Definition of Neural Networks

Samuel Lippl

April 18, 2018

Contents

Neural networks are very general notions that encompass many different varieties. While there have been attempts to restrict make restrictions that are derived from their purpose (see, for instance, [2]), I argue that these restrictions make the definition more complicated to prove but not more powerful. Therefore, I provide a definition in the sense of [7]. However, I have attempted to make the notation more compact for my purposes and I have worked out the different aspects of neural networks more clearly. Finally, my definition is broader than Rojas' definition in [7].

1 Structure of Neural Networks

1.1 General Structure

1.1 Definition (Elementary Structure of Neural Networks). A *neural network* is defined by its *elementary structure* which is a directed graph $(\mathcal{V}, \mathcal{C}), \mathcal{C} \subseteq \mathcal{V} \times \mathcal{V}$, that is *connected*.
We call the nodes in $\mathcal{V}$ *Processing Units* (PU) und the edges in $\mathcal{C}$ *Unit Connections* (UC). We will leave open the possibility to associate a class attribute with PUs or UCs that may restrict different network aspects (see, for instance, example 1.16).

1.2 Notation. We associate $\mathcal{C}$ with the relations $\rightarrow_{\mathcal{C}}$ and $--_{\mathcal{C}}$ on $\mathcal{V}$, i. e.

$$\begin{aligned} v \rightarrow_{\mathcal{C}} w &\equiv (v, w) \in \mathcal{C} \equiv v \leftarrow_{\mathcal{C}} w \\ v --_{\mathcal{C}} w &\equiv v \rightarrow_{\mathcal{C}} w \vee w \rightarrow_{\mathcal{C}} v \end{aligned} \tag{1}$$

Where the context is clear, we write $\to$ and $--$ instead of $\to_{\mathcal{C}}$ and $--_{\mathcal{C}}$.

We now complete Definition 1.1:

1.3 Definition (Complete Connectedness). We call a graph completely connected if and only if

$$\underset{v,w\in\mathcal{V}}{\forall}\underset{k\in\mathbb{N}^+}{\exists}\underset{u_1,\ldots,u_k\in\mathcal{V}}{\exists} u_1 = v \wedge \underset{i\in\{1,\ldots,k-1\}}{\forall} u_i --u_{i+1} \wedge u_k = w \tag{2}$$

1.4 The restriction that a neural network must be completely connected is sensible. It prevents networks from incorporating PUs or sets of PUs that are independent from the rest of the network. A single isolated PU may simply be ignored and a set of isolated PUs may be considered as a separate network[1].
(Where does this restriction come to use?)

1.5 Notation (Preceding and succeeding units). If $u \in \mathcal{V}$ is a PUs we define the set of *preceding* PUs

$$\mathcal{C}_u := \{v \in \mathcal{V} | v \to u\} \tag{3}$$

and the set of *succeeding* PUs

$$\mathcal{C}^u := \{w \in \mathcal{V} | u \to w\} \tag{4}$$

1.6 Definition (Acyclic Networks). The network $(\mathcal{V}, \mathcal{C})$ is called *feed-forward* if and only if it is an acyclic graph.

1.7 Definition. $v \in \mathcal{V}$ is called a *highest-level* PU (resp. *lowest-level* PU) if and only if $\mathcal{C}_v = \emptyset$ (resp. $\mathcal{C}^v = \emptyset$).

1.8 Proposition. A finite feed-forward network $\mathcal{N}$ can be divided into L layers, $L \in \mathbb{N}$ such that

$$\mathcal{V} = \mathcal{V}^{(0)} \dot\cup \cdots \cup \mathcal{V}^{\dot(L)} \tag{5}$$

and

$$v \to w \Rightarrow v \in \mathcal{V}^{(l_1)}, w \in \mathcal{V}^{(l_2)}, l_1 < l_2 \tag{6}$$

The minimal value of L is called the *depth* of $\mathcal{N}$ and can be found by the following algorithm: 1) Take the highest-level PUs of $\mathcal{C}$ and define them as $\mathcal{V}^{(0)}$. 2) Take the highest-level PUs of $\mathcal{C}$ without $\mathcal{V}^{(0)}$ and define them as $\mathcal{V}^{(1)}$. 3) Repeat this process until a partition of $\mathcal{V}$ has been found. The final layer number is the depth of $\mathcal{N}$.

Proof. Proof by induction on $|\mathcal{V}|$ over the proposition and the predicate concerning the depth L of $\mathcal{V}$

$$P(\mathcal{V}, L) \equiv \underset{v_1,\ldots,v_L}{\exists}\underset{i=1,\ldots,L-1}{\forall} v_i \to v_{i+1} \tag{7}$$

Note that $P(\mathcal{V}, L)$ proves that L is the minimal number of layers as any number lower than L could obviously not satisfy condition (6).
Suppose $|\mathcal{V}| = 1$. As the network is feed-forward, $\mathcal{C} = \emptyset$ and (6) is automatically true. $\mathcal{V} = \mathcal{V}^{(0)}$ produces the layered structure of $\mathcal{N}$.

[1] Of course, a single PU could also be considered as a neural network.

Suppose $|\mathcal{V}| = n$ and the proposition is proven for all $|\mathcal{V}| < n$. As the network is feed-forward, there is at least one lowest-level PU of $\mathcal{C}$. Define the set of lowest-level PUs as $\mathcal{V}^{(\max)}$. Then, the proposition and the predicate are proven for $\mathcal{V}' := \mathcal{V} \setminus \mathcal{V}^{(\max)}$. If the depth of $\mathcal{V}'$ is L', the depth of $\mathcal{V}$ is $L := L' + 1$. The last step of the algorithm now corresponds to setting $\mathcal{V}^{(L)} := \mathcal{V}^{(\max)}$.
Obviously, (5) holds true. We now prove (6): if $l_2 < L$, this is true by induction hypothesis. If $l_2 = L$, $l_1 < L = l_2$ because v is not a lowest-level PU.
Finally, $P(\mathcal{V}, L)$ holds: by $P(\mathcal{V}', L')$ we can find $v_1, \dots, v_{L-1}$ such that (7) is satisfied. We know that there must be some $v_L \in \mathcal{V}$ such that $v_{L-1} \to v_L$ because v_{L-1} is not a lowest-level PU and therefore $P(\mathcal{V}, L)$. □

1.9 Many neural networks are structured into L layers of PUs where $v_1^{(l)}, \dots, v_{n_i}^{(l)} \in \mathcal{V}$ and $\mathcal{C} = \left\{(v_i^{(l)}, v_j^{(l+1)}) | l = 0, \dots, L-1, i = 1, \dots, n_l, j = 1, \dots, n_{l+1}\right\}$ where input is provided to layer 0 and the network produces some kind of output at layer L. It may helpful to use such a structure to visualize certain notions. However, as far as I see it, a general feed-forward architecture provides all the benefits of a layered structure which is supported by propositions 1.8 and **??**.

1.10 More generally, we may analyze unit connections (v, w) with respect to some layered structure (5). If $v \in \mathcal{V}^{(l_1)}, w \in \mathcal{V}^{(l_2)}$, we call (v, w) *feed-forward* if $l_1 < l_2$, *feedback* if $l_1 > l_2$ and *lateral* if $l_1 = l_2$.

1.11 The elementary structure of a neural network only specifies how the PUs are connected but not how they use these connections. With respect to this task, we have to differentiate between the *network assumptions* that cannot be change during its applications and its *parameters* that can be changed during learning (see chapter **??**). While specific values of the latter should not be considered as part of the network, the values they may take are as relevant as the way they parametrize the network.

1.12 Definition (Functional Structure). We define the *parameter space* Θ and the *integrative functions* $\mathcal{G} = (g_u)_{u \in \mathcal{V}}$ where

$$g_u : \mathbb{R}^{\mathcal{C}_u} \times \Theta \to \mathbb{R} \qquad (x, \theta) \mapsto g_u(x|\theta) \tag{8}$$

We call $(\mathcal{G}, \Theta)$ the *functional structure* of the neural network and $(\mathcal{V}, \mathcal{C}, \mathcal{G}, \Theta)$ its *structure*.

1.13 I will note that we have not associated these parameters with a specific PU or UC. The main reason for this is the possibility of shared parameters, see definition **??**. I will also note that this chapter is actually not concerned with how the network generates its values which will be of concern in chapter 2. The reader may however have inferred from (8) that I expect the PUs to take real values. This is only because there has been no reason to generalize this definition. The relevant concepts can easily be generalized to more general spaces, however.

1.2 Additive neural networks

1.14 Definition (Additive neural network). An *additive neural network* is defined by certain restrictions on the functional structure. The parameter space is now $\Theta \subseteq \mathbb{R}^{\mathcal{C}}$ and we define a set of *unit functions* $\mathcal{F} = (f_u)_{u \in \mathcal{V}}$ where

$$f_u : \mathbb{R} \to \mathbb{R} \tag{9}$$

Then, the integrative functions are defined by

$$g_u(x|\theta) := \sum_{v \in \mathcal{C}_u} \theta_{(v,u)} f_v(x_v) \tag{10}$$

By defining

$$\theta_u^v \equiv \theta_{(v,u)} \qquad \theta^v \equiv (\theta_u^v)_{u \in \mathcal{C}^v} \qquad \theta_u \equiv (\theta_u^v)_{v \in \mathcal{C}_u} \tag{11}$$

$$f_V : \mathbb{R}^V \to \mathbb{R}^V \quad f_V(x) := (f_v(x_v))_{v \in V} \quad V \subseteq \mathcal{V} \tag{12}$$

and using the ordinary definition of matrix multiplication we can notate

$$g_u(x|\theta) = \theta_u^T f_{\mathcal{C}_u}(x) \tag{13}$$

where θ^T is the transpose of θ.

1.15 In many neural networks all unit functions are provided by a unique function f. Most unit functions map their input to $(-1, 1)$ or $(0, 1)$. Popular choices are the sigmoid function $f(x) = \frac{1}{1+e^{-cx}}$ or the tangens hyperbolicus $f(x) = \tanh(x)$. If we define $f = \mathrm{id}$, the resulting network provides a linear model.
Finally, many neural networks work in layers: they are structured into L layers of PUs where $v_1^{(l)}, \ldots, v_{n_i}^{(l)} \in \mathcal{V}$ and $\mathcal{C} = \left\{(v_i^{(l)}, v_j^{(l+1)}) | l = 0, \ldots, L-1, i = 1, \ldots, n_l, j = 1, \ldots, n_{l+1}\right\}$ where input is provided to layer 0 and the network produces some kind of output at layer L. It may helpful to use such a structure to visualize certain notions. However, as far as I see it, a general feed-forward architecture provides all the benefits of a layered structure which may be justified by Proposition **??**.

1.16 Example (Inhibitory Connections). In the brain, we may distinguish between inhibitory and excitatory synaptic connections. In case of the latter, a firing presynaptic neuron increases postsynaptic firing, in case of the former, presynaptic firing decreases or inhibits postsynaptic firing. We may model this constellation by distinguishing between two classes $\mathcal{C}_i$ and $\mathcal{C}_e$ of connections and defining $\Theta \subseteq \mathbb{R}^{\mathcal{C}}$ such that $\theta_c \leq 0$ if $c \in \mathcal{C}_i$ and $\theta_c \geq 0$ if $c \in \mathcal{C}_e$.

1.3 Predictive Coding

1.17 (Predictive Coding I: The Generative Model) Predictive Coding serves as a model of how our brain processes sensations into perceptions and is based on [4]. It poses that our brain processes data by inferring hidden or perceptible *causes* which are, in this case, "simply the states of processes generating sensory data" [1, p. 819]. Higher-level causes produce lower-level causes which, in turn produce sensory data. To quote Friston [1]

> It is not easy to ascribe meaning to these states without appealing to the way that we categorize things, either perceptually or conceptually. Causes may be categorical in nature, such as the identity of a face or the semantic category to which an object belongs. Others may be parametric, such as the position of an object.[1, p. 819]

We can therefore define a *generative model* that describes how causes relate to sensations by a feed-forward neural network $\mathcal{P}_{\text{gen}} := (\mathcal{V}_P, \mathcal{C}_{\text{gen}})$ where we may represent the process by the deterministic model

$$v = g_v(\mathcal{C}_v; \Theta) \tag{14}$$

As we have no means of representing causes perfectly, we may want to model the outcomes stochastically. We would therefore posit a generative distribution

$$p_v(v|\mathcal{C}_v;\Theta) \tag{15}$$

for every $v \in \mathcal{V}$. For now, we will also assume that v is independent from $\mathcal{V} \setminus \mathcal{C}_v$.

1 Open Problem (Dependent consequences). How do we generalize predictive coding to dependent variables? Friston [1] proposes a framework for decorrelating normal variables that is relatively straight-forward. I will need to extend that to general hierarchies (which should be easy using proposition 1.8). How may we approach general dependence structures?

It is an open question to me, whether this is a concern of structure or a concern of states (see chapter 2).

2 Open Problem (Stochastic Networks). Is randomness a structure or a state function? My current suggestion would be the former.

Either way, the lowest-level PUs of $\mathcal{C}_{\text{gen}}$ correspond to the sensory input. While this may be a more accurate description of reality, our brain has no means of directly perceiving the hidden causes. We refer to this goal as *inference* and it will be the subject of 1.18 and **??**.

1.18 (Predictive Coding II: Inference 1) I will conclude this section with some structural remarks on inference that may elucidate some previous connective notions. If we suppose a layered structure as in 1.17, then the generative connections $\mathcal{C}_{\text{gen}}$ are obviously feed-forward. If we want to infer the hidden causes from the lower-level PUs starting with the sensory input, we would therefore have to define a feedback structure $\mathcal{C}_{\text{inf}}$. For reasons of simplicity, we will consider a slightly changed structure in **??**.

1.4 Parallelity

1.19 One of the discerning characteristics of neural networks in comparison to other mechanisms of inference is their parallelity as we will discover in the next two paragraphs.

1.20 (Transformations) A huge amount of parallelity allows us to transform the input space. This is what happens in the case of predictive coding. We read out the pixel values of the visual input and infer abstract concepts like spatial orientation or even higher concepts like faces. This is only possible because of the massive amount of parallel processing in applied examples like artificial visual recognition systems (see [7, pp. 70-73]). This is also an important advantage for biological networks where the input to our retina is transformed to different characteristics like orientation or relative brightness (see [5, pp. 257-276]). Indeed, predictive coding has been demonstrated to imitate some of the properties of the visual cortex [6]. On the other hand, there are certain advantages to parallelity with respect to biological networks that do not directly extend to artificial networks which I will discuss in the next section (see **??**).

1.21 For now the last paragraph may provide some proof that parallelity is indeed a useful property. On the other hand, a network that does not use parallel processing does not need to be formulated in

the same framework. These considerations have led Guresen and Kayakutlu [2] to claim parallelity as a necessary feature of neural networks. In their proposed definition, they require that "at least two of the multiple [Processing Units are] connected in parallel" [2, p. 428] because, they agure, "structures [...] with one or more [Processing Units] connected serially cannot be referred as [a neural network] because it will lose the power of parallel computing and starts to act more like existing computers than a brain" [2, p. 428].
In principle, I agree with their point. I would argue, however, that it is not helpful to include such a criterion in the definition. After all, if there is one parallel pathway in a massive serial network, this would correspond to a neural network according to their notion even though the difference to a uniquely serial network is insubstantial.
The definition therefore seems to raise an issue that cannot be resolved, analogous to the definition of a "heap of sand", another instance of the Sorites paradox (see [3]): without doubt, one grain of sand is necessary to have a heap but it is not sufficient and we can probably not find a hard definition of such a vague statement. We would be better advised to simply define the underlying property a "heap" refers to, i. e. the *number* of grains of sand that allows us to use more rigorous notions.
Comparably, parallel processing such that it is advantageous cannot be covered by such a broad definition – not only does parallelity exist on a spectrum, it also strongly depends on the context. I would therefore suggest that an attempt to find a way to talk about parallelity may be more fruitful than Guresen and Kayakutlu's hard borders. I have not found the definition of a comparable *degree of parallelity* and Guresen and Kayakutlu themselves also do not elaborate on what they mean by parallel connections.

3 Open Problem (Degree of parallelity). How may we define the *degree of parallelity* of a neural network?

2 States of Neural Networks

2.1 General States

2.1 Definition (States). We call $s \in \mathbb{R}^{\mathcal{V}}$ a state of the network where we denote the state of the PU v by $s(v)$ and the state of vector of PUs $\overline{v} = \begin{pmatrix} v_1 & \cdots & v_n \end{pmatrix}$ by $s(\overline{v}) = \begin{pmatrix} s(v_1) & \cdots & s(v_n) \end{pmatrix}$. s is *balanced* if and only if

$$\underset{u \in \mathcal{V}}{\forall} s(u) = g_u \left(s \left(\mathcal{C}_u \right) | \theta \right) \tag{16}$$

2.2 Definition (State Function). We call $T : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{V}}$ a *state function* where $\mathcal{I} \subseteq \mathcal{V}$ is the *input*. A state function T_{step} is called a *state step* if and only if $\mathcal{I} = \mathcal{V}$ and

$$\underset{u \in \mathcal{V}}{\forall} T(s)(u) = g_u \left(s(\mathcal{C}_u) | \theta \right) \tag{17}$$

T is called *balanced* if and only if every $T(s)$ is balanced.
T is called *invariant* in $\mathcal{J} \subseteq \mathcal{I}$ if and only if

$$\underset{j \in \mathcal{J}}{\forall} T(s)(j) = s(j) \tag{18}$$

2.3 Proposition. The balanced states are the fixed points of a state step of the network.

Proof. The fixed point equation for the state step corresponds exactly to (16). □

2.4 The value attribution of neural networks is often defined together with their structure or even implicitly by their structure. As we will discover below, this is not a problem for feed-forward networks, however, recurrent networks lead to significant difficulties, both in terms of clear and concise notation. Distinguishing between state steps and balanced functions allows to describe, in separate terms, how the neural network updates its values (which should correspond directly to its structure in terms of (17)) and what the result of these operations is.

2.5 Proposition. A finite feed-forward network has a unique balanced state function

$$T : \mathbb{R}^{\mathcal{V}^{(0)}} \to \mathbb{R}^{\mathcal{V}} \tag{19}$$

that is invariant in $\mathcal{I}$. It is defined recursively by

$$\begin{aligned} u \in \mathcal{V}^{(0)} &\Rightarrow T(s)(u) = s(u) \\ u \notin \mathcal{V}^{(0)} &\Rightarrow T(s)(u) = g_u\left(T(s)(\mathcal{C}_u)|\theta\right) \end{aligned} \tag{20}$$

Proof. The state function is well-defined because the network structure is acyclic. (20) is equivalent to invariance and balancedness and therefore the reason for both existence and uniqueness. □

2.6 (Free Energy in invariant functions) Many neural networks can be considered as structures that try to make sense of data. In this framework, every PU corresponds to some hidden or empirical property of the world and the network tries to infer the remaining states from a given input of states $i \in \mathbb{R}^{\mathcal{I}}$. This corresponds to a state function that is invariant in $\mathcal{I}$: the input corresponds to values that are true by observation and therefore may not be changed. We can describe the behaviour of such a network by a *free energy function*

$$F : \mathbb{R}^{\mathcal{V}} \to \mathbb{R} \tag{21}$$

where a low $F(s)$ hints at a calmer and therefore more probable state. A popular choice of F is a (log-)likelihood function and a sensible invariant state function may be the *global* optimum

$$\begin{aligned} T_F^g &: \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{V}} \\ T_F^g(i)(\mathcal{I}) &:= i \\ T_F^g(i)(\mathcal{V} \setminus \mathcal{I}) &:= \arg\min_{s \in \mathbb{R}^{\mathcal{V}\setminus\mathcal{I}}} F\begin{pmatrix} s \\ i \end{pmatrix} \end{aligned} \tag{22}$$

We often consider as state step a gradient descent

$$T_F^{\text{step},\alpha} : \mathbb{R}^{\mathcal{V}} \to \mathbb{R}^{\mathcal{V}} \qquad T_F^{\text{step},\alpha}\begin{pmatrix} s \\ i \end{pmatrix} := \begin{pmatrix} s + \alpha \frac{\partial F}{\partial s}\begin{pmatrix} s \\ i \end{pmatrix} \\ i \end{pmatrix} \tag{23}$$

and as state function some iterative application of gradient descent, for instance

$$T_F^{\lim,\alpha} : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{V}} \qquad T_F^{\lim,\alpha}(i) := \lim_{n\to\infty} \left(T_F^{\text{step},\alpha}\right)^n \circ \text{In}_s \tag{24}$$

where we initialize the gradient descent by some values $s \in \mathbb{R}^{\mathcal{V}\setminus\mathcal{I}}$

$$\text{In}_s : \mathbb{R}^{\mathcal{I}} \to \mathbb{R}^{\mathcal{V}} \qquad \text{In}(i) := \begin{pmatrix} s \\ i \end{pmatrix} \tag{25}$$

2.7 (Network State and Structure) Regardless of these applied questions, the general considerations demonstrate an important principle: while we have formally defined state functions on top of a network structure, these considerations often precede specification of functional or even general structure. In this context, balancedness is more a property of a network structure than of a state function. (Of course, $T_F^{\lim,\alpha}$ is only balanced for an appropriately defined structure if $T_F\text{step}, \alpha$ has a fixed point it reaches by repeated application.) This is also the natural order of considerations in the following section where I will introduce predictive coding as a free energy approach.

References

[1] Karl Friston. "A theory of cortical responses". In: *Philosophical Transactions of the Royal Society B* 360 (2005), pp. 815–836. DOI: 10.1098/rstb.2005.1622.

[2] Erkam Guresen and Gulgun Kayakutlu. "Definition of artificial neural networks with comparison to other networks". In: *Procedia Computer Science* 3 (2011), pp. 426–433. DOI: 10.1016/j.procs.2010.12.071.

[3] Dominic Hyde and Diana Raffman. *Sorites Paradox*. 2014. URL: https://plato.stanford.edu/archives/win

[4] D. Mumford. "On the computational architecture of the neocortex - II The role of cortico-cortical loops". In: *Biological Cybernetics* 66.3 (1992), pp. 241–251. ISSN: 03401200. DOI: 10.1007/BF00198477. arXiv: arXiv:1408.1149.

[5] Dale Purves et al. *Neuroscience*. 5th ed. Sinauer Associates, Inc., 2012.

[6] Rajesh P N Rao and Dana H Ballard. "Predictive Coding in the Visual Cortex: a Functional Interpretation of Some Extra- classical Receptive-field Effects". In: *Nature Neuroscience* 2.1 (1999). DOI: 10.1038/4580.

[7] Raúl Rojas. *Neural Networks*. Berlin: Springer-Verlag, 1996.