 1.309

Adversarial Examples: MNIST

Accuracy drop from 99% to 11% accuracy (over test set)

https://github.com/Yaffa1607/Tensorflow_HVASS/blob/master/12_Adversarial_Noise_MNIST.ipynb

More: <http://cs231n.github.io/understanding-cnn/>



Automatic Text Generation

- Train a RNN on a collection of documents
- The network learns to predict next word(s)
(*better: probability distribution of “next word”*)
- Apply the network to an initial phrase to generate more text
- More Idea's:
 - try this approach on the Europarl text corpora
 - author recognition
 - detection of unusual phrases (“cheating”?)
 - detection of a “change of style / rhetoric” (e.g. for politicians)
 - ...
- Text Summarization <https://arxiv.org/abs/1506.01057>