Introduction to Robotics for cognitive science

Dr. Andrej Lúčny

KAI FMFI UK

lucny@fmph.uniba.sk

Web page of the subject

www.agentspace.org/kv

Emotions

- various list of some global states of mind (7 – 40)

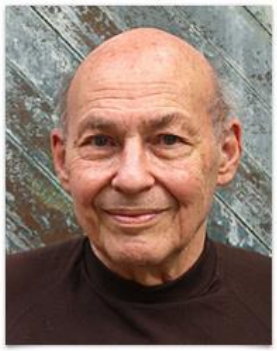
Plutchik's theory

Fear → feeling of being afraid, frightened, scared.
Anger → feeling angry. A stronger word for anger is rage
Sadness → feeling sad. Other words are sorrow, grief
Joy → feeling happy. Other words are happiness, gladness
Disgust → feeling something is wrong or nasty. Strong disapproval.
Surprise → being unprepared for something.
Trust → a positive emotion; admiration is stronger; acceptance is weaker.
Anticipation → in the sense of looking forward positively to something which is going to happen. Expectation is more neutral

Emotions

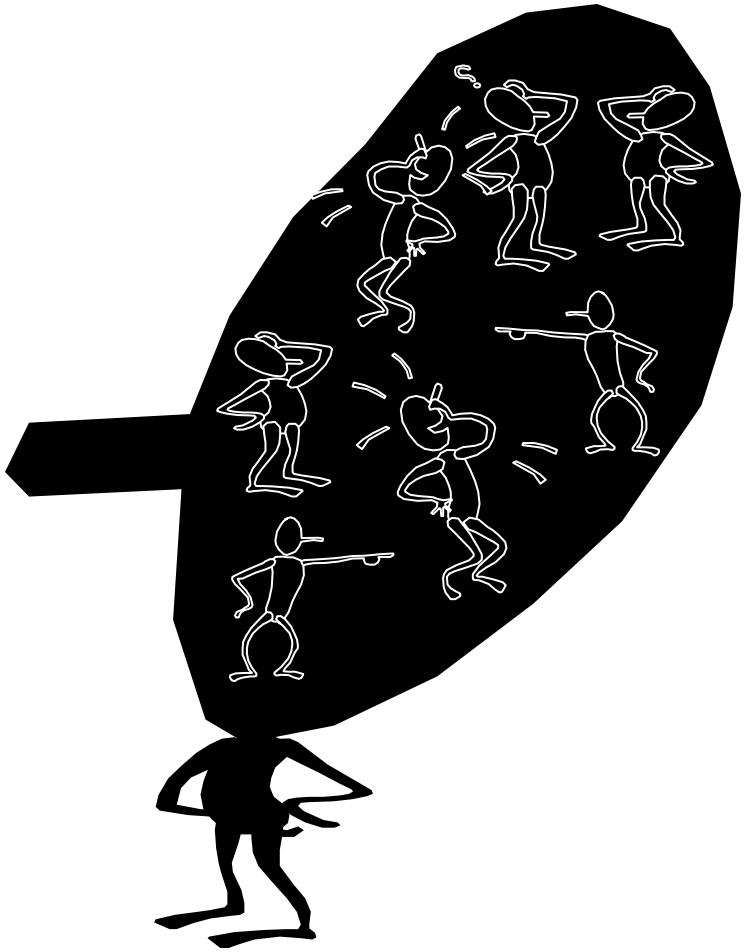
- Important tool for human-robot interaction
- But not only!
- Many theories claim they are important also as significant part of the control mechanism of any robot which should look like to be alive

Minsky's model "Society of Mind"

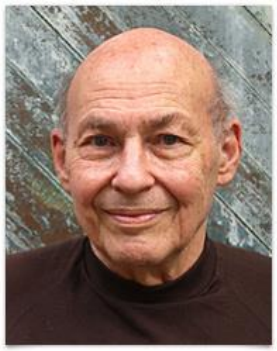


[Minsky 1985]

- System containing many parallel modules (agents, resources)
- Control = activation of a proper set of modules at a proper situation



Emotion Machine



[Minsky 2006]

- Emotions are evolutionary older than anything else in brain, it is residuum from an early control system of our predecessors
- Emotions can be modelled in computer as global parameters of all processes in the control system

Example: Freud theory of "Pleasure pump"

- We can express the theory as a process provided by several agents and model them in an implementation of Minsky mind model (like the Agent-Space Architecture)
- Then we can apply this control structure on any creature, even on the robot with two-wheel gear
- [Kelemen, Lucny 2012]

Freud's theory of "Pleasure Pump"

- Theory describing logic and phases of "courting couple" act putting focus on emotions

1. partner perception
2. activation of emotional representation
3. hormones production
4. emotional tension
5. action stimulation
6. direct act
7. tension release
8. discharge leading to sense of pleasure

Freud's theory of "Pleasure Pump"

- Is it correct ?
- Is it complete ?

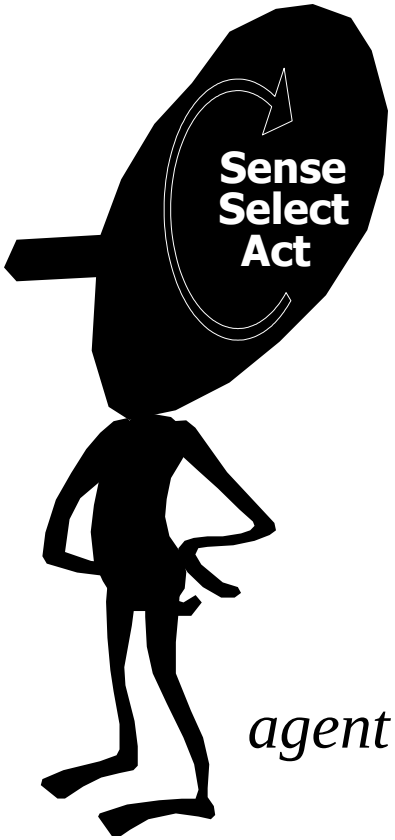
- Can we use it for production of particular behavior in an abstracted situation in virtual reality which looks like "coutring couple" ?
- Are we able to use psychological theory as insipiration for implementation in computer ?

Implementation of Minsky's model

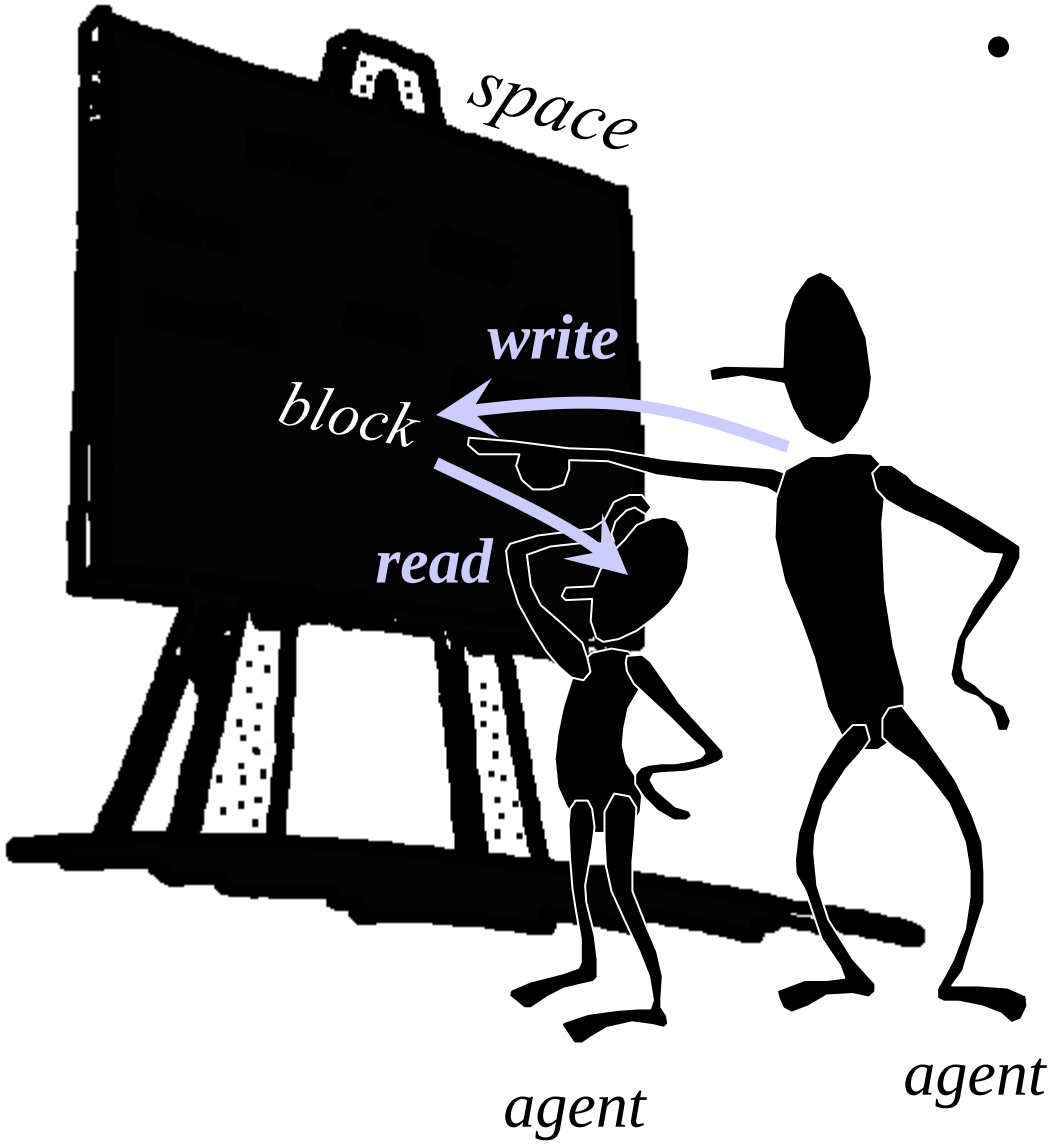
- Agent-Space architecture [Lucny 2004], www.agentspace.org
- agents can be implemented as objects equipped with an own thread of control and a mechanism of a mutual data exchange including sensation and action of the system environment

Nature of agents

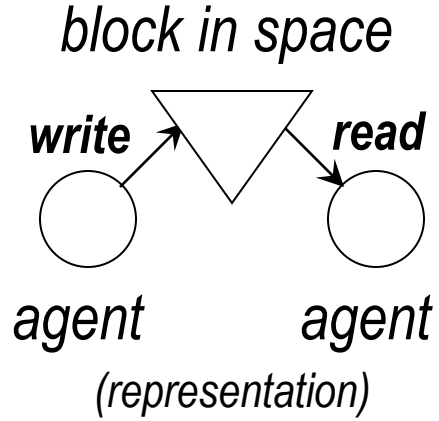
- *each agent is endlessly running a sense-select-act cycle. Any course through this cycle calculates some actions upon information sensed from environment or provided by other agents*



Communication among agents

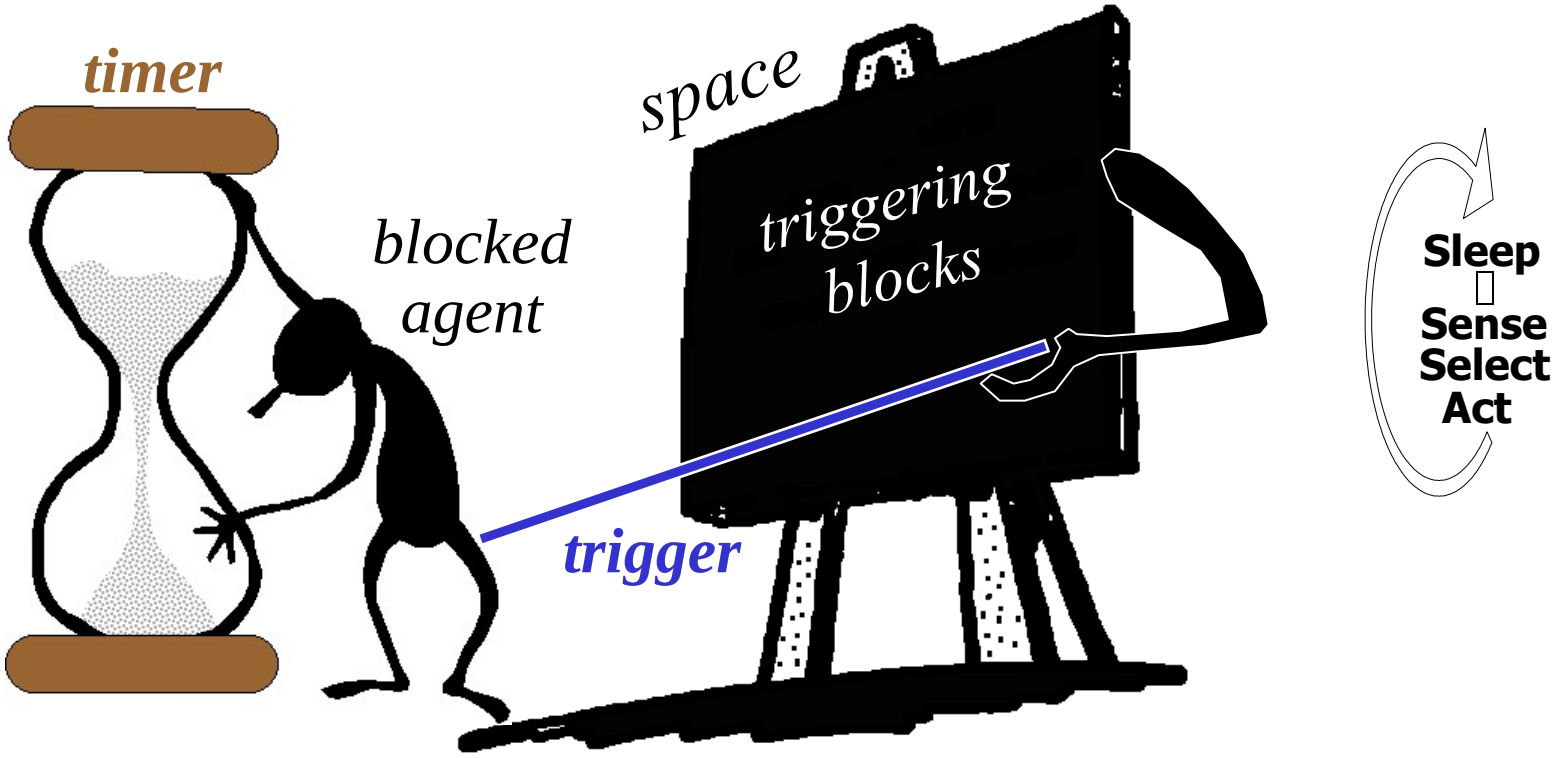


- *indirect communication through sophisticated blackboard (called space)*

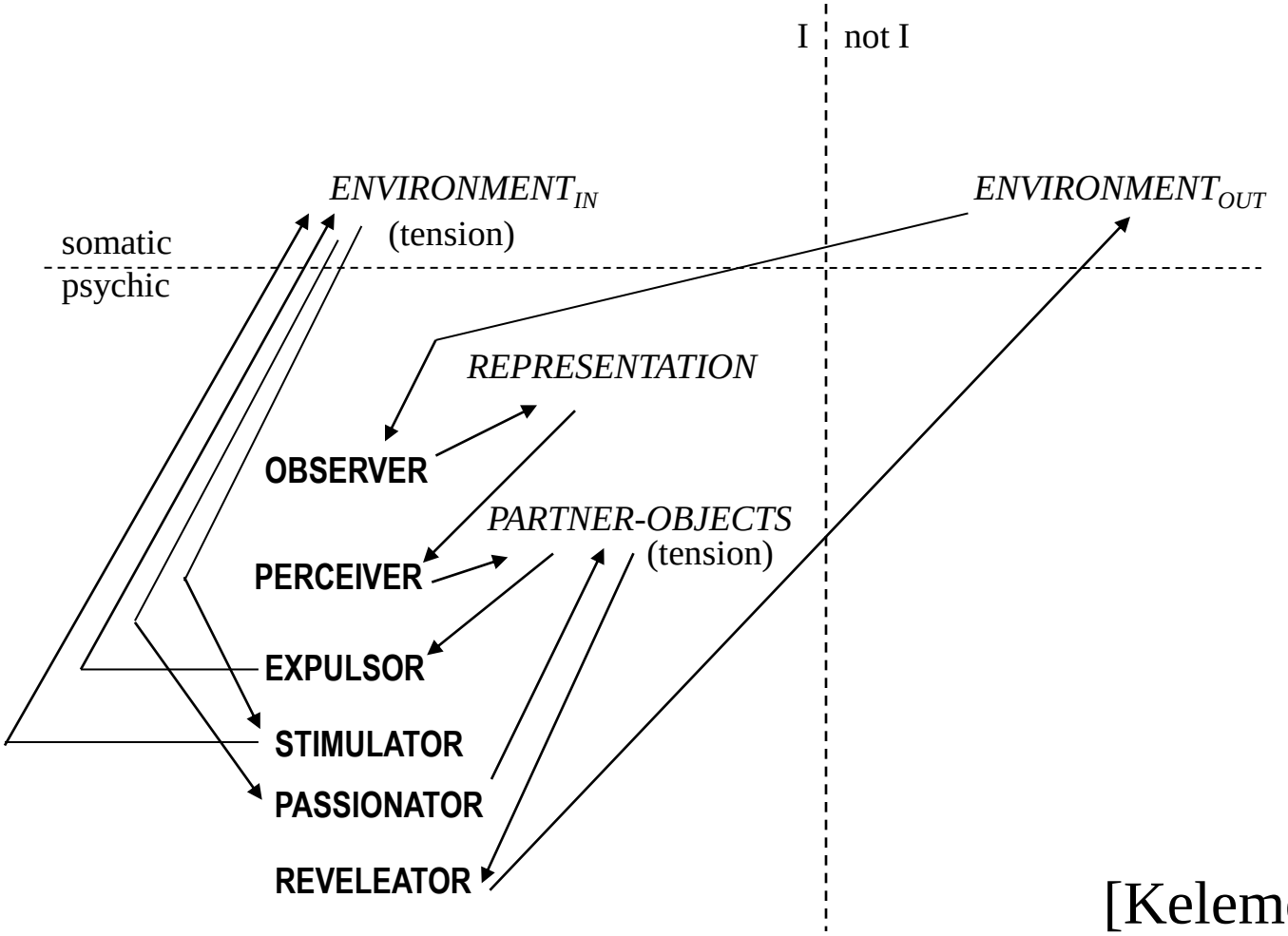


Real-time operation

- *timers and triggers*

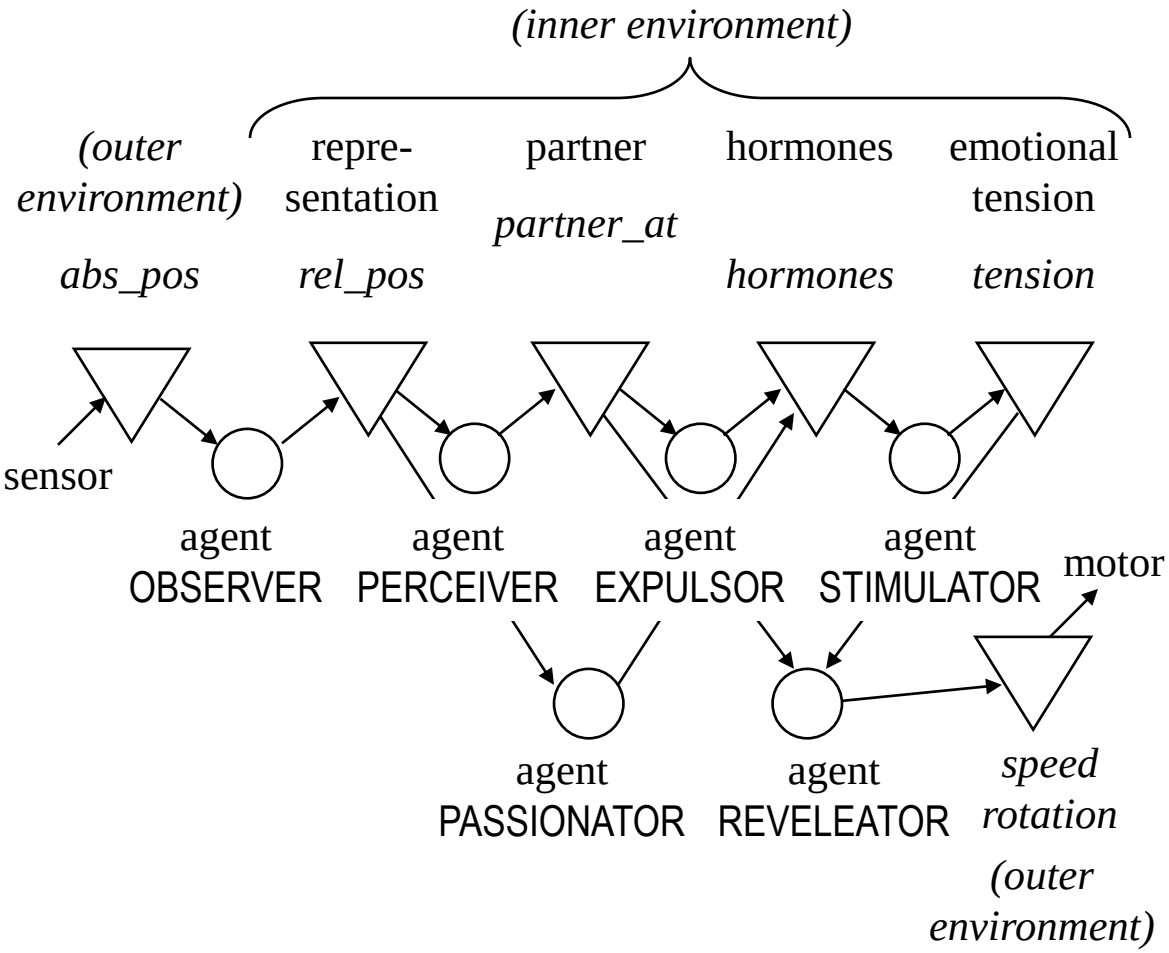


Design of “Pleasure Pump” model

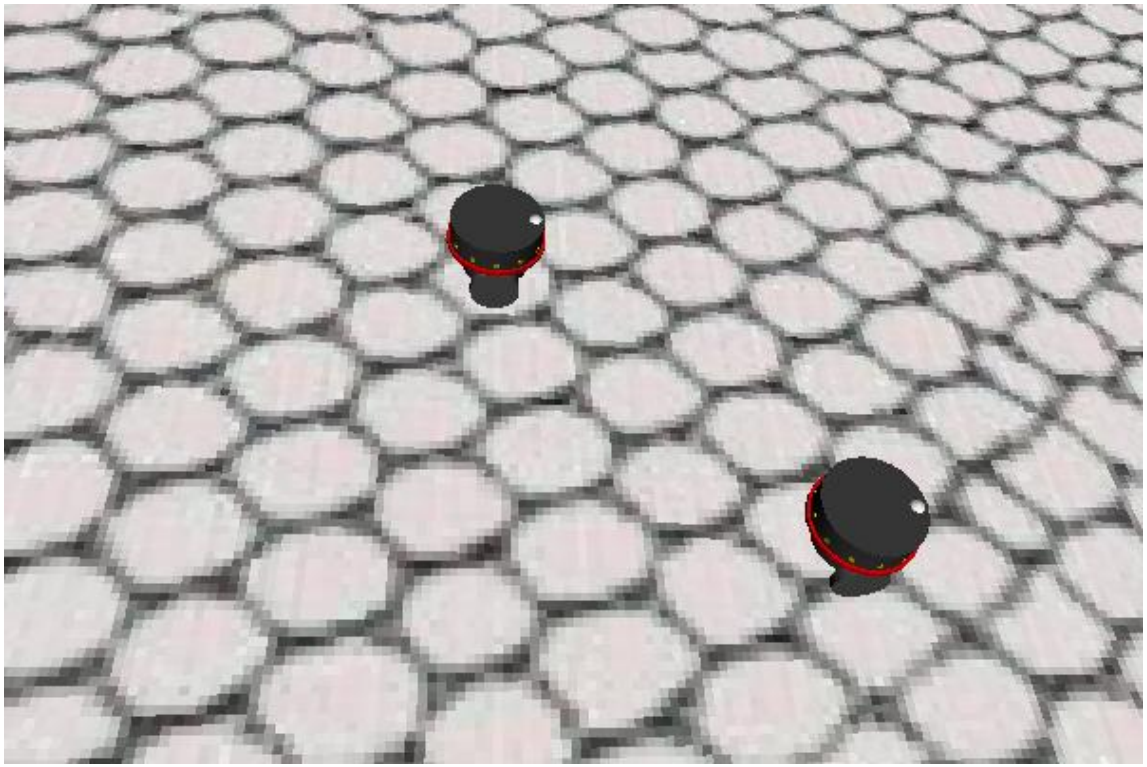


[Kelemen 2011]

Implementation of the model



Used technologies

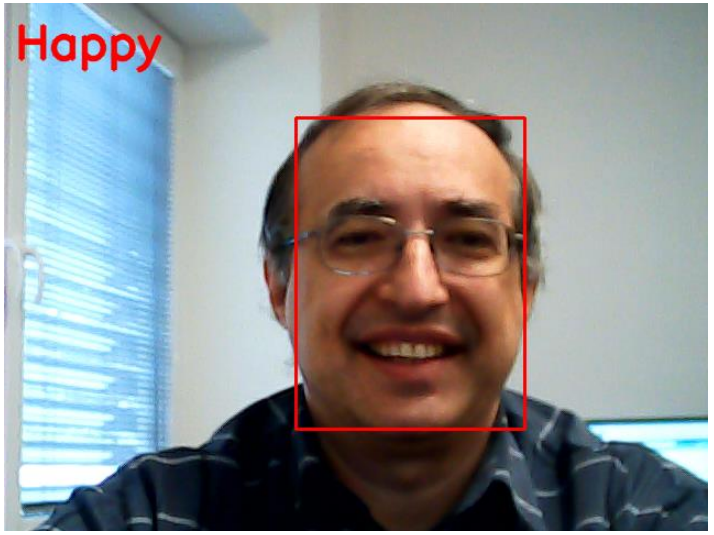


https://youtu.be/8-A16BVjKIc
www.agentspace.org/download/Pleasure Pump-ver3.zip

Facial expressions of emotions

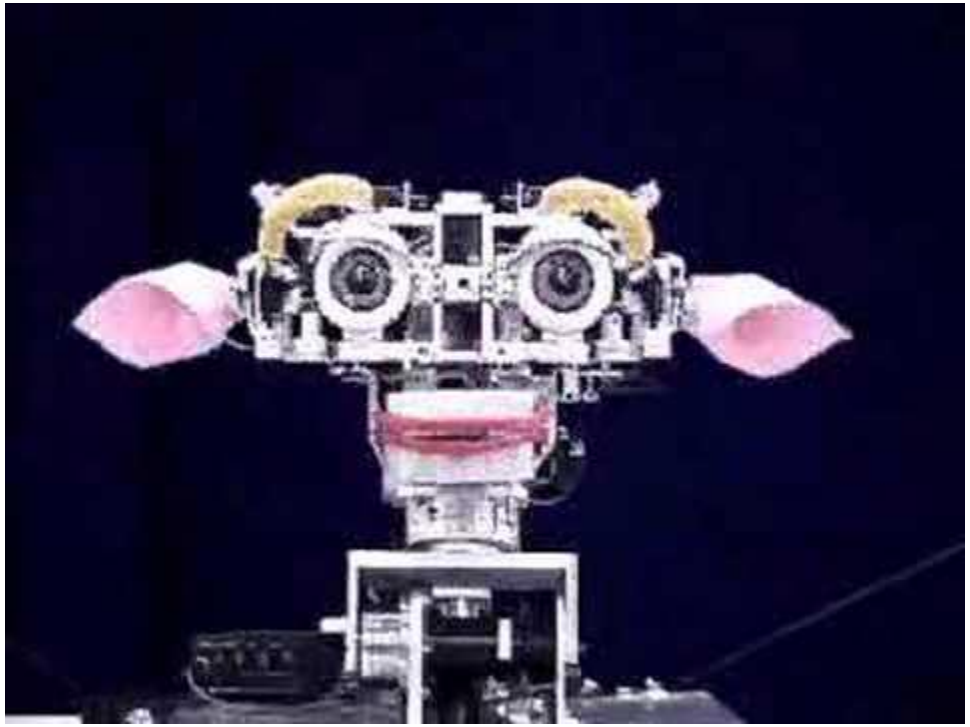
- Generation of facial expressions of emotions (easy)

- Recognition of facial expressions of emotions (difficult)



Generation of facial expressions of emotions

- KISMET [Breazeal 1990]



Furhat, 2018

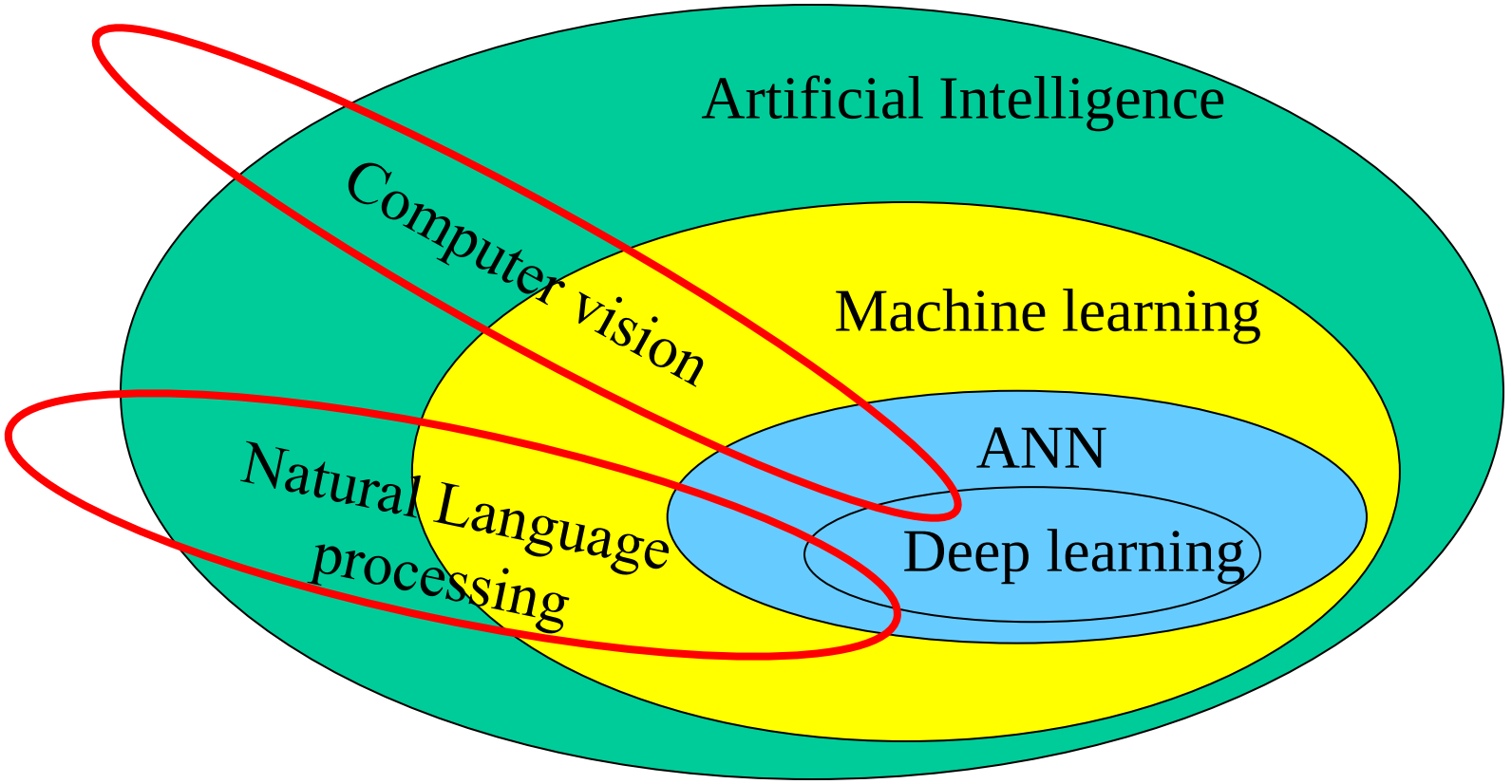
Recognition of the Facial Expressions

- Older approach:
  - Face detector (Haar or HOG)
  - Facial Landmark detector (Cascade regressor)
  - LDA or SVM

- Modern approach:
  - Deep learning

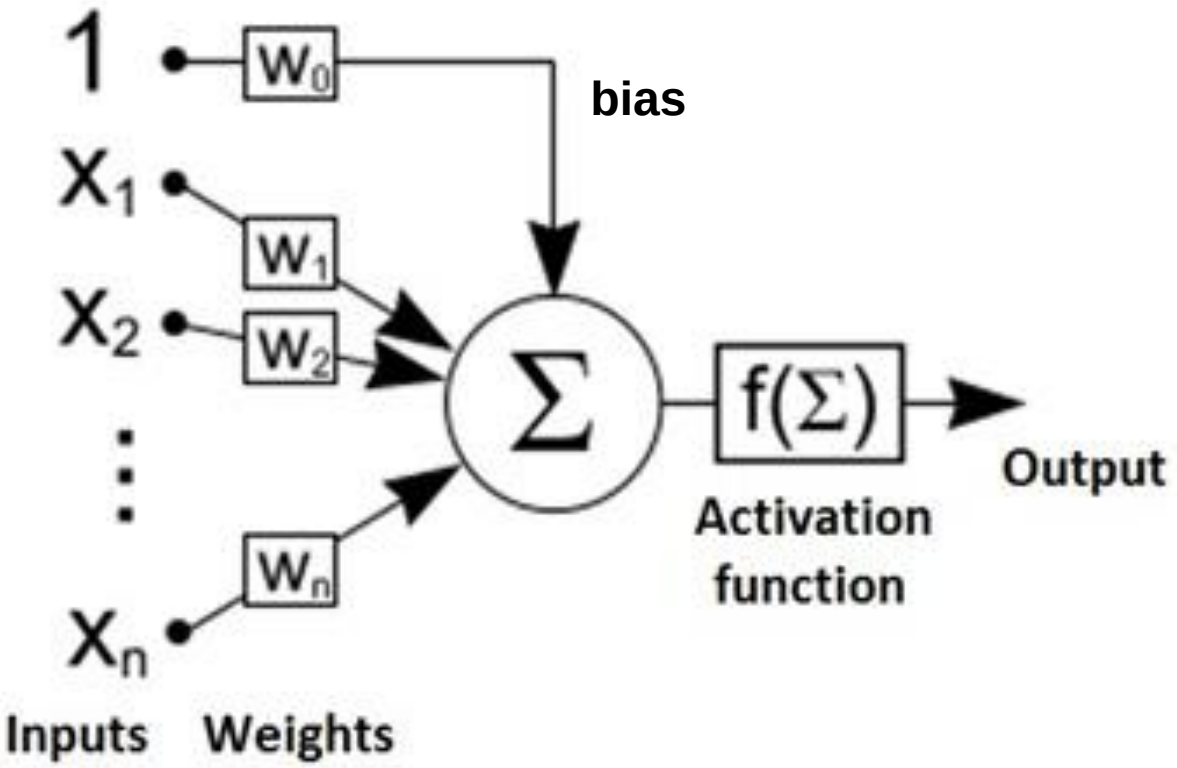
[Alex Krizhevsky, 2012]

Deep Learning



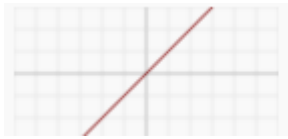
ANN … Artificial Neural Networks

Neuron


1
x_1
x_2
x_n
w_0
w_1
w_2
w_n
bias
Σ
$f(\Sigma)$
Output
Activation function
Inputs
Weights

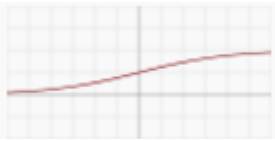
Activation functions

linear



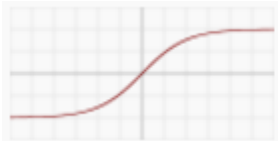
$$f(x) = x$$

sigmoid



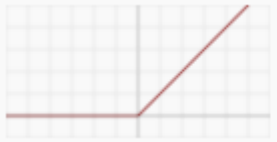
$$f(x) = \frac{1}{1 + e^{-x}}$$

tanh



$$f(x) = \tanh(x) = \frac{2}{1 + e^{-2x}} - 1$$

relu

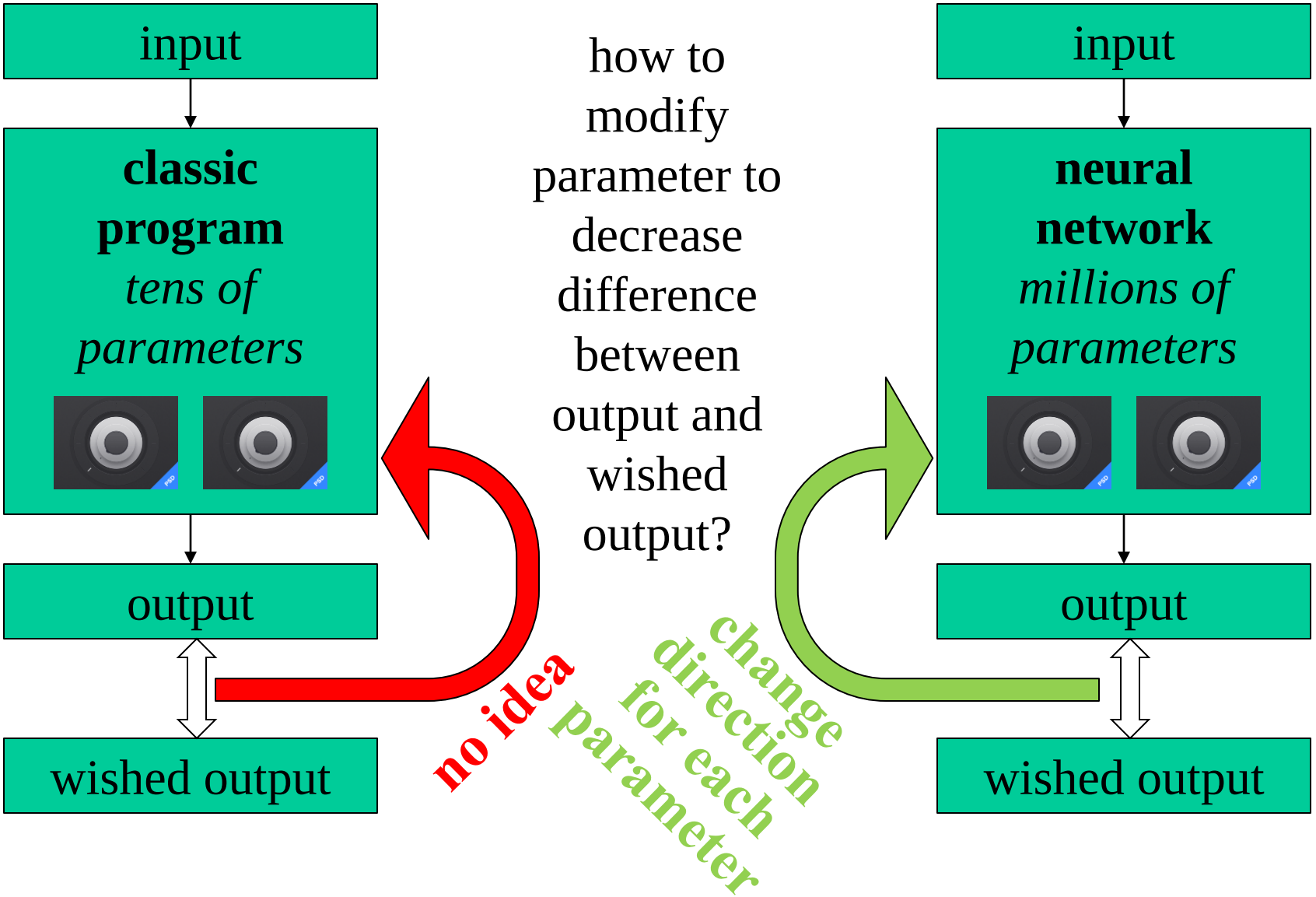


$$f(x) = \begin{cases} 0 & \text{for} \quad x < 0 \\ x & \text{for} \quad x \geq 0 \end{cases}$$

softmax

$$\sigma(z)_j = \frac{e^{z_j}}{\sum_{k=1}^{K} e^{z_k}}$$

What is neural network



Building blocks of neural network

- anything corresponding to function which can be expressed and derivated via symbolic way

Then we can define error function, typicaly sum of squares of differences between output and wished output for all samples and calculate its partial derivatives by indivual parameters

Value of the partial derivative for the current values of parameters gives the direction in which we need to modify the parameter to decrease value of the error function

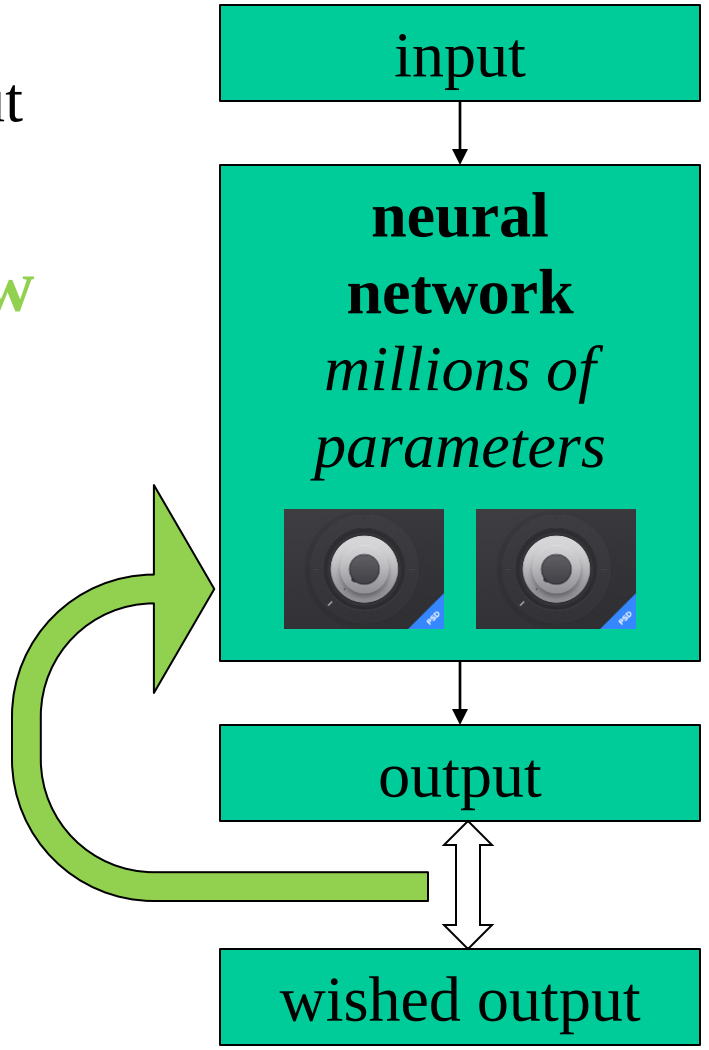
Training (Learning)

how to modify parameter to decrease difference between output and wished output?

for each parameter we know correct direction, even we know that one parameter needs to be modified more than other

but we do not know how much

so we need to guess and try and return back and that is training or learning process



Training algorithms

According from how many samples we derive gradient

- Gradient Descent
- Stochastic Gradient Descent
- Batch Gradient Descent
- Minibatch Gradient Descent

According how we estimate suitable multiple of the opposite gradient vector:

- rmsprop
- ADAM

Simple example of Convolution NN

vertical edges detector

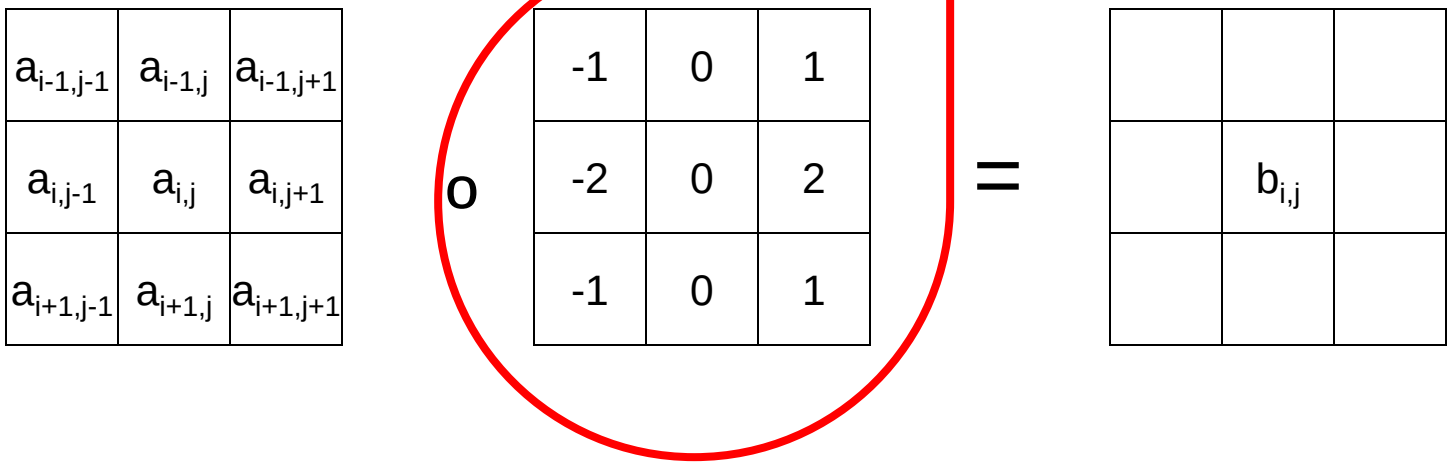
a

b

Vertical edges can be provided by:

kernel 3x3

Sobel kernel



$$b_{i,j} = | a_{i-1,j+1} + 2a_{i,j+1} + a_{i+1,j+1} - a_{i-1,j-1} - 2a_{i,j-1} - a_{i+1,j-1} |$$

Implementation via classic program

```
import cv2
import numpy as np

kernel = np.array([[-1,0,1],[-2,0,2],[-1,0,1]])

output_image = cv2.filter2D(input_image,-1,kernel)
```

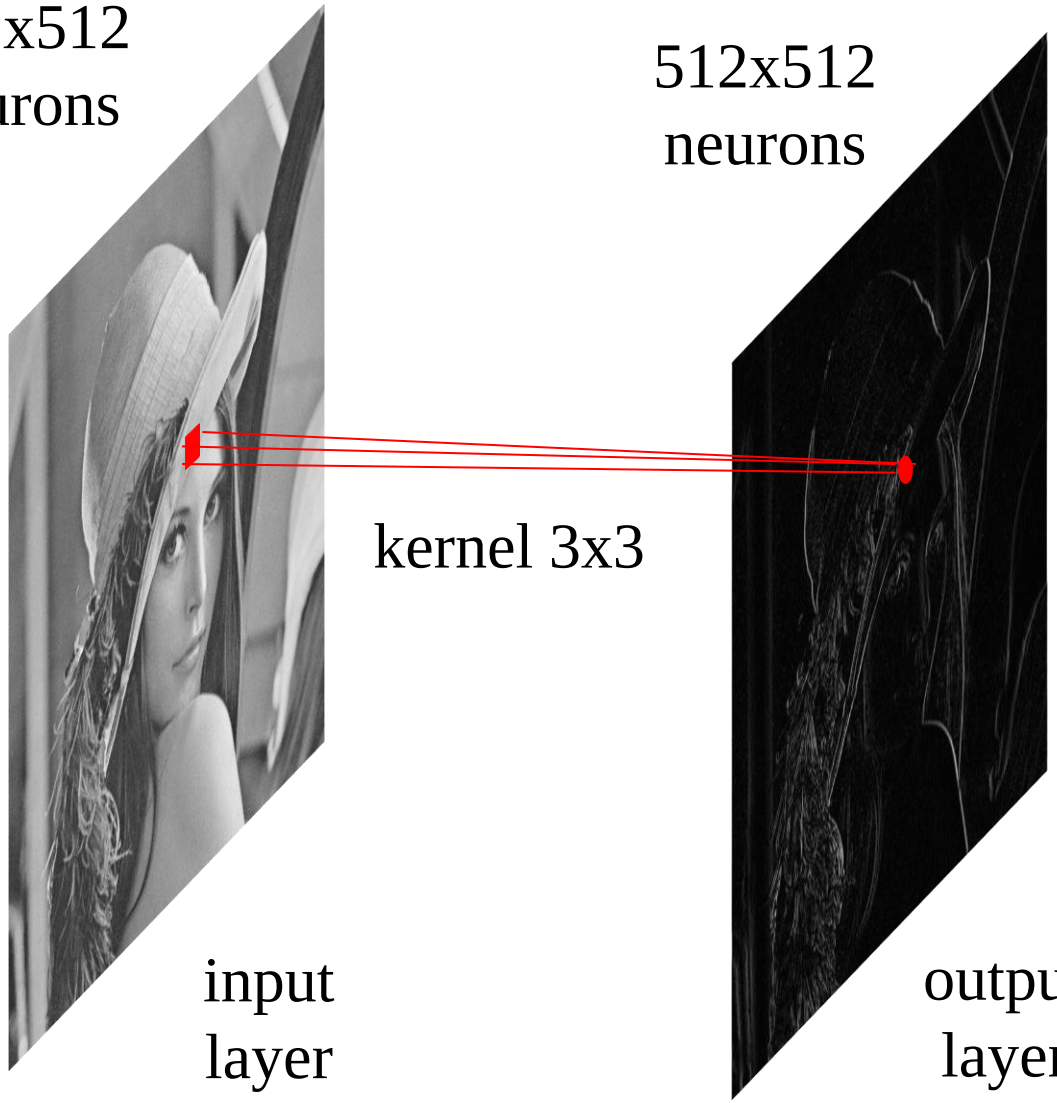
Implementation by neural network in Keras

Such simple task can be realized by single building block –

- by one layer with one convolutional neural network (CNN)

- Contains various building blocks of neural neworks and various training algorithms
- is Python front-end for Deep learning over back-end TensorFlow (Theano, MxNet)
- Keras – Francois Chollet, projekt ONEIROS, 2015

Convolutional neural network



Each neuron of the output layer has connections to 3x3 = 9 neurons in the input layer

Each neuron has the same weights on the connections, i.e. this CNN has just 9 parameters

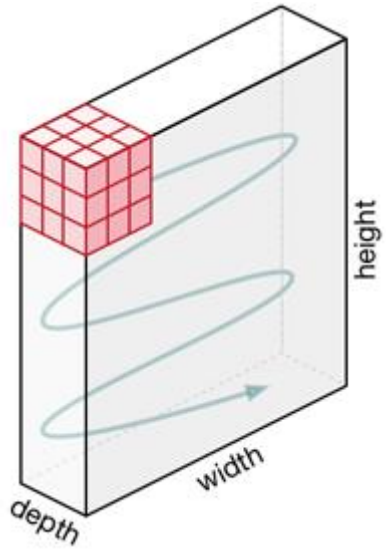
Implemention in Keras

```
import keras

inp = Input(shape=(None,None,1))
out = Conv2D(1, (3, 3), kernel_initializer="normal",
       use_bias=False, padding="same")(inp)
model = Model(inputs=inp, outputs=out)

w = np.array([[ [[-1]],[[0]],[[1]]], [[-2]],[[0]],[[2]]],
       [[-1]],[[0]],[[1]]] ])
model.layers[1].set_weights(w)

input_images = np.array([input_image.reshape(...)])
output_images = model.predict(input_images)
output_image = output_images[0].reshape(...)
```

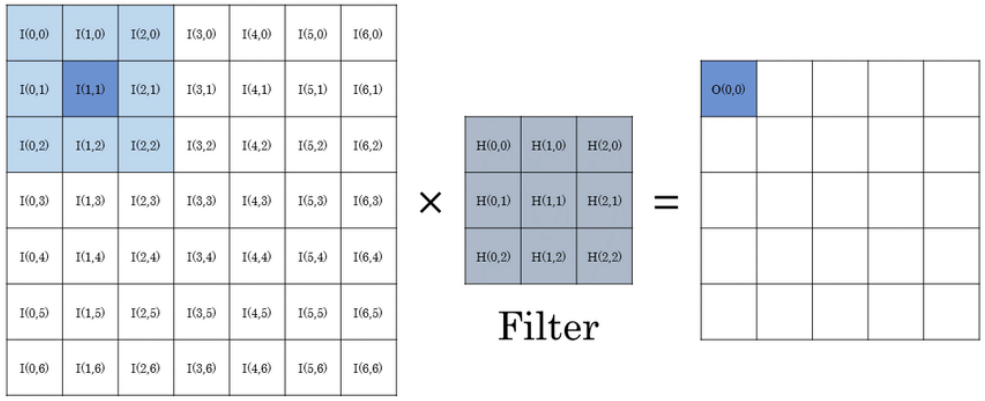
Training in Keras

```
# employ input_images, output_images and w from the
# previous example

inp = Input(shape=(None,None,1))
out = Conv2D(1, (3, 3), kernel_initializer="normal",
       use_bias=False, padding="same")(inp)
model = Model(inputs=inp, outputs=out) # the same model

model.compile(optimizer='rmsprop',loss='mse',metrics=[‘acc'])
model.fit(input_images, output_images, batch_size=1, epochs =
15000)

model.layers[1].get_weights() # is approximately w now

model.predict(input_images) # returns edges
```

Tensor



Kernel



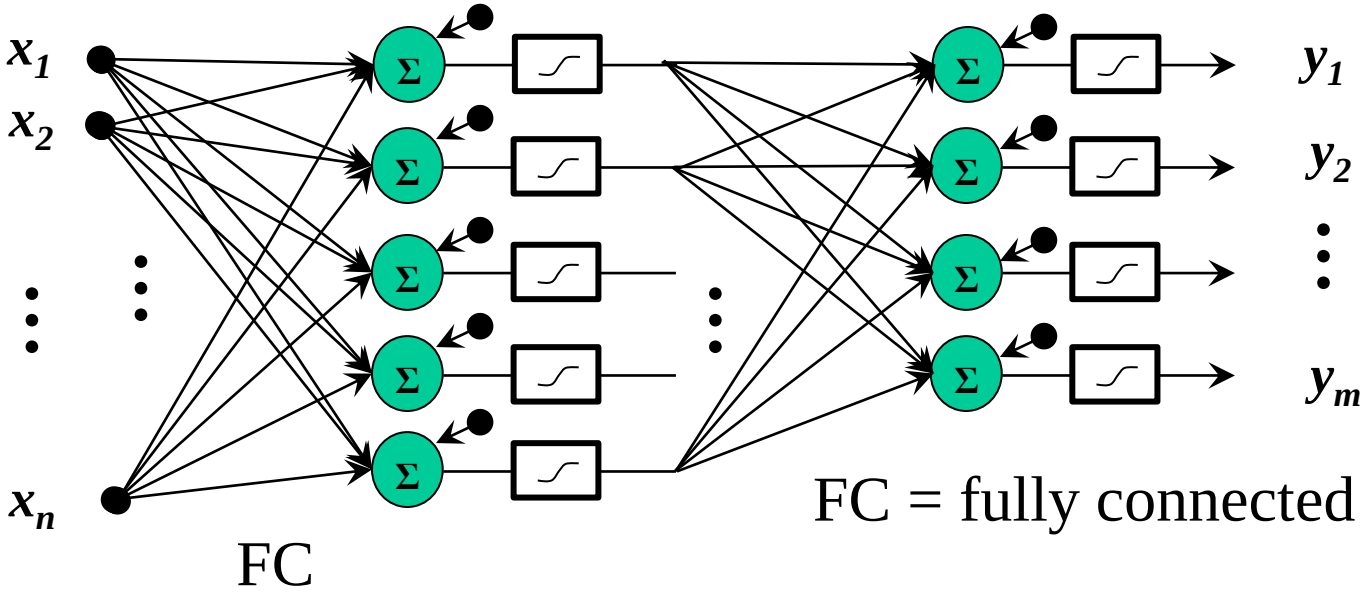
Input image

Filter

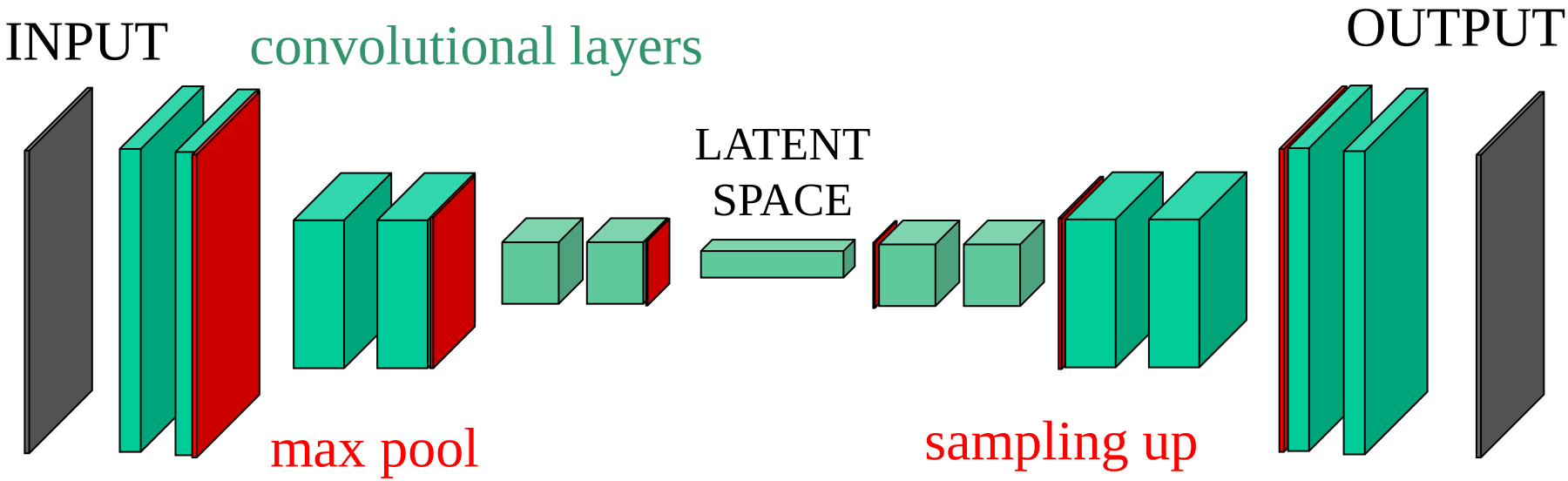
Output image

Perceptron

- Two or more fully connected layers is universal approximator [Cybenko 1989]
- Theoretically it is a very strong machine, but in practice it is not very useful

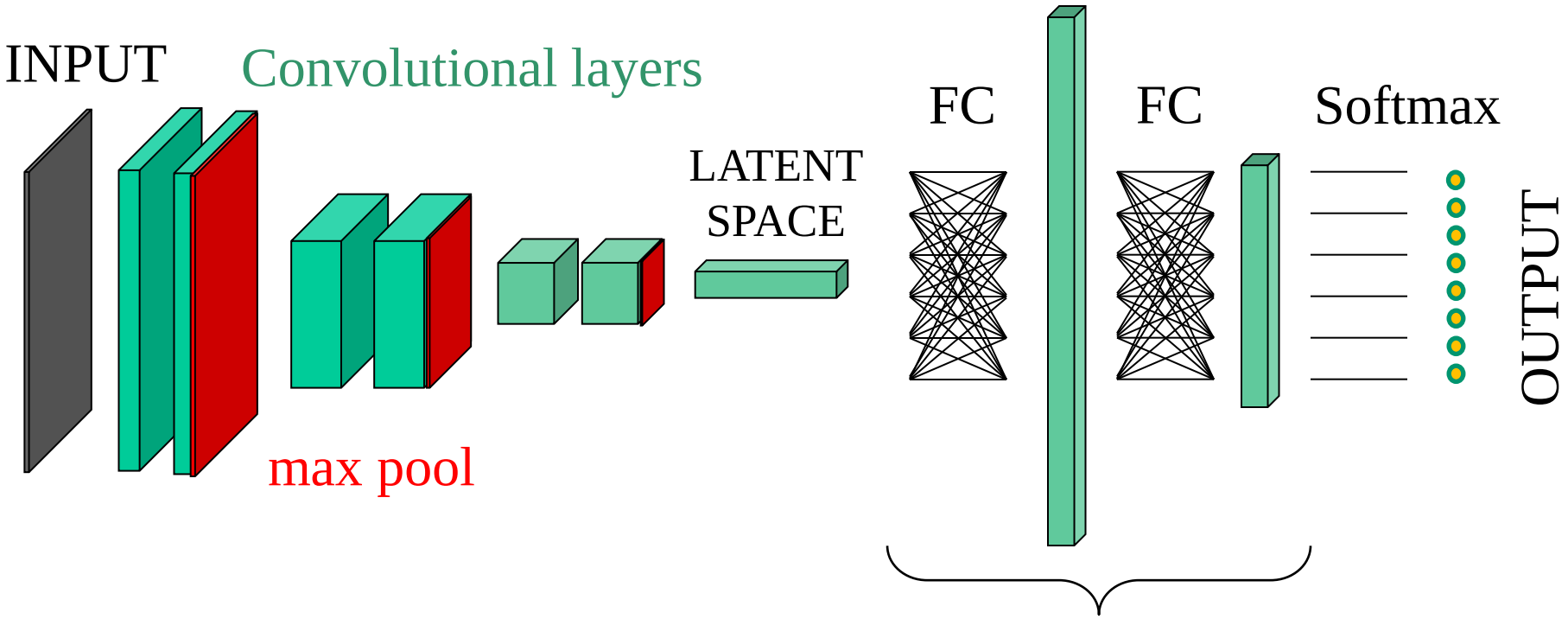


(Convolutional) autoencoder



Encoder (the first half of autoencoder) transforms image to features (point in the latent space) ...

Classifier



... and for features perceptron works also in practice

Emotion recognition

- Having dataset

- We are able train deep model which out probabilities of individual emotions
- Of course, prior to emotion recognition we need to localize face on image, that can be provided by another deep model (detector)