Maxout Networks

Hien Quoc Dang

Outline

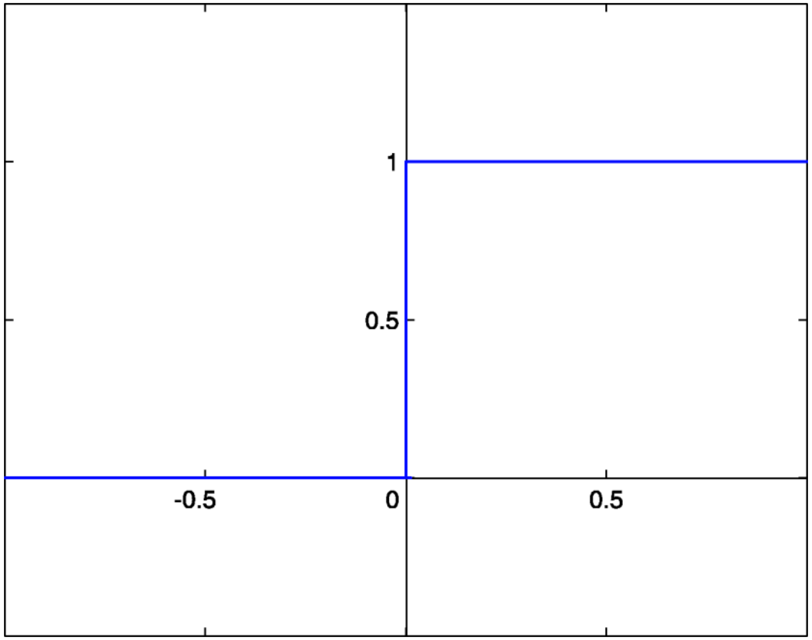
- Introduction
- Maxout Networks
  - Description
  - A Universal Approximator & Proof
- Experiments with Maxout
- Why does Maxout work?
- Conclusion

Introduction

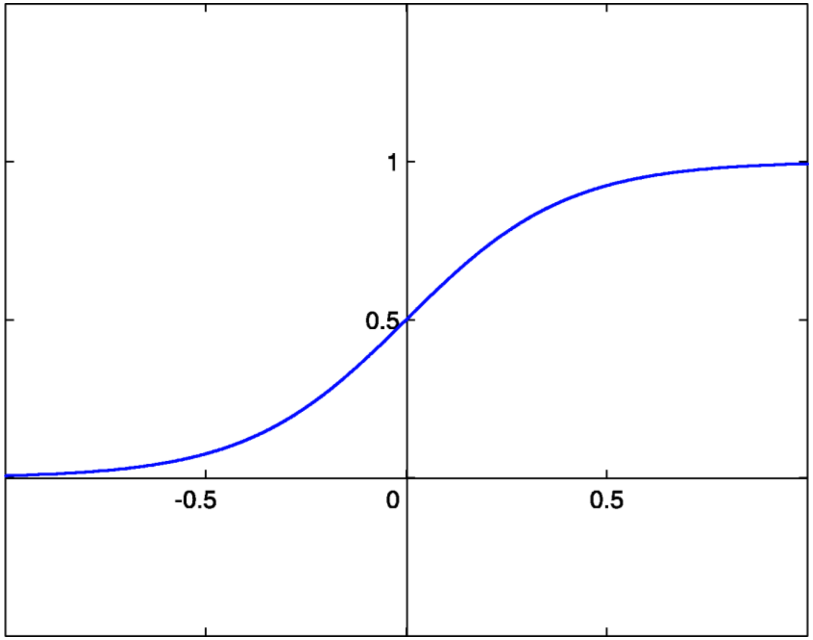
- Generalization
  - Adding noise
  - Training multiple models and use the average model of those
- Dropout
  - Drop a hidden unit with probability of 0.5
  - Maximal 2^h models ($2^{64} = 1.8\text{x}10^{19}$)
  - Approximation to geometric mean
  - Fast averaging technique (divide weights by 2)
- Maxout *(Goodfellow et al)*
  - Facilitate dropout's optimization
  - Improve accuracy of dropout's fast approximate model averaging technique

Idea of Maxout

- Traditional activation functions



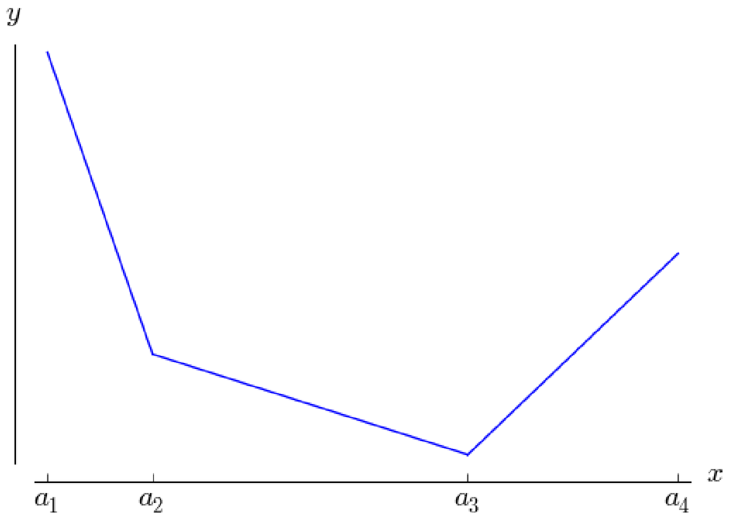
Threshold function



Sigmoid function

Idea of Maxout

- Do not use a fixed activation function
- But learn the activation function
- Piecewise Linear Function
  - Can approximate any continuous function *(Stone-Weierstrass)*
  - Linear almost everywhere, except k-1 points



Piecewise linear function

Idea of Maxout

- Maxout unit
  - k linear models
  - Output is the maximal value from k models from the given input x
- Formal:

$$h_i(x) = \max_{j \in [1,k]} z_{ij}$$

Where $z_{ij} = x^T W_{\ldots ij} + b_{ij}$

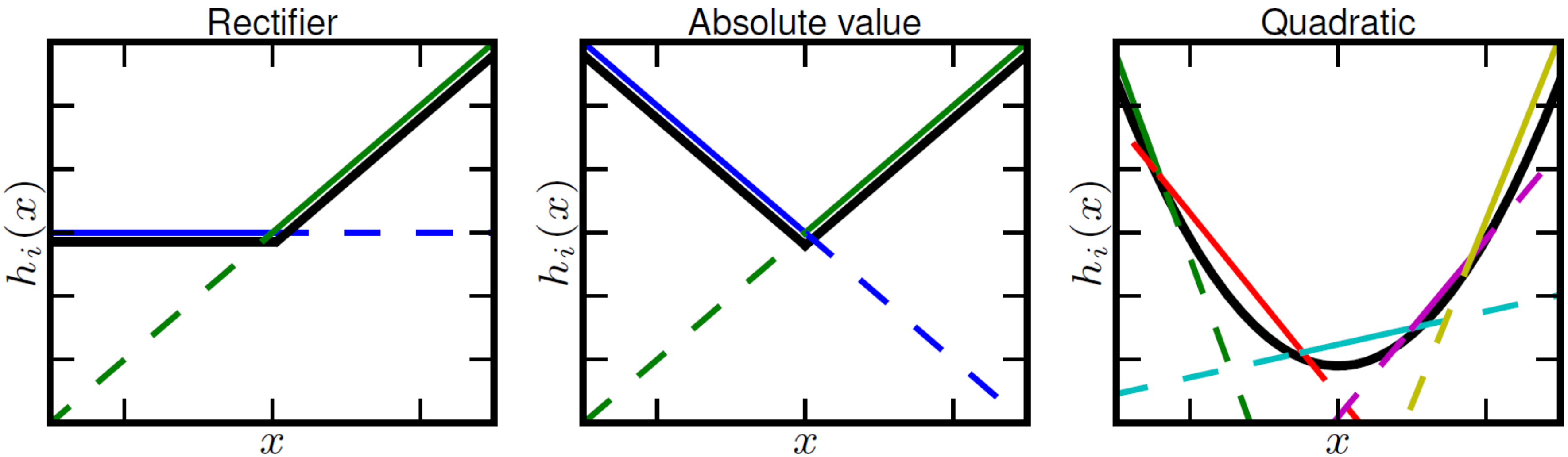
$W \in R^{d \times m \times k}$ and $b \in R^{m \times k}$

m: number of hidden units

d: size of input vector (x)

k: number of linear models

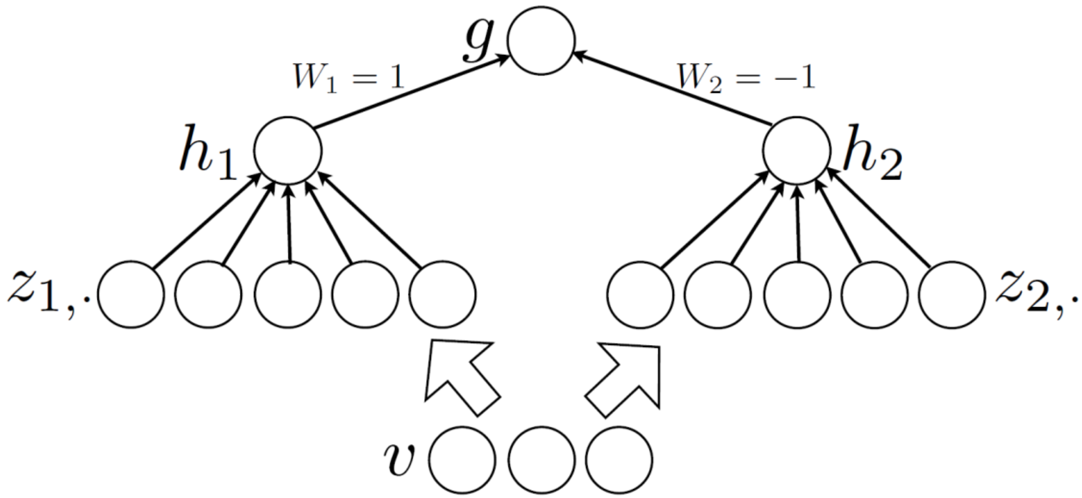
Idea of Maxout



k=2 k=2 k=5

Maxout : universal approximator

- Maxout networks with two hidden units:



Maxout : universal approximator

- Universal approximator theorem:

> *Any continuous function f can be approximated arbitrarily well on a compact domain $C \subset \mathbb{R}^n$ by a maxout network with two maxout hidden units.*

- Proof
  - (Wang, 2004) Any continuous function can be expressed as a difference of 2 convex functions

$$g(x) = h_1(x) - h_2(x) \qquad (1)$$

  - *(Stone-Weierstrass) Any continuous function can be approximated by a piecewise linear function*

$$| f(x) - g(x) | < \mathcal{E} \qquad (2)$$

Experiment on benchmark datasets

Name	Classes	Training	Test	Image	Color
MNIST	10	60 000	10 000	28x28	Grayscale
CIFAR-10	10	50 000	10 000	32x32	Color
CIFAR-100	100	50 000	10 000	32x32	Color
SVHN	10	73 257	26 032	32x32	Color

- SVHN dataset also consists of 521,131 additional samples

MNIST

- *Permutation invariant* MNIST

- Maxout multilayer perceptron (MLP):
  - Two *maxout layers* followed by a *softmax layer*
  - Dropout
  - Training/Validation/Test : 50,000/10,000/10,000 samples

- Error rate: 0.94%

- This is the best result without pre-training

MNIST

- Without permutation invariant restriction

- Best model consists of:
  - 3 convolutional maxout hidden layers with spatial max pooling
  - Followed by a softmax layer

- Error rate is 0.45%

- There are better results by augmenting standard dataset

CIFAR-10

- Preprocessing
  - Global constrast normalization
  - ZCA whitening
- Best model consists of
  - 3 convolutional maxout layers
  - A fully connected maxout layer
  - A fully connected softmax layer
- Error rate
  - Without data augmentation 11.68 %
  - With data augmentation 9.35 %

CIFAR-100

TECHNISCHE UNIVERSITÄT DARMSTADT

- Use the same hyperparameters as in CIFAR-10

- Error rates
  - Without retraining using entire training set : 41.48 %
  - With retraining : 38.57 %

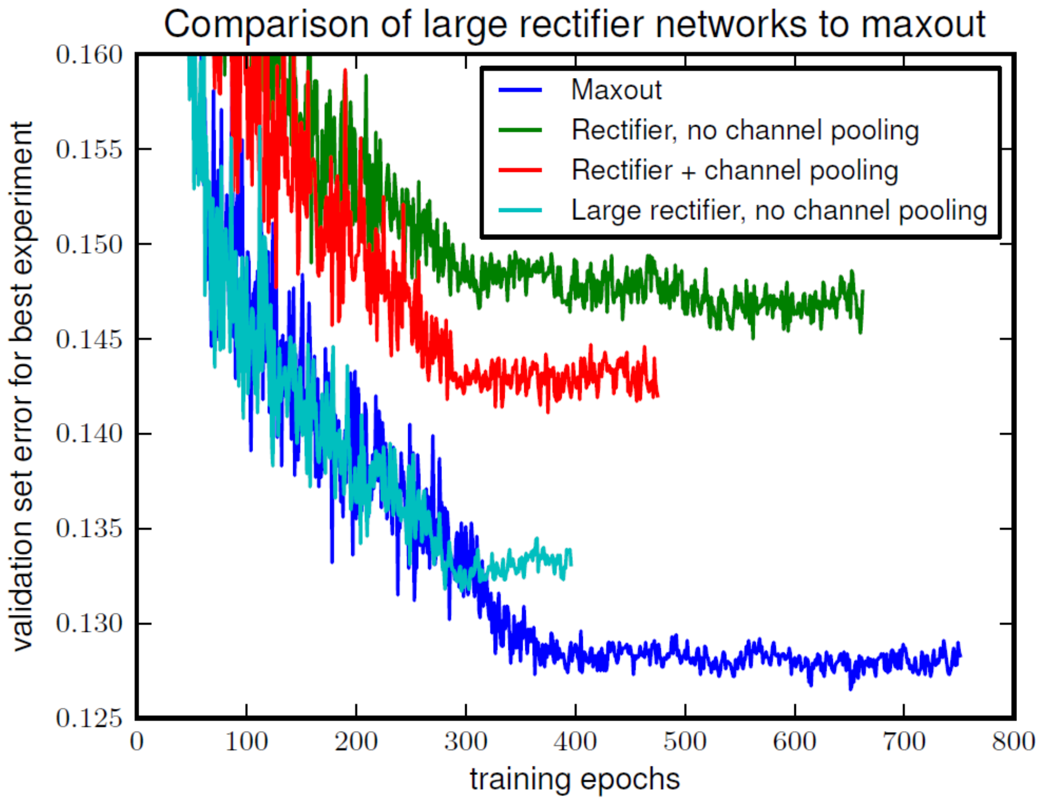
SVHN

- Local contrast normalization preprocessing
- 3 convolutional maxout hidden layers
- 1 maxout layer
- Followed by a softmax layer

- Error rate is 2.47%


Local contrast normalization
(Zeiler&Fergus 2013)

Comparison to rectifiers



What does Maxout work?

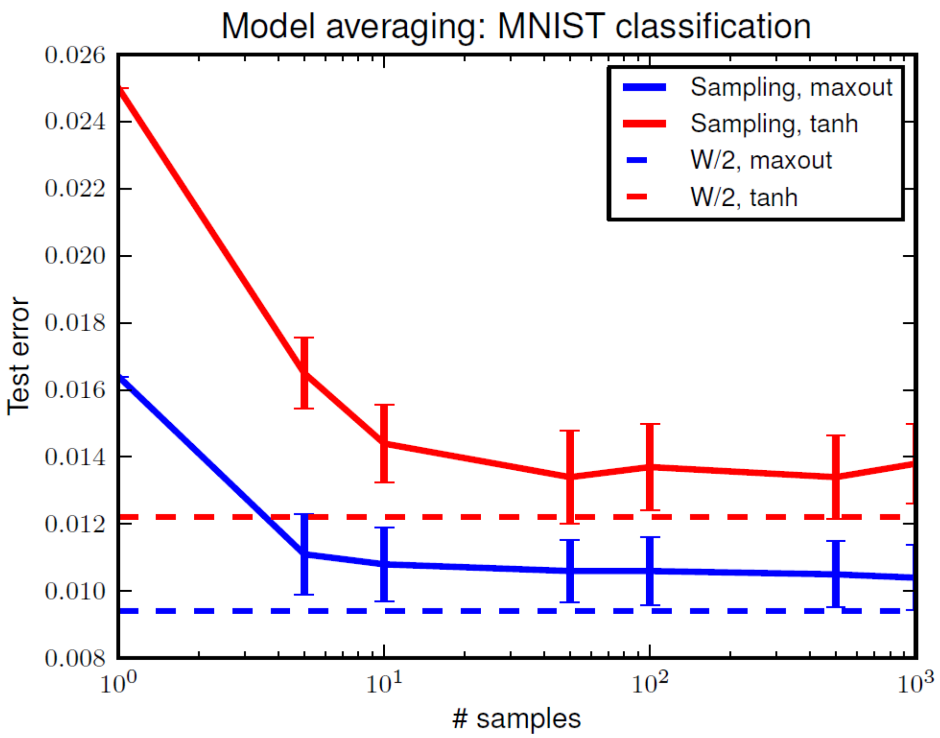
- Enhance accuracy of dropout model averaging technique
- Maxout using with dropout improves optimization
- Maxout improves bagging training style on deeper layer

Model Averaging

- Dropout performs model averaging

- Comparing of geometric mean of sample's subsets and full model of dropout with half of the weight W

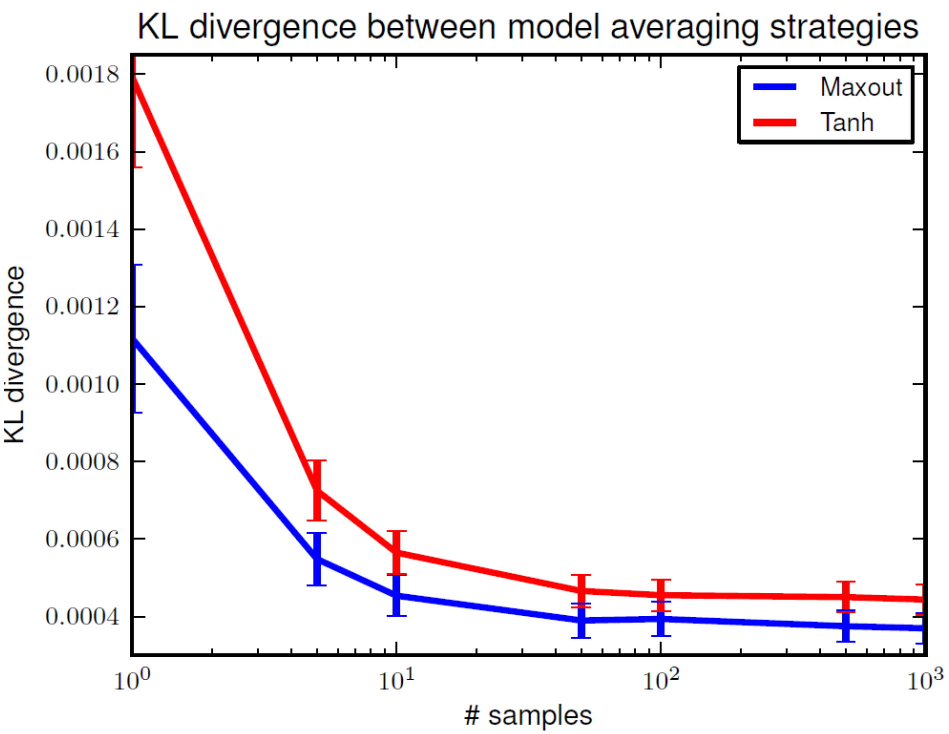
- Maxout improves accuracy of dropout



Model Averaging

- Kullback-Leibler divergence between geometric mean of sample's subset and dropout averaged model

- The approximation is more accurate for maxout units

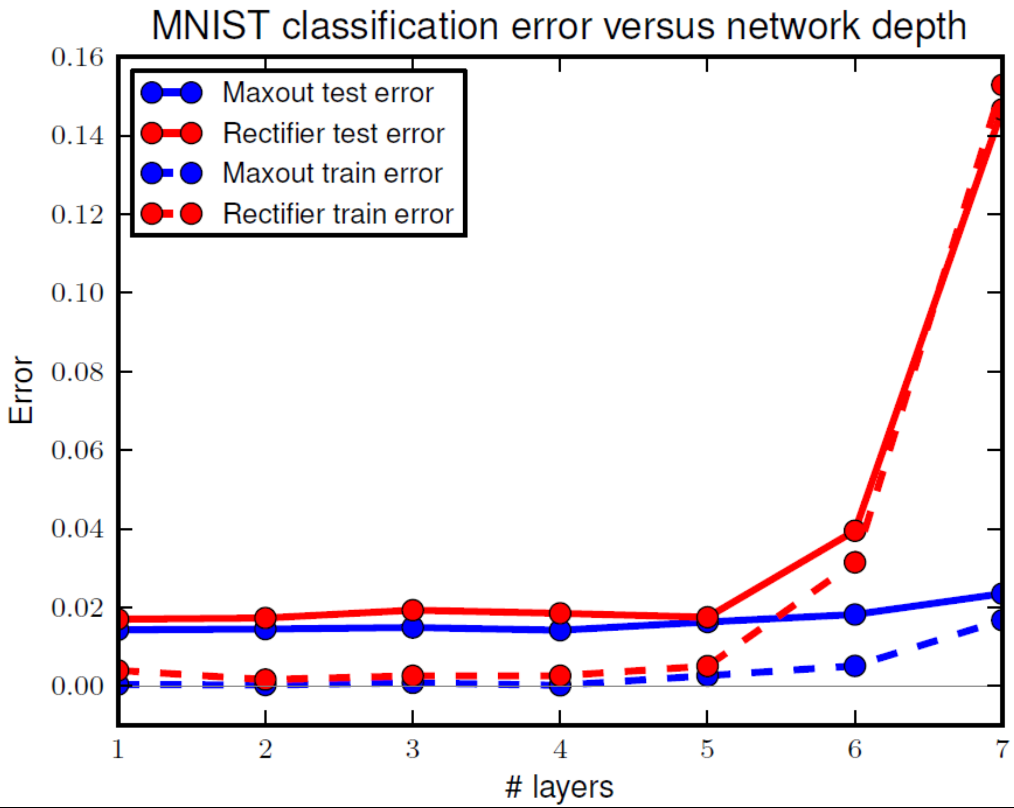


Optimization

- Maxout works better than max pooled rectified linear units
  - Small model on large dataset
    - 2 convolutional layers
    - Training with big SVHN dataset (600,000 samples)
  - Error rate
    - Maxout error : 5.1%
    - Rectifier error : 7.3%

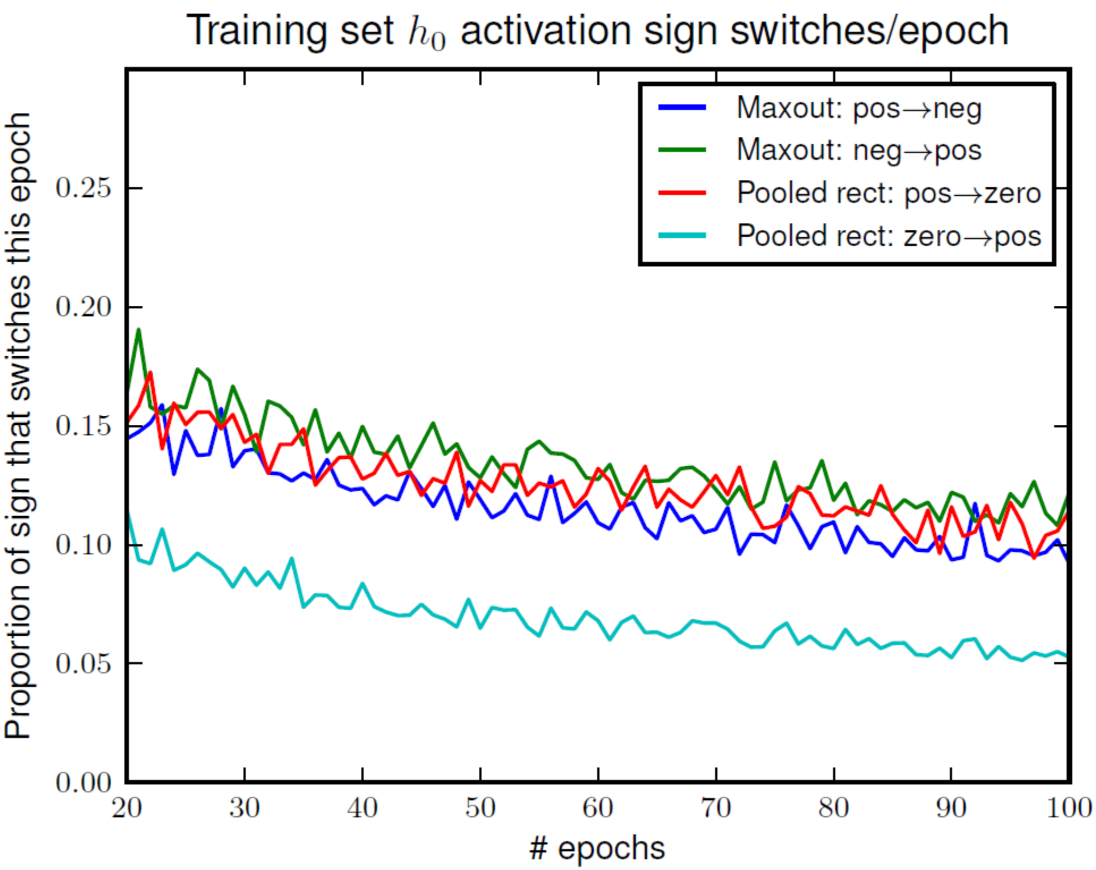
Optimization

- Maxout works better than max pooled rectified linear units
  - Comparison on network depth



Saturation

- Maxout:
  - Rate of sign switches is equals
  - >99.99 % filters used
- Rectifier:
  - *"death rate"* is bigger than *"birth rate"*
  - 40% filters are unused



Conclusion

- A new activation function which is suited with dropout
- Proof of a universal approximator with 2 maxout hidden units
- Maxout model benefits more from dropout than other activation functions
- Set new state of the art on 4 benchmark datasets

References

- Goodfellow et al., Maxout Networks, Proceedings of International Conference on Machine Learning (ICML), 2013
- Hinton, Geoffrey E., Srivastava, Nitish, Krizhevsky, Alex, Sutskever, Ilya, and Salakhutdinov, Ruslan. Improving neural networks by preventing co-adaptation of feature detectors. Technical report, arXiv:1207.0580, 2012.
- Zeiler, Matthew D. and Fergus, Rob. Stochastic pooling for regularization of deep convolutional neural networks. In ICLR 2013