Deep Learning - Assignment 1

Introduction

In this assignment we requested to implement Feedforward Neural Network for multiclass problems. The activation functions used for all layers but the last is ReLU, the final layer will apply the softmax activation function.
In this report we will present our work and it will include: data details, experimental settings, dropout explanation and results.

Data

We used our network to classify MNIST dataset. we used the predefined division between train and test sets, randomly chose 20% of the train set to validation set. Input matrix size (m,784), while m is the number of samples.

Experiments

Our main goals are defining most efficient hyperparameters and analyzing the network performance (memory and running time) by training our network in the following ways:

1. batchnorm option OFF
2. batchnorm option ON
3. dropouts option ON

while dropouts and batchnorm implemented as described in the class lectures.

<u>Configurations</u>: We would like to test the following batch sizes: 50, 100,500. We train our model with a max number of 900 epochs and calculate the cost one time in 100 epochs. The keep-rate we chose for the dropout is 0.8. The learning rate was 0.009.

<u>Stopping criteria</u>: The condition to stop training the model is by calculating the validation set cost compared to the validation set cost on the last cost calculation (the validation set cost calculation executed one time in 100 epochs). We have set a requirement of 5% increase in the cost as our stopping criteria because we recognized it as optimal.

Dropout Adjustments

To enable Dropout we modified both our forward and out backward.
Given a drop-rate p for the drop out and the required length, we defined a random vector:

DropOut = $bernoulli$ (1$-p$,len) / (1$-p$)

This random is generated in each forward for each layer except for the final layer.

In forward we masked our activation A using the DropOut vector:

A = A * DropOut

We incorporated the dropout in the backprop by applying it on A's derivative:

dA = dA * DropOut

Our forward is implemented in the new function:

"linear_activation_forward_da"

Our changes regarding to Dropout were incorporated in

"linear_activation_backward"

Results and Discussion

We will show here the stature of the results. Each experiment has a different configuration.
The parameters in our configurations are:

- Batch size: Number of samples in each batch of the training
- Batchnorm: Use the feature described in 1.H in the assignment file
- Dropout: Use the Dropout feature as instructed in the bonus question

The output of the evaluation returns

- A list of the train\validation losses that were generated during training (each 100 epochs)
- Accuracy of training data
- Accuracy of validation data
- Accuracy of test data
- Runtime of the entire procedure
- A plot showing cost to epoch

Because we bring extensive evaluations so we will first write our discussion that is based on those evaluations.
Here we will draw conclusions from the results that are attached below:

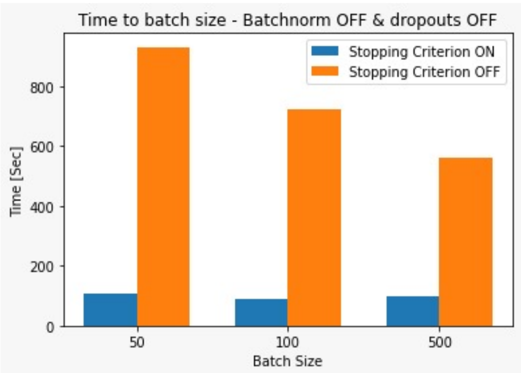
Stopping criterion

We used the instructed technique of early stopping. We see that in the vast majority of experiments the stopping criterion was met within the first 200 epochs. This suggests that the architecture was quick to converge and did not require more than 200 to reach optimal validation accuracy. Beyond the requested experiments, we ran experiments with the stopping criterion; in the majority of those experiments the network improved training accuracy but decreased validation accuracy (overfitting).

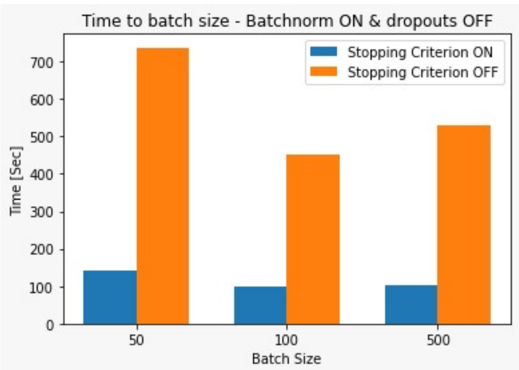
Overfitting

Overfitting occurs when a model fits the training data significantly more than test or validation data. We can observe this phenomena in all our experiments in which we did not use the stopping criterion.

Running time and batch size comparison

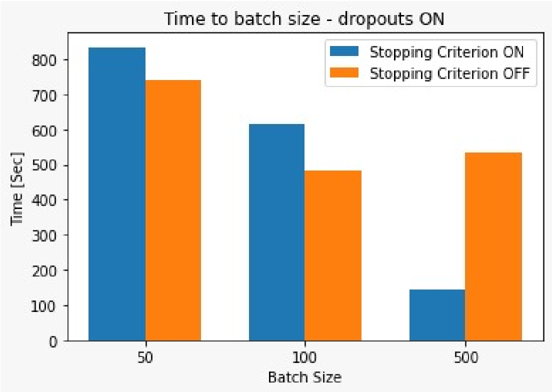


We can see in this figure the running time with batchnorm off and dropout off is much shorter. Using stopping criterion resulted in accuracy of 92-94% accuracy for all batch sizes. Specifically for stopping criterion enabled, the test set accuracy is 93% for all batch sizes, and there is no significant difference of time between the different batch sizes.



As to batchnorm on and dropouts off, here the running time is shorter, also while using stopping criterion. The accuracy of the test set for all batch sizes is 94%. There is no significant time difference between the different batch sizes.

The time difference between batchnorm on and off is almost the same for all batch sizes, the accuracy for the test set is 93-94% for both cases while stopping criterion is on.



As to the dropouts option while the batchnorm option is on: the time difference is insignificant on 50,100 batch sizes. Stopping criterion on led to a major time difference when batch size is 500. But it is important to say the test set accuracy for batch size 50 without stopping criterion is much higher - 89% compared to 10%. For 100 70% compared to 64%, and for 500 75% compared to 78%.

It seems like the training with dropout is much slower, so it might be better to trade in running time and choose a smaller batch size for high accuracy. If 80% accuracy is enough, we suggest to consider batch size 500 with more training epochs.
The running time with dropout compared to without dropout is significantly longer except for when the stopping criterion is on and batch size is 500. But the accuracy in this case is lower with 78% compared to 94%.

Batchnorm effect
We see that using applying a batchnorm changes the training costs but results in similar accuracy. We can see in the plots that the difference between the validation cost and the training cost is less significant when applying batch normalization. However, we see that the accuracy results on the test set are similar.

Dropout effect
We see a drop in accuracy after applying the dropout. We see that the training accuracy also drops. This drop can be explained by the small size of the network. Because the network is too small, the dropout harms its ability to learn the training data too much. This may be described as low bias.

Results:

Batchnorm OFF & Dropouts OFF

Stopping Criterion OFF:

BATCH_SIZE=50

Costs:

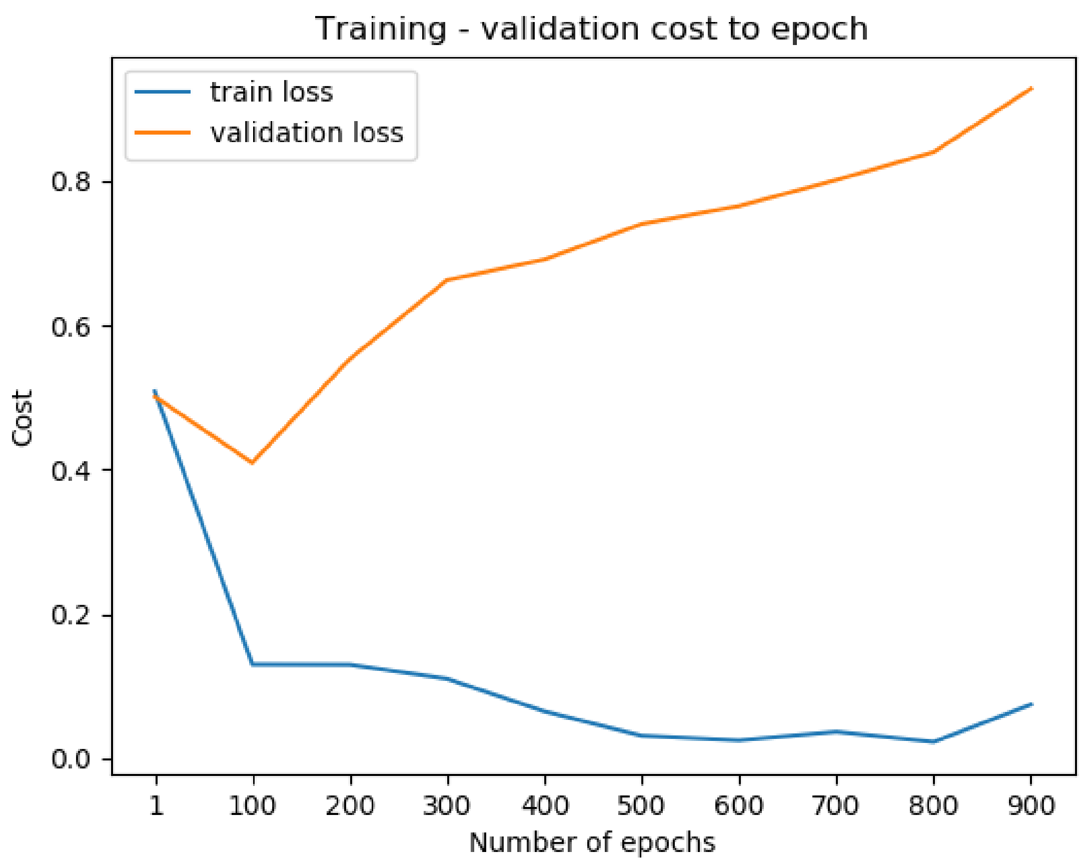
{'train': [0.5082789842679647, 0.1303328257451941, 0.12999565166306662, 0.11041366641284221, 0.06532208224518155, 0.03162423638880292, 0.025145109289119727, 0.03705437183691637, 0.023020493131092584, 0.07478374350717373], 'validation': [0.5008389727461765, 0.4093661619318791, 0.552719422940617, 0.6621132631321681, 0.6908305282896151, 0.7399323726595646, 0.7648271423846214, 0.800871515153788, 0.8394299160235605, 0.9272561505849622]}

Train Prediction: 0.9893125

Validation Prediction: 0.9336666666666666

Test Prediction: 0.9352

Time: 930.3675806522369 sec



BATCH_SIZE=100

Costs:

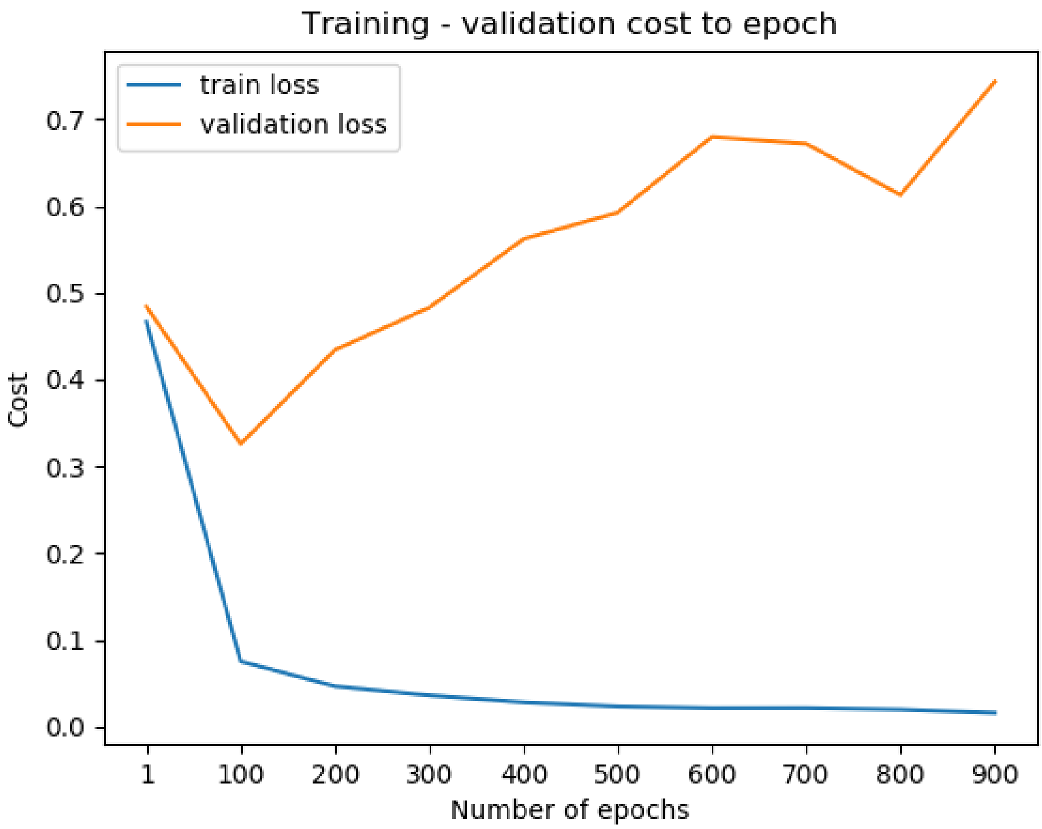
{'train': [0.46664215100084433, 0.07510774903491142, 0.04628083097406596, 0.03585415158321505, 0.027877406144754313, 0.023172413783988113, 0.02122867560458811, 0.021316175687171278, 0.019628045598432415, 0.015800965024977954], 'validation': [0.48408390774408994, 0.32568378779393553, 0.4340138180904153, 0.48285469694527433, 0.5618087147762039, 0.592406423624665, 0.6796668949247384, 0.6717282648503659, 0.6127970200055952, 0.7432024241356756]}

Train Prediction: 0.9875

Validation Prediction: 0.9346666666666666

Test Prediction: 0.9426

Time: 720.7830522060394 sec



BATCH_SIZE=500

Costs:

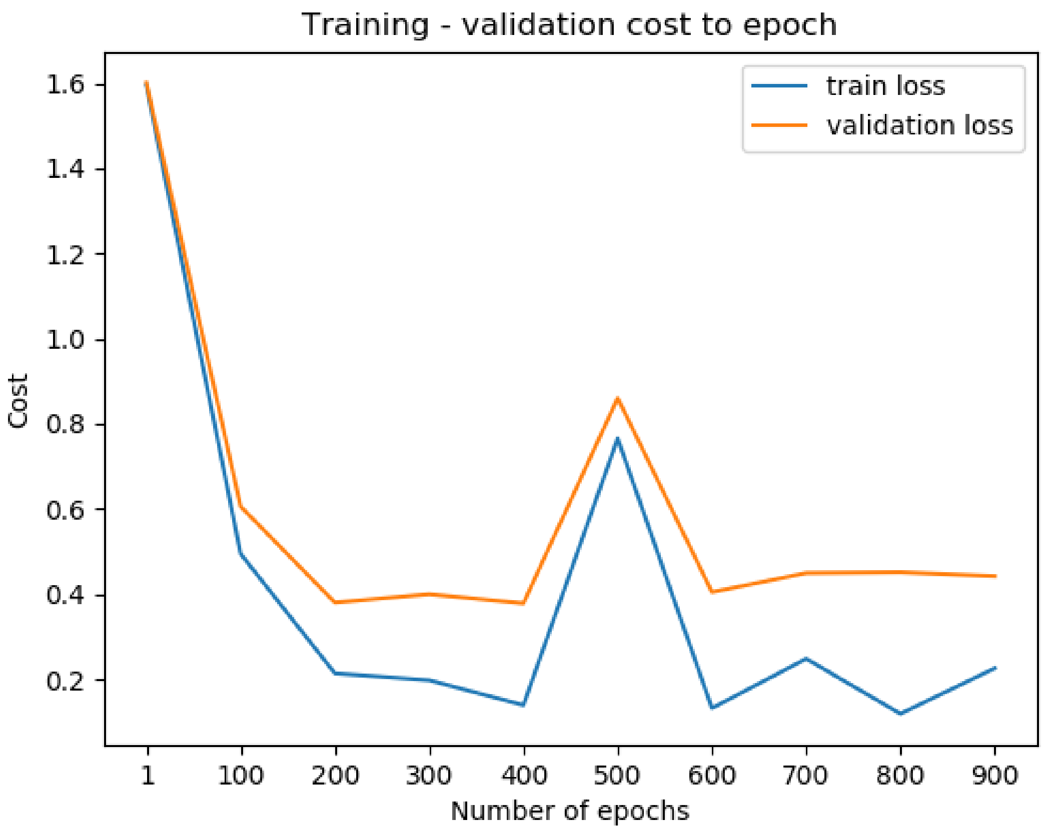
{'train': [1.595144292022794, 0.49457588126781116, 0.21380678012629672, 0.19768448524404278, 0.13933800297303506, 0.7655448848310842, 0.13206283495361806, 0.24808253719296625, 0.11899325623357487, 0.225784131106805], 'validation': [1.600544316012071, 0.6057243911511208, 0.38007817043175046, 0.39946564577903293, 0.3782840874705025, 0.8603941947032128, 0.40454371028228364, 0.44940823184536355, 0.4512068928012909, 0.44202044481463953]}

Train Prediction: 0.9536041666666667

Validation Prediction: 0.9155

Test Prediction: 0.9265

Time: 560.0103430747986 sec



Stopping Criterion ON:

BATCH_SIZE=50

epochs: 200. no improvement - breaking

acc_val_cost > last_cost : 0.5836196209518164>0.4349795984409255

Costs:

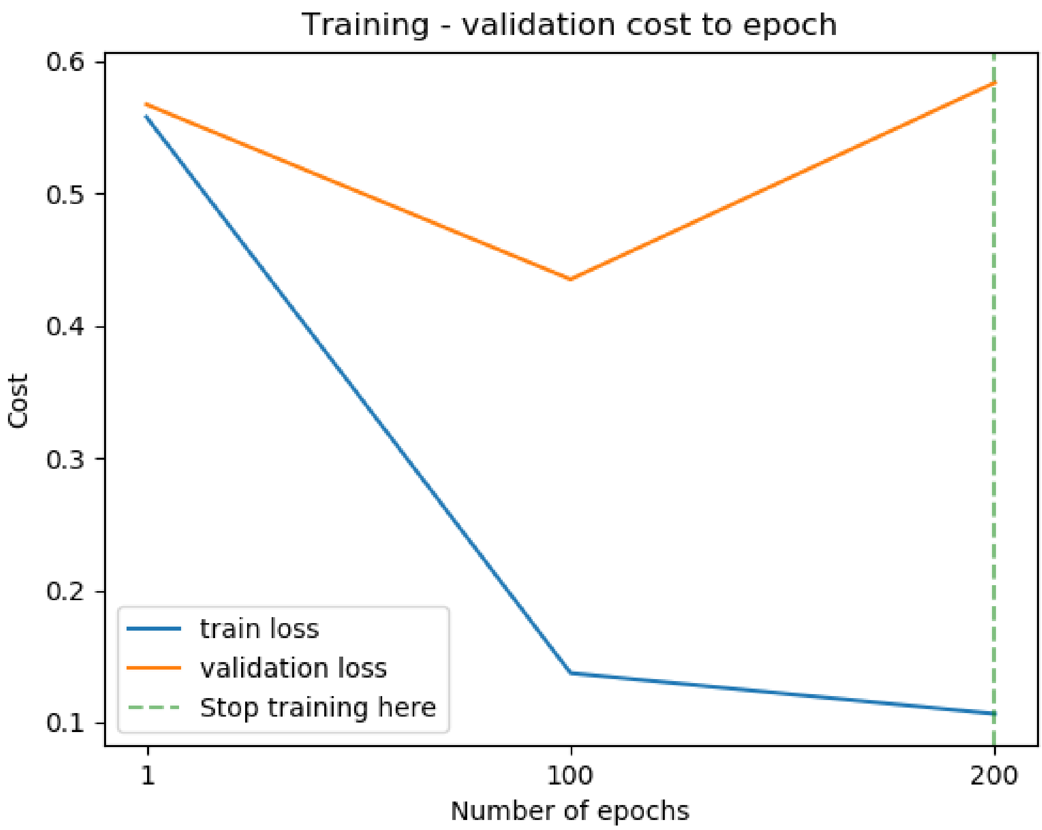
{'train': [0.5578557820336871, 0.13719664268257192, 0.10641728881852291], 'validation': [0.5672408618274775, 0.4349795984409255, 0.5836196209518164]}

Train Prediction: 0.97125

Validation Prediction: 0.9299166666666666

Test Prediction: 0.9324

Time: 107.31928372383118 sec



BATCH_SIZE=100

epochs: 200. no improvement - breaking

acc_val_cost > last_cost : 0.5188385715292886>0.33259655924177195

Costs:

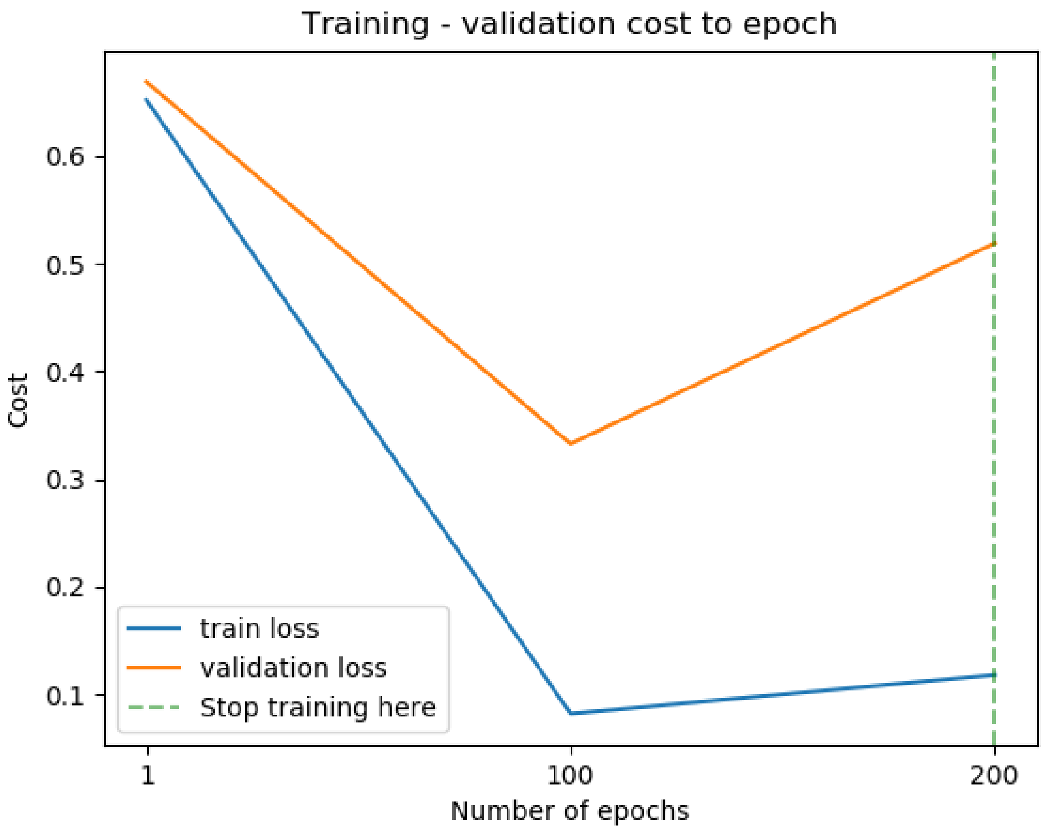
{'train': [0.6518731877528445, 0.08222999096678661, 0.11816129757283238], 'validation': [0.668656451819238, 0.33259655924177195, 0.5188385715292886]}

Train Prediction: 0.97175

Validation Prediction: 0.93075

Test Prediction: 0.9372

Time: 87.66167783737183 sec



BATCH_SIZE=500

epochs: 200. no improvement - breaking

acc_val_cost > last_cost : 0.2945259775547916>0.2329232734742629

Costs:

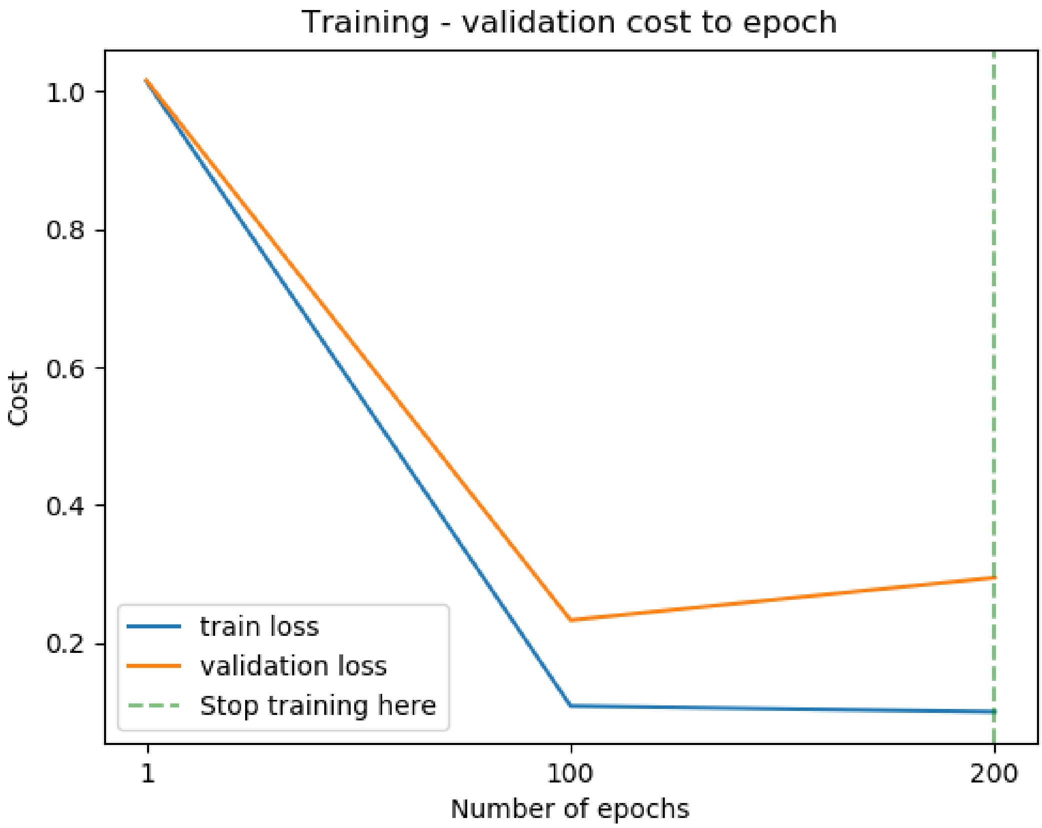
{'train': [1.0148302153752957, 0.1084084599792513, 0.10003646401987294], 'validation': [1.0163809929593695, 0.2329232734742629, 0.2945259775547916]}

Train Prediction: 0.9715625

Validation Prediction: 0.94275

Test Prediction: 0.9362

Time: 97.9766206741333 sec



Batchnorm ON & Dropouts OFF

Stopping Criterion OFF:

BATCH_SIZE=50

Costs:

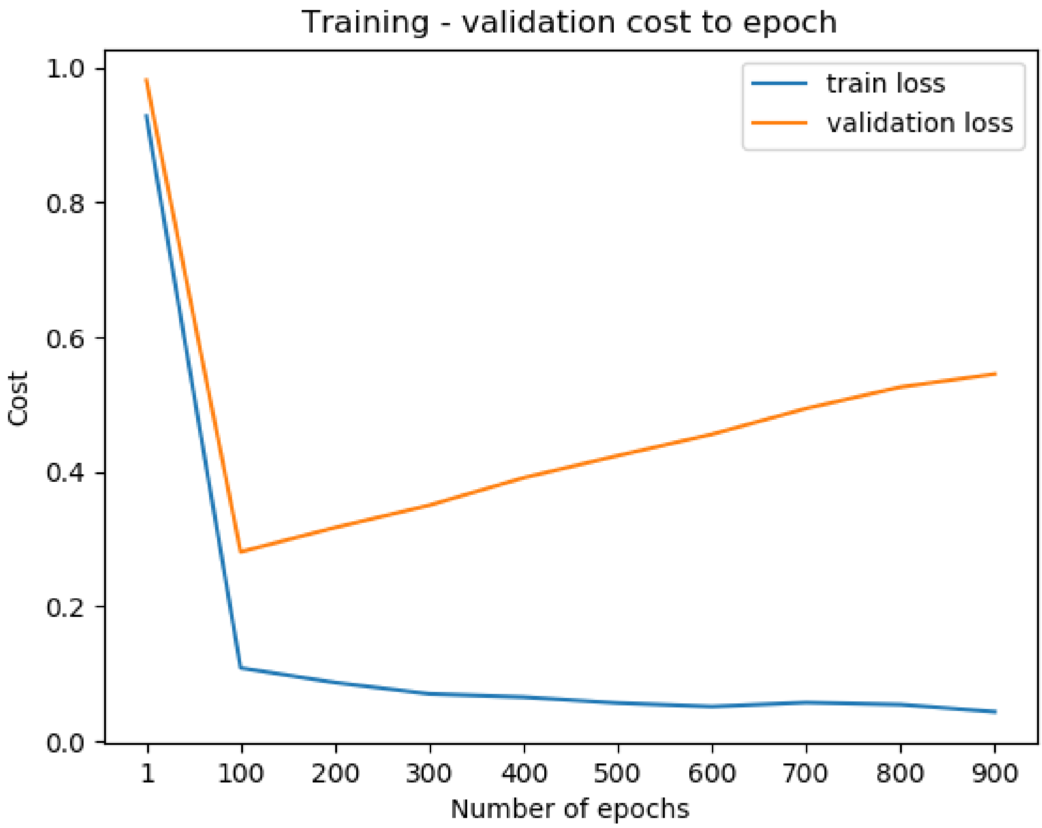
{'train': [0.9284862229936838, 0.10884840942158588, 0.08711700994588477, 0.07029651489462674, 0.06559522216779799, 0.05690560586869969, 0.051551884676878955, 0.057577946452081225, 0.05450759580243267, 0.04390196605512011], 'validation': [0.9814249463820257, 0.2813517314239972, 0.3171424256716902, 0.35035166121557465, 0.39124621000702214, 0.42397047703965746, 0.4557704398313779, 0.49421899737164865, 0.525794629429015, 0.5452266398238427]}

Train Prediction: 0.9846041666666666

Validation Prediction: 0.9339166666666666

Test Prediction: 0.9371

Time: 737.4231295585632 sec

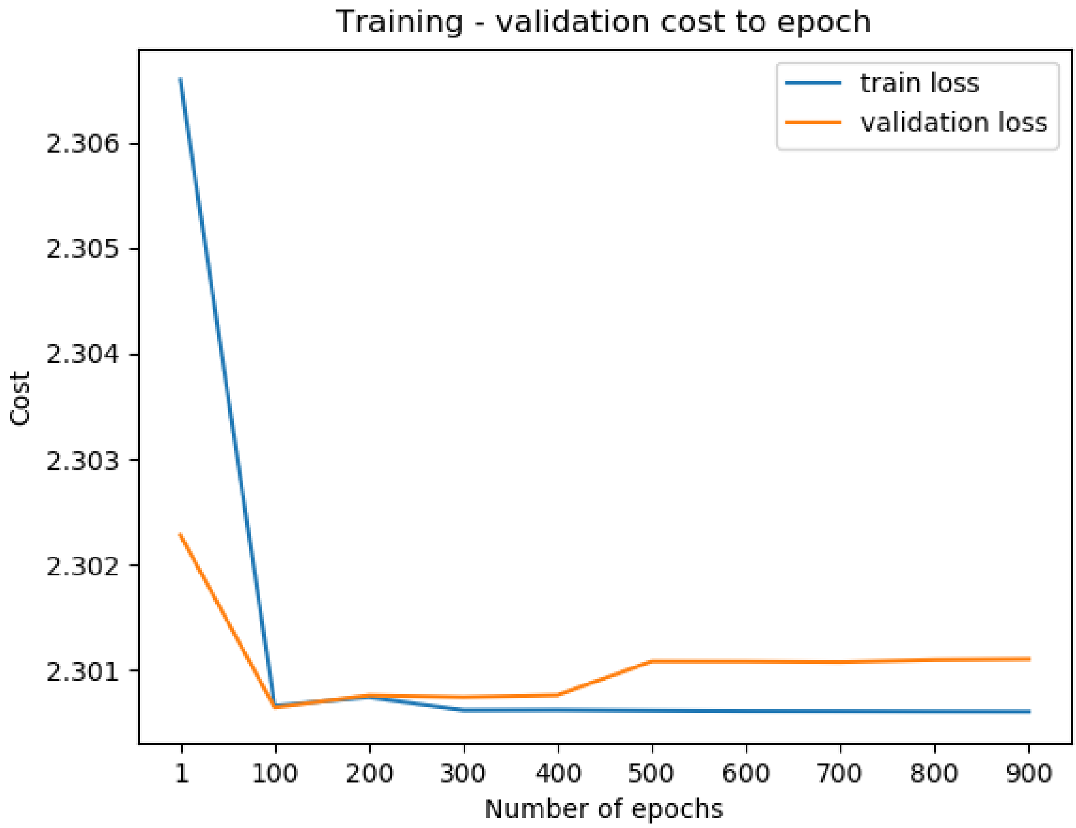


BATCH_SIZE=100

Costs:

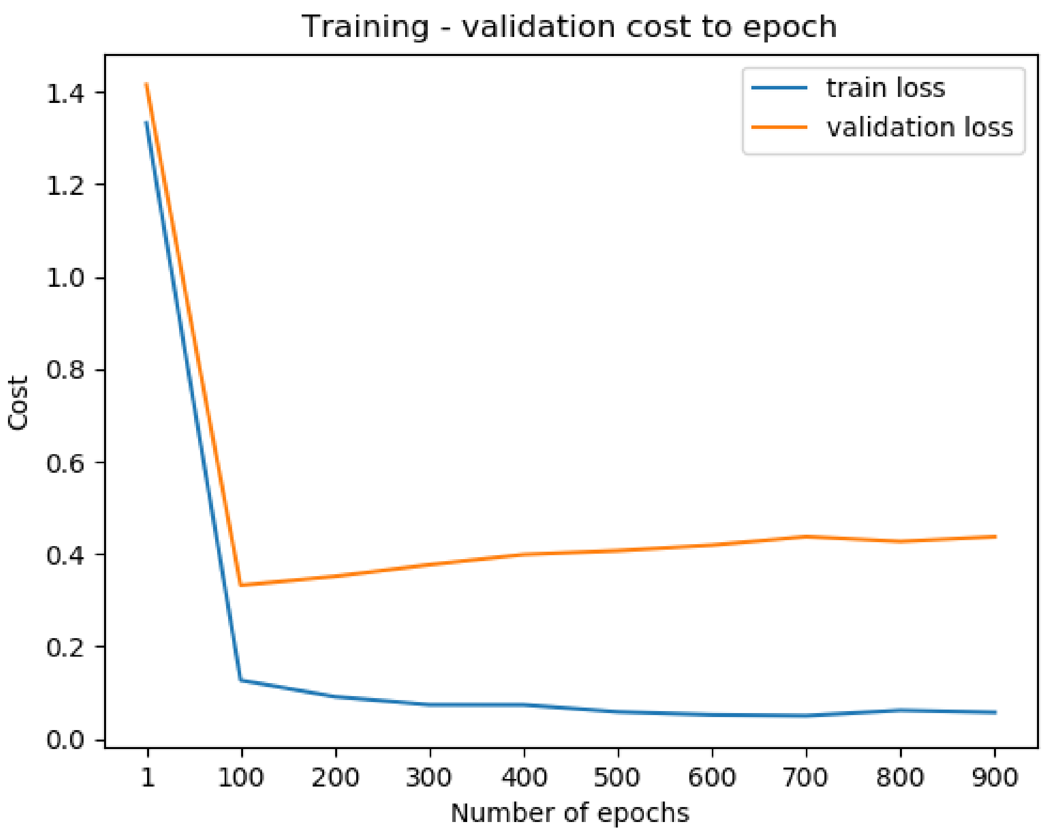
{'train': [2.3065918559681693, 2.300662001080181, 2.3007416875143485, 2.3006195412209935, 2.3006211493158744, 2.300615808843463, 2.3006109595053563,

2.300609557653545, 2.3006069746986615, 2.300605444134483], 'validation':
[2.30227911747127, 2.300644478335962, 2.300760631015892, 2.3007427011928234,
2.300762684317441, 2.301081816126308, 2.3010806419013123, 2.301074617333286,
2.301096372981096, 2.3011030686429828]}
Train Prediction: 0.11254166666666666
Validation Prediction: 0.11316666666666667
Test Prediction: 0.1136
Time: 451.45704221725464 sec



BATCH_SIZE=500
Costs:
{'train': [1.3330097749564593, 0.1270348416905377, 0.09125814654286177,
0.07381731775199306, 0.07358864225178784, 0.058401902023458394,
0.05192817661033889, 0.04996933118481673, 0.06182277650009774,

0.05731872045137752], 'validation': [1.4157093150137003, 0.33235259200296735, 0.35177912624885105, 0.376686478273422, 0.39881299723566344, 0.4071556020685893, 0.41919494663743595, 0.4376198312760928, 0.4272952053444901, 0.437420790643502]}
Train Prediction: 0.983375
Validation Prediction: 0.93675
Test Prediction: 0.9438
Time: 530.8337666988373 sec



Stopping Criterion ON:

BATCH_SIZE=50

epochs: 200. no improvement - breaking

acc_val_cost > last_cost : 0.31828026717959285>0.26448782224983136

Costs:

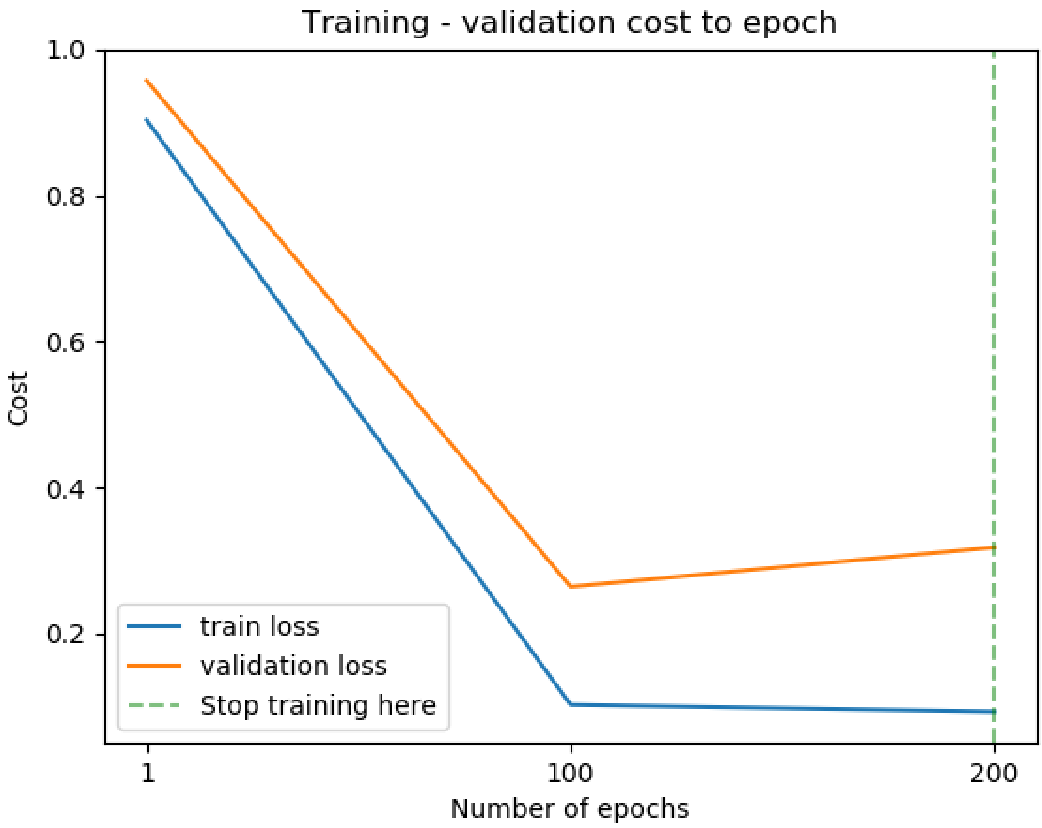
{'train': [0.9030986276313538, 0.10286208453416473, 0.0935505248343101], 'validation': [0.9571748489477532, 0.26448782224983136, 0.31828026717959285]}

Train Prediction: 0.971

Validation Prediction: 0.94

Test Prediction: 0.9418

Time: 142.55429935455322 sec



BATCH_SIZE=100

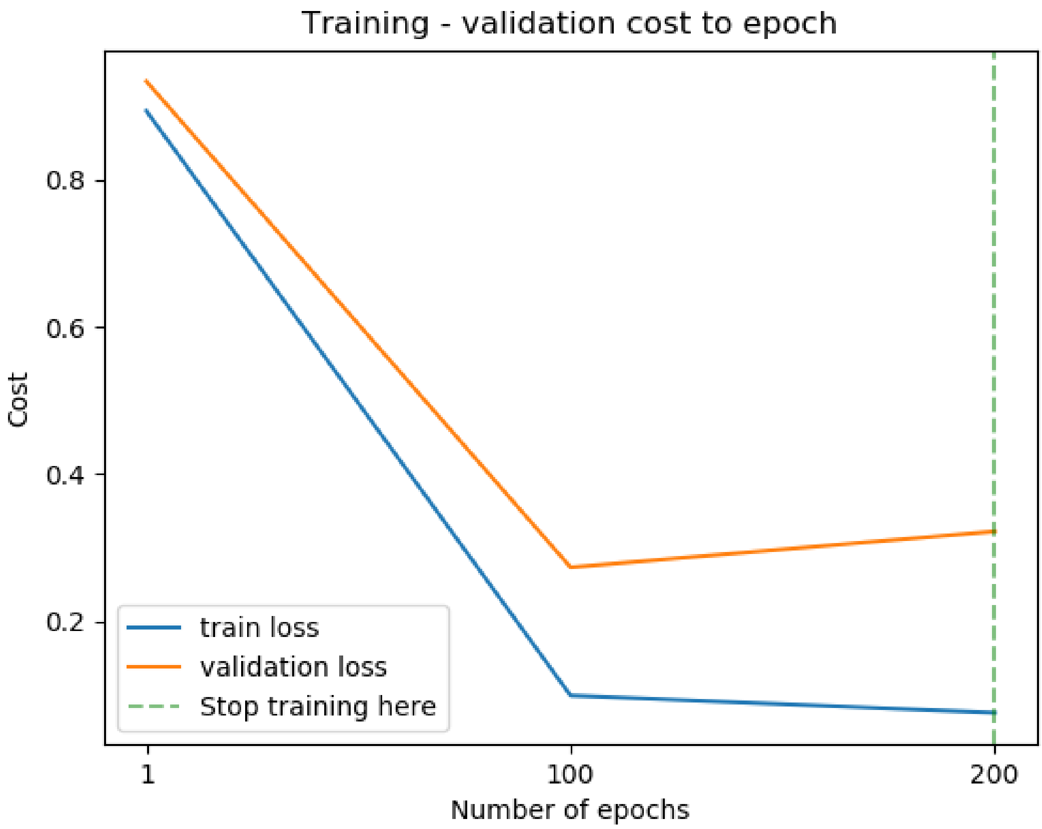
epochs: 200. no improvement - breaking

acc_val_cost > last_cost : 0.32208372804150803>0.2733789476130033

Costs:

{'train': [0.8933230062828992, 0.09922469351506741, 0.07593202891244079], 'validation': [0.9335287509218375, 0.2733789476130033, 0.32208372804150803]}

Train Prediction: 0.9756041666666667
Validation Prediction: 0.943
Test Prediction: 0.9438
Time: 99.85729050636292 sec

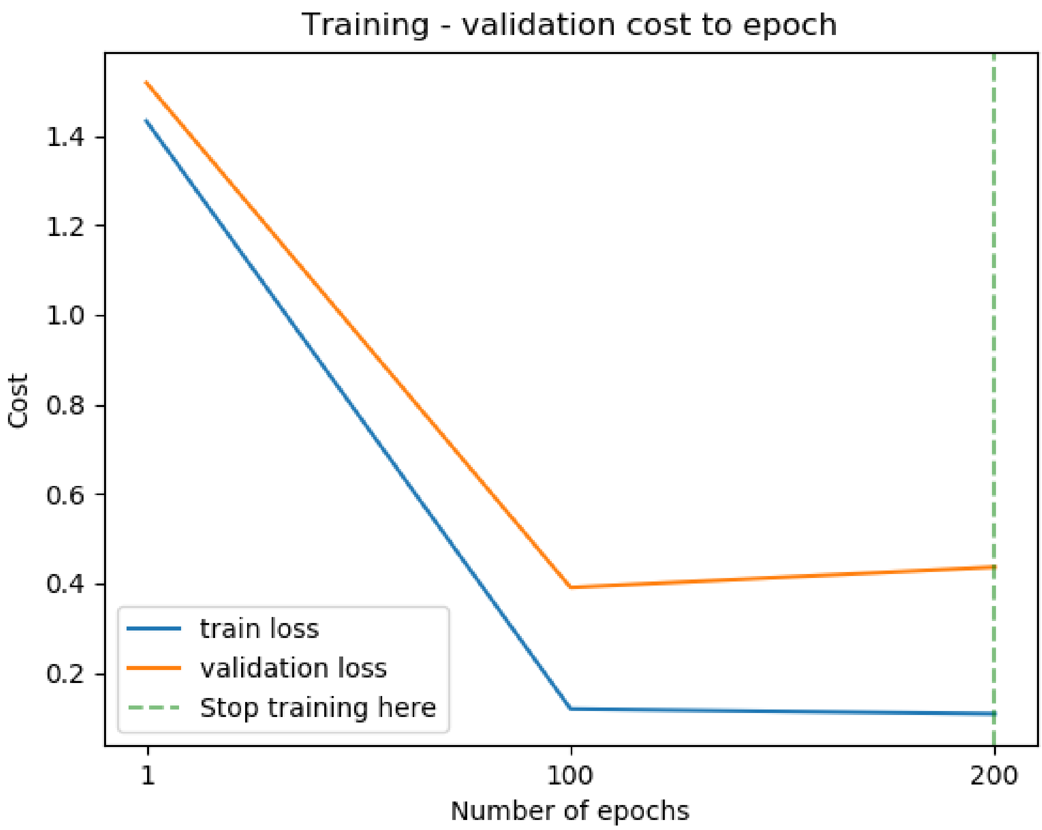


BATCH_SIZE=500
epochs: 200. no improvement - breaking
acc_val_cost > last_cost : 0.43598827064201856>0.39042315864394267
Costs:
{'train': [1.4324990139360179, 0.11909334969485591, 0.10825502986893844], 'validation': [1.5181025628915343, 0.39042315864394267, 0.43598827064201856]}
Train Prediction: 0.9683958333333333

Validation Prediction: 0.9354166666666667

Test Prediction: 0.9387

Time: 102.97545337677002 sec



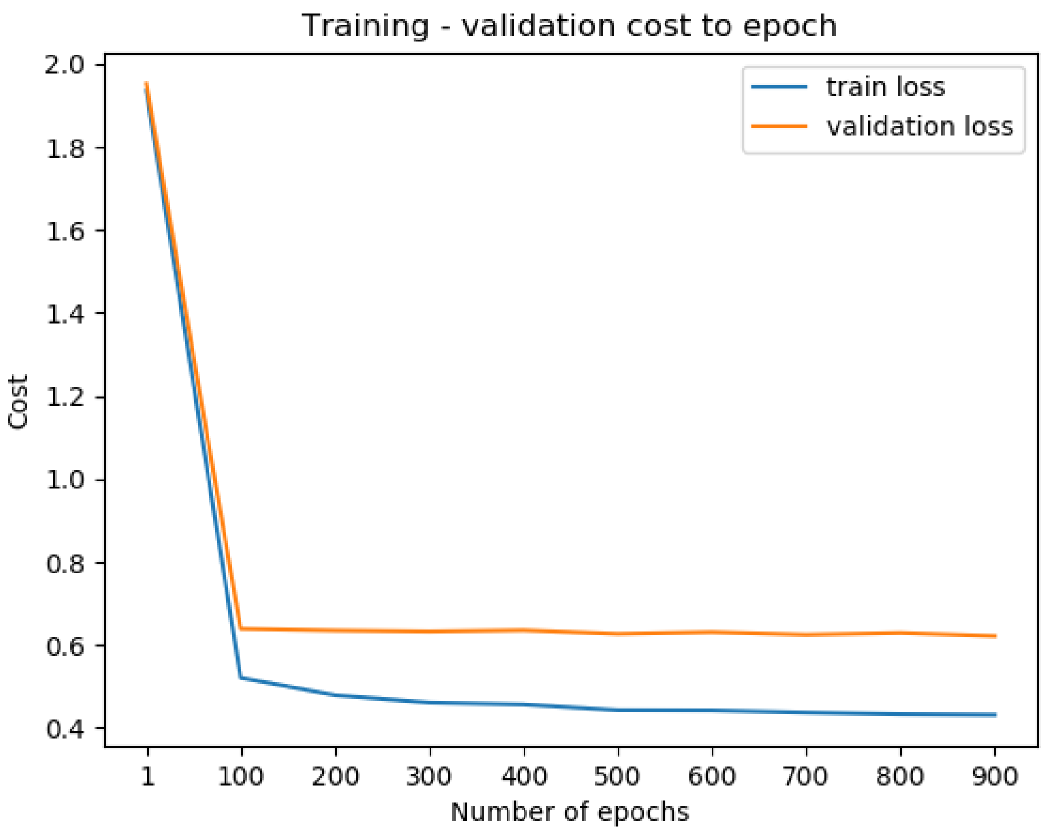
Batchnorm ON & Dropouts ON, keep_prob = 0.8

Stopping Criterion OFF:

BATCH_SIZE=50

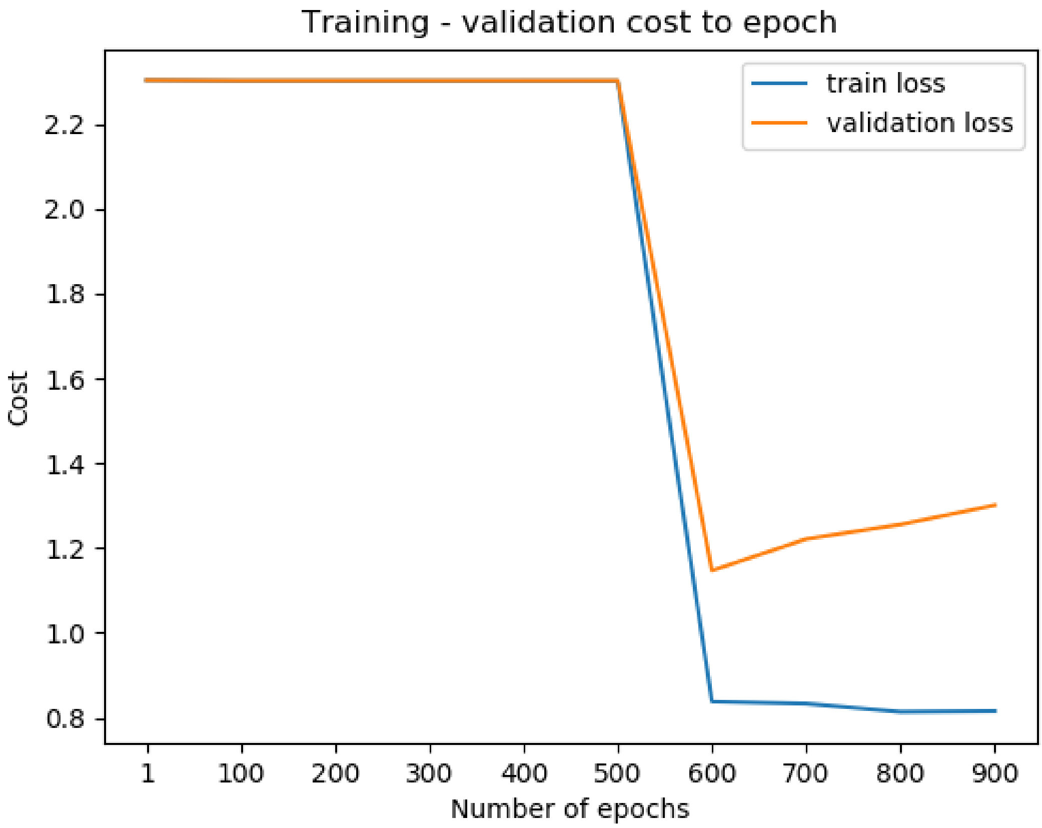
Costs:

{'train': [1.9356825724364153, 0.5209604186237891, 0.47889678851683637, 0.46097136074818834, 0.456716447692823, 0.4428530077707783, 0.4423450066325796, 0.4373086211596905, 0.4332648824476149, 0.43173713623856863], 'validation': [1.9525579763730183, 0.6390568439256422, 0.6349890618030624, 0.6326153756015156, 0.6354966881777185, 0.6267372771597458, 0.6309016594205729, 0.6247566032607696, 0.6290353729624704, 0.6216698748175982]}
Train Prediction: 0.9128333333333334
Validation Prediction: 0.8910833333333333
Test Prediction: 0.8951
Time: 739.0075478553772 sec



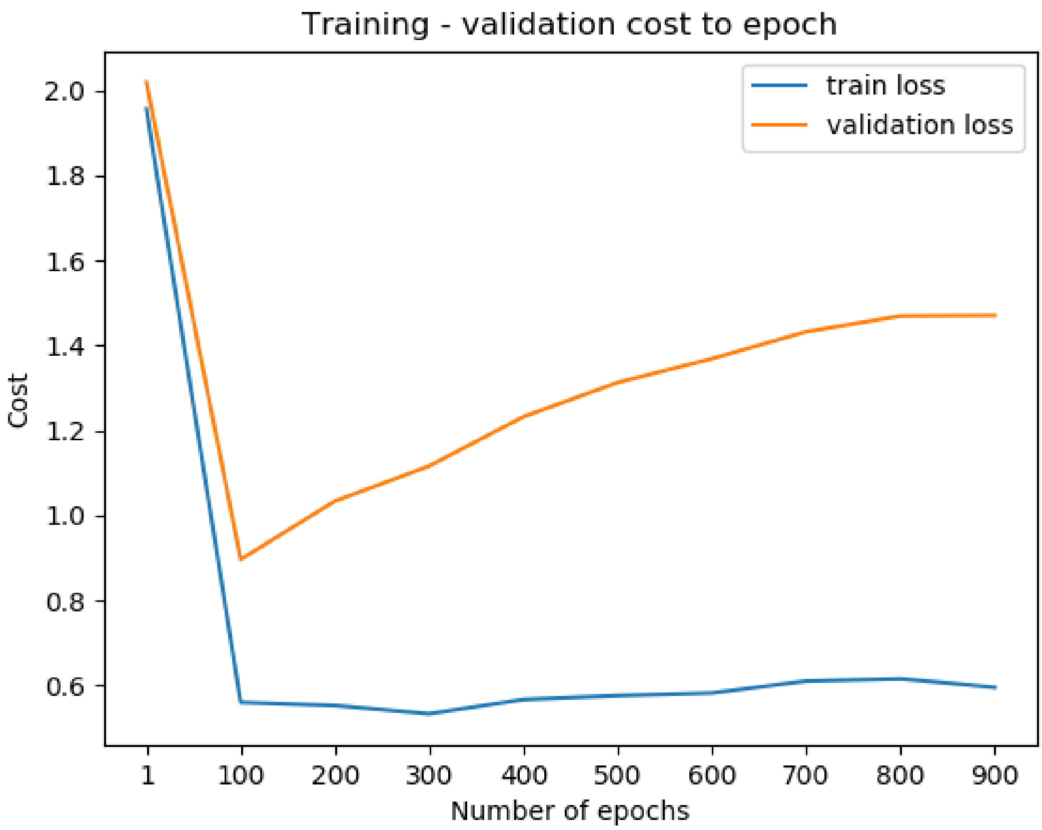
BATCH_SIZE=100
Costs:
{'train': [2.3025566339108012, 2.301174172886774, 2.3011689668686643, 2.3011622330975556, 2.3011633114658436, 2.301054796442107, 0.838544528080757, 0.8334765654873347, 0.8143098345071359, 0.8160076781843274], 'validation': [2.3026330271185613, 2.3017092113572355, 2.301693058634869, 2.3017102393795166,

2.30170928841952, 2.301748439687659, 1.1470438333433142, 1.2213362804072763, 1.2554519837393447, 1.3010386354132082]}
Train Prediction: 0.7067083333333334
Validation Prediction: 0.6926666666666667
Test Prediction: 0.6972
Time: 482.5156424045563 sec



BATCH_SIZE=500
Costs:
{'train': [1.9561248534766489, 0.5608504009902994, 0.5533607047375741, 0.5340542712921504, 0.5674291318045365, 0.5764308160729893, 0.5825779117607293, 0.6106357733206484, 0.6160482759634033, 0.5959350317686666], 'validation': [2.0182179763120986, 0.8967102384557792, 1.033809420076585, 1.1161751608481596,

1.2321244455272653, 1.3124922143239455, 1.3684364268834754, 1.432017386945704, 1.4688688991816101, 1.4702160366343344]}
Train Prediction: 0.759875
Validation Prediction: 0.7525
Test Prediction: 0.7493
Time: 534.5833220481873 sec



Stopping Criterion ON:

BATCH_SIZE=50

Costs:

{'train': [2.302307168429934, 2.3013807245361484, 2.3013804288758752, 2.301380387749918, 2.30138194774789, 2.301380636420932, 2.30138129675889,

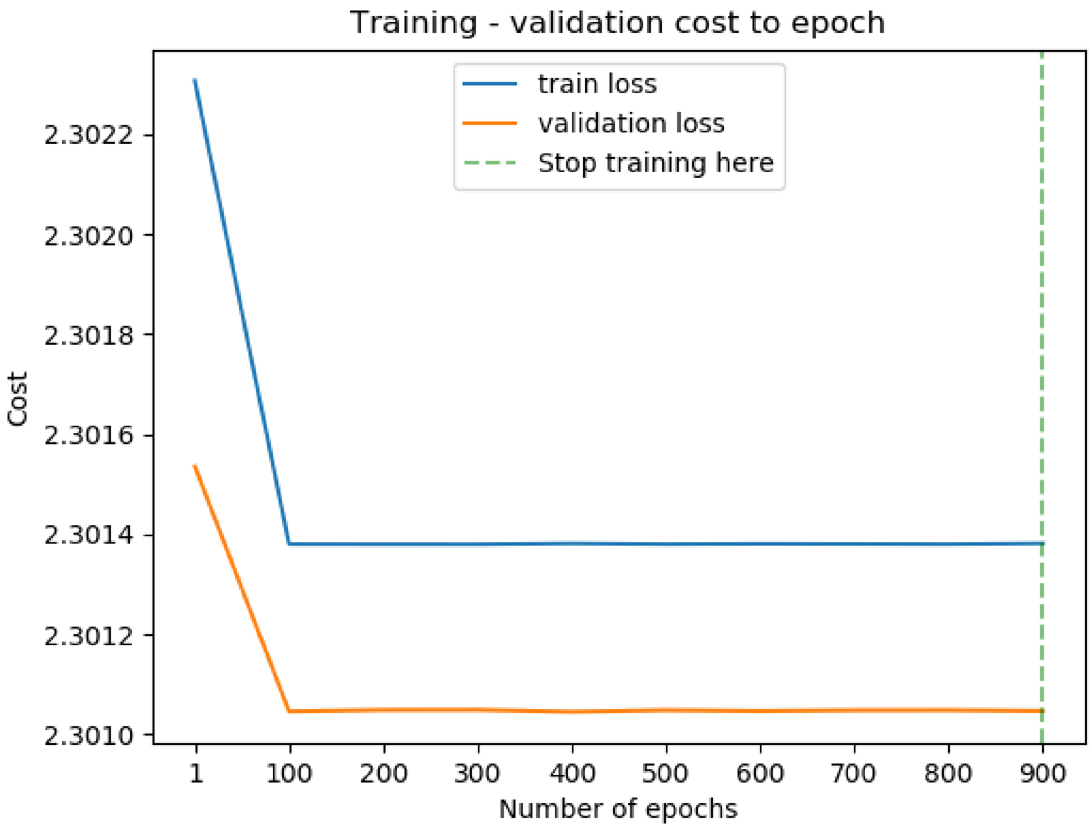
2.3013810167631843, 2.3013806073329564, 2.3013820334712727], 'validation':
[2.3015361568680577, 2.301046148325889, 2.3010489948445327, 2.301049194146403,
2.3010453594383797, 2.3010485582740596, 2.301046901963561, 2.301048368142177,
2.3010485807856296, 2.3010470230657134]}

Train Prediction: 0.11202083333333333

Validation Prediction: 0.11375

Test Prediction: 0.1135

Time: 833.9831893444061 sec

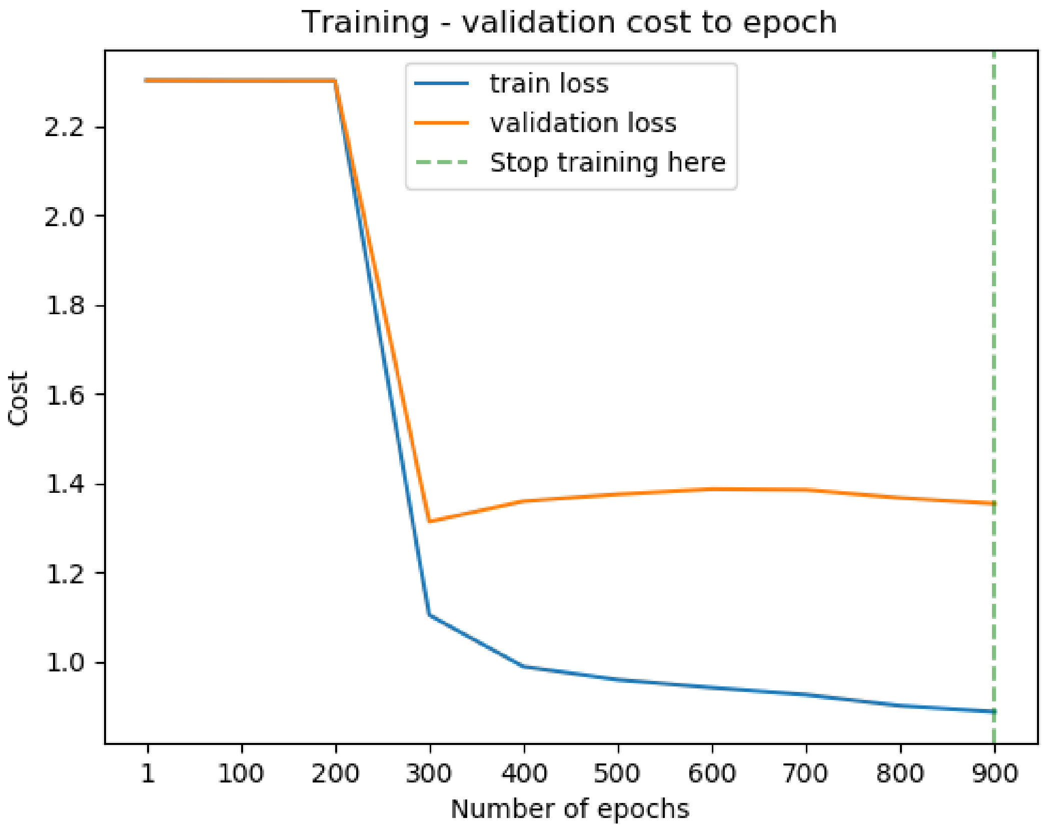


BATCH_SIZE=100

Costs:

{'train': [2.3016620301800597, 2.3012299208806115, 2.3011695543592507,
1.1038003736890125, 0.9880758315925232, 0.9586719137459219, 0.9408716222742965,
0.9251112425535332, 0.9004677692678243, 0.8870940936283636], 'validation':
[2.302113829103343, 2.301207385373336, 2.301148580286567, 1.312807898076706,

1.3588068527142878, 1.3742838045577588, 1.3860483060019326, 1.384343911030215, 1.366177629369207, 1.3534539683542381]}
Train Prediction: 0.65775
Validation Prediction: 0.652
Test Prediction: 0.6467
Time: 614.525110244751 sec



BATCH_SIZE=500
epochs: 200. no improvement - breaking
acc_val_cost > last_cost : 0.9199253142681199>0.8585539239476091
Costs:
{'train': [2.2507323315691776, 0.655430460643252, 0.6341755172124173], 'validation': [2.3262475260675246, 0.8585539239476091, 0.9199253142681199]}
Train Prediction: 0.7933958333333333

Validation Prediction: 0.7840833333333334

Test Prediction: 0.7788

Time: 144.7028157711029 sec