



# Interface de génération de modèles de réseaux de neurones

Projet de programmation - Master 1

# Plan

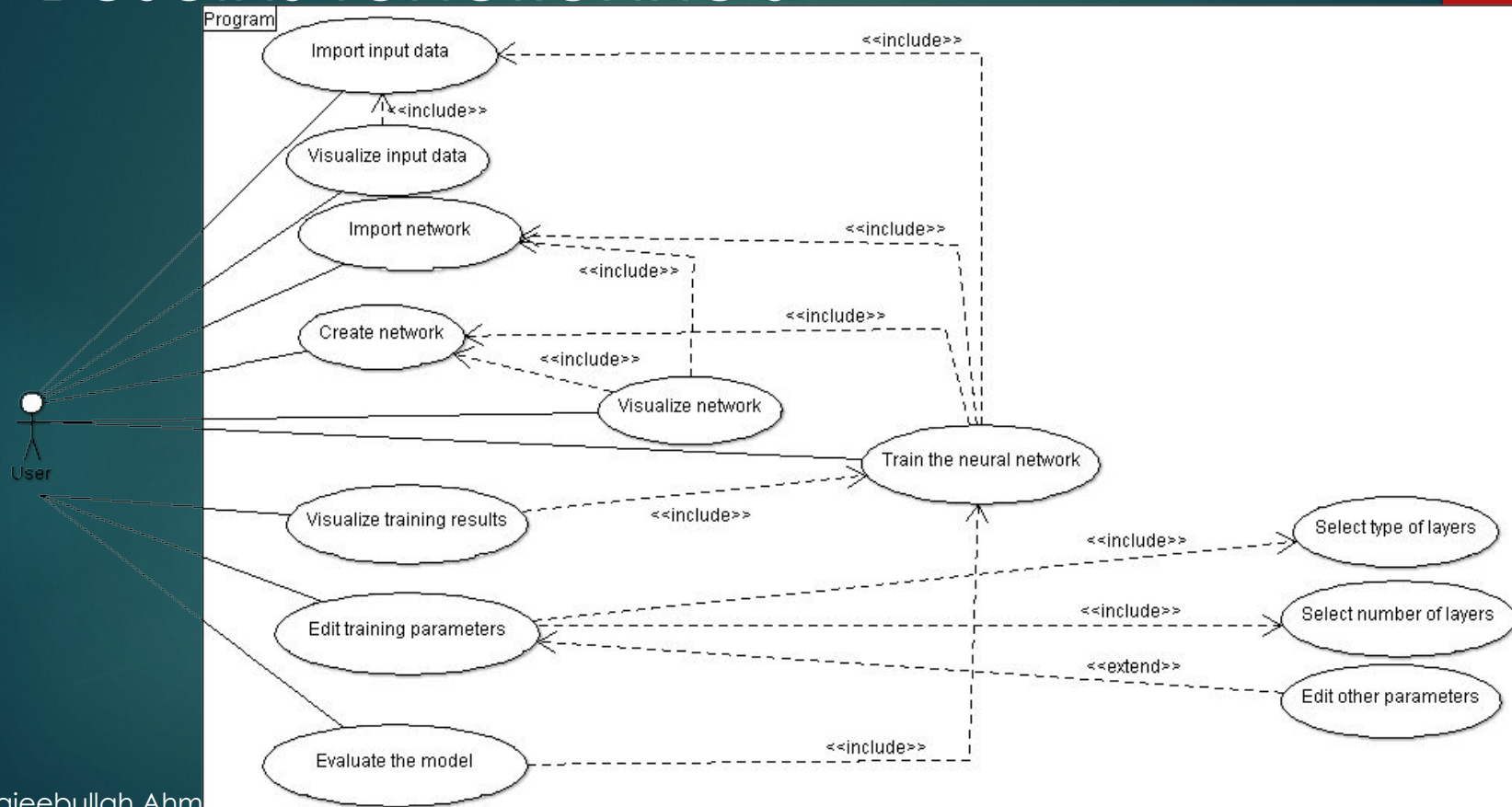
2 / 10

- ▶ Présentation
- ▶ Besoins fonctionnels
- ▶ Architecture du projet
- ▶ Première release
- ▶ Bilan

# Présentation du projet

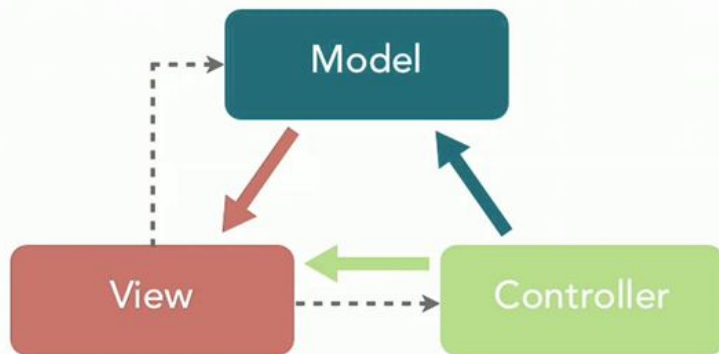
- ▶ Interface graphique pour PyTorch
- ▶ Existant : Caffe et Espresso
- ▶ Implémentation de différents modules
  - ▶ Création d'un réseau de neurones
  - ▶ Edition de modèles existants
  - ▶ Lancer les calculs
  - ▶ Création d'un format de sauvegarde des modèles
  - ▶ Visualisation du réseau et résultats obtenus

# Besoins fonctionnels



# Architecture

## the Model-View-Controller architecture pattern



### Model

contains business logic

### Controller

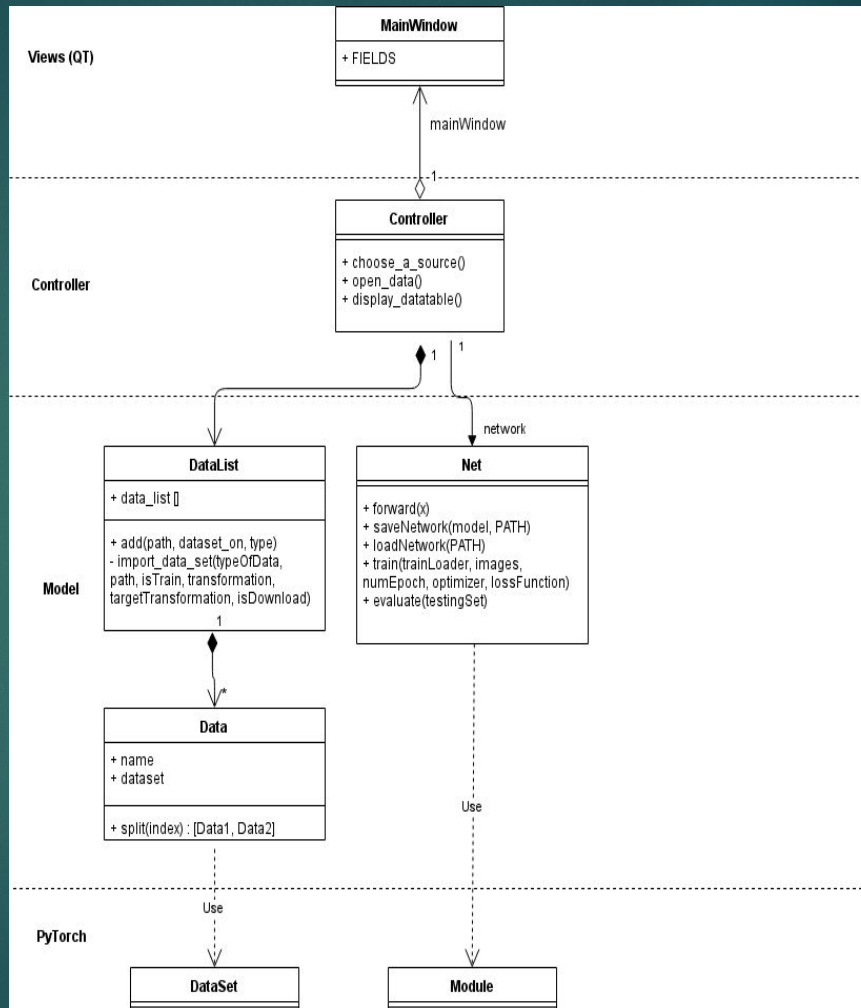
interacts with **Model** to create data for the **View**

### View

renders content to the user and relays user commands to the **Controller**

# Architecture

6 / 10





```

0%|          | 0/2 [00:00<?, ?it/s][1, 2000] loss: 2.132
[1, 4000] loss: 1.853
[1, 6000] loss: 1.676
[1, 8000] loss: 1.589
[1, 10000] loss: 1.517
[1, 12000] loss: 1.470
50%|██████    | 1/2 [02:27<02:27, 147.31s/it][2, 2000] loss: 1.399
[2, 4000] loss: 1.371
[2, 6000] loss: 1.345
[2, 8000] loss: 1.324
[2, 10000] loss: 1.310
[2, 12000] loss: 1.283
100%|██████████| 2/2 [04:47<00:00, 145.21s/it]
Finished Training
tensor([[ 1.3256e+00,  5.1789e+00, -1.2718e+00, -1.4851e+00, -3.0551e+00,
        -2.4034e+00, -5.4453e-01, -2.5971e+00,  1.0911e+00,  3.8840e+00],
        [-1.0477e+00, -1.9003e+00, -7.1758e-01, -2.0721e+00,  3.2378e+00,
        -1.3525e+00, -4.3941e+00,  9.1267e+00, -6.6105e+00,  8.6473e-02],
        [-5.5381e-01,  5.6087e-01, -7.3433e-01, -4.8403e-02, -1.8189e+00,
        -5.6753e-01, -2.1641e-02, -7.7669e-01, -7.5677e-01,  2.8661e+00],
        [-5.2328e-01,  5.8017e-03, -3.9163e-01,  8.7763e-01, -8.4791e-01,
         6.2978e-02,  1.0314e+00, -2.7497e-01, -1.2565e+00,  9.0932e-01]],
        grad_fn=<AddmmBackward>)
('Predicted: ', ' car horse truck frog')
Accuracy of the network on the 10000 test images: 55 %
0%|          | 0/10 [00:00<?, ?it/s]Accuracy of plane : 62 %
Accuracy of car : 66 %
Accuracy of bird : 17 %
Accuracy of cat : 29 %
Accuracy of deer : 52 %
Accuracy of dog : 51 %
Accuracy of frog : 72 %
Accuracy of horse : 72 %
Accuracy of ship : 63 %
Accuracy of truck : 63 %
100%|██████████| 10/10 [00:00<00:00, 31161.25it/s]

```

# Tests

8 / 10

- ▶ 1 seul test (import, entraînement)
- ▶ Unitaires
- ▶ Intégration



# Perspectives d'évolution

- ▶ Entrées plus nombreuses
- ▶ Visualisation des différents types d'entrées
- ▶ Paramètres plus nombreux
- ▶ Edition des paramètres de l'entraînement
- ▶ Visualisation graphique
- ▶ Tests

Merci de nous avoir écouté.