

Deep Learning

Grundlagen Grundbegriffe Grundübungen

Alfatraining Kurs 2025-08 Dozent: Karsten Flügge

Einführung

**Begrüßung, Vorstellungsrunde, Einführung in die
Unterrichtsform, alfaview & digitale Lernumgebung**

Einführung Deep Learning

Kleine Historie Machine Learning

Mathematische Hintergründe

Notwendige Grundlagen

Deep Learning als eine Art von Machine Learning

Intelligence $\approx$ Intelligenz

AI, Künstliche Intelligenz

SOTA State of the Art

Überblick SOTA

Kleine Geschichte

1960 trucks vs cars Eliza

1980 AI Winter (ging nicht recht weiter mit Theorie)

2012 AlexNet (8 Schichten tiefes Netz für Bilderkennung)

Was brachte den Durchbruch:

mehr Rechenpower mit GPUs

Theoretische / Praktische Lösungen von vorigen Problemen:

Exploding / Vanishing Gradient

Over / Underfitting

Deep Learning (viele Schichten)

dank dropout, residuals

Themen des Kurses

Woche 1

Tag 1

Grundlagen

Hello World

Tag 1

Perceptron

Parameter, Optimierung

Rechenschemen: Backpropagation

Tag 3

Tensor

Bibliotheken

Regression vs. Klassifikation <<

Lernkurven

Tag 4

Überanpassung und Regularisierung

Hyperparameter-Optimierung

Stochastischer Gradienten Abstieg (SGD) Descent

Tag 5

Momentum

Adam Optimizer

Lernrate

Themen des Kurses

Woche 2

Tag 6

Convolutional Neural Network (CNN) Bildklassifizierung
CNN-Architekturen ImageNet-Competition
Convolutional-Schichten, Pooling-Schichten
Reshaping-Schichten, Flatten, Global-Average-Pooling

Tag 7

Vertiefung Regularisierung
Tiefe neuronale Netze, Vanishing Gradients
Skip-Verbindungen, Batch-Normalization

Tag 8

Transfer Learning
Finetuning: Anpassen von Modellen
Unüberwachtes Vortrainieren
Image-Data-Augmentation, Explainable AI

Tag 9

Regional CNN
Objekterkennung und Lokalisierung
Regressionsprobleme
Verzweigte neuronale Netze

Tag 10

Methoden der kreativen Bilderzeugung
Generative Adversarial Networks (GAN)
Deepfakes
Diffusionsmodelle

Themen des Kurses

Woche 3

Tag 11

Recurrent Neural Networks (RNN) Sequenzanalyse, Rekurrente Schichten
Backpropagation through time (BPTT)
Analyse von Zeitreihen

Tag 12

Exploding und Vanishing Gradient Probleme
LSTM (Long Short-Term Memory), GRU (Gated Recurrent Unit)
Deep RNN, Deep LSTM

Tag 13

Textverarbeitung durch neuronale Netze
Text-Preprocessing, Embedding-Schichten
Text-Klassifizierung, Sentiment-Analyse
Transfer Learning in NLP

Tag 14

Sequence-to-Sequence Verfahren,
Encoder-Decoder Architektur
Übersetzungen

Tag 15

Sprachmodelle
BERT, GPT
Attention-Schichten, Transformers, Textgeneration-Pipelines
Summarization, Chatbots

Themen des Kurses

Woche 4

Tag 16

Deep Reinforcement Learning

Steuerung dynamischer Systeme

Agentensysteme, Training durch Belohnungen

Spiel Agenten

Policy Gradients,

Deep-Q-Learning

Tag 17

Bonus je nach Interesse, z.B.:

Bayes'sche neuronale Netze

Unsicherheiten in neuronalen Netzen

Statistische Bewertung von Prognosen, Konfidenz, Standardabweichung

Unbalancierte Daten, Sampling-Methoden

Letzte 3 Tage

Projektarbeit

AI Grundverständnis

Legendär verständlich sind die online Kurse von Andrew Karpathy (youtube/blogs)

Auf verschiedenem Niveau: Von Anfänger bis Experte (trotzdem recht verständlich)

Koryphäen:

- **Geoff Hinton** **Nobelpreis für Physik Boltzmann Modelle**
- **Andrew Ng**
- Yann LeCun
- Yoshua Bengio
- Jürgen Schmidhuber (hat alles schon 1990 erfunden)

(es ist es Wert, von jedem das beste Paper / Video anzusehen)

Erfolg 2010...

Gelöste Probleme der AI:

Vanishing / Exploding Gradient heisst numerische Instabilitäten

Overfitting (no more early stopping)

Lösungen:

Regularisierung

- **Dropout** (zufällige Deaktivierung von Perzeptronen, Verteilung des Wissens "Multiplikatoren")
- **Residual** Connections (Schicke Signal unverändert zu tieferen Schichten "Überbleibsel der original Informationen des Input")
- Batch Normalization (Mean/Zentrum wird subtrahiert und Varianz dividiert)
- Convolutional Networks
- Architecture does not matter (**mehr compute + mehr trainingsdaten**) Andrew Ng
- Precision does NOT matter
 - Last paper: 1.8 bit averaged information is **better(more efficient) than** f16, f32
 - Quantisierte LLMs **4 bit** "floats" pseudo reals bit $\approx$ Eigenschaft ja/nein
 - 7 GB 30 GB LLMs sind brauchbar für Spezialanwendungen auf eigenem Rechner

20 questions

SOTA State Of The Art

Stand der Dinge

LLM Large Language Models Sprache 'gelöst'

Multimodal Text, Ton & Bild (bald Video)

Bild Erzeugung, Erkennung und Beschreibung

Video Erzeugung (<https://openai.com/index/sora/> ...) Gemini

3D Modell Erzeugung (Gaussian Splatter, schlechte Topographie)

LLMs als generelle Agenten (Software, Spiele)

Mathematik Olympiade Gold Medaille *Lean*

Mathematik erster eigenes Theorem und Beweis

Kaggle Wettbewerbe

LLM Arena Elo

Saturated Benchmarks

ARC AGI

Small LLMs auf eigenem Rechner



SOTA State Of The Art

Wie gut ist mein Netzwerk?

Baseline (Richtline)

Minimales Grundmodell, oder SoTA

Data is King

Andrew Ng: "Wenn wir genug Daten haben und genug Compute, dann spielt die Architektur eine Nebenrolle".

Precision does not matter!

Float64 Float32 -> "Quantisieren" auf 1.8

Funktioniert *fast* genauso gut.

Bugs werden oft umgangen!

Bei Fehlern übernehmen oft andere Bereiche die Aufgabe.

(Vor und Nachteil)

SOTA State Of The Art

LLM Arena Elo "Blindbewertungen"
<https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard>

Rank* (UB)	Model	Arena Score	95% CI	Votes	Organization	License	Knowledge Cutoff
1	GPT-4o-2024-05-13	1287	+3/-3	65952	OpenAI	Proprietary	2023/10
1	GPT-4o-mini-2024-07-18	1279	+9/-8	6849	OpenAI	Proprietary	2023/10
2	Claude 3.5 Sonnet	1272	+4/-3	36080	Anthropic	Proprietary	2024/4
3	Gemini-Advanced-0514	1266	+2/-3	48674	Google	Proprietary	Online
4	Gemini-1.5-Pro-API-0514	1261	+3/-3	58617	Google	Proprietary	2023/11
5	Gemini-1.5-Pro-API-0409-Preview	1257	+4/-3	55674	Google	Proprietary	2023/11
5	GPT-4-Turbo-2024-04-09	1257	+3/-2	77358	OpenAI	Proprietary	2023/12
8	GPT-4-1106-preview	1251	+3/-3	88859	OpenAI	Proprietary	2023/4
8	Claude 3 Opus	1248	+3/-2	148688	Anthropic	Proprietary	2023/8
8	GPT-4-0125-preview	1245	+3/-3	82138	OpenAI	Proprietary	2023/12
8	Athene-70b	1240	+10/-8	3477	NexusFlow	CC-BY-NC-4.0	2024/07
10	Yi-Large-preview	1240	+3/-3	50864	01 AI	Proprietary	Unknown

Small LLMs auf eigenem Rechner

SOTA State Of The Art

LLM Arena Elo "Blindbewertungen"

<https://huggingface.co/spaces/lmsys/chatbot-arena-leaderboard>

TextView →				WebDev🕒 2 days ago			
Rank (UB) ↑	Model ↓	Score ↑↓	Votes ↑↓	Rank (UB) ↑	Model ↓	Score ↑↓	Votes ↑↓
1	🌐 gemini-2.5-pro	1457	31,991	1	🌀 GPT-5 (high)	1481	4,012
1	🌀 gpt-5-high	1455	9,162	1	🤖 Claude Opus 4.1 thinking-1...	1474	1,604
1	🤖 claude-opus-4-1-20250805...	1451	6,440	3	🤖 Claude Opus 4.1 (20250805)	1436	2,011
2	🌀 o3-2025-04-16	1445	37,814	4	🌐 Gemini-2.5-Pro	1405	7,348
3	🌀 chatgpt-4o-latest-20250326	1442	34,489	4	🦋 DeepSeek-R1-0528	1392	4,800
3	🤖 claude-opus-4-1-20250805	1439	9,097	5	🤖 Claude Opus 4 (20250514)	1383	9,233
3	🌀 gpt-4.5-preview-2025-02-27	1439	15,271	6	🔢 GLM-4.5	1368	1,532
8	📌 grok-4-0709	1425	16,211	6	🔢 GLM-4.5-Air	1364	1,411
8	🌀 gpt-5-chat	1423	6,383	7	🦋 Qwen3-Coder	1365	6,687

Medium to Large LLMs auf eigenem Rechner

SOTA State Of The Art

Exam Subject	Best Model(s)	Score (%)	Gap to Next (%)
LSAT	Claude 3.5	80.0	2.3
SAT Reading	Claude 3.5	82.4	1.6
SAT Math	Claude 3.5	95.8	1.1
GMAT Quant.	Llama 405B	96.0	4.0
GMAT Verbal	GPT-4o	95.5	3.0
GRE Physics	Claude 3.5	90.7	1.4
AP Art History	Llama 405B	86.7	2.3
AP Biology	Multiple	100.0	0.0
AP Calculus	GPT-4o	91.4	2.8
AP Chemistry	Llama 70B	96.9	0.0
AP English Lang.	GPT-4o	98.1	3.8
AP English Lit.	GPT-4o	88.9	0.0
AP Env. Sci.	Llama 405B	93.5	4.4
AP Macro Eco.	Llama 70B/405B	98.3	1.8
AP Micro Eco.	GPT-4o/Claude 3.5	97.9	4.1
AP Physics	Llama 405B	92.9	14.3
AP Psychology	Multiple	100.0	0.0
AP Statistics	Claude 3.5	96.3	3.7
AP US Gov.	GPT-4o/Claude 3.5	100.0	2.4
AP US History	Llama 70B/405B	97.6	2.5
AP World History	Multiple	100.0	0.0
AP Average	Llama 405B	93.5	0.5

Objektive Tests: SAT Scholastic Assessment Test / Advanced Placement (AP) / Abitur

SOTA State Of The Art

Objektive Tests: MMLU (Massive Multitask Language Understanding) benchmark

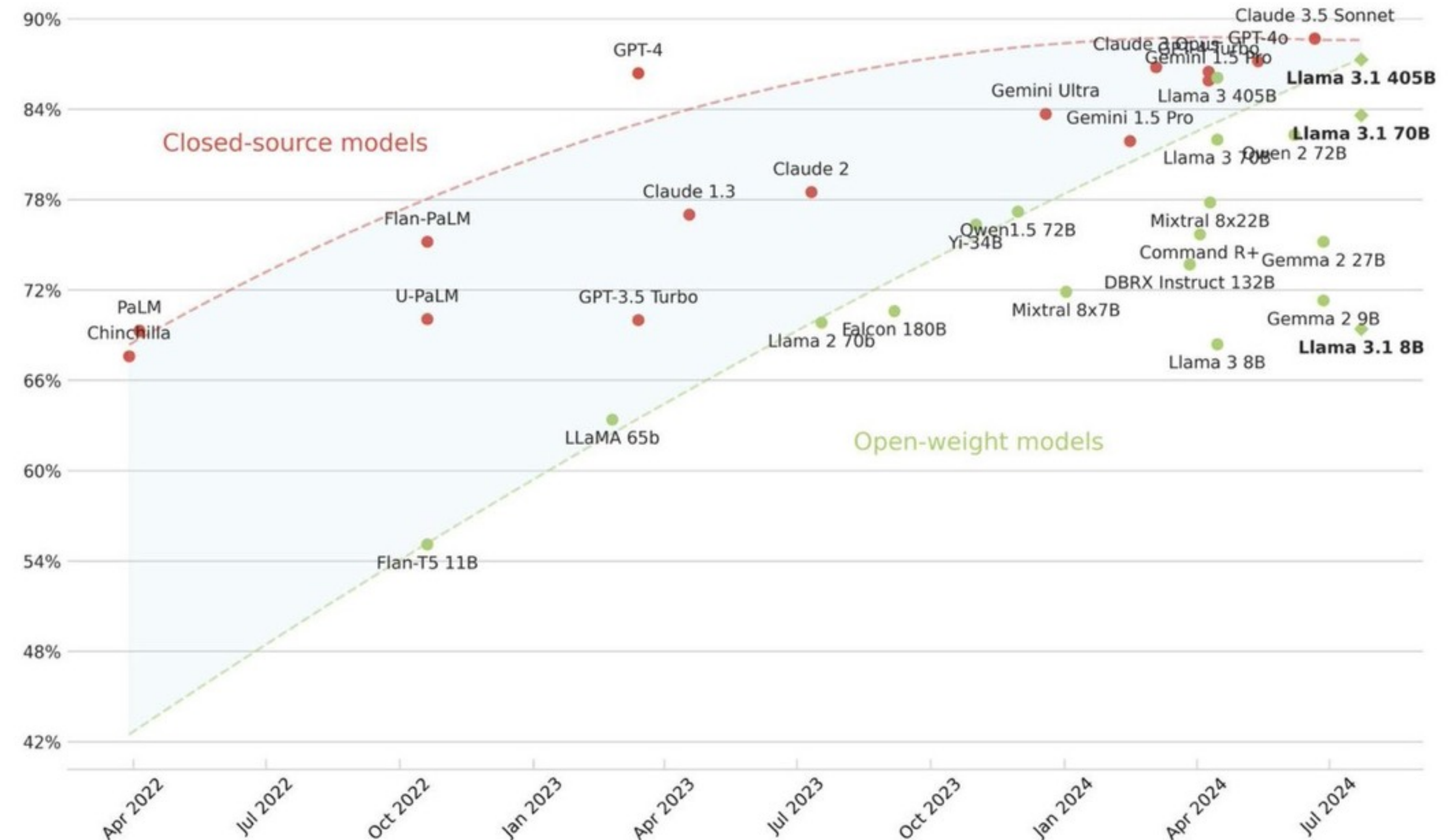
LLM Large Language Models, Small LLMs auf eigenem Rechner

```
`ollama run llama3.1`
```

Closed-source vs. open-weight models

Llama 3.1 405B closes the gap with closed-source models for the first time in history.

MMLU (5-shot)



SOTA State Of The Art

🥈 International Mathematical Olympiad 🥈 Silber!

<https://deepmind.google/discover/blog/ai-solves-imo-problems-at-silver-medal-level/>

IQ Test?

Verbal IQ of the ChatGPT was 155

<https://www.scientificamerican.com/article/i-gave-chatgpt-an-iq-test-heres-what-i-discovered/>

"Was schwer für Menschen ist, ist einfach für AI und umgekehrt."

Neue IQ Tests werden gesucht! Moving target.

EQ Tests? "ChatGPT Outshines Humans in Emotional Tests"

<https://neurosciencenews.com/chatgpt-emotion-awareness-23231/>

3D taktiler, physikalischer Verständnis

SOTA State Of The Art

Bild Verständnis

User

What is funny about this image? Describe it panel by panel.



SOTA State Of The Art

'The Internet is just a fad'

1999 Internet Blase aber man kann nicht investieren

SOTA State Of The Art

Foundation Models

Grundlagenmodell?

Ein Model kann verschiedenste Aufgaben lösen

Eierlegende Wollmilchsau.

Ja, aber noch nicht ganz für agentische (RL) aufgaben.

SOTA State Of The Art

Autonome Fabriken

Autonomes Fahren

Tesla / Uber / Waymo

FSD? Full self driving.

Noch nicht flächendeckend.

Noch nicht in Deutschland.

Statistisch schon sicherer (als Menschen)?

"The Future is here just not evenly distributed"

SOTA State Of The Art

Reasoning Models

HRM Hierarchic Reasoning Model

Wie viele Parameter sind essenziell für Logik?

Meta Hintergründe

NVIDIA CUDA GPU (graphical processing units) *blackbox* cpp parallel

Apple Intelligence "AI" *flop*

Skalierung

7 Billionen (trillion) dollar Sam Altman OpenAI

1% GDP

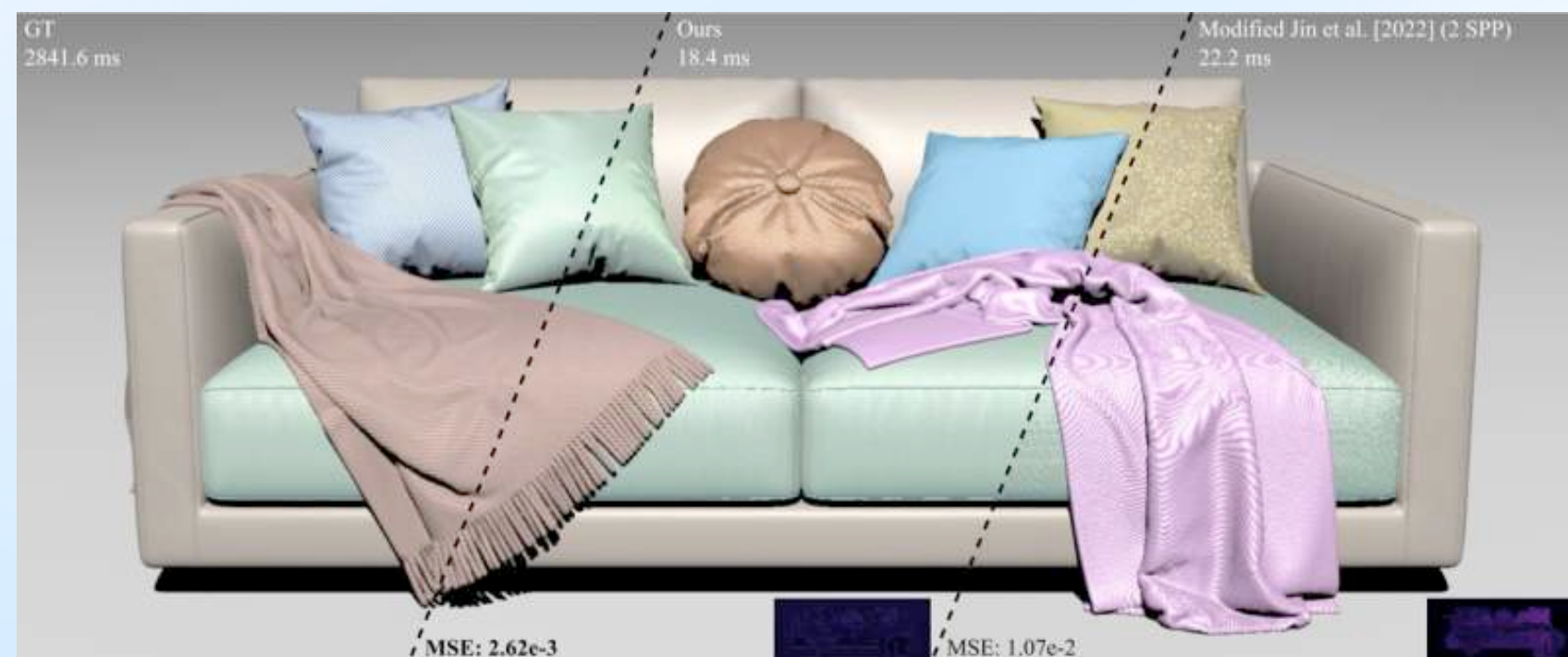
Skynet

Akademia vs Industrie

Eigene Ideen

Anhand von Standard Problemen Testen

MNist (leider schon 99.99% "gelöst")



<https://techxplore.com/news/2024-07-lightweight-neural-network-enables-realistic.html>

Level von Verständnis

X erkennen

X benennen

X passiv nutzen

X Teile erkennen

X Teile benennen

X aktiv nutzen

Teile verstehen

Teile austauschen

Teile tunen

X in größere Zusammenhänge einordnen

Modell simulieren

Modell bauen

Modell bauen, welches funktioniert

eigene Teile / Komponenten erfinden

eigene Teile / Komponenten selbst implementieren

X komplett neu erfinden

Level von Verständnis

Auto passiv nutzen (0 Jahre alt)

Auto erkennen (1/2 Jahr alt)

Auto benennen (2 Jahre alt)

Auto Teile erkennen (Reifen, ... als Kind. Keilriemen eventuell nie;)

Auto Teile benennen - ' ' -

Auto aktiv nutzen (17 Jahre)

Teile verstehen (relativ)

Teile austauschen (eventuell schon als Kind, eventuell nie;)

Teile tunen - ' ' -

Auto in größere Zusammenhänge einordnen (in Schule, Leben)

Modell simulieren (an Uni / Job?)

Modell bauen (Holz: Grundschule?)

Modell bauen, welches funktioniert (Genie!)

eigene Teile / Komponenten erfinden

eigene Teile / Komponenten selbst implementieren

Auto komplett neu erfinden ;)

No free lunch theorem

"Es gibt keine Architektur die bei beliebigen Problemen zu den besten Ergebnissen führt."

Sehr genau auf die Axiome schauen unter welchen das Theorem gültig ist.

In der Praxis gibt es Vorwissen und Ähnlichkeiten: Text, Bild, ...

Für solche Probleme kann man sagen dass unsere Deep Learning Architekturen universell besser sind als Vorgänger Architekturen. (Bei begrenztem compute)

Vorbereitungen

IDE Einrichten (**PyCharm** oder VsCode oder **CoLab**, Spyder, Notebooks jupyter ...)

Freiwillig: **Github account anlegen** (und Name pasten für Kooperation)

Stark Empfohlen: Versionierungssoftware: alte Stände gehen nicht verloren

Empfohlen: **ChatGPT** oder Anthropic **Claude** account

Vorgriff zu Mittwoch: gegebenenfalls schon mal

```
`pip install torch`  
`pip install tensorflow`
```

(oder per IDE)

IDEs

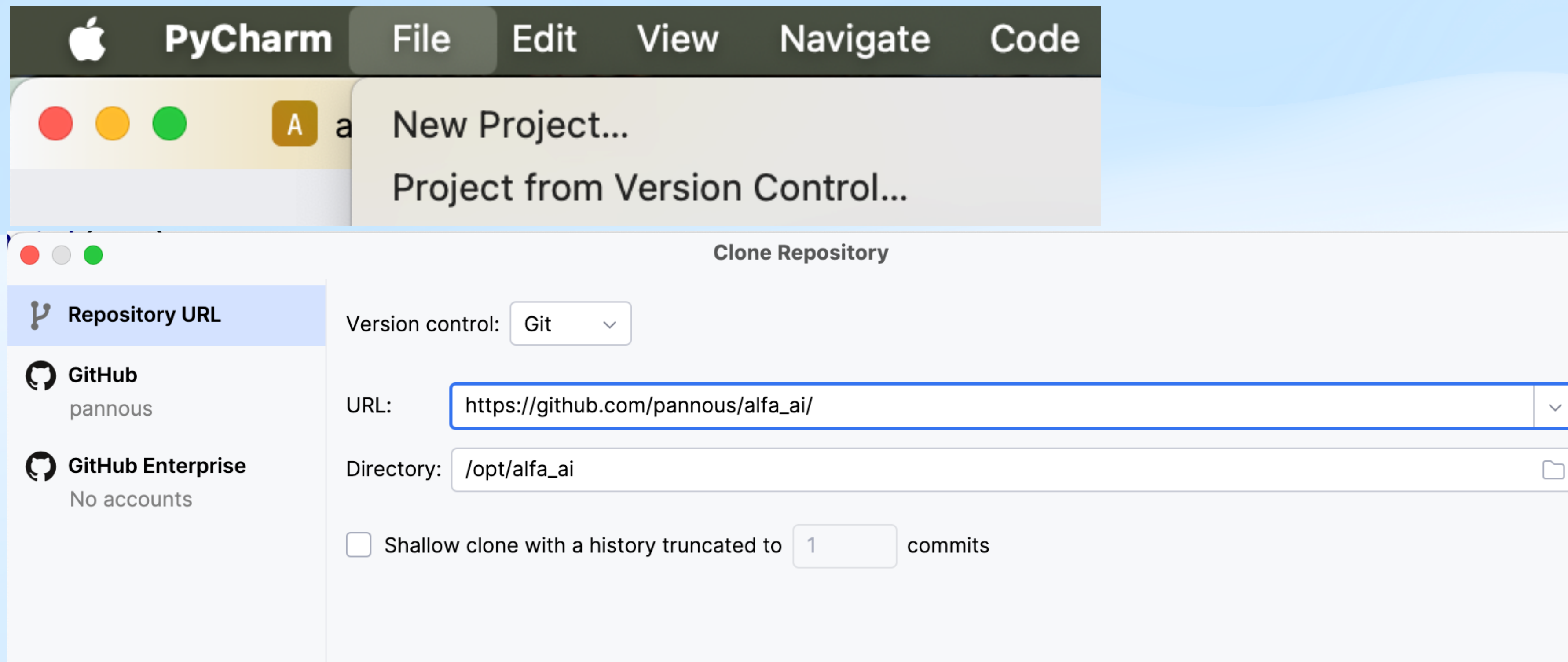
PyCharm: wichtigste Funktion: **Ctrl-Shift-A** "Alles Durchsuchen"

VisualStudio Code: wichtigste Funktion: **Ctrl-Shift-P** "Alles Durchsuchen"

Immer wenn ihr eine Datei, Funktion, ... oder **Befehl** sucht!

IDEs

Unser Github Projekt 'auschecken':
https://github.com/pannous/alfa_ai/

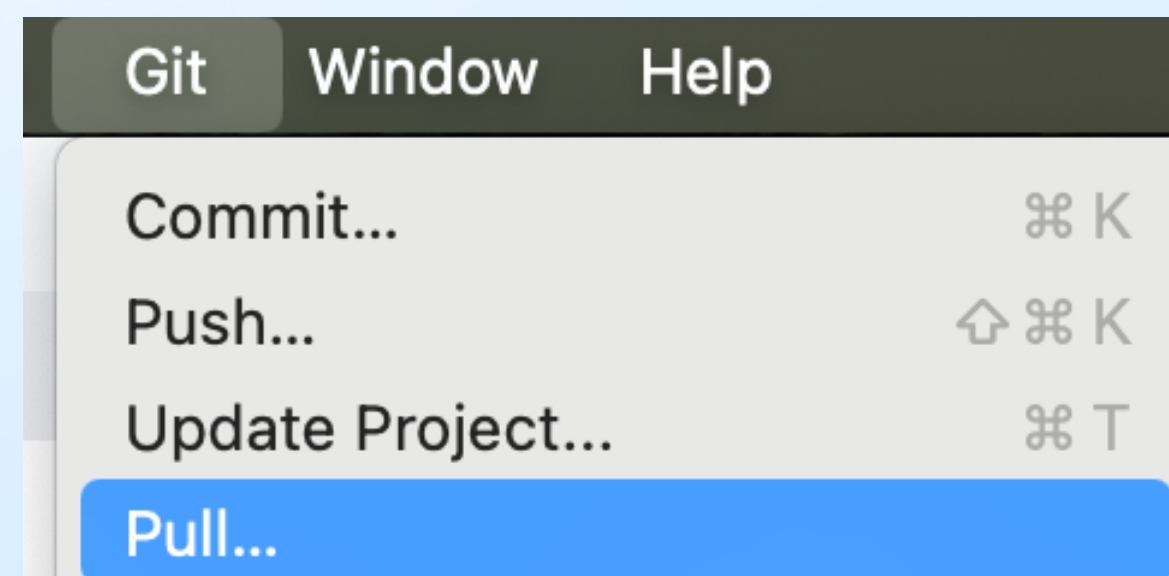


Github

Unser Github Projekt 'clonen' / 'auschecken':

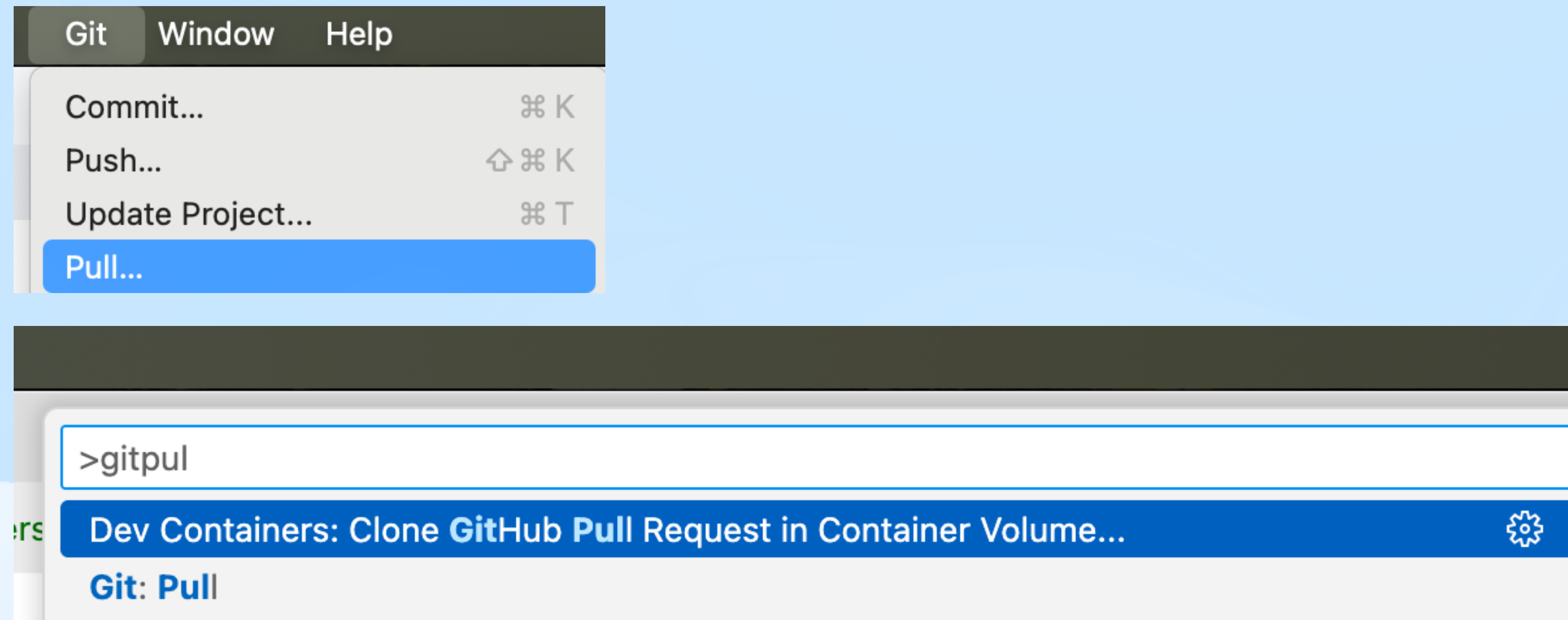
https://github.com/pannous/alfa_ai/

- 1) IDE VSCode ↗ ⌘P → “Git: Clone” ODER**
- 1) GitHub Desktop Client ODER für Profis:**
- 2) git.exe Kommando `git clone git@github.com:pannous/alfa\_ai.git` ODER**
- 3) gh.exe Kommando `gh repo clone pannous/alfa\_ai`**



Github

1) Ab und zu neueste Version holen mit "Git Pull"



Saubere Trainings Daten

anormale Ausreisser? Lücken?

skalieren ja/nein?

Texte dekodieren?

regression/klassifikation?

Plausibilität?

Bei uns: erst mal alles in Rohform mitlernen!

Später per Clusterung Ausreisser etc analysieren.

Finetuning zur Sonderbehandlung 'unerwünschter Artefakte'.

Saubere Trainings Daten

Splitten Auftrennen:

1) Trainings / Testdaten (Verhältnis / sampling!)

2) Trainings / Testdaten / Validierungsdaten

"p hacking" Hyperparameter auf Testdaten optimieren. Nur für Akademiker.

Bei uns ist Training auf Trainingsdaten oft 'genug' um einen guten Eindruck zu bekommen.

Supervised / Unsupervised learning

Supervised $\approx$ Labeled

- **Klassifikation**
- **Regressionsziele**

Unsupervised

- **Strukturen erkennen / Clustering**
- **Vortrainieren**
- **LLM Vorhersage von Sprache = Chat**
- **Bildinterpolation zwischen Bildern mischen/morphen**

Vorbereitungen

Colab:

Unser Sinus Kurven "Hello World" :

<https://colab.research.google.com/drive/1B-MRv6VD7zTnvTBFaxwIWhD8H2-9B9Sh>

Bei harten Problemen kurz auf GPU/TPU umstellen, danach aber wieder zurück auf CPU

Change runtime type

Runtime type

Python 3

Hardware accelerator [?](#)

☒ CPU

☐ T4 GPU

☐ A100 GPU

☐ L4 GPU

☐ TPU v2

Want access to premium GPUs? [Purchase additional compute units](#)

Cancel

Save

Buch

TEIL I Die Grundlagen des Machine Learning (Woche 1)

Ein Machine-Learning-Projekt von A bis Z

Bis Kapitel 4 (nicht: Support Vector Machines etc!!)

Teil II Neuronale Netze und Deep Learning

10 Einführung in künstliche neuronale Netze mit Keras Seite 333 Woche 1

14 Deep Computer Vision mit Convolutional Neural Networks Seite 519 Woche 2

17 Autoencoder, GANs und Diffusionsmodelle .. 679 Woche 2

15 Verarbeiten von Sequenzen mit RNNs Seite 577 Woche 3

16 Verarbeitung natürlicher Sprache mit RNNs und Attention Seite 619 Woche 3

18 Reinforcement Learning Seite 727 Woche 4

<https://github.com/ageron/handson-ml3>

<https://pytorch.org/tutorials>