

The Logic of Neural Networks

J.L. Castro¹ and E. Trillas²

¹Dpto. Ciencias Comp. e I.A., ETSI Informática, Univ. Granada.
Avda. Andalucía 38. 18071 Granada, Spain.

² Dpto. Inteligencia Art. Fac. Informática Univ. Politécnica Madrid
Campus Montegancedo, 28660 Boadilla del Monte, Madrid, Spain.
e-mail: castro@robinson.ugr.es - trillas@fi.upc.es

Abstract

This paper establishes the equivalence between multilayer feedforward networks and linear combinations of Lukasiewicz propositions. In this sense, multilayer forward networks have a logic interpretation, which should permit to apply logical techniques in the neural networks framework.

Keywords. Feedforward Networks, Lukasiewicz Logic, Universal Approximators, Squashing functions.

1 Introduction

In the last several years researches have begun to explore the ability of multilayer feedforward networks to approximate general mappings. Recently, this research has virtually exploded with impressive successes across a wide variety of applications. The scope of these applications is too broad to mention useful specifics here; the interested reader is referred to the proceeding of recent conference IEEE Conferences on Neural Networks for a sampling of examples [6-13].

The ability of sufficiently elaborate feedforward networks to approximate quite well nearly any function has been established in a very specific and satisfying sense [4,5]. Nevertheless, neural networks are until now black boxes whose mechanism has not a logic explication. In this way, in [15] Williams gives a logic interpretation of activation functions. In this paper we will give a first step in order to obtain a general explication of the neural network mechanism.

Before enunciating the result we will give definitions and notation which enable us to speak about the class of multilayer feedforward networks under consideration.

Let RN be a single hidden layer feedforward networks with squashing at the hidden layer and no squashing at the output layer. Remember that a function $\psi : \mathbb{R} \rightarrow [0, 1]$ is a squashing function if it is non-decreasing, $\lim_{\lambda \rightarrow \infty} \Psi(\lambda) = 1$, and $\lim_{\lambda \rightarrow -\infty} \Psi(\lambda) = 0$.

The output function of RN will be

$$f(x) = \sum_{j=1}^k \beta_j \psi \left(b_j + \sum_{i=1}^n \omega_i^j x_i \right). \quad (1)$$

where ψ is a squashing function, $x = (x)_i$ corresponds to network input, w_i^j corresponds to networks weights from input to intermediate layer, b_j corresponds to a bias, and the scalars β_j correspond to network weights from hidden to output layer (see fig. 1).

Fig.1. Neural Networks

Hornik et al. [5] established that fixed ψ , this kind of neural networks are universal approximators, that is, they are capable of approximating any Borel measurable function and any continuous function from one finite dimensional space to another to any desired degree of accuracy, provided sufficiently many hidden units are available. Concretely, they proved that the class $\sum^n(\psi)$ of functions defined by (I) is uniformly dense on compacta in the set C^n of all the continuous function from \mathbb{R}^n to \mathbb{R} , and fixed a probability measure μ on $(\mathbb{R}^n, \mathbb{B}^n)$ –the Borel σ -field of \mathbb{R}^n –, $\sum^n(\psi)$ is p_μ -dense in the set M^i of all Borel measurable functions from \mathbb{R}^n to \mathbb{R} .

In this paper we will prove that for any bounded set $\mathbb{K} \subseteq \mathbb{R}$, if ψ verifies $\psi(\lambda) = 0$ if $\lambda \leq 0$ and $\psi(\lambda) = 1$ if $\lambda \geq 1$ (see fig. 2), these functions, and thus these networks, over \mathbb{K} are lineal combinations of transformations by ψ of propositions of Kleene-Lukasiewicz logic:

$$f(x) = \sum_{j=1}^k \beta_j \psi \left(P_j \left(\alpha_1^j (x_1 + M_1), \dots, \alpha_n^j (x_n + M_n), \alpha_{n+1}^j 1 \right) \right) \quad (II)$$

or equivalently

$$f(x) = \sum_{j=1}^k \beta_j \psi \left(P_j \left(\alpha_1^j(x_i + M), \dots, \alpha_n^j(x_n + M), \alpha_{n+1}^j \right) \right)$$

where

- 1) $\alpha_i^j \in [-1, 1]$ is calculated from w_i^j , and such that $\alpha_i^j(x_i + M_i) \in [0, 1]$,
- 2) P_j is a function from $[0, 1]^n$ to $[0, 1]$ defined as the truth function of propositions in the Kleene-Lukasiewicz propositional logic. That is, they are defined by means of the logic operators:

$$\begin{aligned} a \oplus b &= \min\{1, a + b\} \\ a \otimes b &= \max\{0, a + b - 1\} \\ a^c &= 1 - a. \end{aligned}$$

Fig. 2. Squashing function of the result
In scheme, we can represent the result by:

$$\begin{array}{ccc} \mathbb{K} & \xrightarrow{f} & \mathbb{R} \\ t_j \downarrow & & \uparrow \sum_{j=1}^k \beta_k \psi \circ P_j \circ t_j \\ [0, 1]^n & \xrightarrow{P_j} [0, 1] \xrightarrow{\psi} & [0, 1] \end{array}$$

where $t_j(x_1, \dots, x_n) = \left(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j \right)$.

Important consequences of this result are:

- a) It gives a logic interpretation of neural networks, they are linear combinations of logical propositions (modified by ψ), which are applied to affine transformations of inputs to the unit n-cube $[0, 1]^n$. Moreover, if ψ is interpreted as a linguistic modifier of the truth value [16], then we will have a linguistic interpretation of the network.

- b) It shows that the class $CL^n(\psi)$ of functions defined as linear combinations of logical propositions (modified by ψ), which are applied to affine transformations of inputs to the unit n-cube $[0, 1]^n$, is uniformly dense on compacta in the set C^n , and fixed a probability measure μ on $(\mathbb{R}^n, \mathbb{B}^n)$, $CL^n(\psi)$ is p_μ -dense in the set M^r . Thus, they are also capable of approximating any Borel measurable function and any continuous function from one finite dimensional space to another to any desired degree of accuracy.
- c) The equivalence between neural networks and linear combinations of logical propositions is an open door to study one concept from the view of the other. For example, we can think on using algebraic logic tools for reducing neural networks, or think on learning a logic proposition by means of neural networks in order to reduce the complexity of the logic expression.

The plain of this paper is as follows. In section 2 we present the translation of neural networks into linear combinations of propositions. In Section 3 we will see that any linear combination of propositions is a multilayer feedforward networks. Finally, section 4 contains discussion of our results, directions for further research and some concluding remarks.

2 Neural Networks are linear combinations of propositions

We begin with definitions and notation which enable us to speak precisely about the propositions under consideration.

Definition. The set \mathbb{P}_n of the well formed formulas (in brief w.f.f.) over a set propositional variables $X = \{p_1, \dots, p_n\}$ is defined in Lukasiewicz logic by [1,14]:

1. If $p \in X$, then p is a w.f.f.
2. If P and Q are w.f.f., also $(P \wedge Q)$, $(P \vee Q)$ and $(\neg P)$ are w.f.f.
3. All the w.f.f. are obtained by finite applications of 1 and 2.

Definition. The truth function of a proposition $P \in \mathbb{P}_n$ is the mapping from $[0, 1]^n$ to $[0, 1]$ defined recursively by [1,14]:

1. $p_i(a_1, \dots, a_n) = a_i$.
2. $1(a_1, \dots, a_n) = 1, 0(a_1, \dots, a_n) = 0,$
3. $P \wedge Q(a_1, \dots, a_n) = P(a_1, \dots, a_n) \otimes Q(a_1, \dots, a_n)$
 $P \vee Q(a_1, \dots, a_n) = P(a_1, \dots, a_n) \oplus Q(a_1, \dots, a_n)$
 $\neg P(a_1, \dots, a_n) = P(a_1, \dots, a_n)^c = 1 - P(a_1, \dots, a_n).$

Remark. Note that these functions can be extended to functions from \mathbb{R}^n to \mathbb{R} by the same definition.

Let ϕ be the ramp squashing function defined by $\phi(x) = (1 \wedge x) \vee 0$ (see fig. 3).

Fig. 3. Squashing function ϕ

Lemma 1. For any $x \in \mathbb{R}$ and any $a \in [0, 1]$,

$$\phi(x+a) = (\phi(x) \oplus a) \otimes (\Phi(-x))^c.$$

Proof. If $x+a > 1$, then $x > 1-a$ and $-x < a-1 < 0$. Thus, $\phi(x) \geq \phi(1-a) = 1-a$ and $\phi(-x) = 0$, hence $(\phi(x) \oplus a) \otimes (\Phi(-x))^c = (\phi(x) + a) \vee 1 = \phi(x+a)$. If $x+a < 0$, then $x < -a < 0$ and $-x > a$, thus $\phi(x) = 0$ and $\phi(-x) \geq \phi(a) = a$, hence $(\phi(x) \oplus a) \otimes (\Phi(-x))^c \geq a \otimes a^c = 0 = \phi(x+a)$. Finally, if $0 \leq x+a \leq 1$, then $-a \leq x \leq 1-a$. If $0 \leq x \leq 1-a$, then $\phi(x) = x$ and $\phi(-x) = 0$, hence $(\phi(x) \oplus a) \otimes (\Phi(-x))^c = x+a = \phi(x+a)$. If $-a \leq x < 0$, then $\phi(x) = 0$ and $\phi(-x) = -x$. Thus $(\phi(x) \oplus a) \otimes (\Phi(-x))^c = a \otimes (-x)^c = a+1 - (-x) - 1 \wedge 1 = a+x = \phi(x+a)$. ■

Lemma 2. For any $\alpha_1, \dots, \alpha_n \in [-1, 1]$, there exists a proposition $P = P(\alpha_1, \dots, \alpha_n) \in \mathbb{P}_n$ such that

$$\phi\left(\sum_{i=1}^n \alpha_i a_i\right) = P(|\alpha_1|a_1, \dots, |\alpha_n|a_n) \text{ for any } a_1, \dots, a_n \in [0, 1].$$

Proof. We will prove it by induction over n . If $n = 1$, there are two cases:

- If $\alpha_1 < 0$, then $\phi(\alpha_1 a_1) = 0 = p_1 \wedge \neg p_1 (|\alpha_1|a_1)$.
- If $\alpha_1 \geq 0$, then $\phi(\alpha_1 a_1) = p_1 (|\alpha_1|a_1) = |\alpha_1|a_1$.

Let us suppose that it is verified by a fixed n , and let us prove that it is verified for $\phi\left(\sum_{i=1}^{n+1} \alpha_i a_i\right)$.

- If $\alpha \leq 0 \forall 1 \leq i \leq n+1$, $\sum_{i=1}^{n+1} \alpha_i a_i \leq 0$ and $\phi\left(\sum_{i=1}^{n+1} \alpha_i a_i\right) = 0 = p_1 \wedge \neg p_1(|\alpha_1|a_1, \dots, |\alpha_n|a_n)$.
- If there exists some α_i such that $\alpha_i > 0$, we can suppose without loss of generality that it is $\alpha_{n+1} > 0$, and since $\alpha_{n+1}a_{n+1} \in [0, 1]$, by lemma 1,

$$\begin{aligned} \phi\left(\sum_{i=1}^{n+1} \alpha_i a_i\right) &= \phi\left(\alpha_{n+1}a_{n+1} + \sum_{i=1}^n \alpha_i a_i\right) = \\ &= \left[\phi\left(\sum_{i=1}^n \alpha_i a_i\right) \oplus \alpha_{n+1}a_{n+1}\right] \otimes \left[\phi\left(\sum_{i=1}^n (-\alpha_i) a_i\right)\right]^c. \end{aligned}$$

By induction hypothesis, $\exists P, Q \in \mathbb{P}_n$ such that

$$\phi\left(\sum_{i=1}^n \alpha_i a_i\right) = P(|\alpha_1|a_1, \dots, |\alpha_n|a_n)$$

and

$$\phi\left(\sum_{i=1}^n (-\alpha_i) a_i\right) = Q(|\alpha_1|a_1, \dots, |\alpha_n|a_n),$$

hence taking $R = (P \vee p_{n+1}) \wedge \neg Q$, we have that $R \in \mathbb{P}_{n+1}$ and

$$\phi\left(\sum_{i=1}^{n+1} \alpha_i a_i\right) = R(|\alpha_1|a_1, \dots, |\alpha_n|a_n, |\alpha_{n+1}|a_{n+1}). \quad \blacksquare$$

Lemma 3. Let \mathbb{K} be a bounded subset of \mathbb{R}^n such that for every $i=1\dots n$ it is verified that $x \leq 0$ for every $x \in \mathbb{K}$ or $x_i \geq 0$ for every $x \in \mathbb{K}$, and let $w_1, \dots, w_n \in \mathbb{R}$ be. There exist $M = M(\mathbb{K}) \in \mathbb{N}$, $\alpha_1, \dots, \alpha_n \in [0, 1]$ – $\alpha_i = \alpha_i(w_1, \dots, w_n)$ – and a proposition $P = P(\mathbb{K}, w_1, \dots, w_n) \in \mathbb{P}_n$ such that

$$\phi\left(\sum_{i=1}^n w_i x_i\right) = P(\alpha_1|x_1/M|, \dots, \alpha_n|x_n/M|) \quad \forall x \in \mathbb{K}.$$

Proof. Since that \mathbb{K} is bounded, there exists a $M \in \mathbb{N}$ such that if $x \in \mathbb{K}$, then $|x_i| \leq M$ for all $1 \leq i \leq n$. Let M be fixed. Let $N = 1 + \text{Ent}(\max(|w_1|, \dots, |w_n|))$ where $\text{Ent}(x)$ means the integer part of x . Then

$$\left(\sum_{i=1}^n w_i x_i\right) = M.N \left(\sum_{i=1}^n \delta_i x_i/M\right), \text{ where } \delta_i = w_i/N \in [-1, 1].$$

Thus,

$$\left(\sum_{i=1}^n w_i x_i \right) = \sum_{j=1}^{n.M.N} \delta'_j a_j, \text{ where}$$

for every $i = 1..n$ and for every $j = (i-1).N.M..i.N.M$,

$$\begin{aligned} \delta'_j &= \delta_i \text{ if } x_i \geq 0 \text{ for every } x \in \mathbb{K} \\ &= -\delta_i \text{ if } x < 0 \text{ for every } x \in \mathbb{K} \end{aligned}$$

and

$$\begin{aligned} a_j &= x_i/M \text{ if } x_i \geq 0 \text{ for every } x \in \mathbb{K} \\ &= -x_i/M \text{ if } x_i < 0 \text{ for every } x \in \mathbb{K}. \end{aligned}$$

Since that $\delta'_j \in [-1, 1]$ and $a_j \in [0, 1]$, by lemma 2, there exists a $Q \in \mathbb{P}_{n.N.M}$ such that

$$\phi \left(\sum_{i=1}^n w_i x_i \right) = Q (|\delta_1|a_1, \dots, |\delta_{n.N.M}|a_{n.N.M}), \text{ for every } x \in \mathbb{K},$$

and since there are only n elements $|\delta_j|a_j$, we have that by substitution in Q of every p_j by p_i being i such that $(i-1).N.M < j \leq i.N.M$, we have a proposition $P \in \mathbb{P}_n$ such that

$$\phi \left(\sum_{i=1}^n w_i x_i \right) = P (\alpha_1|x_1/M|, \dots, \alpha_n|x_n/M|) \quad \forall x \in \mathbb{K},$$

where

$$\alpha_i = |\delta'_i| = |w_i/N|. \quad \blacksquare$$

Lemma 4. Let \mathbb{K} be a bounded subset of \mathbb{R}^n and $w_1, \dots, w_n \in \mathbb{R}$. There exist $M \in \mathbb{R}$, $\alpha_1, \dots, \alpha_n, \alpha_{n+1} \in [0, 1]$ - $\alpha_i = \alpha_i(w_1, \dots, w_n, \mathbb{K})$ - and $P \in \mathbb{P}_{n+1}$ such that

$$\phi \left(b + \sum_{i=1}^n w_i x_i \right) = P (\alpha_1(x_1 + M), \dots, \alpha_n(x_n + M), \alpha_{n+1}1) \quad \forall x \in \mathbb{K}.$$

Proof. Since that \mathbb{K} is bounded, $\exists M \in \mathbb{R}$ such that $\forall x \in \mathbb{K}$, $|x_i| \leq M$ for every $i = 1..n$. Thus, $x + M \geq 0$ for every $i = 1..n$. Since

$$b + \sum_{i=1}^n w_i x_i = b + \sum_{i=1}^n w_i (x_i + M) - M \sum_{i=1}^n w_i = b - M \sum_{i=1}^n w_i + \sum_{i=1}^n w_i (x_i + M)$$

By applying lemma 3 to $\left(\sum_{i=1}^{n+1} w_i (x_i + M) \right)$; with $w_{n+1} = b - M \sum_{i=1}^n w_i$, and $x_{n+1} = 1$, over the set $\mathbb{K} + M_x\{1\} = \{(x_1 + M, \dots, x_n + M, 1) / (x_1, \dots, x_n) \in \mathbb{K}\}$, we conclude the proof of the lemma. \blacksquare

In order to simplify the complexity of the affine transformations (M might be a very big number), we will prove the following

Lemma 4 bis. Let \mathbb{K} be a bounded subset of \mathbb{R}^n and $w_1, \dots, w_n \in \mathbb{R}$. There exist $M_1, \dots, M_n \in \mathbb{R}$ - $M_i = M_i(\mathbb{K})$ -, $\alpha_1, \dots, \alpha_n, \alpha_{n+1} \in [-1, 1]$ - $\alpha_i = \alpha_i(w_1, \dots, w_n, \mathbb{K})$ - an a proposition $P \in \mathbb{P}_{n+1}$ such that

$$\phi \left(b + \sum_{i=1}^n w_i x_i \right) = P(\alpha_1(x_1 + M_1), \dots, \alpha_n(x_n + M_n), \alpha_{n+1}1) \quad \forall x \in \mathbb{K}.$$

Proof. Since that \mathbb{K} is bounded, for every $i = 1..n$ there exist N_i^1 and $N_i^2 \in \mathbb{R}$ such that $N_i^1 \leq x_i \leq N_i^2$ for every $x \in \mathbb{K}$. Taking $M' = \min\{N_i^1, N_i^2\}$ and $M_i = M'_i$ if $N_i^1 \leq N_i^2$ or $M_i = -M'_i$ if $N_i^1 > N_i^2$, we have that the set

$$\mathbb{K} + (M_1, \dots, M_n) x\{1\} = \{(x_1 + M_1, \dots, x_n + M_n, 1) / (x_1, \dots, x_n) \in \mathbb{K}\}$$

verifies the conditions of lemma 3 and by applying this lemma to

$$\left(\sum_{i=1}^n w_i (x_i + M_i) \right); \text{ with } w_{n+1} = b - \sum_{i=1}^n M_i w_i, x_{n+1} = 1 \text{ and } M_{n+1} = 0,$$

we conclude the proof. ■

As a immediate consequence we have the following results:

Theorem 1. Let $w_i^j \in \mathbb{R}$, $\beta_j, b_j \in \mathbb{R}$, $i = 1..n$, $j = 1..k$, and let f be the function defined by

$$f(x) = \sum_{j=1}^k \beta_j \phi \left(b_j + \sum_{i=1}^n \omega_i^j x_i \right).$$

For every bounded subset \mathbb{K} of \mathbb{R}^n , there exist k propositions $P_1, \dots, P_k \in \mathbb{P}_{n+1}$, $(n+1) \cdot k$ real numbers $\alpha_i^j \in [-1, 1]$ ($i = 1..n+1, j = 1..k$) and n real numbers M_1, \dots, M_n , such that

$$f(x) = \sum_{j=1}^k \beta_j P_j \left(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j 1 \right), \quad \forall x \in \mathbb{K}. \quad (III)$$

We can express in scheme this result by:

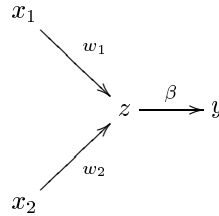
$$\begin{array}{ccc} \mathbb{K} & \xrightarrow{f} & \mathbb{R} \\ t_j \downarrow & & \uparrow \sum_{j=1}^k \beta_j P_j \\ [0, 1]^{n+1} & \xrightarrow{P_j} & [0, 1] \end{array}$$

where $t_j(x_1, \dots, x_n) = (\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j)$.

Theorem 1 bis. (Neural Networks Version). Let RN be a single k -nodes hidden layer feedforward networks with squashing function ϕ at the hidden layer and no squashing at the output layer. For every bounded subset \mathbb{K} of \mathbb{R}^n , there exist k propositions $P_1, \dots, P_k \in \mathbb{P}_{n+1}$, $(n+1) \cdot k$ real numbers $\alpha_i^j \in [-1, 1]$ ($i = 1..n+1$, $j = 1..k$) and n real numbers M_1, \dots, M_n , such that the function f calculated by the net over \mathbb{K} is

$$f(x) = \sum_{j=1}^k \beta_j P_j(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j) \quad (IV)$$

Examples: 1) If the net has the form



- If $w_1 x_1 \in [0, 1]$ and $w_2 x_2 \in [0, 1]$ for each x_1, x_2 in the domain, then the output will be

$$\begin{aligned} \beta\phi(w_1 x_1 + w_2 x_2) &= \beta[(\phi(w_1 x_1) \oplus w_2 x_2) \otimes \phi(-w_1 x_1)^c] = \\ &= \beta[(\phi(w_1 x_1) \oplus w_2 x_2) \otimes 0^c] = \\ &= \beta[(\phi(w_1 x_1) \oplus w_2 x_2) \otimes 1] = \beta[(\phi(w_1 x_1) \oplus w_2 x_2)] = \\ &= \beta[w_1 x_1 \oplus w_2 x_2] \end{aligned}$$

hence the equivalent linear combination is $\beta(p_1 \vee p_2)(w_1 x_1, w_2 x_2)$.

- If $w_1 x_1 \in [0, n_1]$ and $w_2 x_2 \in [0, n_2]$ in the domain, then the output will be

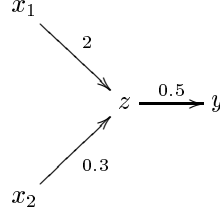
$$\begin{aligned} \beta\phi(w_1 x_1 + w_2 x_2) &= \beta[\phi(w_1/n_1 x_1 + w_2/n_2 x_2 + w_1/n_1 + w_2/n_2)] = \\ &= \beta[w_1/n_1 x_1 \oplus w_2/n_2 x_2 \oplus w_1/n_1 \oplus w_2/n_2], \end{aligned}$$

hence, the equivalent linear combination is

$$\beta(n_1 * p_1 \vee n_2 * p_2)(w_1/n_1 x_1, w_2/n_2 x_2),$$

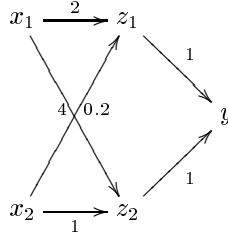
where $n * p$ represents the proposition $p \vee \dots \vee p$.

For example the net



where $x_1 \in [0, 1]$ and $x_2 \in [1, 3]$

calculate the function $0.5 (x_1 \oplus x_1 \oplus 0.3x_2) = 0.5[(p_1 \vee p_1 \vee p_2) (x_1, 0.3x_2)]$; and the net



where $x_1 \in [0, 4]$ and $x_2 \in [0, 1]$

calculate the function

$$(x_2 \oplus 4) \oplus x_2 \oplus (1/4x_1) \oplus 8) \oplus (1/4x_1) - (x_2 \oplus 0.2x_1) = \\ (8 * p_1 \vee 4p_2) (1/4x_1, x_2) - (p_1 \vee p_2) (0.2x_1, x_2) .$$

We can unify the affine transformation by

$$[(40 * p_1 \vee 4 * p_2) - (4 * p_1 \vee p_2)] (1/20x_1, x_2) .$$

As is well-known, networks with this squashing function do not learning by backpropagation. Thus, it is necessary to extend the result to networks with others squashing function.

Lemma 5. Let ψ be a squashing function verifying $\psi(x) = 0$ if $x \leq 0$ and $\psi(x) = 1$ if $x \geq 1$. Then, $\psi(x) = \psi \circ \phi(x)$ for each $x \in \mathbb{R}$.

Proof. If $x \leq 0$, then $\psi(x) = 0 = \psi(0) = \psi \circ \phi(x)$. If $x \geq 1$, then $\psi(x) = 1 = \psi(1) = \psi \circ \phi(x)$. If $0 \leq x \leq 1$, then $\phi(x) = x$ and $\psi(x) = \psi \circ \phi(x)$. ■

Useful examples of this kind of squashing functions are the threshold function, $\psi(x) = 1_{\{x \geq 0\}}$ (where $1_{\{\cdot\}}$ denotes the indicator function), and the cosine squasher of Gallant and White over $[0, \pi/2]$:

$$\psi(x) = 1_{\{x \geq \pi/2\}} + (1 + \cos[x + 3\pi/2]) (1/2) 1_{\{\pi/2 > x \geq 0\}}.$$

Theorem 2. Let ψ be a squashing function verifying $\psi(x) = 0$ if $x \leq 0$ and $\psi(x) = 1$ if $x \geq 1$. Let $w_i^j \in \mathbb{R}$ and $\beta_j, b_j \in \mathbb{R}, i = 1..n, j = 1..k$. Let f the function

$$f(x) = \sum_{j=1}^k \beta_j \psi \left(b_j + \sum_{i=1}^n \omega_i^j x_i \right). \quad (I)$$

For every bounded subset \mathbb{K} of \mathbb{R}^n , there exist k propositions $P_1, \dots, P_k \in \mathbb{P}_{n+1}$, $(n+1) \cdot k$ real numbers $\alpha_i^j \in [-1, 1]$ ($i = 1..n+1, j = 1..k$) and n real numbers M_1, \dots, M_n , such that $f(x) = \sum_{j=1}^k \beta_j \psi \left(P_j \left(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j 1 \right) \right)$, $\forall x \in \mathbb{K}$.

Theorem 2 bis. (Neural Networks Version).

Let ψ be a squashing function verifying $\psi(x) = 0$ if $x \leq 0$ and $\psi(x) = 1$ if $x \geq 1$. Let RN be a single k -nodes hidden layer feedforward networks with squashing function ψ at the hidden layer and no squashing at the output layer. For every bounded subset \mathbb{K} of \mathbb{R}^n , there exist k propositions $P_1, \dots, P_k \in \mathbb{P}_{n+1}$, $(n+1) \cdot k$ real numbers $\alpha_i^j \in [-1, 1]$ ($i = 1..n+1, j = 1..k$) and n real numbers M_1, \dots, M_n , such that the function f calculated by la red over \mathbb{K} is

$$f(x) = \sum_{j=1}^k \beta_j \psi \left(P_j \left(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j 1 \right) \right) \quad (II)$$

$$\begin{array}{ccc} \mathbb{K} & \xrightarrow{f} & \mathbb{R} \\ t_j \downarrow & & \uparrow \sum_{j=1}^k \beta_k \psi \circ P_j \circ t_j \\ [0, 1]^n & \xrightarrow{P_j} [0, 1] \xrightarrow{\psi} & [0, 1] \end{array}$$

A leading case occurs when ψ is a linguistic modifier of the truth value [Zadeh], for example $\psi(x) = \mu_{\text{very}}(x) = x^2$ for every $x \in [0, 1]$. Then $\psi \circ P$ is the ‘‘fuzzy’’ proposition ‘‘very P ’’, and thus, f is a linear combination of fuzzy propositions.

Let us observe that if we use a different squashing function ψ_j in each node of the hidden layer, then the theorem holds with

$$f(x) = \sum_{j=1}^k \beta_j \psi_j \left(P_j \left(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j 1 \right) \right) \quad (II).$$

3 Linear combinations of Lukasiewicz propositions are neural networks

From theorem 2 it is followed:

Corollary 1. Let ψ be a squashing function verifying $\psi(x) = 0$ if $x \leq 0$ and $\psi(x) = 1$ if $x \geq 1$. Let us denote $CL^n(\psi)$ the class of function defined by (II), where each P_j is seen as a function from \mathbb{R}^{n+1} to \mathbb{R} , that is, linear combinations of Lukasiewicz propositions applied to affine transformation of \mathbb{R}^n . $CL^n(\psi)$ is uniformly dense on compacta in C^n , and give a probability measure p_μ over \mathbb{B}^n , $CL^n(\psi)$ is p_μ -dense on M^n . We will denote $CL^n(\phi) = CL^n$.

As an immediate consequence of theorem 1, “any continuous function can be approximate over each compact by functions expressed by means of scalar product, $+$, \oplus , \otimes and c^n ”.

Definition. Given a subset \mathbb{K} of \mathbb{R}^n , we will say that a function $f \in CL^n(\psi)$ defined by II is logically defined over \mathbb{K} if for each $x \in \mathbb{K}$, $\alpha_i^j(x_i + M_i) \in [0, 1]$ for every $i = 1..n$, $j = 1..k$. We will denote $CL^n(\psi, \mathbb{K})$ the class of the functions in $CL^n(\psi)$ logically defined over \mathbb{K} .

Let us observe that $CL^n(\psi)$ is a real vectorial space, and $CL^n(\psi, \mathbb{K})$ is either the empty set if \mathbb{K} is unbounded or a vectorial subspace of $CL^n(\psi)$ if \mathbb{K} is bounded.

Lemma 6. Let \mathbb{K} be a bounded subset of \mathbb{R}^n and let $\alpha_1, \dots, \alpha_n \in [-1, 1]$, $M_1, \dots, M_n \in \mathbb{R}$ such that $\alpha_i(x_i + M_i) \in [0, 1]$ for every $x \in \mathbb{K}$. Let $P \in \mathbb{P}_n$. There exists a neural network RN_P with squashing function ϕ in all its nodes such that the function calculated by RN_P is:

$$f(x) = P(\alpha_1(x_1 + M_1), \dots, \alpha_n(x_n + M_n)) \text{ for every } x \in \mathbb{K}.$$

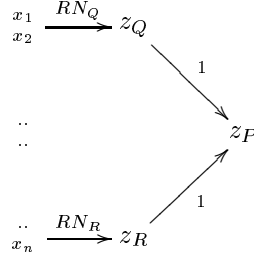
Proof. We will prove it by induction over the recursive construction of P . If $P = p$ is a simple proposition, the net

$$x_i \xrightarrow{\alpha_i} z_P$$

calculate $z_P = \phi(\alpha_i(x_i + M_i)) = \alpha_i(x_i + M_i) = \phi(P(\alpha_1(x_1 + M_1), \dots, \alpha_n(x_n + M_n)))$.

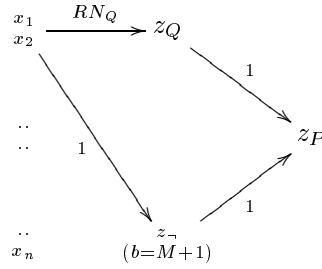
If P is a complex proposition, then there exist three cases:

- a) $P = Q \vee R$. By hypothesis of induction Q and R will be calculated by two neural networks RN_Q and RN_R , then the net



calculate $z_P = \phi(z_Q + z_R) = \min(z_Q + z_R, 1) = z_Q \oplus z_R = P(\alpha_1(x_1 + M_1), \dots, \alpha_n(x_n + M_n))$.

- b) $P = \neg Q$. By hypothesis of induction Q will be calculated by a net RN_Q , and the net



where M is an integer such that if $x \in \mathbb{K}$, then $|x_1| \leq M$, calculates $z_P = 1 - z_Q = P(\alpha_1(x_1 + M_1), \dots, \alpha_n(x_n + M_n))$, since $-M \leq x_1$ and as $1 \leq x_1 + M + 1, z_{\neg} = \phi(x_1 + M + 1) = 1$.

- c) $P = Q \wedge R$. Then $P = \neg(\neg Q \vee \neg R)$ and by applying a) and b) we obtain the net RN_P from RN_Q and RN_R . ■

Lemma 7. Let \mathbb{K} be a bounded subset of \mathbb{R}^n and let $\alpha_1, \dots, \alpha_n, \alpha_{n+1} \in [-1, 1], M_1, \dots, M_n \in \mathbb{R}$ such that $\alpha_i(x_i + M_i) \in [0, 1]$ for every $x \in \mathbb{K}$. Let $P \in \mathbb{P}_{n+1}$. There exists a neural network RN_P with squashing function ϕ in all its nodes such that the function calculated by RN_P is:

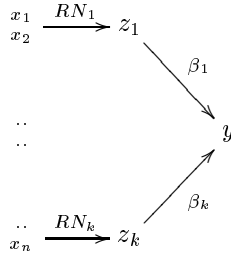
$$f(x) = P(\alpha_1 x_1 + M_1, \dots, \alpha_n(x_n + M_n), \alpha_{n+1}) \text{ for each } x \in \mathbb{K}.$$

Proof. It is enough to apply lemma 6 to $\mathbb{K}x\{1\}$. ■

Theorem 3. Let $f \in CL^n$, i.e., $f(x) = \sum_{j=1}^k \beta_j P_j(\alpha_1^j(x_1 + M_1), \dots, \alpha_n^j(x_n + M_n), \alpha_{n+1}^j 1)$.

For every bounded subset of \mathbb{R}^n where f is logically defined there exist a feedforward network RN with squashing function ϕ at the hidden layers and no squashing at the output layer such that over \mathbb{K} RN calculates exactly $f(x)$.

Proof. By lemma 7 for every P_j there exist a net RN_j with squashing function ϕ in all nodes calculating $P_j(\alpha_1^j x_1 + M_1, \dots, \alpha_n^j x_n + M_n, \alpha_{n+1}^j 1)$. Then, the net



calculates f . ■

4 Discussion and Concluding Remarks

The results of section 2 establish that standard multilayer feedforward networks can be logically interpreted, in a very specific and satisfying sense. It is important to observe that the construction of the linear combination of propositions from a neural network is algorithmic. These results would provide a fundamental basis for rigorously establishing the reasoning used by multilayer feedforward networks in learning. We can associate a succession of linear combinations of propositions to the different configurations of a net during learning, and thus we can hope a “logic connection” among the linear combination of propositions. If it were possible, then we could talk about a “logic learning” in neural networks!

In section 3 we prove that linear combination of propositions can be seen as neural networks. Now, if we have a logic learning in $CL^n(\psi)$ of a function, we can associate a network learning which could help the classical network logic learning. Conversely, we can use neural networks in order to learn logical connections from examples [3].

Other important related area for further research is the construction of neural networks with a priori knowledge. We must translate the knowledge in propositions and then to add this propositions as a subnet before starting the learning. In this paper only Lukasiewicz logic is considered. But many others logics are used in Approximate Reasoning [2] for representing knowledge. An interesting and open question is: exist there a similar result for these logics?.

The results given here are clearly only one step in a rigorous general investigation of the intrinsic logic of multilayer feedforward networks. We can say that the equivalence between neural networks and linear combinations of logical propositions is an open door to study one concept from the view of the other.

References

- [1] Ackermann, R. (1967) *An Introduction to Many-valued Logics*. Roulledge & Kegan Paul Ltd. Londres.
- [2] Castro, J.L., Trillas, E. and Cubillo, S. (1994). On Consequence in Approximate Reasoning. *Journal of Applied Non-Classical Logics*, 4, pp. 91-103.
- [3] Castro, J.L., Delgado, M. and Trillas, E. (1994). Inducing Implication Relations, *International Journal of Approximate Reasoning* , 10, pp. 235-250.
- [4] Hecht-Nielsen, R. (1987). Theory of the back propagation neural Networks. In *Proceedings of IEEE First International Conference on Neural Networks*, pp. 11-14. San Diego, SOS Printing.
- [5] Hornik, K., Stinchcombe, M. White, H. (1989). Multilayer Feedforward Networks are Universal Approximators. *Neural Networks* 2, pp. 359-366.
- [6] *Proceedings of IEEE First International Conference on Neural Networks* (1987) San Diego, SOS Printing.
- [7] *Proceedings of IEEE Second International Conference on Neural Networks* (1988) San Diego, SOS Printing.
- [8] *Proceedings of IEEE third International Conference on Neural Networks* (1989) San Diego, SOS Printing.
- [9] *Proceedings of IEEE 4th International Conference on Neural Networks* (1990). San Diego, SOS Printing.
- [10] *Proceedings of IEEE 5th International Conference on Neural Networks* (1991). San Diego, SOS Printing.
- [11] *Proceedings of IEEE 6th International Conference on Neural Networks* (1992). San Diego, SOS Printing.
- [12] *Proceedings of IEEE 7th International Conference on Neural Networks* (1994). San Diego, SOS Printing.
- [13] *Proceedings of IEEE 8th International Conference on Neural Networks* (1994). San Diego, SOS Printing.
- [14] Rescher, N. (1969). *Many-valued Logic*. McGraw-Hill.
- [15] Williams, R.J. (1986). The logic of activation functions. In D. E. Rumelhart & J.L. McClelland (Eds.), *Parallel distributed processing: Explorations in the microstructures of cognition* (Vol 1, pp. 423-443). Cambridge: MIT Press.
- [16] Zadeh, L.A. (1987). The concept of a Linguistic Variable and its Application to Approximate Reasoning, in R. Yager, S. Ovchinnikov, M.R. Tom and T.H. Nguyen (eds.), *Fuzzy Sets and Applications: Selected Papers by L.A. Zadeh*, pp. 219-329.