Lehren und Lernen im KI-Zeitalter

KI zur Programmierunterstützung

Dr. Norman Meuschke & Prof. Dr. Bela Gipp

Georg-August Universität & SUB Göttingen
Lehrstuhl Scientific Information Analytics

www.gipplab.org

Diskussionsveranstaltung KI in der Lehre, Georg-August-Universität Göttingen, 21.9.2023

Tageblatt-Abstimmung

Auf welchem Sportplatz in Göttingen und der Region gibt es die beste Bratwurst?

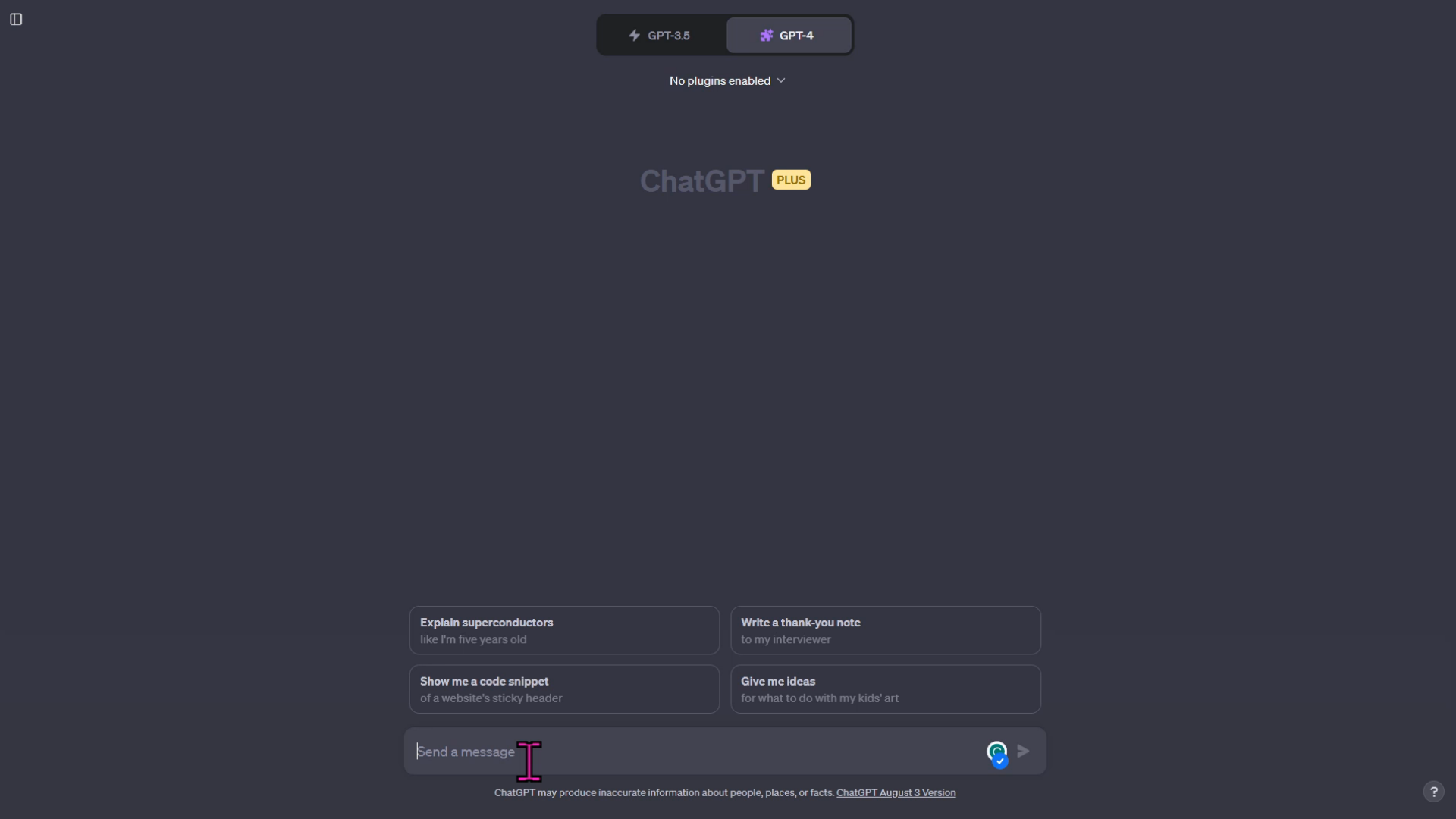
Gericht bestätigt: Es gab keinen Managementvertrag

Bushido gewinnt erneut Zivilprozess: Arafat Abou-Chaker schuldet ihm mehr als zwei Millionen

Gofundme

Querschnittsgelähmt nach E-Scooter-Unfall: Spendenaktion für Familienvater

TOD VON LOTHAR KOCH


GPT-3.5
GPT-4
No plugins enabled
ChatGPT
PLUS
Explain superconductors
like I'm five years old
Write a thank-you note
to my interviewer
Show me a code snippet
of a website's sticky header
Give me ideas
for what to do with my kids' art
Send a message
ChatGPT may produce inaccurate information about people, places, or facts. ChatGPT August 3 Version

Pseudocode

ME

I am writing an academic paper in Machine Learning for NeurIPS. I have a Pseudocode in the paper that is in PyTorch-style. Could you translate it into JAX-style Pseudocode?

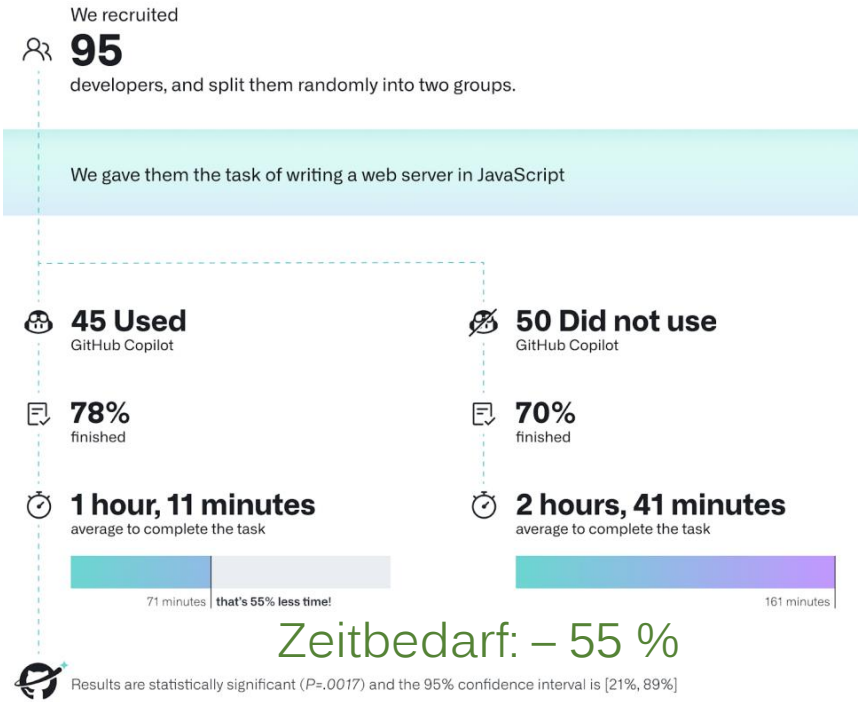
```
# PyTorch style pseudo-code
model = ... # neural network
loss = ... # original loss fn
optimizer = ... # optim. of model
for data, label in data_loader: # apply a neural network model y = model(data)
 # compute gradient of orig. loss
 grad = gradient(loss(y, label), y) # compute Hessian (or alt. Fisher) hess = hessian(loss(y, label), y)
 # compute the projected optimum
 z_star = y - grad @ inverse(hess + tik_l * eye(N))
 # detach / stop the gradient
 z_star = z_star.detach() # compute the Newton loss
 l = MSELoss()(y, z_star)
 # backpropagate and optim. step
 l.backward()
 optimizer.step()
```

python

Copy code

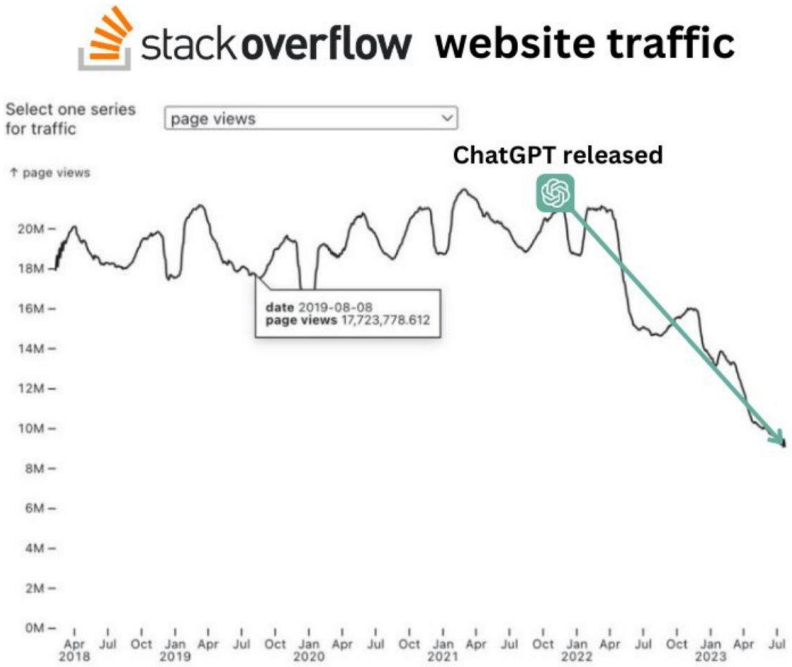
```python
import jax
import jax.numpy as jnp
from jax import grad, vmap, jit
from jax.scipy.linalg import solve

model = ...  # neural network (assumed to be a pure function)
loss = ...  # original loss function (also a pure function)
params = ...  # parameters of the model
opt_init, opt_update, get_params = ...  # optimizer from jax.experimental.op
state = opt_init(params)

@jit
def compute_newton_update(data, label, params):
    y = model(params, data)

    # Gradient and Hessian functions
    loss_grad = grad(loss, 1)
    loss_hessian = jax.jacfwd(jax.jacrev(loss, 1))

    grad_val = loss_grad(y, label)
    hessian_val = loss_hessian(y, label)
```

Produktivitätssteigerung bei Programmierung



Zeitbedarf: – 55 %

Quelle: [1]



Quelle: [2]

Folien & Videos
des Vortrags:

www.gipplab.org/ki-lehre-23

Dr. Norman Meuschke
meuschke@uni-goettingen.de
Twitter: @MeuschkeN
www.gipplab.org/meuschke

Prof. Dr. Bela Gipp
gipp@uni-goettingen.de
Twitter: @BelaGipp
www.gipplab.org/gipp

Quellen

[1] Kalliamvakou, E., "Research: quantifying GitHub Copilot's impact on developer productivity and happiness", *The GitHub Blog*, Sep. 2022. [Online]. Available: https://github.blog/2022-09-07-research-quantifying-github-copilots-impact-on-developer-productivity-and-happiness/.

[2] Sali Zumberi via LinkedIn: https://www.linkedin.com/posts/activity-7102006503108136960-li4C?utm_source=share&utm_medium=member_ios