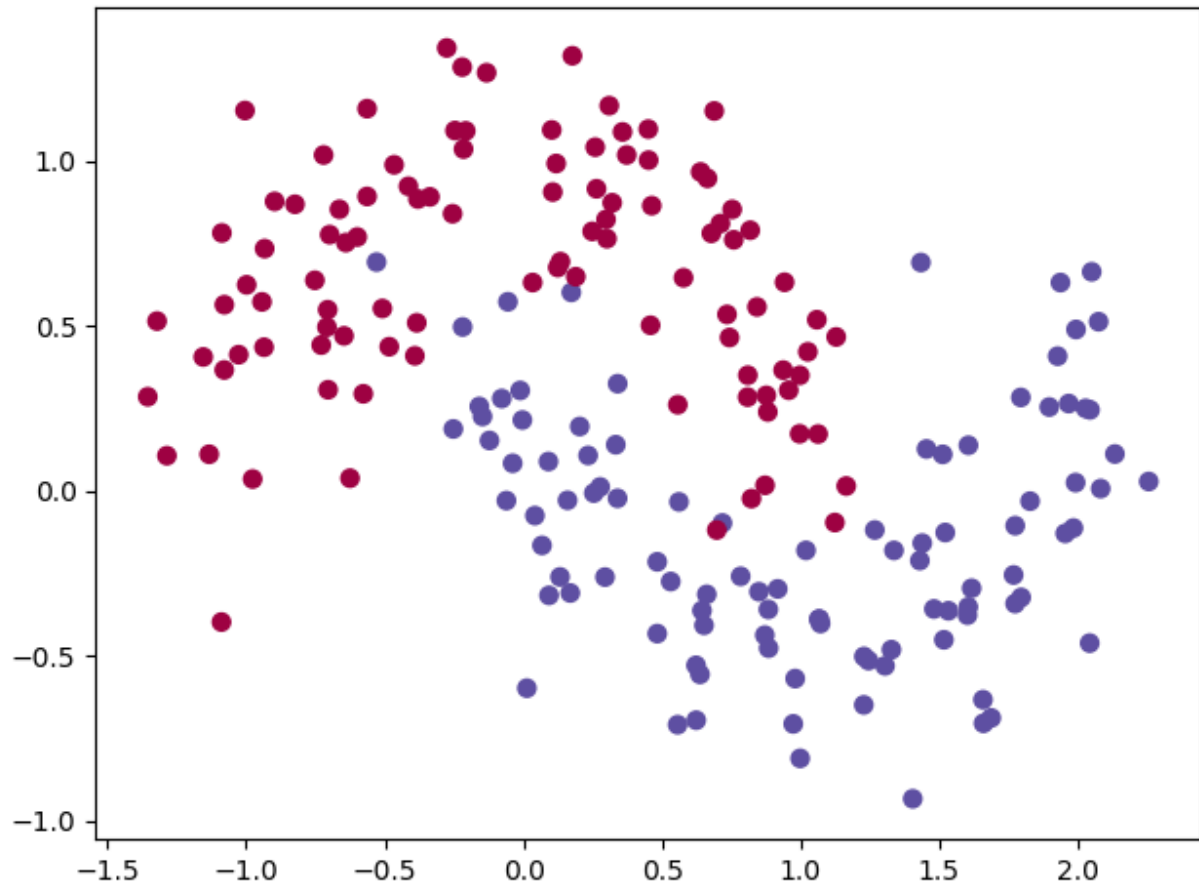


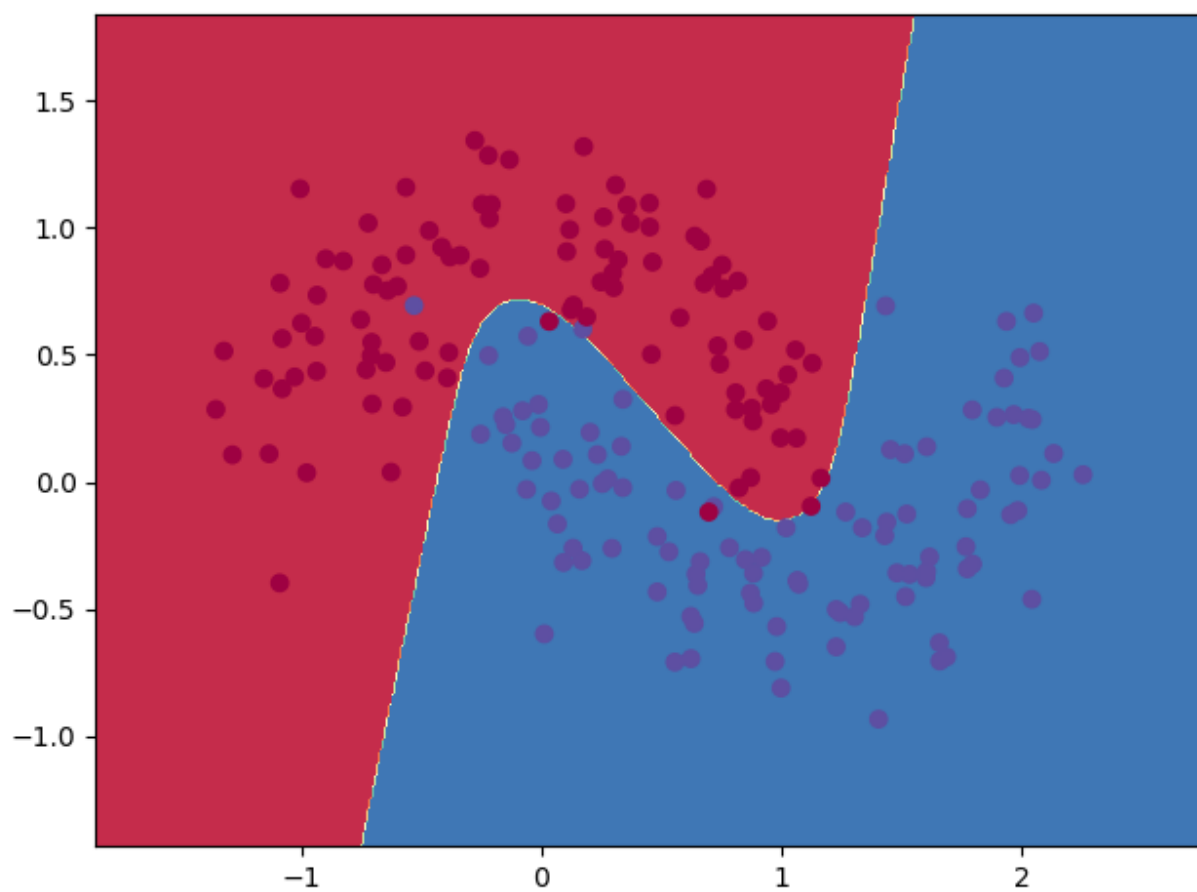
1. Backpropagation in a Simple Neural Network

a. Dataset

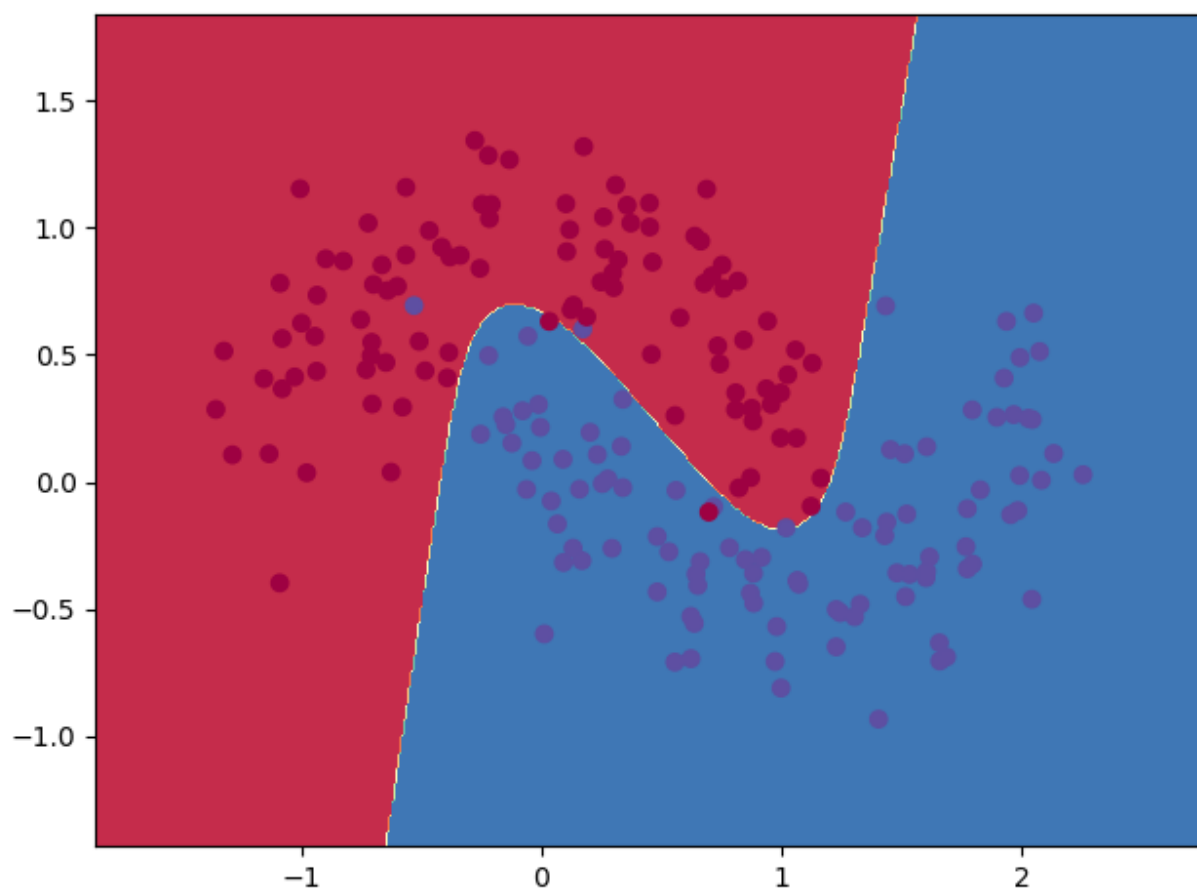


- b. Train the network using different activation functions (Tanh, Sigmoid and ReLU). We can notice that the ReLU function will create a sharp decision boundary, while both Tanh and Sigmoid functions will create a smooth decision boundary.

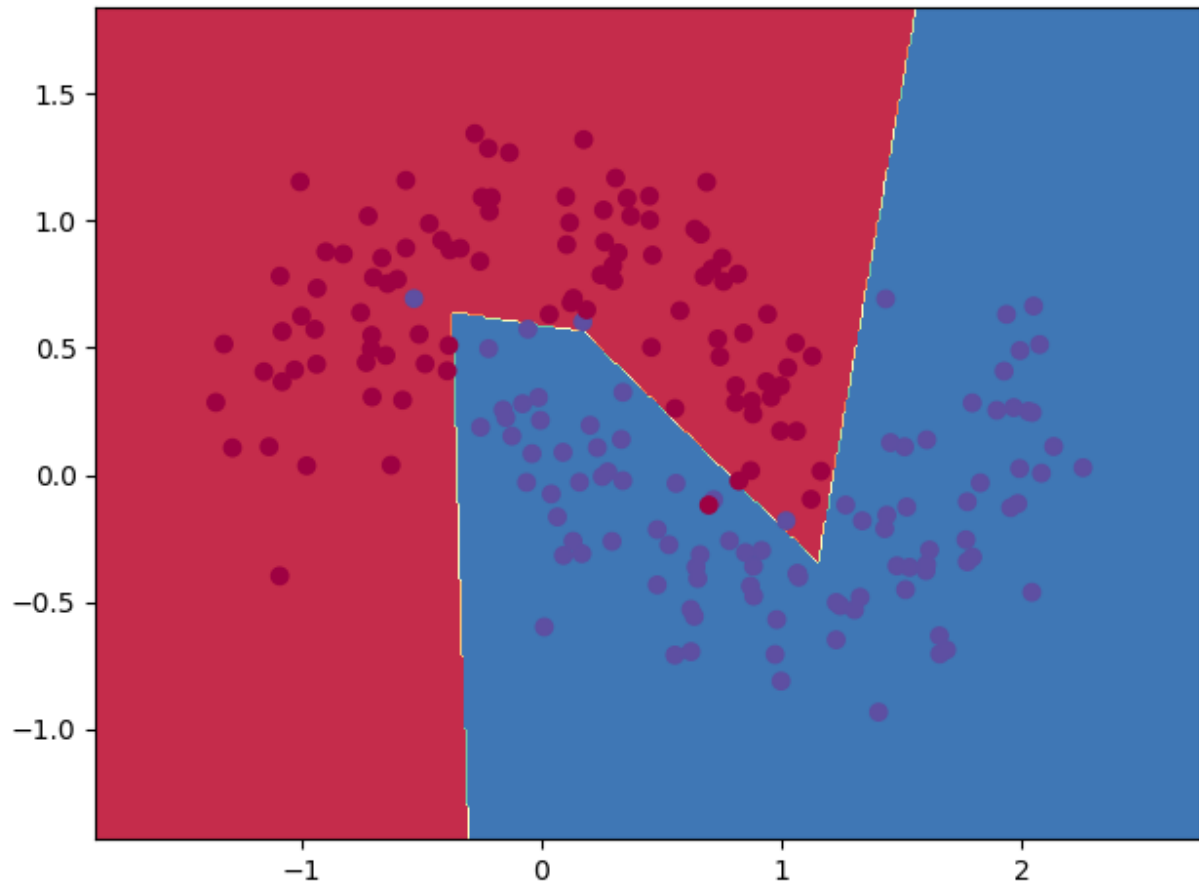
Tanh



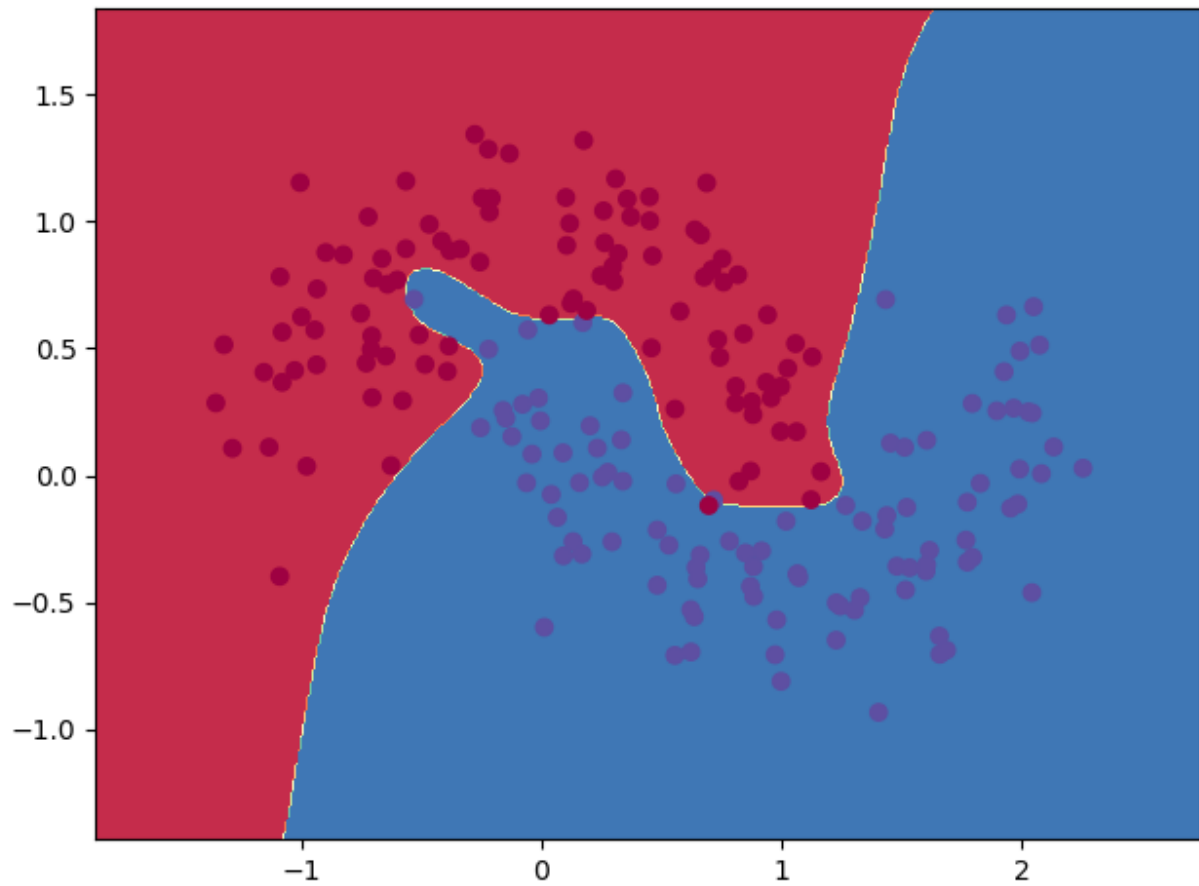
Sigmoid



ReLU

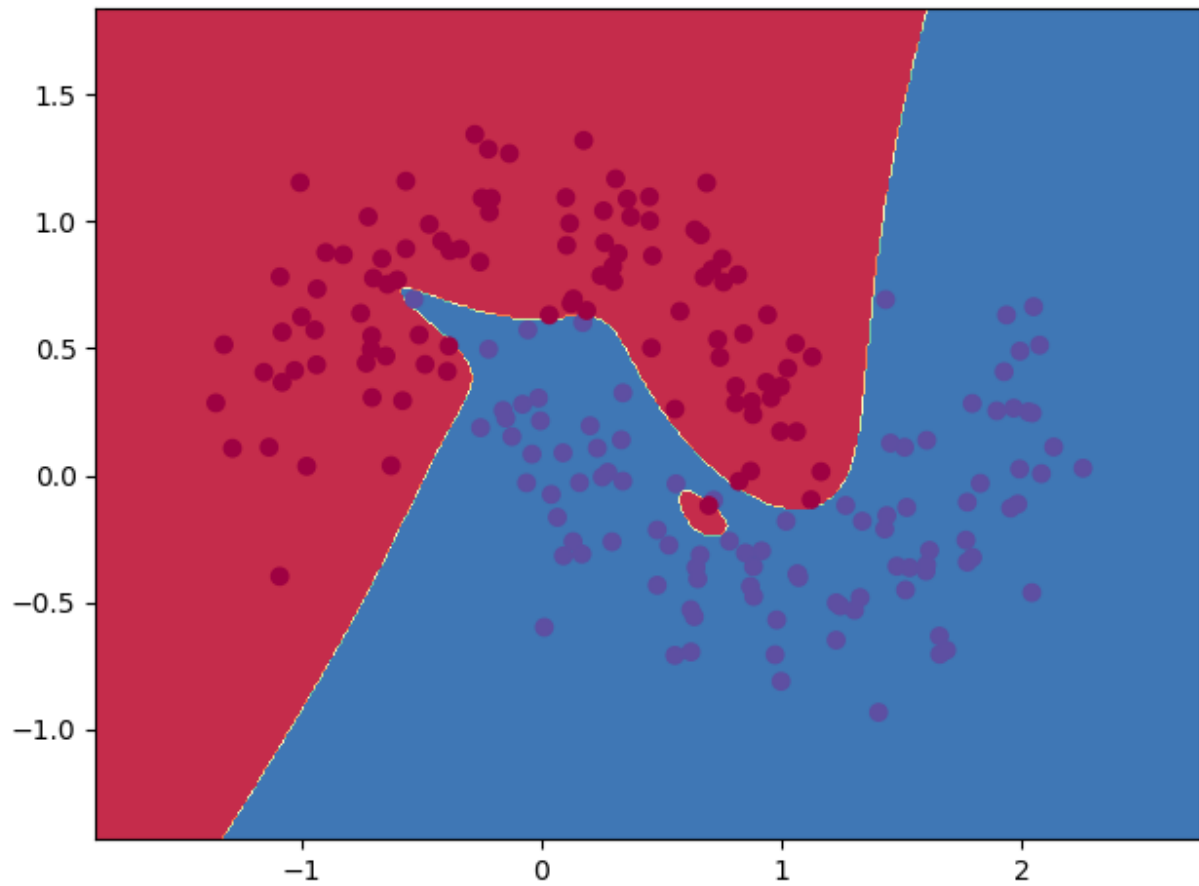


More hidden units with Tanh (`nn_hidden_dim=8`). When I increased the number of hidden units to 8, the decision boundary became overfitting to the data points. The reason for overfitting is that the model now is too complex and it closely fits to a limited set of data points



More fun -Training a Deeper Network:

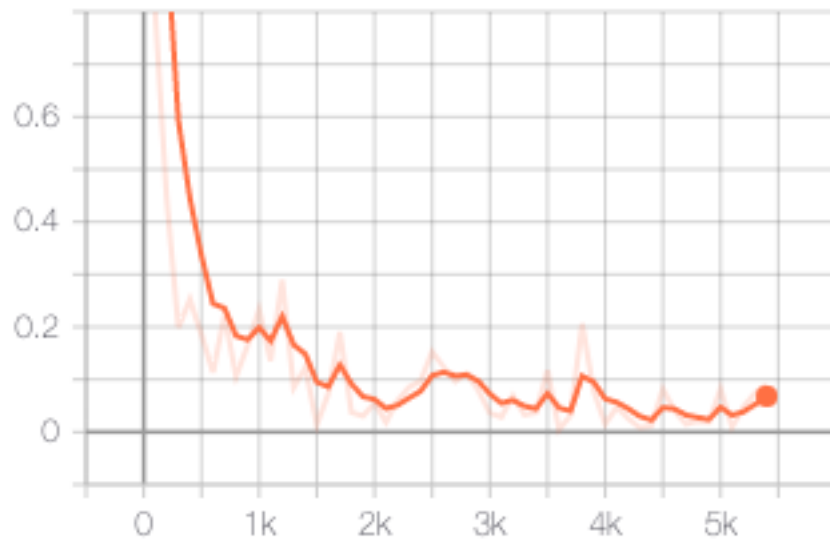
I implemented a `n_layer_neural_network` class that inherits the `three_layer_neural_network`. Then I called `DeepNeuralNetwork(n_hidden=5, input_dim=2, hidden_dim=8, output_dim=2, actFun_type="Tanh")` and the result is displayed as below.



2. Training a Simple Deep Convolutional Network on MNIST

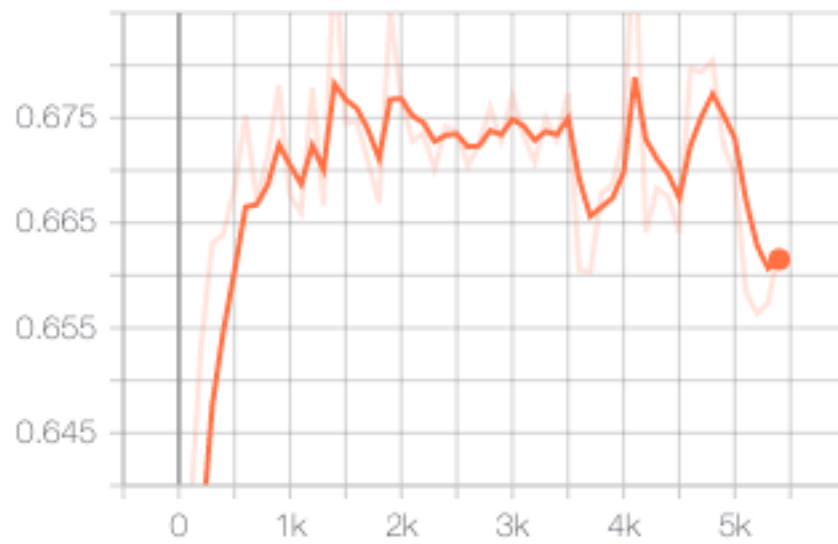
a. Visualize Training

CrossEntropyLoss_1

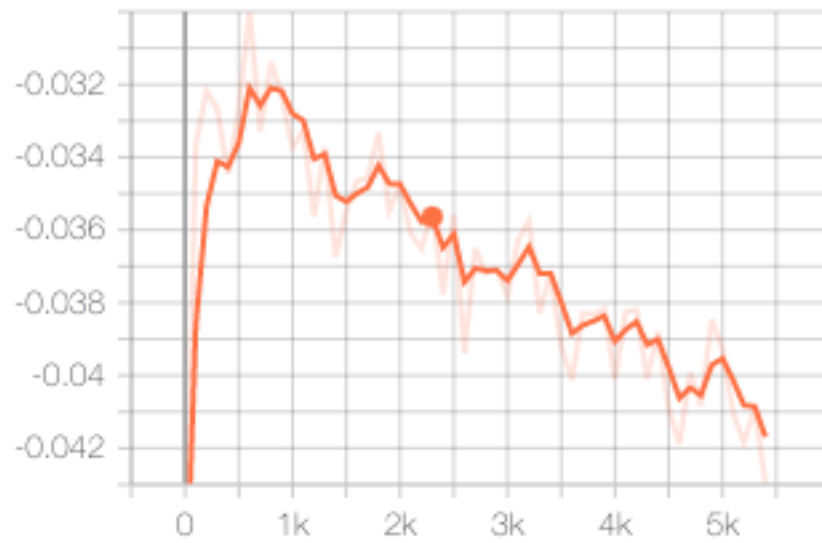


b. More on Visualizing Your Training

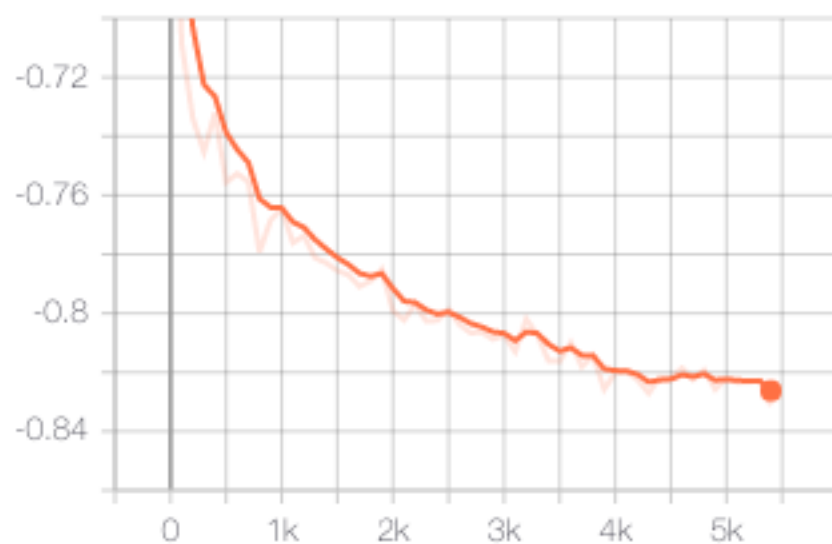
conv1_activation_max



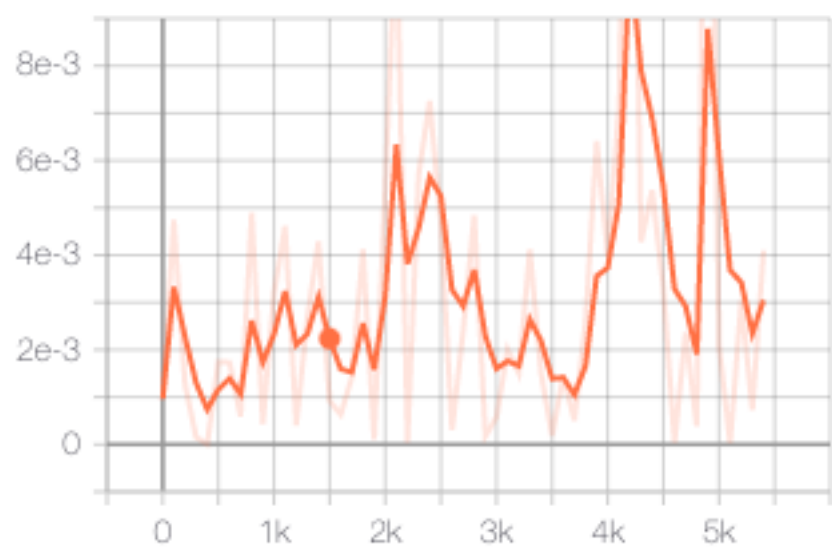
conv1_activation_mean



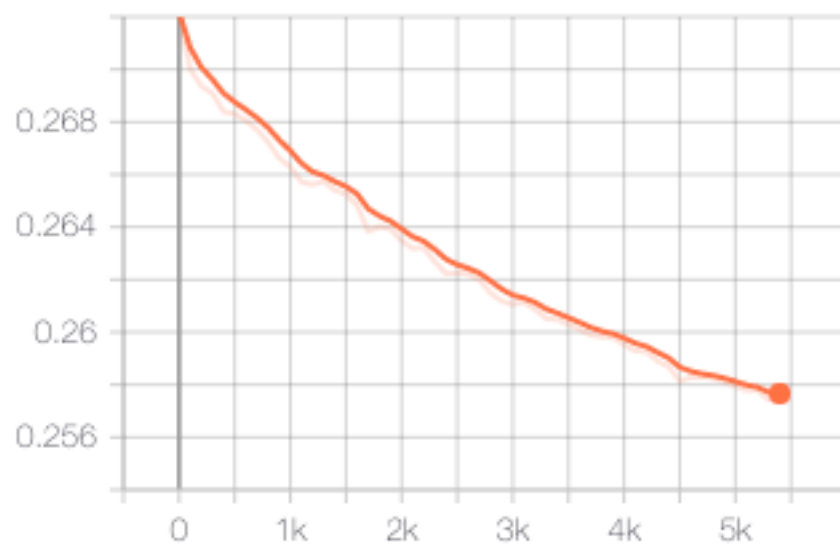
conv1_activation_min



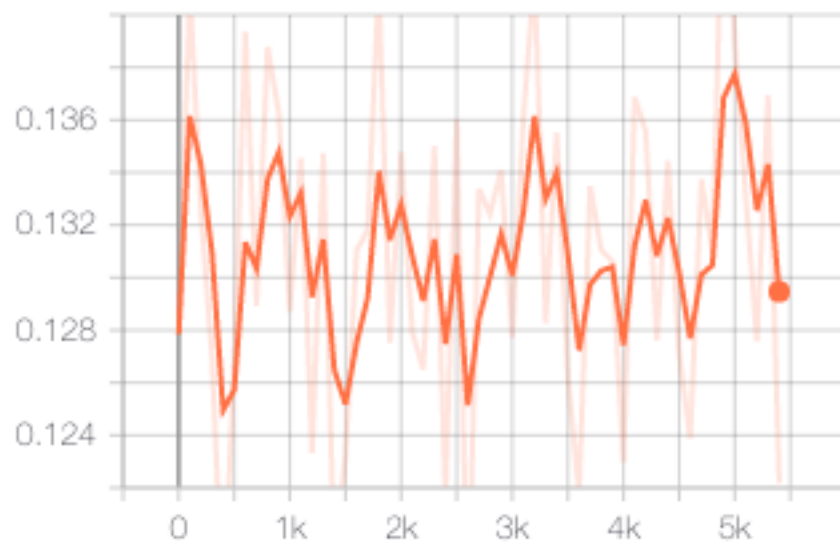
conv1_activation_std



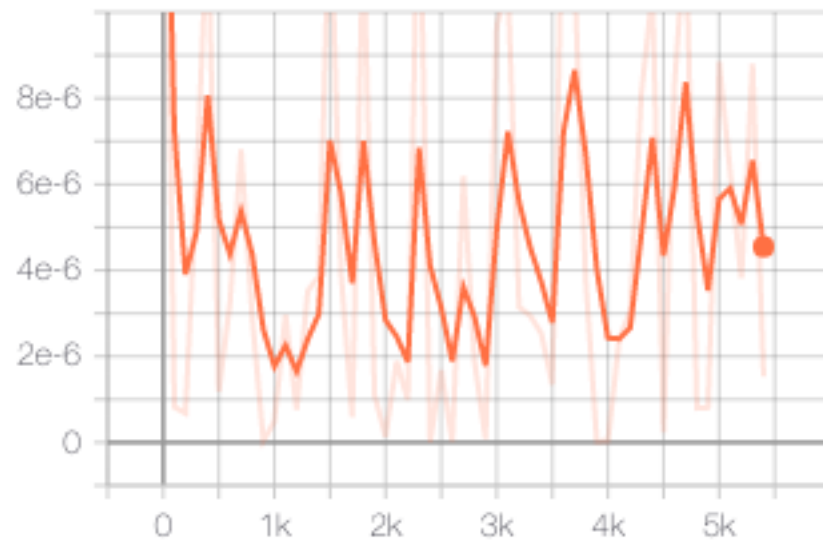
conv1_bias_max



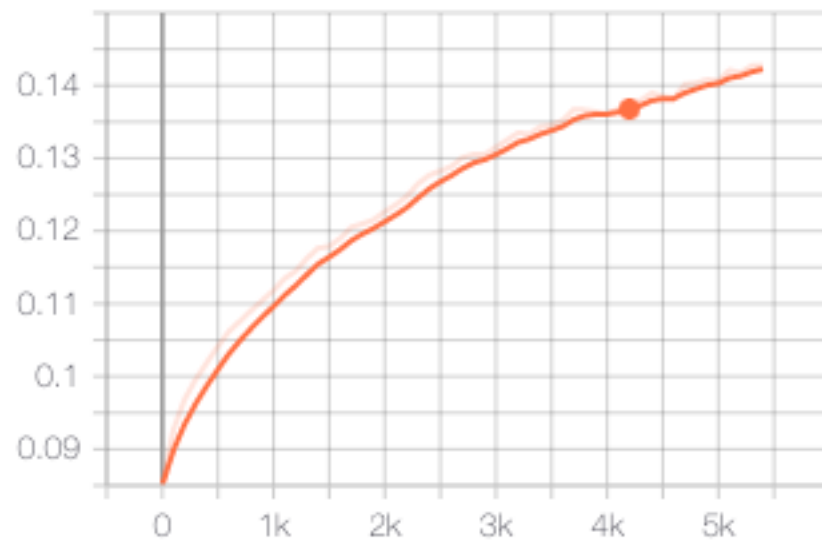
conv1_input_mean



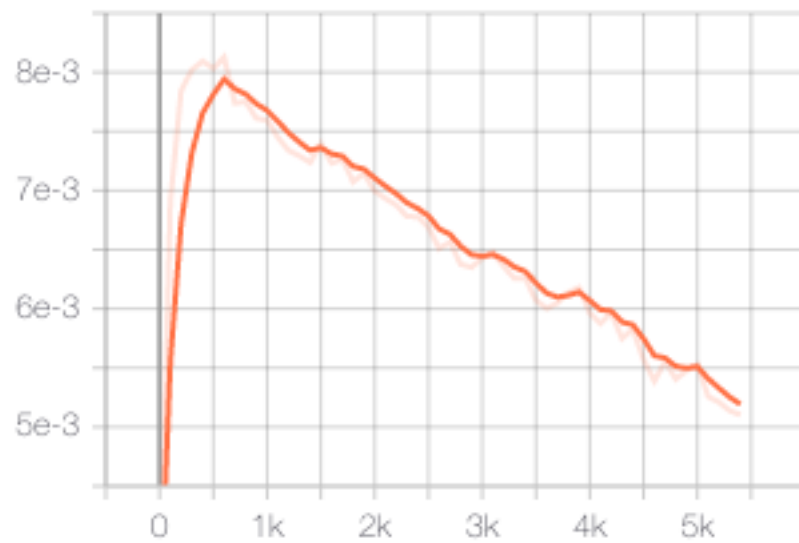
conv1_input_std



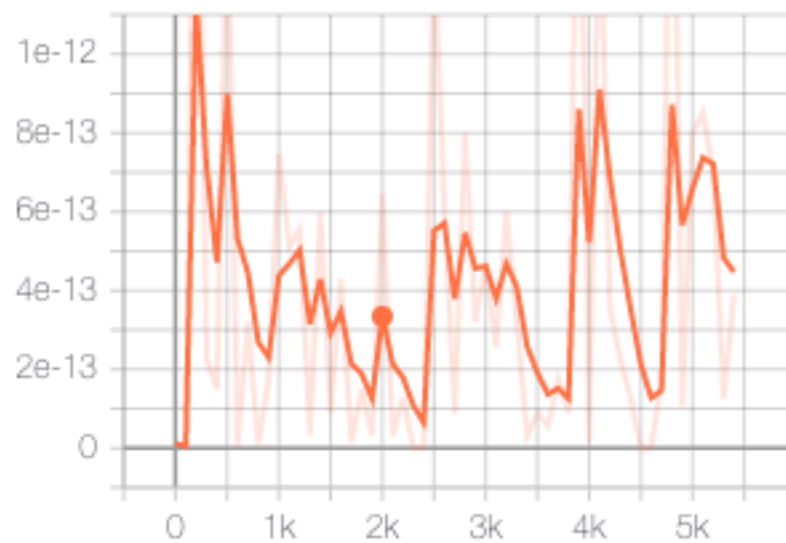
conv1_weight_max



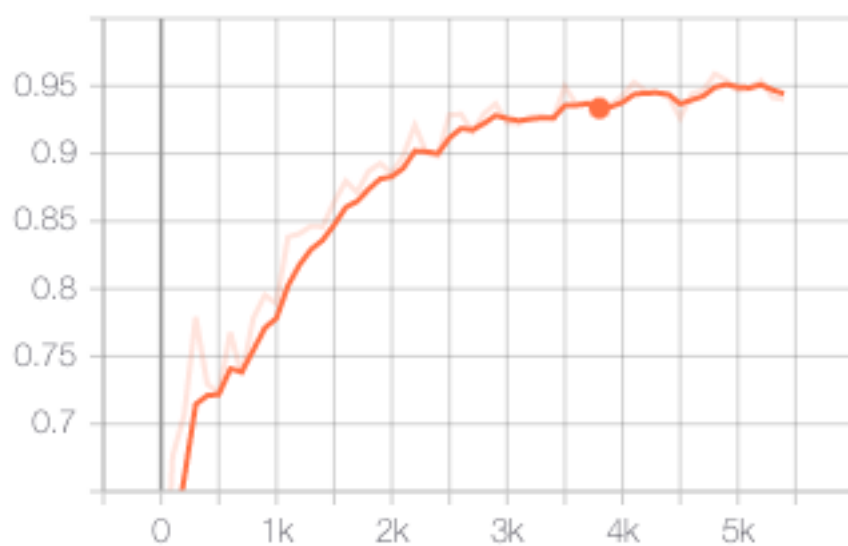
conv1_weight_mean



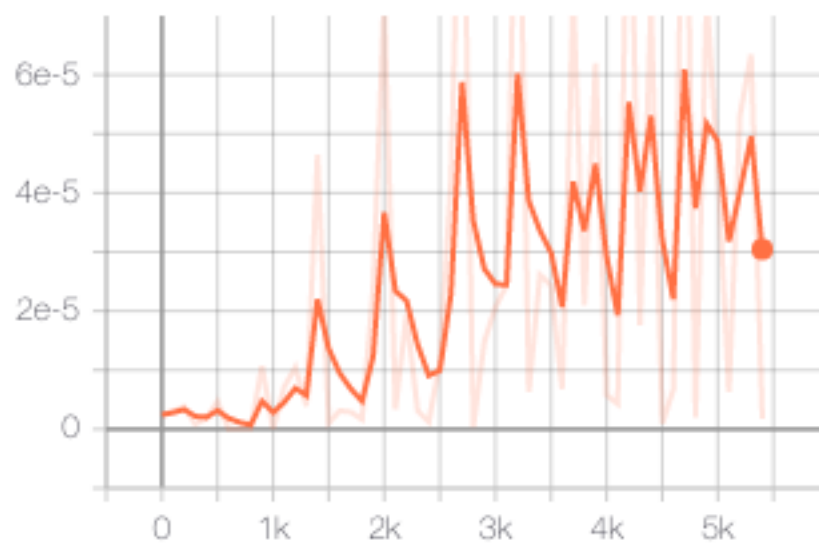
conv1_weight_std



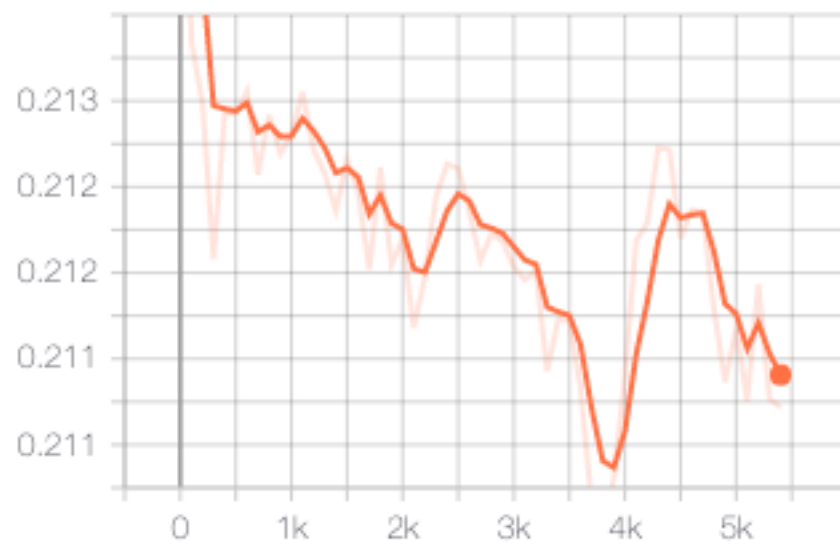
conv2_activation_max



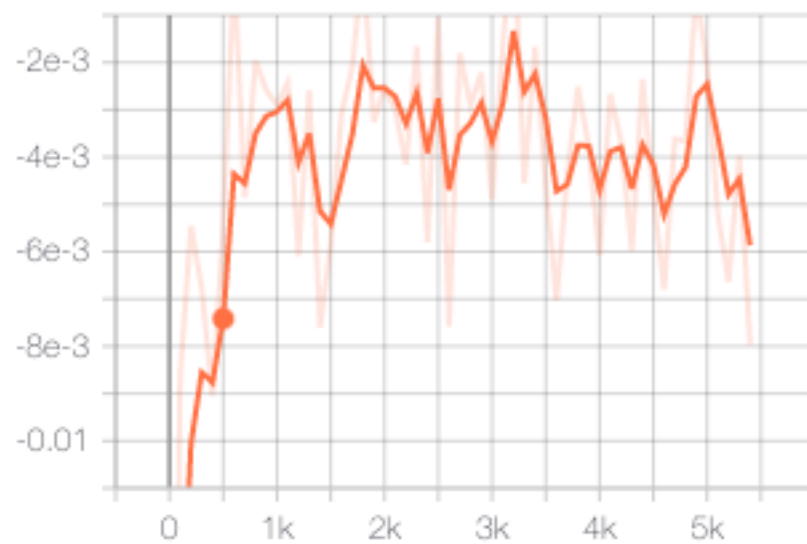
conv2_activation_std



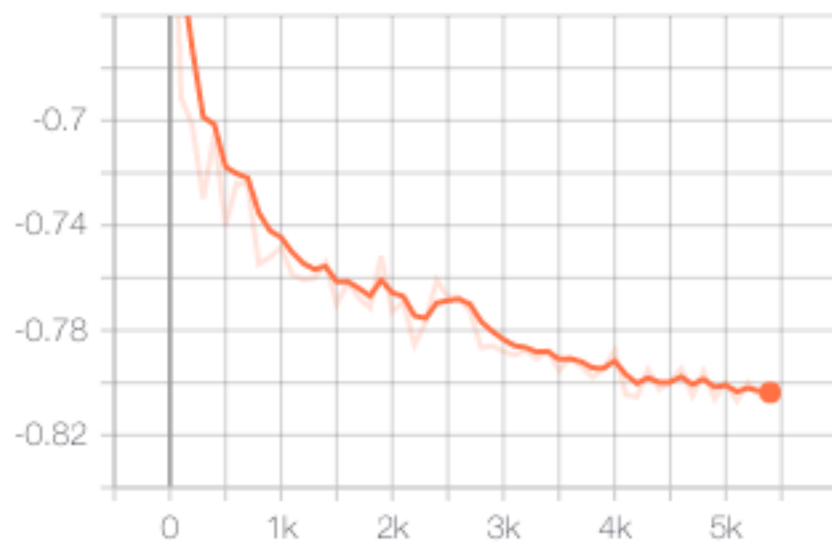
conv2_bias_max



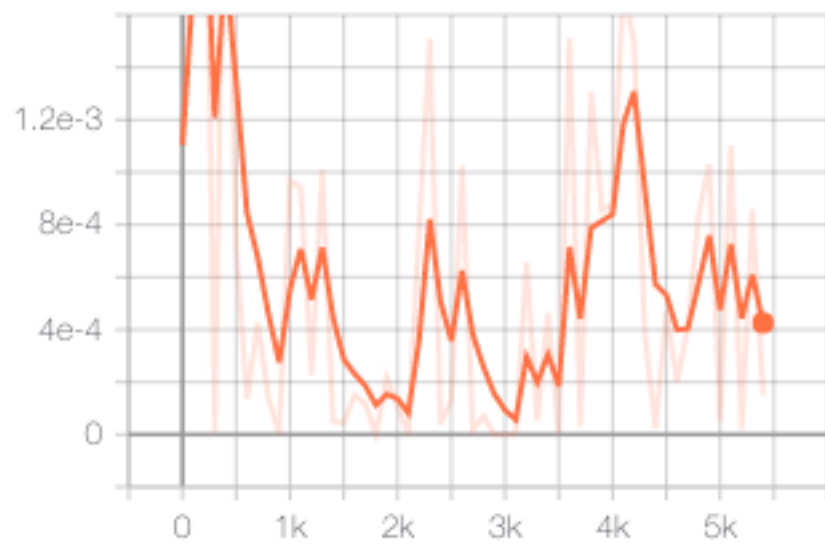
conv2_input_mean



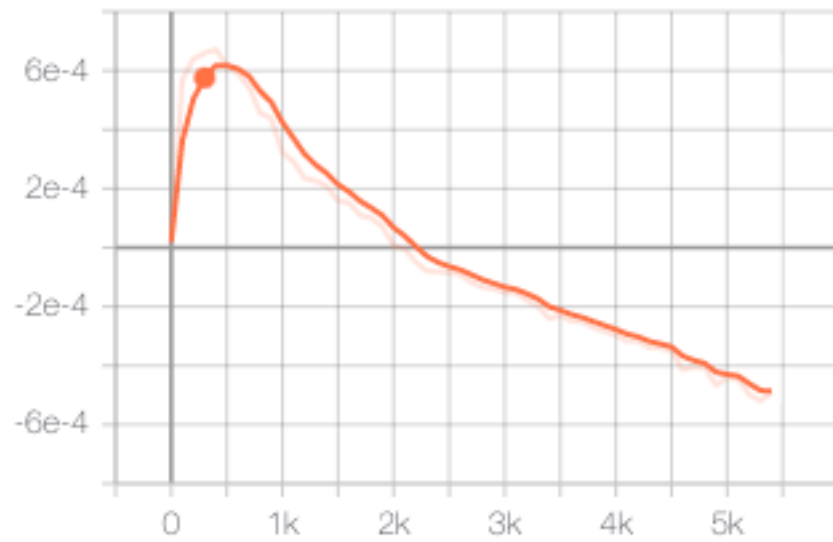
conv2_input_min



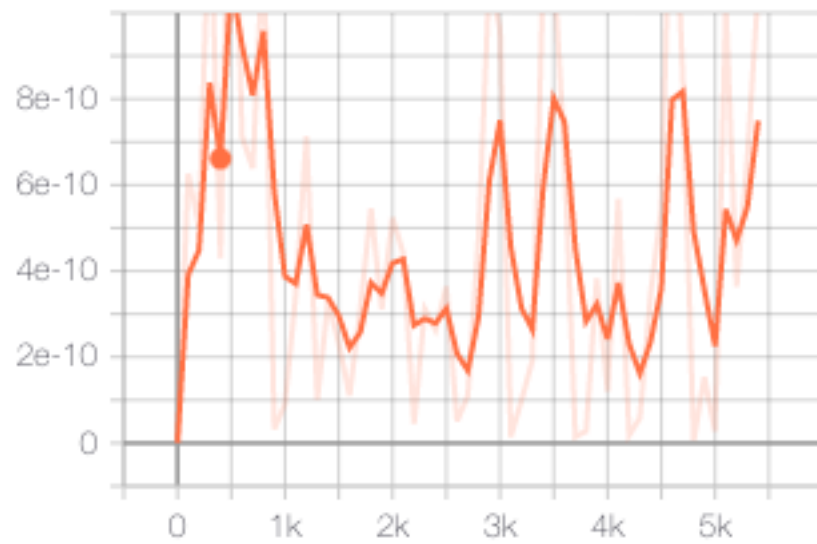
conv2_input_std



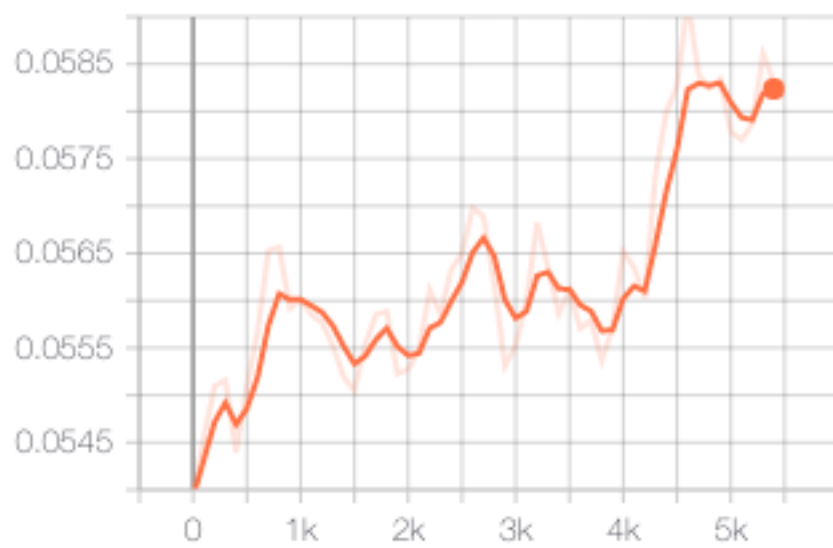
conv2_weight_mean



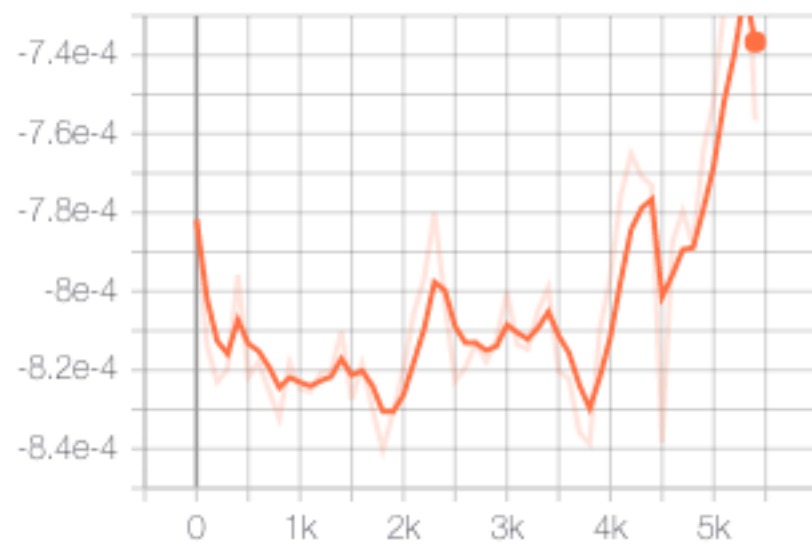
conv2_weight_std



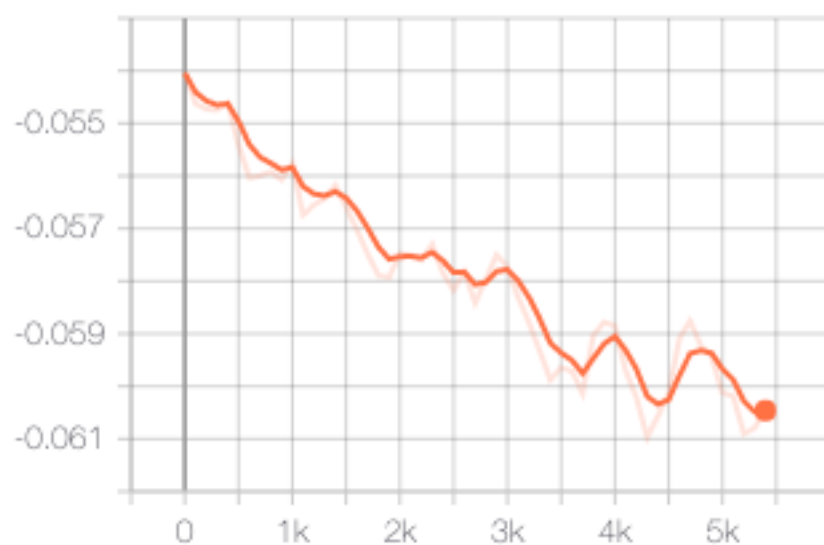
fc_bias_max



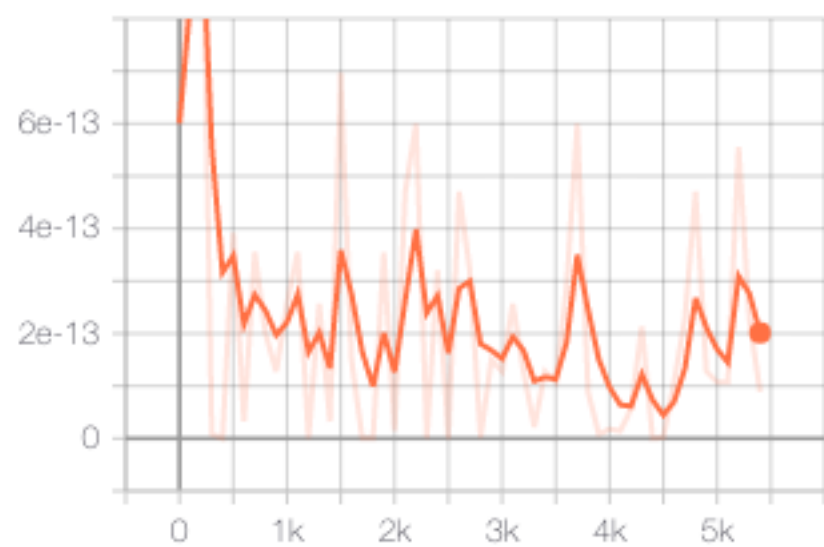
fc_bias_mean



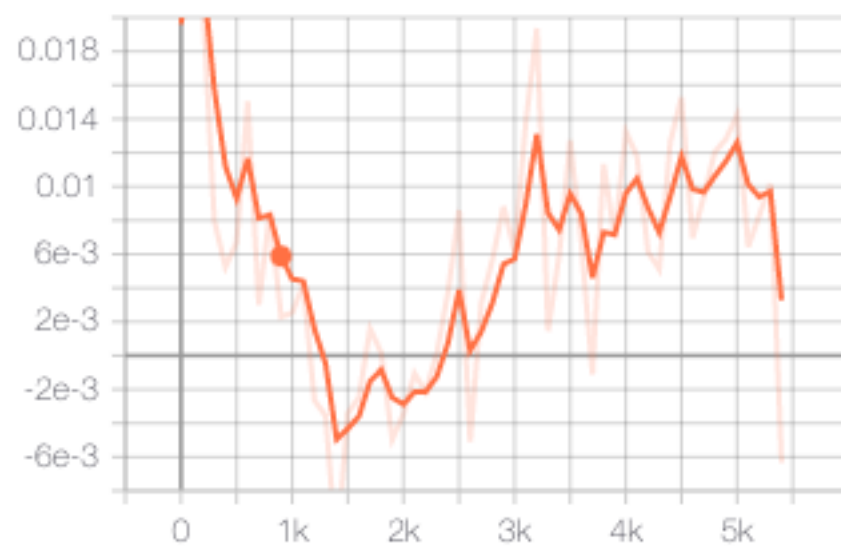
fc_bias_min



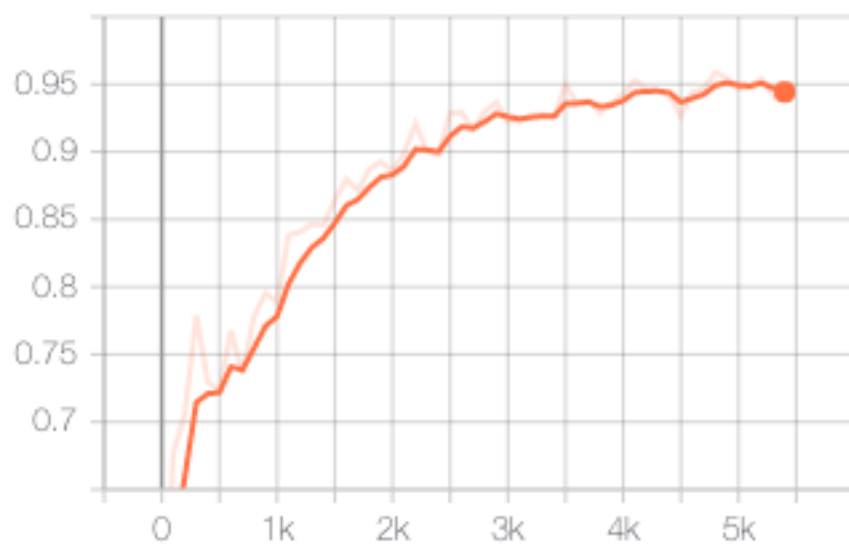
fc_bias_std



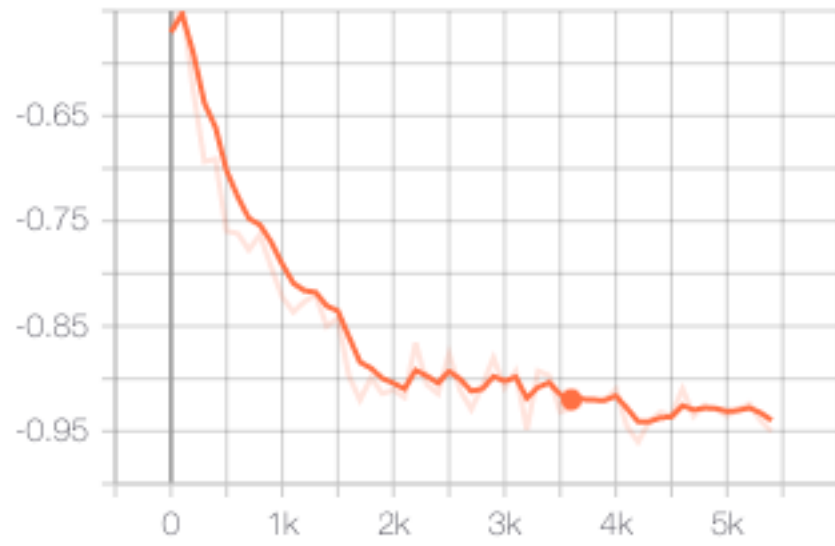
fc_input_mean



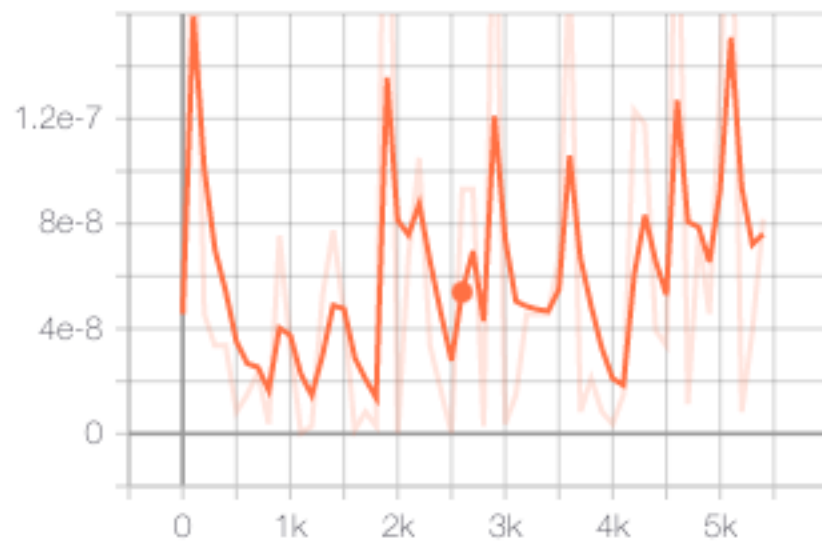
fc_input_max



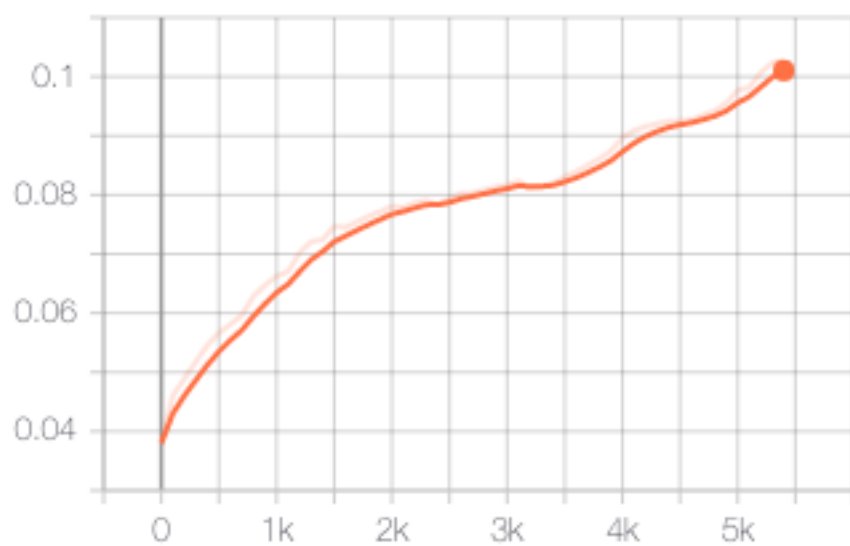
fc_input_min



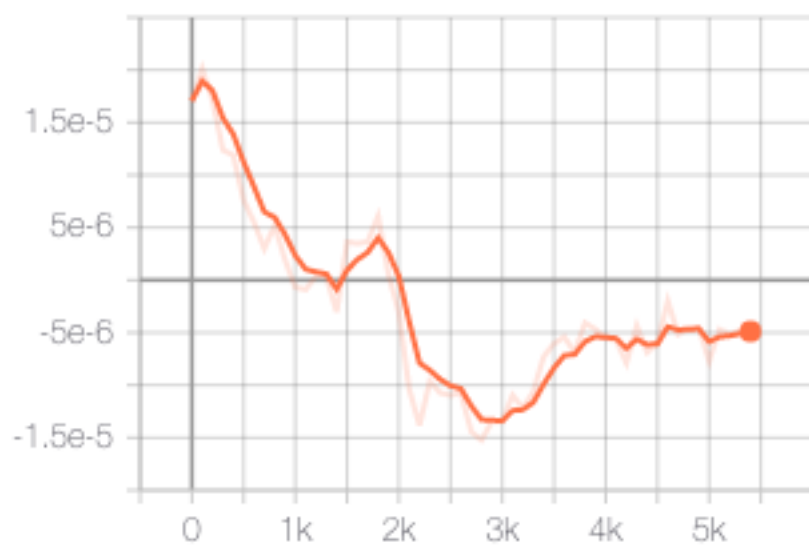
fc_input_std



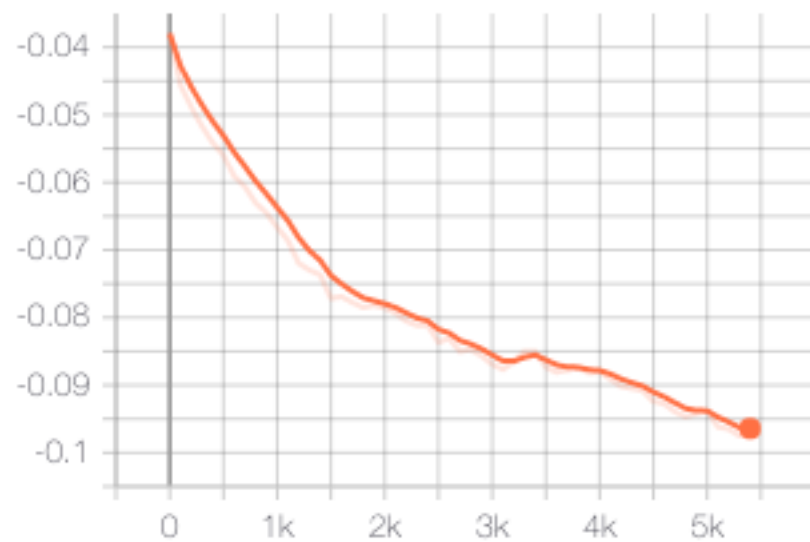
fc_weight_max



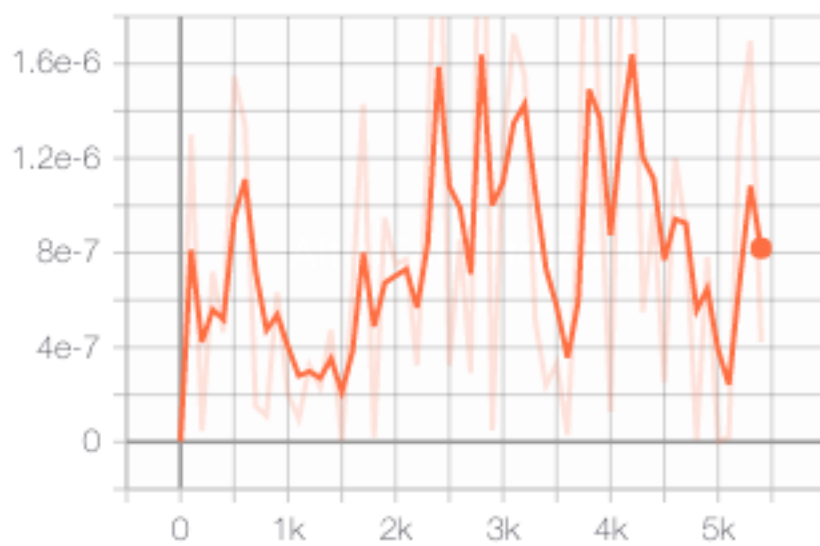
fc_weight_mean



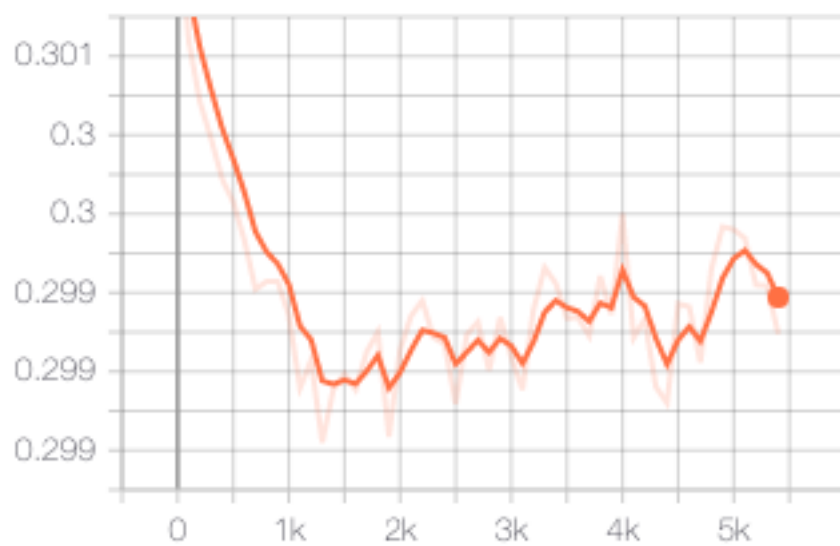
fc_weight_min



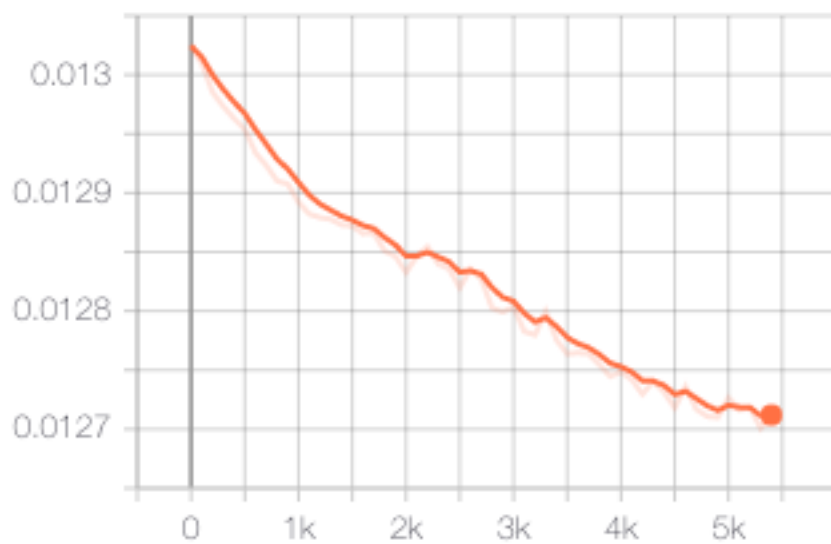
fc_weight_std



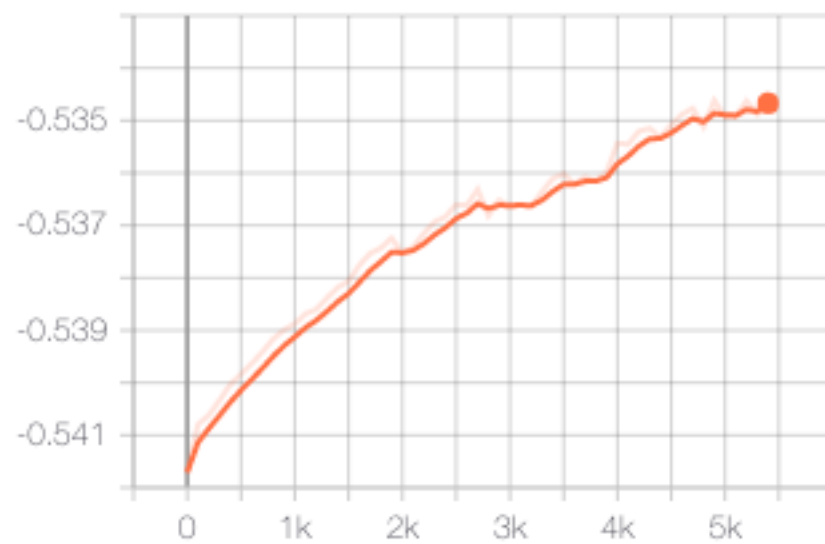
softmax_bias_max



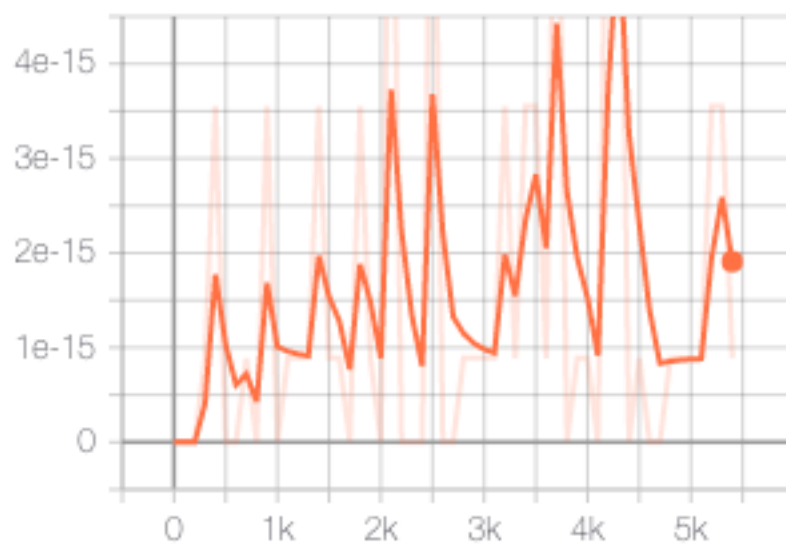
softmax_bias_mean



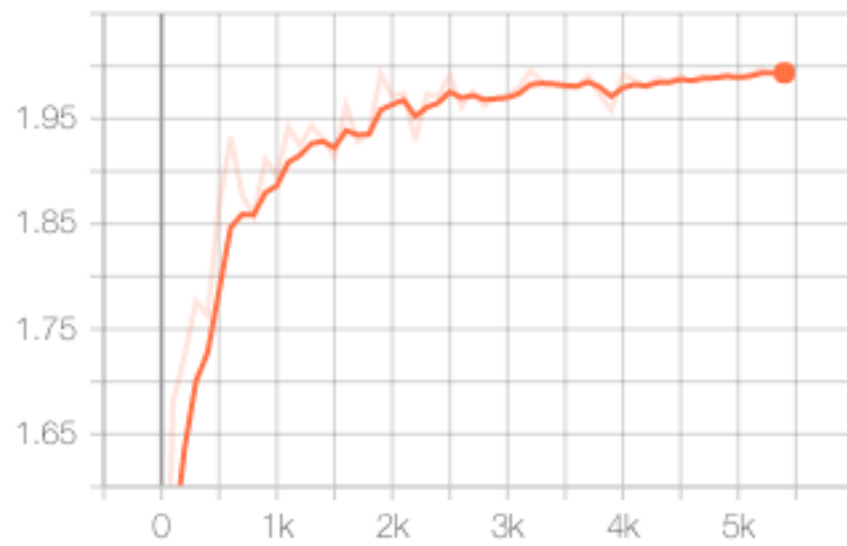
softmax_bias_min



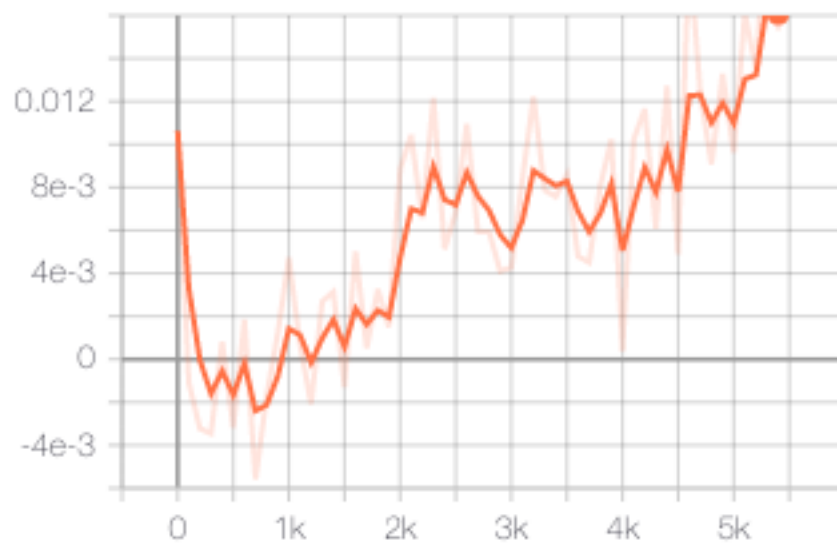
softmax_bias_std



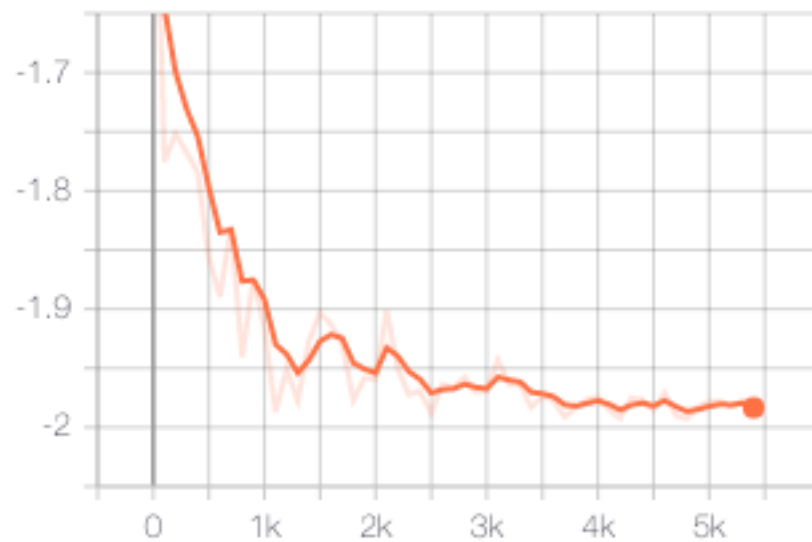
softmax_input_max



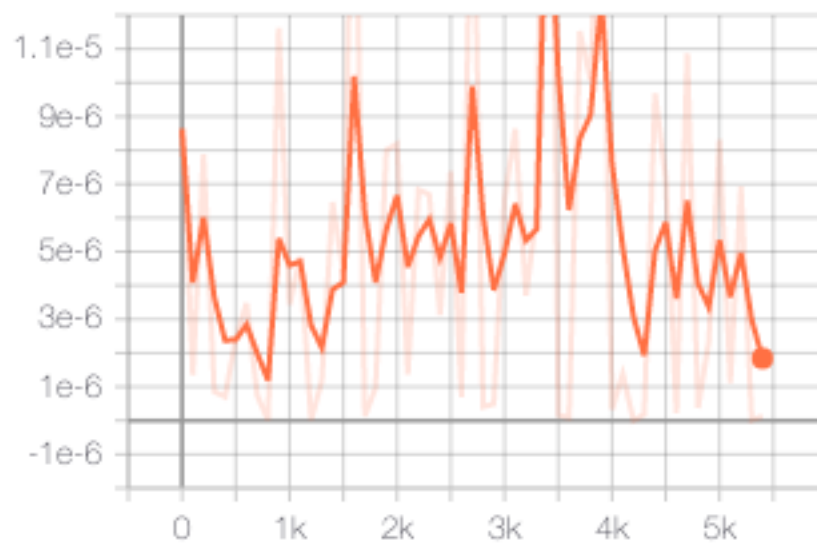
softmax_input_mean



softmax_input_min

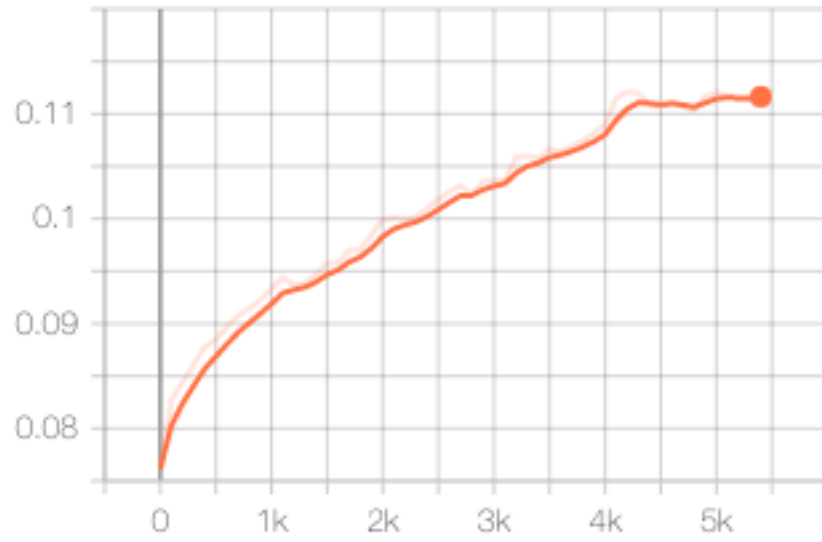


softmax_input_std

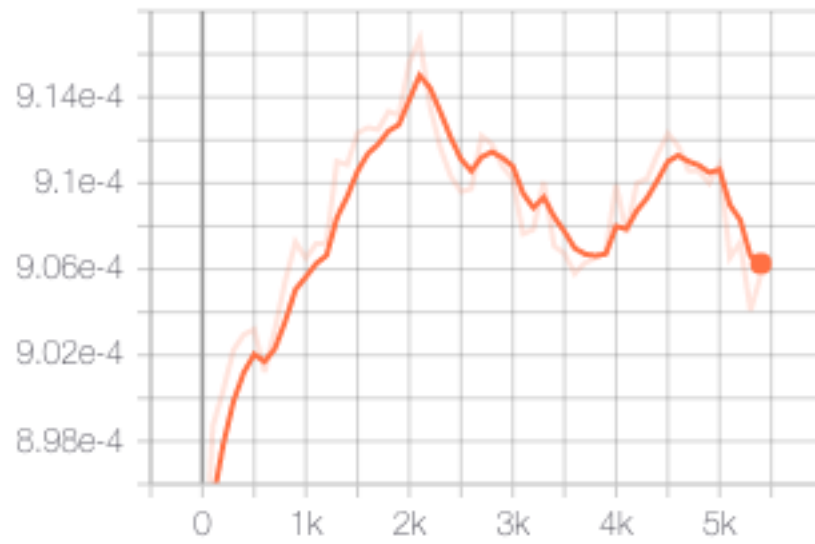


-- — —

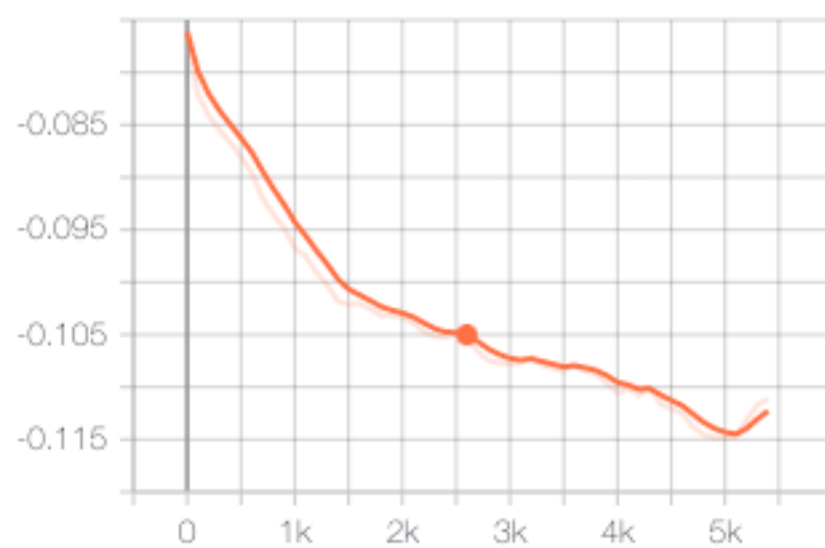
softmax_weight_max



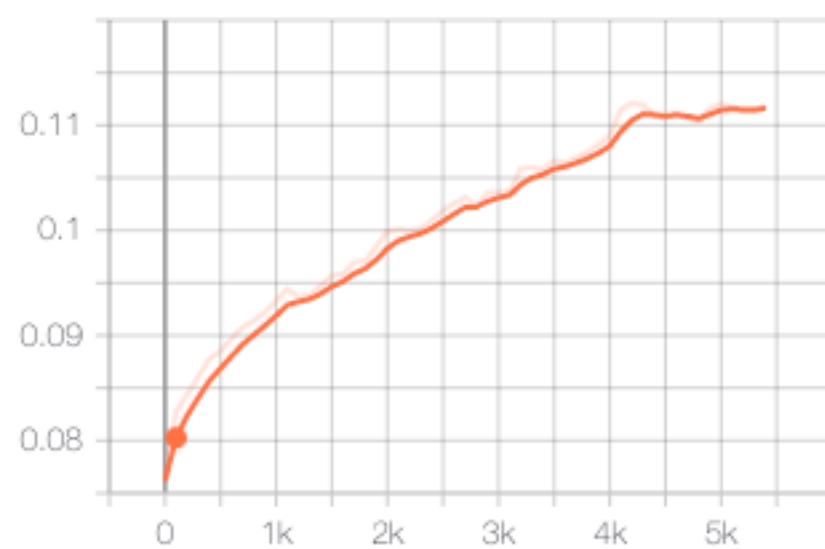
softmax_weight_mean



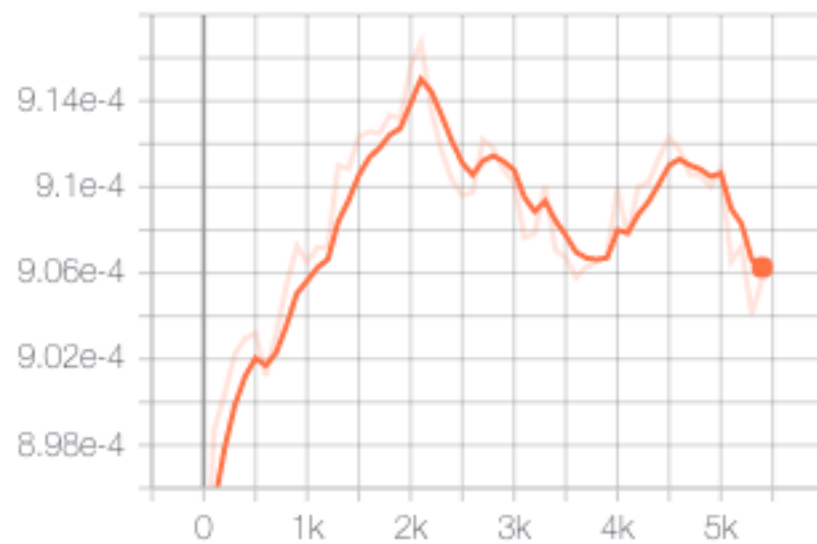
softmax_weight_min



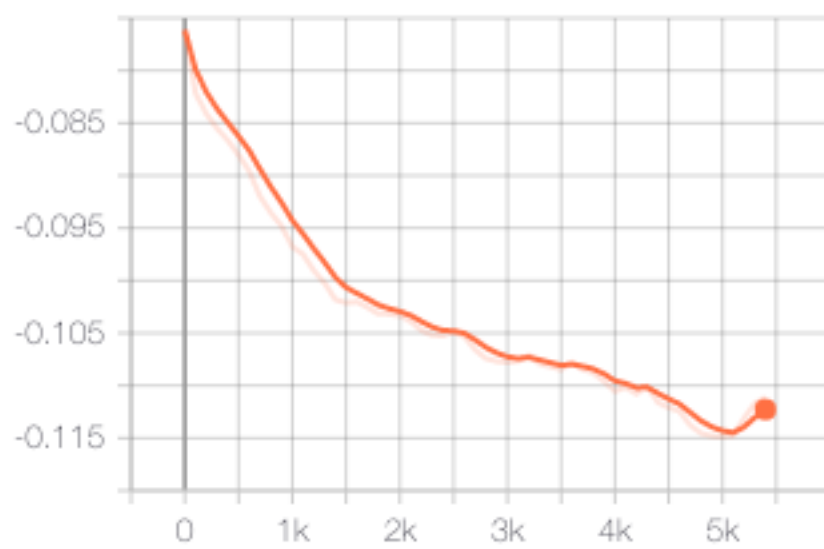
softmax_weight_max



softmax_weight_mean



softmax_weight_min



softmax_weight_std

