Explaining and Harnessing Adversarial Examples

Subham Ghosh

CONS

Results

the validation set error rate has not decreased for 100 epochs. We found that while the validation set error was very flat, the *adversarial* validation set error was not. We therefore used early stopping on the *adversarial validation set error*. Using this criterion to choose the number of epochs to train for, we then retrained on all 60,000 examples. Five different training runs using different seeds for the random number generators used to select minibatches of training examples, initialize model weights, and generate dropout masks result in four trials that each had an error rate of 0.77% on the test set and one trial that had an error rate of 0.83%. The average of 0.782% is the best result reported on the permutation invariant version of MNIST, though statistically indistinguishable from the result obtained by fine-tuning DBMs with dropout (Srivastava et al., 2014) at 0.79%.

Results

the validation set error rate has not decreased for 100 epochs. We found that while the validation set error was very flat, the *adversarial* validation set error was not. We therefore used early stopping on the *adversarial validation set error*. Using this criterion to choose the number of epochs to train for, we then retrained on all 60,000 examples. Five different training runs using different seeds for the random number generators used to select minibatches of training examples, initialize model weights, and generate dropout masks result in four trials that each had an error rate of 0.77% on the test set and one trial that had an error rate of 0.83%. The average of 0.782% is the best result reported on the permutation invariant version of MNIST, though statistically indistinguishable from the result obtained by fine-tuning DBMs with dropout (Srivastava et al., 2014) at 0.79%.

- No formal results section with tabular comparison across different kinds of networks

‘Not So Linear’ Models

- Mostly linear models explored
- What about more complex models?
- LSTMs?
- Do the same kind of methods work on such models?

Generalization

An intriguing aspect of adversarial examples is that an example generated for one model is often misclassified by other models, even when they have different architecures or were trained on disjoint training sets. Moreover, when these different models misclassify an adversarial example, they often agree with each other on its class. Explanations based on extreme non-linearity and over-

Citations?

Thank You