Exercise 05

Querying Neural Networks

Reliable and Interpretable Artificial Intelligence
ETH Zurich

Problem 1 (Coding)**.** This task can either be done in `task.py` or `task.ipynb` (whichever you prefer). In this exercise, you will implement the approach taught in the lecture for querying neural networks. You are provided with an image of a handwritten digit (`nine`), along with the ConvNet (`NN1`) and feed-forward network (`NN2`) for MNIST from the previous exercises. In the provided code, you can find several hints to solve this problem.

1. Implement the loss of the following constraint (similar to the one in slide 26). For now, ignore the box constraints:

```
find i[28, 28]
where i in [0, 1],
    i[0, 0:16,:] = nine[0, 0:16,:],
    class(NN1(i)) = 8,
    class(NN2(i)) = 9
```

   Run gradient descent to optimize the loss with `i` initialized to `nine`.

2. We now consider an alternative translation to the `class` keyword (slide 24). Replace the probability output vector with the logits (the inputs to the softmax layer), accessible through `NN1_logits` and `NN2_logits`. Run gradient descent to optimize the loss with `i` initialized to `nine`. Which translation is better?

3. Repeat tasks 1 and 2 with `i` initialized to the zero vector.

4. Add the box constraints to the loss function and repeats tasks 1 to 3.

5. L-BFGS-B is an optimizer which can explicitly handle box constraints on the input variables. Repeat task 2 with L-BFGS-B and add the box constraints.

6. In slide 17, we defined $d(a, b) = |a - b|$. We now define $d(a, b) = (a - b)^2$. Repeat the previous tasks for the new definition.

7. **Optional:** Replace the box constraint with the following box constraint:

```
i[0, 0:16, :] = nine[0, 0:16, :],
i[0, 16:29, 0:7] = nine[0, 16:29, 0:7],
i[0, 16:29, 14:29] = nine[0, 16:29, 14:29]
```

   Repeat the previous tasks.

Problem 2 (Translation)**.** In this question, we will show how to support negations.

1. Translate the constraint φ using the rules shown in the lecture.

```
φ = (i[0, 0] = nine[0, 0] ∧ i[1, 0] ≠ nine[1, 0]) ∨
(i[0, 0] ≤ nine[0, 0] ∧ i[1, 0] < nine[1, 0])
```

2. How can we transform negated formulas (e.g., $\neg\varphi$) to be in the fragment described in slide 17?

3. Translate the constraint $\neg\varphi$.

Problem 3 (Properties of the translation)**.** Let ϕ be a quantifier-free formula (as discussed in the lecture) with free variables $x_1, ..., x_n$. Prove or disprove the following claims:

1. For any assignment to the variables $x_1 \leftarrow y_1, \ldots, x_n \leftarrow y_n$, if $T(\phi)(y_1, \ldots, y_n) = 0$, then $y_1, ..., y_n$ satisfy ϕ.

2. For any assignment to the variables $x_1 \leftarrow y_1, \ldots, x_n \leftarrow y_n$, if $y_1, \ldots, y_n$ satisfy ϕ, then $T(\phi)(y_1, \ldots, y_n) = 0$.