🤖 AI-ML Class Notes — January 17, 2025

100xSchool Bootcamp 1.0

Artificial Intelligence & Machine Learning Foundations

PYTHON | OPENAI | JUPYTER | TRANSFORMERS

📚 Class Overview

This directory contains resources and notes from the **AI & Machine Learning** class. Today's session focused on understanding the internal mechanics of Large Language Models (LLMs), tracing the history of AI from rule-based systems to the modern Transformer architecture, and exploring the frontiers of prompt engineering and model security.

📖 Table of Contents

▶ **Click to Expand Navigation**

#	Topic	Summary
1️⃣7️⃣	Future (2024-2025)	Multimodality, Reasoning, Agents

🎯 Topics Covered

▼

1️⃣ Visualizing Large Language Models

Understanding LLMs requires peeking inside the "black box". We explored an interactive 3D visualization to see how tokens are processed.

🧠 Key Concepts

Component	Function
Embeddings	Convert words into numerical vectors that capture semantic meaning.
Self-Attention	Weighs the importance of different words in a sentence relative to each other.
Layer Normalization	Keeps data values balanced as they flow through the network.
Feed-Forward Networks (MLP)	Processes information to extract higher-level features.
Softmax	Converts final scores into probabilities for the next token.

> [!TIP] Use the LLM Visualization Tool to step through the inference process token by token. It's the best way to build a mental model of how GPT works.

▼

2️⃣ AI Security & Jailbreaking

We discussed the boundaries of AI safety and how "jailbreaking" helps researchers and developers understand model limitations and vulnerabilities.

🌓 L1B3RT4S (Libertas)

- **Concept**: A project exploring "liberation prompts" to bypass standard model constraints.
- **Goal**: To empower users and test the robustness of AI alignment.
- **Methods**: Using cryptic, high-complexity prompts to confuse or override safety filters (e.g., "GODMODE", "JAILBREAK").

> [!CAUTION] Jailbreaking is for **educational and research purposes only**. Always use AI responsibly and ethically.

▼

3️⃣ Fundamentals: Learning, Knowledge, & Intelligence

We explored the philosophical and technical definitions that underpin Artificial Intelligence.

🌱 What does it mean to "Learn"?

At its core, **learning is the process of changing yourself based on experiences.** It's about making adjustments to improve future performance.

The Learning Loop:

```
graph LR
    A[🧪 Try Something] --> B(📊 Get Feedback)
    B --> C{🔧 Adjust}
    C --> A
```

> *Try ➡️ Feedback ➡️ Adjust ➡️ Repeat*

🧠 The Nature of Knowledge

Type	Definition	Example
Explicit Knowledge	Rules, facts, and logic that can be clearly articulated.	"Water boils at 100°C" or "If X, then Y."
Implicit Knowledge	Intuitive understanding based on patterns, hard to explain.	Recognizing a friend's face or riding a bike.

> [!NOTE] AI has historically struggled with *implicit* knowledge, but Deep Learning has bridged this gap by learning patterns from data rather than hard-coded rules.

💡 Defining Intelligence

Intelligence is the ability to achieve goals in a wide range of situations.

Type	Description
Narrow AI	Extremely efficient at a single task (e.g., Calculator) but fails outside its domain.
General Intelligence	Adaptable, capable of applying skills to new and unforeseen problems (e.g., Humans).

▼

4️⃣ The Evolution: From Rules to Learning

We traced the history of how we moved from rigid instructions to adaptive systems.

🤖 What is AI?

AI is the Goal. It is the science of making machines do things that would require intelligence if done by a human.

📃 Attempt #1: Expert Systems (Rule-Based AI)

The first approach was to **write the rules manually**. Humans explicitly programmed every logical step.

Example: A simple Spam Filter

```
def classify_email(email):
    if "free money" in email:
        return "SPAM"
    elif "click here" in email:
        return "SPAM"
    else:
        return "INBOX"
```

The Limitation: This works for clear-cut logic but fails with nuance.

🚧 The Problem with Rules

How do you write rules for intuitive tasks?

- **Recognizing a Face**: You can't describe a face with just `if-else` statements.
- **Understanding Sarcasm**: "Great weather!" could mean sunny or terrible depending on context.
- **Decoding Idioms**: "It's raining cats and dogs" has nothing to do with animals.

These are examples of Implicit Knowledge that rule-based systems struggle with.

🧠 Attempt #2: Machine Learning

Instead of writing rules, **we let the machine learn them.**

The Core Idea: Show the machine thousands of examples and let it figure out the patterns itself.

> **Analogy**: It's like a child learning by trial and error.
>
> 1. **Show examples** (Input)
> 2. **Machine makes a guess** (Prediction)
> 3. **Tell it if it's right or wrong** (Feedback)
> 4. **Machine adjusts slightly** (Learning)
> 5. **Repeat millions of times**

▼

5️⃣ History: Early Successes, Winters, & The Boom

We looked at the timeline of AI development, from early promises to the modern revolution.

📈 Early Machine Learning: Modest Success

Early algorithms found success in specific domains:

- **Spam Filters**: Became much more effective.
- **Recommendation Systems**: Netflix and Amazon started predicting what we wanted.
- **Basic Image Recognition**: Could identify simple objects.

The Bottleneck: Despite these wins, AI couldn't hold a conversation or understand a complex paragraph.

📉 The Limits & AI Winters

Why was early AI limited? **Not enough Data** and **Not enough Compute**.

❄️ AI Winters: When Hope Died Twice

Era	What Happened
1970s	Early promises failed to materialize, and funding dried up.
1980s-90s	Expert systems were too brittle for the real world. They couldn't adapt, leading to another crash.

🚀 The Explosion (Post-2012)

Three key factors converged to create the modern AI boom:

1. **Massive Data**: The Internet provided a repository of human knowledge.
2. **GPUs**: Originally for gaming, they turned out to be perfect for the parallel math ML needs.
3. **Deep Learning**: Researchers finally cracked the math to train "deep" multi-layered models.

▼

6 The Deep Learning Revolution

The modern AI boom didn't happen by accident. It was ignited by a specific moment in history.

⌛ The AlexNet Moment (2012)

In 2012, a team led by **Geoffrey Hinton** (including Alex Krizhevsky and Ilya Sutskever) entered the **ImageNet Challenge**—a contest to identify objects in millions of images.

Metric	Before AlexNet	AlexNet
Error Rate	~26%	**~15%**

- **The Secret**: They proved that **Deep Neural Networks** trained on **GPUs** with **Massive Data** could outperform any human-crafted rules.

> [!IMPORTANT] This moment marks the birth of the modern Deep Learning era.

▼

7️⃣ Language: The Final Frontier

While computers got good at images, language remained broken for a long time.

🧩 Why is Language Hard?

Challenge	Example
Ambiguity	"I saw the man with the telescope." (Did I have the telescope, or did he?)
Context Dependence	A single word changes meaning based on its neighbors.
Messiness	Humans use slang, idioms, and sarcasm. Computers demand precision.

📖 Attempt #1: The Dictionary Approach (Symbolic AI)

The early idea was to just look up words in a database.

- **Problem**: Polysemy (Multiple meanings).
- *Example*: "Apple" could be a 🍎 (fruit), a 🏢 (company), or a 🎵 (record label). Without context, a dictionary is useless.

📊 Attempt #2: Statistical Patterns (N-grams)

The next step was counting phrases.

- **Logic**: If "New" is followed by "York", predict "City".
- **Problem**: **Pattern Matching ≠ Understanding**. The machine had no concept of what a "City" actually was.

▼

8️⃣ The Breakthrough: Vectors & Embeddings

How do we make a computer *understand* meaning? We turn words into numbers.

🔢 Word2Vec (2013)

This paper changed everything by introducing the concept of **Word Embeddings**.

The Big Idea: Instead of representing "Apple" as a simple ID, we represent it as a **list of numbers (a Vector)**.

Dimensions of Meaning:

Word	Royalty	Gender (M)	Edibility
King	0.98	0.95	0.01
Queen	0.97	0.05	0.02

Word	Royalty	Gender (M)	Edibility
Apple	0.02	0.00	0.94

Notice how "King" and "Queen" have similar scores for Royalty, but opposite scores for Gender.

🗺 Words as Positions in Space

If a word is a list of numbers, it's essentially a **coordinate on a map**.

- **Proximity = Meaning**: Words with similar meanings cluster together.
- **Distance**: "King" is close to "Queen", but far from "Apple".

🧮 The Magic of Embeddings (Word Math)

Since words are numbers, we can perform arithmetic:

```
King - Man + Woman = Queen
Paris - France + Italy = Rome
```

The computer "understands" relationships without explicitly being told!

▼

9️⃣ The Next Challenge: Context & Sequence

Embeddings were a huge leap, but they had a fatal flaw.

🛑 The Problem: One Word = One Position

In basic Word2Vec, each word has **exactly one** vector.

- **Sentence 1**: "I ate an **Apple**." (Fruit)
- **Sentence 2**: "**Apple** released a new iPhone." (Company)

To the model, these are the exact same "Apple". It can't distinguish meaning based on context.

🔄 Sequence Models (RNNs)

To fix this, we need to read the sentence like a human: **from left to right**.

Recurrent Neural Networks (RNNs):

- **Idea**: Process the sentence one word at a time.
- **Memory**: The model maintains a "hidden state" (memory) of what it has read so far.
- **Result**: The meaning of "Apple" changes depending on the words that came before it.

🙈 The Problem: Forgetting (Long-Range Dependencies)

RNNs have a short attention span. They process information linearly and have limited memory capacity.

Example Failure:

> "**The cat**, which was sitting on the mat that I bought from the store near the old church on the corner, **was** happy."

By the time the model reaches "was", it has often forgotten that the subject was "The cat" at the very beginning.

🛠️ The Fix: LSTMs & GRUs

To solve this, researchers invented **Long Short-Term Memory (LSTM)** and **Gated Recurrent Units (GRU)** networks.

- **Concept**: Specialized "gates" that control what to remember and what to forget over long distances.
- **Result**: Better at context, but still **slow** because they processed one word at a time.

▼

🔟 The Transformer Revolution (2017)

By 2017, the stage was set: we had Word Embeddings, powerful GPUs, and tons of internet data. But we needed a better architecture.

⚡ The Big Idea: Parallelism

Old Way (RNNs)	New Way (Transformers)
Sequential. Read a book word-by-word.	**Simultaneous**. Look at *every word* at the same time.
Slow, hard to learn long sequences.	Fast, leverages GPU parallelism.

🎯 The Secret Sauce: Self-Attention

The Transformer uses a mechanism called **Self-Attention**.

- **Concept**: The model "attends" to the most relevant words, no matter how far apart they are.

Example 1: Attention in Action

> "The **animal** didn't cross the street because **it** was too tired." *When the model processes "it", the attention mechanism screams **ANIMAL**.*

Example 2: Context Sensitivity

> "The animal didn't cross the **street** because **it** was too wide." *Change "tired" to "wide", and now "it" refers to **STREET**.*

💪 Why Attention is Powerful

Feature	Benefit

Feature	Benefit
No Forgetting	Every word "sees" every other word simultaneously. Distance is irrelevant.
Speed	Parallel processing means training is much faster than RNNs.
Deep Understanding	Builds a map of how every word relates to every other word.

🤖 The Transformer Architecture

> "An architecture built entirely on the mechanism of **attention**." — *Attention Is All You Need (2017)*

- ❌ **No Recurrence (RNNs)**
- ❌ **No Convolution (CNNs)**
- ✅ **Just Attention.**

This is the foundation of all modern Large Language Models (LLMs).

📉 The Evolutionary Path

```
graph LR
    A[RNN] --> B[LSTM]
    B --> C[GRU]
    C --> D[🚀 Transformer]
    style D fill:#f9f,stroke:#333,stroke-width:2px
```

Model	Description	Limitation
RNN	Reading word-by-word.	Forgets quickly.
LSTM	Explicit memory cells.	Still too slow (sequential).
GRU	Efficient LSTM.	Still sequential.
Transformer	Everything at once + Attention.	**State of the Art** ✅

▼

1️⃣1️⃣ How ChatGPT Works (Simplified)

We can break down modern AI into a simple equation:

$$\text{Transformer} + \text{Internet Data} + \text{Prediction} = \text{ChatGPT}$$

Component	Role	Description
🏗️ **The Engine**	Architecture	A massive neural network built entirely on the **Attention** mechanism.
📚 **The Knowledge**	Training Data	Trillions of words from books, articles, and the entire public internet.

Component	Role	Description
🎯 **The Task**	Objective	A simple goal: Give a sequence, **predict the most likely next word**.

▼

1️⃣2️⃣ The Power of Next Word Prediction

Why does predicting the next word lead to intelligence? To predict correctly, you need to understand *everything*.

Skill	Example
Grammar & Syntax	"The cat sat on the..." → Requires a **noun**.
Factual Knowledge	"The capital of France is..." → **Paris**.
Logic & Reasoning	"If John is older than Mary, and Mary is older than Bob, then John is..." → **Older than Bob**.

> [!IMPORTANT] The model isn't just memorizing; it's **internalizing the structure of reality** to make better guesses.

▼

1️⃣3️⃣ How It Generates Text

Generation is an **Iterative Loop**.

Step	Input	Prediction
1	"The"	→ **quick** (85%)
2	"The quick"	→ **brown** (92%)
3	"The quick brown"	→ **fox** (88%)
4	"The quick brown fox"	→ ...

> [!NOTE] The model predicts an array of next words with their probabilities. It usually picks the highest one, but sometimes adds randomness (**Temperature**) to be creative.

▼

1️⃣4️⃣ SLMs: Small Language Models

Not all AI needs to be massive.

Aspect	Details

Aspect	Details
Definition	Models with fewer than **~5 Billion** parameters.
Benefit	Lightweight enough to **run on your phone** or laptop.
Use Cases	Privacy-focused apps, edge devices, specific fast tasks.
Examples	Phi-3, Gemma, Llama 3.2 (smaller variants)

▼

1️⃣5️⃣ The Unexpected Discovery: Bigger = Smarter

We discovered a "Scaling Law": simply making the models bigger made them smarter.

Factor	Scaling
More Tokens	Millions → **Trillions** of words of training data.
More Size	Millions → **Hundreds of Billions** of parameters.
More Power	Days → **Months** of training on thousands of GPUs.

> [!IMPORTANT] **The Result: Emergent Capabilities** Reasoning, coding, and logic appear *spontaneously* as models get larger. These abilities were not explicitly programmed.

▼

1️⃣6️⃣ Foundation Models

This shift created a new paradigm in AI.

Old Paradigm	New Paradigm
Task-Specific AI	**General-Purpose AI (Foundation Models)**
Independent models for Translation, Summarization, Sentiment, etc.	**One massive model** trained on *everything*, capable of performing *any* language task through prompting.

> The "Foundation" is the base knowledge. We no longer build tools from scratch; we build on top of these giants.

▼

1️⃣7️⃣ Where We Are (2024-2025)

We are now moving beyond just text. The current frontier is defined by three key trends:

🎯 Trend 1: Multimodality

AI is no longer just text. It can **see images, hear voices, and speak back** in real-time.

- *Examples*: GPT-4V, Gemini, Claude Vision.

🧠 Trend 2: Reasoning

New models are designed to **"think" before they speak**, solving complex math and logic problems.

- *Examples*: OpenAI o1/o3, DeepSeek R1.

🕵️ Trend 3: Agents

The shift from Chatbots to Agents.

Chatbot	Agent
Talks to you.	Can **use tools**, browse the web, and complete multi-step tasks on your behalf.

- *Examples*: Claude Computer Use, Manus.

🔑 Key Takeaways

#	Concept	One-Liner
1	**Learning**	Try → Feedback → Adjust → Repeat.
2	**The Shift**	From *writing rules* to *letting machines learn*.
3	**The Boom**	Data + GPUs + Deep Learning = Modern AI.
4	**Embeddings**	Words as numbers. Meaning as position in space.
5	**The Transformer**	Parallelism + Self-Attention = All you need.
6	**LLMs**	Next-word prediction at a massive scale.
7	**Scaling Laws**	Bigger models → Emergent intelligence.
8	**The Future**	Multimodal models that can *see*, *think*, and *act*.

🗂 Folder Structure

```
17-01-2025/
├── 📂 Codes/                # Python scripts and notebooks
├── 📂 Notes_&_Screenshots/  # Class slides and diagrams
└── 📄 README.md             # You are here!
```

🔗 Resources

📝 Class Materials

- **Class Notes (Google Drive)**

🛠️ Tools & Visualizations

- **LLM Visualization (bbycroft)**
- **TensorFlow Playground**
- **Transformer Explainer**

🔓 Research & Repos

- **L1B3RT4S (GitHub)**
- **Hugging Face**
- **Attention Is All You Need (Paper)**

📖 Documentation

- **OpenAI API Docs**
- **LangChain**

📅 Class Date: **January 17, 2025**

Made with 💜 during 100xSchool Bootcamp 1.0

Next Up: Deep Dive into Neural Networks 🧠