Machine Learning with Insufficient Data Amount (Or SVM and CNN?)

Phan Minh Nguyen — Stanford University

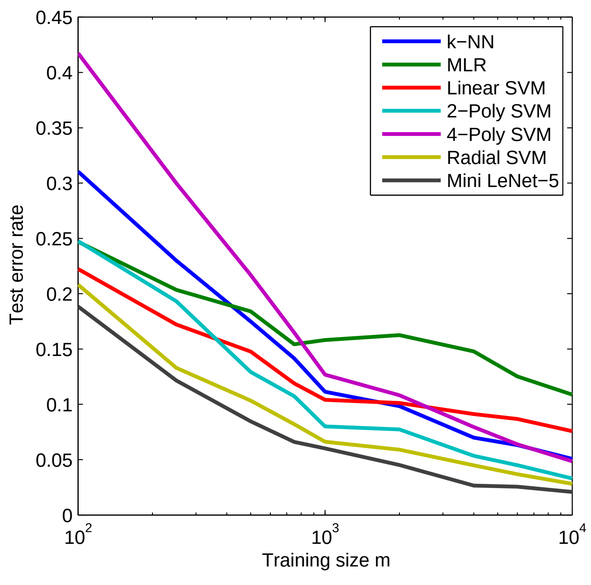
Introduction

The big data regime posits a basic assumption of data abundance. **What should we do when we have a lot of training data, but also a lot of parameters to learn?** We already know this hurts regression tasks. Overfitting is a major concern.

The project explores this high-dimensional setting for supervised digit recognition on MNIST data set, containing 28×28 images. We test the performance for different training sizes $m = 100, 250, ..., 10000$, and draw empirical conclusions for k-nearest neighbor (k-NN), multi-class logistic regression (MLR), SVM and (small, relatively deep) convolutional neural network (CNN).

Overall: CNN beats all

We test several representative algorithms, without regularization.

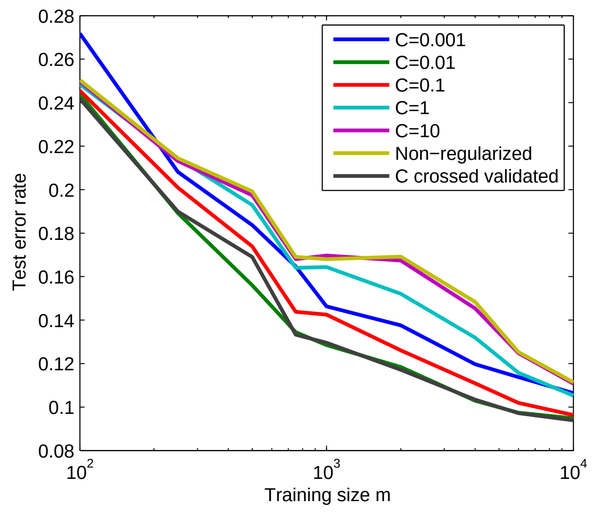


Observations:

- CNN works consistently best. MLR is consistently not great.
- k-NN and strongly kernelized SVM can get degraded, although SVM with radial kernel is the runner-up.

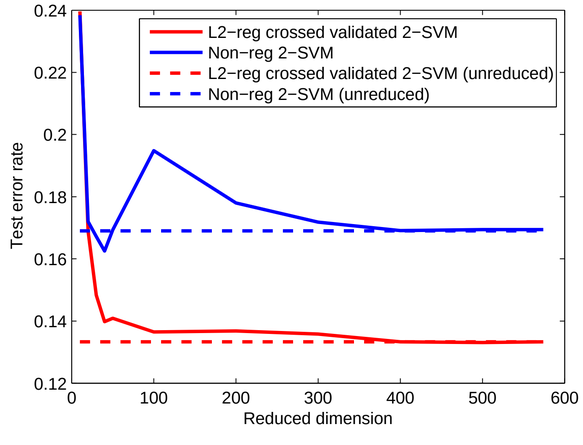
SVM: ℓ_2^2 regularization is (still) very helpful

We test linear SVM with ℓ_2^2 regularization. Smaller C imposes heavier ℓ_2^2 penalization. Oddly, SVM should be insensitive to overfitting...



PCA may not be a terrific idea

We test linear SVM with PCA preprocessing for $m = 750$. Had PCA worked, we should have seen a clear drop in error rate at some intermediate dimension.

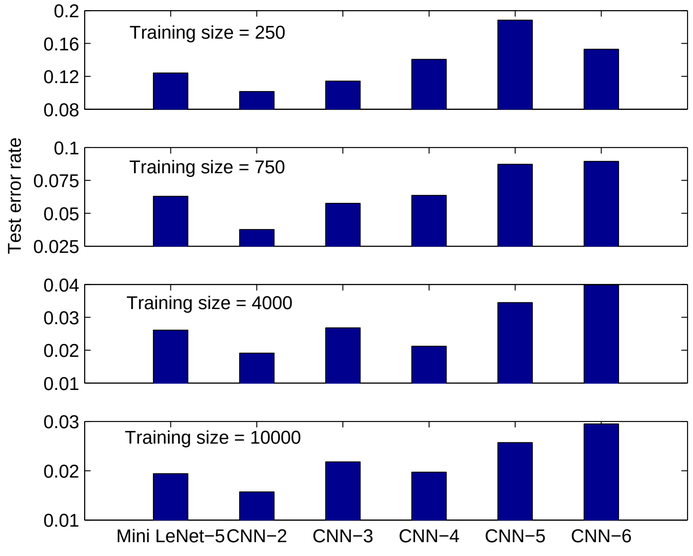


CNN: (Un)predictable

We test several CNN architectures. We list below the structure of convolutional layers, where C1, C3, C5 refer to conv layers of original LeNet-5, and 20 refers to a conv layer with 20 maps (same for 40, 60, 120). In-between subsampling layers and the last fully connected layer are not shown. All layers have trainable weights.

We test their performances without regularization.

	Conv. structure	Depth	# Conv weights
Mini LeNet-5	C1-C3-C5	6	50.8k
CNN-2	C1-C3-20-120	8	13.7k
CNN-3	C1-C3-20-40-120	10	12.1k
CNN-4	C1-C3-20-40-120	10	50.5k
CNN-5	C1-C3-20-40-60-120	12	45.7k
CNN-6	C1-C3-20-40-60-60-120	14	38.5k



Note that even with $m = 10000$ or more, we are still in the regime # training data $\ll$ # parameters. Observations:

- Changing the training size may alter the relative performance.
- Reducing the number of trainable weights may help when there is less data. But not necessarily.
- Increasing depth may not help.