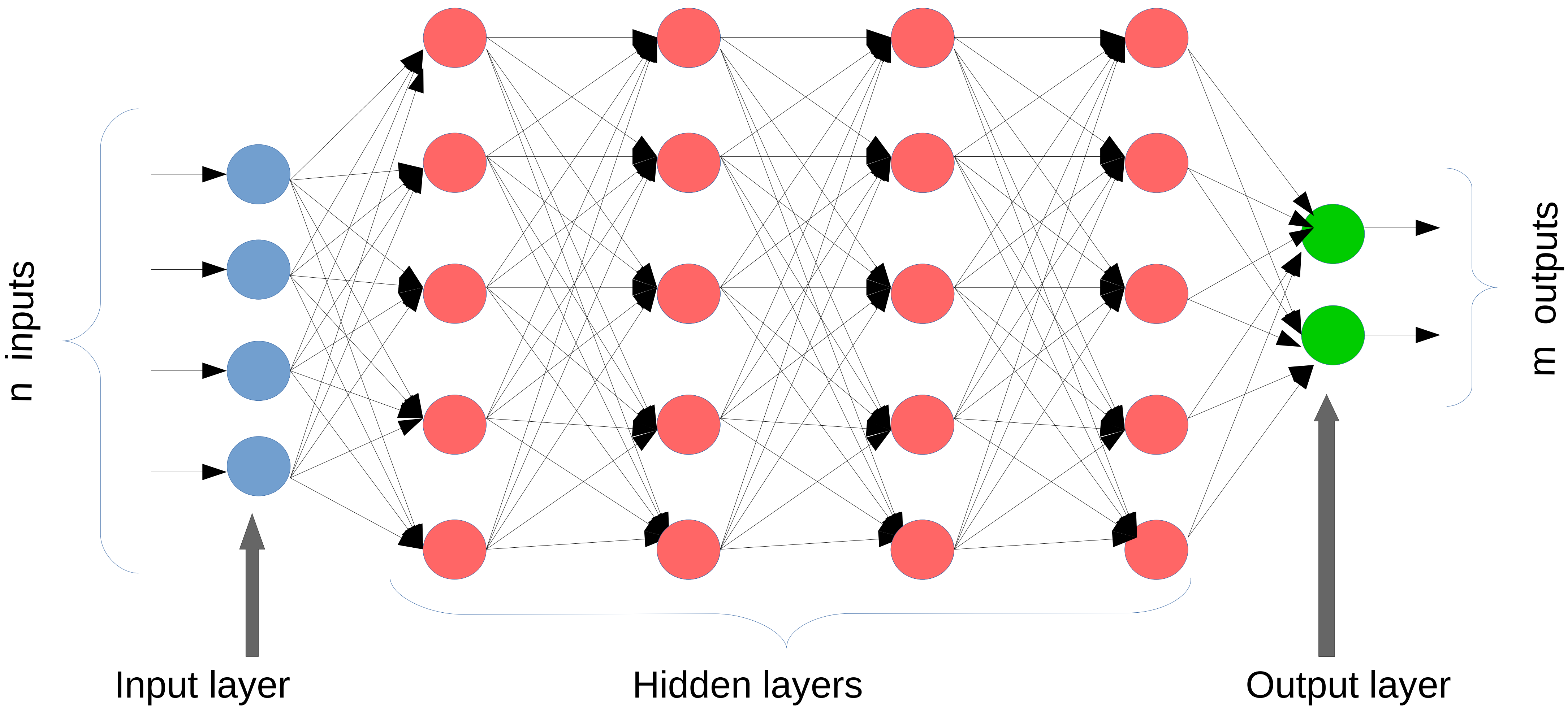
CS620: FM in ML

Specifying Properties of Neural Networks

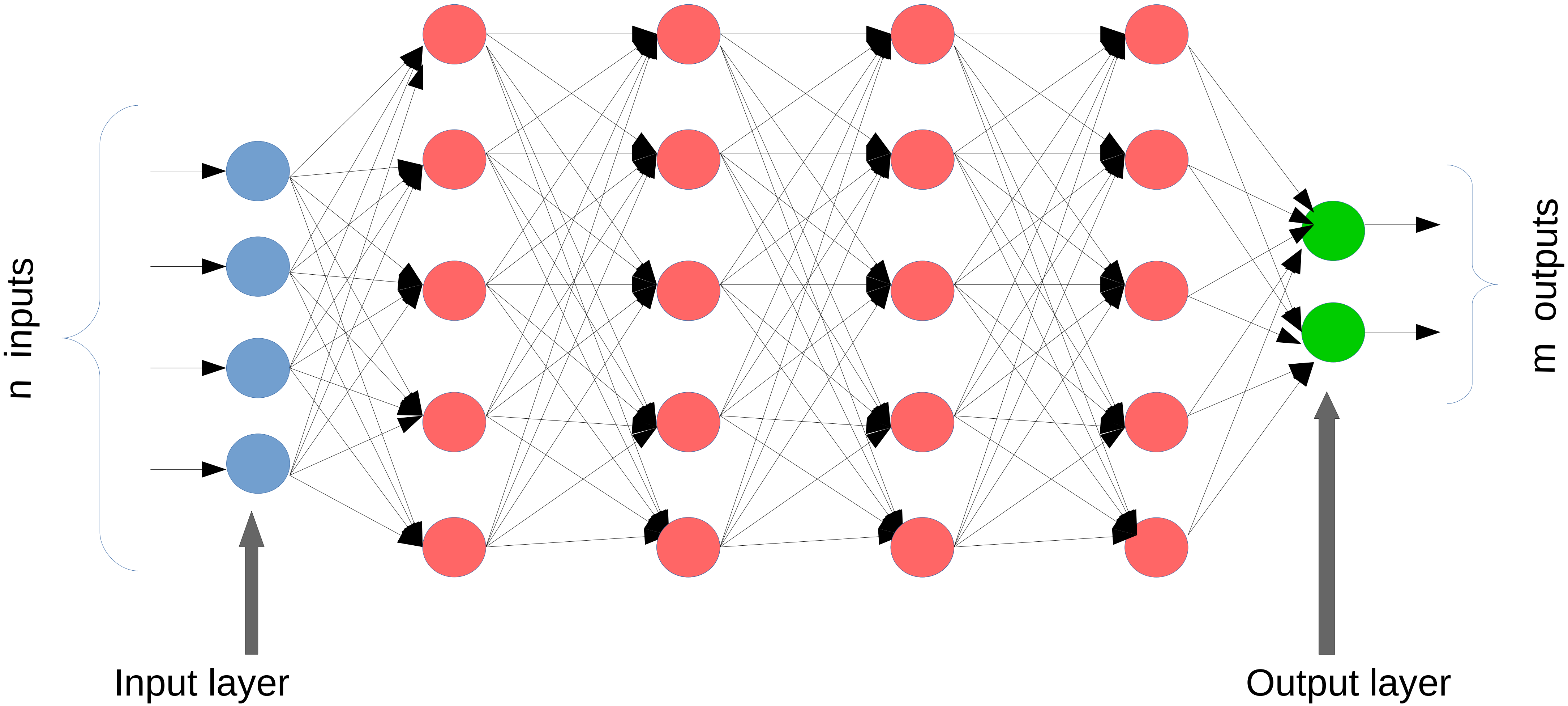
(Week 3)

Supratik Chakraborty

A Typical Neural Network



A Typical Neural Network



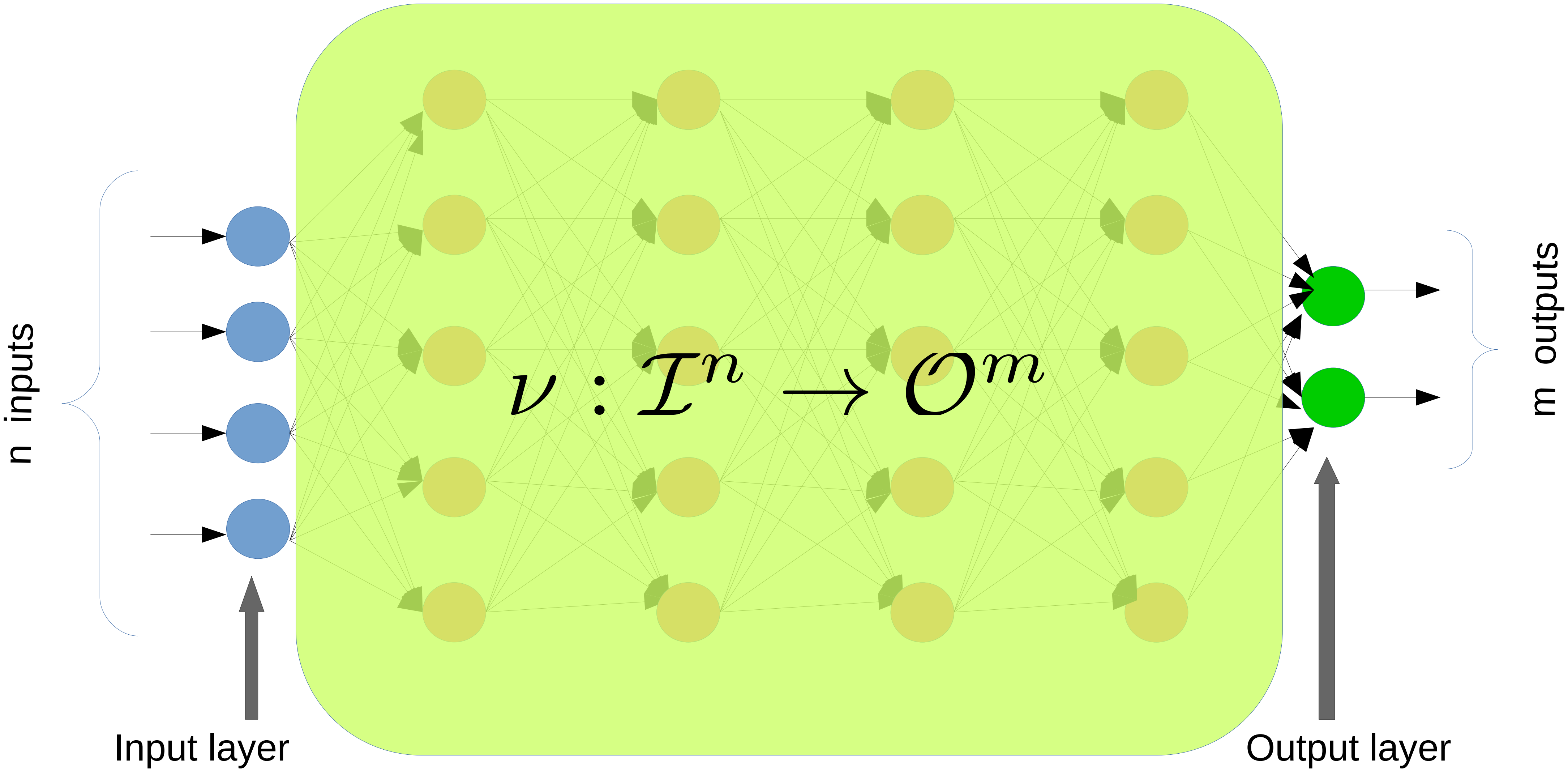
Input Domain: $\mathcal{I}$

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

Output Domain: $\mathcal{O}$

Output space: $\mathcal{O}^m$

A Typical Neural Network



Input Domain: $\mathcal{I}$

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

Output Domain: $\mathcal{O}$

Output space: $\mathcal{O}^m$

A Transformative Program

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

Input: $x \in \mathcal{I}^n$

First order logic formula

$y \leftarrow \nu(x);$

Output: $y \in \mathcal{O}^m$

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

Hoare triples similar to those used in program verification

Semantics of Hoare Triple

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

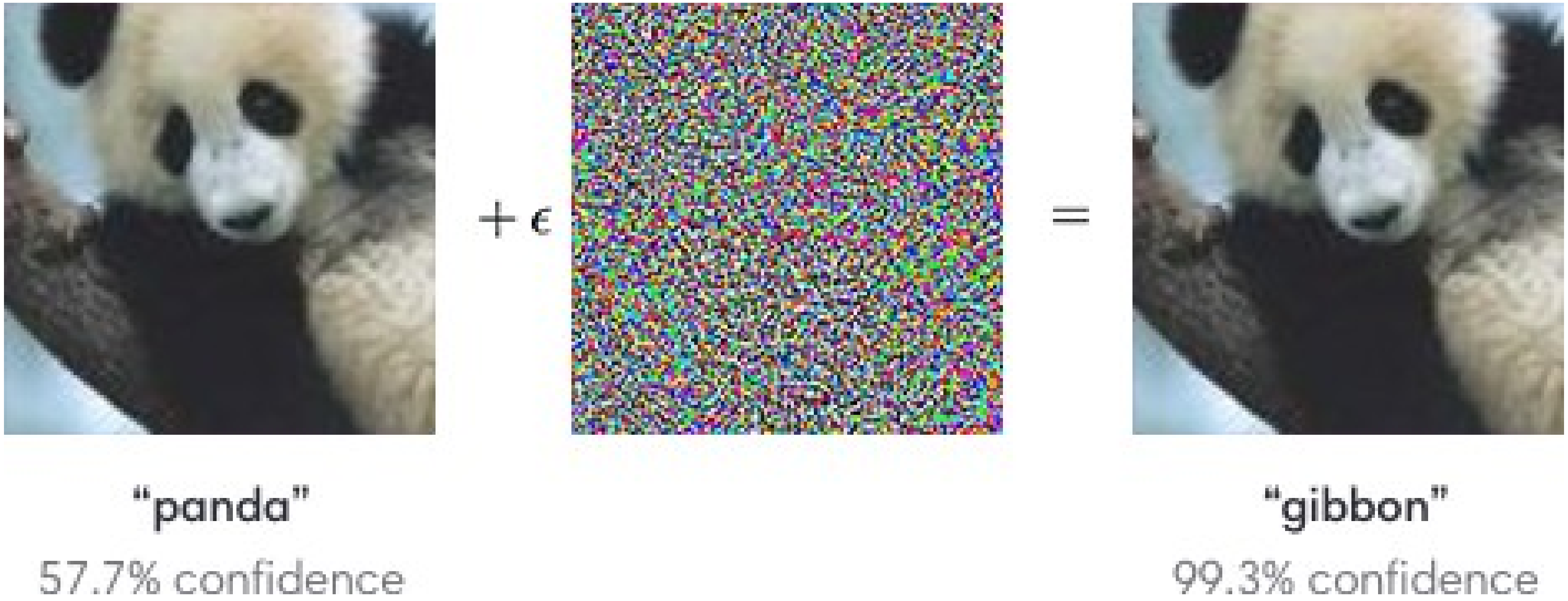
$$y \leftarrow \nu(x); \cdots \text{ "Program"}$$

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

Validity of Hoare triple

If x satisifes $\varphi(x)$,
"program" terminates and encounters no memory exception,
then output y always satisfies $\psi(y)$

Property Specification Example 1

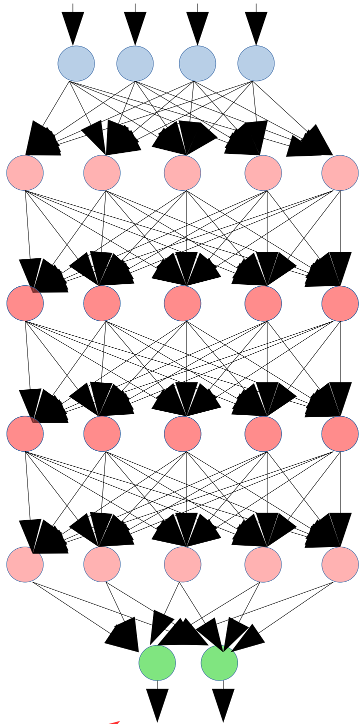


Source: Goodfellow, Shlens, Szegedy, “Explaining and Harnessing Adversarial Examples”,2015

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

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

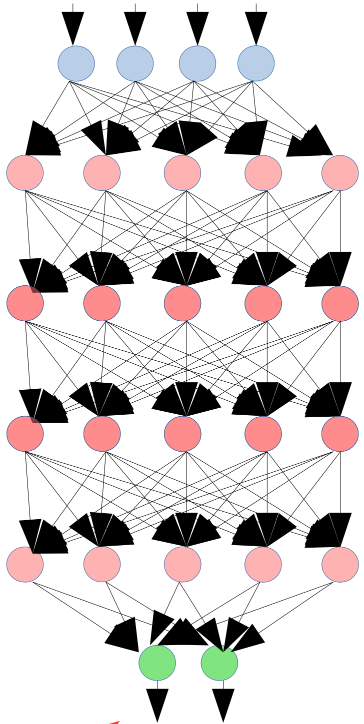
$$\{p > g + \delta\}$$

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

$$\{\|x - x^*\| \leq \varepsilon\}$$

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

$$(p, g) \leftarrow \nu(x);$$

$$\{p > g + \delta\}$$

Spec as a logical requirement

$$\forall r_1 \forall g_1 \forall b_1 \cdots \forall r_N \forall g_N \forall b_N \forall p \forall g$$

$$\left(\begin{array}{c} \bigwedge_{i=1}^{N} \left(|r_i - r_i^*| \le \varepsilon_r\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|g_i - g_i^*| \le \varepsilon_g\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|b_i - b_i^*| \le \varepsilon_b\right) \wedge \\ (p, g) = \nu\big(r_1, g_1, b_1, \ldots r_N, g_N, b_N\big) \\ \\ \Longrightarrow \\ \\ p > g + \delta \end{array} \right)$$

$$\{\|x - x^*\| \le \varepsilon\}$$

$$\begin{array}{l} \bigwedge_{i=1}^{N} \left(|r_i - r_i^*| \le \varepsilon_r\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|g_i - g_i^*| \le \varepsilon_g\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|b_i - b_i^*| \le \varepsilon_b\right) \end{array}$$

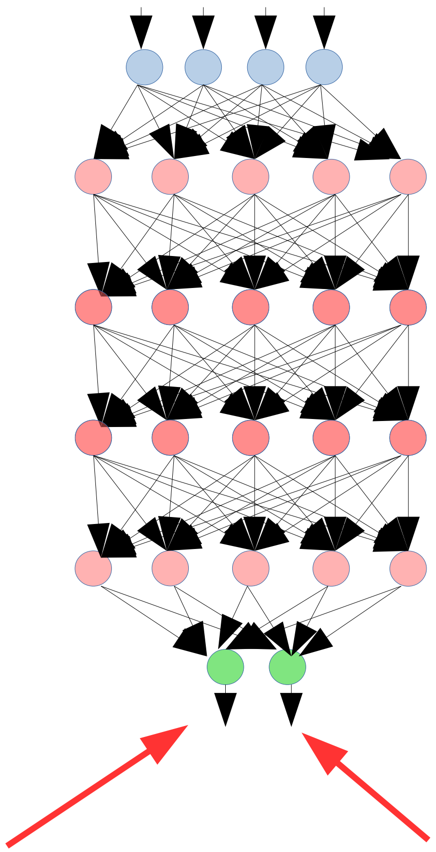
$$(p, g) \leftarrow \nu(x);$$

$$\{p > g + \delta\}$$

A logical implication

Property Specification Example 2

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



$$\{\|x - x^*\| \leq \varepsilon\}$$

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

$$\begin{Bmatrix} (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \end{Bmatrix}$$

Property Specification Example 2

Pause n Reflect

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

$$\{\|x - x^*\| \leq \varepsilon\}$$

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

$$\begin{Bmatrix} (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \end{Bmatrix}$$

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

$$\forall r_1 \cdots \forall b_N \forall r_1^* \cdots \forall b_N^* \forall s_1 \forall s_2 \forall s_1^* \forall s_2^*$$

$$\left(\begin{array}{c} \bigwedge_{i=1}^{N} \left(|r_i - r_i^*| \leq \varepsilon_r\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|g_i - g_i^*| \leq \varepsilon_g\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|b_i - b_i^*| \leq \varepsilon_b\right) \wedge \\ (s_1, s_2) = \nu\left(r_1, g_1, b_1, \ldots r_N, g_N, b_N\right) \wedge \\ (s_1^*, s_2^*) = \nu\left(r_1^*, g_1^*, b_1^*, \ldots r_N^*, g_N^*, b_N^*\right) \wedge \\ \\ \implies \\ \\ (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \end{array}\right)$$

$$\{\|x - x^*\| \leq \varepsilon\}$$

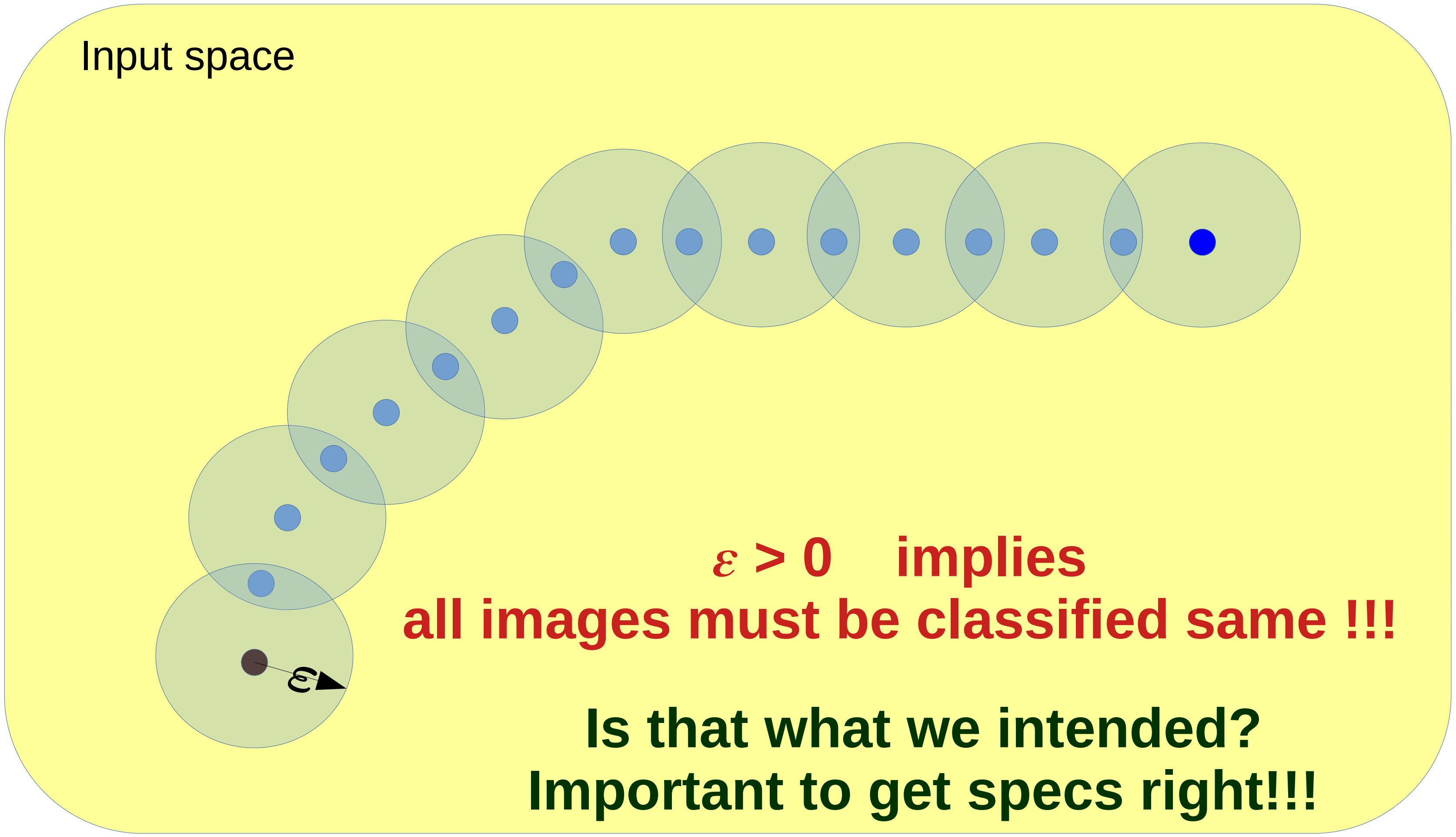
$$\begin{array}{c} \bigwedge_{i=1}^{N} \left(|r_i - r_i^*| \leq \varepsilon_r\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|g_i - g_i^*| \leq \varepsilon_g\right) \wedge \\ \bigwedge_{i=1}^{N} \left(|b_i - b_i^*| \leq \varepsilon_b\right) \end{array}$$

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

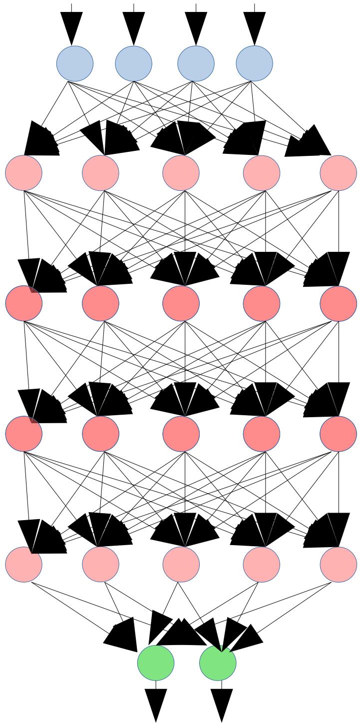
$$\left\{ \qquad \right\}$$

Problem with Specification 2

Pick any two arbitrary images in the input space



Taking a step back to re-look



Arbitrary input

$$\{\|x - x^*\| \leq \varepsilon\}$$

Specific input

$$(p, g) \leftarrow \nu(x)$$

$$\{p > g + \delta\}$$

Arbitrary input

$$\{\|x - x^*\| \leq \varepsilon\}$$

Arbitrary input

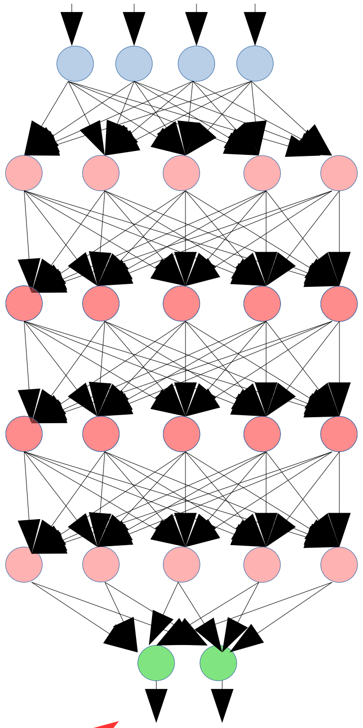
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

$$\left\{\begin{array}{l}(s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta)\end{array}\right\}$$

Attempting a Fix

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

Arbitrary image x



$$\{\|x - x^*\| \leq \varepsilon\}$$

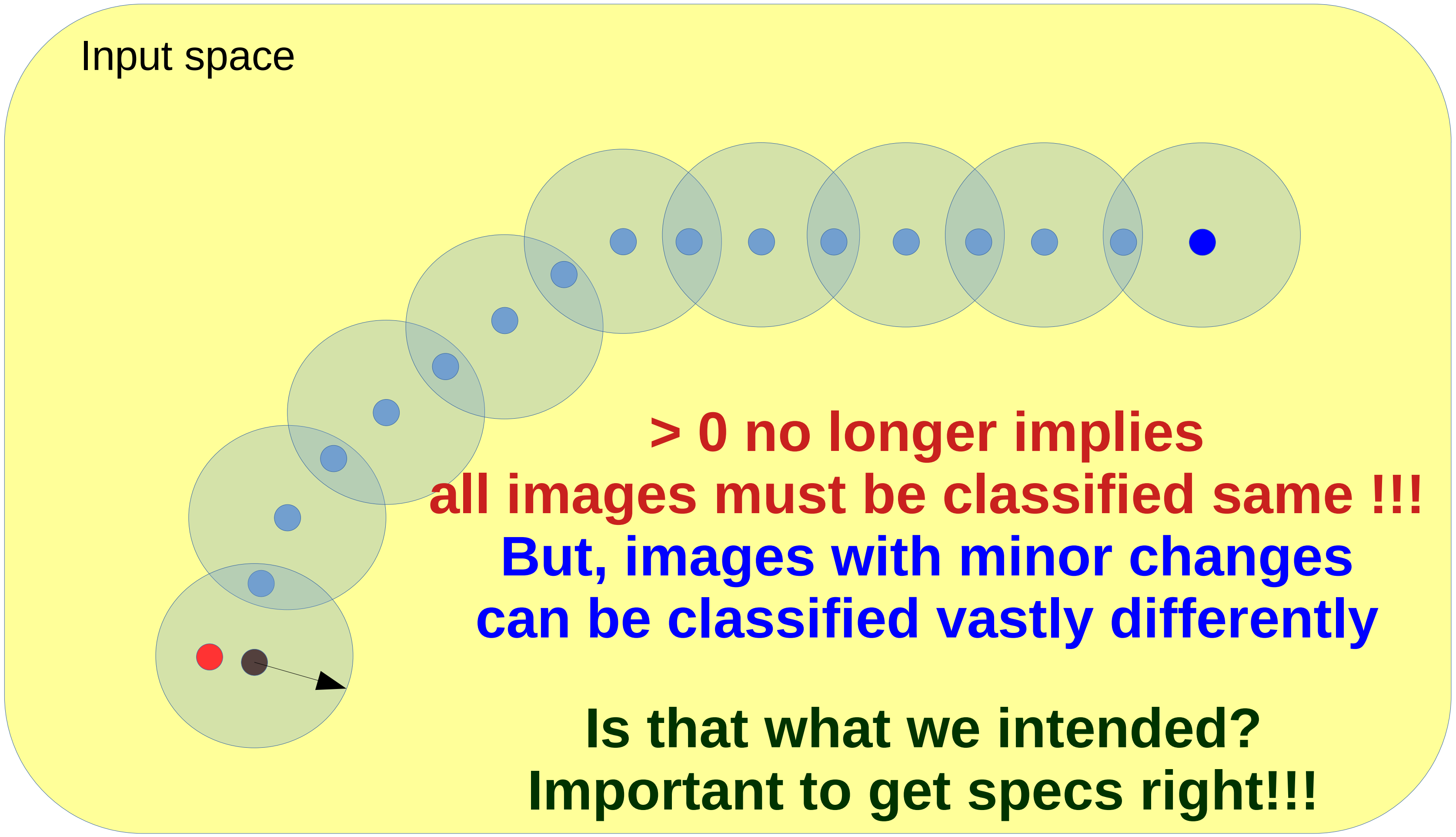
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

$$\left\{ \begin{array}{l} (s_1 > s_2 + \delta) \implies (s_2^* \leq s_1^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_1^* \leq s_2^* + \delta) \end{array} \right\}$$

Score for class 1: s1 **Score for class 2: s2**

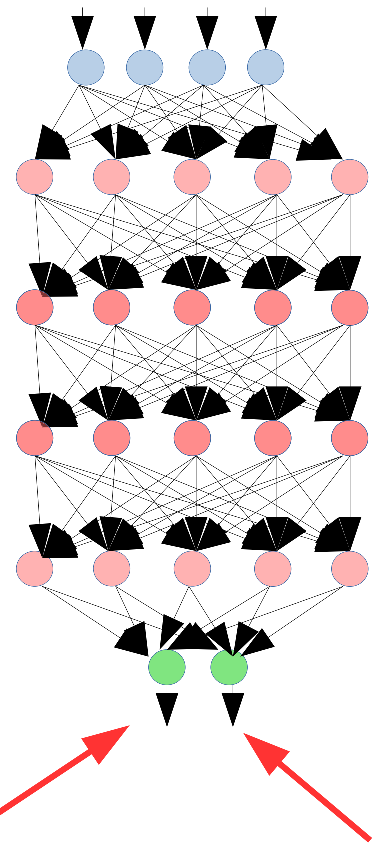
Did It Fix?

Pick any two arbitrary images in the input space



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



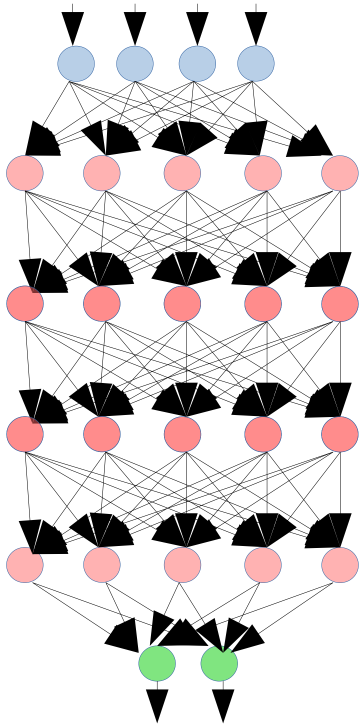
$$\left\{ \left(\|x - x^*\| \leq \varepsilon \right) \wedge \left(\sigma(x) \approx \sigma(x^*) \right) \right\}$$

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

$$\left\{ \begin{array}{l} (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \end{array} \right\}$$

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



$$\{(\|x - x^*\| \leq \varepsilon) \wedge (\sigma(x) \approx \sigma(x^*))\}$$

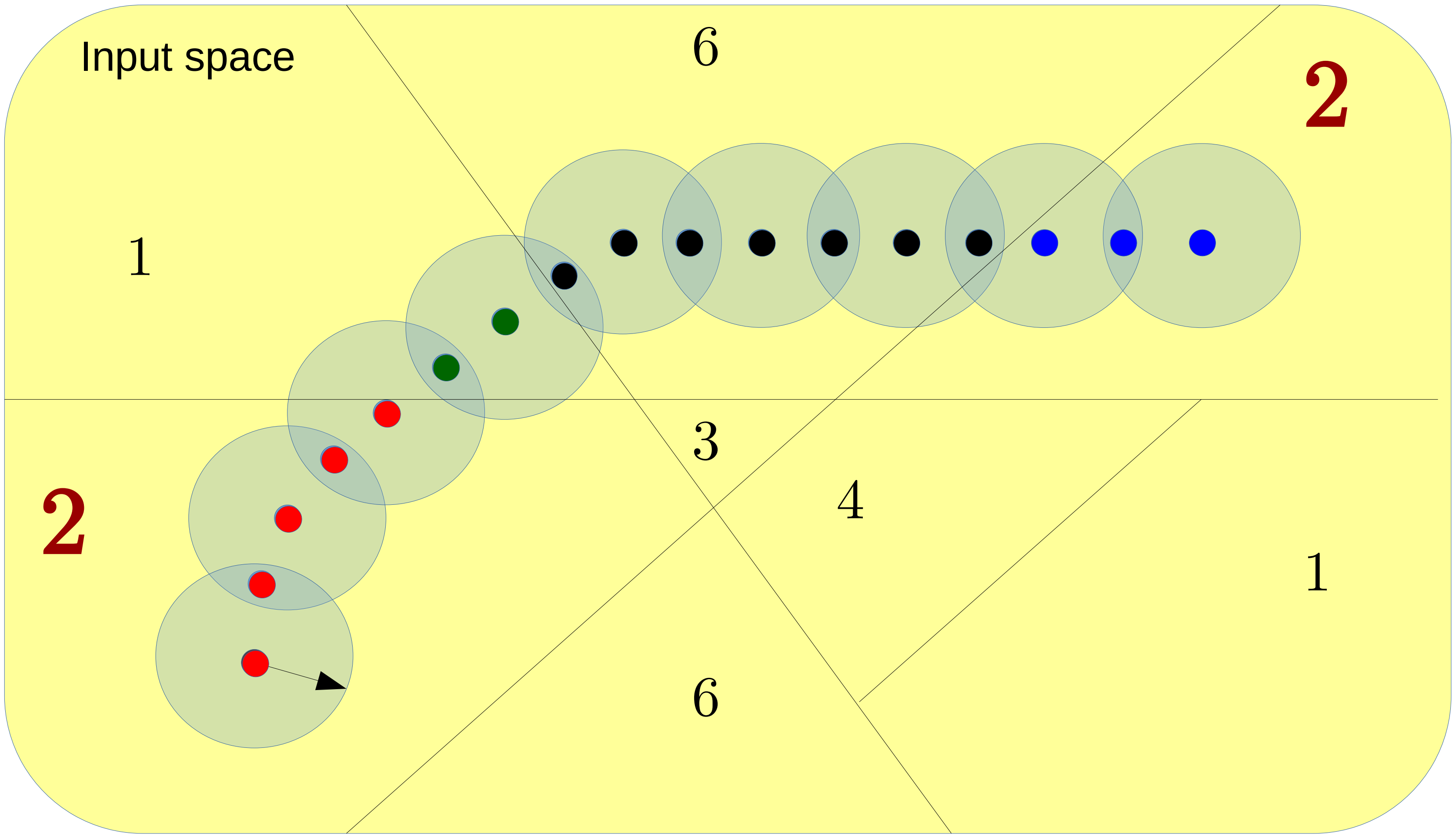
User-defined semantic features,
Not necessarily network-defined

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

$$\left\{\begin{array}{l}(s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta)\end{array}\right\}$$

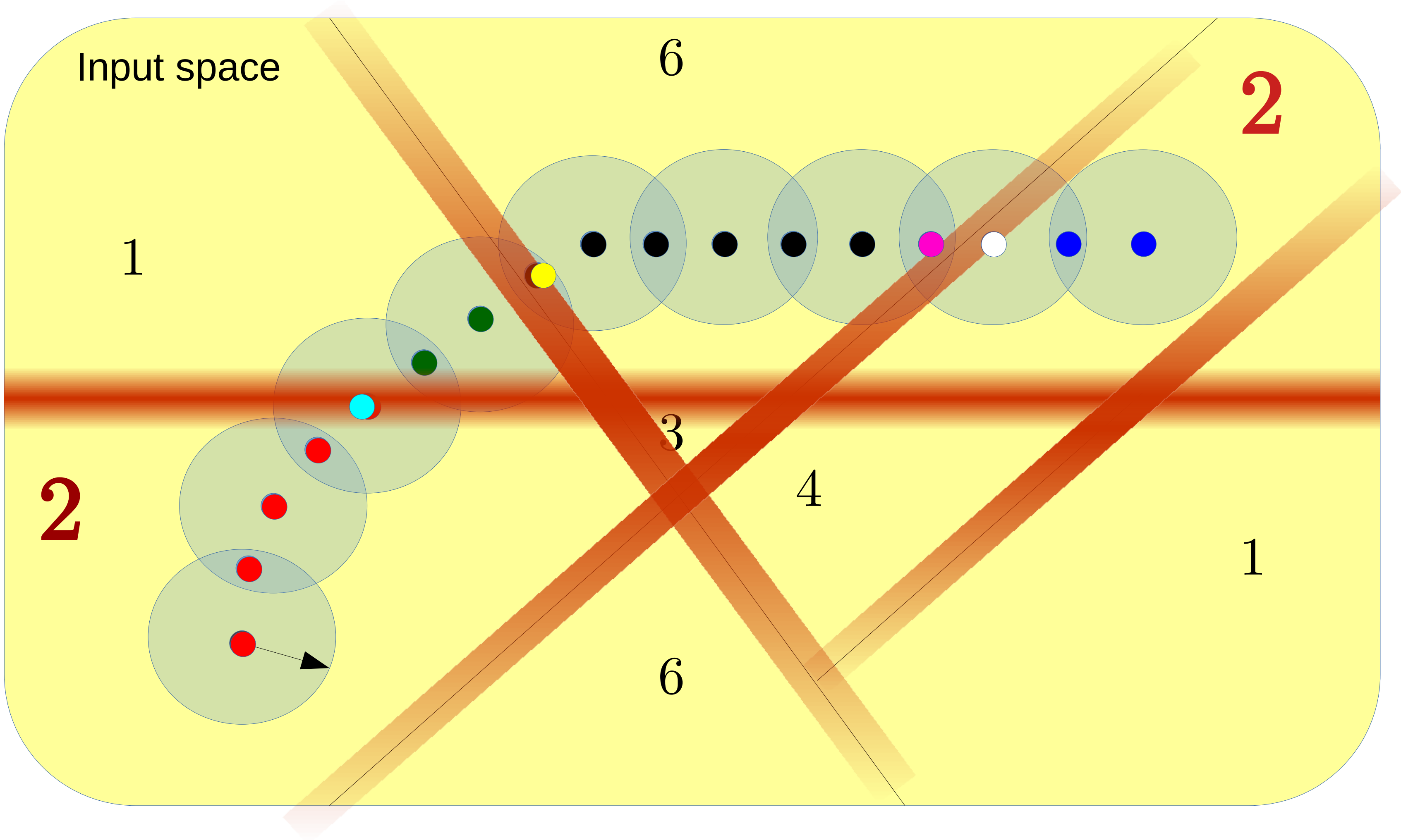
Possibilities with New Spec

Pick any two arbitrary images in the input space



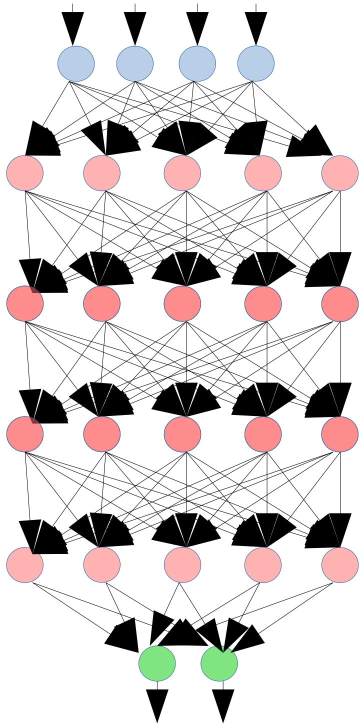
Possibilities with Newer Spec

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



$$\{(\|x - x^*\| \leq \varepsilon) \wedge (\sigma(x) \approx \sigma(x^*))\}$$

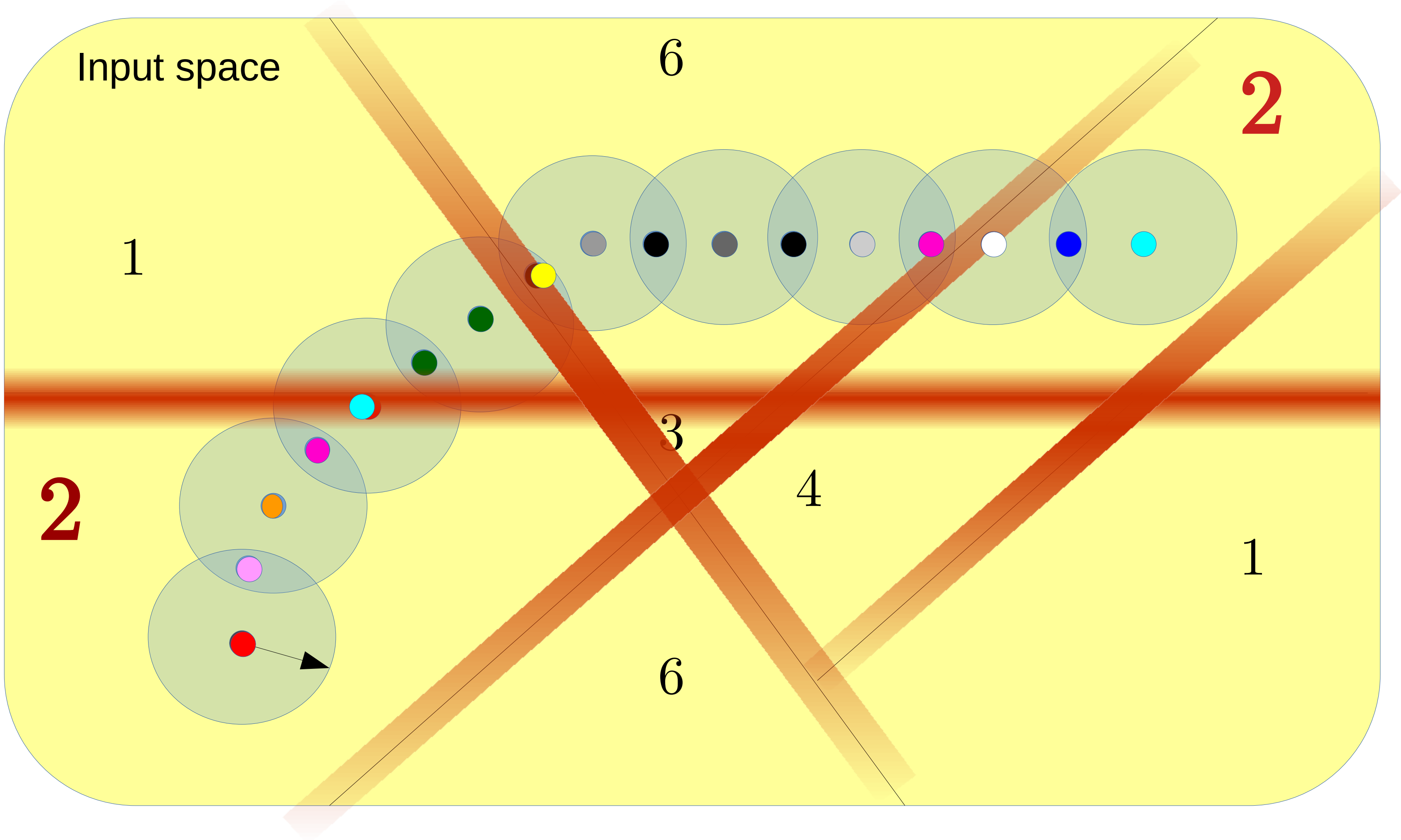
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

$$\{(\lambda(s_1, s_2) \simeq \lambda(s_1^*, s_2^*))\}$$

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?

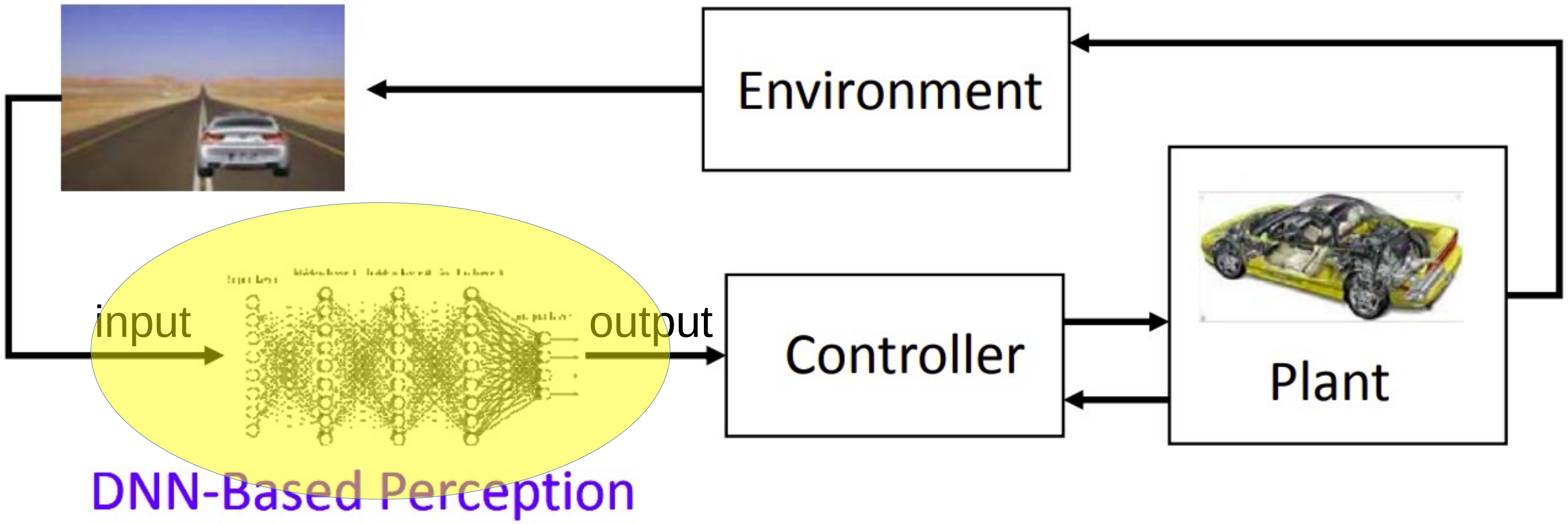
Is it easier to arrive at

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"



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

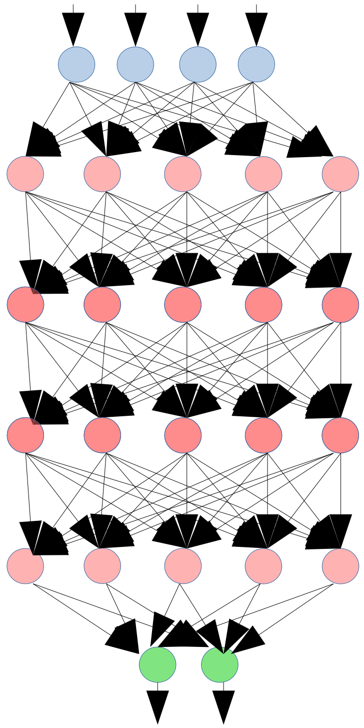
Collect a bunch of **desired/undesired** (input, output) pairs
- Not necessarily what DNN is actually doing
- Instead, what DNN's environment "expects" it to do

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)

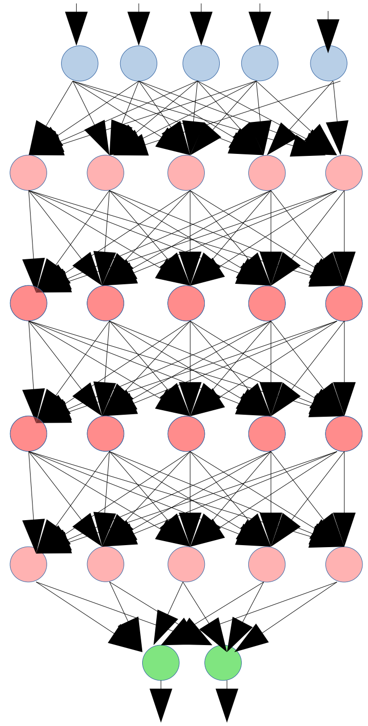


Perceptual Spec

“Too congested to accelerate”

Unmanned drone

Flight parameters

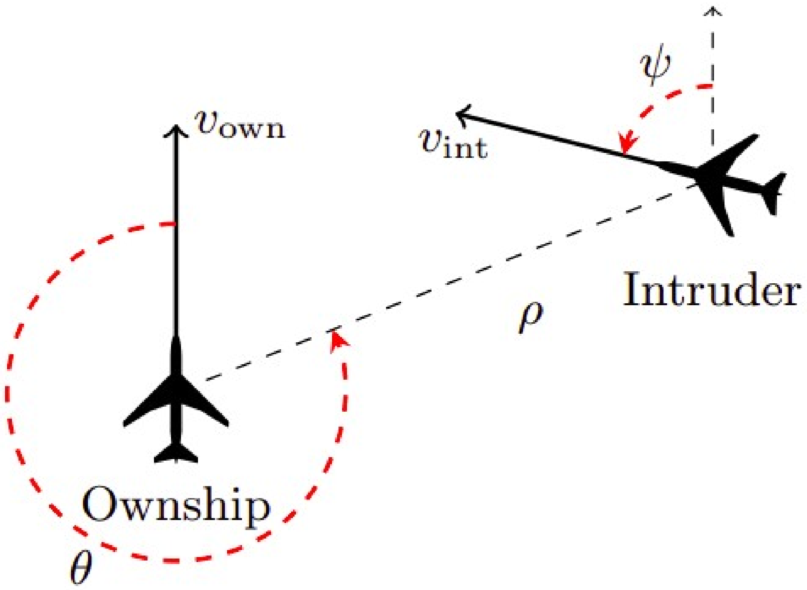


Non-Perceptual Spec

Score
(Horizontal Advisory)

Non-Perceptual DNN Specs

ACAS-Xu



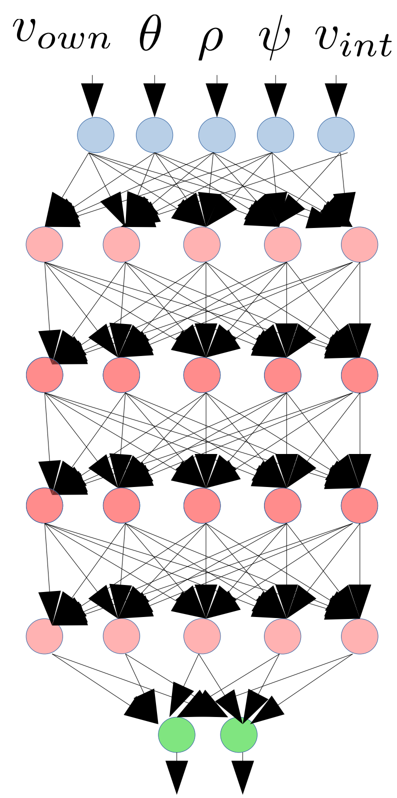
$$\{(\rho \geq 55947.691ft) \wedge (v_{own} \geq 1145ft/s) \wedge (v_{int} \leq 60ft/s)\}$$

$$\textbf{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)$$

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

Clear-of-Conflict

Flight parameters



Score (Horizontal Advisory)

Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

Non-Perceptual DNN Specs

ACAS-Xu

Flight parameters

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

$\{\mathbf{Score}[\mathrm{COC}] \leq 1500\}$

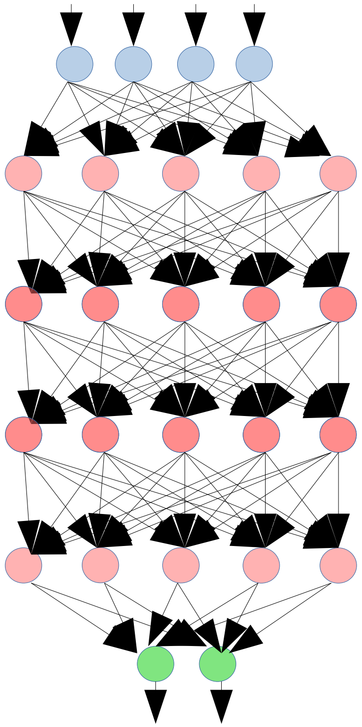
Score (Horizontal Advisory)

Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

Perceptual DNN Specs

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

Image (road scene)



“Too congested to accelerate”

Good spec:

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

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

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

{ y = “Too congested to accelerate” }

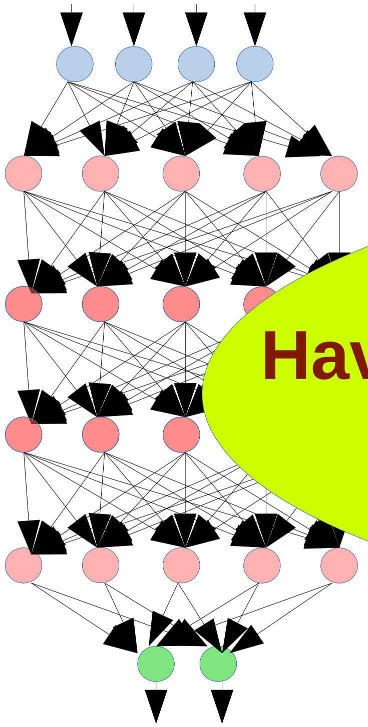
CR(...) = true iff road is “too congested to accelerate”

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

Perceptual DNN Specs

$(r_1, g_1, b_1, \dots r_N, g_N, b_N)$

Image (road scene)



“Too congested to accelerate”

Good spec:

CR $(r_1, g_1, b_1, \dots r_N, g_N, b_N)$

$\{$ ~~$(r_1, g_1, b_1, \dots r_N, g_N, b_N)$: image of congested road~~ $\}$

$g_N, b_N);$

e” }

sted to accelerate”

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

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

Specifying Properties of Perceptual DNNs

Pause n Reflect

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

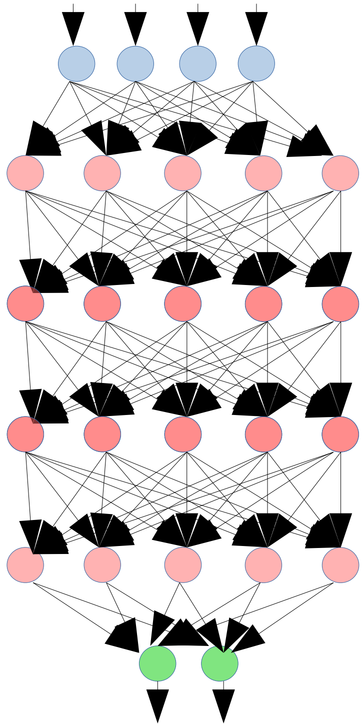
Can we specify anything formally beyond this?

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

Any Hope for Perceptual DNNs?

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

Image (road scene)



“Too congested to accelerate”

Input i

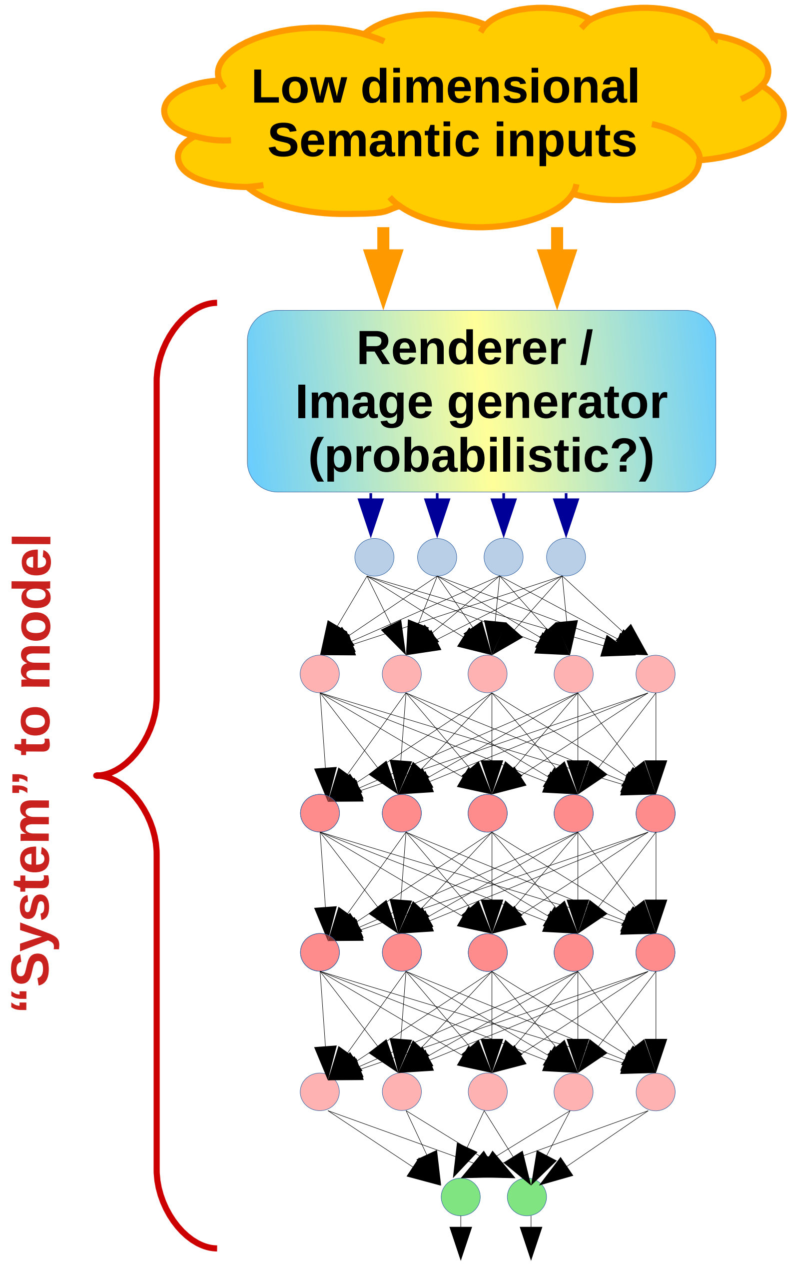
High dimensional, large input space

| Input Space |

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?



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}

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

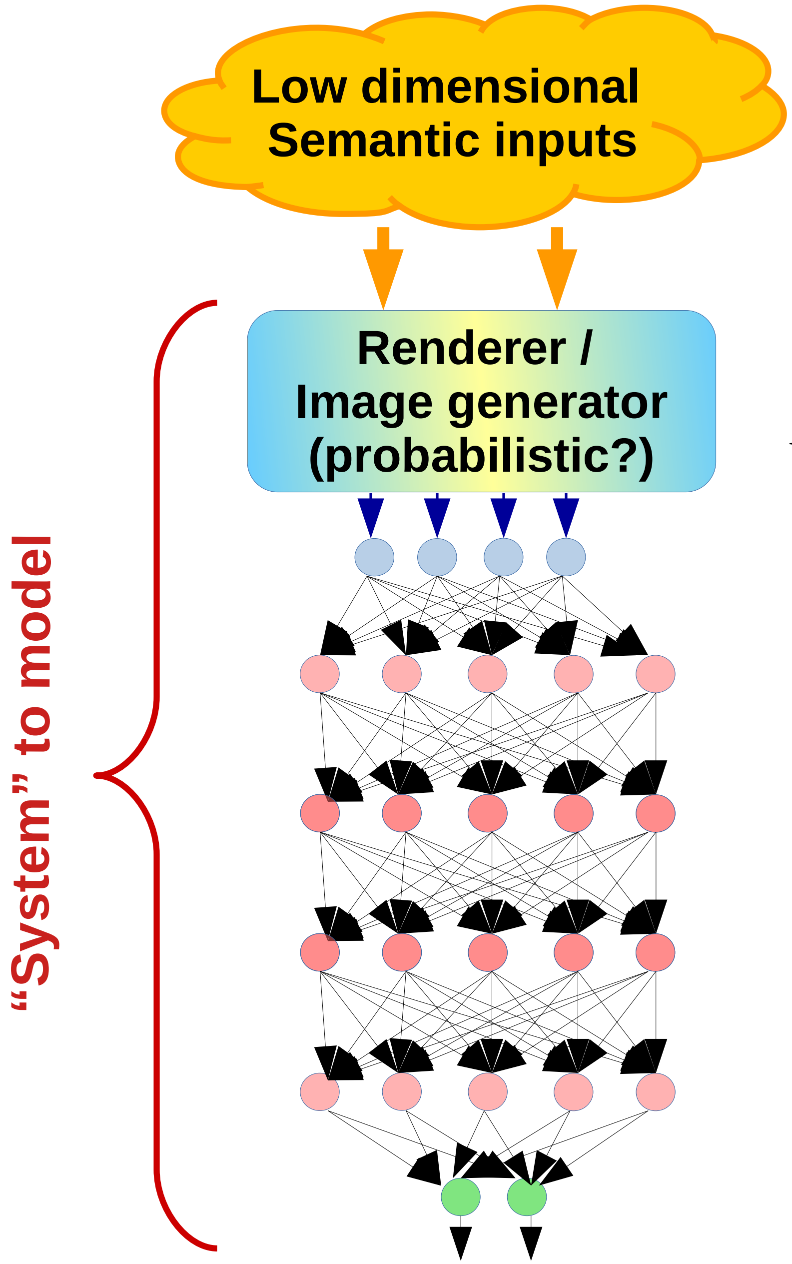
Dimensions of image inp space = 100x100x3 = 30000
|Image inp space| = $256^{100x100x3}$

{ Pre-condition on semantic inputs **s** }

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

{ Post-condition on y}

Any Hope for Perceptual DNNs?



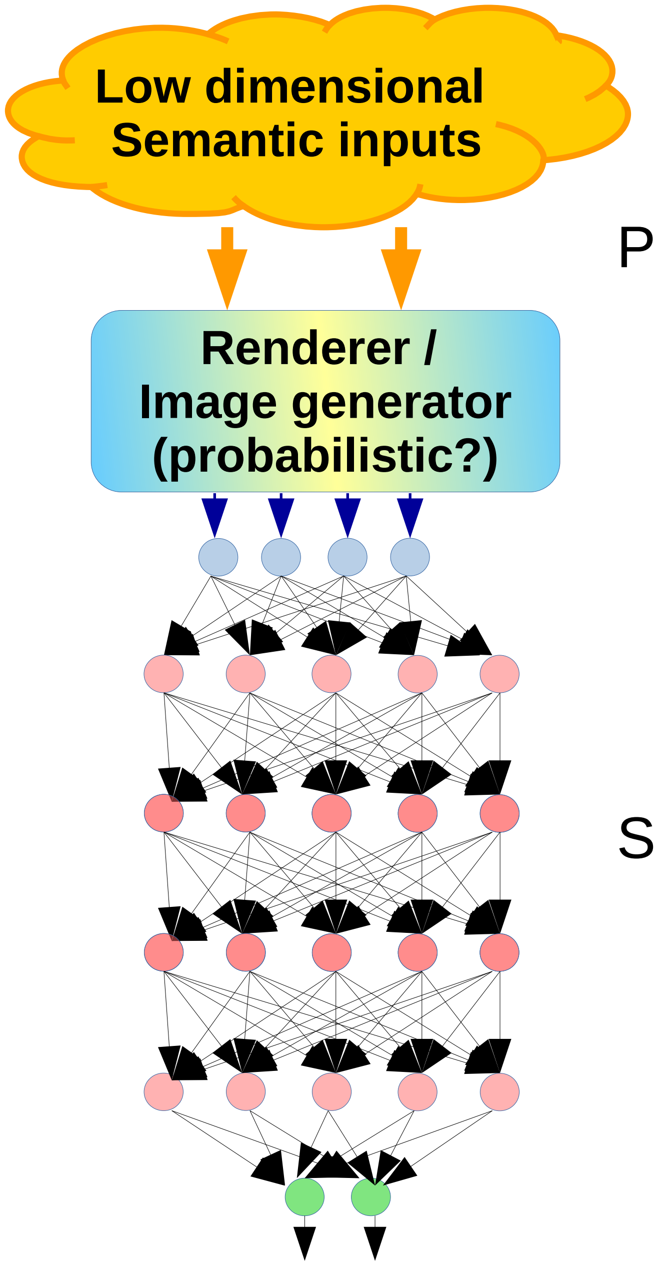
T: {Morning, Noon, Afternoon, Dusk, Night}
W: {Clear, Cloudy, Snowing, Raining}
L: {Wide, Medium, Narrow, None}
Rd: {Straight, Bending}
O: {0, 1-3, 4-8, 9-15, > 15}
B: {Lane disciplined, Chaotic}

$$
\begin{aligned}
&\{(O > 15) \;\bigvee \\
&\;(L = \mathrm{W}) \;\wedge ((O \geq 9) \wedge (B = \mathrm{Ch})) \;\bigvee \\
&\;(L = \mathrm{M}) \;\wedge\; ((O \geq 9) \vee ((O \geq 4) \wedge (B = \mathrm{Ch}))) \;\bigvee \\
&\;((L = \mathrm{N}) \vee (L = \mathrm{None})) \;\wedge\; ((O \geq 4) \vee ((O \geq 1) \wedge (B = \mathrm{Ch})))\}
\end{aligned}
$$

$$
\begin{aligned}
&i \leftarrow \rho(T, W, L, Rd, O, B);\ //\rho\text{: Model of renderer} \\
&y \leftarrow \nu(i);\ //\nu\text{: Model of perceptual DNN}
\end{aligned}
$$

{ y = “Too congested to accelerate” }

Any Hope for Perceptual DNNs?



Potential **"problems"**:

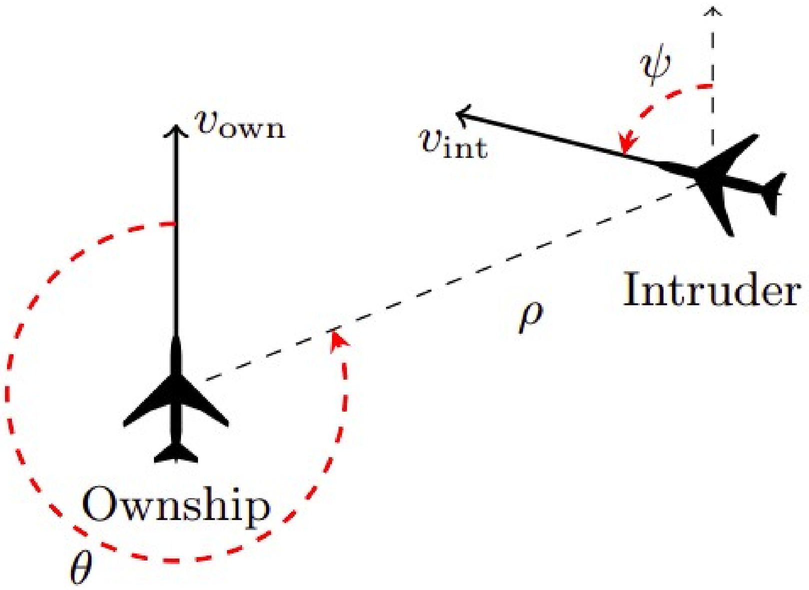
- Doesn't cover entire input space
  - Enrich semantic space to cover most/all meaningful inputs
  - Use richer rendering modules
- Need to model renderer
  - Use abstract / non-deterministic / probabilistic models

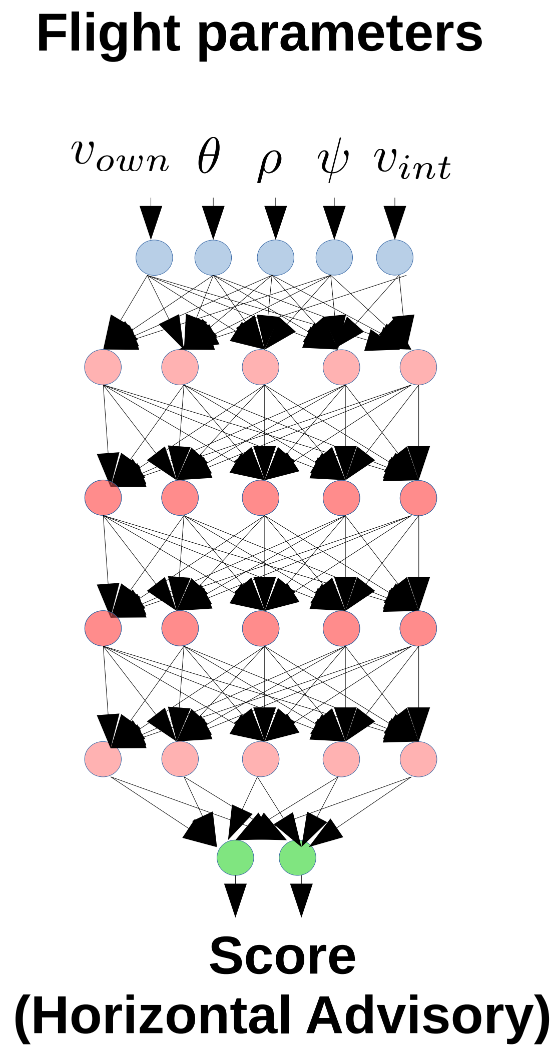
Significant **"benefits"**:

- 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





$$\{(\rho \geq 55947.691ft) \wedge (v_{own} \geq 1145ft/s) \wedge (v_{int} \leq 60ft/s)\}$$

$$\mathbf{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)$$

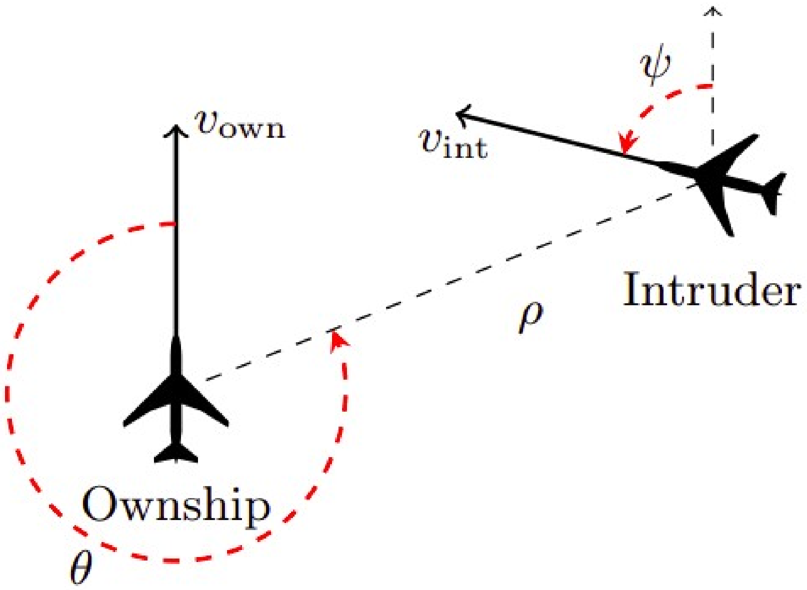
$$\{\mathbf{Score}[\text{COC}] \leq 1500\}$$

Spec 1

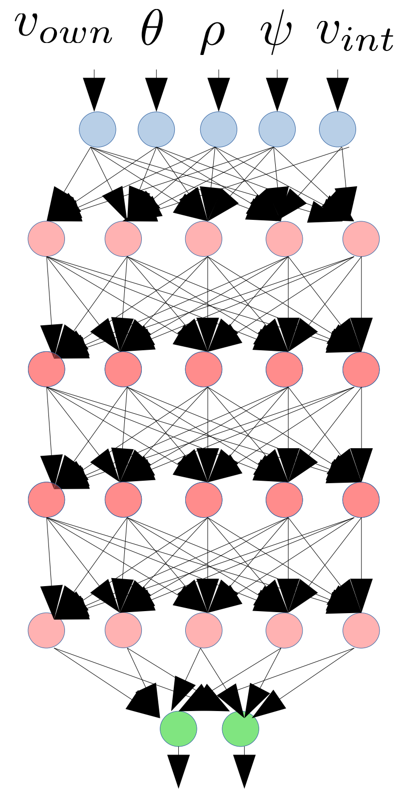
Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

One Spec vs Multiple Sub-specs

ACAS-Xu



Flight parameters



Score
(Horizontal Advisory)

$$\{(0 \leq \rho \leq 60760ft) \wedge (1000 \leq v_{own} \leq 1200ft/s) \wedge$$
$$(0 \leq v_{int} \leq 1200ft/s) \wedge (-3.141592 \leq \theta, \psi \leq 3.141592)\}$$

$$\mathbf{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)$$

Spec 7

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

Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

One Spec vs Multiple Sub-specs

ACAS-Xu

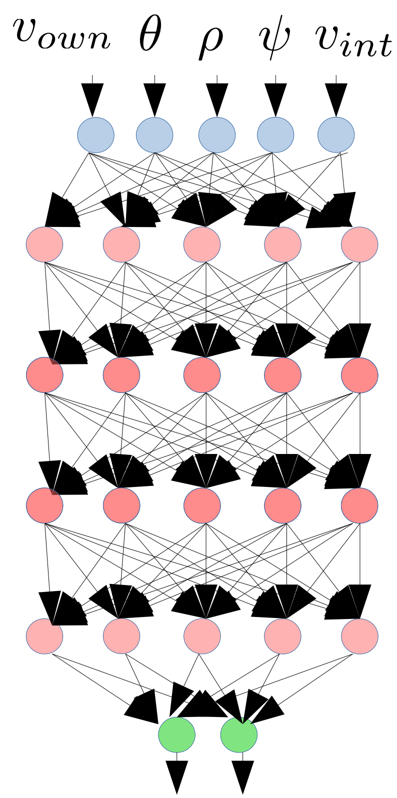
$$\{(v_{own} \geq 1000ft/s) \ \wedge \ (0 \leq v_{int} \leq 1200ft/s)\}$$

$$\textbf{Score} \leftarrow \nu(\rho, v_{own}, v_{int}, \theta, \psi)$$

$$\left\{ \begin{array}{l} (\rho \geq 55947.691ft) \wedge (v_{own} \geq 1145ft/s) \wedge (v_{int} \leq 60ft/s) \\ \qquad \Rightarrow \quad \textbf{Score}[\text{COC}] \leq 1500 \\ \\ \qquad\qquad\qquad\qquad \bigwedge \\ \\ (0 \leq \rho \leq 60760ft) \wedge (1000 \leq v_{own} \leq 1200ft/s) \wedge \\ \quad (0 \leq v_{int} \leq 1200ft/s) \wedge (-3.141592 \leq \theta, \psi \leq 3.141592) \\ \qquad \Rightarrow \quad \text{argmin}_x \textbf{Score}[\text{x}] \notin \{\text{StrongRight}, \text{StrongLeft}\} \end{array} \right\}$$

Specs 1+7

Flight parameters



Score (Horizontal Advisory)

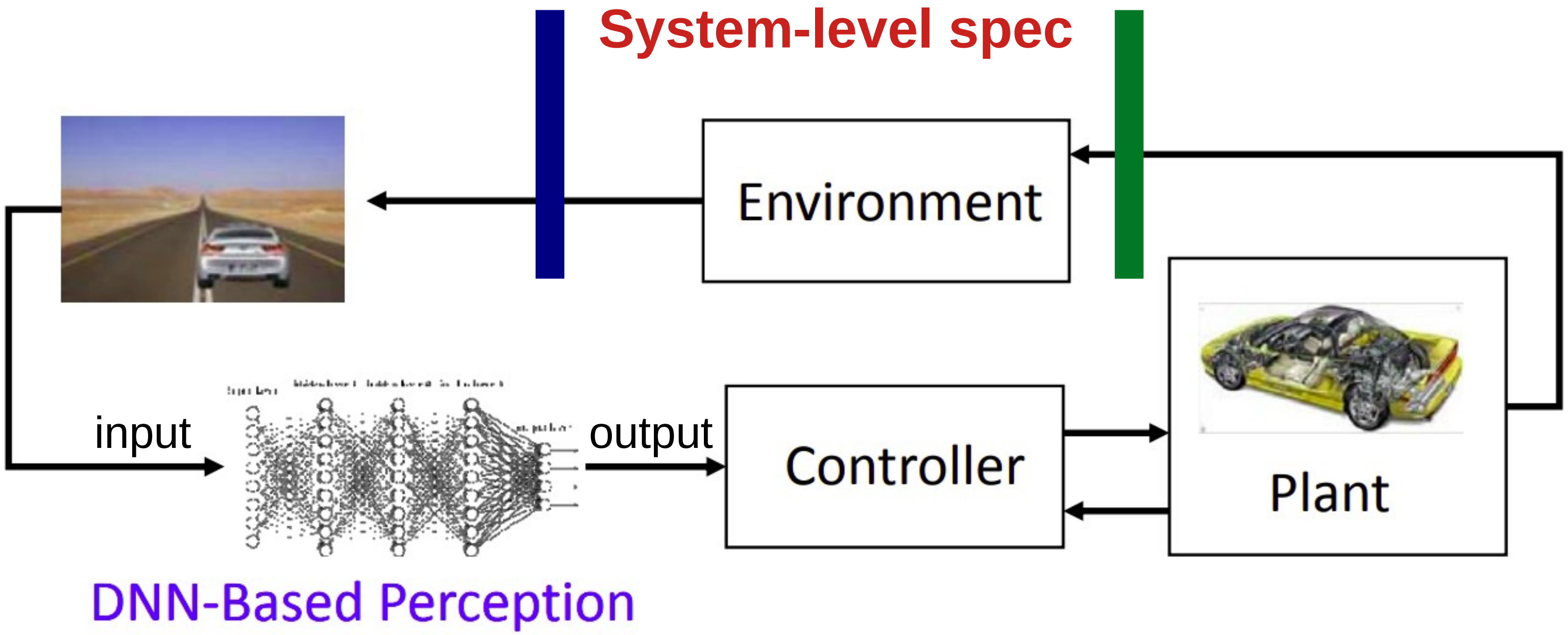
Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

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 km/h) and (road_straight_ahead) and (vehicles_within_5m = 0) }

Model of DNN + Controller + Plant

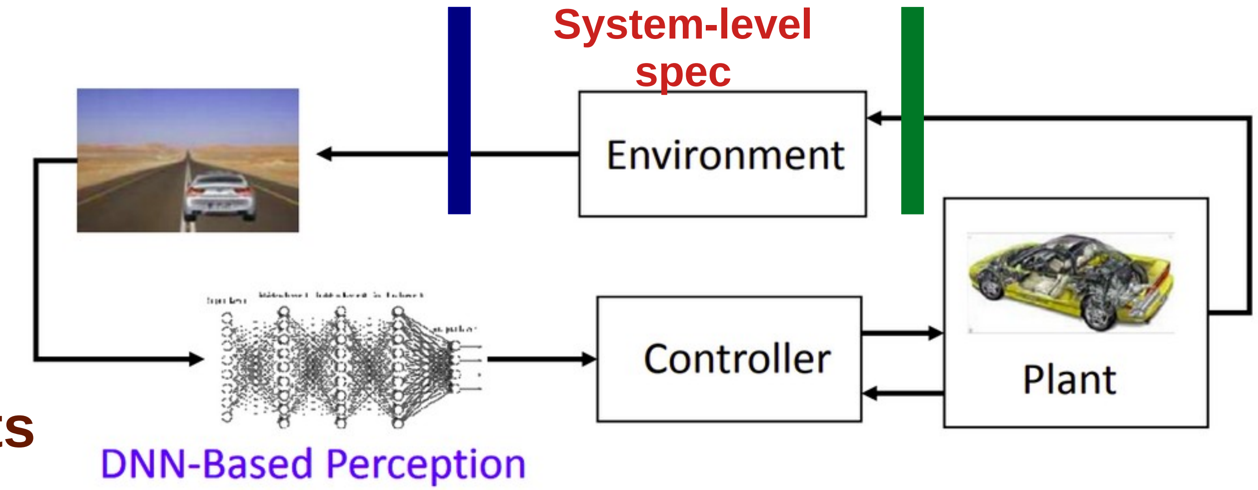
{ Steering = straight }

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



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

{ (own_velocity > 30 km/h) and (road_straight_ahead) and (vehicles_within_5m = 0) }

Model of DNN + Controller + Plant

{ Steering = straight }

Specifying Properties of Neural Networks

Pause n Reflect

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?