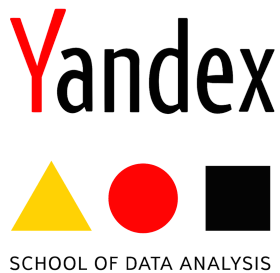
Deep Learning

things happening in science today

Alex Rogozhnikov & Tatiana Likhomanenko

Yandex School of Data Analysis

Imperial College London, 2017



Motivation:

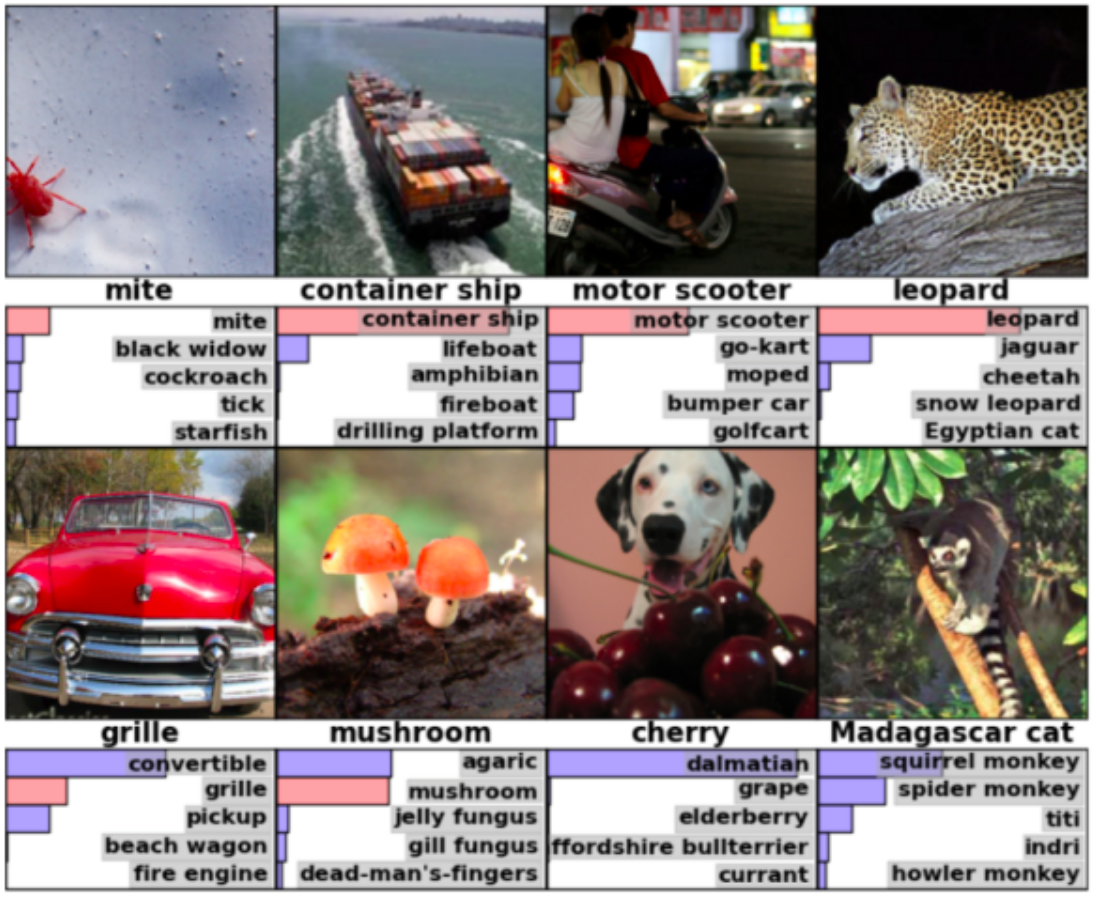
- newsmakers today use the word "Artificial intelligence"
  - while scientist say 'machine learning' about the same things
  - in most cases this is deep learning
- some bridge needed between 'that research in news' and things we studied
- and it is simply interesting

Reminder

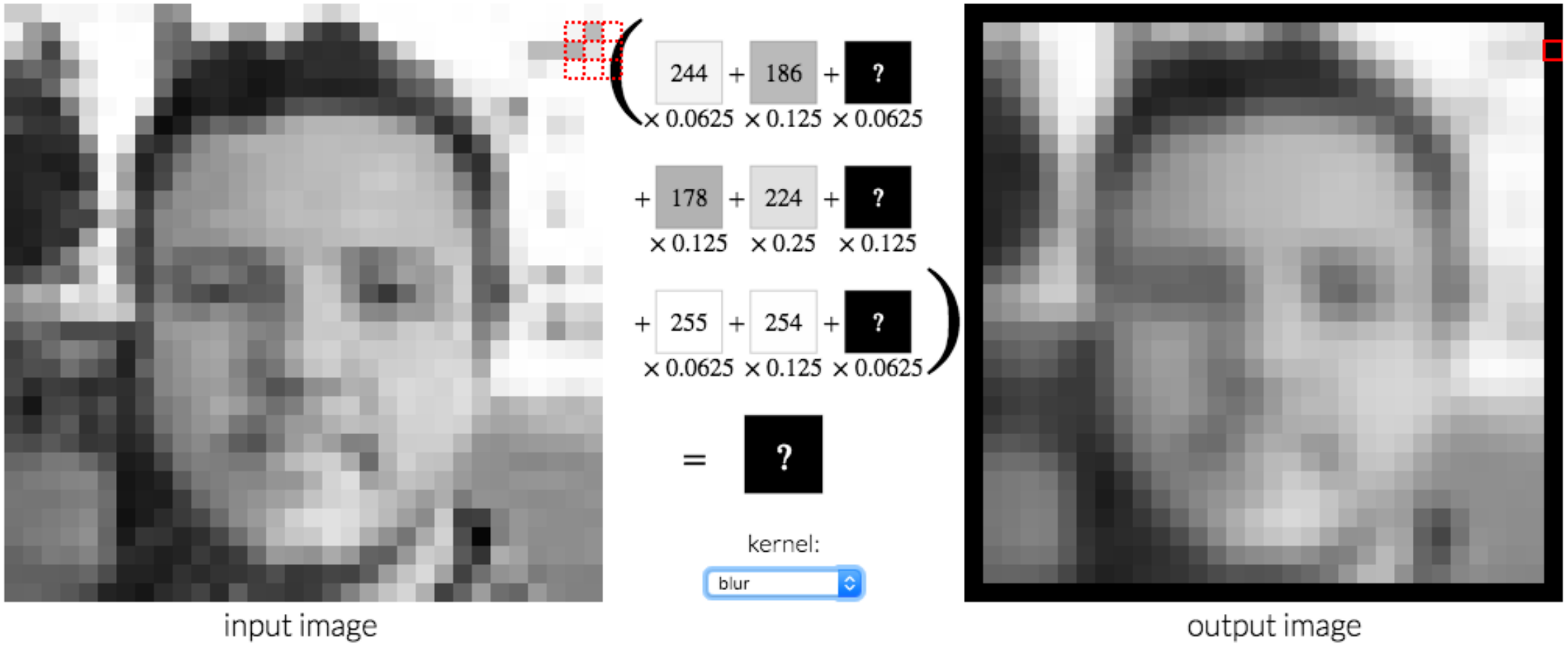
- we studied theano, which operates with tensors in a symbolic way
- wrote a multilayer perceptron
  - provided an expression in tensors to compute result and a function to minimize
  - gradient was derived by theano
  - so one only needs to solve optimization problem
- till the rule used by our model is differentiable, we can apply this approach

Image recognition today

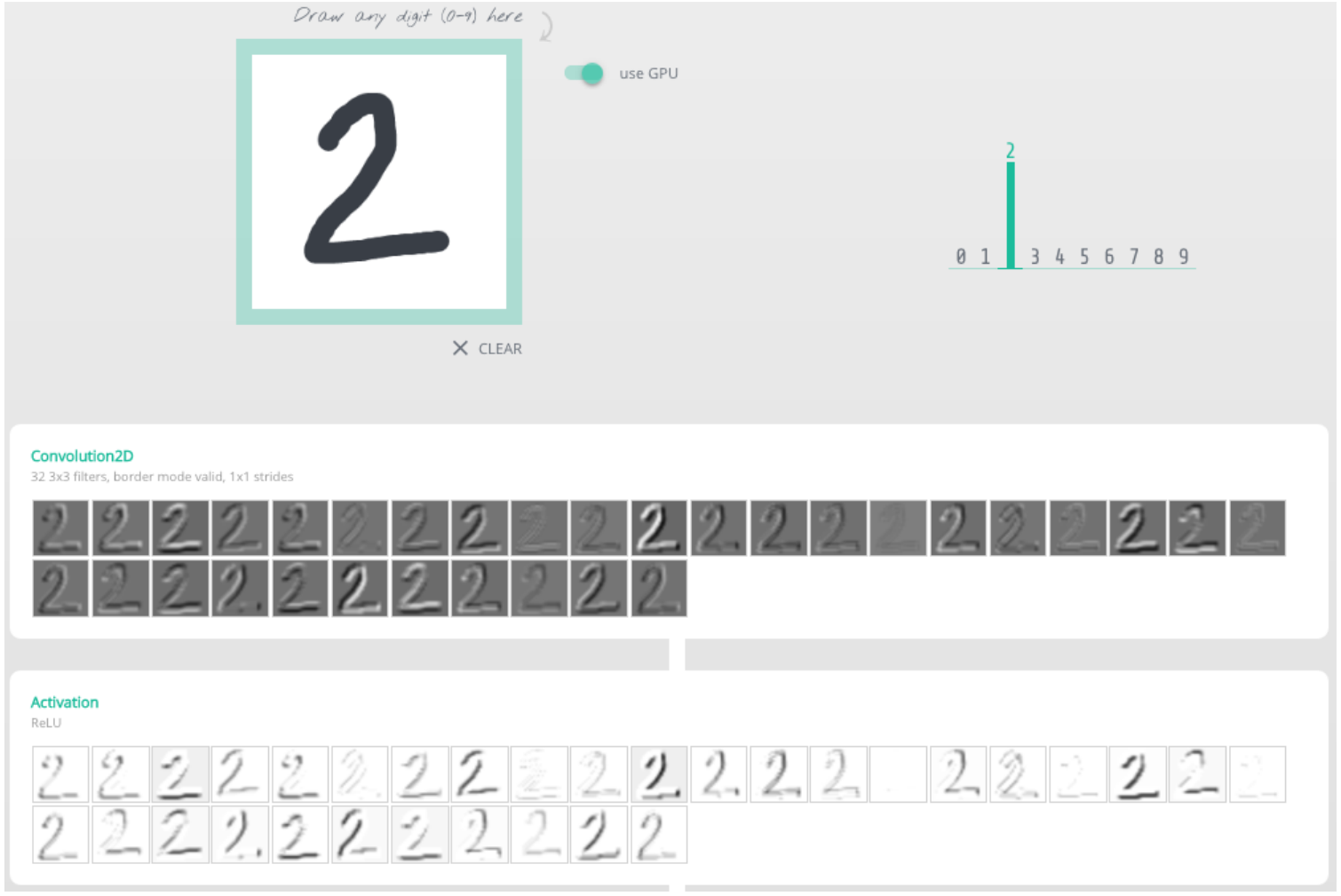
- can classify objects on images with very high accuracy
- even if those are images of 500 x 500 and we have only 400 output neurons, that's $500 * 500 * 400 = 100'000'000$ parameters
  - and that's only linear model!
- can we reduce the number of parameters?



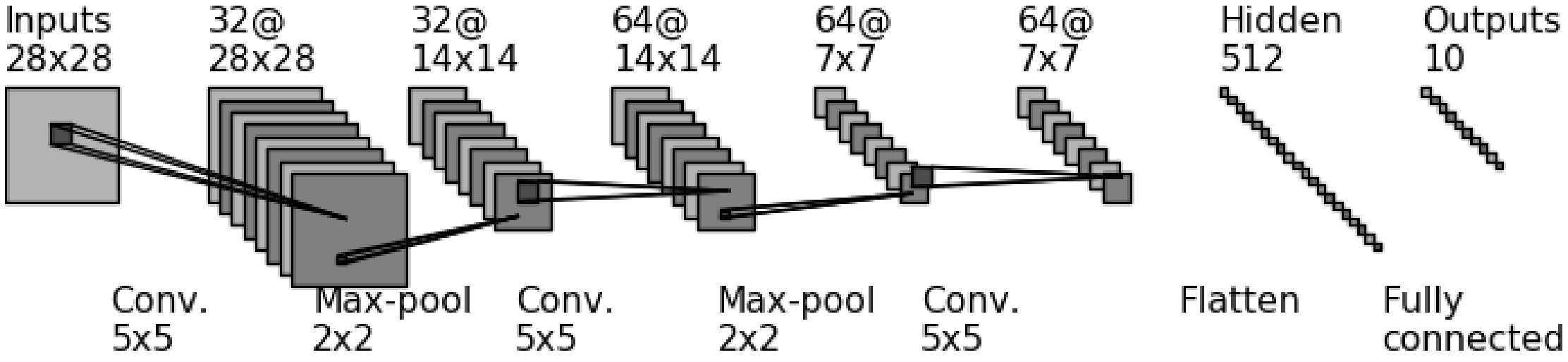
Convolutions



Convolutional neural network (also)



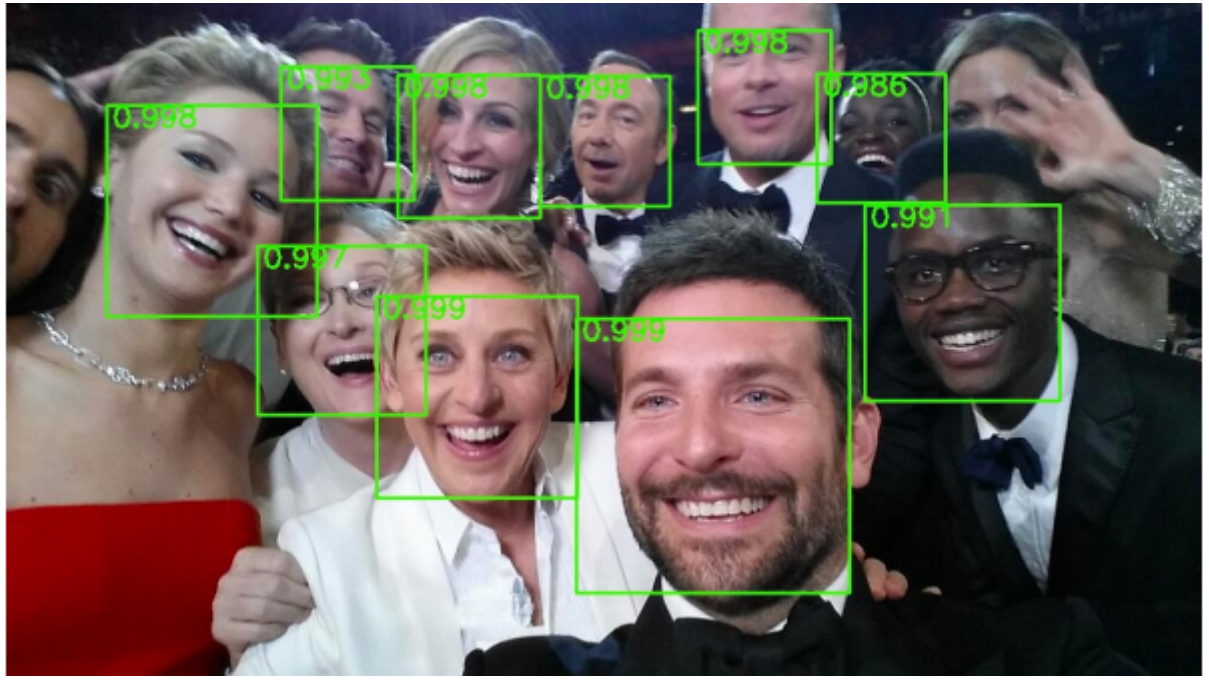
CNN architecture



Components:

- convolutions
- pooling (reducing the size)
- flattening

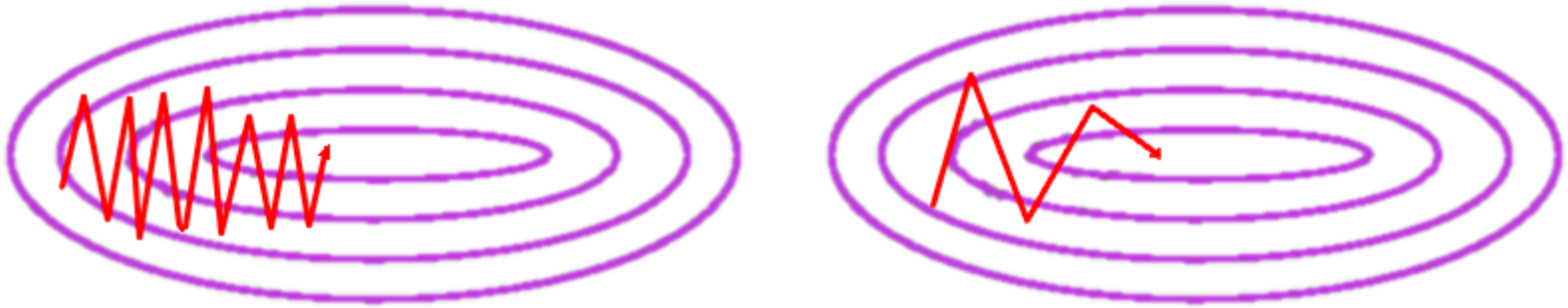
Some idea behind



Detecting particular parts of a face is much easier.

Some idea behind

Problems with deep architectures

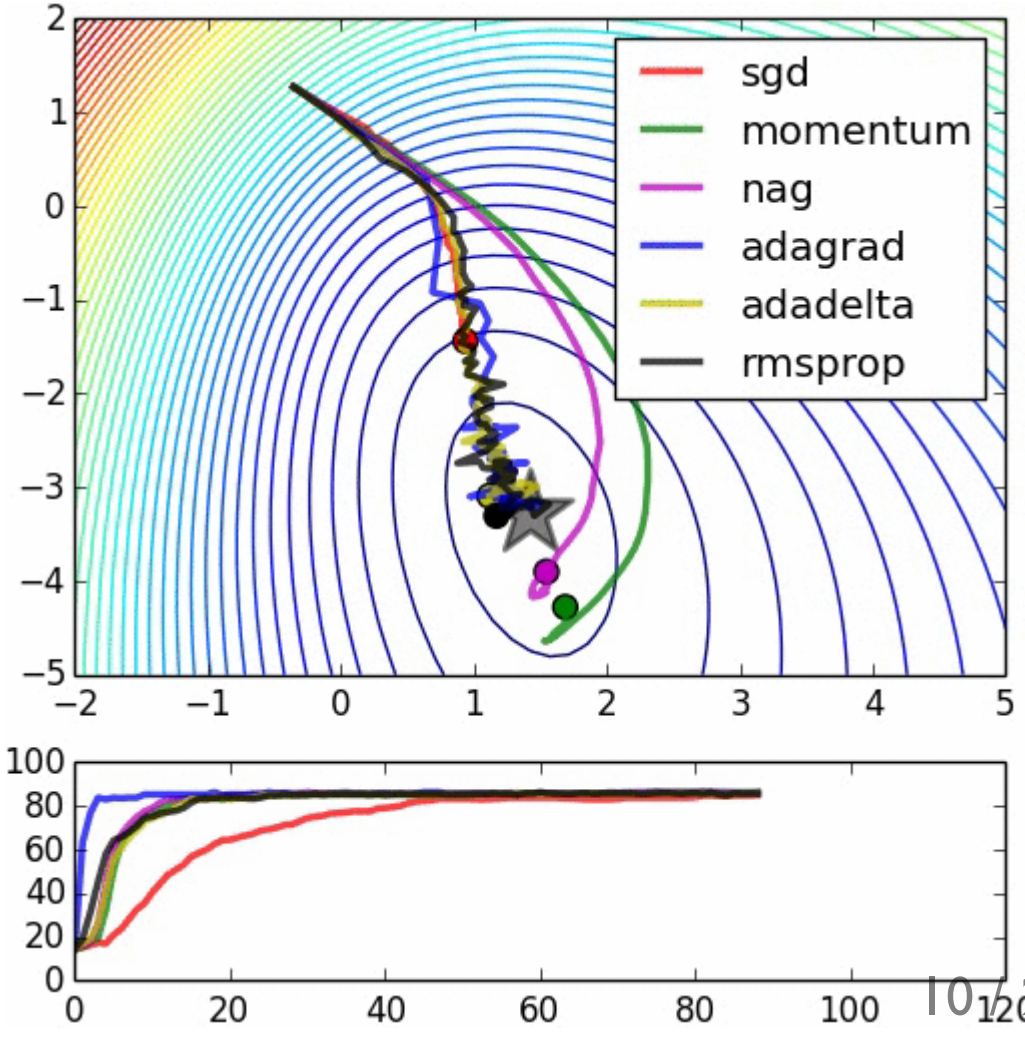


In deep architectures, the gradient varies between parameters drastically (100-1000 times difference).

This problem usually referred to as 'gradient diminishing'.

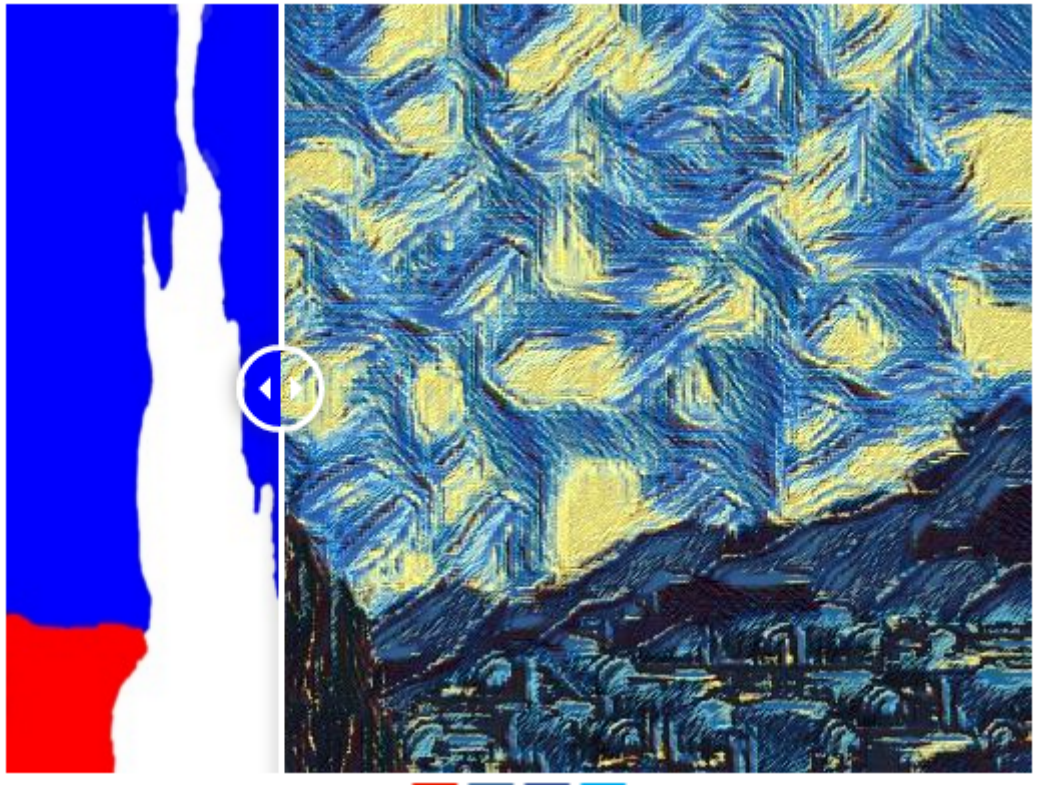
Stochastic optimizations

some toy 2-parameter optimization



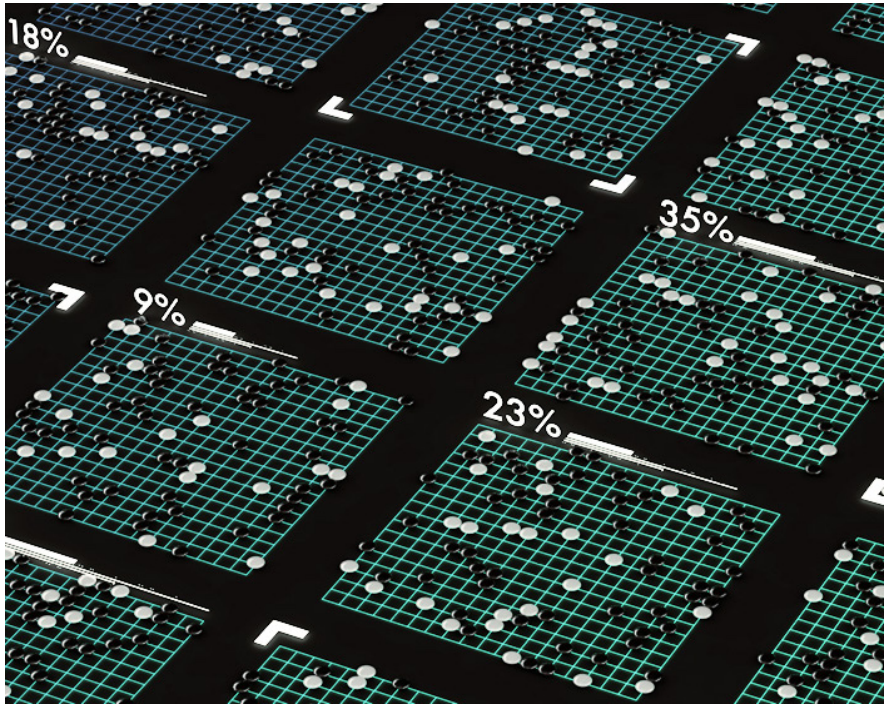
Picture to picture

Sketch to picture (texturing)



AlphaGo

- too many combinations to check
- no reliable way to assess the quality of position

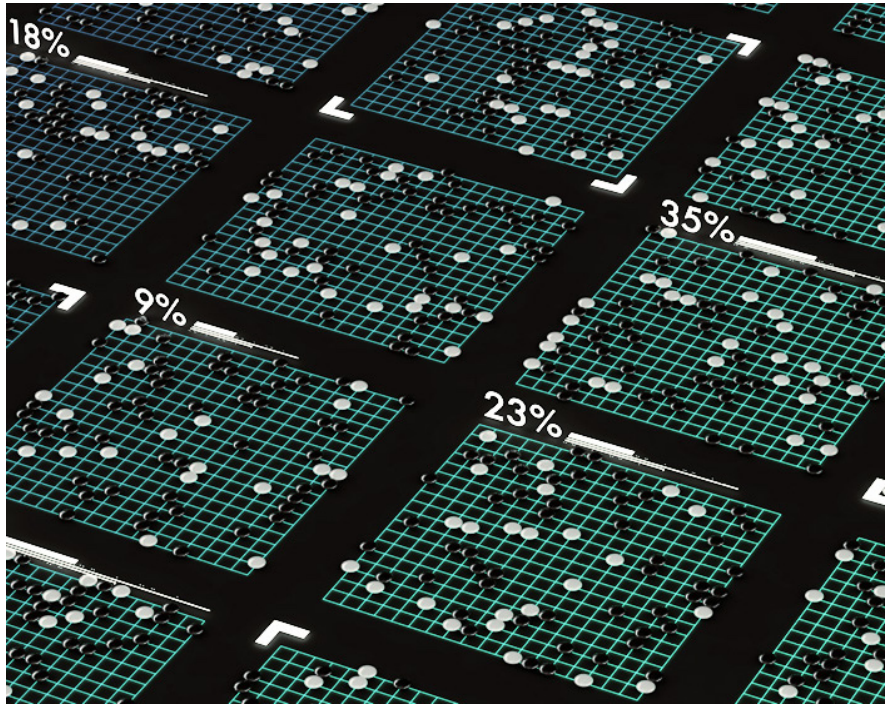


AlphaGo

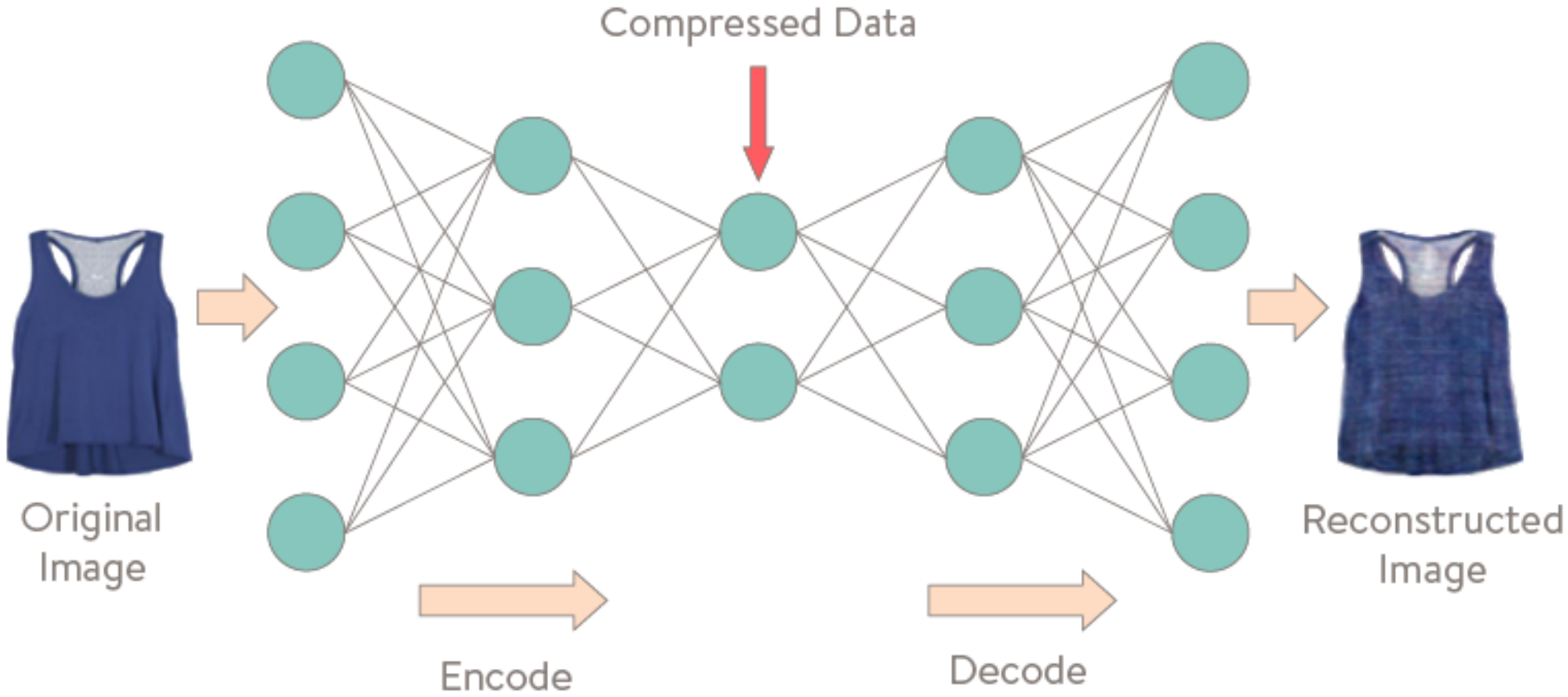
- too many combinations to check
- no reliable way to assess the quality of position

AlphaGo has 2 CNNs:

- value network predicts probability to win
- policy network predicts the next step of a player
- both are given the board (with additional features for each position on the board)

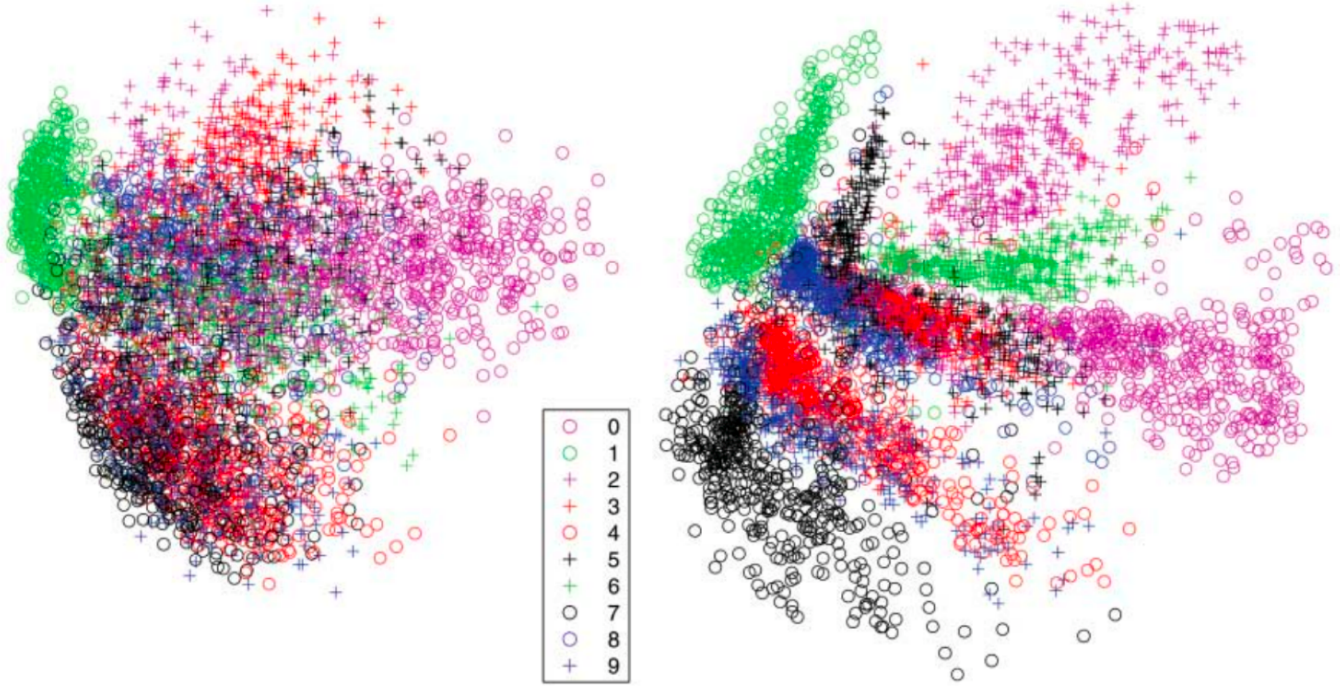


Autoencoding



- compressed data is treated as a result of an algorithm

Autoencoding



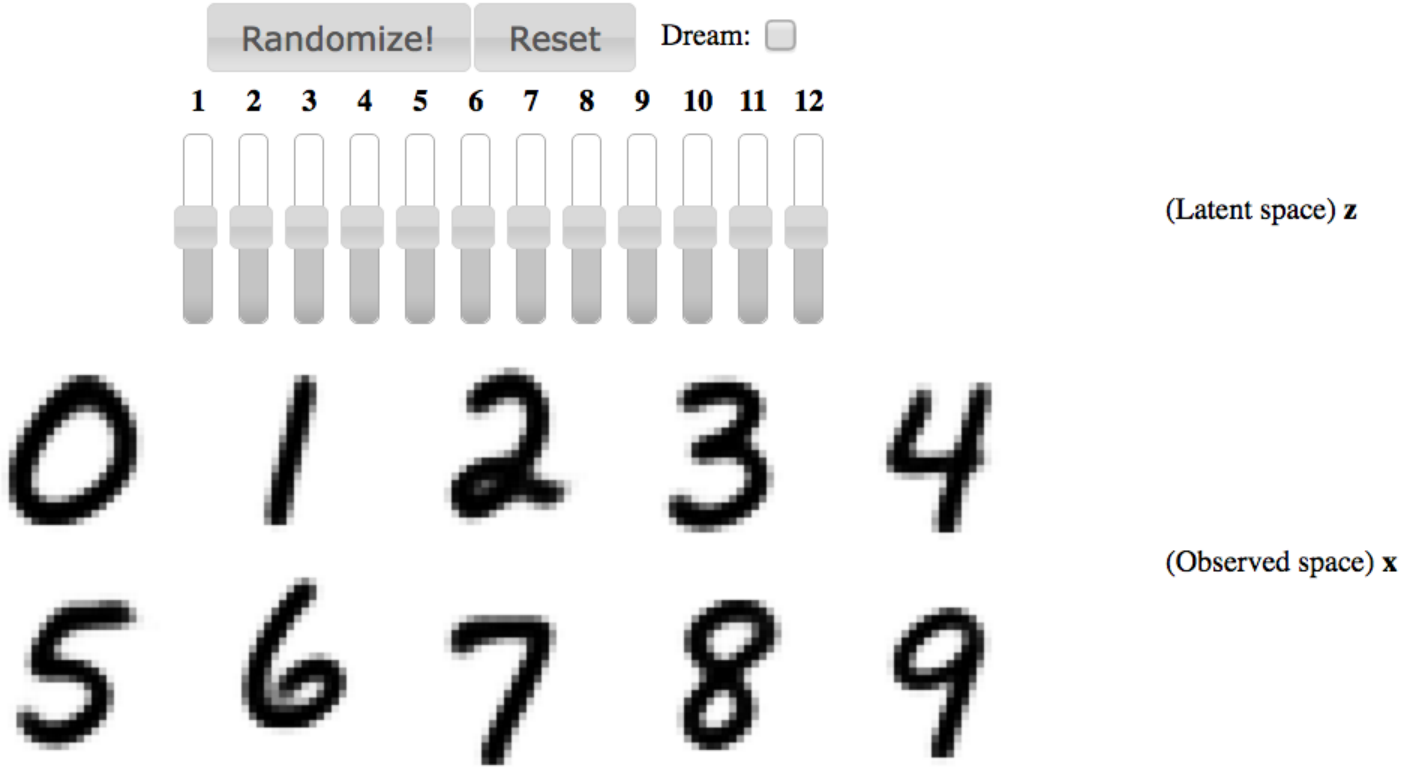
Comparison of PCA and autoencoder

(with 2 neurons in the middle layer)

You can also play with it

Variational autoencoder

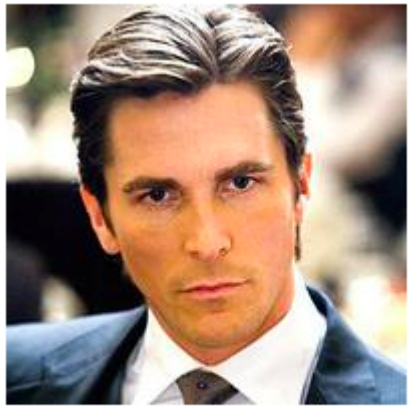
(an example to play with the variables of latent space)



An example with fonts

autoencoder trained to reconstruct symbols from different fonts can generate new fonts when

Gaze correction

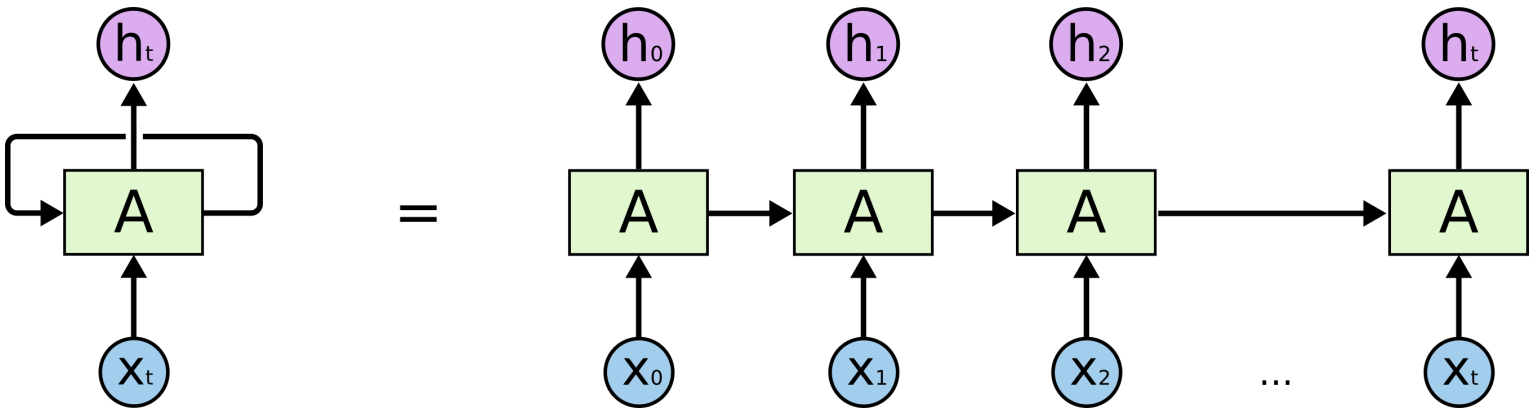


- shift correction + light correction applied
- training data was collected specially for this task

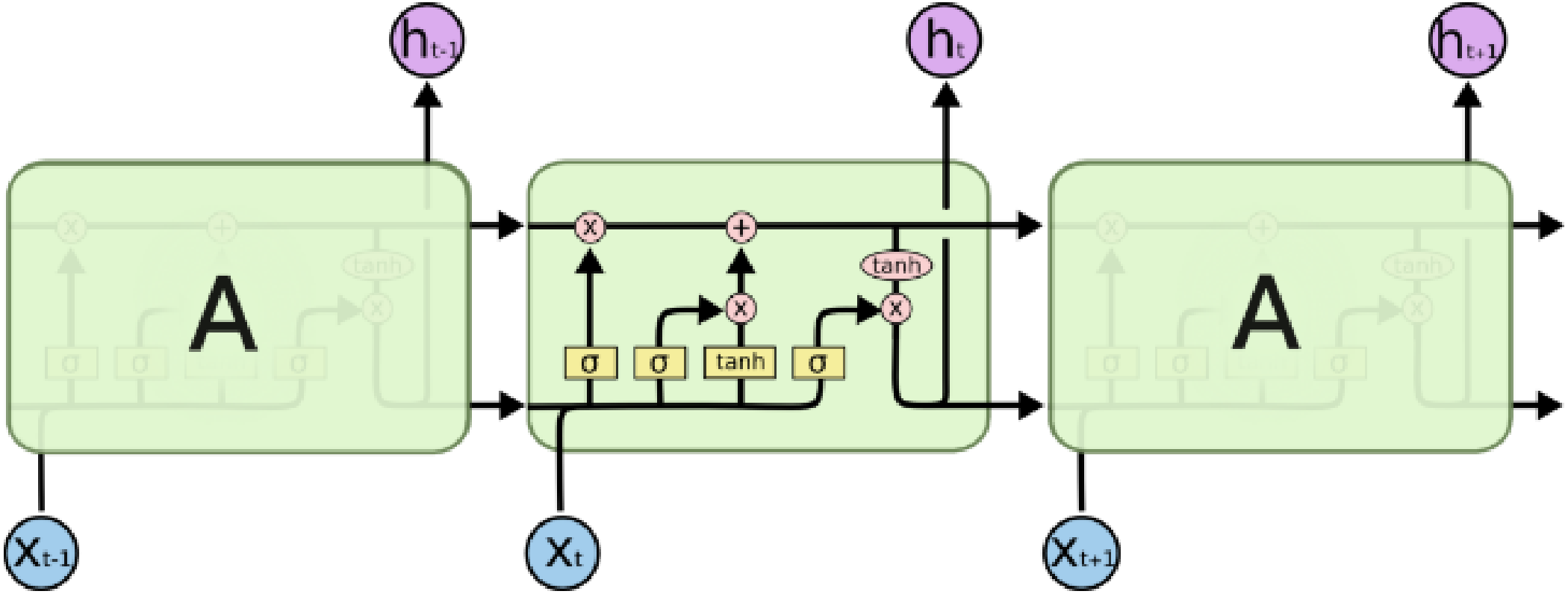
Sequences and Recurrent NN

- text, sound
- any information that is time dependent (daily revenues, stocks prices...)

Inspiration: brain contains information about past and is able to process the information given in sequences.

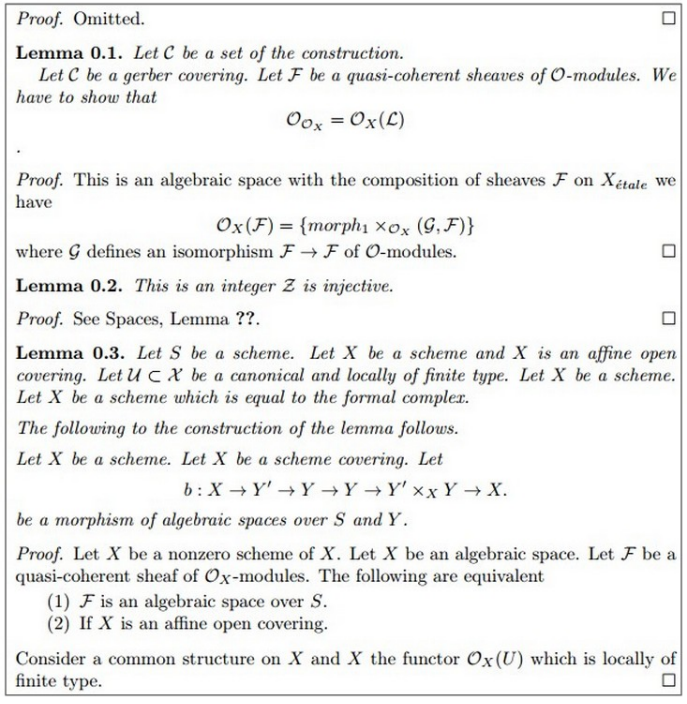


LSTM: a popular recurrent unit



Those are all tensor operations, see post for details

Character-level text generation



Proof. Omitted. □

Lemma 0.1. *Let $\mathcal{C}$ be a set of the construction.*
Let $\mathcal{C}$ be a gerber covering. Let $\mathcal{F}$ be a quasi-coherent sheaves of $\mathcal{O}$-modules. We have to show that

$$\mathcal{O}_{\mathcal{O}_X} = \mathcal{O}_X(\mathcal{L})$$

.

Proof. This is an algebraic space with the composition of sheaves $\mathcal{F}$ on $X_{\acute{e}tale}$ we have

$$\mathcal{O}_X(\mathcal{F}) = \{morph_1 \times_{\mathcal{O}_X} (\mathcal{G}, \mathcal{F})\}$$

where $\mathcal{G}$ defines an isomorphism $\mathcal{F} \to \mathcal{F}$ of $\mathcal{O}$-modules. □

Lemma 0.2. *This is an integer $\mathcal{Z}$ is injective.*

Proof. See Spaces, Lemma ??. □

Lemma 0.3. *Let S be a scheme. Let X be a scheme and X is an affine open covering. Let $\mathcal{U} \subset \mathcal{X}$ be a canonical and locally of finite type. Let X be a scheme. Let X be a scheme which is equal to the formal complex.*

The following to the construction of the lemma follows.

Let X be a scheme. Let X be a scheme covering. Let

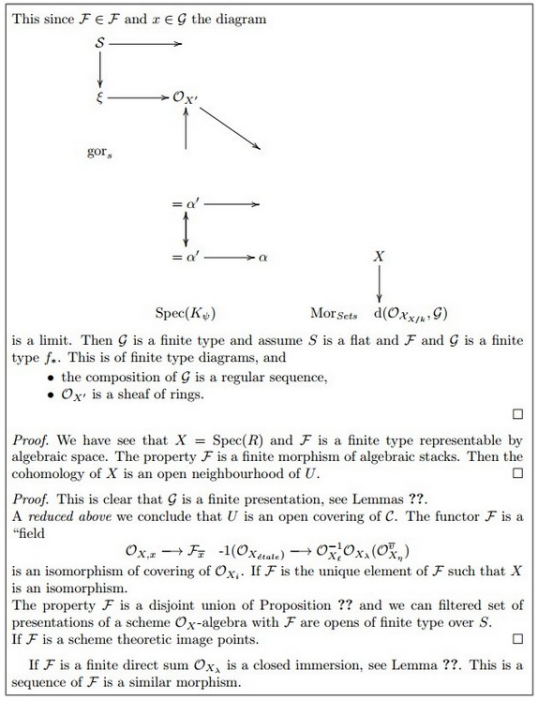
$$b : X \to Y' \to Y \to Y \to Y' \times_X Y \to X.$$

be a morphism of algebraic spaces over S and Y.

Proof. Let X be a nonzero scheme of X. Let X be an algebraic space. Let $\mathcal{F}$ be a quasi-coherent sheaf of $\mathcal{O}_X$-modules. The following are equivalent

(1) $\mathcal{F}$ is an algebraic space over S.
(2) If X is an affine open covering.

Consider a common structure on X and X the functor $\mathcal{O}_X(U)$ which is locally of finite type. □



This since $\mathcal{F} \in \mathcal{F}$ and $x \in \mathcal{G}$ the diagram

$$\begin{array}{ccc} \mathcal{S} & \longrightarrow & \\ \downarrow & & \\ \xi & \longrightarrow & \mathcal{O}_{X'} \\ & & \uparrow \\ \text{gor}_s & & \end{array} \qquad \begin{array}{ccc} = \alpha' & \longrightarrow & \\ \updownarrow & & \\ = \alpha' & \longrightarrow & \alpha \end{array} \qquad \begin{array}{c} X \\ \downarrow \\ \text{d}(\mathcal{O}_{X_{X/k}}, \mathcal{G}) \end{array}$$

$$\text{Spec}(K_\psi) \qquad \text{Mor}_{Sets}$$

is a limit. Then $\mathcal{G}$ is a finite type and assume S is a flat and $\mathcal{F}$ and $\mathcal{G}$ is a finite type f_*. This is of finite type diagrams, and

- the composition of $\mathcal{G}$ is a regular sequence,
- $\mathcal{O}_{X'}$ is a sheaf of rings.

□

Proof. We have see that $X = \text{Spec}(R)$ and $\mathcal{F}$ is a finite type representable by algebraic space. The property $\mathcal{F}$ is a finite morphism of algebraic stacks. Then the cohomology of X is an open neighbourhood of U. □

Proof. This is clear that $\mathcal{G}$ is a finite presentation, see Lemmas ??.
A *reduced above* we conclude that U is an open covering of $\mathcal{C}$. The functor $\mathcal{F}$ is a "field

$$\mathcal{O}_{X,x} \longrightarrow \mathcal{F}_{\overline{x}} \quad \text{-1}(\mathcal{O}_{X_{\acute{e}tale}}) \longrightarrow \mathcal{O}_{X_t}^{-1}\mathcal{O}_{X_\lambda}(\mathcal{O}_{X_\eta}^{\overline{w}})$$

is an isomorphism of covering of $\mathcal{O}_{X_t}$. If $\mathcal{F}$ is the unique element of $\mathcal{F}$ such that X is an isomorphism.
The property $\mathcal{F}$ is a disjoint union of Proposition ?? and we can filtered set of presentations of a scheme $\mathcal{O}_X$-algebra with $\mathcal{F}$ are opens of finite type over S.
If $\mathcal{F}$ is a scheme theoretic image points. □

If $\mathcal{F}$ is a finite direct sum $\mathcal{O}_{X_\lambda}$ is a closed immersion, see Lemma ??. This is a sequence of $\mathcal{F}$ is a similar morphism.

Given a text, network predicts next character. Then this nework is used to generate a text character-by-character.

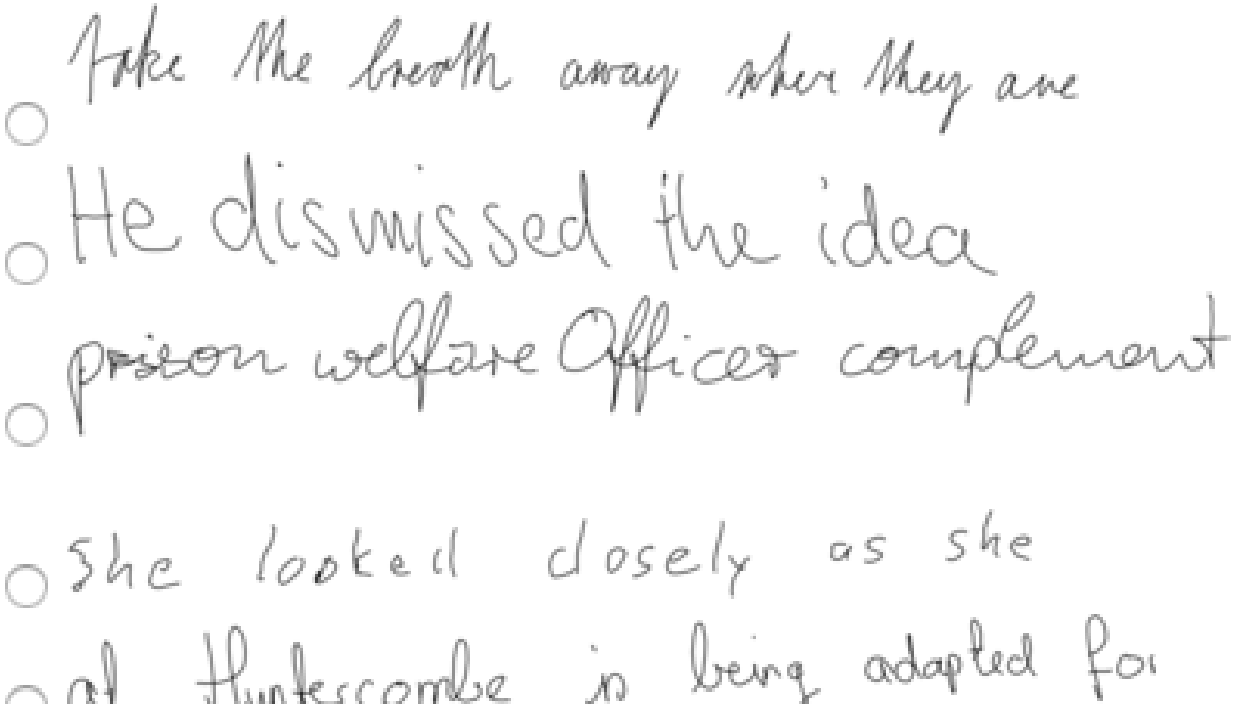
More examples

Code generation

Network trained on the linux source:

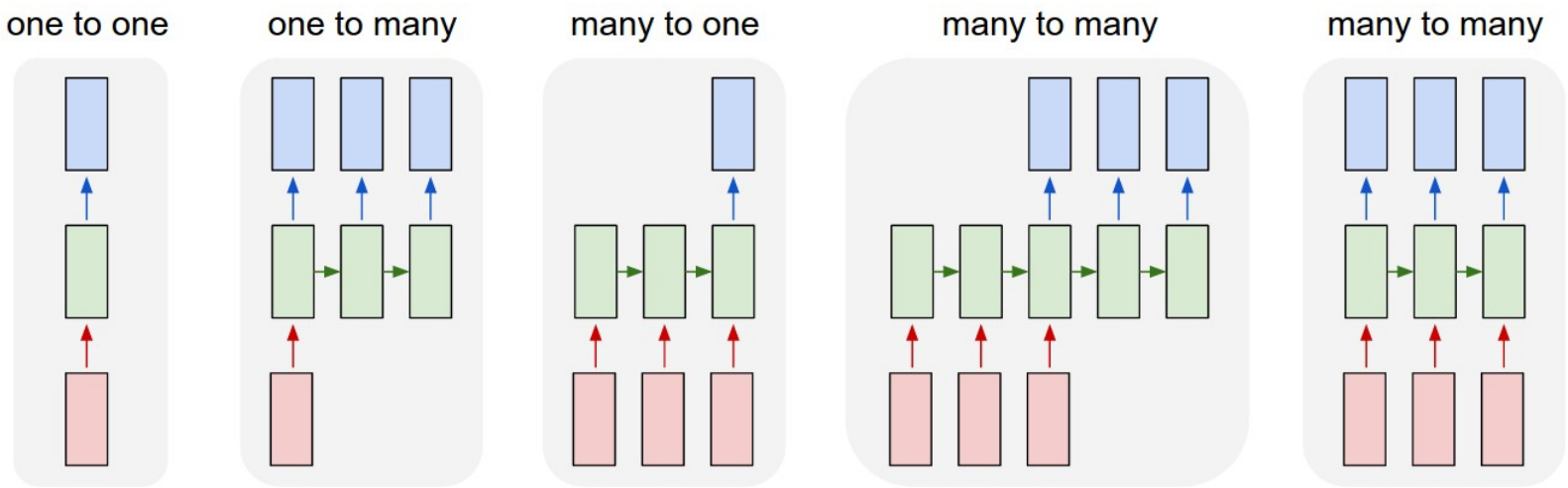
```
/*
 * Increment the size file of the new incorrect UI_FILTER group information
 * of the size generatively.
 */
static int indicate_policy(void)
{
  int error;
  if (fd == MARN_EPT) {
    /*
     * The kernel blank will coeld it to userspace.
     */
    if (ss->segment < mem_total)
      unblock_graph_and_set_blocked();
    else
      ret = 1;
    goto bail;
  }
  segaddr = in_SB(in.addr);
  selector = seg / 16;
  setup_works = true;
  for (i = 0; i < blocks; i++) {
    seq = buf[i++];
    bpf = bd->bd.next + i * search;
    if (fd) {
      current = blocked;
    }
  }
  rw->name = "Getjbbregs";
  bprm_self_clearl(&iv->version);
  regs->new = blocks[(BPF_STATS << info->historidac)] | PFMR_CLOBATHINC_SECONDS << 12;
  return segtable;
}
```

Style in RNNs



Position of the pen in the next moment is predicted.

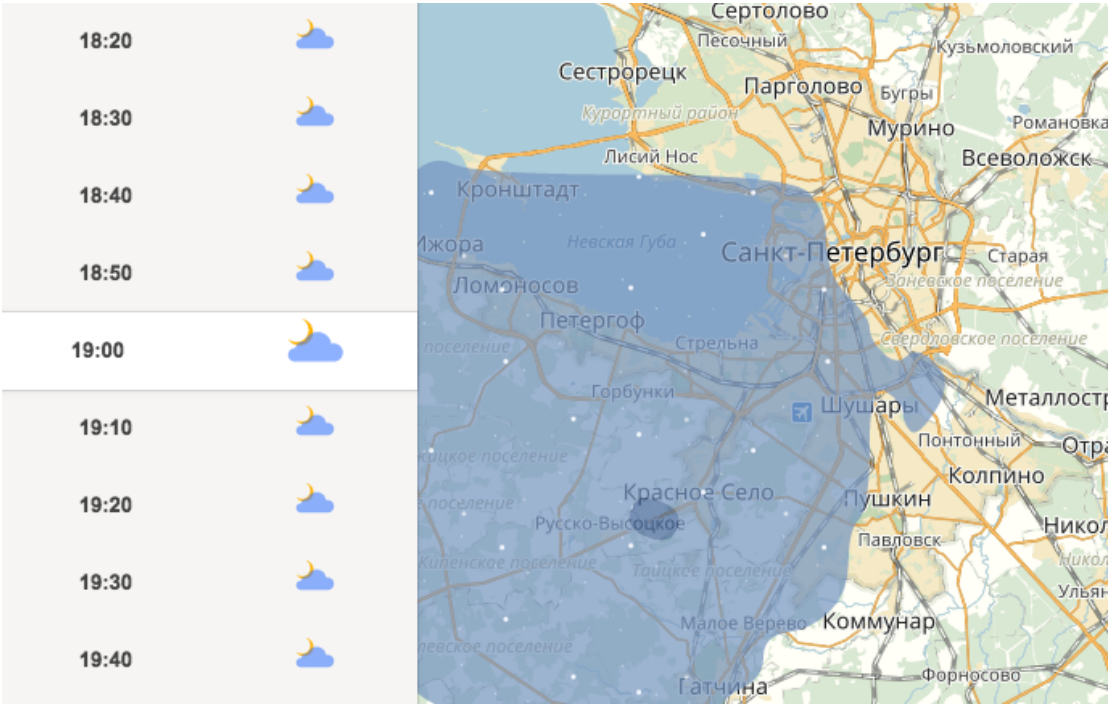
Different strategies for prediction



- one can predict a single observation from sequence
- or map sequence to sequence

Mixing RNN and CNN

Sometimes you need both to analyze dependencies in space and time. For example, in weather forecasting a sequence of photoes from sattelites [can be used](#) to predict snow/rain



Why looking at sequence of images?

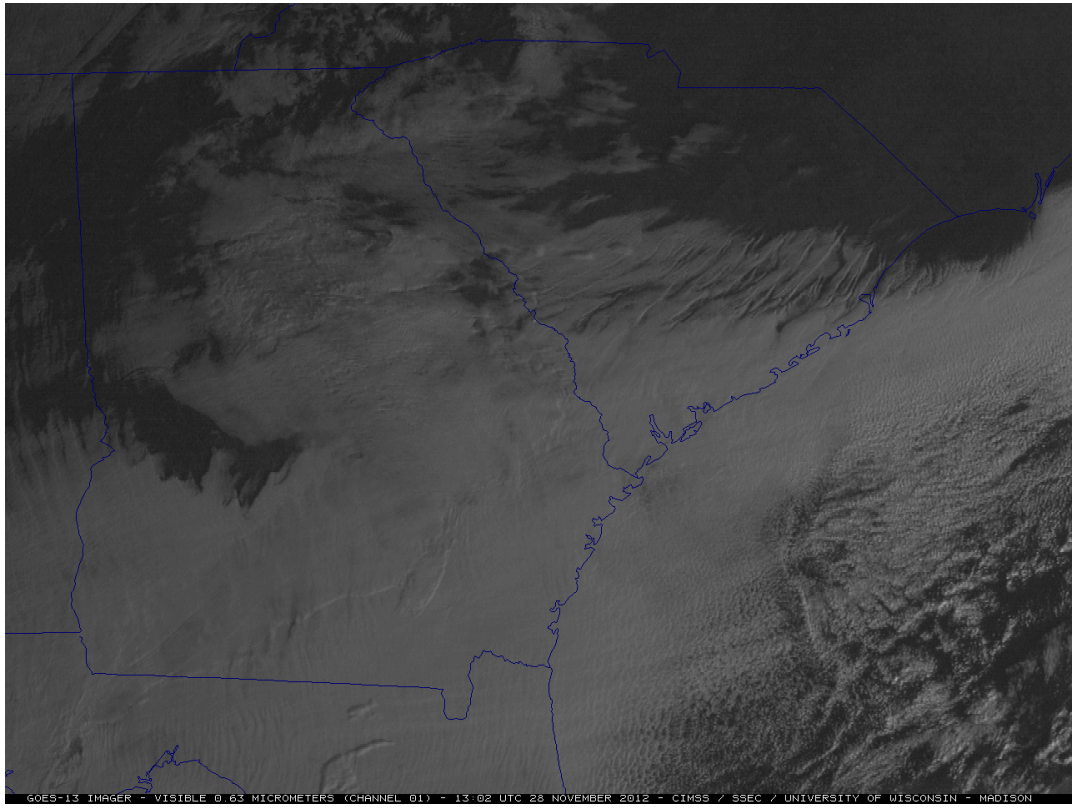


Image taken from CIMSS Satellite Blog

Instead of conclusion

- NNs are very popular today and are a subject of intensive research
- differentiable operations with tensors provide a very good basis for defining useful predicting models
- Structure of a problem can be effectvely used in structure of the model
- DL requires much data and many resources that became available recently (and also numerous quite simple tricks)
- NNs are awesome, but not guaranteed to work well on problems without specific structure

The end!