Regularization

Lyle Ungar

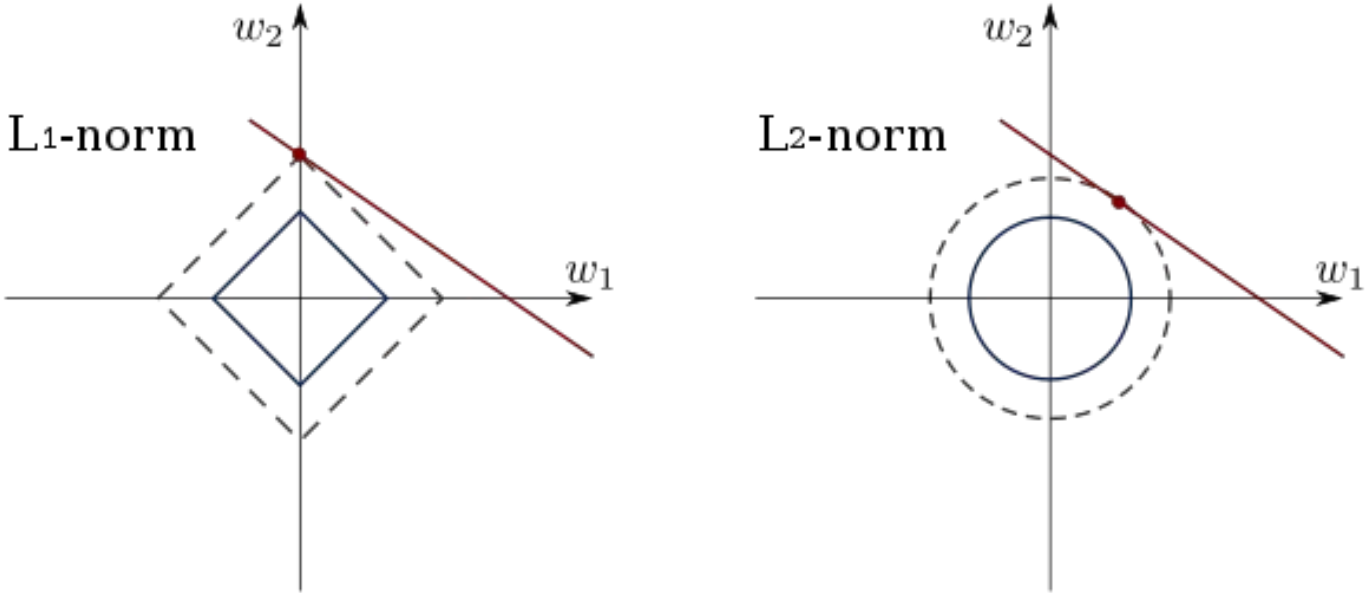
Regularization via penalties

Lyle Ungar

Regularization penalties

- Train to minimize normal loss + c * L_1(weights)
  - L_1: sum of the absolute values of the weights; like lasso regression
  - Drives some weights to 0
- Train to minimize normal loss + c * L_2(weights)
  - L_2: sum of the squares of the weights; like ridge regression
  - Makes the biggest weights smaller
- Train to minimize normal loss - but don't let the weights get too big
  - Like an L-infinity penalty

L1 vs L2 Regularization



Regularization via data augmentation

Lyle Ungar

Data augmentation

- We never have enough labeled data, so generate more

- Take labeled images and
  - flip them left/right
  - shift them up/down/right/left by a couple pixels
  - add small noise
  - ...

Kritzhevsky, Shutskever, & Hinton, 2012

Regularization via SGD

Lyle Ungar

SGD does regularization

- Initialize with small random weights
- Weights get bigger as one iterates
- Use early stopping to avoid overfitting

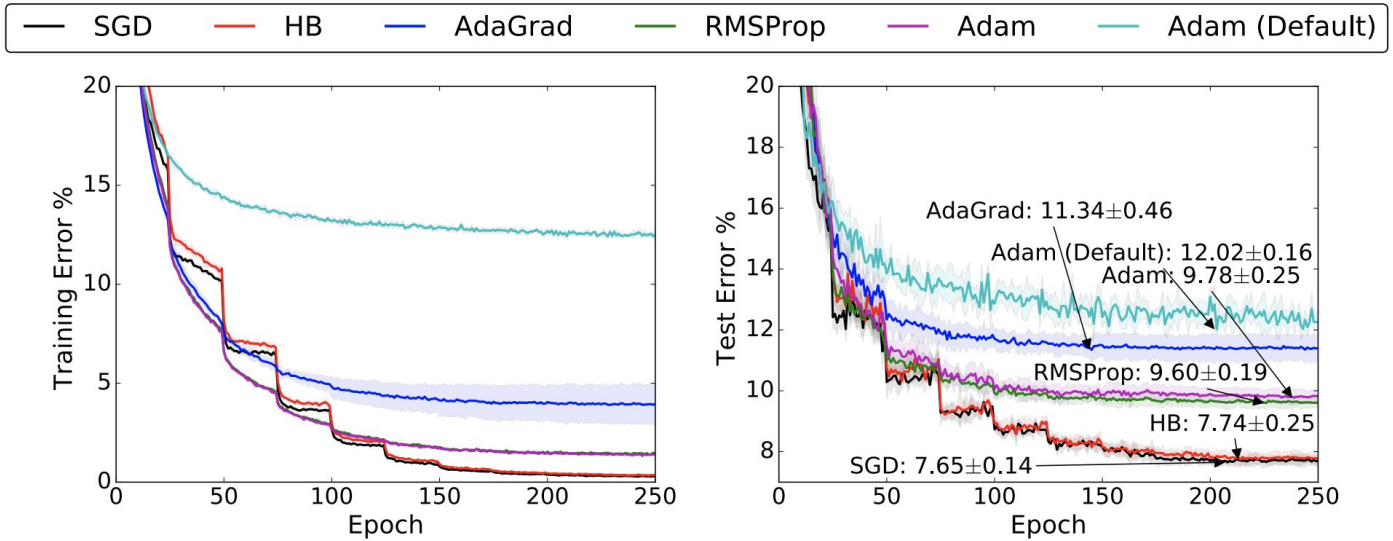
SGD finds ‘good’ minima

- Deep learning often uses more parameters than observations
  - ‘should’ massively overfit
- Deep learning on CIFAR10 and IMAGENET
  - gives 0 training error with small test error
  - gives 0 training error with randomized labels
- **SGD converges to flat minima, which generalize better**

Gradient descent is magic: it tends to find ‘good’ (smooth/regularized) solutions

Zhang, Bengio, Hardt, Recht, and Vinyals 2017

Gradient descent methods converge to different solutions



(a) CIFAR-10 (Train)

(b) CIFAR-10 (Test)

The marginal value of adaptive gradient methods in ML: Wilson.. & Recht

How learning rate affects learning

- **Small learning rates are more likely to find a deep minimum**
  - this could be good or bad
- **Large learning rates regularize**
  - Deep minima often generalize badly
  - Broader, flatter minima may be more robust
  - Larger learning rate misses the deep minima, finds the shallower ones

In Theory!

Small batch size ⇔ large learning rate

- Small batch size => noisy estimates of gradient
- Also can improve generalization
- Increasing the batch size is like decreasing the learning rate
- ½ learning rate can be interpreted as either 2 x batch size or 4 x batch size, depending on perspective

A Modern Take on the Bias-Variance Tradeoff in Neural Networks

Brady Neal, Sarthak Mittal, Aristide Baratin, Vinayak Tantia, Matthew Scicluna, Simon Lacoste-Julien, Ioannis Mitliagkas

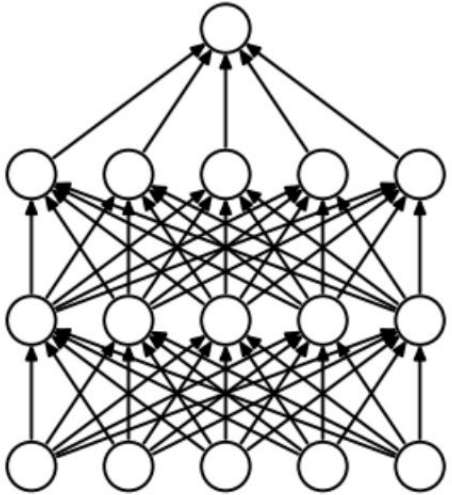
The bias-variance tradeoff tells us that as model complexity increases, bias falls and variances increases, leading to a U-shaped test error curve. However, recent empirical results with over-parameterized neural networks are marked by a striking absence of the classic U-shaped test error curve: test error keeps decreasing in wider networks. This suggests that there might not be a bias-variance tradeoff in neural networks with respect to network width, unlike was originally claimed by, e.g., Geman et al. (1992). Motivated by the shaky evidence used to support this claim in neural networks, we measure bias and variance in the modern setting. We find that both **bias and variance can decrease as the number of parameters grows**. ...

Regularization via dropout

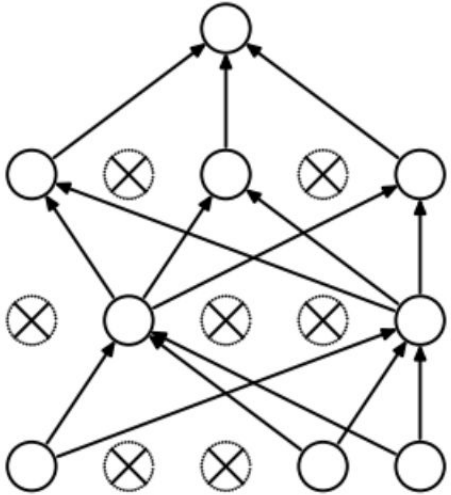
Lyle Ungar

Dropout

- Randomly remove a fraction p of the nodes for each mimibatch
  - Usually p = ½
- For the final network
  - use all the weights
  - but shrink them by p



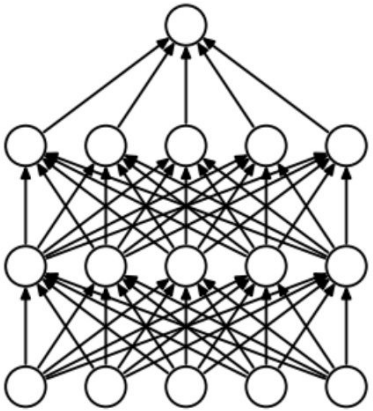
(a) Standard Neural Net



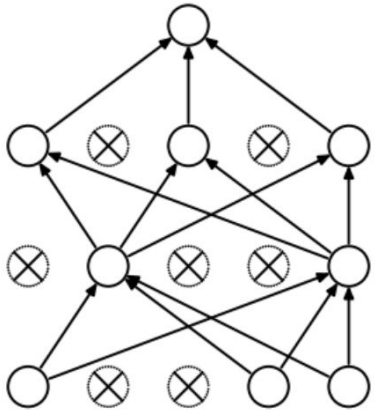
(b) After applying dropout.

Dropout

- This samples (in theory) over exponentially many networks
  - in effect ensembling them
- Bounces the network out of local minim



(a) Standard Neural Net



(b) After applying dropout.

Dropout

- Adds lots of noise
- Prevents overfitting by breaking brittle predictions
- Distributes learning signal
- Enforces distributed representation
- Forces "dead branches" to learn
- In a way approximates ensemble methods

Hyperparameter tuning

Lyle Ungar

People combine many types of regularization

- L0, L1, L2 penalties
- early stopping
- data augmentation
- dropout

Hyperparameter tuning is search

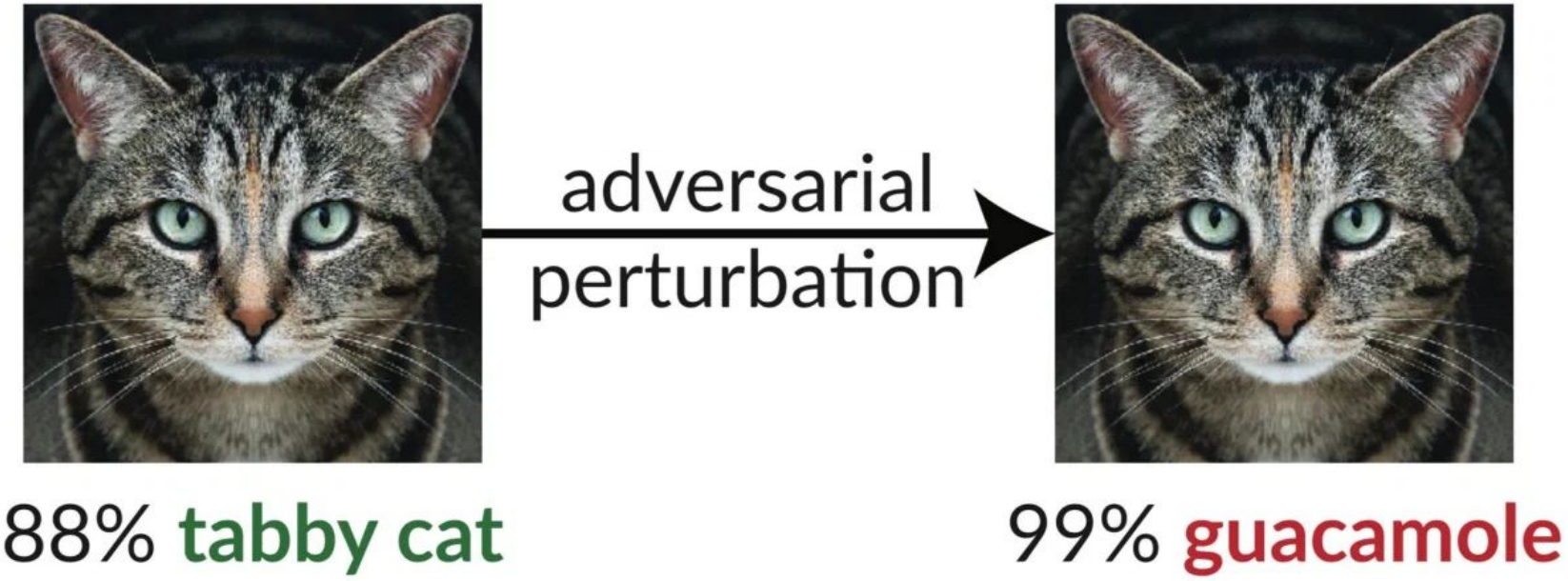
- **grid search**
  - try all possible combinations of hyperparameters
- **randomized search**
  - randomly try different combinations
- **coordinate-wise gradient descent**
  - change them one at a time, accepting any changes that reduce testing error
- **Bayesian optimization/AutoML**
  - start with hyperparameters that worked well for similar problems

more info

Adversarial attacks and defenses

Lyle Ungar

Adversarial perturbations



Adversarial examples

$\boldsymbol{x}$

"panda"
57.7% confidence

$+ .007 \times$

$\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

"nematode"
8.2% confidence

$=$

$\boldsymbol{x} + \epsilon\text{sign}(\nabla_{\boldsymbol{x}} J(\boldsymbol{\theta}, \boldsymbol{x}, y))$

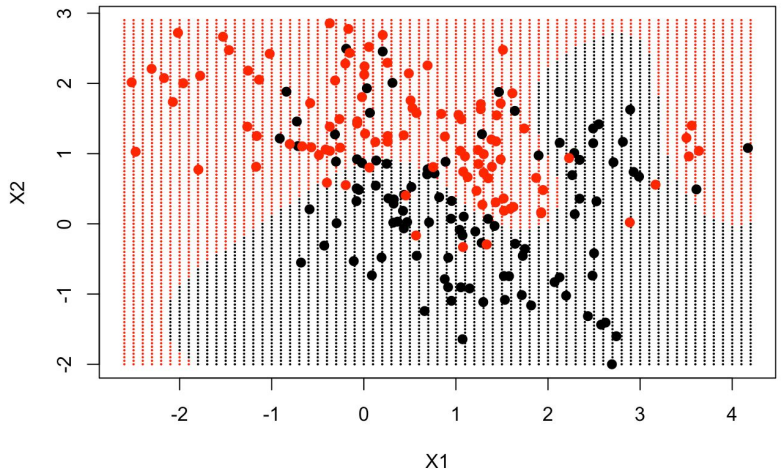
"gibbon"
99.3 % confidence

learn more


rifle
shield, buck
revolver, si

Adversarial examples

- Inevitable consequence of learning in a high-dimensional space?



Adversarial attacks

- Can one defend against them?
  - regularize: ‘defensive distillation’ or ‘feature squeezing’
  - pick weights to minimize the objective function that the adversary is trying to maximize (when searching over perturbations)

Towards Deep Learning Models Resistant to Adversarial Attacks
Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, Adrian Vladu

Lastly

Make sure you provide feedback.

Continuous improvement!