Experimental training of a convolutional network in PyTorch

A neural network was implemented that closely follows one proposed in "The analysis of VERITAS muon images using convolutional neural networks" by Feng, Lin.

The network has three convolutional layers with filter size 3*3 interspersed with pooling layers and followed by two fully connected layers with 256 and 64 neurons, respectively. The last is an output layer of two neurons, estimating the chances of the event being caused by a gamma quantum or a proton. For the training purposes, dropout was applied after each convolutional layer (with p=0.25) and each fully connected layer (with p=0.5).

All neurons except the output have rectifier activations, and the output neurons have sigmoid activation.

Quality of recognition

Let's suppose $E = \Gamma + P$ is the number of events where Γ is a number of gamma events and P is a number of proton events, and let's suppose the network N correctly recognizes N_Γ of the gamma and N_P of the proton events, respectively, and let $N'_\Gamma = \Gamma - N_\Gamma$, $N'_P = P - N_P$. Then we define the quality of recognition as

$$Q(N) = \frac{N_\Gamma/\Gamma}{\sqrt{N'_P/P}}$$

if $N_\Gamma \geqslant N'_\Gamma$, and 0 otherwise.

Results

The network was trained using a set of some 12000 events, with a cross-validation set of over 5000 events. After about 4 days of training, the best quality on cross-validation set was 2.883, with $N_{\Gamma} / \Gamma \sim 0.567$ and $N'_{P} / P \sim 0.0387$.

The best result so far was achieved when training the network without applying dropout, with quality of recognition ~3.493.