CS620: FM in ML
Specifying Properties of Neural Networks
(Week 3)

Supratik Chakraborty
A Typical Neural Network

- **Input layer**
- **Hidden layers**
- **Output layer**

$n$ inputs → $m$ outputs
A Typical Neural Network

Input layer

Input Domain: $\mathcal{I}$
Input Space: $\mathcal{I}^n$

Output layer

Output Domain: $\mathcal{O}$
Output space: $\mathcal{O}^m$
A Typical Neural Network

\[ \nu : \mathcal{I}^n \rightarrow \mathcal{O}^m \]

Input Domain: \( \mathcal{I} \)
Input Space: \( \mathcal{I}^n \)

Output Domain: \( \mathcal{O} \)
Output Space: \( \mathcal{O}^m \)
A Transformative Program

First order logic formula

\{ \varphi(x) \} \quad \cdots \quad \text{Pre-condition}

Input: \( x \in I^n \)

\( y \leftarrow \nu(x) \);

Output: \( y \in O^m \)

\{ \psi(y) \} \quad \cdots \quad \text{Post-condition}

Hoare triples similar to those used in program verification
Semantics of Hoare Triple

\{ \varphi(x) \} \cdots \text{Pre-condition}

y \leftarrow \nu(x); \cdots \quad \text{”Program”}

\{ \psi(y) \} \cdots \text{Post-condition}

Validity of Hoare triple

If \( x \) satisfies \( \varphi(x) \),

”program” terminates and encounters no memory exception,
then output \( y \) always satisfies \( \psi(y) \).
Property Specification Example 1

Wish to specify that the above never happens for a given image, for a specified max perturbation.

Source: Goodfellow, Shlens, Szegedy, “Explaining and Harnessing Adversarial Examples”, 2015
Property Specification Example 1

Specified image: $x^*$

\[ \{ \| x - x^* \| \leq \varepsilon \} \]

Max perturbation of input

\[ (p, g) \leftarrow \nu(x) \]

Separation threshold for “confident” classification

\[ \{ p > g + \delta \} \]

Score for panda: $p$

Score for something else: $g$
Property Specification Example 1

Specified image: $x^*$

Score for panda: $p$
Score for something else: $g$

\[
\{(x - x^*) \leq \varepsilon\}
\]

\[
\bigwedge_{i=1}^{N} (|r_i - r_i^*| \leq \varepsilon_r) \land
\bigwedge_{i=1}^{N} (|g_i - g_i^*| \leq \varepsilon_g) \land
\bigwedge_{i=1}^{N} (|b_i - b_i^*| \leq \varepsilon_b)
\]

\[
(p, g) \leftarrow \nu(x);
\]

\[
\{p > g + \delta\}
\]
Spec as a logical requirement

\[ \forall r_1 \forall g_1 \forall b_1 \cdots \forall r_N \forall g_N \forall b_N \forall p \forall g \]

\[
\bigg( \bigwedge_{i=1}^{N} (|r_i - r_i^*| \leq \varepsilon_r) \bigwedge \\
\bigwedge_{i=1}^{N} (|g_i - g_i^*| \leq \varepsilon_g) \bigwedge \\
\bigwedge_{i=1}^{N} (|b_i - b_i^*| \leq \varepsilon_b) \bigwedge \\
(p, g) = \nu(r_1, g_1, b_1, \ldots r_N, g_N, b_N) \bigg)
\]

\[
\Rightarrow \\
p > g + \delta
\]

A logical implication

\[
\{ ||x - x^*|| \leq \varepsilon \}
\]

\[
\bigwedge_{i=1}^{N} (|r_i - r_i^*| \leq \varepsilon_r) \bigwedge \\
\bigwedge_{i=1}^{N} (|g_i - g_i^*| \leq \varepsilon_g) \bigwedge \\
\bigwedge_{i=1}^{N} (|b_i - b_i^*| \leq \varepsilon_b)
\]

\[
(p, g) \leftarrow \nu(x);
\]

\[
\{ p > g + \delta \}
Property Specification Example 2

Given two arbitrary images that differ within prescribed limits, the network must never “confidently” classify them differently.

\[
\{ \| x - x^* \| \leq \varepsilon \}
\]

\[
(s_1, s_2) \leftarrow \nu(x);
\]

\[
(s_1^*, s_2^*) \leftarrow \nu(x^*);
\]

\[
\begin{align*}
(s_1 > s_2 + \delta) & \implies (s_1^* > s_2^* + \delta) \\
(s_2 > s_1 + \delta) & \implies (s_2^* > s_1^* + \delta)
\end{align*}
\]
Property Specification Example 2

Given two images that differ within prescribed limits, the network must never “confidently” classify them differently

\[
\{ \| x - x^* \| \leq \varepsilon \}
\]

\[
(s_1, s_2) \leftarrow \nu(x);
(s_1^*, s_2^*) \leftarrow \nu(x^*);
\]

\[
\begin{align*}
(s_1 > s_2 + \delta) & \implies (s_1^* > s_2^* + \delta) \land \\
(s_2 > s_1 + \delta) & \implies (s_2^* > s_1^* + \delta)
\end{align*}
\]

Pause n Reflect

Are there any unintended consequences of the specification?

Can a neural network satisfying the specification do anything meaningful?

How easy/hard is it to design a neural network satisfying this specification?
Spec as a logical requirement

\[ \forall r_1 \cdots \forall b_N \forall r^*_1 \cdots \forall b^*_N \forall s_1 \forall s_2 \forall s^*_1 \forall s^*_2 \]

\[
\left( \bigwedge_{i=1}^{N} (|r_i - r^*_i| \leq \varepsilon_r) \right) \wedge \\
\left( \bigwedge_{i=1}^{N} (|g_i - g^*_i| \leq \varepsilon_g) \right) \wedge \\
\left( \bigwedge_{i=1}^{N} (|b_i - b^*_i| \leq \varepsilon_b) \right) \wedge \\
(s_1, s_2) = \nu(r_1, g_1, b_1, \ldots, r_N, g_N, b_N) \wedge \\
(s^*_1, s^*_2) = \nu(r^*_1, g^*_1, b^*_1, \ldots, r^*_N, g^*_N, b^*_N) \wedge \\
\implies \\
(s_1 > s_2 + \delta) \iff (s^*_1 > s^*_2 + \delta) \wedge \\
(s_2 > s_1 + \delta) \iff (s^*_2 > s^*_1 + \delta)
\]

\[
\{ \| x - x^* \| \leq \varepsilon \} \\
\left( \bigwedge_{i=1}^{N} (|r_i - r^*_i| \leq \varepsilon_r) \right) \wedge \\
\left( \bigwedge_{i=1}^{N} (|g_i - g^*_i| \leq \varepsilon_g) \right) \wedge \\
\left( \bigwedge_{i=1}^{N} (|b_i - b^*_i| \leq \varepsilon_b) \right) \wedge \\
(s_1, s_2) \leftarrow \nu(x); \\
(s^*_1, s^*_2) \leftarrow \nu(x^*); \\
\}
\]
Problem with Specification 2

Pick any two arbitrary images in the input space

$\varepsilon > 0$ implies all images must be classified same !!!

Is that what we intended? Important to get specs right!!!
Taking a step back to re-look

\[ \{ \| x - x^* \| \leq \varepsilon \} \]

\[ (p, g) \leftarrow \nu(x) \]

\[ \{ p > g + \delta \} \]

\[ \{ \| x - x^* \| \leq \varepsilon \} \]

\[ (s_1, s_2) \leftarrow \nu(x); \]
\[ (s_1^*, s_2^*) \leftarrow \nu(x^*); \]

\[ \left\{ \begin{align*}
(s_1 > s_2 + \delta) & \implies (s_1^* > s_2^* + \delta) \land \\
(s_2 > s_1 + \delta) & \implies (s_2^* > s_1^* + \delta)
\end{align*} \right\} \]
Attempting a Fix

Given two arbitrary images that differ within prescribed limits, the network must never “confidently” classify them differently.

\[
\{ \|x - x^*\| \leq \varepsilon \}
\]

\[
(s_1, s_2) \leftarrow \nu(x);
\]

\[
(s_1^*, s_2^*) \leftarrow \nu(x^*);
\]

\[
(s_1 > s_2 + \delta) \implies (s_2^* \leq s_1^* + \delta) \land
\]

\[
(s_2 > s_1 + \delta) \implies (s_1^* \leq s_2^* + \delta)
\]
Did It Fix?

Pick any two arbitrary images in the input space

> 0 no longer implies all images must be classified same !!!
But, images with minor changes can be classified vastly differently

Is that what we intended?
Important to get specs right!!!
Property Specification Example 2
Second attempt!

Given two arbitrary images that differ pixel-wise within prescribed limits and have “similar” semantic features, the network must never “confidently” classify them differently.

\[
\begin{align*}
\{ (\| x - x^* \| \leq \varepsilon) \land (\sigma(x) \approx \sigma(x^*)) \} \\
(s_1, s_2) \leftarrow \nu(x); \\
(s_1^*, s_2^*) \leftarrow \nu(x^*); \\
\{ (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \land (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \}
\end{align*}
\]
Second attempt!

Given two arbitrary images that differ pixel-wise within prescribed limits and have “similar” semantic features, the network must never “confidently” classify them differently.

\[
\{(\|x - x^*\| \leq \varepsilon) \land (\sigma(x) \approx \sigma(x^*))\}
\]

\[
(s_1, s_2) \leftarrow \nu(x);
\]

\[
(s_1^*, s_2^*) \leftarrow \nu(x^*);
\]

\[
\begin{aligned}
&\left\{ (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \land \\
&(s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \right\}
\end{aligned}
\]

User-defined semantic features, Not necessarily network-defined.
Possibilities with New Spec

Pick any two arbitrary images in the input space
Possibilities with Newer Spec

Pick any two arbitrary images in the input space.
Property Specification Example 2

Third attempt!

Given two arbitrary images that differ pixel-wise within prescribed limits and have “similar” semantic features, the network must produce “similar” classifications.

Network-defined labeling function: “final” layer(s)
Possibilities with New Spec

Pick any two arbitrary images in the input space
Property Specification

Pause n Reflect

Why is it so hard to get specifications right?

Is it easier to arrive at

THE RIGHT SPECIFICATION that covers all aspects of behaviour

OR

A bunch of sub-specifications that cover parts of the behaviour space?
A Day In The Life of A “Specifier”

Collect a bunch of *desired/undesired* (input, output) pairs
- Not necessarily what DNN is actually doing
- Instead, what DNN’s environment “expects” it to do

Is there a *formalizable relation* between inputs and desired outputs?
- Did we miss out corner cases?
- Sufficiently constrained to preclude all undesired behaviour?
- Sufficiently relaxed to allow all desired behaviour?
Input-Output Relation: How hard is it to formalize?

Self-driving car

Image (road scene)

“Too congested to accelerate”

Unmanned drone

Flight parameters

Score (Horizontal Advisory)
Non-Perceptual DNN Specs

ACAS-Xu

\[(\rho \geq 55947.691 \text{ ft}) \land (v_{\text{own}} \geq 1145 \text{ ft/s}) \land (v_{\text{int}} \leq 60 \text{ ft/s})\]  

\[\text{Score} \leftarrow \nu(\rho, v_{\text{own}}, v_{\text{int}}, \theta, \psi)\]  

\{\text{Score[COC] \leq 1500}\}

Non-Perceptual DNN Specs

ACAS-Xu

Rules for ACAS-Xu when directly implemented takes > 2GB memory

45 Non-perceptual DNNs for same take < 3MB of memory

Having a good spec for a non-perceptual DNN doesn’t make the DNN irrelevant !!!

Specs NOT SAME AS Rules

Perceptual DNN Specs

Image (road scene)

Good spec:

$$\text{CR} (r_1, g_1, b_1, \ldots r_N, g_N, b_N)$$

\{ (r_1, g_1, b_1, \ldots r_N, g_N, b_N): \text{image of congested road} \}

$$y \leftarrow \nu (r_1, g_1, b_1, \ldots r_N, g_N, b_N);$$

\{ y = \text{“Too congested to accelerate”} \}

CR( ... ) = \text{true iff road is “too congested to accelerate”}

"Too congested to accelerate"

If we know CR (...), why design and train a DNN ???
Perceptual DNN Specs

\((r_1, g_1, b_1, \ldots r_N, g_N, b_N)\)

Image (road scene)

Good spec:

\[
\text{CR}(r_1, g_1, b_1, \ldots r_N, g_N, b_N)
\]

\[
\{ (r_1, g_1, b_1, \ldots r_N, g_N, b_N) : \text{image of congested road} \}
\]

CR

Having the ideal spec for a perceptual DNN would make the DNN irrelevant !!!

“Too congested to accelerate”

If we know CR (...), why design and train a DNN ???
Specifying Properties of Perceptual DNNs

Are we in a chicken-and-egg conundrum for perceptual DNNs?

Is there any meaningful way out?
  We can talk about robustness of classification w.r.t. a specific image
  Can we specify anything formally beyond this?
Points to Ponder

Are we in a chicken-and-egg conundrum for perceptual DNNs?

Is there any meaningful way out?
   We can talk about robustness of classification w.r.t. a specific image
   Can we specify anything formally beyond this?

Is it better to write a single all-encompassing spec or multiple sub-specs for different behavioural requirements?
Any Hope for Perceptual DNNs?

High dimensional, large input space

Too congested to accelerate

Most images inconsequential, have no semantic similarity to what can possibly arise on a road

Can we restrict specs to a lower dimensional, smaller, meaningful input space?
Any Hope for Perceptual DNNs?

Renderer / Image generator (probabilistic?)

Low dimensional Semantic inputs

Dimensions of semantic inp space = 6
|Semantic inp space| = 5x4x4x2x5x2 = 1600

Dimensions of image inp space = 100x100x3 = 30000
|Image inp space| = 256^{100x100x3}

Time of Day: {Morning, Noon, Afternoon, Dusk, Night}
Weather: {Clear, Cloudy, Snowing, Raining}
Lanes: {Wide, Medium, Narrow, None}
Road direction: {Straight, Bending}
Other vehicles within 10m: {0, 1-3, 4-8, 9-15, > 15}
Behaviour of other vehicles: {Lane disciplined, Chaotic}

{ Pre-condition on semantic inputs s }

\[ i \leftarrow \rho(s); \quad // \rho: \text{Model of renderer} \]
\[ y \leftarrow \nu(i); \quad // \nu: \text{Model of perceptual DNN} \]

{ Post-condition on y}
Any Hope for Perceptual DNNs?

Low dimensional Semantic inputs

Renderer / Image generator (probabilistic?)

T: {Morning, Noon, Afternoon, Dusk, Night}
W: {Clear, Cloudy, Snowing, Raining}
L: {Wide, Medium, Narrow, None}
Rd: {Straight, Bending}
O: {0, 1-3, 4-8, 9-15, > 15}
B: {Lane disciplined, Chaotic}

\[
\{(O > 15) \lor \\
(L = W) \land ((O \geq 9) \land (B = Ch)) \lor \\
(L = M) \land ((O \geq 9) \lor ((O \geq 4) \land (B = Ch))) \lor \\
((L = N) \lor (L = None)) \land ((O \geq 4) \lor ((O \geq 1) \land (B = Ch)))\}\]

\[i \leftarrow \rho(T, W, L, Rd, O, B); \quad //\rho: \text{Model of renderer}\]
\[y \leftarrow \nu(i); \quad //\nu: \text{Model of perceptual DNN}\]

\{ y = “Too congested to accelerate” \}
Any Hope for Perceptual DNNs?

Low dimensional Semantic inputs

Potential "problems":
- Doesn’t cover entire input space
- Enrich semantic space to cover most/all meaningful inputs
- Use richer rendering modules
- Need to model renderer
- Use abstract / non-deterministic / probabilistic models

Significant "benefits":
- Can eliminate large parts of irrelevant/meaningless input space
- Provide guarantees over large parts of meaningful input space
One Spec vs Multiple Sub-specs

ACAS-Xu

\[(\rho \geq 55947.691ft) \land (v_{own} \geq 1145ft/s) \land (v_{int} \leq 60ft/s)\]

\[\text{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)\]

\[\{\text{Score[COC]} \leq 1500\}\]

One Spec vs Multiple Sub-specs

**ACAS-Xu**

\[\{(0 \leq \rho \leq 60760\text{ft}) \land (1000 \leq v_{own} \leq 1200\text{ft/s}) \land (0 \leq v_{int} \leq 1200\text{ft/s}) \land (-3.141592 \leq \theta, \psi \leq 3.141592)\}\]

**Score** \[\leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)\]

\[\{\text{argmin}_x \text{Score}[x] \notin \{\text{StrongRight, StrongLeft}\}\}\]

One Spec vs Multiple Sub-specs

**ACAS-Xu**

\[
\{(v_{own} \geq 1000 \text{ft/s}) \land (0 \leq v_{int} \leq 1200 \text{ft/s})\}
\]

\[
\text{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)
\]

\[
(\rho \geq 55947.691 \text{ft}) \land (v_{own} \geq 1145 \text{ft/s}) \land (v_{int} \leq 60 \text{ft/s})
\]

\[
\Rightarrow \quad \text{Score[COC]} \leq 1500
\]

\[
\land
\]

\[
(0 \leq \rho \leq 60760 \text{ft}) \land (1000 \leq v_{own} \leq 1200 \text{ft/s}) \land
\]

\[
(0 \leq v_{int} \leq 1200 \text{ft/s}) \land (-3.141592 \leq \theta, \psi \leq 3.141592)
\]

\[
\Rightarrow \quad \arg\min_x \text{Score}[x] \not\in \{\text{StrongRight, StrongLeft}\}
\]

**Specs 1+7**

One Spec vs Multiple Sub-specs

Multiple sub-specs generally preferred over one all-encompassing spec
- Separation of concerns
- Easy understandability
- Proofs often easier
- Modularly build spec over time
Other Ways of Specifying Properties

Source: Seshia et al, Formal Verification of Deep Neural Networks, 2018

\{(own\_velocity > 30 \text{ km/h}) \text{ and } (road\_straight\_ahead) \text{ and } (vehicles\_within\_5m = 0)\} \quad \text{Model of DNN + Controller + Plant}

\{ Steering = straight \}
Other Ways of Specifying Properties

No need for perceptual specs
  • Often easier to specify

Require models of other components
  • May be harder to verify

Classification errors of DNN may not translate to system level spec violations

\{(\text{own\_velocity} > 30 \text{ km/h}) \ \text{and} \ (\text{road\_straight\_ahead}) \ \text{and} \ (\text{vehicles\_within\_5m} = 0)\} \}

Model of DNN + Controller + Plant

\{(\text{Steering} = \text{straight})\}
Specifying Properties of Neural Networks

DNNs are intended to mimic human reasoning
   Is ideal human reasoning amenable to formal specification?

There are “boundaries” of acceptable/unacceptable human behaviour
Can we specify these boundaries?
   Rules, laws, code of conduct
      Do they have unique interpretations?
      Do they evolve?

Is there a counterpart for neural networks?