Quick, Draw! Image Classification

Group 3

Problem to Solve: Quick, Draw! Image Classification

We focus on a subset of the Kaggle competition

- Image classification
-

Data

- Google originally collects data from the user drawings.
- We used .npy files that Google has provided.
- Compressed after using ….. Algorithm.
- >100,000 rows with 784 features.

Our Strategy

Neural Network Architecture
- Fully Connected Neural Network
- Convolutional Neural Network (CNN)

Bias vs. Variance
- Overfitting
- Underfitting

Multi-Class Classification
- Accuracy vs. Number of Classes
- Accuracy vs. Datapoints
- Imbalanced classes

Neural Network Architecture

Human classification

- 85% accuracy
- About 30 mins to complete task

Neural Network Architecture

Fully Connected Neural Network

- Designed with Tensorflow
- 3 layers (ReLU and Softmax activation functions)
- Issues with overfitting
    - Regularization using dropout
    - Added training data
- Results
    - 75% accuracy TODO
    - TODO: cost over epochs

Neural Network Architecture

Convolutional Neural Network (CNN)

- Designed using Keras with Tensorflow backend.
- Tried out different architectures.
- For e.g., changed number of layers, number of filters, added/removed dropout
- Results

Evaluation of model

Fully connected

- Human comparison - 2 team members manually classified 100 images.
- Accuracy 85% and x%
- Our model beat us.
- ROC curves.
- Accuracy
- Confusion matrix.
- Loss measures.

Evaluation of model

CNN using keras

- Human comparison - 2 team members manually classified 100 images.
- Accuracy 85% and x%
- Our model beat us.
- ROC curves.
- Accuracy
- Confusion matrix.
- Loss measures.

Bias vs. Variance

Seeing the Big Picture

* Neural Networks differ from many other models in that bias vs. variance is less of a tradeoff
  - We can fix bias, then separately fix variance

Overfitting

* overfitting is bad and here's how it happens
* here's how we dealt with it

Underfitting

* also bad, here's how we deal with it

Multi-Class Classification

- We classify multiple quickly drawn images.

Accuracy vs. Number of Classes

- Accuracy seemed to go down as the number

of classes went up

Meta Analysis of Neural Network Underfit

We increase the **number of categories** the Neural Network had to model, holding the number of **example per categories** constant.

We increase **example per categories** holding the **number of categories constant**.

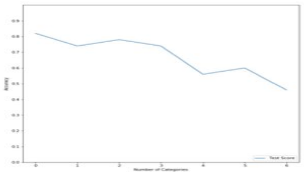
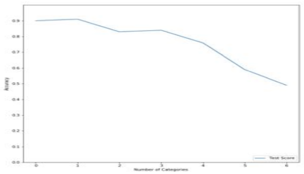
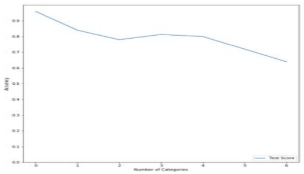
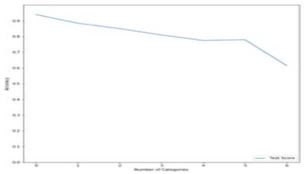
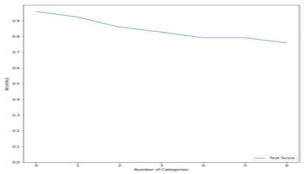
We recorded our accuracy observations for each permutation of the above in steps of 100, from 200 to 2000 observations and in steps of 1, for 3 to 10 categories

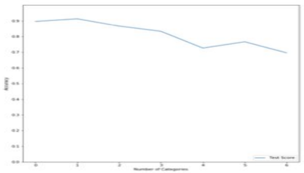
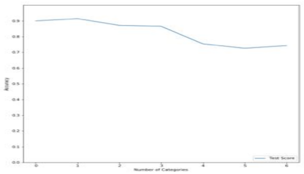
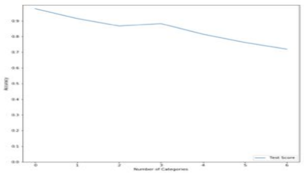
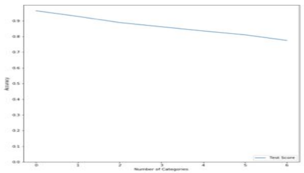
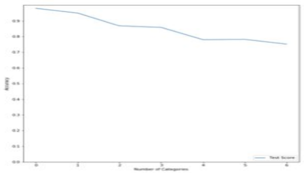
Neural Network Specifications

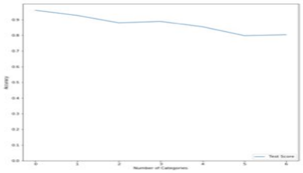
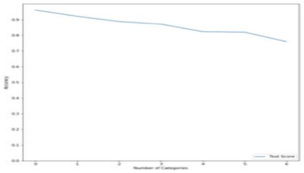
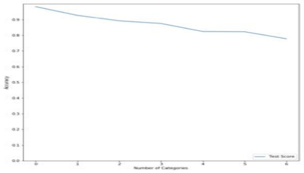
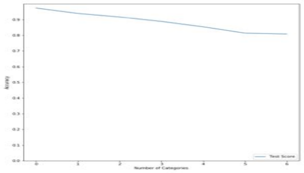
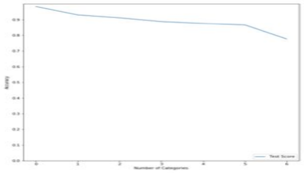
Layer (type)	Output Shape	Param #
conv2d_589 (Conv2D)	(None, 28, 28, 32)	320
conv2d_590 (Conv2D)	(None, 28, 28, 64)	8256
max_pooling2d_197 (MaxPoolin	(None, 14, 14, 64)	0
dropout_589 (Dropout)	(None, 14, 14, 64)	0
conv2d_591 (Conv2D)	(None, 13, 13, 64)	16448
dropout_590 (Dropout)	(None, 13, 13, 64)	0
flatten_197 (Flatten)	(None, 10816)	0
dense_393 (Dense)	(None, 128)	1384576
dropout_591 (Dropout)	(None, 128)	0
dense_394 (Dense)	(None, c)	387

c = [3,4,5,6,7,8,9]
epochs = 10
batch_size = 16

Meta Analysis Results

100	200	300	400	500
				

600	700	800	900	1000
				

1100	1200	1300	1400	1500
				

Meta Analysis Results

OLS Regression Results

click to scroll output; double click to hide		**R-squared:**	0.793
Model:	OLS	**Adj. R-squared:**	0.790
Method:	Least Squares	**F-statistic:**	249.5
Date:	Wed, 05 Dec 2018	**Prob (F-statistic):**	3.10e-45
Time:	20:59:47	**Log-Likelihood:**	227.80
No. Observations:	133	**AIC:**	-449.6
Df Residuals:	130	**BIC:**	-440.9
Df Model:	2		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	0.9628	0.014	67.145	0.000	0.934	0.991
no_categories	-0.0356	0.002	-18.620	0.000	-0.039	-0.032
rows_per_category	8.626e-05	6.99e-06	12.343	0.000	7.24e-05	0.000

Omnibus:	63.988	**Durbin-Watson:**	1.193
Prob(Omnibus):	0.000	**Jarque-Bera (JB):**	231.261
Skew:	-1.783	**Prob(JB):**	6.06e-51
Kurtosis:	8.386	**Cond. No.**	4.63e+03

For every additional category, we can expect the test accuracy go down by 3.56%

For every additional datapoint, we can expect the accuracy to go up by .008626%

The results are statistically significant

Imbalanced Data

- We trained and tested the model with an imbalanced dataset.
- We trained the model using different distributions of 2 classes.
- Here are our results from iterations involving 100,000 samples of ‘fork’ and different number of samples for ‘hammer’.
- Training and testing accuracies are for the data used to fit the model and the last 2 columns show accuracy on completely unseen data (20000 samples).

Fork samples	Hammer samples	Train accuracy (%)	Test accuracy (%)	Fork accuracy (%)	Hammer Accuracy (%)
100000	1000	99	99	99	63
100000	5000	99	99	99	85
100000	25000	98	98	99	90
100000	50000	98	98	99	94
100000	75000	98	97	97	97
100000	100000	98	97	98	98

Imbalanced Data

- Similarly, here are our results from iterations involving 50,000 samples of ‘fork’ and different number of samples for ‘hammer’.
- Result- The confusion matrix showed that all the objects were predicted as a ‘fork’ for models having imbalanced datasets.
- As expected, having imbalanced data results in inaccurate results for the undersampled class for unseen data.

Fork samples	Hammer samples	Train accuracy (%)	Test accuracy (%)	Fork accuracy (%)	Hammer Accuracy (%)
50000	100	99	99	100	43
50000	1000	99	99	99	81
50000	5000	99	98	99	88
50000	10000	98	98	98	94
50000	25000	98	97	98	95
50000	50000	98	97	98	98

Summary

Results

Neural networks are good for classifying images containing a hotdog

- Fully connected Neural Network needed a lot more time to give an accuracy comparable to CNN. (1600 epochs vs 10 epochs).
- CNN classification > human classification > fully connected classification
- Imbalanced data-
- Bias vs Variance thing

DONE!!!!