J. Appl. Math. & Informatics Vol. **37***(2019), No. 3 - 4, pp. 271 - 279*
https://doi.org/10.14317/jami.2019.271

FEEDFORWARD NEURAL NETWORKS AND SEPARATION OF GEOMETRIC REGIONS

KYEONGSU PARK

Abstract. We investigate how a feedforward neural network works to separate a geometric region from its complement. Our investigations are restricted to regions in $\mathbb{R}$ or $\mathbb{R}^2$ including an interval, a triangular region, a disk and the union of two disjoint disks. We also examine what happens at each layer of the network.

AMS Mathematics Subject Classification : 68T05.
Key words and phrases : machine learning, feedforward neural network

1. Introduction

A *feedforward neural network* [7] is an artificial neural network represented by a sequence of functions

$$\mathbb{R}^{d_0} \xrightarrow{F^{(1)}} \mathbb{R}^{d_1} \xrightarrow{G^{(1)}} \mathbb{R}^{d_1} \xrightarrow{F^{(2)}} \cdots \xrightarrow{G^{(r)}} \mathbb{R}^{d_r}.$$

The function $F^{(i)}$ is an affine transformation defined by

$$F^{(i)}(x) = b^{(i)} + W^{(i)}x \tag{1}$$

for a *bias vector* $b^{(i)}$ and a *weight matrix* $W^{(i)}$. The function $G^{(i)}$ is called an *activation function.* In many cases, it acts coordinatewise, that is, it has the form

$$G^{(i)}(x_1, \ldots, x_{d_i}) = (g(x_1), \ldots, g(x_{d_i}))$$

for a monotonically increasing smooth function g, which is also called the activation function. A subsequence

$$\mathbb{R}^{d_i} \xrightarrow{G^{(i)}} \mathbb{R}^{d_i}$$

is called an *internal layer* or a *hidden layer.*

For an *input vector* $x \in \mathbb{R}^{d_0}$, the vector

$$y^{(i)} = (G^{(i)} \circ \cdots \circ F^{(0)})(x) \in \mathbb{R}^{d_i}$$

Received February 12, 2019. Revised March 16, 2019. Accepted March 26, 2019.

is called the *i-th output vector*. The final output vector $y^{(r)}$ is called simply the output vector.

A measure of difference between an output vector and the corresponding target vector is called the *loss*. Various losses are introduced in literatures.

For example, the *L2* loss function is defined by

$$J(x) = \frac{1}{2}\|t - y^{(r)}\|^2$$

where t is the target vector corresponding to x. For input vectors $x_1, \ldots, x_p$, target vectors $t_1, \ldots, t_p$ and corresponding output vectors $y_1, \ldots, y_p$, the sum

$$\sum_{j=1}^{p} J(x_j) = \frac{1}{2}\sum_{j=1}^{p} \|t_j - y_j\|^2$$

is called the *total loss*.

The main problem of this area is to determine biases and weights which minimize the total loss locally [3]. To do it is called the *optimization*. An error backpropagation method is commonly used for this [5, 6]. The method works successfully in general, but processes are hard to understand.

In this paper, the separation of a region means that the region and its complement are classified by a network. It is expected that the separation of a geometric region would make intermediate processes more clear and understandable. We will separate some regions in $\mathbb{R}^n$, $n = 1, 2$.

2. Dividing the real line

A frequently used activation function is the logistic sigmoid function [1]

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

which maps the real line $\mathbb{R}$ onto the unit interval $(0, 1)$. The real line is divided into $\sigma^{-1}((0, 0.5))$ and $\sigma^{-1}([0.5, 1))$, which are two intervals $(-\infty, 0)$ and $[0, \infty)$, respectively.

The feedforward neural network

$$\mathbb{R} \xrightarrow{(-aw,w)} \mathbb{R} \xrightarrow{\sigma} \mathbb{R} \tag{2}$$

separates the interval $[a, \infty)$ from its complement for any $w > 0$. That is, outputs of the network for inputs in $(-\infty, a)$ and $[a, \infty)$ is contained in $(-1, 0.5)$ and $[0.5, 1)$ respectively.

We tested in the case $a = 0$. Some real numbers $x_1, \ldots, x_p$ were taken as input vectors. We obtained output vectors $y_j = \sigma(wx_j - wa)$ for all j. Then $y_j \geq 0.5$ if and only if $x_j \geq a$.

To compute the loss, we took target vectors as

$$t_j = \begin{cases} 0, & \text{if } x_j < a \\ 1, & \text{otherwise} \end{cases}$$

for all j. The larger w is, the smaller total loss $\frac{1}{2}\sum_{j=1}^{p}\|t_j - y_j\|^2$ is. Hence we obtain desired total loss by taking a sufficiently large weight w.

Now we separate the closed interval $[a,b]$. To do it, we construct a network of the form

$$\mathbb{R} \xrightarrow{F^{(1)}} \mathbb{R}^2 \xrightarrow{\sigma} \mathbb{R}^2 \xrightarrow{F^{(2)}} \mathbb{R} \xrightarrow{\sigma} \mathbb{R}$$

where σ acts coordinatewise. Applying the gradient descent iteration with some initial affine transformations $F^{(1)}$ and $F^{(2)}$, we obtain final affine transformations which optimize the network to the desired total loss.

Similar to the network (2), initial biases and weights are taken as

$$b^{(1)} = \begin{bmatrix} -aw \\ bw \end{bmatrix},\ W^{(1)} = \begin{bmatrix} w \\ -w \end{bmatrix},\ b^{(2)} = \begin{bmatrix} \lambda \end{bmatrix},\ W^{(2)} = \begin{bmatrix} \mu & \mu \end{bmatrix} \tag{3}$$

for some nonzero w, λ and μ. Two functions $F^{(1)}$ and $F^{(2)}$ are defined as (1), that is, $F^{(1)}(x) = (-aw + wx, bw - wx)$ and $F^{(2)}(x,y) = -\lambda + \mu x + \mu y$.

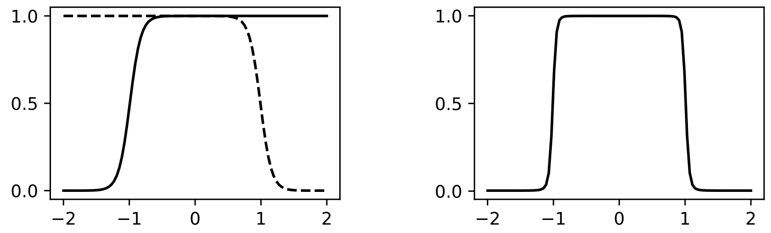
The case $a = -1$ and $b = 1$ was implemented. 100 evenly spaced points in the interval $[-2,2]$ was taken as input vectors. Applying gradient descent iterations starting at $w = 1$, $\lambda = -2$ and $\mu = 1$, biases and weights reached approximately

$$b^{(1)} = \begin{bmatrix} 11.8 \\ 11.8 \end{bmatrix},\ W^{(1)} = \begin{bmatrix} 11.9 \\ -11.9 \end{bmatrix},\ b^{(2)} = \begin{bmatrix} -18.8 \end{bmatrix},\ W^{(2)} = \begin{bmatrix} 13.0 & 13.0 \end{bmatrix}.$$

Graphs of outputs are displayed in Figure 1. Passing through $F^{(1)}$ and the first internal layer, $[-1,\infty)$ and $(-\infty,1]$ are almost separated from their complements, respectively. If we set $(y_1,y_2) = (G^{(1)} \circ F^{(1)})(-1)$, then

$$F^{(2)}(y_1,y_2) = -18.8 + 13.0\,y_1 + 13.0\,y_2$$

is approximately zero. A similar equation holds for the input 1. In the end, $[-1,1]$ is separated from its complement.



(A) the graphs of the first outputs
solid line: the 1st coordinate $y_1^{(1)}$
dashed line: the 2nd coordinate $y_2^{(1)}$

(B) the graph of the final output $y^{(2)}$ as a function of inputs

FIGURE 1. The outputs of a network which separates $[-1,1]$

In general, any n disjoint bounded intervals can be separated. A network is of the form

$$\mathbb{R} \xrightarrow{F^{(1)}} \mathbb{R}^{2n} \xrightarrow{\sigma} \mathbb{R}^{2n} \xrightarrow{F^{(2)}} \mathbb{R} \xrightarrow{\sigma} \mathbb{R}.$$

Initial affine transformations $F^{(1)}$ and $F^{(2)}$ can be taken analogously to the equation (3).

3. Regions in the plane

It is analogous to the above section to construct a network which separates a half plane in $\mathbb{R}^2$. The bias vector and the weight matrix are determined by the equation of its boundary. A strip bounded by two parallel lines can also be separated similarly to the case of a interval.

A *corner plane* is a subset of $\mathbb{R}^2$ bounded by two rays with a common vertex. A network which separates a corner plane is of the form

$$\mathbb{R}^2 \xrightarrow{F^{(1)}} \mathbb{R}^2 \xrightarrow{\sigma} \mathbb{R}^2 \xrightarrow{F^{(2)}} \mathbb{R} \xrightarrow{\sigma} \mathbb{R}.$$

The initial function $F^{(1)}$ can be taken from the equation of rays, and $F^{(2)}$ randomly. Applying gradient descent iterations we may obtain a desired network.

In implementation, a network that separates the first quadrant was built. Since its boundary is composed of two lines $x = 0$ and $y = 0$, initial biases and weights were taken as

$$b^{(1)} = \begin{bmatrix} 0 \\ 0 \end{bmatrix}, \ W^{(1)} = \begin{bmatrix} 1.5 & 0 \\ 0 & 1.5 \end{bmatrix}, \ b^{(2)} = \begin{bmatrix} -2 \end{bmatrix}, \ W^{(2)} = \begin{bmatrix} 1 & 1 \end{bmatrix}.$$

$b^{(2)}$ and $W^{(2)}$ could be taken at random, but the above values were obtained from experiments. Evenly spaced 30×30 points in the square $[-1, 1] \times [-1, 1]$ were taken as input vectors.

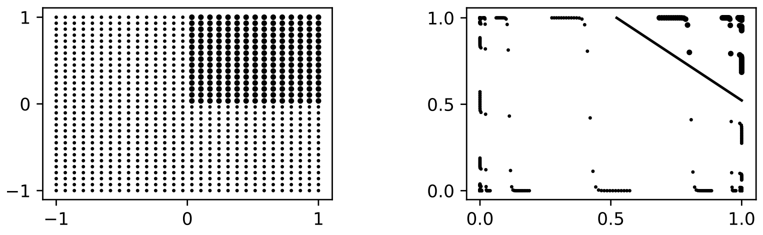
After several iterations, biases and weights reached approximately

$$b^{(1)} = \begin{bmatrix} 0.5 \\ 0.5 \end{bmatrix}, \ W^{(1)} = \begin{bmatrix} 25.4 & -0.6 \\ -0.6 & 25.4 \end{bmatrix}, \ b^{(2)} = \begin{bmatrix} -24.4 \end{bmatrix}, \ W^{(2)} = \begin{bmatrix} 16.0 & 16.0 \end{bmatrix}.$$

The process of separating the first quadrant is displayed in Figure 2. The function $G^{(1)} \circ F^{(1)}$ expands the spacing near the coordinate axes. Then a line is taken by $F^{(2)}$ to separate first outputs.

The network does not completely separate the first quadrant because it depends only on input vectors and target vectors. For example, it maps the point $(1/60, 1/60)$ to 0.21, which is less than 0.5.

The *cross entropy* [2] was tested as a loss function instead of the L2 loss function. The intermediate process and result were not so different. The rectifier ReLU [4] was also tested as an activation function instead of the logistic sigmoid function. The first quadrant was also separated, but intermediate processes were very different. The rectifier seemed more appropriate in this case.



(A) input vectors
big dots: the 1st quadrant

(B) first outputs $y^{(1)}$
big dots: the image of the 1st quadrant
solid line: $F^{(2)}(y^{(1)}) = 0$

Figure 2. Inputs and First outputs of a network which separates the first quadrant

Now we separate a triangular region and a circular region from their complements, respectively. The network is of the form

$$\mathbb{R}^2 \xrightarrow{F^{(1)}} \mathbb{R}^3 \xrightarrow{\sigma} \mathbb{R}^3 \xrightarrow{F^{(2)}} \mathbb{R} \xrightarrow{\sigma} \mathbb{R}. \tag{4}$$

The initial function $F^{(1)}$ can be taken from the equation of the triangle. We may take the same initial function for a circular region if the boundary circle circumscribes the triangle.

In implementation, a network that separates the triangular region bounded by three lines

$$1 - \sqrt{3}\,x - y = 0,\ 1 + \sqrt{3}\,x - y = 0,\ 1 + 2\,y = 0$$

was built. And another network from the same $F^{(1)}$ and $F^{(2)}$ to separate the unit disk was also built because the unit circle circumscribes the triangular region. Evenly spaced 60×60 points in the square $[-2, 2] \times [-2, 2]$ was taken as input vectors. Initial biases and weights were taken as

$$b^{(1)} = \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix},\ W^{(1)} = \begin{bmatrix} -\sqrt{3} & -1 \\ \sqrt{3} & -1 \\ 0 & 2 \end{bmatrix},\ b^{(2)} = \begin{bmatrix} -3 \end{bmatrix},\ W^{(2)} = \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

After several iterations, biases and weights were reached final values with the desired total loss.

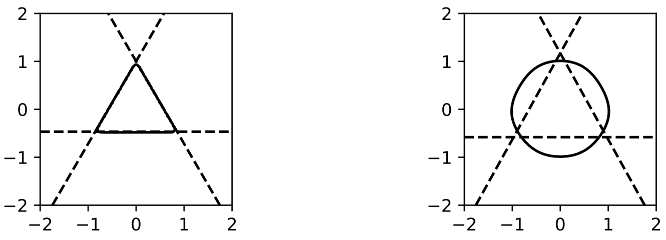
$$b^{(1)} = \begin{bmatrix} 20.6 \\ 20.6 \\ 16.0 \end{bmatrix},\ W^{(1)} = \begin{bmatrix} -35.4 & -20.6 \\ 35.4 & -20.6 \\ 0.0 & 34.3 \end{bmatrix},$$
$$b^{(2)} = \begin{bmatrix} -49.0 \end{bmatrix},\ W^{(2)} = \begin{bmatrix} 21.2 & 21.2 & 16.2 \end{bmatrix},$$

for the triangular region and

$$b^{(1)} = \begin{bmatrix} 2.2 \\ 2.2 \\ 2.4 \end{bmatrix}, \quad W^{(1)} = \begin{bmatrix} -3.4 & -1.9 \\ 3.4 & -1.9 \\ 0.0 & 4.2 \end{bmatrix},$$
$$b^{(2)} = \begin{bmatrix} -74.7 \end{bmatrix}, \; W^{(2)} = \begin{bmatrix} 35.3 & 35.3 & 34.1 \end{bmatrix}.$$

for the unit disk were obtained approximately.

The level curves of the two networks at level 0.5 are shown in the Figure 3. A *level curve* of a network at level k is the inverse image of k, that is, the set of input vectors whose output vector is k. The level curve for the triangular region is a round triangle, and that for circular one is nearly round.



(A) separation of a triangular region solid line: the level curve at level 0.5 dashed line: three lines $F^{(1)}(x) = 0$

(B) separation of a circular region solid line: the level curve at level 0.5 dashed line: three lines $F^{(1)}(x) = 0$

Figure 3. Separation of a triangular region and a circular region

The rectifier ReLU can be used as an activation function. For the triangular region, the result is similar to that of the logistic sigmoid function. However, for unit disks, the level curve at level 0.5 is hexagonal. The rectifier seems to be suitable for the region whose boundary is piecewise linear.

In general, a polygonal region can be separated from its complement. The network is similar to the sequence (4). Since we use boundary equations to define the initial $F^{(1)}$, the dimension of the first internal layer depends on the number of sides of the boundary polygon. The interior region of a regular square was tested. The interior region of the circle that circumscribes it was also tested. The results were similar to those for the triangular region.

In separation of a circular region using a polygon, the more sides it has, the better the circular region is separated. It would be natural.

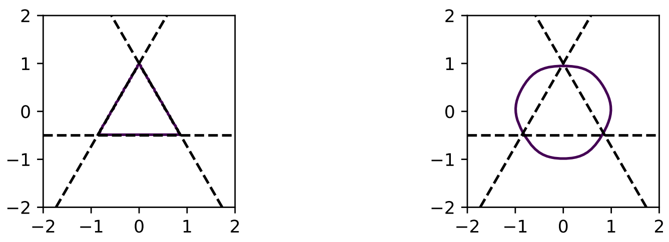
An elliptical region can be separated because it is an image of a circular region under an affine transformation and $F^{(1)}$ is an affine transformation.

The triangle in Figure 3(B) has small portions outside the unit circle. It is wonder if the triangle can be inscribed in the unit circle. It was investigated whether the network with $F^{(1)}$ and $F^{(2)}$ in the form

$$b^{(1)} = \lambda \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix}, \; W^{(1)} = \lambda \begin{bmatrix} -\sqrt{3} & -1 \\ \sqrt{3} & -1 \\ 0 & 2 \end{bmatrix}, \; b^{(2)} = \mu \begin{bmatrix} 1 \end{bmatrix}, \; W^{(2)} = \nu \begin{bmatrix} 1 & 1 & 1 \end{bmatrix}.$$

also separates the unit disk for some λ, μ and ν.

The network was optimized only for three variables λ, μ and ν instead of whole variables. $(\lambda, \mu, \nu) = (43.8, -44.8, 17.7)$ was obtained as optimized values for the triangular region and $(2.6, 84.9, 41.1)$ for the unit disk. In Figure 4 the level curves at level 0.5 and three lines determined by the equation $F^{(1)}(x) = 0$ are shown in both cases.



(A) $(\lambda, \mu, \nu) = (43.8, -44.8, 17.7)$
solid line: the level curve at level 0.5,
dashed line: three lines $F^{(1)}(x) = 0$

(B) $(\lambda, \mu, \nu) = (2.6, -84.9, 41.1)$
solid line: the level curve at level 0.5,
dashed line: three lines $F^{(1)}(x) = 0$

FIGURE 4. Parametrization of networks

By varying the parameters along the line segment from $(43.8, -44.8, 17.7)$ to $(2.6, 84.9, 41.1)$, the region varies from a triangular one to a circular one. Most of them are round triangles.

Networks mentioned in this section so far separate a convex region. Now we build a network which separates the union of two disjoint disks

$$L = \{(x, y) | (x + 1)^2 + y^2 \leq 0.5\}, \; R = \{(x, y) | (x - 1)^2 + y^2 \leq 0.5\}.$$

Since a circular region can be separated using a polygon, a network may be built from two polygons. But we want to build a network as simple as possible. After several attempts, the following network is obtained:

$$\mathbb{R}^2 \xrightarrow{F^{(1)}} \mathbb{R}^4 \xrightarrow{\sigma} \mathbb{R}^4 \xrightarrow{F^{(2)}} \mathbb{R}^3 \xrightarrow{\tau} \mathbb{R}^3 \tag{5}$$

where τ is the softmax function. The cross entropy is used as a loss function.

In implementation, evenly spaced 60×40 points in the square $[-2.3, 2.3] \times [-1.4, 1.4]$ were taken as input vectors. Target vectors were taken by $(1, 0, 0)$, $(0, 1, 0)$ and $(0, 0, 1)$ in L, R and $\mathbb{R}^2 - L \cup R$, respectively. After a few trials at random initial values, biases and weights reached

$$b^{(1)}=\begin{bmatrix}-0.01\\-0.01\\-6.37\\-6.37\end{bmatrix},\ W^{(1)}=\begin{bmatrix}1.80 & 3.28\\-1.80 & 3.28\\-4.39 & 0.02\\4.39 & 0.02\end{bmatrix},$$

$$b^{(2)}=\begin{bmatrix}-5.00\\-5.00\\9.95\end{bmatrix},\ W^{(2)}=\begin{bmatrix}-63.32 & 62.57 & -26.55 & -1.63\\62.40 & -63.16 & -1.57 & -27.17\\-0.25 & -0.08 & 29.22 & 28.59\end{bmatrix}.$$

The level curve of the third coordinate $y_3^{(2)}$ of the second output at level 0.5, and four lines determined by the equation $F^{(1)}(x) = 0$ are shown in Figure 5. It is similar to the double of Figure 3(B).

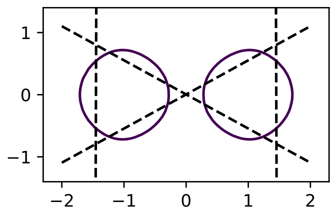


FIGURE 5. Separation of the union of two disjoint disks
solid line: the level curve of $y_3^{(2)}$ at level 0.5
dashed line: four lines $F^{(1)}(x) = 0$

The two level curves of $y_1^{(2)}$ and $y_2^{(2)}$ at level 0.5 are approximately the same as the left and right circles in Figure 5, respectively.

The dimension of the last internal layer can be taken as 1, that is, the targets are 1 and 0 in $L \cup R$ and $\mathbb{R}^2 - L \cup R$, respectively. In this case, the dimension of the first internal layer should be at least 5.

The L2 loss function does not work well in the network (5). However, if the dimension of the first internal layer is 5, we may find a well-working network. If the rectifier is taken as an activation function of the first internal layer, the dimension of the first internal layer should be at least 6 because level curves are polygonal.

4. Conclusion

To separate a region, a network expand the spacing near the boundary of the region, and then cut output data with a hyperplane. Generally, a region bounded by line segments can be separated in $\mathbb{R}^2$. The dimension of the first internal layer is the same as the number of sides of the boundary polygon. A circular region is also separated well. Dimensions are subtle but require less than expected.

References

1. D. Barber, *Bayesian Reasoning and Machine Learning*, Cambridge University Press, 2012.
2. P.T. de Boer, S. Mannor, R.Y. Rubinstein, *A Tutorial on the Cross-Entropy Method*, Annals of Operation Research **134** (2005), 19-67.
3. G.W. Flake and B.A. Pearlmuter, *Differentiating functions of the jacobian with respect to the weights*, in Advances in Neural Information Processing Systems, Vol. 12, The MIT Press, 2000.
4. R. Hahnloser, R. Sarpeshkar, M.A. Mahowald, R.J. Douglas, H.S. Seung, *Digital selection and analogue amplification coexist in a cortex-inspired silicon circuit*, Nature **405** (2000), 947-951.
5. D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning Internal Representations by Error Propagation* in Parallel Distributed Processing: Explorations in the Microstructure of Cognition. Vol. 1:Foundations, MIT Press, 1986.
6. D.E. Rumelhart, G.E. Hinton and R.J. Williams, *Learning representations by back-propagating errors*, Nature **323** (1986), 533-536.
7. J. Schmidhuber, *Deep learning in neural networks: An overview*, Neural Networks **61** (2015), 85-117.

Kyeongsu Park received M.Sc. and Ph.D. from Seoul National University. He is a professor at Jeonju University since 1998. His research interests include affine manifold, computational geometry and machine learning.

Department of Game Contents, Jeonju University, Jeonju 55069, Korea.
e-mail: pine@jj.ac.kr