Pruning Neural Networks By Optimal Brain Damage

Chao Liu, CSLT

Nov.25, 2013

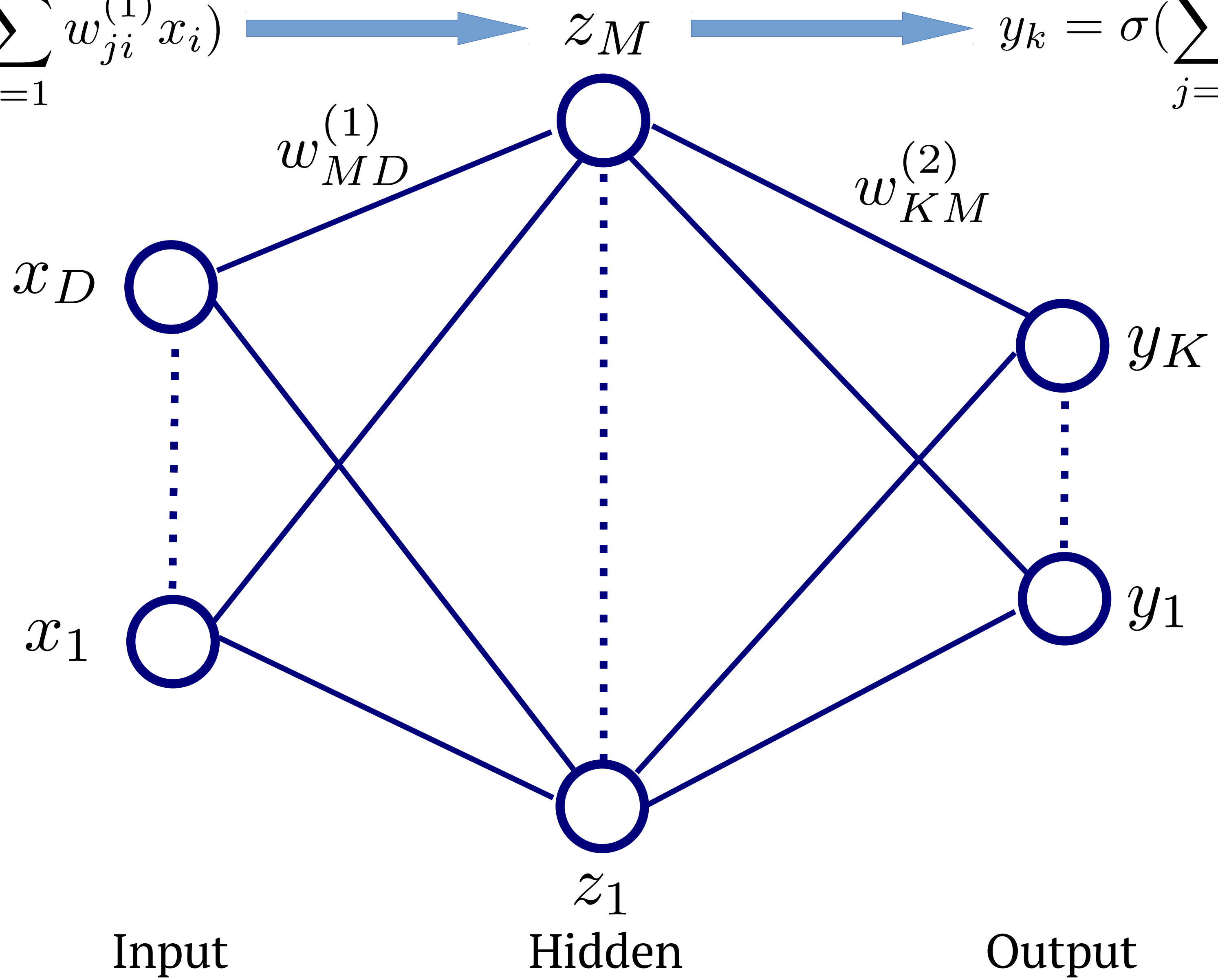
Outline

- Neural Network
- Network Training
- Network Pruning
- Results
- References

z_1

Feed-forward Network

$$z_j = h\left(\sum_{i=1}^{D} w_{ji}^{(1)} x_i\right) \qquad y_k = \sigma\left(\sum_{j=1}^{M} w_{kj}^{(2)} z_j\right)$$



Network Training

- Minimize a mean squared error function,

$$E(\mathbf{w}) = \frac{1}{2}\sum_{n=1}^{N}\|\mathbf{y}_n - \mathbf{t}_n\|^2$$

  or a cross-entropy error function

$$E(\mathbf{w}) = -\sum_{n=1}^{N}\{\mathbf{t}_n ln \mathbf{y}_n + (1-\mathbf{t}_n) ln(1-\mathbf{y}_n)\}$$

- Function is smooth continuous, so smallest value will occur at $\nabla E(\mathbf{w}) = 0$

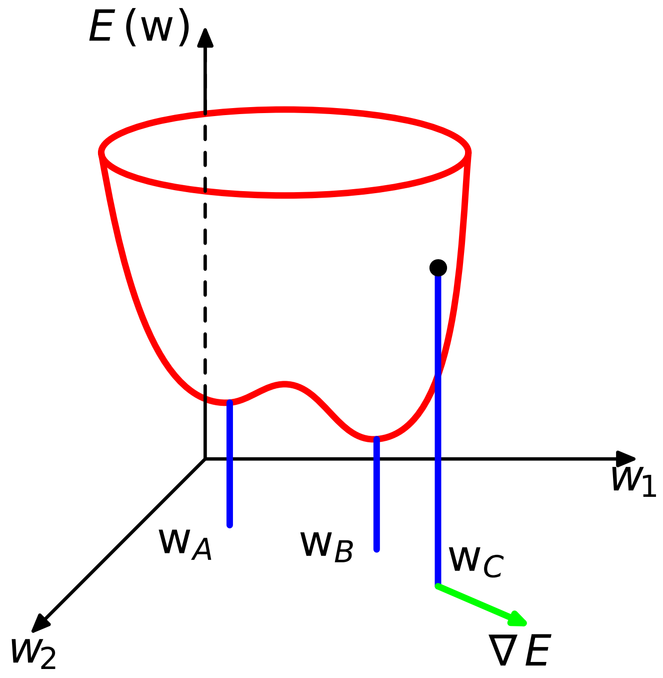
Gradient Descent

- Comprise a small step in the direction of the negative gradient

$$\mathbf{w}^{(\tau+1)} = \mathbf{w}^{(\tau)} - \eta \boxed{\nabla E(\mathbf{w}^{(\tau)})}$$

- Batch Gradient Descent or

  Stochastic Gradient Descent

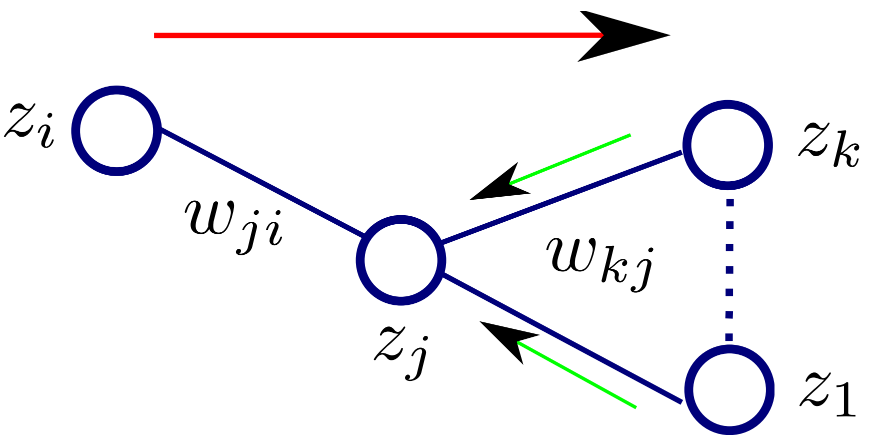


Error Back Propagation

- Apply an input vector, forward propagate through the network
- Evaluate $\frac{\partial E}{\partial a}$ for output units
- Back propagate $\frac{\partial E}{\partial a}$ for each hidden unit

$$\frac{\partial E}{\partial a_j} = h'(a_j) \sum_k w_{kj} \frac{\partial E}{\partial a_k}$$

$$\frac{\partial E}{\partial w_{ji}} = \frac{\partial E}{\partial a_j} z_i$$



Network Pruning

- Too many weights in modern networks, leads to high memory usage and low computing efficiency
- Identify the least significant weights in networks, and then cut them off
- A straightforward way: delete small-magnitude weights

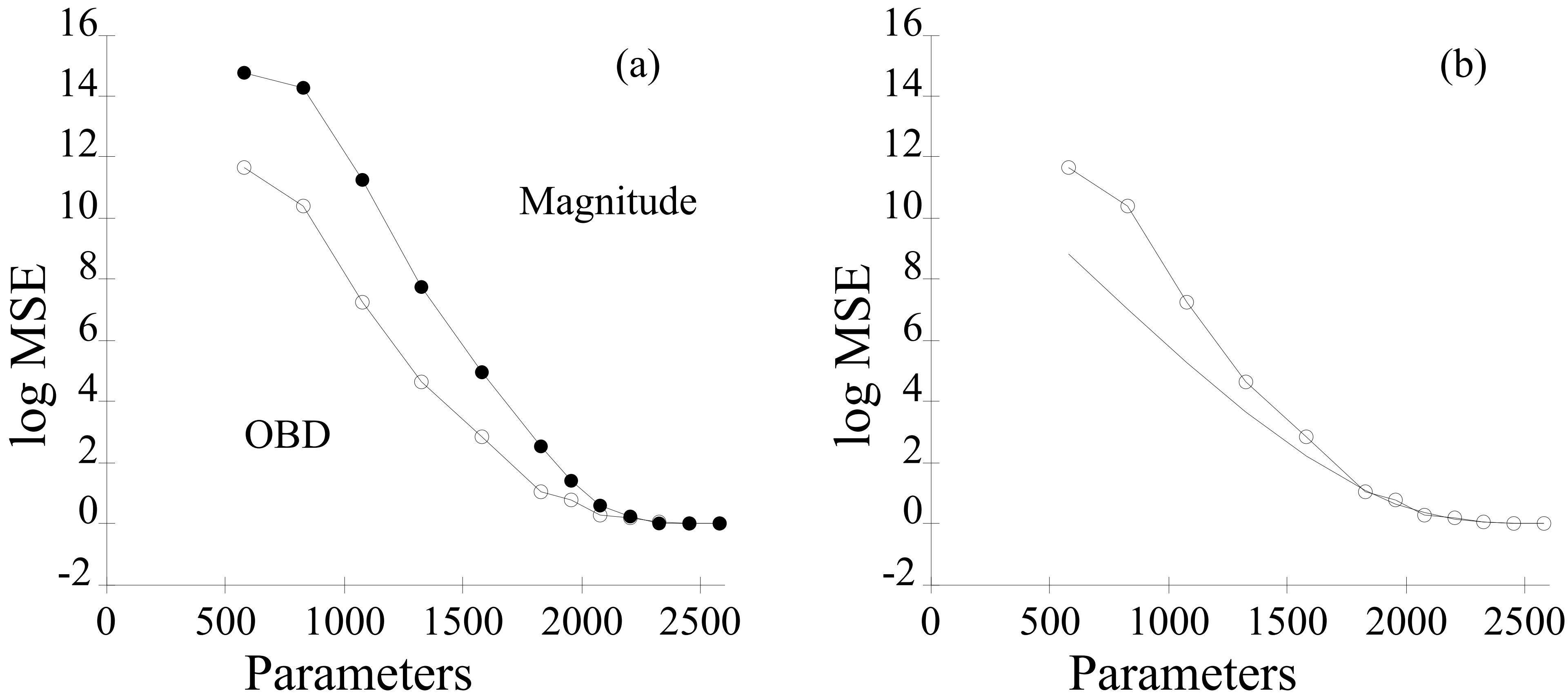
Optimal Brain Damage

- $\delta E = E(w + \delta w) - E(w) \approx \delta w \frac{\partial E}{\partial w} + \frac{1}{2} \delta w^T H \delta w$
- Evaluating full Hessian Matrix cost $O(W^2)$
- Diagonal approximation for Hessian cuts down the time complexity to $O(W)$
- Can be computed in a Back Propagation style

$$\boxed{\frac{\partial^2 E}{\partial a_j^2}} = h'(a_j)^2 \sum_k w_{kj}^2 \boxed{\frac{\partial^2 E}{\partial a_k^2}} + h''(a_j) \sum_k w_{kj} \boxed{\frac{\partial E}{\partial a_k}}$$

$$\frac{\partial^2 E}{\partial w_{ji}^2} = \frac{\partial^2 E}{\partial a_j^2} z_i^2$$

Results



Mean squared error ratio for Magnitude, OBD and predicted OBD based pruning. (Le Cun)

Results

acoustic model	# of nz params	Model Size	Calc Time	Dev QER	Test QER
GMM MPE	1.5M	-	-	34.5	36.2
CD-DNN-HMM	19.2M	100%	100%	28.0	30.4
sparse: 67% nz	12.8M	101%	80%	27.9	30.3
sparse: 46% nz	8.8M	69%	55%	27.7	30.1
sparse: 31% nz	6.0M	47%	37%	27.7	30.1
sparse: 21% nz	4.0M	32%	25%	27.8	30.2
sparse: 12% nz	2.3M	18%	14%	27.9	30.4
sparse: 5% nz	1.0M	8%	6%	29.7	31.7

Model size, computation time, and percent query error rate (QER) with and without pruning by weight magnitude. (MSRA)

References

- Bishop, Christopher M., and Nasser M. Nasrabadi, "Pattern recognition and machine learning," vol. 1, New York: springer, 2006.
- Yann LeCun, John S Denker, Sara A Solla, Richard E Howard, and Lawrence D Jackel, "Optimal brain damage.," in Advances in Neural Information Processing Systems (NIPS), 1989, vol. 2, pp. 598–605.
- Dong Yu, Frank Seide, Gang Li, and Li Deng, "Exploiting sparseness in deep neural networks for large vocabulary speech recognition," in Acoustics, Speech and Signal Processing (ICASSP), 2012 IEEE International Conference on. IEEE, 2012, pp. 4409–4412.

Thanks!

- Q&A