Collaborative Development of Deep Neural Networks for Particle Imaging Detectors

In traditional image analysis with neural networks, first find features (x) and then train to associate them with some output (y). Features examples: vertices, tracks, CNNs made of successive **convolutional layers** that identify location of specific pattern. Network trained to output vector for certain image types by learning the convolutional layer patterns on its own.

**CNN for Image Classification**

**Goal**: Categorize NxM pixel image

**How**: Apply successive convolution & down-sampling operations to transform data into 1D discriminants

**Use the output of deep layer** to estimate the object location in the original image resolution.

**Assumption**: feature maps encode information to deduce higher precision.

**Runs faster** than a traditional object detection algorithm by making a prediction from the reduced image resolution.

**Multi-task Network**: the whole image classification + local feature detection.

**CNN for Pixel-Level Analysis (Segmentation)**

**Pixel Segmentation**: Analyze features at the pixel-level.

**How**: First, extract high-level features by successive convolution & down-sampling operations. **Second**, extrapolate back to a higher resolution using convolution & up-sampling.

**Intermediate, low-resolution**

**Feature tensor**

**Feature extraction** "human face"

**Feature extraction** "written text"

**CNN for Pixel-Level Analysis (Segmentation)**

**Softwares**: Many good open-source deep learning (DL) softwares developed in tech. industries. LArCV is a bridge between your experiment & those softwares.

**LArCV**: Our software provides event processing framework with file format & I/O, data structure designed for 2D/3D information, multi-threaded fast data-read for network training, and extensive Python APIs to interface DL softwares.

**Open Data Sample**: Data is the heart of modern machine learning. We provide open physics simulation samples for collaborative development work.