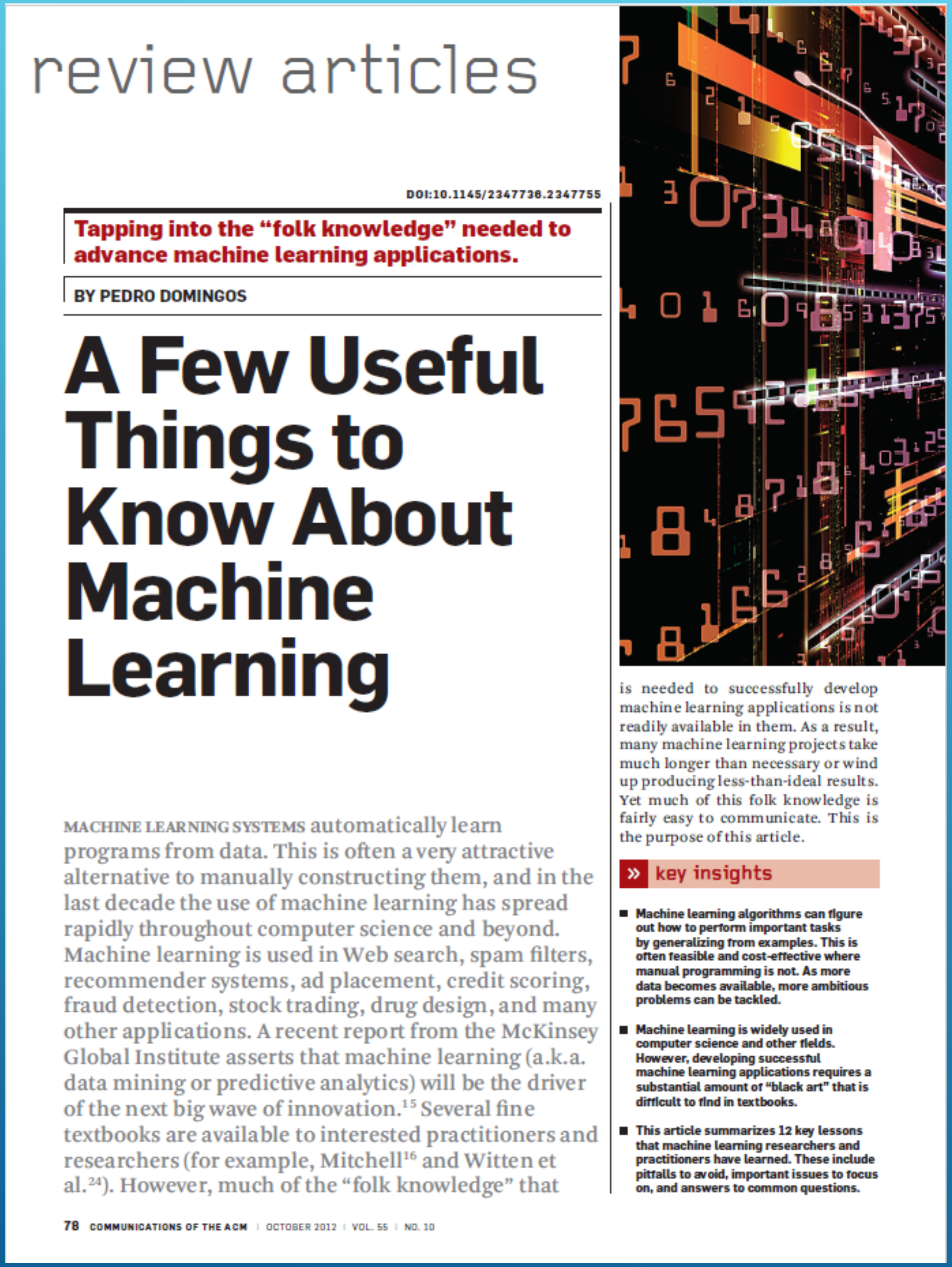
ANATOMY OF A LEARNING ALGORITHM

Scott O'Hara
Metrowest Developers Machine Learning Group
01/13/2021

- Concise definitions and well-thought-out examples.
- ~140 pages.
- Covers mostly supervised learning.
- Provides pointers to things it doesn’t cover.
- Generally, a good summary of things you should know about ML.

ONLINE RESOURCES

- Website: http://themlbook.com/
- Wiki: http://themlbook.com/wiki/doku.php
- Git: https://github.com/aburkov/theMLbook



review articles

DOI:10.1145/2347736.2347755

Tapping into the "folk knowledge" needed to advance machine learning applications.

BY PEDRO DOMINGOS

A Few Useful Things to Know About Machine Learning

MACHINE LEARNING SYSTEMS automatically learn programs from data. This is often a very attractive alternative to manually constructing them, and in the last decade the use of machine learning has spread rapidly throughout computer science and beyond. Machine learning is used in Web search, spam filters, recommender systems, ad placement, credit scoring, fraud detection, stock trading, drug design, and many other applications. A recent report from the McKinsey Global Institute asserts that machine learning (a.k.a. data mining or predictive analytics) will be the driver of the next big wave of innovation.[15] Several fine textbooks are available to interested practitioners and researchers (for example, Mitchell[16] and Witten et al.[24]). However, much of the "folk knowledge" that is needed to successfully develop machine learning applications is not readily available in them. As a result, many machine learning projects take much longer than necessary or wind up producing less-than-ideal results. Yet much of this folk knowledge is fairly easy to communicate. This is the purpose of this article.

» key insights

- Machine learning algorithms can figure out how to perform important tasks by generalizing from examples. This is often feasible and cost-effective where manual programming is not. As more data becomes available, more ambitious problems can be tackled.
- Machine learning is widely used in computer science and other fields. However, developing successful machine learning applications requires a substantial amount of "black art" that is difficult to find in textbooks.
- This article summarizes 12 key lessons that machine learning researchers and practitioners have learned. These include pitfalls to avoid, important issues to focus on, and answers to common questions.

- Author: Dr. Pedro Domingos
- Communications of the ACM – October 2012.
- Written in 2012, Domingos attempted to convey the "black art" of machine learning in an article covering 12 topics i.e. the "folk knowledge" and rules of thumbs that are know by ML practitioners but that are not taught in universities.

THE 100 PAGE ML BOOK: CONTENTS

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Supervised learning algorithm can be reduced to three components:

- **Representation**
- **Evaluation**
- **Optimization**

Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

LEARNING = **REPRESENTATION** + EVALUATION + OPTIMIZATION

- A **classifier** must be represented in some formal language that the computer can handle.
- Choosing a representation is tantamount to choosing the set of classifiers that can be learned. This is called the ***hypothesis space***.
- If a classifier is not in the hypothesis space, it cannot be learned.

Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

LEARNING = REPRESENTATION + **EVALUATION** + OPTIMIZATION

- An **evaluation function** is needed to distinguish good classifiers from bad ones.
- Sometimes called the **objective function** or the **scoring function.**

Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

1. LEARNING = REPRESENTATION + EVALUATION + **OPTIMIZATION**

- A method is needed to search the set of classifiers for the highest-scoring one.
- The choice of optimization technique determines the efficiency of the learner.
- The optimization technique uses the evaluation function.

Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Table 1. The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

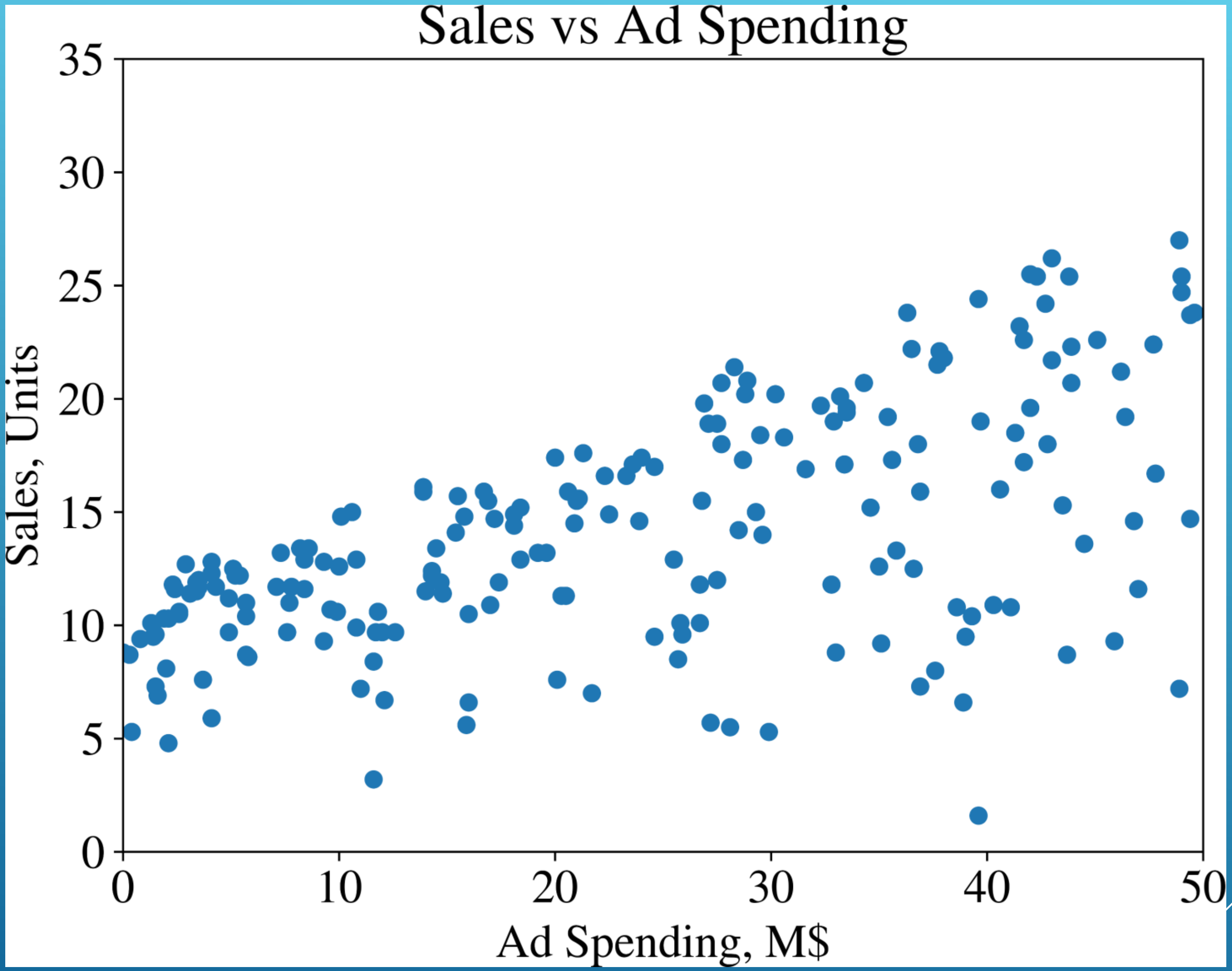
Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

LINEAR REGRESSION EXAMPLE

- Representation: **line**
- Evaluation: **minimize mean squared error**
- Optimization: **gradient descent**

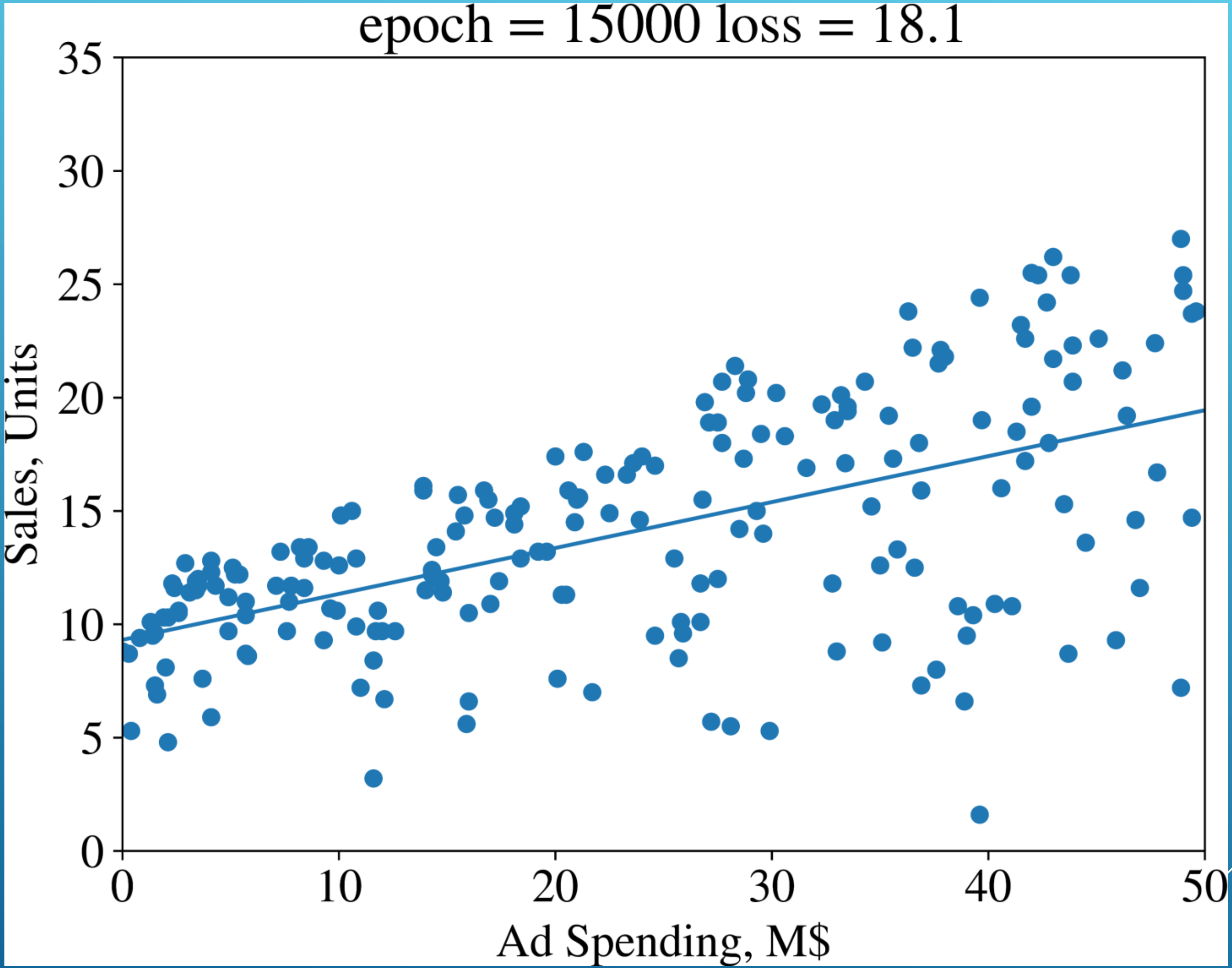
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

EXAMPLE: SALES VS AD SPENDING



Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

FIT REGRESSION LINE TO DATA TO PREDICT UNIT SALES GIVEN AD SPENDING



Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

ADVERTISING.CSV

	A	B	C	D	E
1		TV	radio	newspaper	sales
2	1	230.1	37.8	69.2	22.1
3	2	44.5	39.3	45.1	10.4
4	3	17.2	45.9	69.3	9.3
5	4	151.5	41.3	58.5	18.5
6	5	180.8	10.8	58.4	12.9
7	6	8.7	48.9	75	7.2
8	7	57.5	32.8	23.5	11.8
9	8	120.2	19.6	11.6	13.2
10	9	8.6	2.1	1	4.8
11	10	199.8	2.6	21.2	10.6

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

REPRESENTATION: THE LINEAR REGRESSION MODEL

Model Parameters

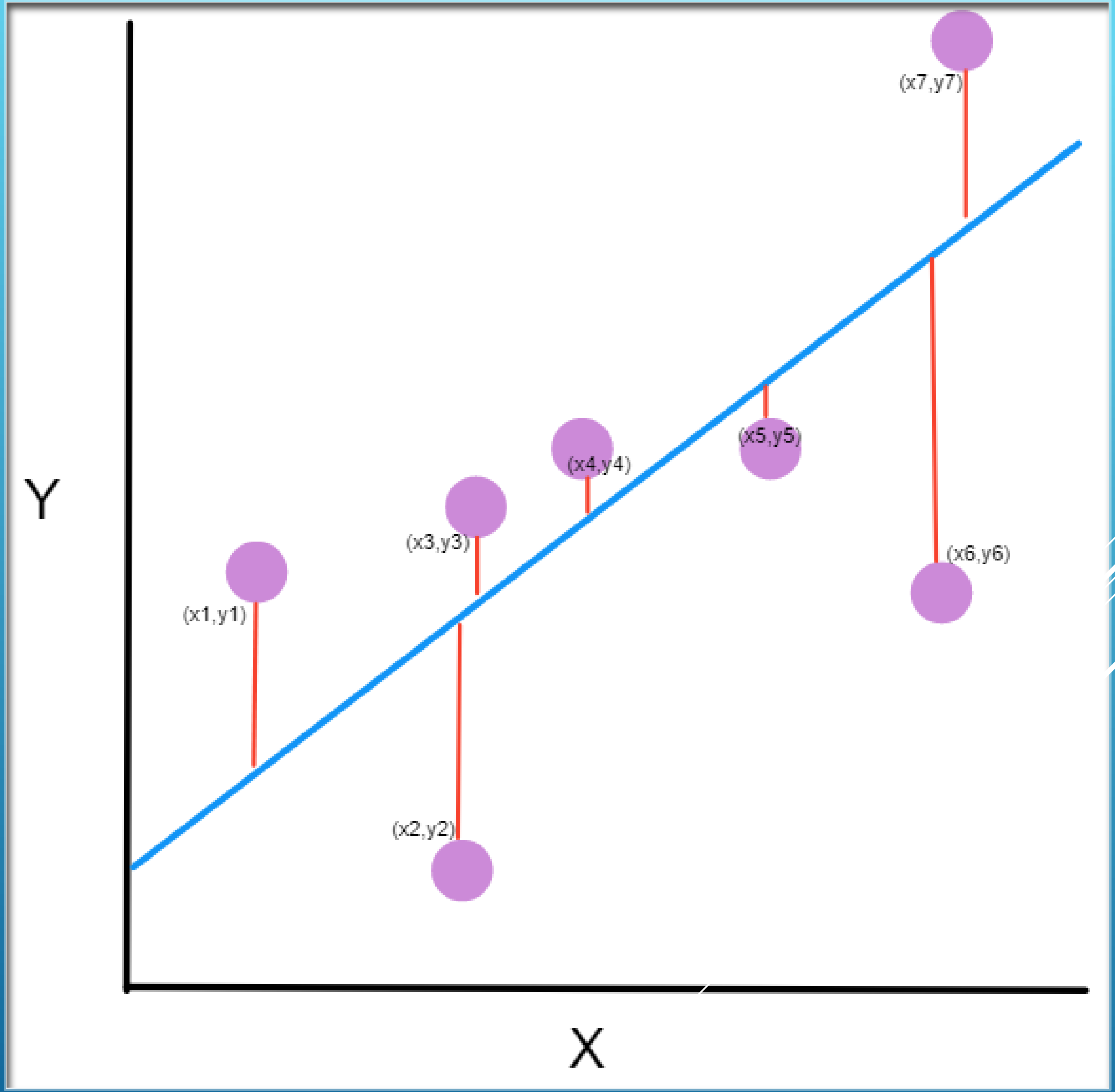
$$f(x) = wx + b$$

Unit Sales

Radio Ad Spending

EVALUATION:

MINIMIZE
MEAN-SQUARED ERROR



Credit: image from https://www.freecodecamp.org/news/machine-learning-mean-squared-error-regression-line-c7dde9a26b93/. Retrieved 01/13/2021.

EVALUATION: MINIMIZE *MEAN-SQUARED ERROR*

$$l \stackrel{\text{def}}{=} \frac{1}{N} \sum_{i=1}^{N} (y_i - (wx_i + b))^2 .$$

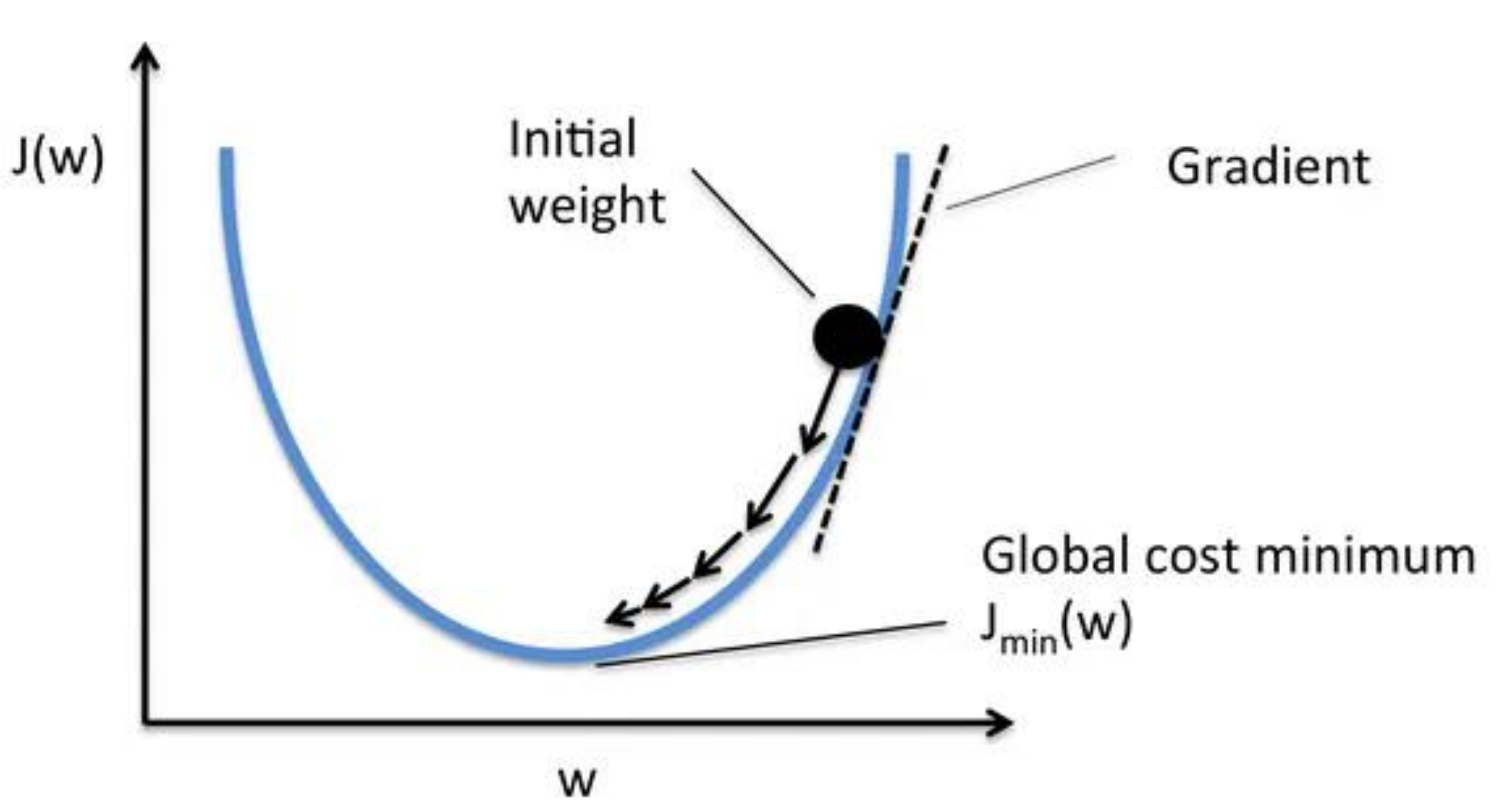
Loss

Size of Training Set

Actual Value

Model Prediction

OPTIMIZATION: GRADIENT DESCENT



OPTIMIZATION: GRADIENT DESCENT

we look for such values for w and b that minimize the mean squared error:

$$l \stackrel{\text{def}}{=} \frac{1}{N}\sum_{i=1}^{N}(y_i - (wx_i + b))^2.$$

Gradient descent starts with calculating the partial derivative for every parameter:

$$\frac{\partial l}{\partial w} = \frac{1}{N}\sum_{i=1}^{N} -2x_i(y_i - (wx_i + b));$$

$$\frac{\partial l}{\partial b} = \frac{1}{N}\sum_{i=1}^{N} -2(y_i - (wx_i + b)).$$

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

GRADIENT DESCENT: EPOCHS, LEARNING RATE

Gradient descent proceeds in **epochs**. An epoch consists of using the training set entirely to update each parameter. In the beginning, the first epoch, we initialize[2] $w \leftarrow 0$ and $b \leftarrow 0$. The partial derivatives, $\frac{\partial l}{\partial w}$ and $\frac{\partial l}{\partial b}$ given by eq. 1 equal, respectively, $\frac{-2}{N}\sum_{i=1}^{N} x_i y_i$ and $\frac{-2}{N}\sum_{i=1}^{N} y_i$. At each epoch, we update w and b using partial derivatives. The **learning rate** α controls the size of an update:

$$\begin{aligned} w &\leftarrow w - \alpha\frac{\partial l}{\partial w}; \\ b &\leftarrow b - \alpha\frac{\partial l}{\partial b}. \end{aligned} \tag{2}$$

We subtract (as opposed to adding) partial derivatives from the values of parameters because derivatives are indicators of growth of a function.

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

GRADIENT DESCENT: UPDATE_W_AND_B

The function that updates the parameters w and b during one epoch is shown below:

```
def update_w_and_b(spendings, sales, w, b, alpha):
    dl_dw = 0.0
    dl_db = 0.0
    N = len(spendings)

    for i in range(N):
        dl_dw += -2*spendings[i]*(sales[i] - (w*spendings[i] + b))
        dl_db += -2*(sales[i] - (w*spendings[i] + b))

    # update w and b
    w = w - (1/float(N))*dl_dw*alpha
    b = b - (1/float(N))*dl_db*alpha

    return w, b
```

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

MAIN LOOP: LOOP OVER MULTIPLE EPOCHS

The function that loops over multiple epochs is shown below:

```python
def train(spendings, sales, w, b, alpha, epochs):
    for e in range(epochs):
        w, b = update_w_and_b(spendings, sales, w, b, alpha)

        # log the progress
        if e % 400 == 0:
            print("epoch:", e, "loss: ", avg_loss(spendings, sales, w, b))

    return w, b
```

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

GRADIENT DESCENT: AVG_LOSS

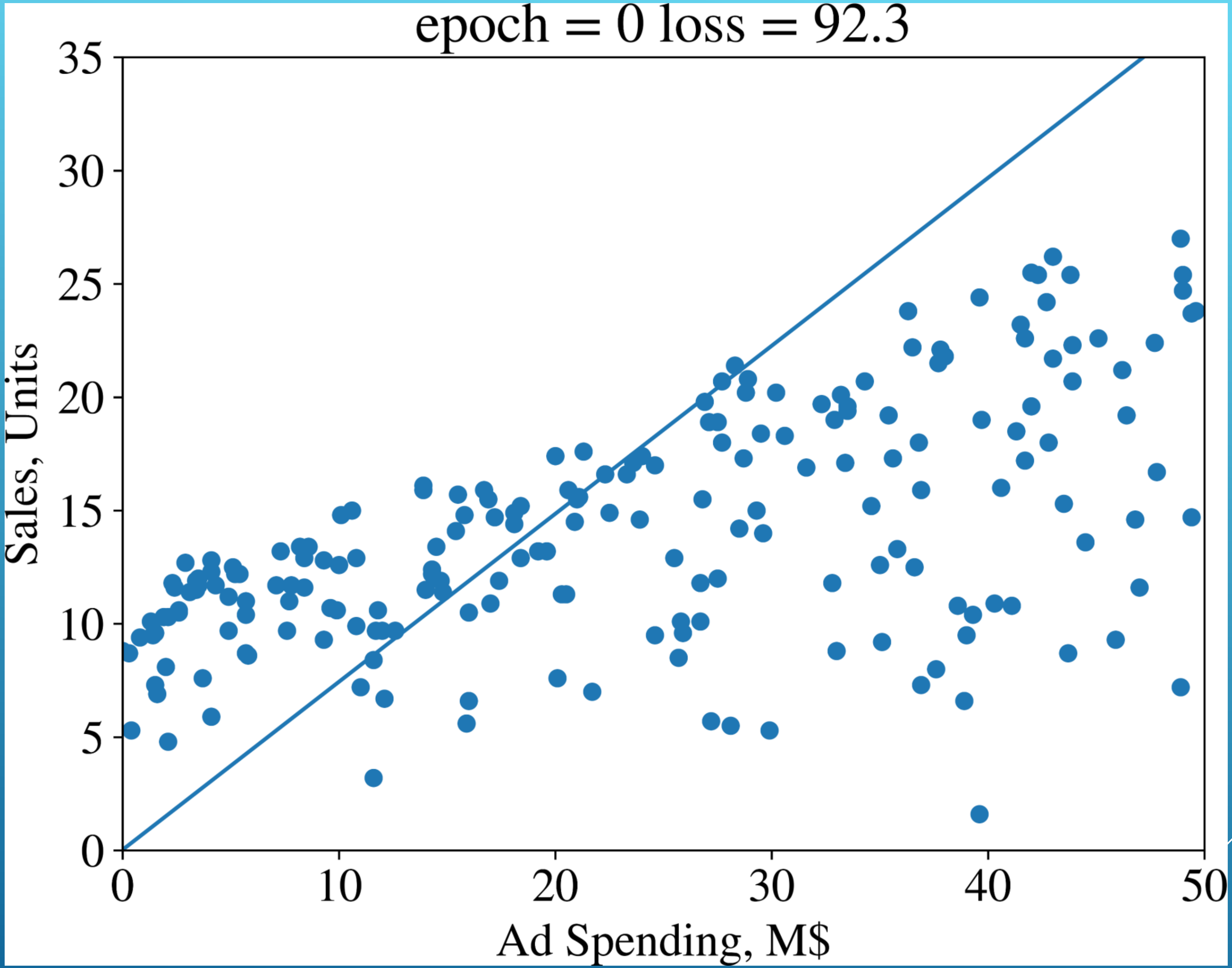
The function *avg_loss* in the above code snippet is a function that computes the mean squared error. It is defined as:

```
def avg_loss(spendings, sales, w, b):
    N = len(spendings)
    total_error = 0.0
    for i in range(N):
        total_error += (sales[i] - (w*spendings[i] + b))**2
    return total_error / float(N)
```

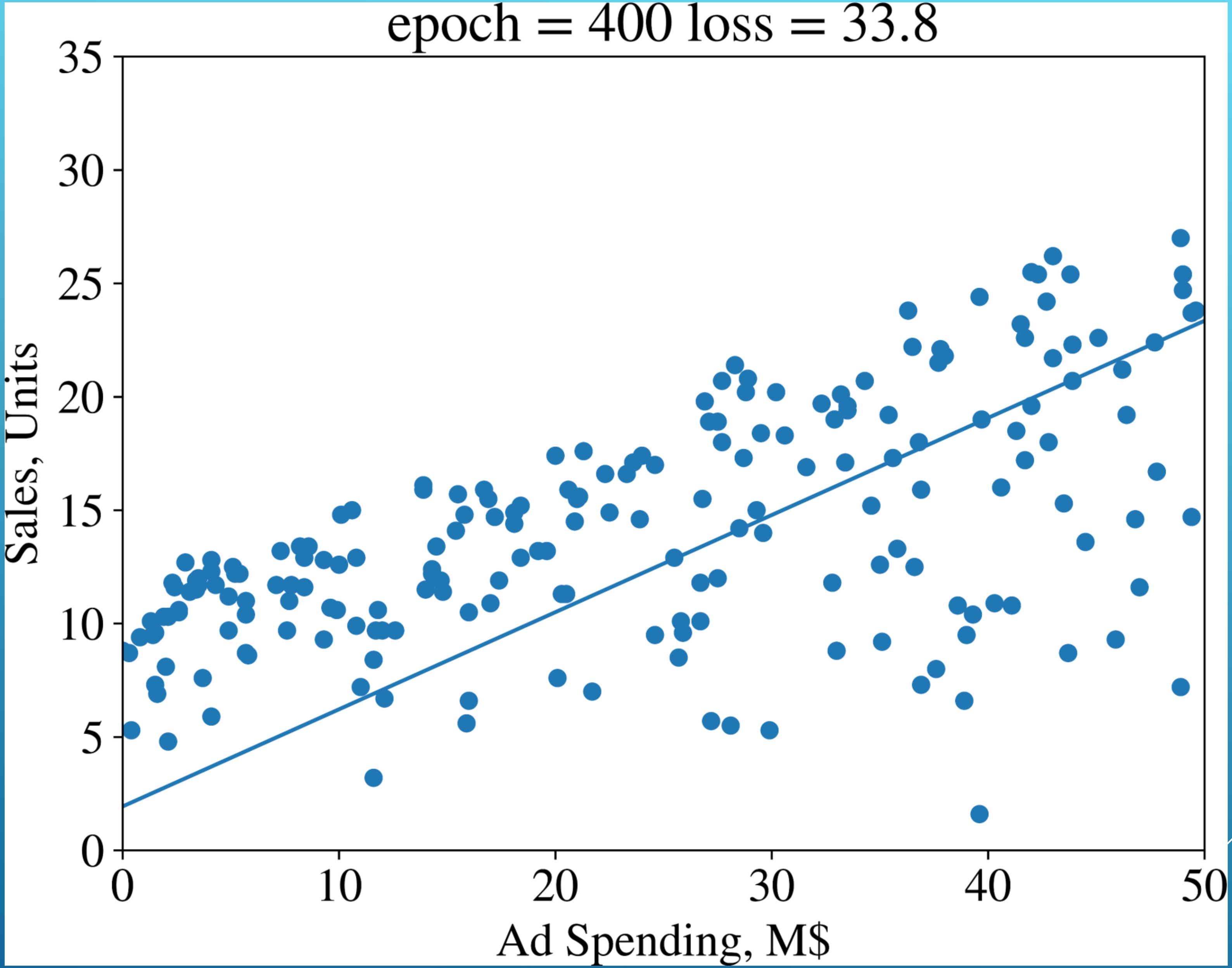
If we run the *train* function for $\alpha = 0.001$, $w = 0.0$, $b = 0.0$, and 15,000 epochs, we will see the following output (shown partially):

```
epoch:  0 loss: 92.32078294903626
epoch:  400 loss: 33.79131790081576
epoch:  800 loss: 27.9918542960729
epoch:  1200 loss: 24.33481690722147
epoch:  1600 loss: 22.028754937538633
...
epoch:  2800 loss: 19.07940244306619
```

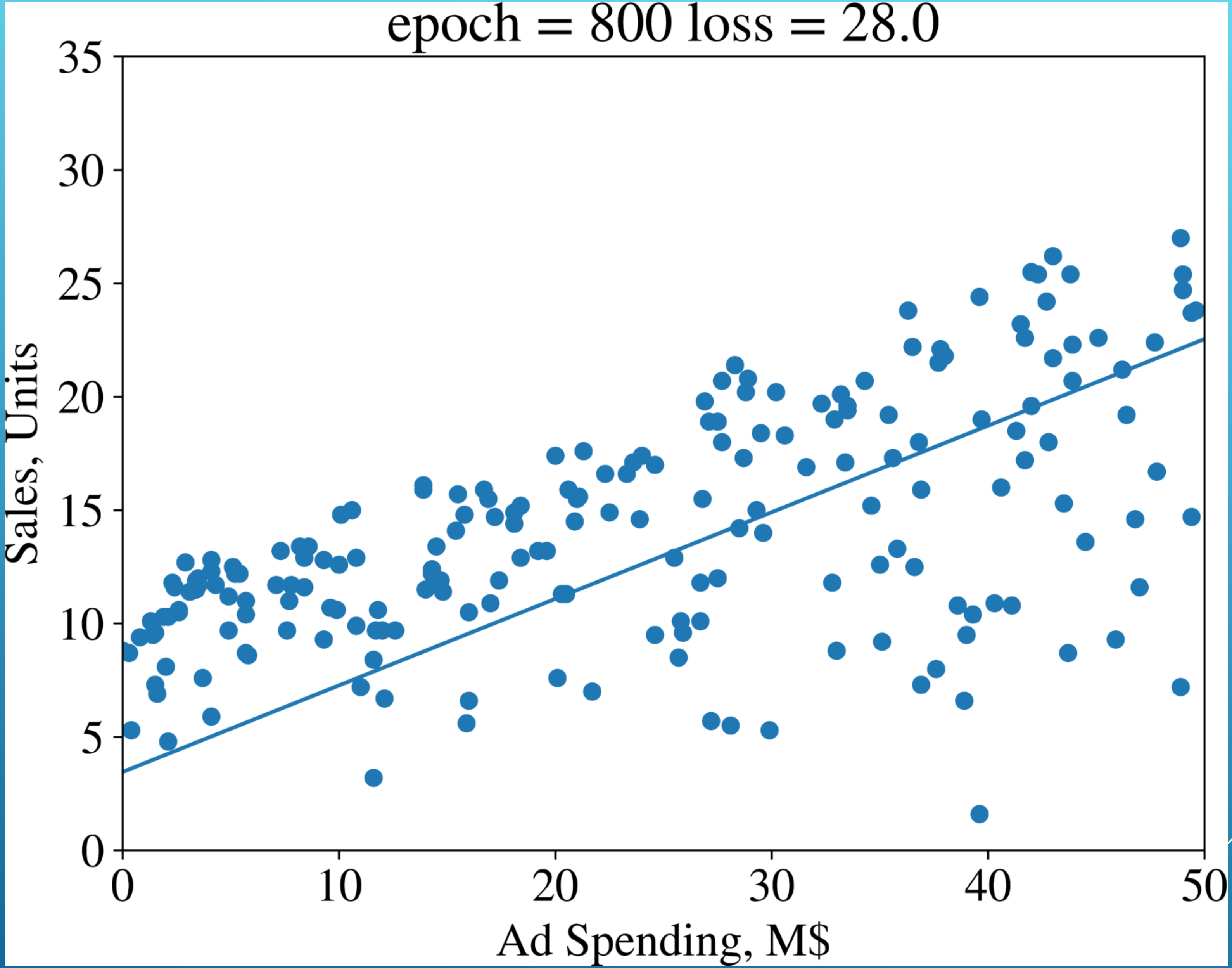
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



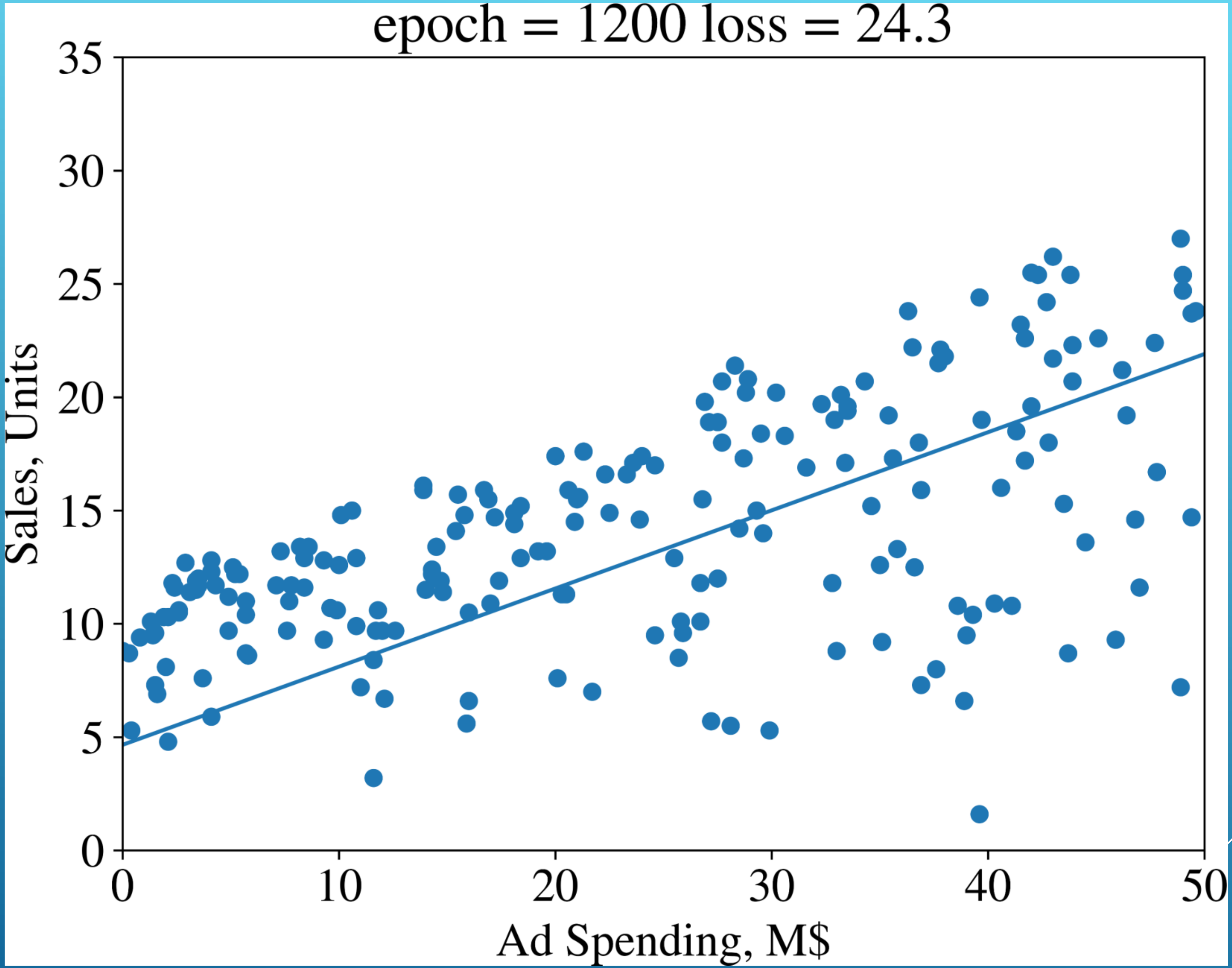
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



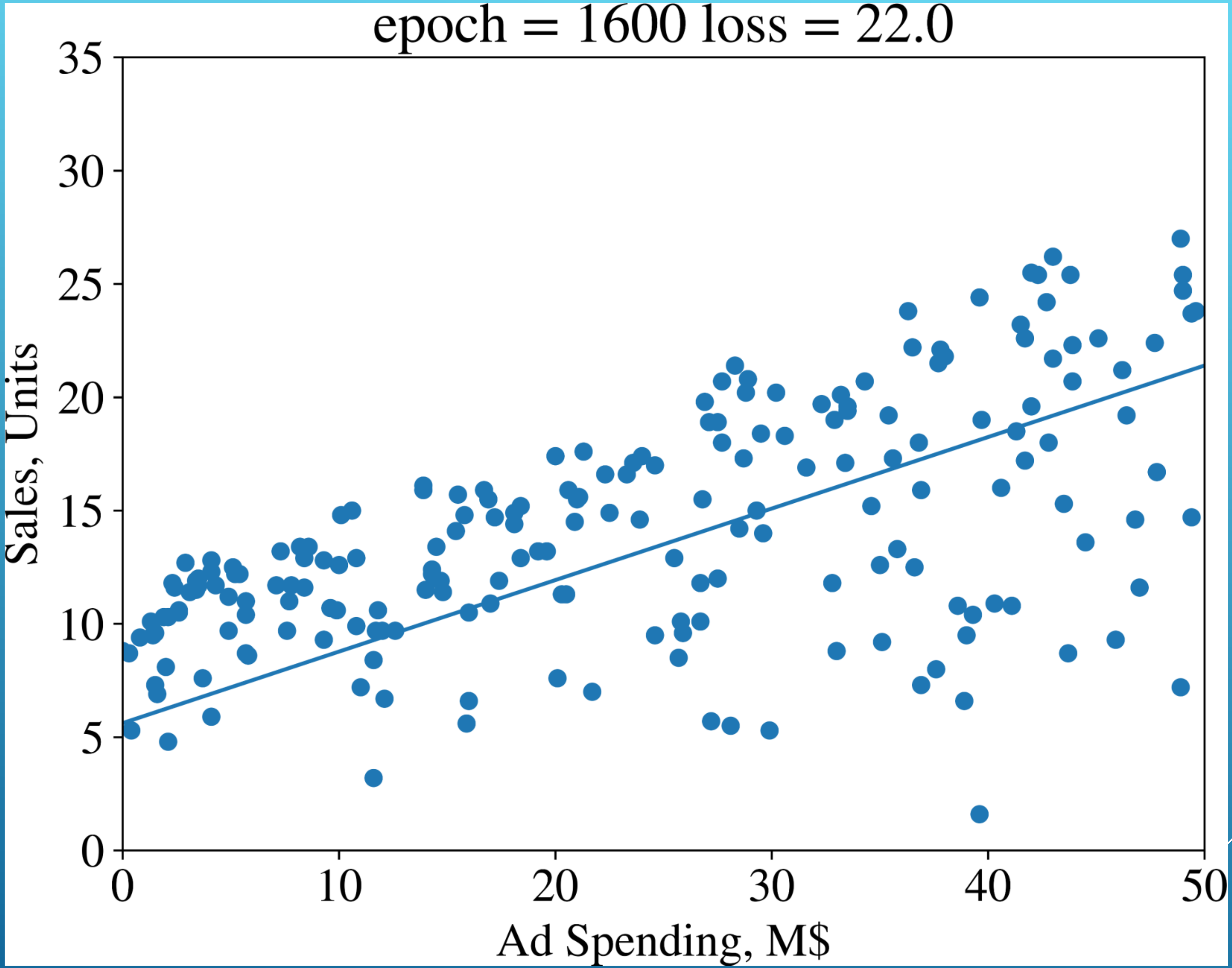
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



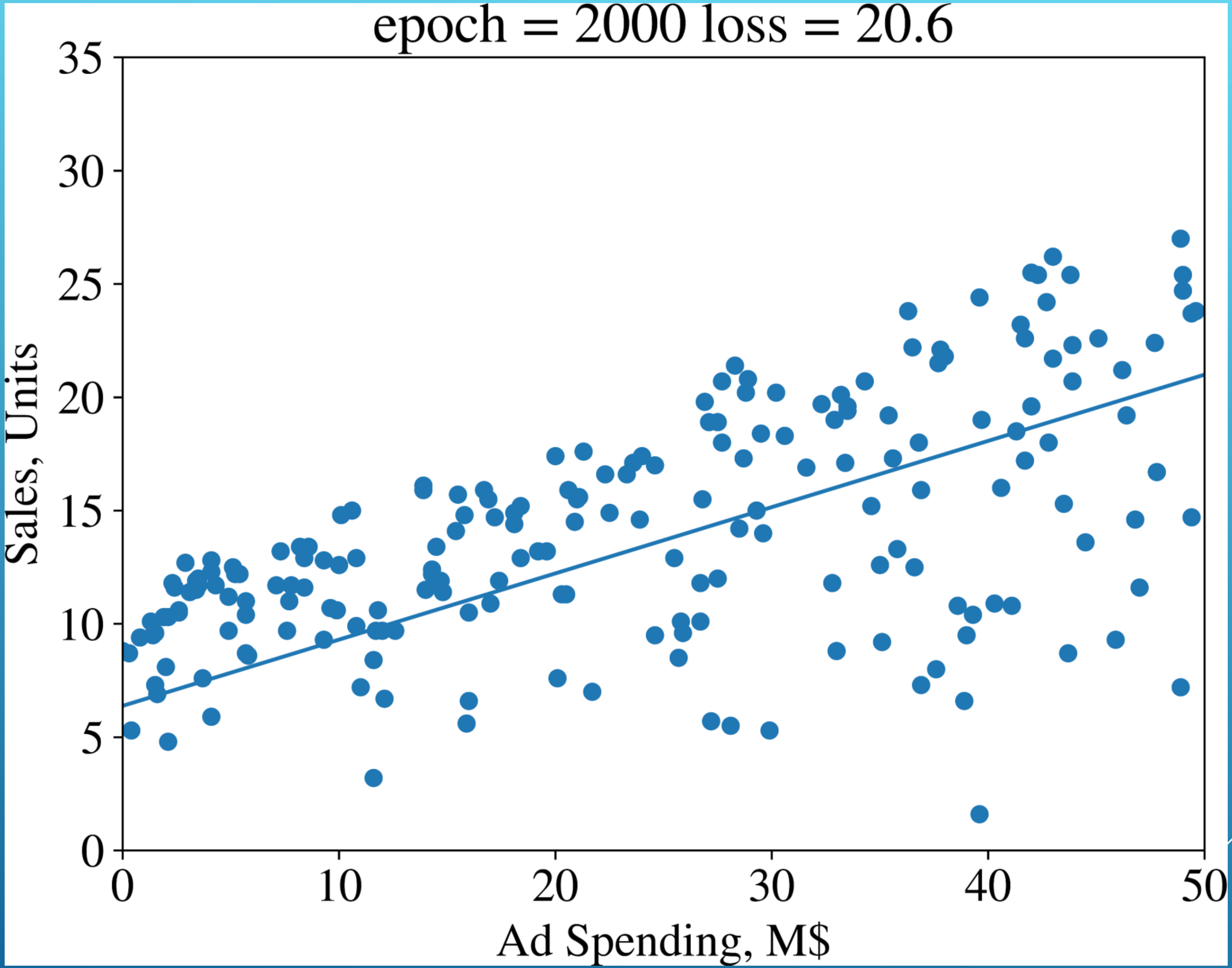
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



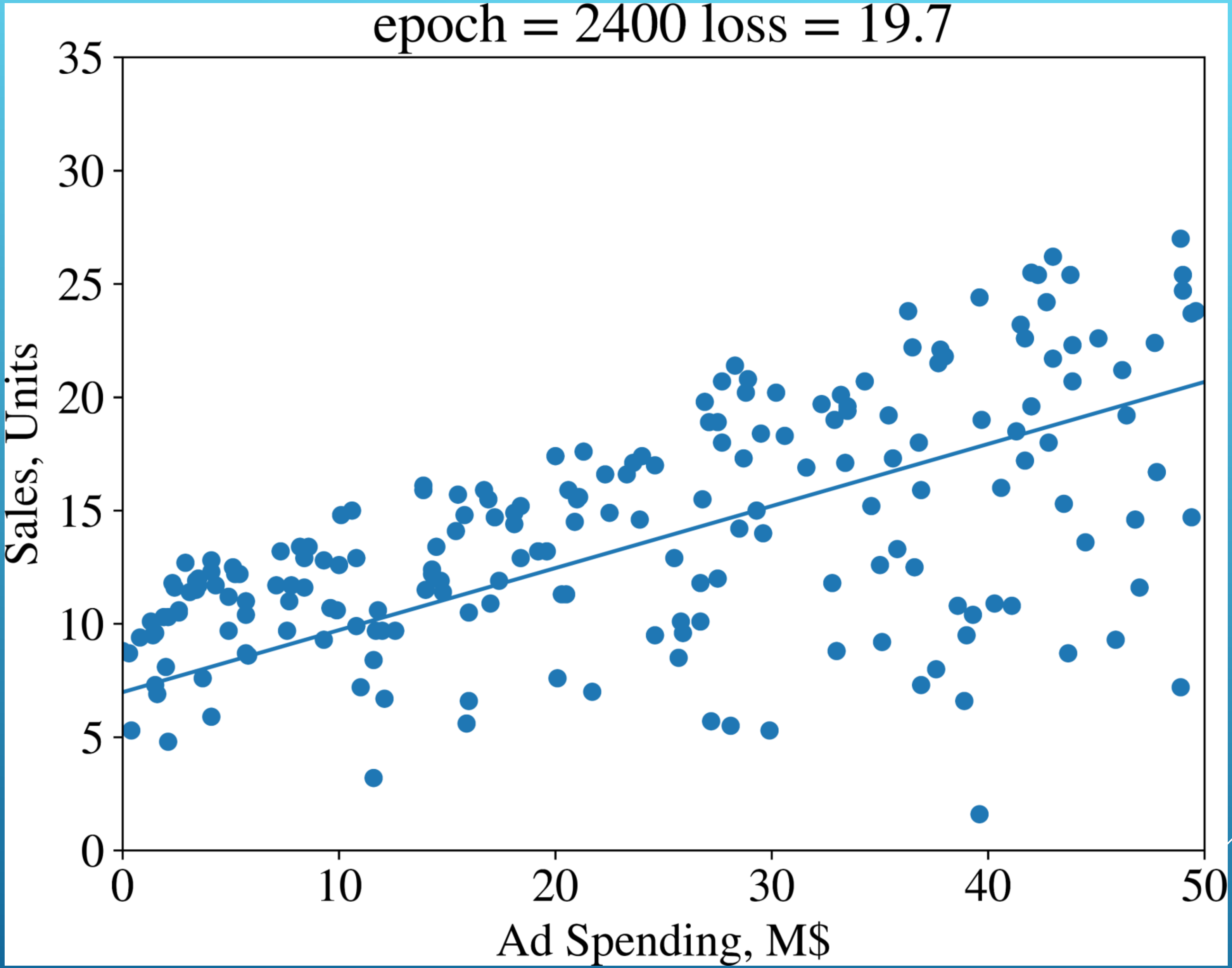
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



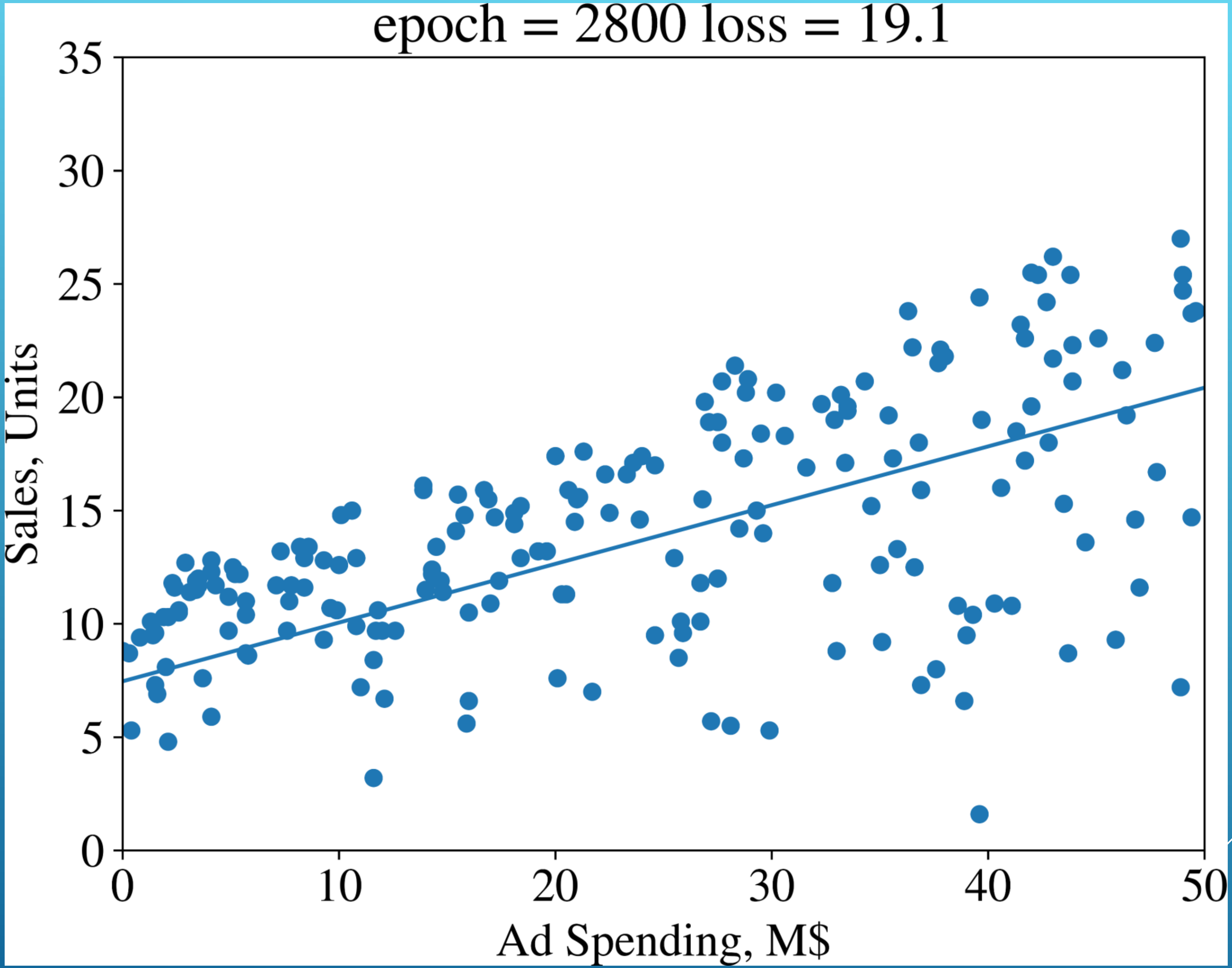
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



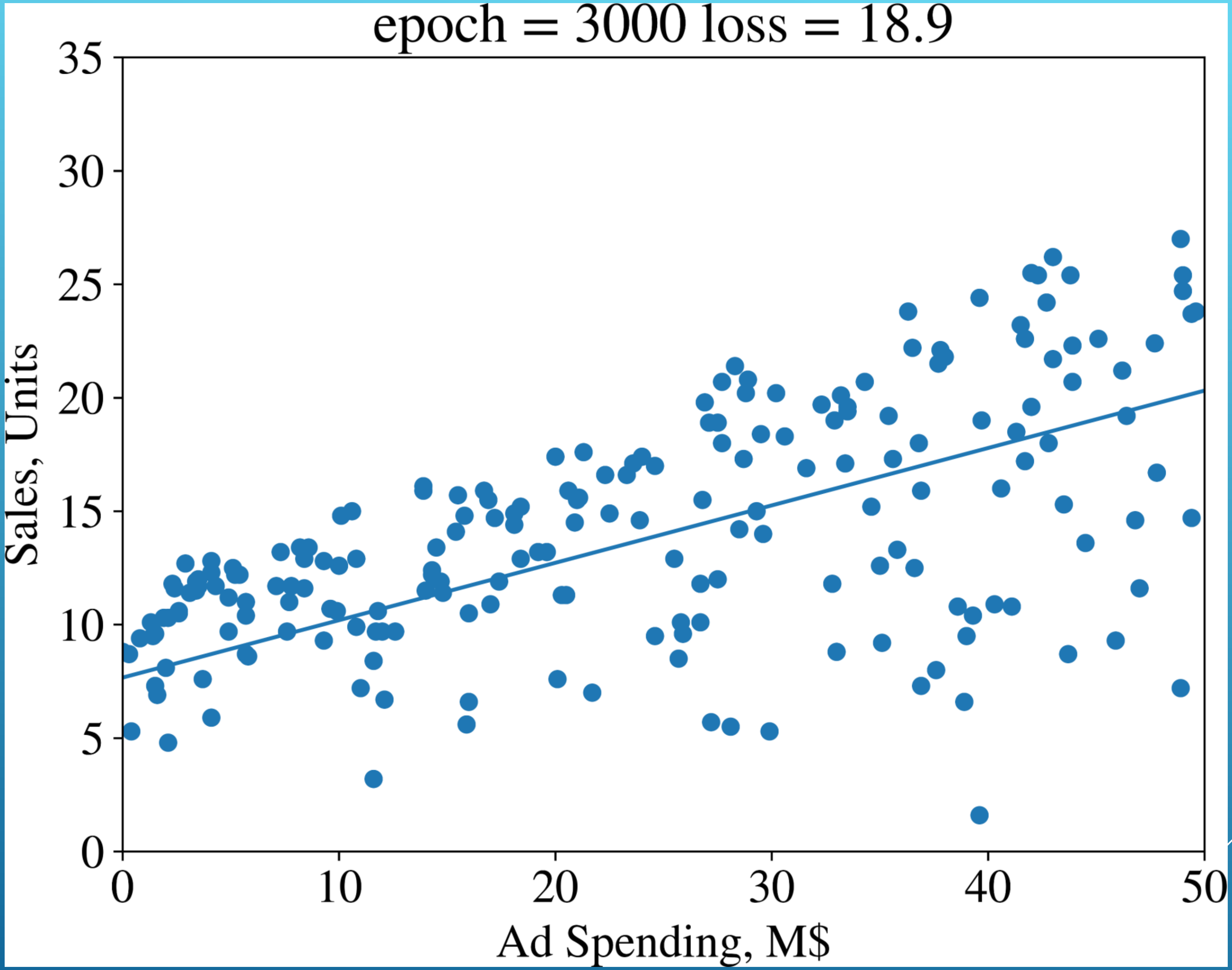
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



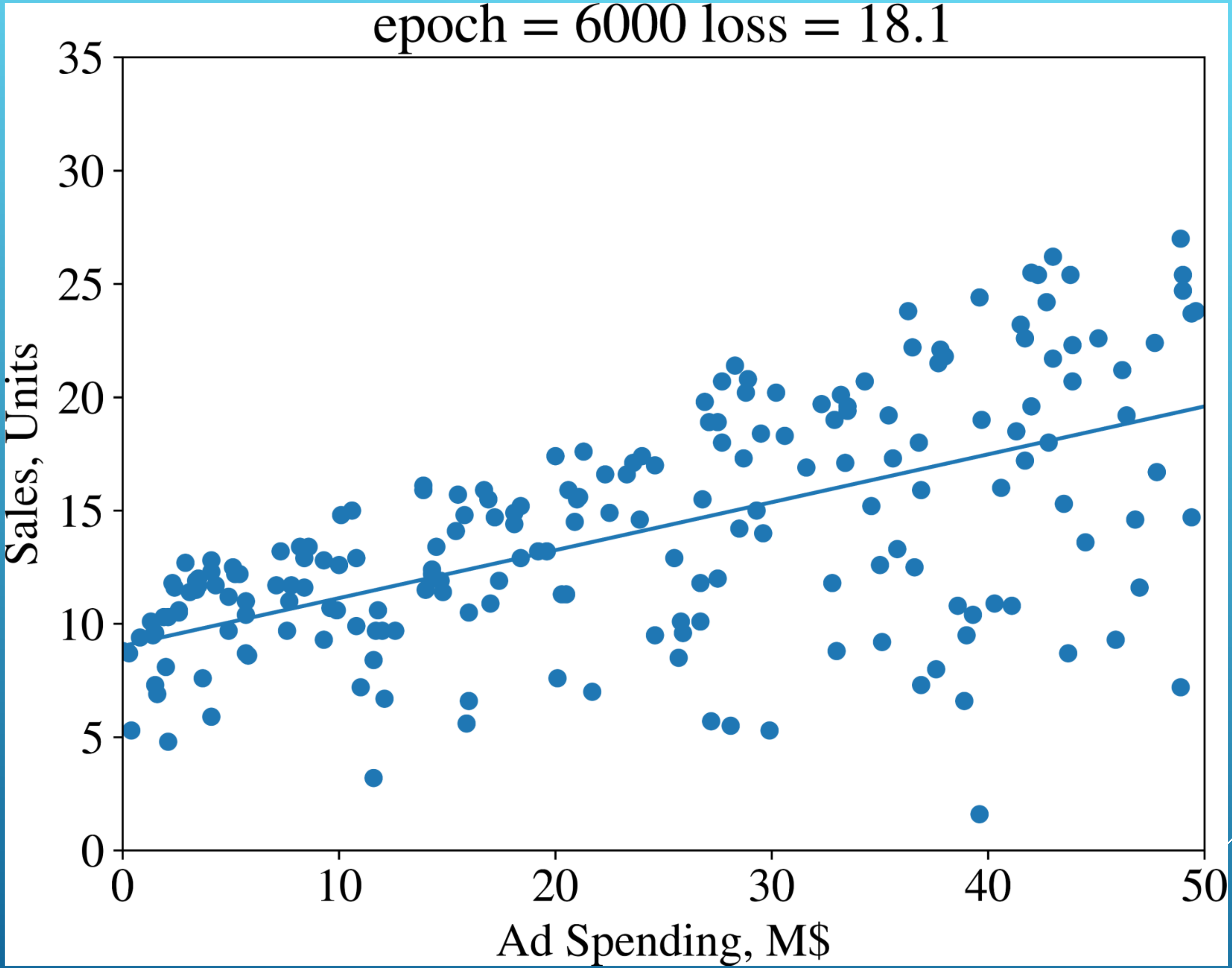
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



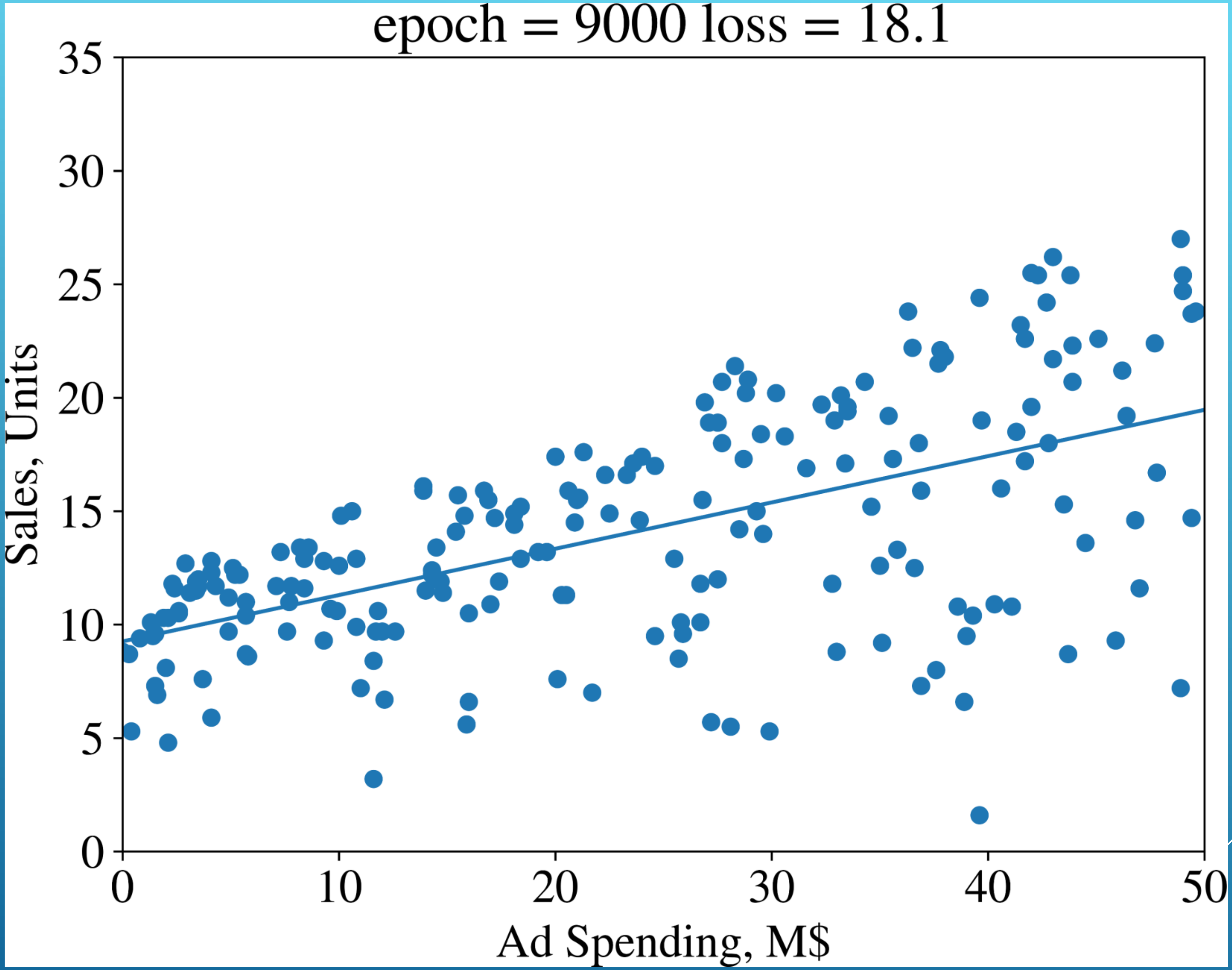
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



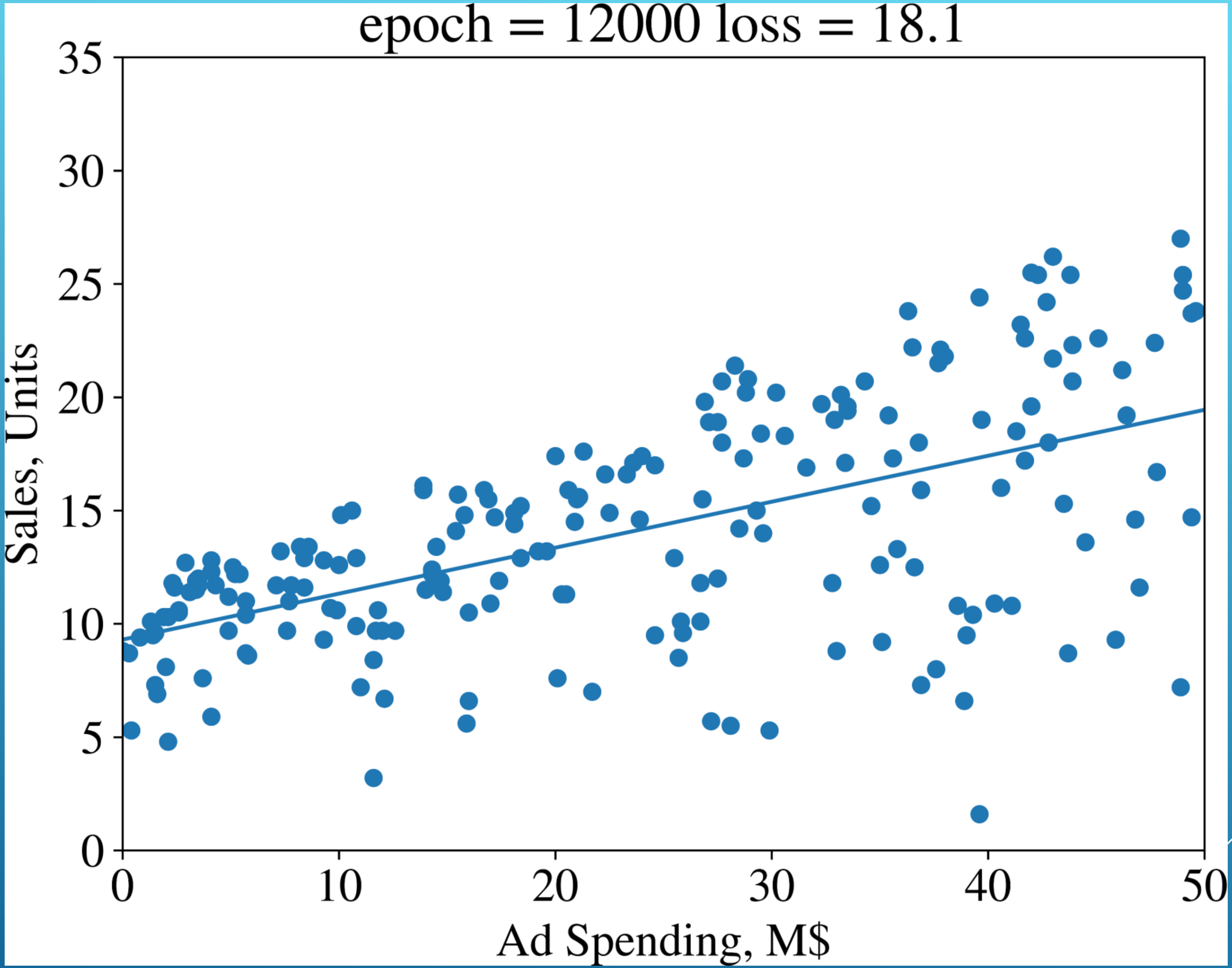
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



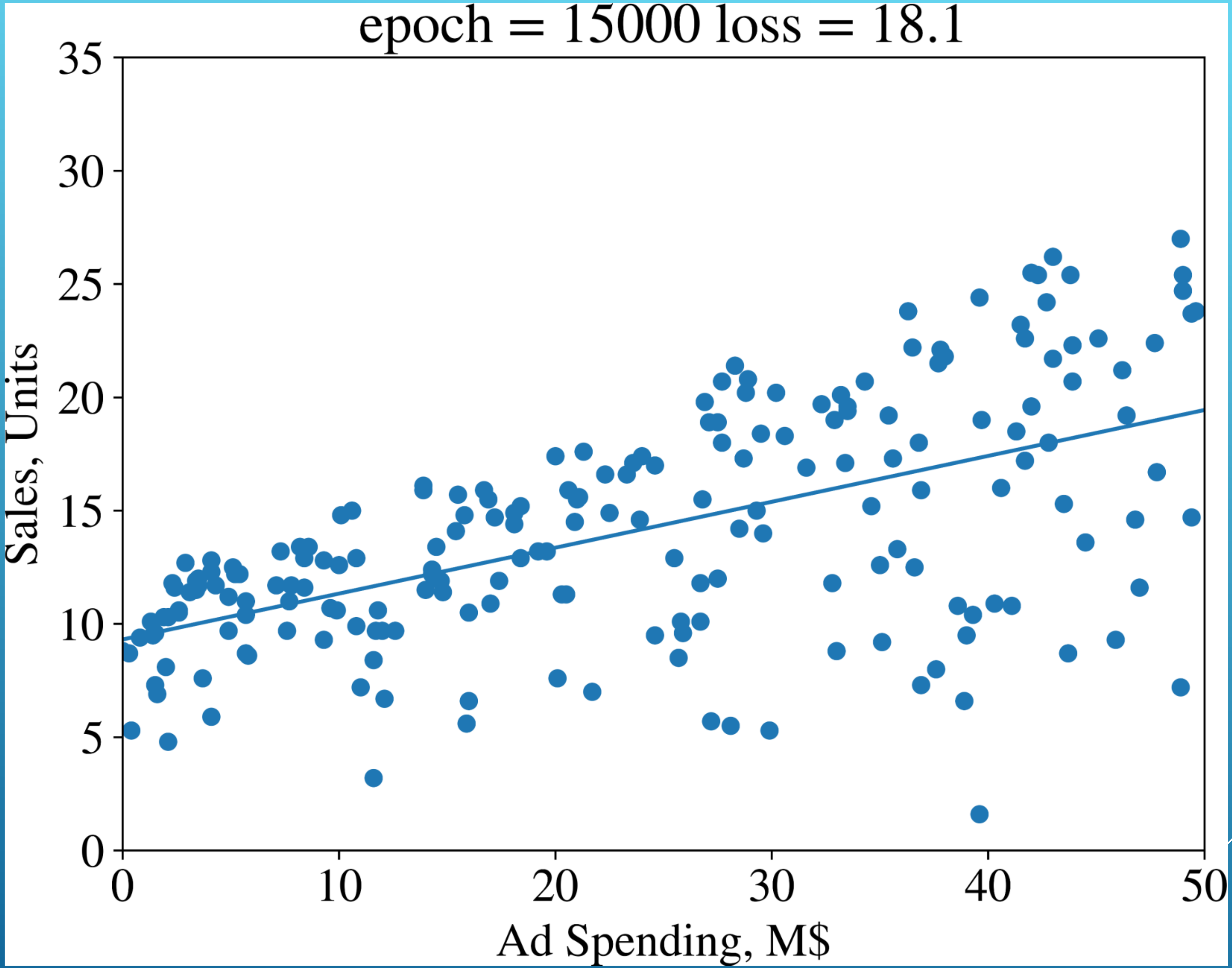
Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*



Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

GRADIENT DESCENT: PREDICT

Finally, once we have found the optimal values of parameters w and b, the only missing piece is a function that makes predictions:

```
def predict(x, w, b):
    return w*x + b
```

Try to execute the following code:

```
w, b = train(x, y, 0.0, 0.0, 0.001, 15000)
x_new = 23.0
y_new = predict(x_new, w, b)
print(y_new)
```

The output is 13.97.

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

- Representation:
  - Provides the model structure.
  - The parameters define the search space of possible models.
- Evaluation:
  - Defines the criteria for successful learning.
  - Gives a method to compare one model to another and decide which one is preferred.
- Optimization:
  - Specifies an algorithmic process to find an optimal model.

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

- Representation: **line**
- Evaluation: **minimize mean squared error**
- Optimization: **gradient descent**

Credit: The Hundred-Page Machine Learning Book, *A. Burkov, 2019*

LEARNING = REPRESENTATION + EVALUATION + OPTIMIZATION

Table 1. The three components of learning algorithms.

Representation	Evaluation	Optimization
Instances	Accuracy/Error rate	Combinatorial optimization
K-nearest neighbor	Precision and recall	Greedy search
Support vector machines	Squared error	Beam search
Hyperplanes	Likelihood	Branch-and-bound
Naive Bayes	Posterior probability	Continuous optimization
Logistic regression	Information gain	Unconstrained
Decision trees	K-L divergence	Gradient descent
Sets of rules	Cost/Utility	Conjugate gradient
Propositional rules	Margin	Quasi-Newton methods
Logic programs		Constrained
Neural networks		Linear programming
Graphical models		Quadratic programming
Bayesian networks		
Conditional random fields		

Credit: "A Few Useful Things to Know About Machine Learning", *P. Domingos, C. of the ACM, Oct 2012, Vol. 55, No. 10*

ANATOMY OF A LEARNING ALGORITHM

Scott O'Hara
Metrowest Developers Machine Learning Group
12/16/2020