Beyond Linear Regression

From Logistic Regression to Neural Networks

Motivating Examples

Suppose that we want to predict whether or not a student can pass an exam.

Here, the dependent variable, Y, is the outcome of the exam (1 = pass, 0 = failure).

The independent variables X are, for example, hours studied for the example, in class participation, background, …

Binary Variables

Binary variables only take two values (e.g., 0 and 1). Here are some examples:

Example outcome: pass vs. fail
Trial result: guilty vs. not guilty
Football result: win vs. lose
Medical testing: positive vs. negative
Gender detection: male vs. female

Logistic Regression

Idea

Recall what you did in linear regression…

You try to find out values of a and b that minimize the "square of errors", i.e., you want to minimize

$$\sum_i (Y_i - a - bX_i)^2$$

For all data points (X_i, Y_i).

Idea

But least square does not work well here… We want to define a and b in other ways. Note that we seek to find out a and b such that

When $Y_i = 1$, $f(X_i) = \Pr[X_i = 1]$ is as large as possible --- this means your prediction is accurate.

When $Y_i = 0$, $f(X_i)$ is as small as possible.

Idea

Let us translate the above reasoning into the following mathematics. For each observation (X_i, Y_i), we hope the following is as large as possible:

$$Y_i f(X_i) + (1 - Y_i)\big(1 - f(X_i)\big)$$

.

$$\text{Likelihood} = Y_i f(X_i) + (1 - Y_i)\big(1 - f(X_i)\big)$$

.

If $Y_i = 1$: The above likelihood is equal to $f(X_i)$, which we want to maximize. So we want to maximize the likelihood.

If $Y_i = 0$: The above likelihood is equal to $1 - f(X_i)$. We want to minimize $f(X_i)$ and again we want to maximize the likelihood.

In either case, we want the likelihood to be as large as possible.

Idea

Let us translate the above reasoning into the following mathematics. For each observation (X_i, Y_i), we hope the following is as large as possible:

$$Y_i f(X_i) + (1 - Y_i)\big(1 - f(X_i)\big)$$

.

And for all observations, we want to maximize the following:

$$\Pi_i Y_i f(X_i) + (1 - Y_i)\big(1 - f(X_i)\big)$$

Idea

This is called "maximum likelihood" --- we want to find out values of a and b that maximize the likelihood that was defined previously.

It is not easy to estimate the values of a and b that maximize the likelihood, and we skip the details here. However, if you are interested, you can check here.

Logistic Regression in R

Let us use an online database on university admission.

```
mydata <-
read.csv("https://stats.idre.ucla.edu/stat/da
ta/binary.csv")
head(mydata)
```

Logistic Regression in R

Let us use an online database on university admission.

```
mydata <-
read.csv("https://stats.idre.ucla.edu/stat/data/binary.csv")
head(mydata)
```

We have variables admit (binary), GRE, GPA, and rank.

Logistic Regression in R

Next, we regress admission on GRE, GPA and rank to see how these factors affect the admission decision, using logistic regression:

```r
logit <- glm(admit ~ gre + gpa + rank, data = mydata, family = "binomial")
summary(logit)
```

Logistic Regression in R

```
Coefficients:
            Estimate Std. Error z value Pr(>|z|)
(Intercept) -3.449548   1.132846  -3.045  0.00233 **
gre          0.002294   0.001092   2.101  0.03564 *
gpa          0.777014   0.327484   2.373  0.01766 *
rank        -0.560031   0.127137  -4.405 1.06e-05 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Logistic Regression in R

This result means that the best logistic model (i.e., the one with maximum likelihood) is

$$\text{Prob}(\text{admit} = 1) = \frac{1}{1 + \exp(-z)}$$

$$z = -3.44 + 0.0023\text{GRE} + 0.777\text{GPA} - 0.56\text{Rank}$$

And all independent variables are significant at 5% level.

Logistic Regression in R

$$\text{Prob}(\text{admit} = 1) = \frac{1}{1 + \exp(-z)}$$

$$z = -3.44 + 0.0023\text{GRE} + 0.777\text{GPA} - 0.56\text{Rank}$$

This result means: A higher GRE or GPA helps you get admitted, while a larger Rank hurts (e.g., rank 2 is larger and worse than rank 1), which is intuitive.

Logistic Regression in R

	Dependent variable:
	admit
gre	0.002**
	(0.001)
gpa	0.777**
	(0.327)
rank	-0.560***
	(0.127)
Constant	-3.450***
	(1.133)
Observations	400
Log Likelihood	-229.721
Akaike Inf. Crit.	467.442
Note:	*p<0.1; **p<0.05; ***p<0.01

We can organize the logistic regression output using the stargazer package. The details can be found in the "Introduction to R" lecture.

```
getwd()
library(stargazer)
stargazer(logit, out = "out.html", type = "html")
```
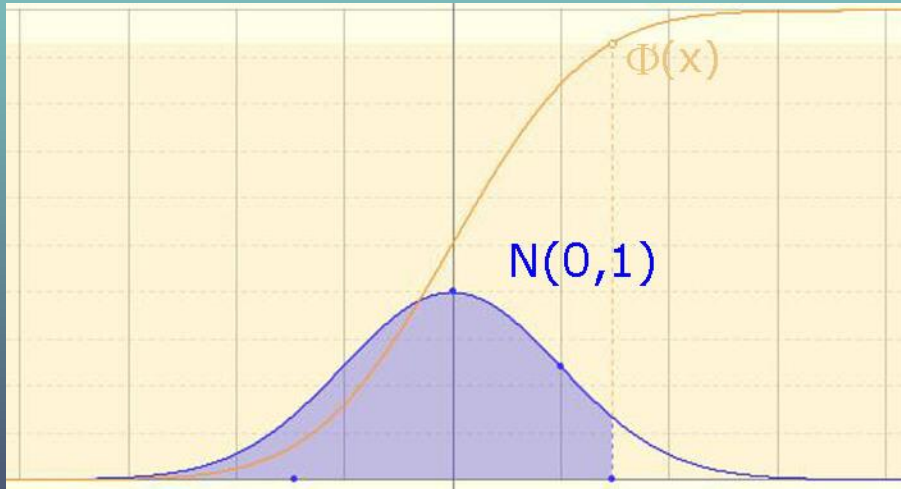
Probit Regression

In logistic regression, we adopt the logistic function to estimate $\Pr[Y = 1|X]$, which satisfies the properties that we listed. However, the logistic function is not the only function that satisfies those properties. Now, we introduce another function that can also make predictions about binary outcomes.

Probit Regression

Here, we use the cumulative distribution function of the standard normal distribution. Mathematically, suppose that $v \sim N(0, 1)$ is a standard normal random variable, then we can define the cumulative distribution function Φ as

$$\Phi(Z) = \Pr[v \leq Z]$$

Probit Regression

The shape of the cumulative function is very similar to the logistic function that we have discussed previously.

Probit Regression

In Probit regression, we assume that the data satisfies the following model:

$$\Pr[Y = 1] = \Phi(a + b\,X)$$

And again, we want to find out the values of a and b.

Probit Regression in R

```r
probit <- glm(admit ~ gre + gpa + rank, data
= mydata, family = binomial(link = "probit"))
summary(probit)
stargazer(probit, out = "out.html", type =
"html")
```

Contrasting Logistic and Probit Regression

	Dependent variable:
	admit
gre	0.002**
	(0.001)
gpa	0.777**
	(0.327)
rank	-0.560***
	(0.127)
Constant	-3.450***
	(1.133)
Observations	400
Log Likelihood	-229.721
Akaike Inf. Crit.	467.442
Note:	*p<0.1; **p<0.05; ***p<0.01

	Dependent variable:
	admit
gre	0.001**
	(0.001)
gpa	0.464**
	(0.195)
rank	-0.332***
	(0.075)
Constant	-2.092***
	(0.672)
Observations	400
Log Likelihood	-229.740
Akaike Inf. Crit.	467.481
Note:	*p<0.1; **p<0.05; ***p<0.01

Probit vs. Logistic Regression

They are similar models that yield similar (though not identical) inferences.

Logistic regression is more popular in healthcare.
Probit regression is more popular in political science.

But in most situations, it does not matter which method you choose to go with. Working with either will be fine.

Problem

In the above discussion, we discuss how to proceed when our dependent variable is binary. Now consider the following scenario:

Suppose that you are analyzing high school students' choice. Each student chooses one of the following options: A general program, an academic program, or a vocational program.

Career Choices

General

Vocational

Academic

Multinomial Models in R

```r
require(foreign)
require(nnet)
require(ggplot2)
require(reshape2)

mydata <-
read.dta("https://stats.idre.ucla.edu/stat/data/hsbdem
o.dta")
```

Multinomial Models in R

```r
mydata$career <- relevel(mydata$prog, ref =
"academic")
```

Here, we set academic as a benchmark and compare other careers with it.

```r
result <- multinom(career ~ read + write + math +
science, data = mydata)
summary(result)
```

Multinomial Models in R

```
multinom(formula = career ~ read + write + math + science, data = mydata)

Coefficients:
         (Intercept)          read         write          math     science
general     4.393599  -0.05638993  -0.03242985  -0.09976556  0.09049715
vocation    8.701066  -0.05844241  -0.06060226  -0.12352885  0.05879027

Std. Errors:
         (Intercept)          read         write          math     science
general     1.437971   0.02806080   0.02792886   0.03302162  0.02913435
vocation    1.549432   0.03046132   0.02781245   0.03578399  0.02936508
```

Multinomial Models in R

	Dependent variable:	
	general	vocation
	(1)	(2)
read	-0.056**	-0.058*
	(0.028)	(0.030)
write	-0.032	-0.061**
	(0.028)	(0.028)
math	-0.100***	-0.124***
	(0.033)	(0.036)
science	0.090***	0.059**
	(0.029)	(0.029)
Constant	4.394***	8.701***
	(1.438)	(1.549)
Akaike Inf. Crit.	356.823	356.823
Note:	*p<0.1; **p<0.05; ***p<0.01	

Again, we can organize the result using R's stargazer package.

Multinomial Models in R

	Dependent variable:	
	general	vocation
	(1)	(2)
read	-0.056**	-0.058*
	(0.028)	(0.030)
write	-0.032	-0.061**
	(0.028)	(0.028)
math	-0.100***	-0.124***
	(0.033)	(0.036)
science	0.090***	0.059**
	(0.029)	(0.029)
Constant	4.394***	8.701***
	(1.438)	(1.549)
Akaike Inf. Crit.	356.823	356.823
Note:	*p<0.1; **p<0.05; ***p<0.01	

We can see that when a student's read score is higher, he or she is less likely to join general and vocation programs. In other words, he or she is more likely to join an academic program. Interestingly, when one's science score is higher, he or she is likely to join a general program.

Multinomial Models in R

	Dependent variable:	
	general (1)	vocation (2)
read	-0.057** (0.028)	-0.059* (0.031)
write	-0.031 (0.028)	-0.054* (0.028)
math	-0.101*** (0.033)	-0.125*** (0.036)
science	0.090*** (0.029)	0.059** (0.030)
factor(schtyp)private	-0.643 (0.539)	-1.781** (0.800)
Constant	4.529*** (1.457)	8.694*** (1.568)
Akaike Inf. Crit.	353.849	353.849
Note:	*p<0.1; **p<0.05; ***p<0.01	

Now we revise the model to include school type into the logistic regression. We can see that, compared with public school students, private school students are less likely to join vocation programs.

Other Applications of Multinomial Model

Multinomial models are commonly used for discrete choice problems.

Which brand of milk is the consumer going to buy?

Is the consumer going to buy an EV, a traditional car, or simply do not buy?

Which university to apply for?

Occupational choice…

Other Dependent Variables

So far, we have linear regression models and logistic/multinomial logit models. How to choose your models?

Y is a real number: linear regression

Y = A, B, C, … multinomial logit models.

Y = 0, 1, 2, 3: what should we choose?

$Y \geq 0$: what should we choose?

Other Dependent Variables

When considering decisions such as purchase quantity (i.e., number of bottle waters to purchase from the supermarket), your dependent variable is something like $Y = 0, 1, \dots$

In this case, you can use Poisson/Negative Binomial models.

Poisson Regression

In many scenarios our dependent variable is a nonnegative integer. Examples include:

The number of children in a family
The number of classes you take in a semester
The number of books a consumer purchases

Poisson Regression

Consider the following model: We have independent variables X and we want to use them to predict the dependent variable, Y. Here, the dependent variable is a nonnegative integer.

For example, given the consumer's age, gender, income, we want to predict how many bottles of milk the consumer will purchase in a week.

Poisson Regression (Optional Topics)

Idea: we assume that the dependent variables follows the Poisson distribution:

$$\Pr[Y_i = y | \lambda_i] = \frac{\exp(-\lambda_i)\lambda_i^y}{y!}$$

where $y!$ is the factorial function and

$$\lambda_i = \exp(\beta X)$$

We are looking for the parameter β.

Poisson Regression in R

```
mydata <-
read.csv("https://stats.idre.ucla.edu/stat/data/poisson_sim
.csv")
```

Here, prog refers to the type of the program: 1 for general program, 2 for academic program, and 3 for vocational program.

Poisson Regression in R

```
result = glm(num_awards ~ factor(prog) + math,
family="poisson", data=mydata)
summary(result)
```

Because prog is simply a notation and does not have any numerical meaning, we take it as a fixed effect, and run the Poisson regression.

Other Dependent Variables

There are also occasions in which your parameters are "censored". For example, we want to predict how various factors affect a person's salary in Canada. The salary database in Canada is public, but it only discloses individuals whose salary is above CAD 100,000. In other words, salaries below 100,000 are "censored".

In this case, you can use Tobit models.

Regression Analysis
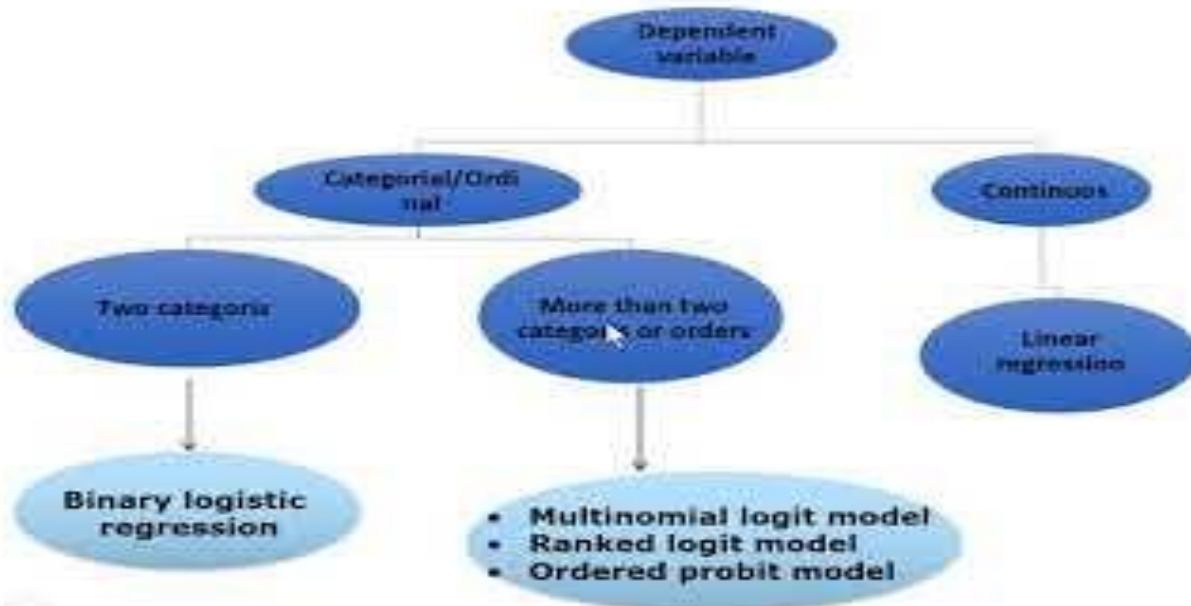
Going Beyond

Now, let's go beyond logistic regression in the marketing setting.

Let's consider some "machine learning" problems.

Going Beyond

Medical imaging:

    Input: medical scans
    Output: whether the
    patient has the disease

This is exactly what logistic
regression is doing!

Going Beyond

Recognizing hand-writing digits ---

Input: hand-writing
Output: Y_0 to Y_9

This is exactly what multinomial models are doing!

Going Beyond

Autonomous driving ---

Input: road condition
Output: Y =
turning left, turining right, ...

This is like multinomial logit models.

Going Beyond

Alpha Go ---

Input: current situation
Output: Y =
the place to move next

This is also like multinomial logit models.

Going Beyond

There are also a large number of similar scenarios such as

Gender detection (male or female?)

Facial recognition (Alice or not Alice?)

Voice recognition (which word?)

Neural Network

While multinomial logit models can help us make predictions about discrete outcomes, they are not powerful enough to address all the problems described above. We need to strengthen the model to make better predictions of these complex models.

One of such powerful models is neural network.

Neural Network

A basic neural network has three layers: an input layer, a hidden layer, and an output layer.

The input layer reads an input (e.g., the current road condition).

The hidden layer makes some calculation.

The output layer generates an output (e.g., turning right).

Neural Network
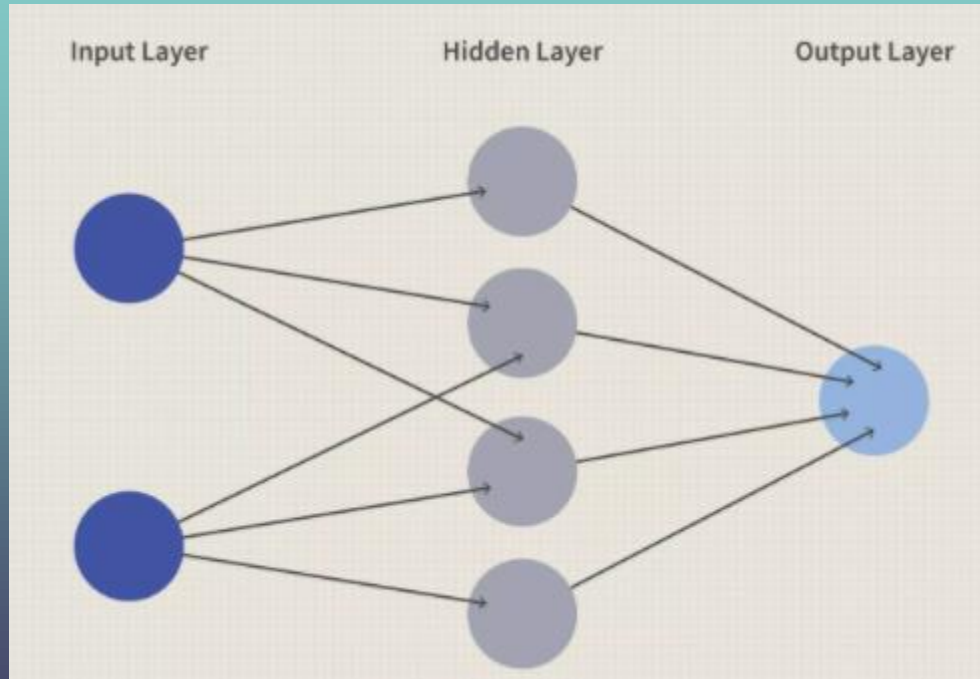
Neural Network

As shown in the figure, in a neural network, each layer consists of a number of units called "neurons". Neurons in the hidden layers are responsible of receiving information from the input layer, make calculation, and send output to the output layer.

What type of calculations does the hidden layer computes? It is essentially logistic function.

Neural Network

Neural networks are inspired by human brain. As human beings, we rely on our brain to make complex calculation such as "Is this person Alice?" "How to play the chess?" "Should I slow down the car a little bit?"

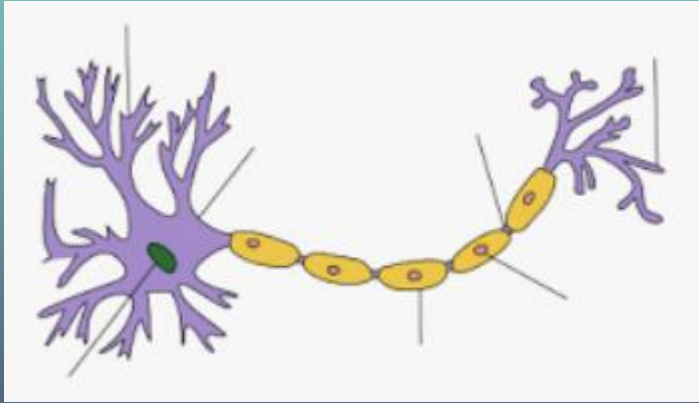While we can fluently handle these tasks, it is not yet clear how our brain makes these calculation…

Neural Network
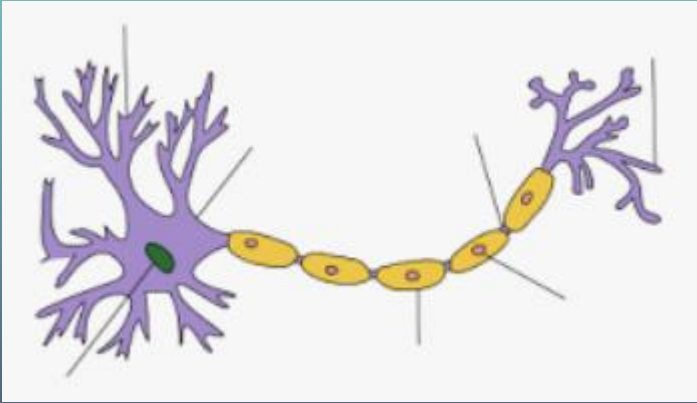
In our brain, there are billions of cells which are called "neurons". These neurons are capable of making calculations.

Neural Network

Some neurons receive input (e.g., light/sound) and convert the input into electric pulse. Some other neurons make calculations (i.e., convert one type of electric pulse into another type). Finally, some neurons output certain electric pulse.

Neural Network

The key is the calculations made in our brain. What type of calculation has been done by our neurons?

Surprisingly, neurons calculate the logistic function!

So, neuron network is just a simulation of our brain. This is why it has the word "neuron" in its name.

The Universal Approximation Theorem

In neural network, there is a key result called the universal approximation theorem.

Basically, it says that, when our neural network is large enough, it would be able to approximate any functions.

So, theoretically, neural networks should be capable of driving a car, recognizing human voice, writing a book or teaching a class.

The Universal Approximation Theorem

The issue is we have billions of neurons in our brain, and they make very complex calculations.

Due to the technology limit, we cannot build a neural network with billions of neurons --- it requires considerable memory and computing power that we cannot afford.

But this may become possible in the future.

Neural Network

To overcome the technology limit, scientists have also revised the basic neural network to make it easier to handle complex tasks. It can have multiple hidden layers, but the idea is more or less the same.
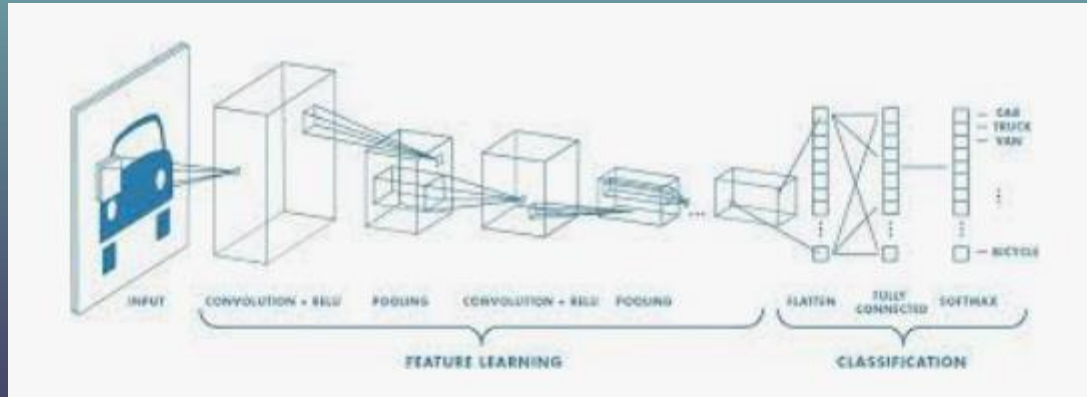
Type: Perceptron
Data Set: MNIST
Hidden Neurons: 2000
Synapses: 1191000
Synapses shown: 2%
Learning: WCor

0 1 2 3 4 5 6 7 8 9

www.cybercontrols.org