A bipotential cost for a neural network

Marius Buliga
mbuliga.github.io

10.06.2021

[Based on the chorasimilarity post [1].]

This is a note about a simple use of convex analysis in relation with neural networks. There are many points of contact between convex analysis and neural networks, but I have not been able to locate this one.

Let's start with a **directed graph** with set of nodes N (these are the neurons) and a set of directed bonds B. Each bond has a source and a target, which are neurons, therefore there are source and target functions

$$s : B \to N \ , \ t : B \to N$$

so that for any bond $x \in B$

- the neuron $a = s(x)$ is the source of the bond,
- the neuron $b = t(x)$ is the target of the bond.

For any neuron $a \in N$:

- let $out(a) \subset B$ be the set of bonds $x \in B$ with source $s(x) = a$,
- let $in(a) \subset B$ be the set of bonds $x \in B$ with target $t(x) = a$.

A **state** of the network is a function

$$u : B \to \mathbb{V}^*$$

where $\mathbb{V}^*$ is the dual of a real vector space $\mathbb{V}$. I'll suppose that $\mathbb{V}$ and $\mathbb{V}^*$ are dual topological vector spaces, with duality product denoted by

$$(u, v) \in V \times \mathbb{V}^* \mapsto \langle v, u \rangle$$

such that any linear and continuous function from $\mathbb{V}$ to the reals is expressed by an element of $\mathbb{V}^*$ and, similarly, any linear and continuous function from $\mathbb{V}^*$ to the reals is expressed by an element of $\mathbb{V}$.

A simple example is

$$\mathbb{V} = \mathbb{V}^*$$

to be finite euclidean vector space with the euclidean scalar product denoted with the $\langle , \rangle$ notation.

A **weight** of the network is a function

$$w : B \to Lin(\mathbb{V}^*, \mathbb{V})$$

Usually the state of the network is described by a function which associates to any bond $x \in B$ a real value $u(x)$. This corresponds to the choice

$$\mathbb{V} = \mathbb{V}^* = \mathbb{R}$$

and $\langle v, u\rangle = uv$. A linear function from $\mathbb{V}^*$ to $\mathbb{V}$ is just a real number w.

The **activation function** of a neuron $a \in N$ gives a relation between the values of the state on the input bonds and the values of the state of the output bonds: any value of an output bond is a function of the weighted sum of the values of the input bonds. Usually (but not exclusively) this is an increasing continuous function.

The integral of an increasing continuous function is a convex function. I'll call this integral the **activation potential** ϕ (we suppose it does not depends on the neuron, for simplicity). For any neuron $a \in N$ and for any bond $y \in out(a)$ we have

$$u(y) = D\phi\left(\sum_{x \in in(a)} w(x)u(x)\right)$$

This relation generalizes to: for any neuron $a \in N$ and for any bond $y \in out(a)$ we have

$$u(y) \in \partial\phi\left(\sum_{x \in in(a)} w(x)u(x)\right) \tag{1}$$

where $\partial\phi$ is the s**ubgradient** of a convex and lower semicontinuous activation potential

$$\phi : V \to \mathbb{R} \cup \{+\infty\}$$

Written like this, we are done with smoothness assumptions about the activation potential, which is one of the strong features of convex analysis. This subgradient relation also explains the maybe strange definition of states and weights with the help of dual vector spaces $\mathbb{V}$ and $\mathbb{V}^*$.

We want to express the subgradient relation (1) as the minimum of a cost function. For this, remark that to any convex function ϕ is associated a **sync** (means "syncronized convex function", notion introduced in [2]).

$$c : V \times \mathbb{V}^* \to \mathbb{R} \cup \{+\infty\}$$

$$c(u, v) = \phi(u) + \phi^*(v) - \langle v, u\rangle$$

where ϕ^* is the **Fenchel dual** of the function ϕ, defined by

$$\phi^*(v) = \sup\{\langle v, u\rangle - \phi(u)\}$$

This sync has the following properties:

- it is convex in each argument
- $c(u, v) \geq 0$ for any $(u, v) \in V \times \mathbb{V}^*$
- $c(u, v) = 0$ if and only if $v \in \partial\phi(u)$.

With the sync we can produce a cost associated to the neuron: for any $a \in N$, the contribution to the cost of the state u and of the weight w is

$$\sum_{y \in out(a)} c\left(\left(\sum_{x \in in(a)} w(x)u(x)\right), u(y)\right)$$

The total **cost function** $C(u, w)$ is

$$C(u, w) = \sum_{a \in N} \sum_{y \in out(a)} c\left(\left(\sum_{x \in in(a)} w(x)u(x)\right), u(y)\right) \tag{2}$$

and it has the following properties:

- $C(u,w) \geq 0$ for any state u and any weight w
- $C(u,w) = 0$ if and only if for any neuron $a \in N$ and for any bond $y \in out(a)$ we have

$$u(y) \in \partial\phi(\sum_{x \in in(a)} w(x)u(x))$$

Example:

- take ϕ to be the **softplus** function $\phi(u) = \ln(1 + \exp(x))$
- then the activation function (i.e. the subgradient) is the **logistic function**
- and the Fenchel dual of the softplus function is the (negative of the) **binary entropy** $\phi^*(v) = v\ln(v) + (1-v)\ln(1-v)$ (extended by 0 for $v = 0$ or $v = 1$ and equal to $+\infty$ outside the closed interval $[0,1]$).

References

[1] M. Buliga, An exercice with convex analysis and neural networks (2016)

[2] M. Buliga, G. de Saxcé, C. Vallée, Blurred maximal cyclically monotone sets and bipotentials, *Analysis and Applications* **8**, no. 4, 1-14 (2010)