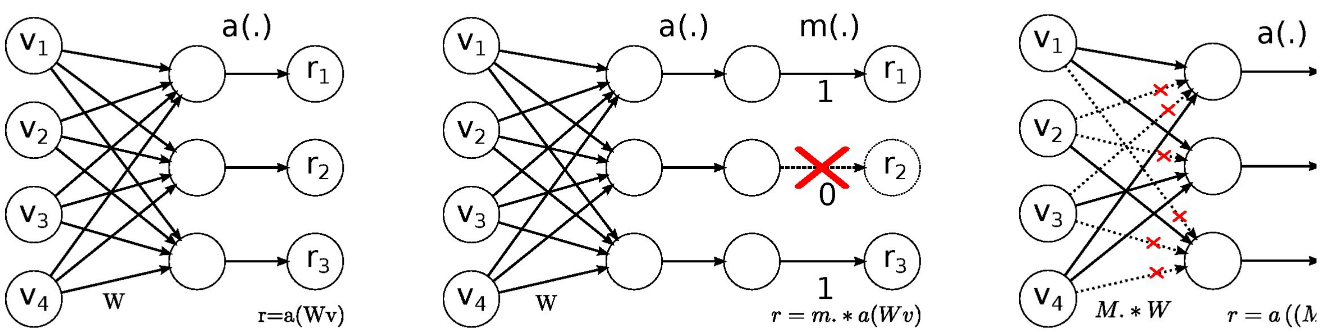
Regularization of Neural Networks using DropConnect

Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus

Dept. of Computer Science, Courant Institute of Mathematical Science, New York University

Introduction

We introduce DropConnect, a generalization of Hinton's Dropout for regularizing large fully-connected layers within neural networks. When training with Dropout, a randomly selected subset of activations are set to zero within each layer. DropConnect instead sets a randomly selected subset of *weights* within the network to zero. Each unit thus receives input from a random subset of units in the previous layer. We derive a bound on the generalization performance of both Dropout and DropConnect.



No-Drop Network

DropOut Network

DropConnect Networ

Motivation

Training Network with Dropout:
Each element of a layer's output is kept with probability p, otherwise being set to 0 with probability $1-p$. If we further assume neural activation function with $a(0)=0$, such as $tanh$ and $relu$ ($\star$ is element-wise multiplication):

$$r = m \star a\left(Wv\right) = a\left(m \star Wv\right)$$

Training Network with DropConnect:
Generalization of Dropout in which each connection, rather than each output unit, can be dropped with probability $1-p$:

$$r = a\left(\left(M \star W\right)v\right)$$

where M is weight mask, W is fully-connected layer weights and v is fully-connected layer inputs.

Mixture Model Interpretation

DropConnect Network is a mixture model of $2^{|M|}$ neural network classifiers $f(x;\theta,M)$:

$$o = \mathbf{E}_M\left[f\left(x;\theta,M\right)\right] = \sum_M p\left(M\right) f\left(x;\theta,M\right)$$

It is not hard to show stochastic gradient descent with random mask M for each data improves the lower bound of mixture model

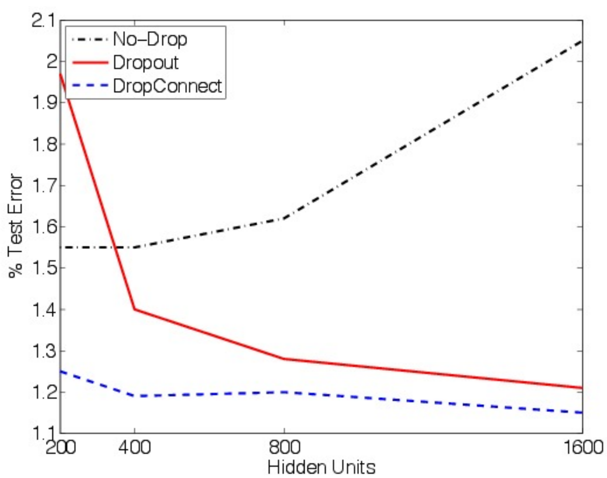
---

Inference

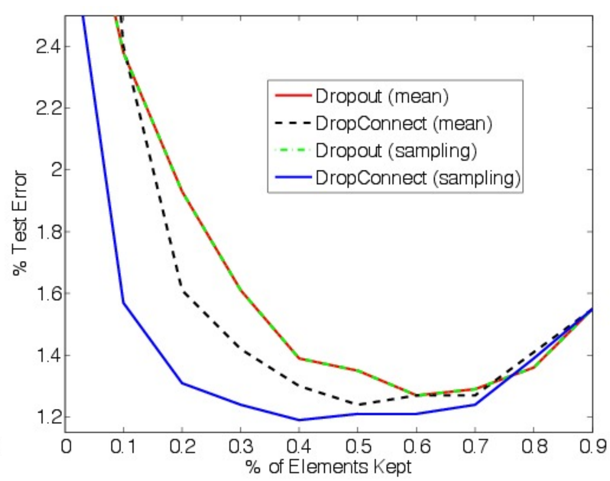
Dropout Network Inference (mean-inference):	$\mathbf{E}_M\left[a\left(M \star W\right)v\right] \approx a\left(\mathbf{E}_M\left[\left(M \star W\right)v\right]\right) = a\left(pWv\right)$
DropConnect Network Inference (sampling):	$\mathbf{E}_M\left[a\left(M \star W\right)v\right] \approx \mathbf{E}_u\left[a(u)\right]$ where $u \sim \mathcal{N}\left(pWv, p\left(1-p\right)\left(W \star W\right)\left(v \star v\right)\right)$, i.e. each neuron activation are approximated by a Gaussian distribution via moment matching.

Experiment Results

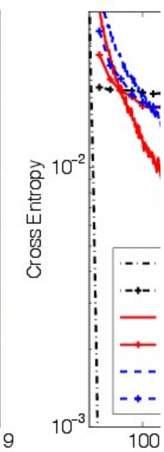
Experiment with MNIST dataset using 2-layer fully connected neural network:



(a)Prevent overfitting as the size of connected layers increase



(b)Varying the drop-rate in a 400-400 network



(c)Conve

Evaluate DropConnect model for regularizing deep neural network of various popular image classification datasets:

Image Classification Error(%) of DropConnect v.s. Dropout

DataSet	DropConnect	Dropout	Previous best result(2013)
MNIST	**0.21**	0.27	0.23
CIFAR-10	**9.32**	9.83	9.55
SVHN	**1.94**	1.96	2.80
NORB-full-2fold	3.23	**3.03**	3.36

Implementation Details

Performance comparison between different implementation of DropConnect layer on NVidia GTX 580 GPU relative to 2.67Ghz Intel Xeon (compiled with -O3 flag). Input and output dimension is 1024 and mini-batch size is 128 (You might not get exactly the same number with my code on your machine):

Efficient Implementation of DropConnect

Implementation	Mask Weight	Total Time(ms)	Speedup
CPU	float	3401.6	1.0 X
CPU	bit	1831.1	1.9 X
GPU	float(global memory)	35.0	97.2 X
GPU	float(tex1D memory)	27.2	126.0 X
GPU	bit(tex2D memory)	**8.2**	414.8 X

Total Time includes: fprop, bprop and update for each mini-batch

Thus, efficient implemention: 1) encode connection information in bits 2) Aligned 2D memory bind to 2D texture for fast query connection status. Texture memory cache hit rate of our implementation is close to 90%.

Why DropConnect Regularize Network

Rademacher Complexity of Model: $\max |W_s| \leq B_s, \max |W| \leq B$, k is the number of classes, $\hat{R}_\ell(\mathcal{G})$ is the Rademacher complexity of the feature extractor, n and d are the dimensionality of the input and output of the DropConnect layer respectively:

$$\hat{R}_\ell(\mathcal{F}) \leq p \left(2\sqrt{k} d B_s n \sqrt{d} B_h\right) \hat{R}_\ell(\mathcal{G})$$

Special Cases of p:

1. $p = 0$: the model complexity is zero, since the input has no influence on the output.
2. $p = 1$: it returns to the complexity of a standard model.
3. $p = 1/2$: all sub-models have equal preference.

Reference

Regularization of Neural Network using DropConnect Li Wan, Matthew Zeiler, Sixin Zhang, Yann LeCun, Rob Fergus
International Conference on Machine Learning 2013 (10 pages PDF) Supplementary Material Slides

---

CUDA code (code Sep-20-2013 update changelog)

Reproduce Experiment Results

The full project code is here in case you want to repeat some of the experiments in our paper. Please refer to here for how to compile the code. Some examples to run the code is here. Unfortunately, the code is a little bit unorganized and I might clean up in the future. Important trained models and config files are also available here(Updated Dec-16-2013).

Zygmunt from FastML has successfully reproduce experiment result on CIFAR-10 on Kaggle CIFAR-10 leadearbord in his artical Regularizing neural networks with dropout and with DropConnect.
A summary of question and my answer for hacking my uncleaned code is Here.