EasyPyTorch - Virtual DSLs for editing of PyTorch code

Dmitry Savelyev, Shengzhe Xu, Frederick Vandermoeten

Overview

- **Intro**: PyTorch and DSLs
- Approach
- Timeline

Example: Existing Project

```
from dsl import db


def x(input, name):
    print("Hello world!")
    if True:
        while 2 < 1:
            change table lectures:
                replace <target> in column lecturer with <value>
                trim column name on both sides


change table students:
    replace Mister in column firstName with Mr.
    trim column secondName on right side

y = 42
y + 10

if y > 42:
    x = 3

```

```
from dsl import db


def x(input, name):
    print("Hello world!")
    if True:
        while 2 < 1:
            with db.change("lectures") as table:
                table.column("lecturer").replace("<target>", "<value>")
                table.column("name").trim("both")


with db.change("students") as table:
    table.column("firstName").replace("Mister", "Mr.")
    table.column("secondName").trim("right")

y = 42
y + 10

if y > 42:
    x = 3

```

PyTorch

- PyTorch is a popular deep learning framework
- Main features:
  - Tensor computing on the GPU
  - Neural Networks build on automatic differentiation
  - Handling of datasets
  - Training and evaluation of models

Domain Specific Language (DSL)

- Specialized language for a specific domain
- "virtual DSL": a DSL that is only present in the code editor

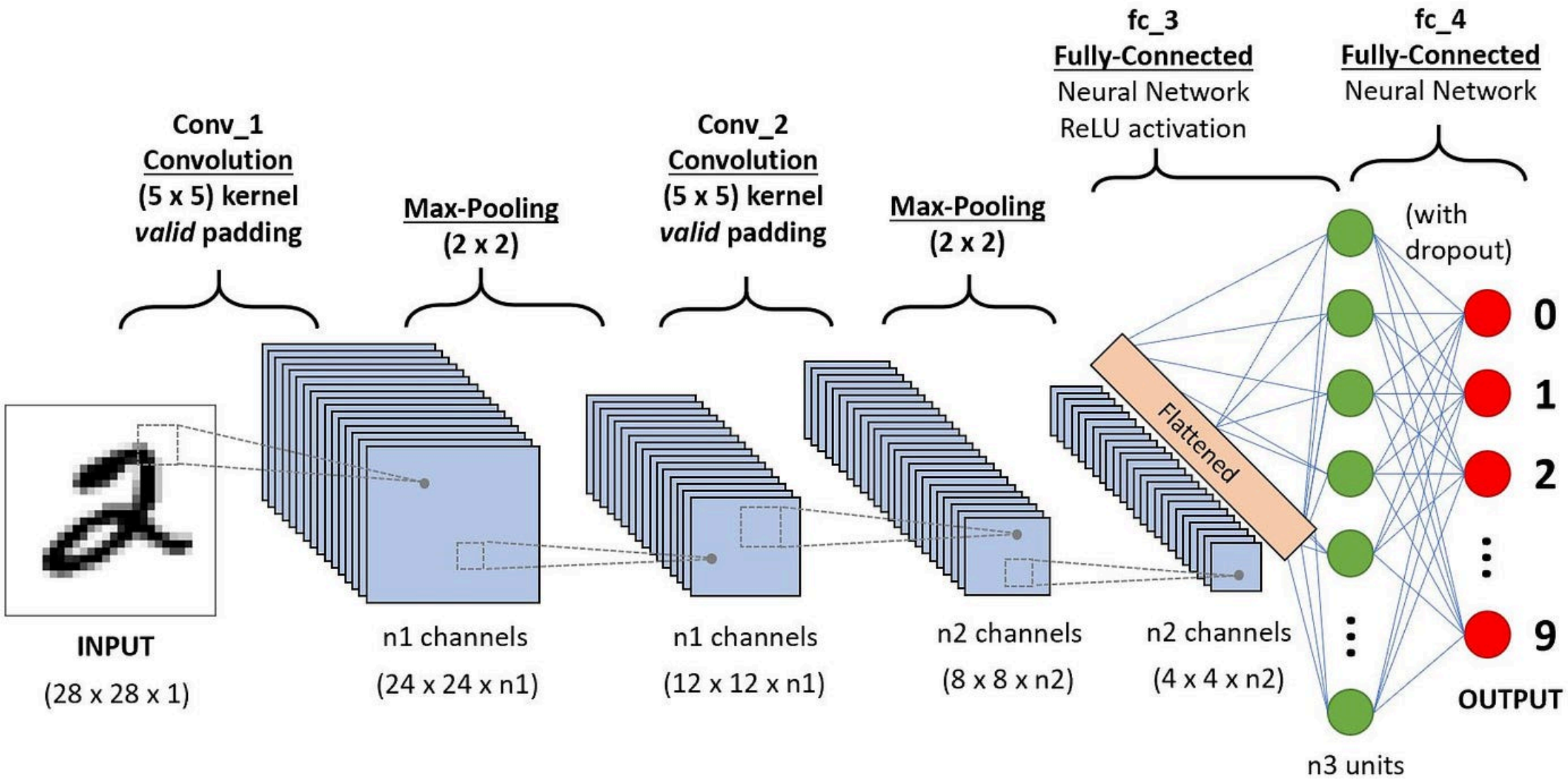
Goals

- **Simplify** the creation and modification of PyTorch code
- Make PyTorch code **easier to understand**

Possible DSL features

- Visual representation of:
  - the neural network
  - Tensor operations
  - Training runs, both setup and results (with e.g. TensorBoard)
  - datasets, e.g. showing the head of the dataset

Example: Neural Network



Prior Work

- vDSL Paper
- MSc Thesis
- PyTorch Lighting, fastai

Tools

- Svelte
- Jupyter, VSCode
- CodeMirror

Timeline

- **first two weeks**: get in touch with earlier groups, get to know the codebase + vscode extension
- week 3-5: analyze PyTorch code for useful widgets + some test widgets
- week 6-8: implementation of no-runtime widgets
- week 9-13: implementation of runtime widgets
- week 14-15: preparing demo and presentation
- last week of the semester (February): presentation + demo
- lecture-free time: finishing touches, finishing documentation
- final meeting & final presentation: early march

Thank you for your attention!

Any questions?