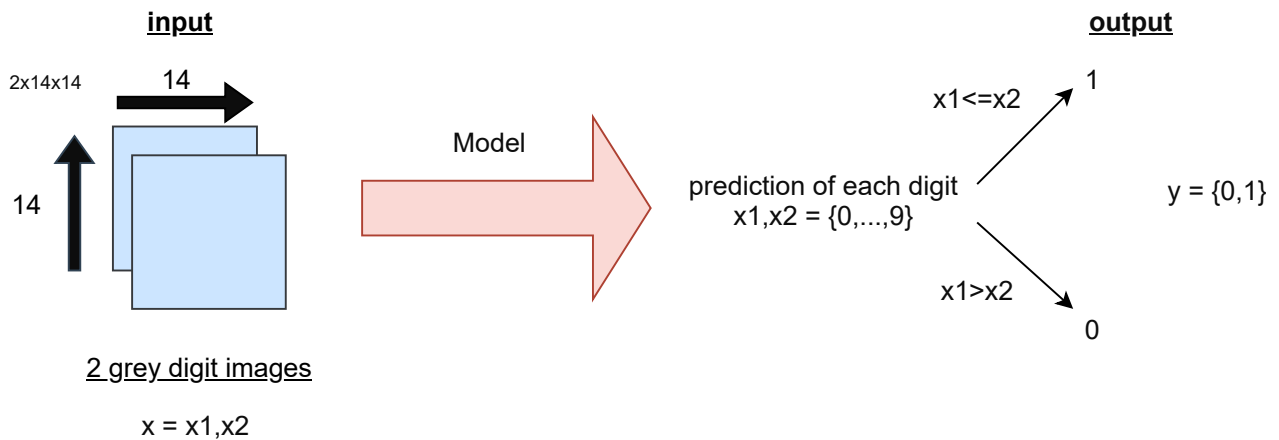


Only pytorch !



No numpy !
No sklearn !

Projet 1



Data

Name	Tensor dimension	Type	Content
train_input	$N \times 2 \times 14 \times 14$	float32	Images
train_target	N	int64	Class to predict $\in \{0, 1\}$
train_classes	$N \times 2$	int64	Classes of the two digits $\in \{0, \dots, 9\}$
test_input	$N \times 2 \times 14 \times 14$	float32	Images
test_target	N	int64	Class to predict $\in \{0, 1\}$
test_classes	$N \times 2$	int64	Classes of the two digits $\in \{0, \dots, 9\}$

We can generate the data sets to use with the function `generate_pair_sets(N)` defined in the file `dlc/practical/prologue.py`.

This function returns the six tensors:

Objective:

The goal of the project is to compare different **architectures**, and assess the performance improvement that can be achieved through **weight** sharing, or using **auxiliary losses**.

auxiliary losses ?
weight sharing ?
different architectures ?

The training can in particular take advantage of the availability of the classes of the two digits in each pair, beside the Boolean value truly of interest.

Methodology

All the experiments should be done with 1, 000 pairs for training and test.

Performance estimates provided in your report should be estimated through 10+ rounds for each architecture, where both data and weight initialization are randomized, and you should provide estimates of standard deviations.