EXISTENCE OF SPARSE BASIS FOR DEEP LEARNING KERNELS ?

Aditya Golatkar and Rudrajit Das
Department of Electrical Engineering,
Indian Institute of Technology Bombay

Problem Introduction & Motivation

- PCA on traditional kernels (KPCA) results in an excellent sparse basis. However, the eigenvectors of the covariance matrix in the kernel space are linear combinations of all the training points!
- In sparse KPCA (SKPCA), we obtain approximate eigenvectors which are linear combinations of only a few of the training points - 2 fold sparsity! SKPCA on the RBF kernel performs very well on real world data.
- Makes us suspect that the RBF kernel can generalize well using only a few training points.
- Modern deep learning (DL) models are trained with a huge number of examples but can the learnt kernels also generalize well using only a few training points? We try to investigate this by performing SKPCA on DL kernels.

SKPCA Objective Function & Algorithm

$$J = \sum_{k=1}^{n} \|\phi(x_k) - (\Phi\alpha_m)\langle(\Phi\beta_m), \phi(x_k)\rangle\|^2 + \lambda\|\beta_m\|^2 + \sum_{j=1}^{m} \lambda_j|\beta_m^j|_1 \text{ subject to } \alpha_m^T K\alpha_m = I_m,$$

$$\Phi = [\phi(x_1), ..., \phi(x_n)]$$

$$J = tr(K) - 2tr({\alpha_m}^T K^2\beta_m) + tr({\beta_m}^T(K^2 + \lambda I)\beta_m) + \sum_{j=1}^{m} \lambda_j|\beta_m^j|_1 \text{ subject to } \alpha_m^T K\alpha_m = I_m$$

SKPCA ALGORITHM:

1. Initialize α_m with the m eigen vectors of K corresponding to the m largest eigen values.
2. First for fixed α_m, solve the elastic net problem in (11) for β_m.
3. Then with the β_m obtained in the previous step, update α_m as $\alpha_m = U\Sigma^{-1/2}U^*V^{*T}$ where $\Sigma^{-1/2}U^T K^2\beta_m = U^*\Sigma^*V^{*T}$ and $K = U\Sigma U^T$.
4. Repeat steps 2-3 till convergence.

Synthetic datasets

1. $x = [x_1, x_2, ..., x_{10}]$ − 10 *dimensional input drawn from* $N(0,1)$,
$y = 1$ *if* $(\sum_{i=1}^{10} x_i^4/10)^{1/4} > 1.2$ *else* $y = 0$

NN architecture used - **10 X 100 X 100 X 1**, Kernel dimension = **100**
Class 1 -
Number of training/test examples for SKPCA = 539/1380
Number of PCs used = 15
Average sparsity of each PC = **27 out of 539 (only 100 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.27%**
Class 0 -
Number of training/test examples for SKPCA = 461/1120
Number of PCs used = 15
Average sparsity of each PC = **22 out of 461 (only 100 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.44%**

3. $x = [x_1, x_2, ..., x_{10}]$ − 10 *dimensional input drawn from* $unif(0,1)$,
$y = 1$ *if* $(\sum_{i=1}^{10} |log(x_i)|/10) > 0.95$ *else* $y = 0$

NN architecture used - **10 X 100 X 100 X 1**, Kernel dimension = **100**
Class 1 -
Number of training/test examples for SKPCA = 499/1225
Number of PCs used = 15
Average sparsity of each PC = **42 out of 499 (only 100 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.28%**
Class 0 -
Number of training/test examples for SKPCA = 501/1275
Number of PCs used = 15
Average sparsity of each PC = **42 out of 501 (only 100 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.22%**

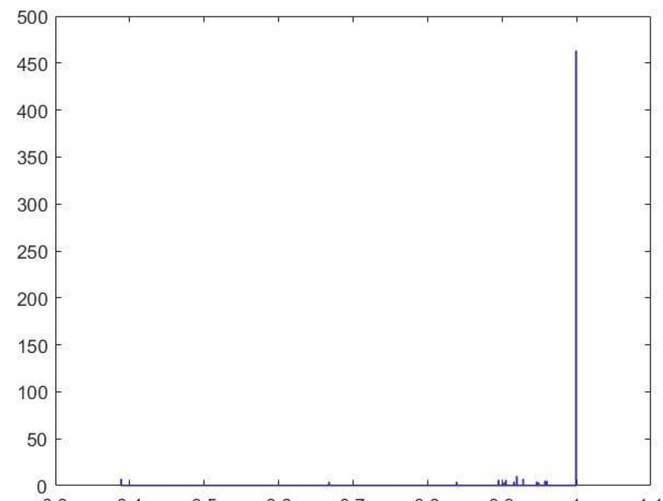
Real world datasets

4. Kaggle Dogs vs. Cat Classification dataset-
Architecture used - **Resnet**, Kernel dimension = **1024**
Class 1 -
Number of training/test examples for SKPCA = 769 (<1024), 2493
Number of PCs used = 15
Average sparsity of each PC = **223 out of 769 (=29%)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.36%**
Class 0 -
Number of training/test examples for SKPCA = 731 (<1024), 2497
Number of PCs used = 15
Average sparsity of each PC = **199 out of 731 (=27.2%)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.30%**

5. Kaggle Breast Histopathology Image Classification dataset (https://www.kaggle.com/paultimothymooney/breast-histopathology-images)-
Architecture used - **Wide Resnet**, Kernel dimension = **512**
Class 1 -
Number of training/test examples for SKPCA = 764/20000
Number of PCs used = 15
Average sparsity of each PC = **113 out of 764 (only 512 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.42%**
Class 0 -
Number of training/test examples for SKPCA = 736/20000
Number of PCs used = 15
Average sparsity of each PC = **114 out of 736 (only 512 required actually)**
RRMSE of reconstruction from first 15 sparse PCs wrt first 15 actual PCs = **1.21%**

Theoretical Justification

Consider the set S_m of m training points in the kernel space $\{\phi(x_1), \phi(x_2), \ldots, \phi(x_m)\}$ such that its elements are i.i.d and also the subset $S_n = \{\phi(x_1), \phi(x_2), \ldots, \phi(x_n)\}$ of $n < m$ points. Here, S_n obeys a particular property. An eigenvector of the covariance matrix can be approximately expressed as a linear combination of the elements of S_n if $\forall\ i > n, \exists\ v_i \in \text{span}\{\phi(x_1), \phi(x_2), \ldots, \phi(x_n)\}$ such that $|\langle v_i, \phi(x_i)\rangle|/(\|v_i\| \times \|\phi(x_i)\|)$ is "very" close to 1 $\forall\ i > n$. The particular property of S_n referred to earlier is precisely the property of having a vector v_i in its span $\forall\ i > n$ such that v_i and $\phi(x_i)$ are "nearly" parallel to each other. Given that we have $\binom{m}{n}$ choices for S_n, we expect to find at least one set satisfying the desired property with high probability.



Histogram of points for i>n (with n = 200 and m = 731) and the maximum normalized absolute dot product with a vector in the span of the first n points for the dog vs. cat dataset

Conclusions

- Based on our experiments, we conjecture that DL kernels also exhibit sparsity in their representation similar to traditional kernels like the RBF kernel. We also provide some theoretical justification to our findings.
- Thus, even though DL models require a huge number of examples to learn the kernel properly, the kernel itself can generalize well using very few examples.