BitBrain and *Sparse Binary Coincidence* (SBC) memories

Presentation for NICE 2022

Michael Hopkins

The University of Manchester

31st March 2022

michael.hopkins@manchester.ac.uk

So what is the *BitBrain* idea all about?

An innovative working mechanism (the *SBC memory*) and surrounding infrastructure (*BitBrain*) based upon a novel synthesis of ideas from sparse coding, computational neuroscience and information theory.

- Single-pass supervised learning; currently follows an unsupervised phase where parameters are learned quickly in a simple, `local' and highly parallel way that avoids attempted global optimisation over very high-dimensional continuous spaces or calculation of derivatives.

- Accurate inference (classification for now) that is robust against imperfect inputs & uncertainty.

- Support for continuous adaptive learning – with or without `forgetting'.

- Designed to be implemented with low energy-latency product - both training and inference - on conventional CPU and memory architectures, and on current and future neuromorphic devices.

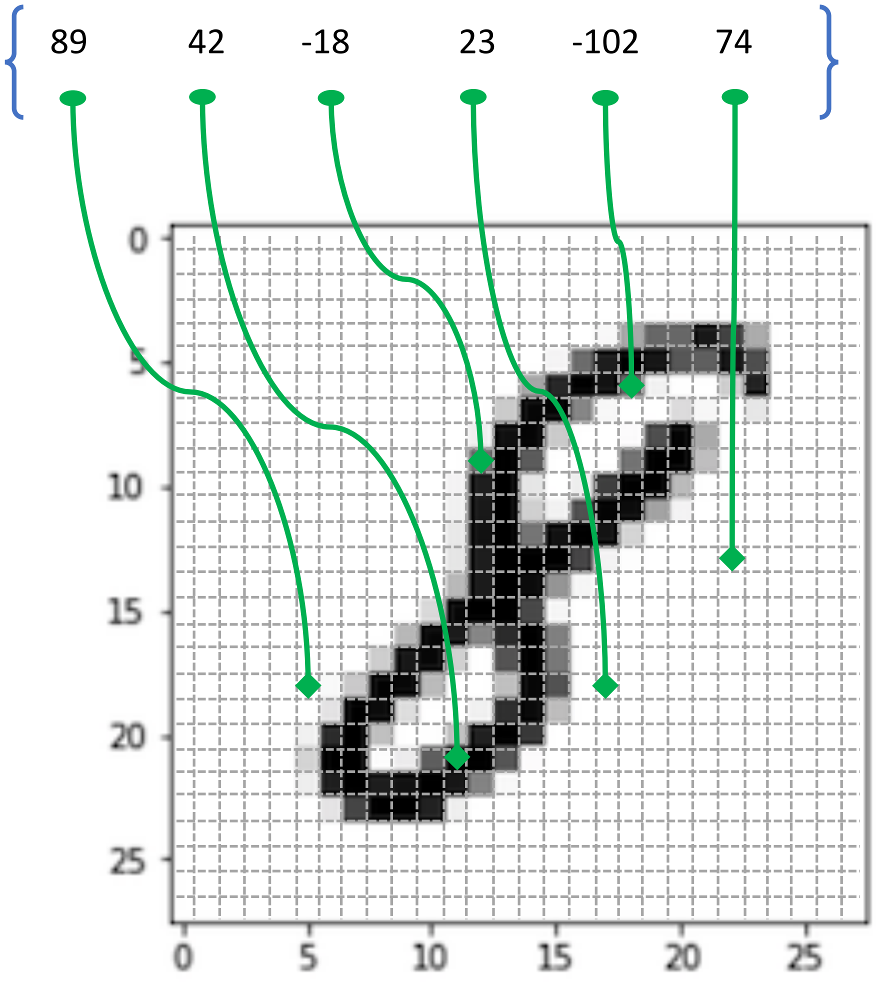
- A natural target for the increasing number of event-based sensors such as silicon retinas, enabling further energy and bandwidth gains to be exploited.

Patent GB 2113341.8 filed at the UKIPO on 17th Sept 2021 by myself and Steve Furber.

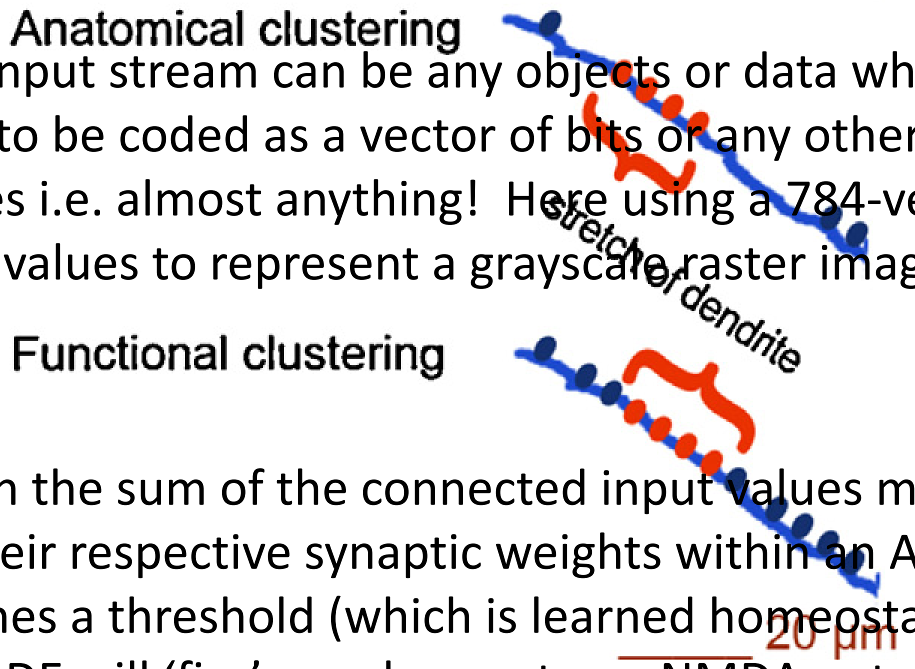
Address Decoder Elements (ADEs)

Each ADE samples a small subset of the input data e.g. the pixels of an MNIST digit.

An example ADE which contains multiple synapses with individual weights which can signify strength and/or longevity of connection.



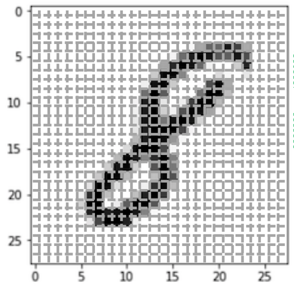
The input stream can be any objects or data which are able to be coded as a vector of bits or any other scalar values i.e. almost anything! Here using a 784-vector of 8-bit values to represent a grayscale raster image.

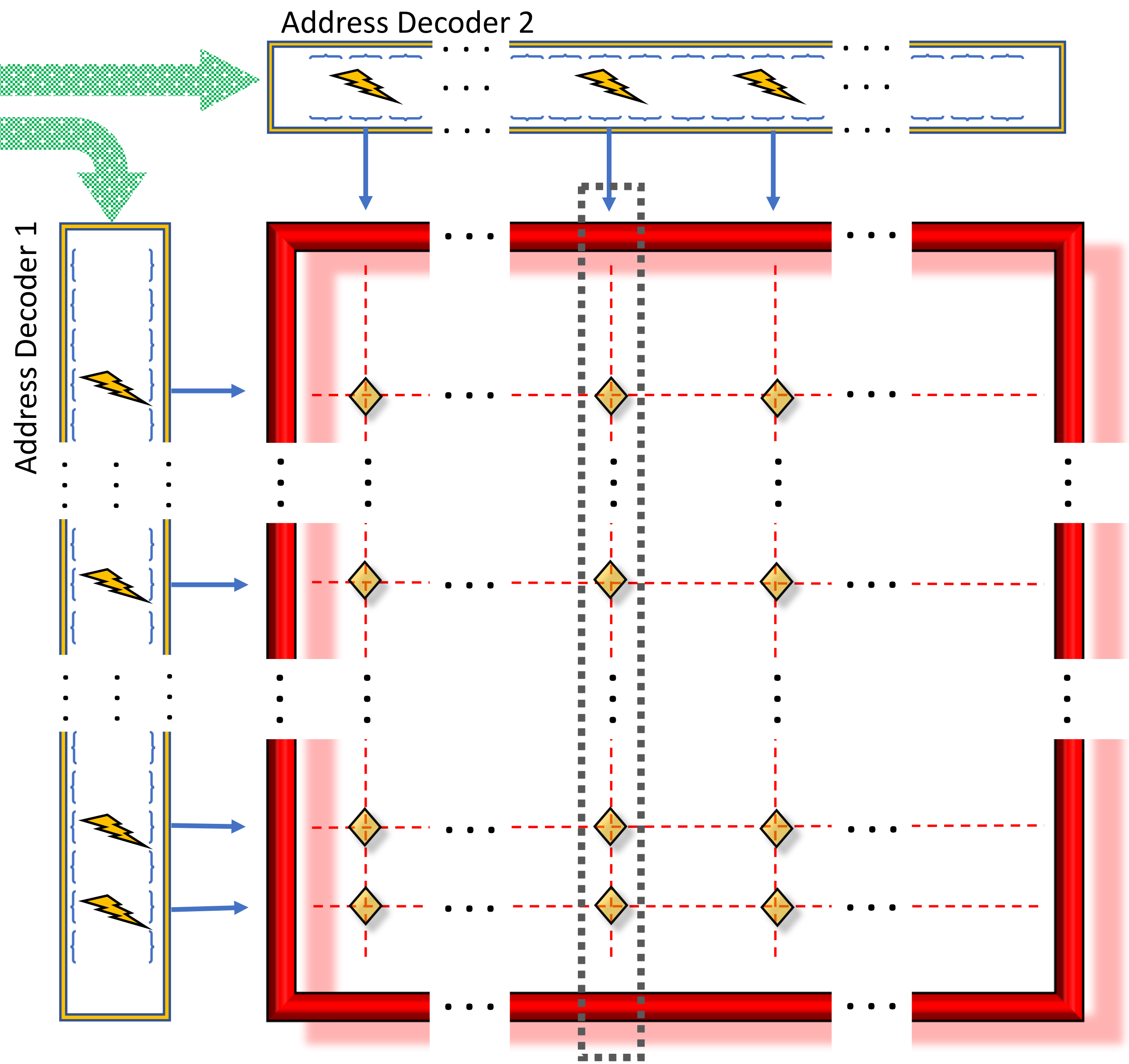


When the sum of the connected input values multiplied by their respective synaptic weights within an ADE reaches a threshold (which is learned homeostatically), the ADE will 'fire' - analogous to an NMDA potential from a synaptic cluster within a dendrite.

From *Synaptic clustering within dendrites: An emerging theory of memory formation.* Kastellakis et al (2015)

Address Decoders (ADs) accessing a 2D SBC memory





This is a 2D memory. 3D and higher (using more ADs) are also of interest.

Activation pattern is *sparse* i.e. only a small percentage of the ADEs in each AD will fire for any given input.

size of SBC match the length of ADs e.g. 1,024

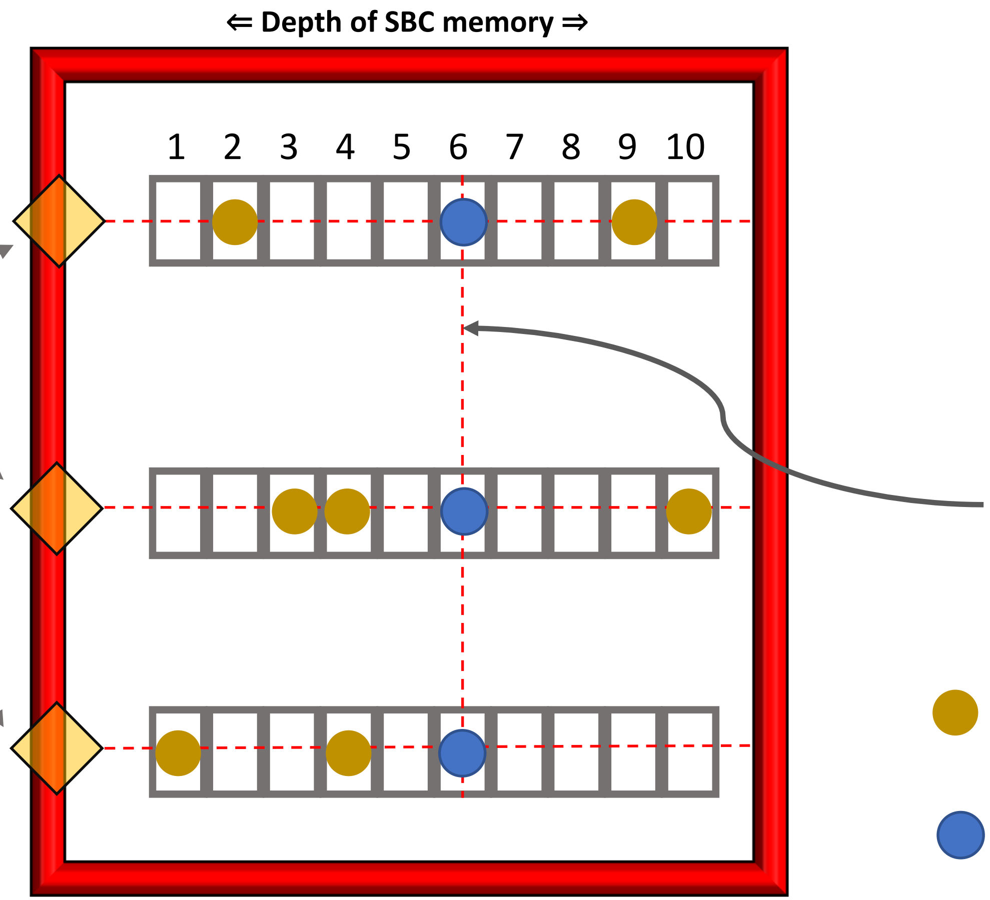
Each *coincidence* of active ADEs between ADs activates a memory location that reads or writes information about the class which has activated it.

depth = *f*(# classes)

◇ <- activated memory position

Class information held within SBC memory

Memory positions activated by coincident ADEs



‘Side view’ of SBC memory, showing ‘depth’ which varies with number of classes in problem. In this case, 10 classes with ‘one-hot’ coding meaning 10 bit cells per memory position.

3 activations are shown and in this case the input is from class 6.

<- Bits that were set previously

<- Bits set by this input

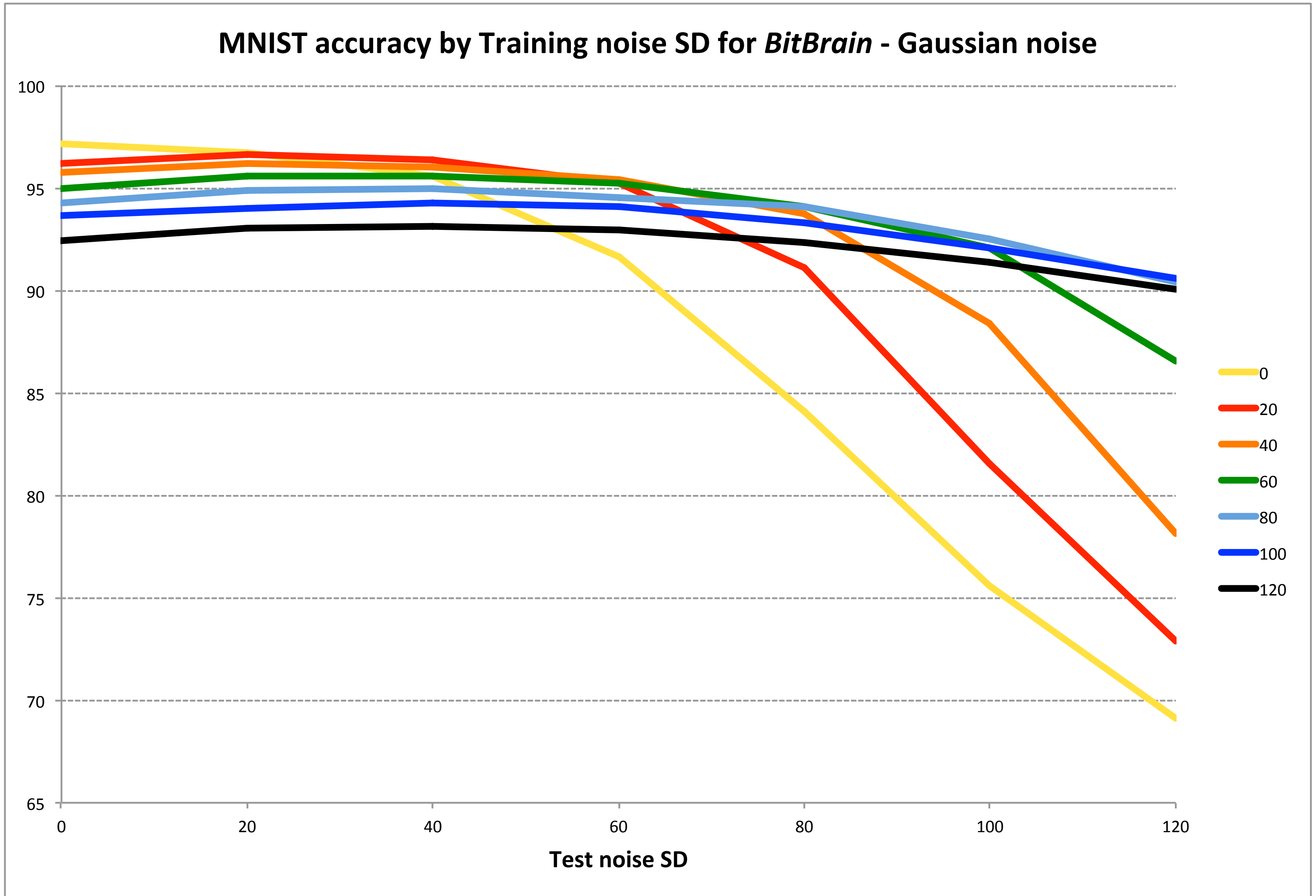
Writing to the SBC: go to all activated memory positions & set the relevant class bits if they are not already set.

Reading from the SBC: count bits set over all activated memory positions & choose class with the highest sum.

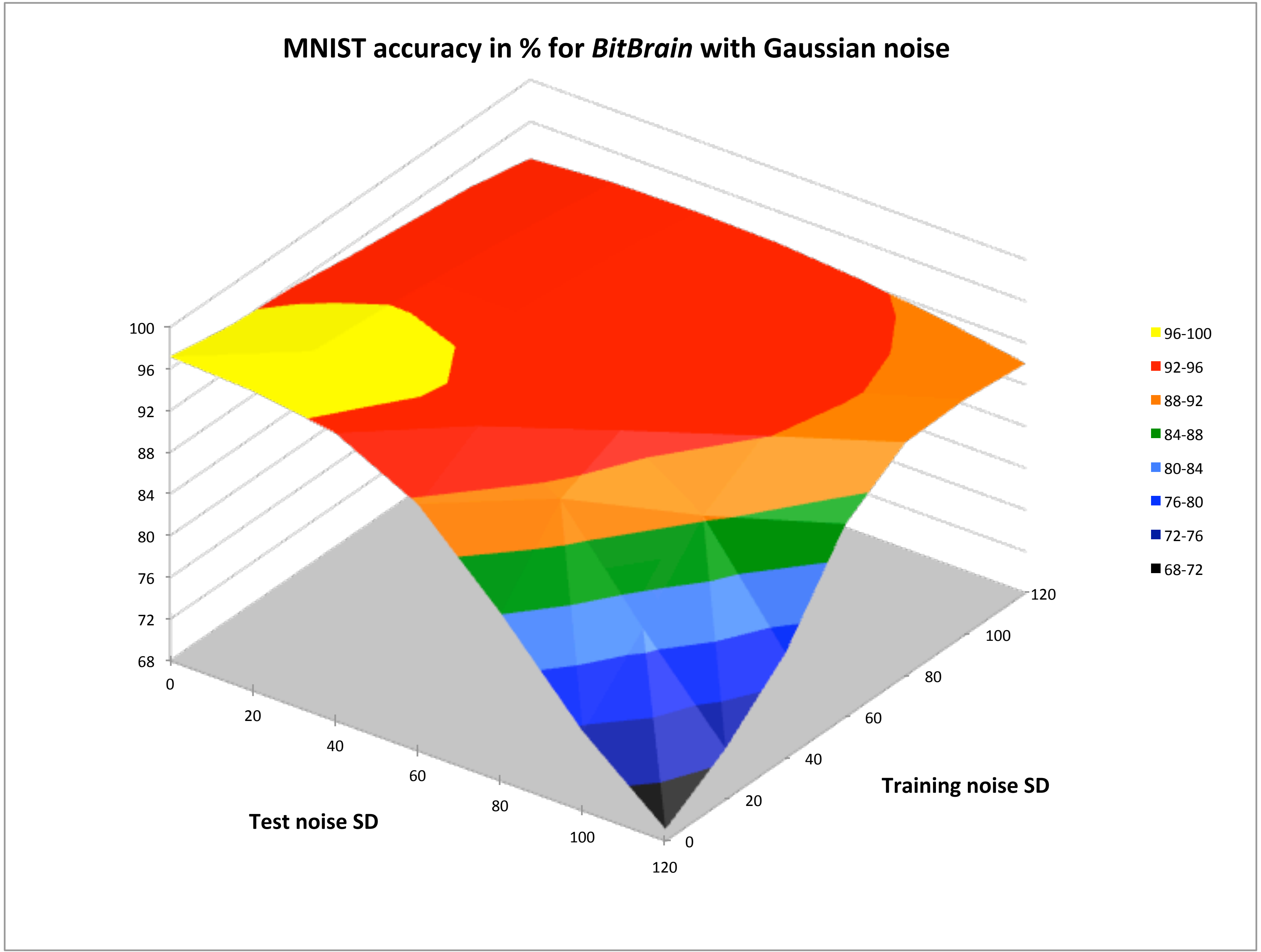
Assumes ‘one-hot’ encoding. If classes are coded differently then another encoding/decoding process required.

Basic results from MNIST (10 classes balanced)

AD lengths = 2,048. 4x ADs with { 6, 8, 10, 12 } synapse cluster size. AD target firing ≈1% per input.
Synapses spatially clustered and then within-cluster structural plasticity used to home in on features.
10x 2D SBCs; 6x full-size between ADs, 4x half-size within ADs.
42MB memory for full occupancy. Typically ≈15MB stored with opportunities for high levels of compression (≈1,000x).

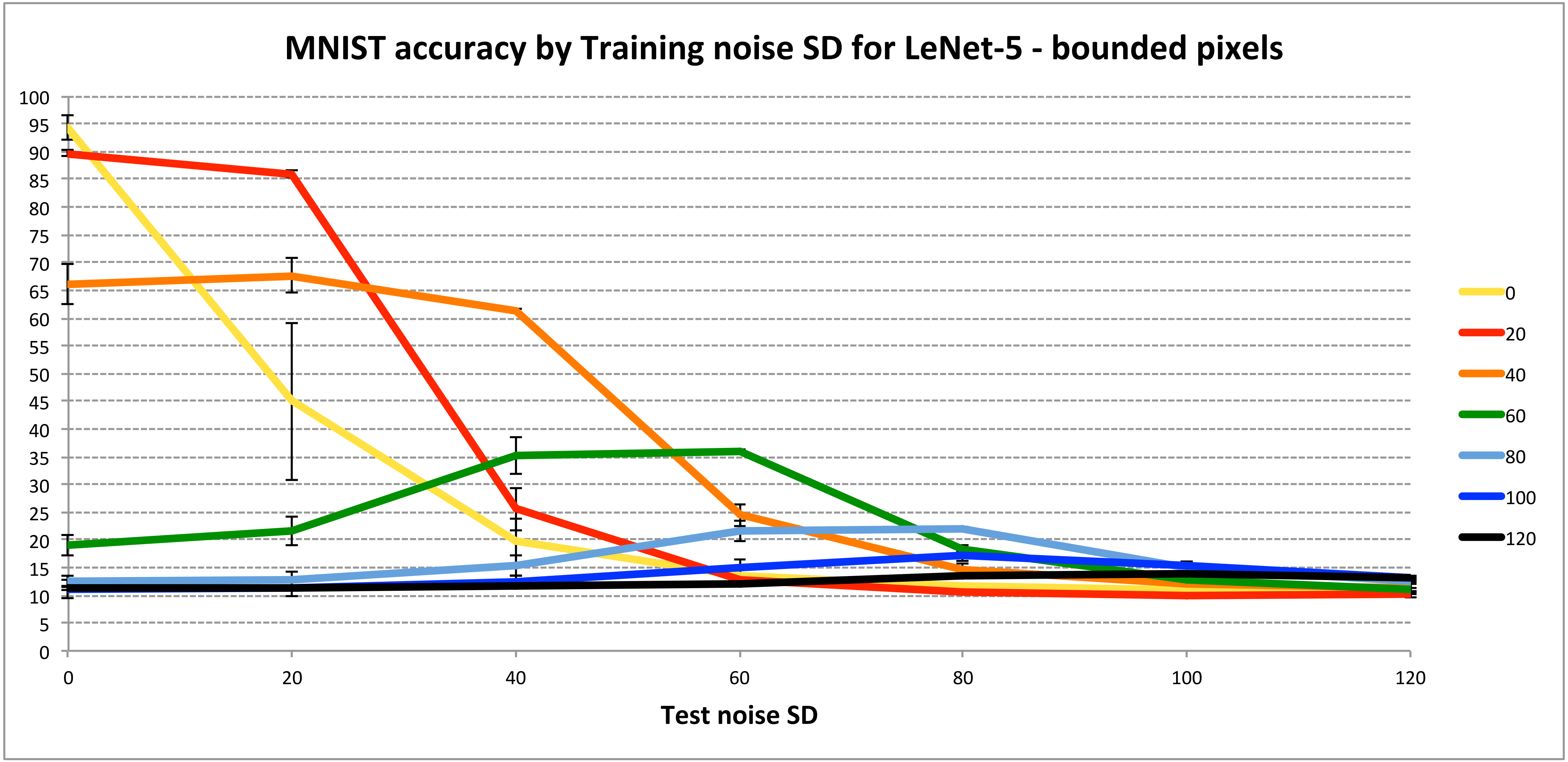


Basic results from MNIST (10 classes balanced)

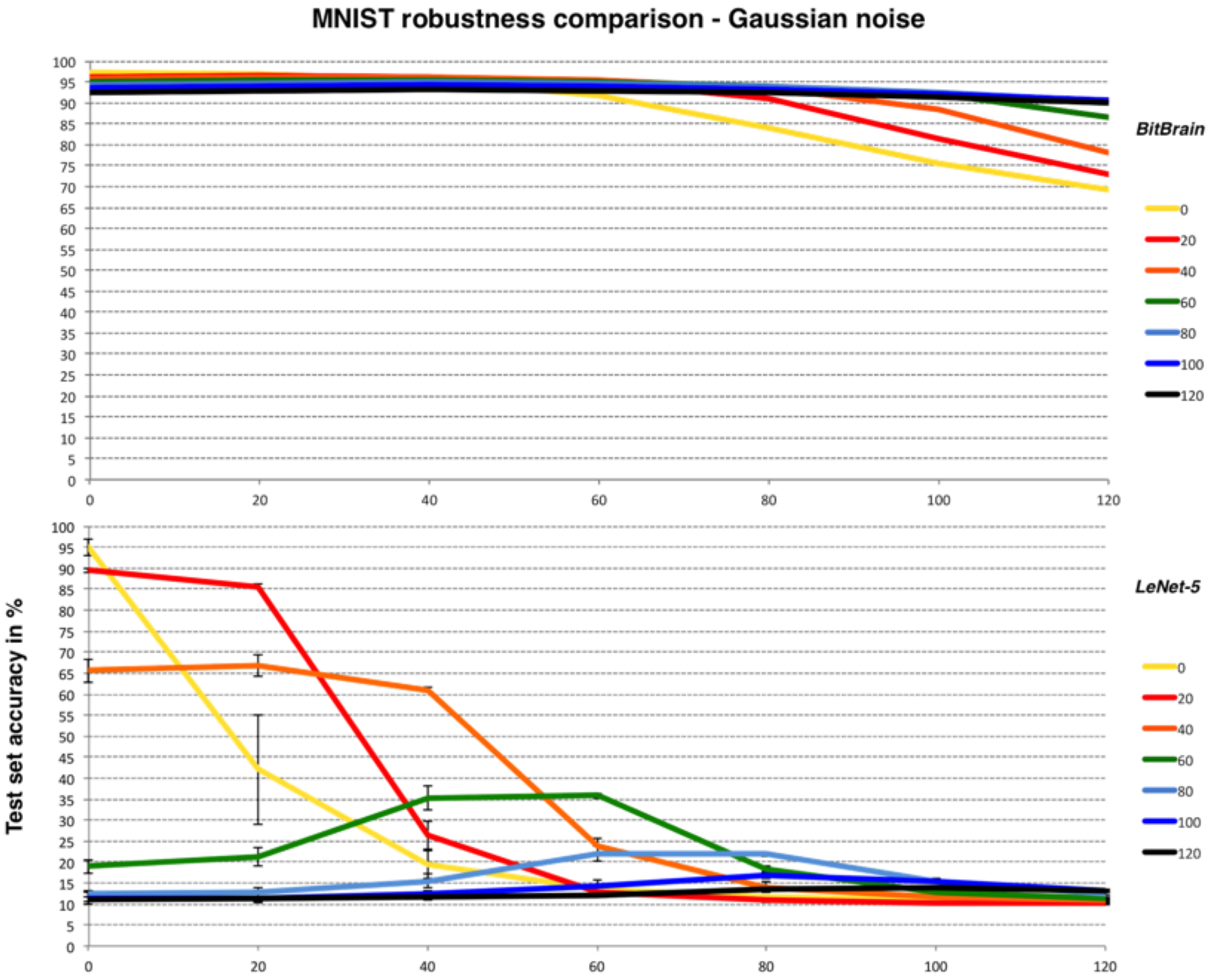


MNIST from *LeNet-5*

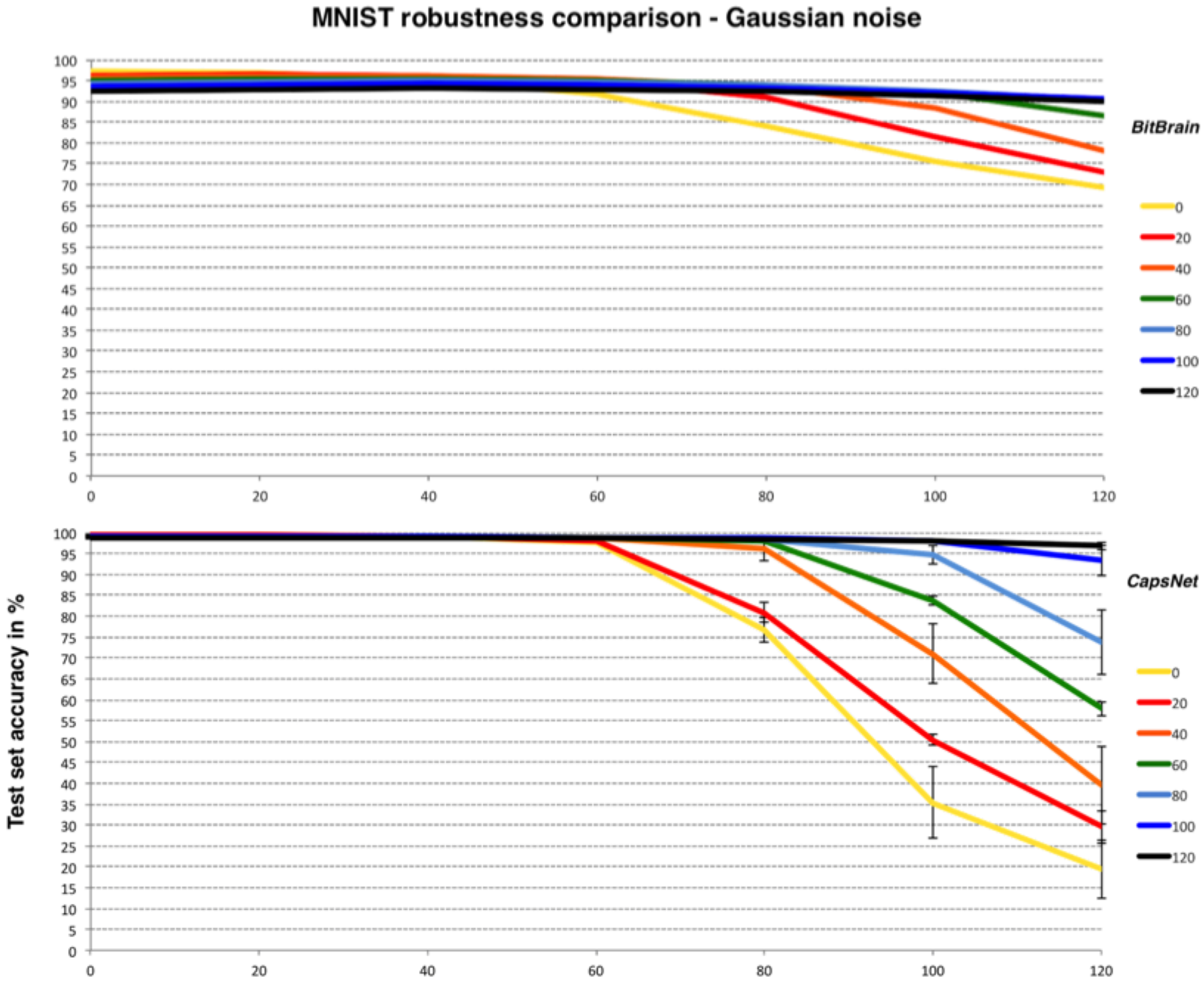
Early but respected CNN designed for character recognition, see https://en.wikipedia.org/wiki/LeNet
Max 100 epochs, early stopping with `patience' = 5. Sigmoidal activations, `static' noise



MNIST robustness comparison *BitBrain* vs *LeNet-5*



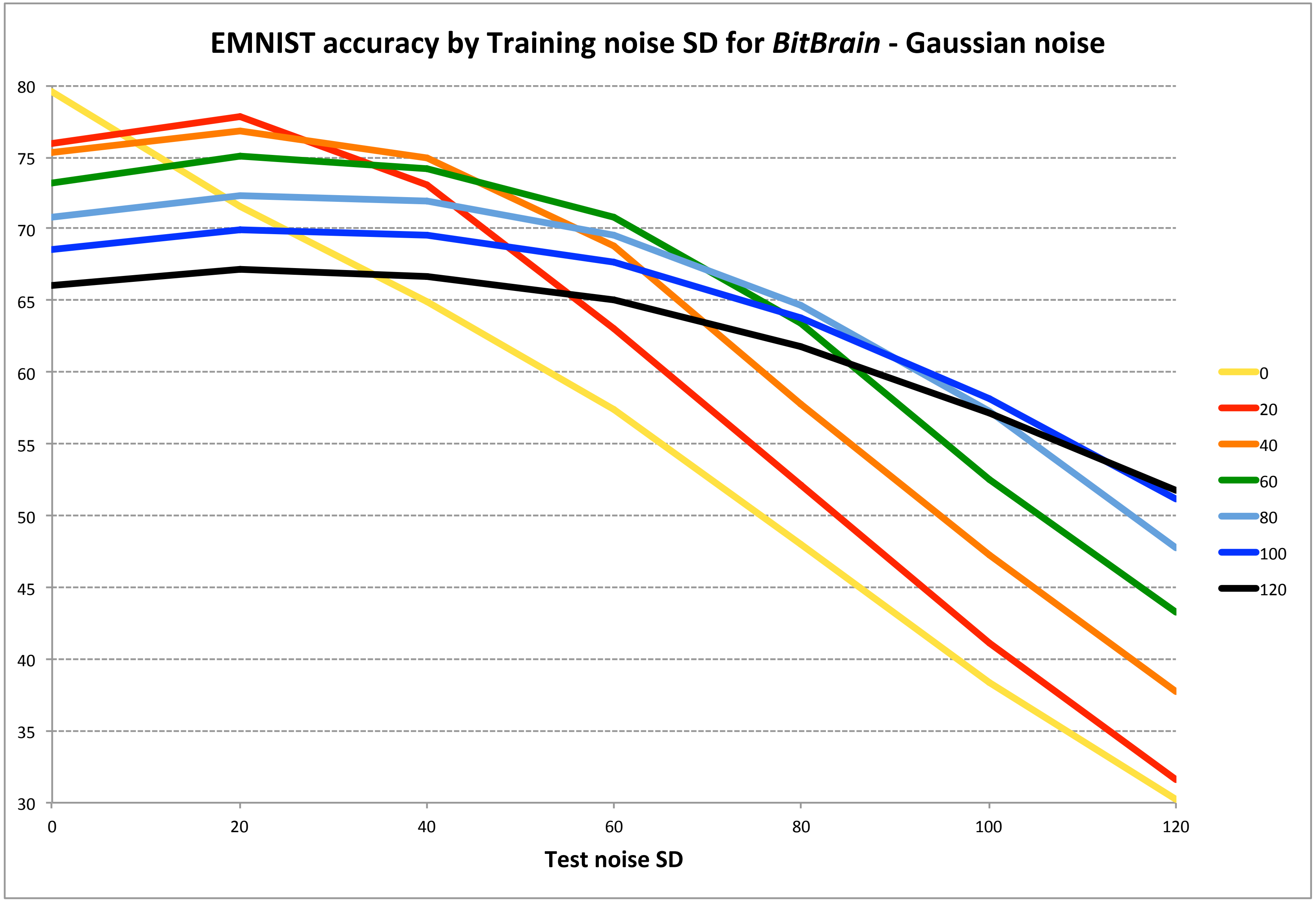
MNIST robustness comparison *BitBrain* vs *CapsNet*



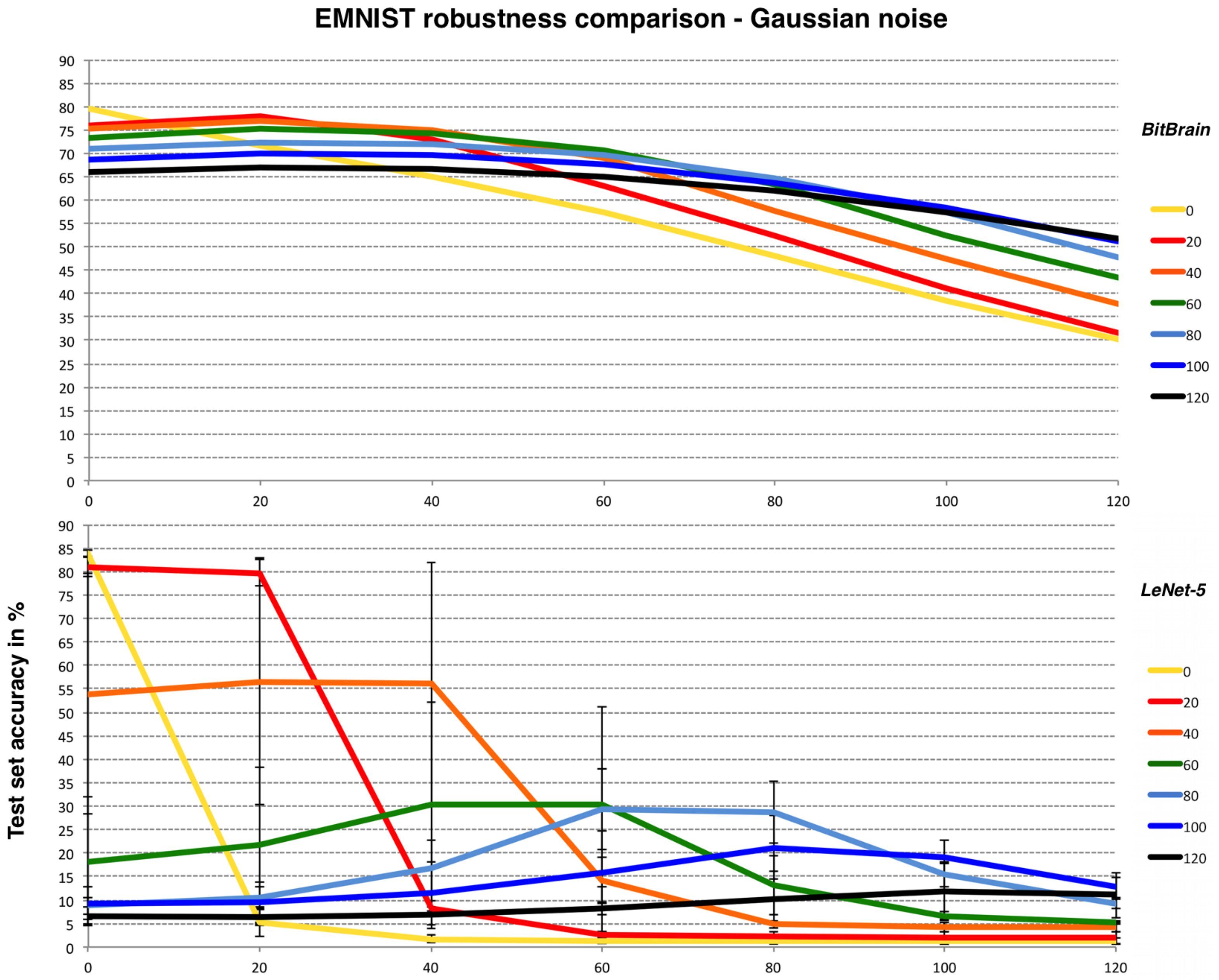
Basic results from EMNIST (62 classes unbalanced)

BitBrain setup identical to MNIST. Much harder problem. Very unbalanced and natural class aliasing:

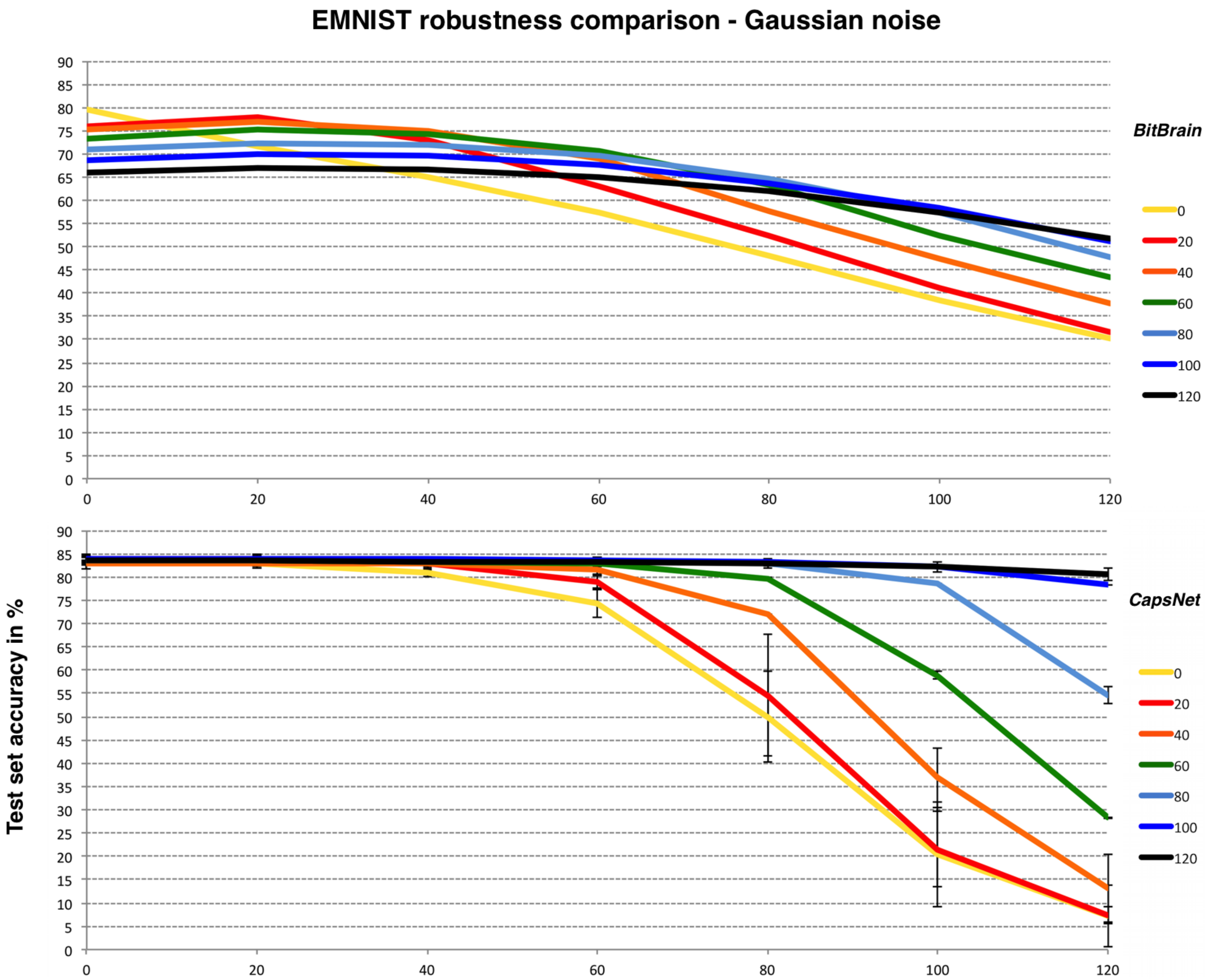
{ o, O, 0 }, { i, I, l, 1 }, { s, S, 5 }, { B, 8 }



EMNIST robustness comparison *BitBrain* vs *LeNet-5*

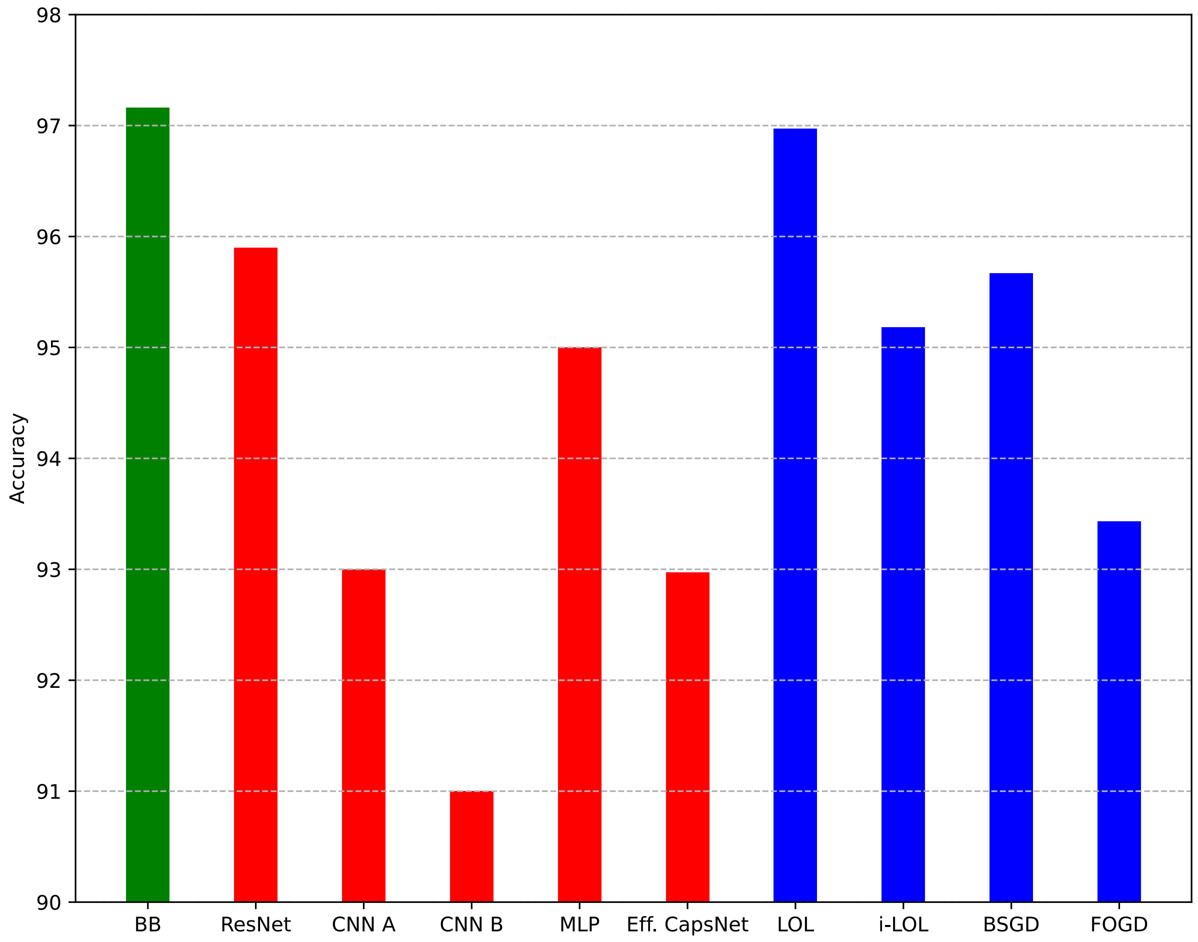


EMNIST robustness comparison *BitBrain* vs *CapsNet*



MNIST comparison with other single-pass methods - 1

Red bars are CNNs trained for only one epoch
Blue bars are specifically designed single-pass classification methods
References for all methods are in our upcoming paper



MNIST comparison with other single-pass methods - 2

Another publication compares single-pass SVM methods for two-class problems.

Their table 1 giving % accuracy on the test set for differentiating *0* vs *1* and *8* vs *9* is reproduced below, including *BitBrain* results. Best results are in **bold**.

	libSVM	*Perceptron*	*Pegasos 1*	*Pegasos 20*	*LASVM*	*StreamSVM 1*	*StreamSVM 2*	*BitBrain*
0 vs *1*	99.52	99.47	95.06	99.48	98.82	99.34	99.71	**99.95**
8 vs *9*	96.57	95.90	69.41	90.62	90.32	84.75	94.70	**98.49**

Reproduced from: Piyush Rai, Hal III, and Suresh Venkatasubramanian. *Streamed learning: One-pass SVMs.* IJCAI International Joint Conference on Artificial Intelligence, 2009.

Conclusions

BitBrain status:

- Novel single-pass learning mechanism
- Accurate classification – best in single-pass class on MNIST
- Good robustness to imperfect inputs and other forms of uncertainty
- Simple and energy-efficient operation (small integer and bitwise - no floating point)
- Single-thread implementation on on 3.2GHz Apple ARM M1 gives 10k inferences in 0.42 secs
- Improvements investigated: 'Jitter'/data augmentation and weighting of counts by occupancy both gain ≈1% on MNIST

To do and in progress:

- More challenging 2D image benchmarks: CIFAR-10 & -100, German traffic sign database, many others…
- CNN or other biologically-inspired front end
- Continuous adaptive learning, with or without forgetting
- Layers of SBC memories connected in novel ways
- Application to different types of data: time series, DNA/biology, abstract codes, 3D volumetric, …
- Differing time delays on synaptic connections for automatic spatio-*temporal* pattern classification
- More underlying theory; particularly connections to Kernel methods, SDM, VSA/VFA, …
- Good mappings to GPU, FPGA, SpiNNaker, other neuromorphic platforms, specialised hardware?