Implementing ANNs with TensorFlow

Session 01 - Introduction

Agenda

1. Introduction
   A. What are ANNs?
   B. Why do we study ANNs?
   C. Applications of ANNs (Teaser)
   D. Further Research on ANNs
2. Organizational Issues

Part I

Introduction

What are ANNs?

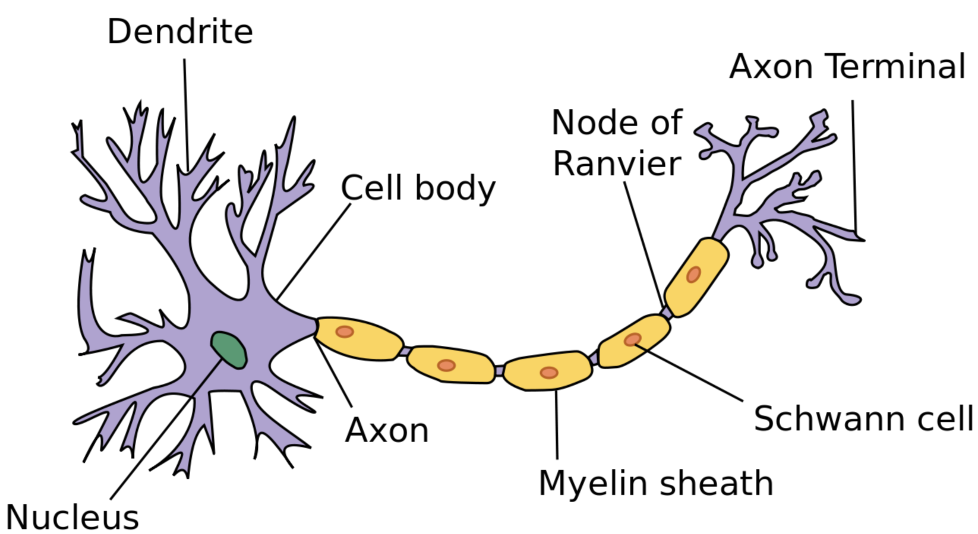
What are ANNs?

- ANN is short for Artificial Neural Network.
- Networks of artificial neurons - in contrast to networks of biological neurons (brains!).

What is an artificial neuron?

- It is **not** about building “real” neurons out of silicon.

- It is about mathematical/computational models of neurons.



[w1]

$$I = C_m \frac{\mathrm{d}V_m}{\mathrm{d}t} + \bar{g}_\mathrm{K} n^4 (V_m - V_K) + \bar{g}_\mathrm{Na} m^3 h (V_m - V_{Na}) + \bar{g}_l (V_m - V_l),$$

$$\frac{dn}{dt} = \alpha_n(V_m)(1-n) - \beta_n(V_m)n$$

$$\frac{dm}{dt} = \alpha_m(V_m)(1-m) - \beta_m(V_m)m$$

$$\frac{dh}{dt} = \alpha_h(V_m)(1-h) - \beta_h(V_m)h$$

Computational Models of Neurons

- Building any kind of model the big question is: Which aspects of the real thing do you want to model?
- There are many different computational models of neurons out there for different purposes!
- e.g. building models to understand how neurons behave in a dynamic system requires quite detailed models of neurons (-> Lecture 8.3047: Neurodynamics).
- This lecture focuses on how networks of neurons can process information!

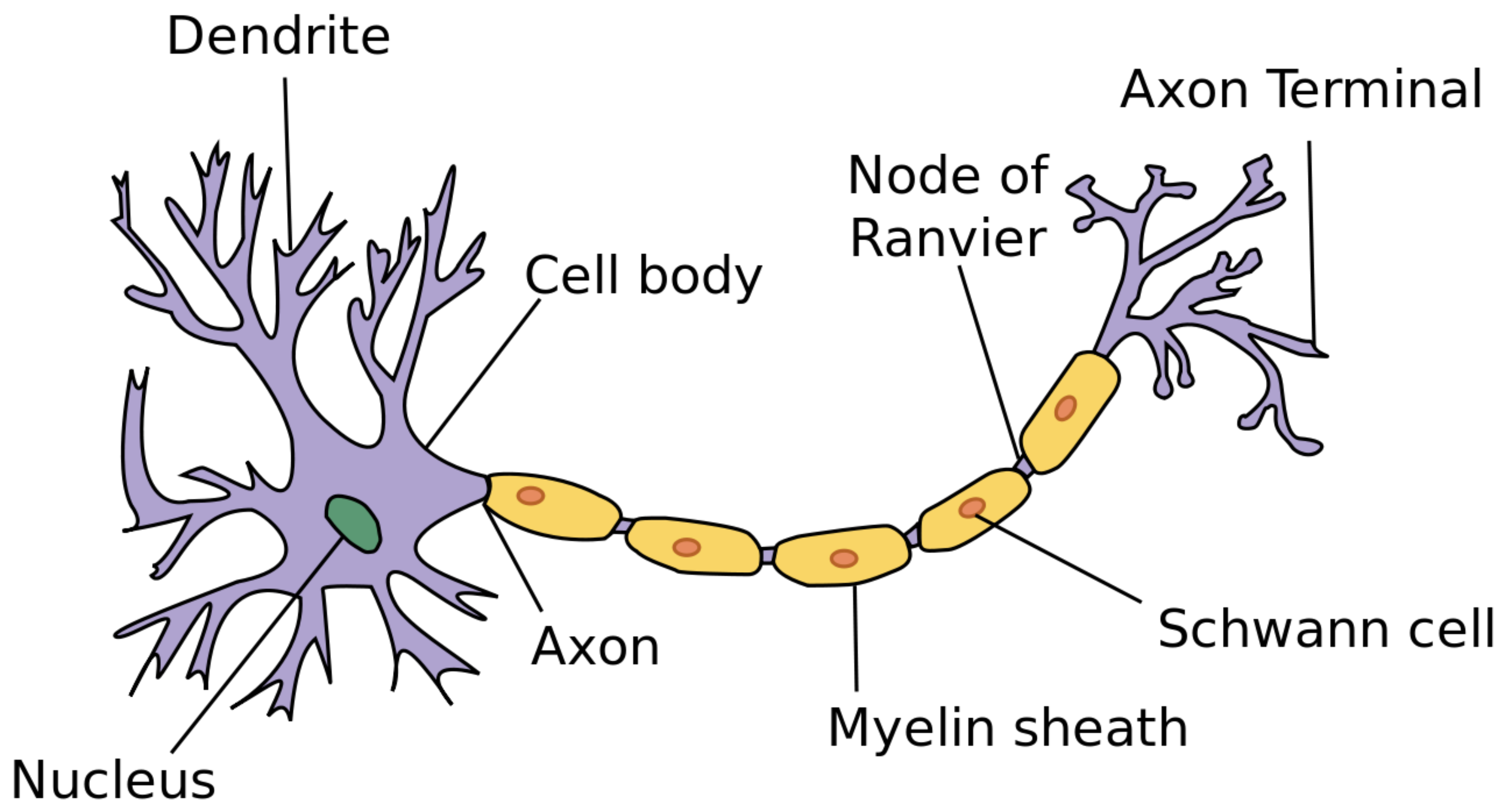
Information Processing with Neurons

- Which properties of neurons do we need to model the information processing abilities of biological neural networks?

- **Up to date this question is not answered!**

- But a very simple (maybe the easiest) model, the **McCulloch-Pitts neuron**, can be quite effective and is the base for the current machine learning hype around **deep learning**.

Neuron

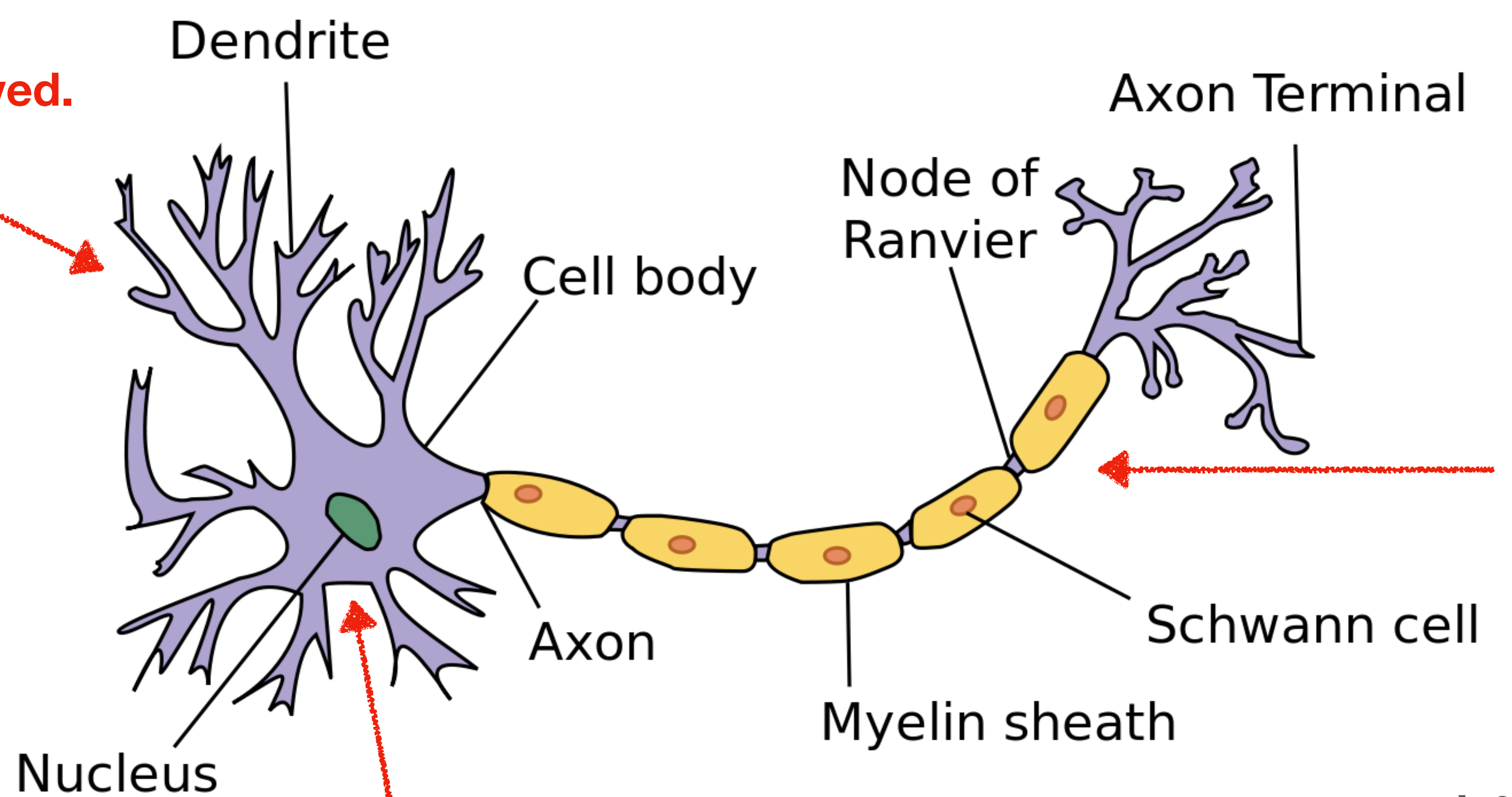

Dendrite
Axon Terminal
Node of Ranvier
Cell body
Schwann cell
Axon
Myelin sheath
Nucleus

[w1]

McCulloch-Pitts Neuron

- The model makes the following (severe) simplifications.

Signals are received.

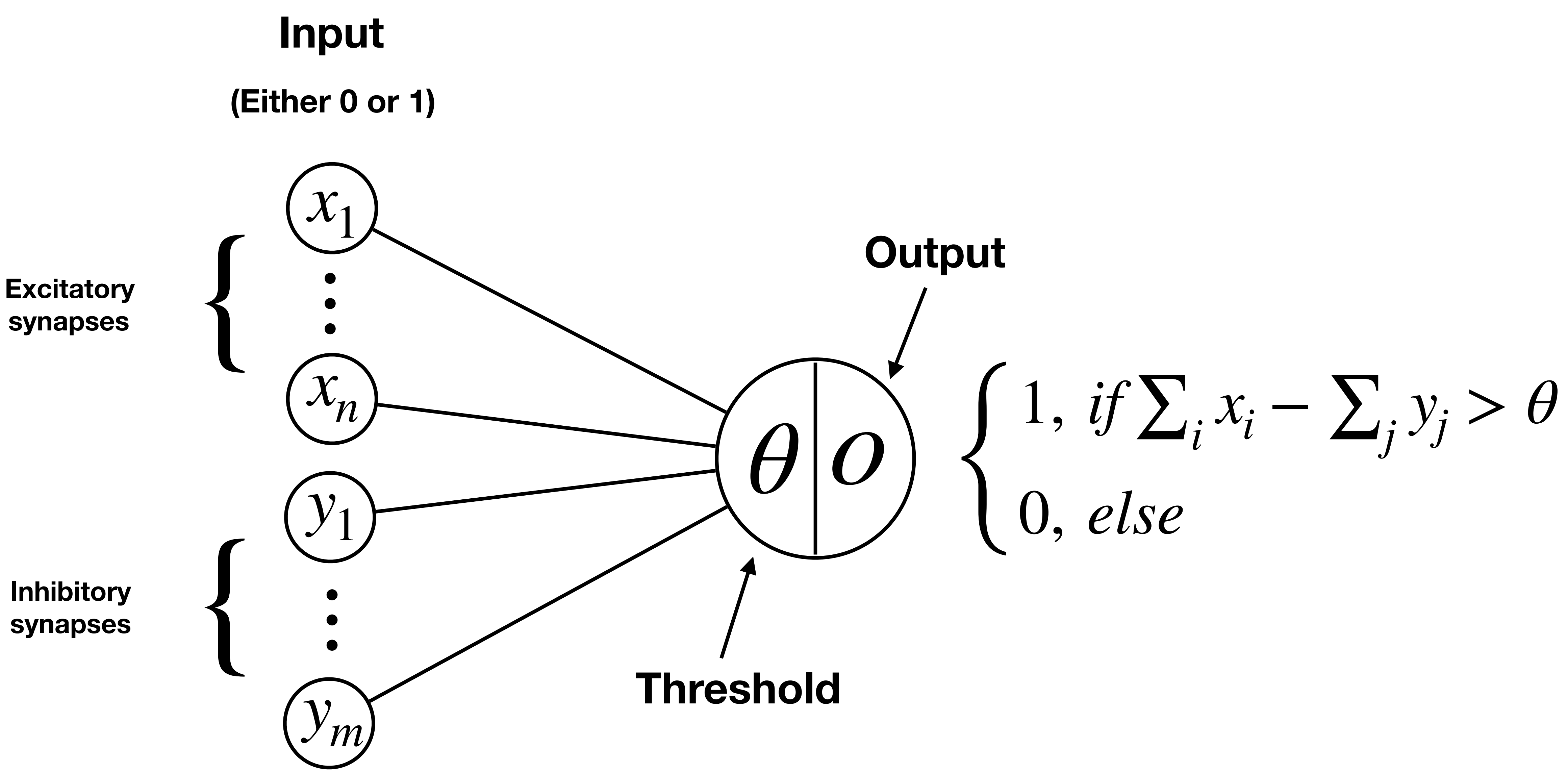


[w2]

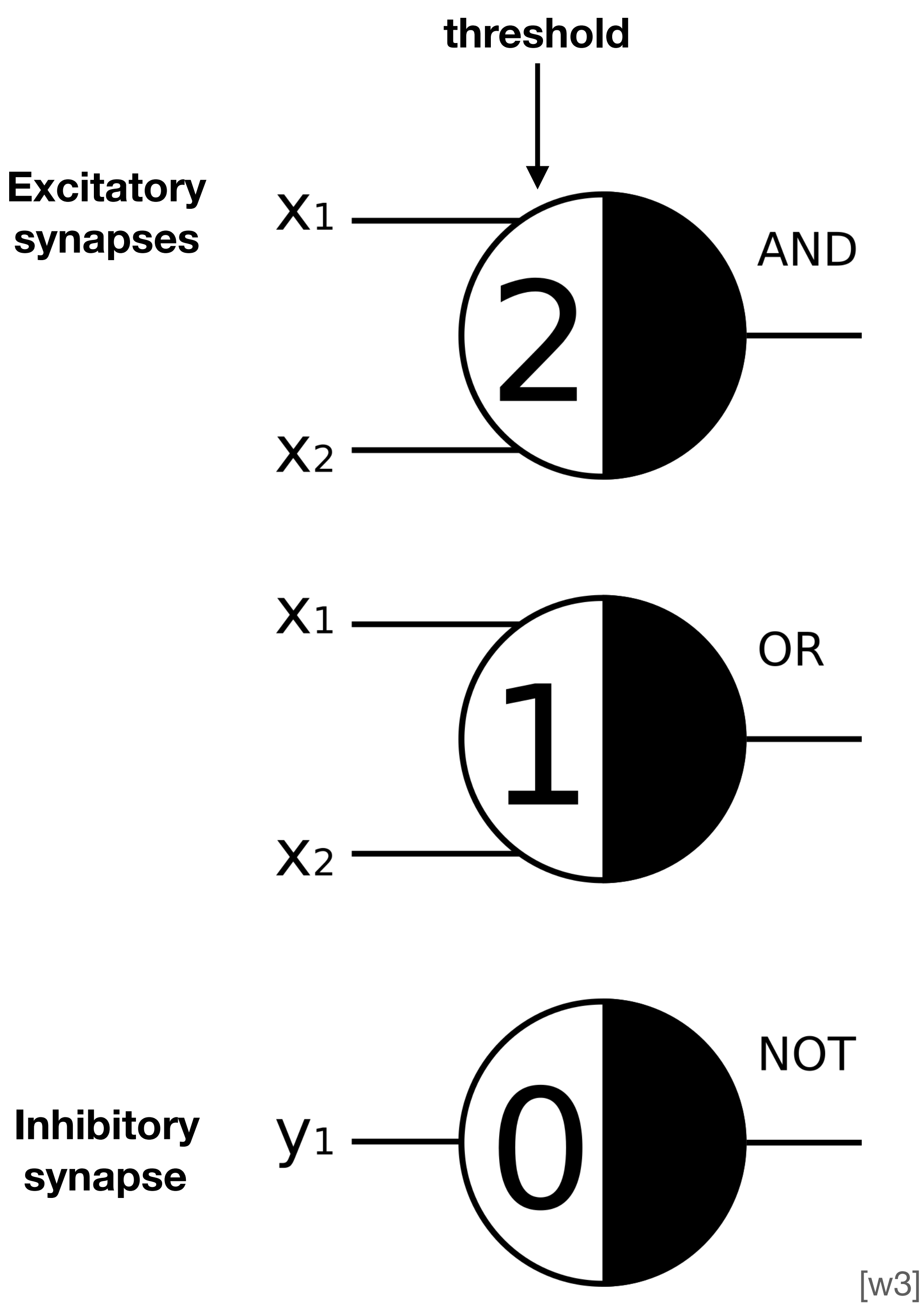
New signal is transmitted and sent to other neurons.

Signals are summed up and activate the neuron, if a certain threshold is surpassed.

McCulloch-Pitts Neuron



Example: Logical Gates



Intuition

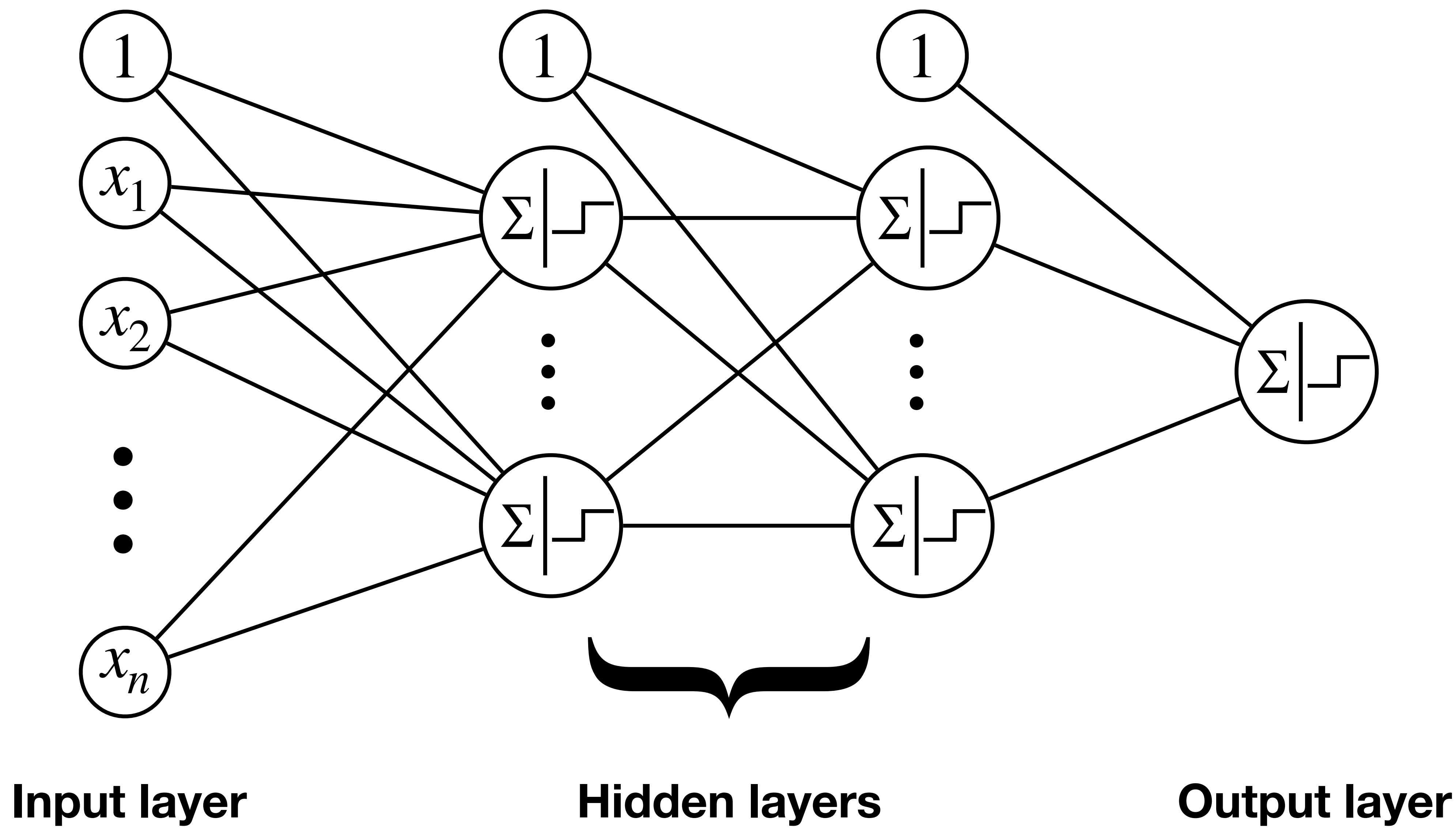
Only if both synapses get inputs their sum is 2. Therefore only in this case the neuron is activated.

The sum gets larger than 1 if at least one synapse gets an input.

If the inhibitory synapse gets an input the “sum” is smaller than 0. Therefore if there is no input the neuron gets activated.

Deep Neural Networks

Large networks based on this basic principle are able to solve complex tasks, that seem to require “intelligence”.



Why do we study ANNs?

Why do we study ANNs?

Results from cognitive (neuro)science help in building better models for AI.

Cognitive (Neuro)Science

ANNs can help in explaining how the brain achieves certain tasks on a neural level.

Artificial Intelligence/ Machine Learning

ANNs are a successful machine learning model for certain kinds of tasks.

Artificial neural network models from AI are used to test computational hypothesis about the brain.

Why do we study ANNs?

Results from cognitive (neuro)science help in building better models for AI.

Focus of this course!

Cognitive (Neuro)Science

ANNs can help in explaining how the brain achieves certain tasks on a neural level.

Artificial Intelligence/ Machine Learning

ANNs are a successful machine learning model for certain kinds of tasks.

Artificial neural network models from AI are used to test computational hypothesis about the brain.

ANNs for Artificial Intelligence

- To understand why we use ANNs in AI and machine learning we have to make a quick detour into AI history.

What does it mean to build an intelligent system?

ANNs for Artificial Intelligence

- To understand why we use ANNs in AI and machine learning we have to make a quick detour into AI history.

What does it mean to build an intelligent system?

AI has a long history of choosing tasks that are assumed to require a certain level of intelligence.

Example: Chess

- One classic example is **chess**.
- As only intelligent people are very good at it, that seems like a good task to start.

Once we built a machine that will beat the greatest chess masters…

Example: Chess

- One classic example is **chess**.
- As only intelligent people are very good at it, that seems like a good task to start.

Once we built a machine that will beat the greatest chess masters…

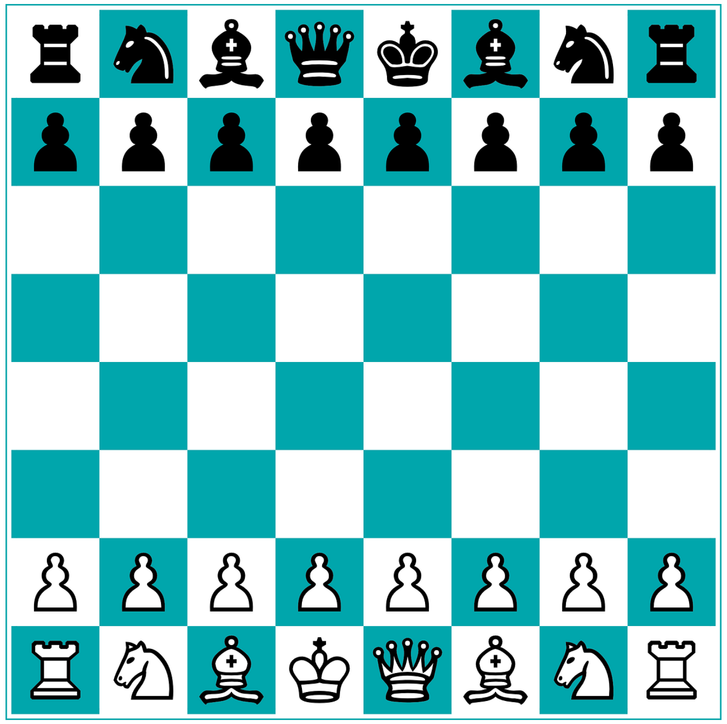
DeepBlue beating Kasparov in 1997!

[f1]

Where is AGI?

The Symbolic Approach to AI

- Idea: Intelligence mainly arises from the ability to manipulate symbols and use logical rules.



How to build a chess AI?

- **Environment**: chess board, different figures
- **Rules**: allowed moves for different figures
- **Heuristics**: e.g. queen is more valuable than horse
- **Algorithm**: how to compute the next move

From today's point of view tasks of that type do not seem hard for computers.

The Symbolic Approach to AI

- Can we use this approach to build AGI?
- **No!** Why?
- You have to know how to solve a problem, to be able to formulate it as an algorithm.
- Turns out humans actually solve a lot of problems without knowing how!

Not available due to copyright reasons.

Any ideas what I mean by that?

Example: Handwritten Digit Recognition

Which digit can be seen in the image on the right?

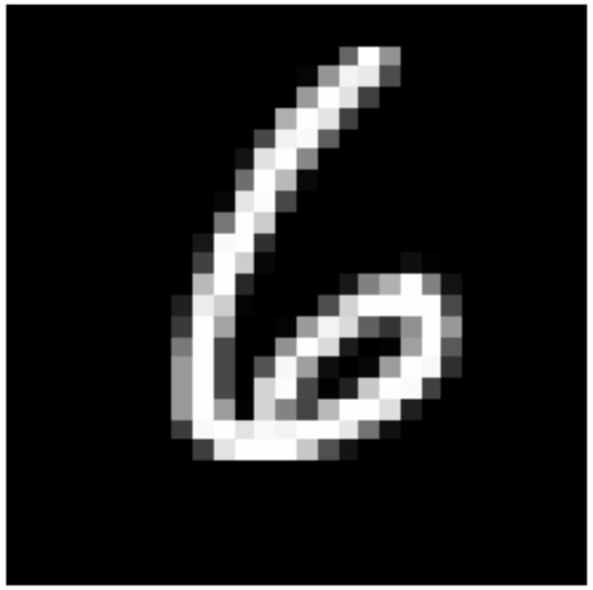
A six.

How do you know it is a six?

Well I can see a bow that is curved to the right and circle attached to it at the bottom.

But how do you know that there is a bow and a circle and that the circle is attached to the bow at the bottom.

Well uuh…



[ylc]

Although we are very familiar with handwritten digits, it is hard to formulate how we actually recognize it.
(example for Moravec's paradox)

Example: Handwritten Digit Recognition

Finally you would need to be able to provide a formal algorithm that **given the pixels of an image - returns the correct digit label.**

[w4]

Sub-symbolic Approach to AI

- **The intelligence to solve tasks like that is not symbolic.**
- It emerges from **simple computational units** (like neurons in the brain).
- Can we built an AI that learns itself how to solve these tasks?

Not available due to copyright reasons.

3Blue1Brown Series on Neural Networks

Intermediate Conclusion

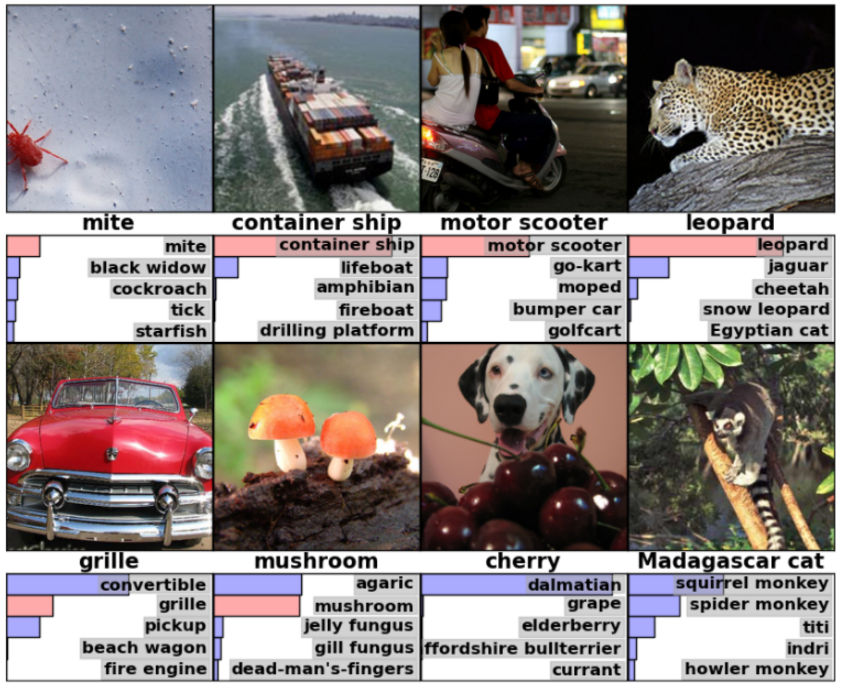
Tasks that are seemingly very simple for humans have solutions that are not trivial to formalize.

We will use Artificial Neural Networks to solve them!

Applications and Tasks

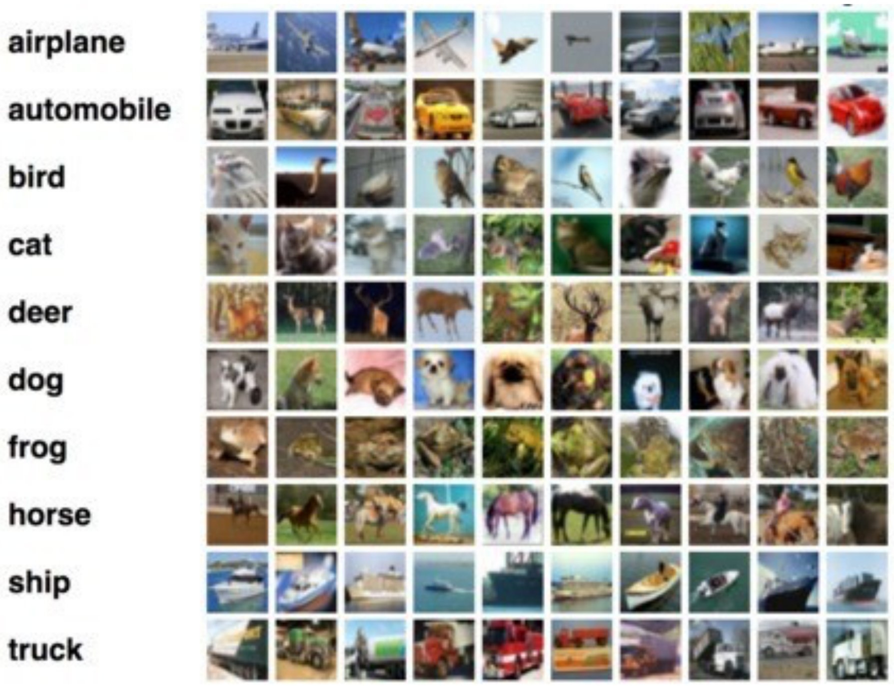
Object Recognition

- Going beyond handwritten digit recognition…



[ak1]

ImageNet



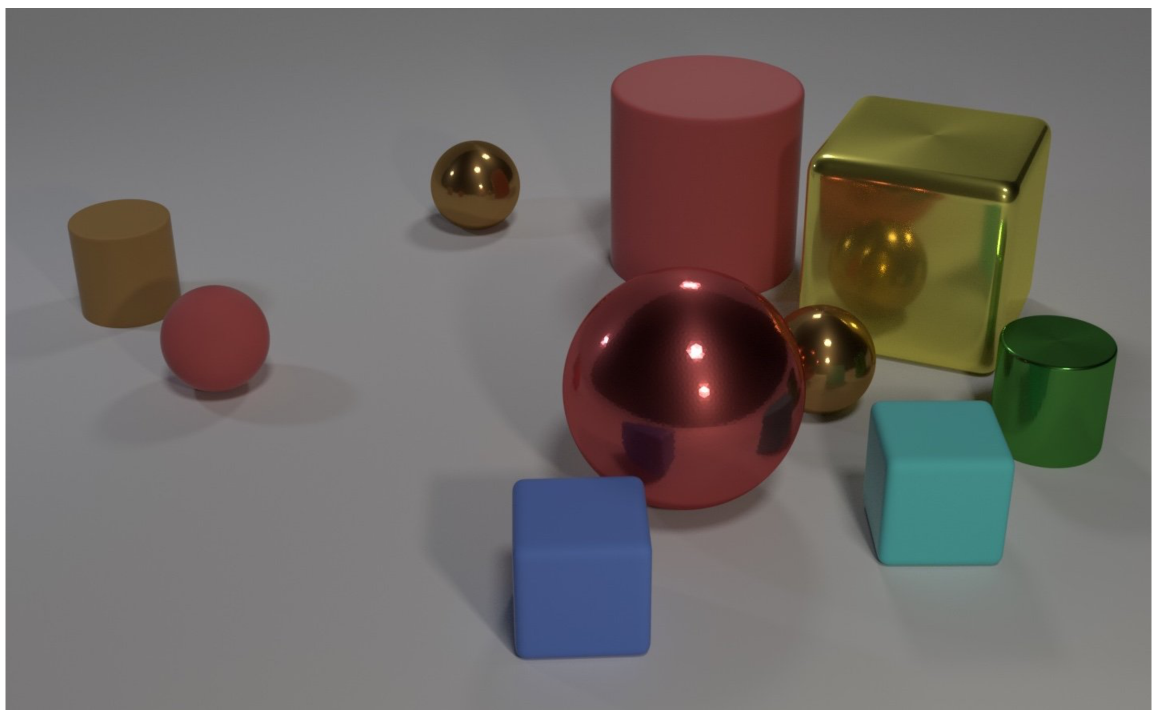
[ak2]

CIFAR10

Other Computer Vision Task

[rm]

PASCAL (Semantic Segmentation)



[ji]

CLEVR (Visual Reasoning)

Example question:

Are there an equal number of large things and metal spheres?

Image Generation

[tk]

Face Generator

thispersondoesnotexist.com

Deepfakes

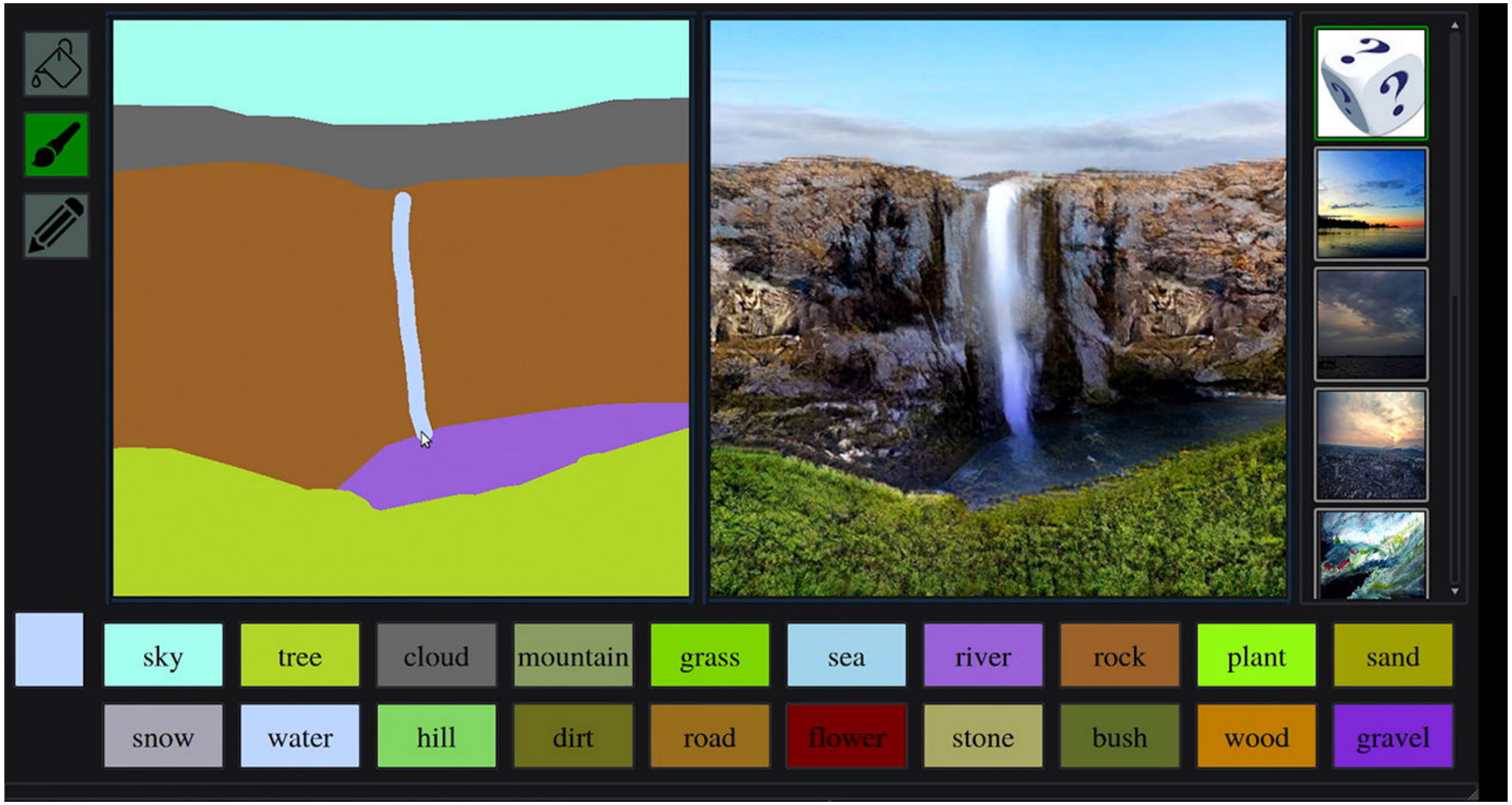
https://www.youtube.com/watch?v=cQ54GDm1eL0

Image Generation

[ab]

Natural Image Generator

Cool Application: GauGAN



[tp]

Try it yourself!

http://nvidia-research-mingyuliu.com/gaugan/

Text-to-Speech

Parametric

Concatenative

WaveNet (Artificial Neural Network)

Speech generation without text

[dm1]

Text Comprehension

The city councilmen refused the demonstrators a permit because they [feared/advocated] violence.

Given one of the two words in the squared brackets to which of the two subjects is “they” referring?

“feared” ⟶ **“councilmen”**

“advocated” ⟶ **“demonstrators”**

Generate Text

SYSTEM PROMPT (HUMAN-WRITTEN)	*In a shocking finding, scientist discovered a herd of unicorns living in a remote, previously unexplored valley, in the Andes Mountains. Even more surprising to the researchers was the fact that the unicorns spoke perfect English.*
MODEL COMPLETION (MACHINE-WRITTEN, 10 TRIES)	The scientist named the population, after their distinctive horn, Ovid's Unicorn. These four-horned, silver-white unicorns were previously unknown to science. Now, after almost two centuries, the mystery of what sparked this odd phenomenon is finally solved. Dr. Jorge Pérez, an evolutionary biologist from the University of La Paz, and several companions, were exploring the Andes Mountains when they found a small valley, with no other animals or humans. Pérez noticed that the valley had what appeared to be a natural fountain, surrounded by two peaks of rock and silver snow. Pérez and the others then ventured further into the valley. "By the time we reached the top of one peak, the water looked blue, with some crystals on top," said Pérez. Pérez and his friends were astonished to see the unicorn herd. These creatures could be seen from the air without having to move too much to see them – they were so close they could touch their horns.

[oa1]

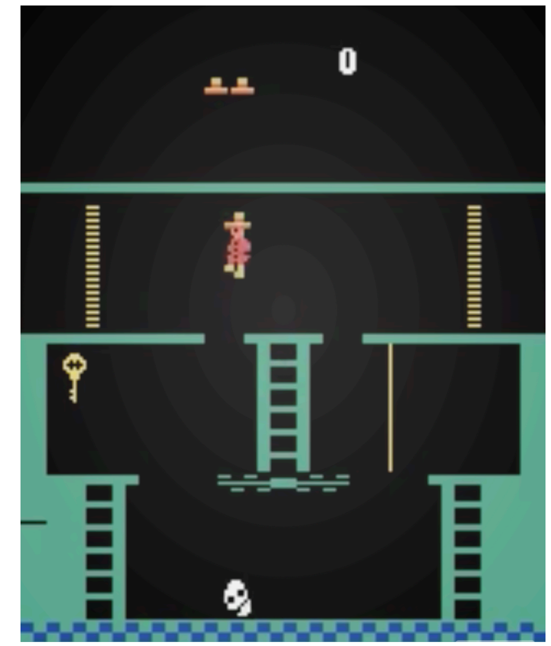
https://talktotransformer.com

Playing Games

StarCraft

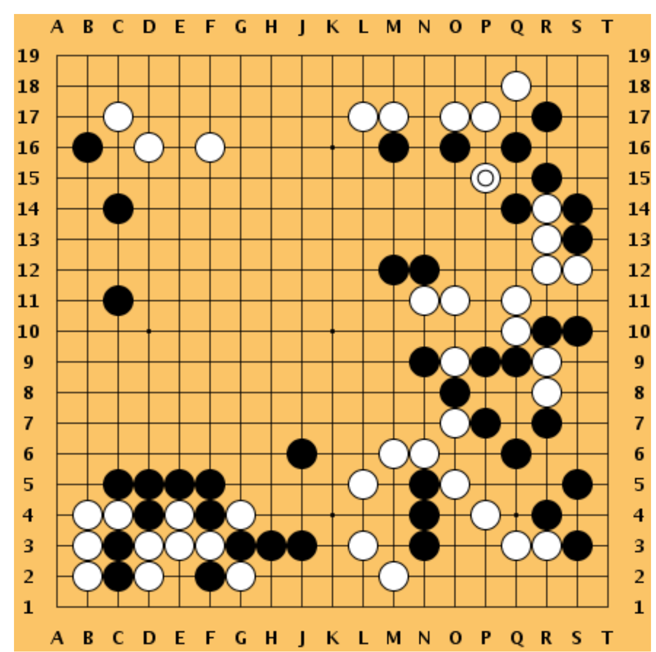
[dm2]

Atari Games



[oa2]

Go



Robotics

https://www.youtube.com/watch?v=kVmp0uGtShk

Research

Black Box

- If you ask a neural network **why** it did **what** it did you won't get any answer.
- A neural network is just thousands or millions of numbers. There are no logical rules.
- Problematic if you want to use neural networks for important decisions.

Why didn't I get that credit?!

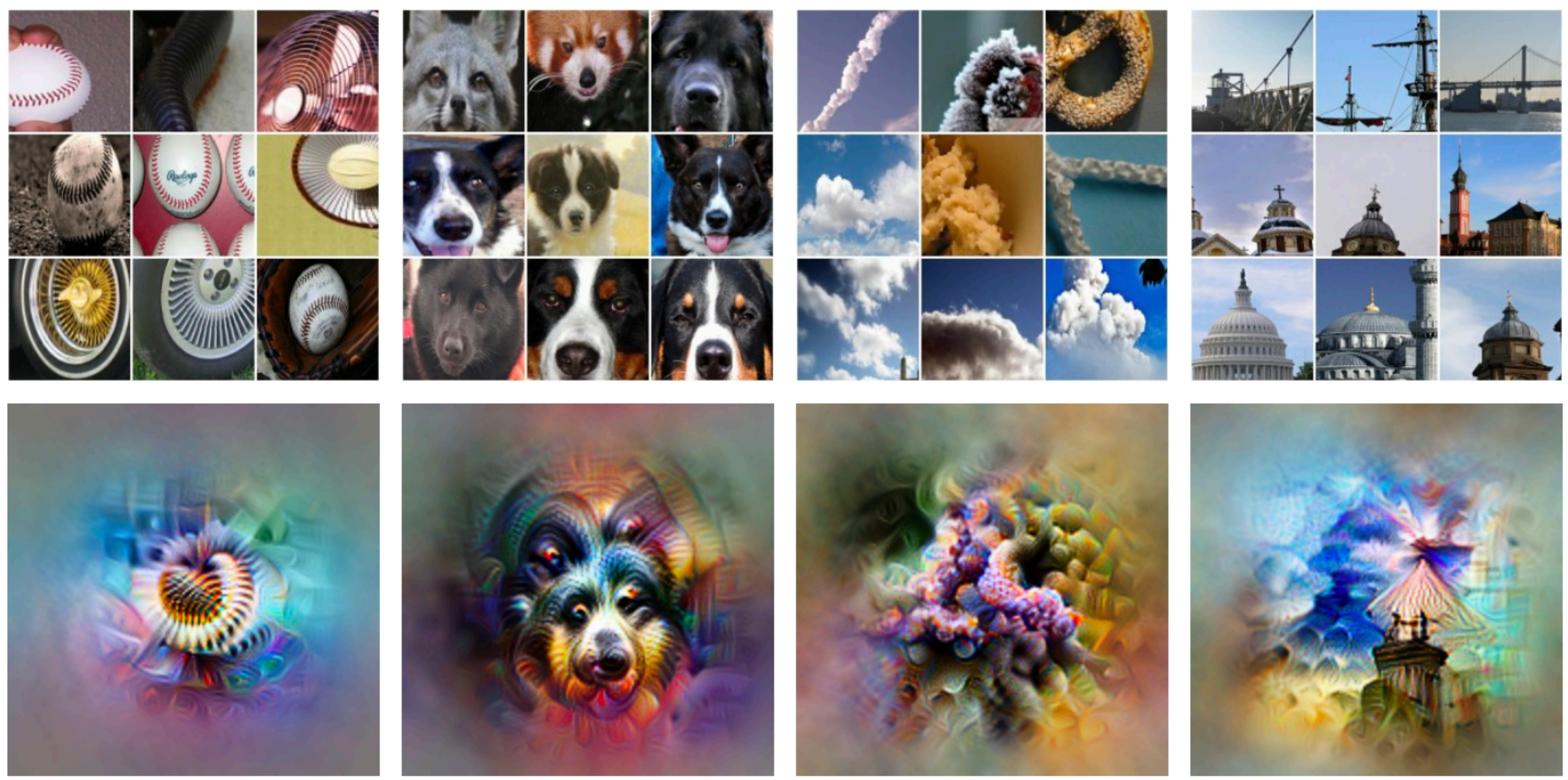
Not available due to copyright reasons.

Feature Visualization

In computer vision there are efforts to bring forward so called explainable AI by visualizing what neurons in a DNN are “seeing”.

Dataset Examples show us what neurons respond to in practice

Optimization isolates the causes of behavior from mere correlations. A neuron may not be detecting what you initially thought.



Baseball—or stripes? *mixed4a, Unit 6*

Animal faces—or snouts? *mixed4a, Unit 240*

Clouds—or fluffiness? *mixed4a, Unit 453*

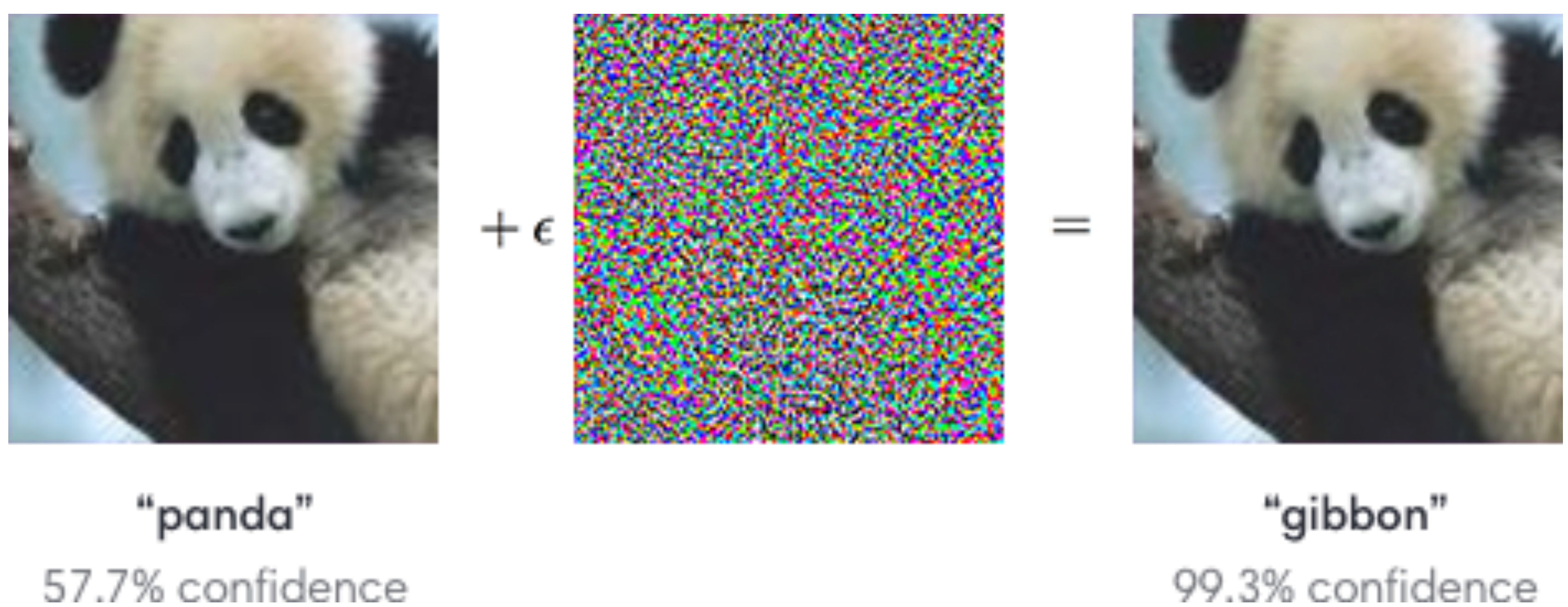
Buildings—or sky? *mixed4a, Unit 492*

[co]

Distill.pub article

Adversarial Attacks

But there is also research showing that these networks do not behave as we would like them to!

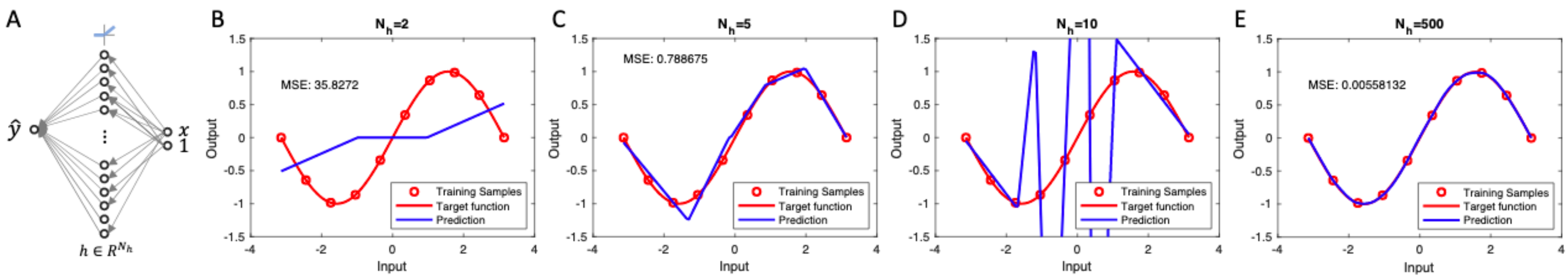


[ig]

Why do ANNs learn that good actually?

- It is also not quite clear why actually ANNs work as well as they do.

- Neural networks are heavily overparameterized, still they find good solutions.



[yb]

Part II

Organizational Issues

Team

Luke

Sahar

Leon

About Me

- PhD Candidate in Gordon Pipa's Neuroinformatics Lab.
- My work focuses around how to teach ANNs to learn more human-like. (If you're interested in bachelor or master theses regarding this topic, write me an E-mail!)
- Previously M.Sc. Cognitive Science in Osnabrueck and B.Sc. Mathematics at HHU Duesseldorf.

Course Goal

- This course is the only one covering deep neural networks.

- The course tries to teach you the basics, such that you yourself know how to proceed if you want to dive deeper:

  1. Basic concepts of deep learning.

  2. Implementing deep neural networks with TensorFlow.

  3. Introductions to various applications and research possibilities.

Sessions

- **Lecture: Tuesday 16.15-17.45 in 32/110**

Mostly theoretical background on artificial neural networks and their applications. Sometimes implementational topics. Held by me.

- **Homework Tutorial Session: Wednesday 12.15 - 13.45 in 35/E01**

Tutorial session to discuss the last homework. Presented by Sahar.
Starting on 06.11.19/13.11.19 (infos will follow).

- **Open Q&A Homework Session: Fridays 14.15 - 15.45 in 35/E16**

Homework session to work on the current homework. Supervised by Leon.
Starting on 08.11.19.

Recordings

- The lecture should usually be recorded.
- **Do not bet on it! It often fails!**
- Tutorial and QnA are not recorded.

Course Website

- The course is hosted on the website:

  https://lukeeffenberger.github.io/IANNWTF-2019/

- The recordings will probably be on Stud.IP only though.

Prerequisites

- There are no formal prerequisites!
- This does not mean that this course will be easy for all of you!
- I presuppose:
  - Basic coding skills in Python (incl. NumPy).
  - Basic mathematical understanding of linear algebra and calculus.

The homework this week will include questions for you to get on board!

Setup

- You need to program for this course.
- We will use Python, NumPy, Jupyter Notebook and TensorFlow 2.0.
- The homework includes some high-level guidance on how to setup everything on your laptop.
- I recommend using Google Colab - computing in the cloud; no need to install anything! (You need a Google account though.)

Requirements

This course gives 8 ECTS. To get those there are three formal requirements.

A. Homework

B. Midterm

C. Final Project

Homework

- There will be a homework each week to apply what you have learned in the lecture.
- Homework will be mostly coding (first two weeks in Python, then in TensorFlow), but might sometimes include theoretical questions.
- Published as a Jupyter notebook! (On website and directly via Colab.)
- Homework will be published after the lecture (Tuesday 18.00).
- Deadline is before the next lecture (Tuesday 14.00).

Homework Groups

- Please find a group to work on the homework.
- Groups of 3.
- If you can’t find anyone, you can write something in the “Forum” for this course on Stud.IP.
- Please enroll in a group on Stud.IP:
  Participants -> Groups

Homework Corrections

- The homework corrections this year are somewhat experimental.
- **You have to rate your own homework!**
- There are four possible ratings:
  - Outstanding
  - Done
  - Not Done
  - Fail

PASS

There will be 10-12 homeworks. You have to pass at least eight of them!

Homework Ratings

- Outstanding: Your homework runs perfectly. You put a lot of effort in cleaning up your code and using sensible variable names. Other students would benefit from looking at your code.

  - This rating will be double-checked by Leon & Sahar.

  - If the rating is approved, you will get 0.05+ for the final grade (for each outstanding homework), which might lift you one grade up in the end.

  - The homework will finally be published for other students to benefit from it.

Homework Corrections

- Done: You solved the homework. It runs and does what it should. But besides that your code is kind of messy. Others should not use it as guidance.
    - This rating will only randomly be double-checked by Leon & Sahar.
- Fail: You did not really work on the homework and did not solve it.
    - In this case you do not submit anything.

Homework Corrections

- Not Done: You put effort into solving the homework, but you couldn’t manage it.
  - Here you have to include a short statement in your submission on the problem you were not able to solve and what you tried to solve it.
  - When the homework solution is published you have one week to submit another short statement on whether and if how your problem was resolved.
  - Both statements will be reviewed by Leon and Sahar. If both statements are sufficient the rating “Not Done” counts as passing the homework.

Cheating

- Please be honest!
- If you submit a homework as **Outstanding** or **Not Done** but we downgrade it to **Pass** or **Fail** nothing will happen as long as you are not trying to cheat. You will get a mail explaining what your submission missed.
- If you constantly or clearly give a wrong rating you will get a **Fail**.
- If you submit a homework as **Done** although your code is not running this counts as a **Fail** and we will subtract one **Outstanding**!

Homework Submission

- You can submit your homework directly via Colab:
  Visit Sahar's tutorial on 6th of November.

- Or you can submit it via Stud.IP in your group folder.
  Please stick to the following naming convention:

```
G{group_number}_H{homework_number}_{rating}.ipynb
```

Outstanding - O
Done - D
Not Done - ND
Fail - F

Examples

```
G01_H03_O.ipynb
G15_H07_ND.ipynb
...
```

Midterm Exam

- There will be a midterm exam shortly before the christmas break.
- Presumably on 17.12./18.12.
- The exam will be designed such that you will easily succeed, if you actively took part in the course (visiting the lectures, working on the homework).
- No coding. It will test your understanding of basic concepts around deep learning.
- There will be no retake, but oral exams. Only if you failed.
- If you are sick on that day, please bring a certificate from a doctor.

Final Project

- Instead of a final exam you will have to work on a final project.
- Deadline: 30.03.2020. (You have the whole semester break!)
- You will write up your project in a blog post.
- For those who are okay with it, I would like to publish your projects as a resource for understanding how to implement stuff in TensorFlow 2.0

Topics for Final Project

There are two possibilities for your final project:

1. Reimplementing a paper. (**Strongly recommended!**)
    - You choose a paper you're interested in (I will supply you with a list for possible ones) .
    - You understand the paper and implement it in TensorFlow 2.0.
    - In the blog post you explain the important parts of the paper and how you implemented it.

2. Implementing your own application. (**Not recommended!**)
    - You have your own idea about a task that you would like to solve.
    - You make this task the center of your project and try to find a solution.
    - In the blog post you explain the task and how you solved it.
    - If you can't solve it you have to come up with a hypothesis for why that is and explain what you tried to solve it.

Topics for Final Project

- I will regularly update a list of possible topics on the website.
- Feel free to approach me after the lectures, if you want to talk about an idea.
- For all paper and tasks it is first-come-first-serve.
- **Deadline for fixing a topic: 14.02.2020**

Grade

- The final grade will emerge as follows:
    - 65% final project
    - 35% midterm exam
    - + bonus for outstanding homework

Content

Before Christmas break: Basics

Week	Topic
1	Introduction
2	Perceptron, MLP
3	Backpropagation, Gradient Descent
4	TensorFlow
5	More on Deep Neural Networks
6	Convolutional Neural Networks
7	Training Deep Neural Networks

Content

After Christmas break: Introduction to various applications.

Possible Topics
Advanced Large Scale CNNs
Recurrent Neural Networks, LSTMs
Unsupervised Learning (VAEs, GANs)
Transformer Networks
Deep Reinforcement Learning
Feature Visualization & Adversarial Attacks
One-Shot Learning
Active Research Topics in Deep Learning

Feedback

- I am always happy about constructive feedback (regarding everything).
- Either approach me after the lecture or write me an Email.
- If you wish to stay anonymous, feel free to approach Leon or Sahar.

Questions

- Organizational questions:
  - To me: leffenberger@uos.de
- Content/Homework questions:
  - To Leon or Sahar: lschmid@uos.de, sniknam@uos.de

Please use always the following subject line:

[TF]*yourSubjectLine*

Acknowledgement

Lastly would like to thank Lukas Braun, who when he was a Bachelor student, designe the outline of this course.

Not available due to copyright reasons.

Any questions left?

See you next week!

Resources

[w1] Quasar Jarosz at English Wikipedia (https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg)

[w2] Adapted from Quasar Jarosz at English Wikipedia (https://commons.wikimedia.org/wiki/File:Neuron_Hand-tuned.svg)

[w3] Adrian Lange at German Wikipedia (https://commons.wikimedia.org/wiki/File:Diagram_of_a_McCulloch-Pitts-cell_showing_boolean_functions_and,_or_and_not.svg)

[f1] Jonathan V at Flickr (https://www.flickr.com/photos/hugonorsing/97830254)

[ylc] Y. LeCun et al., "Gradient-based learning applied to document recognition." *Proceedings of the IEEE*, 86(11):2278-2324, November 1998.

[w4] Josef Steppan at English Wikipedia (https://commons.wikimedia.org/wiki/File:MnistExamples.png)

[ak1] A. Krizhevsky et al., "Imagenet classification with deep convolutional neural networks" *Proceedings of the NeurIPS 2012*, 2012.

[ak2] A. Krizhevsky, G. Hinton. "Learning multiple layers of features from tiny images", 2009.

[jj] J. Johnson et al., "CLEVR: A Diagnostic Dataset for Compositional Language and Elementary Visual Reasoning", *2017 IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

[rm] R. Mottaghi et al., "The Role of Context for Object Detection and Semantic Segmentation in the Wild", *2014 IEEE Conference on Computer Vision and Pattern Recognition*, 2014.

[tk] T. Karras et al., "A Style-Based Generator Architecture for Generative Adversarial Networks", *2019 IEEE Conference on Computer Vision and Pattern Recognition*, 2019.

Resources

[ab] A. Brock et al., "Large Scale GAN Training for High Fidelity Natural Image Synthesis", *ICLR 2019*.

[tp] T. Park et al., "Semantic Image Synthesis with Spatially-Adaptive Normalization", *2019 IEEE Conference on Computer Vision and Pattern Recognition, 2019*.

[dm1] https://deepmind.com/blog/article/wavenet-generative-model-raw-audio

[oa1] https://openai.com/blog/better-language-models/

[dm2] https://deepmind.com/blog/article/alphastar-mastering-real-time-strategy-game-starcraft-ii

[oa2] https://gym.openai.com/envs/MontezumaRevenge-v0/

[co] C. Olah et al., "Feature Visualization", *Distill,* 2017.

[ig] I. Goodfellow et al., "Explaining and Harnessing Adversarial Examples", *ICLR,* 2014.

[yb] Y. Bansal et al., "Minnorm training: an algorithm for training over-parameterized deep neural networks", *arXiv:1806.00730,* 2018.