Dog Breed Identification

Kaitlyn Mulligan
Marist College

May 8, 2019

Overview

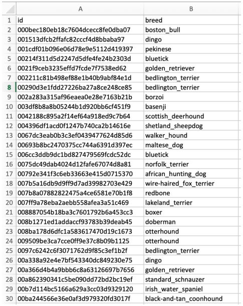
1. Motivation
2. Data
3. Initial Ideas
4. Xception and MLP
5. Implementation
6. Confusion Matrices
7. Predictions
8. References

Motivation

- Learn how to use a classification tool on images in order to classify the image
  - CNN
  - Xception
  - MLP
- In this project, the goal is to input an image of a dog and output the dog breed represented in the image.

Data

- Kaggle Dog Breed Identification
- 10,222 images
- 120 dog breeds
- CSV file of ID numbers of images for respective dog breeds



	A	B
1	id	breed
2	000bec180eb18c7604dcecc8fe0dba07	boston_bull
3	001513dfcb2ffafc82cccf4d8bbaba97	dingo
4	001cdf01b096e06d78e9e5112d419397	pekinese
5	00214f311d5d2247d5dfe4fe24b2303d	bluetick
6	0021f9ceb3235effd7fcde7f7538ed62	golden_retriever
7	002211c81b498ef88e1b40b9abf84e1d	bedlington_terrier
8	00290d3e1fdd27226ba27a8ce248ce85	bedlington_terrier
9	002a283a315af96eaea0e28e7163b21b	borzoi
10	003df8b8a8b05244b1d920bb6cf451f9	basenji
11	0042188c895a2f14ef64a918ed9c7b64	scottish_deerhound
12	004396df1acd0f1247b740ca2b14616e	shetland_sheepdog
13	0067dc3eab0b3c3ef0439477624d85d6	walker_hound
14	00693b8bc2470375cc744a6391d397ec	maltese_dog
15	006cc3ddb9dc1bd827479569fcdc52dc	bluetick
16	0075dc49dab4024d12fafe67074d8a81	norfolk_terrier
17	00792e341f3c6eb33663e415d0715370	african_hunting_dog
18	007b5a16db9d9ff9d7ad39982703e429	wire-haired_fox_terrier
19	007b8a07882822475a4ce6581e70b1f8	redbone
20	007ff9a78eba2aebb558afea3a51c469	lakeland_terrier
21	008887054b18ba3c7601792b6a453cc3	boxer
22	008b1271ed1addaccf93783b39deab45	doberman
23	008ba178d6dfc1a583617470d19c1673	otterhound
24	009509be3ca7cce0ff9e37c8b09b1125	otterhound
25	0097c6242c6f3071762d9f85c3ef1b2f	bedlington_terrier
26	00a338a92e4e7bf543340dc849230e75	dingo
27	00a366d4b4a9bbb6c8a63126697b7656	golden_retriever
28	00a862390341c5be090dd72bd2bc19ef	standard_schnauzer
29	00b7d114bc5166a629a3cc03d9329120	irish_water_spaniel
30	00ba244566e36e0af3d979320fd3017f	black-and-tan_coonhound

Initial Ideas

Convolutional Neural Network

1. 2 convolutional layers, 5 full connection layers
    - Full connection layers consisting of: relu, sigmoid, and dropout
    - Batch size of 512, 500 epochs
    - 10% accuracy rate

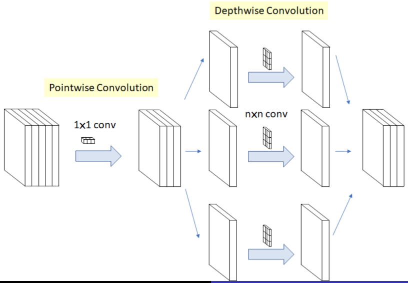
Initial Ideas

Convolutional Neural Network

1. 2 convolutional layers, 5 full connection layers
    - Full connection layers consisting of: relu, sigmoid, and dropout
    - Batch size of 512, 500 epochs
    - 10% accuracy rate
2. Variation of initial CNN, no convolutional layers
    - Wasn't working well
    - Decided to try Logistic Regression with Xception

Xception

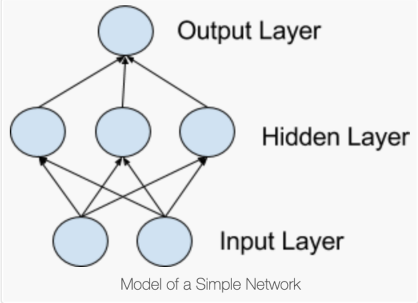
- 36 convolutional stages
- Depthwise separable convolutions
- Better performance due to more efficient use of model parameters
- Performs 1 by 1 convolution first, then the channel wise spatial convolution
- No intermediate ReLU non-linearity
- Without intermediate activation it has the highest accuracy compared to others



MLP

Multilayer Perceptron

- Deep, artificial neural network
- Composed of:
  1. Input layer which receives the signal
  2. Arbitrary number of hidden layers
  3. Output layer makes the decision/prediction regarding the input
- Trains on the data to learn the model
- Goal is to minimize the error by adjusting parameters



Implementation

1. Load Xception
2. Split into training and testing
3. Run training and testing through Xception

Implementation

1. Load Xception
2. Split into training and testing
3. Run training and testing through Xception

In order to find the best model, I needed the best set of neurons and best value of eta. Therefore, I made for-loops to save the best values for each of these. Before running it with everything, I wanted to make sure it all worked.

Implementation

Problems run into:

- Takes a long time to run
- Errors in code
- No time left

Based on these problems, my current model is not optimal. The program is now working, but I did not have time to run it with all of the neurons and values of eta.

Implementation

Based on what I have run so far:

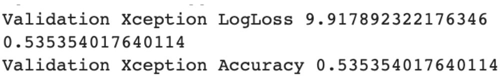
- Best set of neurons: (500,)
- Best value of eta: 0.001

Implementation

Based on what I have run so far:

- Best set of neurons: (500,)
- Best value of eta: 0.001

Using these values, I ran an MLPClassification to obtain a model. This resulted in:



```
Validation Xception LogLoss 9.917892322176346
0.535354017640114
Validation Xception Accuracy 0.535354017640114
```

Confusion Matrices

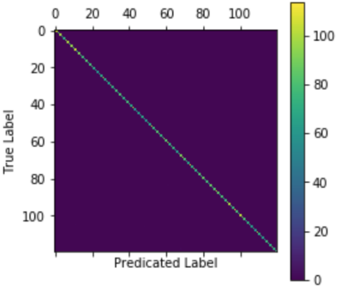


Figure 1: Training Confusion Matrix

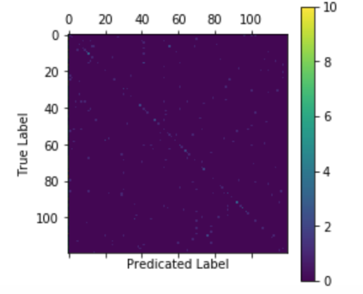
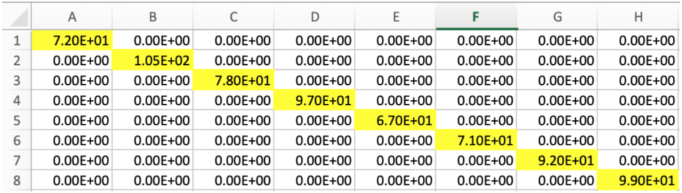


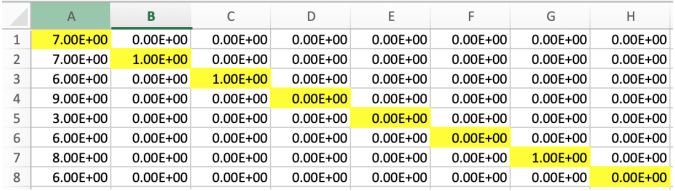
Figure 2: Testing Confusion Matrix

Confusion Matrices



	A	B	C	D	E	F	G	H
1	7.20E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	0.00E+00	1.05E+02	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	0.00E+00	0.00E+00	7.80E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	0.00E+00	0.00E+00	0.00E+00	9.70E+01	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	0.00E+00	0.00E+00	0.00E+00	0.00E+00	6.70E+01	0.00E+00	0.00E+00	0.00E+00
6	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	7.10E+01	0.00E+00	0.00E+00
7	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.20E+01	0.00E+00
8	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	9.90E+01

Figure 3: Training Confusion Matrix



	A	B	C	D	E	F	G	H
1	7.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
2	7.00E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
3	6.00E+00	0.00E+00	1.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
4	9.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
5	3.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
6	6.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00
7	8.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	1.00E+00	0.00E+00
8	6.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00	0.00E+00

Figure 4: Testing Confusion Matrix

Predictions with Test Data

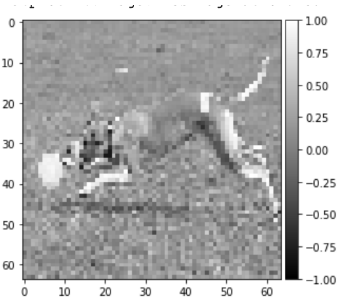
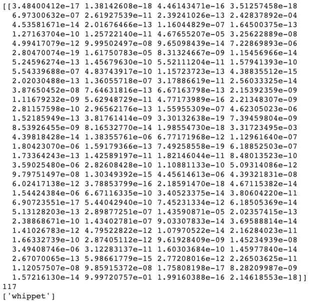
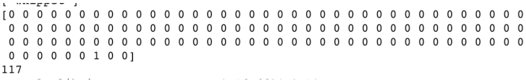


Figure 5: Whippet



```
[[3.48400412e-17 1.38142608e-18 4.46143471e-16 3.51257458e-18
  6.97300632e-07 2.61927539e-11 2.39241026e-13 2.42837892e-04
  4.53581671e-14 2.01676466e-13 1.16044829e-07 1.64500375e-13
  1.27163704e-10 1.25722140e-11 4.67655207e-05 3.25622889e-08
  4.99417079e-12 9.99502497e-08 9.65098439e-14 7.22869893e-06
  2.80470074e-19 1.61750783e-05 8.31324667e-09 1.15456966e-14
  5.24596274e-13 1.45679630e-10 5.52111204e-11 1.57941393e-10
  5.54339688e-07 4.83743917e-10 1.15723723e-13 4.38835512e-15
  2.02030488e-13 1.36055718e-07 3.17886619e-11 2.56033325e-14
  3.87650452e-08 7.64631816e-13 6.67163798e-13 2.15392359e-09
  1.11679232e-09 5.62948729e-11 4.77173989e-16 2.21348307e-09
  2.81157598e-10 2.96562176e-13 1.55955309e-07 4.62305023e-06
  1.52185949e-13 3.81761414e-09 3.30132638e-19 7.39459804e-09
  8.53926455e-09 8.16532770e-14 1.98554730e-18 3.31723495e-03
  4.39818428e-14 1.38355761e-06 6.77171968e-12 1.12961640e-07
  1.80423070e-06 1.59179366e-13 7.49258558e-19 6.18852503e-07
  1.73364243e-13 1.42589197e-11 1.82146044e-11 8.48013523e-10
  3.59025480e-06 2.82608428e-10 1.10881133e-10 5.09314086e-12
  9.79751497e-08 1.30349392e-15 4.45614613e-06 4.39321831e-08
  6.02417138e-12 3.78853799e-16 2.18591470e-18 4.67115382e-14
  1.54424384e-06 6.67116335e-10 3.40523375e-14 3.80604220e-11
  6.90723551e-17 5.44042940e-10 7.45231334e-12 6.18505369e-14
  5.13128203e-13 2.89877251e-07 1.43590871e-05 2.02357415e-13
  2.38868671e-10 1.43402781e-07 9.03307833e-14 3.69588814e-14
  1.41026783e-12 4.79522822e-12 1.07970522e-14 2.16284023e-11
  1.66332739e-10 2.87405112e-12 9.61928409e-09 1.45234939e-08
  3.49408746e-06 3.12283137e-11 1.60303684e-10 1.45977840e-14
  2.67070065e-13 5.98661779e-15 2.77208016e-12 2.26503625e-11
  1.12057507e-08 9.85915372e-08 1.75808198e-17 8.28209987e-09
  1.57216130e-14 9.99720757e-01 1.99160388e-16 2.14618553e-18]]
117
['whippet']
```

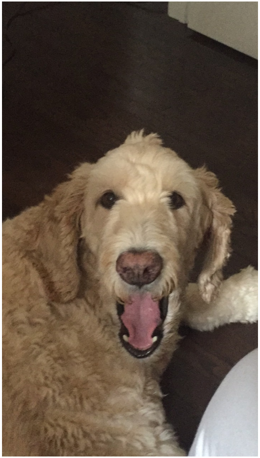
Figure 6: Prediction

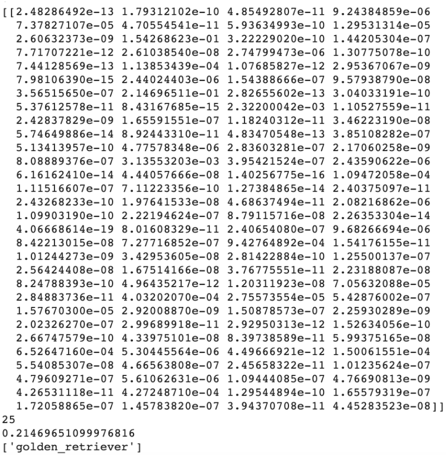


```
[0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
 0  0  0  0  0  0  1  0  0]
117
```

Figure 7: Actual

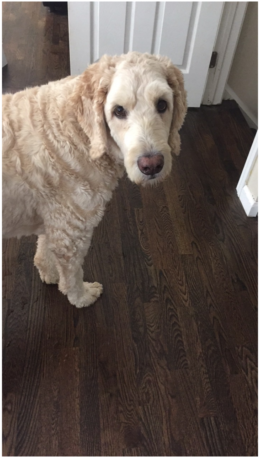
Predictions with Outside Data

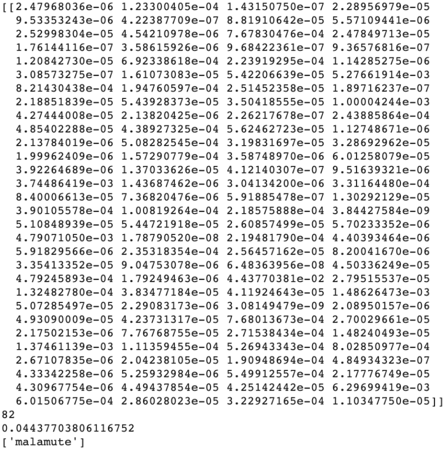




```
[[2.48286492e-13 1.79312102e-10 4.85492807e-11 9.24384859e-06
  7.37827107e-05 4.70554541e-11 5.93634993e-10 1.29531314e-05
  2.60632373e-09 1.54268623e-01 3.22229020e-10 1.44205304e-07
  7.71707221e-12 2.61038540e-08 2.74799473e-06 1.30775078e-10
  7.44128569e-13 1.13853439e-04 1.07685827e-12 2.95367067e-09
  7.98106390e-15 2.44024403e-06 1.54388666e-07 9.57938790e-08
  3.56515650e-07 2.14696511e-01 2.82655602e-13 3.04033191e-10
  5.37612578e-11 8.43167685e-15 2.32200042e-03 1.10527559e-11
  2.42837829e-09 1.65591551e-07 1.18240312e-11 3.46223190e-08
  5.74649886e-14 8.92443310e-11 4.83470548e-13 3.85108282e-07
  5.13413957e-10 4.77578348e-06 2.83603281e-07 2.17060258e-09
  8.08889376e-07 3.13553203e-03 3.95421524e-07 2.43590622e-06
  6.16162410e-14 4.44057666e-08 1.40256775e-16 1.09472058e-04
  1.11516607e-07 7.11223356e-10 1.27384865e-14 2.40375097e-11
  2.43268233e-10 1.97641533e-08 4.68637494e-11 2.08216862e-06
  1.09903190e-10 2.22194624e-07 8.79115716e-08 2.26353304e-14
  4.06668614e-19 8.01608329e-11 2.40654080e-07 9.68266694e-06
  8.42213015e-08 7.27716852e-07 9.42764892e-04 1.54176155e-11
  1.01244273e-09 3.42953605e-08 2.81422884e-10 1.25500137e-07
  2.56424408e-08 1.67514166e-08 3.76775551e-11 2.23188087e-08
  8.24788393e-10 4.96435217e-12 1.20311923e-08 7.05632088e-05
  2.84883736e-11 4.03202070e-04 2.75573554e-05 5.42876002e-07
  1.57670300e-05 2.92008870e-09 1.50878573e-07 2.25930289e-09
  2.02326270e-07 2.99689918e-11 2.92950313e-12 1.52634056e-10
  2.66747579e-10 4.33975101e-08 8.39738589e-11 5.99375165e-08
  6.52647160e-04 5.30445564e-06 4.49666921e-12 1.50061551e-04
  5.54085307e-08 4.66563808e-07 2.45658322e-11 1.01235624e-07
  4.79609271e-07 5.61062631e-06 1.09444085e-07 4.76690813e-09
  4.26531118e-11 4.27248710e-04 1.29544894e-10 1.65579319e-07
  1.72058865e-07 1.45783820e-07 3.94370708e-11 4.45283523e-08]]
25
0.21469651099976816
['golden_retriever']
```

Predictions with Outside Data





```
[[2.47968036e-06 1.23300405e-04 1.43150750e-07 2.28956979e-05
  9.53353243e-06 4.22387709e-07 8.81910642e-05 5.57109441e-06
  2.52998304e-05 4.54210978e-06 7.67830476e-04 2.47849713e-05
  1.76144116e-07 3.58615926e-06 9.68422361e-07 9.36576816e-07
  1.20842730e-05 6.92338618e-04 2.23919295e-04 1.14285275e-06
  3.08573275e-07 1.61073083e-05 5.42206639e-05 5.27661914e-03
  8.21430438e-04 1.94760597e-04 2.51452358e-05 1.89716237e-07
  2.18851839e-05 5.43928373e-05 3.50418555e-05 1.00004244e-03
  4.27444008e-05 2.13820425e-06 2.26217678e-07 2.43885864e-04
  4.85402288e-05 4.38927325e-04 5.62462723e-05 1.12748671e-06
  2.13784019e-06 5.08282545e-04 3.19831697e-05 3.28692962e-05
  1.99962409e-06 1.57290779e-04 3.58748970e-06 6.01258079e-05
  3.92264689e-06 1.37033626e-05 4.12140307e-07 9.51639321e-06
  3.74486419e-03 1.43687462e-06 3.04134200e-06 3.31164480e-04
  8.40006613e-05 7.36820476e-06 5.91885478e-07 1.30292129e-05
  3.90105578e-04 1.00819264e-04 2.18575888e-04 3.84427584e-09
  5.10848939e-05 5.44721918e-05 2.60857499e-05 5.70233352e-06
  4.79071050e-03 1.78790520e-08 2.19481790e-04 4.40393464e-06
  5.91829566e-06 2.35318354e-04 2.56457162e-05 8.20041670e-06
  3.35413352e-05 9.04753078e-06 6.48363956e-08 4.50336249e-05
  4.79245893e-04 1.79249463e-06 4.43770381e-02 2.79515537e-05
  1.32482780e-04 3.83477184e-05 4.11924643e-05 1.48626473e-03
  5.07285497e-05 2.29083173e-06 3.08149479e-09 2.08950157e-06
  4.93090009e-05 4.23731317e-05 7.68013673e-04 2.70029661e-05
  2.17502153e-06 7.76768755e-05 2.71538434e-04 1.48240493e-05
  1.37461139e-03 1.11359455e-04 5.26943343e-04 8.02850977e-04
  2.67107835e-06 2.04238105e-05 1.90948694e-04 4.84934323e-07
  4.33342258e-06 5.25932984e-06 5.49912557e-04 2.17776749e-05
  4.30967754e-06 4.49437854e-05 4.25142442e-05 6.29699419e-03
  6.01506775e-04 2.86028023e-05 3.22927165e-04 1.10347750e-05]]
82
0.04437703806116752
['malamute']
```

Goal

Continuing forward:

- Run it with all sets of neurons and values of eta
- Out of these, find and save the best values
- Run the MLPClassification again and obtain a higher accuracy rate (hopefully!)
- See how these new results compare to the current results

References

"A Beginner's Guide to Multilayer Perceptrons (MLP)," Skymind.

"Dog Breed Identification," Kaggle. [Online]. Available: https://www.kaggle.com/c/dog-breed-identification. [Accessed: 08-May-2019].

J. Brownlee, "When to Use MLP, CNN, and RNN Neural Networks," Machine Learning Mastery, 23-Jul-2018. [Online]. Available: https://machinelearningmastery.com/when-to-use-mlp-cnn-and-rnn-neural-networks/. [Accessed: 08-May-2019].

S.-H. Tsang, "Review: Xception - With Depthwise Separation Convolution, Better Than Inception-v3 (Image Classification)," Towards Data Science, 25-Sep-2018.

Thank you!

Questions?