Supplemental Material: "Deep Learning for Drug Discovery and Cancer Research: Automated Analysis of Vascularization Images"

Gregor Urban, Kevin Bache, Duc T.T. Phan, Agua Sobrino, Alexander K. Shmakov, Stephanie J. Hachey, Christopher C.W. Hughes, and Pierre Baldi

Hyperparameter Optimization

The following sections summarize all hyperparameters and their ranges that were optimized for the various machine learning models.

1 Baseline Models

We trained the logistic regression model on raw pixel values for all combinations of input image sizes (128×128px, 192×192px, 256×256px, 320×320px), and L2 regularization strengths (0.025, 0.05, 0.1, 0.2, 0.5, 1). For the SIFT/SURF-feature based logistic regression and SVM classifiers we tested all possible combinations of the following hyperparameters: (1) bag of words (BoW) representations using either SIFT or SURF features; (2) binary or non-binary histogram encoding of the BoW features; (3) size of the BoW representation: 100, 200, 300, 400; (4) L2 regularization (0.025, 0.05, 0.1, 0.2, 1, 2, 4, 10).

2 Pre-Trained Convolutional Architecture

The hyperparameters that we optimized when training the pre-trained InceptionV3 network were: the input image size (192×192px, 256×256px, 320×320px; we skip the 128×128px version because InceptionV3 requires input images to be at least 139px×139px), the number of neurons in the first fully connected layer (16, 32, 64, 128, 256), dropout probability for the dropout layer immediately after the first fully connected layer (0.0 to 0.99), the number of neurons in the second fully connected layer (16, 32, 64, 128, 256, 512, 1024), dropout probability for the dropout layer immediately after the second fully connected layer (0.0 to 0.99), the optimization batch size (16 to 64), $\log_{10}$ of the learning rate (-3.0 to 0.0), $\log_{10}$ of the L1 penalty applied to the weights of the network (-9.0 to -1.0), $\log_{10}$ of the L2 penalty applied to the weights of the network (-9.0 to -1.0).

3 Custom Convolutional Architecture

The hyperparameter ranges that we considered for the optimization of the custom convolutional network model were: the input image size (128×128px, 192×192px, 256×256px, or 320×320px), the number of convolutional layers in the model (2 to 7), the number of convolutional filters at the start of the convolutional cascade (16, 32, 64, 128 or 256), the number of convolutional filters at the end of the cascade (16, 32, 64, 128 or 256; filter counts were linearly interpolated across the 2 to 7 convolutional layers between the number of filters at the start and the number of filters at the end), the number of convolutional layers between max pooling layers (1 to 4; the first pooling layer was fixed after the second convolutional layer), the size of the max pooling receptive fields (2 to 7; stride was fixed to match pooling size), dropout probability for a dropout layer immediately after the convolutional layers (0.00 to 0.99), the number of neurons in the first fully connected layer (16, 32, 64, 128, 256), dropout probability for the dropout layer immediately after the first fully connected layer (0.0 to 0.99), the number of neurons in the second fully connected layer (16, 32, 64, 128, 256, 512, 1024), dropout probability for the dropout layer immediately after the second fully connected layer (0.0 to 0.99), the optimization batch size (1 to 8), $\log_{10}$ of the learning rate (-3.0 to 0.0), $\log_{10}$ of the L1 penalty applied to the weights of the network (-9.0 to -1.0), $\log_{10}$ of the L2 penalty applied to the weights of the network (-9.0 to -1.0).