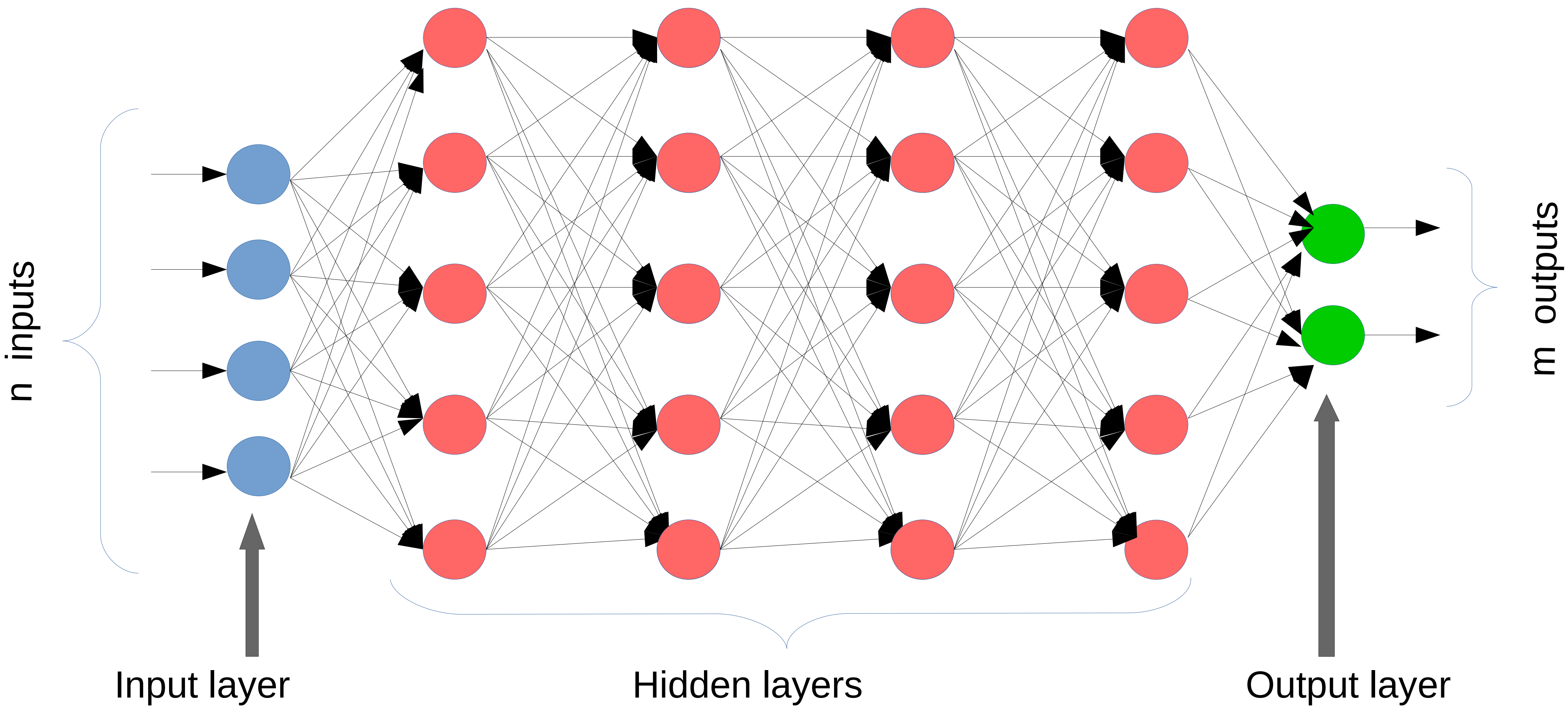
CS781: FM in ML

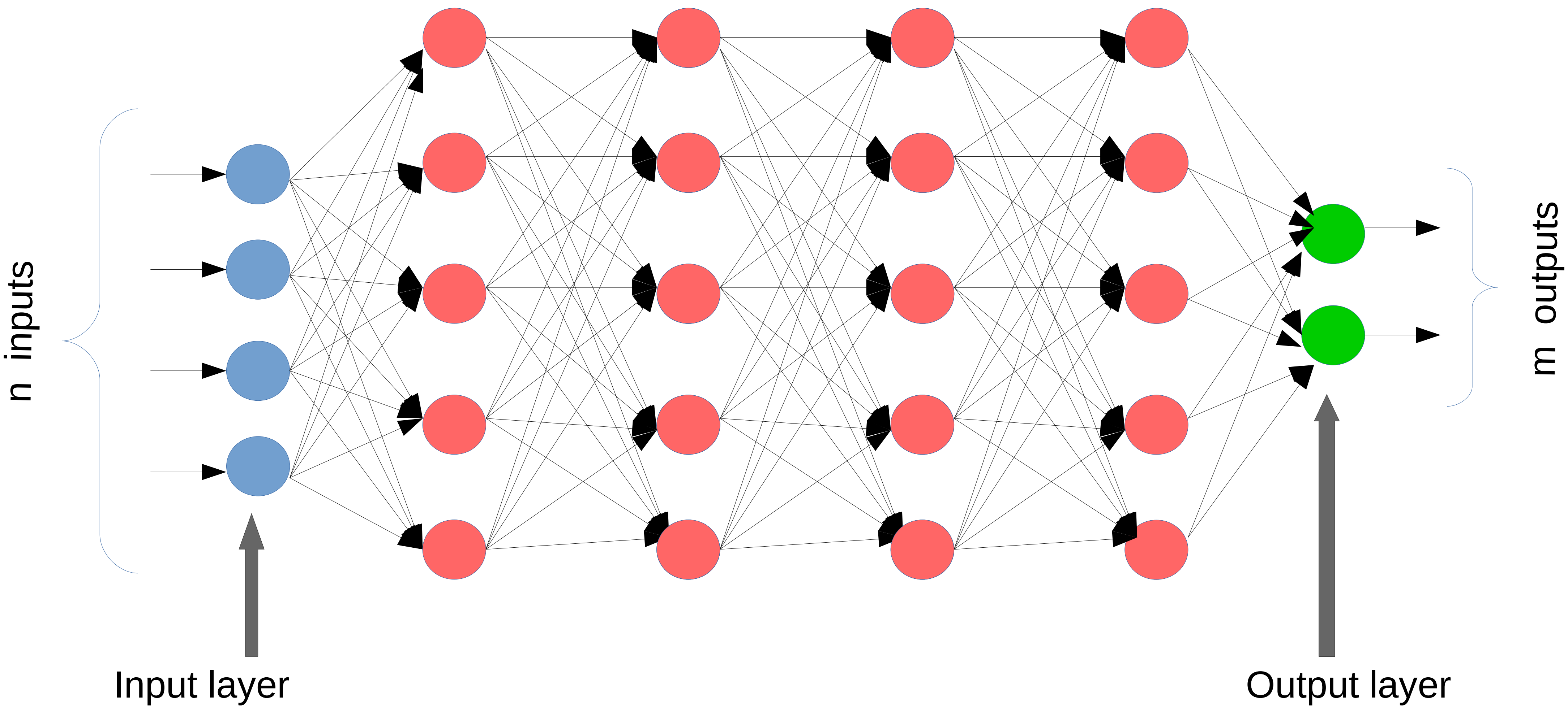
Specifying Properties of Neural Networks

Supratik Chakraborty

A Typical Neural Network



A Typical Neural Network



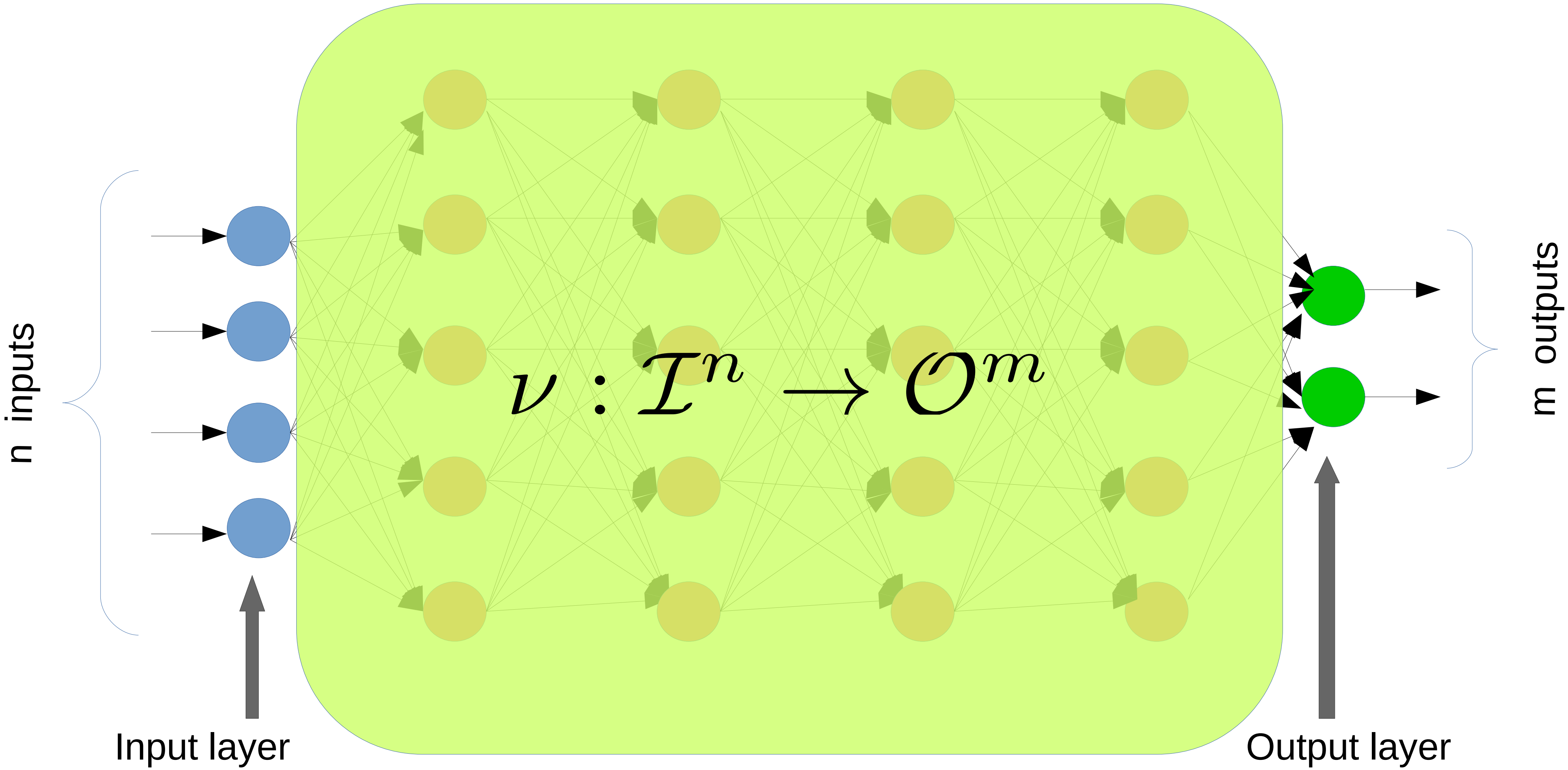
Input Domain: $\mathcal{I}$

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

First order logic formula

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

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

$y \leftarrow \nu(x);$

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

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

Hoare triples similar to those used in program verification

Semantics of Hoare Triple

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

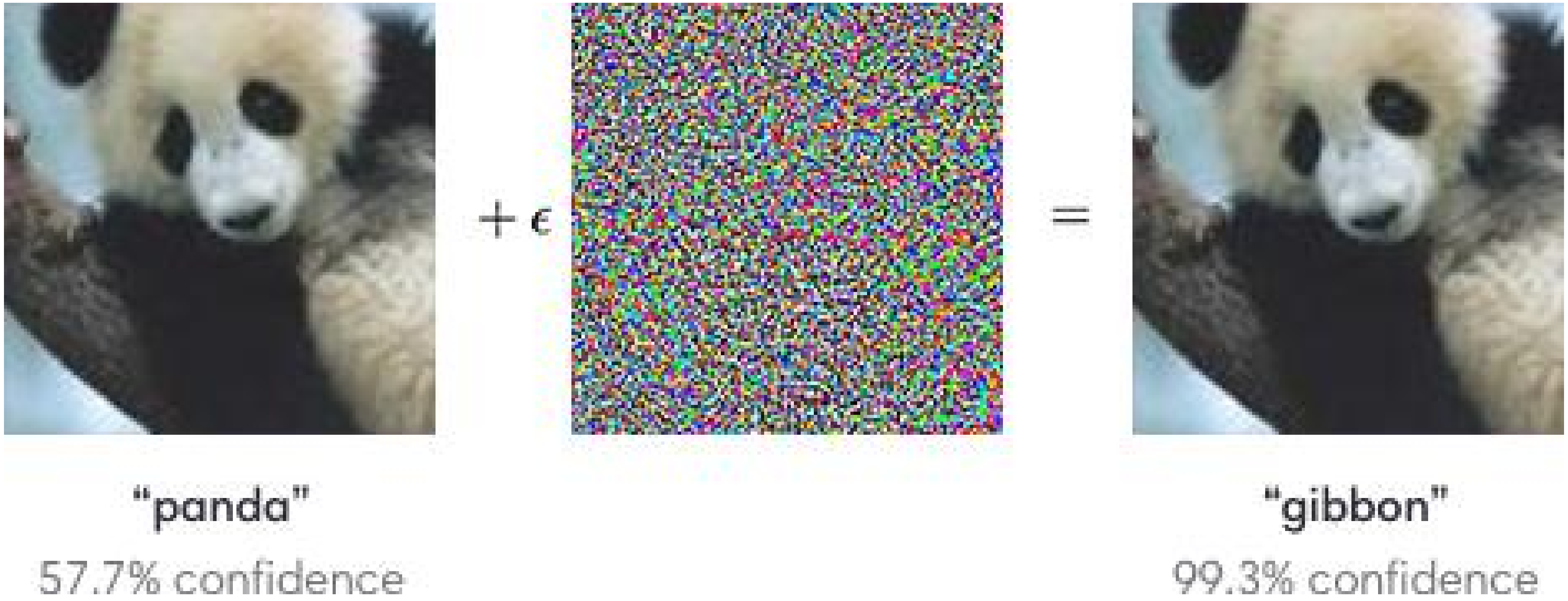
$$y \leftarrow \nu(x); \cdots \text{ ”Program”}$$

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

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

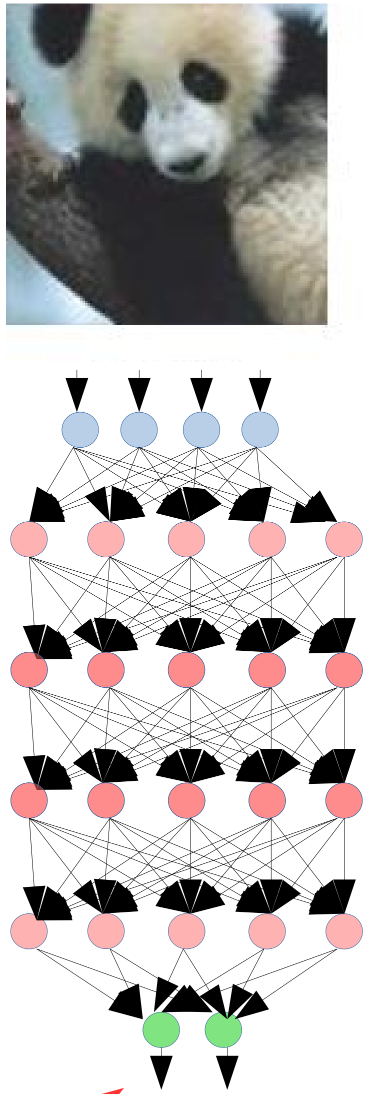


Source: Goodfellow, Shlens, Szegedy, "Explaining and Harnessing Adversarial Examples",2015

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

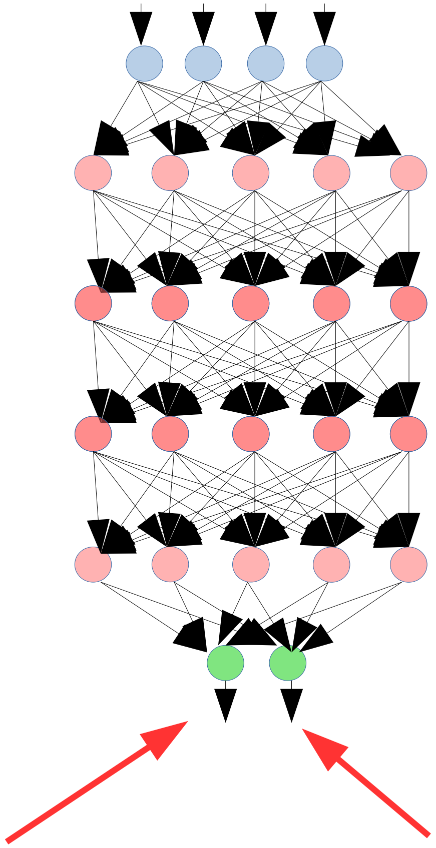
$$\{p > g + \delta\}$$

Score for panda: p

Score for something else: g

Property Specification Example 1

Specified image: x*



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

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

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

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

Score for panda: p

Score for something else: g

Spec as a logical requirement

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

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

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

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

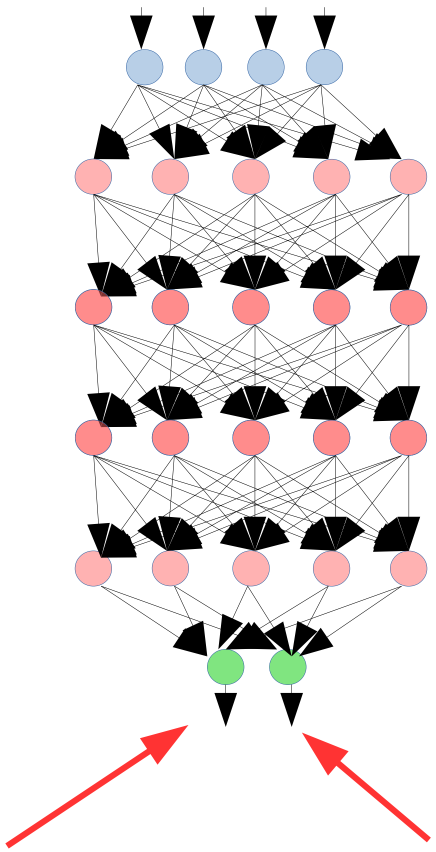
$$(p, g) \leftarrow \nu(x);$$

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

A logical implication

Property Specification Example 2

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



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

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

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

Property Specification Example 2

Pause n Reflect

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

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

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

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

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

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

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

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

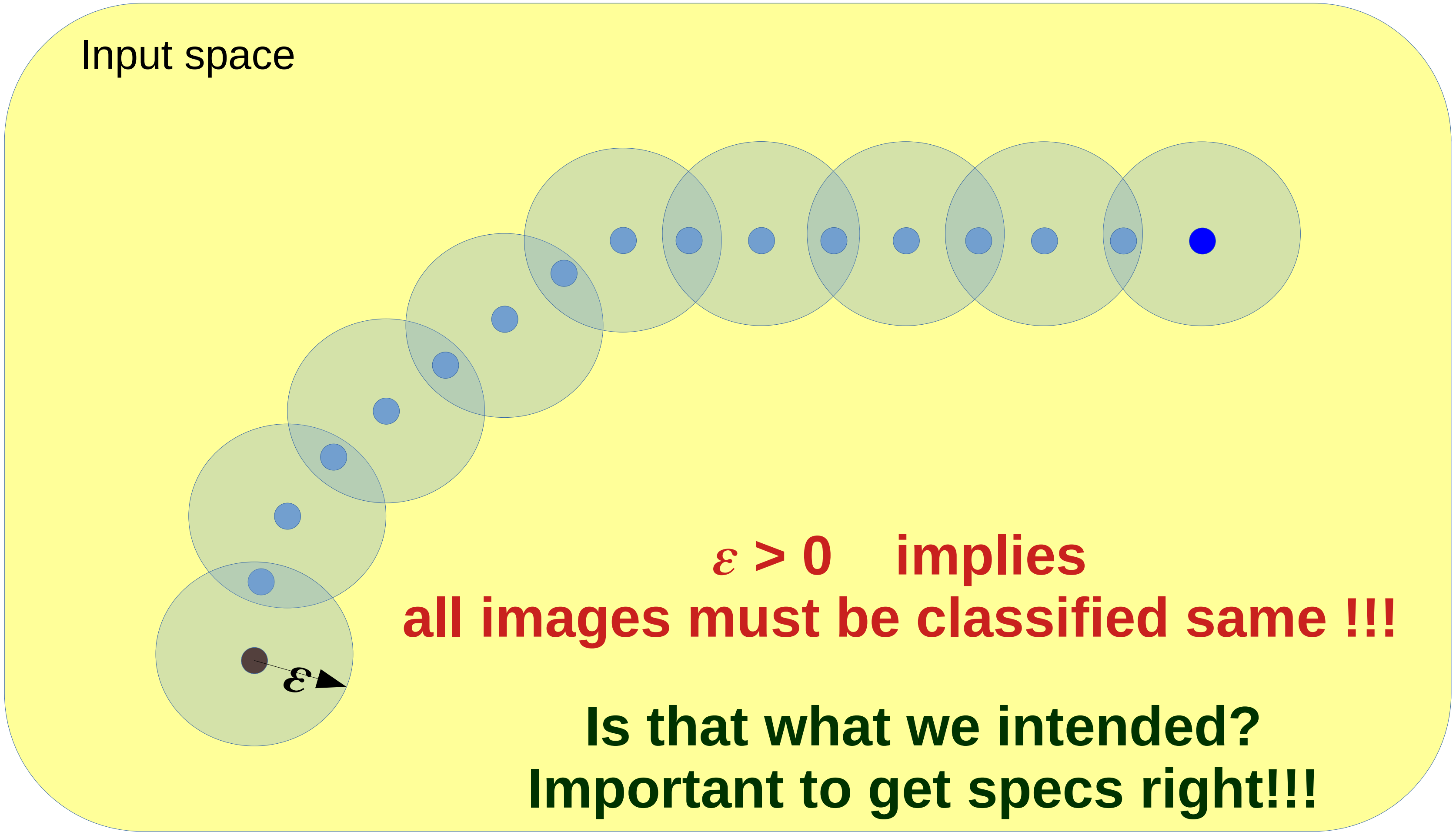
$$\bigwedge_{i=1}^{N} \left(|r_i - r_i^*| \leq \varepsilon_r\right) \wedge$$
$$\bigwedge_{i=1}^{N} \left(|g_i - g_i^*| \leq \varepsilon_g\right) \wedge$$
$$\bigwedge_{i=1}^{N} \left(|b_i - b_i^*| \leq \varepsilon_b\right)$$

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

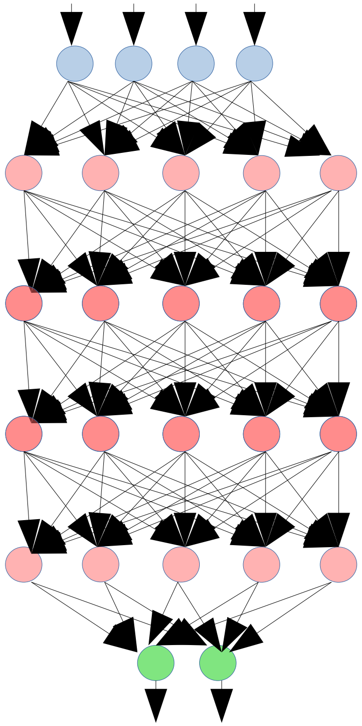
$$\left\{ \quad \right\}$$

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

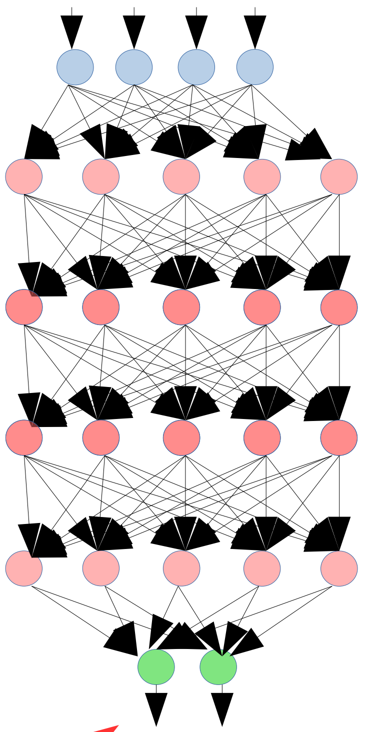
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

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

Attempting a Fix

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

Arbitrary image x



Score for class 1: s1

Score for class 2: s2

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

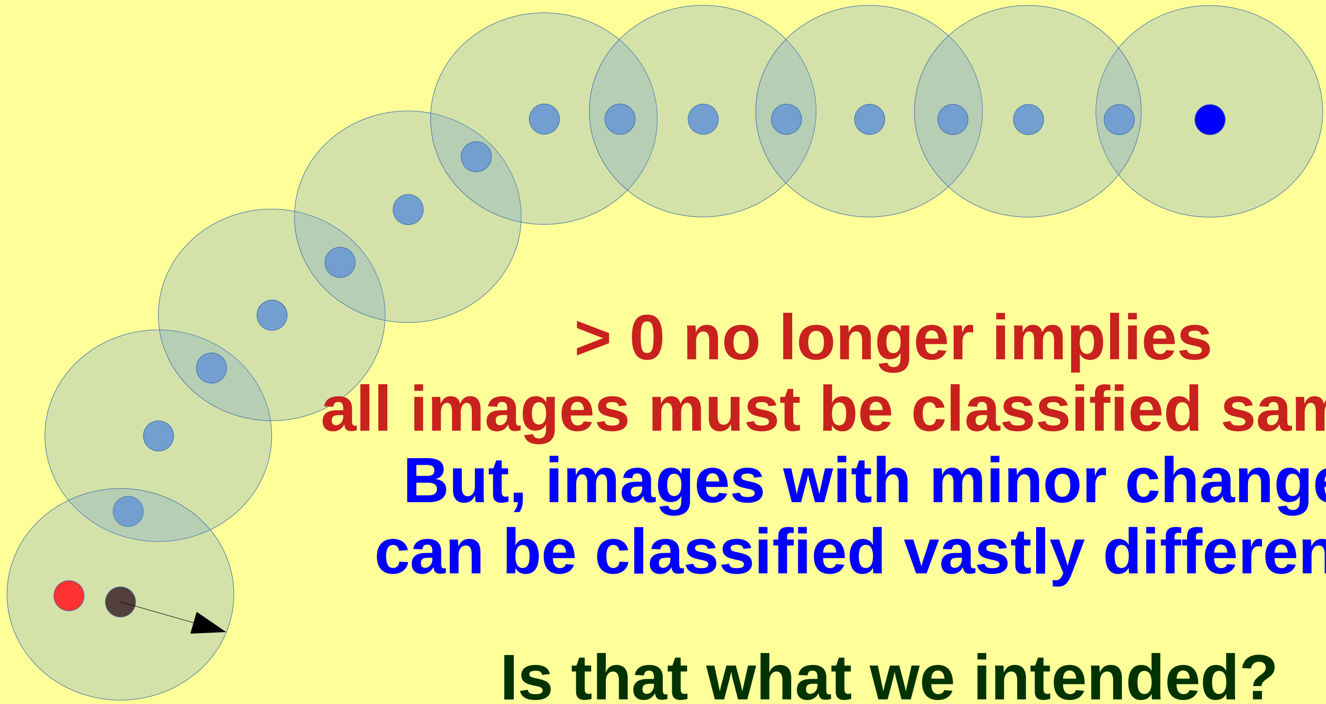
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

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

Did It Fix?

Pick any two arbitrary images in the input space

Input space



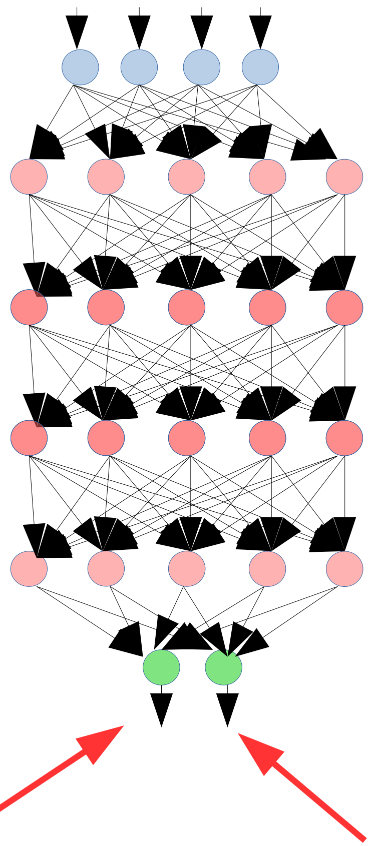
> 0 no longer implies
all images must be classified same !!!
But, images with minor changes
can be classified vastly differently

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



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

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

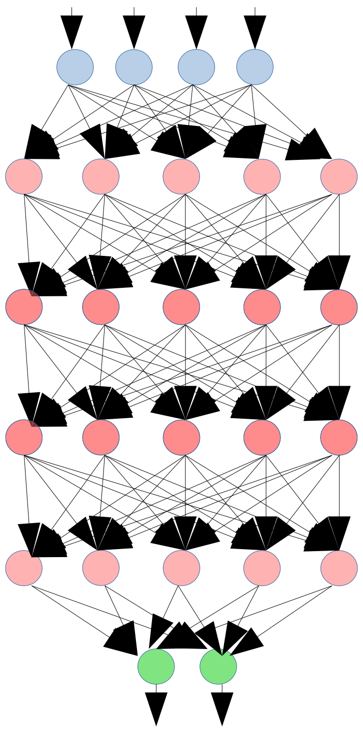
$$\left\{ \begin{array}{l} (s_1 > s_2 + \delta) \implies (s_1^* > s_2^* + \delta) \wedge \\ (s_2 > s_1 + \delta) \implies (s_2^* > s_1^* + \delta) \end{array} \right\}$$

Score for class 1: s1

Score for class 2: s2

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



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

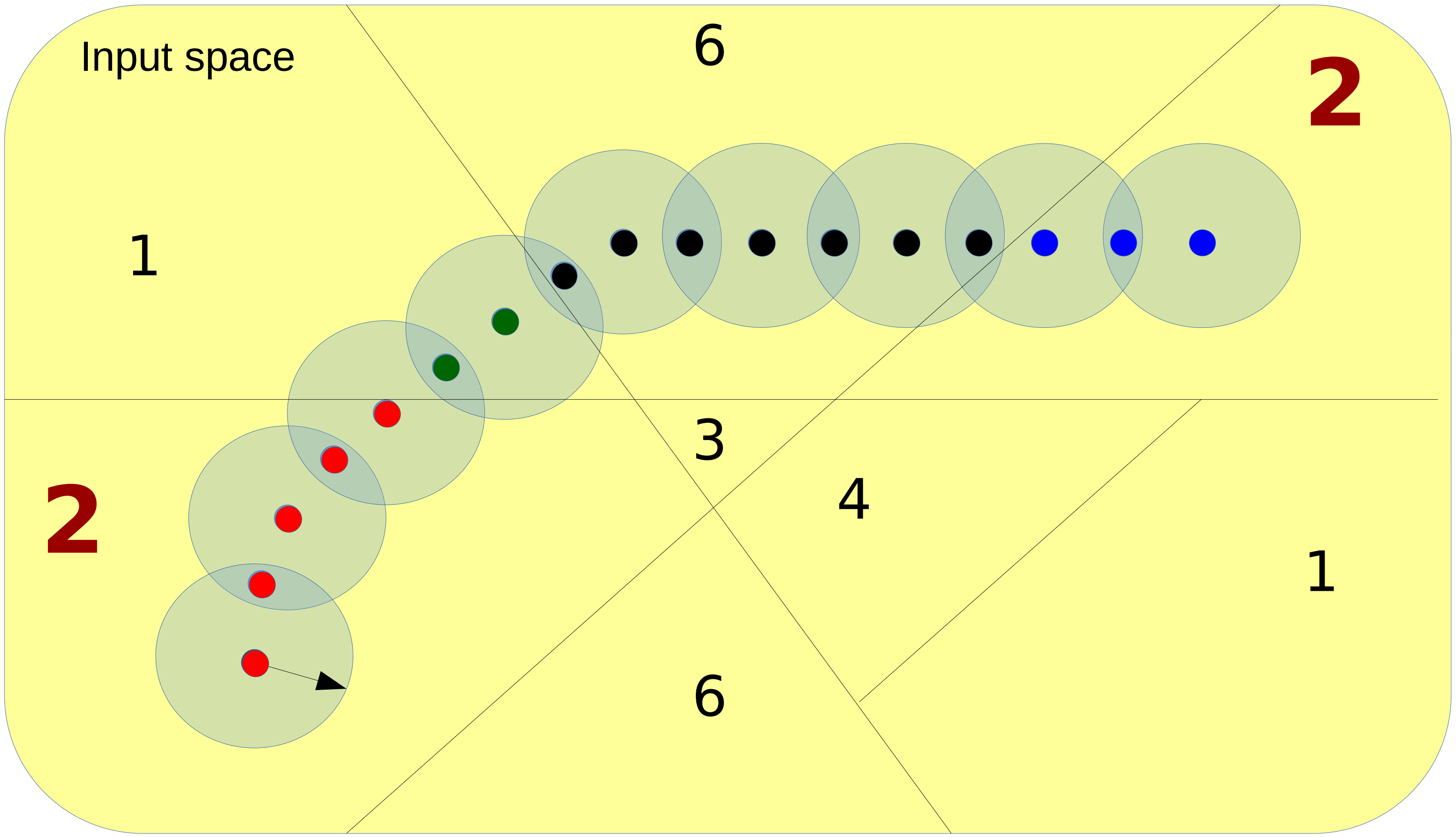
User-defined semantic features,
Not necessarily network-defined

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

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

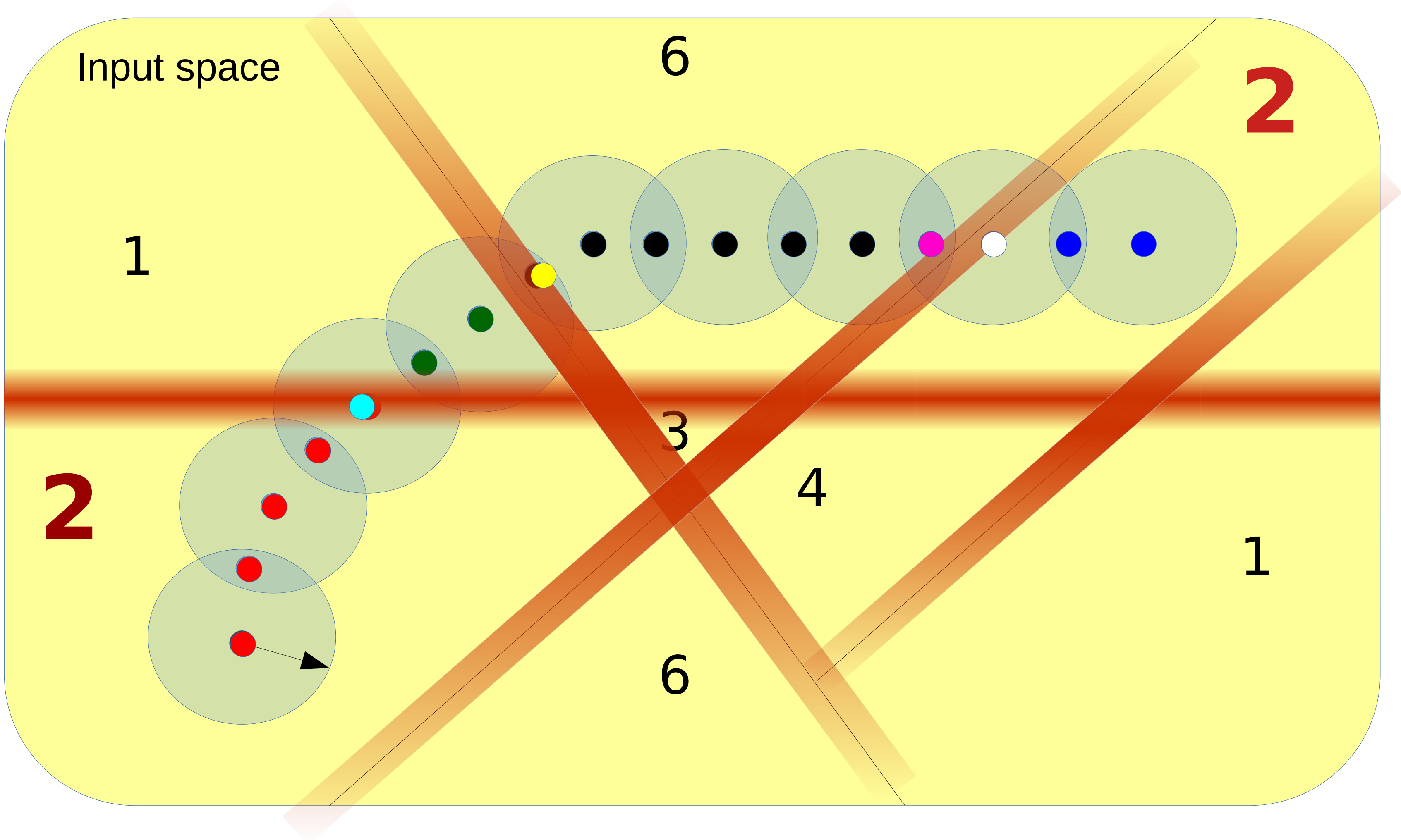
Possibilities with New Spec

Pick any two arbitrary images in the input space



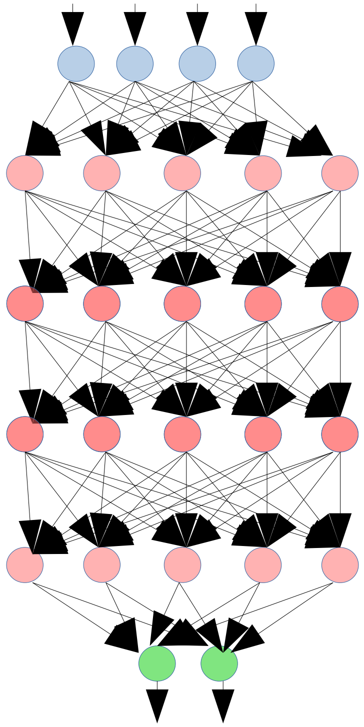
Possibilities with Newer Spec

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



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

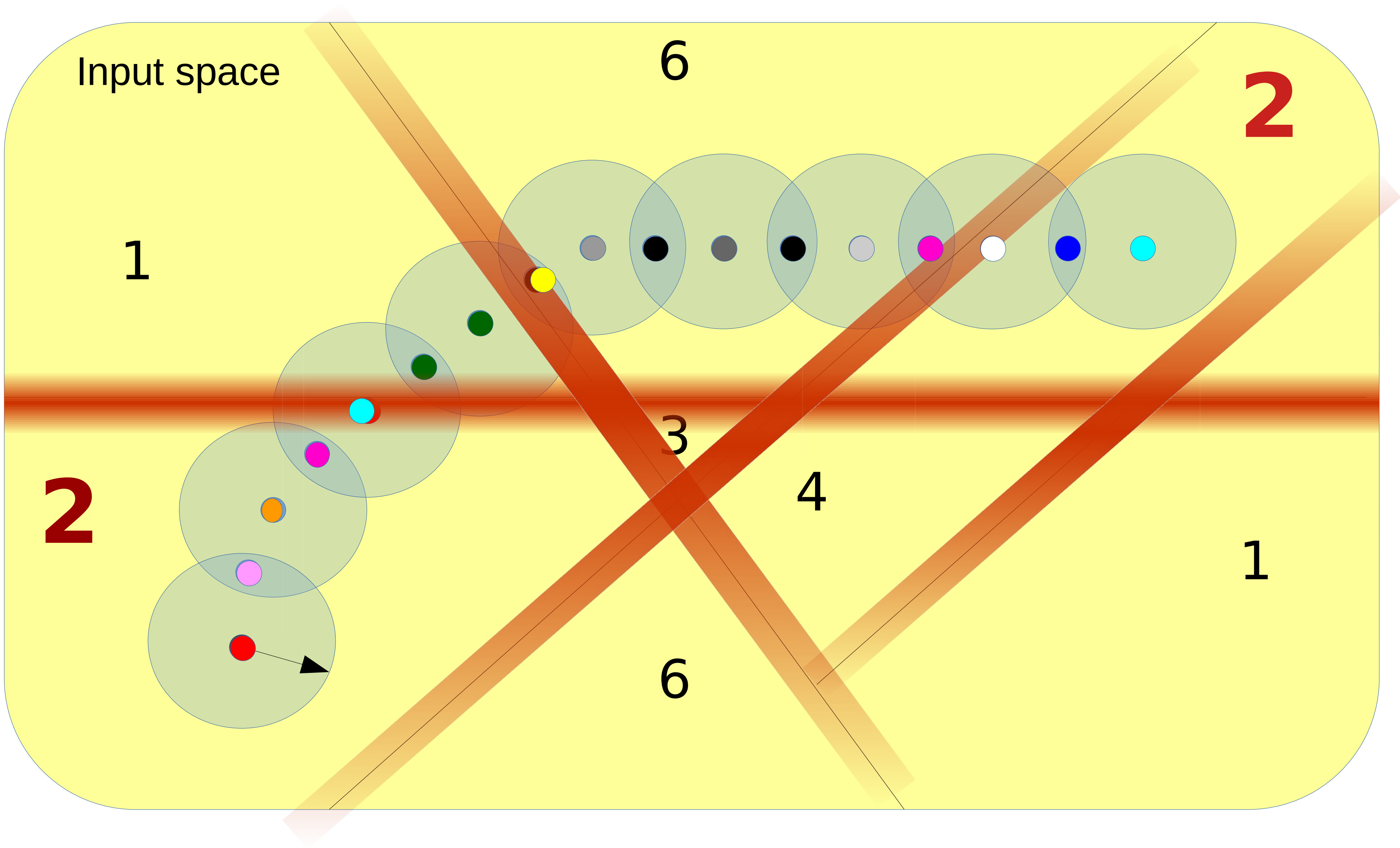
$$(s_1, s_2) \leftarrow \nu(x);$$
$$(s_1^*, s_2^*) \leftarrow \nu(x^*);$$

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

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?

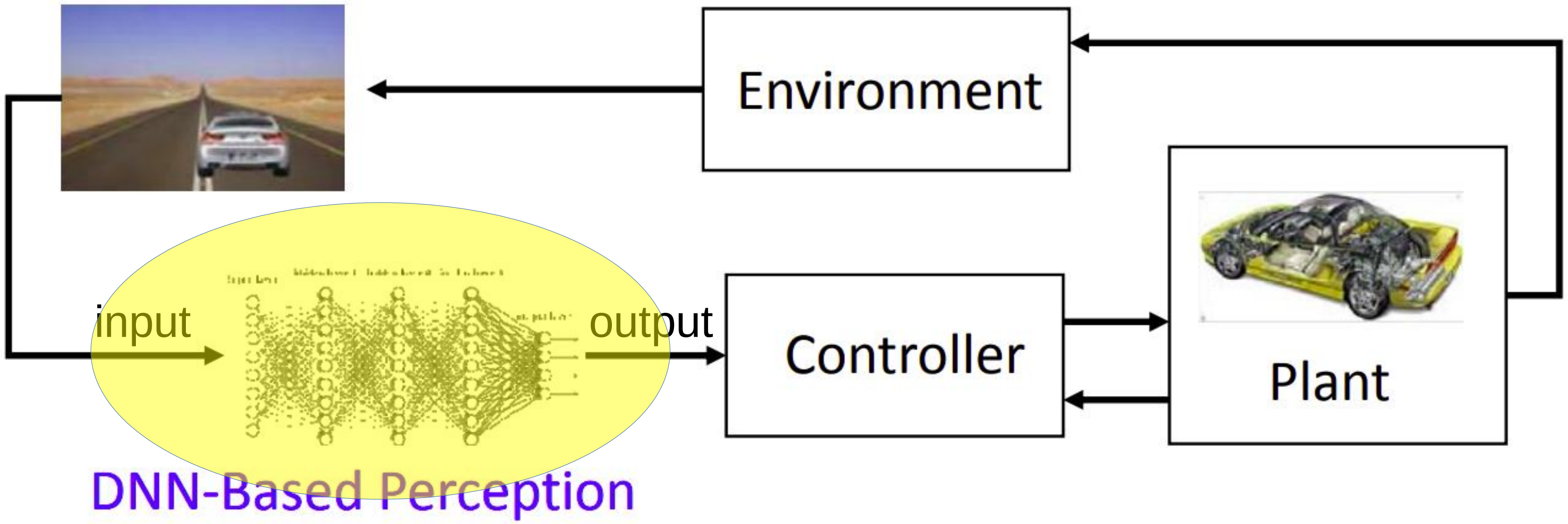
Is it easier to arrive at

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

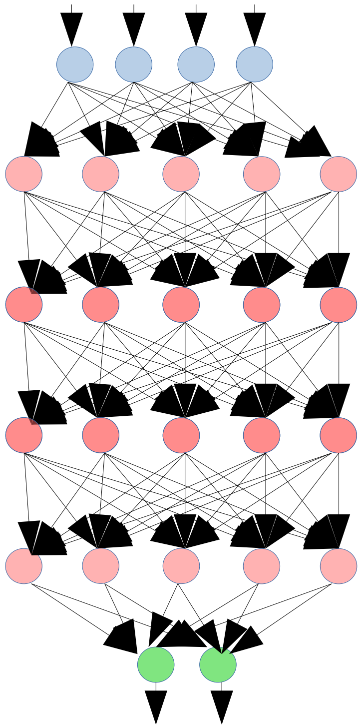
Is there a **formalizable relation** between inputs and desired outputs?

- 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)

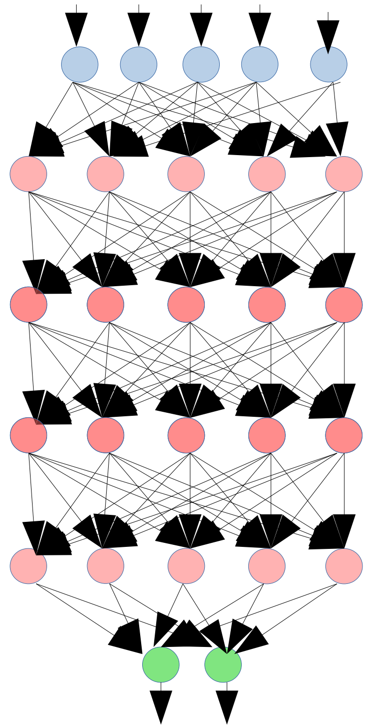


Perceptual Spec

“Too congested to accelerate”

Unmanned drone

Flight parameters

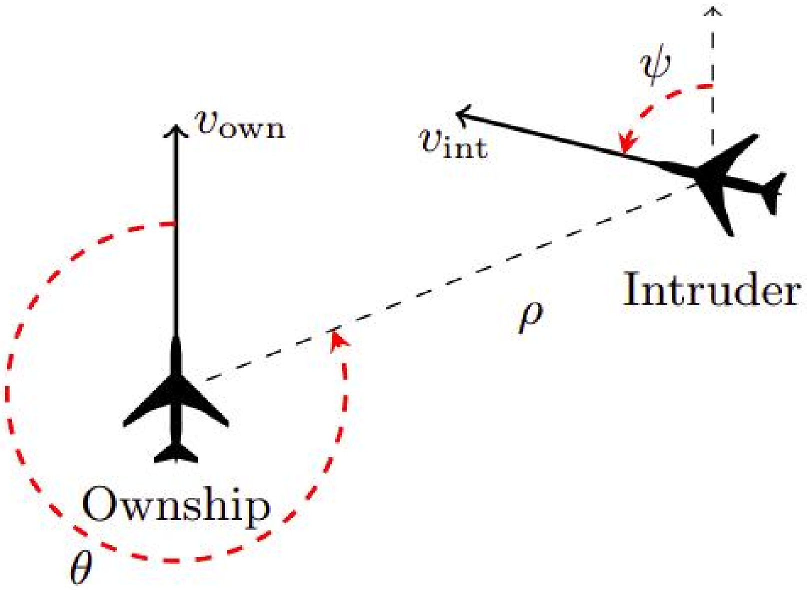


Non-Perceptual Spec

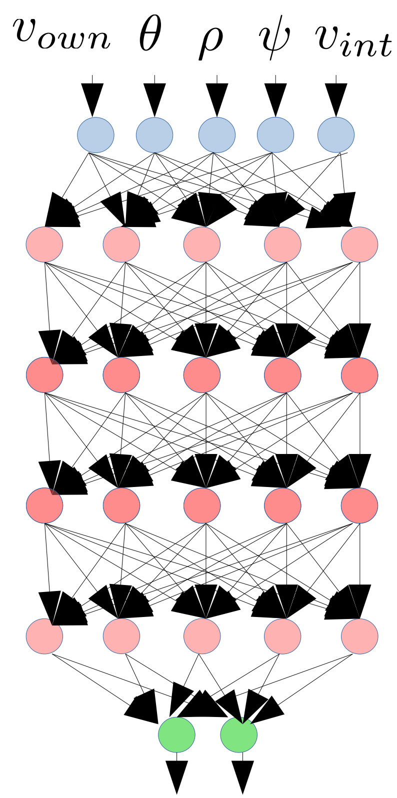
Score
(Horizontal Advisory)

Non-Perceptual DNN Specs

ACAS-Xu



Flight parameters



Score
(Horizontal Advisory)

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

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

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

Clear-of-Conflict

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

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

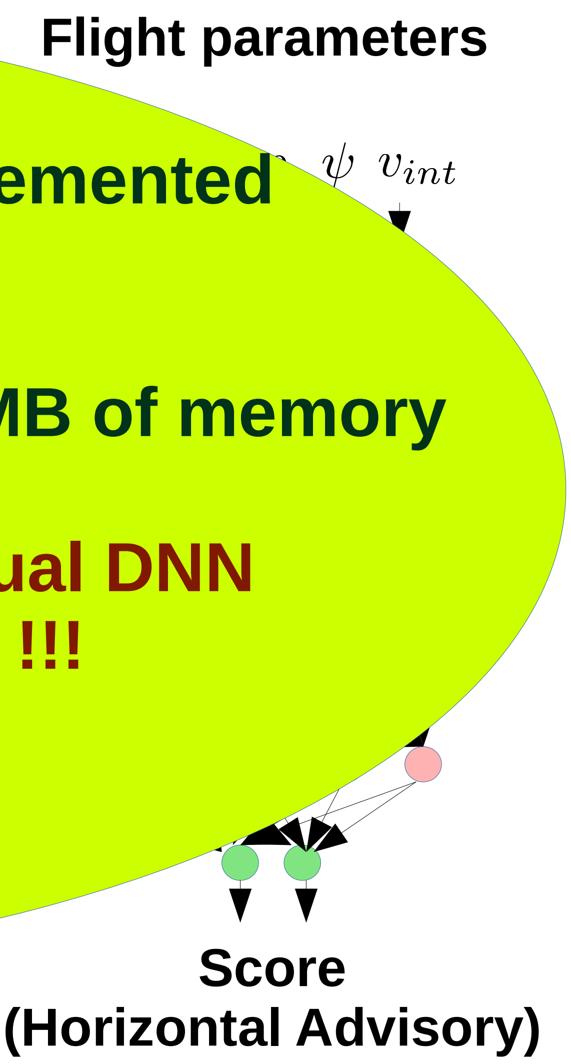
Having a good spec for a non-perceptual DNN doesn’t make the DNN irrelevant !!!

Specs NOT SAME AS Rules

$\{($

Sc

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

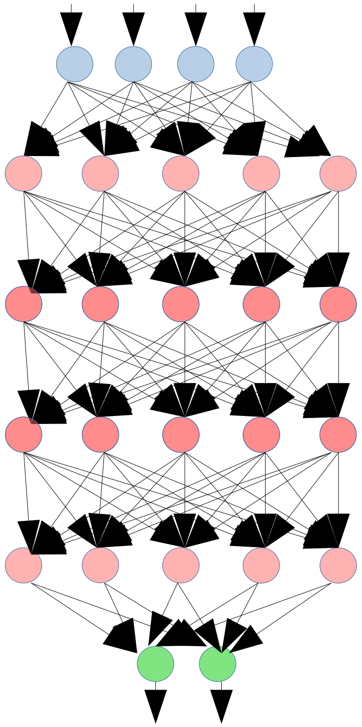


Source: Reluplex: An Efficient SMT Solver for Verifying Deep Neural Networks, by Katz et al, 2017

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” }

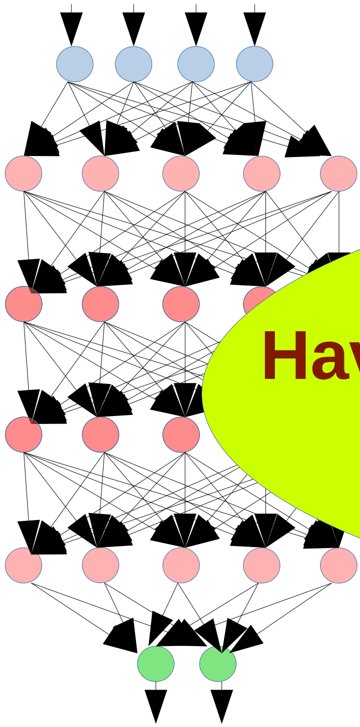
CR(...) = true iff road is “too congested to accelerate”

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

Perceptual DNN Specs

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

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~~ }

$g_N, b_N);$

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?

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