ECE521: Assignment 2

Due on Wednesday, February 17, 2016

Davi Frossard

February 17, 2016

Task 1

The training procedure goes up to a maximum of 500 epochs with a learning rate of 0.01 and momentum of 0.05. The validation cost is checked every 10 epochs and compared to the last value checked, if it is more than 1% bigger, an early stop is triggered. As can be see in Figure 1, an early stop is not triggered and there are no strong signs of over-fitting. From the model we obtain the results shown in Table 1, as evaluated on the test set.

Log-Likelihood Cost	**Accuracy (%)**
0.4408	88.7868

Table 1: Evaluation of the model on test set.

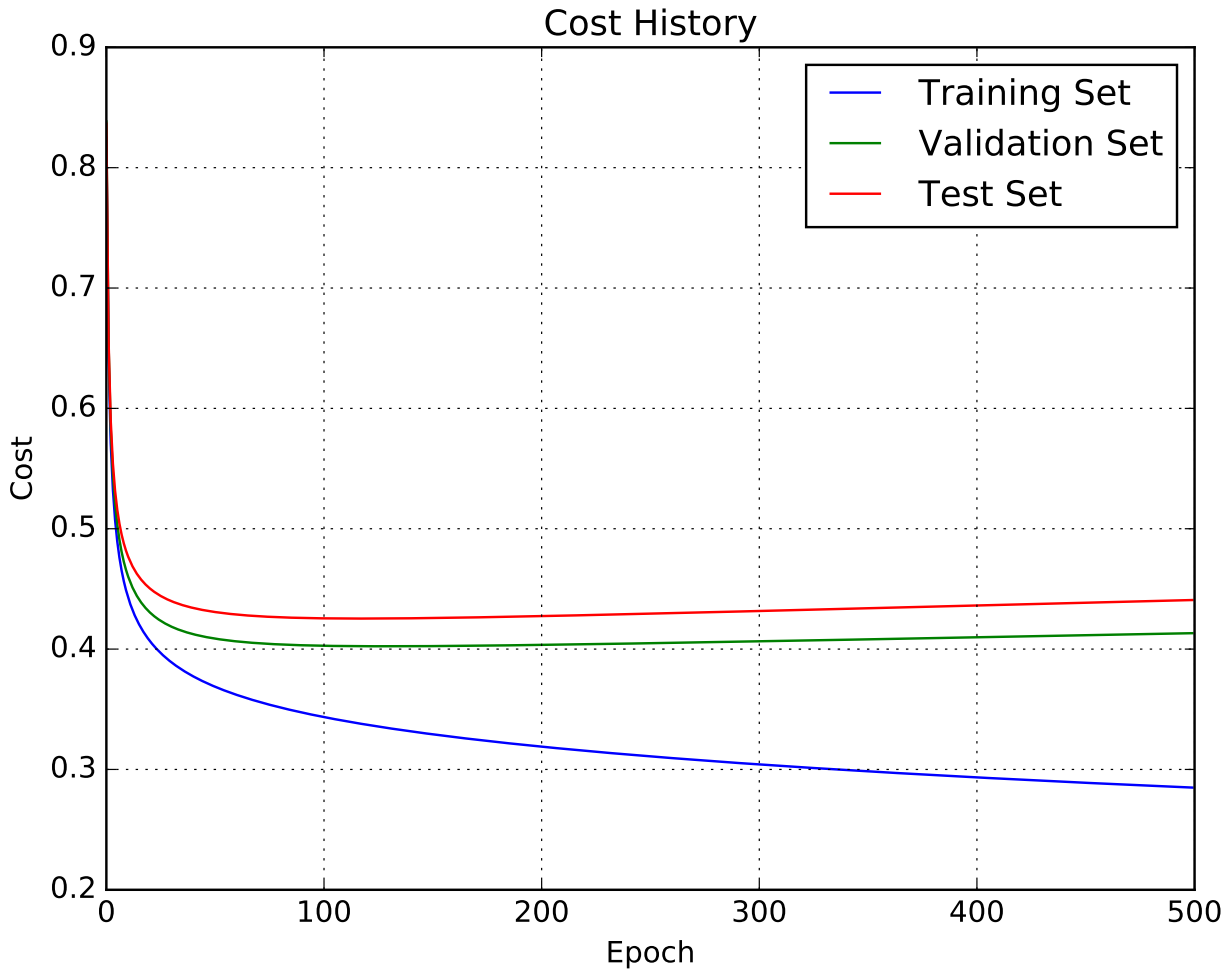

Figure 1: Log likelihood cost for each epoch.

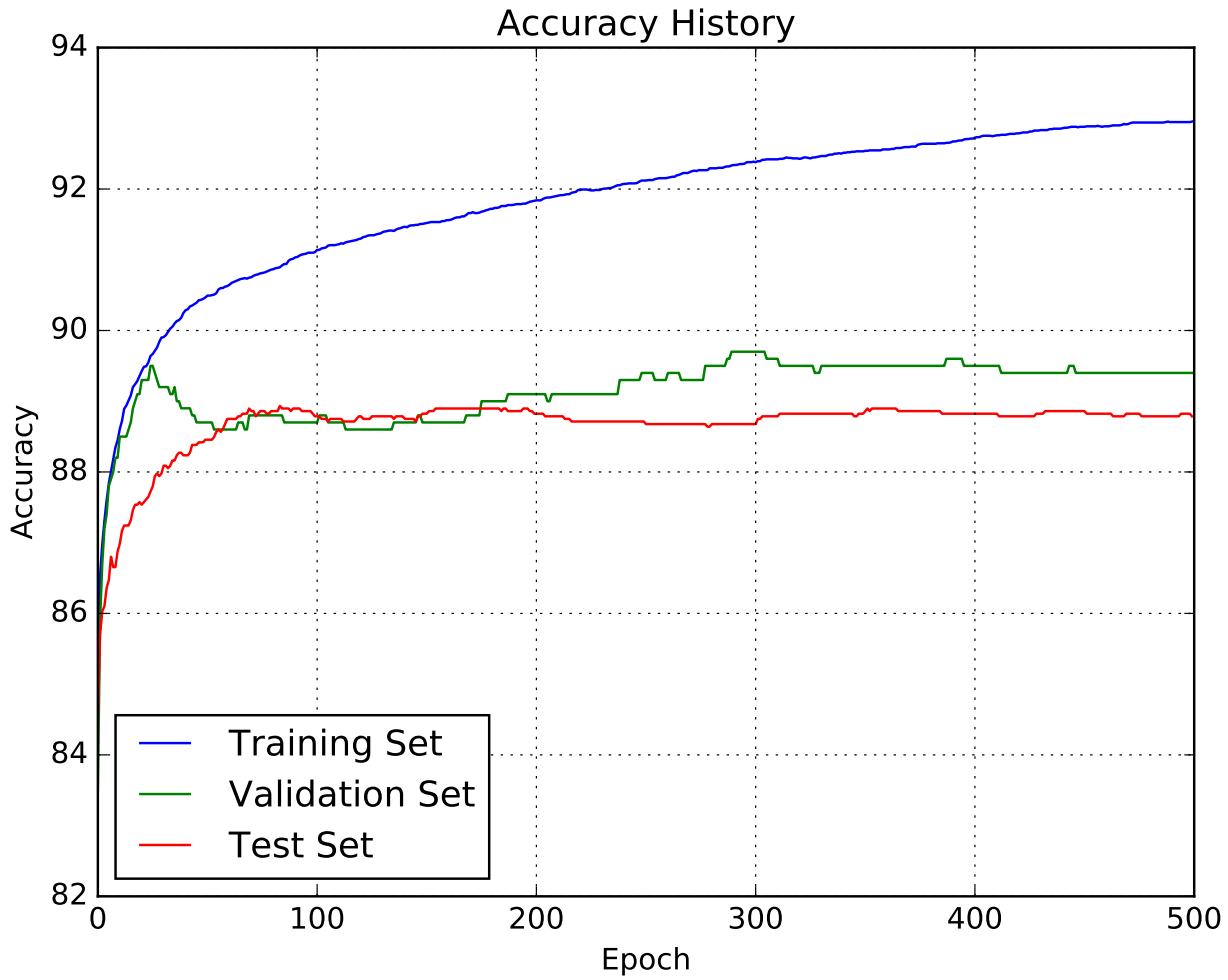

Figure 2: Accuracy of the model for each epoch

Task 2

In order to determine reasonable values for the hyperparameters we randomize a set of 5 rates from 0.001 to 0.09 and fix the momentum at 5 times the learning rate, which proved to be a reasonable value both in Part 1 and throughout this experiment. From the swipe we obtain the values shown in Table 2. We then select the best model according to the validation log-likelihood cost, from which we obtain Table 3. Plotting the training curves for said model we obtain Figure 3 and Figure 4.
Using the same criteria for early-stopping as in Part 1, we notice that bigger learning rates trigger early stops more often than smaller ones and this method proved reasonably effective in returning a good model, although its trigger criteria requires some tuning.

Hyperparameters		**Cost**			**Accuracy (%)**		
Learning	**Momentum**	**Training**	**Validation**	**Test**	**Training**	**Validation**	**Test**
0.0020	0.0100	0.2377	0.3334	0.3470	93.6733	90.6000	89.8897
0.0900	0.4500	0.0435	0.3172	0.3308	99.2333	91.9000	91.2132
0.0050	0.0250	0.1032	0.3004	0.3207	97.8733	91.6000	90.8456
0.0100	0.0500	0.0270	0.3248	0.3662	99.8400	91.6000	90.9559
0.0400	0.2000	0.0503	0.3041	0.3313	99.2733	91.7000	91.0294

Table 2: Hyperparameter search.

Learning	**Momentum**	**Cost**	**Accuracy (%)**
0.0050	0.0250	0.3207	90.8456

Table 3: Evaluation of best model from swipe on test set.

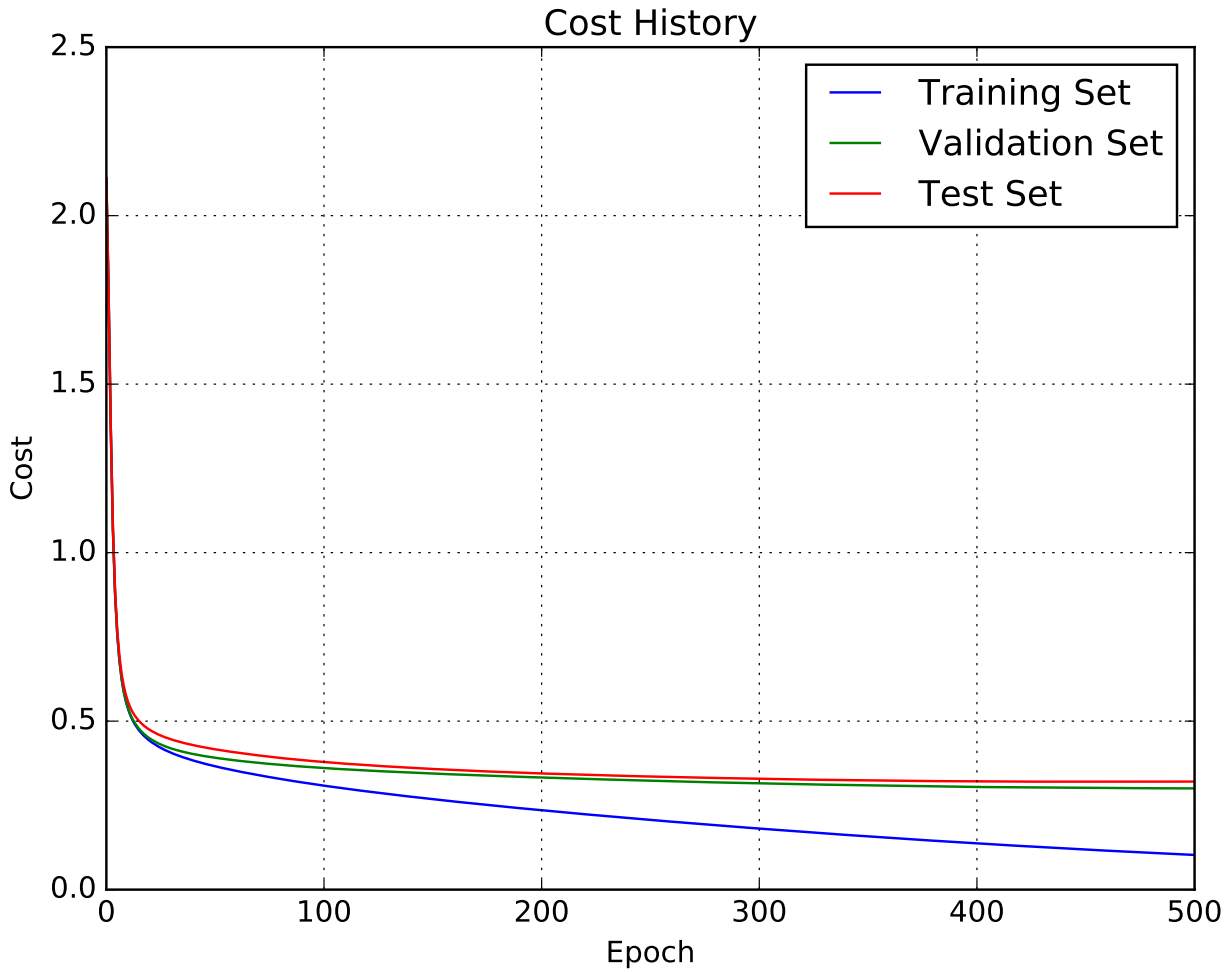

Figure 3: Log likelihood cost for each epoch in the training of the best model.

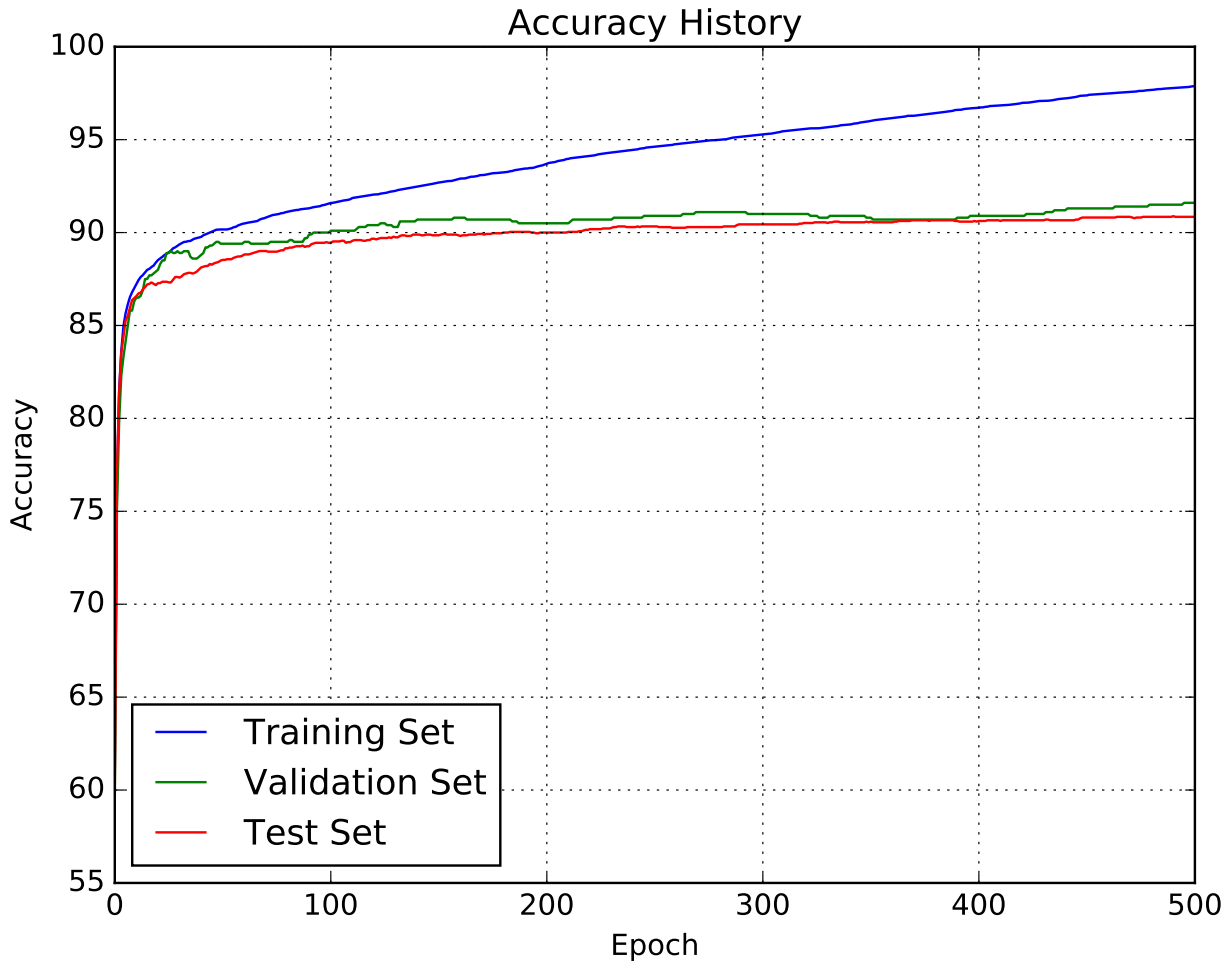

Figure 4: Accuracy of the model for each epoch in the training of the best model

Task 3

In order to determine the best topology from the ones suggested, we use the optimum hyperparameters found in Part 2 and train three neural networks, each with one of the topologies, obtaining Table 4. We then select the one with the minimum log-likelihood cost for the validation set, which gives us Table 5. Plotting the training curves we get Figure 5 and Figure 6.
Using the validation cost as the comparison reference, as we have been doing so far, we notice no major impact caused by the number of hidden layers in this range, they all produced reasonable results. By our metrics the best model is the one with 500 hidden units.

Topology	**Cost**			**Accuracy (%)**		
	Training	**Validation**	**Test**	**Training**	**Validation**	**Test**
100.0	0.1309	0.3021	0.3436	96.7200	91.5000	90.4044
500.0	0.1107	0.3008	0.3305	97.5733	91.6000	90.6250
1000.0	0.1026	0.3027	0.3215	97.8600	91.5000	90.6985

Table 4: Topology Swipe.

Topology	**Cost**	**Accuracy (%)**
500	0.3305	90.6250

Table 5: Evaluation of best model from swipe on test set.

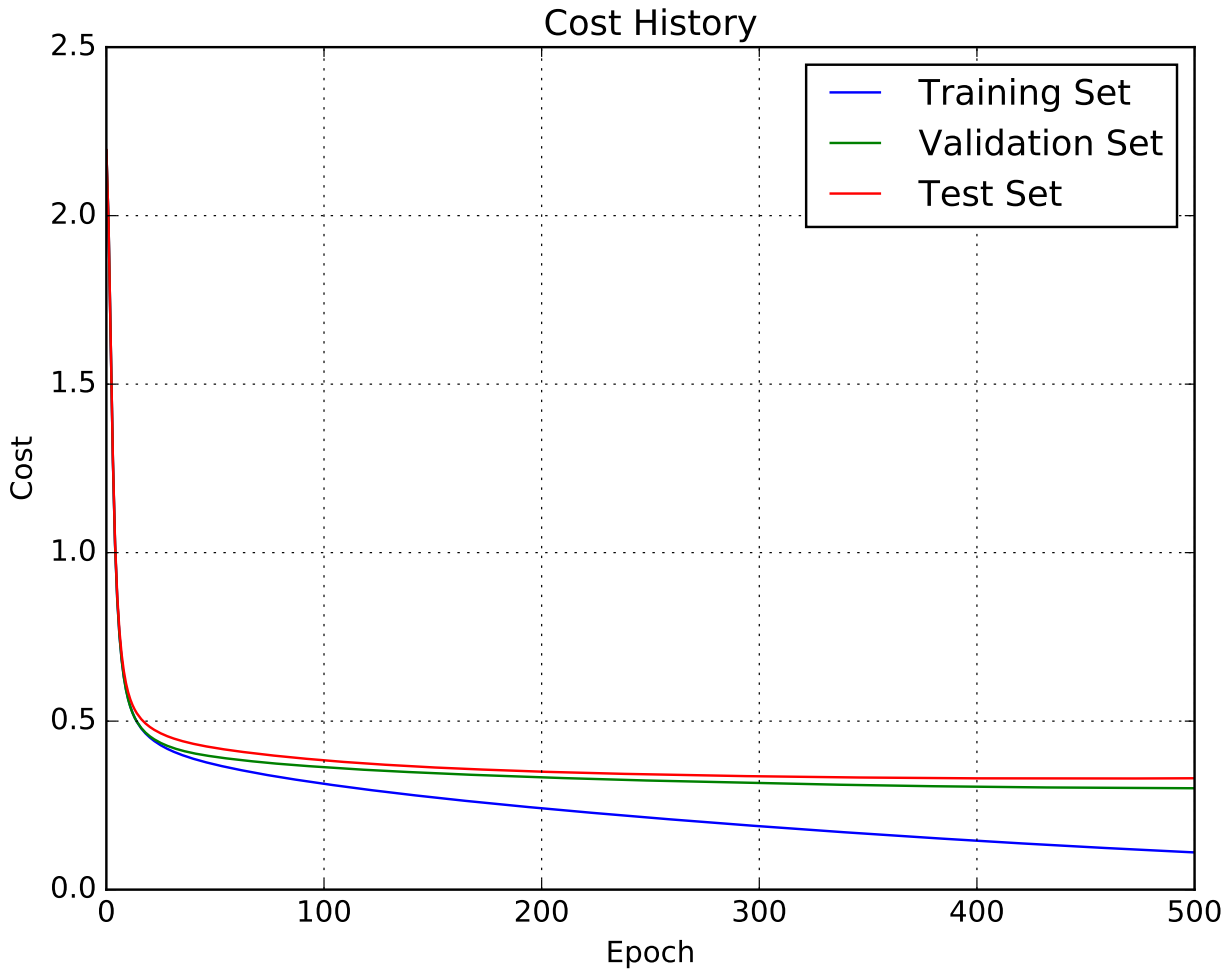

Figure 5: Log likelihood cost for each epoch in the training of the best model.

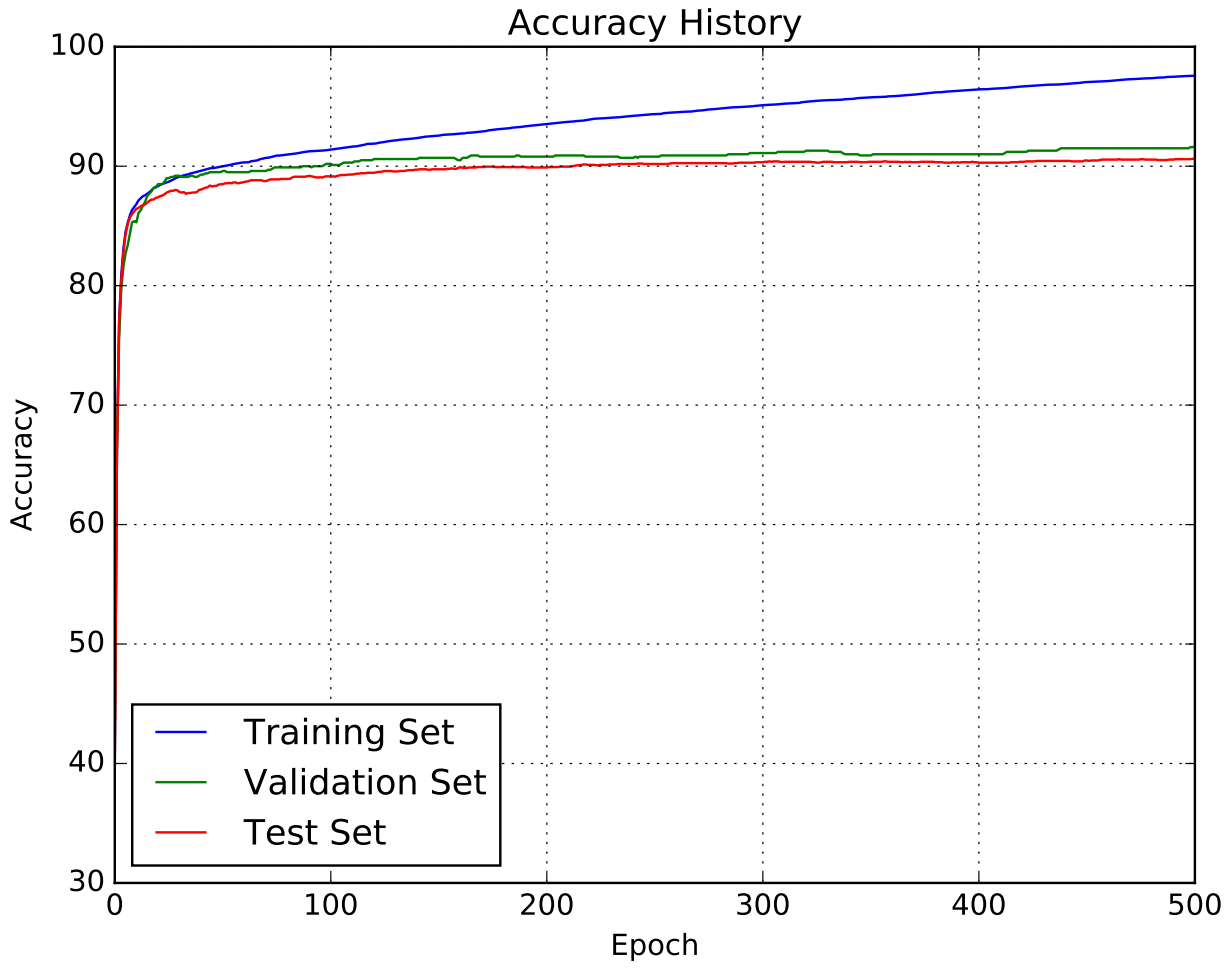

Figure 6: Accuracy of the model for each epoch in the training of the best model

Task 4

Instead of using a single layer and varying the number of hidden units, we now train a neural network with two layers with 500 units in each one. We obtain the results shown in Table 6. With the training curves shown in Figure 7 and Figure 8.
From the experiment we conclude that a two layer topology provides results in the same range as those obtained with a single hidden layer. During the parameter tuning phase we also noticed that this topology is more prone to overfitting.

Log-Likelihood Cost	**Accuracy (%)**
0.3825	90.9191

Table 6: Evaluation of the model on test set.

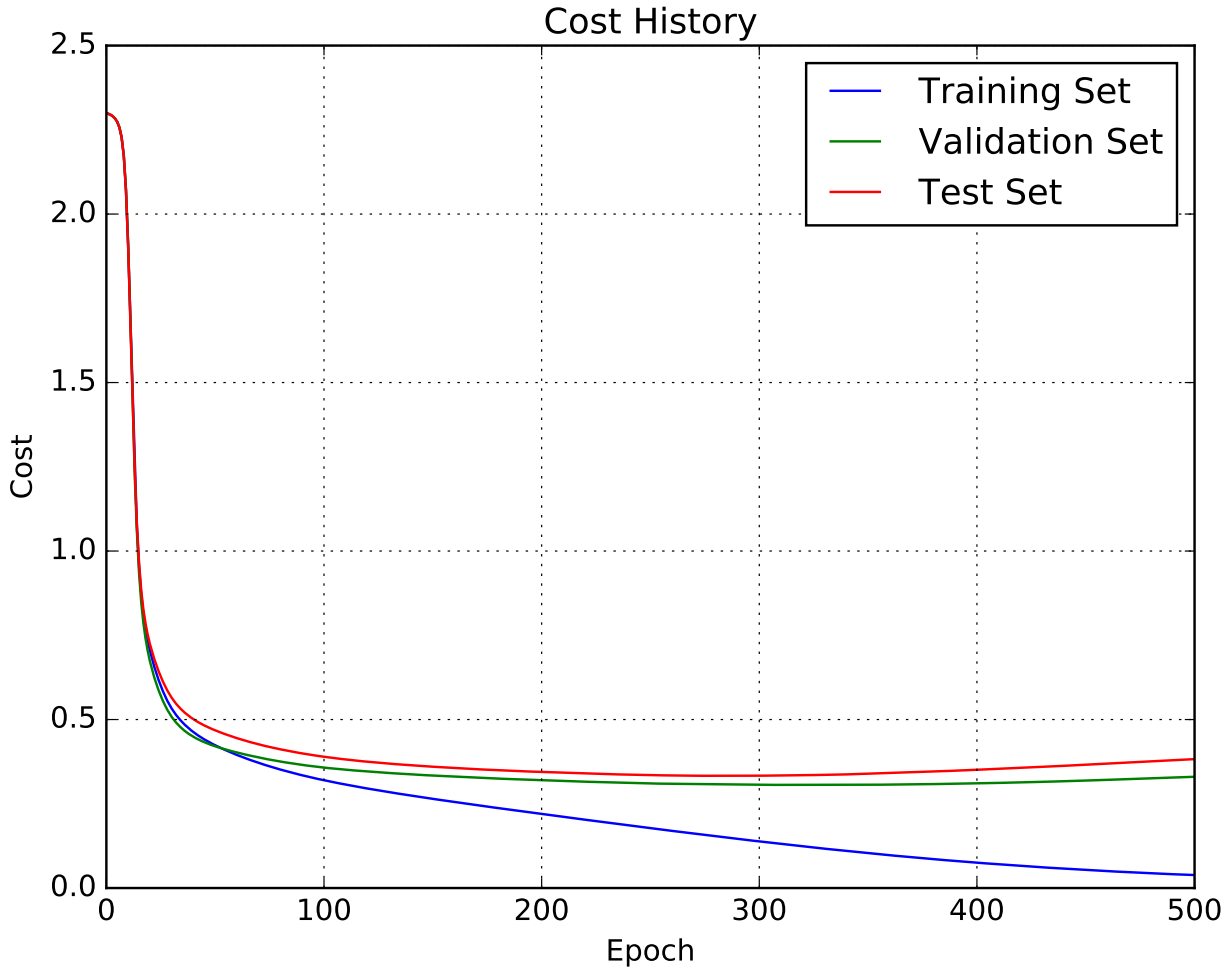

Figure 7: Log likelihood cost for each epoch in the training of the best model.

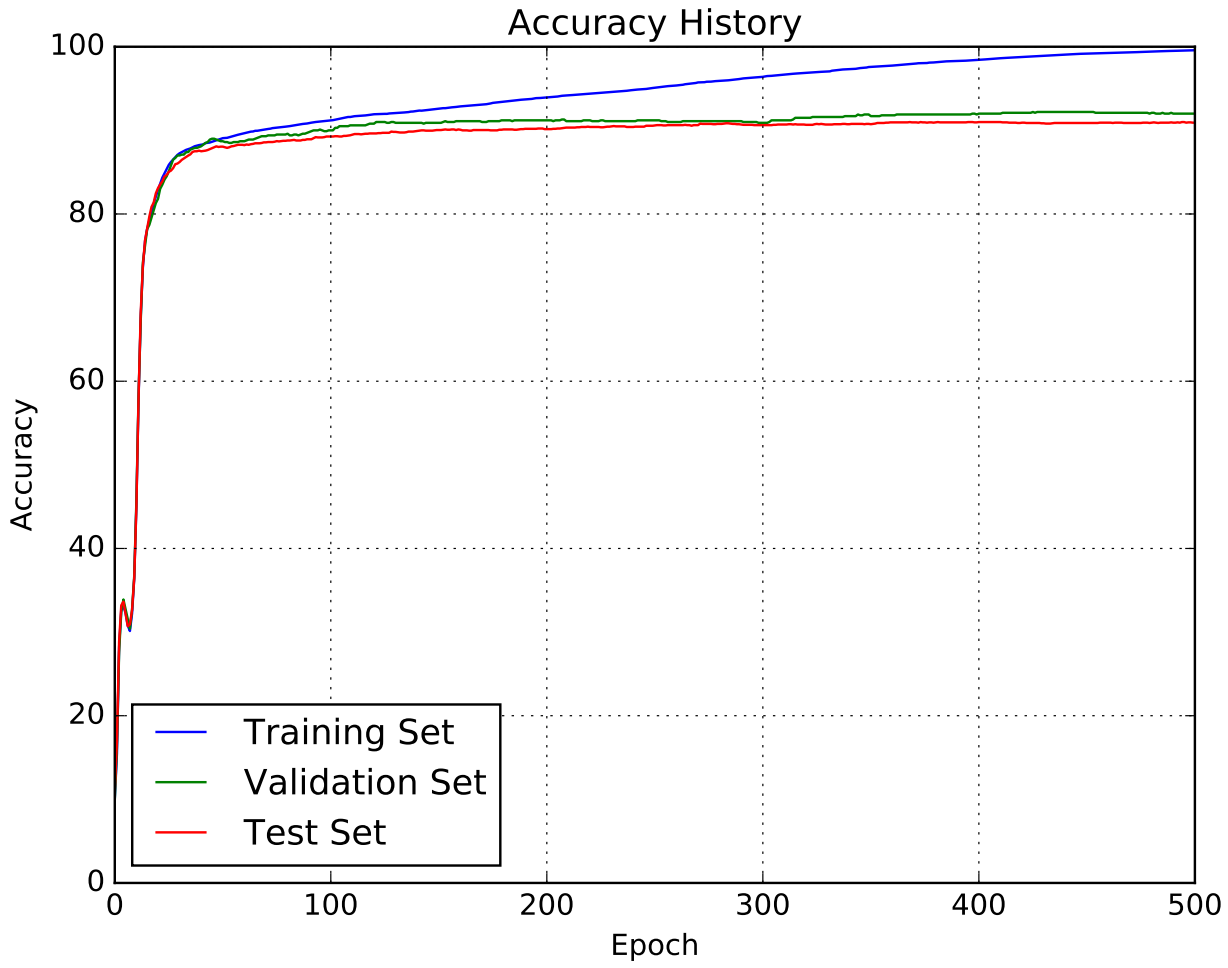

Figure 8: Accuracy of the model for each epoch in the training of the best model

Task 5

To evaluate dropout as a method to avoid overfitting we disable early stopping and use a 50% chance of keeping the weights. With this model we obtain the results shown in Table 7, with training curves shown in Figure 9 and Figure 10.
Comparing to the model obtained in Part 2 with no dropout, we notice that dropout is effective in avoiding overfitting without the need of subtle analysis of how the validation cost is behaving as in early stopping.

Log-Likelihood Cost	**Accuracy (%)**
0.2982	91.2868

Table 7: Evaluation of the model on test set.

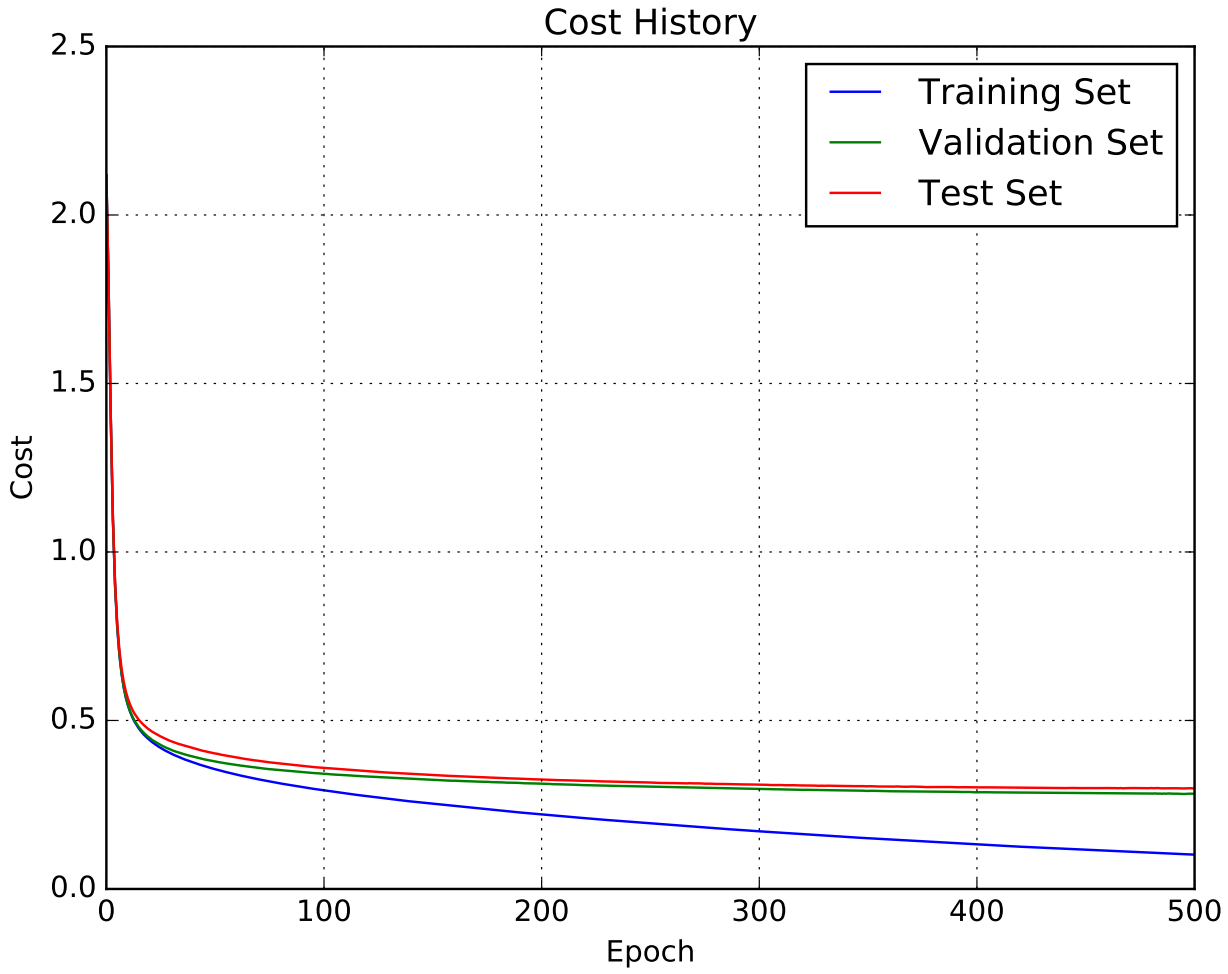

Figure 9: Log likelihood cost for each epoch in the training of the best model.

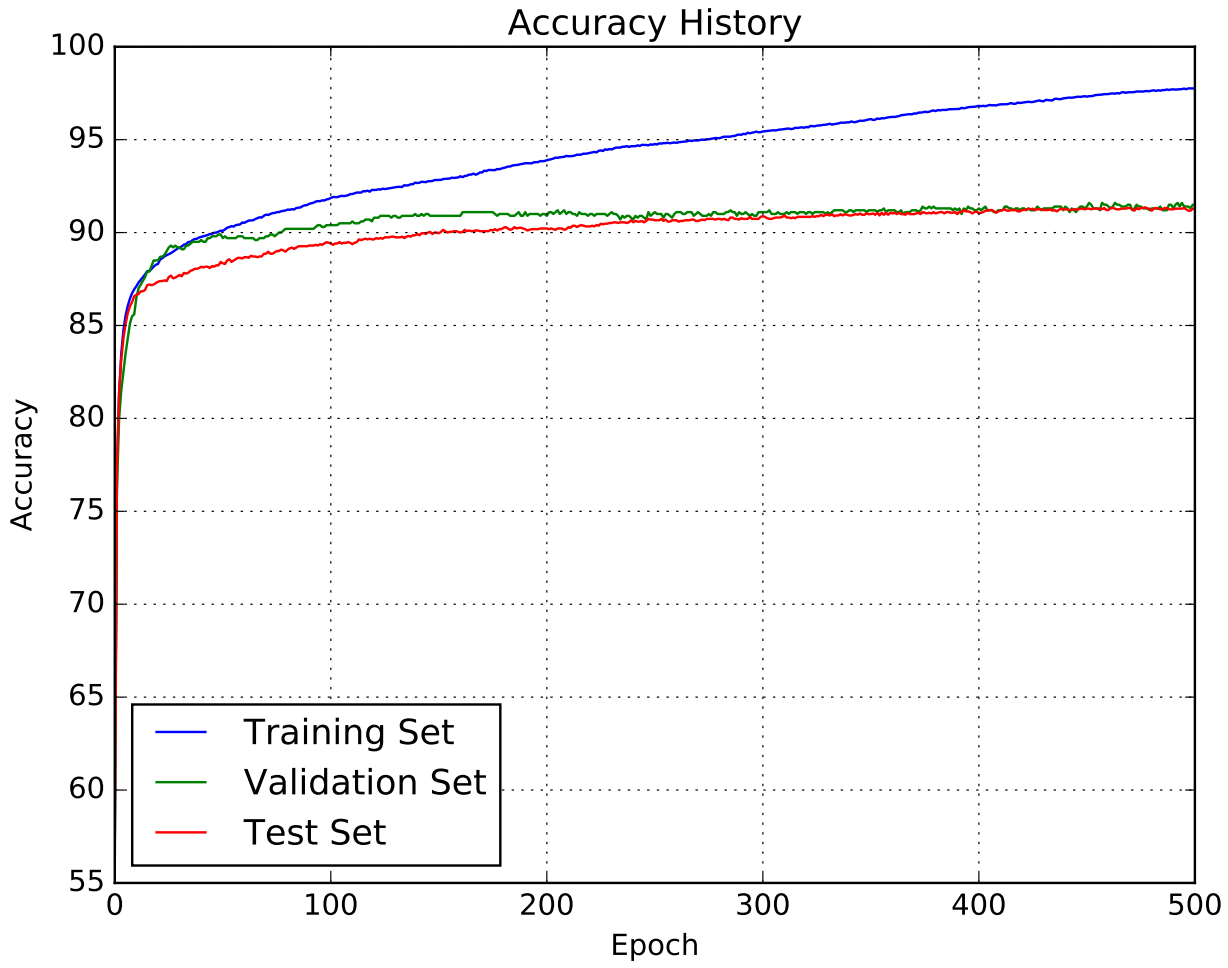

Figure 10: Accuracy of the model for each epoch in the training of the best model

Task 6

For this experiment we train 40 different neural networks. Each time a learning rate from 0.09 to 0.0001 is used, momentum is always double the learning rate (we are more conservative since higher multipliers were leading to divergence in some cases), the number of layers varies from 1 to 3 and each layer might have from 100 to 500 units. For each model we train it with and without dropout, early stopping is never used. With these parameters we obtain the results shown in Table 8, sorted by the validation cost.

Hyperparameters				**Validation**		**Test**	
Learning	**Topology**	**Dropout**	**Momentum**	**Cost**	**Accuracy**	**Cost**	**Accuracy**
0.0040	[120 309]	0.5	0.0080	0.2883	91.3000	0.3228	90.7353
0.0080	[243]	0.5	0.0160	0.2919	92.1000	0.3058	91.2132
0.0030	[424 396 339]	0.5	0.0060	0.2959	91.1000	0.3565	90.5515
0.0020	[204 484]	0.5	0.0040	0.2975	91.2000	0.3311	90.0735
0.0030	[280]	0.5	0.0060	0.3017	91.2000	0.3191	90.5515
0.0040	[120 309]	1	0.0080	0.3129	91.8000	0.3520	90.2574
0.0080	[243]	1	0.0160	0.3130	91.6000	0.3553	90.8824
0.0020	[472]	0.5	0.0040	0.3159	91.1000	0.3288	89.8529
0.0020	[204 484]	1	0.0040	0.3182	91.0000	0.3489	89.8897
0.0030	[280]	1	0.0060	0.3211	90.8000	0.3399	90.0368
0.0200	[283]	0.5	0.0400	0.3230	92.6000	0.3602	91.3235
0.0020	[472]	1	0.0040	0.3343	90.6000	0.3502	89.8529
0.0010	[175]	0.5	0.0020	0.3423	89.8000	0.3657	89.1176
0.0300	[165]	0.5	0.0600	0.3453	92.8000	0.4095	91.3235
0.0030	[424 396 339]	1	0.0060	0.3480	91.4000	0.4196	90.4044
0.0010	[175]	1	0.0020	0.3708	89.9000	0.3904	89.0074
0.0006	[277]	0.5	0.0012	0.3709	89.5000	0.3942	88.5294
0.0070	[485 430]	1	0.0140	0.3739	92.7000	0.4255	90.8824
0.0006	[277]	1	0.0012	0.3882	89.2000	0.4153	88.6029
0.0090	[127 401 375]	0.5	0.0180	0.3931	91.7000	0.5194	90.6985
0.0500	[219]	0.5	0.1000	0.4155	93.1000	0.4721	91.4338
0.0070	[260 398 495]	0.5	0.0140	0.4190	91.8000	0.4751	91.2500
0.0200	[283]	1	0.0400	0.4198	91.7000	0.4567	90.6985
0.0500	[219]	0.5	0.1000	0.4249	92.5000	0.4739	91.5441
0.0500	[393 207]	0.5	0.1000	0.4538	93.4000	0.5391	92.2794
0.0400	[356 347]	0.5	0.0800	0.4578	93.0000	0.5303	92.2426
0.0400	[387 319]	0.5	0.0800	0.4587	93.1000	0.5252	91.9485
0.0300	[165]	1	0.0600	0.4618	92.0000	0.5134	91.0662
0.0500	[219]	1	0.1000	0.5122	92.3000	0.5501	90.6985
0.0500	[219]	1	0.1000	0.5132	92.3000	0.5488	90.9926
0.0500	[393 207]	1	0.1000	0.5176	93.1000	0.6288	91.3603
0.0400	[387 319]	1	0.0800	0.5419	92.4000	0.6258	91.2868
0.0400	[356 347]	1	0.0800	0.5450	92.2000	0.6250	91.2132
0.0070	[260 398 495]	1	0.0140	0.6026	90.8000	0.7344	90.2574
0.0090	[127 401 375]	1	0.0180	0.7482	90.3000	0.8841	89.8162
0.0008	[434 470 303]	1	0.0016	0.7911	78.4000	0.8725	78.9338
0.0008	[434 470 303]	0.5	0.0016	1.0135	72.5000	1.0648	72.0221
0.0006	[380 432 420]	1	0.0012	1.7981	30.6000	1.7817	31.4338
0.0006	[380 432 420]	0.5	0.0012	2.2564	19.3000	2.2543	20.9559

Table 8: Hyperparameter search.