CS620 Mid-semester Exam (Spring 2021)

Max marks: 45 **Duration: 2 hours**

- *Be brief, complete and stick to what has been asked.*
- *Untidy presentation of answers, and random ramblings will be penalized by negative marks.*
- *Unless asked for explicitly, you may cite results/proofs covered in class without reproducing them.*
- ***If you need to make any assumptions, state them clearly.***
- ***Do not copy solutions from others. Penalty for offenders: FR grade.***
- **Expected time to solve:** ≤ 120 **mins.**
- **You will have an additional 30 mins to revise, scan your answer papers and upload on Moodle.**

1. Consider an image classification DNN $\mathcal{N}$ with an associated input-output transformer ν. Suppose the input domain, denoted **Im**, consists of 32 pixel $\times$32 pixel gray-scale images of animals, where each pixel is a real number (hence, each pixel can take infinitely many values). Suppose the output domain is the set of labels $\{\text{Cat}, \text{Dog}, \text{Other}\}$.

   Let $\Delta : \mathbf{Im} \times \mathbf{Im} \to \mathbb{R}^{\geq 0}$ be a carefully designed image similarity metric that maps a pair of input images to a non-negative real number, also called their *similarity score*. Let $\varepsilon > 0$ be a small positive real number, called *similarity threshold*. You are told that for every pair of input images $I_1, I_2 \in \mathbf{Im}$, the following hold: (i) $\Delta(I_1, I_2) = 0$ iff $I_1 = I_2$, and (ii) $\Delta(I_1, I_2) = \Delta(I_2, I_1)$. Additionally, for every finite set of images $\{I_1, I_2, \ldots I_k\} \subseteq \mathbf{Im}$, there exists at least one image $I_{k+1} \in \mathbf{Im}$ such that $\bigwedge_{j=1}^{k} (\Delta(I_{k+1}, I_j) > M)$ holds, where $M = \max_{i,j \in \{1,\ldots k\}} \Delta(I_i, I_j)$.

   (a) *[10 marks]* We want a network like $\mathcal{N}$ to not classify similar looking images differently. A convenient way of expressing this is via the Hoare triple HT_1:

   $$\{\Delta(I_1, I_2) < \varepsilon\} \quad \ell_1 = \nu(I_1); \ell_2 = \nu(I_2); \quad \{\ell_1 = \ell_2\}.$$

   However, we have seen in the lectures that HT_1 may (rather unexpectedly) require $\mathcal{N}$ to classify everything with the same label, rendering HT_1 meaningless.

   It turns out, however, that under certain conditions on the input image space, the triple HT_1 can indeed be meaningful and can express our intuitive ask, i.e. similar looking images should be classified the same.

   Give a first order logic sentence (i.e. formula with no free variables), say α, over elements of **Im** such that if **Im** satisfies α, then indeed HT_1 specifies the desired (or intended) property of $\mathcal{N}$. Give as weak a sentence α as you can, so that the restriction on **Im** is as mild as possible. Explain your answer clearly.

   (b) *[10 marks]* A student has written the following Hoare triple HT_2 for $\mathcal{N}$:

   $$\{\Delta(I_1, I_2) > \varepsilon\}\ \ell_1 = \nu(I_1); \ell_2 = \nu(I_2)\ \{\ell_1 \neq \ell_2\}$$

   Intuitively, the student wishes to express the property that if two input images are hugely dissimilar, then they should not be classified the same.

   Prove that if HT_2 holds, then for every pair of images $I_1, I_2 \in \Im$, we must have $\Delta(I_1, I_2) < \varepsilon$. In other words, the only way for HT_2 to hold is vacuously, i.e. the pre-condition itself is unsatisfiable.

2. A saturating ReLU is a function that behaves as follows:

$$y = \max(0, \min(z, c)),$$

where $c > 0$ is a given saturation value.

Consider the DNN shown in Fig. 1 below. Assume that each node in the hidden layer uses a saturating

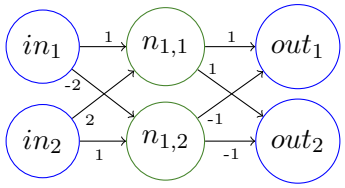

Figure 1: A simple DNN

ReLU with $c = 10$, and each node in the output layer uses a standard ReLU.

- *[7.5 marks]* Write the node constraint for $n_{1,2}$ in the above DNN using a boolean combination of linear constraints. You *must not* use max or min functions in writing the node constraint.
- *[7.5 marks]* Give the best over-approximation of the above constraint (for node $n_{1,2}$) that you can, as a conjunction of linear constraints (i.e. linear equalities and inequalities).
- *[10 marks]* A student wants to use Reluplex to prove properties of the DNN shown in Fig. 1. However, since Reluplex doesn't handle saturated ReLUs natively, it is not possible to use Reluplex directly.

  You are required to help the student by constructing a new DNN that mimics the input-output behaviour of the DNN shown in Fig. 1, but uses only ReLUs as the non-linear activation functions. This will allow the student to use Reluplex directly to reason about the network in Fig. 1. Effectively, you are required to mimic the behaviour of a saturated ReLU using one or more ReLUs, and appropriate interconnections between them. You may also use constant valued inputs of the new DNN and additional hidden layers/additional nodes, if needed.

  Draw the new DNN that uses only ReLUs (not saturated ReLUs) and clearly explain how the behaviour of each saturating ReLU in the original network is mimicked in the new network.