Supplementary Material

MLP Binary Connect Architecture

Dropout p = 0.2
Fully Connected Layer (units = 2048, bias = False)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)

Dropout p = 0.2
Fully Connected Layer (units = 2048, bias = False)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)

Dropout p = 0.2
Fully Connected Layer (units = 2048, bias = False)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)

Dropout p = 0.2
Fully Connected Layer (units = 2048, bias = False)
Batch Normalization Layer (gain = 1, bias = 0)
Softmax

VGG Binary Connect Architecture

Convolutional Layer (channels = 128, kernel-size = 3×3, bias = False, padding = same)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)
Convolutional Layer (channels = 128, kernel-size = 3×3, bias = False, padding = same)
ReLU
Max Pooling Layer (size = 2×2, stride = 2×2)
Batch Normalization Layer (gain = 1, bias = 0)

Convolutional Layer (channels = 256, kernel-size = 3×3, bias = False, padding = same)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)
Convolutional Layer (channels = 256, kernel-size = 3×3, bias = False, padding = same)
ReLU
Max Pooling Layer (size = 2×2, stride = 2×2)
Batch Normalization Layer (gain = 1, bias = 0)

Convolutional Layer (channels = 512, kernel-size = 3×3, bias = False, padding = same)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)
Convolutional Layer (channels = 512, kernel-size = 3×3, bias = False, padding = same)
ReLU
Max Pooling Layer (size = 2×2, stride = 2×2)
Batch Normalization Layer (gain = 1, bias = 0)

Fully Connected Layer (units = 1024, bias = False)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)

Fully Connected Layer (units = 1024, bias = False)
ReLU
Batch Normalization Layer (gain = 1, bias = 0)

Fully Connected Layer (units = 10, bias = False)
Batch Normalization Layer (gain = 1, bias = 0)
Softmax

MLP Binary Connect Architecture for Continual Learning

Fully Connected Layer (units = 100, bias = False) ReLU Batch Normalization Layer (gain = 1, bias = 0)
Fully Connected Layer (units = 100, bias = False) ReLU Batch Normalization Layer (gain = 1, bias = 0)
Fully Connected Layer (units = 100, bias = False) ReLU Batch Normalization Layer (gain = 1, bias = 0) Softmax

LRNet Architecture (MNIST)

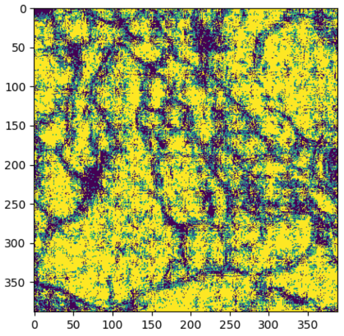
Convolutional Layer (channels = 32, kernel-size = 5×5, bias = False, padding = same) Max Pooling Layer (size = 2×2, stride = 2×2) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Convolutional Layer (channels = 64, kernel-size = 5×5, bias = False, padding = same) Max Pooling Layer (size = 2×2, stride = 2×2) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Fully Connected Layer (units = 512, bias = False) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Fully Connected Layer (units = 10, bias = False) Batch Normalization Layer (gain = 1, bias = 0) Softmax

LRNet Architecture (CIFAR-10)

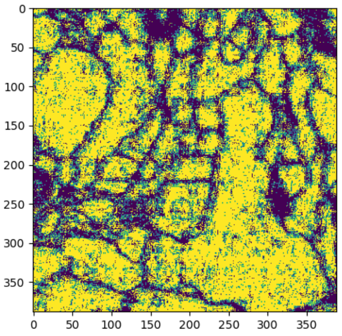
Convolutional Layer (channels = 128, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Convolutional Layer (channels = 128, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) Max Pooling Layer (size = 2×2, stride = 2×2) ReLU
Convolutional Layer (channels = 256, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Convolutional Layer (channels = 256, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) Max Pooling Layer (size = 2×2, stride = 2×2) ReLU
Convolutional Layer (channels = 512, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Convolutional Layer (channels = 512, kernel-size = 3×3, bias = False, padding = same) Batch Normalization Layer (gain = 1, bias = 0) Max Pooling Layer (size = 2×2, stride = 2×2) ReLU
Fully Connected Layer (units = 1024, bias = False) Batch Normalization Layer (gain = 1, bias = 0) ReLU
Fully Connected Layer (units = 10, bias = False) Batch Normalization Layer (gain = 1, bias = 0) Softmax

Semantic Segmentation using BayesBiNN with augmented dataset

We generated 1260 images from 30 original images using the rotation, random horizontal flip, random vertical flip operations. The result for BayesBiNN with this extended dataset was still very poor and inconsistent with the other methods (STE and Full Precision). The results presented in Section 5.4 were to show the extent of difficulty to train BayesBiNN for segmentation task as even with such a small dataset and large number of epochs, it was still not even able to overfit. Following are some of the images obtained by using BayesBiNN with this bigger dataset:



(a) Mask example 1



(b) Mask example 2