Xlabel = [1,5,10,15,20,25,30,35,40,45,50,55,60,65,70,75,80,85,90,95,100]

Adam: [0.45763, 0.83, 0.84406817, 0.84838975, 0.846762, 0.852032956, 0.8516501, 0.85055, 0.8549172550, 0.855722375, 0.85384, 0.843596, 0.85344, 0.850130538, 0.8544031, 0.8561226, 0.871593, 0.85144095, 0.855948577, 0.85122618, 0.856454]

SGD(m=0.9) = [0.6921402, 0.83, 0.84406817, 0.846762, 0.852032956, 0.852032956, 0.8516501, 0.85055, 0.8549172550, 0.855722375, 0.85384, 0.843596, 0.85344, 0.850130538, 0.85440331, 0.8561226, 0.871593, 0.85144095, 0.855948577, 0.85122618, 0.856454]

SGD(m=0.8) = [0.6657678488, 0.8051370, 0.8389284, 0.84571221, 0.84756267, 0.847168, 0.85625669, 0.85555704, 0.860804256, 0.8583543, 0.8589860, 0.867118, 0.86698484, 0.8666589, 0.86837586, 0.86320.0., 0.84241810, 0.869826, 0.867632411, 0.868703858, 0.8701014]

Adagrad = [0.6379970290254808, 0.7894496988600398, 0.851253424714051, 0.8627262275995934, 0.846011989048035, 0.8661018072107464, 0.8551510100144373, 0.864344062516451, 0.8718083899814623, 0.8534955285075888, 0.8595995529524659, 0.862239975617787, 0.862627043660901, 0.8698519632871318, 0.8672679524212658, 0.864623352051099, 0.8650294481638516, 0.870097995315265, 0.8726112344768182, 0.8675029334802827]

RMSProp = [0.3291191464546527, 0.4918080570020013, 0.43498794370915705, 0.5838441679781767, 0.7463403755579366, 0.5817452528430777, 0.8365179721092124, 0.8180399801917451, 0.8454145088190996, 0.8269698980333701, 0.8371695686933892, 0.7957581475914745, 0.8547453822152182, 0.869872287683797, 0.8554992041112977, 0.866489847716382, 0.8670796400880978, 0.8506484952314738, 0.8628464391489793, 0.8635574197455511, 0.8648893165260956]

SGD(nesterov) = [0.7002808010155198, 0.8557790572477775, 0.8520047420776546, 0.8617436798709177, 0.866782907249554, 0.8628787995622069, 0.8636731965701104, 0.8586414480804553, 0.8589208637433143, 0.873631518350729, 0.8596824280592925, 0.8751195437174017, 0.8702186077236794, 0.8711319960855295, 0.8763131919893714, 0.8678721159978628, 0.8668664565972957, 0.8688247002347154, 0.864363982550693, 0.8770761070861137, 0.872899130432508]

SGD(no m) = [0.468269250101342, 0.7154090740204379, 0.7758618091539917, 0.7994153428796171, 0.817523036318296, 0.8085991143446851, 0.7949029618033869, 0.8039057427553494, 0.8132992803625051, 0.8201306383961027, 0.8070245228688597, 0.8237062171503006, 0.8214190189550733, 0.8203200873945968, 0.8169636854098913, 0.8073268005338567, 0.8155641077229628, 0.8144292535189243, 0.8085751711319016, 0.8174121074342768, 0.8120615302343881]

In the FCN network structure, the best performance is achieved by using 3x3 kernel size with 1 padding and 1 stride. After each five convolutional blocks, there is a batch normalization layer to avoid overfitting and maxpool layer to extract the features. And the output size of each blocks are (4,1,96,96), (4,16,48,48), (4,32,24,24), (4,128,6,6) and (4,256,3,3). And upsampling layer is used after convolutional part is finished. And to make the prediction more accuracy, we use 2x, 2x, and 8x as upsampling factor instead of 32x directly. Because it adds the feature maps which is the output of 3rd and 4th convolutional part. Finally, with the best loss function Adam with 0.01 learning and best optimizer which is found by other group member, we obtain over 0.93 accuracy for training, over 0.88 accuracy for validation. And for test set, we submitted to Kaggle, it achieved 0.88431, and it’s the best accuracy in these 2 models we implemented.

For the optimizer, each algorithm doesn’t have a large difference between them, the performance is quite well. We choose currently the most popular one – Adam as our optimizer. It is an alternative stochastic gradient descent optimization algorithm for training deep learning models, and it combines the best features of the AdaGrad and RMSProp algorithms to provide an optimization algorithm that can deal with sparse gradients on noise problems. In addition, Adam’s memory requirement is low and suitable for a large and high dimension dataset, especially image problem.