ONNX in EDDL

Lab 0: ECVL + EDDL environment

Winter School 24/01/2022

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Contents

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

What is ONNX?

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

What is ONNX?

Open Neural Network Exchange

- Open format to represent ML models
- Defined by a set of **Operators** to build computational graphs
- Uses **Protocol Buffers** as the mechanism to serialize the models

Key benefits

- Frameworks **interoperability**
- Inference hardware **optimizations**

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Operators support

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Operators support

EDDL coverage of the ONNX operators

We **don't support** the full ONNX operators set

All the main layers of the EDDL are supported. The **exceptions** are:

- **Data transformations:** Most of them are not in ONNX standard (Coverage list)
  - The ones supported are **Pad** and **Scale**
- **Data augmentation layers:** Not in ONNX standard (Coverage list)
- **Noise layers:** Not in ONNX standard (Coverage list)

Complete layers coverage list in eddl_progress.md

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

EDDL ONNX API

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

EDDL ONNX API

Functionalities of the ONNX module in EDDL

- Export/import your EDDL models
- Import pretrained models from our model Zoo
- Import models from other libraries*

*Remember that not all the operators are supported. Errors may appear

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

EDDL ONNX API

Export example

```
# Create your model
model = ...

# The model must be built to export it
eddl.build(model, ...)

eddl.save_net_to_onnx_file(model, 'my_model.onnx')
```

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

EDDL ONNX API

Import example

```python
model = eddl.import_net_from_onnx_file('my_model.onnx')
# or you can change the input layer of the model to fit your data
model = eddl.import_net_from_onnx_file('my_model.onnx', input_shape=(3, 512, 512))


eddl.build(model,
           o=eddl.adam(0.001),                  # Optimizer
           lo=['categorical_cross_entropy'],    # Losses
           me=['accuracy'],                     # Metrics
           cs=eddl.CS_GPU(),                    # Computing Service
           init_weights=False)                  # Avoid reinitializing the weights
```

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Model Zoo

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Model Zoo

Models included

One of the main advantages of ONNX in the EDDL is to import pretrained models for **transfer learning**

We currently support some popular topologies for image classification (pretrained with ImageNet):

- **VGG:** The 16 and 19 variants, with and without BatchNormalization
- **ResNet:** 18, 34, 50, 101 and 152 variants
- **DenseNet-121**

You can check them in the documentation here

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Model Zoo

Full model import example

```python
# Import the complete model (with the classifier)
feature_extractor = eddl.download_resnet18(top=False)
# or, you can also change the input shape of the imported model
feature_extractor = eddl.download_resnet18(top=False, input_shape=(3, 512, 512))

eddl.build(model,
           o=eddl.adam(0.001),                  # Optimizer
           lo=['categorical_cross_entropy'],    # Losses
           me=['accuracy'],                     # Metrics
           cs=eddl.CS_GPU(),                    # Computing Service
           init_weights=False)                  # Avoid reinitializing the weights
```

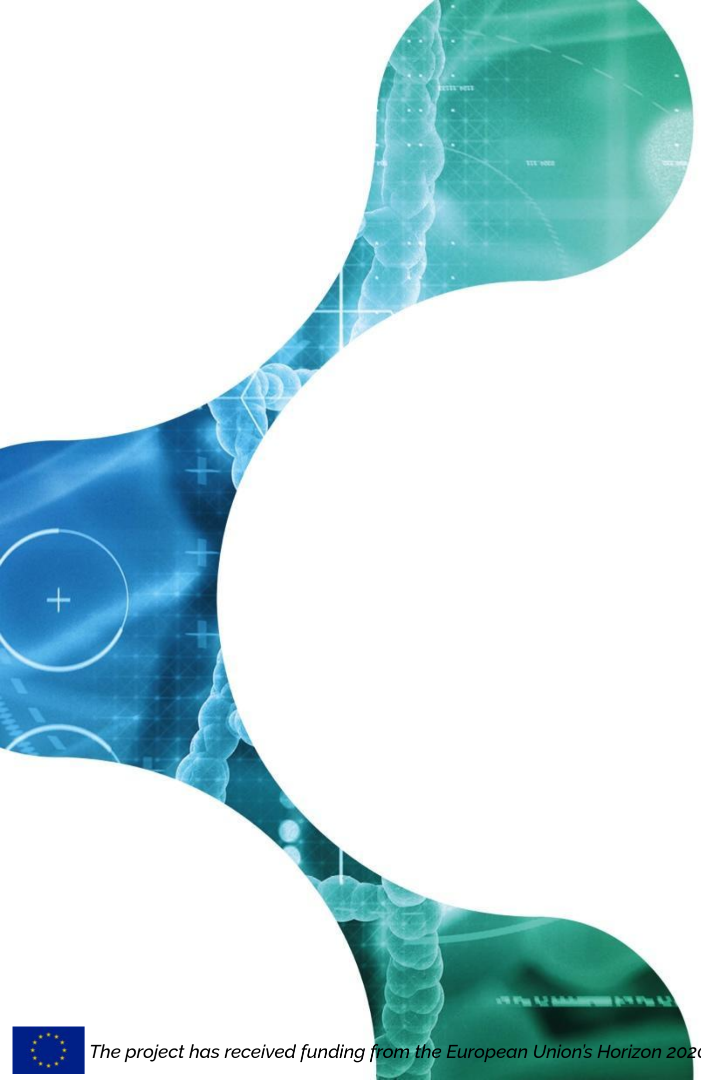
The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Model Zoo

Feature extractor import example

```python
# Import the model without the classifier
feature_extractor = eddl.download_resnet18()
# or, you can also change the input shape of the imported model
feature_extractor = eddl.download_resnet18(input_shape=(3, 512, 512))

# Get the last layer with the extracted features
top_layer = eddl.getLayer(feature_extractor, 'top')

# Add the classifier
dense0 = eddl.ReLu(eddl.Dense(top_layer, 512, name='dense0'))
out_ = eddl.Softmax(eddl.Dense(dense0, 10, name='dense1'))  # 10 output classes

# Get the input layer to build the new model
in_ = eddl.getLayer(feature_extractor, 'input')

# Create the final model
model = eddl.Model([in_], [out_])

eddl.build(model,
           o=eddl.adam(0.001),                  # Optimizer
           lo=['categorical_cross_entropy'],    # Losses
           me=['accuracy'],                     # Metrics
           cs=eddl.CS_GPU(),                    # Computing Service
           init_weights=False)                  # Avoid reinitializing the weights

# Initialize the new layers that are not pretrained
for l_name in ['dense0', 'dense1']:
    eddl.initializeLayer(model, l_name)
```

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.



Resources

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

Resources

Additional links with useful information

EDDL ONNX documentation

https://deephealthproject.github.io/eddl/model/onnx.html#

Netron

ONNX models visualizer (web app here)

ONNX Simplifier

Tool to avoid some importing errors (github here)

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.

DEEPHEALTH

Thank you!

Álvaro López Chilet

allochi@prhlt.upv.es

The project has received funding from the European Union's Horizon 2020 research and innovation programme under grant agreement No 825111.