Random Projection in Deep Learing

Shai Kimhi

September 2021

1 Introduction

Link to project repository.

This work attempts to learn and verify the efficiency of Random Projection (RP) dimensionality reduction in Deep Learning.

Random Projection is a linear method of dimensionality reduction which is not dependant on the projected examples but just on their dimension and the size of the training dataset.

Deep Learning is based on Deep Neural Networks (DNNs) and through training tries to approximate a non-linear function that fits the dataset distributions, we will be using it for supervised learning tasks, the ability of DNN to learn and achieve satisfiying results on the projected examples of the dataset is also dependant on the capacity of the DNN model which is dependant among other factors mainly on the network's depth.

In this work there is an implementation of Fully Connected Deep Neural Networks with a Random Projection layer, the RP layer is not learned linear layer with weights initiation matching Random Projection initiation models.

This work examines different Random Projection initializations Such as Gaussian, Achlioptas, Li, Fast-Johnson Lindenstrauss also known as Subsampled Randomized Hadamard Transform (SRHT) (Nir Ailon and Bernard Chazelle 2006)

In order to examine the validity of RP for the mentioned DNN models, this work tests different real world datasets, including high-dimensional real datasets.

This work also attempts to comparse the efficiency of Deep Learning with Random Projection in comparison to both learning

with unprojected data and learning with Principle Component Analysis (PCA) dimensionality reduction. PCA is a linear but not random dimensionality reduction based on SVD of the dataset (after normalization), this method is best possible linear dimensionality reduction for a set of points as it preserves the presentation of vectors through the singular vectors matching the eigenvalues which represents the largest variance in the sampled dataset.

PCA is less computationally efficient and takes

$$\mathcal{O}(n^2m + m^3) \text{ time for } A \in R^{n \times m}$$

which can be non-feasible for certain datasets unlike Random Projection which could be much faster.

2 Deep Neural Networks

Deep Neural Networks are a unique and advanced Machine Learning model which allows to learn large varity of functions from examples without crafting new features, this model allows learning non-linear complex functions using several learnable linear layers with non-linear non-learnable activations between them. Every layer increases the number of operations the model needs to make in inference and the number of parameters in the model gradient.

For a long time, training Neural Networks with a large number of layers was impossible, espacially for a high-dimensional input, as the number of parameters in a linear fully connected layer is $N_{in} \times N_{out}$ where N_{in} is the input dimension to that layer, and N_{out} is the output dimension for that layer.

With developments in GPU parallelism modern computers can train these networks and espacially use a trained network in real time. That being said, it is still hard to train large models on large datasets, and thus methods of preprocessing dimensionality reduction for Machine Learning and specifically Deep Learning are present.

ReLU Rectifier

ReLU is a non-linear activation popular in deep learning, it is defined as $f(x) = x^+ = max(0, x)$. When this function is applied on a multi-dimensional tensor, the output is a tensor of the same dimensions with element-wise activation of this function.

Softmax

Softmax is a generalization of a logistic function for multiple dimensions, it is define as $\sigma_i(x) = \frac{e^{x_i}}{\sum_{j=1}^{K} e^{x_i}}$ where K is the size of the input vector. It is used mainly in multinomial classification and as so is useful with Deep Neural Networks of this type, it is usually applied as the final activation of a Neural Network in that case.

3 Random Projection

Random Projection is a computationally efficient and algorithmically simple technique of linear dimensionality reduction, this technique preserves the L2 norm of every vector from a given closed set of vectors with a certain probability and up to a certain small possible error. Therefore and since of the linearity of the method, the method preserves as well the distance between any two points selected from the set, this property allows different machine learning algorithms such as K-Nearest-Neighbours and Linear Regression on the projected datasets instead on original datasets while maintaining the same ability to learn from the dataset after projection and achieiving compatible results on the dataset as achieved on the original dataset.

Since Fully Connected Deep Neural Networks are comprised of several layers of learned linear projection and an activation after hidden layers, and from the given seperability preservation after linear projection, we can assume that DNNs could perform well on projected dataset, that could vary depending on the depth of the network.

In deep learnign in general there is also the problem of models are overfitting by learning function which represents the given train dataset perfectly but they do not perform as good on the dataset as a whole, usually models with the capacity to learn all the training dataset almost perfectly with no need to find a more generalized approximated function, are overfitting. This work will check wheter in certain occasions the test accuracy of a model with Random Projection layer could be better than the one of the same network without Random Projection layer.

Gaussian Random Projection

Let our dataset be matrix

$$A \in R^{n \times m}$$

and so our projected data will be

$$\hat{A} = MA^t \in R^{k \times m} \text{ s.t } M \sim N(0, 1/k)$$

Achlioptas Random Projection

Achliotas, 2001 proposed two different distributions for the Random Projection matrix, that could be calculated more efficiently than Gaussian Distribtuion, that is such that for RP matrix M:

$$M_{i,j} = \sqrt{\frac{1}{k}} \cdot \begin{cases} 1 & \text{w.p 1/2} \\ -1 & \text{w.p 1/2} \end{cases}$$

$$M_{i,j} = \sqrt{\frac{1}{k}} \cdot \begin{cases} 1 & \text{w.p 1/6} \\ 0 & \text{w.p 2/3} \\ -1 & \text{w.p 1/6} \end{cases}$$

We will be using the later the distribution as it allows a more sparse matrix and so the multiplication of this matrix with data vectors could be computationally easier.

Li Random Projection

Li et al, 2006 proposes a different distribution which includes Achlioptas distribution:

$$M_{i,j} = \sqrt{\frac{s}{k}} \cdot \begin{cases} 1 & \text{w.p 1/2s} \\ 0 & \text{w.p 1-1/s} \\ -1 & \text{w.p 1/2s} \end{cases}$$

This distribution matches Achlioptas distribution for s=3, Li et al showed that it is possible to choose an s as high as d/log d. In this work we use s = $\sqrt{d}$ as is used in Random Projection in Deep Neural Networks, 2018.

Fast Johson Lindenstrauss Transform(FJLT)

When encountaring sparse vectors in our dataset, other methods with sparse Random Projection matrix might suffer from distortions, Nir Ailon and Bernard Chazelle 2006 proposed a RP

matrix comprised of multiplication of 3 matrices such as one matrix acts as random densifier in order to prevent the method to be inacurate for sparse matrices. The RP matrix is defined:

$$M = \frac{1}{\sqrt{k}} SHD$$

where:

$$D \in R^{m \times m} \text{ such that } D_{i,j} = \begin{cases} 1 & \text{w.p 1/2} \\ -1 & \text{w.p 1/2} \end{cases}$$

H = $\sqrt{\frac{1}{m}} H_m \in R^{m \times m}$ where H_m is the Walsh-Hadmard matrix, for m which is not a power of d, we would padd the examples with zeros such that the examples vectors will be from dimension $2^{ceil(log(m))}$.

S $\in R^{k \times m}$ such that $S_{i,j} = \begin{cases} 0 & \text{w.p 1-q} \\ \sim N(0, 1/q) & \text{w.p q} \end{cases}$

Where q = $\mathcal{O}(\log^2(n)/d)$
In this work we chose q = $\log_2^2(n)/d$

This transform is also known as *Subsampled Randomized Hadamard Transform (SRHT)*

Count-Sketch

Count Sketch Algorithm was proposed first by Charikar et al, 2004, Weinberger et al 2009 proposed dimensionality reduction based on Count-Sketch and Dasgupta et al, 2010 introduced the Count-Sketch dimensionality reduction in the form of a matrix:

$$M = CD \text{ where } D \text{ is defined as in FJLT and}$$

$$C_{-,i} \sim Uni\{ \begin{pmatrix} 1 \\ 0 \\ ... \\ 0 \end{pmatrix}, ..., \begin{pmatrix} 0 \\ ... \\ 0 \\ 1 \end{pmatrix} \} \text{ is the i'th row of } C \in R^{k \times m}$$

The distribution of M, in which a single element is picked uniformly at random, and set to be either -1 or 1 at random, is also called Rademacher distribution.
The Count-Sketch algorithm could be activated in linear time i.e $\mathcal{O}$(nd), and Clarkson and Woodruff shows that, for sparse data, the time complexity can be decreased to $\mathcal{O}$(nnz(A)) where nnz(A) is the number of non-zero elements in the dataset matrix A.

Principal Component Analysis

Principal Component Analysis (PCA), is a Singular Value Decomposition based method for dimensionality reduction, given a set of vectors represented as a matrix $A \in R^{n \times m}$, the by evaluating the matrix $\widetilde{A}$ by reducing the average vector $\bar{a}$ from each one of the n vectors in A. Let: $\widetilde{A} = USV^T$

By calculating the Singular Value Decomposition of this matrix (Eigenvalue Decomposition is not always possible) and ordering the matrix V (the singular vectors matrix) by the order of the singular values from largerst to smallest and cropping the reordered matrix V to remain with first k columns, we are guaranteed the best possible preservation across all given vectors from A. When encountering a problem which is seperated to seen and unseen data, PCA dimensionality reduction will work if the unseen data distribtutes similarly to the seen data. Generally and in that issue in particular, PCA is a learning algorithm based on given data, and unlike Random Projection is derived directly from a known dataset, moreover, for large databases the PCA over all the datset is intractible and could be obtained through a large enough random sub sample of the dataset, for a too large number of features in each vector, PCA is harder to get for the entire dataset, in such cases random dimensionality reduction could be the only possible way to process the dataset.

4 Our Model

As we have stated in the *Introduction*, we propose a model of Deep Neural Network with a several hidden layers, a relu activation after each one of them, a dimensionality reduction layer as the first layer, not followed by activation and followed after (that is, since the RP layer is fixed and thus an activation would distort the data and make it difficult for the later layers to learn) and at the output of the network we have an output layer from the last hidden layer output dimension to the number of classes and a softmax layer after that, we used Cross Entropy Loss as the classification loss, as defined in *Deep Learning* section. Each hidden layer has an ouput of half of the input the layer has.

The dimensionality reduction method we use will be linear matrix multiplication, the matrix would be selected either randomly from a Random Projection distribution, or as a Principle Component Analysis matrix from the given dataset.

The models are trained on the different datasets through batch gradient descent with fixed batch size of 32 as commonly used in models training.

The reason for the dimesnionality reduction layer in the model is to reduce the amount of parameters of the network, this allows training of a deeper network, the dimesnionality reduction layer reorganizes the input in a manner which the network can still distinguish between different inputs, the network is allowed to overcome this reoganization in 10-20 iterations on the dataset, we are testing models with one hidden layer (in addition to an output layer), and with three hidden layers (a deeper model), the deeper model has more parameters, but not considerably more. Thus, allowing it to train with not much more time than the model with one hidden layer.

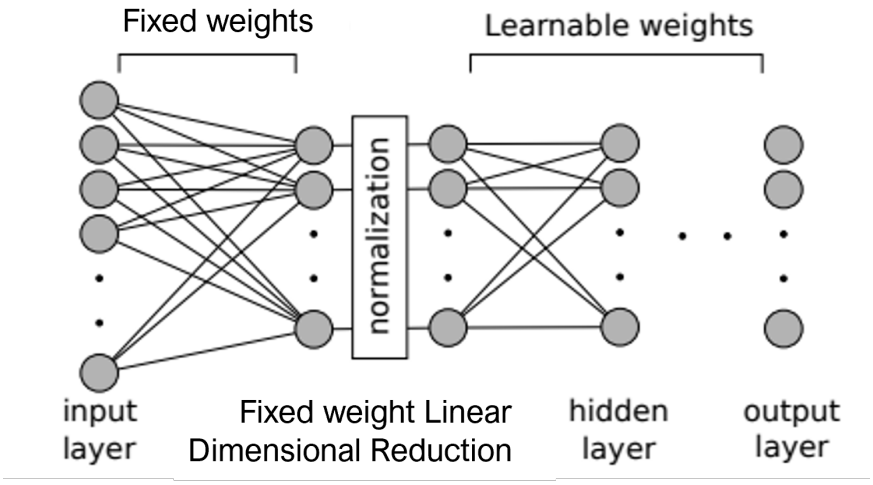

Figure 4.1: Deep Neural Network with fixed Linear Dimensionality reduction layer.

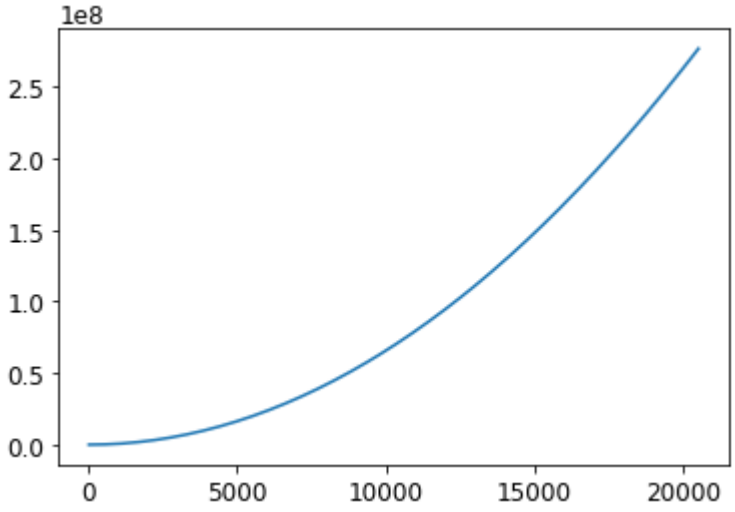

Figure 4.2: Model with three hidden layers number of trainable parameters depending on the input dimension (After RP dimensionality reduction).

The original input is transperant to the trained part of the network and as can be seen, for smaller dimensions the change of parameters is slow (due to the second and third hidden layers and the output layer changing less than the first hidden layer) later we see that the increase in trainable parameters is linear to the input of the network.

5 Experimental Results

Gene Expression cancer RNA-Seq Data Set

The dataset, is a real world dataset containing 801 examples each with 20531 features, the datset is split into training and test sets with equal ratio (training set contains 401 examples and the rest are in the test set). The data is fitting for classification problem where the different classes are mapped to integer from 0-4 (there are 5 different classes).

PRAD is mapped to 0, LUAD is mapped to 1, BRCA is mapped to 2, KIRC is mapped to 3 and COAD is mapped to 4.

Model with one hidden layer

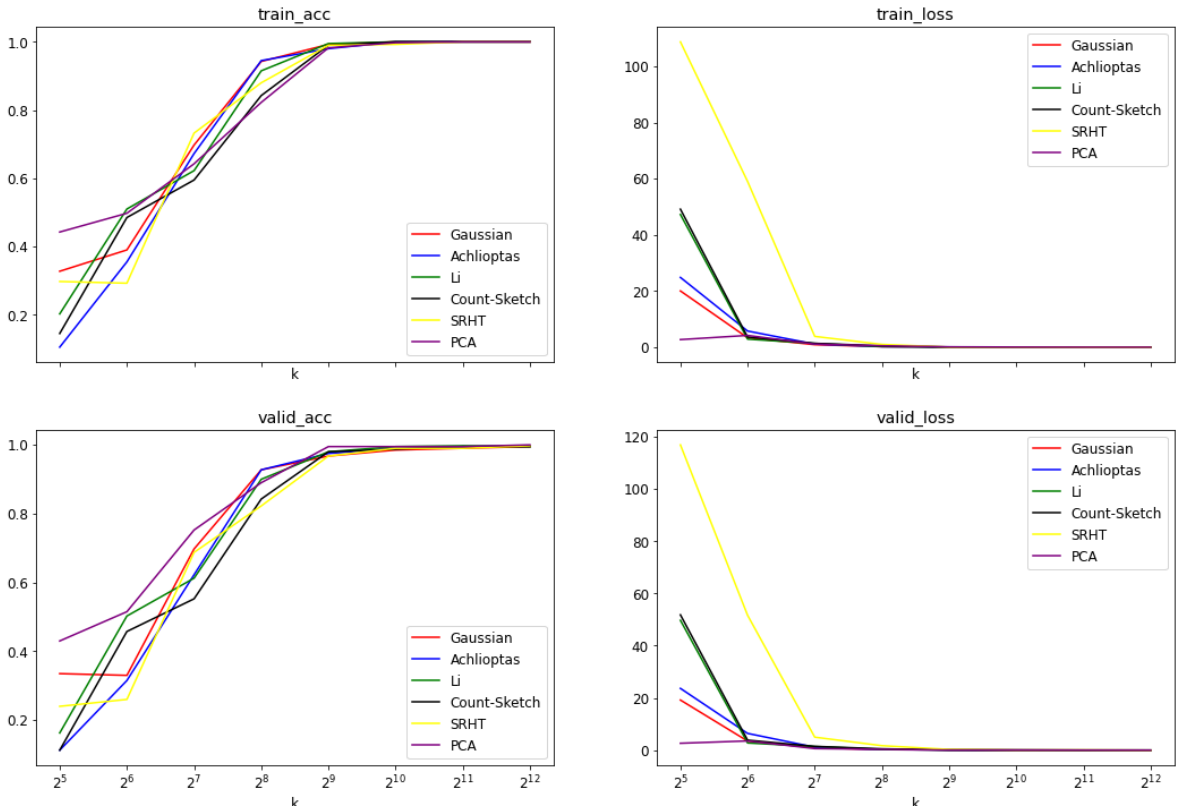

Figure 5.1: The accuracy and loss terms for both training and validation(test) sets, for projected dimension k between 2^5 and 2^{12}, for model with 1 hidden layer.

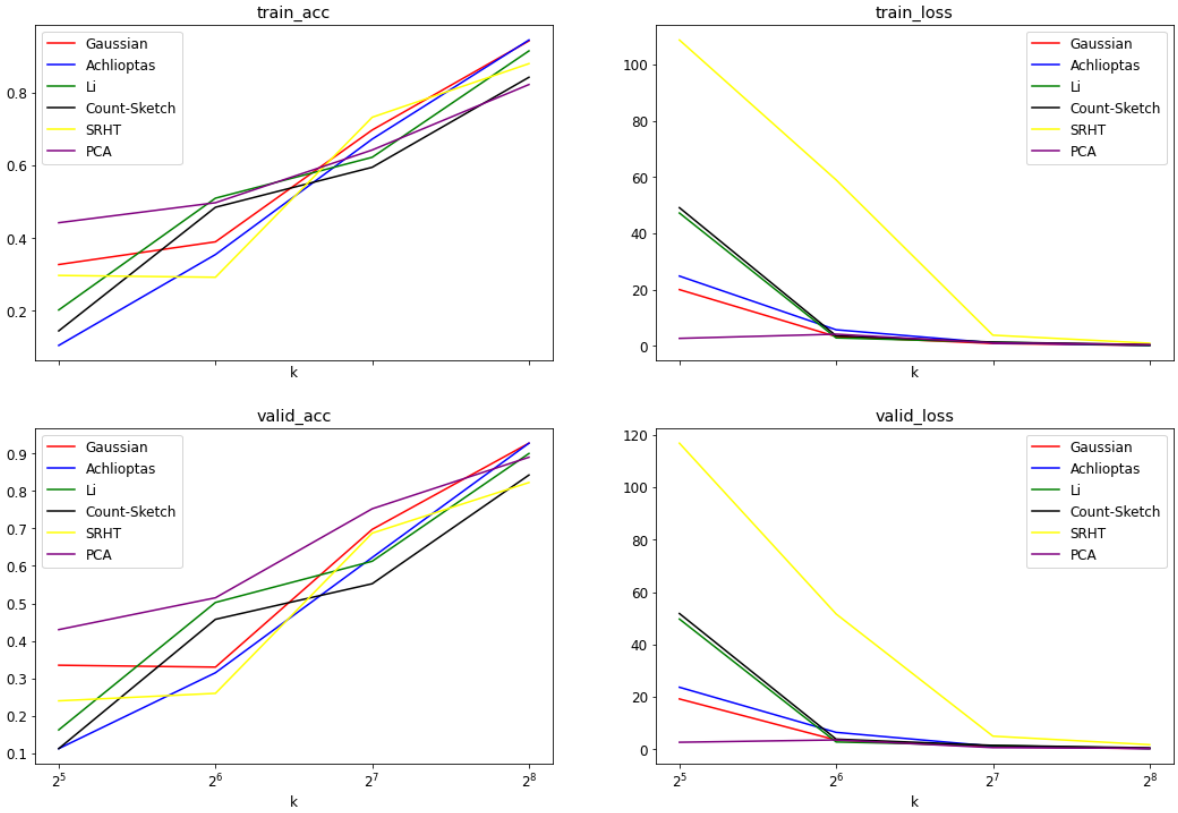

Figure 5.2: The accuracy and loss terms for both training and validation(test) sets, for projected dimension k between 2^5 and 2^8, for model with 1 hidden layer.

We see that for larger projected dimension, Gaussian and Achlioptas generally perform better than other RP models. We can also see that for smaller projected dimensions, Li and Count-Sketch perform better than most other models, SRHT/FJLT seems to perform competitvely mostly at larger projected dimensions, that could be improved by changing the density argument of the projection q. We can see that Random Projection methods are comparable for this dataset to the PCA method (PCA over the training set only) and in for certain k values Achlioptas and Gaussian are even achieving slightly better results, this can be explained by the small size of the datset which causes the PCA method to overfit the training set and not to fit percisely for all of the dataset, also since the original dimensions are very high, the RP layers achieve good results in comparison to the PCA dimensionality reduction and achieve optimal solution for 20 times dimensionality reduction (from 20531 to 1024).

Model with three hidden layer

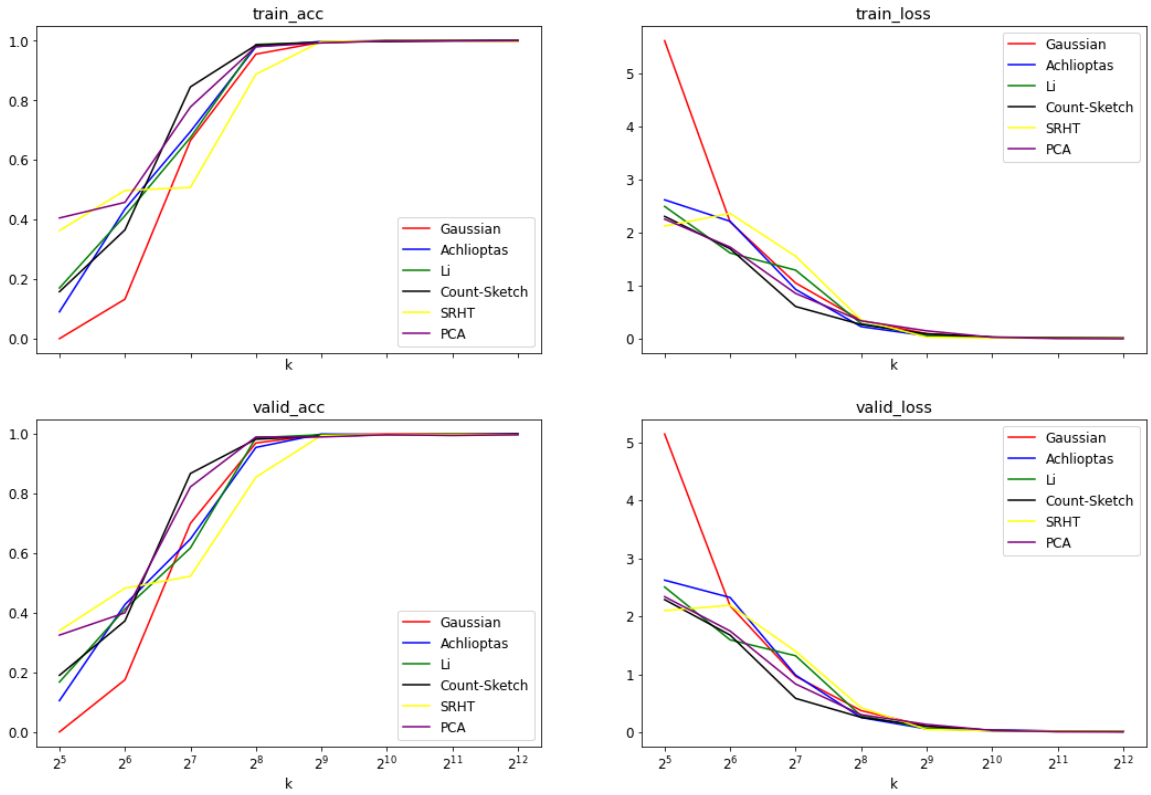

Figure 5.3: The accuracy and loss terms for both training and validation(test) sets, for projected dimension k between 2^5 and 2^{12}, for model with 3 hidden layers.

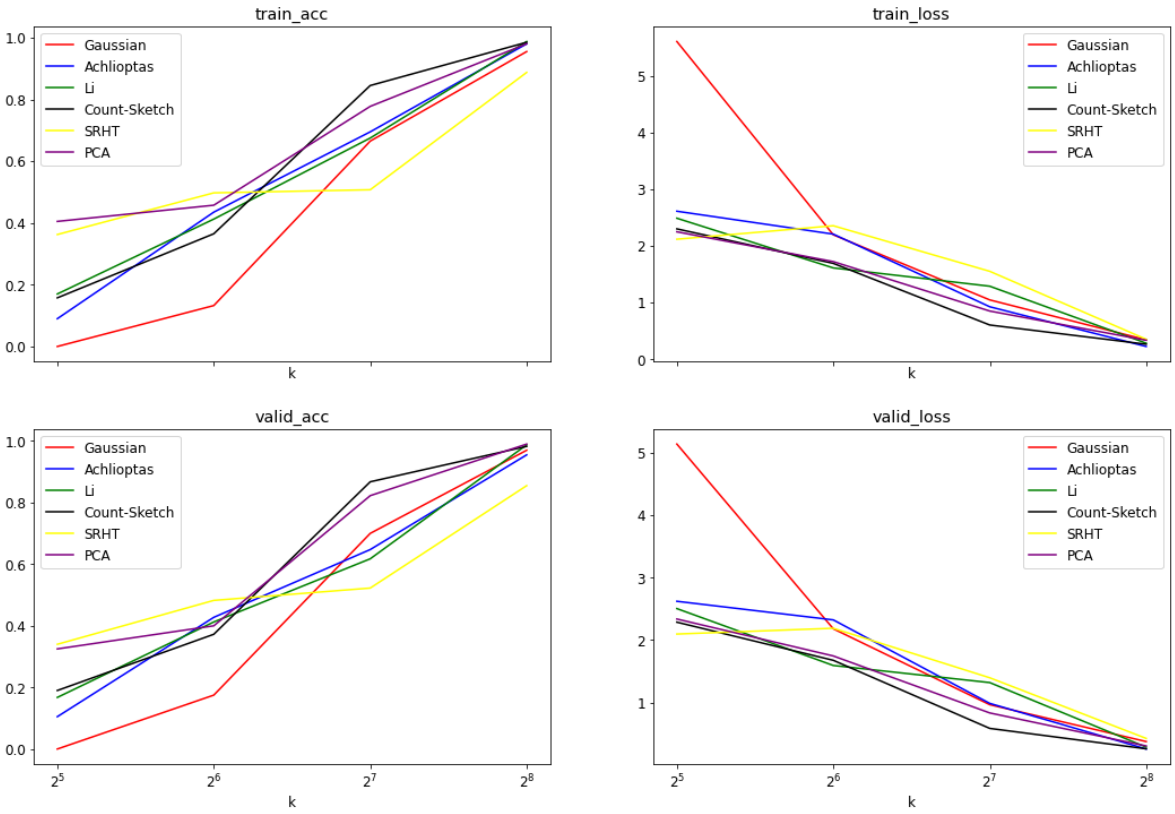

Figure 5.4: The accuracy and loss terms for both training and validation(test) sets, for projected dimension k between 2^5 and 2^8, for model with 3 hidden layers.

For the deeper model we see that for very low projected dimension k, the best RP method is SRHT. Count Sketch achieves the best results for larger projected dimension values, the deeper model achieves better results in comparison to PCA than the previous model. SRHT achieves considerably less accuracy from the other methods for $k \in \{128, 256\}$. For these k values, the Achlioptas method slightly overfits (by achieving better results on train set than on test set in compare to other methods). The deeper model outperforms the model with 1 hidden layer for $k \in \{128, 256\}$

MNIST dataset

MNIST dataset is a dataset of images of handwritten digits (0-9), each image of $28 \times 28 = 784$ pixels, we used a train and test sets of 60,000 images each.

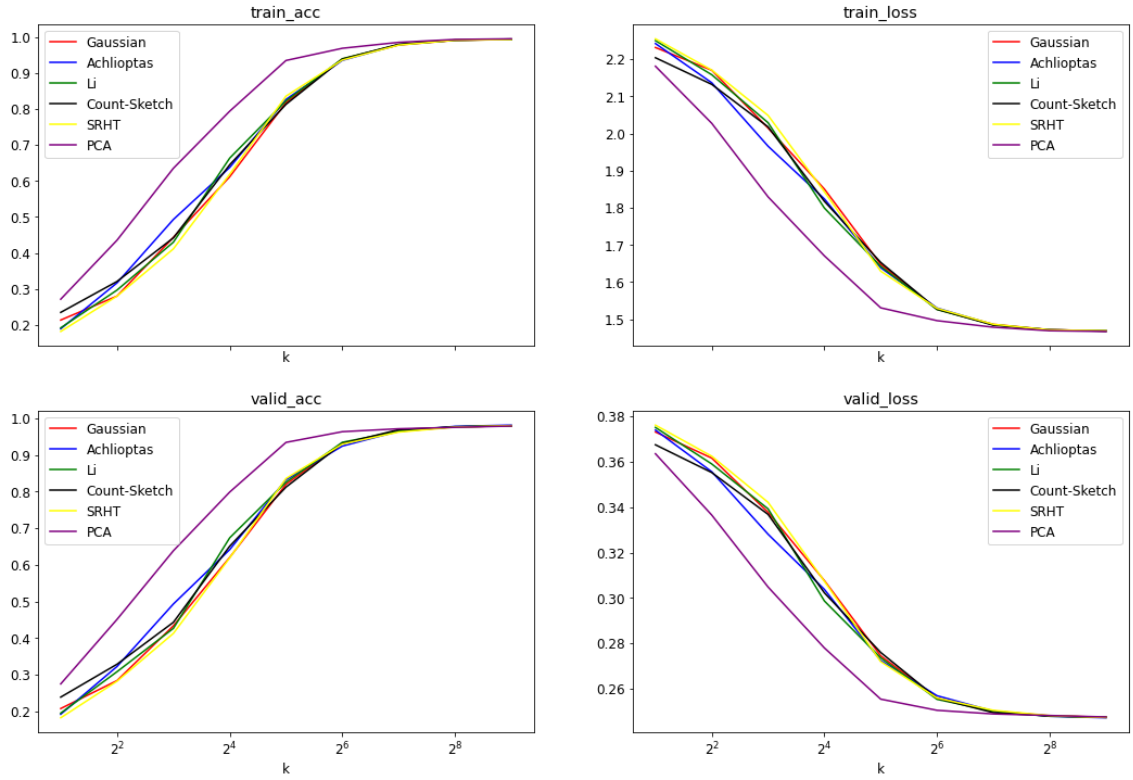

Figure 5.5: The accuracy and loss terms for both training and validation(test) sets,for projected dimension k between 2 *and* 2^9.

CIFAR-10 dataset

CIFAR-10 dataset

PCA results for smaller k values are significantly better than RP methods, for these values, Achlioptas an Li RP methods achieve good results in comparison to other RP methods, for larger k values, the gap between PCA and RP is decreasing and RP methods close this gap for $k \geq 2^7$.

For $k = 2^5$, SRHT achieves better results than other RP methods and for $k = 2^7$ Count Sketch achieves the better results than other RP methods.

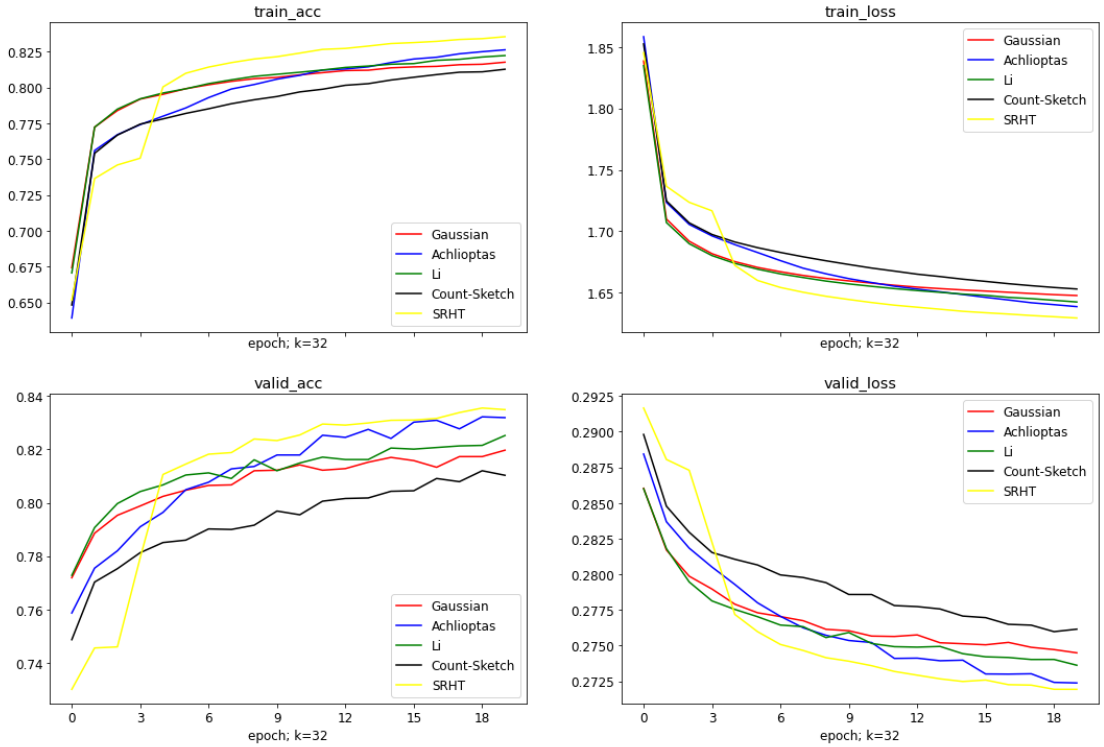

Figure 5.6: The accuracy and loss terms for both training and validation(test) sets, for projected dimension k=32 for each epoch(20 epochs.)

By checking convergance over time of the model, we can see that for the first 4 epochs Li RP achieved the best results while Li and Gaussian RP models outperformed the other models, at the 5 epoch (epoch index 4), SRHT achieves better results and from the 12th epoch, Achlioptas and SRHT outperform the other RP models.

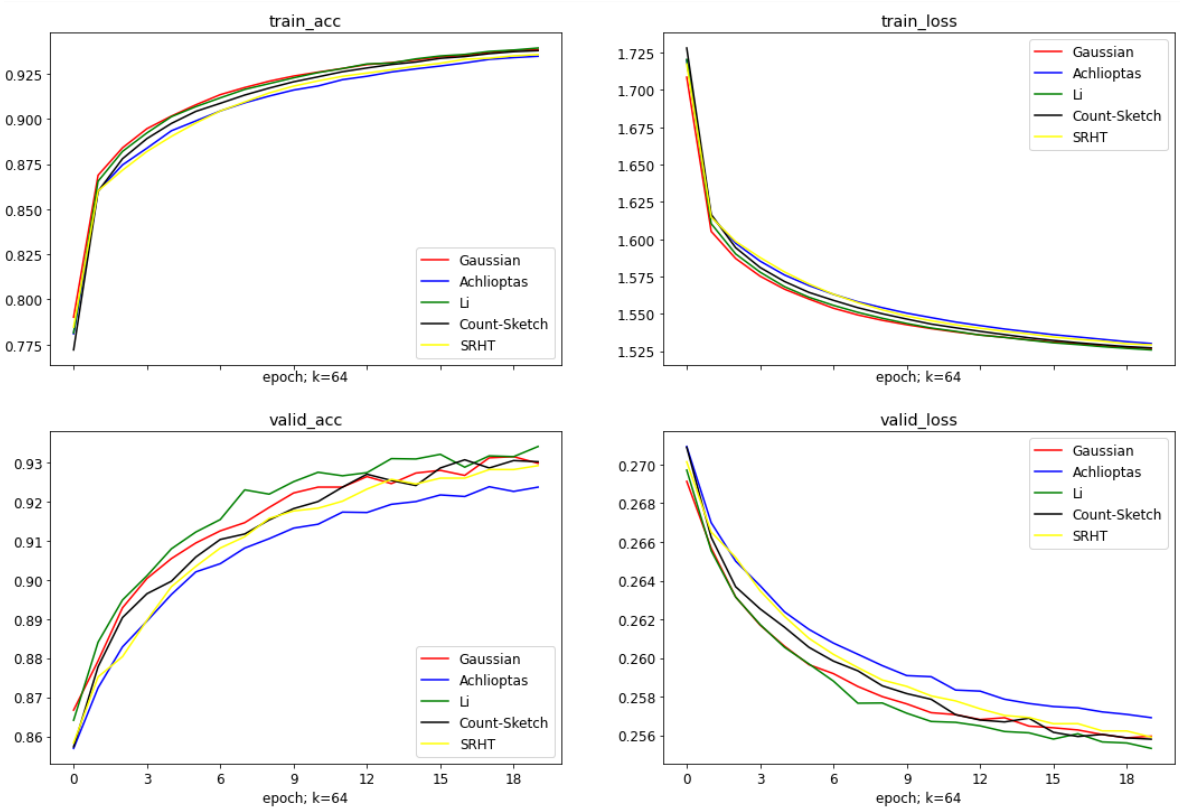

Figure 5.7: The accuracy and loss terms for both training and validation(test) sets,for projected dimension k=64 for each epoch(20 epochs.)

For k=64, the Li RP method outerperforms all other methods by a small margin for all of the training process in the validation accuracy and loss, but on training accuracy it achieves similar results to other methods.

CIFAR-10 dataset

CIFAR-10 dataset is a dataset which contains RGB images each with width and length of 32 pixels (RGB i.e 3 channels, $3 \times 32 \times 32 = 3072$ input size. Each image is classified in the dataset as one of 10 possible Classes. The dataset contains 60,000 examples of which we are using 50,000 as train set and 10,000 as test set. The model tested has 3 hidden layers similiarly to the Genetic dataset model.

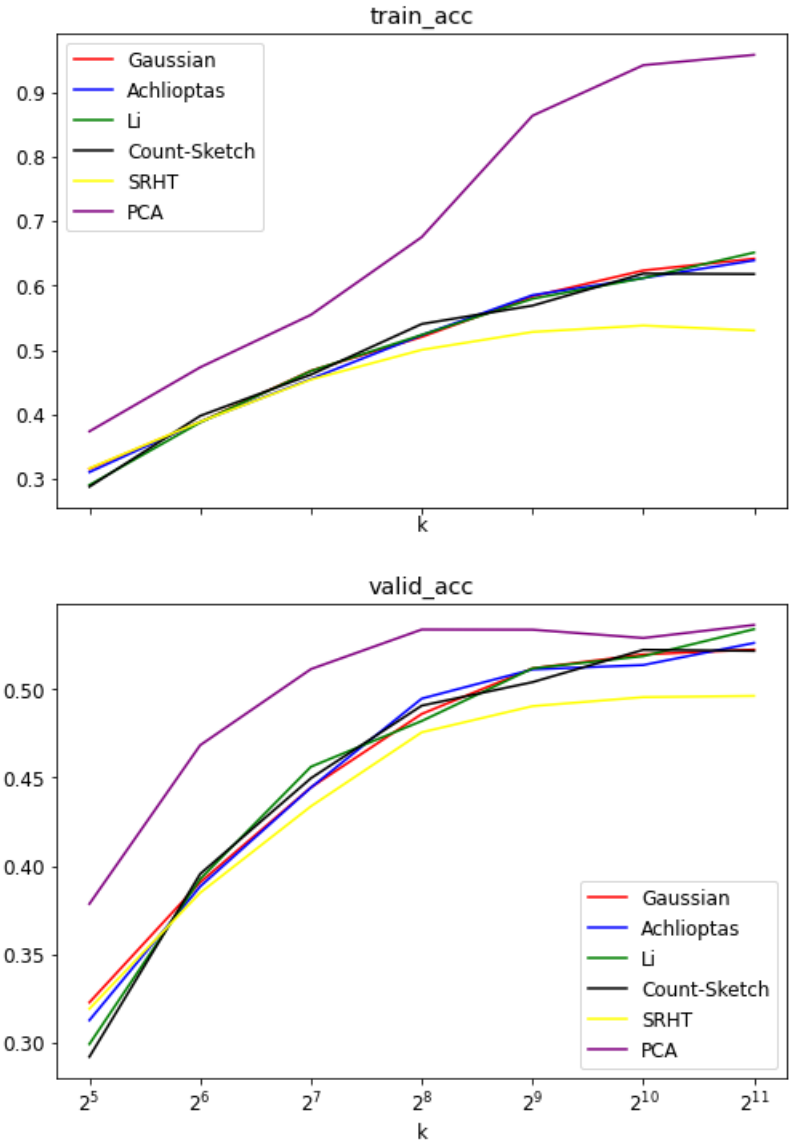

Figure 5.8: train and validation(test) accuracy for different projected dimension k values between $2^5 - 2^{11}$.

In this dataset we see that the PCA random projection achieves decent training accuracy as it preserves the data much better than RP methods, but it extermely overfits the data unlike RP methods which has much smaller gap between train and validation results, although these methods do overfit, is it possibly since RP methods act as a implicit regularization for the model. generally, for smaller projected dimensions, Gaussian and SRHT perform slightly better, but for larger projected dimensions,

Count Sketch, Li and Achlioptas achieve better results with the later two achiving the best results out of RP methods.