Capsule Networks for Low-Data Transfer Learning

Andrew Gritsevskiy
Mentor: Maksym Korablyov
Mentor: Gil Alterovitz

MIT PRIMES

May 20, 2018

Neural networks

Neural networks

- Universal function approximator

Neural networks

- Universal function approximator
  - Is this a dog?

Neural networks

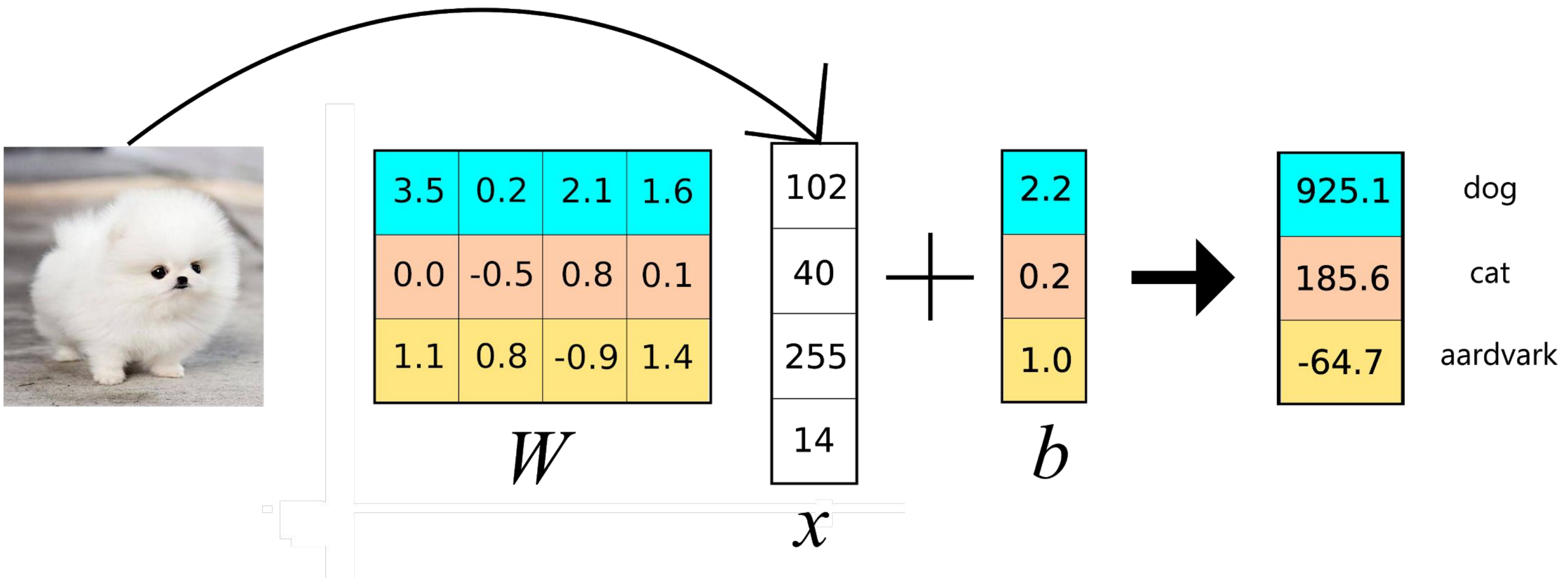
- Universal function approximator
  - Is this a dog?

Neural networks

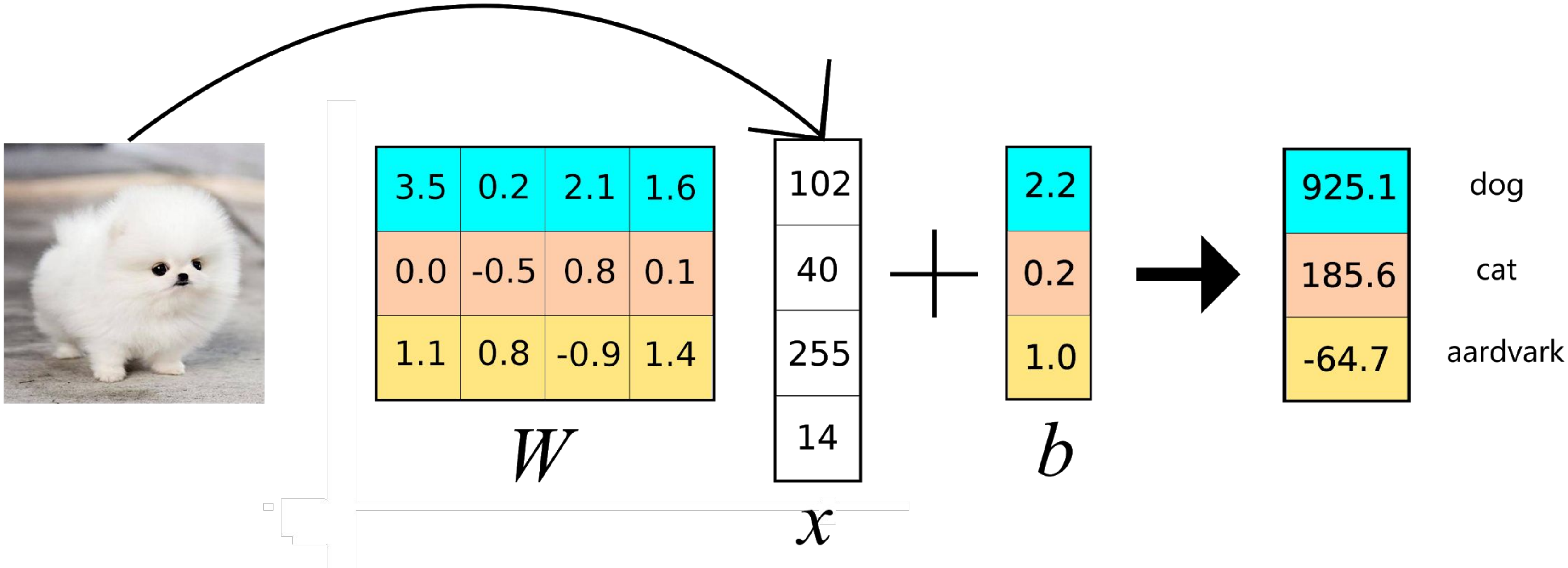
- Universal function approximator
  - Is this a dog?

Structure of a (linear) classifier

Structure of a (linear) classifier



Structure of a (linear) classifier



- Loss function

Adjusting weights

Adjusting weights

- Method I: Random
  - Accuracy: 15.5%

Adjusting weights

- Method I: Random
  - Accuracy: 15.5%
- Method II: Random local search
  - Accuracy: 21.4%

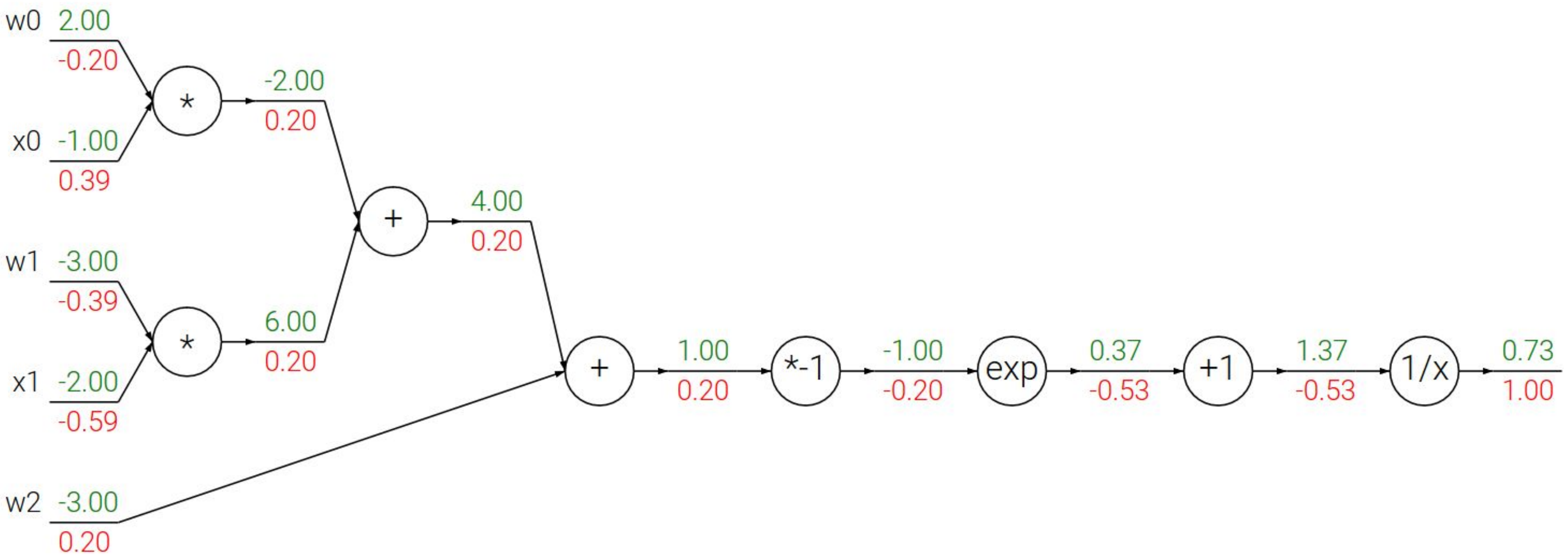
Adjusting weights

- Method I: Random
  - Accuracy: 15.5%
- Method II: Random local search
  - Accuracy: 21.4%
- Method III: Gradient descent

Gradient descent

```
while True:
    gradients = calculate_gradient(loss_function, data, weights)
    weights += - step_size * gradients
```

Backpropagation



Capsule networks

Capsule networks

- Neurons store information as vectors

Capsule networks

- Neurons store information as vectors
- Vectors store pose information

Capsule networks

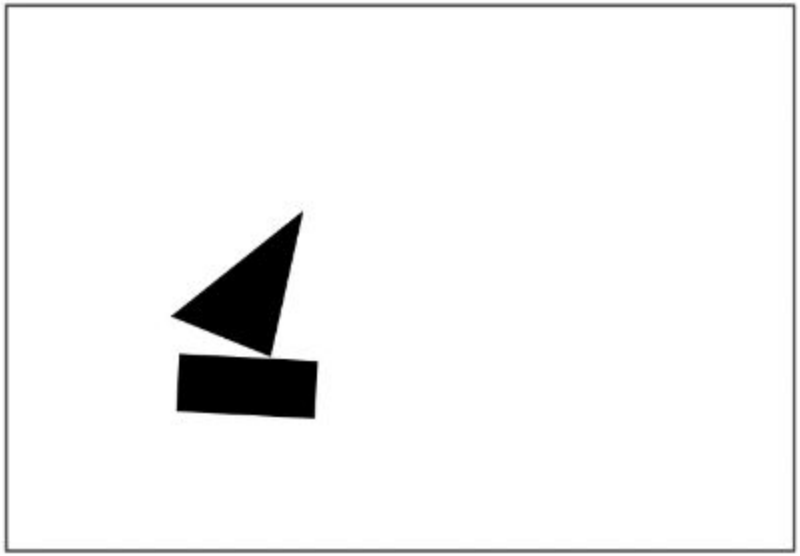
- Neurons store information as vectors
- Vectors store pose information
  - Vector points in direction of object orientation

Capsule networks

- Neurons store information as vectors
- Vectors store pose information
  - Vector points in direction of object orientation
  - Length of vector is probability that object exists

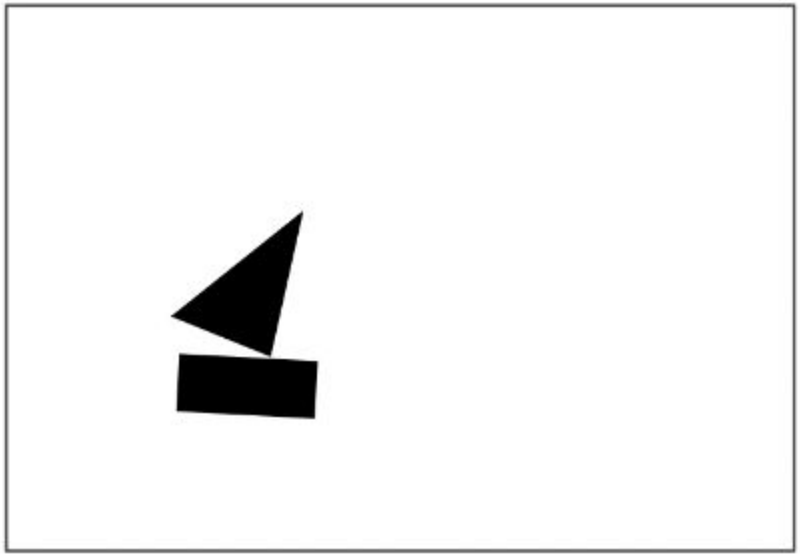
Capsule networks

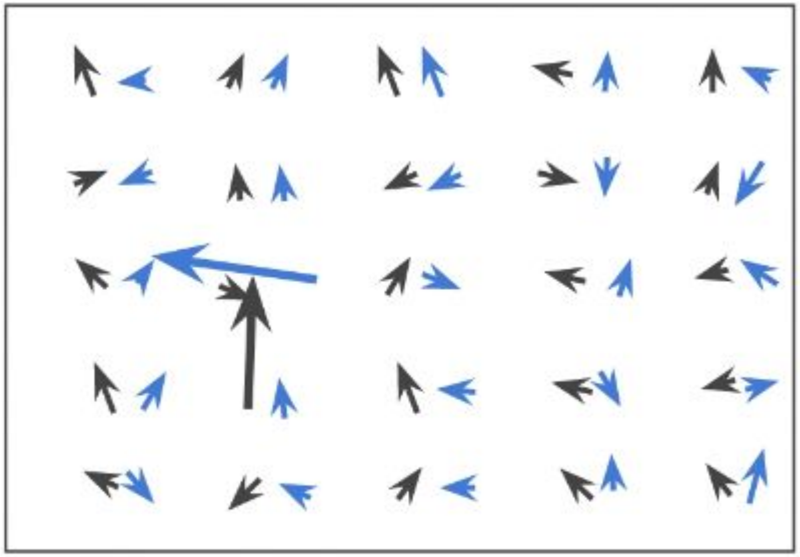
- Neurons store information as vectors
- Vectors store pose information
  - Vector points in direction of object orientation
  - Length of vector is probability that object exists



Capsule networks

- Neurons store information as vectors
- Vectors store pose information
  - Vector points in direction of object orientation
  - Length of vector is probability that object exists



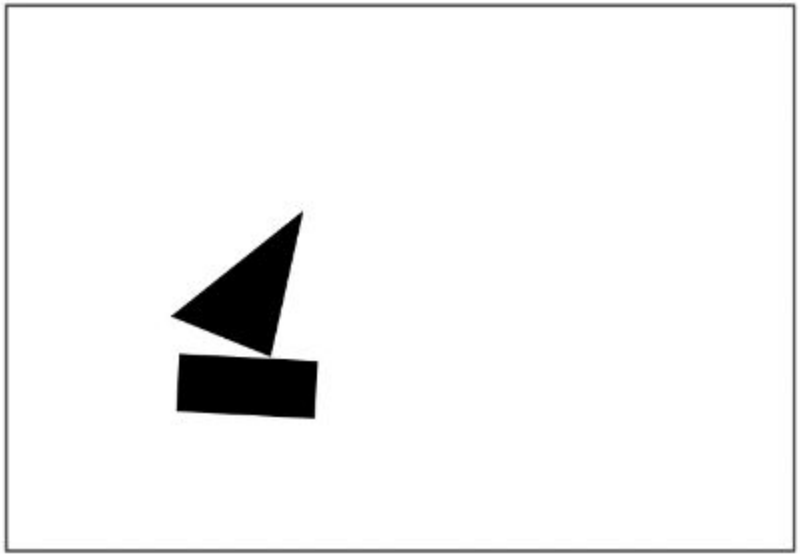


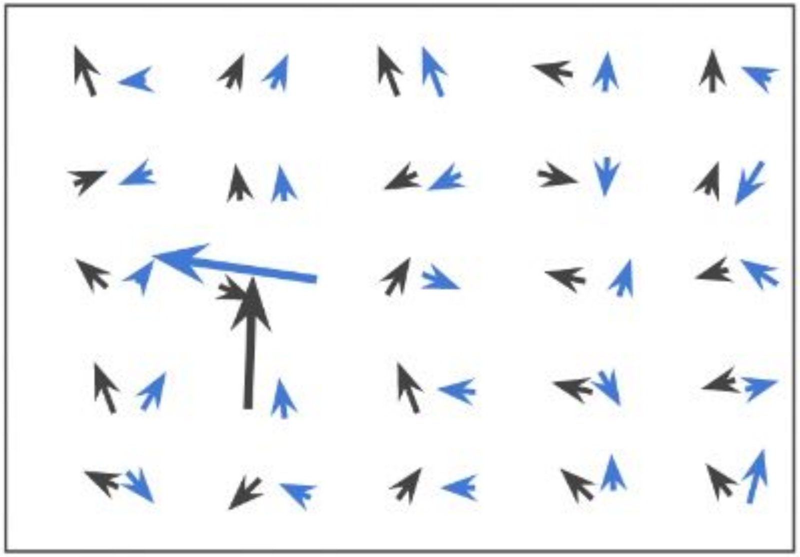
Capsule networks

- Routing by agreement

Capsule networks

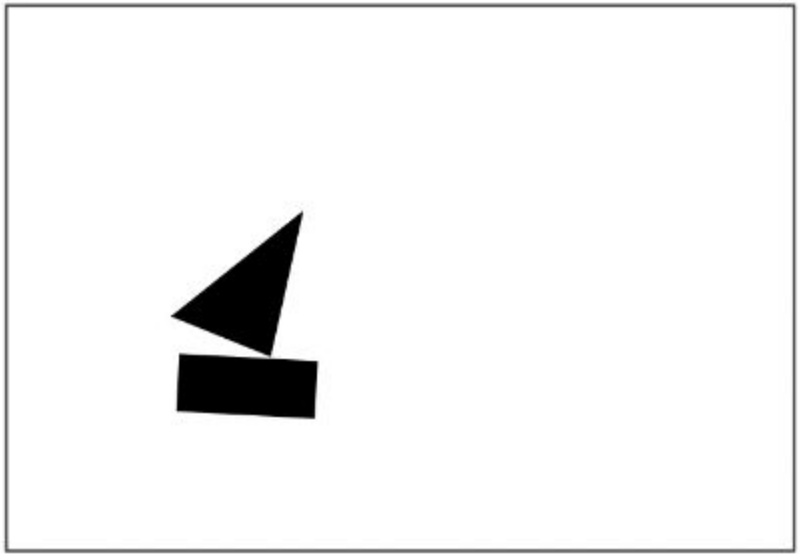
- Routing by agreement

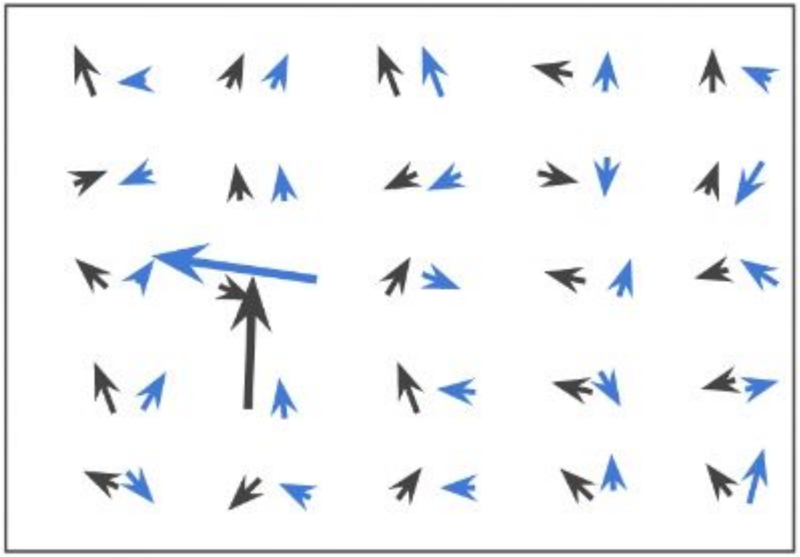




Capsule networks

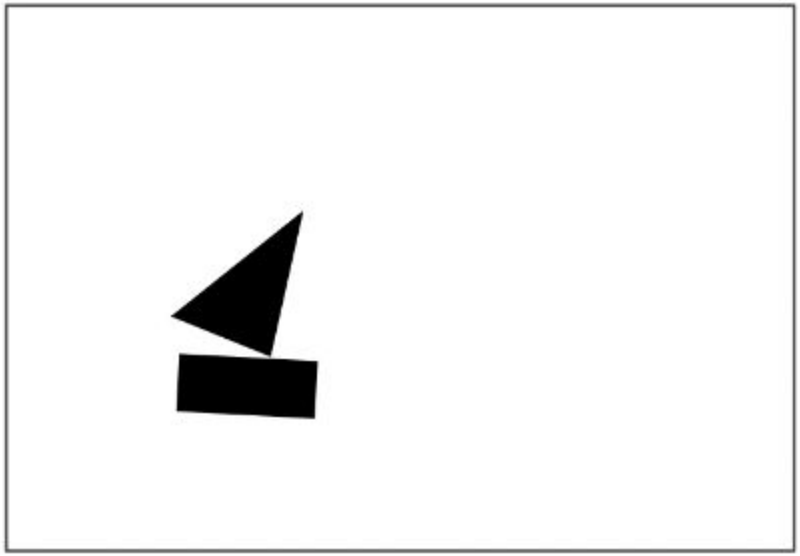
- Routing by agreement
  - Image segmentation?

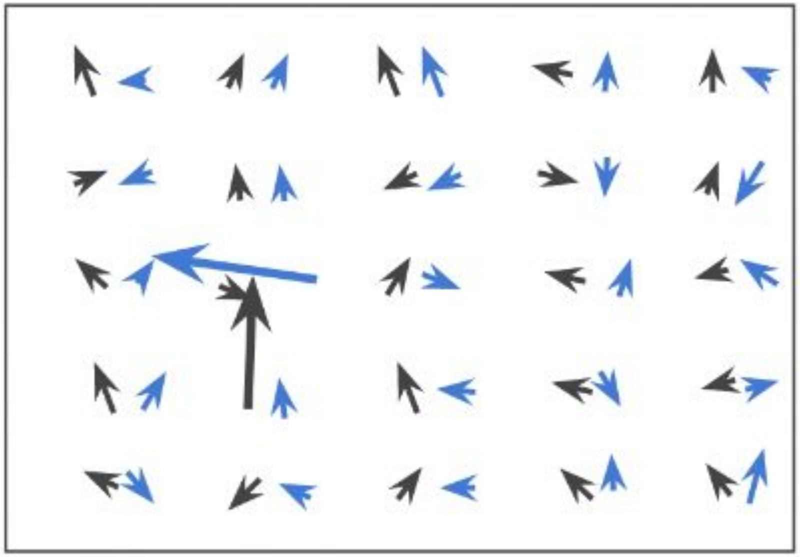




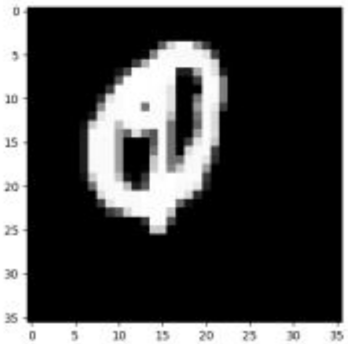
Capsule networks

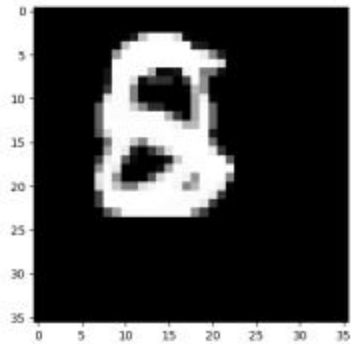
- Routing by agreement
  - Image segmentation!

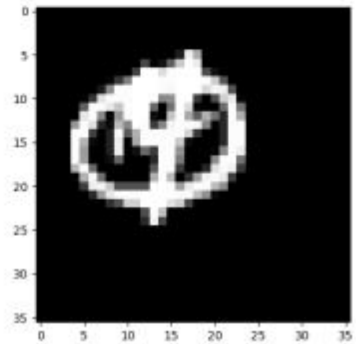




MultiMNIST

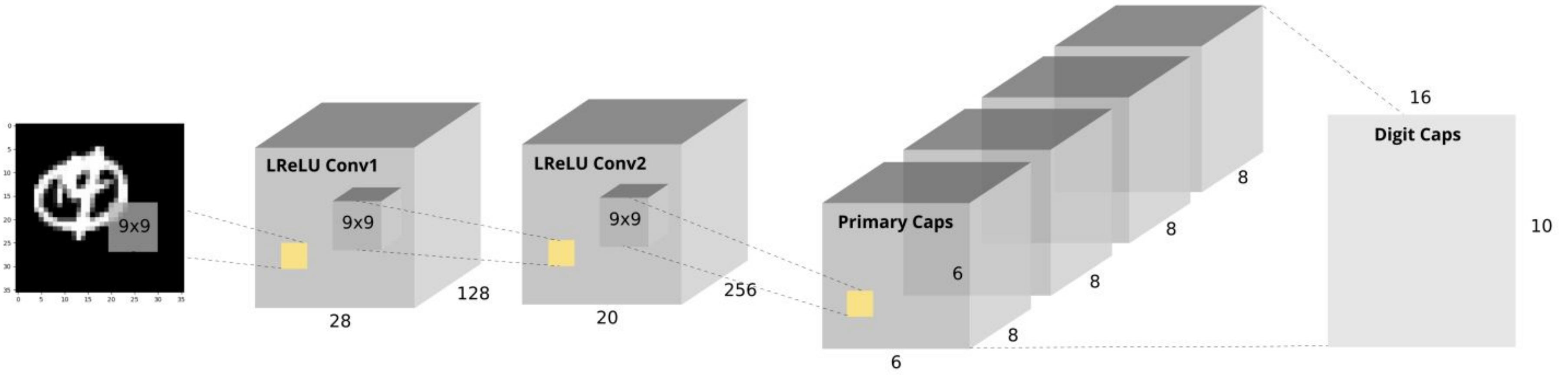






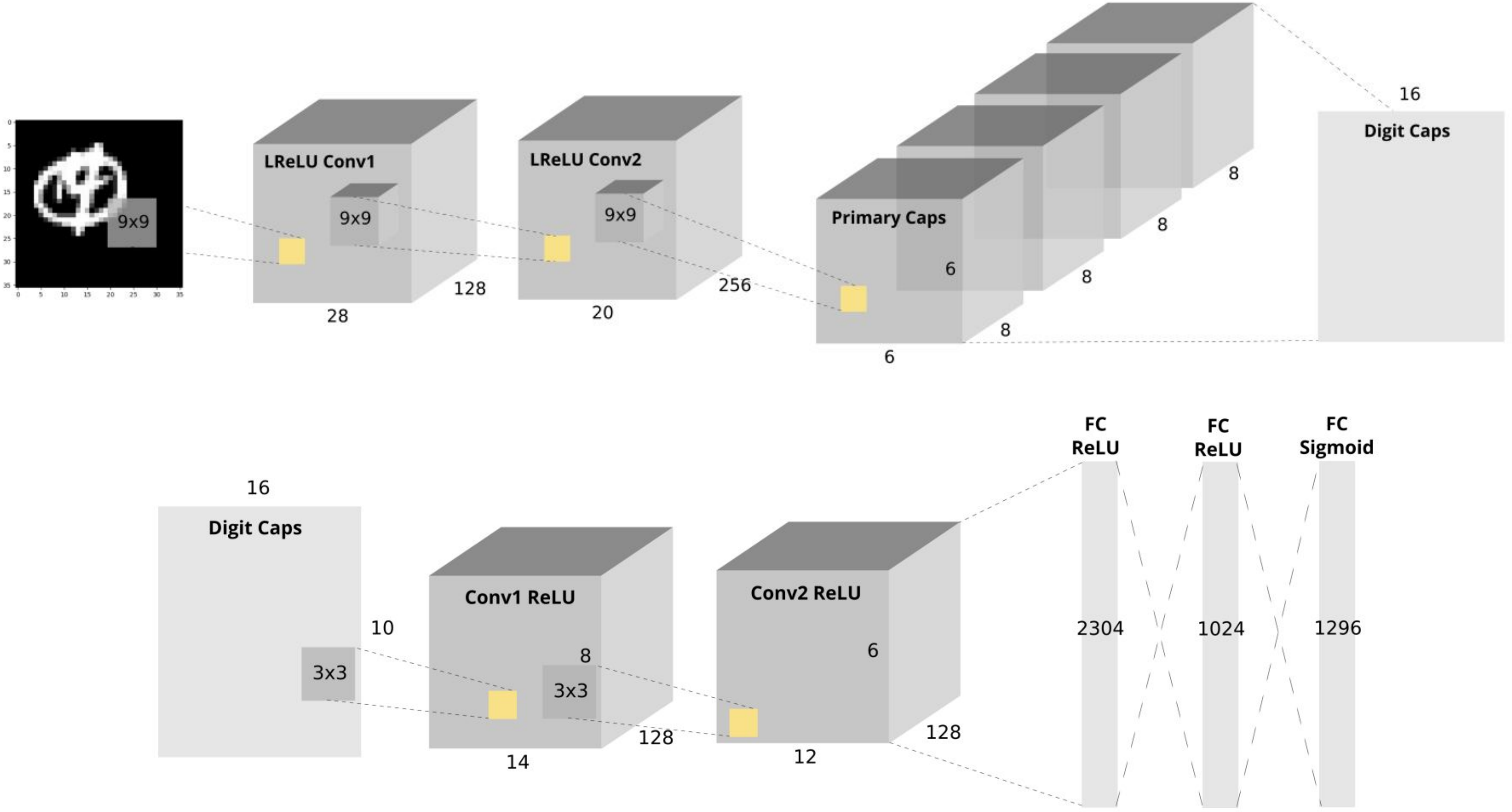
MultiMNIST

MultiMNIST



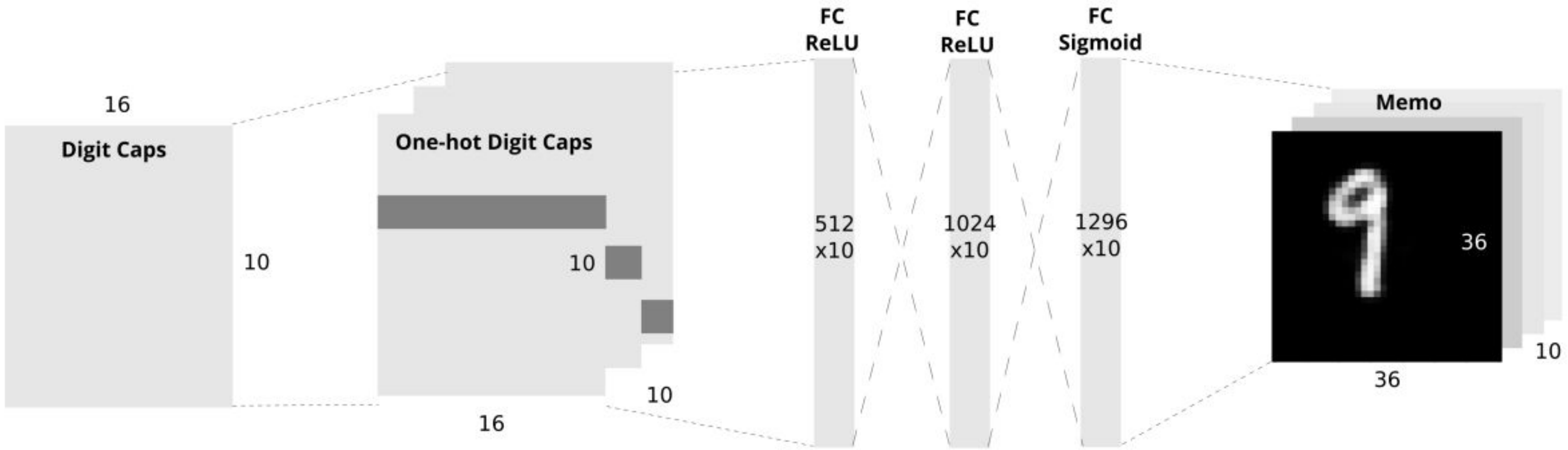
9x9
LReLU Conv1
9x9
128
28
LReLU Conv2
9x9
256
20
Primary Caps
6
8
6
8
8
8
16
Digit Caps
10

MultiMNIST

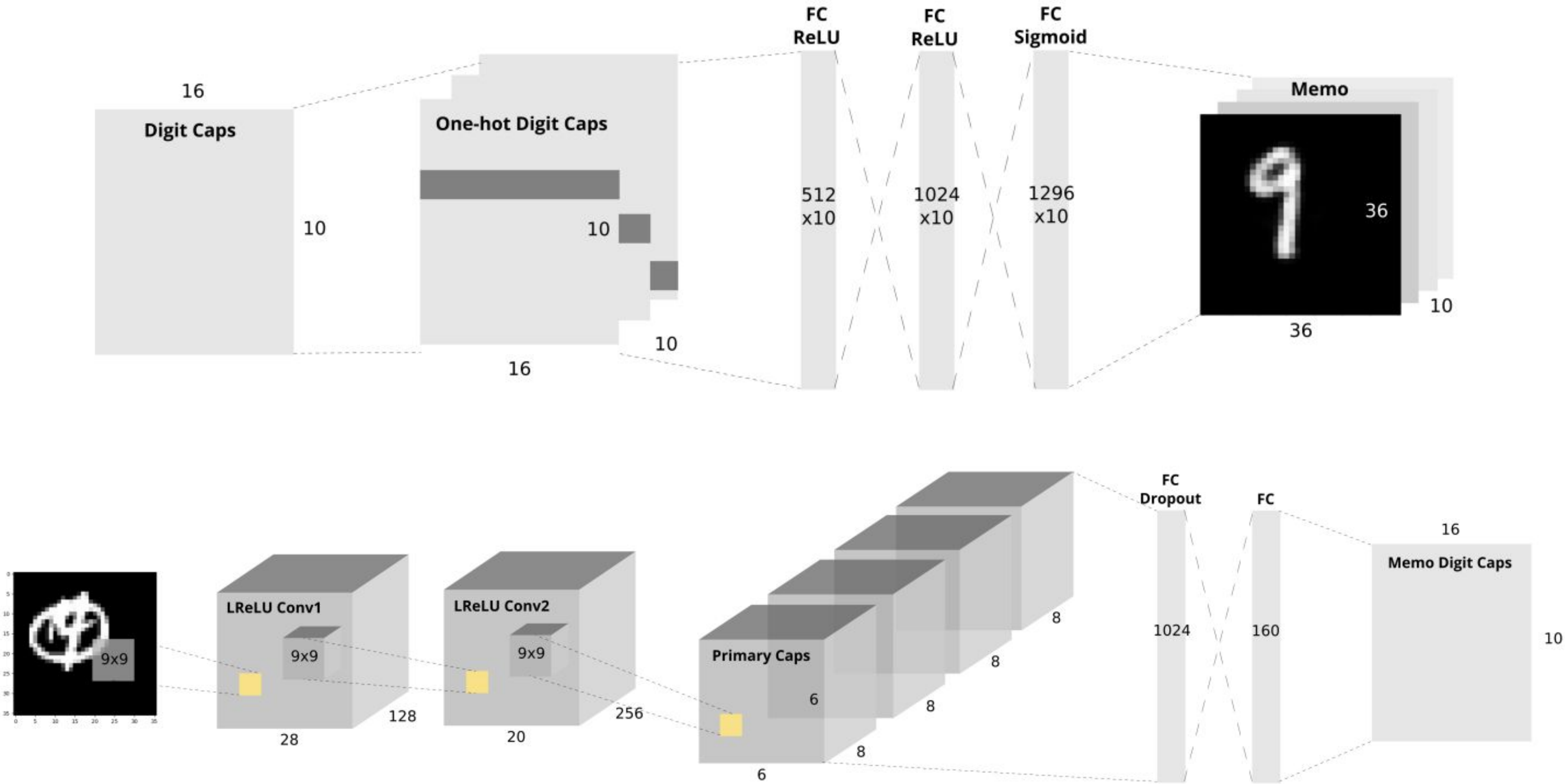


9x9
LReLU Conv1
9x9
128
28
LReLU Conv2
9x9
256
20
Primary Caps
6
8
6
8
8
8
16
Digit Caps
10
16
Digit Caps
10
3x3
Conv1 ReLU
8
3x3
128
14
Conv2 ReLU
6
128
12
FC
ReLU
2304
FC
ReLU
1024
FC
Sigmoid
1296

MultiMNIST


16
Digit Caps
10
One-hot Digit Caps
10
10
16
FC
ReLU
512
x10
FC
ReLU
1024
x10
FC
Sigmoid
1296
x10
Memo
36
36
10

MultiMNIST



16
Digit Caps
10
One-hot Digit Caps
10
10
16
FC
ReLU
512
x10
FC
ReLU
1024
x10
FC
Sigmoid
1296
x10
Memo
36
36
10
9x9
LReLU Conv1
9x9
128
28
LReLU Conv2
9x9
256
20
Primary Caps
6
6
8
8
8
8
FC
Dropout
1024
FC
160
16
Memo Digit Caps
10

Transfer learning

Transfer learning

- Use a model pre-trained on one dataset to learn another dataset

Transfer learning

- Use a model pre-trained on one dataset to learn another dataset
- subMMNIST dataset

Transfer learning

- Use a model pre-trained on one dataset to learn another dataset
- subMMNIST dataset
  - The MMNIST dataset, but without one digit

Transfer learning

- Use a model pre-trained on one dataset to learn another dataset
- subMMNIST dataset
  - The MMNIST dataset, but without one digit
- The idea:
  - Train on subMMNIST
  - Load full MMNIST dataset
  - See how the network does

The three networks

- Regular convolutional network

The three networks

- Regular convolutional network
- Regular capsule network

The three networks

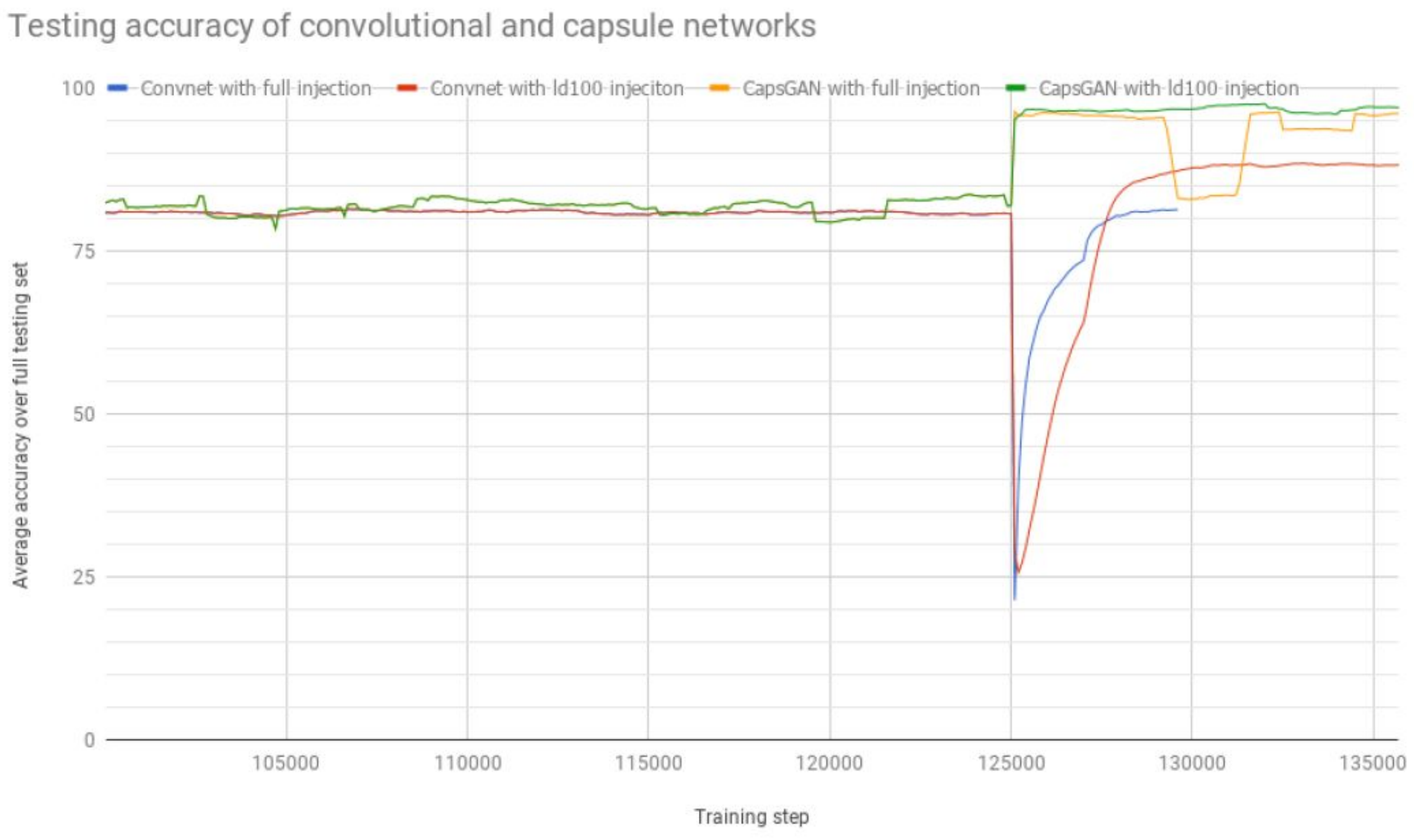
- Regular convolutional network
- Regular capsule network
- Generative capsule network (CapsGAN)

Experiment I: CapsGAN vs Convnet

- Injection after 125,000 iterations

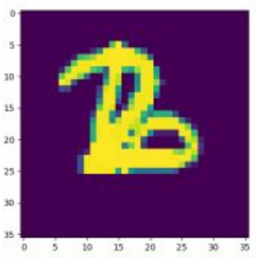
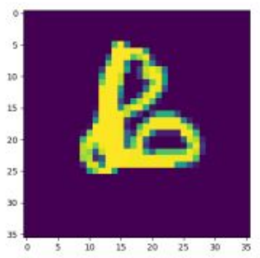
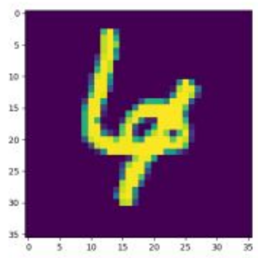
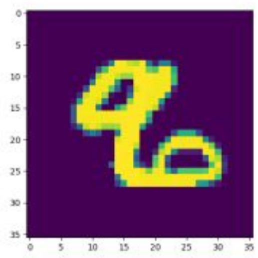
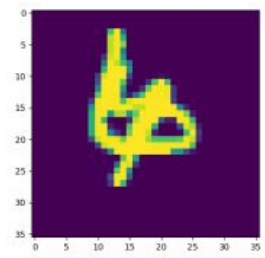
Architecture	Iterations to reach initial accuracy	Pre-injection accuracy	Peak accuracy on full test set with full injection	Peak accuracy on full test set with ld-100 injection
Convolutional	2700	80.7%	<82%	88.4%
Generative capsule	**<100**	81.9%	**96.3%**	**97.5%**

Experiment I: CapsGAN vs Convnet

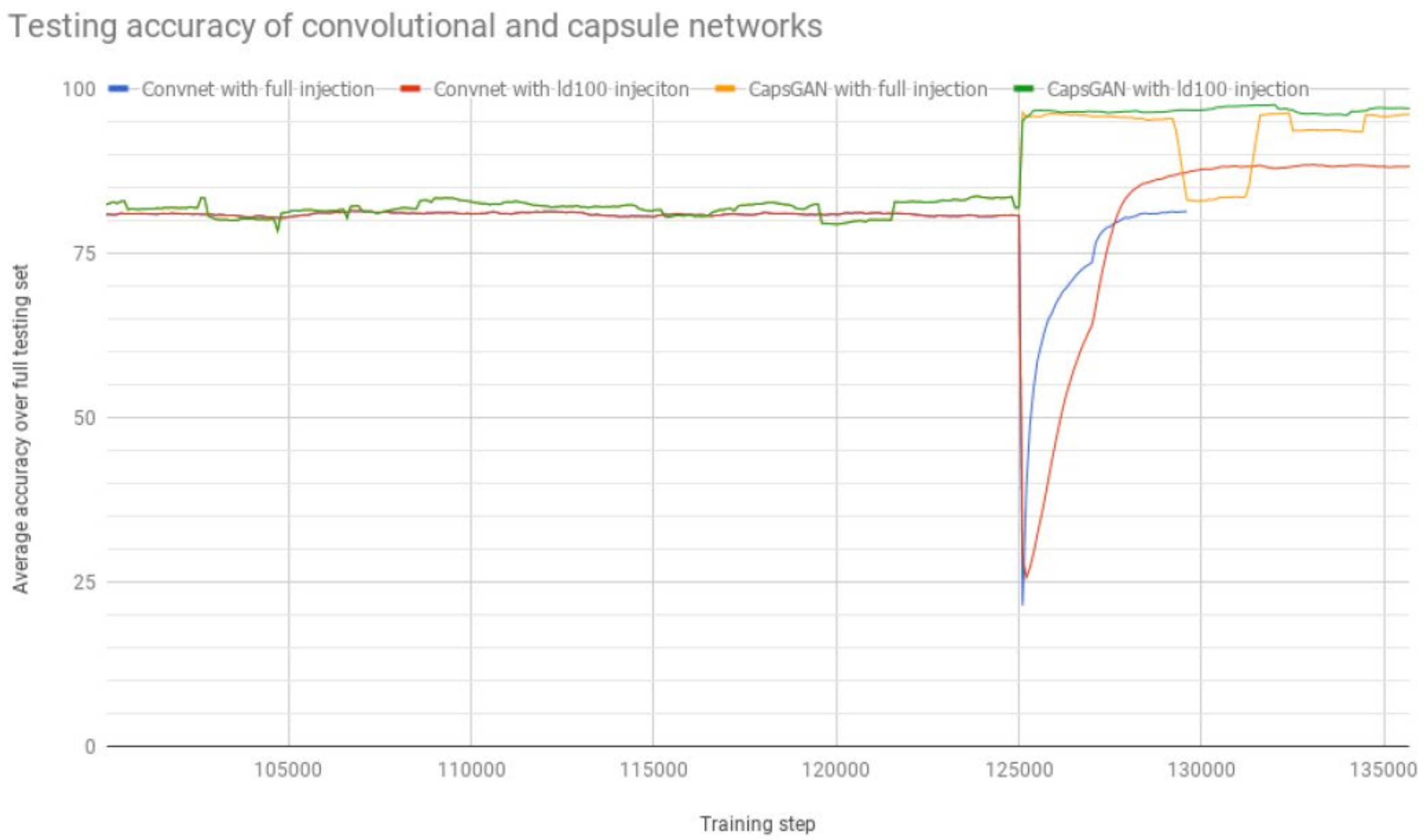


The LD dataset

- Use a small number of new examples

Experiment I: CapsGAN vs Convnet



Experiment I: CapsGAN vs Convnet

- Injection after 125,000 iterations

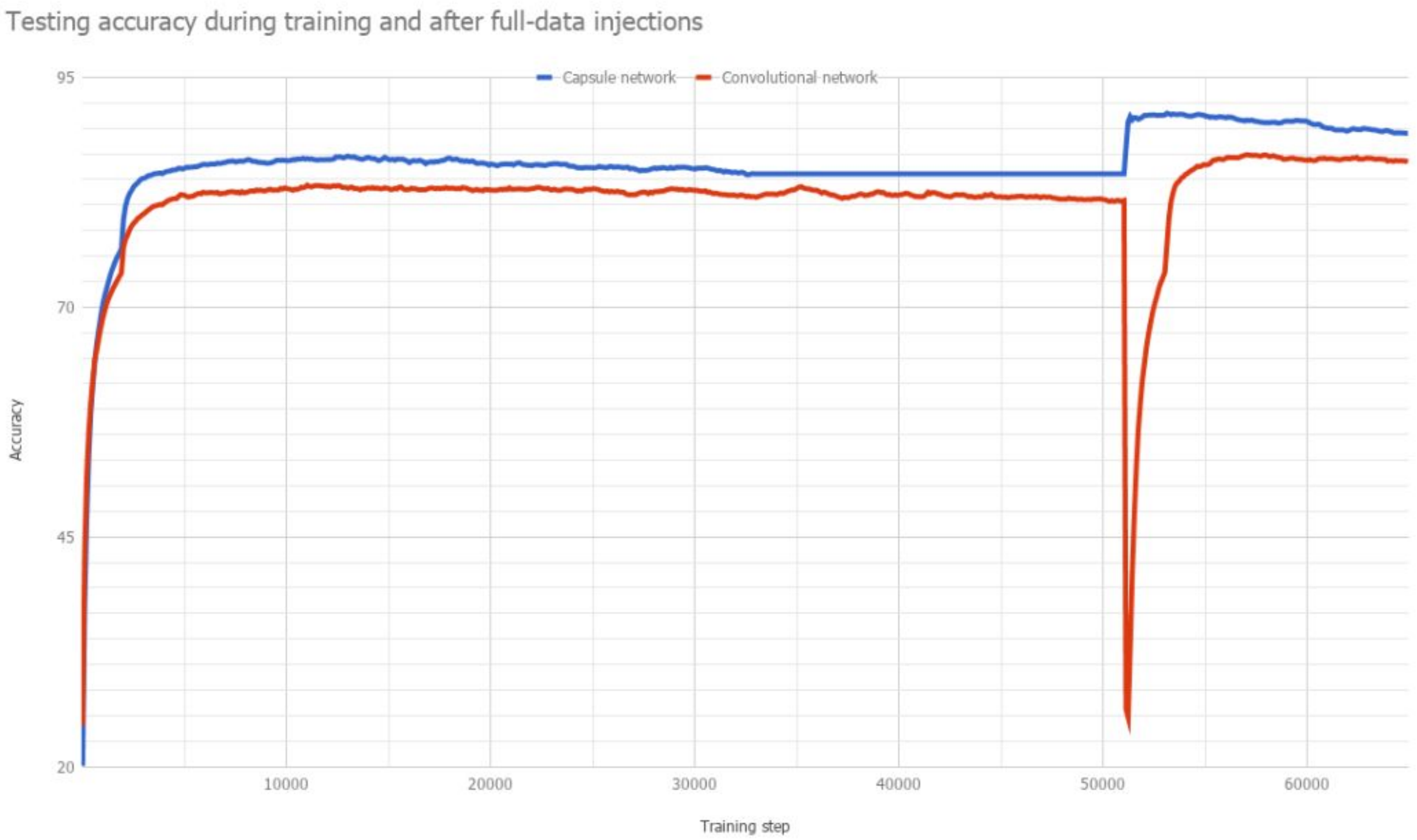
Architecture	Iterations to reach initial accuracy	Pre-injection accuracy	Peak accuracy on full test set with full injection	Peak accuracy on full test set with ld-100 injection
Convolutional	2700	80.7%	<82%	88.4%
Generative capsule	**<100**	81.9%	**96.3%**	**97.5%**

Experiment II: Capsnet vs Convnet

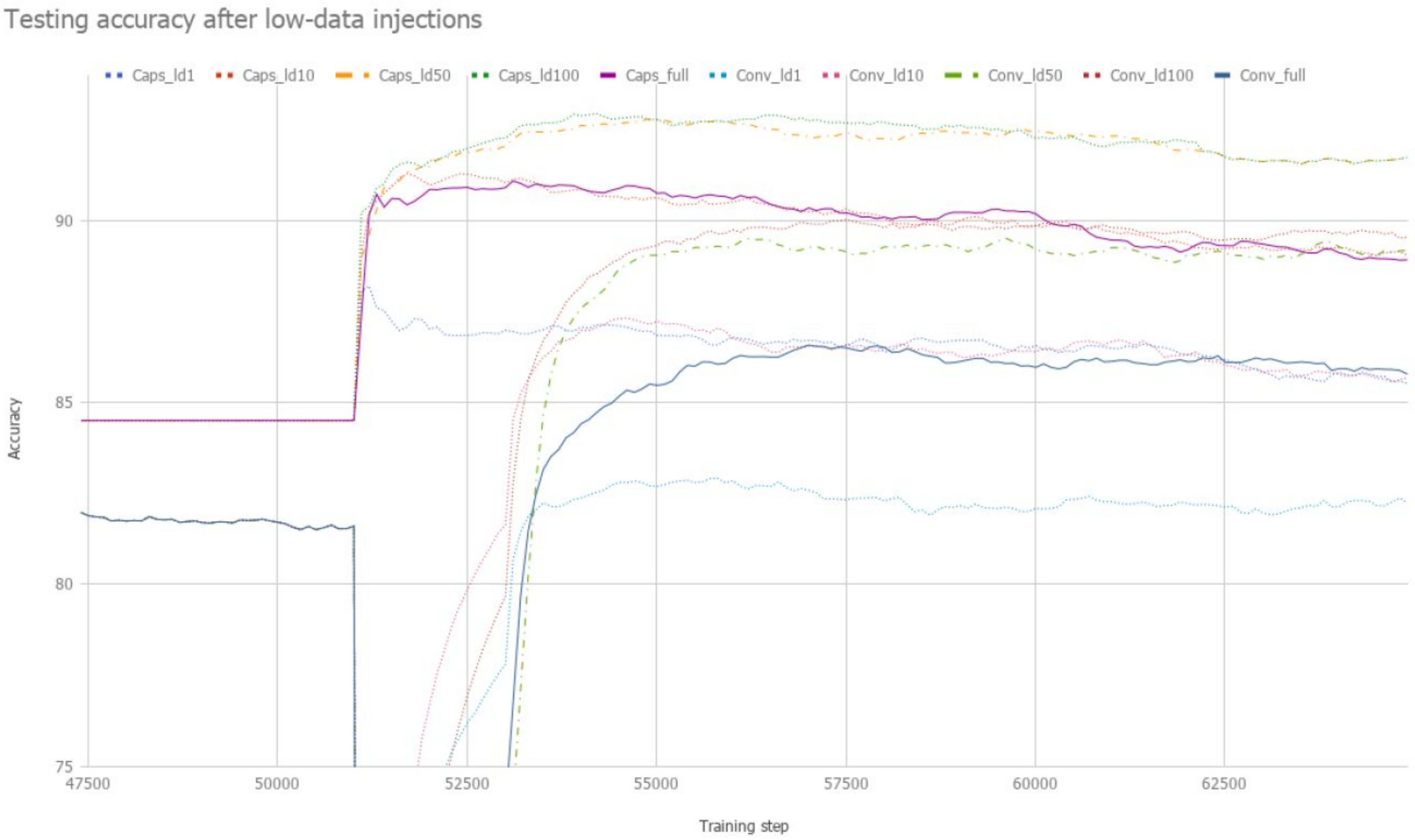
- Injection after ~50k iterations

			Peak accuracy on dataset after injection				
Architecture	Iterations to reach initial accuracy	Pre-injection accuracy	ld-1	ld-10	ld-10	ld-100	full
Convolutional	2300	81.4%	82.9%	87.3%	89.5%	90.0%	86.5%
Capsule	<100	84.5%	88.2%	91.3%	92.8%	93.0%	91.1%

Experiment II: Capsnet vs Convnet



Experiment II: Capsnet vs Convnet



Experiment II: Capsnet vs Convnet

- Injection after ~50k iterations

			Peak accuracy on dataset after injection				
Architecture	Iterations to reach initial accuracy	Pre-injection accuracy	ld-1	ld-10	ld-10	ld-100	full
Convolutional	2300	81.4%	82.9%	87.3%	89.5%	90.0%	86.5%
Capsule	<100	84.5%	88.2%	91.3%	92.8%	93.0%	91.1%

The Unused Capsule effect

- During training, nine of ten pathways are used

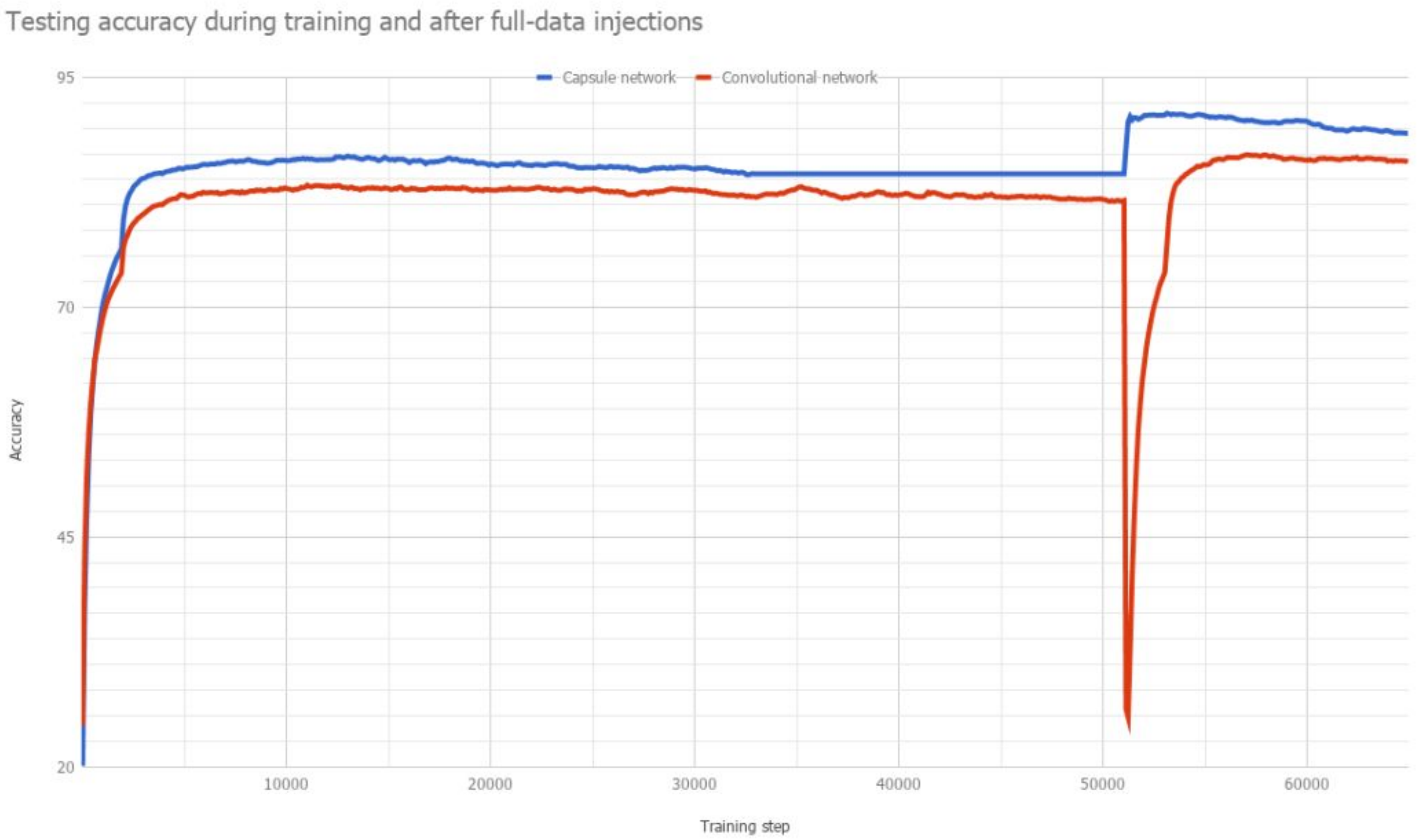
The Unused Capsule effect

- During training, nine of ten pathways are used
- Network recognizes that new data does not fit the existing pathways

The Unused Capsule effect

- During training, nine of ten pathways are used
- Network recognizes that new data does not fit the existing pathways
- Tenth pathway is now used

Experiment II: Capsnet vs Convnet



Future directions

- Dynamic addition of pathways

Future directions

- Dynamic addition of pathways
- Automated “guided” learning
  - Pre-injection is 7 p.p. less
  - Post-injection is 2.5 p.p. better

Future directions

- Dynamic addition of pathways
- Automated “guided” learning
  - Pre-injection is 7 p.p. less
  - Post-injection is 2.5 p.p. Better
- How much data is best?

Future directions

- Dynamic addition of pathways
- Automated “guided” learning
  - Pre-injection is 7 p.p. less
  - Post-injection is 2.5 p.p. Better
- How much data is best?
- More advanced tasks

Acknowledgements

- Slava Gerovitch, Pavel Etingof, Tanya Khovanova, Srinivas Devadas and the MIT PRIMES program
- Maksym Korablyov and Dr. Joseph Jacobson
- My parents

Questions?