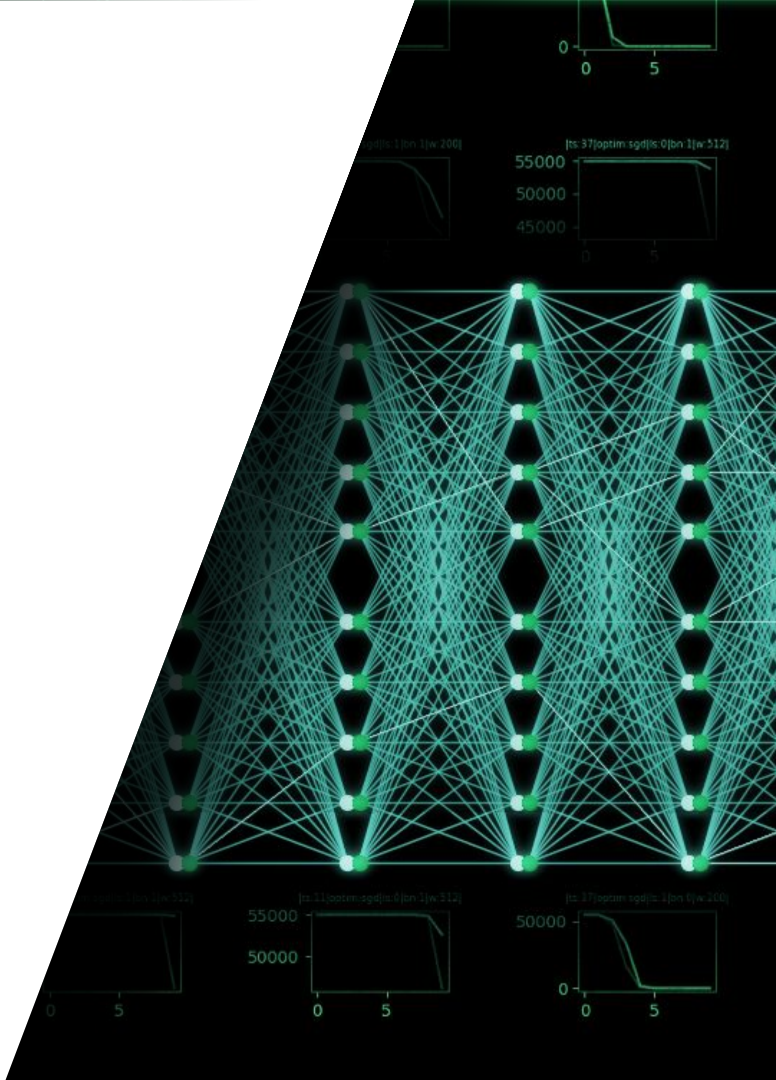
DNNs as layers of cooperating classifiers

Davel, Theunissen, Pretorius & Barnard

Marelie Davel

North-West University, South Africa
Centre for Artificial Intelligence Research



Overview

- Generalization in DNNs
- Inside a fully-connected feedforward network
  - Regularities in node behavior
  - Mechanisms that cause them
  - ‘Two collaborating systems’
- Implications

Generalization in DNNs

Understanding DNNs

- Expressivity
- Trainability
- Generalization

The ‘apparent paradox’ (Kawaguchi et al., 2019)

- Low-capacity class $\implies$ generalization (Vapnik, 1998)
- DNNs extemely large capacity (Zhang et al., 2017; Dinh et al., 2017)

Generalization in DNNs

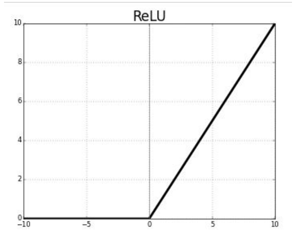
Many approaches:

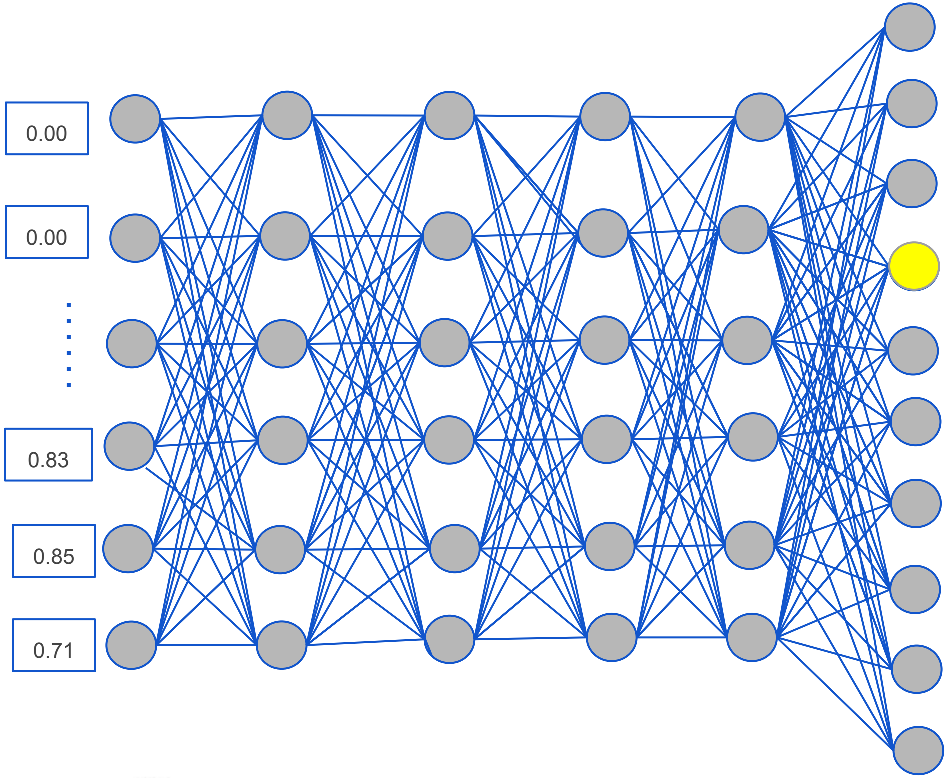
- Complexity of the hypothesis space; regularised network capacity; small norms
- Geometry (smoothness) of the loss surface; flat minima
- Statistical measures: uniform stability, robustness
- Large classification margins

Framework for characterizing generalisation behavior in general circumstances remains elusive

Architecture: start simple

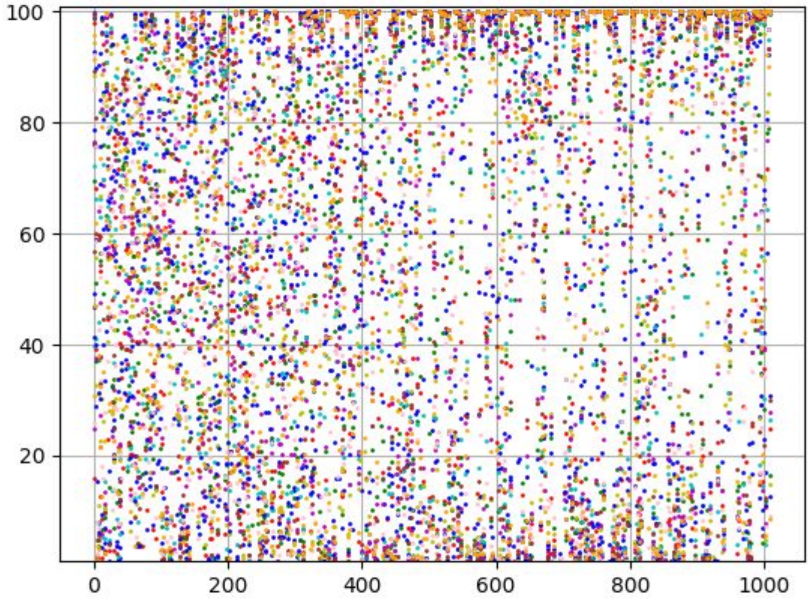
- Feedforward fully-connected
- Classification
- ReLU hidden activations



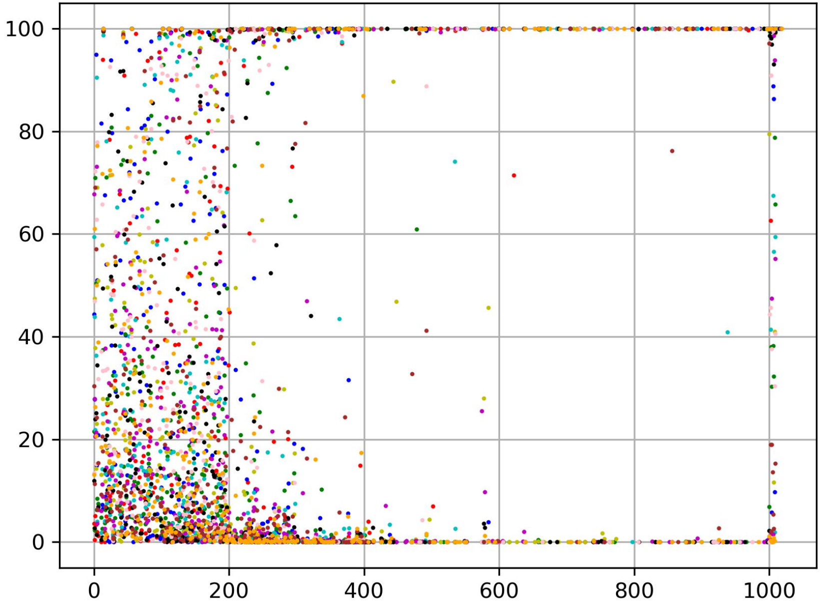


Node behavior: activation patterns

MNIST initialized



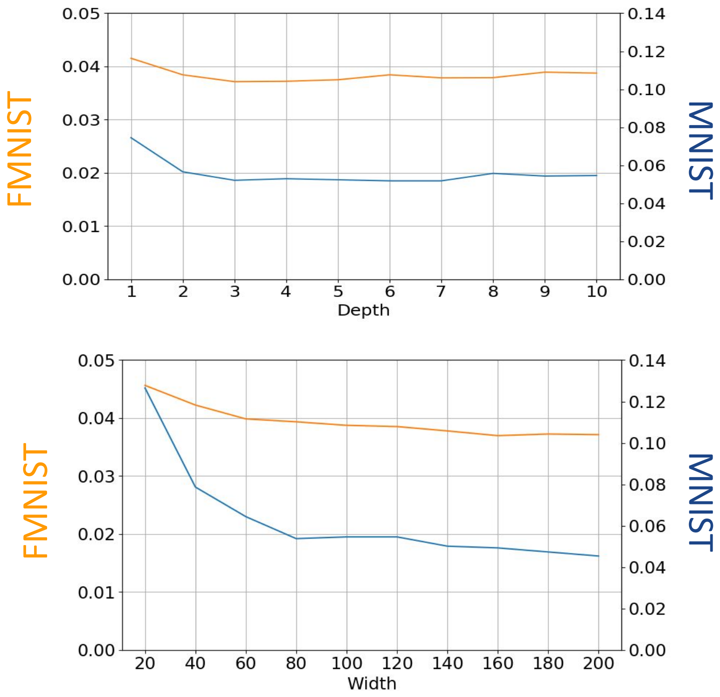
MNIST trained



Experimental setup

- MNIST, FMNIST
- Different architectures:
  - Depth: 1-10 layers by 100 nodes
  - Width: 10 layers by 20-200 nodes
- Standard setup:
  - SGD, Adam, early stopping, normalized uniform init, lr grid search, random training seeds
- No batchnorm, dropout, data augmentation

Test error



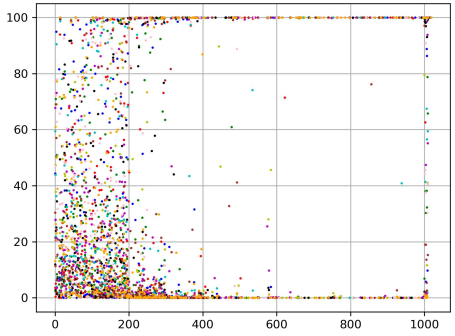
Node behavior: layer perplexity

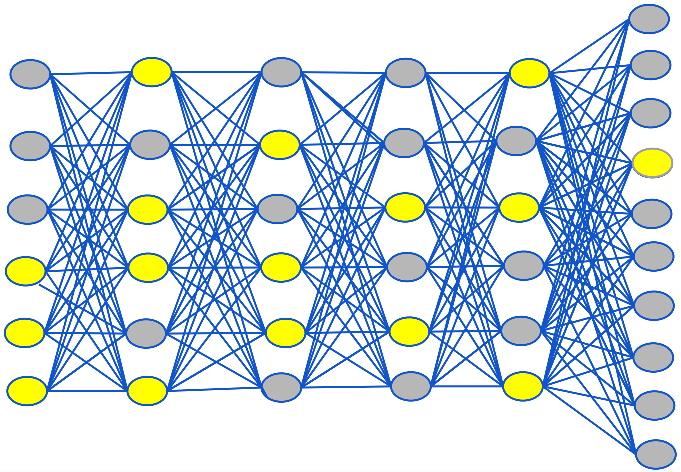
- Discrete representations per layer

- Average perplexity per class

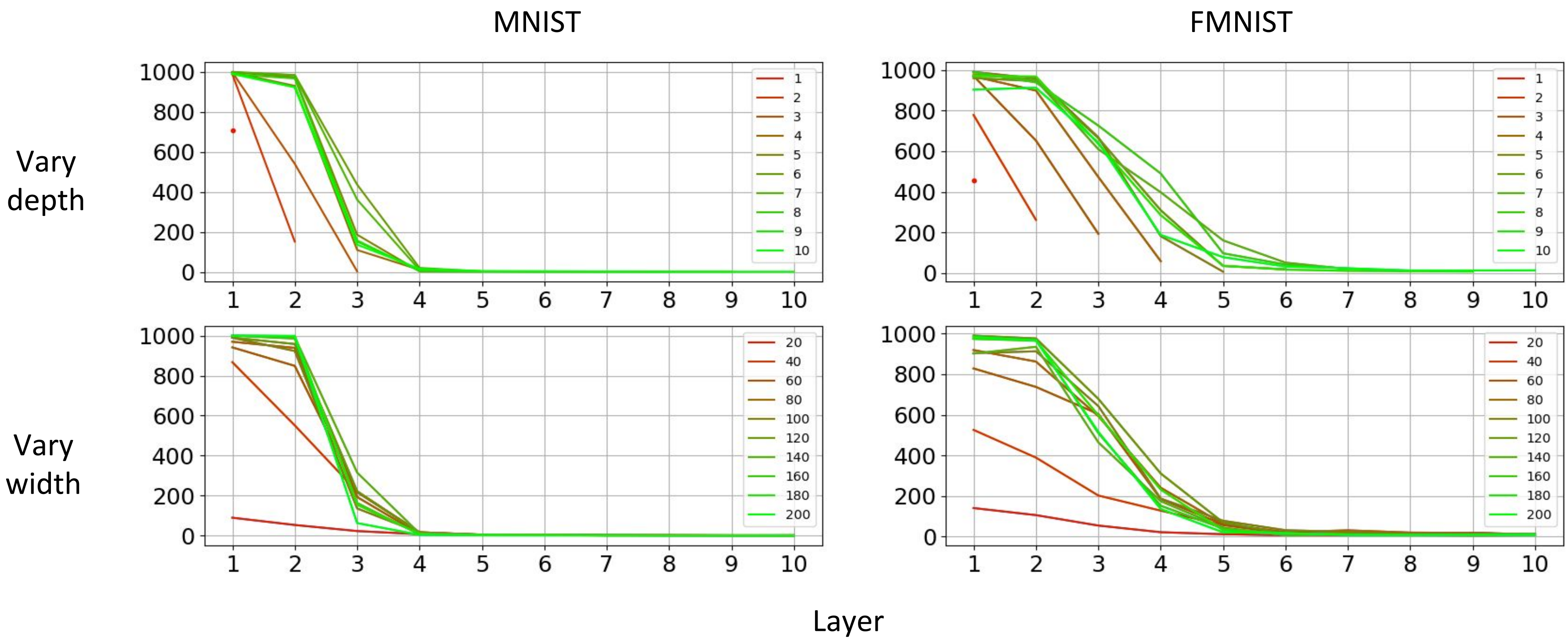
$$H(c,l) = -\sum_{n \in K(c,l)} \frac{n}{N_c} ln\left(\frac{n}{N_c}\right),$$

$$P(c,l) = e^{H(c,l)}$$





Node behavior: layer perplexity

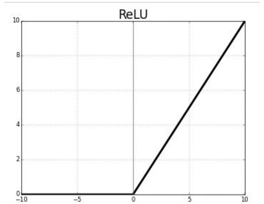


Iterative SGD

- Weight update: recursive calculation $$\Delta w_{i,j,k} = -\eta \frac{\partial E}{\partial w_{i,j,k}}$$

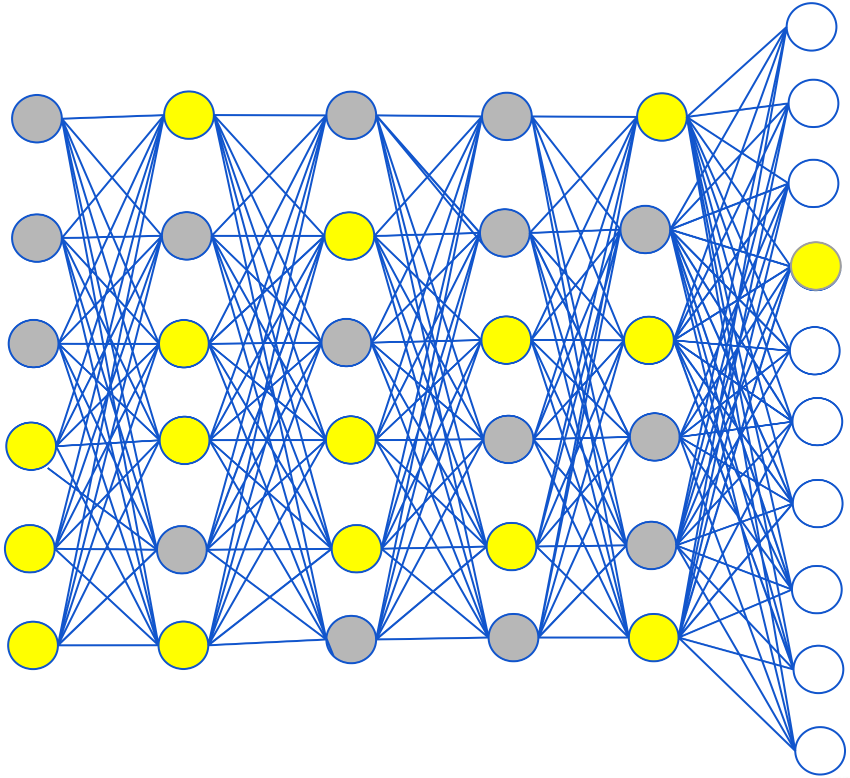
- Use

$$\begin{aligned} Relu(x) &= xT(x) \\ \text{where } T(x) &= \begin{cases} 1 \text{ if } x > 0 \\ 0 \text{ if } x \leq 0 \end{cases} \end{aligned}$$



- Incorporate bias in first layer as extra weight
- Enumerate all components

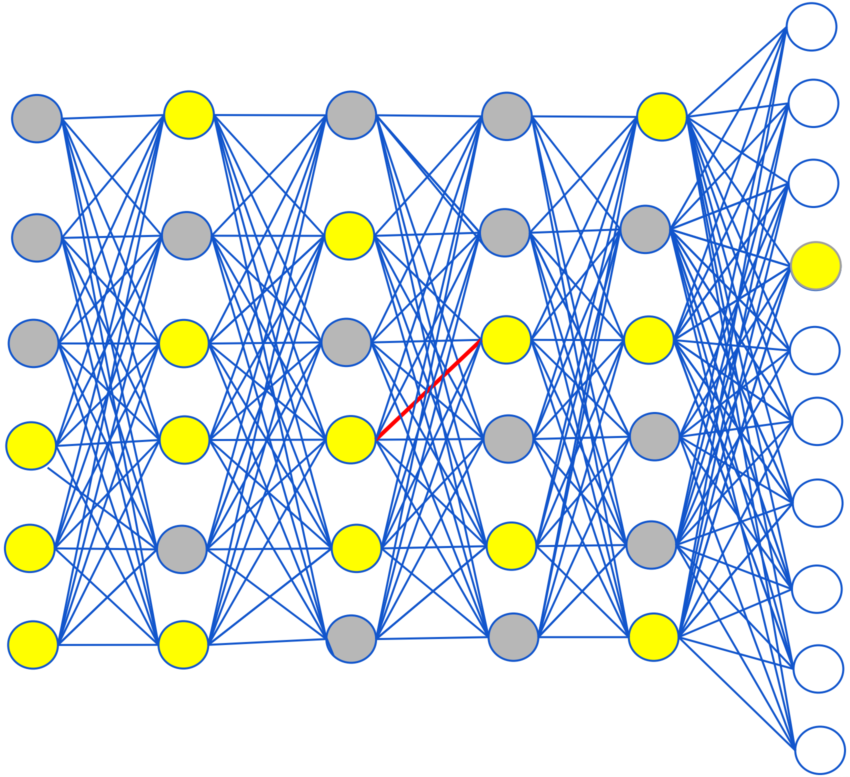
Iterative SGD



$$\Delta W_{i,j,k} = \eta\, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

ReLU + CrossEntropy + Softmax

Iterative SGD

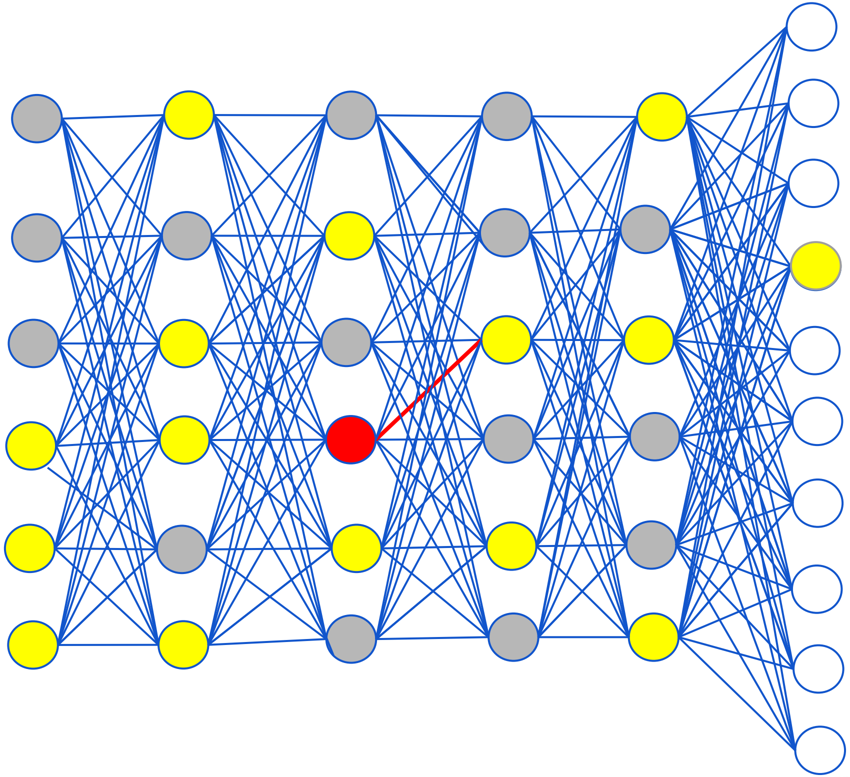


weight update

$$\Delta W_{i,j,k} = \eta \, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

ReLU + CrossEntropy + Softmax

Iterative SGD



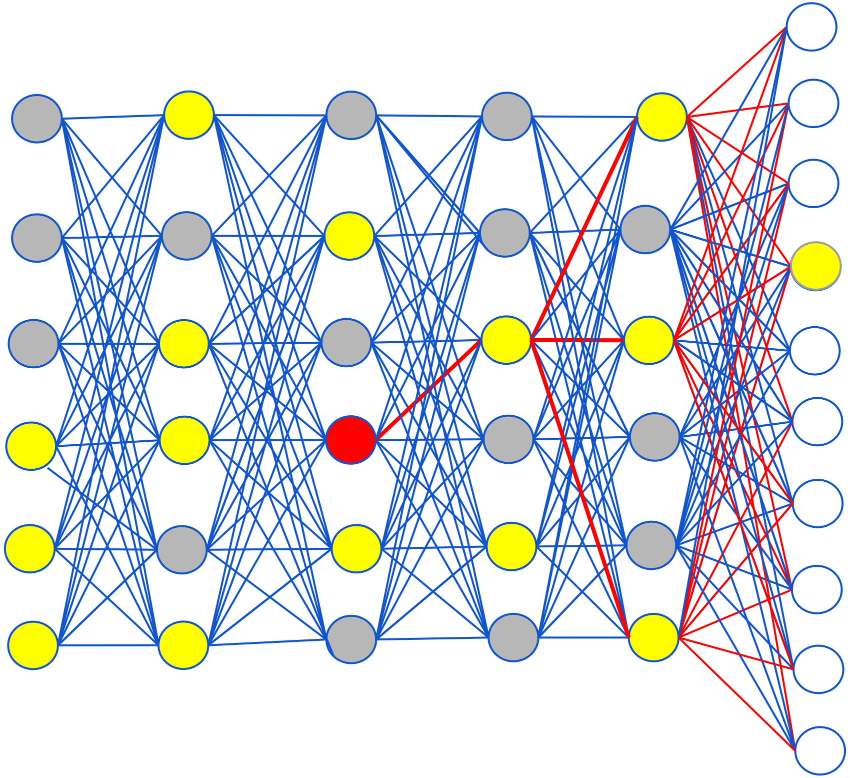
activation
feeding into
weight

$$\Delta W_{i,j,k} = \eta\, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

weight
update

ReLU + CrossEntropy + Softmax

Iterative SGD



activation
feeding into
weight

1 if path
active,
0 if not

$$\Delta W_{i,j,k} = \eta\, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

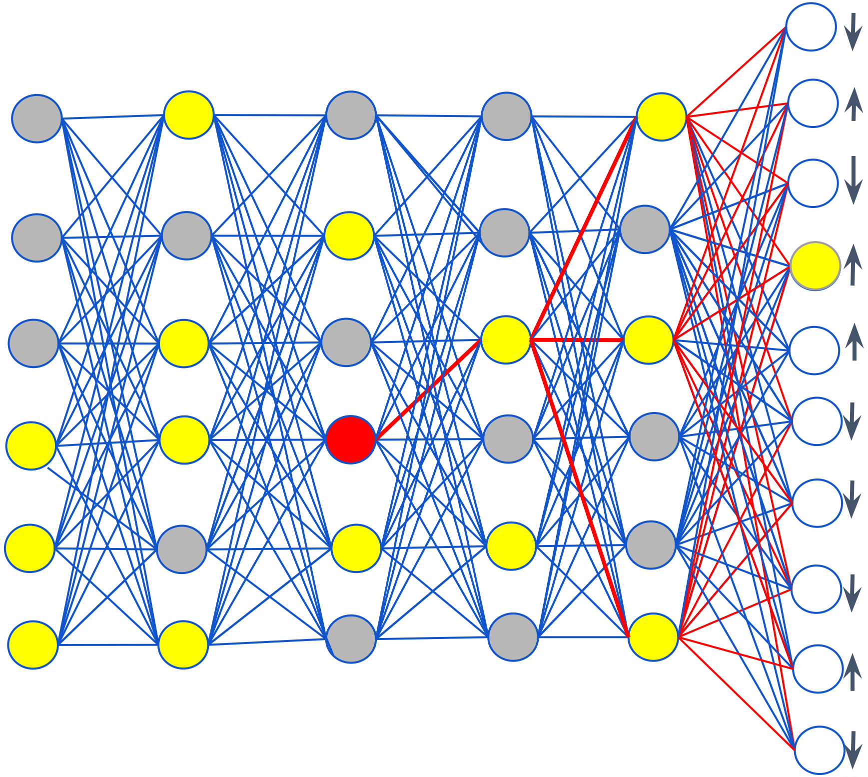
weight
update

sum over
all possible
paths

product of
weights on
this path

ReLU + CrossEntropy + Softmax

Iterative SGD



activation feeding into weight

1 if path active, 0 if not

$$\Delta W_{i,j,k} = \eta\, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

weight update

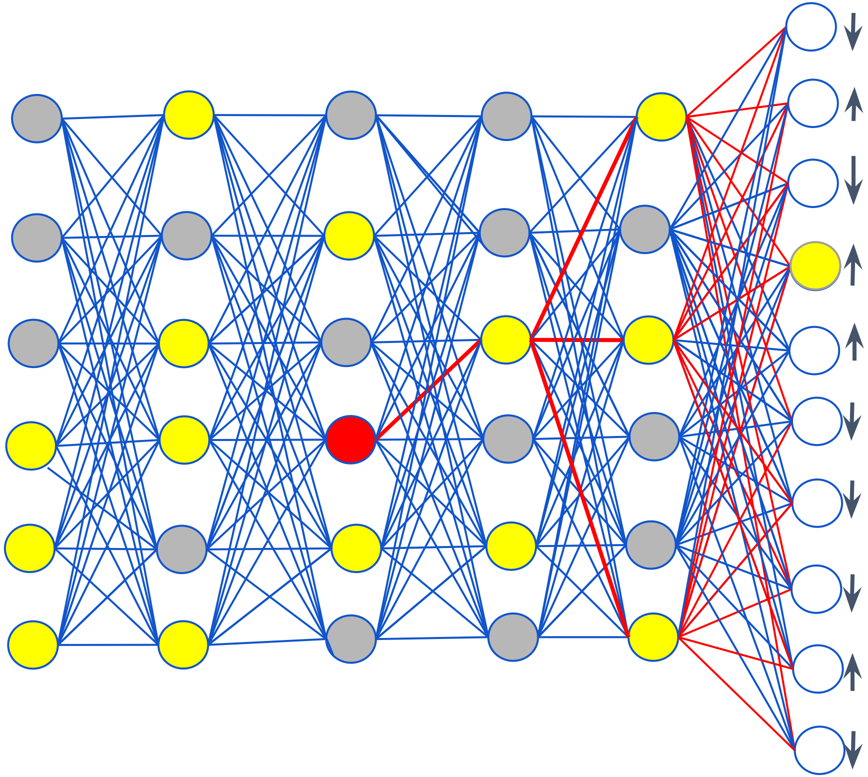
sum over all possible paths

product of weights on this path

target - actual in last layer

ReLU + CrossEntropy + Softmax

Iterative SGD



activation feeding into weight

1 if path active, 0 if not

learning rate

$$\Delta W_{i,j,k} = \eta \, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

weight update

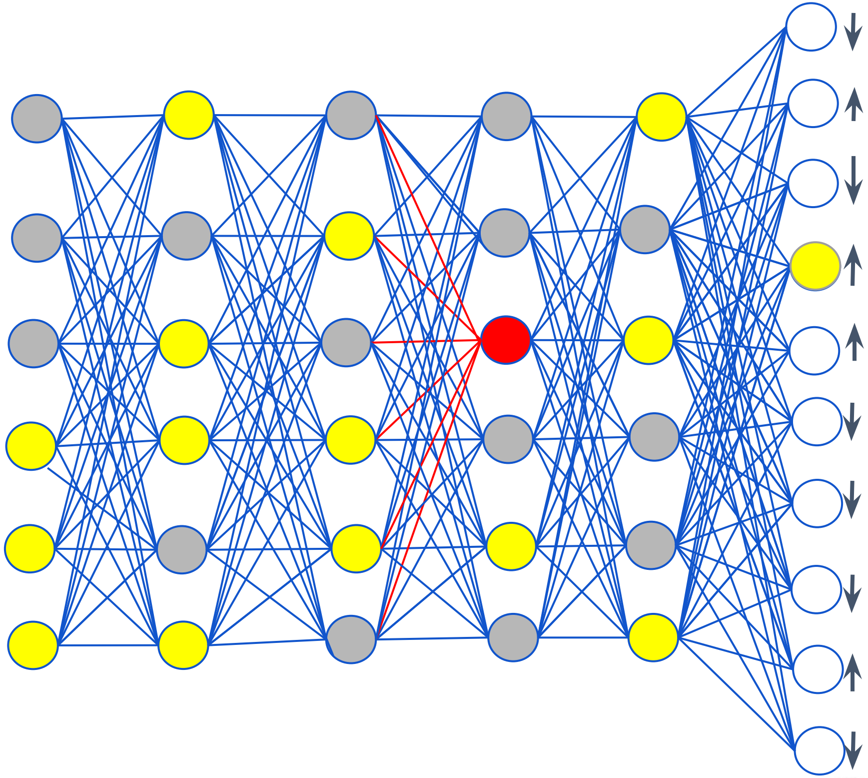
sum over all possible paths

product of weights on this path

target - actual in last layer

ReLU + CrossEntropy + Softmax

Iterative SGD



activation feeding into weight

1 if path active, 0 if not

learning rate

$$\Delta W_{i,j,k} = \eta\, a_{i-1,k} \sum_{b=0}^{B_i-1} \prod_{g=i}^{N-1} T(z_{g,I(g,b)}) \prod_{r=i+1}^{N} W_{r,I(r,b),I(r-1,b)} \left(y_{I(N,b)} - h_{N,I(N,b)}\right)$$

weight update

sum over all possible paths

product of weights on this path

target - actual in last layer

ReLU + CrossEntropy + Softmax

Two synergistic systems

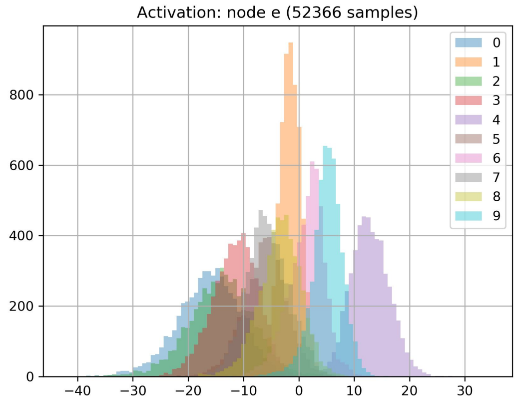
- Forward process:
  - Group samples with regard to features; create ‘sample sets’
  - Discrete process
- Adaptation process:
  - Optimise nodes locally
  - Attune node to features relevant to local sample set
  - Continuous process
- Nodes collaborate globally

Can we measure the effect of these systems?

Nodes as classifiers

- Estimate P(class|obs) at each node
  - Continuous: fit KDE
  - Discrete: count samples
  - Combined: continuous if node on, discrete if node off
- Combine over a layer

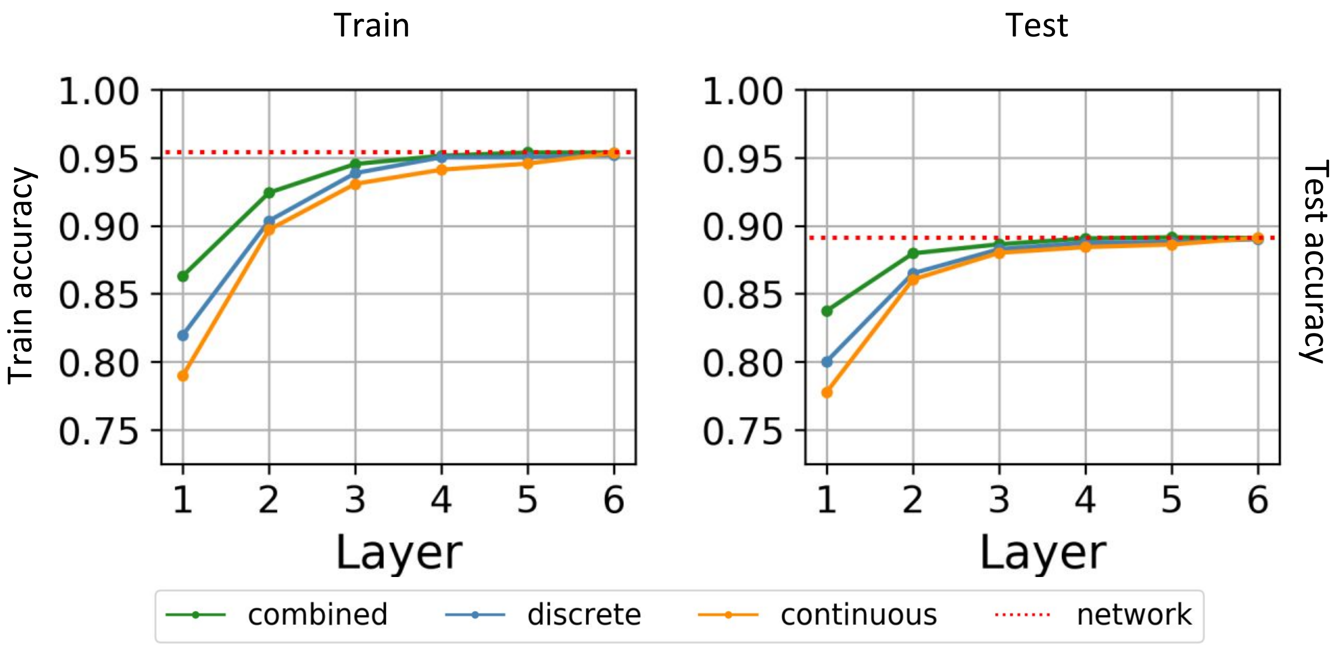
continuous



discrete

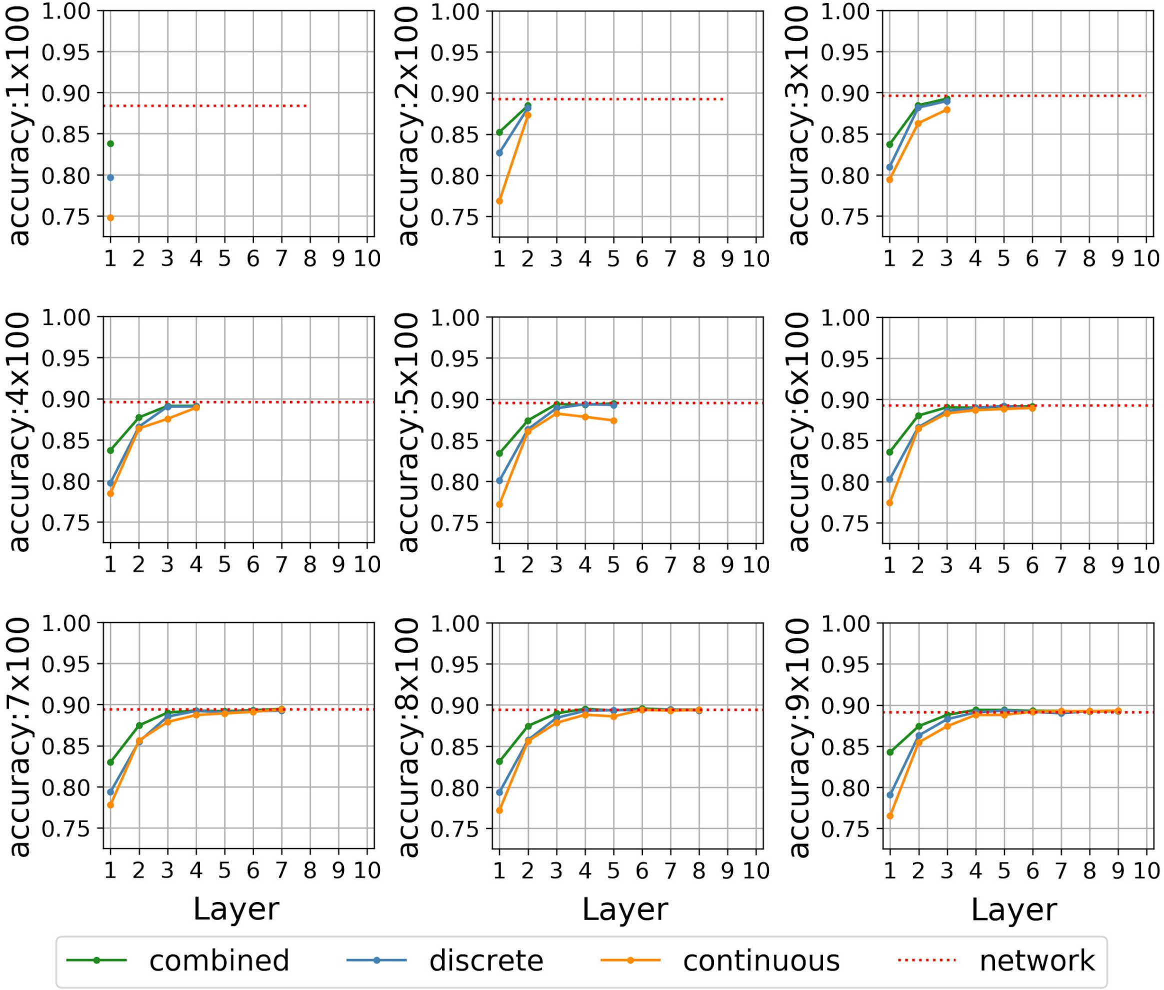
class	% active
1	0.854
2	0.432
3	0.105
...	...

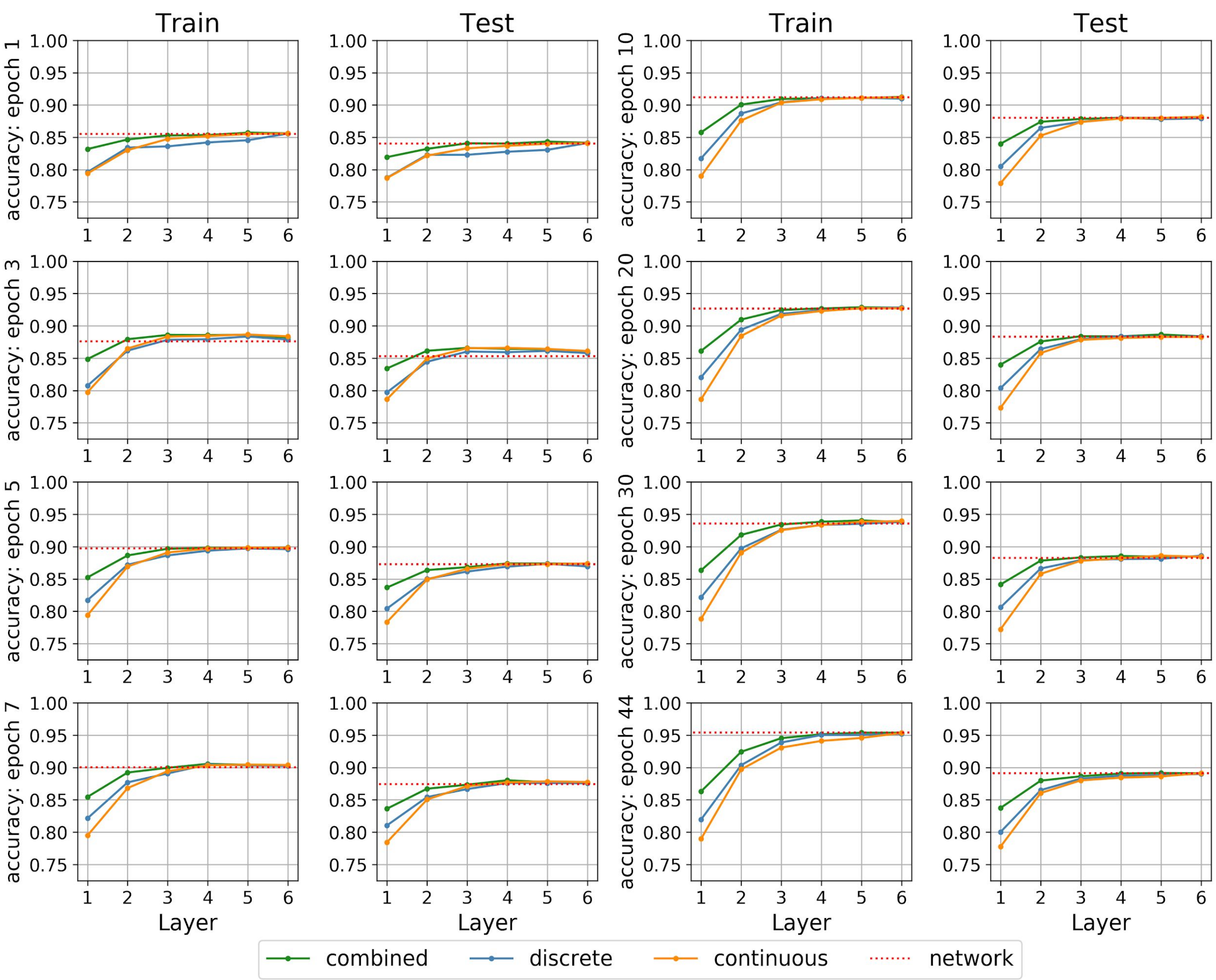
Two synergistic systems



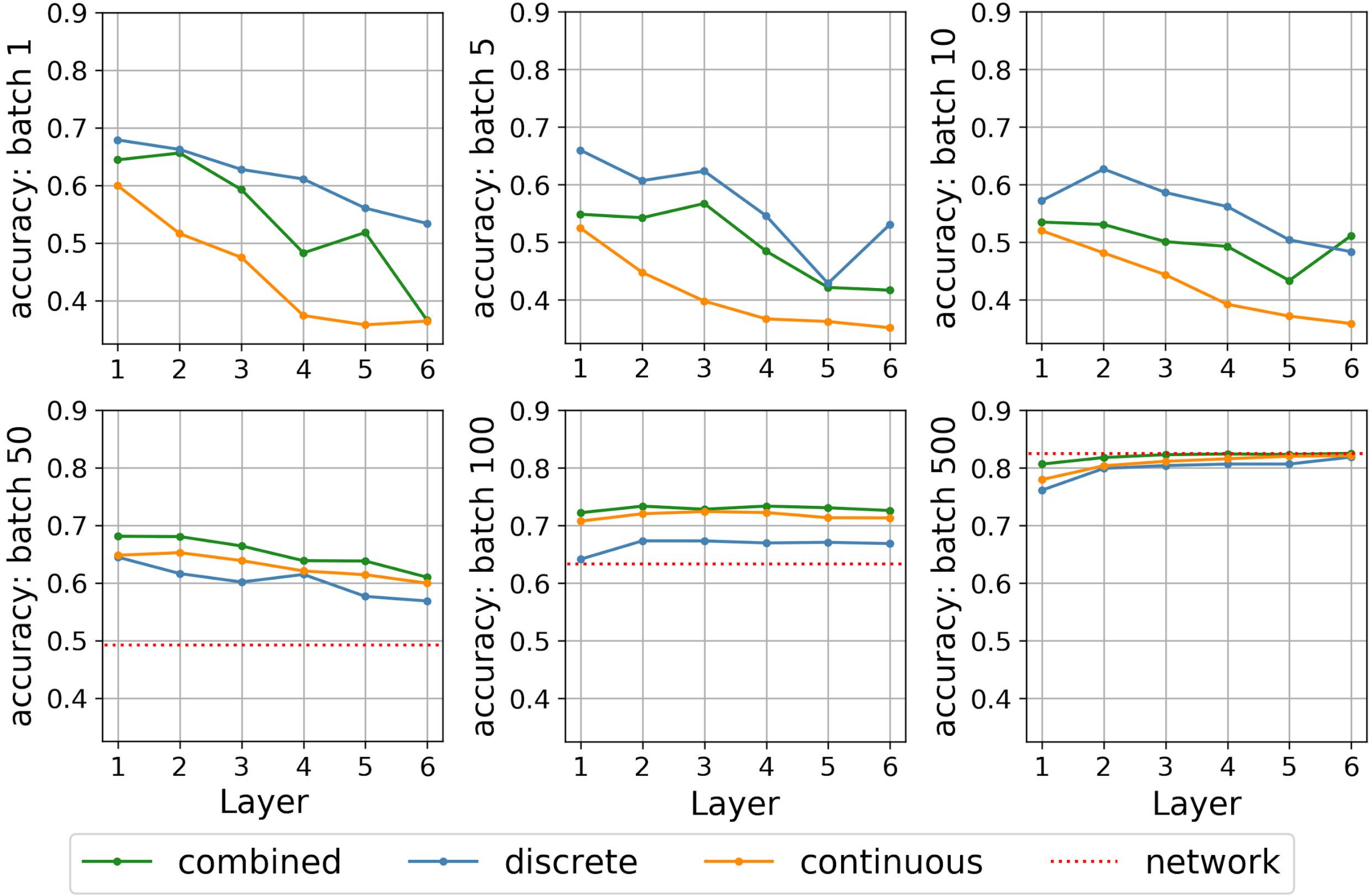
6x100 trained network, MNIST

Trained networks, different depths (FMNIST)



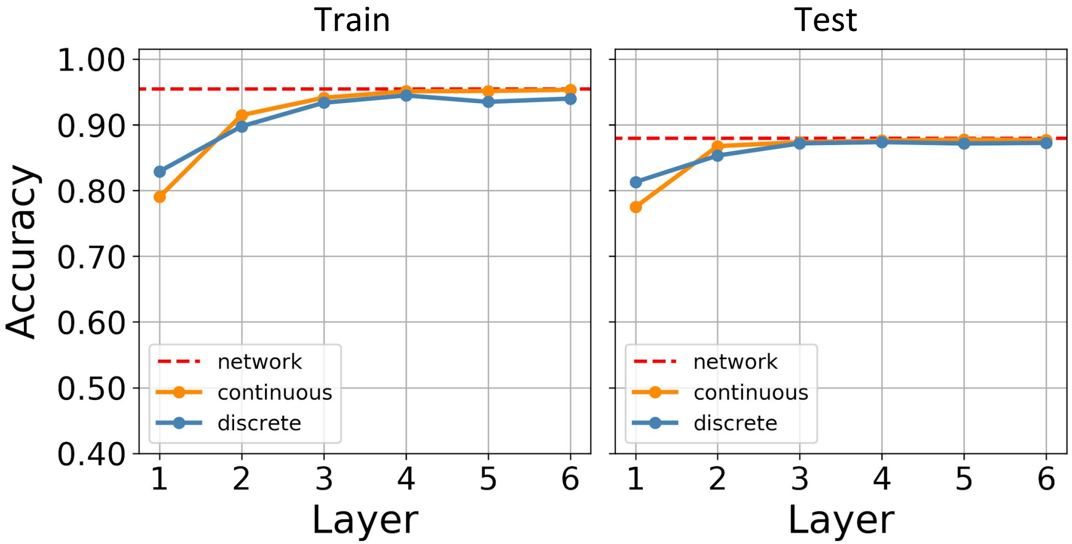


6x100 network during training (FMNIST)



6x100 network during training (FMNIST), first epoch

What if not ReLU?

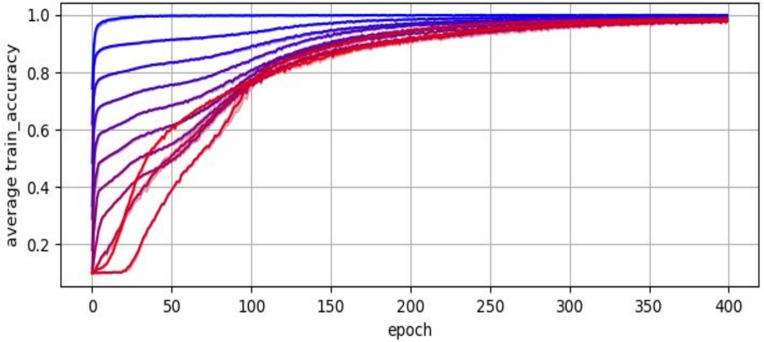


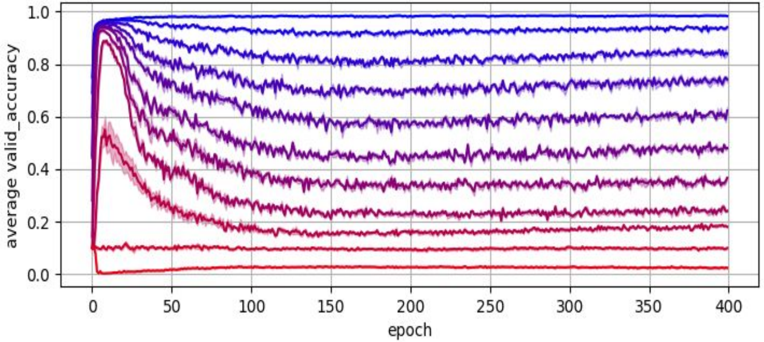
7x100 trained network,sigmoid activations, FMNIST

Why relevant?

- Viewpoint + analysis tools
  - Can probe generalization ability of a network
  - Shed light on role of sub-components in solving sub-tasks
- Investigate balance between opposing goals
  - Grouping / separating samples
  - General / specialist behavior
  - Local / global optimization

Label corruption

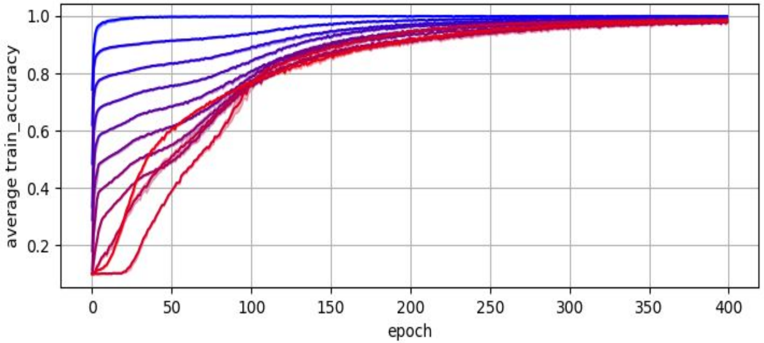


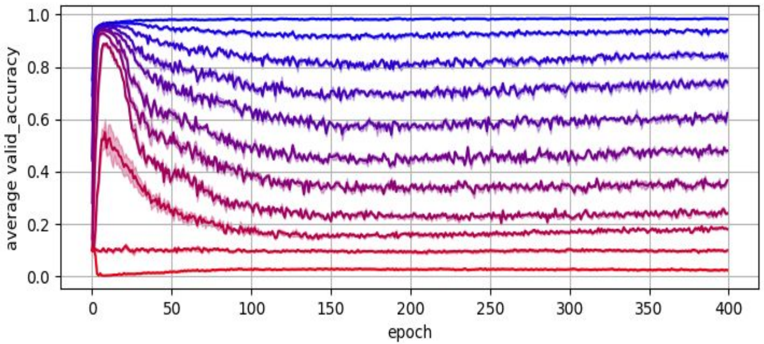


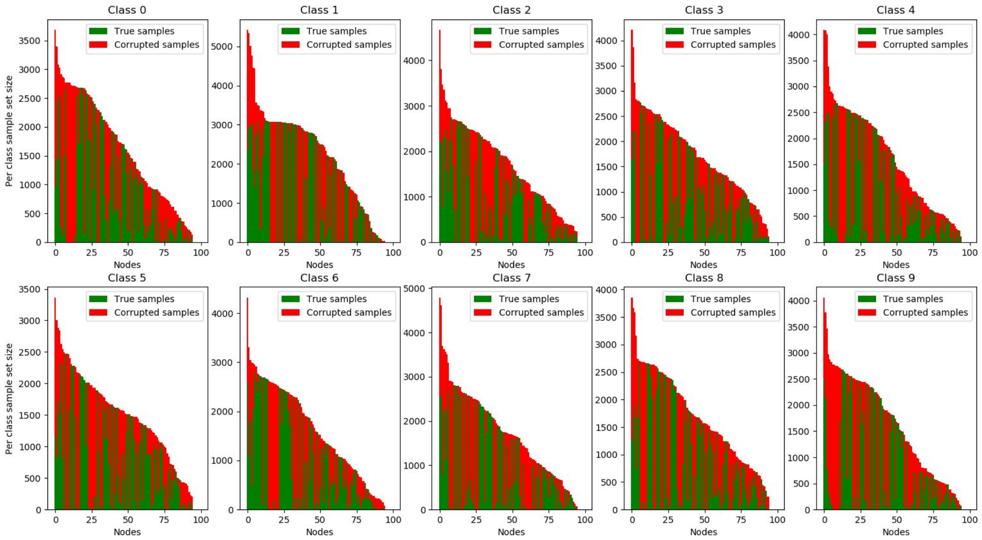
Theunissen, Davel & Barnard, 2019. "Insights regarding overfitting on noise in deep learning"

Reproduced based on Zhang et al, 2017. "Understanding deep learning requires rethinking generalization"

Label corruption

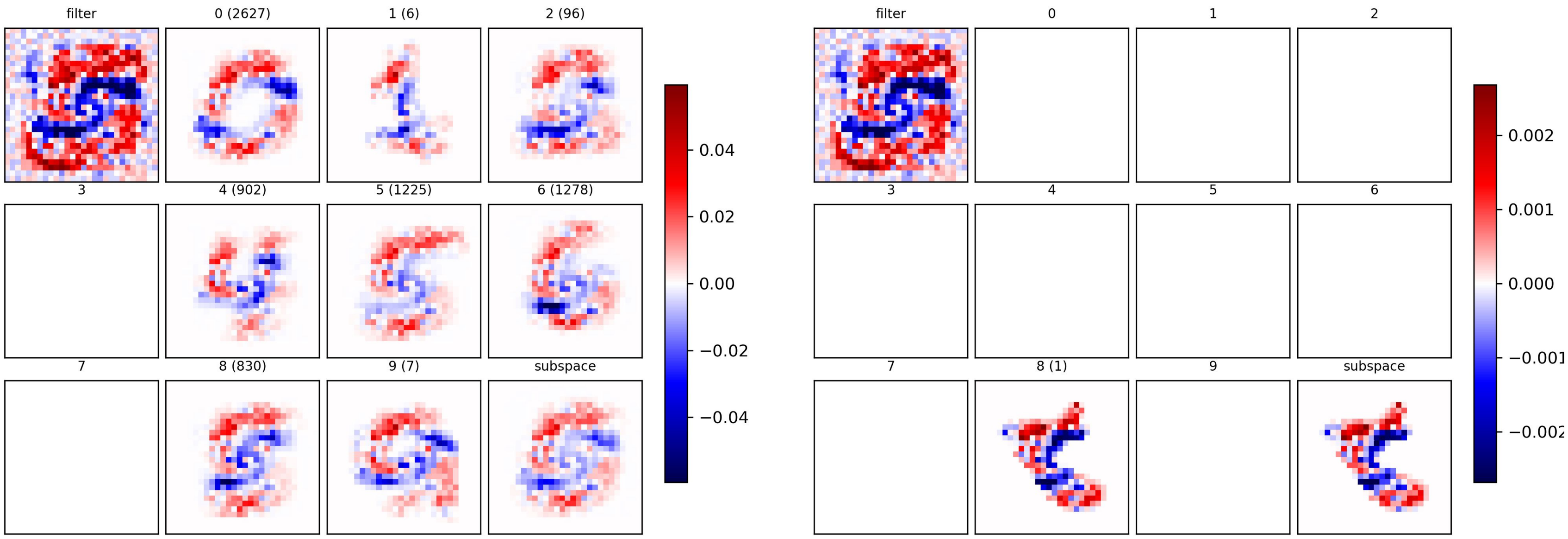






Theunissen, Davel & Barnard, 2019. "Insights regarding overfitting on noise in deep learning"

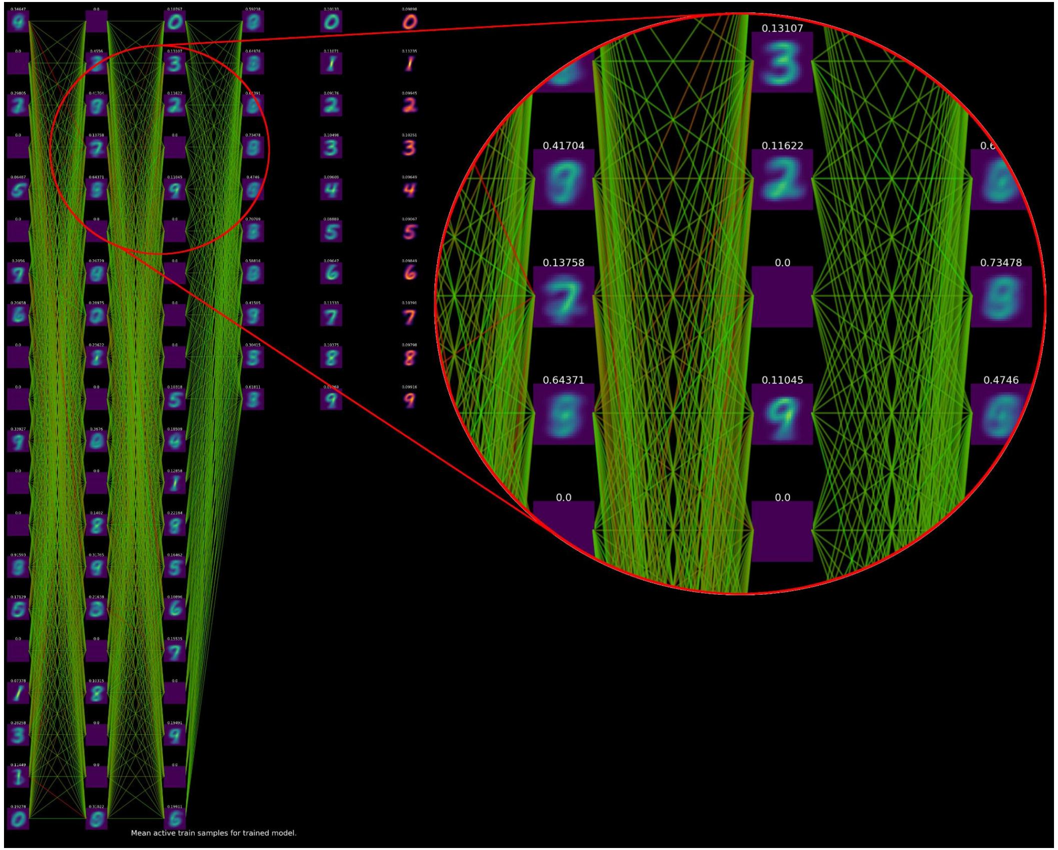
Quarantine paths



In summary

- Individual nodes can be regarded as classifiers, making accurate predictions at individual layers
- SGD creates two interacting systems:
  - Discrete system that groups samples according to local relevancy
  - Continuous system that attunes each weight vector (flowing into a node) to the most relevant local features
- Generalization strength emerges from collaboration among distinct classifiers, each addressing a sub-population of the data
- Two-system analysis a conceptual tool for exploring further

Thank you!



MW Theunissen (in prep)