CS156b Project1

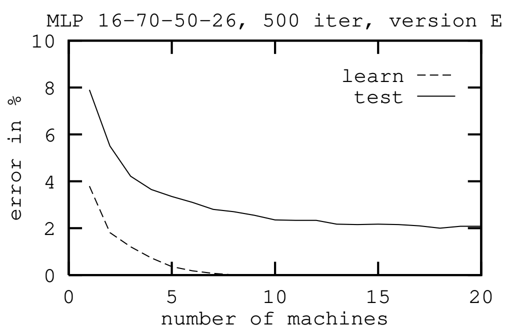
Letter Recognition — AdaBoost & More layers

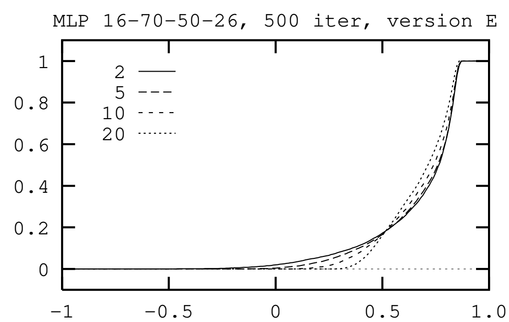
Ling Li
Learning Systems Group, Caltech
November 13, 2001

What can be improved

Last week I showed some results of bagging and boosting with 16-50-26 nets. The testing error decreases from 11% (one hypothesis) to 5% (simple boosting/AdaBoost.M1). Errors of bagging are higher.

According to Schwenk and Bengio's Boosting Neural Networks, the errors should be 7% (one hypothesis) and below 2% (AdaBoost.M2).





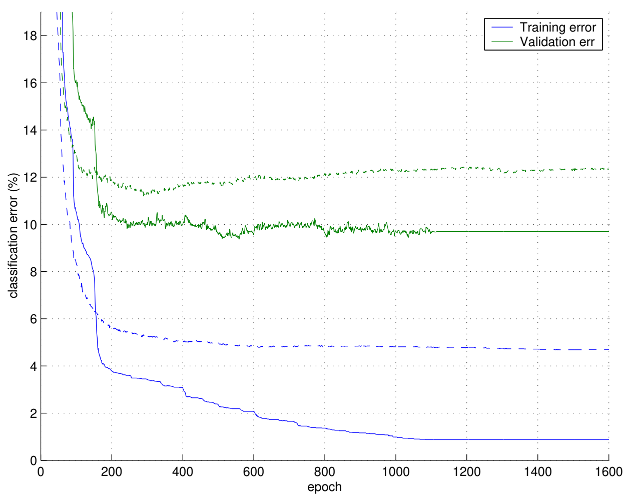
Figures from Schwenk's paper

I should improve

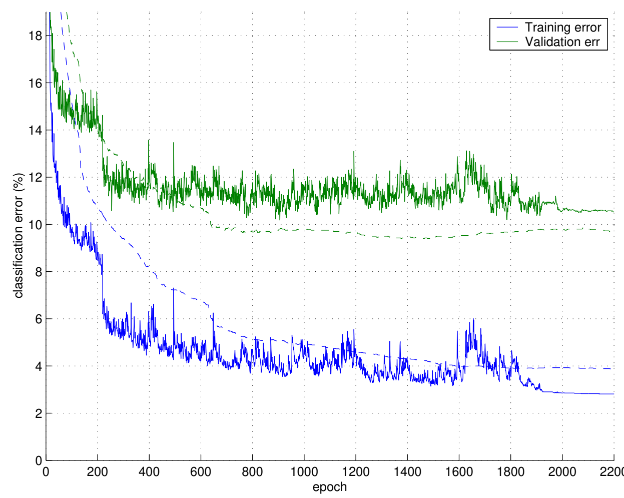
- the base architecture and
- the aggregation algorithm.

More Layers

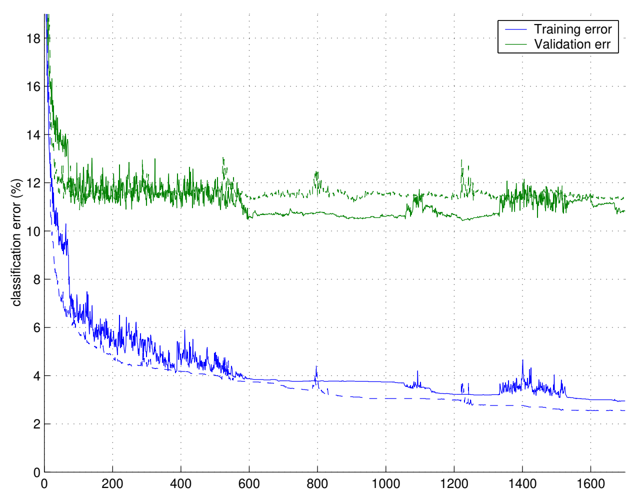
I compared 16-70-50-26 with 16-50-26.



Conjugate-Gradient



Momentum



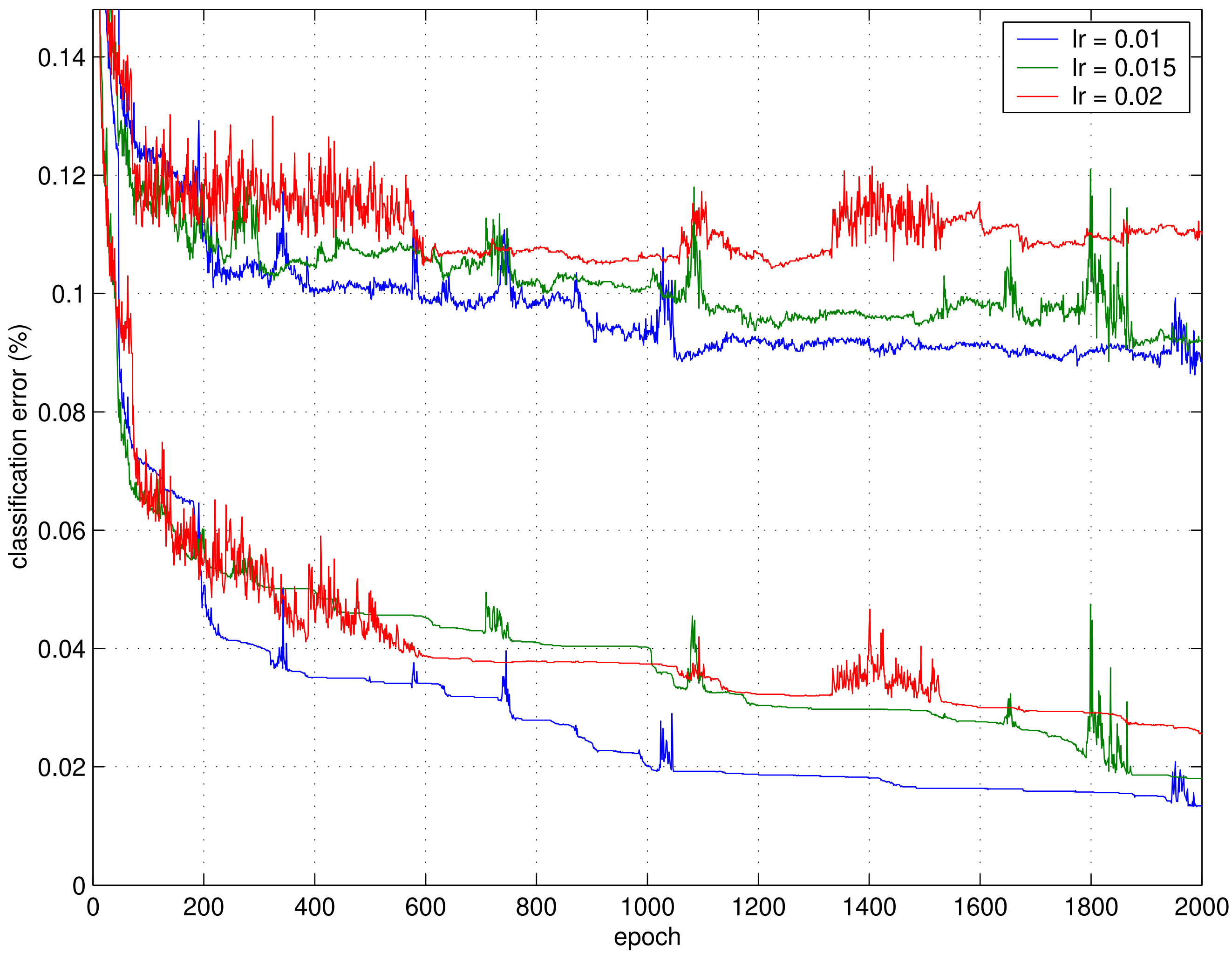
Stochastic Gradient Descent

Stochastic Gradient Descent

Randomly select a sample according to distribution D, for each feed-forward / back-propagation.

Schwenk claimed this is better than conjugate-gradient, since there are so many examples.

I tested this method with different learning rates.



AdaBoost

Freund and Schapire. Experiments with a New Boosting Algorithm.

- We have a "weak" learning algorithm; want to make it "strong"
- Boost it! Combine many "weak" hypotheses
- Different from bagging: emphasize on "bad" samples; combine hypotheses with different weights

AdaBoost.M1

We have $D_t(i)$ as the weight for i^{th} sample (x_i, y_i) at iteration t

- generate h_t according to samples and $D_t(i)$
- error is $\epsilon_t = \sum_i D_t(i) \cdot (h_t(x_i) \neq y_i)$
- $\beta_t = \epsilon_t^{-1} - 1$; h_t has weight $\log \beta_t$.
- Update D as $D_{t+1}(i) \propto D_t(i) \beta_t^{(h_t(x_i) \neq y_i)}$

We need $e_t < 0.5$. For two-class problem, it is easy; however, ...

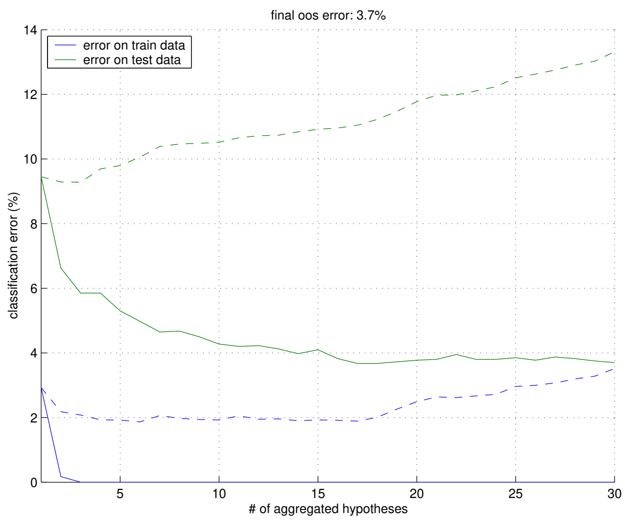
AdaBoost.M2 for multi-class problems

We have $D_t(i, y)$ as the weight for x_i and $y \neq y_i$ at iteration t

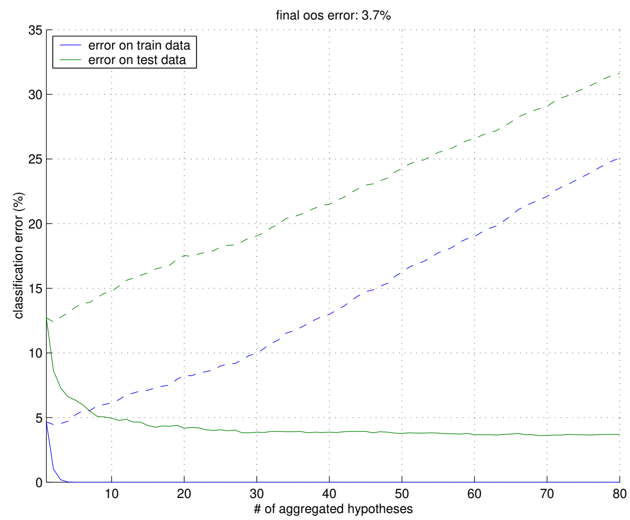
- generate h_t according to samples and $D_t(i, y)$
- *pseudo-loss* is $\epsilon_t = \frac{1}{2} \sum_{i,y} D_t(i) \cdot (1 - h_t(x_i, y_i) + h_t(x_i, y))$
- $\beta_t = \epsilon_t^{-1} - 1$; h_t has weight $\log \beta_t$.
- Update D as $D_{t+1}(i, y) \propto D_t(i, y) \beta_t^{\frac{1}{2}(1 - h_t(x_i, y_i) + h_t(x_i, y))}$

Now boosting algorithm has more communication with the "weak" one. $h_t(i, y)$ is the likelihood that x_i belongs to class y.

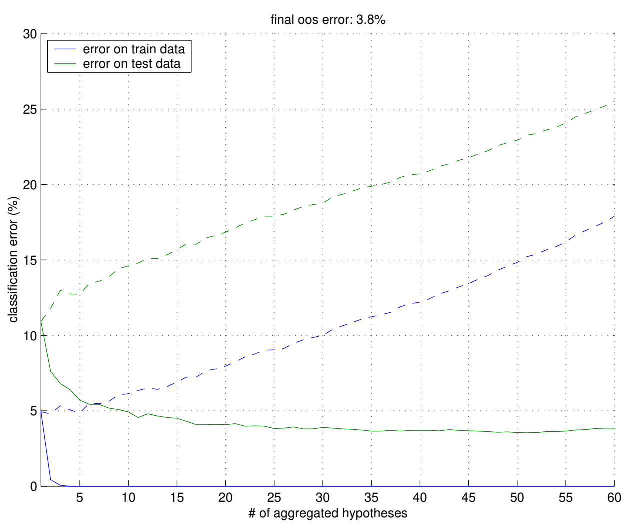
Results with AdaBoost.M2



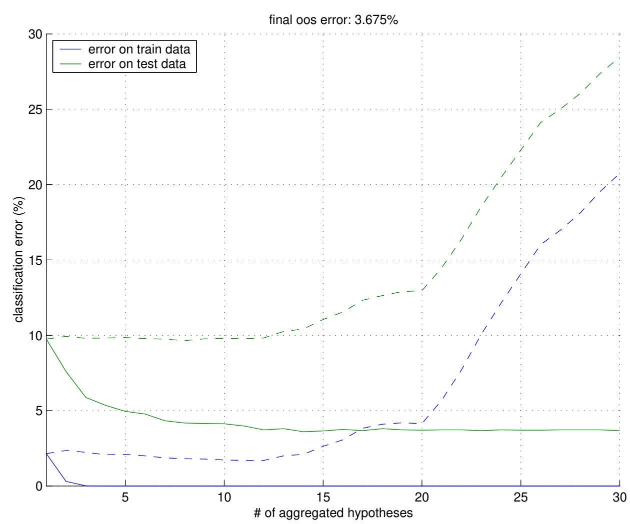
SG, 1000, 30, 0.01



SG, 500, 80, 0.02



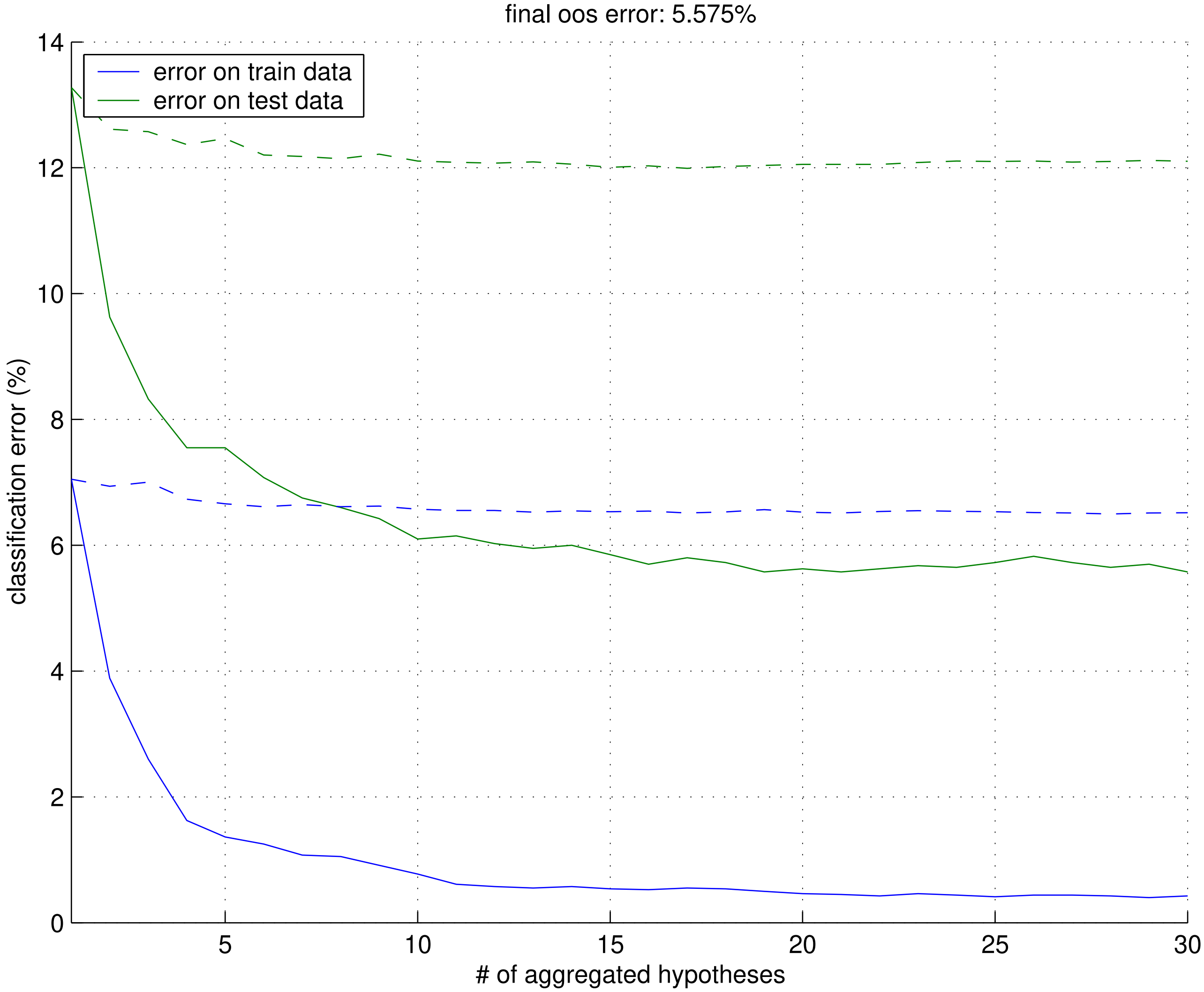
MM, 500, 60, 0.01



CG, 500, 30, 0.02

Results with Bagging

Bagging still does a worse job.



SG, 500, 30, 0.01