GANNs: A New Theory of Representation for Nonlinear Bounded Operators

William Guss
Machine Learning at Berkeley
April 22, 2016
Introduction: What’s up with continuous data?

- **All** of the data we deal with is discrete thanks to Turing.
- But, most of it models a continuous process.
- **Examples**
  - Audio: We take $> 100k$ samples of something we could describe with $f : \mathbb{R} \rightarrow \mathbb{R}$! Trick Question: Which is easier to use? (a) $\nu \in \mathbb{R}^{100000}$ or (b) $f$.
  - Images: We take $100k \times 100k$ samples of something we could describe with $f : \mathbb{R}^2 \rightarrow \mathbb{R}$.
- Why do we use discrete data? No computer known can really store $f$. End of story.
Introduction: Abusing continuity

- $f$ can’t be *that* bad. Can it?
- If $f$ is smooth it’s easy to draw:

![Graph of $f(x) = x^2$](image)

- I can even name $f$ most of the time: $f : x \mapsto x^2$ or even super precisely $g : x \mapsto \sum_{i}^{\infty} a_n x^n$.
- Moral: Smooth functions are mostly very manageable.
Introduction: Abusing continuity

- So why do we do this:

- To classify this:

  frequency: **20kHz**

  sampling rate: **44.1kHz**
Artificial Neural Networks

Definition

We say $\mathcal{N} : \mathbb{R}^n \rightarrow \mathbb{R}^m$ is a feed-forward neural network if for an input vector $\mathbf{x}$,

\[
\mathcal{N} : \sigma^{(l+1)}_j = g \left( \sum_{i \in Z^{(l)}} w^{(l)}_{ij} \sigma^{(l)}_i + \beta^{(l)} \right) 
\]

(1)

\[
\sigma^{(0)}_i = x_i,
\]

where $1 \leq l \leq L - 1$. Furthermore we say $\{\mathcal{N}\}$ is the set of all neural networks.
Operator Neural Networks

Let’s get rid of $\mathbb{R}^{100000}$ and use $f$.

**Definition**

We call $\mathcal{O} : L^p(X) \rightarrow L^1(Y)$ an operator neural network if,

$$
\mathcal{O} : \sigma^{(l+1)}(j) = g \left( \int_{R^{(l)}} \sigma^{(l)}(i)w^{(l)}(i,j) \, di \right)
$$

$$
\sigma^{(0)}(j) = f(j).
$$

Furthermore let $\{\mathcal{O}\}$ denote the set of all functional neural networks.

Well that was easy. In fact $\{\mathcal{O}\} \supset \{\mathcal{N}\}$

These definitions looks really similar? Is there some more general category or structure containing them.
Generalized Artificial Neural Networks

**Definition**

If $A, B$ are (possibly distinct) Banach spaces over a field $\mathbb{F}$, we say $\mathcal{G} : A \to B$ is a generalized neural network if and only if

$$
\mathcal{G} : \sigma^{(l+1)} = g \left( T_l \left[ \sigma^{(l)} \right] + \beta^{(l)} \right)
$$

$$
\sigma^{(0)} = \xi
$$

for some input $\xi \in A$, and a linear form $T_l$.

**Claim:** "Neural networks" are powerful because they can move bumps anywhere!

*How?* $T_l$ is a linear form. It can move $\sigma^{(l)}$ anywhere, and $g$ is a bump of some sort.
Moving bumps around

- The sigmoid function

\[ g = \frac{1}{1 + e^{-x}} \]  

is a bump, that we can move around with weights!
$T_l$ as the layer type.

**Definition**

We suggest several classes of $T_l$ as follows

- $T_l$ is said to be $\sigma$ operational if and only if

\[
T_l = \sigma : L^p(R^{(l)}) \to L^1(R^{(l+1)})
\]

\[
\sigma \mapsto \int_{R^{(l)}} \sigma(i) w^{(l)}(i, j) \, di.
\]

(5)

- $T_l$ is said to be $n$ discrete if and only if

\[
T_l = n : \mathbb{R}^n \to \mathbb{R}^m
\]

\[
\bar{\sigma} \mapsto \sum_{j} \bar{\sigma}_j \sum_{i} \sigma_i w^{(l)}_{ij}
\]

(6)

where $\bar{\sigma}_j$ denotes the $j^{th}$ basis vector in $\mathbb{R}^m$. 
Definition

- \( T_l \) is said to be \( n_1 \) transitional if and only if
  \[
  T_l = n_1 : \mathbb{R}^n \rightarrow L^q(\mathbb{R}^{(l+1)})
  \]
  \[
  \vec{\sigma} \mapsto \sum_{i}^n \sigma_i w_i^{(l)}(j).
  \]  
  \( \quad (7) \)

- \( T_l \) is said to be \( n_2 \) transitional if and only if
  \[
  T_l = n_2 : L^p(\mathbb{R}^{(l)}) \rightarrow \mathbb{R}^m
  \]
  \[
  \sigma(i) \mapsto \sum_{j}^m \vec{e}_j \int_{\mathbb{R}^{(l)}} \sigma(i) w_j^{(l)}(i) \, di
  \]  
  \( \quad (8) \)
Neural networks as diagrams!

This generalization is nice from a creative standpoint. I can come up with new sorts of "classifiers" on the fly. **Examples:**

- A three layer neural network is just
  \[ \mathcal{N}_3 : \mathbb{R}^{10000} \xrightarrow{g^\circ n} \mathbb{R}^{30} \xrightarrow{g^\circ n} \mathbb{R}^3. \] (9)

- A three layer operator network is simply
  \[ \mathcal{O}_3 : L^p(R) \xrightarrow{g^\circ o} L^1(R) \xrightarrow{g^\circ o} C(R). \] (10)

- We can even classify functions!
  \[ \mathcal{C} : L^p(R) \xrightarrow{g^\circ o} L^1(R) \xrightarrow{g^\circ o} \ldots \xrightarrow{g^\circ o} L^1(R) \xrightarrow{g^\circ n^2} \mathbb{R}^n. \] (11)
Results: Did abusing continuity help?

For every layer $a$ has weights

$$w^{(l)}(i, j) = \sum_{b} \sum_{a} k_{a,b}^l i^a j^b.$$  \hspace{1cm} (12)

**Theorem**

Let $C$ be a GANN with only one $n_2$ transitional layer with $O(1)$ weight polynomial. If a continuous function, say $f(t)$ is sampled uniformly from $t = 0$, to $t = N$, such that $x_n = f(n)$, and if $G$ has an input function which is piecewise linear with $O(N^2)$ weights.

$$\xi = (x_{n+1} - x_n) (z - n) + x_n$$ \hspace{1cm} (13)

for $n \leq z < n + 1$, then there exist some discrete neural network $\mathcal{N}$ such that $G(\xi) = \mathcal{N}(x)$. 
Results: Did abusing continuity help?

**WHAT?!?!?** How did $C$ reduce the number of weights from $O(N^2)$ to $O(1)$?

- The infinite dimensional versions of $N$, in particular $O$ and $C$ are invariant to input quality. Takes the idea behind Convnets to an extreme!
- This is easy to see.
Results: Representation Theory

How good are Continuous Classifier Networks, \( \{C\} \) as algorithms?

**Theorem**

Let \( X \) be a compact Hausdorff space. For every \( \epsilon > 0 \) and every continuous bounded functional on \( L^q(X) \), say \( f \), there exists a two layer continuous classifier

\[
C : L^q(x) \xrightarrow{g \circ n_2} \mathbb{R}^m \xrightarrow{n} \mathbb{R}^n
\]

such that

\[
\| f - C \| < \epsilon.
\]
How good are Operator Networks and GANNs as algorithms? They should be able to approximate the important operators, eg. **Fourier Transform, Laplace Transform, Derivation**, etc.

**Theorem**

*Given a operator neural network $O$ then some layer $l \in O$, then let $K : C(R^{(l)}) \rightarrow C(R^{(l)})$ be a bounded linear operator. If we denote the operation of layer $l$ on layer $l - 1$ as $\sigma^{(l+1)} = g \left( \Sigma_{l+1} \sigma^{(l)} \right)$, then for every $\epsilon > 0$, there exists a weight polynomial $w^{(l)}(i,j)$ such that the supremum norm over $R^{(l)}$*

$$\left\| K \sigma^{(l)} - \Sigma_{l+1} \sigma^{(l)} \right\|_{\infty} < \epsilon \quad (16)$$

**Proof.**

See paper. Nice!
We want to show the following better theorem.

**Theorem**

Given a operator neural network $\mathcal{O}$ then some layer $l \in \mathcal{O}$, let $K : C(R^{(l)}) \rightarrow C(R^{(l)})$ be a bounded **continuous** operator. If we denote the operation of layer $l$ on layer $l-1$ as $\sigma^{(l+1)} = g\left(\Sigma_{l+1} \sigma^{(l)}\right)$, then for every $\epsilon > 0$, there exists a weight polynomial $w^{(l)}(i, j)$ such that the supremum norm over $R^{(l)}$

$$\left\| K \sigma^{(l)} - \Sigma_{l+1} \sigma^{(l)} \right\|_\infty < \epsilon$$  \hspace{1cm} (17)

But how? **Dirac Spikes!**
Results: Stronger Representation Theory

Proof.
Proof.

- Fix $\epsilon > 0$. Given $K: \xi \mapsto f$, let $K_j: \xi \mapsto f(j)$ be a functional on $L^q$. 
Results: Stronger Representation Theory

Proof.

- Fix $\epsilon > 0$. Given $K : \xi \mapsto f$, let $K_j : \xi \mapsto f(j)$ be a functional on $L^q$.
- We can find a $C_j : L^q(R) \xrightarrow{g_{\text{on}2}} \mathbb{R}^m(j) \xrightarrow{n} \mathbb{R}^1$ so that for all $\xi$,
  \[ |C_j(\xi) - K_j(\xi)| = |C_j(\xi) - f(j)| < \epsilon/2. \]  (18)
Results: Stronger Representation Theory

Proof.

We know that

\[ C_j(\xi) = \sum_{k=1}^{m(j)} a_{jk} g \left( \int_R \xi(i) w_{kj}(i) \, d\mu(i) \right) \]  \hspace{1cm} (18)
Results: Stronger Representation Theory

Proof.

- We wish to turn $C_j$ into a two layer $O$. Let,

$$w^{(0)}(i, \ell) = \begin{cases} w_{kj}(i), & \text{if } \ell = j + k, \ k \in 1, \ldots, m(j) \\ 0 & \text{otherwise} \end{cases}$$
Results: Stronger Representation Theory

Proof.

Then

\[ C_j(\xi) = \sum_{k=1}^{m} a_{jk} g \circ o[\xi](k + j) \]  

(18)
Results: Stronger Representation Theory

Proof.

Then

\[ C_j(\xi) = \sum_{k=1}^{m} a_{jk} g \circ o[\xi](k + j) \]  \hspace{1cm} (18)

How do we turn this finite sum into an integral? Dirac time!
Proof.

We define a dirac spike as follows for every \( n \):

\[
\delta_{nkj}(\ell) = cn \exp(-bn^2|\ell - (j + k)|^2)
\]  

(18)

where \( c, b \) are set so that \( \int_\mathbb{R} \delta_{nk} = 1 \)
Proof.

Now let the second weight function be:

\[ w_n^{(1)}(\ell, j) = \sum_{k=1}^{m} a_{jk} \delta_{nkj}(\ell) \]  

(18)
Proof.

- Putting everything together, for every $n$ let
  \[ O_n : L^p(R) \to L^1([0, 1]) \]

  \[ O_n : \xi \mapsto \int_R w^{(1)}(\ell, j) \circ [\xi](\ell) \, d\mu(\ell). \]  

  Clearly $O_n \to \sum_{k=1}^m a_{jk} g \circ \circ [\xi](k + j)$
Proof.

Therefore for every $\epsilon > 0$ there exists an $N$ such that for all $n > N$, for all $\xi$, and for all $j$,

$$|O_n[\xi](j) - C_j[\xi]| \leq \|O_n[\cdot](j) - C_j[\cdot]\| < \epsilon/2. \quad (18)$$
Proof.

- Therefore for every $\epsilon > 0$ there exists an $N$ such that for all $n > N$, for all $\xi$, and for all $j$,

$$|O_n[\xi](j) - C_j[\xi]| \leq \|O_n[\cdot](j) - C_j[\cdot]\| < \epsilon/2. \quad (18)$$

- Recall that for every $j$, $\|K_j - C_j\| < \epsilon/2$. 
By the triangle inequality we have that for all $j$

$$\|K_j - O_n(k)\| = \|K_j - O_n(j) + C_j - C_j\|$$

$$\leq \|K_j - C_j\| + \|O_n(j) - C_k\| < \epsilon.$$
Results: Stronger Representation Theory

Proof.

- By the triangle inequality we have that for all $j$

  \[
  \|K_j - O_n(k)\| = \|K_j - O_n(j) + C_j - C_j\| \\
  \leq \|K_j - C_j\| + \|O_n(j) - C_k\| < \epsilon. \tag{18}
  \]

- Therefore $\|K - O\| < \epsilon$