Dropout is a special case of the stochastic delta rule: faster and more accurate deep learning

Radek Bartyzal

Let's talk ML in Prague

28. 2. 2019

Dropout

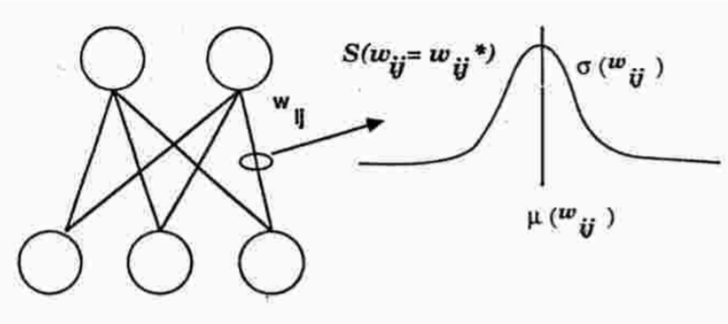
- add stochasticity = escape local minima
- removes hidden units according to a Bernoulli random variable with probability p prior to each update
- $\implies$ gradient updates affect non-removed neurons only
- $\implies$ exponential number of networks averaged over updates
- $\implies$ increase generalization by model averaging

Stochastic Delta Rule: Motivation

- neural transmissions involve noise
- neuron stimulated with same stimuli will never result in the same response
- smooth neural rate functions = averaging over many stimulation trials
- $\implies$ synapse between two neurons could be modeled with a distribution with fixed parameters

Stochastic Delta Rule: Idea

- each weight w_{ij} = random variable with mean $\mu_{w_{ij}}$ and standard deviation $\sigma_{w_{ij}}$
- we assume Gaussian but can be other distr.
- weight random variable is sampled on each forward activation



Stochastic Delta Rule: Details

- exponential number of potential networks with shared weights

- Both parameters are updated according to prediction error
- $\implies$ weight noise injections reflecting local history of prediction error $=$ bigger error $\implies$ bigger σ
- $\implies$ local model averaging
- model averaging may smooth out ravines in the error surface [Hinton]

- simulated annealing per weight
- each weight is updated based on its sampled contribution $=$ gradient is a random variable

Stochastic Delta Rule: Update rules

Forward pass samples weights w_{ij}^* from $N(\mu_{w_{ij}}, \sigma_{w_{ij}})$:

$$S(w_{ij} = w_{ij}^*) = \mu_{w_{ij}} + \mu_{w_{ij}} \theta(w_{ij}; 0, 1)$$

Classic gradient update to mean:

$$\mu_{w_{ij}}(n+1) = \alpha\left(\frac{\partial E}{\partial w_{ij}^*}\right) + \mu_{w_{ij}}(n)$$

Bigger error $\implies$ bigger σ = increase temperature:

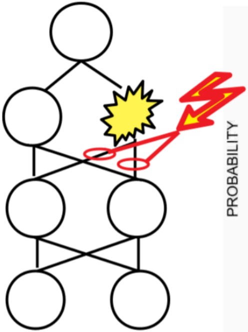
$$\sigma_{w_{ij}}(n+1) = \beta\left|\frac{\partial E}{\partial w_{ij}^*}\right| + \sigma_{w_{ij}}(n)$$

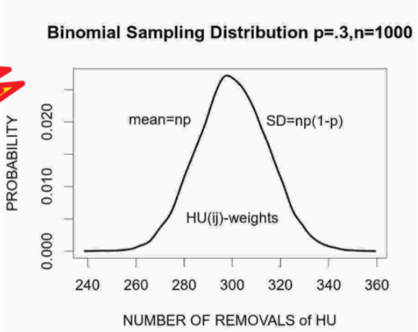
Exponentially lower σ = lower temperature = converge:

$$\sigma_{w_{ij}}(n+1) = \zeta \sigma_{w_{ij}}(n+1), \zeta < 1.$$

Dropout is Stochastic Delta Rule

- Bernoulli random variable over many trials results in a Binomial distribution with mean np and standard deviation $(np(1-p))$.
- The random variable is the number of removals over learning
- Dropout = hidden unit Binomial sampling





Experiments

- DenseNet-40, DenseNet-100, DenseNet-BC 250
- original parameters kept
- dropout $= 0.2$
- α/LR dropping at 50% and 75% of the run
- around $\alpha = 0.25$, $\beta = 0.05$, $\gamma = 0.7$
- annealed γ to reduce the influence of σ as the model converges
- σ updated twice every epoch, in the middle and at the end, for DenseNet-BC 250 and DenseNet-100 and after every batch for the others
- number of updates per epoch affects performance $=$ new hyperparameter
- earlier layers have $\gamma = 0.9 * \gamma$

Results

Table 1. Top-1 error validation rates at end of training of DenseNet-SDR compared to DenseNet with Dropout.

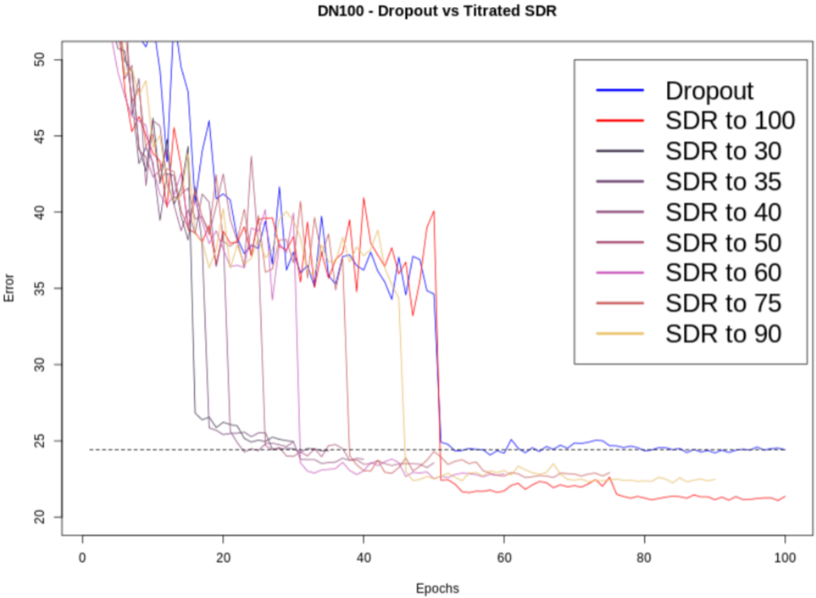
Model	Dataset CIFAR-10	Dataset CIFAR-100
DenseNet-40 (k=12)	6.88	27.88
DenseNet-100 (k=12)	-	24.67
DenseNet-BC 250 (k=12)	-	23.91
DenseNet-40 with SDR (k=12)	**5.91**	**24.58**
DenseNet-100 with SDR (k=12)	-	**21.72**
DenseNet-BC 250 with SDR (k=12)	-	**19.79**

Results

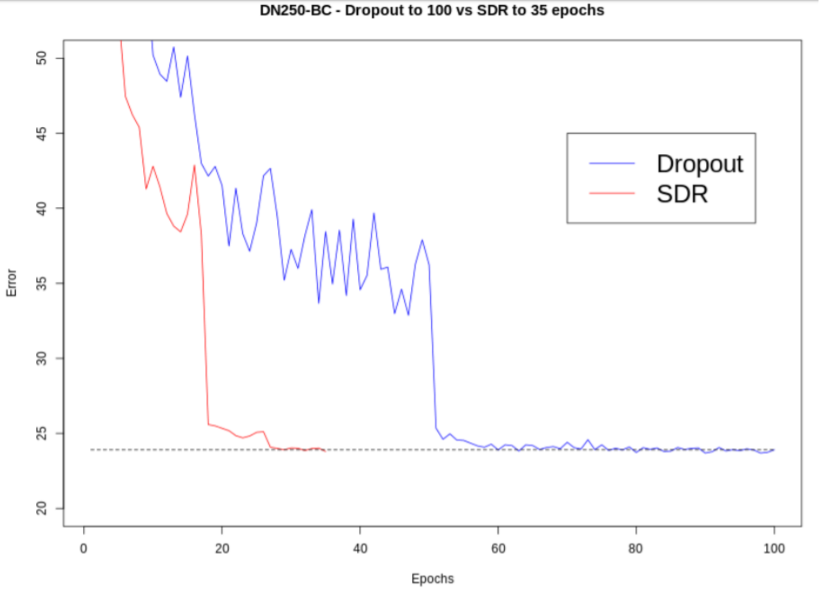
Table 2. Training losses of DenseNet-SDR compared to DenseNet with Dropout at end of training.

Model	Dataset: CIFAR-10	Dataset: CIFAR-100
DenseNet-40 (k=12)	1.85	10.01
DenseNet-100 (k=12)	-	1.17
DenseNet-BC 250 (k=12)	-	1.24
DenseNet-40 with SDR (k=12)	**0.24**	**0.89**
DenseNet-100 with SDR (k=12)	-	**0.15**
DenseNet-BC 250 with SDR (k=12)	-	**0.11**

Results


DN100 - Dropout vs Titrated SDR
Error
Epochs
50
45
40
35
30
25
20
0
20
40
60
80
100
Dropout
SDR to 100
SDR to 30
SDR to 35
SDR to 40
SDR to 50
SDR to 60
SDR to 75
SDR to 90

Results



DN250-BC - Dropout to 100 vs SDR to 35 epochs
Dropout
SDR
Error
Epochs
0
20
40
60
80
100
20
25
30
35
40
45
50

Sources

1. Frazier-Logue, Noah, and Stephen José Hanson. "Dropout is a special case of the stochastic delta rule: faster and more accurate deep learning." arXiv preprint arXiv:1808.03578 (2018).
https://arxiv.org/pdf/1808.03578v2.pdf
Code: https://github.com/noahfl/sdr-densenet-pytorch