# Discrete neural nets and polymorphic learning (Part 1)

Charlotte Aten

University of Denver

University of Denver Machine Learning Seminar
2023 October 2

# Introduction

- I have been an instructor/organizer of an annual data science and machine learning REU at the University of Rochester for several years now.
- In the the 2021 iteration of this program, I ran a project which led to the preprint I'm discussing today.
- This year, I began work on a sequel with a new group of students.

# Introduction

- You can find the preprint "Discrete neural nets and polymorphic learning" at https://arxiv.org/abs/2308.00677.
- Relevant code appears at https://github.com/caten2/Tripods2021UA.

# Talk outline

- Universal algebra and universal approximation
- Discrete neural nets
- Polymorphisms
- Example: binary images

# Universal algebra and universal approximation

- Some time during the 2020-2021 academic year, I noticed that some early results in universal approximation for neural nets (name Cybenko's from the 1980s) were extremely similar to the some results on primality of random finite algebras in universal algebra (namely Murskiĭ's from the 1970s).
- You can find the original video pitch where I propose exploiting this similarity at https://www.youtube.com/watch?v=nr0KbcloYW4.

# Universal algebra and universal approximation

- The idea was to use operations on a finite set instead of continuous activation functions.
- What my students and I quickly realized, however, is that this unrestricted hypothesis class led to overfitting much faster than in the continuous case.
- We found that we could do a lot better by only using activation functions which were polymorphisms of a relevant relational structure.

# Discrete neural nets

**Definition (Neural net)**

A *neural net* $(V_1, \ldots, V_r, E, \Phi)$ with $r$ layers consists of

1. a finite digraph $(V, E)$ (the *architecture* of the neural net) and
2. for each $v \in V \setminus V_1$ a function $\Phi(v)\colon \mathbb{R}^{\rho(v)} \to \mathbb{R}$ (the *activation function* of $v$)

where

1. $V := \bigcup_{i=1}^{r} V_i$,
2. the only edges in $E$ are from vertices in $V_i$ to vertices in $V_{i+1}$ for $i < r$,
3. $\rho(v)$ is the indegree of $v$ in $(V, E)$, and
4. if $i \neq r$ then every vertex $v \in V_i$ has nonzero outdegree.

# Discrete neural nets

- A typical neural net. A node $v_{ij} \in V_i$ is called a *neuron* in layer $i$. We will denote $\Phi(v_{ij})$ by $\phi_{ij}$.

- We think of the labels $x_j$ as variables.

# Discrete neural nets

**Definition (Neural net)**

A *discrete neural net* $(V_1, \ldots, V_r, E, \Phi)$ with $r$ layers on a finite set $A$ consists of

1. a finite digraph $(V, E)$ (the *architecture* of the neural net) and
2. for each $v \in V \setminus V_1$ a function $\Phi(v)\colon A^{\rho(v)} \to A$ (the *activation function* of $v$)

where

1. $V := \bigcup_{i=1}^{r} V_i$,
2. the only edges in $E$ are from vertices in $V_i$ to vertices in $V_{i+1}$ for $i < r$,
3. $\rho(v)$ is the indegree of $v$ in $(V, E)$, and
4. if $i \neq r$ then every vertex $v \in V_i$ has nonzero outdegree.

# Polymorphisms

**Definition (Structure, universe, basic operations/relations)**

A *structure* $\mathbf{A} := (A, F, \Theta)$ conisists of a set $A$ (the *universe* or *underlying set* of the structure) as well as indexed collections $F := \{f_i\}_{i \in I}$ and $\Theta := \{\theta_j\}_{j \in J}$ of operations on $A$ (the *basic operations* of $\mathbf{A}$) and of relations on $A$ (the *basic relations* of $\mathbf{A}$). We require that the index sets $I$ and $J$ be disjoint.

# Polymorphisms

**Definition (Homomorphism (of structures))**

Given structures $\mathbf{A} := (A, F, \Theta)$ and $\mathbf{B} := (B, G, \Psi)$ where $F := \{f_i\}_{i\in I}$, $G := \{g_i\}_{i\in I}$, $\Theta := \{\theta_j\}_{j\in J}$, and $\Psi := \{\psi_j\}_{j\in J}$, both of the same signature $\rho\colon I \cup J \to \mathbb{N}$, we say that a function $h\colon A \to B$ is a *homomorphism* from $\mathbf{A}$ to $\mathbf{B}$ when $h\colon (A, F) \to (B, G)$ is a homomorphism of algebras and for each $j \in J$ we have for all $a_1, \ldots, a_{\rho(j)} \in A$ that if

$$(a_1, \ldots, a_{\rho(j)}) \in \theta_j$$

then

$$(h(a_1), \ldots, h(a_{\rho(j)})) \in \psi_j.$$

# Polymorphisms

**Definition (Polymorphism)**

Given a structure $\mathbf{A}$ we say that a homomorphism $f\colon \mathbf{A}^n \to \mathbf{A}$ is a *polymorphism* of $\mathbf{A}$.

- For example, a group homomorphism $f\colon \mathbb{Z}^n \to \mathbb{Z}$ is a polymorphism of the group $\mathbb{Z}$.

# Example: binary images

**Definition (Hamming graph)**

Given $n \in \mathbb{N}$ we define the *n-Hamming graph* to be

$$\mathbf{Ham}_n := (A_n, \{\, (a_1, a_2) \in A_n^2 \mid d(a_1, a_2) \leq 1 \,\})$$

where $A_n$ is the set of all $n \times n$ images consisting of black and white pixels only and $d$ is the Hamming distance.

# Example: binary images

- Endomorphisms and automorphisms of $\mathbf{Ham}_n$ are easy to come by.
- The dihedral group acts on $\mathbf{Ham}_n$.
- Any bitwise operation with a fixed image will yield an endomorphism of $\mathbf{Ham}_n$.

# Example: binary images

- Higher-arity polymorphisms are harder to come by.
- These are graph homomorphisms

$$f\colon \mathbf{Ham}_n^k \to \mathbf{Ham}_n\,.$$

# Example: binary images

**Definition (Multi-linear indicator)**

Given $b \in B_n$ and $c \in A_n^k$ the *multi-linear indicator polymorphism* for $(b, c)$ is the map $g_{b,c} \colon A_n^k \to A_n$ given by

$$g_{b,c}(a_1, \dots, a_k) := \left( \prod_{i=1}^{k} a_i \cdot c_i \right) b$$

where $x \cdot y := \sum_{i,j} x_{ij} y_{ij}$ denotes the standard dot product in $\mathbb{F}_2^{[n]^2}$.

# Example: binary images

- The preprint contains a discussion of some even more involved/interesting polymorphisms.
- I am working with this year's group of students to extend these constructions to higher-arity relations.

# References


- **aten2022**
- **aten2023-1**