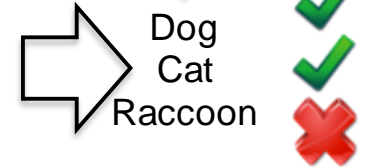
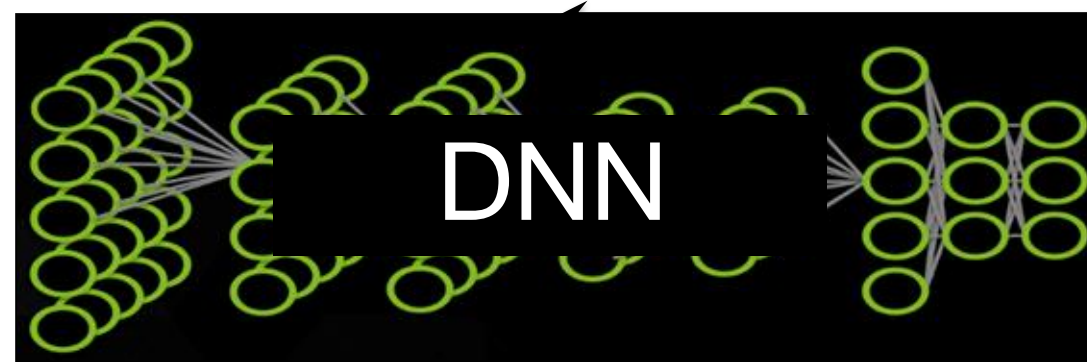
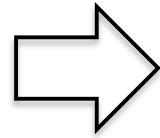


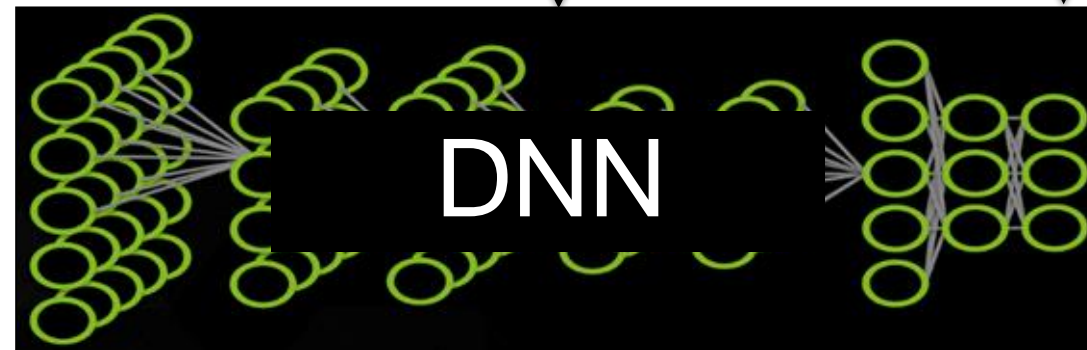
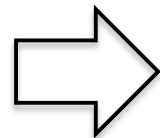
Deployment

Deep Learning Approach

Train:

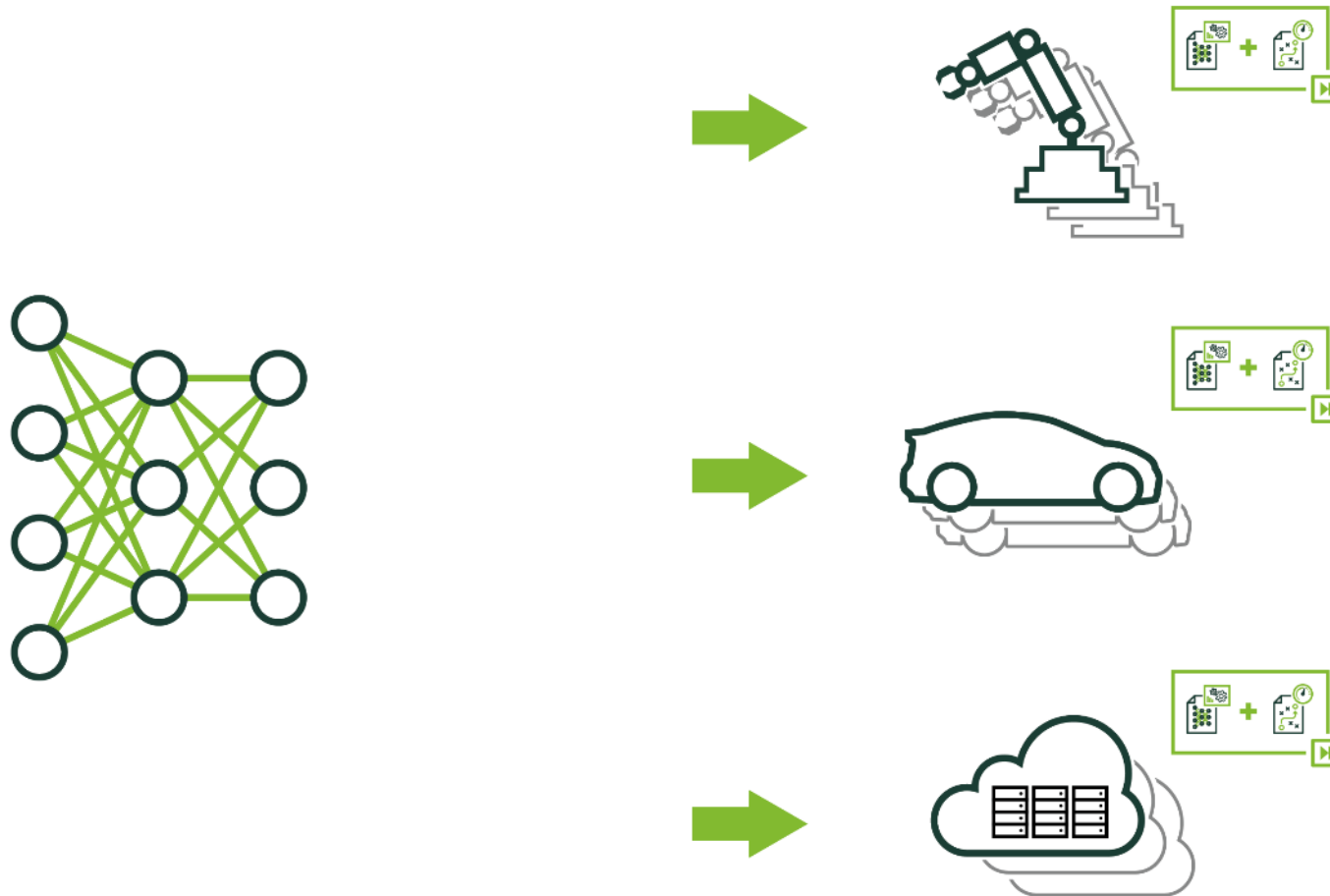


Deploy:



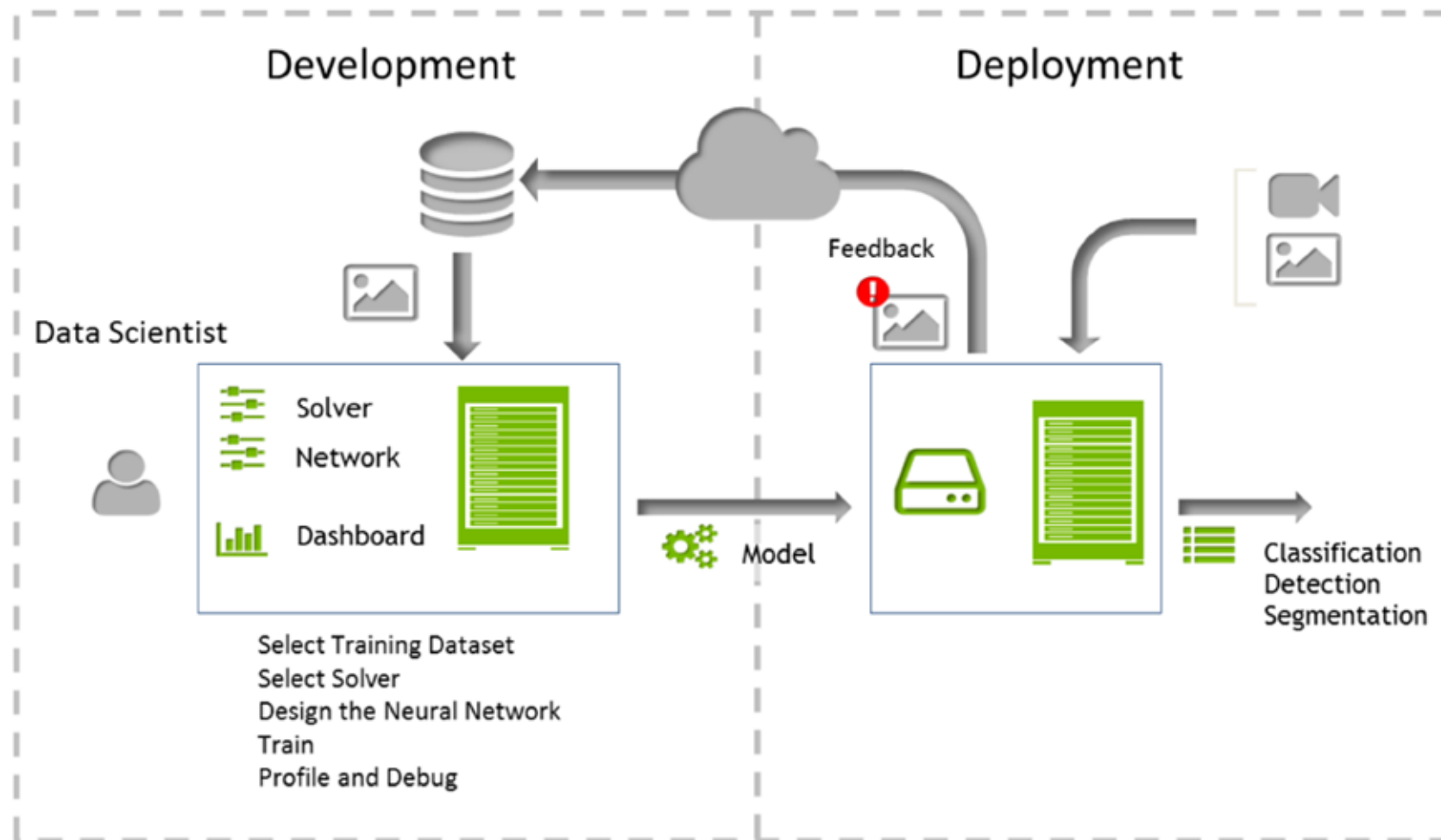
Deployment

How do I use a trained neural network as part of a solution?

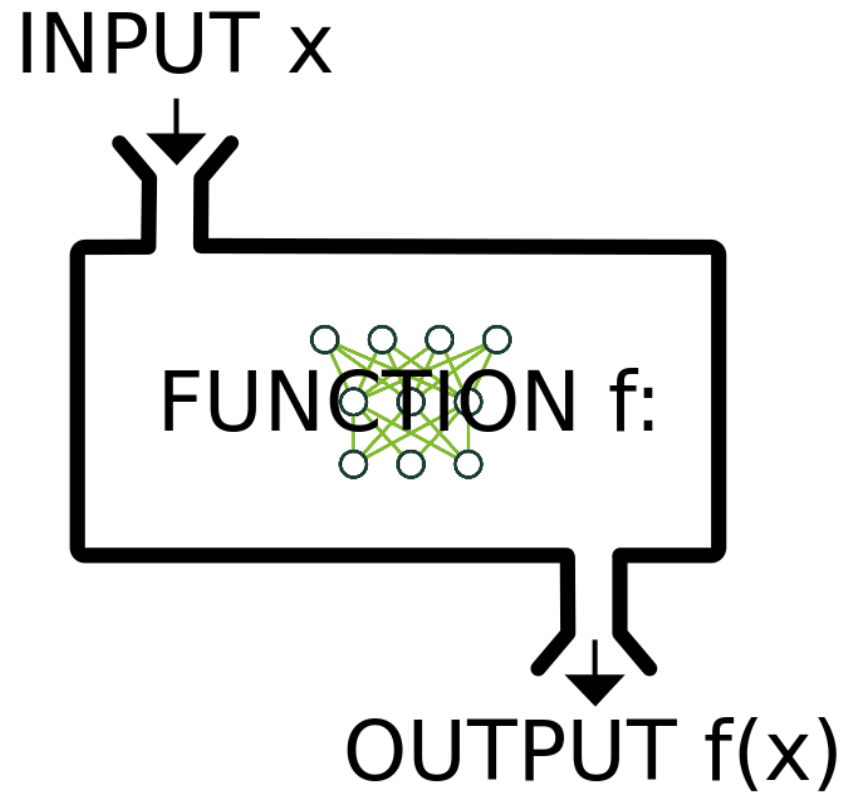


Deep Learning Approach

Neural network training and inference



Expected Inputs and Useful Outputs



Our current architecture

FRAMEWORK

We've been working in a framework called Caffe.

Each framework requires a different way (syntax) of describing architectures and hyperparameters.

Other frameworks include TensorFlow, MXNet, etc.

NETWORK

We've been working with a network called AlexNet.

Each network can be described and trained using ANY framework.

Different networks learn differently: different training rates, methods, etc. Think different learners.

TOOL - UI

We've been working with a UI called DIGITS

The community works to make model building and deployment easier.

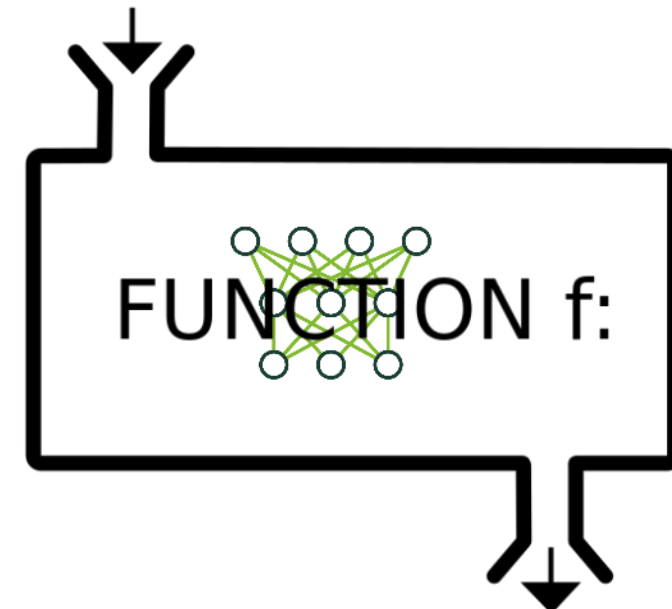
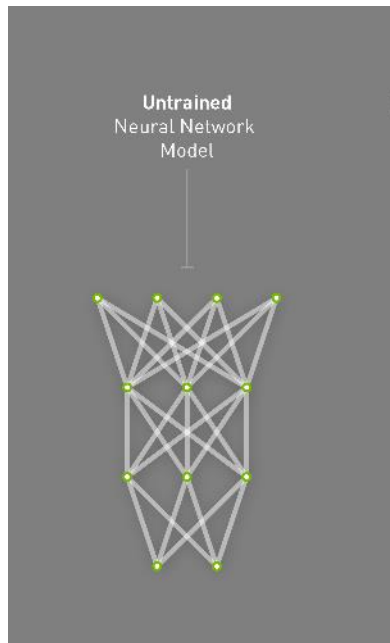
Other tools include Keras, Tensorboard, or APIs with common programming languages.

Components of a Model

Model Architecture = [deploy.prototxt](#)

Learned Weights = `***.caffemodel`

Model



Caffe files

- *.caffemodel: a binary file containing the weights for the model at the iteration it was saved
- *.prototxt: a text file describing the network model and its layers
- image_mean.binaryproto: the image mean of the dataset, the model requires this to be subtracted from each image before classifying

Deploying Our Model: GPU Task 3

Deploying our Model: GPU Task 3

[VIEW UNIT IN STUDIO](#)

[Bookmark this page](#)



Select **Start** to launch our GPU task.