Sparsity in the Neocortex, and its Implications for Machine Learning

Subutai Ahmad and Jeff Hawkins
sahmad@numenta.com, jhawkins@numenta.com

SUMMARY

1) Most deep learning systems rely on dense representations. This is in stark contrast to the neocortex, which relies on highly sparse representations.

2) The neocortex is sparse in at least two very different ways: a) the instantaneous activity of neurons is highly sparse, and b) the connectivity between neurons is also extremely sparse.

3) We show that deep learning can benefit significantly by moving to networks that are sparse in both activations and connections.

Contributions:

- Sparse representations are extremely robust, particularly when dimensionality is high.

- We train networks with sparse activations and weights, trained by back propagation.

- We demonstrate sparse networks are more robust than dense networks on speech and vision datasets.

- We show sparse networks can be extremely efficient, >50 times more efficient than dense networks.

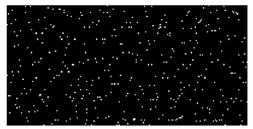
SPARSITY IN THE NEOCORTEX

Population sparsity

How many neurons are active right now?

Best estimates:
0.5% to 2% of cells are active at a time

(Willmore & Tolhurst, 2001; Attwell & Laughlin, 2001; Lennie, 2003; Graham & Field, 2007)

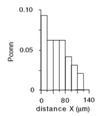


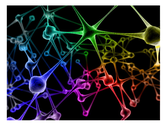
Connection sparsity

If a layer of cells projects to another layer, what percentage are connected?

Best estimates:
1% - 10% of possible neuron to neuron connections exist

(Holmgren et al., 2003; Lennie, 2003)





Deep learning systems are nothing like this.
Activations are far more dense (close to 50%).
Weight matrices are 100% dense.

Can deep learning networks benefit from sparsity?

SPARSE REPRESENTATIONS IMPROVE EXPONENTIALLY WITH DIMENSIONALITY

Sparse representations are highly "stable" and robust to perturbations and noise. The dot product between two vectors is the fundamental operation in neural networks. We can quantify the robustness by measuring probability of matches to random vectors.

Each dendritic segment has s synapses and represented by a binary vector D with n components and s "1" bits:

Activity in presynaptic region at time t represented by a binary vector A_t with n components and a_t active cells:

Probability of a random input matching a dendrite:

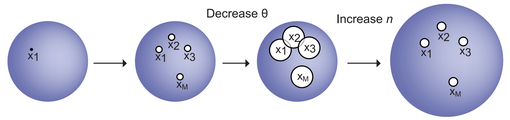
$$P(\mathbf{A}_t \cdot \mathbf{D} \geq \theta) = \frac{\sum_{b=\theta}^{s} |\ \Omega_{\mathbf{D}}(n, a_t, b)\ |}{\binom{n}{a_t}}$$

$|\Omega_{\mathbf{D}}(n,a,b)|$ counts the number of input vectors that exactly match *b* synapses on the dendrite

$$|\Omega_{\mathbf{D}}(n,a,b)| = \binom{s}{b} \times \binom{n-s}{a-b}$$

Number of ways to select exactly *b* out of *s* synapses

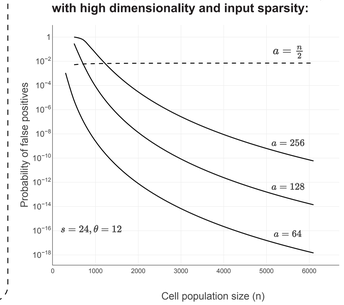
Number of vectors that have *a-b* bits on and no overlap with dendrite.



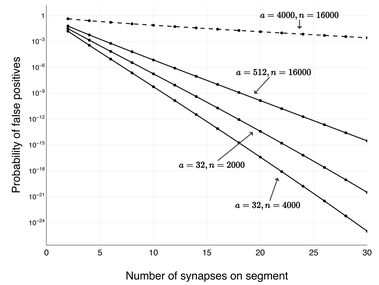
White circles represent all points whose dot products are within θ of some target point. As you decrease θ, the set of matching points increases, but there is an increasing risk of false matches (overlapping white circles).

For sparse representations, as you increase the dimensionality the space between white circles increases much faster than the size of the white circles, even though the number of non-zero components is unchanged. This means that the representations become far more robust to matches due to random noise.

The probability of error decreases dramatically with high dimensionality and input sparsity:

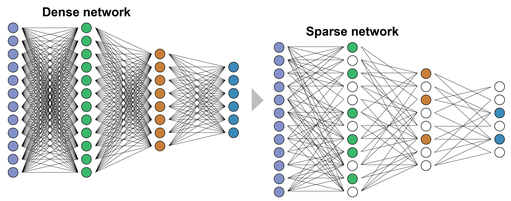


A tiny number of synapses, subsampling from a much larger pattern, is sufficient for robust recognition:

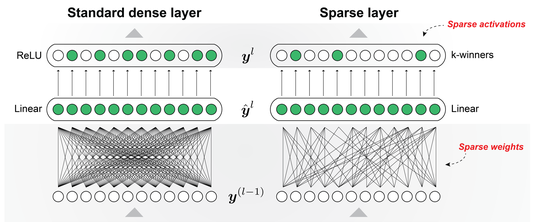


TRAINING SPARSE NETWORKS

Modeling biology, we create networks with both sparse activations and sparse connections



Each layer of the sparse network:
1) Contains sparse weights enforced by a binary mask
2) Replaces ReLU with a k-winner-take-all activation function
2) Incorporates a boosting function to maximize entropy and prevent dead units



The networks are trained using standard backpropagation.
For implementation details, please see: https://arxiv.org/abs/1903.11257

SPARSE NETWORKS ARE MORE ROBUST

Google Speech Commands Dataset (GSC)

Dataset of one word spoken commands with 65,000 utterances.
State of the art accuracy is between 95 - 97.5%

We trained dense and sparse convolutional networks, and tested their average accuracies under a wide range of noise values.

Network	Test Score	Noise Score	Params
Dense CNN	97.05± 0.20	31.08± 2.46	1.7M
Sparse CNN	97.03± 0.14	44.45± 2.54	160,952
Dense Small1	96.14± 0.73	26.57± 2.39	536,008
Dense Small2	95.89± 0.51	26.29± 3.11	270,376

Sparse networks had a significantly better noise score, even with 10% as many weights. Small dense networks did worse, showing the benefits of sparsity and high dimensionality.

CIFAR-10

Dataset of labeled color images with 10 total categories.

Noise	DenseNet	NotSoDenseNet	VGG19-Dense	VGG19-Sparse
0.0%	92.80	93.09	93.24	92.10
2.5%	86.34	87.50	85.07	86.21
5.0%	77.19	79.10	75.88	79.00
7.5%	66.22	69.52	63.60	71.34
10.0%	55.10	61.13	52.41	64.18
12.5%	45.79	52.10	42.25	56.49
15.0%	38.67	45.25	35.25	50.86
17.5%	33.03	39.60	29.37	45.00

Sparse networks again performed significantly better under noise.

SPARSE NETS ARE FAR MORE EFFICIENT

FPGA implementations show that sparse networks can be >50X faster

FPGA (Field Programmable Gate Arrays) platforms are ideal for sparse computations.
We implemented our sparse GSC network on three different Xilinx FPGA platforms.

Alveo U250 | Zynq™ UltraScale+ ZCU104 | Zynq™ UltraScale+ ZU3EG

Overall throughput is more than 50X higher for sparse networks

Name of chip	Network type	Throughput for single network	Speedup over dense	Number of networks on chip	Full chip throughput	Full chip speedup
Alveo U250	Dense	3,049	-	4	12,195	-
Alveo U250	Sparse	31,250	10.25	20	625,000	51.25
ZCU104	Dense	6,410	-	1	6,410	-
ZCU104	Sparse	26,667	4.16	3	80,000	12.48
ZU3EG	Dense	0	-	0	0	-
ZU3EG	Sparse	21,053	Infinite	1	21,053	Infinite

Each network is >10X faster

Dense network does not even fit on the small chip

Overall >50X throughput

>25X improved energy efficiency

Name of chip	Network type	System power	Words / Watt	Relative efficiency (compared to best dense network)
Alveo U250	Dense	225	54	0.507
Alveo U250	Sparse	225	2,778	26.00
ZCU104	Dense	60	107	1.0
ZCU105	Sparse	60	1,333	12.48
ZU3EG	Dense	24	0	-
ZU3EG	Sparse	24	877	8.211

>25X efficiency

>10X faster than NVIDIA Tesla V100

Platform	Network type	Batch size	Overall throughput
Alveo U250	Dense	500	12,195
Alveo U250	Sparse	N/A (streaming)	625,000
Tesla K80	Dense	256	16,024
Tesla K80	Dense	1024	17,710
Tesla K80	Dense	8192	20,118
Tesla V100	Dense	256	45,450
Tesla V100	Dense	1024	61,638
Tesla V100	Dense	8192	54,301

NETWORK DETAILS

Network	L1 Channels	L2 Channels	L3 N
GSC			
denseCNN2	64	64	1000
sparseCNN2	64	64	1000
denseSmall1	32	32	300
denseSmall2	32	64	300

Network	L1 Activation Sparsity	L2 Activation Sparsity	L3 Activation Sparsity
GSC			
denseCNN2	ReLU	ReLU	ReLU
sparseCNN2	90.5%	87.5%	90.0%
denseSmall1	ReLU	ReLU	ReLU
denseSmall2	ReLU	ReLU	ReLU

Network	L1 Weight Sparsity	L2 Weight Sparsity	L3 Weight Sparsity
GSC			
denseCNN2	0.0%	0.0%	0.0%
sparseCNN2	50.0%	80.0%	90.0%
denseSmall1	0.0%	0.0%	0.0%
denseSmall2	0.0%	0.0%	0.0%

Key network parameters for the GSC network used to test robustness. Networks used in the FPGA implementation were similar but larger and sparser.

References

J. Hawkins, S. Ahmad, Why Neurons Have Thousands of Synapses, a Theory of Sequence Memory in Neocortex, Front. Neural Circuits. 10 (2016) 1–13.

Cui, Y., Ahmad, S., & Hawkins, J. (2017). The HTM Spatial Pooler – a neocortical algorithm for online sparse distributed coding. Frontiers in Computational Neuroscience, 11, 111. https://doi.org/10.3389/FNCOM.2017.00111

Ahmad, S., & Scheinkman, L. (2019). How Can We Be So Dense? The Benefits of Using Highly Sparse Representations. ArXiv:1903.11257 [Cs.LG]. http://arxiv.org/abs/1903.11257