Language Models and Transformers

Deep Learning course - SKEMA 2025

Mastère Spécialisé® Chef de Projet Intelligence Artificielle

Salem Lahlou

Recap: Key Machine Learning Concepts (1/2)

The foundations of our journey:

- **Generalization**: The ability of a model to work on new, unseen data (our ultimate goal!)
- **Overfitting**: When a model performs well on training data but poorly on new data (our enemy)
- **Loss Functions**: How we measure model performance (MSE, BCE, Cross-Entropy)
- **Data Splitting**: Dividing data into training, validation, and test sets
- **Parametrized Models**: Systems with adjustable "knobs" we can tune for better performance
- **Gradient**: The direction we need to adjust our model parameters to improve
- **Gradient Descent**: The process of repeatedly adjusting parameters to minimize errors

Recap: Key Machine Learning Concepts (2/2)

Building on the foundations:

- **Neural Networks**: Flexible models that can handle complex data without manual feature engineering
- **Universal Approximation**: Neural networks can theoretically represent almost any function
- **Activation Functions**: Sigmoid and softmax convert outputs into probabilities
- **Forward Propagation**: How data flows through the network to make predictions
- **Backpropagation**: How we calculate gradients to update the network
- **Mini-batch Gradient Descent**: A practical compromise between accuracy and speed
- **Hyperparameters**: Settings like learning rate that we tune on validation data
- **CNNs**: Special networks designed for images using convolutions
- **Preventing Overfitting**: Techniques like early stopping and regularization

The Challenge of Understanding Language

Why is language hard for computers?

- Words have **multiple meanings** depending on context
  - "The bank is closed" (financial institution or riverbank?)
- Words relate to each other in **complex ways**
  - "The trophy wouldn't fit in the suitcase because it was too big"
  - What was too big? The trophy or the suitcase?
- Understanding requires **world knowledge**
  - "She put the ice cream in the fridge because it was melting"
- Language has **long-range dependencies**
  - "The dog, which had been barking all morning, finally fell asleep"

Language Models: The Big Idea

At their core, language models do one simple thing:

They predict the next word in a sequence.

> "The chef cooked a delicious ______"

What word comes next? Probably: meal, dinner, steak, dish...

> "I need to charge my ______"

Likely completions: phone, laptop, battery, device...

This simple task forces the model to understand language!

Why Next-Word Prediction is Powerful

By learning to predict the next word, models learn:

- **Grammar**: Sentences must be structured correctly
- **Meaning**: Words must make sense in context
- **Facts**: Common knowledge appears in training data
- **Reasoning**: Logical connections between concepts

And the best part:

We can train models on virtually unlimited text from the internet without needing humans to label anything! The text itself provides the labels.

From Prediction to Generation

Once a model can predict the next word, it can generate text:

1. Start with a prompt: "Once upon a time"
2. Predict the next word: "there"
3. Add it to the sequence: "Once upon a time there"
4. Predict the next word again: "was"
5. Repeat until we have a complete text!

This is how ChatGPT and other AI assistants work at their core:

- Predict next word → add to text → predict next word → and so on

Traditional Language Models: N-grams

Early approach: Look at the last few words to predict the next one

> "The dog chased the ___"

With N-gram models:

- Look at frequently occurring patterns in text
- Count how often "cat" follows "The dog chased the"
- Count how often "ball" follows "The dog chased the"
- Pick the most likely word

Limitations:

- Can only use a small window of previous words
- Can't understand longer contexts
- Limited by what exact phrases it has seen before

The Sequential Nature of Language

Why are standard neural networks not enough?

- **Varying length**: Sentences can be any length
- **Order matters**: "Dog bites man" ≠ "Man bites dog"
- **Context changes meaning**: The same word means different things in different contexts

We need models that can:

1. Handle sequences of any length
2. Remember information from earlier in the sequence
3. Understand how words relate to each other

Representing Words as Vectors

The challenge: How do we represent words for computers?

- Computers work with numbers, not text
- Each word needs a numerical representation

Traditional approach: One-hot encoding

- Each word gets a unique position in a very long list
- "Cat" might be [0, 0, 1, 0, 0, 0, ...]
- "Dog" might be [0, 0, 0, 0, 1, 0, ...]
- **Problem**: All words appear equally different to the computer!

With one-hot encoding, "cat" and "kitten" seem just as different as "cat" and "skyscraper"

Word Embeddings: Words in a Meaningful Space

Word embeddings place words in a "meaning space":

- Each word becomes a list of 100-300 numbers
- Similar words have similar vectors
- "Cat" might be [0.2, -0.5, 0.1, ...]
- "Kitten" might be [0.25, -0.45, 0.15, ...]

What makes this powerful:

- Words with similar meanings cluster together
- Relationships between words become mathematical operations
- For example:
  - King - Man + Woman ≈ Queen
  - Paris - France + Italy ≈ Rome
- The model learns these relationships from reading text!

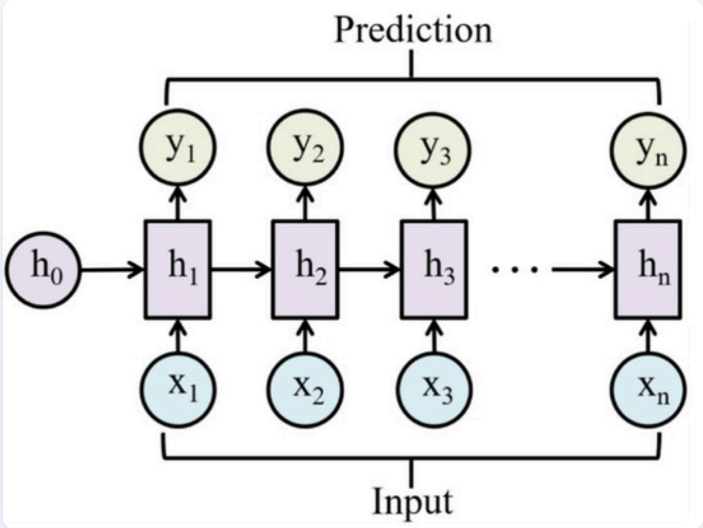
Recurrent Neural Networks (RNNs)

The first neural networks designed for sequences:

> Think of an RNN as a person reading a book one word at a time, while taking notes about what they've read so far.

- Process one word at a time
- Maintain a "memory" of what came before
- Use that memory to help predict the next word
- Use the same "reading process" for each word (shared parameters)

This gave us the first neural language models!



The Problem with Basic RNNs

RNNs struggle with long texts:

Imagine trying to remember every detail from a book you started reading last month!

The "vanishing gradient" problem:

- Information from early in the text gradually fades away
- The model effectively "forgets" what happened many words ago

Real example:
"The cats, which were sitting on the mats that had been placed there yesterday by the owner who lives next door, **purr**."

RNNs might predict "purrs" (agreeing with "owner") instead of "purr" (agreeing with "cats")

The Attention Mechanism: A Breakthrough

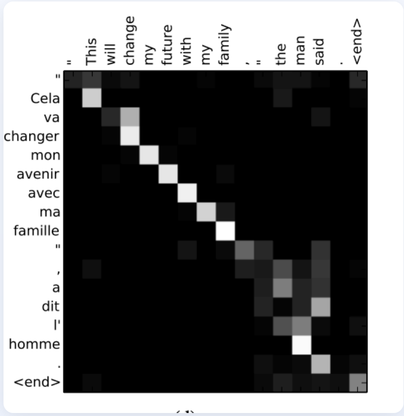
Key insight: Not all previous words are equally important!

> Attention is like being able to look back at the whole text and focus on just the relevant parts.

When predicting a word, attention allows the model to:

- Look at all previous words
- Assign each one an "importance score"
- Focus most on the relevant words
- Largely ignore irrelevant words

Example: When translating "The cat is black" to French as "Le chat est noir", attention connects "Le" with "The", "chat" with "cat", etc.



Self-Attention: Connecting Words to Each Other

A more powerful form of attention:

> Self-attention lets every word in a sentence directly "look at" every other word.

This helps with:

- **Pronouns**: Connecting "she" to the person it refers to
- **Relationships**: Understanding how words relate to each other
- **Long-range dependencies**: Connecting related parts even if they're far apart

Example:
"The animal didn't cross the street because it was too wide."
Self-attention directly connects "it" with "street" (not "animal")

The Transformer Architecture

Combines several innovations into one powerful model:

1. **Self-attention**: Connect any word directly to any other word
2. **Multiple attention "heads"**: Different types of relationships
3. **No recurrence**: Process all words in parallel for speed
4. **Position encoding**: Keep track of word order
5. **Feed-forward layers**: Additional processing power

Result: Much better language understanding and generation

> Transformers are the technology behind GPT, BERT, ChatGPT, Claude, and most modern AI language systems!

GPT

GPT (Generative Pre-trained Transformer):

- Sees text from left to right only
- Specialized for generation tasks:
  - Writing completion
  - Translation
  - Creative writing

How Large Language Models Learn

Modern language models like GPT & Claude learn in stages:

1. Pre-training:

- Learn general language patterns from vast amounts of text (books, websites, etc.)
- Trillions of words of training data
- Learn grammar, facts, reasoning, etc.

2. Fine-tuning:

- Additional training on high-quality examples
- Learn to follow instructions
- Avoid harmful outputs

3. RLHF (Reinforcement Learning from Human Feedback):

- Humans rate model responses
- Model learns which outputs humans prefer
- Leads to more helpful, harmless, honest responses

In-Context Learning: A Surprising Ability

Large language models can learn from examples in your prompt!

Example prompt:

```
Translate English to French:
English: The house is blue.
French: La maison est bleue.
English: The cat is black.
French: Le chat est noir.
English: What time is it?
French:
```

The model will likely respond: "Quelle heure est-il?"

No additional training needed—it learns from your examples!

Limitations of Current Language Models

Despite impressive abilities, language models have important limitations:

Hallucinations:

- May generate plausible but incorrect information
- Might confidently state something false

Reasoning limitations:

- Struggle with complex logic
- May make mathematical errors

No real understanding:

- No actual experiences of the world
- Understanding based entirely on patterns in text
- No ability to verify information against reality

Enhancing Language Models with Retrieval

Retrieval-Augmented Generation (RAG) helps address limitations:

> RAG is like giving the language model access to a searchable library of reliable information.

How it works:

1. Store trusted information in a database
2. When asked a question, search the database
3. Give the language model both the question and the search results
4. Model generates an answer using both its pre-trained knowledge and the search results

Benefits: More accurate, up-to-date, and verifiable information

Ethical Considerations

The power of language models raises important concerns:

Bias and fairness:

- Models learn biases present in their training data
- May reproduce or amplify stereotypes
- Ongoing research to detect and reduce bias

Misinformation risks:

- Generation of convincing but false content
- Potential for misuse in spreading misinformation
- Need for tools to verify AI-generated content

Privacy concerns:

- Models may memorize sensitive information from training data
- Risk of exposing private information in responses
- Need for careful data handling and model design

Future Directions

Where is language model technology heading?

Multimodal models:

- Combining text with images, audio, video
- Understanding and generating across multiple formats

Tool use and integration:

- Models that can use external tools (calculators, search engines, etc.)
- API integration with other systems
- Example: ChatGPT Plugins, Claude with web search

Smaller, more efficient models:

- More capabilities with less computing power
- Models that can run on personal devices
- Specialized models for specific tasks

Key Takeaways

The big ideas to remember:

1. **Language models predict the next word in a sequence** (surprisingly powerful!)
2. **Attention mechanisms revolutionized language modeling** by connecting any words directly
3. **Transformers** (combining several innovations) power most modern AI language systems
4. **Language models have impressive abilities** but important limitations
5. **Understanding the capabilities and limitations** of these models is crucial for using them effectively