Sai Aditya Varma Vuppalapati

vuppalsa@mail.uc.edu

Abstract

**This report guides through the design choices I have made in building appropriate deep neural net for the data set “Optical Recognition of Handwritten Digits”**

Experimental report

Success Hub interview task

**1. Introduction**

This is a classification task of handwritten digits. most of the postal addresses we write on postcards/couriers have to go through optical recognition in order to identify hand written addresses. there is a need to classify area codes with high precessions. This deep learning system I have build serves a great purpose for this task.

They were many Machine learning models that have been built over this dataset, they primarily used basic ML models such as svm, knn etc. Although they gave pretty good accuracies in detecting numbers with moderate computational resources, but they could only give accuracies in the range 0f 85-92%.

With the availability of economically feasible computational devices such as gpu’s and cloud services which offer this service at affordable prices there is a need in upgrading ML systems to deep learning models to enjoy high accuracies.

In this report I am going to summarize the results of deep learning model and draw a comparison on using Sum-of-squares error vs. cross-entropy error function and a comparison of Logistical sigmoid vs. tanh vs. ReLU activation for hidden units.

**2. Problem Definition and Algorithm**

2.1 Task Definition

This system is built for recognizing handwritten digits with high accuracy. The input to the system will be a preprocessed input matrix of 8x8 bitmaps and the output will be a predicted number from 0 to 9.

2.2 Algorithm Definition

The algorithm I have used is fully connected neural net. The 8\*8 bitmap which we take as input has 64 intensity levels as inputs. this input was given to 64 nodes in the input layer, these are connected to a 4 dense layers with variable node size.

**3. Experimental Evaluation**   
  
3.1 Methodology(3.1 is a Experimental documentation but not a demonstration, for findings please look at 3.2)

This criteria for evaluation of this architecture is based on validation loss, validation accuracies, test accuracies and convergence speed.

This experiment mainly tests the hypothesis that-

1. cross entropy outperforms sum of squared error.

2.Relu out performs Logistical sigmoid and tanh activation functions.

**Systematic experimentation is conducted to prove the hypothesis:**

**1.Preprocessing of data:**

I skipped image data preprocessing because the images were well centered, and the data is not skewed.

**2.Controlling for variance:**

**Variance in the data**

the dataset already came with training/test split, I have used 20%of training data as validation set picked randomly.

**Variance in the model**

Variance in this system is perfectly maintained as I choose to use Keras which initializes random weights in each experiment and also make sure shuffled data is given to input layer in each epoch.

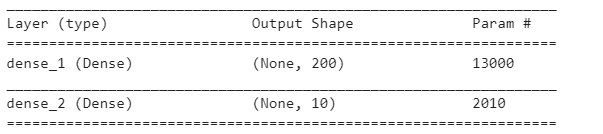
**3.Model and Architecture based Experiments**

**A. Experimenting to discover the best architecture for given dataset.**

I choose to experiment for finding best architecture assuming cross entropy is the best error function and Relu is the best activation function, once I find the best architecture ill cross check weather this architecture gives best results with other loss/activations.

**Trail-1**

Architecture- input layer with 64 nodes ,hidden layer with 200 nodes,o/p layer with 10 nodes.



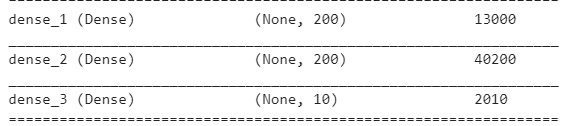


Val\_accu=98.82 val\_loss=0.05041,

This architecture can be verified by loading **archtrail1.hdf5** file

**Trail-2**

Architecture- input layer with 64 nodes ,2 hidden layer with 200 nodes , o/p layer with 10 nodes.





Val\_loss= 0.04426 val\_accuracy=98.69

This architecture can be verified by loading **archtrail2.hdf5** file

**Trail-3**

Architecture- input layer with 64 nodes ,3 hidden layer with 200 nodes , o/p layer with 10 nodes.

the test accuracy is 95.93767390094602

This architecture can be verified by loading **archtrail3.hdf5** file

**Trail-4**

Architecture- input layer with 64 nodes,2 hidden layer with 200 nodes,1 hidden layer with 100 nodes , o/p layer with 10 nodes.

the test accuracy is 96.88369504730106

This architecture can be verified by loading **archtrail4.hdf5** file

**Trail-5**

Architecture- input layer with 64 nodes

1 hidden layer with 200 nodes ,

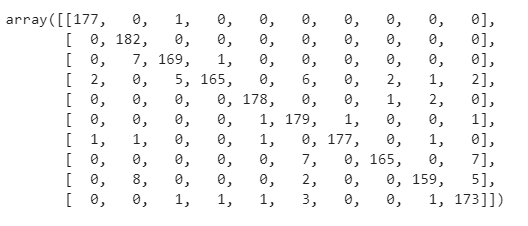
1 hidden layer with 150 nodes ,

o/p layer with 10 nodes.

the test accuracy is 97.60712298274903

val\_loss=0.04267 val\_accuracy=98.82

This architecture can be verified by loading **archtrail5.hdf5** file(1 relu + 1 sigmoid)



**Trail-6**

Architecture- input layer with 64 nodes

1 hidden layer with 200 nodes ,

1 hidden layer with 100 nodes ,

o/p layer with 10 nodes.

the test accuracy is 97.32888146911519

This architecture can be verified by loading **archtrail6.hdf5** file

The best architecture so far was trail-5. proceeding with trail 5 architecture for further experimenting.

**B.Tune-Model Experiments(Task 1 and Task 2)**

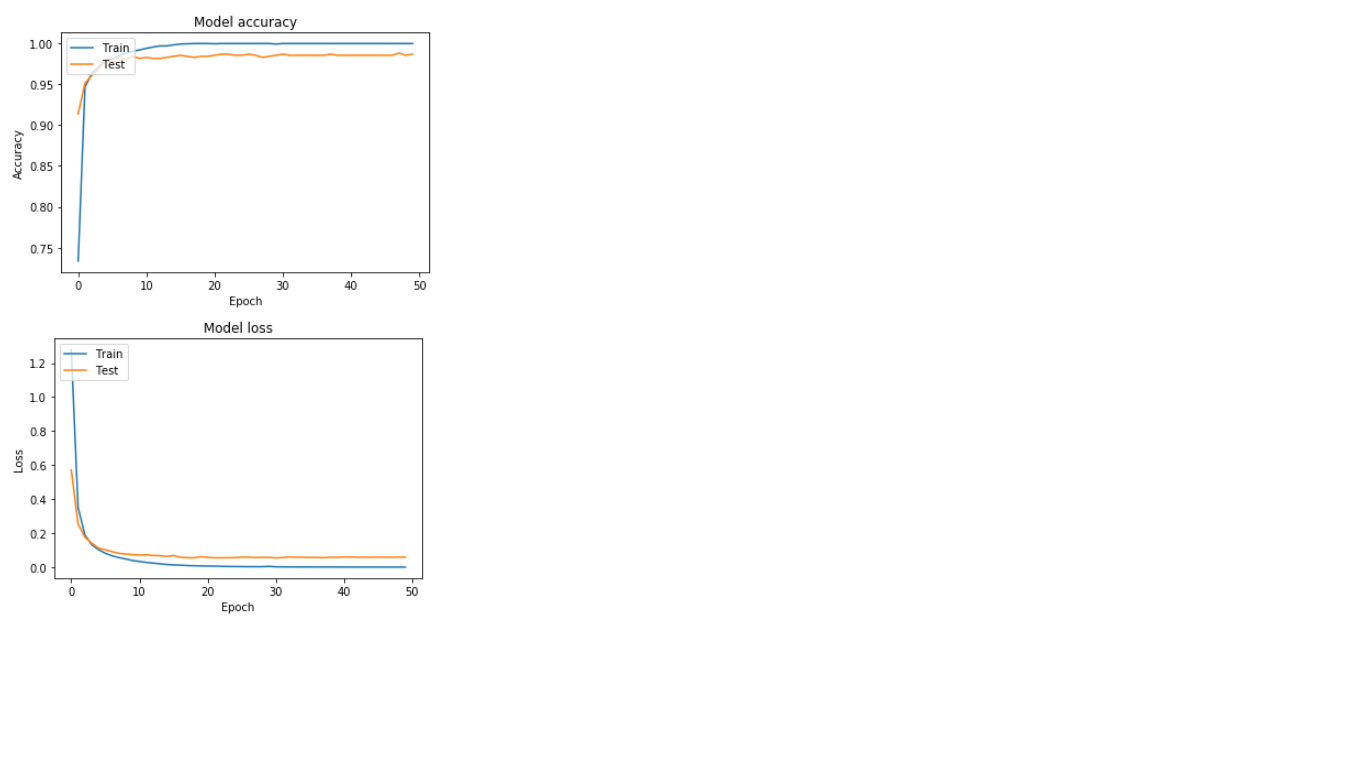
**Experimenting with independent variables (hyper parameters)**

**Sum-of-squares error vs. cross-entropy error function**

**Trail-1a**

**Using trail 5 arch (cross\_entropy) with sigmoid activation functions.**

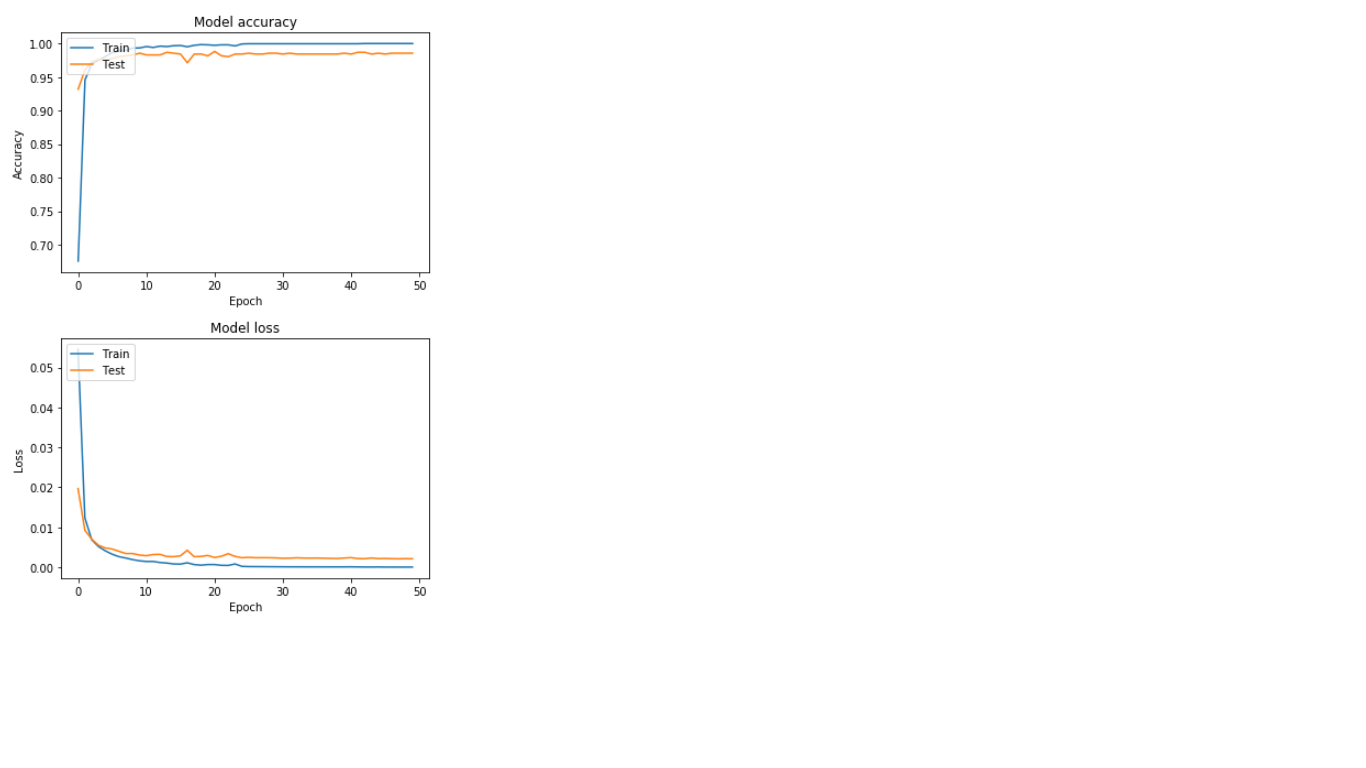
the test accuracy is 96.43850862548692

**tm\_exp1a.hdf5**

**Trail-2a**

**Using trail 5 arch (cross\_entropy) with sigmoid activation functions.**

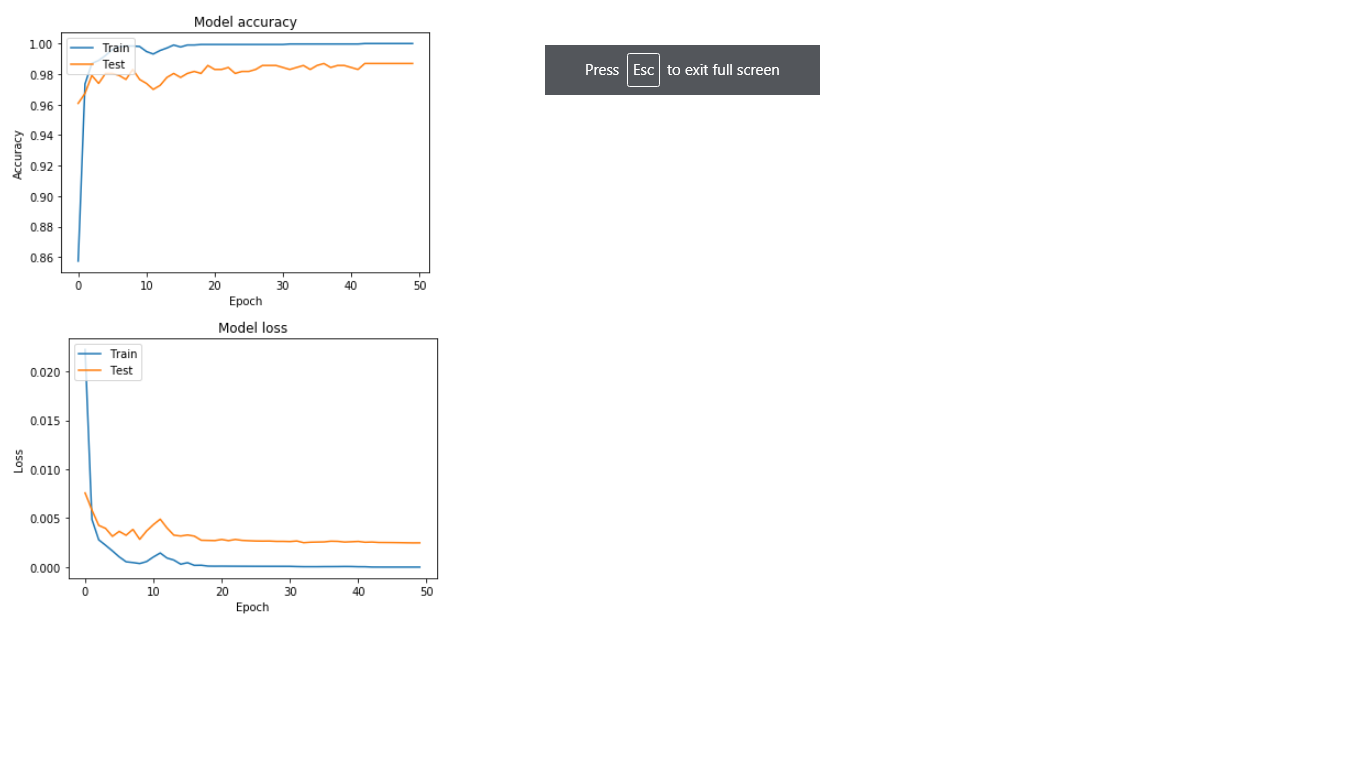
the test accuracy is 96.60545353366722

 **tm\_exp2a.hdf5**

**Trail\_3a:**

tanh + cross entropy

the test accuracy is 96.4941569282137



3.2 Results

**Best architecture**

|  |  |  |
| --- | --- | --- |
| **architecture** | **Val-loss** | **Test accuracy** |
| **Test1** | **0.05041** | **96.54** |
| **Test2** | **0.044** | **97.16** |
| **Test3** |  | **95.93** |
| **Test4** |  | **96.88** |
| **Test5** | **0.04267** | **97.60** |
| **Test6** |  | **97.32** |

**Refer 3.1 for detailed test architecture.**

The best result is obtained at test5.

**Brief experimenting summary:(activation=relu,loss=cross entropy)**

1.started with 1 hidden layer with 200 nodes-got test accuracy of 96.54

2.tried increasing depth of neural net. (2 hidden layers with 200 nodes each), it performs well

And got a accuracy of 97.16.

3.added 1 more hidden layer(3 hidden layers with 200 nodes each),model started over fitting.

4.added regularization(dropout) to test 3 model, that did not help,

Till this point test 2 was the best result.

5.i tried checking weather model 2 overfits or not by decreasing the number of nodes in last hidden layer and it turned out that’s true.

Got the best accuracy till now.

6.checked even if model 5 overfits by decreasing nodes, it did not work.

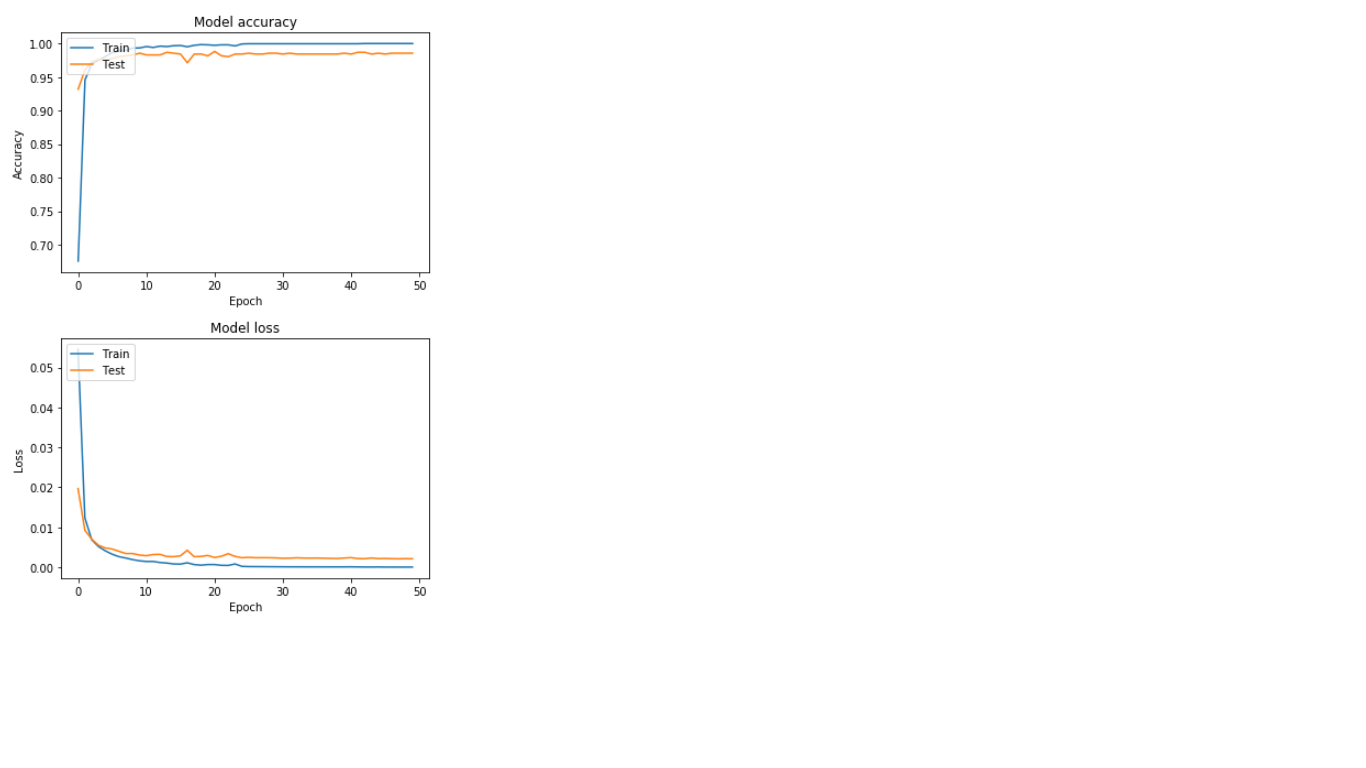
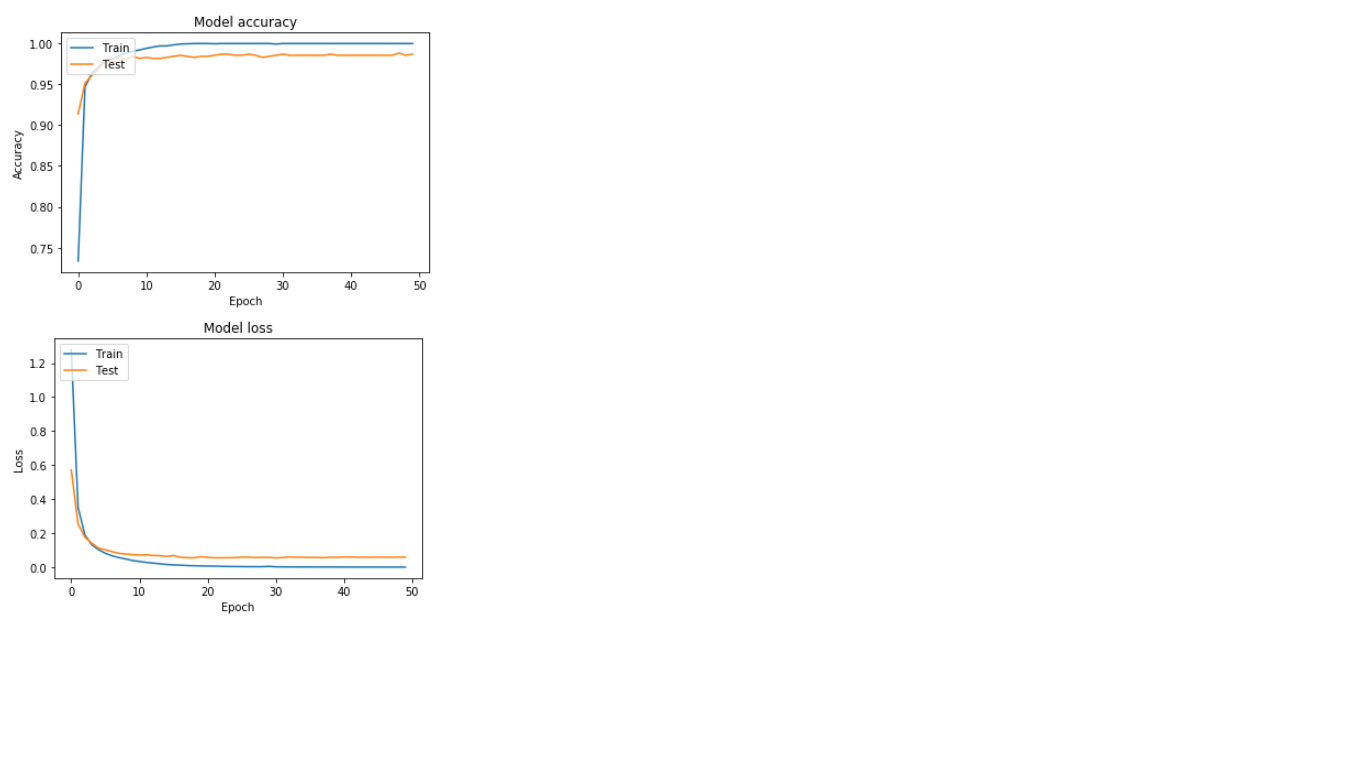
So, concluded that test5 architecture works best for this dataset.

Task-1:

using the above architecture with sigmoid as activation

|  |  |
| --- | --- |
| Error function | Test accuracy |
| Cross-entropy | 96.43 |
| Sum of squared error | 96.60 |

Both the models gave similar results, but sse outperformed.



Both the models converged before 20 th epoch and continued to descent until 30 th epoch and reached their global minima.

Tried different learning rates and the best result was given by 0.01, as we increase the lr there may be chances of missing minima.

Coming to momentum Adam comes with nesterov momentum so no need to tune it.

Task-2:

|  |  |
| --- | --- |
| activation | Test accuracy |
| sigmoid | 96.43850862548692 |
| tanh | 96.4941569282137 |
| relu | 97.60(test5) |

On tuning parameters relu was found to be giving best results.