Protein-Ligand Scoring with Convolutional Neural Networks

Surabhi S Nath

2016271

1 Abstract

This paper describes the use of protein-ligand binding data for deep learning to predict binding affinities and poses. The scoring function which ranks and scores protein-ligand structures is developed with the help of Deep Learning. The authors have developed a CNN which can enable automatic learning of key features of protein-ligand interactions and how they correlate to binding. They have trained the network on 3D grid representations of protein-ligand structures generated through docking. They have proved that their CNN based scoring function outperforms the scoring function of AutoDock Vina.
Three tasks are performed -

1. Pose Prediction - Selection of correct binding mode
2. Virtual Screening - Distinguishing between binders and non-binders
3. Affinity Prediction - Predict binding affinities

2 Datasets

For the Pose Prediction task, they use the CSAR dataset. The following steps are performed to make the input from this dataset:

- Filter ligands based on binding affinity
- Docked the filtered ligands against the reference receptor to generate upto 20 poses
- Convert to positive and negative binding labels based on RMSD value
- RMSD < 2Å is considered POSITIVE (correct pose)
- RMSD > 4Å is considered NEGATIVE (incorrect pose)
- This resulted in 745 POSITIVE and 3251 NEGATIVE cases

For the Virtual Screening task, they use the DUDE dataset. The following steps are performed to make the input from this dataset:

- Dock ligands against reference receptor using SMINA
- Use the top ranked poses
- This resulted in 22645 POSITIVE and 1407145 NEGATIVE cases

However, this dataset was not considered appropriate for deep learning tasks due to the following reasons:

- Hidden Bias leading to misleading performance (ref)
- Highly imbalanced dataset due to cross-docking

Following this, the data is converted into a grid structure to mimic the 3D image RGB structure. They also used 2 independent test datasets for testing.

3 Training

The training procedure was as follows:

- Caffe framework was used
- Multinomial logistic loss function was applied
- Stochastic Gradient Descent was employed
- Cross-validation was performed
- AUC ROC was used for model evaluation

4 Reported Results

4.1 Training Dataset Results

Dataset/AUC for Model	Author's model	Vina
CSAR	0.815	0.645
DUDE	0.864	0.683

Dataset/AUC for Task	Pose Prediction	Virtual Screening
CSAR	0.815	0.66
DUDE	0.56	0.864
2:1 DUDE:CSAR	0.79	0.83

4.2 Testing Dataset Results

Dataset/AUC	CSAR Model	DUDE Model	2:1 DUDE:CSAR model	Vina
PDBbind	0.79	0.45	0.77	0.68
ChEMBL		0.78	0.64	0.67
MUV		0.52	0.50	0.55

5 My Experiments

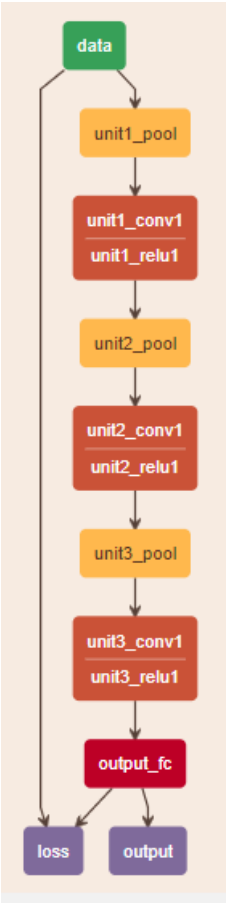
After setting up Caffe and building the GNINA resource, I trained and tested the network on multiple datasets for pose prediction, virtual screening and affinity prediction tasks.
The following experiments were carried out:

1. CSAR dataset training and testing on –
   (a) CSAR All dataset
   (b) CSAR Balanced dataset
2. General dataset training and testing
3. Affinity model testing on General dataset
4. 2:1 CSAR:DUDE testing on –
   (a) General dataset
   (b) CSAR All dataset
   (c) CSAR Balanced dataset

6 My Results

6.1 CSAR Training

The CSAR dataset docked poses were provided by the authors in 2 forms - All dataset and a Balanced dataset. They have also provided a model containing the following structure:



I trained this model on the provided 2 forms of data for 10,000 iterations using 3 fold cross validation on All dataset and held-out set in Balanced dataset. I plotted the following graphs -

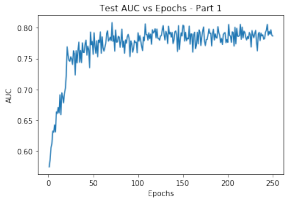
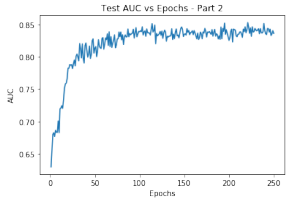
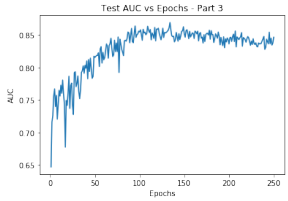
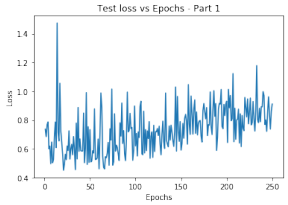
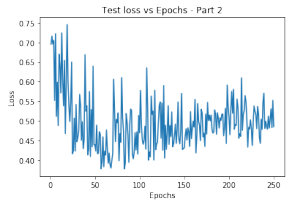
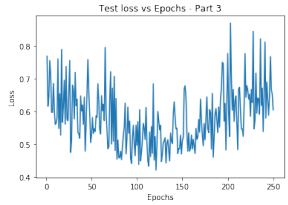
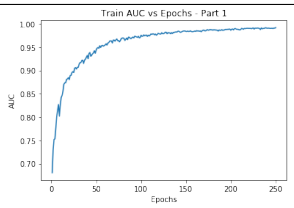
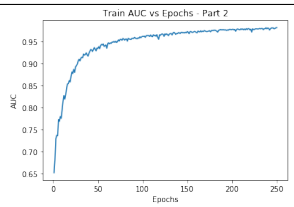
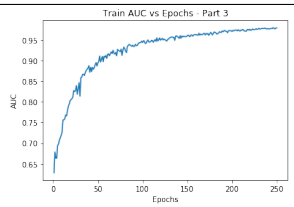
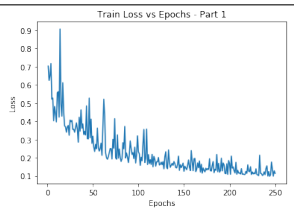
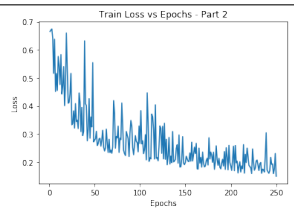
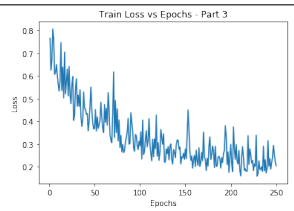
- Test AUC
- Test Loss
- Train AUC
- Train Loss

For each Receptor - Ligand pose pair, a prediction probability (Column 1) between 0 and 1 was obtained indicating if the pair would bind or not. This probability was converted to a binary label 0/1 for binding/not binding using the following rule -

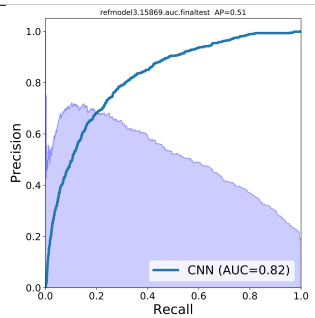
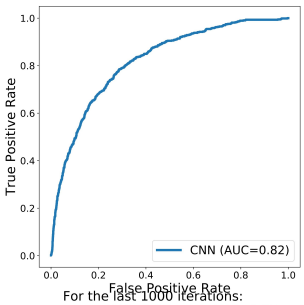
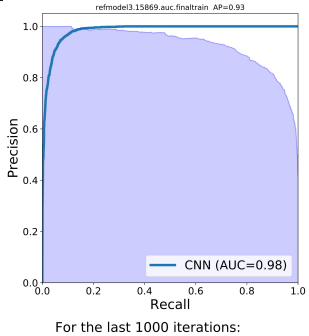
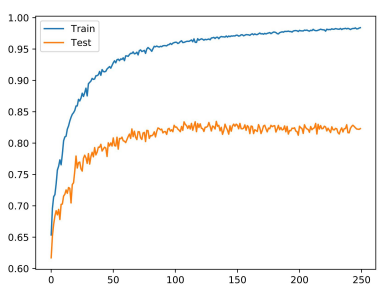
- If probability ≥ 0.5, then 1
- Else 0

6.1.1 CSAR All dataset

Curves for each Fold

Metric/Fold	Fold 1	Fold 2	Fold 3
Test AUC			
Test Loss			
Train AUC			
Train Loss			

Test and Train Curves

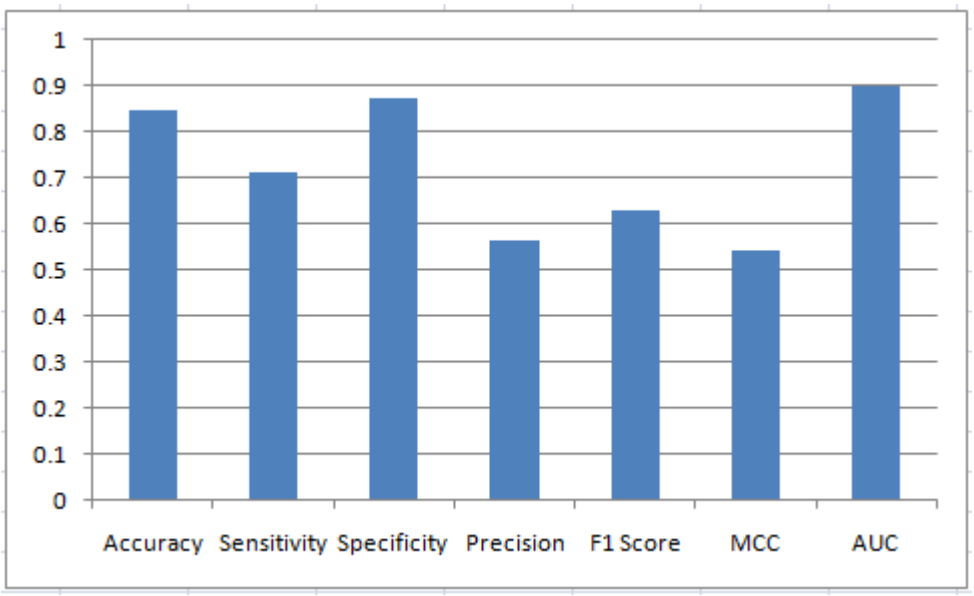
Test Curves	 For the last 1000 iterations: mean AUC=0.82 max AUC=0.85 min AUC=0.76	 For the last 1000 iterations: mean AUC=0.82 max AUC=0.85 min AUC=0.76
Train Curves	 For the last 1000 iterations: mean AUC=0.98 max AUC=0.99 min AUC=0.98	

Predictions

```
0.588790 | 1 set1/159/rec.gninatypes set1/159/docked_0.gninatypes
0.010623 | 0 set1/159/rec.gninatypes set1/159/docked_1.gninatypes
0.030514 | 0 set1/159/rec.gninatypes set1/159/docked_2.gninatypes
0.000665 | 0 set1/159/rec.gninatypes set1/159/docked_3.gninatypes
0.164526 | 1 set1/159/rec.gninatypes set1/159/docked_4.gninatypes
0.085010 | 0 set1/159/rec.gninatypes set1/159/docked_5.gninatypes
0.068489 | 0 set1/159/rec.gninatypes set1/159/docked_6.gninatypes
0.159224 | 0 set1/159/rec.gninatypes set1/159/docked_7.gninatypes
0.006736 | 0 set1/159/rec.gninatypes set1/159/docked_9.gninatypes
0.014252 | 0 set1/159/rec.gninatypes set1/159/docked_11.gninatypes
0.092240 | 0 set1/159/rec.gninatypes set1/159/docked_13.gninatypes
```

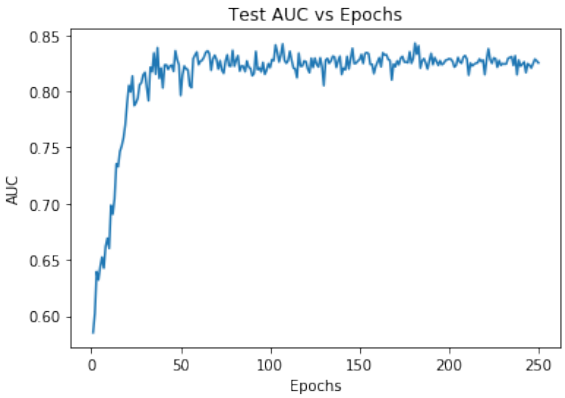
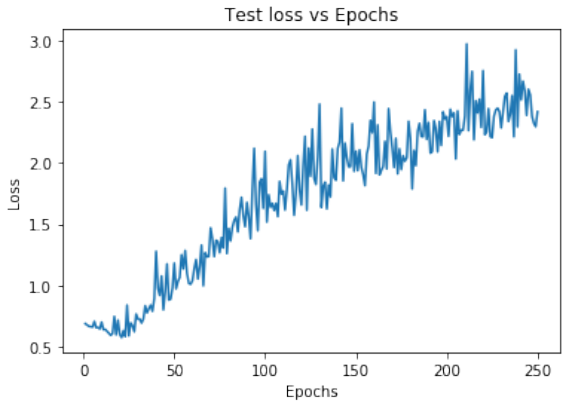
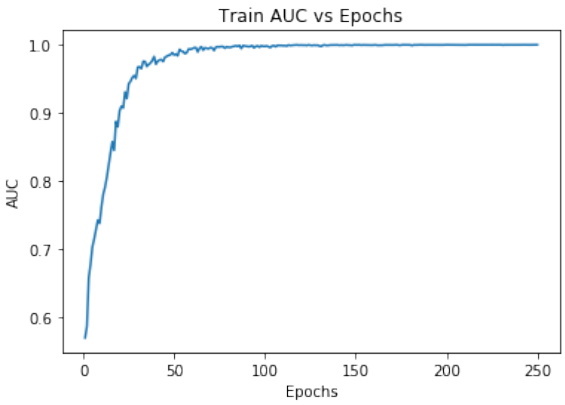
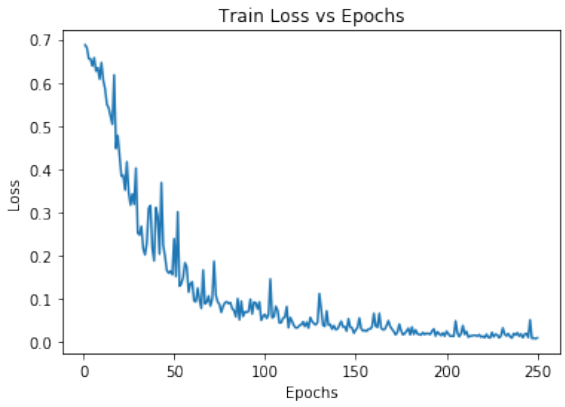
Results

Accuracy	0.846
Sensitivity	0.714
Specificity	0.876
Precision	0.566
F1 Score	0.632
MCC	0.542
AUC	0.902



6.1.2 CSAR Balanced dataset

Test and Train Curves

Test AUC	
Test Loss	
Train AUC	
Train Loss	

Predictions on pre-trained weights

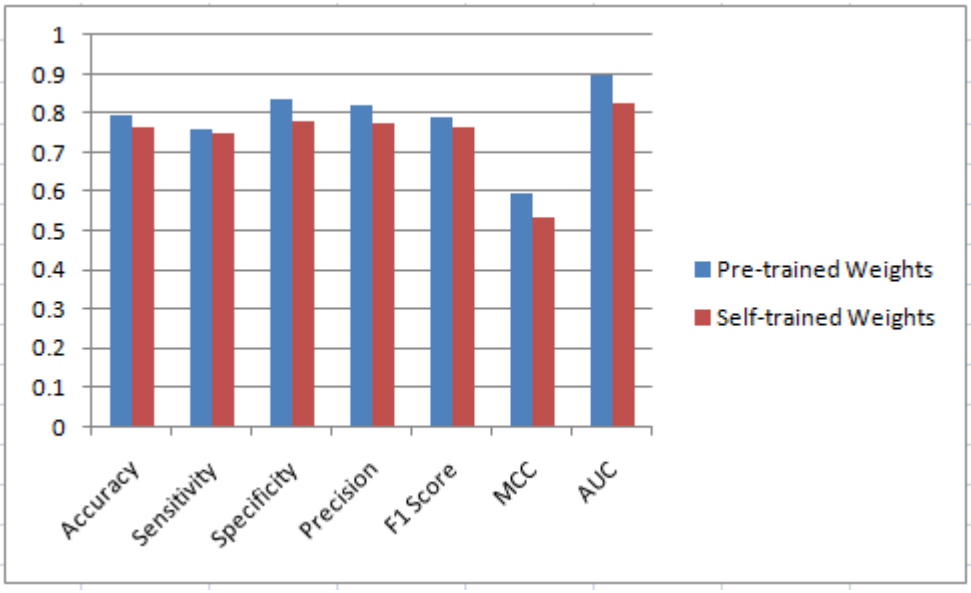
```
0.588790 | 1 set1/159/rec.gninatypes set1/159/docked_0.gninatypes #
0.092240 | 0 set1/159/rec.gninatypes set1/159/docked_13.gninatypes
0.817471 | 1 set1/156/rec.gninatypes set1/156/docked_0.gninatypes #
0.000046 | 0 set1/156/rec.gninatypes set1/156/docked_19.gninatypes
0.978913 | 1 set2/205/rec.gninatypes set2/205/docked_1.gninatypes #
0.365104 | 0 set2/205/rec.gninatypes set2/205/docked_19.gninatypes
0.586794 | 1 set1/157/rec.gninatypes set1/157/docked_0.gninatypes #
0.145610 | 0 set1/157/rec.gninatypes set1/157/docked_19.gninatypes
0.396480 | 1 set1/71/rec.gninatypes set1/71/docked_5.gninatypes # 0
```

Predictions on self-trained weights

```
0.999973 | 1 set1/159/rec.gninatypes set1/159/docked_0.gninatypes
0.999992 | 0 set1/159/rec.gninatypes set1/159/docked_13.gninatype
1.000000 | 1 set1/156/rec.gninatypes set1/156/docked_0.gninatypes
0.000141 | 0 set1/156/rec.gninatypes set1/156/docked_19.gninatype
1.000000 | 1 set2/205/rec.gninatypes set2/205/docked_1.gninatypes
1.000000 | 0 set2/205/rec.gninatypes set2/205/docked_19.gninatype
1.000000 | 1 set1/157/rec.gninatypes set1/157/docked_0.gninatypes
0.008663 | 0 set1/157/rec.gninatypes set1/157/docked_19.gninatype
0.791656 | 1 set1/71/rec.gninatypes set1/71/docked_5.gninatypes #
```

Results

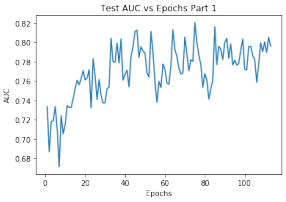
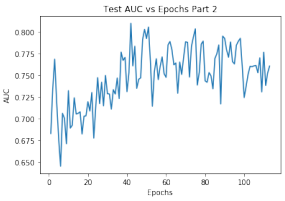
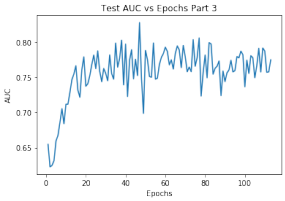
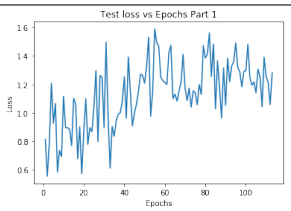
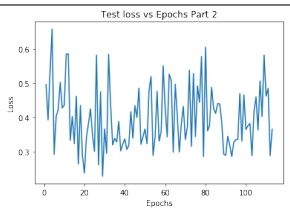
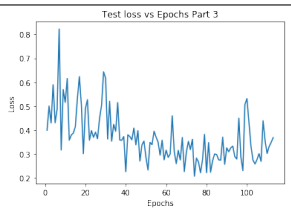
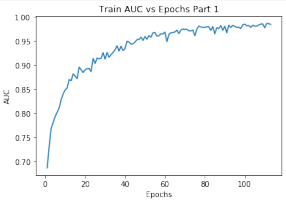
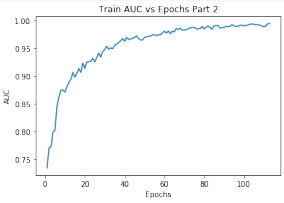
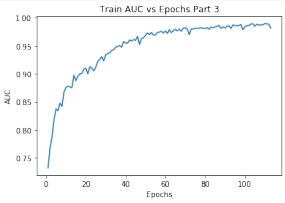
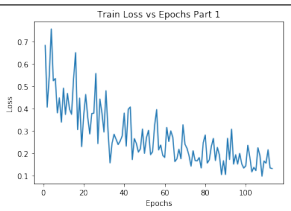
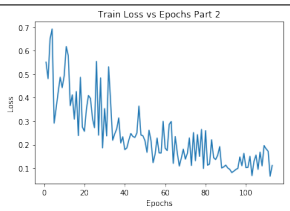
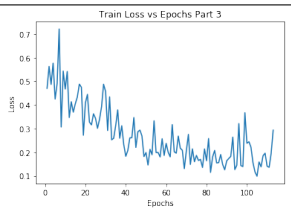
	Pre-trained Weights	Self-trained Weights
Accuracy	0.797	0.766
Sensitivity	0.760	0.750
Specificity	0.833	0.781
Precision	0.820	0.774
F1 Score	0.789	0.762
MCC	0.595	0.532
AUC	0.898	0.824



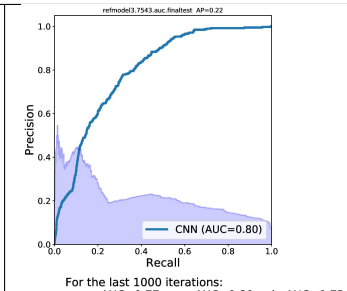
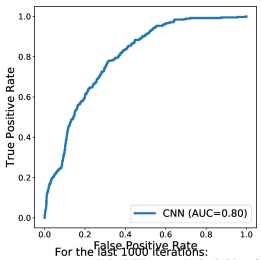
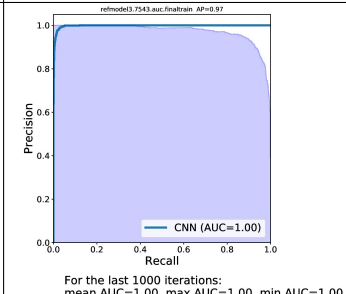
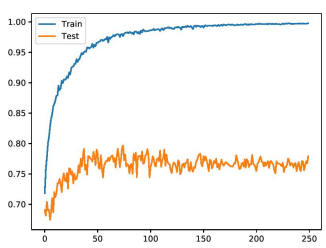
6.2 General Dataset Training

I trained a General dataset provided by the authors on the same network as used above in CSAR training using 3-fold cross validation.

Curves for each Fold

Metric/Fold	Fold 1	Fold 2	Fold 3
Test AUC			
Test Loss			
Train AUC			
Train Loss			

Test and Train Curves

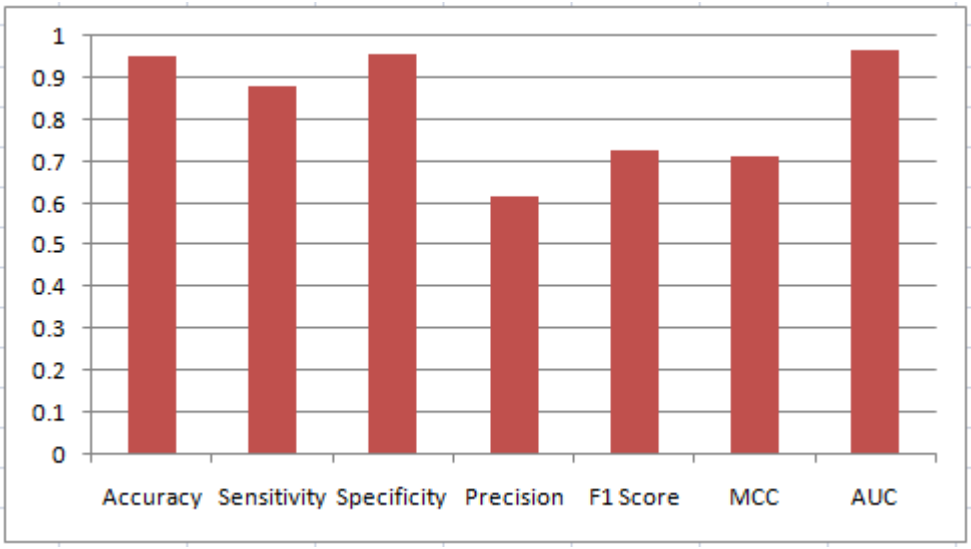
Test Curves	 For the last 1000 iterations: mean AUC=0.77 max AUC=0.80 min AUC=0.73	 For the last 1000 iterations: mean AUC=0.77 max AUC=0.80 min AUC=0.73
Train Curves	 For the last 1000 iterations: mean AUC=1.00 max AUC=1.00 min AUC=1.00	

Predictions

```
0.000014 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_1.gninatypes
0.053690 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_2.gninatypes
0.000007 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_3.gninatypes
0.000228 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_4.gninatypes
0.000128 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_5.gninatypes
0.000091 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_7.gninatypes
0.000113 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_8.gninatypes
0.190215 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_9.gninatypes
0.000423 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_10.gninatypes
0.000023 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_11.gninatypes
0.000729 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_12.gninatypes
```

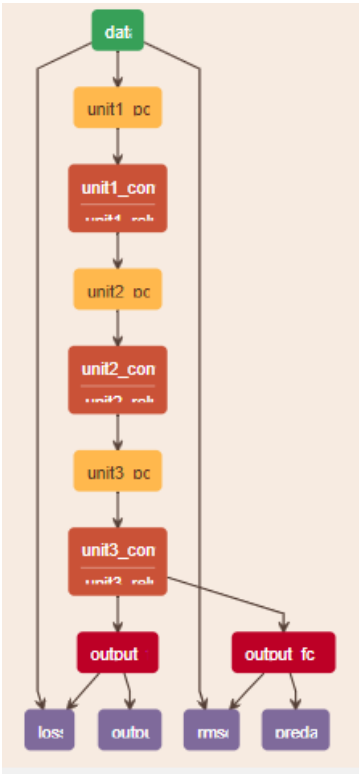
Results

Accuracy	0.952
Sensitivity	0.878
Specificity	0.958
Precision	0.617
F1 Score	0.724
MCC	0.712
AUC	0.968



6.3 Affinity Model Testing

I tested various datasets on the affinity model to predict the binding affinities of receptor-ligand binding. The following model provided by the authors was used -



6.3.1 General Dataset

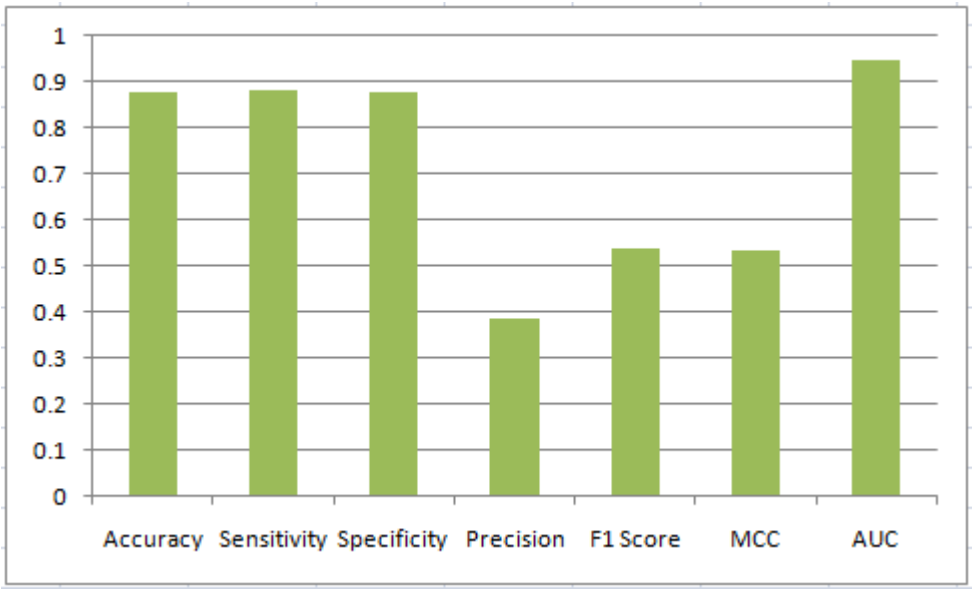
Predictions

```
0.014886 4.315974 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_1.gninatypes
0.779990 5.628354 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_2.gninatypes
0.000007 2.470673 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_3.gninatypes
0.000406 3.394661 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_4.gninatypes
0.009257 4.233458 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_5.gninatypes
0.024640 4.406263 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_7.gninatypes
0.003725 3.757120 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_8.gninatypes
0.790298 6.137585 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_9.gninatypes
0.001585 3.390300 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_10.gninatypes
0.000107 2.734729 | 0 -5.2200 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_11.gninatypes
```

It can be seen here that the predictions contain both the probability of binding (Column 1) and affinity prediction (Column 2).

Binder/non-binder Results

Accuracy	0.879
Sensitivity	0.884
Specificity	0.879
Precision	0.388
F1 Score	0.539
MCC	0.536
AUC	0.950



Binding Affinity Results

Binding Affinity Avg Loss	1.26
RMSD	2.52

6.4 2:1 DUDE:CSAR Dataset

In this experiment, the model used in CSAR training was trained on 2:1 DUDE:CSAR data. The pre-trained weights were used to test on 2 datasets - General dataset and CSAR All dataset.

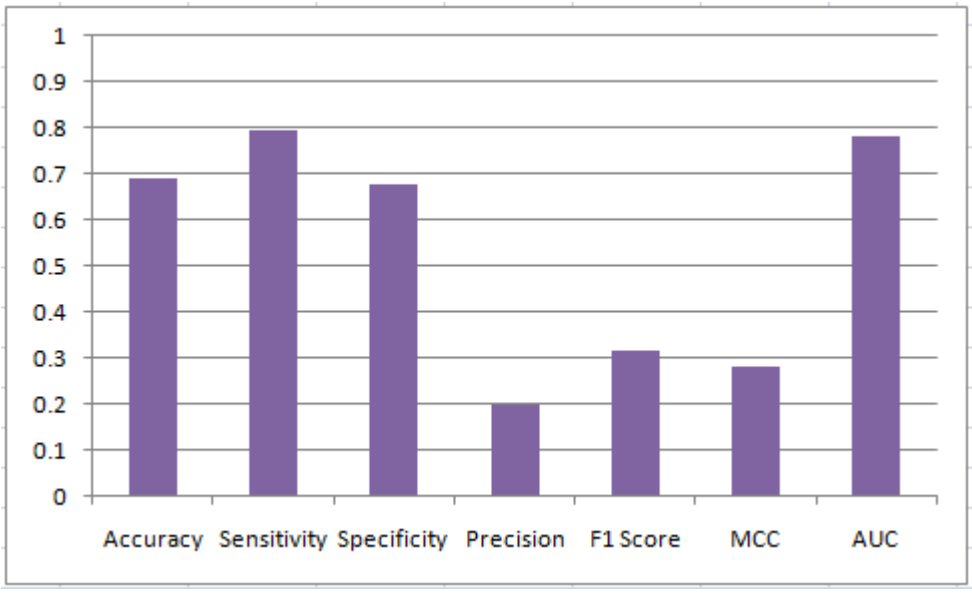
6.4.1 General Dataset

Predictions

```
0.048232 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_1.gninatypes
0.408276 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_2.gninatypes
0.046899 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_3.gninatypes
0.235612 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_4.gninatypes
0.077797 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_5.gninatypes
0.029147 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_7.gninatypes
0.031668 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_8.gninatypes
0.754530 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_9.gninatypes
0.522344 | 0 16pk/16pk_rec.gninatypes 16pk/16pk_ligand_10.gninatypes
```

Results

Accuracy	0.689
Sensitivity	0.780
Specificity	0.678
Precision	0.197
F1 Score	0.316
MCC	0.282
AUC	0.782



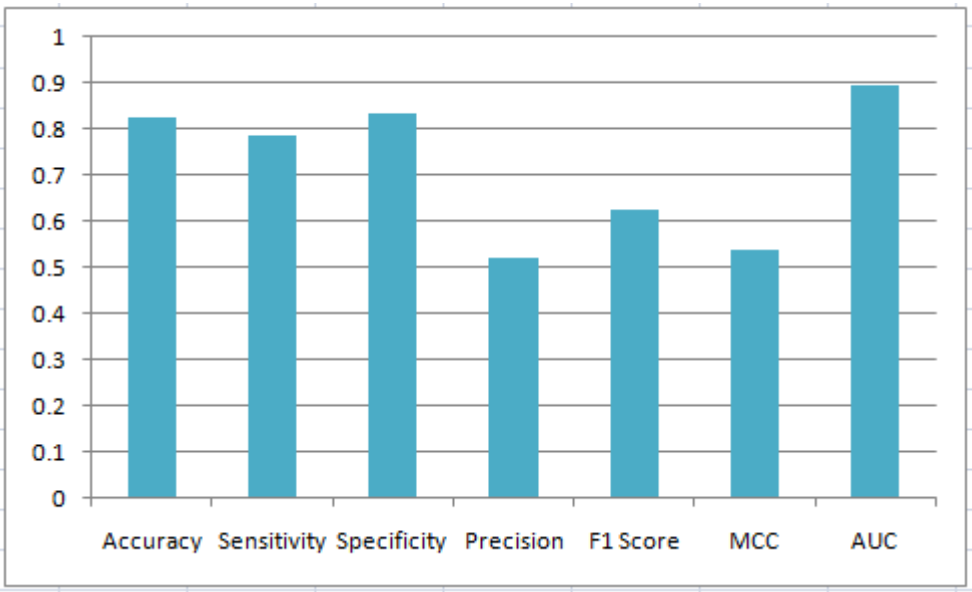
6.4.2 CSAR All Dataset

Predictions

```
0.765258 | 1 set1/159/rec.gninatypes set1/159/docked_0.gninatypes # 1.340370 -10.122400
0.291092 | 0 set1/159/rec.gninatypes set1/159/docked_1.gninatypes # 7.048960 -9.855450
0.337537 | 0 set1/159/rec.gninatypes set1/159/docked_2.gninatypes # 5.218720 -9.642270
0.025988 | 0 set1/159/rec.gninatypes set1/159/docked_3.gninatypes # 5.695560 -9.599200
0.489556 | 1 set1/159/rec.gninatypes set1/159/docked_4.gninatypes # 1.705850 -9.536100
0.698177 | 0 set1/159/rec.gninatypes set1/159/docked_5.gninatypes # 9.192180 -9.486370
0.027133 | 0 set1/159/rec.gninatypes set1/159/docked_6.gninatypes # 9.563520 -9.475520
0.360462 | 0 set1/159/rec.gninatypes set1/159/docked_7.gninatypes # 7.947470 -9.453060
0.008984 | 0 set1/159/rec.gninatypes set1/159/docked_9.gninatypes # 8.909340 -9.347500
0.307089 | 0 set1/159/rec.gninatypes set1/159/docked_11.gninatypes # 4.000570 -9.245450
0.393906 | 0 set1/159/rec.gninatypes set1/159/docked_13.gninatypes # 9.867850 -9.186520
0.069911 | 0 set1/159/rec.gninatypes set1/159/docked_14.gninatypes # 9.149710 -9.173460
0.007634 | 0 set1/159/rec.gninatypes set1/159/docked_15.gninatypes # 5.010950 -9.167310
0.165120 | 0 set1/159/rec.gninatypes set1/159/docked_16.gninatypes # 9.430620 -9.162590
0.017999 | 0 set1/159/rec.gninatypes set1/159/docked_17.gninatypes # 9.747710 -9.111870
```

Results

Accuracy	0.827
Sensitivity	0.788
Specificity	0.836
Precision	0.522
F1 Score	0.628
MCC	0.540
AUC	0.894



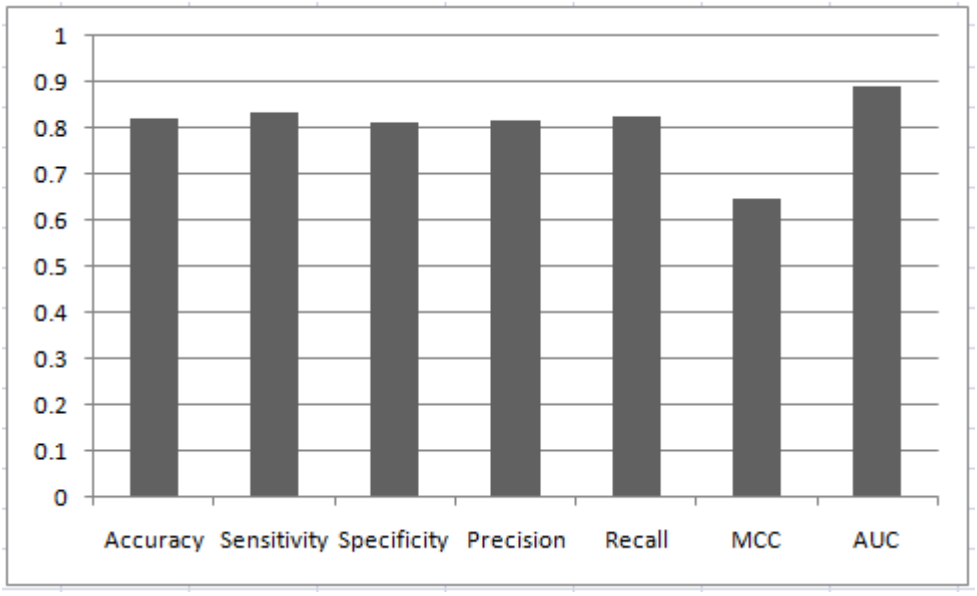
6.4.3 CSAR Balanced Dataset

Predictions

```
0.765258 | 1 set1/159/rec.gninatypes set1/159/docked_0.gninatype
0.393906 | 0 set1/159/rec.gninatypes set1/159/docked_13.gninatyp
0.826082 | 1 set1/156/rec.gninatypes set1/156/docked_0.gninatype
0.003223 | 0 set1/156/rec.gninatypes set1/156/docked_19.gninatyp
0.904438 | 1 set2/205/rec.gninatypes set2/205/docked_1.gninatype
0.110552 | 0 set2/205/rec.gninatypes set2/205/docked_19.gninatyp
0.912413 | 1 set1/157/rec.gninatypes set1/157/docked_0.gninatype
0.126138 | 0 set1/157/rec.gninatypes set1/157/docked_19.gninatyp
0.654405 | 1 set1/71/rec.gninatypes set1/71/docked_5.gninatypes
0.060174 | 0 set1/71/rec.gninatypes set1/71/docked_7.gninatypes
```

Results

Accuracy	0.824
Sensitivity	0.834
Specificity	0.813
Precision	0.817
F1 Score	0.826
MCC	0.648
AUC	0.893



7 Conclusion

The objective of this work was to find if a given pair of receptor-ligand will bind or not and if so with what binding affinity. Ligands are docked in different poses and the network learns the suitable poses where binding can occur. The authors have conducted a range of experiments which I have tried to replicated. I trained and tested 3 different networks on CSAR All, CSAR Balanced and General Datasets. Alongside, I also tested the Affinity prediction model, 2:1 DUDE:CSAR model on multiple datasets and tabulated the results. The authors have only reported AUC score for all their experiments but I have reported several other metrics like Accuracy, Sensitivity, Specificity, Precision, F1 Score, MCC for a threshold of 0.5 and also reported Average Loss and RMSD for the affinity prediction task. I derive the following analysis from my experiments:

- On CSAR dataset, the reported test AUC is 0.815. The model is trained on part of CSAR and tested on a separtae part. I attained a similar AUC of 0.824 by training and testing on Balanced CSAR dataset. The slight difference in value where I am getting a slightly higher score could be since I tested on fewer examples.
- For the 2:1 DUDE:CSAR dataset training, the authors have reported 0.83 AUC and I reached an AUC of 0.89. This difference could be because I only tested on CSAR since DUDE ligand poses and ground truth data was not available. However, it is seen that since the model is trained on DUDE and CSAR datasets, it is not able to perform as well on an external dataset like the General dataset where it merely achieves 0.78 AUC.
- I trained a network using the General dataset. Here, the network is trained on part data and tested on the other part. The network obtained an very high AUC of 0.96. This indicates that the models are capable of learning the nature of the data of train set and is able to generalize it to similar test data to give accurate predictions.
- I also tested the affinity model which along with predicting the binder/non binder binary label, also estimates the binding affinity. This model was pretrained on the General dataset hence it was able to generate good results when tested on General dataset test set. The average loss and average RMSD were as low as 1.26 and 2.5 for around 4000 predictions.

Overall, the models perform very well. However, there are a few signs of overfitting which can be seen through the validation curves. Despite the 3D representation, the input is not as complex as images hence the network learns much faster as compared to usual CNN training on images.

As part of future work, newer datasets can be used and developed. Also, metadata such as pH, temepature, ionic strength which impact the receptor-ligand interaction can be factored-in to enable better predictions. Finally, presence of inhibitors and determining selectivity among competing ligands can be explored to develop a system in a more realistic context.