Review Series of Recent Deep Learning Papers:

Parameter Prediction Paper: Diet Networks: Thin Parameters for Fat Genomics

Adriana Romero,Pierre Luc Carrier,Akram Erraqabi,Tristan Sylvain, Alex Auvolat, Etienne Dejoie, Marc-André Legault, Marie-Pierre Dube, Julie G. Hussin, Yoshua Bengio

ICLR 2017

Reviewed by : Arshdeep Sekhon

[1]Department of Computer Science, University of Virginia
https://qdata.github.io/deep2Read/

August 25, 2018

'FAT' genomic data

1. the number of input features orders of magnitude larger than the number of training examples
2. For example, genetic variation data: miilions of SNPs
3. When these features are used as input, number of free parameters increases.
4. Diet Networks: Reparametrize the network to reduce the number of free parameters

Diet Networks

1. TASK: predicting genetic ancestry from SNP data

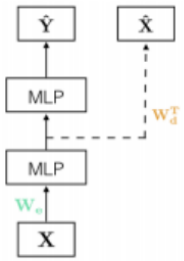
Diet Networks

1. TASK: predicting genetic ancestry from SNP data
2. Diet Networks:
   1. Parameter Prediction Network
   2. Primary Network

Diet Networks

1. TASK: predicting genetic ancestry from SNP data
2. Diet Networks:
    1. Parameter Prediction Network
        1. Input: Single feature
        2. Output: Parameters for Primary Network for that feature
        3. Shared between all features
    2. Primary Network
        1. Classifier Network

Diet Network



1. $\boldsymbol{X} \in R^{N \times N^d}$ ($\mathbb{N}^d$dimensional N examples)
2.

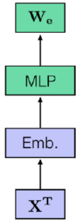
$$\boldsymbol{h}_i = f(\boldsymbol{x}_i) \tag{1}$$
$$y_i = g(\boldsymbol{h}_i) \tag{2}$$
$$\hat{\boldsymbol{x}}_i = r(\boldsymbol{h}_i) \tag{3}$$

Diet Nets: 'Fat' layer parameters

1. $$\boldsymbol{h}_i = f(\boldsymbol{x}_i) \tag{4}$$
   $$y_i = g(\boldsymbol{h}_i) \tag{5}$$
   $$\hat{\boldsymbol{x}}_i = r(\boldsymbol{h}_i) \tag{6}$$
2. $$\boldsymbol{h}_1 = f_1(\boldsymbol{W}_e \boldsymbol{x}_i) \tag{7}$$
3. first hidden layer: $\mathbb{N}^h$, $\boldsymbol{W}_e \in \mathbb{N}^d \times \mathbb{N}^h$
4. if input has 300K SNPs, then the number of parameters: 30M

Parameter Prediction Network

1.
$$\boldsymbol{h}_1 = f_1(\boldsymbol{W}_e \boldsymbol{x}_i) \tag{8}$$

2. 

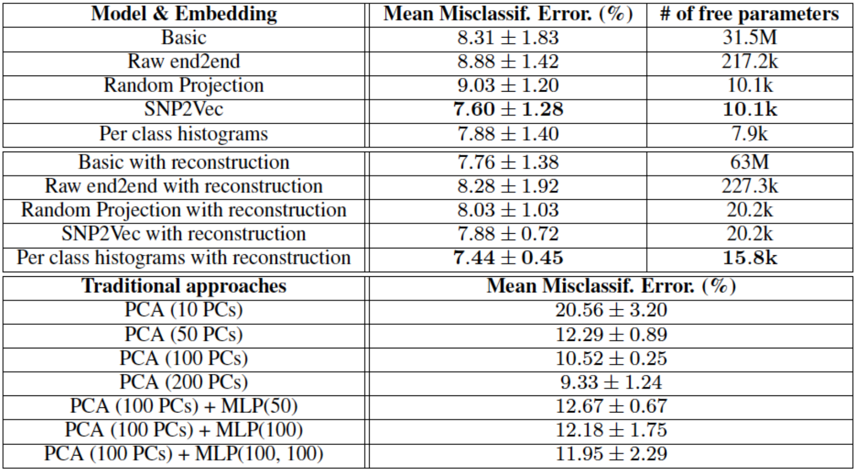
Parameter Prediction Network

3. $\boldsymbol{e}_j = Embedding(\boldsymbol{X}_{j:}^T)$
4. $(\boldsymbol{W}_e)_{j:} = \phi(\boldsymbol{e}_j)$

Feature Embeddings

1. Any kind of embedding that does not lead to an increase in the number of parameters
2. For example, for ancestry prediction task,
   1. Random Projection
   2. embedding as another MLP: end to end learning
   3. SNP per class histogram
   4. Denoising Autoencoder trained on X: learns to recover the values of missing SNPs by using their similarities and cooccurences with other SNPs

Results



Model & Embedding	Mean Misclassif. Error. (%)	# of free parameters
Basic	8.31 ± 1.83	31.5M
Raw end2end	8.88 ± 1.42	217.2k
Random Projection	9.03 ± 1.20	10.1k
SNP2Vec	$\mathbf{7.60 \pm 1.28}$	**10.1k**
Per class histograms	7.88 ± 1.40	7.9k
Basic with reconstruction	7.76 ± 1.38	63M
Raw end2end with reconstruction	8.28 ± 1.92	227.3k
Random Projection with reconstruction	8.03 ± 1.03	20.2k
SNP2Vec with reconstruction	7.88 ± 0.72	20.2k
Per class histograms with reconstruction	$\mathbf{7.44 \pm 0.45}$	**15.8k**

Traditional approaches	Mean Misclassif. Error. (%)
PCA (10 PCs)	20.56 ± 3.20
PCA (50 PCs)	12.29 ± 0.89
PCA (100 PCs)	10.52 ± 0.25
PCA (200 PCs)	9.33 ± 1.24
PCA (100 PCs) + MLP(50)	12.67 ± 0.67
PCA (100 PCs) + MLP(100)	12.18 ± 1.75
PCA (100 PCs) + MLP(100, 100)	11.95 ± 2.29