Deep Learning for Text

Applied Text Mining

Maryam Movahedifar

14-17 July 2025

University of Bremen, Germany
movahedm@uni-bremen.de

Universität Bremen

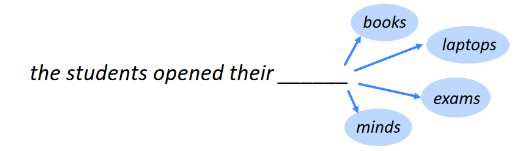
DATA SCIENCE CENTER

Outline

Language Modeling

What is a Language Model?

- A statistical tool that predicts the next word in a sequence.
- Essential in natural language processing (NLP).



Key Points:

- *Predictive Power:* Uses context from earlier words.
- *Applications:* Translation, speech recognition, text generation.
- *Training:* Learns from large-scale text data.

Formula: Given a sequence $x_1, x_2, \ldots, x_t$, it estimates:

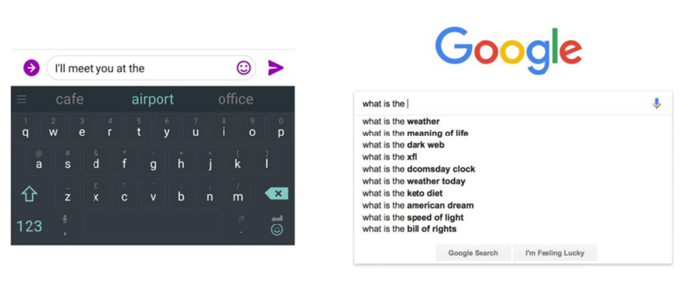
$$P(x_{t+1} \mid x_1, x_2, \ldots, x_t)$$

Another View: What Does a Language Model Do?

- Language model can be as a system that assigns a probability to a piece of text.

$$
\begin{aligned}
P(\boldsymbol{x}^{(1)}, \ldots, \boldsymbol{x}^{(T)}) &= P(\boldsymbol{x}^{(1)}) \times P(\boldsymbol{x}^{(2)} | \, \boldsymbol{x}^{(1)}) \times \cdots \times P(\boldsymbol{x}^{(T)} | \, \boldsymbol{x}^{(T-1)}, \ldots, \boldsymbol{x}^{(1)}) \\
&= \prod_{t=1}^{T} \underbrace{P(\boldsymbol{x}^{(t)} | \, \boldsymbol{x}^{(t-1)}, \ldots, \boldsymbol{x}^{(1)})}_{\text{This is what our LM provides}}
\end{aligned}
$$

- You use language models every day:



Types of Language Models

n-gram Models

Predict the next word using the previous $n - 1$ words. Simple and fast but limited to short context.

Neural Network Models

Use deep learning to understand longer context and generate more natural text. Examples: GPT, BERT.

Source: `http://web.stanford.edu/class/cs224n/`

n-gram Language Models

The sentence: **the students opened their ....**

- **Question:** How to learn a Language Model?
- **Answer (pre-Deep Learning):** Learn an n-gram Language Model!

- **Definition:** An n-gram is a chunk of n consecutive words.

- Unigrams: "the", "students", "opened", "their"
- Bigrams: "the students", "students opened", "opened their"
- Trigrams: "the students opened", "students opened their"
- Four-grams: "the students opened their"

- **Idea:** Collect statistics on how frequent different n-grams are, then use these to predict the next word.

Using Markov Chains in n-gram Language Models

First, we make the **Markov assumption***: the next word* x_{t+1} *depends only on the preceding* $n-1$ *words.*

$$P(\boldsymbol{x}^{(t+1)}|\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(1)}) = P(\boldsymbol{x}^{(t+1)}|\overbrace{\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)}}^{n\text{-1 words}})$$

(assumption)

prob of a n-gram

prob of a (n-1)-gram

$$= \frac{\boxed{P(\boldsymbol{x}^{(t+1)},\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}}{\boxed{P(\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}}$$

(definition of conditional prob)

- **Question:** How do we get these n-gram and (n-1)-gram probabilities?
- **Answer:** By counting their occurrences in a large text corpus!

$$\approx \frac{\mathrm{count}(\boldsymbol{x}^{(t+1)},\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}{\mathrm{count}(\boldsymbol{x}^{(t)},\ldots,\boldsymbol{x}^{(t-n+2)})}$$

Trigram Language Models: Example

Suppose we are learning a **4-gram** Language Model.

~~as the proctor started the clock, the~~ *students opened their* ______

discard

condition on this

$$P(w \mid \text{students opened their}) = \frac{\text{count(students opened their } w)}{\text{count(students opened their)}}$$

For example, suppose that in the corpus:

- "students opened their" occurred 1000 times
- "students opened their books" occurred 400 times
  - → P(books | students opened their) = 0.4
- "students opened their exams" occurred 100 times
  - → P(exams | students opened their) = 0.1

Should we have discarded the "proctor" context?

Generating Text with an n-gram Language Model

You can also use a Language Model to generate text:

today the price of gold per ton , while production of shoe lasts and shoe industry , the bank intervened just after it considered and rejected an imf demand to rebuild depleted european stocks , sept 30 end primary 76 cts a share .

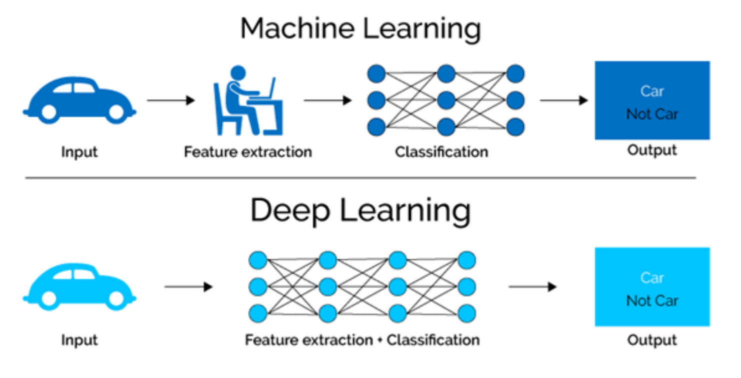
Surprisingly grammatical!

...but incoherent. We need to consider more than three words at a time if we want to model language well.

Deep Learning

What is Deep Learning?

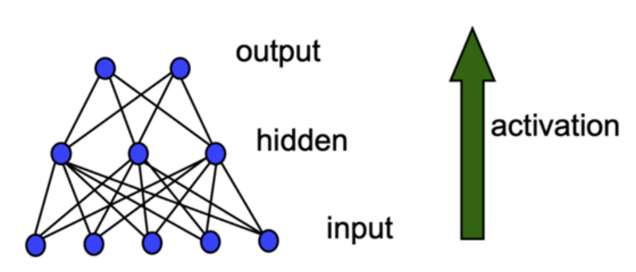
- Subfield of machine learning focused on learning hierarchical representations.
- Exceptionally effective at learning patterns in data.
- Uses multiple layers to build complex feature representations.



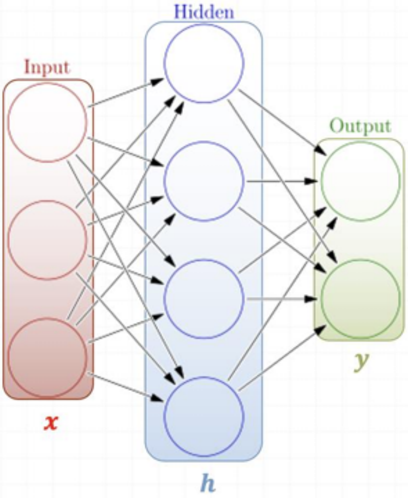
Feed-forward Neural Networks

Architecture

- Multi-layer networks with input, hidden, and output layers.
- Each layer fully connected to the next.
- Activation functions apply non-linearity.



Feed-Forward Neural Networks Structure



Weights

$$h = \sigma(W_1 x + b_1)$$

$$y = \sigma(W_2 h + b_2)$$

Activation functions

4 + 2 = 6 neurons (not counting inputs)
[3 x 4] + [4 x 2] = 20 weights
4 + 2 = 6 biases

26 learnable **parameters**

One Forward Pass Example

Text (input) representation

TFIDF

Word embeddings

....

$$\underbrace{\begin{bmatrix} 0.2 & -0.5 & 0.1 \\ 2.0 & 1.5 & 1.3 \\ 0.5 & 0.0 & 0.25 \\ -0.3 & 2.0 & 0.0 \end{bmatrix}}_{\mathbf{W}} \underbrace{\begin{bmatrix} 0.1 \\ 0.2 \\ 0.3 \end{bmatrix}}_{x_i} + \underbrace{\begin{bmatrix} 1.0 \\ 3.0 \\ 0.025 \\ 0.0 \end{bmatrix}}_{b} = \underbrace{\begin{bmatrix} 0.95 \\ 3.89 \\ 0.15 \\ 0.37 \end{bmatrix}}_{\sigma(x_i; W, b)} \begin{matrix} \text{very positive} \\ \boxed{\text{positive}} \\ \text{negative} \\ \text{very negative} \end{matrix}$$

Training Feed-Forward Networks

- Goal: Learn the best parameters θ by minimizing prediction error.
- Optimize the cost function $J(\theta)$ using gradient descent:

$$\theta_j^{\text{new}} = \theta_j^{\text{old}} - \alpha \frac{\partial J(\theta)}{\partial \theta_j}$$

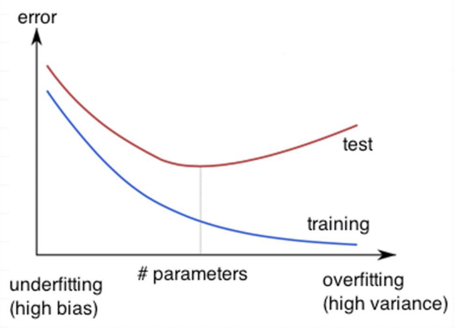
- Use the chain rule to compute gradients layer by layer (backpropagation).
- Train over many epochs, sometimes with random restarts to escape bad solutions.

Overfitting and Regularization

Overfitting: When a model fits the training data too well but performs poorly on new, unseen data.

How to prevent overfitting:

- **Early stopping:** Stop training when validation error starts increasing.
- **Dropout:** Randomly turn off neurons during training to prevent reliance on specific paths.
- **Weight decay (L2):** Penalize large weights to keep the model simpler.
- **Hyperparameter tuning:** Adjust learning rate, number of hidden units, etc.

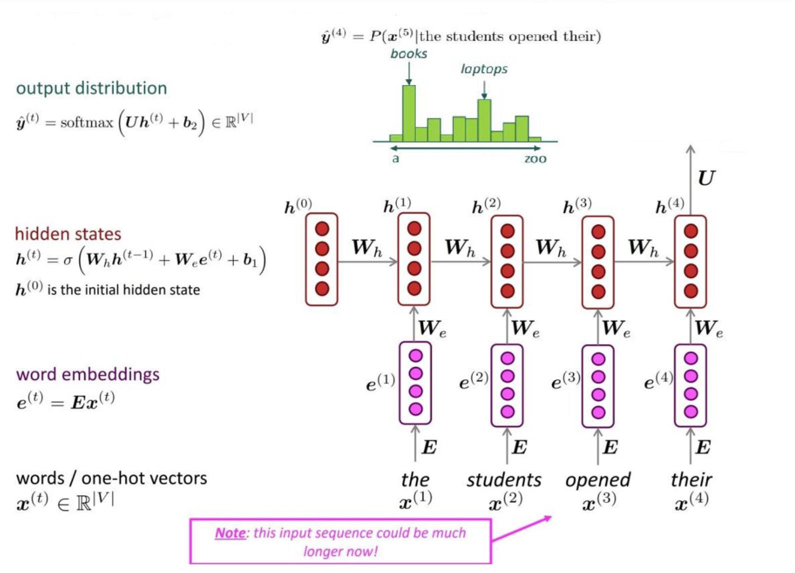


Recurrent Neural Networks

Introduction to RNNs

- Recurrent Neural Networks (RNNs) are designed to process **sequences** of data.
- They have feedback loops that let information flow from one step to the next.
- This allows RNNs to capture **temporal dependencies** in data like text or time series.
- A Simple Recurrent Network (SRN) uses the previous hidden state along with current input to compute the new state.

A Simple RNN Language Model



RNN: Advantages and Disadvantages

Advantages

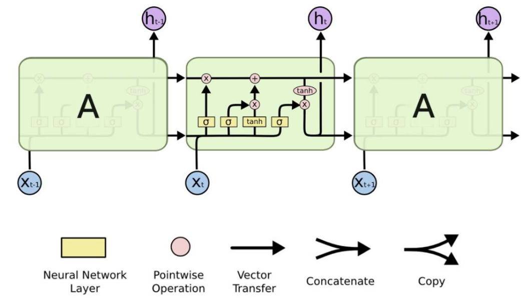
- Can process any length input
- Step t can use past information
- Model size fixed regardless of input length
- Same weights applied at every time step

Disadvantages

- Recurrent computation is slow
- Hard to retain long-term dependencies

LSTM (Long Short-Term Memory) Networks

- Designed to solve the vanishing gradient problem in RNNs.
- Uses a memory cell and three gates:
  - **Forget gate** – discards irrelevant information.
  - **Input gate** – stores new relevant input.
  - **Output gate** – passes on useful memory.
- Enables learning of long-term dependencies in sequential data.



Attention Mechanism

Attention Mechanism

Focus on Relevant Input: Allows the network to dynamically attend to different parts of the input sequence.

Boosts Performance: Greatly improves results in tasks like machine translation, speech recognition, and image captioning.

Dynamic Weights: Computes weights for input elements to decide their importance during processing.

Summary

Summary of Language Models

- **Predict Next Word:** Language models estimate the most likely next word in a sequence.
- **Deep Learning Power:** Automatically extracts useful features from data without manual intervention.
- **Sequential Models:** RNNs process sequences and are improved by LSTM units and attention mechanisms.
- **Variety of Models:** Includes classical n-grams, neural networks, and modern transformers like GPT.

Practical