Mathematical Remarks for Neural Networks

10

The following sections treat mathematical topics that were presupposed in the text (Sect. 10.1 on straight line equations and Sect. 10.2 on regression), or side remarks, which would have disturbed the flow of the exposition (Sect. 10.3 on activation transformation in a Hopfield network).

10.1 Equations for Straight Lines

In this section a few important facts about straight lines and their equations have been collected, which are used in Chap. 3 on threshold logic units. More extensive explanations can be found in any textbook on linear algebra.

Straight lines are commonly described in one of the following forms:

$$\begin{array}{lll} \text{explicit form:} & g \equiv & x_2 = bx_1 + c \\ \text{implicit form:} & g \equiv & a_1 x_1 + a_2 x_2 + d = 0 \\ \text{point-direction form:} & g \equiv & \mathbf{x} = \mathbf{p} + k\mathbf{r} \\ \text{normal form:} & g \equiv & (\mathbf{x} - \mathbf{p})\mathbf{n} = 0 \end{array}$$

with the parameters

b : slope of the line
c : intercept
$\mathbf{p}$: position vector of a point of the line (support vector)
$\mathbf{r}$: direction vector of the line
$\mathbf{n}$: normal vector of the line.

It is a disadvantage of the explicit form that straight lines that are parallel to the x_2-axis cannot be represented. All other forms can represent arbitrary lines.

The implicit form and the normal form are closely related to each other, because the coefficients a_1 and a_2 of the variables x_1 and x_2, respectively, are the coordinates

R. Kruse et al., *Computational Intelligence*, Texts in Computer Science,
DOI 10.1007/978-1-4471-7296-3_10

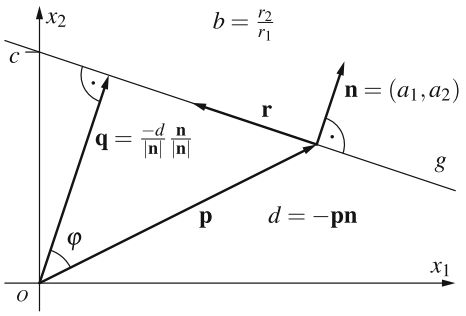

Fig. 10.1 A straight line and the parameters describing it

of a normal vector of the line. That is, we may use $\mathbf{n} = (a_1, a_2)$ in the normal form. Expanding the normal form also shows that $d = -\mathbf{pn}$.

The relations between the parameters of the different forms of stating a straight line are shown in Fig. 10.1. Particularly important is the vector $\mathbf{q}$, which provides an interpretation for the parameter d of the implicit form. The vector $\mathbf{q}$ is obtained by projecting the support vector $\mathbf{p}$ onto the direction normal to the straight line. This is achieved with the scalar product. It is

$$\mathbf{pn} = |\mathbf{p}\,||\mathbf{n}| \cos\varphi.$$

From the diagram we see that $|\mathbf{q}\,| = |\mathbf{p}\,| \cos\varphi$. There fore we have

$$|\mathbf{q}\,| = \frac{|\mathbf{pn}|}{|\mathbf{n}|} = \frac{|d|}{|\mathbf{n}|}.$$

Hence $|d|$ measures the distance of the straight line from the origin of the coordinate system relative to the length of the normal vector. If $\sqrt{a_1^2 + a_2^2} = 1$, that is, if the normal vector has unit length, then $|d|$ measures this distance directly. In this case the normal form is called the **Hessian normal form** of the line equation.

If one takes into account that $\mathbf{pn}$ becomes negative if $\mathbf{n}$ does not point away from the origin (as in the diagram), but toward it, one finally obtains:

$$\mathbf{q} = \frac{\mathbf{pn}}{|\mathbf{n}|} \frac{\mathbf{n}}{|\mathbf{n}|} = \frac{-d}{|\mathbf{n}|} \frac{\mathbf{n}}{|\mathbf{n}|}.$$

Note that $\mathbf{q}$ always points from the origin to the straight line, regardless of whether $\mathbf{n}$ points toward the origin or away from it. Therefore we can read the location of the origin from the sign of d:

$$\begin{aligned} &d = 0 : \text{The straight line contains the origin,} \\ &d < 0 : \mathbf{n} = (a_1, a_2) \text{ points away from the origin,} \\ &d > 0 : \mathbf{n} = (a_1, a_2) \text{ points toward the origin.} \end{aligned}$$

Of course, we can carry out these computations not only for a support vector $\mathbf{p}$ of the straight line, but for an arbitrary vector $\mathbf{x}$ (see Fig. 10.2). Thus we obtain a vector $\mathbf{z}$ that is the projection of the vector $\mathbf{x}$ onto the direction normal to the line.

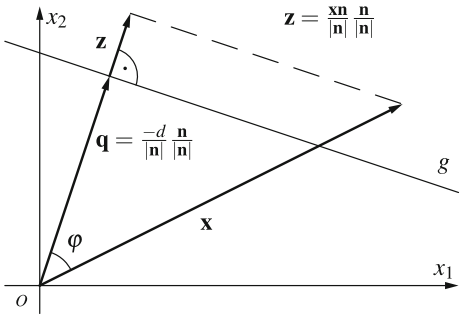

Fig. 10.2 Determining the side of straight line on which a point **x** lies

By comparing this vector to the vector **q** considered above, we can determine on which side of the straight line the point lies that has the position vector **x**:

A point with position vector **x** *lies on the side of the straight line to which the normal vector* **n** *points, if* $\mathbf{xn} > -d$, *and on the other side, if* $\mathbf{xn} < -d$. *If* $\mathbf{xn} = -d$, *the point lies on the straight line.*

It should be clear that these considerations are not restricted to straight lines, but can be transferred immediately to planes and hyperplanes. Thus we can easily determine for them as well on which side a point with given position vector lies.

10.2 Regression

This section recalls the **method of least squares**, also known as **regression**, which is well known in calculus and statistics. It is used to determine best fit straight lines (regression lines) and generally best fit polynomials (regression polynomials). The following exposition follows mainly (Heuser 1988).

(Physical) measurement data rarely show the exact relationship of the measured quantities as it is described by physical laws, since they are inevitably afflicted by errors. If one wants to determine the relationship of the quantities nevertheless (at least approximately), one faces the task to find a function that fits the measurement points as well as possible, so that the measurement errors are somehow "balanced." Naturally, in order to achieve this, we should have a hypothesis about the type of relationship, so that we can choose a function class and thus reduce the problem to the selection of the parameters of a function of a specific type.

For example, if we expect two quantities x and y to exhibit a linear dependence (for instance, because a scatter plot of the measurement points suggests such a relationship), we have to determine the parameters a and b of the straight line $y = g(x) = a + bx$. However, due to the inevitable measurement errors it will generally not be possible to find a straight line in such a way that all n given measurement points (x_i, y_i), $1 \leq i \leq n$, lie exactly on this straight line. Rather we have to try to find a straight line that deviates from the measurement points as little as possible. Therefore it is plausible to determine the parameters a and b in such a way that the

sum of the squared differences

$$F(a,b) = \sum_{i=1}^{n} (g(x_i) - y_i)^2 = \sum_{i=1}^{n} (a + bx_i - y_i)^2$$

is minimized. That is, the y-values that can be computed from the line equation should deviate (in total) as little as possible from the measured values. The reasons for choosing the squared deviations are basically the same as those given in Sect. 4.3: in the first place using squares makes the error functions continuously differentiable everywhere. In contrast to this, the derivative of the absolute value, which would be an obvious alternative, does not exist/is not continuous at 0. Secondly, squaring the deviations weights large deviations more heavily than small ones, so that there is a tendency to avoid individual large deviations from the measured data.[1]

A necessary condition for a minimum of the error function $F(a, b)$ defined above is that the partial derivatives of this function w.r.t. the parameters a and b vanish:

$$\frac{\partial F}{\partial a} = \sum_{i=1}^{n} 2(a + bx_i - y_i) = 0 \quad \text{and}$$
$$\frac{\partial F}{\partial b} = \sum_{i=1}^{n} 2(a + bx_i - y_i)x_i = 0.$$

From these equations we obtain, after a few simple transformations, the so-called **normal equations**

$$na + \left(\sum_{i=1}^{n} x_i\right) b = \sum_{i=1}^{n} y_i$$
$$\left(\sum_{i=1}^{n} x_i\right) a + \left(\sum_{i=1}^{n} x_i^2\right) b = \sum_{i=1}^{n} x_i y_i,$$

that is, a linear equation system with two equations and two unknowns a and b. It can be shown that this equation system has a unique solution unless the x-values of all measurement points are identical (that is, unless $x_1 = x_2 = \ldots = x_n$), and that this solution indeed describes a minimum of the function F (Heuser 1988). The straight line $y = g(x) = a + bx$ determined in this way is called the **best fit (straight) line** or the **regression line** for the data set $(x_1, y_1), \ldots, (x_n, y_n)$.

To illustrate the procedure, we consider a simple example. Let the data set consisting of eight data points $(x_1, y_1), \ldots, (x_8, y_8)$ be given that is shown in the following table (Heuser 1988) (see also Fig. 10.3):

[1] Note, however, that this can also be a disadvantage. If the data set contains "outliers" (that is, measurement values that due to random, disproportionally large measurement errors deviate strongly from the true value), the position of the regression line may be influenced heavily by very few measurement points (precisely the outliers), which can lead to an unusable result.

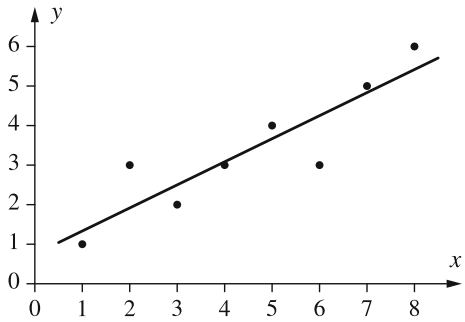

Fig. 10.3 An example data set and a regression line that was computed with the method of least squares

x	1	2	3	4	5	6	7	8
y	1	3	2	3	4	3	5	6

In order to set up the system of normal equations, we compute

$$\sum_{i=1}^{8} x_i = 36, \qquad \sum_{i=1}^{8} x_i^2 = 204, \qquad \sum_{i=1}^{8} y_i = 27, \qquad \sum_{i=1}^{8} x_i y_i = 146.$$

Thus we obtain the equation system (normal equations)

$$\begin{aligned} 8a + 36b &= 27, \\ 36a + 204b &= 146, \end{aligned}$$

which possesses the solution $a = \frac{3}{4}$ and $b = \frac{7}{12}$. Therefore the regression line is

$$y = \frac{3}{4} + \frac{7}{12}x.$$

This line is shown, together with the data points we started from, in Fig. 10.3.

The method we just considered is, of course, not limited to straight lines, but can be extended at least to polynomials. In this case one tries to find a polynomial

$$y = p(x) = a_0 + a_1 x + \ldots + a_m x^m$$

with a given, fixed degree m that approximates the n data points $(x_1, y_1), \ldots, (x_n, y_n)$ as well as possible. In this case we have to minimize

$$F(a_0, a_1, \ldots, a_m) \;=\; \sum_{i=1}^{n} (p(x_i) - y_i)^2 \;=\; \sum_{i=1}^{n} (a_0 + a_1 x_i + \ldots + a_m x_i^m - y_i)^2.$$

Necessary conditions for a minimum are again that the partial derivatives w.r.t. the parameters a_0 to a_m vanish, that is,

$$\frac{\partial F}{\partial a_0} = 0, \quad \frac{\partial F}{\partial a_1} = 0, \quad \ldots, \quad \frac{\partial F}{\partial a_m} = 0.$$

In this way we obtain the system of normal equations (Heuser 1988)

$$\begin{aligned}
na_0 + \left(\sum_{i=1}^{n} x_i\right) a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^m\right) a_m &= \sum_{i=1}^{n} y_i \\
\left(\sum_{i=1}^{n} x_i\right) a_0 + \left(\sum_{i=1}^{n} x_i^2\right) a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^{m+1}\right) a_m &= \sum_{i=1}^{n} x_i y_i \\
\vdots \qquad\qquad \vdots \qquad\qquad\qquad \vdots \qquad\qquad \vdots & \\
\left(\sum_{i=1}^{n} x_i^m\right) a_0 + \left(\sum_{i=1}^{n} x_i^{m+1}\right) a_1 + \ldots + \left(\sum_{i=1}^{n} x_i^{2m}\right) a_m &= \sum_{i=1}^{n} x_i^m y_i,
\end{aligned}$$

from which the parameters a_0 to a_m can be derived with the usual methods of linear algebra (Gaussian elimination, Cramer's rule, inverting the coefficient matrix, etc.). The resulting polynomial $p(x) = a_0 + a_1 x + a_2 x^2 + \ldots + a_m x^m$ is called **best fit polynomial** or **regression polynomial** of degree m for the data set $(x_1, y_1), \ldots, (x_n, y_n)$.

Furthermore the method of least squares cannot only be used, as considered up to now, to compute regression polynomials, but may as well be employed to fit functions with more than one argument. This case is called **multiple** or **multivariate regression**. We consider, as an example, only the special case of **multilinear regression** and confine ourselves to a function with two arguments. That is, we consider, how one can find a best fitting function of the form

$$z = f(x, y) = a + bx + cy$$

for a given data set $(x_1, y_1, z_1), \ldots, (x_n, y_n, z_n)$ in such a way that the sum of squared errors is minimized. In this case the normal equations are derived in a perfectly analogous way. We have to minimize

$$F(a, b, c) = \sum_{i=1}^{n} (f(x_i, y_i) - z_i)^2 = \sum_{i=1}^{n} (a + bx_i + cy_i - z_i)^2.$$

Necessary conditions for a minimum are

$$\begin{aligned}
\frac{\partial F}{\partial a} &= \sum_{i=1}^{n} 2(a + bx_i + cy_i - z_i) &&= 0, \\
\frac{\partial F}{\partial b} &= \sum_{i=1}^{n} 2(a + bx_i + cy_i - z_i) x_i &&= 0, \\
\frac{\partial F}{\partial c} &= \sum_{i=1}^{n} 2(a + bx_i + cy_i - z_i) y_i &&= 0.
\end{aligned}$$

Therefore we obtain the system of normal equations

$$\begin{aligned}
na &+ \left(\sum_{i=1}^{n} x_i\right) b &+ \left(\sum_{i=1}^{n} y_i\right) c &= \sum_{i=1}^{n} z_i \\
\left(\sum_{i=1}^{n} x_i\right) a &+ \left(\sum_{i=1}^{n} x_i^2\right) b &+ \left(\sum_{i=1}^{n} x_i y_i\right) c &= \sum_{i=1}^{n} z_i x_i \\
\left(\sum_{i=1}^{n} y_i\right) a &+ \left(\sum_{i=1}^{n} x_i y_i\right) b &+ \left(\sum_{i=1}^{n} y_i^2\right) c &= \sum_{i=1}^{n} z_i y_i
\end{aligned}$$

from which a, b and c can easily be computed.

It should be immediately clear that the method of least squares can also be extended to polynomials in multiple variables. How it may also be extended, under certain conditions, to other function classes is demonstrated in Sect. 5.3 with the help of the example of **logistic regression**.

A program for multivariate polynomial regression that uses ideas from dynamic programming to quickly compute the needed power products can be found at

`http://www.borgelt.net/regress.html`

10.3 Activation Transformation

In this section we demonstrate how the weights and thresholds of a Hopfield network that works with activations 0 and 1 can be transformed into the corresponding parameters of a Hopfield network that works with the activations -1 and $+1$ (and vice versa). This shows that the two network types are essentially equivalent, and thus that it was justified to choose in Chap. 8 whichever form was more suitable for the specific task under consideration.

In the following we indicate by an upper index of the considered quantities what the range of activation values of the neural network is, to which they refer:

$$\begin{aligned}
^{0} &: \text{quantity of a network with } \mathrm{act}_u \in \{\ \ 0, 1\}, \\
^{-} &: \text{quantity of a network with } \mathrm{act}_u \in \{-1, 1\}.
\end{aligned}$$

Clearly we must always have

$$\mathrm{act}_u^0 = \frac{1}{2}(\mathrm{act}_u^- + 1) \qquad \text{and} \qquad \mathrm{act}_u^- = 2\mathrm{act}_u^0 - 1.$$

That is, the neuron u either has activation 1 in both networks or it has activation 0 in one network and activation -1 in the other. In order to achieve that both network types exhibit the same behavior, it must also hold that:

$$s(\mathrm{net}_u^- - \theta_u^-) = s(\mathrm{net}_u^0 - \theta_u^0),$$

where

$$s(x) = \begin{cases} 1, \text{ if } x \geq 0, \\ -1, \text{ otherwise.} \end{cases}$$

Only if this is the case the activation changes are the same in both networks. The above equation clearly holds if

$$\text{net}_u^- - \theta_u^- = \text{net}_u^0 - \theta_u^0.$$

(Note that this is a sufficient, but not a necessary condition.) Using the relations between the activations stated above, we obtain from this equation

$$\begin{aligned} \text{net}_u^- - \theta_u^- &= \sum_{v \in U - \{u\}} w_{uv}^- \text{act}_u^- - \theta_u^- \\ &= \sum_{v \in U - \{u\}} w_{uv}^- (2\text{act}_u^0 - 1) - \theta_u^- \\ &= \sum_{v \in U - \{u\}} 2w_{uv}^- \text{act}_u^0 - \sum_{v \in U - \{u\}} w_{uv}^- - \theta_u^- \\ &\stackrel{!}{=} \text{net}_u^0 - \theta_u^0 \\ &= \sum_{v \in U - \{u\}} w_{uv}^0 \text{act}_u^0 - \theta_u^0 \end{aligned}$$

This equation holds if we choose

$$\begin{aligned} w_{uv}^0 &= 2w_{uv}^- \qquad \text{and} \\ \theta_u^0 &= \theta_u^- + \sum_{v \in U - \{u\}} w_{uv}^-. \end{aligned}$$

For the opposite direction we obtain

$$\begin{aligned} w_{uv}^- &= \frac{1}{2} w_{uv}^0 \qquad \text{and} \\ \theta_u^- &= \theta_u^0 - \sum_{v \in U - \{u\}} w_{uv}^- \;=\; \theta_u^0 - \frac{1}{2} \sum_{v \in U - \{u\}} w_{uv}^0. \end{aligned}$$

Reference

H. Heuser, *Lehrbuch der Analysis, Teil 1+2* (Teubner, Stuttgart, Germany, 1988)