M. Babaeizadeh et al., *NoiseOut: A Simple Way to Prune Neural Networks*, 29th Conference on Neural Information Processing Systems (NIPS 2016)

Alexander Astrup, Thomas Halberg og Sidsel Winther
Advanced Methods in Applied Statistics, The Niels Bohr Institute

March 6, 2018

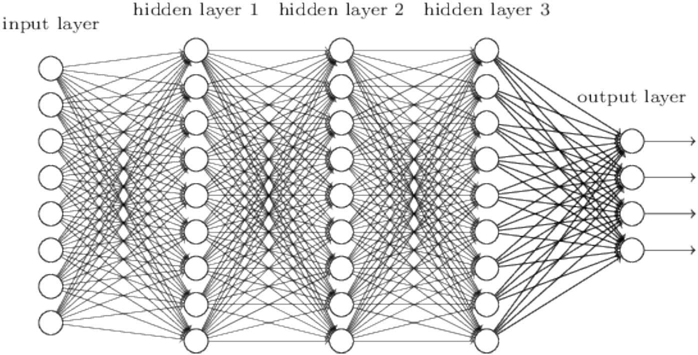
Overview

Introduction

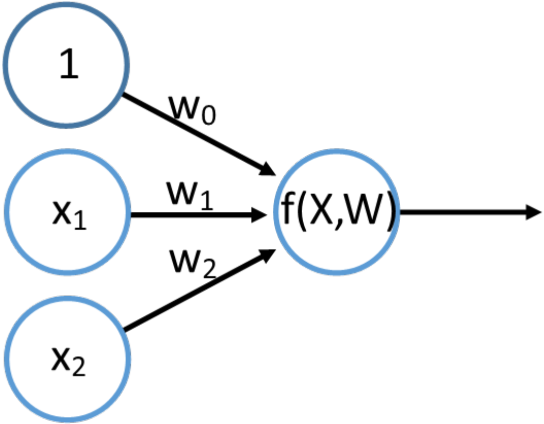
- Neural networks provide state-of-the-art results within various fields of research.
- But, often overparameterization results in an excessive usage of memory and computations.
- One solution to this issue is pruning.
- The technique presented here merges highly correlated neurons, while maintaining the accuracy obtained by the network.

- Fun fact: Pruning is naturally occurring in the brains of (especially) babies.

Basic introduction to artificial neural networks



Basic introduction to artificial neural networks



$$f(X, W) = \frac{1}{1+\exp(-(x_1 w_1 + x_2 w_2 + 1 \cdot w_0))}$$

Proposed method

NoiseOut:

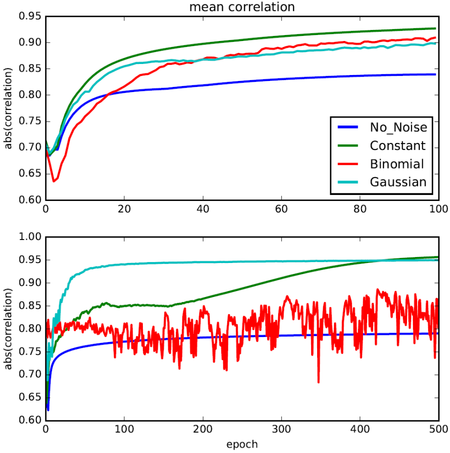
procedure Train(X,Y):
  W $\leftarrow$ initialize_weights()
  for each iteration **do**:
    Y_N $\leftarrow$ generate_random_noise()
    Y' $\leftarrow$ concatenate(Y, Y_N)
    W $\leftarrow$ back_prob(X,Y')
    while cost(W) $\leq$ threshold:
      A, B $\leftarrow$ find_most_correlated_neurons(W,X)
      α, β $\leftarrow$ estimate_parameters(W,X,A,B)
      W' $\leftarrow$ remove_neurons(W,A)
      W' $\leftarrow$ adjust_weights(W',B,α,β)
      W $\leftarrow$ W'
  return W

Pruning of a single neuron:

1. For each i,j,l calculate $\rho\left(h_i^{(l)}, h_j^{(l)}\right)$
2. Find $u, v, l = \arg\max \left|\rho\left(h_i^{(l)}, h_j^{(l)}\right)\right|$
3. Calculate $\alpha, \beta := \arg\min \left(h_u^{(l)} - \alpha h_v^{(l)} - \beta\right)$
4. Remove neuron u in layer l
5. For each neuron k in layer $l+1$:
   - Update the weight $w_{v,k}^{(l)} = w_{v,k}^{(l)} + \alpha w_{u,k}^{(l)}$
   - Update the bias $b_k^{(l+1)} = b_k^{(l+1)} + \beta w_u^{(l)}$

Encouraging correlations between neurons

- Correlations between the neurons is a key factor in pruning the network.
- Correlations are increased by adding *noise outputs*.
- Three noise models have been investigated: Binomial, Gaussian and Constant. These models were tested versus the non-noise case.
- The results only show correlation, not model improvement.



Experiments

Table 1: Pruning Lenet-300-100 on MNIST. In all of the experiments the error rate is 3.05%

Method	Noise Neurons	Layer 1 Neurons	Layer 2 Neurons	Parameters	Removed Parameters	Compression Rate
Ground Truth	-	300	100	266610	-	-
No_Noise	-	23	14	15989	94.00%	16.67
Gaussian	512	20	9	15927	94.02%	16.73
Constant	512	20	7	15105	94.33%	17.65
Binomial	512	19	6	11225	95.78%	23.75
No_Noise	-	13	12	10503	96.06%	20.89
Gaussian	1024	16	7	12759	95.21%	18.58
Constant	1024	18	7	14343	94.62%	17.61
Binomial	1024	19	7	15135	94.32%	25.38

Table 2: Pruning a convolutional network trained on SVHN dataset with 93.39% accuracy

Method	Dense Layer Neurons	Parameters	Removed Parameters
Ground Truth	1024	1236250	-
No_Noise	132	313030	74.67%
Gaussian	4	180550	85.39%
Constant	25	202285	83.63%
Bionomial	17	194005	84.30%

Table 3: Pruning Lenet-5 on MNIST. In all of the experiments the error rate is 0.95%

Dense Layer Neurons	Parameters	Removed Parameters
512	605546	-
313	374109	38.21%
3	13579	97.75%
33	48469	91.99%
26	40328	93.34%

Conclusion

- Introduction of noise outputs can increase the correlation between neurons in the hidden layers.
- Pruning according to this scheme can drastically decrease the number of neurons while maintaining accuracy for highly performing networks.
- Without decreasing accuracy for the MNIST dataset:
  - the number of parameters were decreased by 96% for the LeNet-300-100 network.
  - the number of parameters were decreased by 98% for the LeNet-5 network.