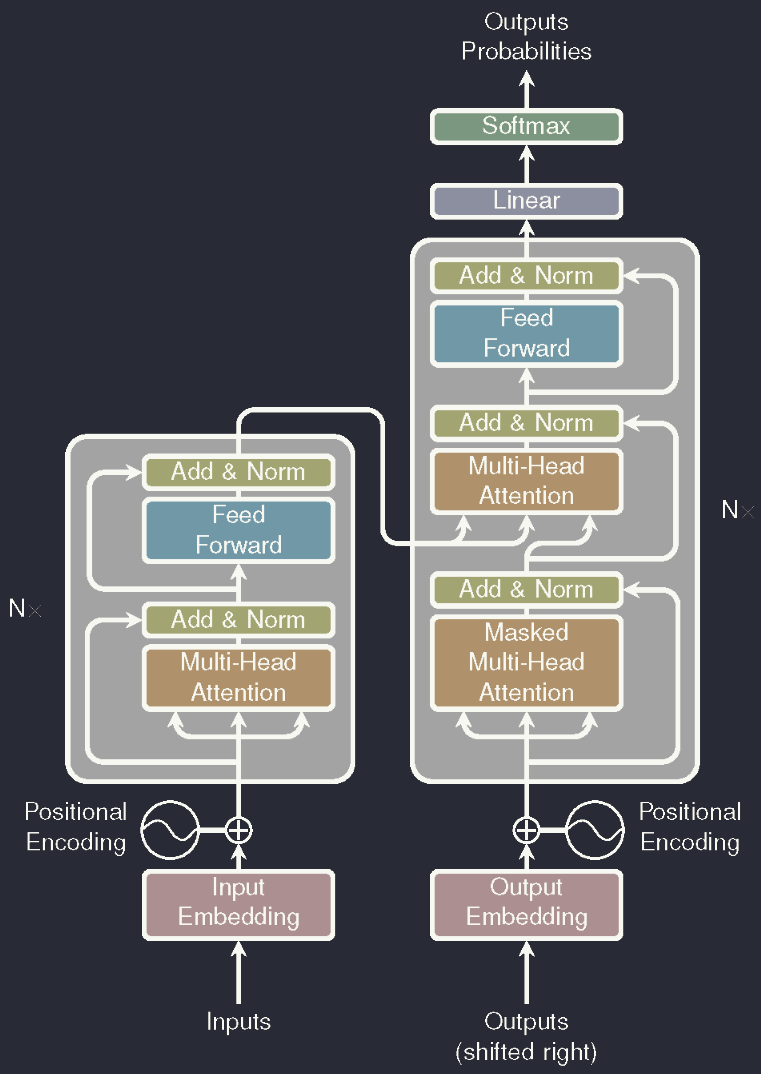

Figure 1: "The Transformer" Recreated in TikZ from "Attention Is All You Need"

Machine Learning Notes (Book 2)

"An Introduction To Statistical Learning"

Author

Paul Beggs
PaulBeggs03@gmail.com

TABLE OF CONTENTS

CHAPTER 1

INTRODUCTION

1.1 Notation and Simple Matrix Algebra

The following text is taken directly from the book:

Choosing notation for a textbook is always a difficult task. For the most part, we adopt the same notational conventions as ESL.

We will use n to represent the number of distinct data points, or observations, in our sample. We will let p denote the number of variables that are available for use in making predictions.

For example, the `wage` data set consists of 11 variables for 3,000 people, so we have $n = 3000$ observations and $p = 11$ variables (such as `year`, `age`, `race`, and more).

Note that throughout this book, we indicate variable names using colored font: `Variable Name`.

In some examples, p might be quite large, such as on the order of thousands or even millions; this situation arises quite often, for example, in the analysis of modern biological data or web-based advertising data.

1.1.1 Notation

In general, we let x_{ij} represent the value of the jth variable for the ith observation, where $i = 1, 2, \ldots, n$ and $j = 1, 2, \ldots, p$.

Throughout this book, i will be used to index the samples or observations (from 1 to n) and j will be used to index the variables (from 1 to p).

We let $\mathbf{X}$ denote an $n \times p$ matrix whose (i, j)th element is x_{ij}. That is,

$$\mathbf{X} = \begin{pmatrix} x_{11} & x_{12} & \cdots & x_{1p} \\ x_{21} & x_{22} & \cdots & x_{2p} \\ \vdots & \vdots & \ddots & \vdots \\ x_{n1} & x_{n2} & \cdots & x_{np} \end{pmatrix}.$$

For readers who are unfamiliar with matrices, it is useful to visualize $\mathbf{X}$ as a spreadsheet of numbers with n rows and p columns.

1.1.2 Row and Column Vectors

At times we will be interested in the rows of $\mathbf{X}$, which we write as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_n$. Here $\mathbf{x}_i$ is a vector of length p, containing the p variable measurements for the ith observation. That is,

$$\mathbf{x}_i = \begin{pmatrix} x_{i1} \\ x_{i2} \\ \vdots \\ x_{ip} \end{pmatrix}. \tag{1.1}$$

For example, for the `wage` data, $\mathbf{x}_i$ is a vector of length 11, consisting of year, age, race, and other values for the ith individual.

At other times we will instead be interested in the columns of $\mathbf{X}$, which we write as $\mathbf{x}_1, \mathbf{x}_2, \ldots, \mathbf{x}_p$. Each is a vector of length n. That is,

$$\mathbf{x}^{(j)} = \begin{pmatrix} x_{1j} \\ x_{2j} \\ \vdots \\ x_{nj} \end{pmatrix}.$$

For example, for the `wage` data, $\mathbf{x}_1$ contains the $n = 3000$ values for year.

Using this notation, the matrix $\mathbf{X}$ can also be written as

$$\mathbf{X} = \begin{pmatrix} \mathbf{x}_1 & \mathbf{x}_2 & \cdots & \mathbf{x}_p \end{pmatrix}, \quad \text{or} \quad \mathbf{X} = \begin{pmatrix} \mathbf{x}_1^T \\ \mathbf{x}_2^T \\ \vdots \\ \mathbf{x}_n^T \end{pmatrix}.$$

Here, the superscript ^{T} denotes the transpose of a matrix or vector. So, for example,

$$\mathbf{X}^T = \begin{pmatrix} x_{11} & x_{21} & \cdots & x_{n1} \\ x_{12} & x_{22} & \cdots & x_{n2} \\ \vdots & \vdots & \ddots & \vdots \\ x_{1p} & x_{2p} & \cdots & x_{np} \end{pmatrix}, \quad \mathbf{x}_i^T = \begin{pmatrix} x_{i1} & x_{i2} & \cdots & x_{ip} \end{pmatrix}.$$

1.1.3 Output Variable

We use y_i to denote the ith observation of the variable on which we wish to make predictions, such as `wage`. Hence, we write the set of all n observations in vector form as

$$\mathbf{y} = \begin{pmatrix} y_1 \\ y_2 \\ \vdots \\ y_n \end{pmatrix}.$$

Then our observed data consists of the pairs

$$\{(\mathbf{x}_1, y_1), (\mathbf{x}_2, y_2), \ldots, (\mathbf{x}_n, y_n)\},$$

where each $\mathbf{x}_i$ is a vector of length p. (If $p = 1$, then $\mathbf{x}_i$ is simply a scalar.)

1.1.4 Vector and Matrix Notation

- A vector of length n is always denoted in lower-case bold font, e.g., $\mathbf{a} = [a_1, a_2, \ldots, a_n]^T$.
- Vectors not of length n (such as feature vectors of length p) are denoted in lower-case normal font, e.g., a.
- Scalars are denoted in lower-case normal font, e.g., a.
- Matrices are denoted using bold capitals, such as $\mathbf{A}$.
- Random variables are denoted using capital normal font, e.g., A, regardless of their dimensions.

In rare cases where the use of lower-case normal font leads to ambiguity, we will clarify the intended use.

1.1.5 Dimensions and Spaces

To indicate the dimension of a particular object:

- Scalar: $a \in \mathbb{R}$
- Vector of length k: $a \in \mathbb{R}^k$
- Vector of length n: $a \in \mathbb{R}^n$
- Matrix of size $r \times s$: $\mathbf{A} \in \mathbb{R}^{r \times s}$

1.1.6 Matrix Multiplication

We have avoided using matrix algebra whenever possible. However, in a few instances it becomes too cumbersome to avoid it entirely. In these rare instances, it is important to understand the concept of multiplying two matrices. Suppose that $\mathbf{A} \in \mathbb{R}^{r \times d}$ and $\mathbf{B} \in \mathbb{R}^{d \times s}$. Then the product $\mathbf{AB}$ is an $r \times s$ matrix, where the (i, j)th element is computed as

$$(\mathbf{AB})_{ij} = \sum_{k=1}^{d} a_{ik} b_{kj}.$$

As an example, consider

$$\mathbf{A} = \begin{pmatrix} 1 & 2 \\ 3 & 4 \end{pmatrix}, \quad \mathbf{B} = \begin{pmatrix} 5 & 6 \\ 7 & 8 \end{pmatrix}.$$

Then,

$$\begin{aligned}\mathbf{AB} &= \begin{pmatrix} 1\cdot 5 + 2\cdot 7 & 1\cdot 6 + 2\cdot 8 \\ 3\cdot 5 + 4\cdot 7 & 3\cdot 6 + 4\cdot 8 \end{pmatrix} \\ &= \begin{pmatrix} 19 & 22 \\ 43 & 50 \end{pmatrix}.\end{aligned}$$

Note that this operation produces an $r \times s$ matrix. It is only possible to compute the product $\mathbf{AB}$ if the number of columns in $\mathbf{A}$ (which is d) is equal to the number of rows in $\mathbf{B}$ (which is also d).

1.2 Organization of the Book

Chapter 2 introduces the basic terminology and concepts behind statistical learning. This chapter also presents the K-nearest neighbor classifier, a very simple method that works surprisingly well on many problems. Chapters 3 and 4 cover classical linear methods for regression and classification. In particular, Chapter 3 reviews linear regression, the fundamental starting point for all regression methods. In Chapter 4 we discuss two of the most important classical classification methods, logistic regression and linear discriminant analysis.

A central problem in all statistical learning situations involves choosing the best method for a given application. Hence, in Chapter 5 we introduce cross-validation and the bootstrap, which can be used to estimate the accuracy of a number of different methods in order to choose the best one.

Much of the recent research in statistical learning has concentrated on non-linear methods. However, linear methods often have advantages over their non-linear competitors in terms of interpretability and sometimes also accuracy. Hence, in Chapter 6 we consider a host of linear methods, both classical and more modern, which offer potential improvements over standard linear regression. These include stepwise selection, ridge regression, principal components regression, and the lasso.

The remaining chapters move into the world of non-linear statistical learning. We first introduce in Chapter 7 a number of non-linear methods that work well for problems with a single input variable. We then show how these methods can be used to fit non-linear additive models for which there is more than one input. In Chapter 8, we investigate tree-based methods, including bagging, boosting, and random forests. Support vector machines, a set of approaches for performing both linear and non-linear classification, are discussed in Chapter 9. We cover deep learning, an approach for non-linear regression and classification that has received a lot of attention in recent years, in Chapter 10. Chapter 11 explores survival analysis, a regression approach that is specialized to the setting in which the output variable is censored, i.e. not fully observed.

In Chapter 12, we consider the unsupervised setting in which we have input variables but no output variable. In particular, we present principal components analysis, K-means clustering, and hierarchical clustering. Finally, in Chapter 13 we cover the very important topic of multiple hypothesis testing.

At the end of each chapter, we present one or more Python lab sections in which we systematically work through applications of the various methods discussed in that chapter. These labs demonstrate the strengths and weaknesses of the various approaches, and also provide a useful reference for the syntax required to implement the various methods. The reader may choose to work through the labs at their own pace, or the labs may be the focus of group sessions as part of a classroom environment. Within each Python lab, we present the results that we obtained when we performed the lab at the time of writing this book. However, new versions of Python are continuously released, and over time, the packages called in the labs will be updated. Therefore, in the future, it is possible that the results shown in the lab sections may no longer correspond precisely to the results obtained by the reader who performs the labs. As necessary, we will post updates to the labs on the book website.

Name	**Description**
`Auto`	Gas mileage, horsepower, and other information for cars.
`Bikeshare`	Hourly usage of a bike sharing program in Washington, DC.
`Boston`	Housing values and other information about Boston census tracts.
`BrainCancer`	Survival times for patients diagnosed with brain cancer.
`Caravan`	Information about individuals offered caravan insurance.
`Carseats`	Information about car seat sales in 400 stores.
`College`	Demographic characteristics, tuition, and more for USA colleges.
`Credit`	Information about credit card debt for 400 customers.
`Default`	Customer default records for a credit card company.
`Fund`	Returns of 2,000 hedge fund managers over 50 months.
`Hitters`	Records and salaries for baseball players.
`Khan`	Gene expression measurements for four cancer types.
`NCI60`	Gene expression measurements for 64 cancer cell lines.
`NYSE`	Returns, volatility, and volume for the New York Stock Exchange.
`OJ`	Sales information for Citrus Hill and Minute Maid orange juice.
`Portfolio`	Past values of financial assets, for use in portfolio allocation.
`Publication`	Time to publication for 244 clinical trials.
`Smarket`	Daily percentage returns for S&P 500 over a 5-year period.
`USArrests`	Crime statistics per 100,000 residents in 50 states of USA.
`Wage`	Income survey data for men in central Atlantic region of USA.
`Weekly`	1,089 weekly stock market returns for 21 years.

Table 1.1: A list of data sets needed to perform the labs and exercises in this textbook. All data sets are available in the `ISLR` package, with the exception of `USArrests`, which is part of the `R` distribution, but accessible from `Python`.