Machine Learning Group

Ovidiu Calin, October 13, 2017

The sigmoid neuron

Perceptron - provides two decisions states, 0 and 1, without no intermediate values.

Sigmoid neuron - takes all states in the interval $(0, 1)$.

• Outputs closer to 1 correspond to decisions that are more likely to occur,

• Outputs closer to 0 represent decisions not that likely to happen.

• The neuron output acts as the likelihood of taking a certain weighted decision.

A *sigmoid neuron* is a computing unit with

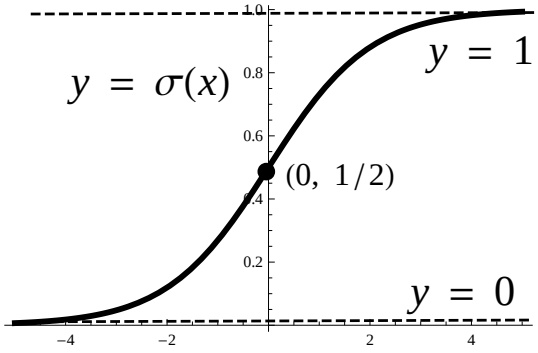
- inputs $x_1, \ldots, x_n$.
- activation function given by the logistic function

$$\sigma(z) = \frac{1}{1 + e^{-z}}. \tag{0.1}$$

- output given by a number in $(0, 1)$, which is given by

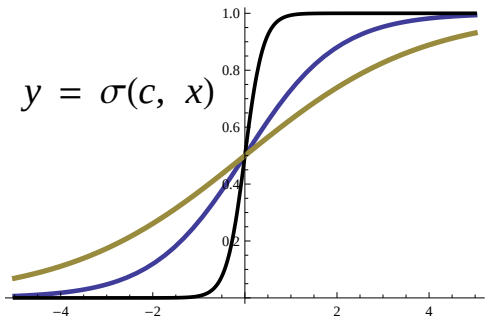
$$y = \sigma(x^T w - b) = \frac{1}{1 + e^{-w^T x + b}}.$$

$w =$ and $b =$ bias.



The logistic function

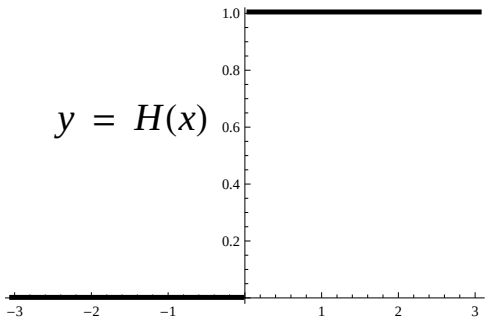
$$\sigma(x) = \frac{1}{1 + e^{-x}}.$$



The scaled logistic function

$$\sigma_c(z) = \frac{1}{1 + e^{-cz}}, \qquad c > 0$$

tends to $H(x)$ as $c \to \infty$.



The scaled logistic function

$$\sigma_c(z) = \frac{1}{1 + e^{-cz}}, \qquad c > 0$$

tends to $H(x)$ as $c \to \infty$.

Advantages of sigmoid neuron:

• it approximates perceptrons;

• the output function is continuous (differentiable) with respect to weights. This is important for the learning process (allows the use of gradient descent method).

• the freedom to choose a sigmoidal function φ, such as: hyperbolic tangent, arctangent function, soft-sign, etc., enabling to accommodate different learning speeds due to different rates of saturation.

Sensitivity of the output w.r.t. weights and biasses

By chain rule, the infinitesimal change in the output $y = \varphi(w^T x - b)$ is

$$\begin{aligned} dy &= \sum_{i=1}^{n} \frac{\partial y}{\partial w_i}\, dw_i + \frac{\partial y}{\partial b}\, db \\ &= \sum_{i=1}^{n} \varphi'(w^T x - b) x_i dw_i + \varphi'(w^T x - b)(-1)\, db \\ &= \varphi'(w^T x - b)\Big(\sum_{i=1}^{n} x_i dw_i - db \Big). \end{aligned}$$

The rate φ' in some cases can be easily computed:

If $\varphi(z) = \sigma(z)$ is the logistic function, then

$$dy = \sigma(w^T x - b)\big(1 - \sigma(w^T x - b)\big)\Big(\sum_{i=1}^{n} x_i dw_i - db\Big).$$

If $\varphi(z) = \mathbf{t}(z)$ is the hyperbolic tangent function, we have

$$dy = (1 - \mathbf{t}^2(w^T x - b))\Big(\sum_{i=1}^{n} x_i dw_i - db\Big).$$

In the gradient descent method we need the gradient of the output $\nabla y = (\nabla_w y, \partial_b y)$, which becomes:

$$\begin{aligned} \nabla y &= (\nabla_w y, \partial_b y) \\ &= (x\varphi'(w^T x - b), -\varphi'(w^T x - b)) \\ &= \varphi'(w^T x - b)(x, -1). \end{aligned}$$

Logistic regression

We present two examples of learning using logistic regression:

- forecasting the probability of default of a company
- binary classification of clusters.

Default probability of a company

Goal: predicting the probability of default of a company for a given time period $[0, T]$, using a sigmoid neuron.

Neuron input x is a vector which contains some information regarding the company, such as: cash reserves, revenue, costs, labor expenses, etc.

Training set consists on n pairs (x_i, z_i), with x_i inputs as before and $z_i \in \{-1, 1\}$.

• $z_i = 1$ means the ith company defaulted during $[0, T]$;

• $z_i = -1$ means that the ith company had not defaulted during $[0, T]$.

• $(x_i, 1)$ means that a company with the features x_i defaulted during $[0, T]$;

• $(x_i, -1)$ means that a company with the features x_i had not defaulted during $[0, T]$;

The measurements $(x_i, z_i)_{1 \leq i \leq n}$ represent empirically a pair of random variables (X, Z),
where $Z \in \{-1, 1\}$.

The conditional probability $P(Z|X)$ is given by the table

z	-1	1
$P(z\|x)$	$1 - h(x)$	$h(x)$

for $h : \mathbb{R}^m \to [0, 1]$, where m is the dimension of the input X. One convenient way to choose $h(x)$ is using the sigmoid function

$$h(x) = \sigma(w^T x).$$

Risk score

• The inner product $w^T x$ is a weighted sum of the inputs and can be interpreted as a "risk score" .

• Consider $\sigma(w^T x)$ as a probability.

• A high (positive) score $w^T x$ implies a value of $\sigma(w^T x)$ close to 1 (high probability of default)

• A low (negative) score $w^T x$ implies a value of $\sigma(w^T x)$ close to 0 (low probability of default).

The default probability writes as $\sigma(zw^T x)$, where $z \in \{-1, 1\}$. Why?

If $z = 1$, we recover the above formula,

$$h(x) = \sigma(w^T x).$$

If $z = -1$, using the property $\sigma(-x) = 1 - \sigma(x)$, we have

$$\sigma(zw^T x) = \sigma(-w^T x) = 1 - \sigma(w^T x) = 1 - h(x).$$

Hence, the previous table

z	-1	1
$P(z\|x)$	$1-h(x)$	$h(x)$

can be written as

z	-1	1
$P(z\|x)$	$\sigma(-w^T x)$	$\sigma(w^T x)$

The probability distribution depends now on the parameter w.

What is the weight w for which the above model distribution approximates in "the best way" the training data $\{(x_i, z_i)\}$, $1 \leq i \leq n$?

- "the best way" = choosing w such that the likelihood of z_i being in a proximity of y_i is maximized.
- method = maximum likelihood method.
- Assume independent training measurements.

$$
\begin{aligned}
w^* &= \arg\max_w P(z_1, \cdots, z_n | x_1, \cdots, x_n) \\
&= \arg\max_w \prod_{i=1}^{n} P(z_i|x_i) = \arg\max_w \ln\Big(\prod_{i=1}^{n} P(z_i|x_i)\Big) \\
&= \arg\max_w \sum_{i=1}^{n} \ln P(z_i|x_i) = \arg\min_w \Big(-\sum_{i=1}^{n} \ln P(z_i|x_i)\Big) \\
&= \arg\min_w \Big(-\frac{1}{n}\sum_{i=1}^{n} \ln P(z_i|x_i)\Big) = \arg\min_w \mathbb{E}^{\widehat{p}_X}\Big[-\ln P(Z|X)\Big].
\end{aligned}
$$

Notations:

$\widehat{p}_X$= input empirical probability

$\mathbb{E}^{\widehat{p}_X}\Big[-\ln P(Z|X)\Big]$ = the cross entropy of the empirical probability $\widehat{p}_X$ with the conditional probability $P(Z|X)$.

The expression of the cross entropy can be computed explicitly.

$$
\begin{aligned}
\mathbb{E}^{\widehat{p}_X}\Big[-\ln P(Z|X)\Big] &= -\frac{1}{n}\sum_{i=1}^{n} \ln P(z_i|x_i) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \ln \sigma(z_i w^T x_i) \\
&= -\frac{1}{n}\sum_{i=1}^{n} \ln \frac{1}{1+e^{-z_i w^T x_i}} \\
&= \frac{1}{n}\sum_{i=1}^{n} \ln(1+e^{-z_i w^T x_i}).
\end{aligned}
$$

The optimal weight is then given by

$$w^* \;=\; \arg\min_{w} \left(\frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-z_i w^T x_i}) \right)$$

Need to minimize the cross-entropy error function by gradient descent method.

Gradient Descent

• There is no closed form solution in this case for the optimum weight w^*.

• We approximate the value w^* by a sequence $w^{(j)}$ using the gradient descent method.

• Consider the error function

$$F(w) = \frac{1}{n} \sum_{i=1}^{n} \ln(1 + e^{-z_i w^T x_i})$$

and compute its gradient

$$\begin{aligned}
\nabla_w F &= -\frac{1}{n}\sum_{i=1}^{n}\frac{z_i x_i e^{-z_i w^T x_i}}{1+e^{-z_i w^T x_i}} = -\frac{1}{n}\sum_{i=1}^{n}\frac{z_i x_i}{1+e^{z_i w^T x_i}} \\
&= \frac{1}{n}\sum_{i=1}^{n} z_i x_i \sigma(-z_i w^T x_i) \\
&= \frac{1}{n}\sum_{i=1}^{n} z_i x_i \sigma(z_i w^T x_i) - \frac{1}{n}\sum_{i=1}^{n} z_i x_i.
\end{aligned}$$

The first term is a weighted sum of the entries (x_i, z_i) with the weights $\rho_i = \sigma(z_i w^T x_i) \in (0,1)$. The last term is the empirical expectation $\widehat{\mathbb{E}}[XZ] = \frac{1}{n}\sum_{i=1}^{n} z_i x_i$.

Given an initial weight initialization $w^{(0)}$, the approximation sequence that approaches the minimum argument, w^*, of the error $F(w)$ is given by

$$\begin{aligned} w^{(j+1)} &= w^{(j)} - \delta \nabla_w F(w^{(j)}) \\ &= w^{(j)} - \frac{\delta}{n} \sum_{i=1}^{n} z_i x_i \, \sigma(-z_i {w^{(j)}}^T x_i), \end{aligned}$$

where $\delta > 0$ is the learning rate.

Assume now we have the training set $\{(x_i, z_i)\}_{i \leq n}$ and consider a given input x describing the parameters of a certain company we would like to evaluate the probability of default.

Then the desired probability is given by

$$h(x) = \sigma(w^{*T} x)$$

where $w^* = \lim_{j \to \infty} w^{(j)}$.

Binary Classifier

Goals:

• use the sigmoid neuron as a binary classifier.

• we train a sigmoid neuron to distinguish between two distinct clusters in the plane, the learning algorithm being the logistic regression.

Problem:

Assume we have two groups of points in the plane: *black* and *white*. We would like to split the points in two groups:

- black points cluster, denoted by $\mathcal{G}_1$ and
- white points cluster, denoted by $\mathcal{G}_2$.

If $x = (x_1, x_2)$ represents the coordinates of a point in the plane, consider the target given by the following decision function

$$z(x_1, x_2) = \begin{cases} 1, & \text{if } x \in \mathcal{G}_1 \\ -1, & \text{if } x \in \mathcal{G}_2, \end{cases}$$

i.e. each black point is associated the label 1, while each white point the label -1.

Decision line The line

$$w_1 x_1 + w_2 x_2 - b = 0,$$

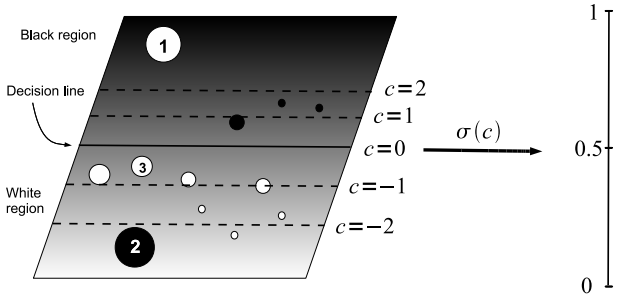
attempts to split the points into clusters $\mathcal{G}_1$ and $\mathcal{G}_2$; say $\mathcal{G}_1$ above the decision line and $\mathcal{G}_2$ below it.

Goal: adjust parameters w_1, w_2 and b such that the aforementioned line represents "the best" split of data into clusters of identic color.

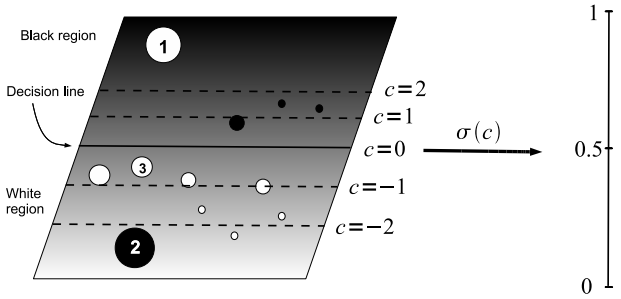
Task: Define "the best"!

Consider the partition of the plane into lines parallel to the decision line

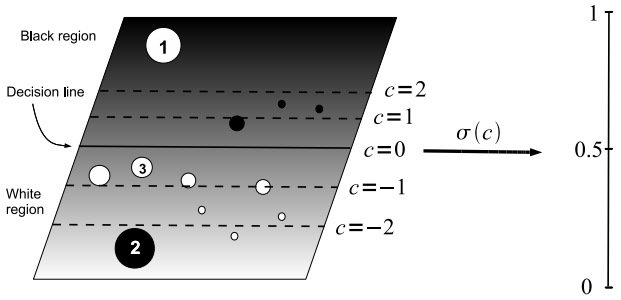
$$\{w_1x_1 + w_2x_2 - b = c;\ c \in \mathbb{R}\},$$



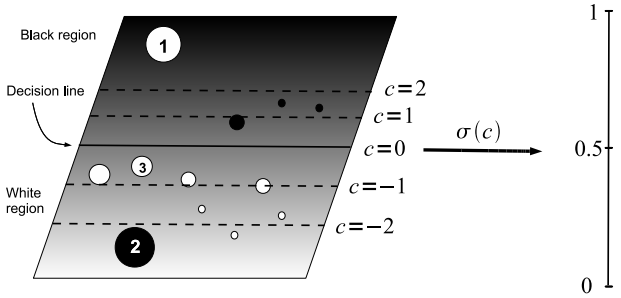
The lines with a positive value of c correspond to the black region, $\mathcal{G}_1$, while the lines with a negative value of c correspond to the white region, $\mathcal{G}_2$. The line corresponding to $c = 0$ is the decision line between the two clusters of distinct colors.



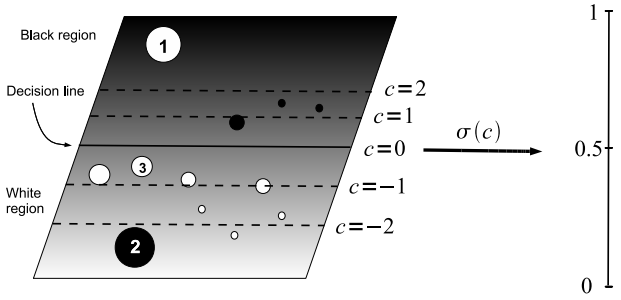
We can think the plane as a gray scale from pure white, when $c = -\infty$, to pure black, for $c = +\infty$. Also the 50% black-white mixture is realized for $c = 0$.



If a given point has a color that is in large discrepancy with the background, then the "surprise" element is large, and this corresponds to a large amount of information.



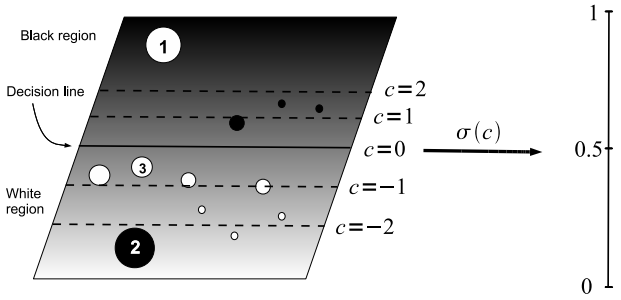
If the point color and the background do not differ much in tone, the amount of information provided by this point is small. We shall construct an information measure based on this color difference effect, subject to be minimized later.



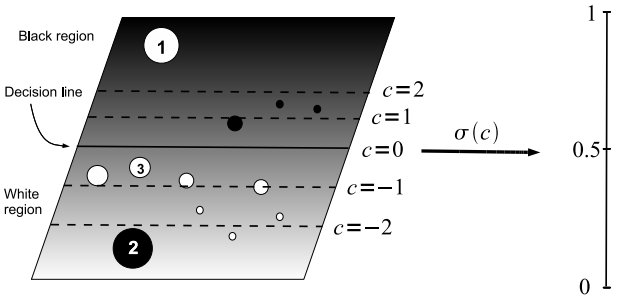
The information associated to a point of a given color with coordinate x is defined as

$$H(x) = -\ln \sigma\big(z(x)(w^T x - b)\big).$$

Why?



Pick an unclassified point of coordinate x (a point of a different color than its neighboring background). For instance, we pick the point with label "1". This is a white point in a very black region, so the information associated with this event is large.



There is $c > 0$ such that this point belongs to the line $w_1x_1 + w_2x_2 - b = c$. Then σ maps c into a value between 0 and 1, which can be considered as a probability, $P(x) = \sigma(c)$.

The information associated with the point x is given by the negative log-likelihood function, $-\ln P(x) = -\ln \sigma(c)$.

If c is large, then $\sigma(c)$ is close to 1 and hence the information $-\ln P(x)$ is close to 0, which does not make sense, since when c is large, the information should be also large (as a white point in very black region).

To fix this problem, we define the information slightly different, using the complementary probability. Since the point is white, $x \in \mathcal{G}_2$, then $z(x) = -1$.

Define the probability as

$$P(x) = \sigma(z(x)c) = \sigma(-c) = 1 - \sigma(c)$$

so when c is large, the probability is close to 0 and hence the information

$$-\ln P(x) = -\ln \sigma(z(x)c) = -\ln \sigma(-c)$$

is also large.

Choose a correct classified point, for instance the white point with label "3". Let x denote its coordinate and $c < 0$ be the constant such that $w^T x - b = c$. The information associated with this point is small if the difference in the tone color with respect to the background is small. This occurs when the constant c is large and negative.

Associate a probability with x, defined by

$$P(x) = \sigma(z(x)c).$$

Then $z(x) = -1$, since the point is white, $x \in \mathcal{G}_2$.

For c large and negative the value of $\sigma(z(x)c)$ is close to 1, and hence its logarithm is close to 0. Therefore, it makes sense to define the information of this point as $-\ln \sigma(z(x)c)$.

Information minimization error

Consider n points in the plane with coordinates x_i and color type $z_i = z(x_i)$. The total information is defined as the sum of all individual points information (both correct and misclassified). This is

$$E(x, z) = \sum_{i=1}^{n} H(x_i) = -\sum_{i=1}^{n} \ln \sigma\big(z_i(w^T x_i - b)\big)$$

The "best decision line" is the line corresponding to the values of w and b that minimize the information.

The optimal values are given by

$$\begin{aligned}
(w^*, b^*) &= \arg\min_{w,b} E(x, z) = \arg\max_{w,b} \sum_{i=1}^{n} \ln \sigma\big(z_i(w^T x_i - b)\big) \\
&= \arg\max_{w,b} \sum_{i=1}^{n} \ln \sigma\big(z_i(w^T x_i - b)\big) \\
&= \arg\max_{w,b} \ln\Big(\prod_{i=1}^{n} \sigma\big(z_i(w^T x_i - b)\big)\Big) \\
&= \arg\max_{w,b} \ln\Big(\prod_{i=1}^{n} P_{w,b}(z_i|x_i)\Big),
\end{aligned}$$

where $P_{w,b}(z|x)$ represents the probability that the color is of type z (black or white), given that the point coordinate is x.

Finding the optimal parameters (w^*, b^*), which define the decision line ${w^*}^T x - b^* = 0$, are obtained using a maximum likelihood method.

Note that the aforementioned error function, $E(x, z)$, can be also regarded as a cross entropy of the empirical probability of X with the conditional probability of Z, given X, as

$$E(x, z) = \mathbb{E}^{\widehat{p}_X}[-\ln P(Z|X)],$$

with the model probability given by

$$P(z|X = x) = -\ln \sigma\big(z(w^T x_i - b)\big).$$

Gradient descent There is no closed form formulae for w^* and b^*.

The gradient has two components

$$\nabla E = (\nabla_w E, \partial_b E).$$

We have

$$E(x,z) = -\sum_{i=1}^{n} \ln \frac{1}{1+e^{-\left(z_i(w^T x_i - b)\right)}} = \sum_{i=1}^{n} \ln(1+e^{-(z_i(w^T x_i - b))}),$$

$$\begin{aligned}
\nabla_w E &= -\sum_{i=1}^{n} \frac{z_i x_i}{1 + e^{-(z_i w^T x_i - z_i b)}} \\
\partial_b E &= \sum_{i=1}^{n} \frac{z_i}{1 + e^{-(z_i w^T x_i - z_i b)}}.
\end{aligned}$$

The approximation sequence is defined recursively by

$$\begin{aligned}
w^{(j+1)} &= w^{(j)} - \delta \nabla_w E(w^{(j)}, b^{(j)}) \\
b^{(j+1)} &= b^{(j)} - \delta \partial_b E(w^{(j)}, b^{(j)}).
\end{aligned}$$

Other error functions: the number of misclassified points. If m_1 and m_{-1} represent the number of misclassified black and white points, respectively, then the error becomes $E = m_1 + m_{-1}$.

The flaw of this error is its discontinuity with respect to parameters w and b, and hence no differentiable methods would work for it.

The neuron with a continuum input

• The inputs x are considered continuum over the interval $[0, 1]$.

• The weight associated with the value x is given by $\mu(dx)$, where μ is a measure on $[0, 1]$.

• The first half of the computing unit will integrate the inputs with respect to the weighting measure μ, obtaining $\int_0^1 x d\mu(x)$.

• The neuron output is

$$y = \sigma\Big(\int_0^1 x d\mu(x) \Big).$$

Particular case

• The input is a random variable X taking values in $[0, 1]$.

• μ represents the distribution measure of X.

• Note that

$$\mathbb{E}[X] = \int_0^1 x d\mu(x).$$

• the output function depends on its expectation,

$$y = \sigma\big(\mathbb{E}[X]\big).$$

A few measures

Dirac measure

Let $x_0 \in [0,1]$ be fixed and consider the Dirac measure δ_{x_0} sitting at x_0, defined by

$$\delta_{x_0}(A) = \begin{cases} 1, & \text{if } x_0 \in A \\ 0, & \text{if } x_0 \notin A, \end{cases}$$

for any measurable set $A \in \mathcal{B}([0,1])$.

In this case the random variable X takes only the value x_0. The output function in this case is a constant

$$y = \sigma\Big(\int_0^1 x d\delta_{x_0}(x) \Big) = \sigma(x_0).$$

Since there are no parameters to adjust, there is nothing to learn here.

Discrete measures

Let $E = \{x_1, \cdots, x_n\}$ be a finite subset of $[0, 1]$ and consider the discrete measure

$$\mu(A) = \sum_{x_i \in E} w_i \delta_{x_i}(A), \quad \forall A \in \mathcal{B}([0, 1]),$$

where the positive number w_i is the mass attached to the point x_i. The output function is given by

$$y = \sigma\Big(\int_0^1 x \, d\mu(x) \Big) = \sigma\Big(\sum_{j=1}^{n} w_j x_j \Big).$$

This corresponds to the classical case. The learning algorithm in this case adjusts the weights w_j of the possible error function

$$F(w) = \frac{1}{2}\Big(\sigma\big(\sum_{i=0}^{n} w_i x_i\big) - z\Big)^2,$$

z being the desired value of the neuron, given the observation $x^T = (x_1, \cdots, x_n)$.

In the case of m observations $\{(x^k, z^k)\}_{k=1,m}$ the error function takes the form

$$F(w) = \frac{1}{2} \sum_{k=1}^{m} \Big(\sigma\big(\sum_{i=0}^{n} w_i x_i^k \big) - z^k \Big)^2.$$

The optimal weights can be found using the gradient descent algorithm.

Lebesgue measures

• Assume there is a non-negative continuous function p on $[0,1]$ such that $d\mu(x) = p(x)dx$.

• Here dx stands for the Lebesgue measure.

• The output function

$$y = \sigma\Big(\int_0^1 x\, d\mu(x)\Big) = \sigma\Big(\int_0^1 xp(x)\, dx\Big)$$

depends on the weight function $p(x)$.

The learning algorithm has to choose the optimal function $p(x)$, which minimizes the error functional

$$F(p) = \frac{1}{2}\Big(\sigma\Big(\int_0^1 xp(x)\,dx\Big) - z\Big)^2.$$

If μ is a probability measure, i.e. $\mu([0,1]) = 1$, then $p(x)$ is a density function. The optimality problem implies that $p(x)$ is a critical value for the the Lagrange multiplier functional

$$L(p) = \frac{1}{2}\Big(\sigma\Big(\int_0^1 xp(x)\,dx\Big) - z\Big)^2 - \lambda\Big(\int_0^1 p(x)\,dx - 1\Big).$$

In the case of a repeatable experiment, the constant output z is replaced by the target random variable $Z : \Omega \to [0, 1]$, where $(\Omega, \mathcal{H}, \mathbb{P})$ is a probability space. The problem asks for finding the random variable $X : \Omega \to [0, 1]$ such that the amount

$$\mathbb{E}\big[\sigma\Big(\mathbb{E}[X]\Big) - Z\big]^2 = \int_\Omega \big[\sigma\Big(\mathbb{E}[X]\Big) - Z(\omega)\big]^2 \, d\mathbb{P}(\omega)$$

is minimized.

From the theory of random variables a necessary condition for this to be achieved is that

$$\mathbb{E}[Z] = \sigma\Big(\mathbb{E}[X]\Big).$$

Consequently, X is a random variables with the mean

$$\mathbb{E}[X] = \sigma^{-1}\Big(\mathbb{E}[Z]\Big).$$