

Lernpfad für LLM-Verständnis – Von Grund auf

1. NOTWENDIGES VORWISSEN (Reihenfolge wichtig)

A. Lineare Algebra & Kalkül (2–3 Wochen)

Warum: Tensoren, Matrizen, Gradienten sind überall

Ressourcen:

- **3Blue1Brown – Essence of Linear Algebra** (YouTube, kostenlos)
 - https://www.youtube.com/playlist?list=PLZHQObOWTQDPD3MizzM2xVFitgF8hE_ab
 - ~15 Videos, sehr intuitiv
- **3Blue1Brown – Essence of Calculus** (YouTube, kostenlos)
 - Chain rule, partial derivatives verstehen

Takeaways:

- Vektoren & Matrizen (Shape, Multiplikation)
 - Gradienten & partielle Ableitungen
 - Backpropagation (konzeptuell)
-

B. Wahrscheinlichkeit & Statistik (2 Wochen)

Warum: Cross-Entropy Loss, Softmax, Sampling verstehen

Ressourcen:

- **MIT OpenCourseWare: Probability** (kostenlos)
 - <https://ocw.mit.edu/courses/6-041sc-probabilistic-systems-analysis-and-applied-probability-fall-2013/>
- **StatQuest with Josh Starmer** (YouTube)
 - Bayes Theorem, Maximum Likelihood, etc.

Takeaways:

- Wahrscheinlichkeitsverteilungen
- Maximum Likelihood Estimation
- Cross-Entropy Loss
- Softmax (Umwandlung von Logits zu Wahrscheinlichkeiten)

2. DEEP LEARNING GRUNDLAGEN (4–6 Wochen)

Best Option: Stanford CS231N + MIT 6.S191

Stanford CS231N – Convolutional Neural Networks (kostenlos online)

- <https://cs231n.github.io/>
- Nicht nur CNNs – die ersten Vorlesungen sind fundamental für alle NNs

Wichtigste Vorlesungen:

1. **Lecture 1:** Computervision History (nur Kontext)
2. **Lecture 2:** Image Classification, Data-driven approach, kNN
3. **Lecture 3:** Loss Functions, Optimization (KRITISCH!)
4. **Lecture 4:** Backpropagation, Chain Rule
5. **Lecture 5:** Convolutional Networks (optional für LLMs, aber gutes Pattern)
6. **Lecture 9:** CNN Architectures (überschießbar)

Zeitaufwand: ~20 Stunden für Lectures + Assignments

MIT 6.S191 – Introduction to Deep Learning (kostenlos)

- <http://introtodeeplearning.com/>
- Sehr gut strukturiert, weniger “Image-fokussiert”

Wichtigste Inhalte:

- Neural Networks
- Convolutional Networks
- Recurrent Neural Networks (RNNs) – wichtig für Sequenzen!

Ergänzung: Andrew Ng’s Coursera (kostenpflichtig, aber ~\$30)

- **Neural Networks and Deep Learning** (1 von 5 Kursen)
- <https://www.coursera.org/learn/neural-networks-deep-learning>
- Sehr gut erklärt, mit Coding Assignments in Python

3. NATURAL LANGUAGE PROCESSING (NLP) (6–8 Wochen)

Kritisch für LLM-Verständnis!

Stanford CS224N – NLP with Deep Learning (kostenlos online)

- <https://web.stanford.edu/class/cs224n/>
- Das ist der Standard-Kurs für NLP

Wichtigste Vorlesungen:

1. **Lecture 1:** NLP Intro, Word Vectors
2. **Lecture 2:** Word Vectors (Word2Vec, GloVe) – Grundkonzept!
3. **Lecture 3:** Neural Networks & Word Vectors
4. **Lecture 4:** Backprop (Wiederholung mit NLP-Fokus)
5. **Lecture 5:** Dependency Parsing
6. **Lecture 6-7:** RNNs, GRUs, LSTMs (WICHTIG!)
7. **Lecture 8:** Machine Translation, Seq2Seq
8. **Lecture 9:** Attention is All You Need (KRITISCH!)
9. **Lecture 10-11:** Transformers (DAS ist das Fundament!)
10. **Lecture 13-15:** Pretraining, BERT, GPT, Fine-tuning

Zeitaufwand: ~50 Stunden für Lectures, ~30 Stunden für Assignments

4. TRANSFORMERS SPEZIALISIERT (2–3 Wochen)

Nach CS224N:

HuggingFace Course (kostenlos)

- <https://huggingface.co/course/chapter1/1>
- **Sehr praktisch**, mit Code zum Mitschreiben

Wichtigste Kapitel:

- Chapter 1: Transformer Models
- Chapter 2: Using Transformers
- Chapter 3: Fine-tuning a Pretrained Model
- Chapter 4: Sharing Models and Tokenizers

“Attention Is All You Need” Paper lesen

- <https://arxiv.org/abs/1706.03762>
- Nach Lectures viel verständlicher
- Nur noch ~2 Stunden zum Durcharbeiten

5. GROSSE LANGUAGE MODELS (Optional aber empfohlen) (2 Wochen)

Nach allem obigen:

Papers zum Lesen:

Fundament:

1. “Language Models are Unsupervised Multitask Learners” (GPT-2 Paper)
 - https://d4mucfpksywv.cloudfront.net/better-language-models/language_models_are_unsupervised_multitask_learners.pdf
2. “Language Models are Few-Shot Learners” (GPT-3 Paper)
 - <https://arxiv.org/abs/2005.14165>

Für LLM-Sicherheit/Alignment:

3. “Training language models to follow instructions with human feedback” (Anthropic InstructGPT/RLHF)
 - <https://arxiv.org/abs/2203.02155>

Für Transformers-Optimierung:

4. “FlashAttention: Fast and Memory-Efficient Exact Attention with IO-Awareness”
 - <https://arxiv.org/abs/2205.14135>

Vorträge statt Papers:

- Andrej Karpathy’s “Lecture on Large Language Models” (YouTube)
 - https://www.youtube.com/watch?v=zjkBMFhNj_g
 - 1 Stunde, sehr gut verständlich nach Grundlagen

KOMPLETTER LERNPLAN (12–16 Wochen / ~200 Stunden)

Woche 1-3: Lineare Algebra + Kalkül (3Blue1Brown)

Time: ~15 Stunden

Woche 4-5: Statistik & Wahrscheinlichkeit

Time: ~15 Stunden

Woche 6-10: Deep Learning Fundamentals

(CS231N Lectures 1-5 + MIT 6.S191)

Time: ~40 Stunden (Lectures + Code)

Woche 11-18: Stanford CS224N

(Lectures 1-15, Assignments)

Time: ~80 Stunden

Woche 19-20: Transformers Deep Dive + Papers

Time: ~20 Stunden

TOTAL: ~200 Stunden ≈ 5-6 Stunden/Woche über 6-7 Monate

SCHNELLER WEG (wenn es schneller sein muss)

Wenn du nur Zeit für 8 Wochen hast:

Woche 1: 3Blue1Brown Linear Algebra (skip Kalkül, lernen unterwegs)

Woche 2: Basic Statistics (nur die wichtigsten Konzepte)

Woche 3-5: MIT 6.S191 (schneller als Stanford CS231N)

Woche 6-8: Stanford CS224N – aber nur Lectures 1-11

(Skip: Dependency Parsing, fokus auf RNNs & Transformers)

+ Papers lesen

MINIMALER PATH (wenn sehr kurzfristig vor Vortrag)

Falls du in 2-3 Wochen presenting:

1. **3Blue1Brown Linear Algebra** – 3 Stunden
2. **MIT 6.S191 Lectures 1-5** – 5 Stunden (Neural Networks)
3. **Stanford CS224N Lecture 9-11** – 4 Stunden (Attention, Transformers)
4. **Andrej Karpathy's LLM Lecture** – 1 Stunde
5. **GPT-2 Paper überfliegen** – 1 Stunde

Total: ~14 Stunden

Das reicht, um die technischen Details meiner vorherigen Erklärung zu verstehen.

KONKRETE SCHRITTE JETZT

1. **Diese Woche:**
 - Schau die 3Blue1Brown Linear Algebra Serie (~15 Videos)
 - Parallel: MIT 6.S191 Lecture 1-2
 2. **Nächste Wochen:**
 - Beende MIT 6.S191 (oder CS231N Lectures 3-5)
 - Starte Stanford CS224N Lecture 1
 3. **Wenn gut vorangehen:**
 - CS224N Lectures 6-11 (das ist wo LLMs wirklich werden)
-

Mein Plan:

1. **Linear Algebra (3Blue1Brown)** – schnell durchziehen (1 Woche)
2. **MIT 6.S191** – kompakter, praktischer (2-3 Wochen)
3. **HuggingFace Course** – hands-on coding (1-2 Wochen)
4. **Stanford CS224N Lectures 6-11** – tieferes Verständnis (3-4 Wochen)

→ Kann in 6-8 Wochen gemacht werden mit ~10 Stunden/Woche

Danach verstehst du genug, um:

- Meine ursprüngliche Erklärung zu verstehen
 - Dein LLM-Projekt konkret zu starten
 - Den Vortrag intelligent zu halten
-