The Mathemmatical Engineering of Deep Learning

Chapter 2 - Lecture 2 - Part (1/3)

B. Liquet[1,2] and S. Moka[3] and Y. Nazarathy[3]

[1] Macquarie University [2] LMAP, Université de Pau et des Pays de L'Adour' [3] The University of Queensland

At the moment we use a significant number of figures and illustrations created by other authors.
We have attempted to properly attribute all such usage of figures and illustrations.
In cases where omissions are present, please accept our apology beforehand and we will rectify ASAP.

Outline of Lecture

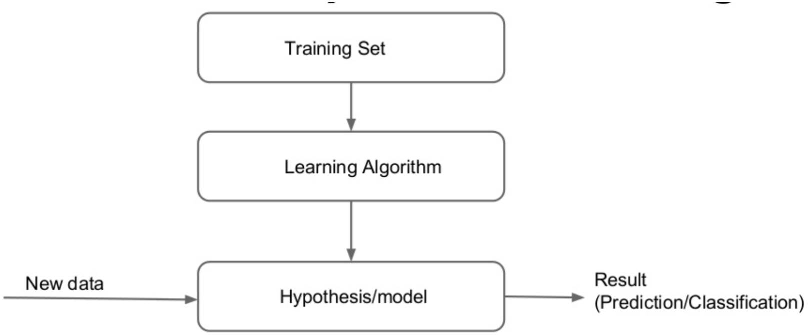
- Summary Lecture 1 (10 minutes)
- Logistic regression (20 minutes)
    - Statistical view
    - Machine Learning framework
- Softmax regression (20 minutes)
    - Statistical view
    - Machine Learning framework

DEFINITION:

- Samuel Muller (1959): https://en.wikipedia.org/wiki/Arthur_Samuel
  - "Machine Learning is the field of study that gives the computer the ability to learn without being explicitly programmed."
- Tom Mitchell (1998): https://en.wikipedia.org/wiki/Tom_M._Mitchell
  - "A computer program is said to learn from experience E with respect to some class of tasks T and performance measure P, if its performance at tasks in T, as measured by P, improves with experience E."

Supervised Learning: Big Picture

Process of Supervised Learning



Supervised Learning

Aim

To estimate the function f (**the model**) in the relationship

$$\boxed{Y = f(X) + \text{“error”},}$$

using observed input/output data

$$\text{data} = \{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_m, y_m)\}.$$

why:

- Prediction: Using the learned model $\widehat{f} \approx f$ we can predict

  $$\widehat{Y} = \widehat{f}(X)$$

  the value of a new unseen input X (“test input”).

- Inference: The model $\widehat{f}$ can help us to understand the relationships between input and output variables. *Useful for decision making but also to advance our knowledge and to construct better models*

Supervised Learning

- Regression: when the output Y is quantitative:
  - Marketing: Y= *housing price*
  - Climate models: Y = *increase in global temperature*
- Classification: when the output Y is qualitative
  - Diagnosis cancer ($Y \in \{\text{"Malignant"}, \text{"Benign"}\}$)
  - Spam filters ($Y \in \{\text{"spam"}, \text{"good email"}\}$)
  - Image classification: MNIST ($Y \in \{0, 1, \ldots, 9\}$)

Regression: linear case

$$
\begin{aligned}
Y &= \beta_0 + \sum_{j=1}^{d} X_j \beta_j + \varepsilon \\
&= \beta^T X + \varepsilon,
\end{aligned}
$$

where β is the parameters composed by the **"weights"** β_j and the offset (**"bias"**/"intercept") term β_0,

$$
\beta = \begin{pmatrix} \beta_0 & \beta_1 & \beta_2 & \cdots & \beta_d \end{pmatrix}^T,
$$

$$
X = \begin{pmatrix} 1 & X_1 & X_2 & \cdots & X_d \end{pmatrix}^T.
$$

How to estimate this model?

- Loss function
- Likelihood approach

Linear case: Loss function

Training Step: we want to make $\widehat{f}(X)$ close to Y.

- Closeness between $\widehat{f}(X)$ and Y is evaluated using loss function
- Linear case: Squared loss

$$L(Y, \widehat{f}(X)) = (Y - \widehat{f}(X))^2 \qquad (\Rightarrow \text{MSE})$$

- Testing Step: It is more common to use same loss: function
  - when training the model (minimizing loss of training data)
  - when testing the model (evaluating loss at test inputs)

Minimize the Cost function

Cost function for the $\mathrm{data} = \{(x_1, y_1),\ (x_2, y_2),\ \ldots,\ (x_m, y_m)\}$. $(x_j \in \mathfrak{R}^d)$

$$
\begin{aligned}
J(\beta) &= \frac{1}{m}\sum_{i=1}^{m} L(y_i - \widehat{f}(x_i)) \\
&= \frac{1}{m}\sum_{i=1}^{m} (Y_i - \beta^T X_i)^2 \\
&= \frac{1}{m}(Y - X\beta)^T (Y - X\beta)
\end{aligned}
$$

TO DO NOW: derive an estimate of $\beta \in \mathfrak{R}^d$

Solution

Likelihood approach

Reminder Likelihood for an i.i.d. sample $\boldsymbol{y} = (y_1, \ldots, y_n)$

General case: $Y_i \sim f(y; \theta)$

$$L(\theta; \boldsymbol{y}) = \prod_{i=1}^{n} f(y_i; \theta) \quad \text{(since } Y_i \text{ are independent)}$$

Example: $Y_i \sim N(\mu, \sigma^2)$ $(\theta = (\mu, \sigma^2))$

$$\begin{aligned} L(\mu, \sigma^2, \boldsymbol{y}) &= \prod_{i=1}^{n} \frac{1}{\sqrt{2\pi\sigma^2}} \exp\left(-\frac{(y_i - \mu)^2}{2\sigma^2}\right) \\ &= \frac{1}{(2\pi\sigma^2)^{n/2}} \exp\left(-\frac{\sum_{i=1}^{n}(y_i - \mu)^2}{2\sigma^2}\right) \end{aligned}$$

Likelihood approach

Next step: find $\theta = (\mu, \sigma^2)$ which maximises the likelihood

$\hookrightarrow$ Differentiate the **log**-likelihood with respect to the parameter, and set to 0 for the maximum:

$$\frac{\partial \log L(\theta; \boldsymbol{y})}{\partial \theta} = 0$$

Linear model:

$$Y_i = \beta^T X_i + \varepsilon_i, \quad i = 1, \ldots, m$$

where $\varepsilon_i \overset{iid}{\sim} N(0, \sigma^2)$

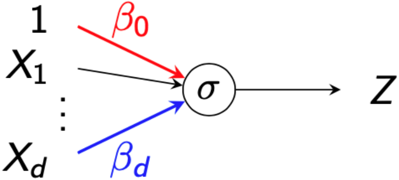
TO DO NOW: derive the maximum likelihood estimate of $\beta \in \mathfrak{R}^d$

Solution

Linear model is a shallow NN ?

Neural Network

A neural network (NN) is a nonlinear function $Y = f(X;\theta) + \varepsilon$ from an input X to an output Y parameterized by parameters θ.

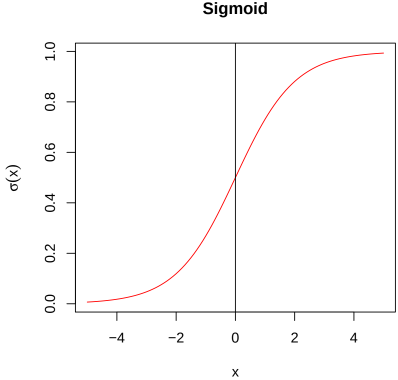


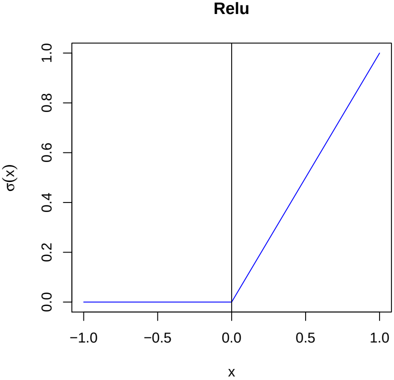
$$
\begin{aligned}
Y &= \sigma(\beta^T X) + \varepsilon, \\
&\quad \text{or equivalently} \\
&= Z + \varepsilon, \qquad \text{with} \qquad Z = \sigma(\beta^T X)
\end{aligned}
$$

- linear model: activation function $\sigma(x)$ is the identity function $\sigma(x) = x$.

Extend the linear model

NN introduces nonlinear transformations of the predictor $\beta^T X$,





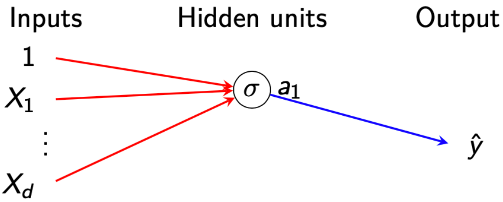
Sigmoid: $\sigma(x) = \frac{1}{1+e^{-x}}$

ReLu: $\sigma(x) = max(0, x)$

Neural Network: construction

Neural Network: construction

A neural network can be viewed as sequential construction of several linear regression models

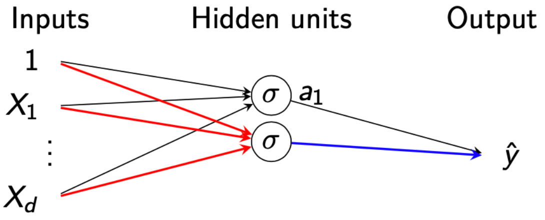


$$a_1 = \sigma(X^T \beta_1^{(1)})$$
$$=$$

$$\widehat{y} = \beta_1^{(2)} a_1$$

Neural Network: construction

A neural network can be viewed as sequential construction of several linear regression models

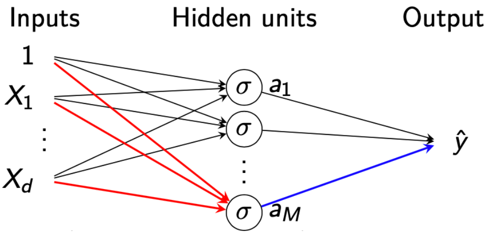


$$
\begin{aligned}
a_1 &= \sigma(X^T \beta_1^{(1)}) \\
a_2 &= \sigma(X^T \beta_2^{(1)})
\end{aligned}
$$

$$\widehat{y} = \beta_1^{(2)} a_1 + \beta_2^{(2)} a_2$$

Neural Network: construction

A neural network can be viewed as sequential construction of several linear regression models



$$\begin{aligned} a_1 &= \sigma(X^T\beta_1^{(1)}) \\ a_2 &= \sigma(X^T\beta_2^{(1)}) \\ &\vdots \\ a_M & \quad \sigma(X^T\beta_2^{(1)}) \end{aligned}$$

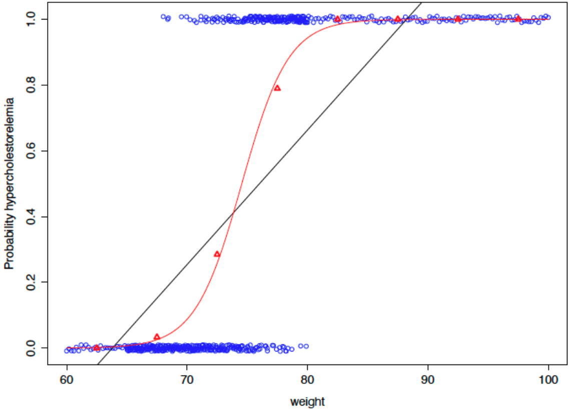
$$\widehat{y} = \sum_{m=1}^{M} \beta_m^{(2)} a_m$$

Example using R

Demo

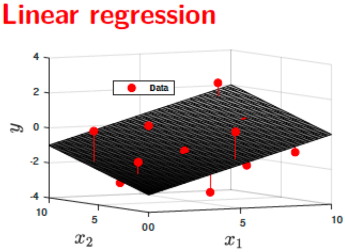
Consider binary classification problems: $y \in \{0, 1\}$

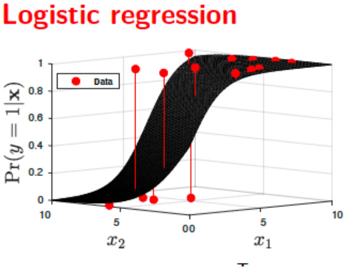
- Example 1: predicting **hypercholesterolemia** ($y = 1$) given weight x



What's wrong with the linear model (black line)

With two features:





Logistic Regression

Consider the data from *Breast Cancer Wisconsin (Diagnostic)* (WBCD) dataset

- Aim: discriminate benign ($Y = 0$) from malignant ($Y = 1$) lumps of a breast mass
- 30 (=d) characteristics of individual cells of breast cancer

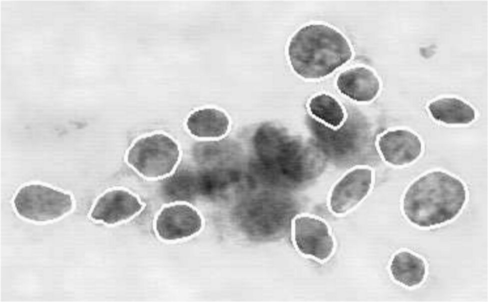
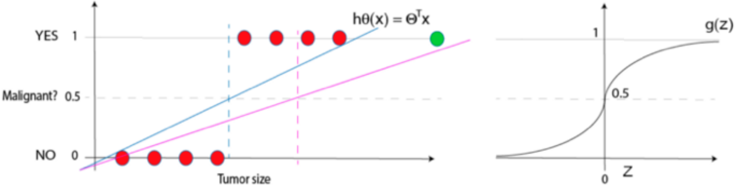


Figure 1: A magnified image of a malignant breast FNA. A curve-fitting algorithm was used to outline the cell nuclei. (Figure from Mangasarian OL., Street WN., Wolberg. WH. Breast Cancer Diagnosis and Prognosis via Linear Programming. Mathematical Programming Technical Report 94–10. 1994 Dec)


YES 1
Malignant? 0.5
NO 0
Tumor size
hθ(x) = ΘTx
g(z)
1
0.5
0
Z

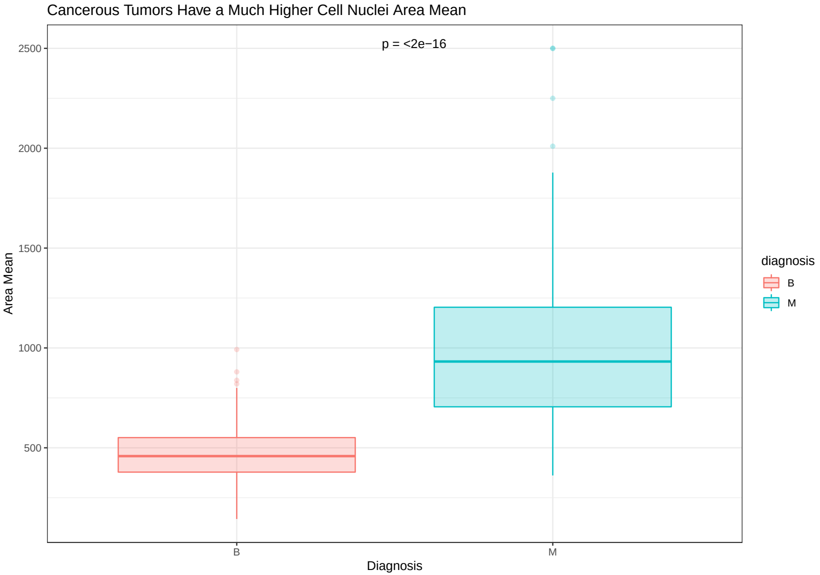
Logistic Regression

- Data: 30 features ..

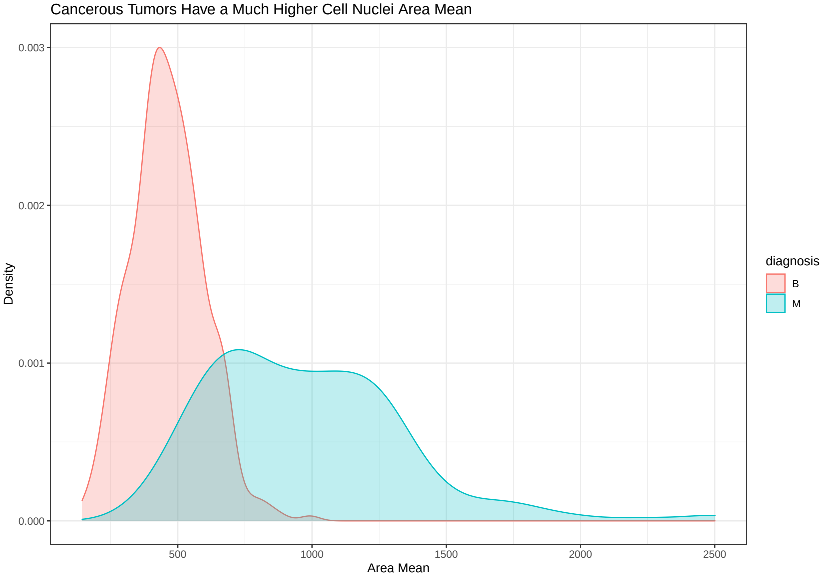
```
  diagnosis radius_mean texture_mean perimeter_mean area_mean
1         M       17.99        10.38         122.80    1001.0
2         M       20.57        17.77         132.90    1326.0
3         M       19.69        21.25         130.00    1203.0
4         M       11.42        20.38          77.58     386.1
5         M       20.29        14.34         135.10    1297.0
6         M       12.45        15.70          82.57     477.1
```

```
[1] 569  32
```

Visualisation



Density plot



Simple logistic model

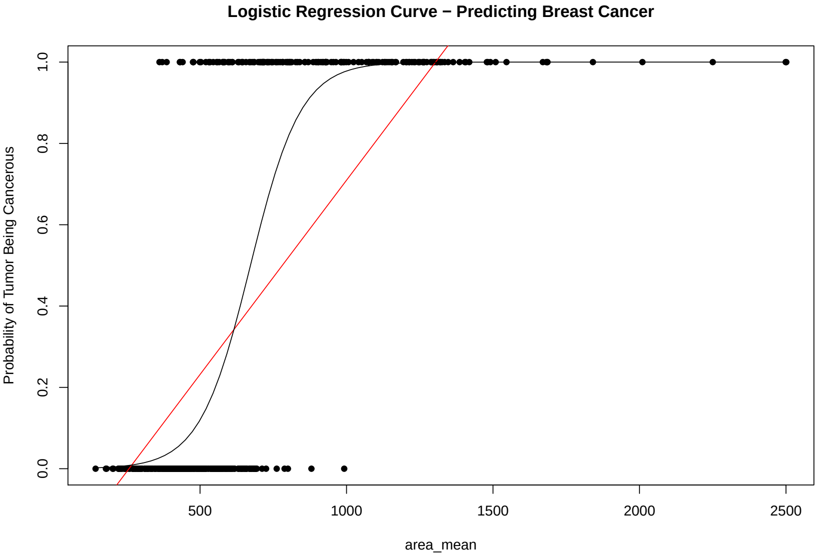
```
Call:
glm(formula = diagnosis_0_1 ~ area_mean, family = "binomial",
    data = train)

Deviance Residuals:
    Min       1Q   Median       3Q      Max
-2.7323  -0.4762  -0.1997   0.1159   2.6929

Coefficients:
             Estimate Std. Error z value Pr(>|z|)
(Intercept) -7.789754   0.742988 -10.484   <2e-16 ***
area_mean    0.011590   0.001191   9.735   <2e-16 ***
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)
```

Probability of Tumor Being Cancerous



Result for a 0.5 cut-off

```
          Reference
Prediction   0   1
         0 263  37
         1  18 138
```

```
Accuracy
0.879386
```

```
We made 401 correct predictions,
```

```
55 incorrect predictions,
```

```
thus giving us an accuracy rating of: 87.9%
```

Why only results on 456 samples and not on 569?

Overfit

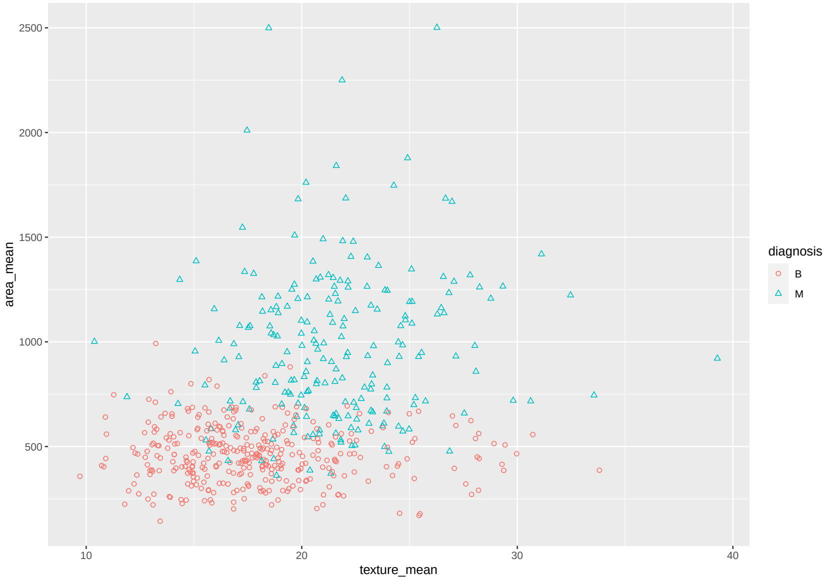
How our model will generalize to new samples that we didn't use to train

Solution to quantify the true **generalization error** is to split the data:

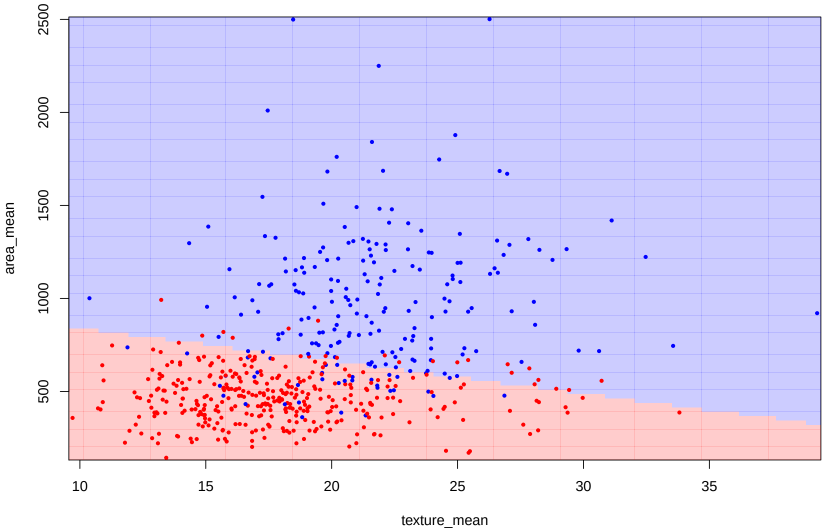
- First version: **holdout cross-validation**

- Second version: **K-fold cross-validation**

Two predictors



Classification with the logistic model

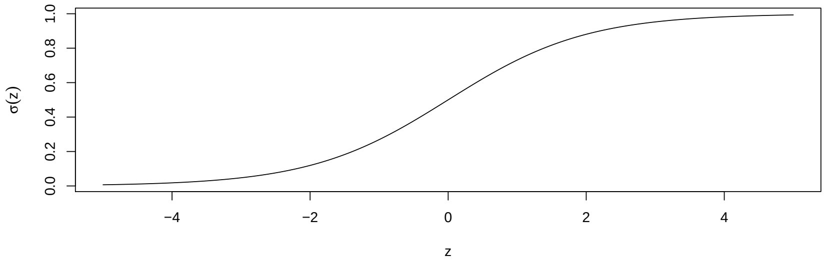


Linear decision bundary

It looks a linear decision boundary while we use a non linear function as the logistic model!!!

- Need further explanation of the logistic model
- sigmoid function $\sigma(\cdot)$, also known as the logistic function, is defined as follows:

$$\forall z \in \mathbb{R}, \quad \sigma(z) = \frac{1}{1 + e^{-z}} \in]0, 1[$$



Definition of logistic model

- A probabilistic model to predict the probability that the outcome variable y is equal to 1.
- $y|x;\theta \sim \mathrm{Bernoulli}(\phi)$.
- Logistic regression is defined by applying **the sigmoid function** to the linear predictor $\theta^T x$:

$$\phi = h_\theta(x) = p(y = 1|x;\theta) = \frac{1}{1 + \exp(-\theta^T x)} = \sigma(\theta^T x)$$

The logistic regression is also presented:

$$\mathrm{Logit}[h_\theta(x)] = logit[p(y = 1|x;\theta)] = \theta^T x$$

where $\mathrm{Logit}(p) = log\left(\frac{p}{1-p}\right)$.

Likelihood of the logistic model

The maximum likelihood estimation procedure:

$$p(y|x;\theta) = \begin{cases} h_\theta(x) & \text{if } y = 1, \text{ and} \\ 1 - h_\theta(x) & \text{otherwise.} \end{cases}$$

which could be written as

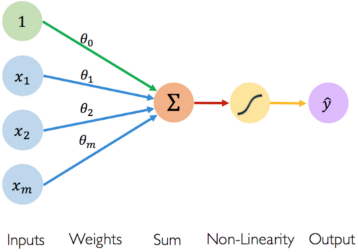
$$p(y|x;\theta) = h_\theta(x)^y (1 - h_\theta(x))^{1-y},$$

Likelihood for m training:samples denoted by $\left\{(x^{(1)}, y^{(1)}), \ldots, (x^{(m)}, y^{(m)})\right\}$

$$\begin{aligned} L(\theta) &= \prod_{i=1}^{m} p(y^{(i)}|x^{(i)};\theta) \\ &= \prod_{i=1}^{m} h_\theta(x^{(i)})^y (1 - h_\theta(x^{(i)}))^{1-y} \end{aligned}$$

$$\ell(\theta) = \log L(\theta) = \sum_{i=1}^{m} \left[y^{(i)} \log h_\theta(x^{(i)}) + (1 - y^{(i)}) \log (1 - h_\theta(x^{(i)})) \right]$$

Shallow Neural Network

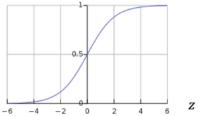


Activation Functions

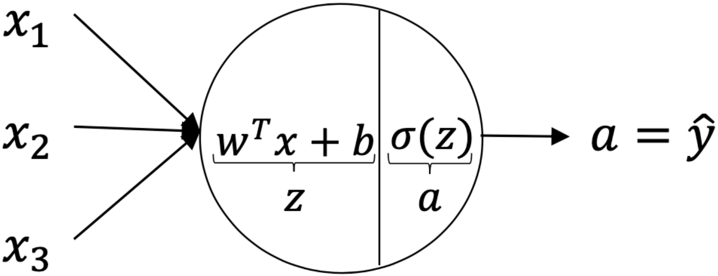
$$\hat{y} = g\left(\theta_0 + \boldsymbol{X}^T \boldsymbol{\theta} \right)$$

- Example: sigmoid function

$$g(z) = \sigma(z) = \frac{1}{1 + e^{-z}}$$



MIT: Alexander Amini, 2018 introtodeeplearning.com


x_1
x_2
x_3
$w^T x + b$
$\sigma(z)$
z
a
$a = \hat{y}$

Cross-entropy Loss

- For binary class problem, **Cross-entropy** loss is the most popular (due to property of convexity)
- The **cross-entropy** is defined for one sample (x, y):

$$
\begin{aligned}
L_{CE}(y, \widehat{y}) &= \begin{cases} -\log \widehat{y} & \text{if } y = 1 \\ -\log(1 - \widehat{y}) & \text{if } y = 0 \end{cases} \\
&= -y \log \widehat{y} - (1 - y) \log(1 - \widehat{y})
\end{aligned}
$$

Cost function:

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\widehat{y^{(i)}}, y^{(i)})$$

Connection with the log-likelihood

$$J(w,b) = \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\widehat{y^{(i)}}, y^{(i)})$$

Further, it is easy to see the connection with the log-likelihood function of the logistic model:

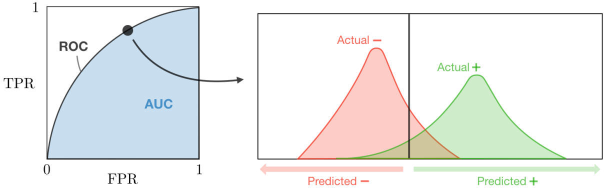
$$\begin{aligned} J(w,b) &= \frac{1}{m}\sum_{i=1}^{m}\mathcal{L}(\widehat{y^{(i)}}, y^{(i)}) \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log a^{(i)}) + (1-y^{(i)})\log(1-a^{(i)})\right] \\ &= -\frac{1}{m}\sum_{i=1}^{m}\left[y^{(i)}\log h_\theta(x^{(i)}) + (1-y^{(i)})\log(1-h_\theta(x^{(i)}))\right] \\ &\equiv -\frac{1}{m}\ell(\theta) \end{aligned}$$

How to maximize?

- Next lecture . . . .

Main Metrics and AUC

- Main Metrics:
  - Precision
  - Recall
  - F_1
- **AUC**. The area under the receiving operating curve, also noted AUC or AUROC



Multiclass Classification

Multiclass classification: predicting a discrete (> 2)-valued target

- predict the value of a handwritten digit
- classify e-mails as spam, travel, work, personal

Multiclass Classification

- Targets form a discrete set $\{1, \ldots, K\}$
- It's often more convenient to represent them as **one-hot vectors**, or **a one-of-K encoding**:

$$y^* = \underbrace{(0, \ldots, 0, 1, 0, \ldots, 0)^T}_{\text{entry k is 1}} \in \Re^K$$

- softmax regression, also called a multiclass logistic regression is used when there are more than 2 outcome classes ($k = 1, \ldots, K$).

Probabilistic Model

multinomial regression model

A GLM model where the distribution of the outcome y is a Multinomial$(1, \pi)$ where $\pi = (\phi_1, \ldots, \phi_K)$ is a vector with probabilities of *success* for each category. This Multinomial$(1, \pi)$ is more precisely called *categorical distribution*.

- The **multinomial regression model** is parameterize by $K - 1$ parameters, $\phi_1, \ldots, \phi_K$, where $\phi_i = p(y = i; \phi)$, and $\phi_K = p(y = K; \phi) = 1 - \sum_{i=1}^{K-1} \phi_i$.
- We set $\theta_K = 0$, which makes the Bernoulli parameter ϕ_i of each class i be such that

$$\phi_i = \frac{\exp(\theta_i^T x)}{\sum_{j=1}^{K} \exp(\theta_j^T x)}, \qquad \text{where} \quad \theta_1, \ldots, \theta_{K-1} \in \mathfrak{R}^{d+1}$$

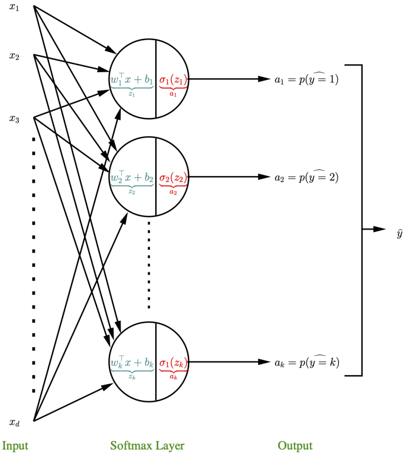
- Output of the model: estimated probability $p(y = i | x; \theta)$, for every value of $i = 1, \ldots, K$.

Likelihood of the softmax model

The maximum likelihood estimation procedure consists to maximizing the log-likelihood:

$$
\begin{aligned}
\ell(\theta) &= \sum_{i=1}^{n} \log p(y^{(i)}|x^{(i)};\theta) \\
&= \sum_{i=1}^{m} \log \prod_{l=1}^{K} \left(\frac{e^{\theta_l^T x^{(i)}}}{\sum_{j=1}^{K} e^{\theta_j^T x^{(i)}}} \right)^{1_{\{y^{(i)}=l\}}}
\end{aligned}
$$

Neural network



$$\widehat{y} = \operatorname*{argmax}_{i \in \{1,\ldots,K\}} a_i$$

Loss function: cross-entropy for categorical variable

- Let consider first one training sample (x, y).
- The cross entropy loss for categorical response variable, also called **Softmax Loss** is defined as:

$$
\begin{aligned}
CE &= -\sum_{i=1}^{K} \widetilde{y}_i \ln p(y = i) \\
&= -\sum_{i=1}^{K} \widetilde{y}_i \ln a_i \\
&= -\sum_{i=1}^{K} \widetilde{y}_i \ln \left(\frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} \right)
\end{aligned}
$$

where $\widetilde{y}_i = 1_{\{y=i\}}$ is a binary variable indicating if y is in the class i.

This expression can be rewritten as

$$CE = -\ln \prod_{i=1}^{K} \left(\frac{\exp(z_i)}{\sum_{j=1}^{K} \exp(z_j)} \right)^{1_{\{y=i\}}}$$

Then, the cost function for the *m* training samples is defined as

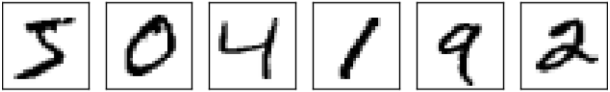
$$\begin{aligned} J(w,b) &= -\frac{1}{m} \sum_{i=1}^{m} \ln \prod_{k=1}^{K} \left(\frac{\exp(z_k^{(i)})}{\sum_{j=1}^{K} \exp(z_j^{(i)})} \right)^{1_{\{y^{(i)}=k\}}} \\ &\equiv -\frac{1}{m}\ell(\theta) \end{aligned}$$

Take Home Message

- Likelihood
- Logistic model
- Sigmoid
- ReLu
- Squared loss
- Cross entropy loss
- Metrics
- Softmax

Home work: handwritten digits

We want to classify images ($28 \times 28 = 784$ pixels) such as these



into 10 classes (0 to 9)

Work to do

- **One versus All** using 10 logistic models
- Softmax regression