An Isabelle Formalization of the Expressiveness of Deep Learning

Alexander Bentkamp — Vrije Universiteit Amsterdam
Jasmin Blanchette — Vrije Universiteit Amsterdam
Dietrich Klakow — Universität des Saarlandes

Motivation

- Case study of proof assistance in the field of machine learning
- Development of general-purpose libraries
- Study of the mathematics behind deep learning

Motivation

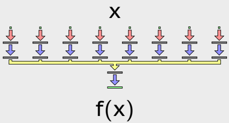
- Case st... of proof ...sistance in the field of machine le...ing
- Development o... ...eral-purpose libraries
- Study of the ...athemati... behind deep learning

Just wanted to formalize something!

Fundamental Theorem of Network Capacity

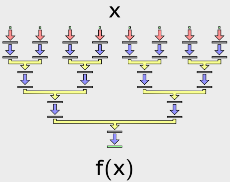
(Cohen, Sharir & Shashua, 2015)

Shallow network



needs exponentially more nodes to express the same function as

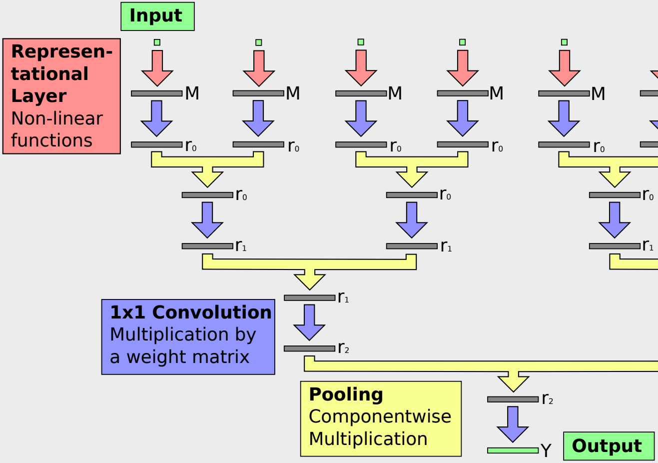
deep network



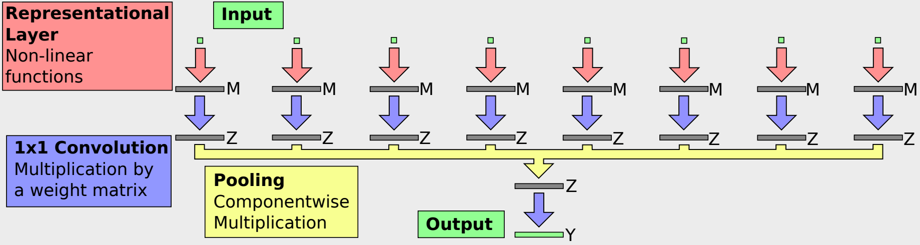
for the vast majority of functions*

* except for a Lebesgue null set

Deep convolutional arithmetic circuit



Shallow convolutional arithmetic circuit



Lebesgue measure

definition lborel :: (α :: euclidean_space) measure

Isabelle's standard probability library

vs.

My new definition

definition $\mathsf{lborel_f}$ :: nat $\Rightarrow$ (nat $\Rightarrow$ real) measure
where $\mathsf{lborel_f}$ n =
$\prod_{\mathsf{M}}$ b $\in$ {.. < n}. (lborel :: real measure)

Matrices

- Isabelle's multivariate analysis library
- Sternagel & Thiemann's matrix library
  (Archive of Formal Proofs, 2010)
- Thiemann & Yamada's matrix library
  (Archive of Formal Proofs, 2015)

Matrices

matrix dimension fixed by the type

- Isabelle's multivariate analysis library
- Sternagel & Thiemann's matrix library
  (Archive of Formal Proofs, 2010)
- Thiemann & Yamada's matrix library
  (Archive of Formal Proofs, 2015)

Matrices

matrix dimension fixed by the type

- Isabelle's multivariate analysis library
- Sternagel & Thiemann's matrix library
  (Archive of Formal Proofs, 2010)

lacking many necessary lemmas

- Thiemann & Yamada's matrix library
  (Archive of Formal Proofs, 2015)

Matrices

- Isabelle's multivariate analysis library ← matrix dimension fixed by the type
- Sternagel & Thiemann's matrix library ← lacking many necessary lemmas
  (Archive of Formal Proofs, 2010)
- Thiemann & Yamada's matrix library
  (Archive of Formal Proofs, 2015)

I added definitions and lemmas for

- matrix rank
- submatrices

Multivariate polynomials

Lochbihler & Haftmann's polynomial library

I added various definitions, lemmas, and the theorem

- "Zero sets of polynomials $\not\equiv 0$ are Lebesgue null sets."

theorem:
fixes p :: real mpoly
assumes $p \neq 0$ **and** vars $p \subseteq \{..< n\}$
shows $\{x \in$ space (lborel$_f$ n). insertion x $p = 0\}$
$\in$ null_sets (lborel$_f$ n)

My tensor library

typedef α tensor $=$
$\{(\text{ds} :: \text{nat list}, \text{as} :: \alpha \text{ list}).\ \text{length as} = \text{prod\_list ds}\}$

- addition, multiplication by scalars, tensor product, matricization, CP-rank
- Powerful induction principle uses subtensors:
  - Slices a $d_1 \times d_2 \times \cdots \times d_N$ tensor into d_1 subtensors of dimension $d_2 \times \cdots \times d_N$

definition subtensor :: α tensor $\Rightarrow$ nat $\Rightarrow$ α tensor

The proof on one slide

- Def1 Define a tensor $\mathcal{A}(w)$ that describes the function expressed by the deep network with weights w
- Lem1 The CP-rank of $\mathcal{A}(w)$ indicates how many nodes the shallow network needs to express the same function
- Def2 Define a polynomial p with the deep network weights w as variables
- Lem2 If $p(w) \neq 0$, then $\mathcal{A}(w)$ has a high CP-rank
- Lem3 $p(w) \neq 0$ almost everywhere

Restructuring the proof

Before

- Def1 Tensors
- Lem1 Tensors, shallow network
- Induction over the deep network
  - Lem2 Polynomials, Matrices
  - Def2 Polynomials, Tensors
  - Lem3a Matrices, Tensors
  - Lem3b Measures, Polynomials

After

- Def1 Tensors
- Lem1 Tensors, shallow network
- Induction over the deep network
  - Def2 Polynomials, Tensors
- Lem2 Polynomials, Matrices
- Induction over the deep network
  - Lem3a Matrices, Tensors
- Lem3b Measures, Polynomials

Restructuring the proof

Before*

- Def1 Tensors
- Lem1 Tensors, shallow network
- Induction over the deep network
  - Lem2 Polynomials, Matrices
  - Def2 Polynomials, Tensors
  - Lem3a Matrices, Tensors
  - Lem3b Measures, Polynomials

* except for a Lebesgue null set

After*

- Def1 Tensors
- Lem1 Tensors, shallow network
- Induction over the deep network
  - Def2 Polynomials, Tensors
- Lem2 Polynomials, Matrices
- Induction over the deep network
  - Lem3a Matrices, Tensors
- Lem3b Measures, Polynomials

* except for a zero set of a polynomial

Type for convolutional arithmetic circuits

datatype α cac $=$

Input nat | Conv α (α cac) | Pool (α cac) (α cac)

Type for convolutional arithmetic circuits

datatype α cac =
  Input nat | Conv α (α cac) | Pool (α cac) (α cac)

fun insert_weights ::
  $\underbrace{\text{(nat} \times \text{nat) cac}}_{\text{network without weights}} \Rightarrow \underbrace{\text{(nat} \Rightarrow \text{real)}}_{\text{weights}} \Rightarrow \underbrace{\text{real mat cac}}_{\text{network with weights}}$

Type for convolutional arithmetic circuits

datatype α cac =
  Input nat | Conv α (α cac) | Pool (α cac) (α cac)

fun insert_weights ::
  $\underbrace{\text{(nat} \times \text{nat) cac}}_{\text{network without weights}} \Rightarrow \underbrace{\text{(nat} \Rightarrow \text{real)}}_{\text{weights}} \Rightarrow \underbrace{\text{real mat cac}}_{\text{network with weights}}$

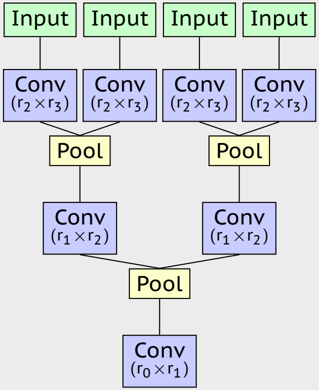
fun evaluate_net :: $\underbrace{\text{real mat cac}}_{\text{network}} \Rightarrow \underbrace{\text{real vec list}}_{\text{input}} \Rightarrow \underbrace{\text{real vec}}_{\text{output}}$

Deep network parameters

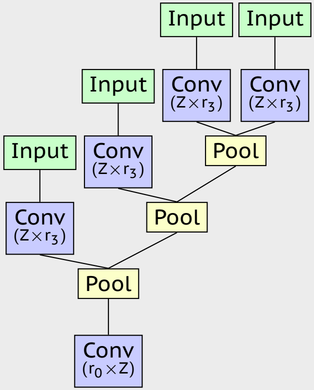
```
locale deep_net_params =
  fixes rs :: nat list
  assumes deep: length rs ≥ 3
  and no_zeros: ⋀r. r ∈ set rs ⟹ 0 < r
```

Deep and shallow networks

deep_net =



shallow_net Z =



Def1 Define a tensor $\mathcal{A}(w)$ that describes the function expressed by the deep network with weights w

definition $\mathcal{A}$ :: (nat $\Rightarrow$ real) $\Rightarrow$ real tensor
  where $\mathcal{A}$ w = tensor_from_net (insert_weights deep_net w)

The function tensor_from_net represents networks by tensors:

fun tensor_from_net :: real mat cac $\Rightarrow$ real tensor

If two networks express the same function, the representing tensors are the same

Lem1 The CP-rank of $\mathcal{A}(\mathrm{w})$ indicates how many nodes the shallow network needs to express the same function

```
lemma cprank_shallow_model:
  shows cprank (tensor_from_net
                  (insert_weights w (shallow_net Z))) ≤ Z
```

- Can be proved by definition of the CP-rank

Def2 Define a polynomial p with the deep network weights w as variables

Easy to define as a function:

definition p_{func} :: (nat $\Rightarrow$ real) $\Rightarrow$ real **where**
  p_{func} w = det (submatrix [$\mathcal{A}_i$ w] rows_with_1 rows_with_1)

But we must prove that p_{func} is a polynomial function

Lem2 If $p(w) \neq 0$, then $\mathcal{A}(w)$ has a high CP-rank

lemma
assumes p_{func} w $\neq 0$
shows $r^{N\_half} \leq$ cprank ($\mathcal{A}$ w)

- Follows directly from definition of p using properties of matricization and of matrix rank

Lem3 $p(w) \neq 0$ almost everywhere

Zero sets of polynomials $\not\equiv 0$ are Lebesgue null sets
$\Longrightarrow$ It suffices to show that $p \not\equiv 0$

We need a weight configuration w with $p(w) \neq 0$

Final theorem

theorem

$\forall_{ae}$ w_d w.r.t. lborel_f weight_space_dim. $\nexists w_s$ Z.
  Z $<$ $r^{\text{N\_half}}$ $\wedge$
  $\forall$is. input_correct is $\longrightarrow$
    evaluate_net (insert_weights deep_net w_d) is $=$
    evaluate_net (insert_weights (shallow_net Z) w_s) is

Conclusion

Outcome

- First formalization on deep learning
  Substantial development ($\sim$ 7000 lines including developed libraries)
- Development of libraries
  New tensor library and extension of other libraries
- Generalization of the theorem
  Proof restructuring led to a more precise result

More information:

http://matryoshka.gforge.inria.fr/#Publications

AITP abstract

M.Sc. thesis

Archive of Formal Proofs entry

ITP paper draft (coming soon)