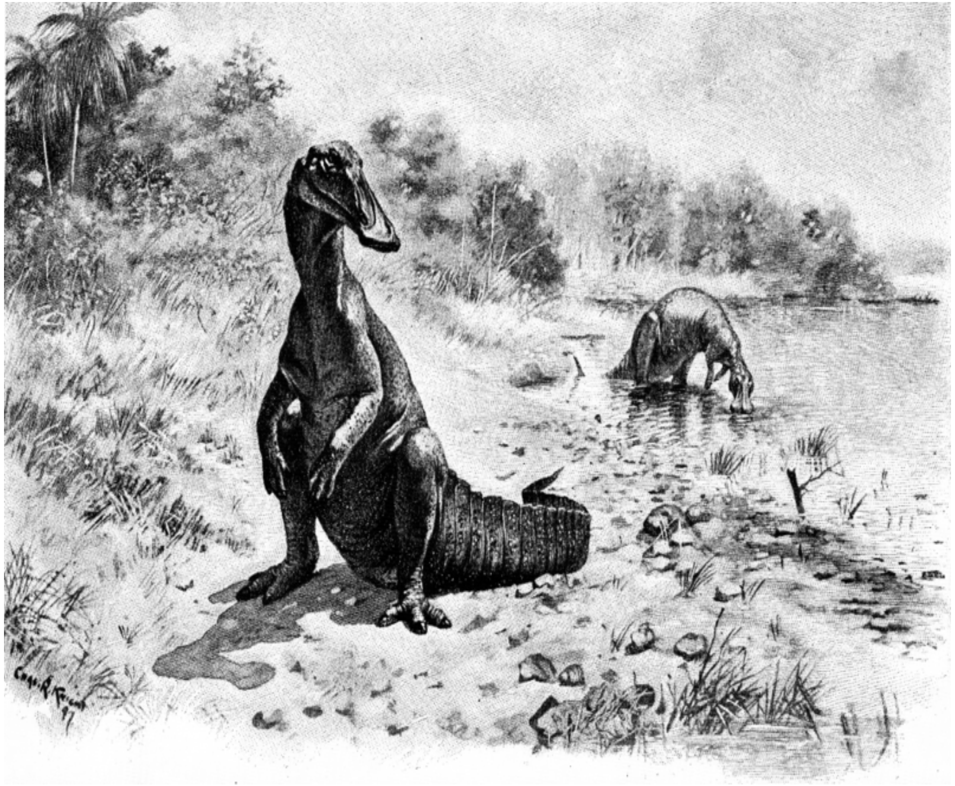
Deep Neural Networks for Image Classification: Architectures, Approaches, and Advice

Kyle Kastner
University of Texas - San Antonio

Previously, on FOX...

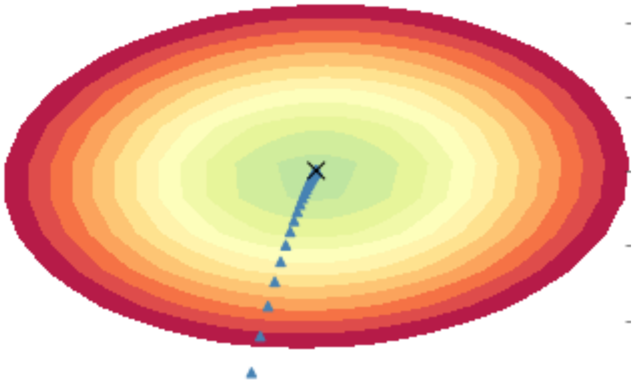
- Datasets
  - CIFAR-10
  - Asirra subset
- Partial success
  - CIFAR10 was good
  - Did not generalize
- Better preprocessing
- New advances
- Now… a recap



http://commons.wikimedia.org

Techniques

- Zero phase components (ZCA) preprocessing
- Global Contrast Normalization (GCN)
- Flip randomly during training
- Trained by SGD
- Convolutional units
- Maxout units
- Dropout
- Softmax classification



http://kastnerkyle.github.io

Preprocessing

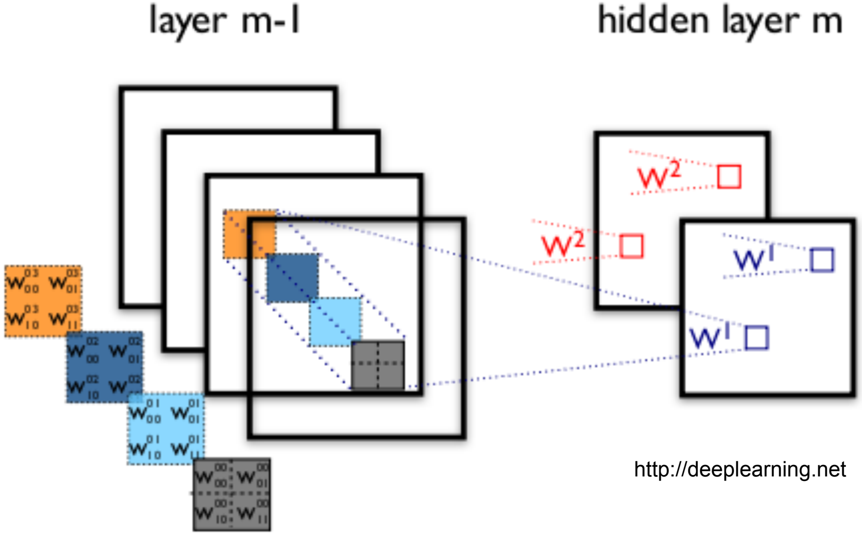
- ZCA mimics the processing of the human visual system
- Similar concept to Principle Component Analysis (PCA)
- Typically used only for images
- **Bell A.J. and Sejnowski T.J. 1996.** *The `Independent Components' of natural scenes are edge filters*

$$W = (XX^T)^{\frac{1}{2}} = ED^{\frac{1}{2}}E^T$$

$$X_w = XW$$

Convolutional Units

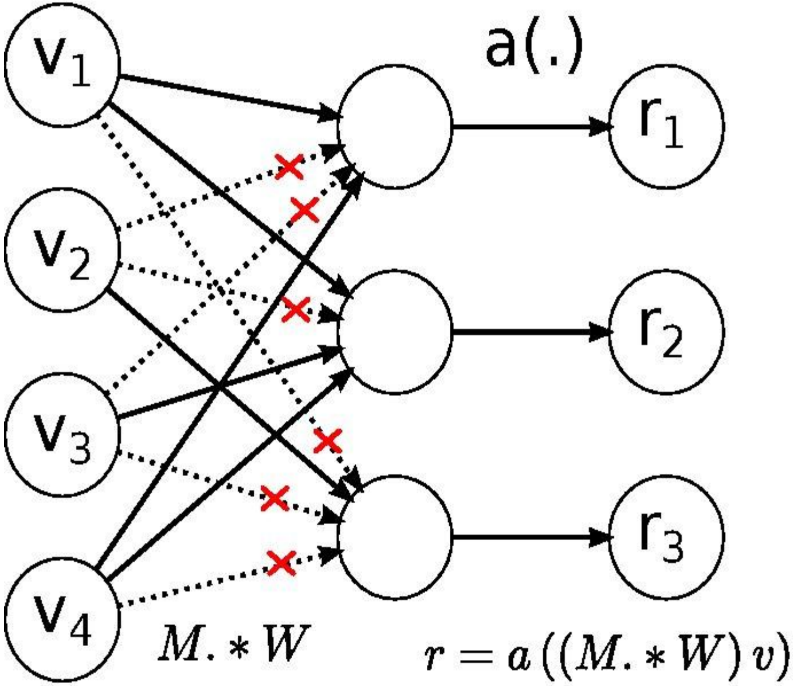
- Inspired by research on cat visual cortex
- Used to exploit correlation in input
- Designed for vision tasks


layer m-1
hidden layer m
W^2
W^2
W^1
W^1
W^{03}_{00} W^{03}_{01}
W^{03}_{10} W^{03}_{11}
W^{02}_{00} W^{02}_{01}
W^{02}_{10} W^{02}_{11}
W^{01}_{00} W^{01}_{01}
W^{01}_{10} W^{01}_{11}
W^{00}_{00} W^{00}_{01}
W^{00}_{10} W^{00}_{11}

http://deeplearning.net

Dropout

- Randomly drop ~50% of input
- Typically keep 80% in first layer
- Equivalent to training many nets
- Very strong regularizer
- Limit weight adaptation
- Typically ReLU or Maxout units



http://cs.nyu.edu

Maxout

- Uses localized “max” units
- Shown to be an approximator for many functions
- Designed to be used in conjunction with dropout

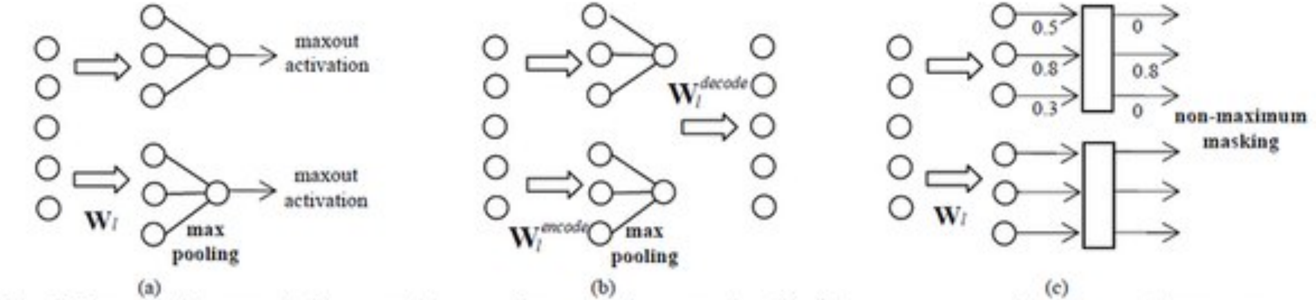
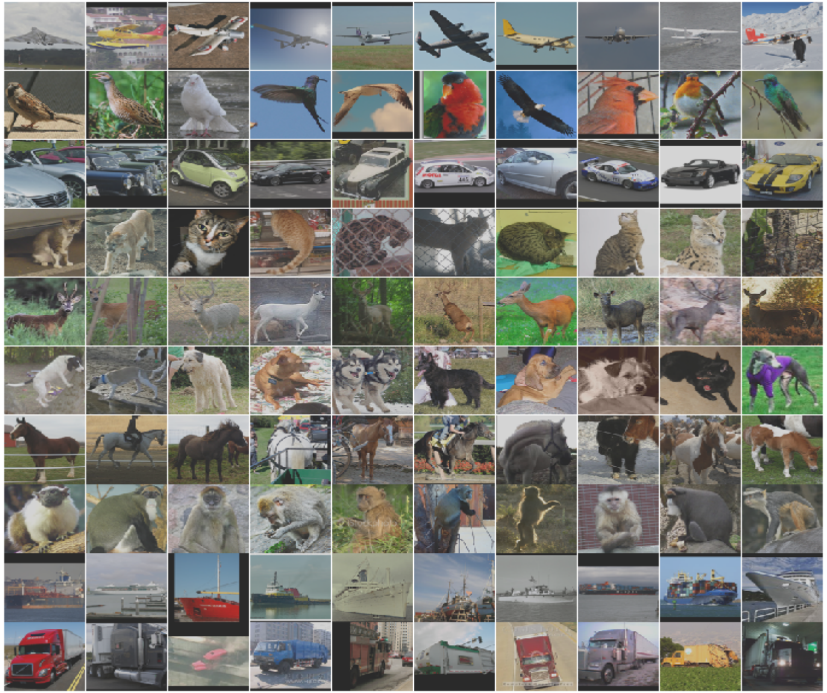


Fig. 1. Maxout architectures in this paper: (a) maxout layer with the group size of 3; (b) maxout autoencoder; (c) sparse feature extractor.

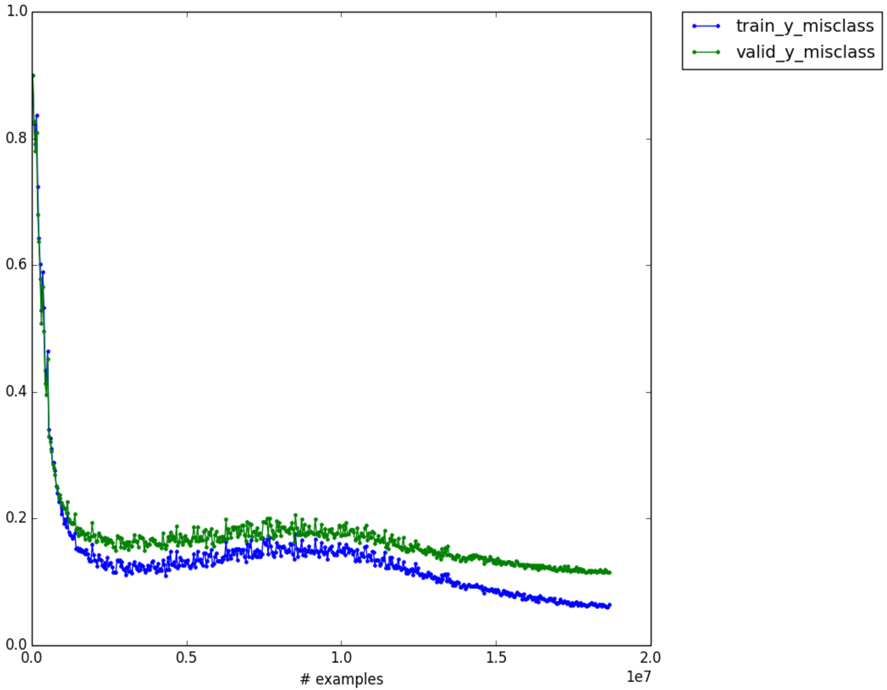
https://portal.xsede.org/publications/-/publications/715

First Experiment

- CIFAR10 dataset
- Standard benchmark for vision
- Replicate performance
- Planned to extend for Asirra
- Results: **88.9%** accuracy
- Similar:
    - CIFAR100
    - ImageNet, STL-10



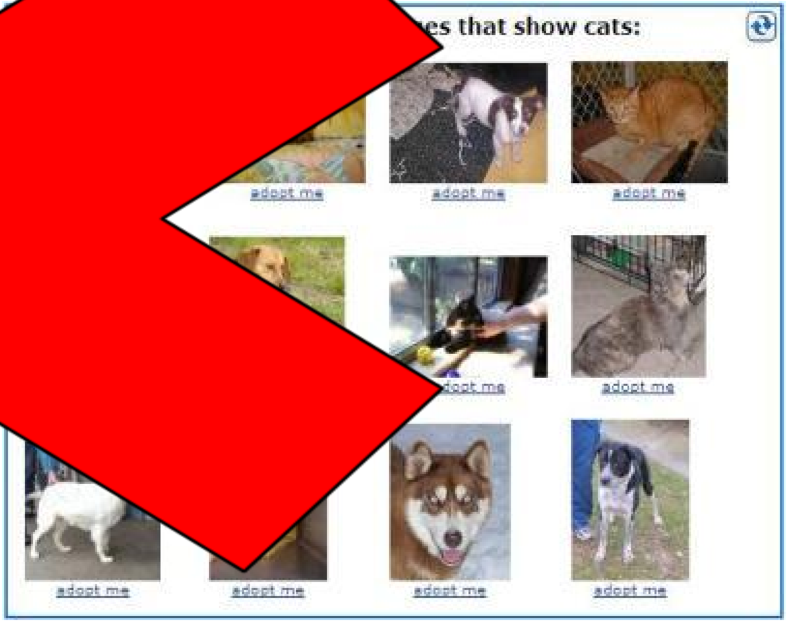
http://www.stanford.edu/~acoates//stl10/



Results from CIFAR10 Training

Failed Extension

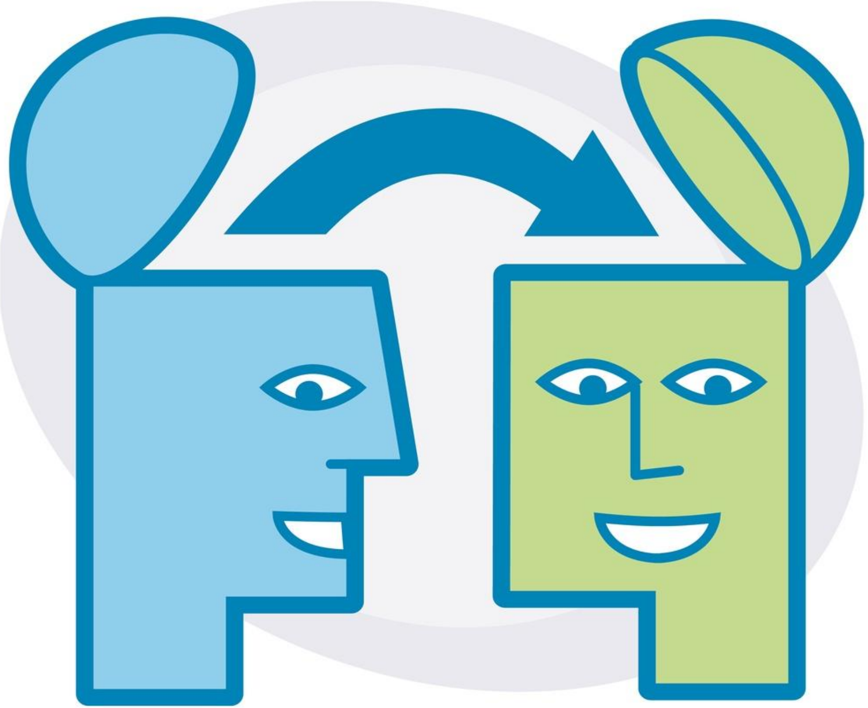
- Image bas
- Developed by
- Kaggle Dogs vs. Ca
- SOTA: **96%**
- Best accu
- This is **BAD!**



http://www.peta.org

Successful Approach

- Transfer learning
- Trained for ImageNet
  - 1000 classes!
- Cut off softmax layer
- Forward propogate new data
- Use output as new features



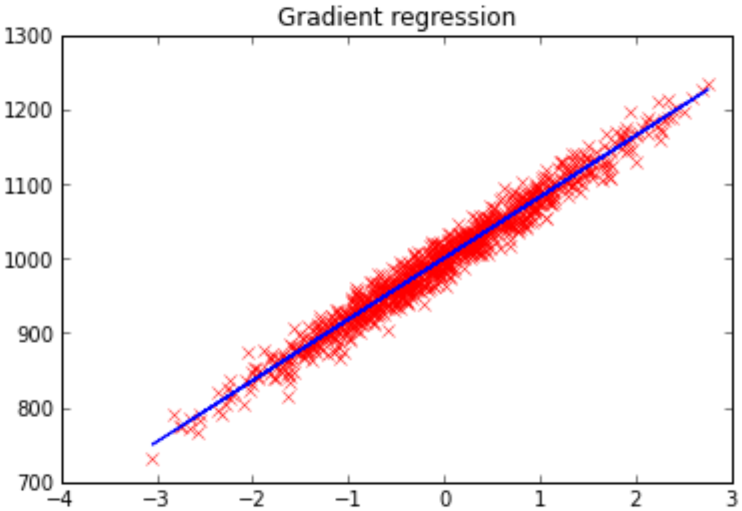
DeCAF

- Used for preprocessing
- Details here:
  - http://decaf.no-ip.org/
- Newer, better version
  - Caffe
  - http://daggerfs.com/caffe/
- ImageNet
  - 120 GB labeled images
  - 1.2 TB total images
  - http://www.image-net.org

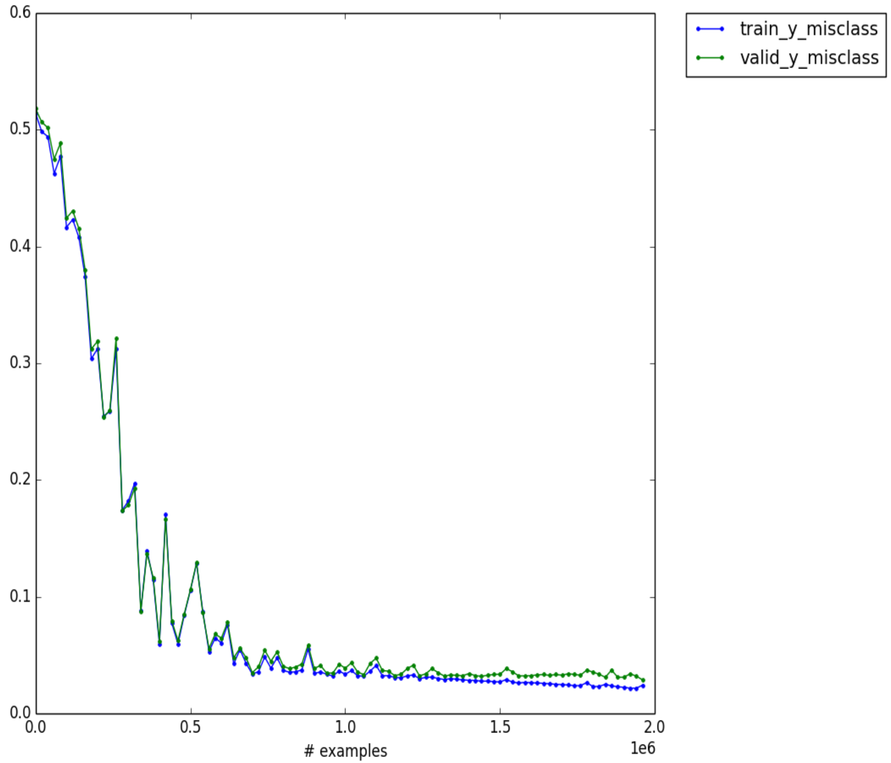
http://commons.wikimedia.org

Results

- Tried several classifiers
  - Logistic Regression: **94.6%**
  - Linear SVM: **94.3%**
  - Polynomial SVM, degree 3: **96.1%**
  - Deep classifier, softmax: **96.67%**
  - Deep classifier, SVM cost: **??%**
- Deep Classifier
  - **5000-5000-5000-5000-2**
  - Learning rate **.001**
  - Rectified Linear units
  - Softmax cost
  - SVM cost *should* improve ~1%



Graphic from unrelated dataset
http://kastnerkyle.github.io



Results from deep classifier training

Discussion

- Reduced computation
    - Network training needs G
    - SVM is possible on embe
- Minimal increases WRT comp
    - Indicates **GOOD** features
    - At least for this task
- SOTA was ~**80%**
    - Currently **97.04%**
    - *And climbing!*

http://www.kno.com

Advice

- Many hyperparameters don’t matter!
  - *Random Search for Hyper-Parameter Optimization*
  - Bergstra, Bengio 2012
  - Sizes and number of layers - ***not very important***
  - Learning rate - ***very important***
  - Type of activation functions - ***very important***
  - Regularization - ***dropout***
- Train and tune network until strongly overfit
  - Then add dropout
- Advanced hyperparameter search packages
  - Hyperopt, Spearmint

More Advice

- Preprocessing is vital!
  - Global Contrast Normalization (GCN)
  - ZCA
  - Flip upside-down randomly (p = .5) during training
- Rectified Linear, Maxout are very good with dropout
- For more, read this paper:
  - http://arxiv.org/abs/1206.5533
- GPU is critical for big networks
  - Use pretrained nets to simplify
  - Transfer learning has great potential

Final Thoughts

- Links
  - https://github.com/lisa-lab/pylearn2
  - https://github.com/UCB-ICSI-Vision-Group/decaf-release
  - https://github.com/kastnerkyle/kaggle-dogs-vs-cats
  - https://github.com/kastnerkyle/kaggle-cifar10
  - http://arxiv.org/abs/1206.5533

Thank you!