*

99%

The well-known issue of recent omics data is a feature-sample ratio which is highly imbalanced means the number of features is way more than the number of samples. Among all the available features, few might be meaningful for distinguishing the samples which belong to different classes and the rest of these are either irrelevant, redundant, or noise (Pirgazi et al., 2019). During classification or clustering the high dimensional data, irrelevant features cause unnecessary computational complexities and decrease the performance. Therefore, it is essential to identify the most relevant features that would have a high contribution to the classificaiton or clustering of the data. During the feature selection process, redundant features are removed because there is a subset of features that carries approximate similar information. In a similar fashion, noise features that provide no information about labels are also be removed from the database. Thus, only relevant features will be remain that will increase the efficiency of any classification or clustering problems (Liu and Motoda, 2012). Any dataset with N number of features has 2^N possible subset of features. The goal of feature selection algorithms is to find the most precise subset of features. Due to having a large number of possible combinations, finding the best subset of N features is computationally challenging and costly Liang et al. (2018).

Filter, wrapper, embedded are the types of feature selection methods. Numerous algorithms have been proposed for each type of feature selection method. In the filtering method, a rank is assigned to each feature depending on the statistical relevance to the class type. In both univariate and multivariate filter method, feature-feature interactions are not considered in the selection process. Some example studies such as Pearson correlation coefficient(PC), t-statistics(TS) (Speed, 2003), F-Test (Ding and Peng, 2005), and ANOVA (Ding and Li, 2015). These methods are effective for selecting features for high-dimensional data because of its fewer computation expenses but failed to provide a good accuracy (Sun et al., 2018). To enhance the performance, the wrapper method is proposed with a learning algorithm and a classifier to find a suitable subset of features. First, it generates a random solution, then it maximize an objective function using a black-box optimization method (Rau et al., 2019) such as Simulated Annealing (Jeong et al., 2018), Particle Swarm Optimization (Xue et al., 2012), Genetic Algorithm (Wu et al., 2011), and Ant Colony Optimization (Kabir et al., 2012). Since these methods evaluate every candidate subset of feature iteratively, they can find a strong relationship between features but it increases computational expenses. Similarly, embedded method do so efficiently as it is a part of its learning phase. Thus, it reduces the computational costs. Some well-known example studies are LASSO (Tibshirani, 1996), recursive feature ellimation with state vector machien estimator (SVM-RFE) (Abdullah, 2019; Guyon et al., 2002; Fang, 2019), random forest (Pouyan and Kostka, 2018; Ram et al., 2017), Adabost (Wang, 2012), KNN (Le et al., 2019), and autoencoder (Lu et al., 2019).

In general, feature selection methods are useful to get insight about large and complex dataset which can simplify the learning process of any machine learning algorithm. The use of feature selection is worthy when using the whole set of features is difficult to collect or costly to execute. For example, the gene expression dataset contains more than 60 thousand features with a very low number of samples. It is normal to ask: *Is it possible to identify important genes those expressions can classify available disease or cancer type?* The domain of feature selection is way more dissimilar than standard dimension reduction techniques such as principal component analysis (PCA) (Hotelling, 1933), and autoencoders (Hinton and Salakhutdinov, 2006). They can preserve maximum variance with a fewer number of features, however, these methods do not provide the original features of the dataset. Thus, it is impossible to eliminate redundant or irrelevant features from the dataset.

○

In this paper, a novel feature subset selection method that increases the power deep autoencoder for differentiable feature selection is proposed. Our method CoRAE introduces a new layer in the autoencoder called concrete distribution of features which allows the model to select a user-defined number of original features. Idea of concrete distribution is adapted from (Maddison et al., 2016; Kingma and Welling, 2013), and reparameterization technique to minimize the loss and reconstruction error from (Abid et al., 2019). We have tested our end-to-end model on coding and non-coding gene expression dataset and it outperforms state-of-the-art feature selection techniques.

To validated the proposed idea, TCGA RNAseq cancer (n=9566) and clinical samples for 33 cancers were downloaded from UCSC Xena database (https://xenabrowser.net). TCGA processed raw RNAseq data using Illumina HiSeq 2000 RNA sequencing platform where per-gene normalized abundance estimation were calculated with FPKM method. RNASeq normalized counts were then log transformed after adding a constant of 1. Later UCSC re-processed using GENCODE v23 transcript annotation to quantify protein coding() and non-coding transcripts() expression (Harrow et al., 2006). Coding genes refers to mRNA whereas non-coding genes refers to long non-coding RNA (lncRNA) in this experiment. To improve the focus on individual feature selection, we separated mRNA and lncRNA expression from combined database using TANRIC (Li et al., 2015) provided standard list of lncRNAs. Another important reason of performing experiment on individual RNA types is because their expression level is different. The number of mRNA and lncRNA are 18731, 12309 respectively. We merged all the cancer samples for individual RNA types for further experiment. Each row is mapped to a unique Ensemble ID, and each column mapped to a patient ID. Normal patients or RNA with missing data were removed from the original dataset. Each RNA expression was further processed using min-max normalization method to achieve good training performance.

The concrete relaxation autoencoder CoRAE is a variation of original autoencoder AE (Hinton and Salakhutdinov, 2006) for dimension reduction. It is a neural network consists of two parts: an encoder that selects latent features and a decoder that uses selected features to reconstruct the output similar to the input. Instead of using a sequence of fully connected layers in the encoder, we propose a concrete relaxation based feature selection layer where user can define the number of nodes (feature), k. This layer selects probabilistic linear arrangement of input features during training, which converge to a discrete set of k features by the end of training and during the testing.

The original features are selected based on the temperature of this layer which is tunned using an annealing schedule. More specifically, the concrete selector layer identifies k number of important features as the temperature decreases to zero. For reconstructing the input, a simple decoder similar to the standard AE is used. This simple neural network can be updated based on the characteristics of the data and its complexity.

Layer that selects the features shown in Figure 1 is called concrete variable selctor layer adpoted from concrete distribution (Maddison et al., 2016) and categorical represntation (Jang et al., 2016). Since, backpropagation does not allow computation of the parameters' gradient through stochastic nodes of standard autoencoder, gumbel $softmax$ distribution g (Gumbel, 1954) is a right choice to pick samples z from categorical distribution with class probabilities α_k.

$$z = \textbf{one-hot}\left(\underset{k}{\operatorname{argmax}}\left[g_k + log\,\alpha_k\right]\right) \tag{1}$$

Because argmax is not differentiable, simple $softmax$ function can be used as a continuous approximation of argmax. The aim of using Concrete random variables is to relax the state of a discrete variable and the ralaxation degree is controlled by a temparature parameter $\tau \in (0, \infty)$. To sample a concrete random variable in z dimensions with prarameter $\alpha \in \mathbb{R}^z > 0$ and τ, one must samples a z-dimensional vector of $i.i.d.$ (independent and identically distributed) samples from a Gumbel distribution, g. Then each element of the sample f from the Concrete distribution can be defined as:

$$f_k = \frac{exp((log\alpha_k + g_k)/\tau)}{\sum_{i=1}^{z} exp((log\alpha_i + g_i)/T)}\ for\ k = 1, ..., z \tag{2}$$

where f_k refers to the k_{th} element in a particular sample vector. With the limit $\tau \rightarrow 0$, the concrete variable uniformly progresses the discrete distribution, producing one-hot vector with $f_k = 1$ with a probabilistic chance of $\alpha_k / \sum_p \alpha_p$. The advantage of using a concrete random discrete variable is that it is differentiable $w.r.t\ \alpha$ using reparameterization technique as mentioned by (Kingma and Welling, 2013).

More concisely, the way original feature is selected using the concrete random variable as follows: a z-dimensional concrete random variable $f^{(i)}$ is sampled for each node of the selector layer with k nodes where i refers to the index of the node, $i \in \{1...k\}$. The output of the i^{th} node is $\mathbf{x}.f^{(i)}$. Although it is a combination of the input feature's weight, every node of the selector layer produces exactly one of the original input features in the limit $\tau \rightarrow 0$. After training the network, a discrete argmax layer is replaced with the concrete selector layer by which $x_{\operatorname{argmax}_j\ \alpha_j^{(i)}}$ is produced as an output of i^{th} node during the testing phase. The value of α_i initially starts with a small positive random number so that it can explore various combinations of input features. As the model is being trained, the value of α_i, in other words the probability of class i becomes more stable. As a result, the model reduces its stochasticity rather increases the confidence in drawing a particular subset of features.

The temperature of the random variable in the selector layer has a significant impact in forming the output of each node. Initially, when τ is high, search space is large since it consideres linear combination of all features. In contrast, the selector layer will not be able to search all possible combinations of features in low τ and thus, model converges to a bad local minima. Instead of using a fixed temperature, a simple annealing scheduling scheme is used for every concrete variable. It starts with an user-defined high temperature (τ_s) and steadily lessening the temperature until it touches the ending bound (τ_e) by every epoch as follows:

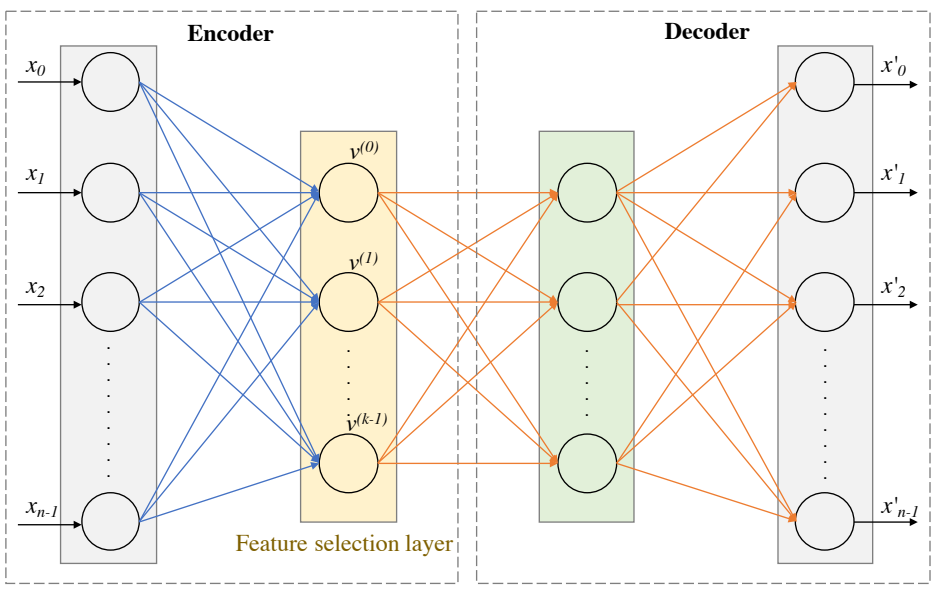
$$\tau(e) = \tau_s(\tau_N/\tau_s)^{e/N} \tag{3}$$

where $T_{(e)}$ is the temperature at epoch e, N refers to the total number of epochs. The proposed annealing schedule is good enough to explore the feature combinations during the training phase and finally lowered temperature enables the model to strict to the best set of features which is shown in Figure 3.

The encoder of the Concrete relaxation autoencoder (CoRAE) architecture is constructed with a hidden layer of k nodes where k being the number of gene selected. The decoder, on the other hand, is consisting of one hidden layer with 3k/2 nodes. The number of nodes in this layer is tunned in a range of [4k/7, 2k/5, 3k/2]. Adam optimizer with a learning

Table 1. Sample Distribution for 33 cancers along with 75-25 split for training and testing.

Sl	Cancer site name	Acronym	#Sample	#Train	#Test	Sl	Cancer site name	Acronym	#Sample	#Train	#Test
1	Adrenocortical Cancer	ACC	77	57	20	18	Lung Squamous Cell Carcinoma	LUSC	498	373	125
2	Bladder Cancer	BLCA	407	305	102	19	Mesothelioma	MESO	86	64	22
3	Breast Cancer	BRCA	1089	816	273	20	Ovarian Cancer	OV	375	281	94
4	Cervical Cancer	CESC	304	228	76	21	Pancreatic Cancer	PAAD	177	132	45
5	Bile Duct Cancer	CHOL	36	27	9	22	Pheochromocytoma & Paraganglioma	PCPG	177	132	45
6	Colon Cancer	COAD	301	225	76	23	Prostate Cancer	PRAD	493	369	124
7	Large B-cell Lymphoma	DLBC	47	35	12	24	Rectal Cancer	READ	95	71	24
8	Esophageal Cancer	ESCA	161	120	41	25	Sarcoma	SARC	258	193	65
9	Glioblastoma	GBM	158	118	40	26	Melanoma	SKCM	465	348	117
10	Head and Neck Cancer	HNSC	499	374	125	27	Stomach Cancer	STAD	378	283	95
11	Kidney Chromophobe	KICH	66	49	17	28	Testicular Cancer	TGCT	132	99	33
12	Kidney Clear Cell Carcinoma	KIRC	527	395	132	29	Thyroid Cancer	THCA	501	375	126
13	Kidney Papillary Cell Carcinoma	KIRP	287	215	72	30	Thymoma	THYM	118	88	30
14	Acute Myeloid Leukemia	LAML	147	110	37	31	Endometrioid Cancer	UCEC	184	138	46
15	Lower Grade Glioma	LGG	507	380	127	32	Uterine Carcinosarcoma	UCS	56	42	14
16	Liver Cancer	LIHC	369	276	93	33	Ocular melanomas	UVM	79	59	20
17	Lung Adenocarcinoma	LUAD	512	384	128		Total		9566	7161	2405



Architecture of Concrete Relaxation Autoencoder. Proposed feature selection architecture consists of an encoder and a decoder. The layer after input layer in encoder is called concrete feature selection layer shown in yellow. This layer has k number of node where each node is for each feature to be selected. During the training stage, the i^{th} node $v^{(i)}$ takes the value $x^T f^{(i)}$. During testing stage, these weights are fixed and the element with the highest value is selected by the corresponding i^{th} hidde n node. The architecture of the decoder remains the same during train and test stage.

rate of 10^-3 is used for all the experiments. The starting temperature of the CoRAE was set to 10 and it ends at 0.01. To avoid overfitting, the dataset is split into the train and test set according to $75/25$ ratio. The training set is used to estimates the learning parameters and the test set is used for performance evaluation. To control the performance, the model is trained for the same number of epoch 100. Performance of CoRAE has been compared with state-of-the-art feature selection techniques such as LASSO and SVM-RFE on both mRNA and lncRNA expression datasets. In LASSO, a regularization parameter α decides the number of most important features. More precisely, the higher the α, the more feature's coefficient shrinks to zero, fewer features would be selected. Recursive feature elimination is a recursive method in which less important features are eliminated in every iteration. In the recursive feature elimination technique, SVM is used as an estimator. Linear kernel with a regularization parameter $C = 0.05$ is used. C controls the tradeoff between the error and norm of the learning weights. GridSearch algorithm is used to estimate the best set of parameters for SVM. In every iteration of RFE, the number of dropped features is set to 100.

We extract a subset of features by varying k from 10 to 500. Towards fair comparison with CoRAE, the same number of genes has been selected by LASSO and SVM-RFE. Then dataset with reduced number of features (expression of selected genes) to the SVM for classifying 33 cancer types on both mRNA and lncRNA expression. Similarly, to reconstruct all the input features, we trained a linear regressor with no regularization and measure the reconstruction mean square error. LASSO and SVM-RFE are developed using scikit learn framwork (Pedregosa et al., 2011) whereas CoRAE is build using Google developed Tensorflow (Abadi et al., 2015) based deep learning framwork Keras (Chollet et al., 2015). Experiments

are parallelized on NVIDIA Quadro K620 GPU with 384 cores and 2GB memory devices. Five different evaluation metrics have been used to record the classification and reconstruction performance such as accuracy, precision, recall, f1 score, and mean squared error (MSE).

Accuracy is the number of correct predictions made by the model over all kinds of predictions made. True positives(TP) and True Negatives(TN) are the correct prediction.

$$Accuracy = \frac{TP + TN}{TP + TN + FP + FN} \tag{4}$$

Precision is the number of correct positive results divided by the number of positive results predicted by the classifier. It indicates the predicted positive portion of the samples.

$$Precision = \frac{TP}{TP + FP} \tag{5}$$

Recall is the number of correct positive results divided by the number of all relevant samples.

$$Recall = \frac{TP}{TP + FN} \tag{6}$$

F1 score is a measure of a test's accuracy. It considers both the precision and the recall of the test to compute the score.

$$F1 = 2 \times \frac{Precision \times Recall}{Precision + Recall} \tag{7}$$

Mean squared error MSE is the avarage of $(\frac{1}{n}\sum_{i=1}^{n})$ of the square of the errors $(Y_i - Y_i')$ where Y_i is a true label and Y_i' is a predicted label. All performance matrices are measured on the predicted labels and true labels of independent test samples.

Top genes can be selected based on two criteria - a) classification accuracy needs to be higher, and b) the number of genes should be as less as possible so that biologists can conduct a wet lab experiment easily. The capabilities of selected genes in pan-cancer classification is visually validated using unsupervised visualization technique t-SNE (Maaten and Hinton, 2008).

A series of experiements is conducted to compare the performance of CoRAE with other state-of-the-art feature selection methods such as LASSO and SVM-RFE. A range of features from 10 to 500 has been selected using all three methods, then train a linear classifier (SVM) using selected coding and non-coding gene expression of 33 cancer patients. Figure **??** shows the classification performance for different number of features. Across the all k, CoRAE has highest accuracy and lowest error for both mRNA and lncRNA expression. Even if the number of feature is low e.g. 10, the accuracy is almost 80% whereas LASSO and SVM-RFE shows poor restuls for lowest number of feature. For more than 50 features, CoRAE shows more than 90% accuracy. Also, it shows less error with less number of features compare to other methods. For mRNA, CoRAE starts with MSE of 38 and quickly recduced to less than 10 within top 100 features. The behaviour in classification is almost similar in both coding and non-coding genes. However, mRNA expression performs slightly better than lncRNA which as shown in Figure **??**.

CoRAE not only able to identify important features but also allows the user to examine relevance by observing the estimated concrete parameter $\alpha^{(i)}$ for each feature. Since CoRAE selects a feature based on the value of vector $\alpha^{(i)}$, the user can check the importance of each feature and find the correlation with others. In Figure **??**, it is visually revealed that the top 100 mRNA or lncRNA is capable of distinguishing 33 cancer types. Also, it is noticeable that the feature selected by CoRAE is carrying more information than among all other features. Thus, influential features are selected in proposed method.

In this research, a new differentiable feature selection method via backpropagation is proposed. In brief, the concrete relaxation autoencoder used reparameterization and concrete random variable technique to allow gradients to pass through a layer that stochastically selects discrete original input features. This randomness of the proposed method enables it to effectively search and converge to a user-defined number of original features which maximizing the objective function and minimizing the loss as discussed in section 2.2. The estimated parameters learned by the models can be further examined by the biologists to interpret biological relevance as discussed in section 3.1. This made CoRAE distinctive from numerous competing approaches based on regularization.

It is shown via several experiments on publicly available gene expression cancer datasets that CoRAE efficiently maximizes the classification accuracy and minimizes the reconstruction error using a selected subset of genes. For both datasets mRNA and lncRNA gene expression, CoRAE outperformed several sophisticated feature selection techniques. This phenomena still remain and minimizes the reconstruction error when only a single hidden layer is used in the decoder. It indicates the power of CoRAE in selecting features from a large dataset.

Since CoRAE is built on standard autoencoder architecture, it is easily scalable to the higher number of samples or dimensions as discussed in section **??** where the features selected by the CoRAE outperformed the competing methods. Moreover, as CoRAE proposed in its generic form, it can be surely prolonged in several fashions. For example, unlike multiple cancer classification, important genes can be extracted during the molecular subtype classification of a single cancer dataset. Also, It allows users to integrate multi-omics data such as gene, protein, RNAseq expression, DNA methylation, copy number and so on. CoRAE is easy to use and it requires only a few lines of modification in implementing it in the popular machine learning algorithm. Moreover, the runtime and space complexity is similar to that of the standard autoencoder. In addition, it enhances parallelization and hardware acceleration which is an obvious demand for deep learning. Starting and ending temperature are the only added hyperparameters used for annealing schedule. The default value used in the experiment is found well enough for the various datasets.

future work: we will conduct more biological validation in our extended work such as survival analysis of 33 cancer patients using selected features to measure the prognostic capabilities. Similarly, pathway analysis of selected coding and non-coding genes will be analyzed in future work as well.

This research is partially funded by NSF CAREER award #1651917 (transferred to #1901628) to AMM.

Martín Abadi, Ashish Agarwal, Paul Barham, Eugene Brevdo, Zhifeng Chen, Craig Citro, Greg S. Corrado, Andy Davis, Jeffrey Dean, Matthieu Devin, Sanjay Ghemawat, Ian Goodfellow, Andrew Harp, Geoffrey Irving, Michael Isard, Yangqing Jia, Rafal Jozefowicz, Lukasz Kaiser, Manjunath Kudlur, Josh Levenberg, Dandelion Mané, Rajat Monga, Sherry Moore, Derek Murray, Chris Olah, Mike Schuster, Jonathon Shlens, Benoit Steiner, Ilya Sutskever, Kunal Talwar, Paul Tucker, Vincent Vanhoucke, Vijay Vasudevan, Fernanda Viégas, Oriol Vinyals, Pete Warden, Martin Wattenberg, Martin Wicke, Yuan Yu, and Xiaoqiang Zheng. TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. URL . Software available from tensorflow.org.

Ananda Mondal Abdullah, Al Mamun. Feature selection and classification reveal key lncrnas for multiple cancers. 2019.

Abubakar Abid, Muhammad Fatih Balin, and James Zou. Concrete autoencoders for differentiable feature selection and reconstruction. *arXiv preprint arXiv:1901.09346*, 2019.

François Chollet et al. Keras. , 2015.

Chris Ding and Hanchuan Peng. Minimum redundancy feature selection from microarray gene expression data. *Journal of bioinformatics and computational biology*, 3(02):185–205, 2005.

Hui Ding and Dongmei Li. Identification of mitochondrial proteins of malaria parasite using analysis of variance. *Amino acids*, 47(2):329–333, 2015.

Jianwen Fang. Tightly integrated genomic and epigenomic data mining using tensor decomposition. *Bioinformatics*, 35(1):112–118, 2019.

Emil Julius Gumbel. Statistical theory of extreme values and some practical applications. *NBS Applied Mathematics Series*, 33, 1954.

Isabelle Guyon, Jason Weston, Stephen Barnhill, and Vladimir Vapnik. Gene selection for cancer classification using support vector machines. *Machine learning*, 46(1-3):389–422, 2002.

Jennifer Harrow, France Denoeud, Adam Frankish, Alexandre Reymond, Chao-Kung Chen, Jacqueline Chrast, Julien Lagarde, James GR Gilbert, Roy Storey, David Swarbreck, et al. Gencode: producing a reference annotation for encode. *Genome biology*, 7(1):S4, 2006.

Geoffrey E Hinton and Ruslan R Salakhutdinov. Reducing the dimensionality of data with neural networks. *science*, 313(5786):504–507, 2006.

Harold Hotelling. Analysis of a complex of statistical variables into principal components. *Journal of educational psychology*, 24(6):417, 1933.

Eric Jang, Shixiang Gu, and Ben Poole. Categorical reparameterization with gumbel-softmax. *arXiv preprint arXiv:1611.01144*, 2016.

In-Seon Jeong, Hong-Ki Kim, Tae-Hee Kim, Dong Hwi Lee, Kuinam J Kim, and Seung-Ho Kang. A feature selection approach based on simulated annealing for detecting various denial of service attacks. *Software Networking*, 2018(1): 173–190, 2018.

Md Monirul Kabir, Md Shahjahan, and Kazuyuki Murase. A new hybrid ant colony optimization algorithm for feature selection. *Expert Systems with Applications*, 39(3):3747–3763, 2012.

Diederik P Kingma and Max Welling. Auto-encoding variational bayes. *arXiv preprint arXiv:1312.6114*, 2013.

Trang T Le, Ryan J Urbanowicz, Jason H Moore, and Brett A McKinney. Statistical inference relief (stir) feature selection. *Bioinformatics*, 35(8):1358–1365, 2019.

Jun Li, Leng Han, Paul Roebuck, Lixia Diao, Lingxiang Liu, Yuan Yuan, John N Weinstein, and Han Liang. Tanric: an interactive open platform to explore the function of lncrnas in cancer. *Cancer research*, 75(18):3728–3737, 2015.

Sen Liang, Anjun Ma, Sen Yang, Yan Wang, and Qin Ma. A review of matched-pairs feature selection methods for gene expression data analysis. *Computational and structural biotechnology journal*, 16:88–97, 2018.

Huan Liu and Hiroshi Motoda. *Feature selection for knowledge discovery and data mining*, volume 454. Springer Science & Business Media, 2012.

Xiaolu Lu, Hong Gu, Yang Wang, Jia Wang, and Pan Qin. Autoencoder based feature selection method for classification of anticancer drug response. *Frontiers in genetics*, 10:233, 2019.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of machine learning research*, 9(Nov):2579–2605, 2008.

Chris J Maddison, Andriy Mnih, and Yee Whye Teh. The concrete distribution: A continuous relaxation of discrete random variables. *arXiv preprint arXiv:1611.00712*, 2016.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830, 2011.

Jamshid Pirgazi, Mohsen Alimoradi, Tahereh Esmaeili Abharian, and Mohammad Hossein Olyaee. An efficient hybrid filter-wrapper metaheuristic-based gene selection method for high dimensional datasets. *Scientific Reports*, 9 (1):1–15, 2019.

Maziyar Baran Pouyan and Dennis Kostka. Random forest based similarity learning for single cell rna sequencing data. *Bioinformatics*, 34(13):i79–i88, 2018.

Malihe Ram, Ali Najafi, and Mohammad Taghi Shakeri. Classification and biomarker genes selection for cancer gene expression data using random forest. *Iranian journal of pathology*, 12(4):339, 2017.

Andrea Rau, Michael Flister, Hallgeir Rui, and Paul L Auer. Exploring drivers of gene expression in the cancer genome atlas. *Bioinformatics*, 35(1):62–68, 2019.

Terry Speed. *Statistical analysis of gene expression microarray data*. Chapman and Hall/CRC, 2003.

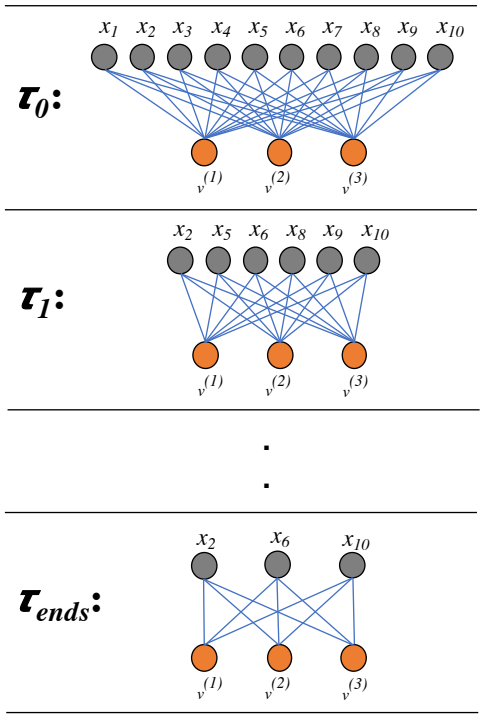
Yingqiang Sun, Chengbo Lu, and Xiaobo Li. The cross-entropy based multi-filter ensemble method for gene selection. *Genes*, 9(5):258, 2018.

Robert Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

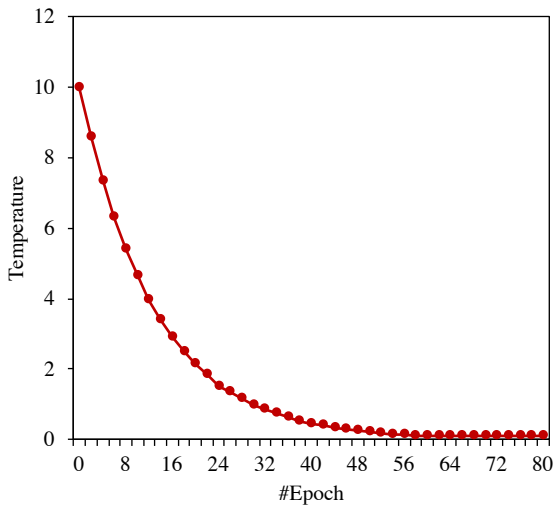
Ruihu Wang. Adaboost for feature selection, classification and its relation with svm, a review. *Physics Procedia*, 25:800–807, 2012.

Yi-Leh Wu, Cheng-Yuan Tang, Maw-Kae Hor, and Pei-Fen Wu. Feature selection using genetic algorithm and cluster validation. *Expert Systems with Applications*, 38(3):2727–2732, 2011.

Bing Xue, Mengjie Zhang, and Will N Browne. Particle swarm optimization for feature selection in classification: A multi-objective approach. *IEEE transactions on cybernetics*, 43(6):1656–1671, 2012.



Temperature Effect.



Annealing schedules for the CoRAE. Effect of different annealing schedules on a concrete autoencoder trained on the mRNA dataset with $k = 100$ selected features. If the temperature is exponentially decayed (the annealing schedule), the feature selected layer (model) converges to informative features.