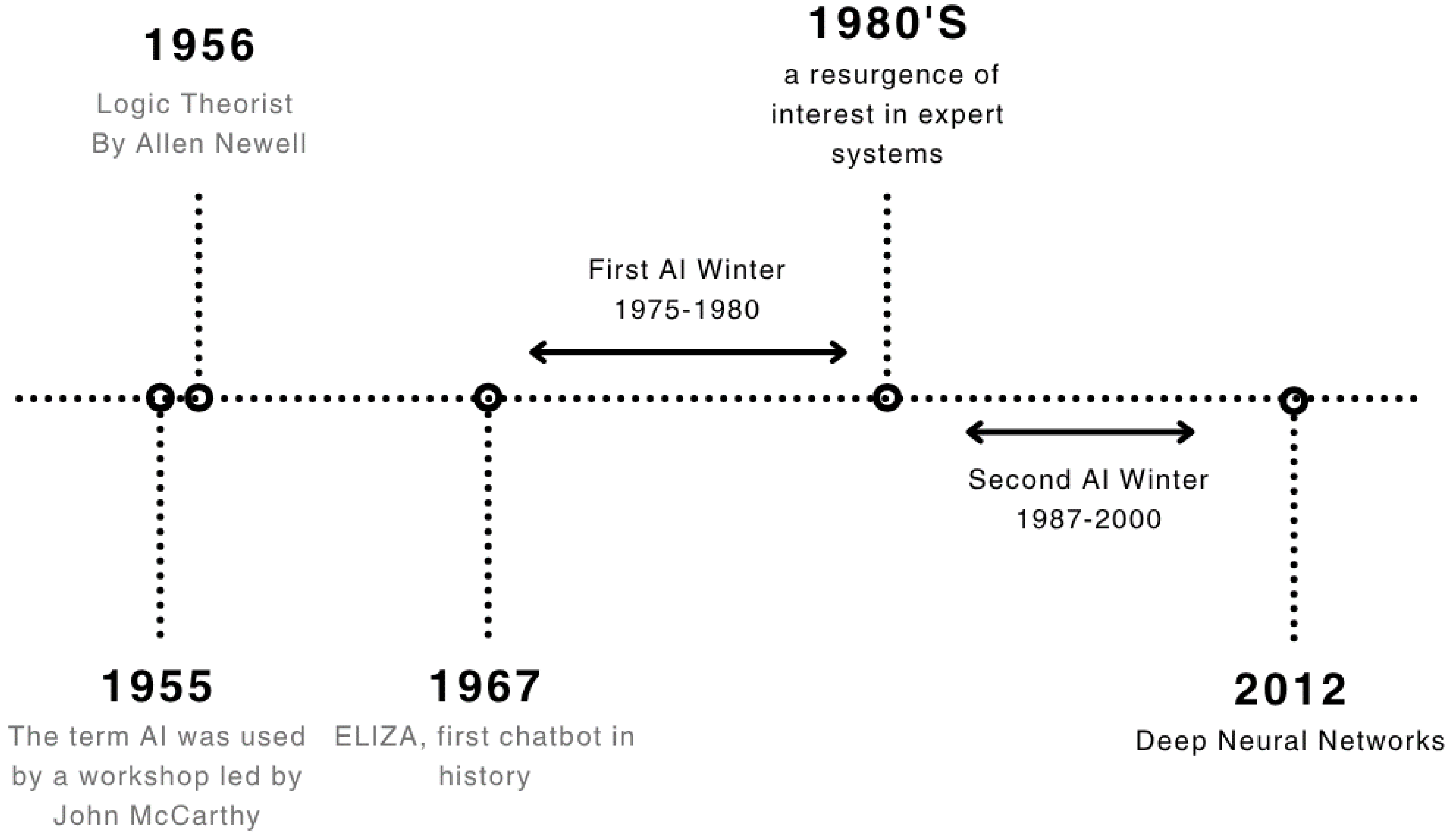
An Overview of Language Modeling

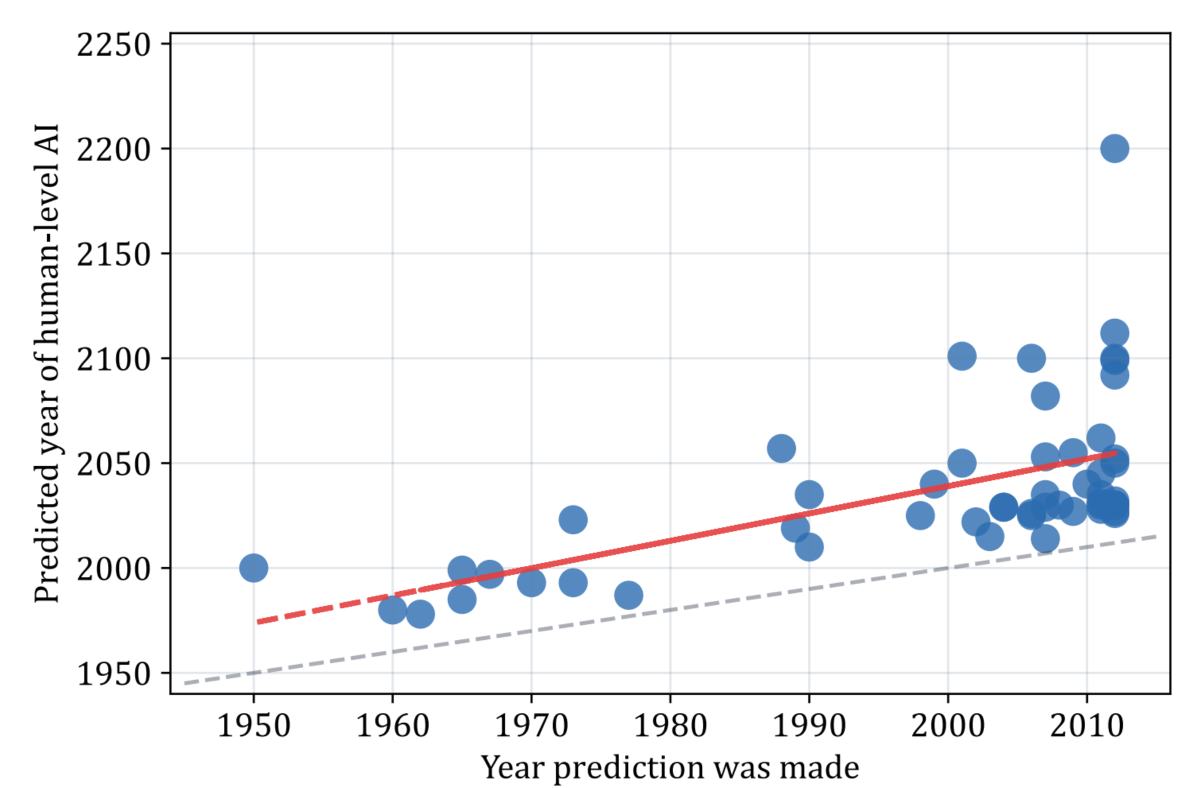
Based on Andriy Burkov’s 100-Page LM Book

Part 1

By Pouria Moradpour
Science Club


1956
Logic Theorist
By Allen Newell
1980'S
a resurgence of
interest in expert
systems
First AI Winter
1975-1980
Second AI Winter
1987-2000
1955
The term AI was used
by a workshop led by
John McCarthy
1967
ELIZA, first chatbot in
history
2012
Deep Neural Networks

“decade after decade, AGI remained roughly 25 years on the horizon”



Basic Definitions

A model is typically represented by a mathematical equation:

$$y = f(x)$$

In machine learning, the goal is to compile a dataset of examples and use them to build f, so when f is applied to a new, unseen x, it produces a y that gives meaningful insight into x.

Now, Let’s see a very basic example

Linear Regression

$$f(x) \coloneqq wx + b$$

$$\mathcal{D} = \{\ (x^{(1)}, y^{(1)}), (x^{(2)}, y^{(2)}), \ \ldots\ , (x^{(N)}, y^{(N)})\ \}$$

- Supervised vs. Unsupervised: our focus is on the former.

Linear Regression: MSE Loss Function

$$e_i := (\hat{y}_i - y_i)^2$$

$$J(w, b) = (1/N) \sum^{n}_{i=1} (\hat{y_i} - y_i)^2$$

$$J(w, b) = (1/N) \sum^{n}_{i=1} (wx_i + b - y_i)^2$$

An Example, House Price & It's Size

$$\mathcal{D} = \{ (150, 200), (200, 600), (260, 500) \}$$

$$J(w, b) = (1/3) [(w \cdot 150 + b - 200)^2 + (w \cdot 200 + b - 600)^2 + (w \cdot 260 + b - 500)^2]$$

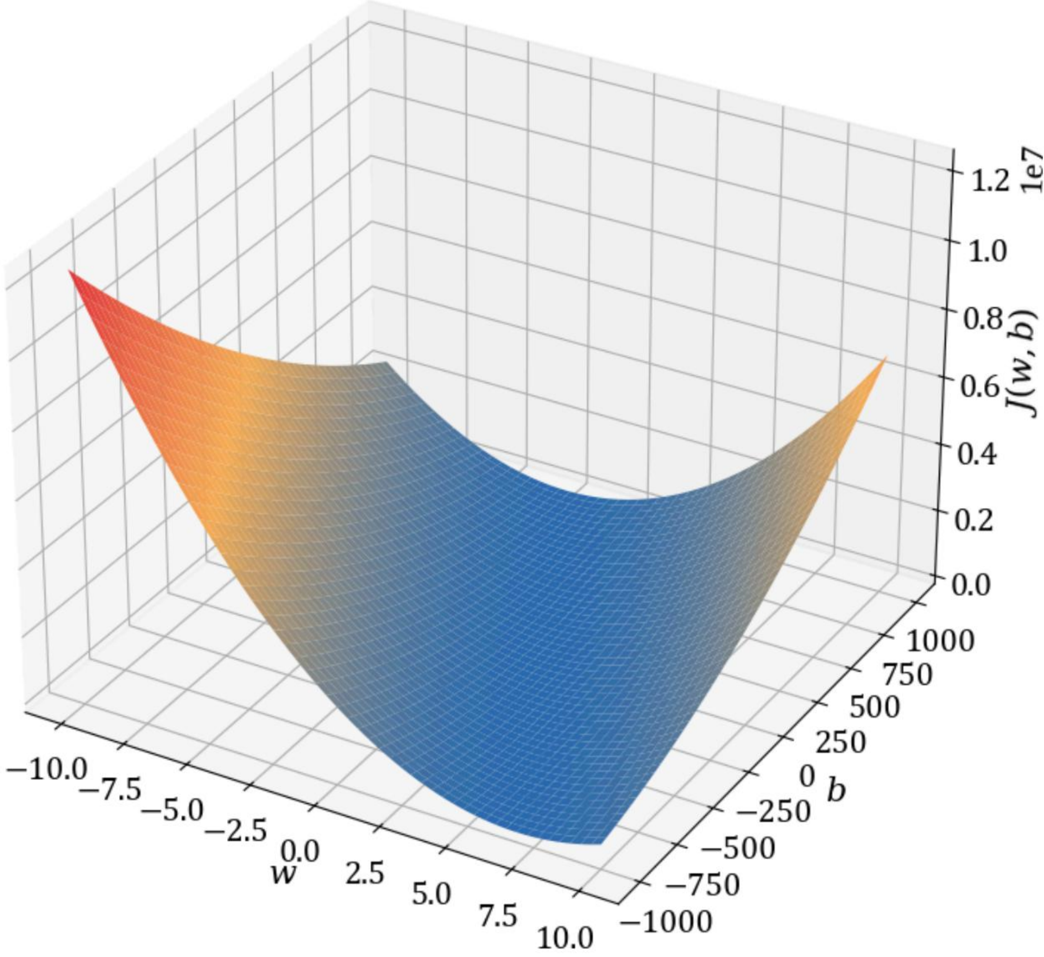
Plotting J

$J(w, b) = (1/3)\,[\,(w\cdot 150 + b - 200)^2 + (w\cdot 200 + b - 600)^2 + (w\cdot 260 + b - 500)^2\,]$

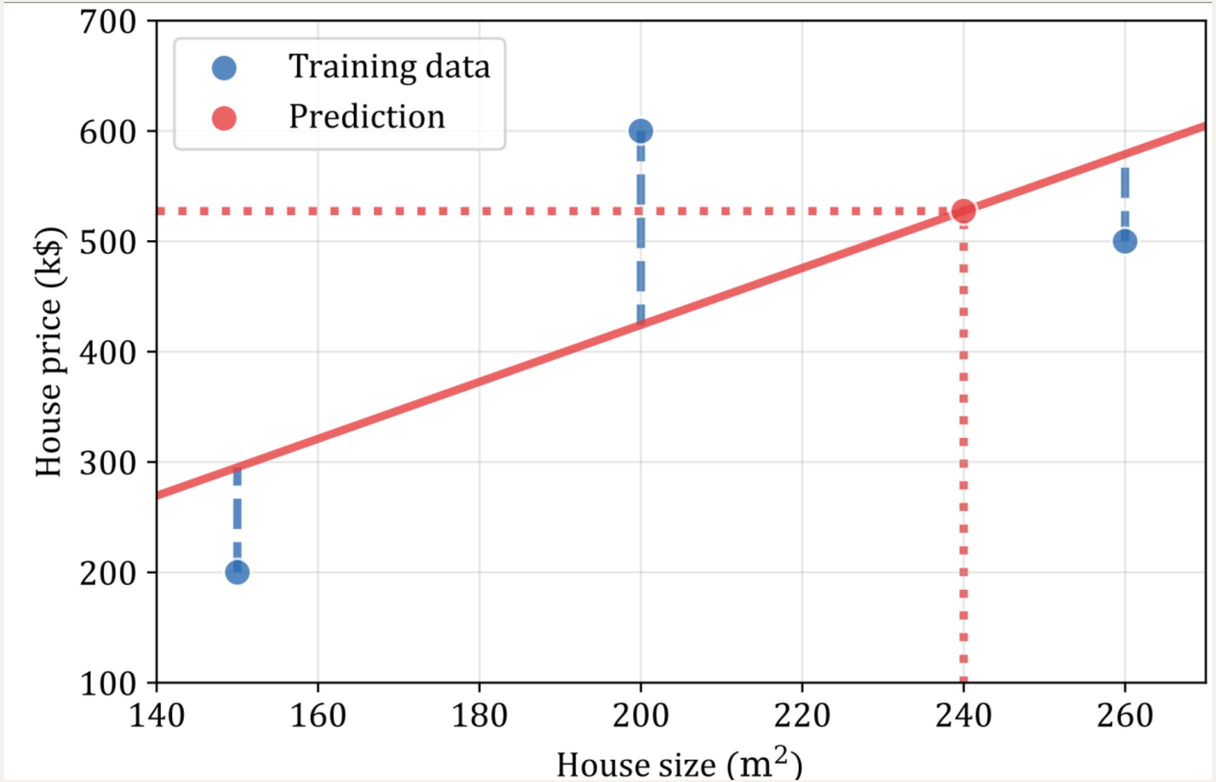
$$w^* = 2.58, b^* = -91.76$$

$$f(x) = 2.58\mathrm{w} - 91.76$$

But how can we find the minimum...?
Requires basic calculus knowledge. (but actually Gradient Descent and Automatic differentiation are used)



Using the Model



Four-Step Machine Learning Process

01

Collect a dataset:

$\mathcal{D} = \{ (x^1, y^1), (x^2, y^2), \dots, (x^N, y^N) \}$

02

Define the model's structure:

$f(x) = w \cdot x + b$

03

Define the loss function

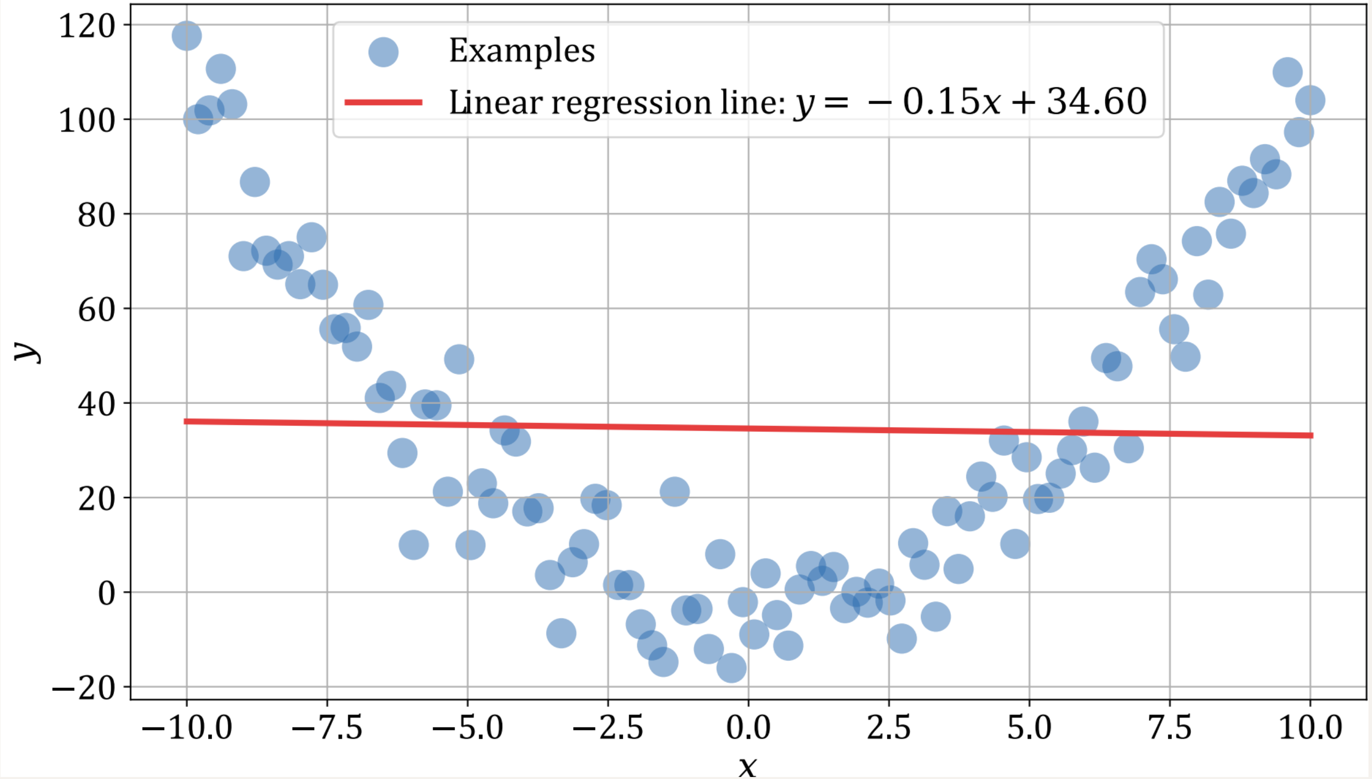
$J(w, b) = (1 / N) \sum^{n}_{i=1} (f(x_i) - y_i)^2$

04

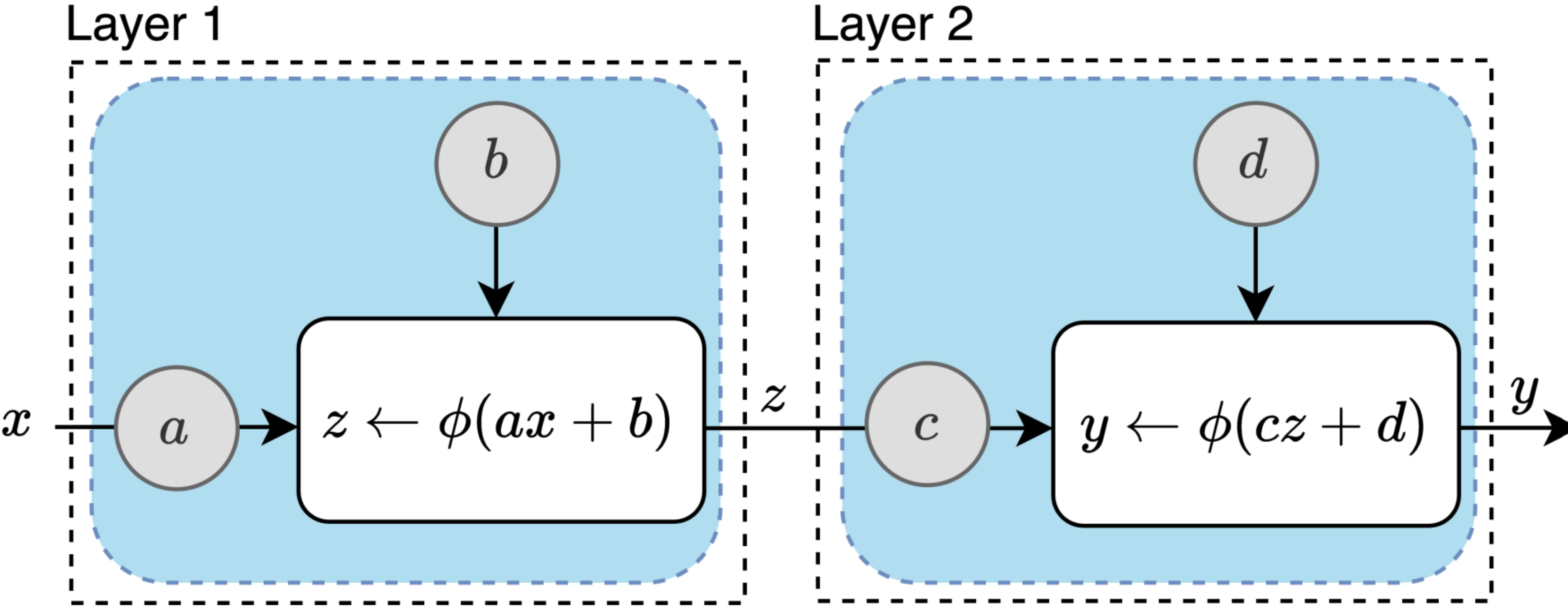
Minimize the loss

Find w^*, b^* such that $J(w^*, b^*)$ is minimized

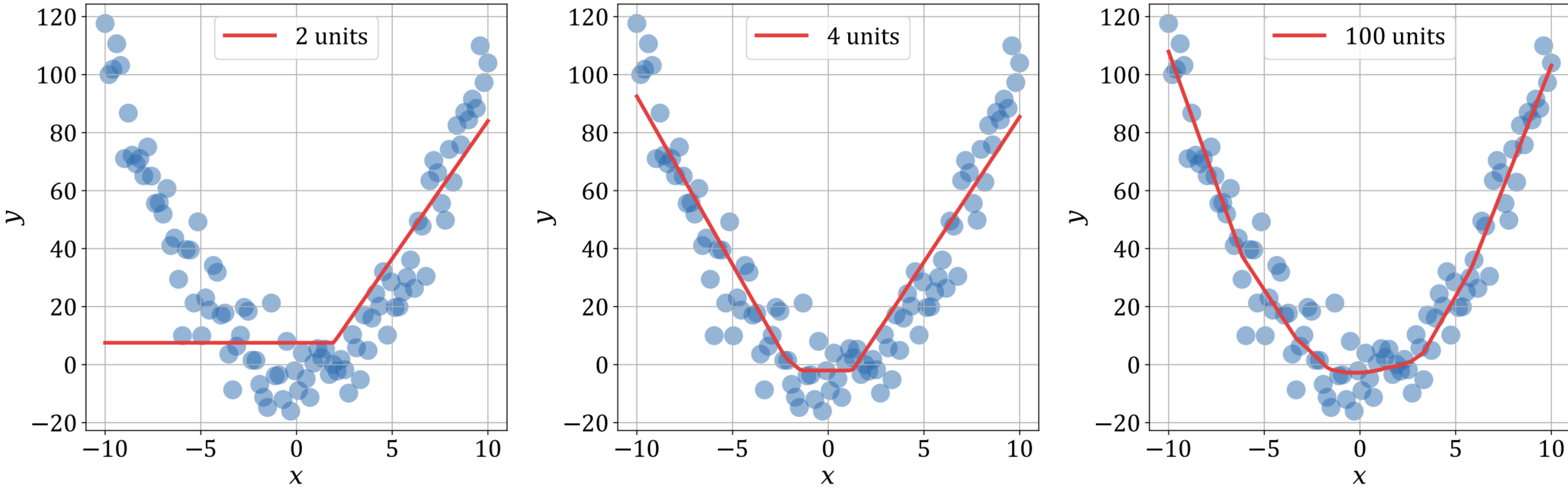
Where Linear Regression Fails



Computational Graph with Activation Functions



Increasing Model size improves performance



Language Modeling

Bag-of-Words

["a", "and", "are", "discovery", "enjoy", "everyone", "folk", "for", "fun", "great", "important", "interesting", "is", "learning", "listen", "math", "movie", "movies", "music", "research", "rock", "science", "to", "today", "very", "watching"]

1. Movies are fun for everyone.
2. Watching movies is great fun.
3. Enjoy a great movie today.
4. Research is interesting and important.
5. Learning math is very important.
6. Science discovery is interesting.
7. Rock is great to listen to.
8. Listen to music for fun.
9. Music is fun for everyone.
10. Listen to folk music!

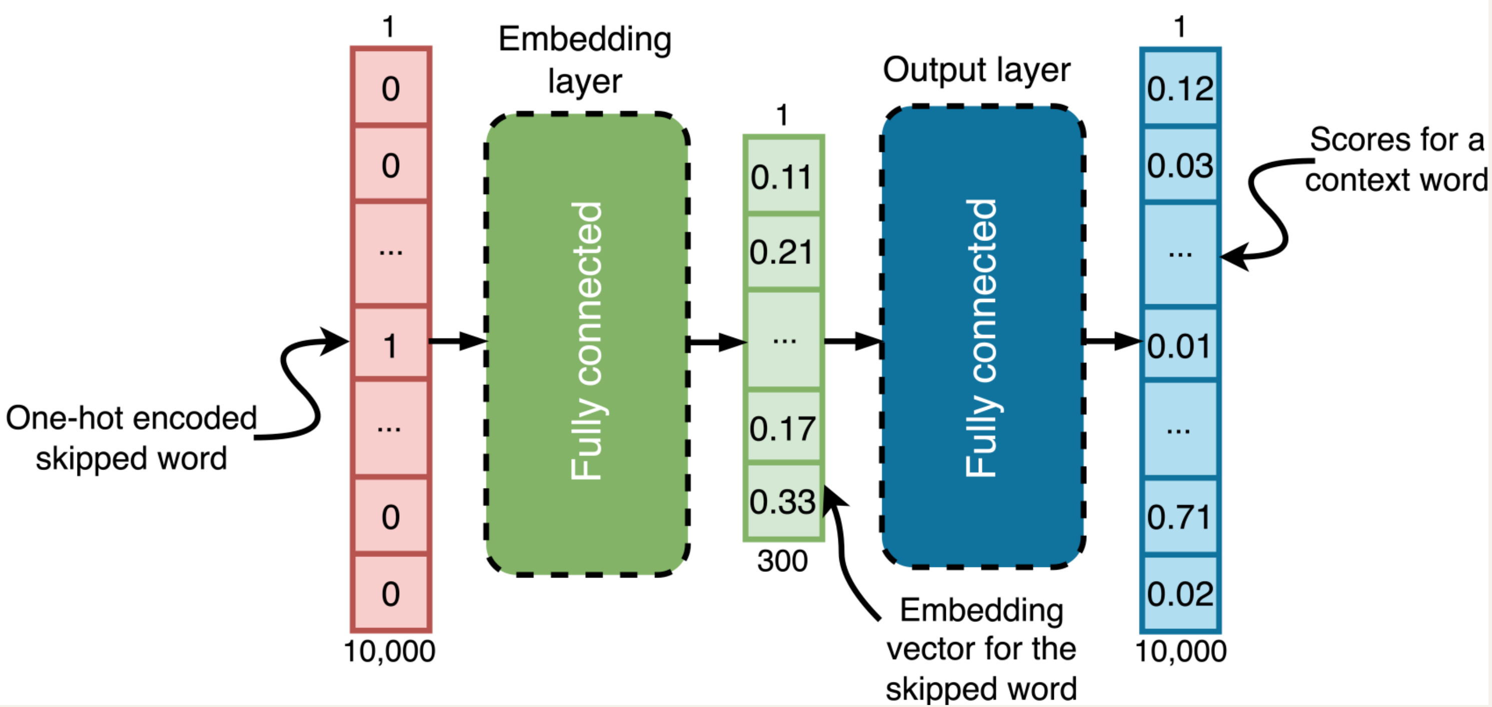
Bag-of-Words: DTM

Doc	a	and	…	fun	…	listen	math	…	science	…	watching
1	0	0	…	1	…	0	0	…	0	…	0
2	0	0	…	1	…	0	0	…	0	…	1
3	1	0	…	0	…	0	0	…	0	…	0
4	0	1	…	0	…	0	0	…	0	…	0
5	0	0	…	0	…	0	1	…	0	…	0
6	0	0	…	0	…	0	0	…	1	…	0
7	0	0	…	0	…	1	0	…	0	…	0
8	0	0	…	1	…	1	0	…	0	…	0
9	0	0	…	1	…	0	0	…	0	…	0
10	0	0	…	0	…	1	0	…	0	…	0

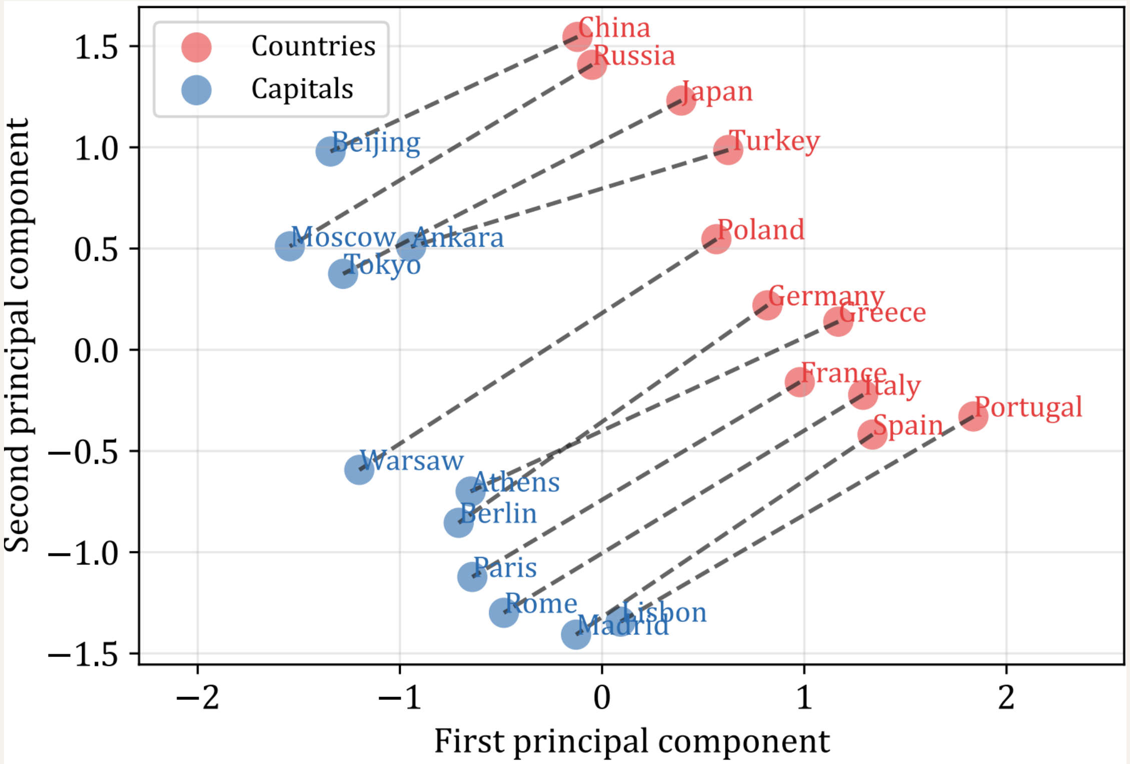
Bag-of-Words: Limitaitions

- **Ignores Word Order**
- **No Contextual Understanding**
- **Sparse and High-Dimensional Vectors**
- **Cannot Handle Out-of-Vocabulary Words**

Word Embedding



Word Embedding: Semantic Similarity



- 200-dimensional projected by PCA
- King – man + woman = queen!

Byte-Pair Encoding

- **Start with characters as tokens**

Example: "lower" → ["l", "o", "w", "e", "r"]

- **Count all adjacent token pairs in the corpus**

- **Find the most frequent pair**

Example: "l o", "o w", "w e", "e r“

- **Merge the most frequent pair into a new token**

"e r" → "er"
"lower" → ["l", "o", "w", "er"]

- **Repeat Steps 2–4 until vocabulary reaches the desired size**

A Language Model

$$P(w^1, w^2, \ldots, w_\mathrm{n}) =$$
$$P(w_1) \cdot P(w_2 \mid w_1) \cdot P(w_3 \mid w_1, w_2) \cdot \ldots \cdot P(w_\mathrm{n} \mid w_1, \ldots, w_{\mathrm{n}-1})$$

- This is a discrete probability distribution

Count-Based Language Model:
A Trigram

$$P(w_{\mathrm{i}} \mid w_{\mathrm{i-2}}, w_{\mathrm{i-1}}) = C(w_{\mathrm{i-2}}, w_{\mathrm{i-1}}, w_{\mathrm{i}}) \; / \; C(w_{\mathrm{i-2}}, w_{\mathrm{i-1}})$$

Count-Based Trigram: Back-Off

Expression	Condition
$\frac{C(t_{i-2}, t_{i-1}, t_i)}{C(t_{i-2}, t_{i-1})}$	if $C(t_{i-2}, t_{i-1}, t_i) > 0$
$\Pr(t_i \mid t_{i-1})$	if $C(t_{i-2}, t_{i-1}, t_i) = 0$ and $C(t_{i-1}, t_i) > 0$
$\Pr(t_i)$	otherwise

Time for Some Code!

Let’s See a Count-Based Language Model in Its Natural Habitat!

Thanks!