Deep Learning in HCI

CHI course computational interaction

Otmar Hilliges

30 August 2018

Advanced Interactive Technologies

Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

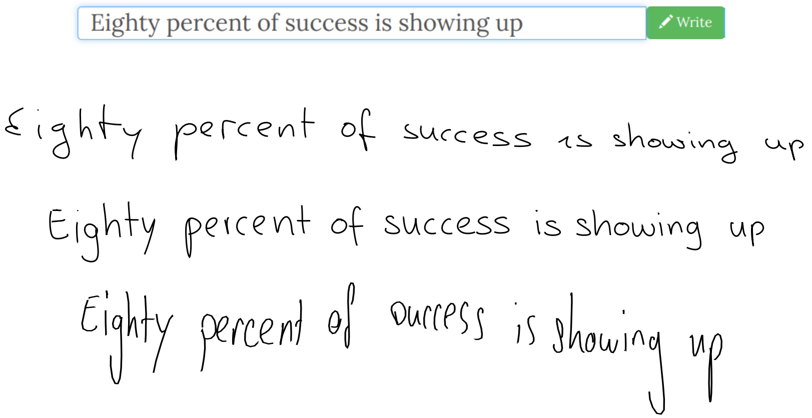
Course Materials

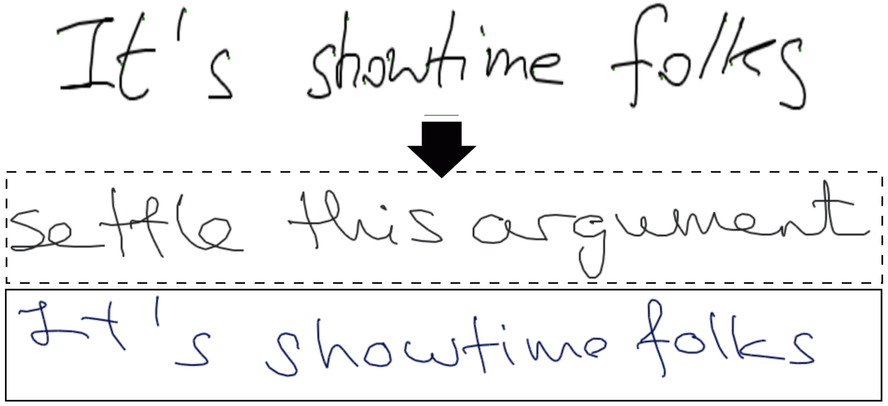
In course exercise:

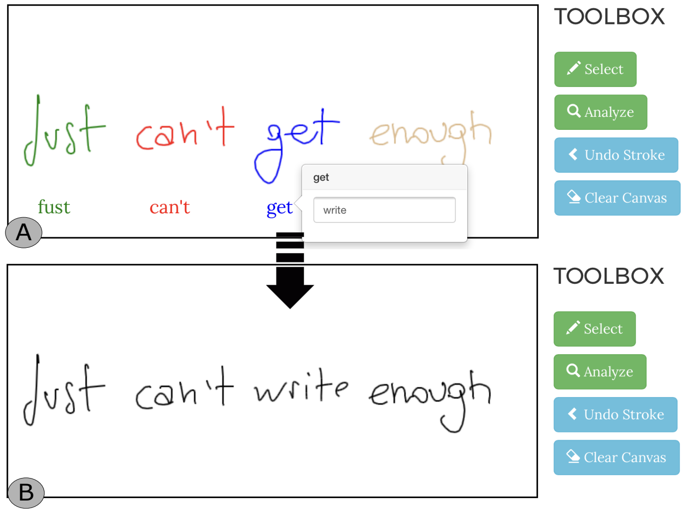
In the Github repo

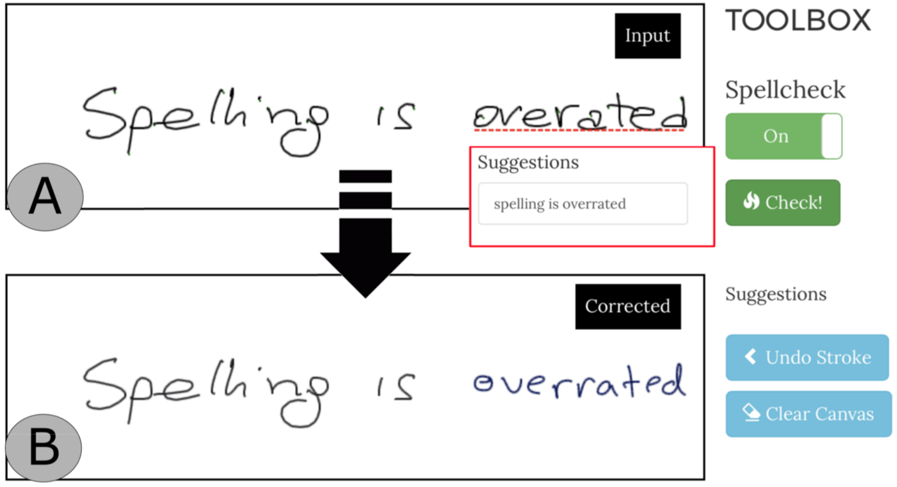
Take home exercise:

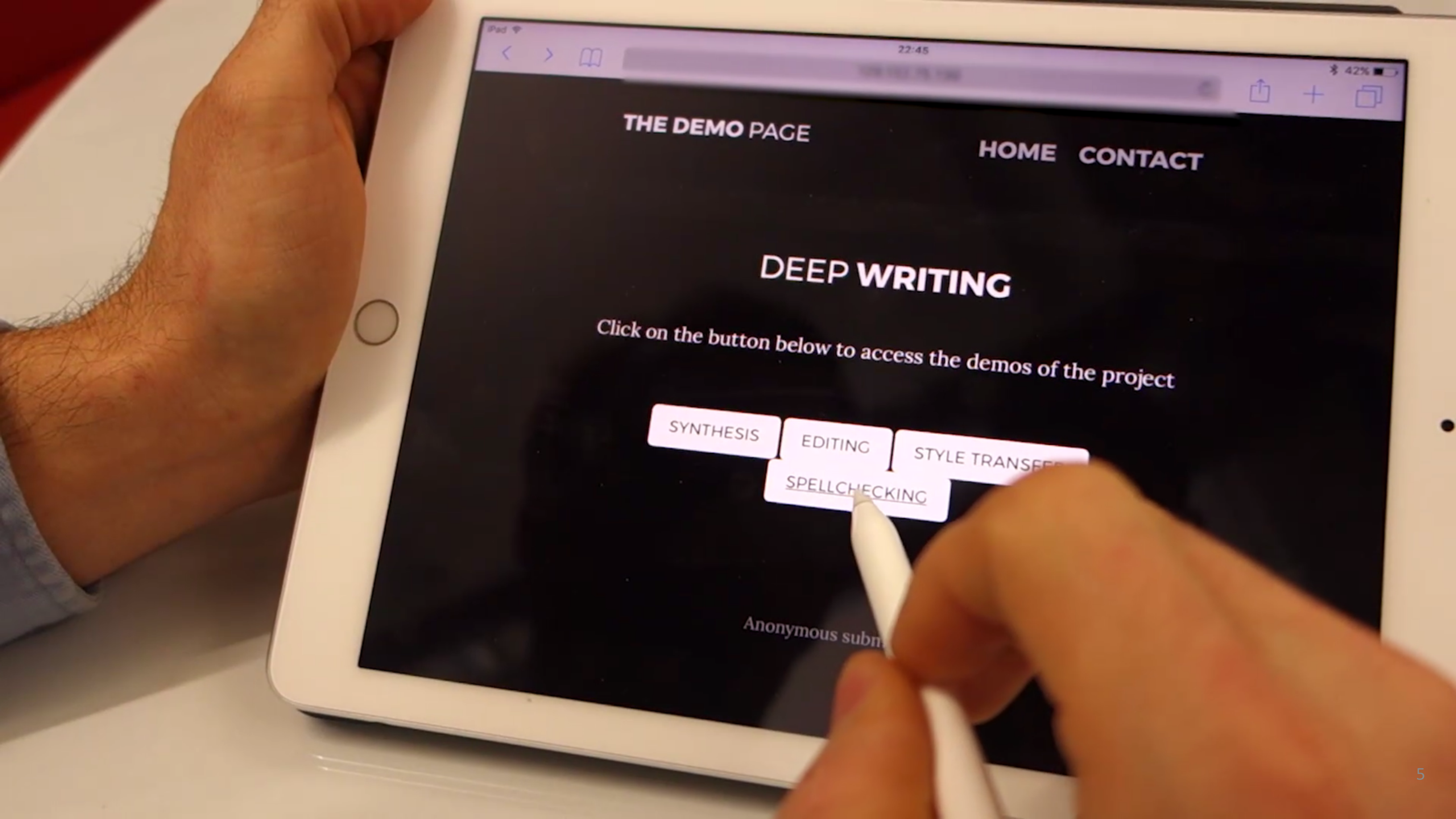
https://goo.gl/fSskTC


Eighty percent of success is showing up
Write
Eighty percent of success is showing up
Eighty percent of success is showing up
Eighty percent of success is showing up

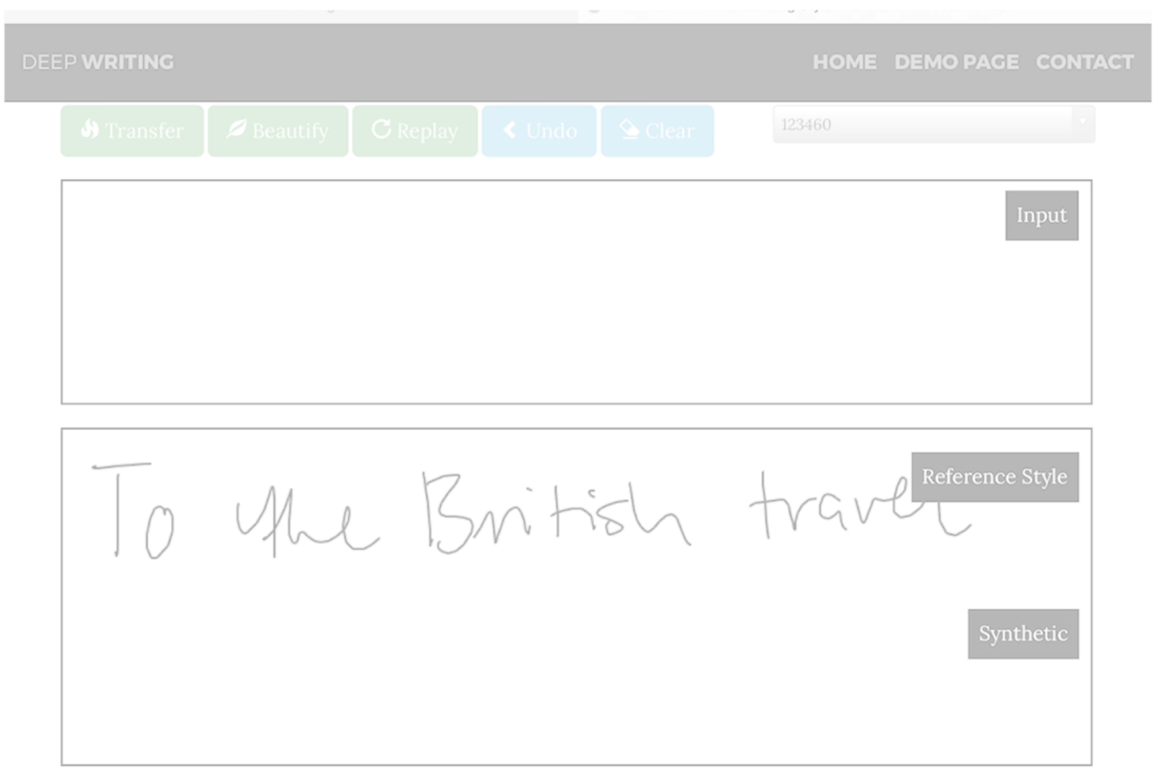

It's showtime folks
settle this argument
It's showtime folks


TOOLBOX
Select
Analyze
Undo Stroke
Clear Canvas
Just can't get enough
fust
can't
get
get
write
A
TOOLBOX
Select
Analyze
Undo Stroke
Clear Canvas
Just can't write enough
B


Input
TOOLBOX
Spellcheck
On
Check!
Spelling is overated
Suggestions
spelling is overrated
A
Corrected
Suggestions
Undo Stroke
Clear Canvas
Spelling is overrated
B


THE DEMO PAGE
HOME
CONTACT
DEEP WRITING
Click on the button below to access the demos of the project
SYNTHESIS
EDITING
SPELLCHECKING

Style Transfer



Digital ink promises to combine the flexibility
and aesthetics of handwriting and the ability
to process, search and edit digital text.

Character recognition converts handwritten text
into a digital representation, albeit at the cost
of losing style and personalized appearance
due to the technical difficulties of separating
the interwoven components of content and style.

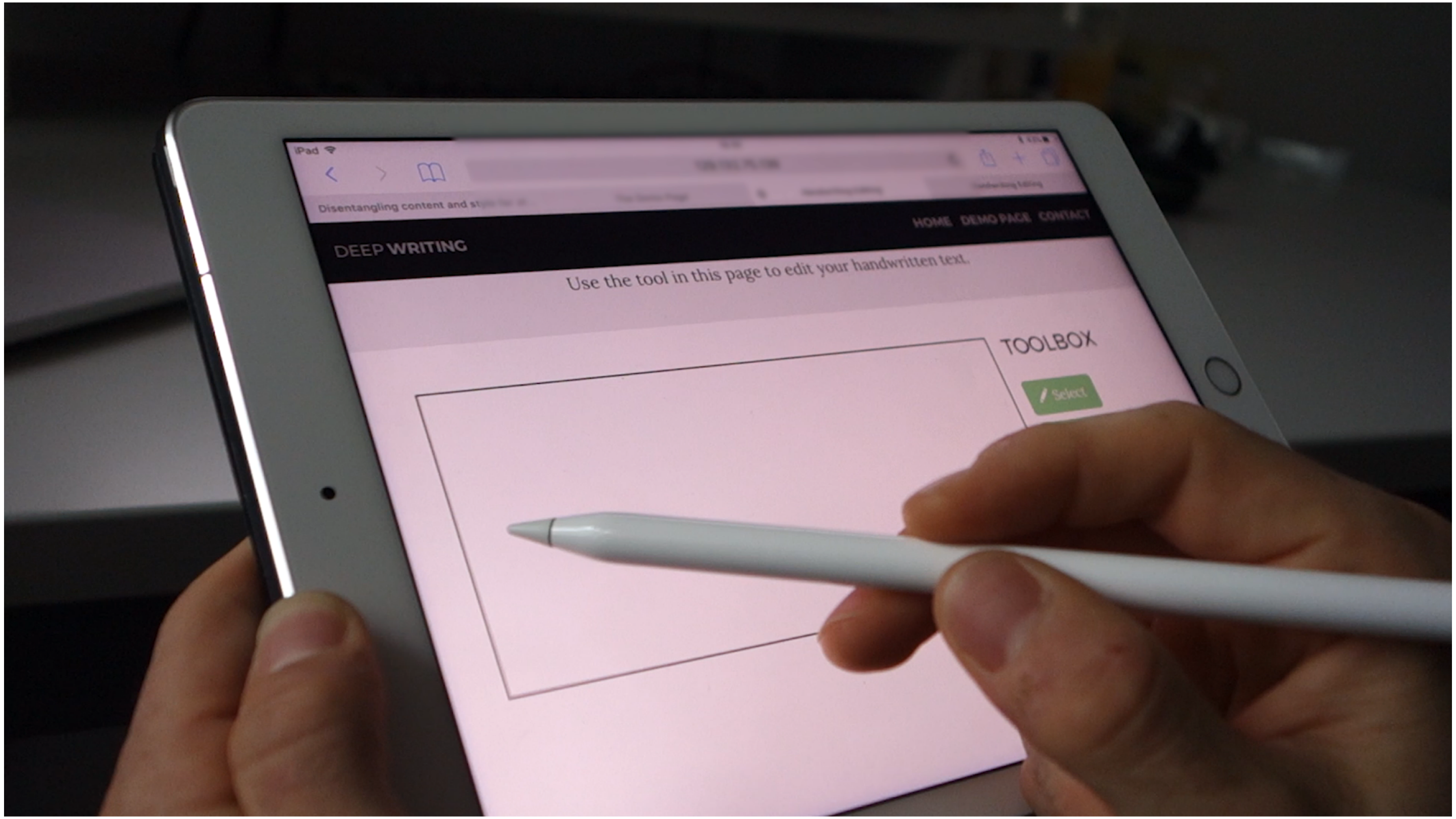
In this paper we propose a novel
generative neural network architecture
that is capable of disentangling style from content
and thus making digital ink editable.

Our model can synthesize arbitrary text
while giving users control over
the visual appearance.

For example allowing for style transfer
without changing the content
editing of digital ink at the character level
and other application scenarios
such as spellchecking and correction of handwritten text.

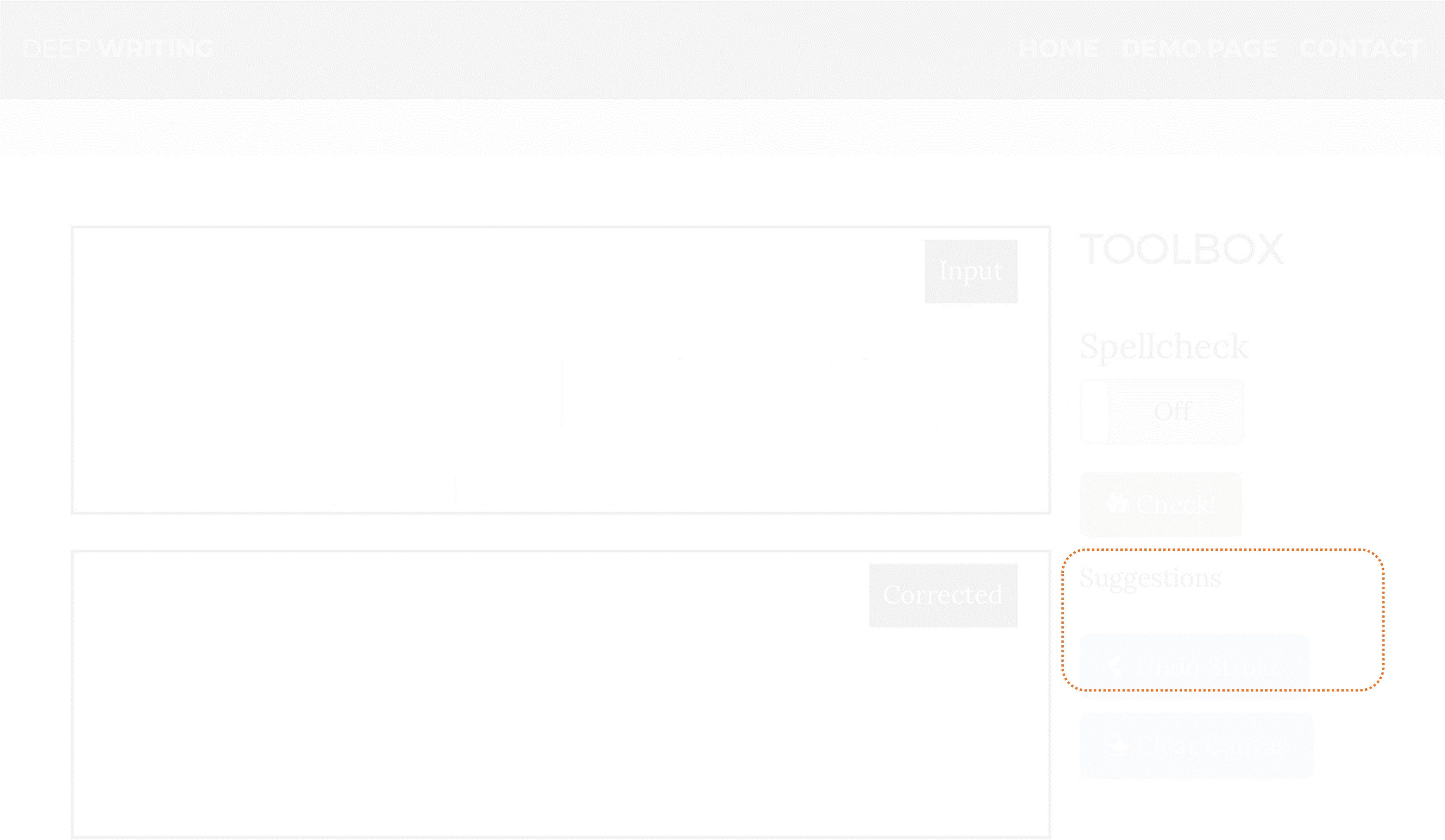
We furthermore contribute a new dataset
of handwritten text with fine grained annotations
at the character level and report results
from an initial user evaluation.

Editing

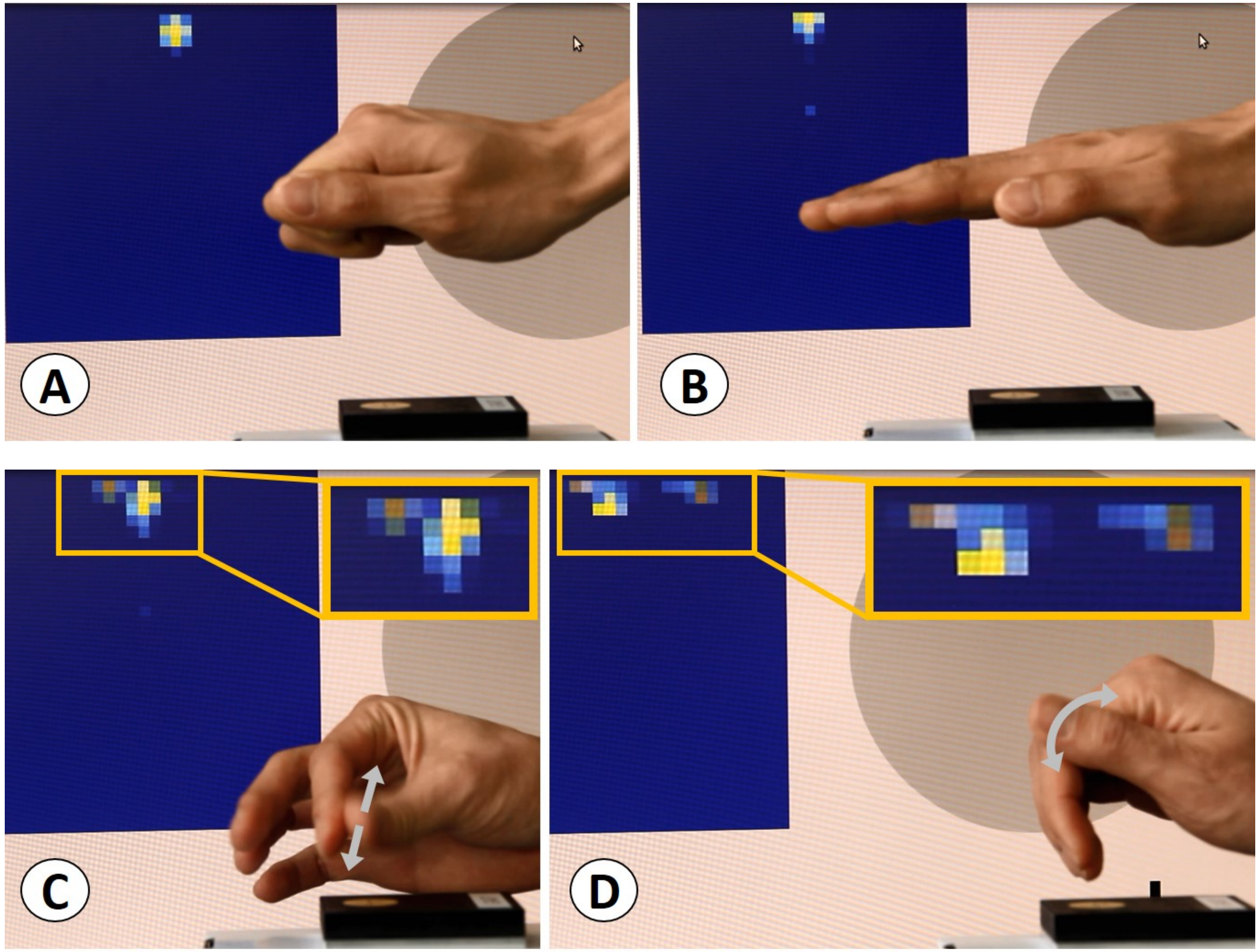


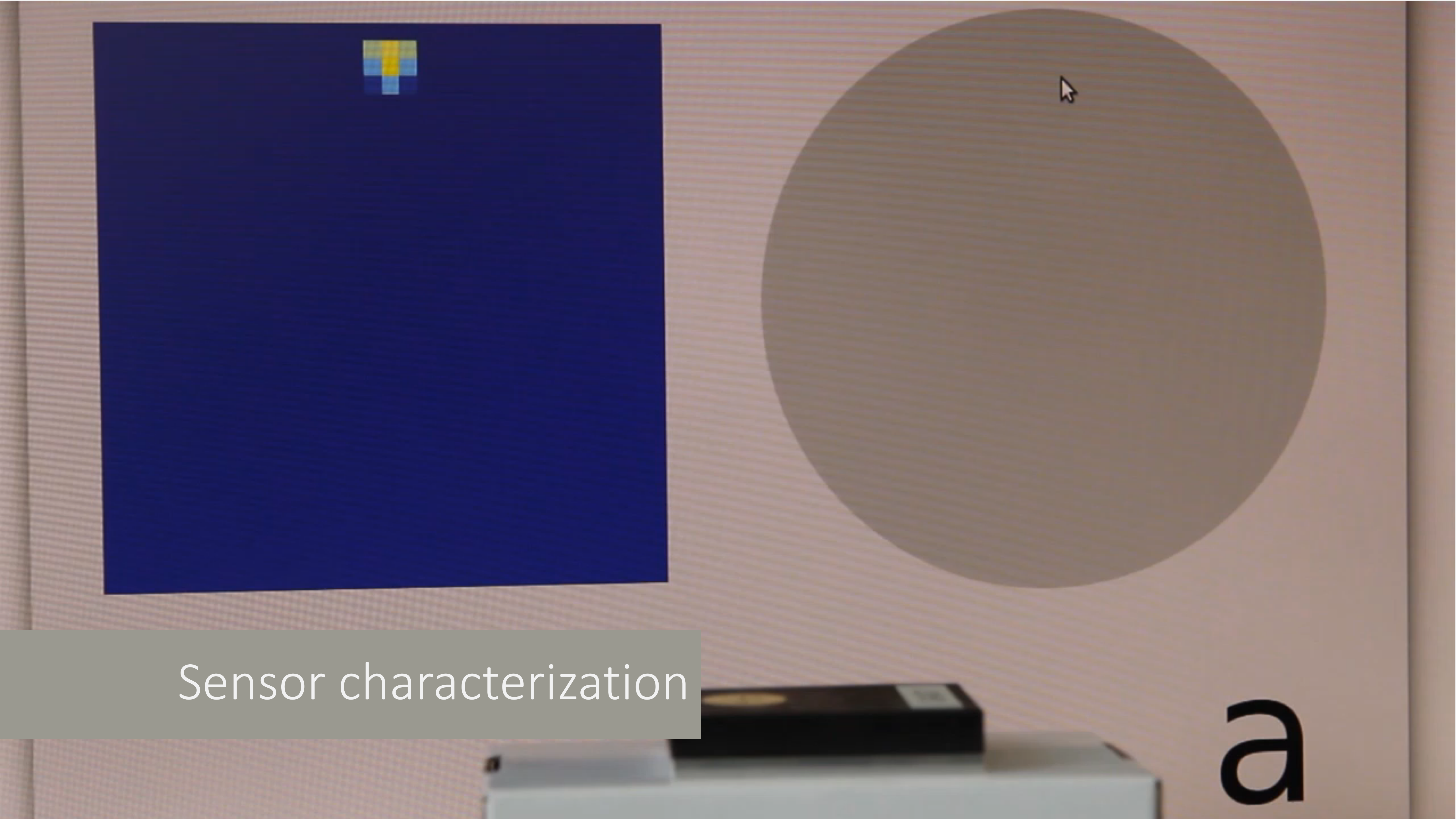
Disentangling content and st
DEEP WRITING
HOME
DEMO PAGE
CONTACT
Use the tool in this page to edit your handwritten text.
TOOLBOX
Select

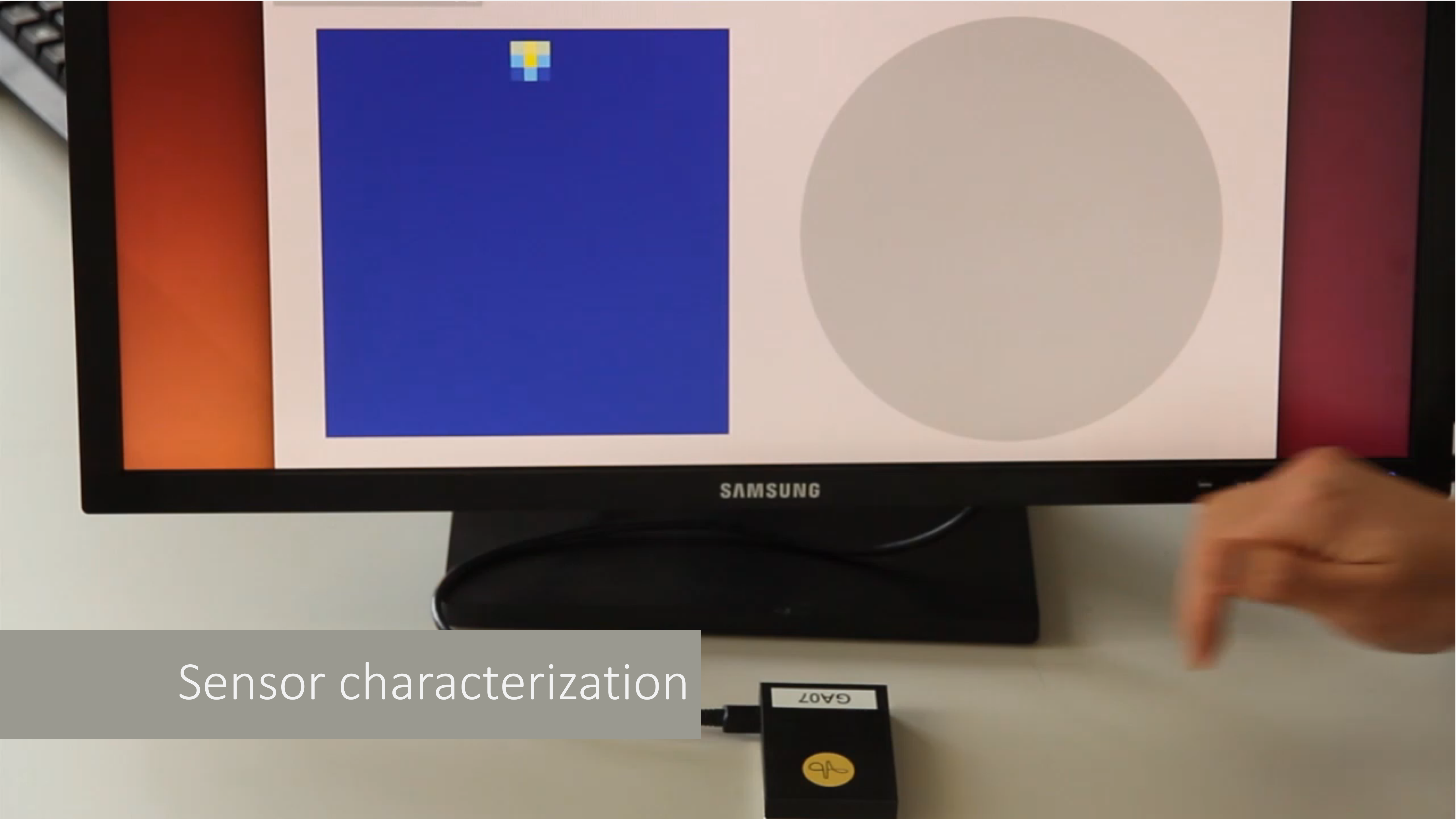
Spell-checking



Motivation: Fine-grained input recognition (UIST '16)

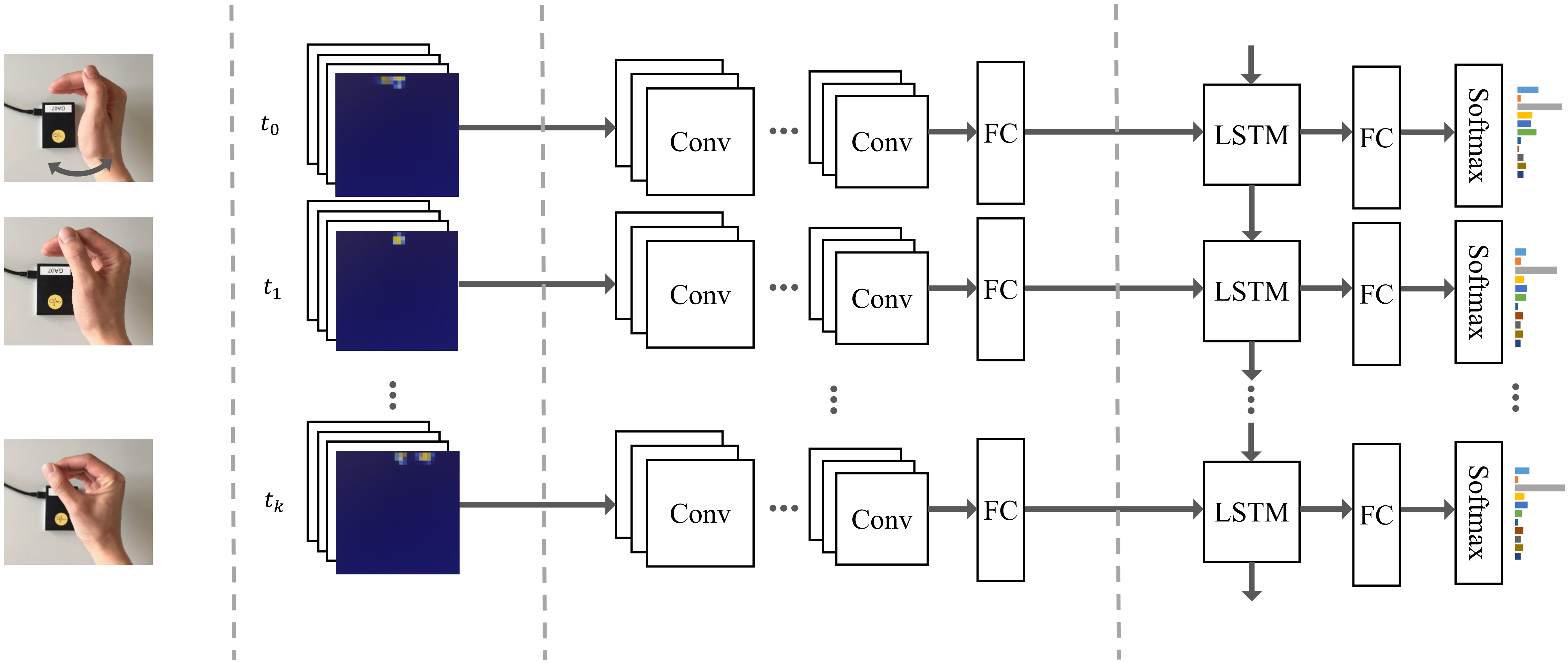



Sensor characterization
a



Sensor characterization

Deep learning pipeline



Schedule for the day

9-10: Introduction to neural networks for machine perception

10-10:30: coffee break

10:30-12:00: practical feed forward network

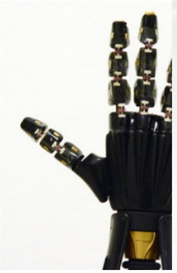
12-1pm: lunch

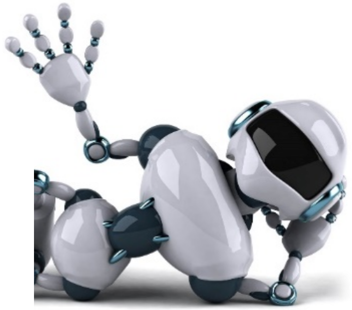
1-2pm: Introduction to CNNs

2-2:30pm: coffee break

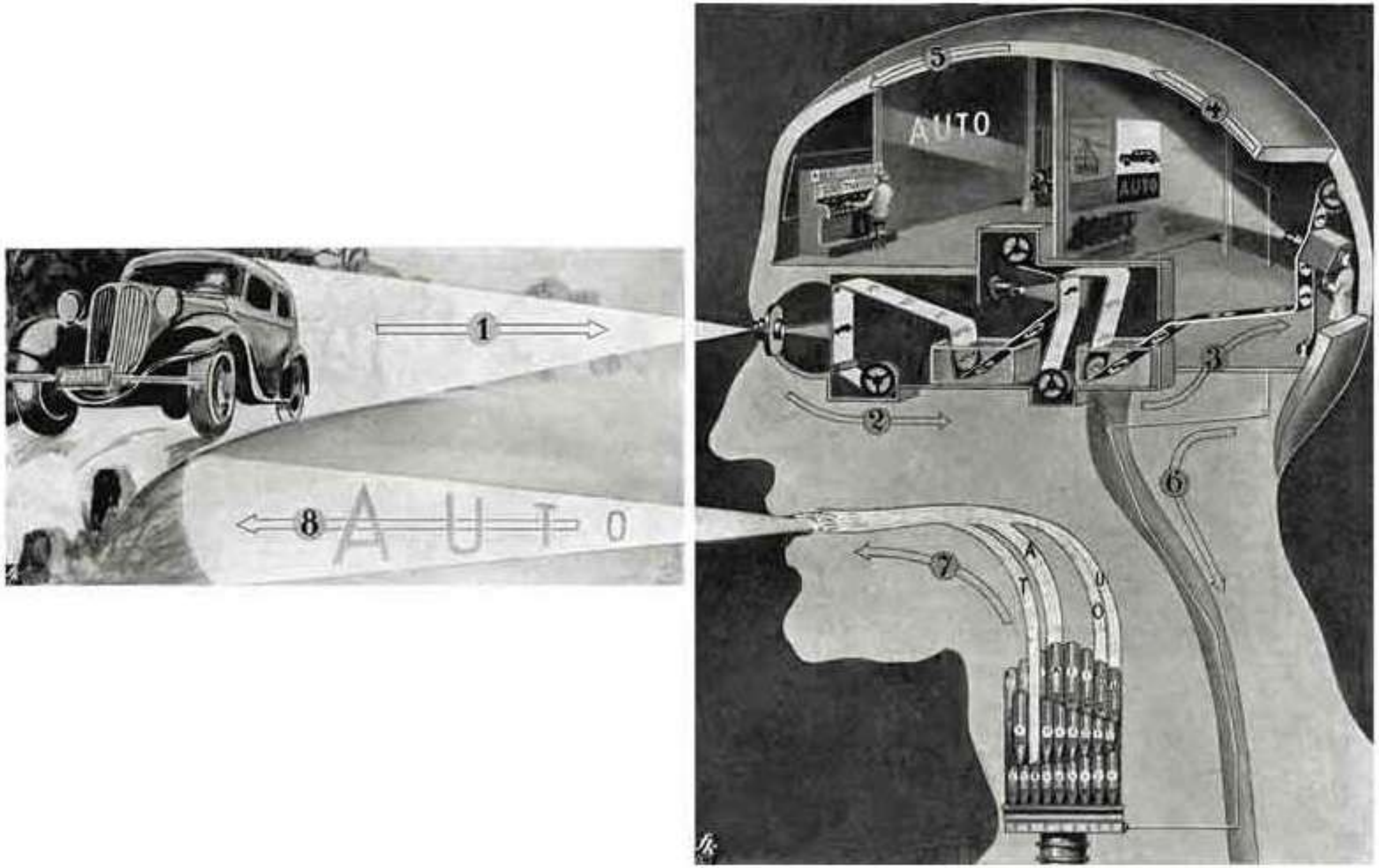
2:30-5:30pm: practical

Machine perception - Motivation





Motivation – How Do I see?



[Fritz Kahn. 1939. Der Mensch Gesund und Krank II.]

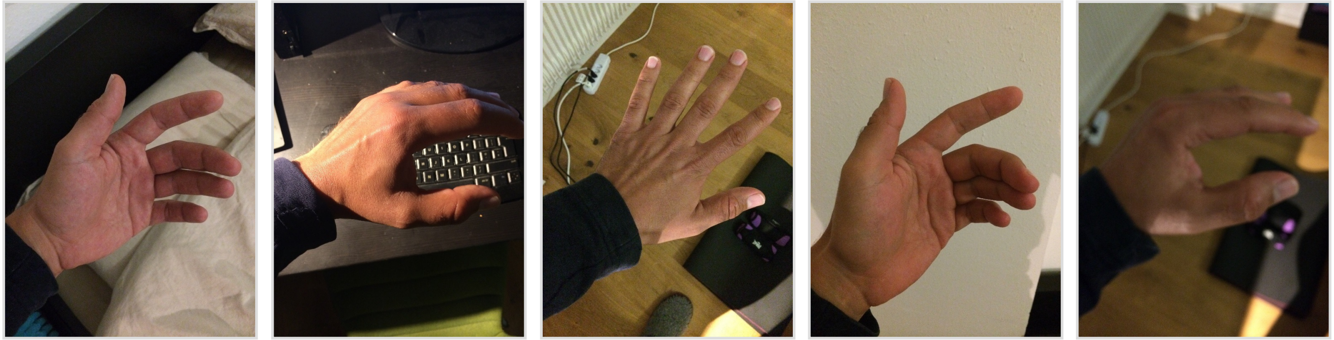
Problem Statement

Challenges

Viewpoint variation

Illumination

Deformation



Occlusion

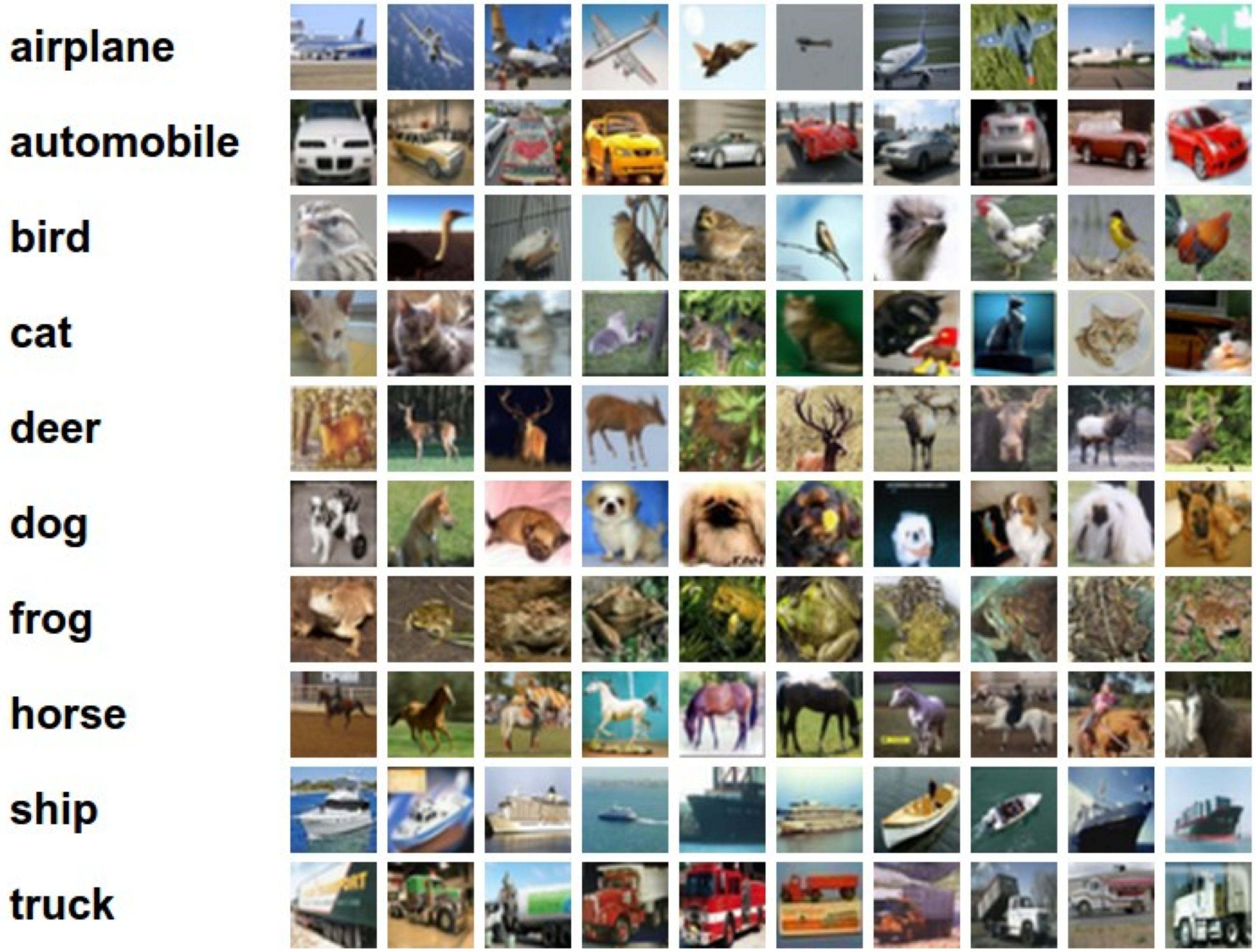
Background clutter

Intra-class variation

Classification: Basic Principles

Mapping of the inputs to a pre-determined set of targets.

$\longrightarrow \; f_\theta \; \longrightarrow$ *{House, Boat} or {+1, -1}*



Example dataset: **CIFAR-10**

10 labels

50,000 training images

each image is **32x32x3**

10,000 test images.

[Slides adopted from: Li, Karpathy]

Parametric approach

image parameters

$$f(\mathbf{x}, \mathbf{W})$$

→

10 numbers, indicating class scores

[32x32x3]
array of numbers 0...1
(3072 numbers total)

[Slides adopted from: Li, Karpathy]

Parametric approach: **Linear classifier**

$$\underbrace{\boxed{f(\boldsymbol{x}; \boldsymbol{W}, \boldsymbol{b})}}_{} = \boxed{\boldsymbol{W}}\boxed{\boldsymbol{x}} + \boxed{\boldsymbol{b}}$$

10x1 10x3072 10x1

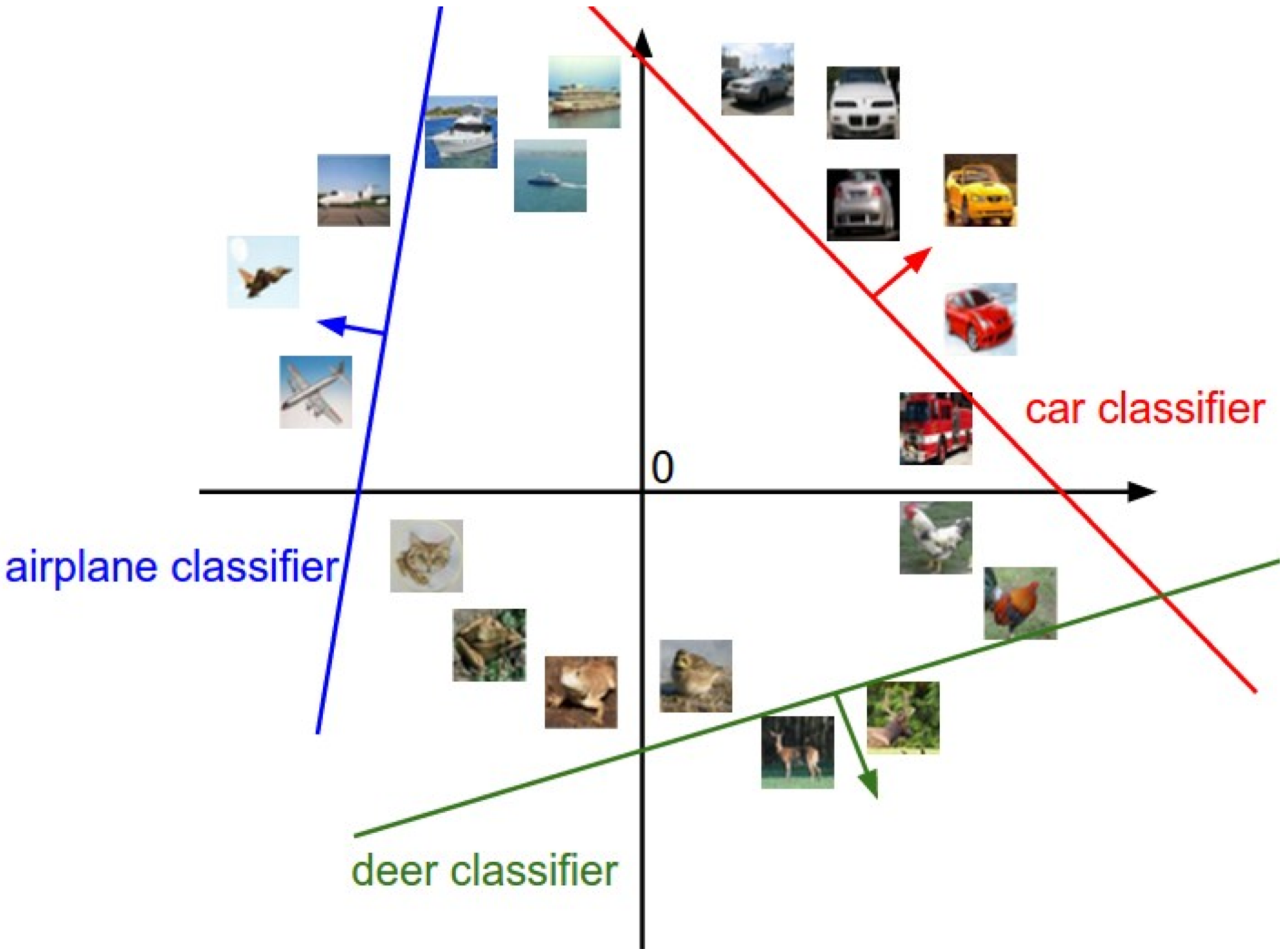
3072x1

[32x32x3]
array of numbers 0...1

10 numbers, indicating class scores

[Slides adopted from: Li, Karpathy]

Interpreting a Linear Classifier



$$f\left(x^{(i)}; W, b\right) = Wx^{(i)} + b$$

[32x32x3]
array of numbers 0...1
(3072 numbers total)

[Slides adopted from: Li, Karpathy]

Parametric approach: **Linear classifier**

$$\underset{10\text{x}1}{\boxed{f(\boldsymbol{x};\boldsymbol{W},\boldsymbol{b})}} = \underset{10\text{x}3072}{\boxed{\boldsymbol{W}}}\,\underset{3072\text{x}1}{\boxed{\boldsymbol{x}}} + \underset{10\text{x}1}{\boxed{\boldsymbol{b}}}$$

[32x32x3]
array of numbers 0...1

10 numbers, indicating class scores

[Slides adopted from: Li, Karpathy]

Example with an image with 4 pixels, and 3 classes (cat/dog/ship)

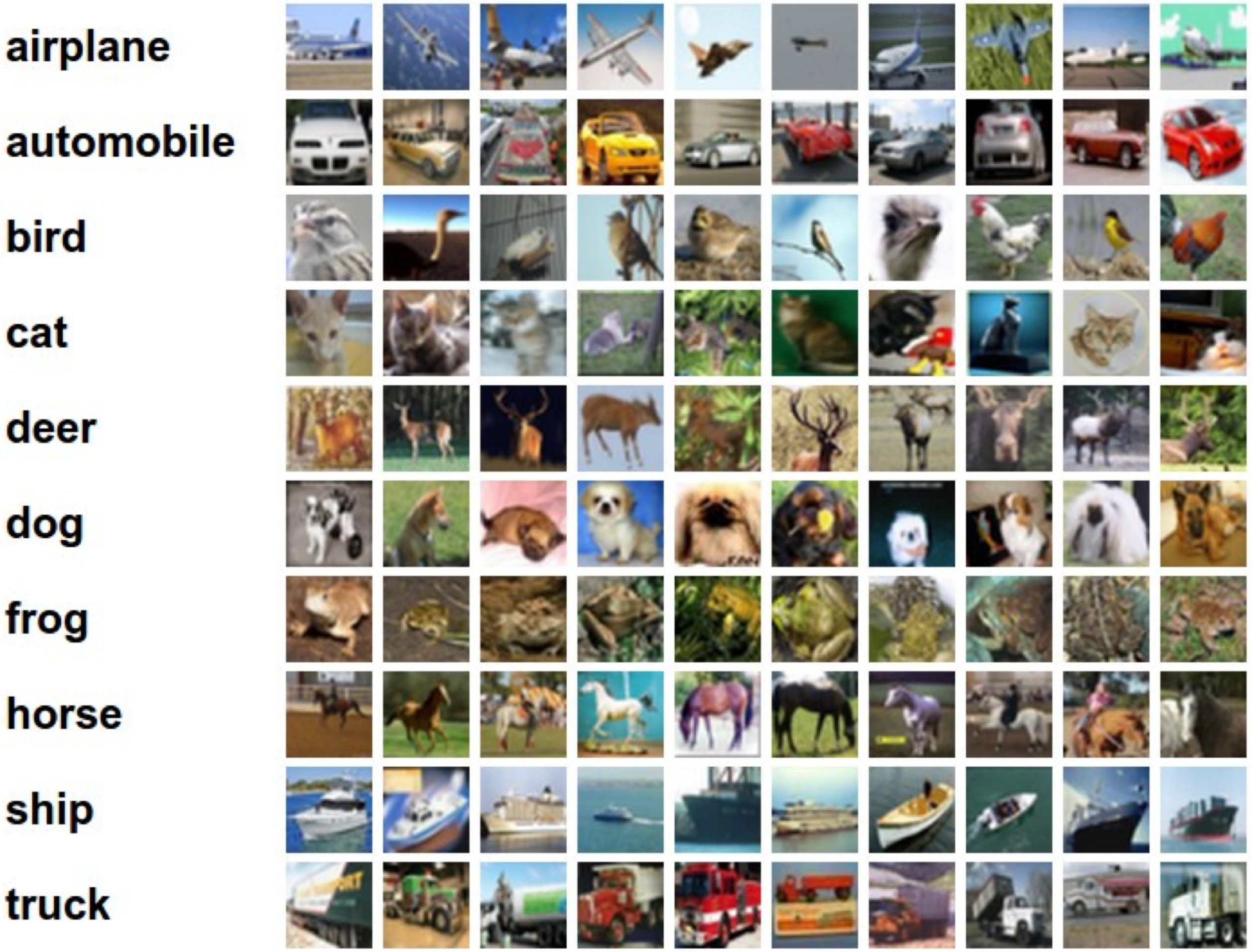
stretch pixels into single column

input image

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 & 2.0 \\ 1.5 & 1.3 & 2.1 & 0.0 \\ 0 & 0.25 & 0.2 & -0.3 \end{bmatrix}}_{\boldsymbol{W}} \underbrace{\begin{bmatrix} 56 \\ 231 \\ 24 \\ 2 \end{bmatrix}}_{\boldsymbol{x}^{(i)}} + \underbrace{\begin{bmatrix} 1.1 \\ 3.2 \\ -1.2 \end{bmatrix}}_{\boldsymbol{b}} \rightarrow \underbrace{\begin{bmatrix} -96.8 \\ 437.9 \\ 61.95 \end{bmatrix}}_{\boldsymbol{f}\left(\boldsymbol{x}^{(i)};\boldsymbol{W},\boldsymbol{b}\right)} \begin{array}{l} \text{cat score} \\ \text{dog score} \\ \text{ship score} \end{array}$$

[Slides adopted from: Li, Karpathy]

Interpreting a Linear Classifier

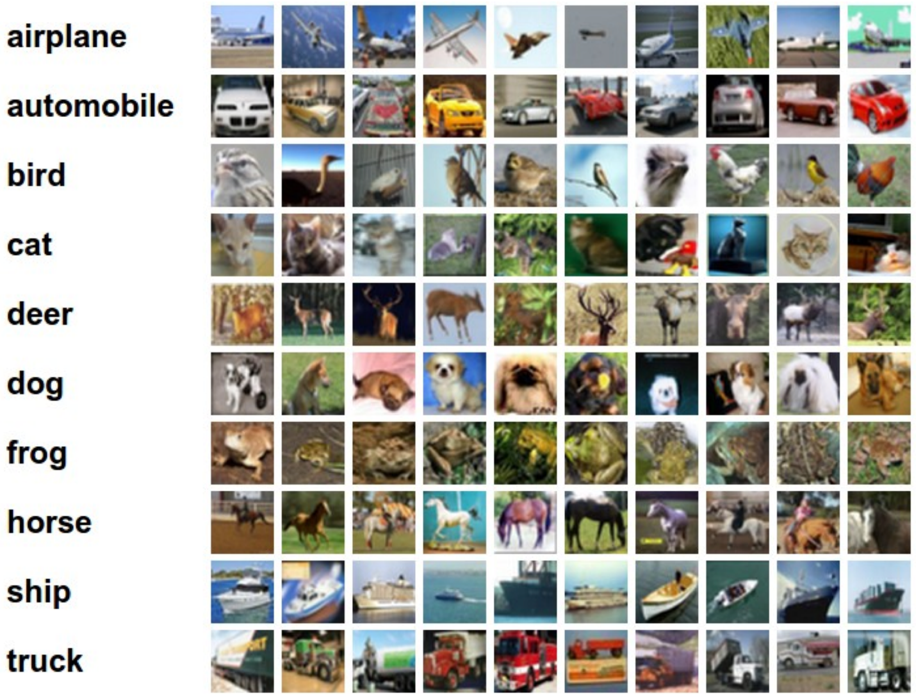


$$f\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}, \boldsymbol{b}\right) = \boldsymbol{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}$$

Q: what does the linear classifier do, in English?

[Slides adopted from: Li, Karpathy]

Interpreting a Linear Classifier

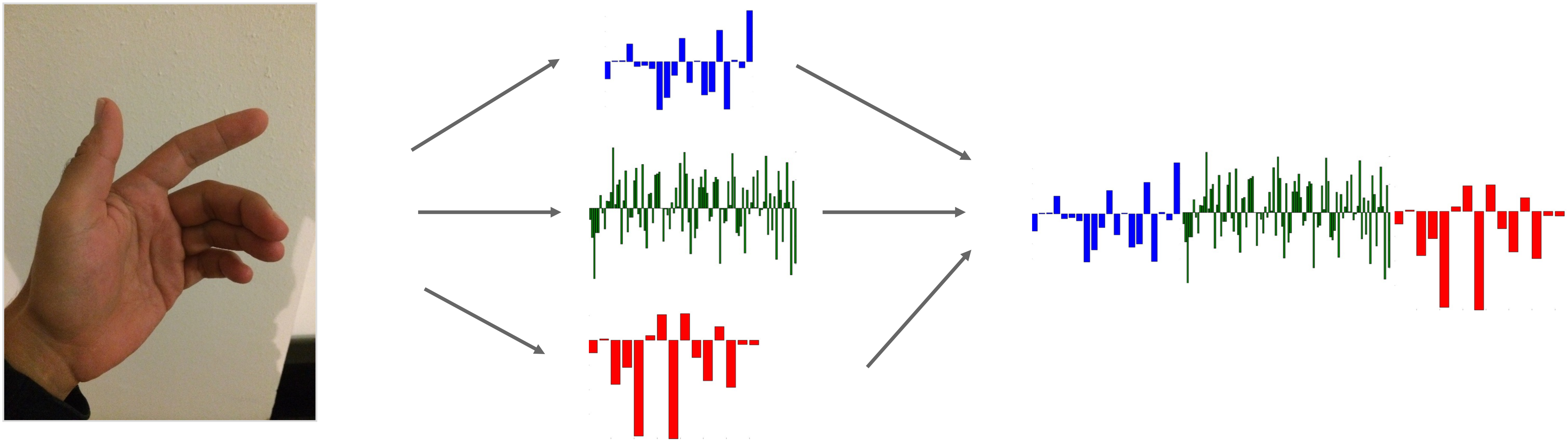


$$f\left(\boldsymbol{x}^{(i)}; \boldsymbol{W}, \boldsymbol{b}\right) = \boldsymbol{W}\boldsymbol{x}^{(i)} + \boldsymbol{b}$$

Example trained weights of a linear classifier trained on CIFAR-10:

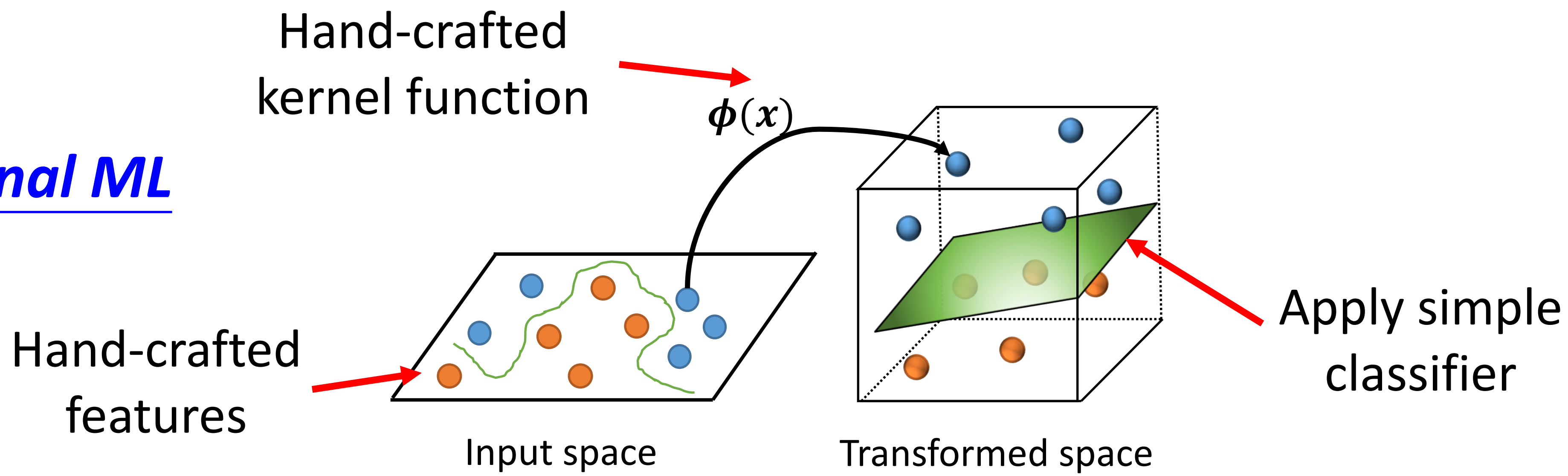
[Slides adopted from: Li, Karpathy]

Aside: Image Features

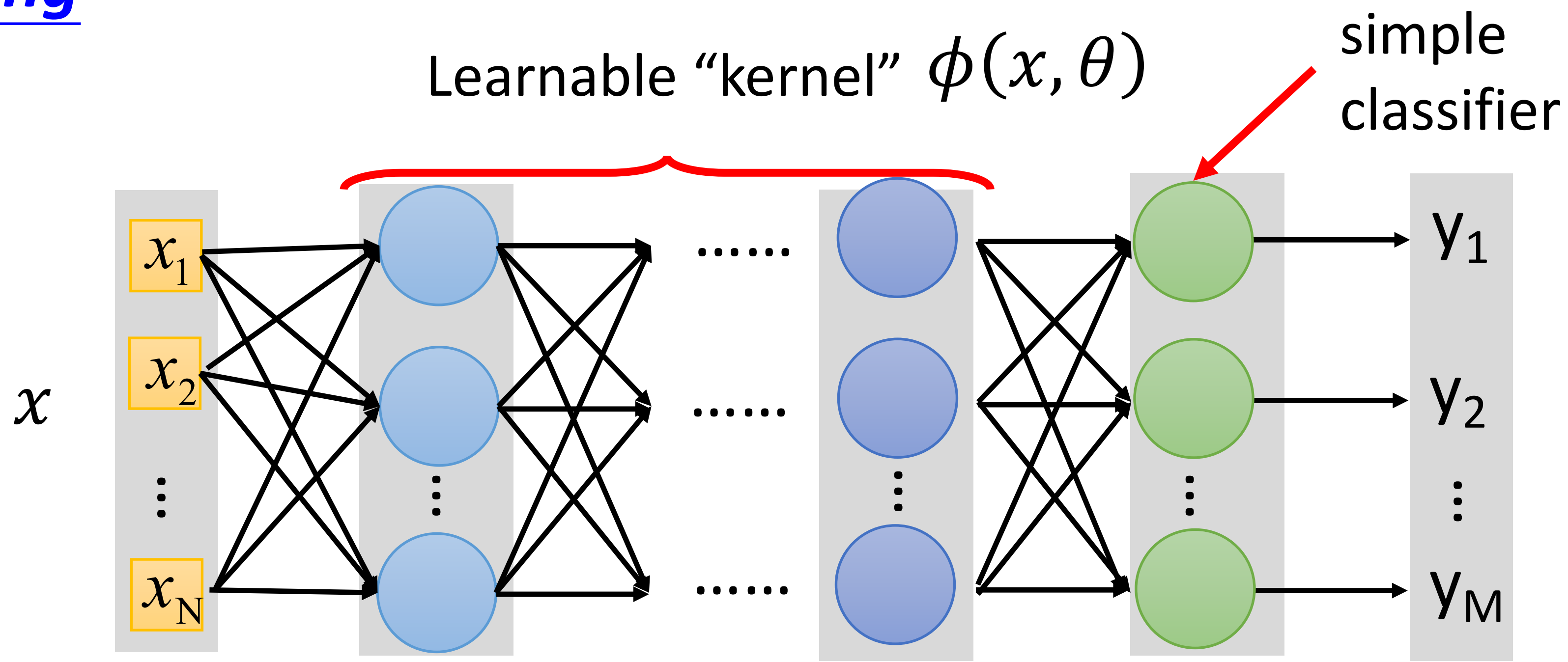


[Slides adopted from: Li, Karpathy]

Traditional ML



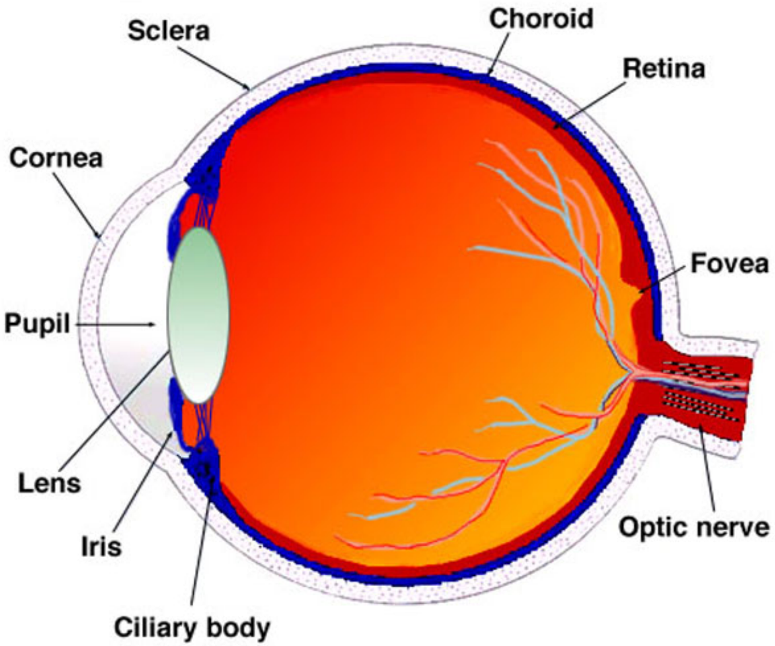
Deep Learning



Representation learning

1. Use generic Kernel function $\phi\left(x^{(i)}\right)$ such as RBF (risks poor generalization
2. Manually engineer $\phi\left(x^{(i)}\right)$ (requires lots of domain knowledge)
3. Learn function and parameters from data $y = f\left(x^{(i)}, \Theta, w\right) = \phi\left(x^{(i)}, \Theta\right)^T w$

Motivation – How Do I See?



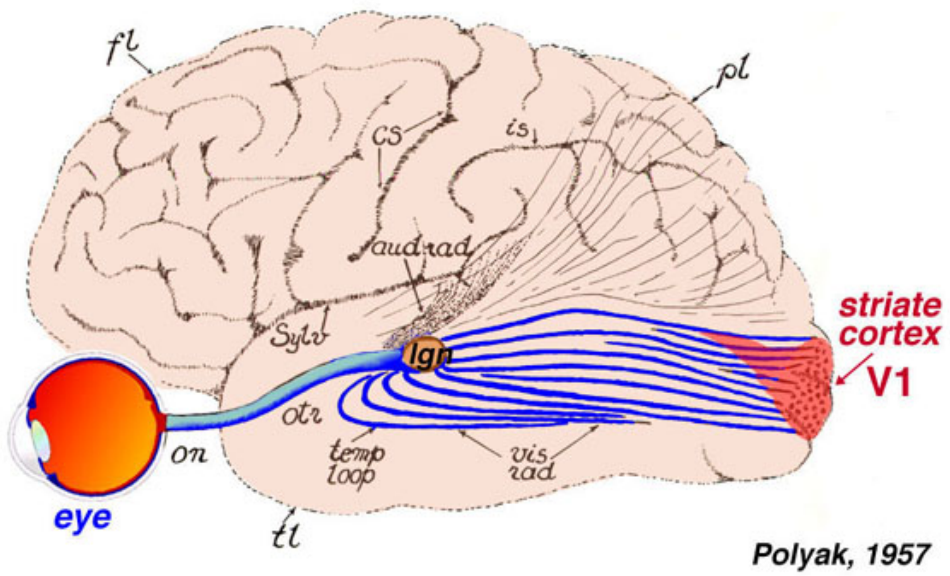
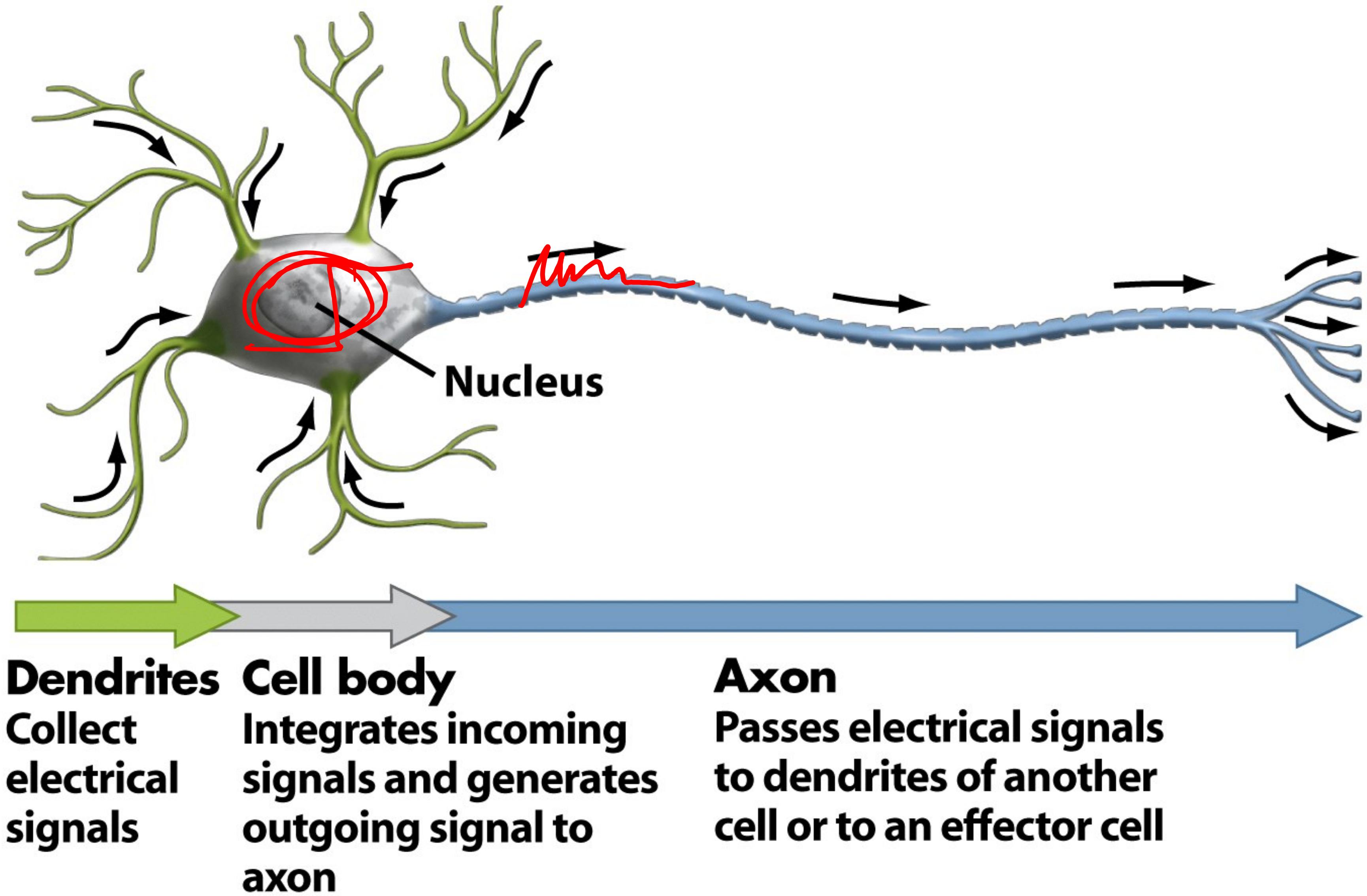


Figure 8. Visual input to the brain goes from eye to LGN and then to primary visual cortex, or area V1, which is located in the posterior of the occipital lobe. Adapted from Polyak (1957).

Information Flow through Neurons



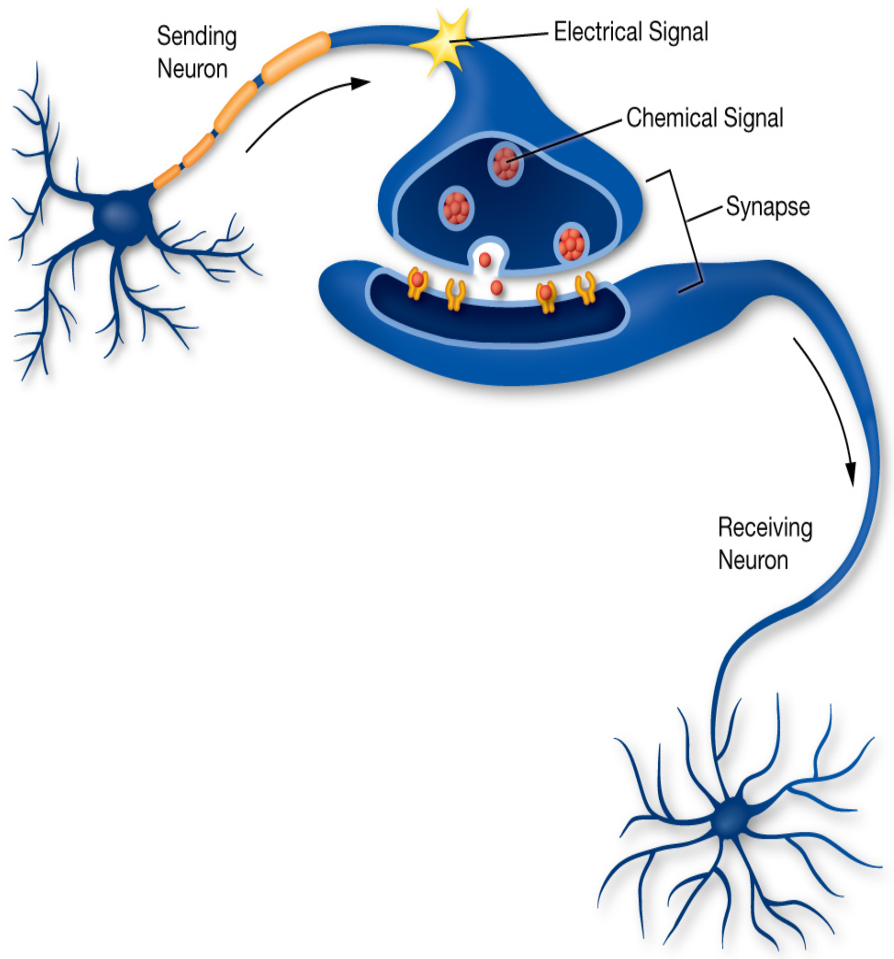
http://www.uic.edu/classes/bios/bios100/lectures/nervous.htm

Synapses transmit signal from one Neuron to another

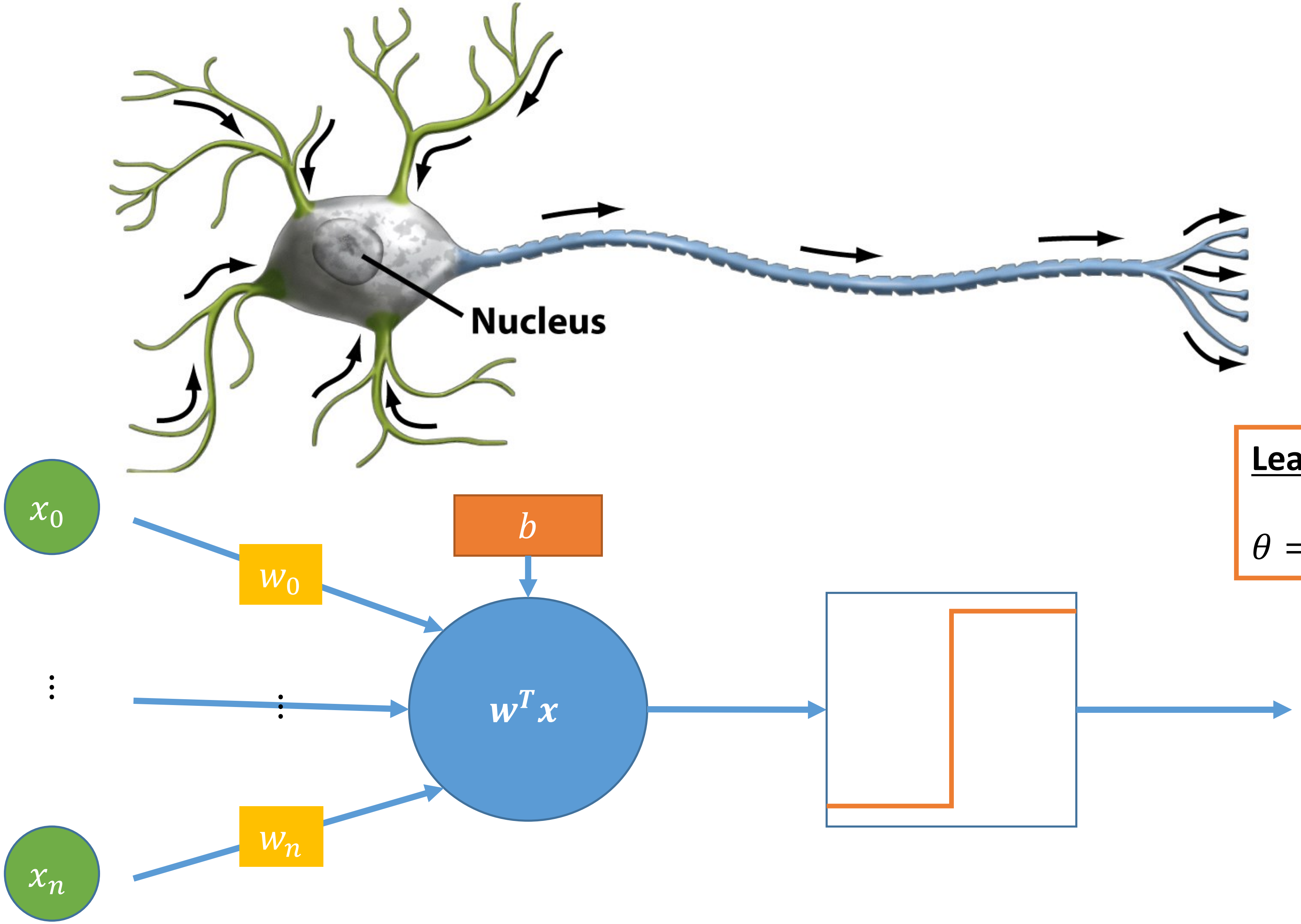
Information from one neuron flows to another neuron across a small gap called a synapse.

Action potentials are translated into chemical signals (neuro transmitters) to cross the gap.

Once on the other side, the signal becomes electrical again.



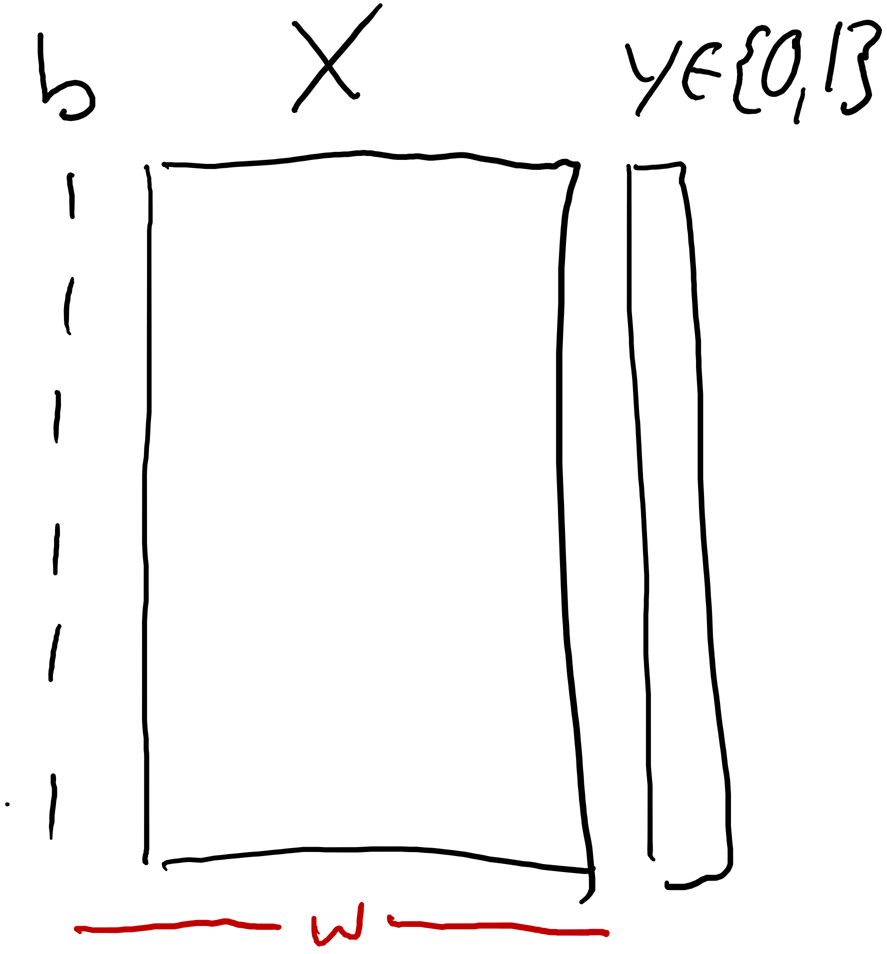
Perceptron – Engineering view of a neuron



Learnable parameters:

$\theta = \{\boldsymbol{w}, b\}$

Perceptron learning



init: $w^0 = 0$

while: $\exists\, i_k$ s.t. $({w^k}^T x^{i_k} \geq 0) \neq y^{i_k}$

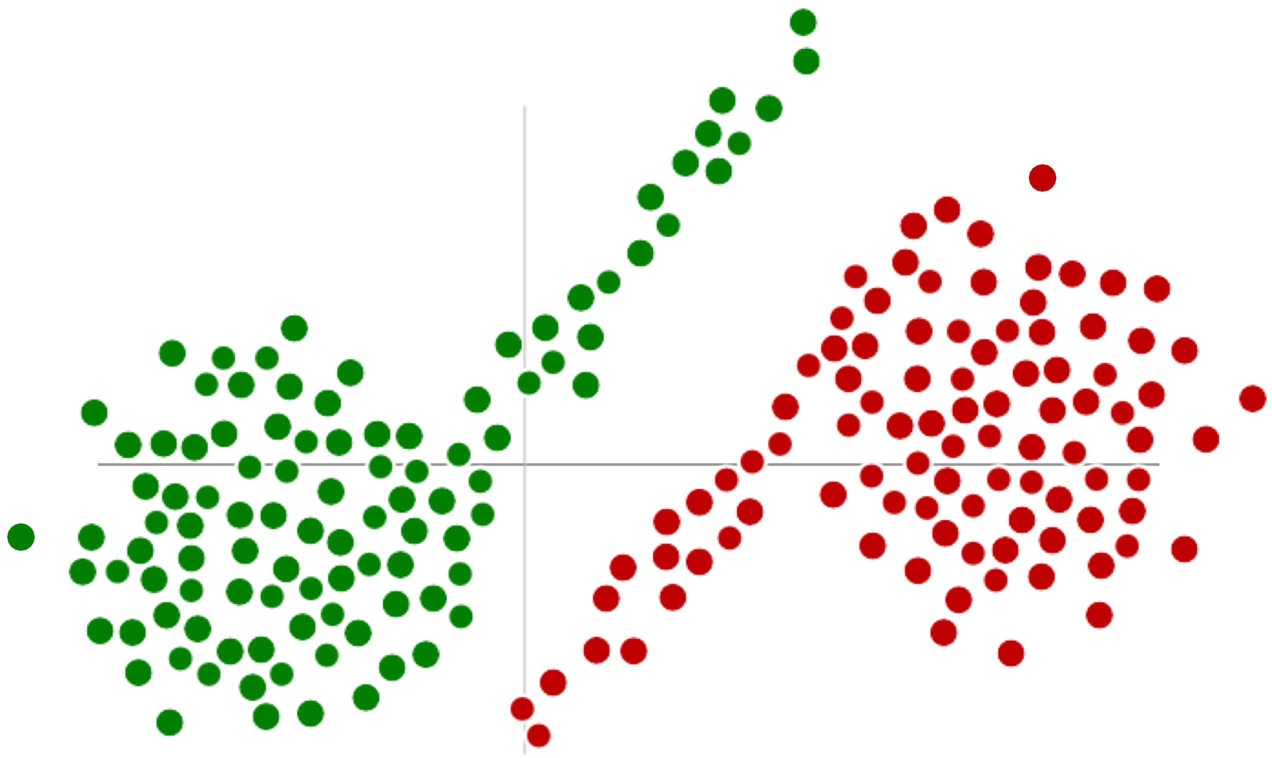
do: $w^{k+1} = w^k + \eta \underbrace{(y - \hat{y})}_{\text{error}} x^{i_k}$

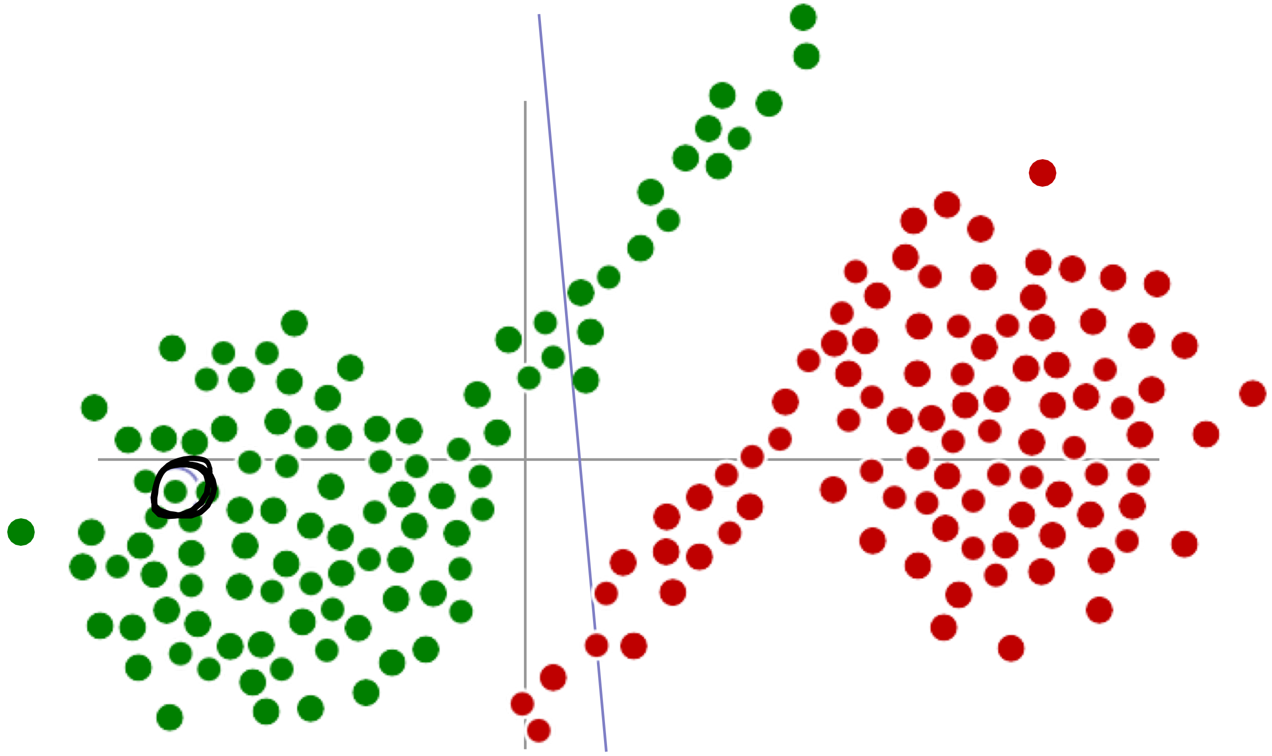
where: $\hat{y} = (w^T x \geq 0)$

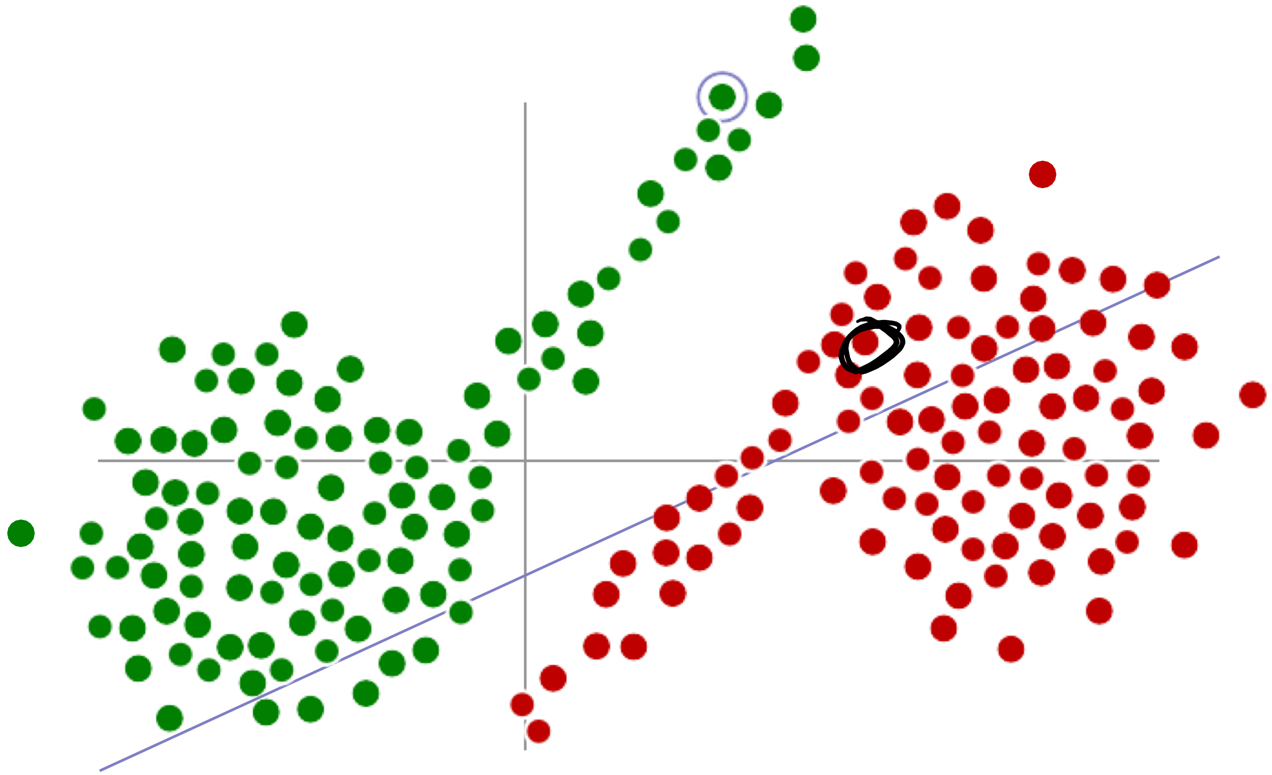
y	$\hat{y}$	$(y - \hat{y})$
0	0	0
0	1	-1
1	0	1
1	1	0

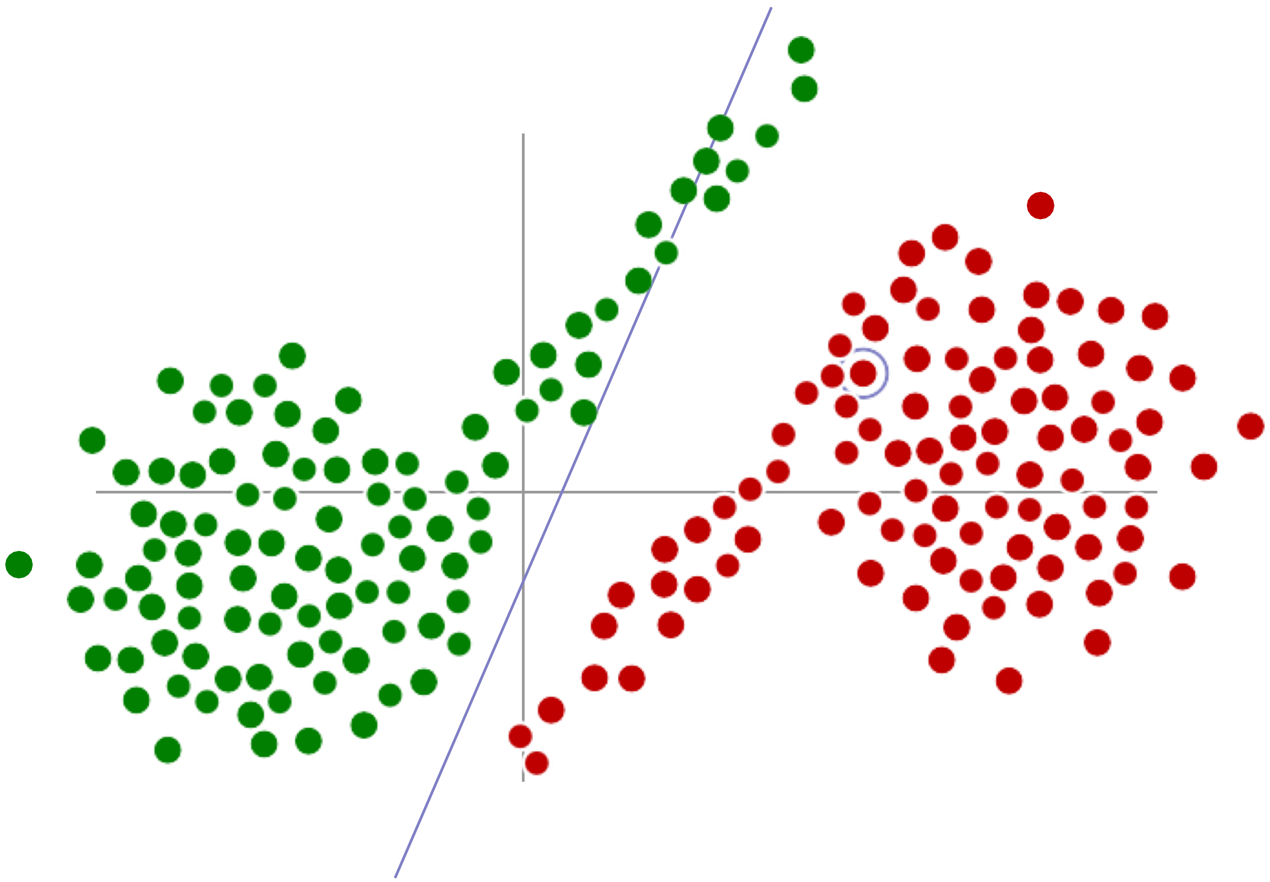
iff linearly sep

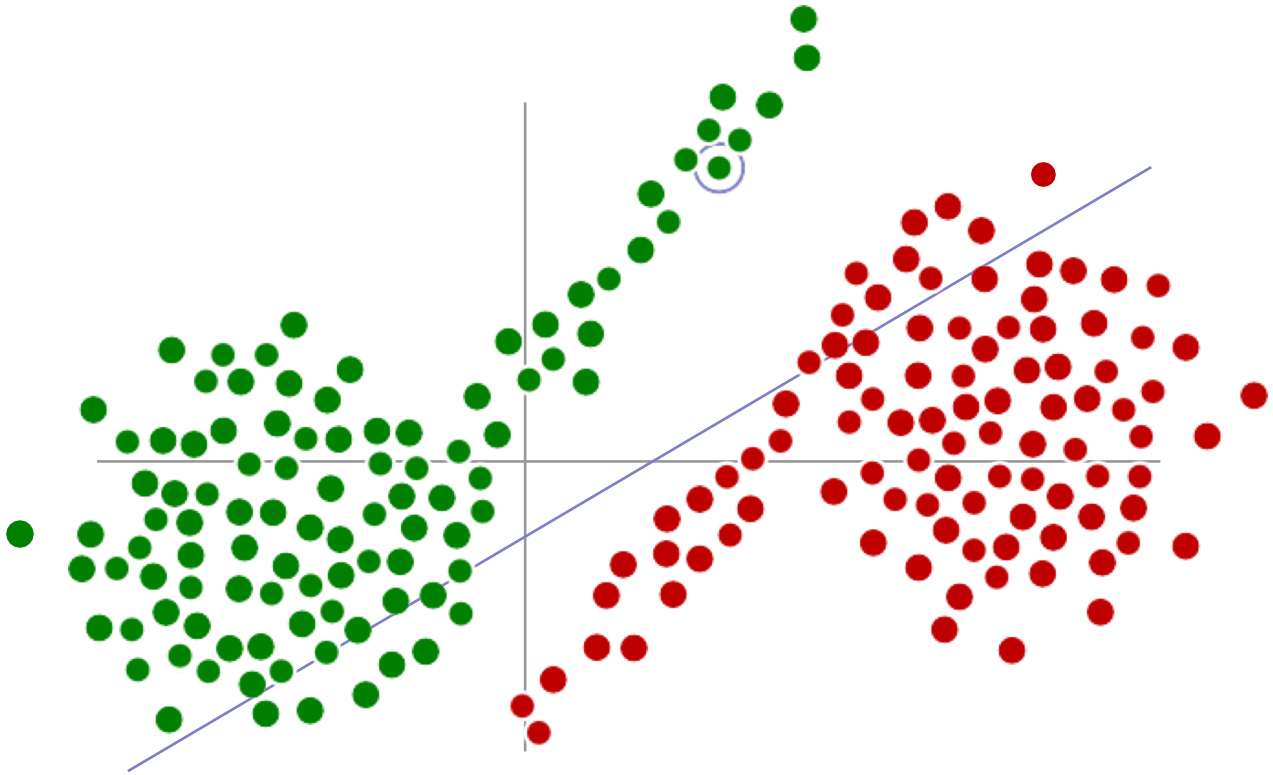
$\Rightarrow$ converges in finite time

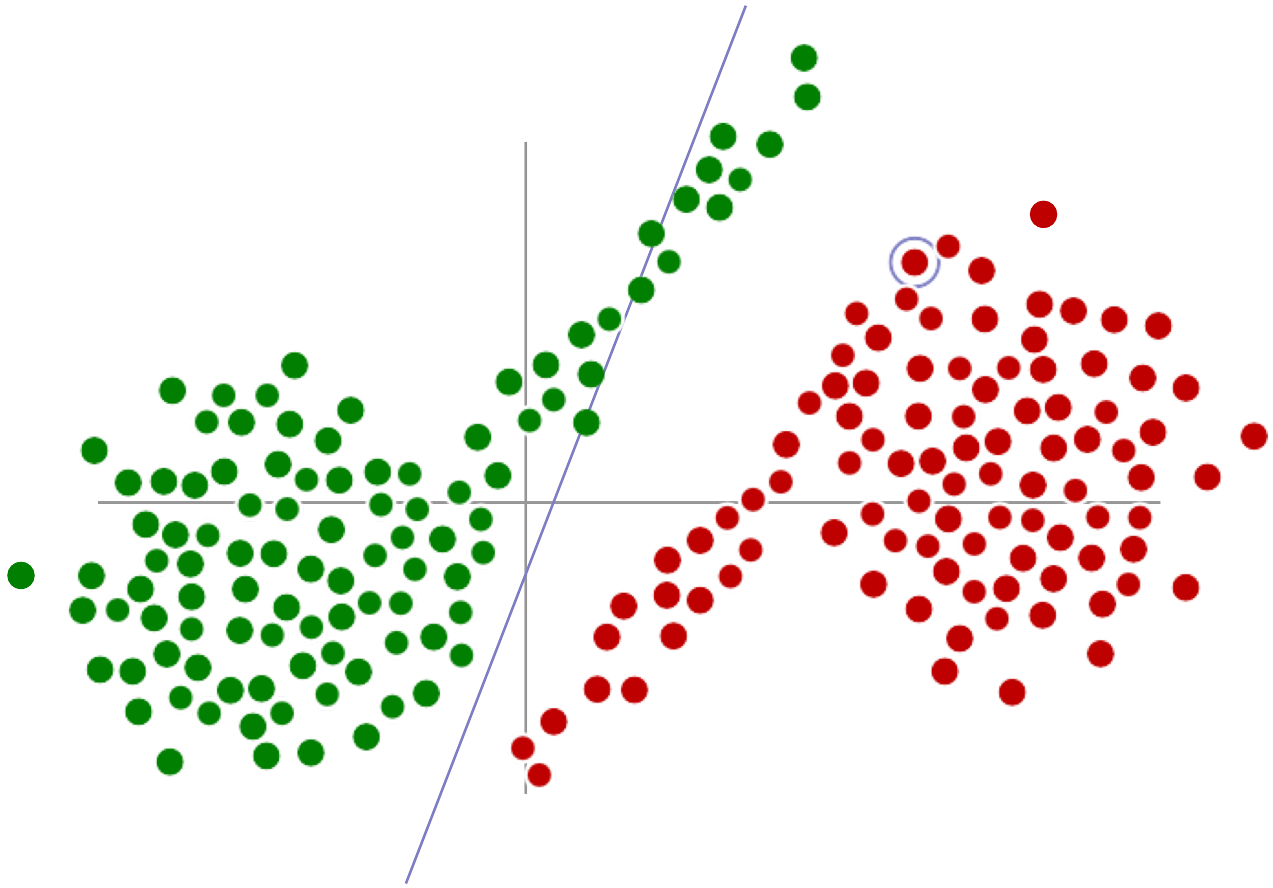


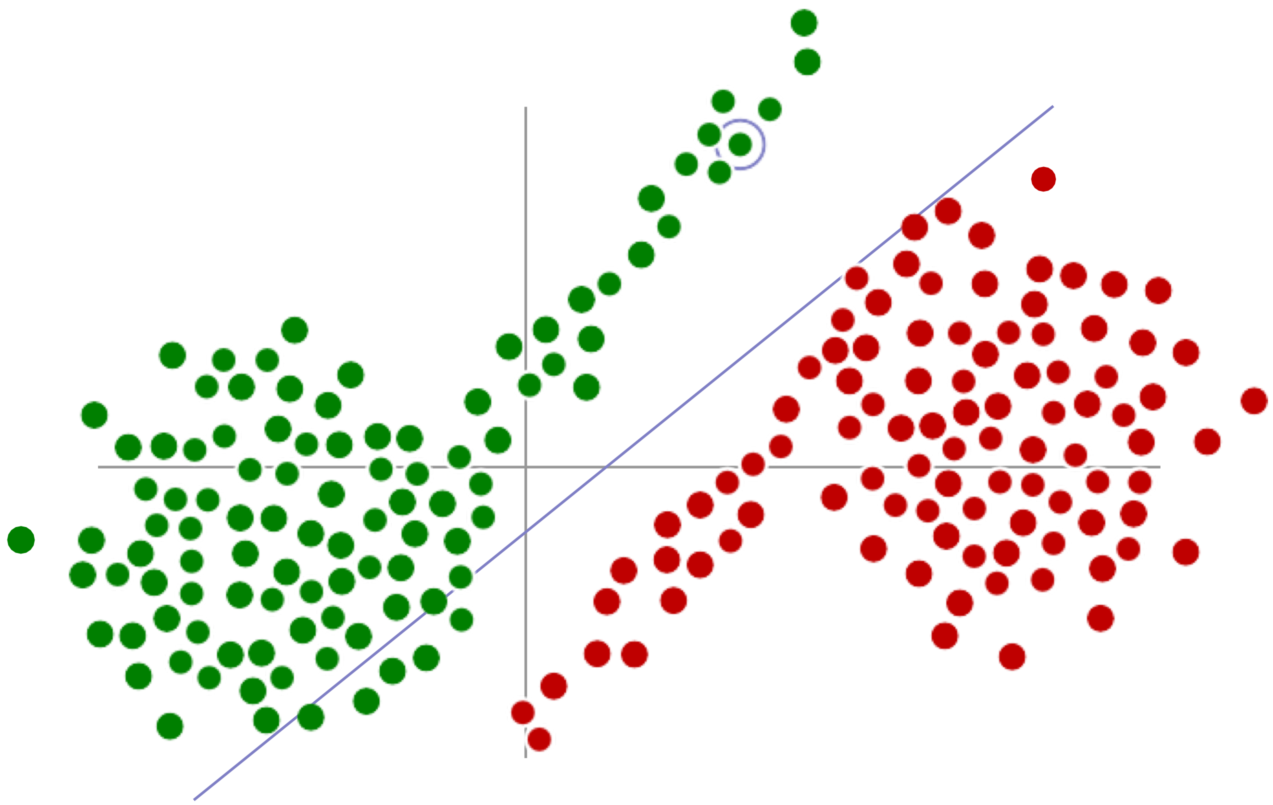


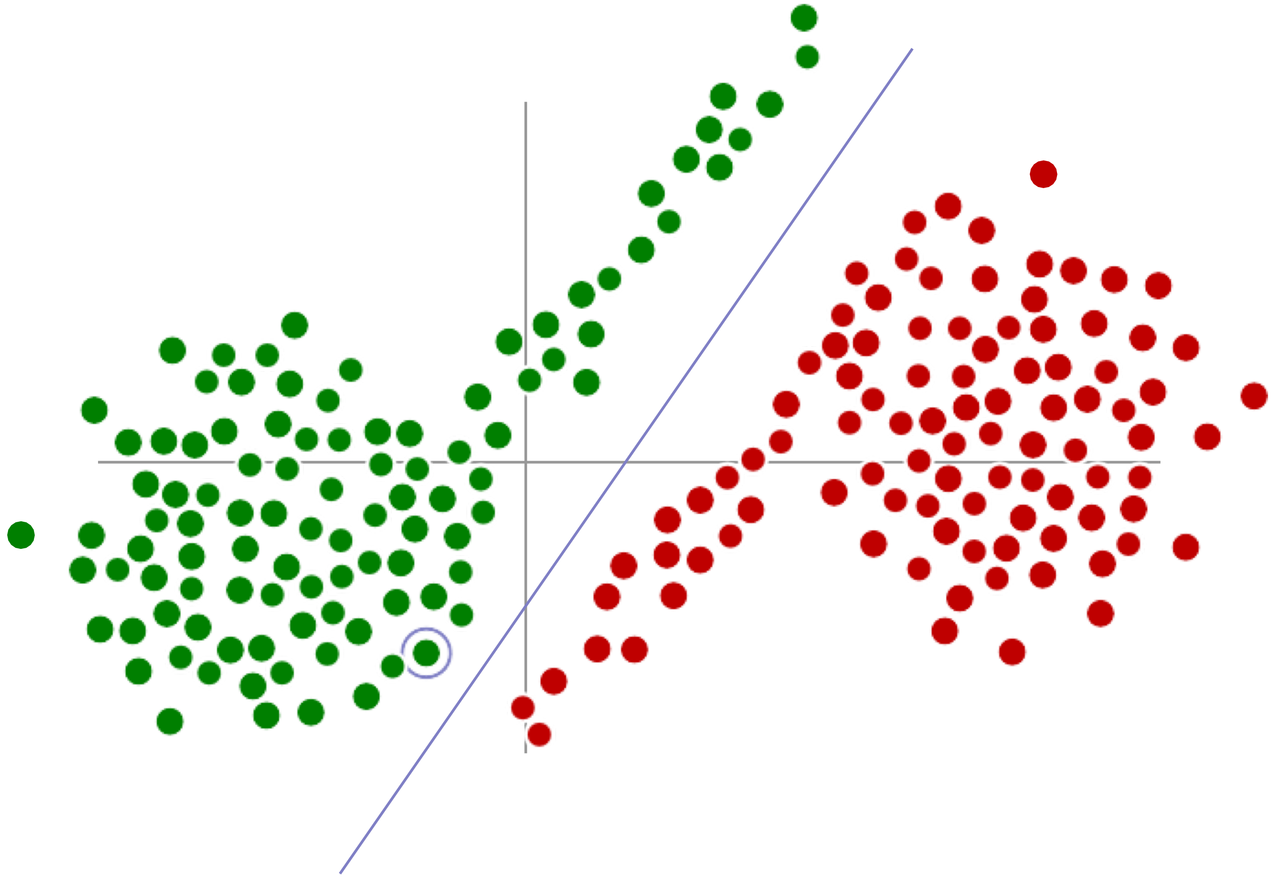


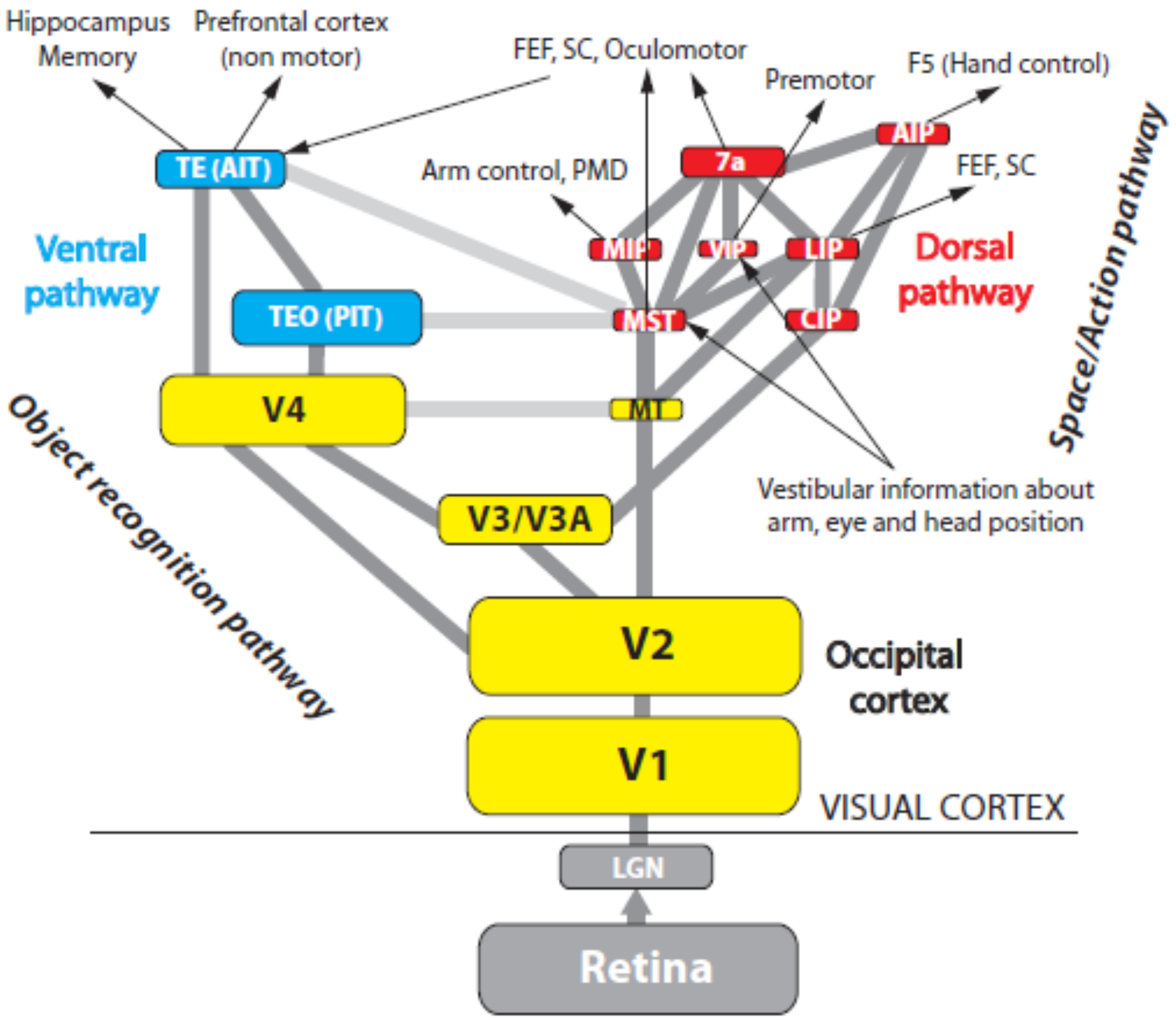


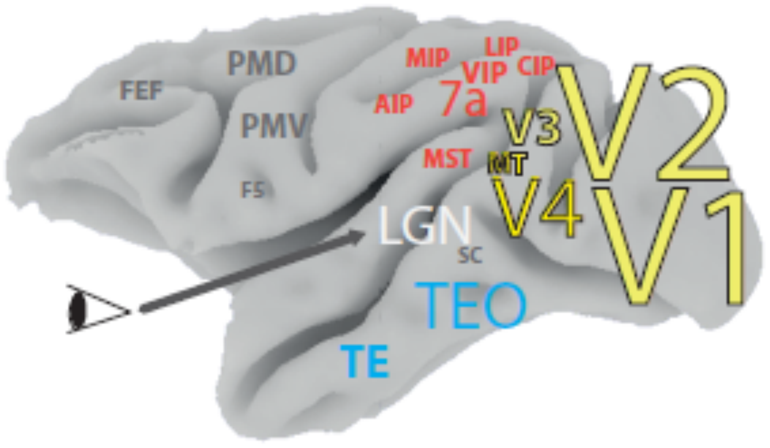






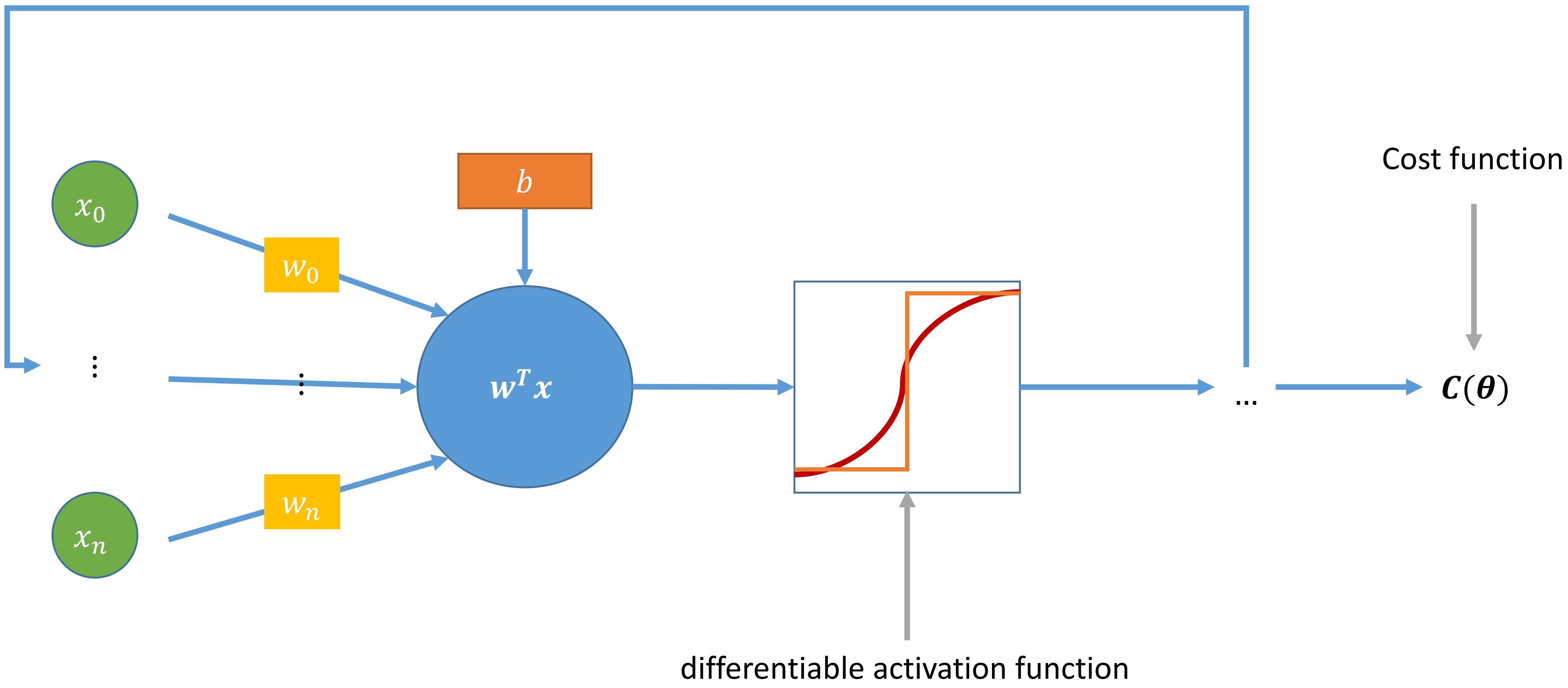






[Kruger, Maersk, Jannssen et al. TPAMI 2012]

Perceptron – Engineering view of the "brain"



Optimization - General procedure

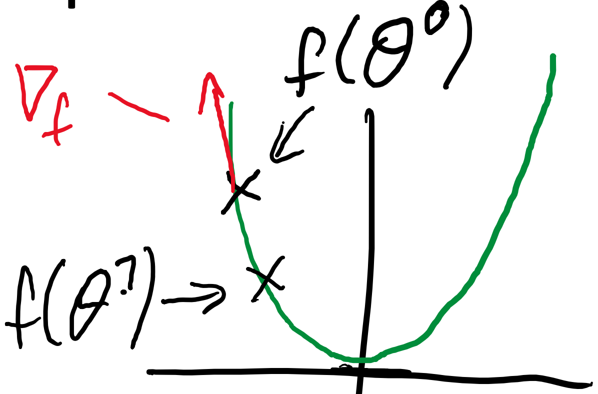
Iterative gradient descent to find parameters Θ:

Initialize weights with small random values

Initialize biases with 0 or small positive values

Compute gradients

Update parameters with SGD



$$\theta^{k+1} = \theta^k - \nabla_f(\theta^k)$$

SGD in a nutshell:

Compute the negative gradient at θ^0

$\Rightarrow -\nabla C(\theta^0)$

Times the learning rate η

$\Rightarrow -\eta \nabla C(\theta^0)$

Chain rule

$$y = g(x) \text{ and } z = f\big(g(x)\big) = f(y)$$

$$\frac{dz}{dx} = \frac{dz}{dy}\frac{dy}{dx}$$

For vector types

$$\frac{\partial z}{\partial x} = \frac{\partial z}{\partial y}\frac{\partial y}{\partial \mathrm{x}}$$

① Backpropagation – single unit

$$* \; f^{(3)} = \|z^{(2)} - y\|^2 = z^{(2)^T} z^{(2)} - 2 z^{(2)} y + y^T y$$

$$\underbrace{\frac{\partial z^{(3)}}{\partial z^{(2)}}}_{----}^{*} = 2 z^{(2)} - 2y = 2(z^{(2)} - y)$$

$$\underbrace{\frac{\partial z^{(2)}}{\partial z^{(1)}}}_{-\cdot-\cdot-} = \sigma(z^{(1)})(1 - \sigma(z^{(1)}))$$

$$\underbrace{\frac{\partial z^{(1)}}{\partial \theta^{(1)}}}_{-\times-\times-} = z^{(0)} = \bar{x}$$

$$\begin{array}{c} z^{(3)} = C(\theta) \\ \uparrow \\ \boxed{f^{(3)} = \|z^{(2)} - y\|^2} \\ z^{(2)} \uparrow \\ \boxed{f^{(2)} = \sigma(z^{(1)})} \\ z^{(1)} \uparrow \\ \boxed{f^{(1)} = \sum \theta^{(1)} z^{(0)}} \\ \uparrow \\ z^{(0)} = \bar{X} \end{array}$$

$$\frac{\partial C}{\partial C} = \frac{\partial z^{(3)}}{\partial z^{(3)}} = 1$$

$$\frac{\partial z^{(3)}}{\partial z^{(2)}} \cdot 1 \leftarrow$$

$$\frac{\partial z^{(2)}}{\partial z^{(1)}} \cdot \frac{\partial z^{(3)}}{\partial z^{(2)}} \cdot 1 \leftarrow$$

$$\frac{\partial z^{(1)}}{\partial \theta^{(1)}} \cdot \frac{\partial z^{(2)}}{\partial z^{(1)}} \cdot \frac{\partial z^{(3)}}{\partial z^{(2)}} \cdot 1 \leftarrow = \frac{\partial C}{\partial \theta} = \underbrace{\frac{\partial z^{(3)}}{\partial z^{(2)}}}_{----} \cdot \underbrace{\frac{\partial z^{(2)}}{\partial z^{(1)}}}_{-\cdot-\cdot-} \cdot \underbrace{\frac{\partial z^{(1)}}{\partial \theta^{(1)}}}_{-\times-\times-}$$

$$= 2(z^{(2)} - y)\,\sigma(z^{(1)})(1 - \sigma(z^{(1)}))\,\bar{x}$$