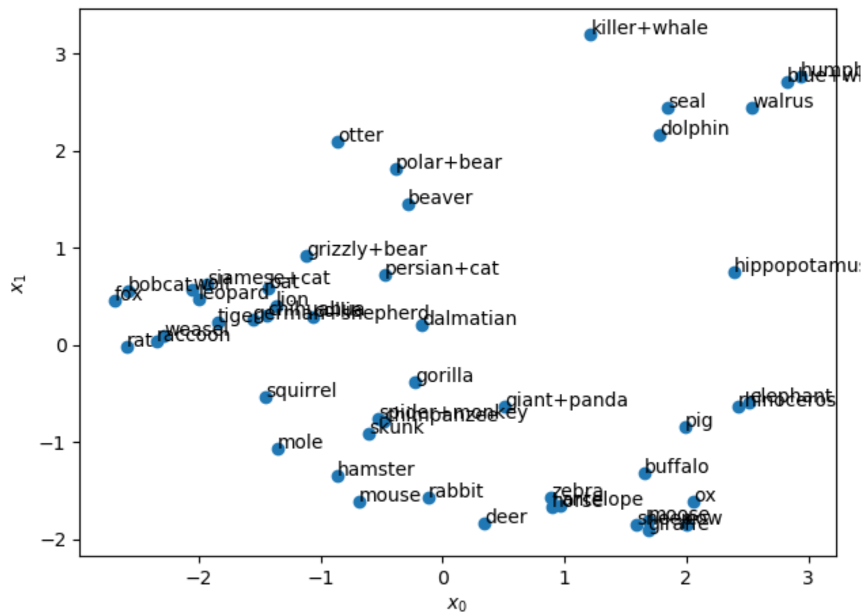
1.

1.1.



All the similar animals are presented together in the graph as observed from the above figure.

1.2.

1.2.1.

32% variance is explained by setting 2-dimentional representation

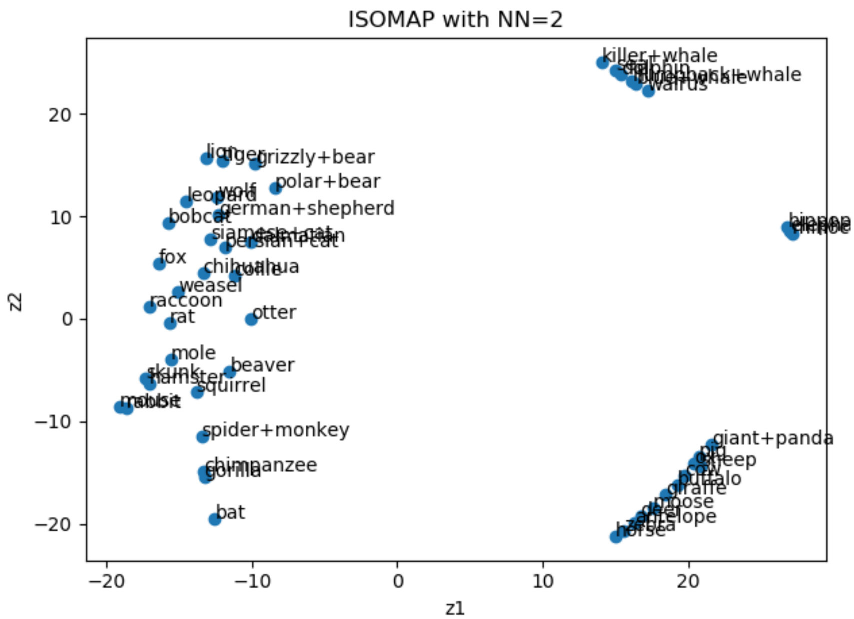
1.2.2.

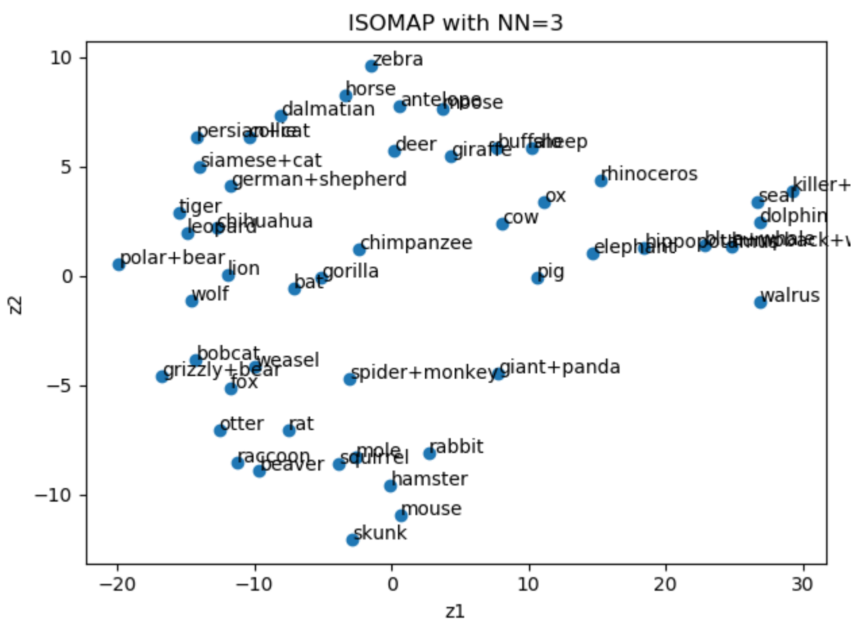
50% of variance is explained by setting at minimum 5-dimentional representation

1.3.

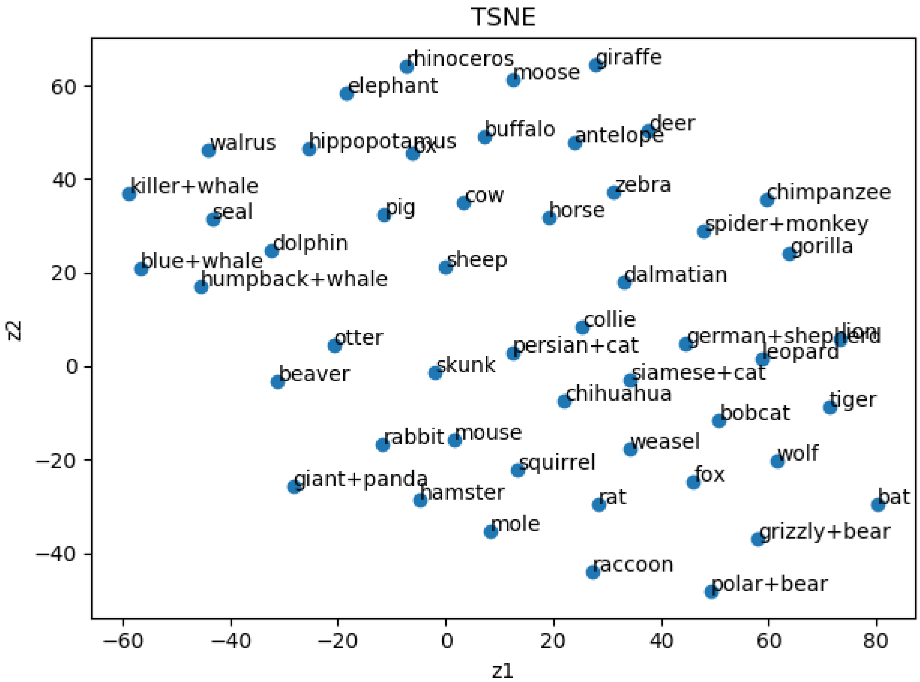
PCA produces a decent graph with most animals in the right group however MDS categorizes the animals comparatively better which is also seen from the MDS objective of MDS solution to be 1776.8 and PCA to be 4942.2.

1.4.


ISOMAP with NN=2
z1
z2
killer+whale
walrus
lion
grizzly+bear
polar+bear
leopard
wolf
bobcat
german+shepherd
fox
chihuahua
weasel
raccoon
rat
otter
mole
beaver
squirrel
spider+monkey
chimpanzee
gorilla
bat
giant+panda
buffalo
giraffe
moose
antelope
horse


ISOMAP with NN=3
z1
z2
zebra
horse
antelope
moose
dalmatian
deer
giraffe
sheep
siamese+cat
german+shepherd
rhinoceros
killer+
seal
ox
dolphin
tiger
leopard
chihuahua
cow
chimpanzee
polar+bear
lion
gorilla
bat
pig
wolf
walrus
bobcat
weasel
grizzly+bear
fox
spider+monkey
giant+panda
otter
rat
rabbit
raccoon
beaver
squirrel
mole
hamster
mouse
skunk

1.5.



According to the figures, ISOMAPS with NN set to 2 does the best as it sorts the various animals pretty well. PCA does a decent job however there are still certain errors with regards to grouping of animals. MDS does a better job compared to PCA and then TSNE produces a good graph however is not as differentiated.

1.6.
TSNE gives a different result and is prone to the initialization and it does map with the discussion in the lecture because TNSE is sensitive to heuristics

2. NAN

2.1.

$$z_i = \begin{matrix} -2 & 2 & -1 \\ 1 & -2 & 0 \end{matrix} X \begin{matrix} -3 \\ -2 \\ 2 \end{matrix}$$

$$= \begin{matrix} 6-4-2 \\ -3+4 \end{matrix} = \begin{matrix} 0 \\ 1 \end{matrix}$$

$$h_z = \begin{matrix} \frac{1}{1+e^{0}} \\ \frac{1}{1+e^{-1}} \end{matrix}$$

$$= \frac{\frac{1}{2}}{\frac{1}{1+\frac{1}{e}}} = \frac{\frac{1}{2}}{\frac{e}{e+1}} \approx \frac{1}{1}$$

$$y_i = \begin{matrix} 3 & 1 \end{matrix} X \begin{matrix} 1 \\ 1 \end{matrix}$$
$$= 4$$

2.2.
Training error: 0.08512
Test error: 0.0811

2.3.
I think stochastic gradient implementation is better compared to gradient descent implementation given the time taken for the code to run and result

2.4.
The changes that were tried were changing the number of hidden layers and the activation.
The first change was changing the hidden layers to (100,100) and the activation to logistic hence increasing the layer and activation to a sigmoid function however this increases the test error probably because of overfitting

The second change was changing the hidden layers to (200,100) and the activation to logistic, this reduced the test error to 0.0171 because of optimizing the function using sigmoid and more hidden layers

3.

3.1.
Yes, it is convex as only positive solutions are considered which are present in first quadrant of an already convex function

3.2.
Sparse solutions is wanted because zi is sparse so calculations are easier to calculate gradient and secondly, because certain parts of the wc objects are randomized

3.3.
ISOMAP is mainly used for unsupervised learning and it is non-parametric in nature

3.4.
Collaborative filtering is better for recommending movies to a new user, as content-based filtering requires an initial user input data compared to collaborative filtering which takes into account other users as well

3.5.
PCA reduces the dimension, whereas collaborative works with the given dimension moreover, collaborative tries to fill in missing values in the matrix whereas PCA does not

3.6.
Neural networks are supervised learning and are non-parametric in nature

3.7.
Regularization becomes more important as nets increases to avoid overfitting

3.8.
It does not matter if the loss went up or down because eventually the loss becomes centered as it goes through the examples, like the pendulum

3.9.
Input layer and hidden layer 10x100 = 1000 parameters with 100 biases makes 1100 parameters
Hidden layer and outer layer 100x3+3 = 300 parameters
Total 1403 parameters

3.10.
Neural network with ReLU activation, with hidden layer as size 1 will yield a convex loss function as, ReLU is convex and L2 function is convex.

3.11.
The vanishing gradient problem with neural networks based on sigmoid non-linearities is that gradients away from origin are reaching zero but never reach zero leading to slow training process and lower prediction accuracy

3.12.

CNN’s are used instead of regular neural networks because it reduces the number of parameters using filters which speeds up training and reduces overfitting