ARTIFICIAL NEURAL NETWORKS ON GRADED VECTOR SPACES

T. SHASKA

Department of Mathematics and Statistics,
College of Liberal Arts and Sciences
Oakland University, Rochester, MI, 48326

Abstract. We develop new artificial neural network models for graded vector spaces, which are suitable when different features in the data have different significance (weights). This is the first time that such models are designed mathematically and they are expected to perform better than neural networks over usual vector spaces, which are the special case when the gradings are all 1s.

Contents

E-mail address: shaska@oakland.edu.

2020 *Mathematics Subject Classification.* xx, yy.

Key words and phrases. artificial neural networks, equivariant networks, weighted varieties, weighted heights.

1. Introduction

Artificial neural networks are widely used in artificial intelligence for a variety of problems, including problems that rise from pure mathematics. A neural network model is a function $f : k^n \to k^m$, for some feld k and in the majority of cases $k = \mathbb{R}$. Many different architectures and models are used for such networks. The coordinates of $\mathbf{v} \in k^n$ are called *input features* and the coordinates of the vector $\mathbf{u} = f(\mathbf{v})$ the *output features*.

There are many scenarios when the input features are characterized by different values from some set, say I. For example, if the entries of the data are document and each one has a different significance and could be associated with different values. Consider for example if $\mathbf{v} = [x_0, \ldots, x_n]$ we can assign to any x_i some value $\mathbf{wt}(x_i) \in I$. Such values are called *weights* . A vector space in which coordinates of each vector are assigned some other value are known in mathematics as graded vector spaces (cf. Section 4). In this paper we investigate whether one can design neural networks over such graded vector spaces. One can think of many scenarios where neural networks defined over graded vector spaces can make a lot os sense from the applications point of view.

Our motivation came from studying the weighted projective space $\mathbb{WP}_{(2,4,6,10),\mathbb{Q}}$ which is the moduli space of genus two curves; see [10], in which case the weights are positive integers. The space of homogenous polynomials graded by their degree is a classical example of such graded vector spaces, when again the grading is done over the set of positive integers.

If one intends to carry the theory of neural networks to such graded vector spaces there are some mathematical obstacles that need to be cleared. Are there linear maps between such spaces? How will the activation functions look like? Will such graded or weighted neural networks have any advantages over the classical neural networks?

This paper is organized as follows. In Section 2 we give the mathematical background of artificial neural networks. We briefly define group action on sets, invariant and equivariant maps, quotient spaces, group and quotient representations, tensor products, topological groups, and state the Clebsch-Gordan decomposition. While some of these definitions are basic knowledge for mathematicians, they become necessary since this paper is intended to a larger audience of the AI community. Section 2 is a prelude to defining equivariant neural networks what we intend to develop for the analog of such networks over graded vector spaces.

In Section 3 we give the basic definitions of equivariant neural networks. We define convolutional neural networks or translation equivariant networks, integral

transforms, square integrable functions, regular translation intertwiners, and describe some of the properties of the translation equivariant local pooling operations. Part of Section 3 are also affine group equivariance and steerable Euclidean convolutional neural networks. For more details on such new and exciting topics the reader can check the wonderful book [12].

In many ways we want to reproduce the results of Section 3 for neural networks over graded vector spaces, but the upshot is to go even further and define such neural networks that are equivariant under coordinate changes (i.e. work for weighted projective spaces) and to do this over any field k so we can study not just applications from everyday life, but to use such neural networks to study arithmetic applications (i.e. when k is a number field), cryptography and cybersecurity (when k is a finite field), etc. Of course, this is probably unrealistic currently since such methods are not fully understood even on classical neural networks.

In Section 4 we go over the mathematical foundations of graded vector spaces. We define gradations, graded linear maps, operations on graded vector spaces, inner graded vector spaces, and discuss how to define a norm on such spaces. Defining a norm is very important since it will be based on this norm that one would define a cost function for the neural network. An adjusted homogenous norm seems as the best option to capture the significance of the weights, similar to the discussion on linear bundles and weighted heights in [8]. This is open to further investigation.

In Section 5 we define graded neural networks, graded activation functions. In general a graded neural network is defined, as expected, as a neural network which handles data where every input feature has a certain weight. It seems as under mild conditions, we can replicate all the machinery of the artificial neural networks to work for such artificial graded neural networks. It is worth pointing out that when the weights are all ones the graded neural network is just the usual neural network. It is interesting both mathematically and from the application point of view to understand the performance of such neural networks and whether they perform better for certain applications.

From the mathematical point of view many questions arise, but the main one is the understanding of the geometry of weighted projective spaces. In view of [8–10] understanding the geometry of such spaces possibly could shed light to many intriguing arithmetic questions on weighted projective varieties.

2. Mathematical foundations of artificial neural networks

In this section we establish the notation and give basic definitions of equivariant neural networks. We assume the reader has basic knowledge on the subject on the level of [12], [7] Throughout this paper k denotes a field, $\mathbb{A}^n(k) := k^n$ the affine space, and $\mathbb{P}^n(k)$ the projective space over k.

2.1. **Artificial Neural Networks.** Let the input vector be $\mathbf{x} = (x_0, \dots, x_m)$ and the output say some $\mathbf{y} = (y_0, \dots, y_n)$. We denote by $\mathcal{X}$ the space of in-features and $\mathcal{Y}$ the space of out-features. A **neuron** is a function $f : k^n \to k$ such that

$$f(\mathbf{x}) = \sum_{i-0}^{n} w_i x_i + b,$$

where $b \in k$ is a constant called **bias**. We can generalize neurons to tuples of neurons via

$$\begin{aligned} L: \ & k^n \to k^n \\ & \mathbf{x} \to (f_0(\mathbf{x}), \dots, f_n(\mathbf{x})) \end{aligned}$$

Then L is a function given by

$$L(\mathbf{x}) = W \cdot \mathbf{x} + \mathbf{b},$$

where W is an $n \times n$ matrix (of weights) with integer entries and $\mathbf{b} \in k^n$. A non-linear function $g : k^n \to k^n$ is called an **activation function**.

A **network layer** is a function

$$\begin{aligned} & k^n \to k^n \\ & \mathbf{x} \to g\left(W \cdot \mathbf{x} + \mathbf{b}\right) \end{aligned}$$

for some g some activation function. A **neural network** is the composition of many layers. The i-th layer

$$\begin{aligned} \cdots \longrightarrow k^n \xrightarrow{L_i} k^n \longrightarrow \cdots \\ \mathbf{x} \longrightarrow L_i(\mathbf{x}) = g_i\left(W_i \mathbf{x} + \mathbf{b}_i\right), \end{aligned}$$

where g_i, W_i, and $\mathbf{b}_i$ are the activation, matrix, and bias of the corresponding to this layer.

After m layers the output (predicted values) will be denoted by $\hat{\mathbf{y}} = [\hat{y_1}, \dots, \hat{y_n}]^t$, where

$$\hat{\mathbf{y}} = L_m\left(L_{m-1}\left(\dots\left(L_1(\mathbf{x})\right)\dots\right)\right),$$

while the true values by $\mathbf{y} = [y_1, \dots, y_n]^t$. The composition of all layers is called the **model function**, say

$$\mathfrak{M} : \mathcal{X} \to \mathcal{Y}$$

2.2. **Symmetries.** Assume that the input has symmetries. The simplest one could be **permuting coordinates**, but other not so obvious symmetries could be present as well.

Example 1 (Symmetric polynomials)**.** *Consider the following:* $\mathbf{x} = (\alpha_1, \dots, \alpha_n)$ *and* $\mathbf{y} = (y_0, \dots, y_{n-1})$ *where coordinates of* $\mathbf{y}$ *are coefficients of the polynomial*

$$F(x) := \prod_{i=1}^{n} (x - \alpha_i)$$

Obviously permuting roots $\alpha_1, \ldots, \alpha_n$ does not affect the outcome here, which is the set of ***elementary symmetric polynomials*** $\mathbf{y} = (s_0, \ldots, s_{n_1})$ *well known in algebra. In this case,*

$$\begin{aligned} s_1 &= \sum_{i=1}^{n} \alpha_i = \alpha_1 + \alpha_2 + \cdots + \alpha_n, \\ s_2 &= \sum_{i \neq j}^{n} \alpha_i \alpha_j, \\ &\vdots \\ s_n &= \prod_{i=1}^{n} \alpha_i = \alpha_1 \cdots \alpha_n \end{aligned}$$

We can generalize this concept by **group actions**, which is a well understood concept from abstract algebra. The symmetric group S_n **acts** on $\{\alpha_1, \ldots, a_n\}$ by permuting the roots. Notice that **symmetric polynomials** $s_0, \ldots, s_n$ are unchanged (**invariant**) under this action.

Can we use this idea for neural networks? In other words, if a model network is given by $\mathfrak{M} : \mathcal{X} \to \mathcal{Y}$ and a group G acts on $\mathcal{X}$ when can we use this action to get a more efficient model? What about if we have a group G acting not only on the space of in-features $\mathcal{X}$, but also on the space of out-features $\mathcal{Y}$? We will explore what conditions have to be met by these actions and the model so that we can make use of it. This lead to two interesting types of neural networks: **invariant networks** and **equivariant networks**.

2.3. **Groups acting on sets.** Let $\mathcal{X}$ be a set and G a group. We say that the group G **acts** on $\mathcal{X}$ if there is a function

$$\blacktriangleright\colon\ G \times \mathcal{X} \to \mathcal{X} \qquad \text{such that} \qquad (g, x) \to g \blacktriangleright x$$

which satisfies the following properties:

i) $e \blacktriangleright x = x$ for every $x \in \mathcal{X}$
ii) $g \blacktriangleright (h \blacktriangleright x) = (gh) \blacktriangleright x$, for every $g, h \in G$.

The set $\mathcal{X}$ is called a G**-set**. When there is no confusion $g \blacktriangleright x$ is simply denoted by gx. Let G acts on $\mathcal{X}$ and $x, y \in \mathcal{X}$. We say that x and y are G**-equivalent** if there exists $g \in G$ such that $gx = y$. If two elements are G-equivalent, we write $x \sim_G y$ or $x \sim y$.

Proposition 1. *Let $\mathcal{X}$ be a G -set. Then, G-equivalent is an* ***equivalence relation*** *in $\mathcal{X}$.*

The **kernel** of the action is the set of elements

$$\ker(f) \ = \ \{g \in G \ \mid \ gx = x, \ \text{for all } x \in \mathcal{X}\}$$

For $x \in \mathcal{X}$, the **stabilizer** of $x \in G$ is defined as

$$\mathrm{Stab}_G(x) = \{g \in G \,|\, gx = x\}$$

sometimes denoted by G_x. The stabilizer $\mathrm{Stab}_G(x)$ is a subgroup of G.

Lemma 1. *Let $\mathcal{X}$ be a G -set and assume that $x \sim y$. Then, the stabilizer $Stab_G(x)$ is isomorphic to the stabilizer $Stab_G(y)$.*

The action of G on $\mathcal{X}$ is called **faithful** if its kernel is the identity. The **orbit** of $x \in \mathcal{X}$ (or G-orbit) is the set

$$\mathrm{Orb}(x) = \{gx \in \mathcal{X} \mid g \in G\}$$

An action is called **transitive** if for every $x, x \in X$, there is $g \in G$ such that $x = gy$.

Lemma 2. *Let G act on a set $\mathcal{X}$ and $x \in \mathcal{X}$. Then, the cardinality of the orbit* $\mathrm{Orb}(x)$ *is the index of the stabilizer* $|\mathrm{Orb}(x)| = [G : Stab_G(x)]$.

A G-set is transitive if it has only one G-orbit. This is equivalent with the above definition of the transitive. Let $\mathcal{X}$ be a finite G-set and $\mathcal{X}_G$ the set of **fixed points** in $\mathcal{X}$ (sometimes **set of invariants**)

$$\mathcal{X}_G = \{x \in \mathcal{X} : gx = x \text{ for every } \ g \in G\}.$$

Since the orbits partition $\mathcal{X}$ we have

$$|\mathcal{X}| = |\mathcal{X}_G| + \sum_{i=k}^{n} |\mathrm{Orb}(x_i)|,$$

where $x_k, \ldots, x_n$ are representative of distinct orbits of $\mathcal{X}$.

For any $g \in G$ the set of **fixed points** of g in $\mathcal{X}$, which we denote with $\mathcal{X}_g$, is the set of all points $x \in \mathcal{X}$ such that $gx = x$. Thus,

$$\mathcal{X}_g = \{x \in \mathcal{X} \mid gx = x\}$$

Theorem 1 (Orbit counting theorem)**.** *Let G be a finite group acting on $\mathcal{X}$. If N is number of orbits, then*

$$N = \frac{1}{|G|} \sum_{g \in G} |X_g|.$$

Hence, the number of orbits is equal to the average number of points fixed by an element of G.

Corollary 1. *Let G be a finite group and $\mathcal{X}$ a finite set such that $|\mathcal{X}| > 1$. If G acts on $\mathcal{X}$ transitively then there exists $\tau \in G$ with no fixed points.*

Proof. Let $|G| = n$. Since the action is transitive then there is only one G-orbit. Form the above theorem we have that

$$|G| = F(1_G) + F(g_1) + \cdots F(g_n) = |\mathcal{X}| + \cdots$$

If $F(\tau) \geq 1$ for all $\tau \in G$ then

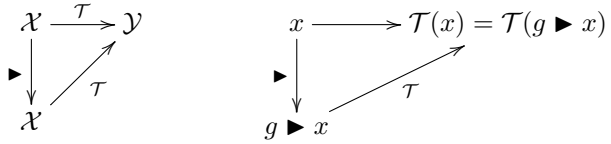
$$|G| = |\mathcal{X}| + \sum_{\sigma \in G} F(\tau) \geq |\mathcal{X}| + (n-1)$$

Thus, $|G| > n$ which is a contradiction. Hence, there must be some $\tau \in G$ such that $F(\tau) = 0$. □

2.4. **Invariant and equivariant maps.** From now on G acts on $\mathcal{X}$ via $\blacktriangleright$: $G \times \mathcal{X} \to \mathcal{X}$. A function $\mathcal{T} : \mathcal{X} \to \mathcal{Y}$ is called **G-invariant** if and only if,

$$\mathcal{T}(g \blacktriangleright x) = \mathcal{T}(x), \quad \forall g \in G, \forall x \in \mathcal{X}$$

In other words,

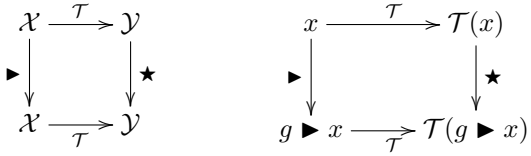


Assume now that G also acts on $\mathcal{Y}$, say G acts on $\mathcal{Y}$ as

$$\bigstar : G \times \mathcal{Y} \to \mathcal{Y}, \quad (g, y) \to g \bigstar y$$

Then, $\mathcal{T} : \mathcal{X} \to \mathcal{Y}$ is called **G-equivariant** if

$$\mathcal{T}(g \blacktriangleright x) = g \bigstar \mathcal{T}(x) \ \ \forall g \in G, \forall x \in \mathcal{X}$$



2.5. **Quotient spaces.** The set of orbits (left) of G acting on $\mathcal{X}$ is denoted by

$$G \backslash \mathcal{X} := \{\mathrm{Orb}(x) \mid x \in \mathcal{X}\}$$

and is called a **quotient space**. The corresponding **quotient map** is called the map

$$\pi : \mathcal{X} \to G \backslash \mathcal{X}, \quad x \to \mathrm{Orb}(x)$$

Notice that in the case of right action, the symbol $\mathcal{X}/G$ is used for the quotient space.

2.6. **Group representations.** Let V be a vector space (finite dimension) over a field k. $\mathrm{GL}(V)$ the **general linear group** of V (i.e., group of invertible linear maps $L : V \to V$). Let G a locally compact group (i.e., finite groups, compact groups, Lie groups are all locally compact) A **linear representation** of G on V is a tuple (ρ, V) such that

$$\rho : G \to \mathrm{GL}(V)$$

is a group homomorphism. V is called the **representation space**. Sometimes (ρ_G, V) is used. If $V = k^n$ then $\forall g \in G$, we have $\rho(g) \in \mathrm{GL}_n(k)$, so $\rho(g)$ is an $n \times n$ invertible matrix when a basis in V is chosen.

Any G-representation (ρ, V) defines an action

$$\triangleright : G \times V \to V, \qquad (g, v) \to \rho(g)v$$

Conversely, from any linear G action $\triangleright : G \times V \to V$ we get a representation

$$\rho_{\triangleright} : \quad G \to \mathrm{GL}(V), \qquad g \to L_g$$

where $L_g(v) = g \triangleright (v)$, for all $v \in V$. Hence, there is a one to one correspondence between G-representations on V and (linear) G-group actions on V. Here are some common representations (we will skip details)

(i) trivial representation ($\rho(g) = \mathrm{id}_V$)
(ii) standart representation ($\rho(g) = g$)
(iii) tensor representation
(iv) regular representation

Let (ρ_1, V_1) and (ρ_2, V_2) be two given G-representations. Let $V_1 \oplus V_2$ be the direct sum and

$$\alpha : \mathrm{GL}(V_1) \times \mathrm{GL}(V_2) \to \mathrm{GL}(V_1 \oplus V_2)$$
$$(v_1, v_2) \to v_1 \oplus v_2$$

Then we can define the **direct sum representation** as given by $\rho_1 \oplus \rho_2 = \alpha \circ (\rho_1 \times \rho_2)$ as

$$G \xrightarrow{\rho_1 \times \rho_2} \mathrm{GL}(V_1) \times \mathrm{GL}(V_2) \xrightarrow{\alpha} \mathrm{GL}(V_1 \oplus V_2)$$
$$g \to (\rho_1(g), \rho_2(g)) \to \rho_1(g) \oplus \rho_2(g)$$

The matrix representation of it (when bases for V_1 and V_2 are chosen) is

$$(\rho_1 \oplus \rho_2)(g) = \begin{pmatrix} \rho_1(g) & 0 \\ 0 & \rho_2(g) \end{pmatrix}$$

2.7. **Quotient representation:** Let $W \subset V$ be a subspace and V/W the quotient space. G acts on V/W via

$$G \times W \to W, \quad (g, v + W) \to \rho(g)v + W$$

With this action V/W is called the **quotient representation** of V under W

Let (ρ, V) be a G-representation and consider a subspace $W \subset V$. W is called **invariant** if it is closed under the action of ρ, i.e., $\rho(g)w \in W$, for any $w \in W$ and $g \in G$. Hence the restriction of ρ in W is a homomorphism:

$$\rho_W : G \longrightarrow \mathrm{GL}(W)$$

Definition. 2. *A representation (ρ, V) is called **irreducible representation** (irrep) it it has only the two trivial subrepresentations $W = V$ and $W = \{0\}$.*

Example 2. *Of course the fact that (ρ, V) is irreducible or not depends on the field k. For example, let $G = \mathrm{SO}(2, \mathbb{R})$. Its real valued irreducible representation are*

$$\rho_m^{G,\mathbb{R}}(\phi) = \begin{pmatrix} \cos(m\phi) & -\sin(m\phi) \\ \sin(m\phi) & \cos(m\phi) \end{pmatrix}, \quad m \in \mathbb{N}$$

However, over $\mathbb{C}$

Let (ρ_1, V_1) nd (ρ_2, V_2) be G-representations. An **intertwiner** between them is an equivariant linear map

$$L : V_1 \longrightarrow V_2, \quad \text{which satisfies} \quad L \circ \rho_1(g) = \rho_2(g) \circ L$$

The space of intertwines is a vector space denoted by $\mathrm{Hom}_G(V_1, V_2)$.

Example 3. *Convolutions are intertwiners*

Definition. 3 (Equivalent (isomorphic) representations)**.** *Two representations (ρ_1, V_1) and (ρ_2, V_2) are called **equivalent** or **isomorphic** if there exists an isomorphism*

$$L : V_1 \longrightarrow V_2, \quad \textit{such that} \quad L \circ \rho_1(g) = \rho_2(g) \circ L, \quad \textit{for all} \quad g \in G$$

This is equivalent as matrix representations $\rho_1(g)$ and $\rho_2(g)$ are similar for every $g \in G$.

Definition. 4 (Endomorphisms)**.** *Intertwines from (ρ, V) to itself are called **endomorphisms**. In other words, an endomorphism is a linear map $L : V \to V$ such that*

$$L \circ \rho(g) = \rho(g) \circ L,$$

for all $g \in G$. The endomorphism space is denoted by $\mathrm{End}_G(V) = \mathrm{Hom}_G(V, V)$

Lemma 3 (Schur's lemma)**.** *Let (ρ_1, V_1) and (ρ_2, V_2) be G-irreps over $k = \mathbb{R}$ or $k = \mathbb{C}$. Then:*

1. *If (ρ_1, V_1) and (ρ_2, V_2) are not isomorphic, then there is no (non-trivial) intertwiner between them*
2. *If $(\rho_1, V_1) = (\rho_2, V_2) =: (\rho, V)$ are identical, any intertwiner is an isomorphism and*
   (a) *If $k = \mathbb{C}$ then*

$$\rho = \lambda \,\mathrm{id}_v, \quad \text{for} \quad \lambda \in \mathbb{C}$$

   (b) *If $k = \mathbb{R}$, then $\mathrm{End}_G(V)$ has dimension 1, 2, or 4 depending on whether (ρ, V) is real, complex, or quaternionic type.*

2.8. **Tensor products.** The **tensor product** $V \otimes_k W$ of two vector spaces V and W over a field k is the k-vector space based on elements $v \otimes w$, and with relations for all $k \in \mathbb{C}$, $v \in V$, $w \in W$

$$\begin{aligned}(v_1 + v_2) \otimes w &= v_1 \otimes w + v_2 \otimes w \\ v \otimes (w_1 + w_2) &= v \otimes w_1 + v \otimes w_2 \\ (k \cdot v) \otimes w &= v \otimes (k \cdot w) = k \cdot (v \otimes w)\end{aligned}$$

If $\{v_1, \ldots, v_n\}$ is a basis for V and $\{w_1, \ldots, w_m\}$ is a basis for W, then $\{v_i \otimes w_j\}$ is a basis for $V \otimes W$.

Let (ρ_1, V_1) and (ρ_2, V_2) be two representations of a group G. The **tensor product representation** $(\rho_1 \otimes \rho_2, V_1 \otimes V_2)$ is defined as

$$(\rho_1 \otimes \rho_2)(g)(v_1 \otimes v_2) := \rho_1(g)(v_1) \otimes \rho_2(g)(v_2)$$

and extended to all vectors in $V \otimes W$ by linearity. It has dimension $\dim(V_1) \cdot \dim(V_2)$.

If $\dim V$, $\dim W < \infty$, then there is a natural isomorphism of vector spaces (preserving G-actions, if defined) from $W \otimes V$ to $\mathrm{Hom}(V, W)$.

2.9. **Topological groups.** A **topological group** is a group which is also a topological space and for which the group operation is continuous. It is called **compact** if it is so as a topological space.

A representation of a topological group G on a finite-dimensional vector space V is a **continuous** group homomorphism

$$\rho : G \to \mathrm{GL}(V),$$

with the topology of $\mathrm{GL}(V)$ inherited from the space $\mathrm{End}(V)$ of linear self-maps. Notice that now, we can naturally replace $\frac{1}{|G|} \sum_{g \in G}$ with $\int_G dg$ (see Haar measure, Borel measure, etc).

2.10. **Clebsch-Gordan decomposition.** Let (ρ_1, V_1) and (ρ_2, V_2) be unitary irreducible G-representations of a compact group G. Let $\widehat{G}$ denote the set of isomorphisms classes f unitary irreducible representations of V

Their tensor product $\rho_1 \otimes \rho_2$ is no necessary irreducible. However, there exists an isomorphism

$$\phi : V_l \otimes V_k \longrightarrow \oplus_{j \in \widehat{G}} \oplus_{s=1}^{m_{j,lk}} V_j$$

such that V_j are irreducible, where $m_{j,lk}$ the multiplicity of irreducible representation j in the tensor product of irreducible representations l and k. This is called **Clebsch-Gordan decomposition**.

Consider a choice of basis for V_j, say

$$\{e_j^1, e_j^2, \ldots, e_j^{\dim V_j}\}$$

Hence, $e_l^m \otimes e_k^n$ are basis elements in $V_l \otimes V_k$ which are mapped to the basis elements of $\oplus_{s=1}^{m_{j,lk}} V_j$. Hence we get a matrix associated to ϕ, its elements are called **Clebsch-Gordan coefficients**

A real-valued function $f : \mathbb{R} \to \mathbb{R}$ is called a **square-integrable function** if

$$\int_{-\infty}^{\infty} |f(x)|^2 \, dx < \infty$$

Let $L^2(\mathbb{R}$ denote the **space of square-integrable functions**. $L^2(\mathbb{R})$ is a vector space and a Hilbert space.

Theorem 5 (Peter-Weyl)**.** *The space of square integrable functions on G is an Hilbert space, direct sum over finite dimensional irreducible representations V*

$$L^2(G) \cong \oplus \operatorname{End}(V)$$

where

$$f \to \int_G f(g) \cdot \rho_V(g) \, dg$$

The inverse map sends $\phi \in \operatorname{End}(V)$ to the function

$$g \to Tr_V(\rho_V(g)^* \phi)$$

Let G be a compact group and $L_k^2(G/H)$ as above. Denote by : $\widehat{G}$ is the set of isomorphism classes of G irreducible representations, $\widehat{(\cdot)}$ is a topological closure, and by $m_j \leq \dim V_j$ is the multiplicity of irreducible representation of V_j in $L_k^2(G/H)$.

Theorem 6. *The quotient representation $\left(\rho_{quot}^{G/H}, L_k^2(G/H)\right)$ decomposes into irreducible subrepresentations*

$$L_k^2(G/H) \cong \widehat{\oplus}_{j \in \widehat{G}} \oplus_{i=1}^{m_j} V_j$$

Notice that if $k = \mathbb{C}$ and $H = \{e\}$ then $m_j = \dim V_j$.

3. Equivariant Neural Networks

Let us see now how to construct some Equivariant Neural Networks. Let $\mathcal{X}$ be the **space of input features** and $\mathcal{Y}$ the **space of output features**. Let $\mathfrak{M} : \mathcal{X} \to \mathcal{Y}$ be a **model**. Usually we want to approximate some **target function**

$$\mathcal{T} : \mathcal{X} \to \mathcal{Y}$$

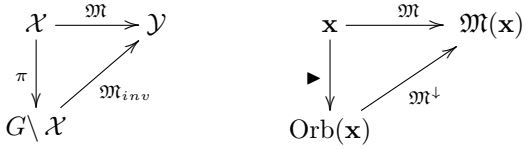
Let $\mathcal{H}_{full}$ denote the space of all models under consideration during the training, we call this the **hypothesis space**. Assume G acts on $\mathcal{X}$ and $\mathcal{Y}$ as

$$G \times \mathcal{X} \to \mathcal{X}, (g, x) \to g \blacktriangleright x \quad \text{and} \quad G \times \mathcal{Y} \to \mathcal{Y}, (g, y) \to g \bigstar y$$

We denote by $\mathcal{H}_{inv}$ the space of invariant models and by $\mathcal{H}_{equiv}$ the space of equivariant models. So we have

$$\mathcal{H}_{inv} \subset \mathcal{H}_{equiv} \subset \mathcal{H}_{full}$$

Consider now instead of having a network sending $\mathbf{x} \to \mathfrak{M}(\mathbf{x})$ we have one which sends $\mathrm{Orb}(\mathbf{x}) \to \mathfrak{M}(\mathbf{x})$.



So $\mathfrak{M}^{\downarrow}$ is an invariant map and $\mathcal{H}_{inv}$ can be thought of the space of models $\mathfrak{M}^{\downarrow}$.

3.1. **Equivariant neural networks.** A feed forward neural network is a sequence

$$\mathcal{X}_0 \xrightarrow{\mathcal{L}_1} \mathcal{X}_1 \xrightarrow{\mathcal{L}_2} \mathcal{X}_2 \cdots \xrightarrow{\mathcal{L}_{N-1}} \mathcal{X}_N$$

of parametrization layers $\mathcal{L}_i : \mathcal{X}_{i-1} \to \mathcal{X}_i$, where $\mathcal{X}_i$ is a feature space (vector space) and $\mathcal{L}_i$ the i-th layer. Constructing equivariant networks typically involves designing each layer to be individually equivariant. Therefore, each feature space $\mathcal{X}_i$ has it's own group action:

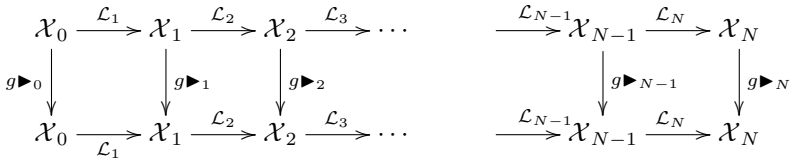
$$\blacktriangleright_i \colon G \times \mathcal{X}_i \to \mathcal{X}_i, \quad (g, x) \to g \blacktriangleright_i x.$$

In the network the input $\mathcal{X}_0$ and output X_N actions $\blacktriangleright_0$ and $\blacktriangleright_N$ are determined by learning task, while the intermediate actions are selected by the user.

For a layer $\mathcal{L}_i$ to be invariant it's input and output actions must satisfies the following,

$$\mathcal{L}_i(g \blacktriangleright_{i-1} x) = g \blacktriangleright_i \mathcal{L}_i(x), \ \forall g \in G, x \in \mathcal{X}_{i-1}$$

The visualization of equivariant neural networks is given below



3.2. **Convolution Neural Networks: translation equivariance.** A **Euclidean feature map** in d dimensions with c channels is a function $F : \mathbb{R}^d \to \mathbb{R}^c$ that assigns a c-dimensional feature vector $F(x)$ for every point $\mathbf{x} \in \mathbb{R}^d$.

Let $\mathcal{E}_{(d,c)}$ be the set of all Euclidean feature maps $\mathbb{R}^d \to \mathbb{R}^c$. A **translation group** is called the additive group of the Euclidean space $V = \mathbb{R}^d$. It acts on $\mathbb{R}^d$ by shifting (or translation)

$$(\mathbb{R}^d, +) \times \mathbb{R}^d \to \mathbb{R}^d, \quad (t, x) \to x + t$$

It induces an action on $\mathcal{E}_{(d,c)}$ via

$$(\mathbb{R}^d, +) \times \mathcal{E}_{(d,c)} \to \mathcal{E}_{(d,c)}, \quad F \to (t \blacktriangleright F)(x) = F(x - t)$$

This action is known as **regular representation**

The feature spaces of translation equivariant Euclidean Convolutional Neural networks are vector spaces

$$\mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c) := \left\{ \mathcal{E}_{d,c} : \mathbb{R}^d \to \mathbb{R}^c \mid \int_{\mathbb{R}^d} ||F(x)||^2 \, dx \right\}$$

And the translation group action on this space as described above.

A translation equivariant network between feature maps with c_{in} inputs channels and c_{out} output channels l are functions:

$$L : L^2(R^d, \mathbb{R}^{c_{in}} \longrightarrow L^2(R^d, \mathbb{R}^{c_{out}}$$

such that the following diagram commutes for $t \in (\mathbb{R}^d, +)$.

$$\begin{array}{ccc} \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{in}}) & \xrightarrow{L} & \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{out}}) \\ {\scriptstyle t\blacktriangleright}\downarrow & & \downarrow{\scriptstyle t\bigstar} \\ \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{in}}) & \xrightarrow[L]{} & \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{out}}) \end{array}$$

Linear translation equivariant functions mapping between feature maps are essentially convolutions.

3.3. **Integral transforms.** Let

$$\mathcal{I}_\kappa : \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{in}}) \to \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{out}})$$

be integral transform map that is parametrized by a **square integrable two-argument kernel** κ

$$\kappa : \mathbb{R}^d \times \mathbb{R}^d \to \mathbb{R}^{c_{in} \times c_{out}}, \ (x, y) \to \kappa(x, y)$$

defined by

$$\boxed{\mathcal{I}_\kappa(F)(x) := \int_{\mathbb{R}^d} \kappa(x, y) \, F(y) \, dy}$$

Let

$$\mathcal{K} : \mathbb{R}^d \to \mathbb{R}^{c_{in} \times c_{out}}, \quad \Delta x \to \mathcal{K}(\Delta x)$$

defined by $\mathcal{K}(\Delta_x) := \kappa(\Delta_x, 0)$.

Theorem 7 (Regular translations intertwiners are convolutions)**.** *The integral transform $\mathcal{I}_\kappa$ is equivariant if and only if the two-argument kernel κ satisfies*

$$\kappa(x + \mathbf{t}, y + \mathbf{t}) = \kappa(\mathbf{x}, \mathbf{y}), \qquad \textit{for any } \mathbf{x}, \mathbf{y}, \mathbf{t} \in \mathbb{R}^d.$$

Moreover, the integral trasform reduces to a convolution integral

$$\mathcal{I}_\kappa(F)(\mathbf{x}) = \int_{\mathbb{R}^d} K(\mathbf{x}, \mathbf{y}) F(\mathbf{y}) \, dy$$

Proof. The integral transformation form $\mathcal{I}_\kappa$ is translation equivariant if $\mathcal{I}_\kappa(t \blacktriangleright F) = t\bigstar\mathcal{I}_\kappa(F)$, for all $\mathbf{t} \in (\mathbb{R}^d, +)$. Begin with the left-hand side of the equality:

$$\mathcal{I}_\kappa(\mathbf{t} \blacktriangleright F)(x) = \int_{\mathbb{R}^d} \kappa(\mathbf{x}, \mathbf{y})(t \blacktriangleright F)(y) \, dy = \int_{\mathbb{R}^d} \kappa(\mathbf{x}, \mathbf{y}) \, F(y - \mathbf{t}) \, dy = \int_{\mathbb{R}^d} \kappa(\mathbf{x}, \tilde{\mathbf{y}} + \mathbf{t}) \, F(\tilde{\mathbf{y}}) \, d\tilde{\mathbf{y}},$$

where $\mathbf{y} = \tilde{\mathbf{y}} + \mathbf{t}$. While the right-hand side is given by

$$\mathbf{t}\bigstar\mathcal{I}_\kappa(F) = \int_{\mathbb{R}^d} \kappa(\mathbf{x} - \mathbf{t}, \mathbf{y}) \, F(\mathbf{y}) \, dy, \quad \forall \mathbf{t} \in (\mathbb{R}^d, +), \forall F \in \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^{c_{in}})$$

implies the **translation invariance constraint**

$$\kappa(\mathbf{x}+\mathbf{t},\mathbf{y}+\mathbf{t})=\kappa(\mathbf{x},\mathbf{y}),\quad \forall \mathbf{x},\mathbf{y},\mathbf{t}\in\mathbb{R}^d$$

of the neural connectivity (spatial weight sharing).

If we let $\mathbf{t}=-\mathbf{y}$ then,

$$\kappa(\mathbf{x},\mathbf{y})=\kappa(\mathbf{x}-\mathbf{y},0)=\kappa(\Delta x,0)=\mathcal{K}(\Delta x).$$

Thus, this makes the integral transform a convolution. □

3.4. **Translation equivariant bias summation.** Let $\mathbf{b}:\mathbb{R}^d\to\mathbb{R}^{c_{in}}$ be a bias field. The bias preserves the number of channels, so $c:=c_{in}=c_{out}$ Consider a bias operation

$$\begin{aligned} B_\beta:\mathcal{L}^2(\mathbb{R}^d,\mathbb{R}^c)&\to\mathcal{L}^2(\mathbb{R}^d,\mathbb{R}^c),\\ F&\to F+\mathbf{b}\end{aligned}$$

be an unconstrained bias summation that is parametrized by a square integrable bias field $\mathbf{b}:\mathbb{R}^d\to\mathbb{R}^c$

Theorem 8 (Translation equivariant bias summation)**.** *The bias summation $B_{\mathbf{b}}$ is equivariant if and only if $\mathbf{b}$ is constant (i.e., $\mathbf{b}(x)=b$ for some $b\in\mathbb{R}^c$).*

Proof. The bias summation $B_{\mathbf{b}}$ is equivariant if $B_{\mathbf{b}}(g\blacktriangleright F)=g\blacktriangleright B_{\mathbf{b}}(F),\quad \forall g\in(\mathbb{R}^d,+)$.

Begin with the left-hand side of the equality:

$$B_{\mathbf{b}}(g\blacktriangleright F)(x)=(g\blacktriangleright F)(x)+\mathbf{b}(x)=F(x-g)+\mathbf{b}(x),$$

While the right-hand side is given by

$$[g\blacktriangleright B_{\mathbf{b}}(F)](x)=[g\blacktriangleright(F+\mathbf{b})](x)=F(x-g)+\mathbf{b}(x-g),$$

Hence

$$\mathbf{b}(x)=\mathbf{b}(x-g)\ \ for\ \ \ arbitrary\ \ \ x,g\in\mathbb{R}^d.$$

The bias field is required to be translation invariant. Thus, $\mathbf{b}(x)=b\,for\,b\in\mathbb{R}^d$. □

3.5. **Translation equivariant local nonlinearities.** Let

$$S_\sigma:\mathcal{L}^2(\mathbb{R}^d,\mathbb{R}^{c_{in}})\to\mathcal{L}^2(\mathbb{R}^d,\mathbb{R}^{c_{out}}),\quad F\to S_\sigma(F)$$

defined by $S_\sigma(F)(x)=:\sigma_x(F(x))$, where σ is a spatially dependent localized nonlinearity

$$\sigma:\mathbb{R}^d\times\mathbb{R}^{c_{in}}\to\mathbb{R}^{c_{out}},\quad (x,y)\to\sigma_x(y)$$

Theorem 9 (Translation equivariant local nonlinearities)**.** *The spatially dependent localized nonlinearity operation S_σ is translation equivariant if and only if $\sigma_x=s$ for some $x\in\mathbb{R}^d$.*

Proof. The spatially dependent localized nonlinearity operation S_σ is translation equivariant if :

$$S_\sigma(\mathbf{t}\blacktriangleright F)=\mathbf{t}\bigstar S_\sigma(F),\quad \forall\mathbf{t}\in(\mathbb{R}^d,+)$$

Begin with the left-hand side of the equality:

$$S_\sigma[\mathbf{t}\blacktriangleright F](\mathbf{x})=\sigma_{\mathbf{x}}[\mathbf{t}\blacktriangleright F](\mathbf{x})=\sigma_{\mathbf{x}}[F(\mathbf{x}-g)]$$

While the right-hand side is given by

$$[\mathbf{t}\bigstar S_\sigma(F)](\mathbf{x})=S_\sigma[F(\mathbf{x}-\mathbf{t})]=\sigma_{\mathbf{x}-\mathbf{t}}[F(\mathbf{x}-\mathbf{t})]$$

Hence $s:=\sigma_{\mathbf{x}}=\sigma_{\mathbf{x}-\mathbf{t}}$ for an arbitrary $\mathbf{x},\mathbf{t}\in\mathbb{R}^d$ □

3.6. **Translation equivariant local pooling operations.** Local max pooling is a nonlinear operation that generates a feature field where the value at a point $x_0 \in \mathbb{R}^d$ is determined by taking the maximum feature value across channels within a defined pooling region $R_{x_0} \subset R^d$ centered around x_0.

$$\mathcal{P}: \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c) \to \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c), \quad F \to \max_{y \in R_x} F(y)$$

Theorem 10 (Translation equivariance local max pooling)**.** *The local max pooling operation. $\mathcal{P}$ is translation equivariant if and only if $g^{-1}R_x = R_{x-g}$ for all $x \in \mathbb{R}^d$, $g \in (\mathbb{R}, +)$.*

Proof. The local max pooling operation $\mathcal{P}$ is translation equivariant if

$$\mathcal{P}(g \blacktriangleright F) = g \blacktriangleright \mathcal{P}(F), \quad \forall x \in \mathbb{R}^d, g \in (\mathbb{R}^d, +)$$

Begin with the left-hand side:

$$\mathcal{P}(g \blacktriangleright F)(x) = \max_{y \in R_x}[g \blacktriangleright F](y) = \max_{y \in R_x}[F(y-g)] = \max_{y \in g^{-1}R_x} F(y)$$

And the right-hand side is given by

$$[g \blacktriangleright \mathcal{P}(F)](x) = \mathcal{P}[F(x-g)] = \max_{y \in R_{x-g}} F(y)$$

□

Definition. 11. *Local average pooling calculates the channel-wise average of the responses.*

$$\mathcal{P}_\alpha : \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c) \to \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c), \quad F \to \alpha \star F$$

where α is a scalar weighting kernel:

$$\alpha : \mathbb{R}^d \to \mathbb{R}.$$

Theorem 12 (Translation equivariance of local average pooling)**.** *The local average pooling operation $\mathcal{P}_\alpha$ is by construction translation equivariant.*

Proof. The local max pooling operation $\mathcal{P}$ is translation equivariant if

$$\mathcal{P}_\alpha(g \blacktriangleright F)(x) = [g \blacktriangleright \mathcal{P}_\alpha(F)](x), \qquad \forall x \in \mathbb{R}^d, g \in (\mathbb{R}^d, +)$$

$$\mathcal{P}_\alpha(g \blacktriangleright F)(x) = \max_{y \in R_x}[g \blacktriangleright F](y) = \max_{y \in R_x}[F(y-g)] = \max_{y \in g^{-1}R_x} F(y)$$

And the right-hand side is given by

$$[g \blacktriangleright \mathcal{P}(F)](x) = \mathcal{P}[F(x-g)] = \max_{y \in R_{x-g}} F(y).$$

. □

3.7. **Affine group equivariance and steerable Euclidean CNNs.** Let $G \leq \mathrm{GL}_d(\mathbb{R})$ be a given group. Affine groups $\mathrm{Aff}(G)$ are semi-direct products of translations $t \in (\mathbb{R}^d, +)$ and G, $\mathrm{Aff}(G) := (\mathbb{R}^d, +) \rtimes G$. The affine group $\mathrm{Aff}(G)$ acts on Euclidean spaces,

$$\begin{aligned} \mathrm{Aff}(G) \times \mathbb{R}^d &\to \mathbb{R}^2 \\ (tg, x) &\to gx + t \end{aligned}$$

Notice that

$$\big((th)^{-1}, x\big) \to g^{-1}(x-t)$$

3.7.1. *Euclidean feature fields and induced affine group representations.* The feature spaces of $\mathrm{Aff}(G)$-equivariant Euclidean steerable CNNs are vector spaces

$$L^2(\mathbb{R}^d, \mathbb{R}^c) := \{\mathbb{R}^d \to \mathbb{R}^c \mid \int_{\mathbb{R}^d} ||F(x)||^2 \, dx \leq \infty\}.$$

of square integrable c-channel feature fields in d spatial dimensions. The are associated to some $\rho : G \to \mathrm{GL}_c(\mathbb{R})$. The affine group acts via

$$\begin{aligned} \blacktriangleright_\rho \colon \mathrm{Aff}(G) \times \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c) &\to \mathcal{L}^2(\mathbb{R}^d, \mathbb{R}^c), \\ (tg, F) &\to tg \blacktriangleright_\rho F := \rho(g) F(tg)^{-1} \end{aligned}$$

This action it corresponds to

$$\begin{aligned} \mathrm{Ind}_G^{\mathrm{Aff}(G)} \rho : \mathrm{Aff}(G) &\to \mathrm{GL}\left(L^2(\mathbb{R}^d, \mathbb{R}^c)\right), \\ tg &\to tg \blacktriangleright_\rho (\cdot) \end{aligned}$$

known as induced representation that turns G-representations to $\mathrm{Aff}(G)$-representations. **Euclidean feature fields** are elements of induced affine group representation spaces $\mathrm{Ind}_G^{\mathrm{Aff}(G)}$. Hence, $\mathrm{Ind}_G^{\mathrm{Aff}(G)} \rho$ is a functor that turns G-representations into $\mathrm{Aff}(G)$-representations. Our goal of the next section is to explore wthether the same machinery can be constructed when we replace the affine space with a graded vector space.

A full feature space of steerable convolutional neural networks (CNNs) comprises multiple individual feature fields $F_i : \mathbb{R}^d ß \mathbb{R}^{c_i}$ of different types ρ_i and dimensionalities c_i. The composite field $F = \oplus_i F_i$ transforms according to the direct sum

$$\oplus_i \mathrm{Ind}_G^{\mathrm{Aff}(G)} \rho_i = \mathrm{Ind}_G^{\mathrm{Aff}(G)} \oplus_i \rho_i$$

and can therefore be viewed as being of type $\oplus_i \rho_i$ The block structure of the direct sum representation guarantees hereby that the individual fields f_i transform independently from each other, that is, their channels do not mix under G-transformations. The following visual illustration is [12, Fig. 4.4]

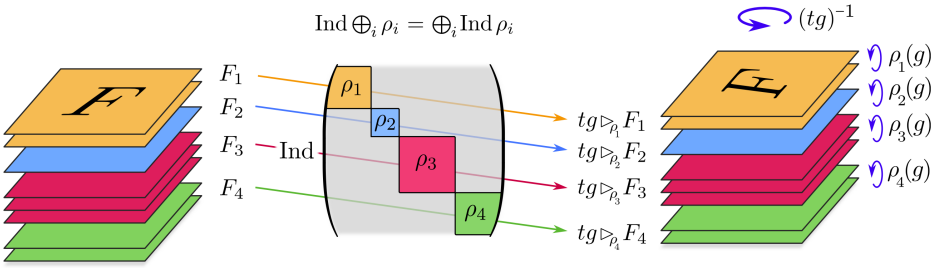


FIGURE 1. Decomposition of steerable CNNs; [12, Fig. 4.4]

The goal for the rest of this paper is to develop a similar theory of what is described in this section for artificial neural networks over graded vector spaces or even more generally over graded modules.

4. Graded vector spaces

Here we give the bare minimum of the background graded vector spaces. The interested reader can check details at [3], [7], [4] among other places.

A graded vector space is a vector space that has the extra structure of a grading or gradation, which is a decomposition of the vector space into a direct sum of vector subspaces, generally indexed by the integers. For the purposes of this paper we will focus on graded vector spaces indexed by integers, but we also give below the definition of such spaces for a general index set I.

4.1. **Integer gradation.** Let $\mathbb{N}$ be the set of non-negative integers. An $\mathbb{N}$–graded vector space, often called simply a **graded vector space** without the prefix $\mathbb{N}$, is a vector space V together with a decomposition into a direct sum of the form

$$V = \bigoplus_{n\in\mathbb{N}} V_n$$

where each V_n is a vector space. For a given n the elements of V_n are then called homogeneous elements of degree n.

Graded vector spaces are common. For example the set of all polynomials in one or several variables forms a graded vector space, where the homogeneous elements of degree n are exactly the linear combinations of monomials of degree n.

Example 4. *Let k be a field and consider $\mathcal{V}_{(2,3)}$ the space of degree 2 and 3 homogenous polynomials in $k[x, y]$. It is decomposed as $\mathcal{V}_{(2,3)} = \mathcal{V}_2 \oplus \mathcal{V}_3$, whwre $\mathcal{V}_2$ is the space of binary quadratics and V_3 the space of binary cubics. Let $\mathbf{u} = [f, g] \in \mathcal{V}_2 \oplus \mathcal{V}_3$. Then the scalar multiplication works as*

$$\lambda \star \mathbf{u} = \lambda \star [f, g] = [\lambda^2 f, \lambda^3 g]$$

We will use this example repeatedly for the rest of the paper. □

Next we give another example that was our main motivation for machine learning models over graded vector spaces.

Example 5 (Moduli space of genus 2 curves)**.** *Assume char $k \neq 2$ and C a genus 2 curve defined over k. Then C has affine equation $y^2 = f(x)$ where $f(x)$ is a degree 6 polynomial. The ismorphism class of C is determined by its invariants J_2, J_4, J_6, J_{10}, which are homogenous polynomials of degree 2, 4, 6, and 10 respectively in terms of the coefficients of C. The moduli space of genus 2 curves defined over k is isomorphic to the weighted projective space $\mathbb{WP}_{(2,4,6,10),k}$.*

4.2. **General gradation.** The subspaces of a graded vector space need not be indexed by the set of natural numbers, and may be indexed by the elements of any set I. An I-graded vector space V is a vector space together with a decomposition into a direct sum of subspaces indexed by elements i of the set I:

$$V = \bigoplus_{i\in I} V_i$$

The case where I is the ring $\mathbb{Z}/2\mathbb{Z}$ (the elements 0 and 1) is particularly important in physics. A $(\mathbb{Z}/2\mathbb{Z})$-graded vector space is known as a **supervector space**.

4.3. **Graded linear maps.** For general index sets I, a linear map between two I-graded vector spaces $f : V \to W$ is called a **graded linear map** if it preserves the grading of homogeneous elements,

$$f(V_i) \subseteq W_i, \quad \text{for all} \quad i \in I.$$

A graded linear map is also called a **homomorphism (or morphism) of graded vector spaces**, or homogeneous linear map.

When I is a commutative monoid (such as $\mathbb{N}$), then one may more generally define linear maps that are homogeneous of any degree i in I by the property

$$f(V_j) \subseteq W_{i+j}, \quad \text{for all} \quad j \in I$$

where "+" denotes the monoid operation. If moreover I satisfies the cancellation property so that it can be embedded into an abelian group A that it generates (for instance the integers if I is the natural numbers), then one may also define linear maps that are homogeneous of degree i in A by the same property (but now "+" denotes the group operation in A). Specifically, for $i \in I$ a linear map will be homogeneous of degree $-i$ if

$$f(V_{i+j}) \subseteq W_j, \ \text{ for all } \ j \in I, \ \text{ while } \ f(V_j) = 0 \ \text{ if } \ j - i \notin I$$

Let us see a simple example of a graded linear map.

Example 6. *Consider $\mathcal{V}_{(2,3)} = V_2 \oplus V_3$ as in Example 4. Then a linear map $L : \mathcal{V}_{(2,3)} \to \mathcal{V}_{(2,3)}$ satisfies*

$$L([\lambda \star \mathbf{u}]) = L([\lambda^2 f, \lambda^3 g]) = [\lambda^2 L(f), \lambda^3 L(g)] = \lambda \star [L(f), L(g)] = \lambda \star L(\mathbf{u})$$

and

$$\begin{aligned} &L\left([f,g] \oplus [f',g']\right) = L\left([f+f', g+g']\right) = [L(f) + L(f'), L(g) + L(g')] \\ &= [L(f), L(g)] \oplus [L(f'), L(g')] = L([f,g]) \oplus L([f',g']) \end{aligned}$$

We can see things more explicitly if we choose a basis for $\mathcal{V}_{(2,3)}$. Since V_2 is the space of binary quadratics $ax^2 + bxy + cy^2$ we can pick the standard basis of monomials for V_2 as $\mathcal{B}_1 = \{x^2, xy, y^2\}$. Similarly a standard basis for V_3 can be chosen as $\mathcal{B}_2 = \{x^3, x^2y, xy^2, y^3\}$. Hence, a basis for $\mathcal{V}_{(2,3)}$ can be picked as

$$\mathcal{B} = \{x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$$

For example, the polynomial $F(x,y) = \left(x^2 + xy + y^2\right) + \left(x^3 + x^2y + xy^2 + y^3\right)$ has coordinates $\mathbf{u} = [1,1,1,1,1,1,1]^t$ in this basis. □

Further details on isomorphisms of graded rings of linear transformations of graded vector spaces can be found in [3], [1], [2], and others.

Notice that the simplest graded linear map is the "multiplication" by a scalar, say $L(\mathbf{x}) = \lambda \mathbf{x}$ which has matrix representation the diagonal matrix

$$\begin{bmatrix} \lambda^{q_0} & 0 & 0 & \dots \\ 0 & \lambda^{q_1} & 0 & \vdots \\ \vdots & 0 & \ddots & 0 \\ \dots & \dots & 0 & \lambda^{q_n} \end{bmatrix}$$

4.4. **Operations over graded vector spaces.** Some operations on vector spaces can be defined for graded vector spaces as well. For example, given two I-graded vector spaces V and W, their direct sum has underlying vector space $V \oplus W$ with gradation

$$(V \oplus W)_i = V_i \oplus W_i$$

If I is a semigroup, then the tensor product of two I-graded vector spaces V and W is another I-graded vector space,

$$(V \otimes W)_i = \bigoplus_{(j,k):j+k=i} (V_j \otimes W_k)$$

We will come back back to tensor products of vectors spaces when more details are needed.

4.5. **Inner graded vector spaces.** Consider now the case when each V_i, is a finite fdimensional inner space and let $\langle \cdot, \cdot \rangle_i$ denote the corresponding inner product. Then we can define an inner product on V as follows. For $\mathbf{u} = u_1 + \ldots + u_n$ and $\mathbf{v} = v_1 + \ldots + v_n$ we define

$$\langle \mathbf{u}, \mathbf{v} \rangle = \langle u_1, v_2 \rangle_1 + \ldots + \langle u_n, v_n \rangle_n$$

which is the standart product. Then the Euclidean norm is as expected

$$\|\mathbf{u}\| = \sqrt{u_1^2 + \ldots + u_n^2}$$

If such V_i are not necessary finite dimensional then we have to assume that V_i is a Hilbert space (i.e. a real or complex inner product space that is also a complete metric space with respect to the distance function induced by the inner product). This case of Hilbert spaces is especially important in machine learning and artificial intelligence as pointed out by Thm. 5.

Obviously having a norm on a graded vector space is important for machine learning if we want to define a cost function of some type. The simpler case of Euclidean vector spaces and their norms was considered in [6], [11].

Example 7. *Let us continuw with the space $\mathcal{V}_{(2,3)}$ from Example 4. We continue with with bases $\mathcal{B}_1 = \{x^2, xy, y^2\}$ and $\mathcal{B}_2 = \{x^3, x^2y, xy^2, y^3\}$ as in Example 6. Hence, a basis for $\mathcal{V}_{(2,3)}$ can be picked as $\mathcal{B} = \{x^2, xy, y^2, x^3, x^2y, xy^2, y^3\}$. Let us see how we could define an inner product in $\mathcal{V}_{(2,3)}$. Let $\mathbf{u}, \mathbf{v} \in \mathcal{V}_{(2,3)}$ given by*

$$\mathbf{u} = \mathbf{a} + \mathbf{b} = \left(u_1\, x^2 + u_2\, xy + u_3\, y^2\right) + \left(u_4\, x^3 + u_5\, x^2y + u_6\, xy^2 + u_7\, y^3\right)$$
$$\mathbf{v} = \mathbf{a}' + \mathbf{b}' = \left(v_1\, x^2 + v_2\, xy + v_3\, y^2\right) + \left(v_4\, x^3 + v_5\, x^2y + v_6\, xy^2 + v_7\, y^3\right)$$

Then

$$\begin{aligned}\langle \mathbf{u}, \mathbf{v} \rangle &= \langle \mathbf{a} + \mathbf{b}, \mathbf{a}' + \mathbf{b}' \rangle = \langle \mathbf{a} + \mathbf{a}' \rangle + \langle \mathbf{b} + \mathbf{b}' \rangle \\ &= u_1v_2 + u_2v_2 + u_3v_3 + u_4v_4 + u_5v_5 + u_6v_6 + u_7v_7\end{aligned}$$

and the Euclidean norm is defined as expected $\|\mathbf{u}\| = \sqrt{u_1^2 + \ldots u_7^2}$. □

There are other ways to define a norm on graded spaces. Consider a Lie algebra $\mathfrak{g}$. It is called **graded** if there is a finite family of subspaces $V_1, \ldots, V_r$ such that $\mathfrak{g} = V_1 \oplus \cdots \oplus V_r$ and $[V_i, V_j] \subset V_{i+j}$, where $[V_i, V_j]$ is the Lie bracket. When $\mathfrak{g}$ is graded we define for $t \in \mathbb{R}^\times$, $\alpha_t : \mathfrak{g} \to \mathfrak{g}$ such that

$$\alpha_t(v_1, \ldots, v_n) = (tv_1, t^2v_2, \ldots, t^r v_r).$$

We define a **homogenous norm** on $\mathfrak{g}$ as

$$\|\mathbf{v}\| = \|(v_1, \ldots, v_n)\| = \left(\|v_1\|_1^{2r} + \|v_2\|_2^{2r-2} + \cdots + \|v_r\|_r^2\right)^{1/2r}$$

where $\|\cdot\|_i$ is the Euclidean norm in V_i. For details see [5, 6]. It is shown in [11] that this norm satisfies the triangle inequality.

A more general approach is considered in [8] defining norms for line bundles and using such norms in the definition of weighted heights on weighted projective variaties.

5. Artificial neural networks over graded vector spaces

Let us now try to design artificial neural networks over graded vector spaces. Let k be a field and for any integer $n \geq 1$ denote by $\mathbb{A}_k^n$ (resp. $\mathbb{P}_k^n$) the affine (resp. projective) space over k. When k is an algebraically closed field, we will drop the subscript. A fixed tuple of positive integers $\mathbf{w} = (q_0, \ldots, q_n)$ is called **set of weights**. The weight of $\alpha \in k$ will be denoted by $\mathbf{wt}(\alpha)$. The set

$$\mathcal{V}_{\mathbf{w}}^n(k) := \{(x_1, \ldots, x_n) \in k^n \mid \mathbf{wt}(x_i) = q_i,\ i = 1, \ldots, n\}$$

is a graded vector space over k. From now on, when there is no confusion we will simply use $\mathcal{V}_{\mathbf{w}}$ for a graded vector space.

We follow the analogy with the classical case of artificial neural networks. A **neuron** on a graded vector space $\mathcal{V}_{\mathbf{w}}$ is a function $f : \mathcal{V}_{\mathbf{w}}^n \to k$ such that

$$\alpha_{\mathbf{w}}(\mathbf{x}) = \sum_{i-0}^{n} w_i^{q_i} x_i + \mathbf{b},$$

where $\mathbf{b} \in k$ is a constant called **bias**. We can generalize neurons to tuples of neurons via

$$\begin{aligned} \phi := \mathcal{V}_{\mathbf{w}}^n(k) &\to \mathcal{V}_{\mathbf{w}}^n(k) \\ \mathbf{x} &\to g\left(\alpha_0(\mathbf{x}), \ldots, \alpha_n(\mathbf{x})\right) \end{aligned}$$

for any gives set of weights $\mathbf{w}_0, \ldots, \mathbf{w}_n$. Then ϕ is a k-linear function with matrix written as

$$\phi(\mathbf{x}) = W \cdot \mathbf{x} + \mathbf{b},$$

for some $\mathbf{b} \in k^{n+1}$ and W an $n \times n$ matrix with integer entries.

Remark 1. *There is a big confusion here when it comes to terminology. The elements w_i are called weights in classical neural networks, but these are different from **weights** of the graded vector space q_i. The matrix W is called the matrix of weights since it is the matrix $W = [w_{i,j}]$, but again those weights are not the same as weights $q_0, \ldots, q_n$.*

A non-linear function $g : \mathcal{V}_{\mathbf{w}}^n \to \mathcal{V}_{\mathbf{w}}^n$ is called an **graded activation function**. A **graded network layer** is a function

$$\begin{aligned} \mathcal{V}_{\mathbf{w}}^n(k) &\to \mathcal{V}_{\mathbf{w}}^n(k) \\ \mathbf{x} &\to g\left(W \cdot \mathbf{x} + \mathbf{b}\right) \end{aligned}$$

for some some activation function g. A **graded neural network** is the composition of many layers. The l-th layer

$$\begin{aligned} \cdots \longrightarrow \mathcal{V}_{\mathbf{w}}^n(k) &\xrightarrow{\phi_l} \mathcal{V}_{\mathbf{w}}^n(k) \longrightarrow \cdots \\ \mathbf{x} &\longrightarrow \phi_l(\mathbf{x}) = g_l\left(W^l \mathbf{x} + \mathbf{b}^l\right), \end{aligned}$$

where g_l, W^l, and $\mathbf{b}^l$ are the activation, matrix, and bias corresponding to this layer. After m layers the output (predicted values) will be denoted by $\hat{\mathbf{y}} = [\hat{y_1}, \ldots, \hat{y_n}]^t$, where

$$\hat{\mathbf{y}} = \phi_m\left(\phi_{m-1}\left(\ldots\left(\phi_1(\mathbf{x})\right)\ldots\right)\right),$$

while the true values by $\mathbf{y} = [y_1, \ldots, y_n]^t \in \mathcal{V}^n_{\mathbf{w}}$.

The *relu activation* function for graded neural networks can be defined analougsly. Let $\mathbf{x} = [x_0, \ldots, x_n]^t \in V^n_{\mathbf{w}}$. Then for each coordinate $i = 0, \ldots, n$ we define

$$\mathrm{ReLu}_i(x_i) = \max\{0, |x_i|^{1/q_i}\}$$

Hence, is defined as

$$\mathrm{ReLu}(\mathbf{x}) := [\mathrm{ReLu}_0(x_0), \ldots, \mathrm{ReLu}_n(x_n)]$$

Example 8. *Consider $\mathcal{V}_{(2,3)}$ as above. Let $\mathbf{u} \in \mathcal{V}_{(2,3)}$. Then, $\mathbf{u} = f + g$ such that $f \in V_2$ and $g \in V_3$. Assume*

$$f = 2x^2 - 9xy + y^2, \qquad \text{and} \qquad g = x^3 - 2x^2y + xy^2 + y^3$$

The coordinated of $\mathbf{u}$ with respect to the basis $\mathcal{B}$ fixed in Example 4 are $\mathbf{u}_{\mathcal{B}} = [2, -3, 1, 1, -2, 11]^t$ and

$$\mathrm{ReLu}(\mathbf{u}) = [\sqrt{2}, 3, 1, 1, \sqrt{2}, 1, 1]^t.$$

It remains to be seen if this activation function or many others which can be adopted in our settings will be efficient. Notice the similarity of this definition with the weighted heights defined in [8, 9].

5.1. **Artificial neural networks on weighted projective spaces.** Our intention is to build a complete theory of equivariant neural networks over graded vector spaces and make it possible to design machine learning models to study weighted projective spaces and weighted varieties among other applications. It is unclear how such models would perform, however mathematically there is every reason to believe that computations on weighted projective varieties are more efficient than over classical projective varieties.

In [10] we used current techniques of machine learning to study the weighted projective space $\mathbb{WP}_{(2,4,6,10)}$ which is the moduli soace of genus two curves. The input features were invariants J_2, J_4, J_6, and J_{10}, which represent a point in a graded space. However, current techniques are designed for classical vector spaces. Hence, while we got some interesting results in [10] we weren't sure what these results represent. In other words, one can't really understand what is happening in a graded space unless the gradation becomes part of the training.

This paper is the first attempt in what we envision as a long project of machine learning in graded vector spaces. Most of mathematical details have yet to be worked out and one has to explore graded Lie algebras, graded manifolds and also different gradings. Especially interesting is the case when the ground field is not $\mathbb{R}$ or $\mathbb{C}$, but $\mathbb{Q}$ or a a field of positive characteristic. Such tasks present challenges mathematically and from the implementation point of view and their performance and efficiency are still open questions.

References

[1] I. N. Balaba, *Isomorphisms of graded rings of linear transformations of graded vector spaces*, Chebyshevskiu i Sb. **6** (2005), no. 4(16), 7–24. MR2455670 ↑17

[2] Vitalij M. Bondarenko, *Linear operators on S-graded vector spaces*, 2003, pp. 45–90. Special issue on linear algebra methods in representation theory. MR1987327 ↑17

[3] N. Bourbaki, *Algebra I*, Springer, 1974. Chapter 3. ↑16, 17

[4] J.-L. Koszul, *Graded manifolds and graded Lie algebras*, Proceedings of the international meeting on geometry and physics (Florence, 1982), 1983, pp. 71–84. MR760837 ↑16

[5] Martin Moskowitz, *An extension of Minkowski's theorem to simply connected 2-step nilpotent groups*, Port. Math. **67** (2010), no. 4, 541–546. MR2789262 ↑19

[6] ———, *The triangle inequality for graded real vector spaces of length 3 and 4*, Math. Inequal. Appl. **17** (2014), no. 3, 1027–1030. MR3224852 ↑18, 19

[7] Steven Roman, *Advanced linear algebra*, Third, Graduate Texts in Mathematics, vol. 135, Springer, New York, 2008. MR2344656 ↑4, 16

[8] S. Salami and T. Shaska, *Local and global heights on weighted projective varieties*, Houston J. Math. **49** (2023), no. 3, 603–636 (English). ↑3, 19, 20

[9] ———, *Vojta's conjecture on weighted projective varieties* (2024), available at arXiv:2309.10300. ↑3, 20

[10] E. Shaska and T. Shaska, *Machine learning for moduli space of genus two curves* (2024), available at arXiv:2403.17250. ↑2, 3, 20

[11] Songpon Sriwongsa and Keng Wiboonton, *The triangle inequality for graded real vector spaces*, Math. Inequal. Appl. **23** (2020), no. 1, 351–355. MR4061546 ↑18, 19

[12] Maurice Weiler, Patrick Forré, Erik Verlinde, and Max Welling, *Equivariant and coordinate independent convolutional networks*, University of Amsterdam, 2023. ↑3, 4, 15