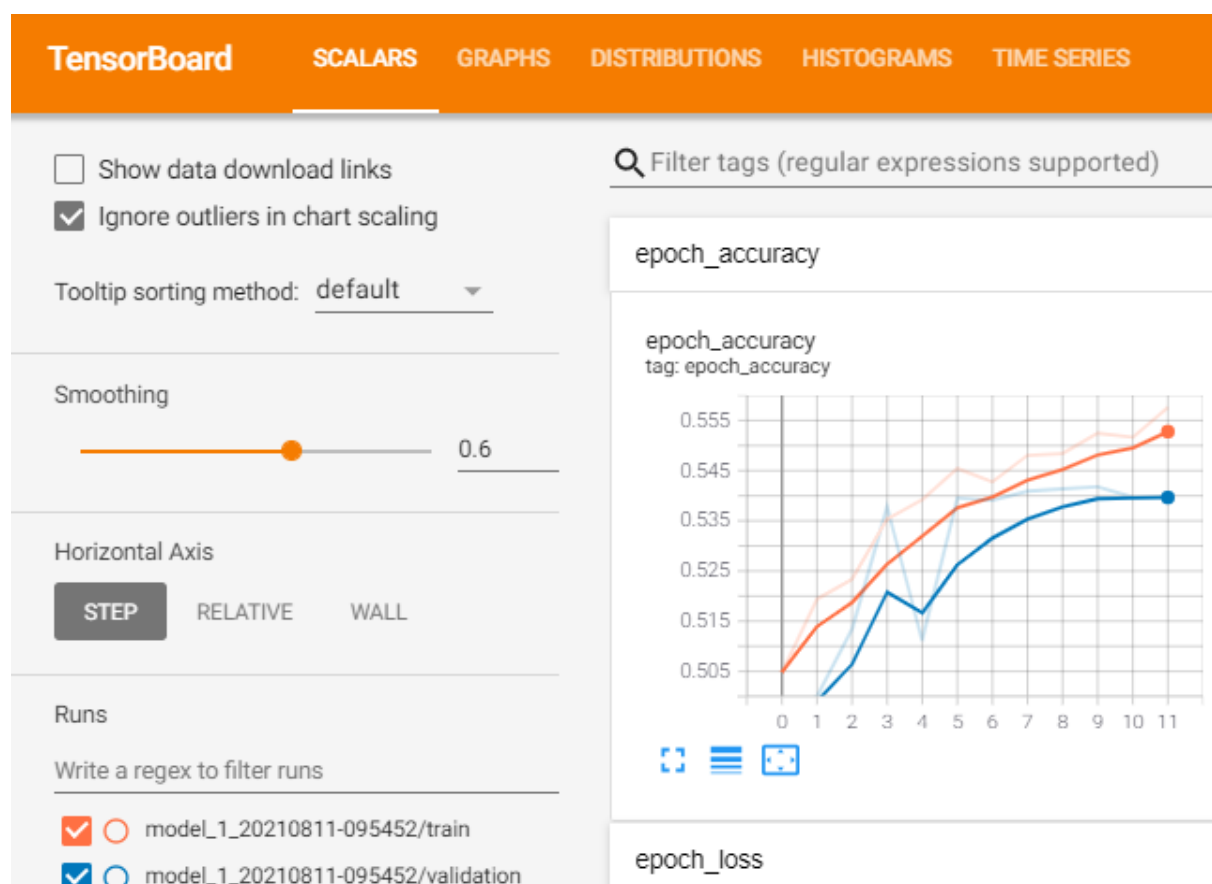


CALLBACK ASSIGNMENT OBSERVATIONS

Model-1

1. Use tanh as an activation for every layer except the output layer.
2. use SGD with momentum as an optimizer.
3. use RandomUniform(0,1) as initializer.
4. Analyze your output and training process.

Accuracy vs Epoch Graph :



Observations:

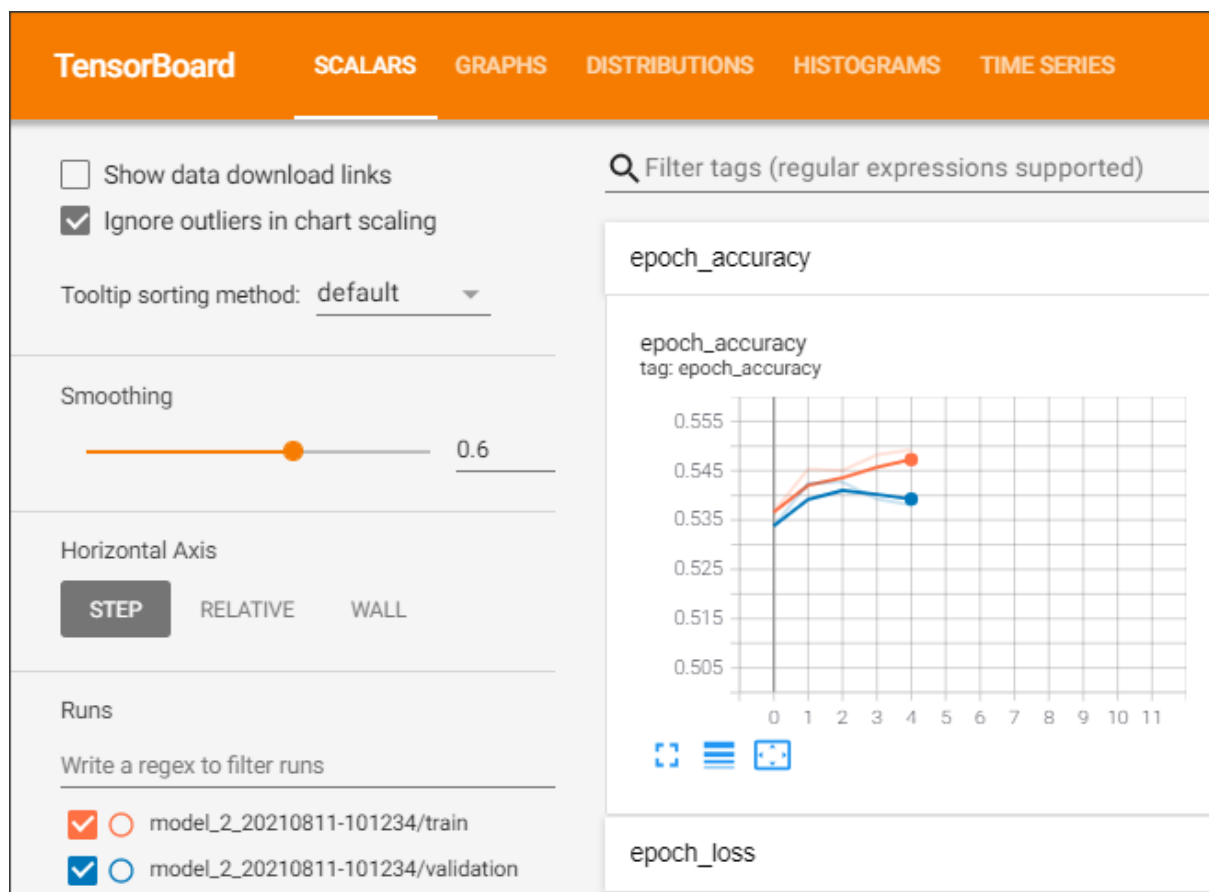
1. We have used tanh as an activation function here. It is similar to sigmoid function, the only difference is that it is symmetric around the origin. Thus the inputs to the next layer will not always be of the same sign.
2. The learning of the model here is quite slow. After the first epoch, the test accuracy is 0.49 which is very low.

3. Test accuracy improves slowly after reducing learning rate.
4. It takes 11 epochs to reach to a test accuracy of 0.54
5. Because of using RandomUniform(0,1) as a weight initializer, the weight update is minor hence making the optimization of our loss function slow.
6. Problem of vanishing gradient is observed.

Model-2

1. Use relu as an activation for every layer except the output layer.
2. use SGD with momentum as an optimizer.
3. use RandomUniform(0,1) as initializer.
3. Analyze your output and training process.

Accuracy vs Epoch Graph :



Observations:

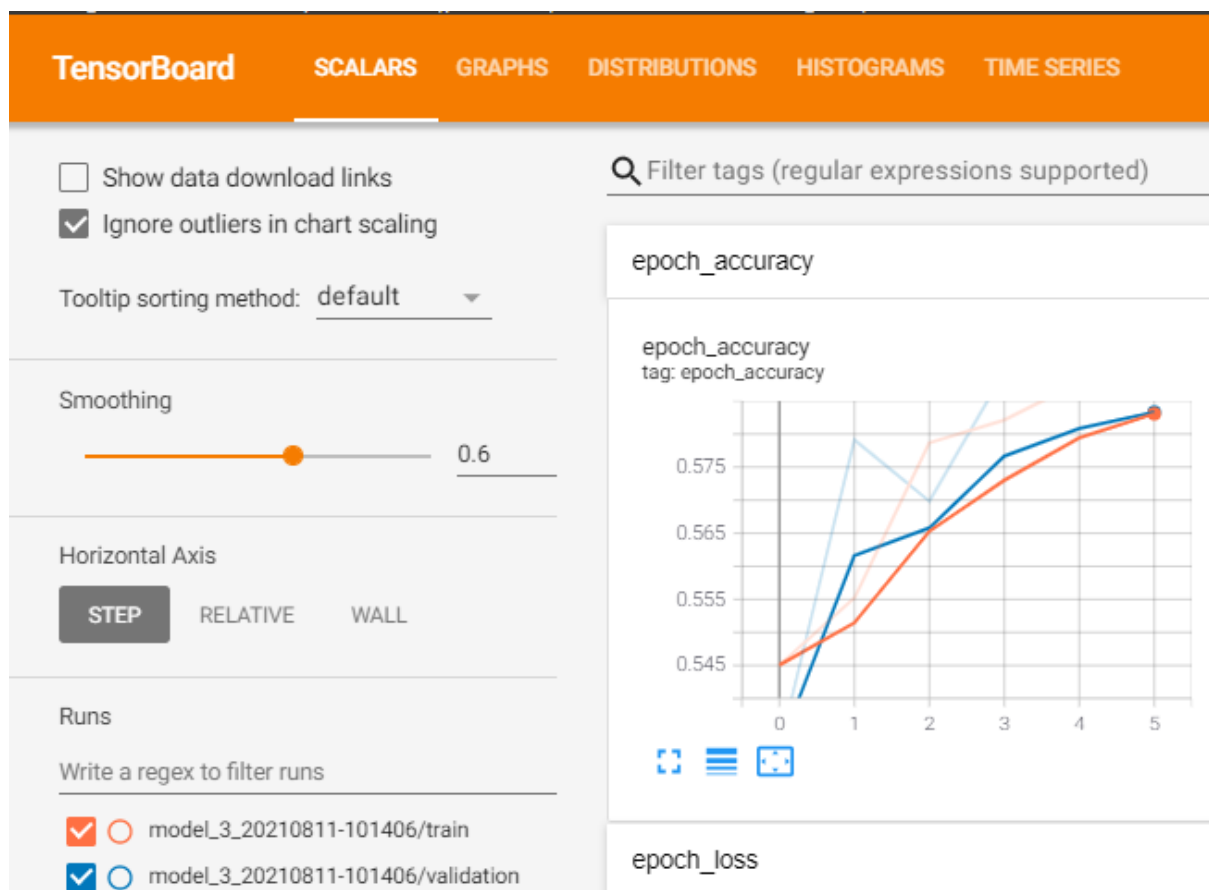
1. In model 2, except for the activation function, everything is the same as Model 1. Here we have used Relu as an activation function.

2. As Relu does not activate all the neurons at the same time, it is far more computationally efficient and fast as compared to the tanh function used in model 1.
3. Vanishing gradient is not a problem here as we are using Relu.
4. There is not any improvement in the test accuracy as compared to model 1 (i.e. 0.54 in both the cases) however because of Relu the convergence is faster.
5. Model reaches a test accuracy of 0.54 in 5 epochs only.

Model-3

1. Use relu as an activation for every layer except the output layer.
2. use SGD with momentum as an optimizer.
3. use he_uniform() as initializer.
3. Analyze your output and training process.

Accuracy vs Epoch Graph :



Observations:

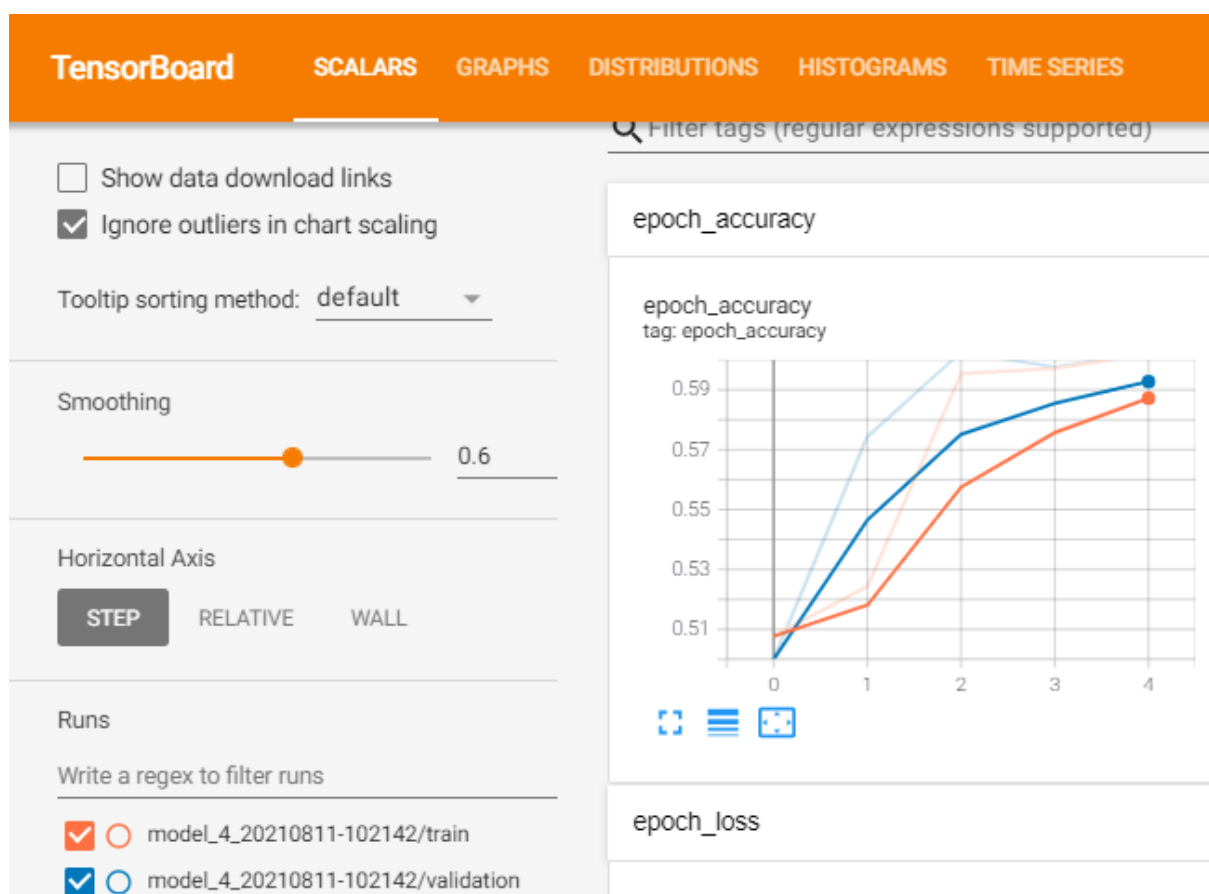
1. Here we are using Relu and HeUniform as weights initializers.

2. As observed in the previous model, Relu speeds up the convergence and helps with vanishing gradient problems.
3. Using it with HeUniform improves the test accuracy of the model to 0.589 which was achieved in just 6 epochs.

Model-4 (Custom)

1. Using tanh as an activation for every layer except the output layer.
2. using SGD with momentum as an optimizer.
3. using GlorotNormal() as initializer.

Accuracy vs Epoch Graph :



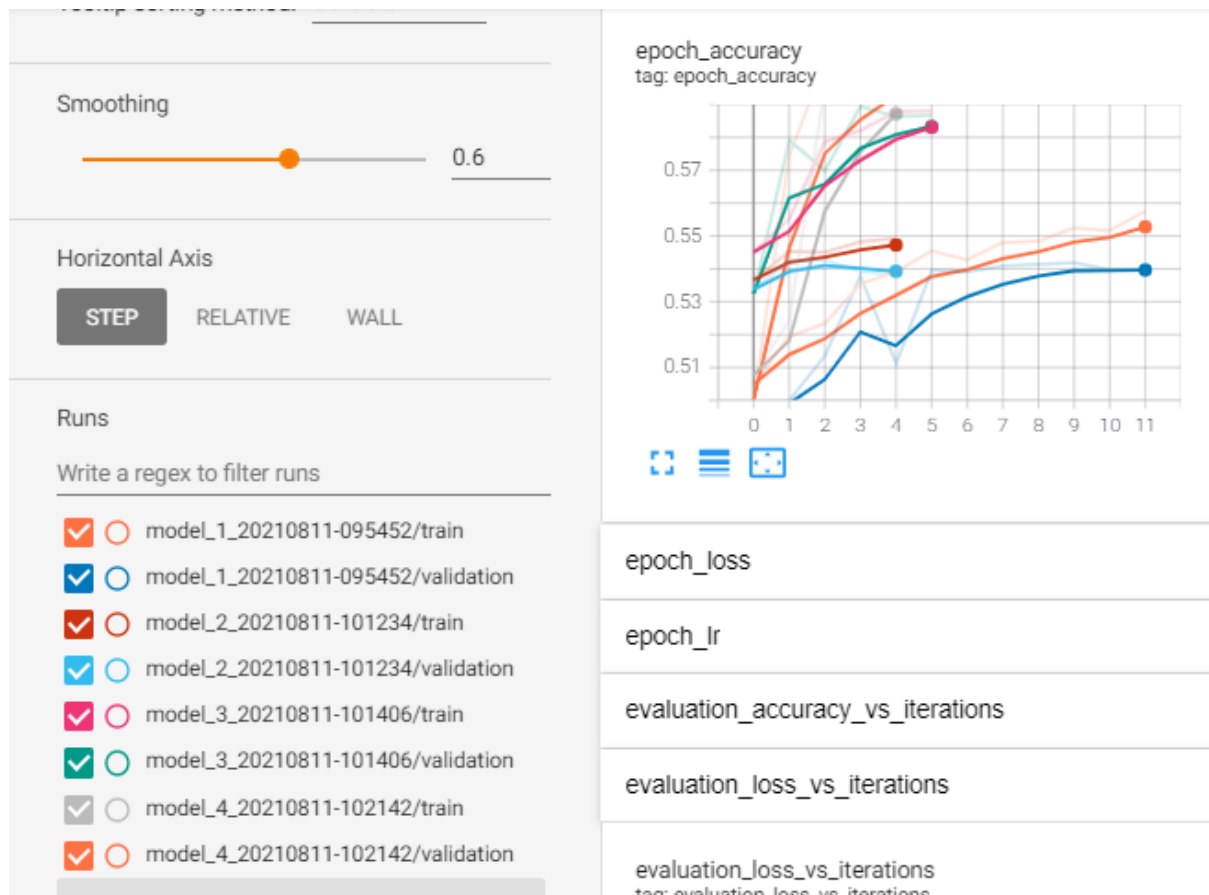
Observations

1. This model performs the best.
2. Here I have used tanh as an activation function as it is binary class classification.
3. However, to overcome the vanishing gradient problem and slow convergence, I have used Glorot / Xaviers heuristics.

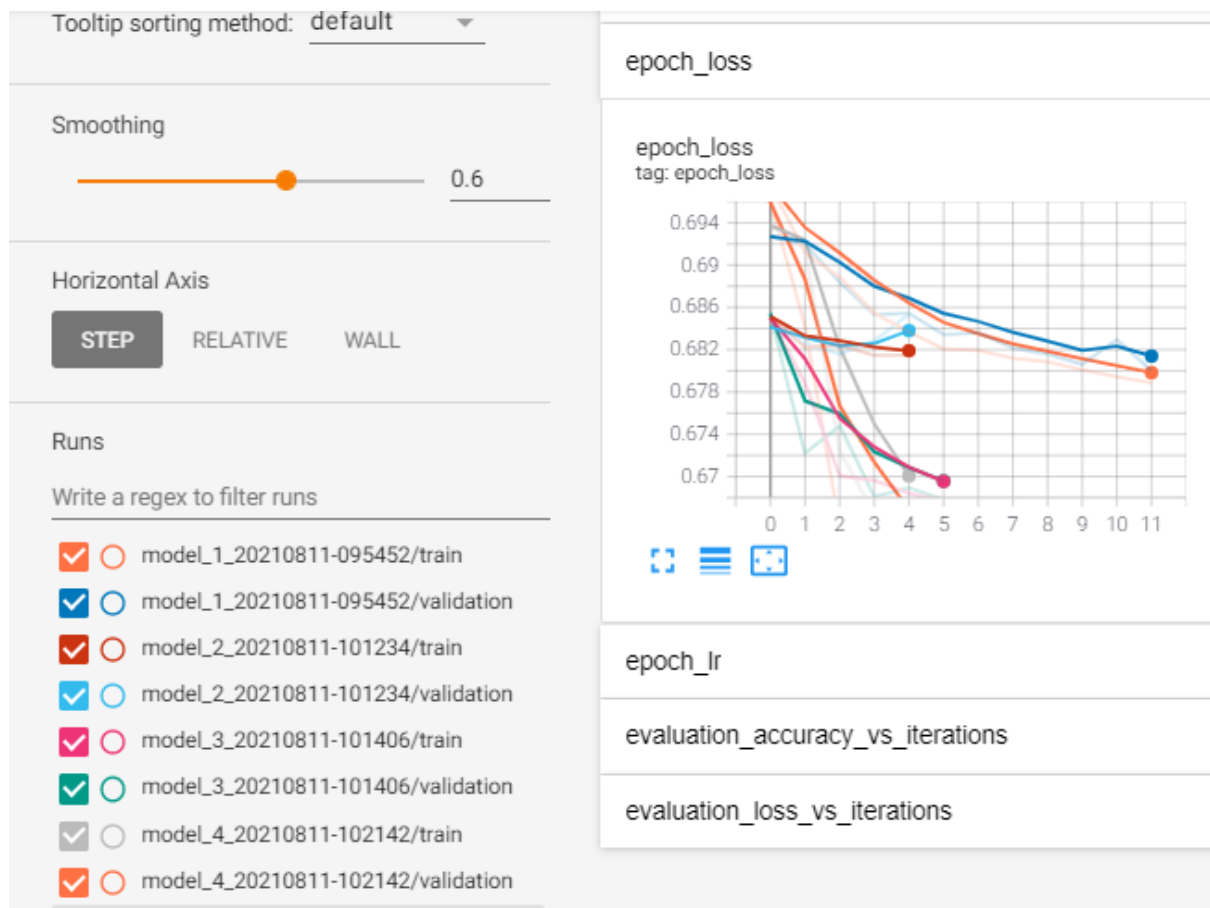
4. As a result the test accuracy is better as compared to other models and it is 0.60
5. Model achieved this accuracy within 5 epochs before plateauing off.

Combined graphs

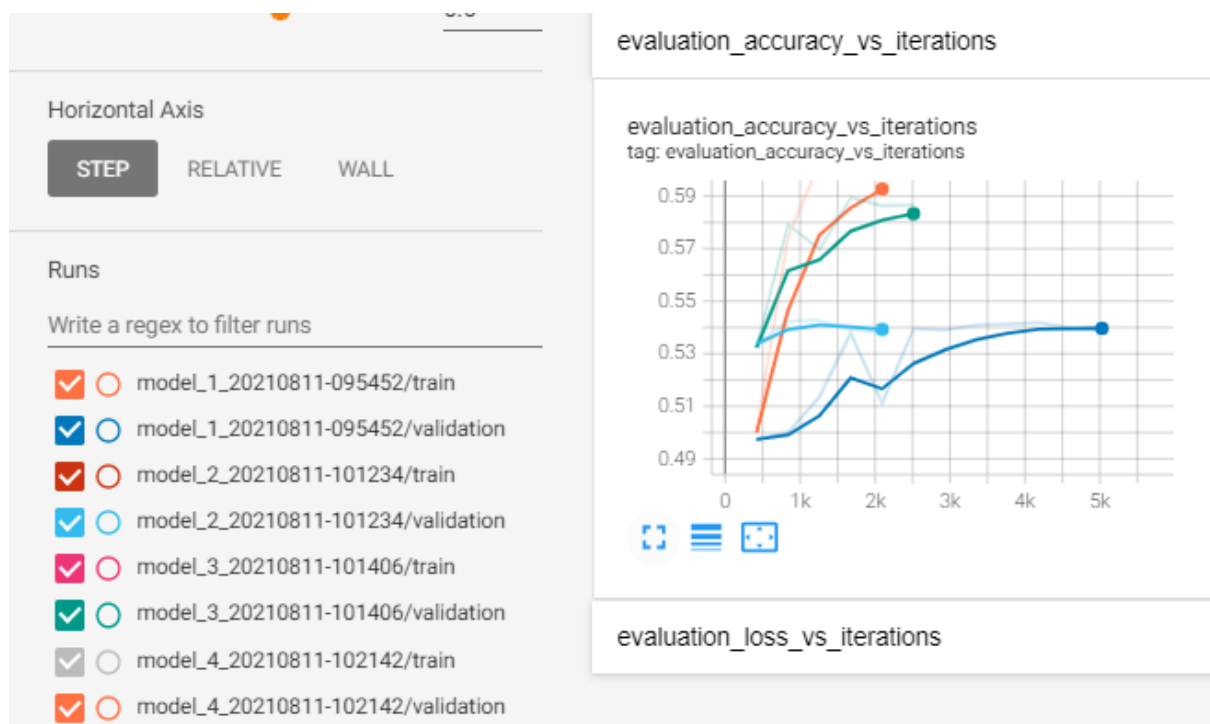
Train & Test Accuracy vs Epochs



Train & Test Loss vs Epochs



Test Accuracy vs iterations



Test Loss vs iterations

