Lab 3 Report

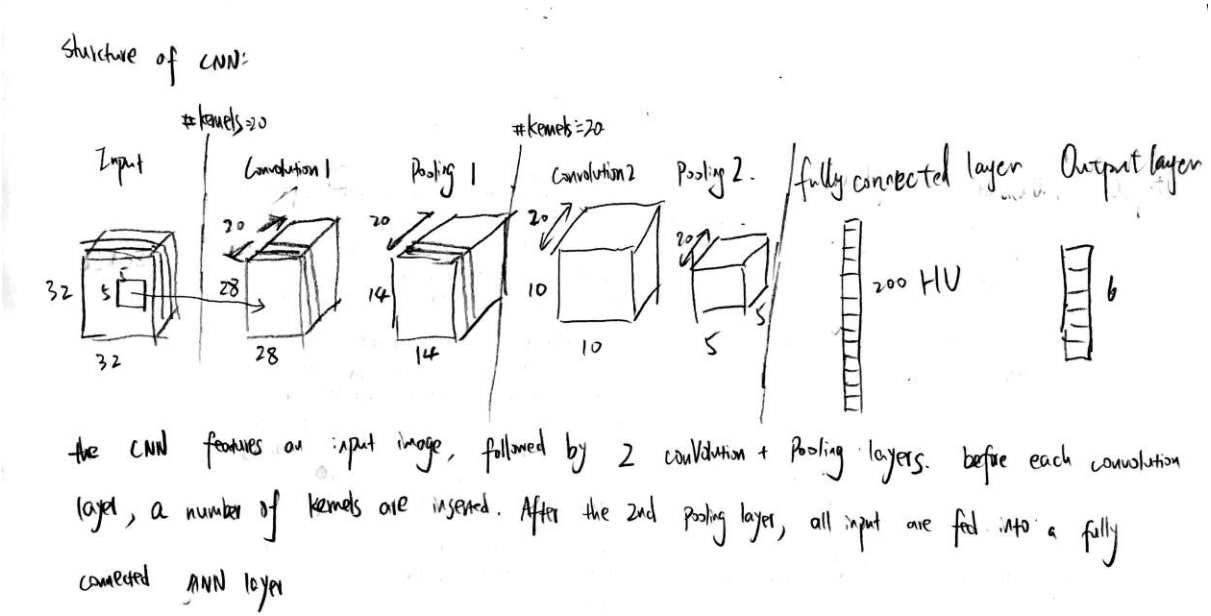
Zhanpeng Zeng (zzeng38)
Qinyuan Sun (qsun28)
Jingyi Zhao (jzhao62)

Tasks:

Zhanpeng Zeng: Program Implementation, Program Framework design
Qinyuan Sun: Program Implementation, Mathematical Derivation of BackProp
Jingyi Zhao: Experimental design and evaluation

Network Configuration Detail

The deep net structure is shown in figure below. We used convolution-maxpooling-convolution-maxpooling-fullyconnected structure. For convolution and max-pooling layers, we output 20 plates. For fully-connnected layer, we used 200 hidden units with leaky rectified linear activation function. We use 5-by-5 filters for convolution layer and applied none zero padding and stride 0 convolution in forwardprop. Each plate in max-pooling operates on corresponding convolution plate. Input of convolution layer comes from all the max-pooling plate. For backprop, we applied the simple gradient descent. In addition, we implemented dropouts for all layers.



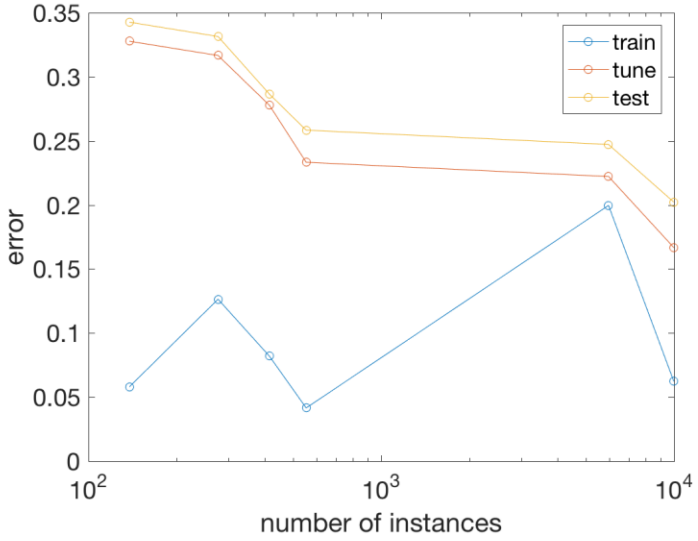
We ran multiple experiments. Our best result come from the following setting. We used 9980 samples in total and divided them in the same way. The extra samples are generated using provided code. Furthermore, we balanced the number of samples for each class. No dropout is used.

By inspecting the confusion matrix, we observed that most error are produced by predicting starfish to flower. Since we use 32-by-32 images, most details are lost. We might rely on shape, color or edges of the image. By inspecting the image, we notice that some of the starfish are very similar to flowers in terms of color and shape.

	Airplanes (predicted)	Butterfly (predicted)	Flower (predicted)	Piano (predicted)	Starfish (predicted)	Watch (predicted)
Airplanes (actual)	37	0	0	0	2	2
Butterfly (actual)	0	8	2	2	3	3
Flower (actual)	0	0	30	0	7	0
Piano (actual)	2	0	0	15	1	1
Starfish (actual)	0	1	0	3	11	2
Watch (actual)	2	2	1	0	0	41

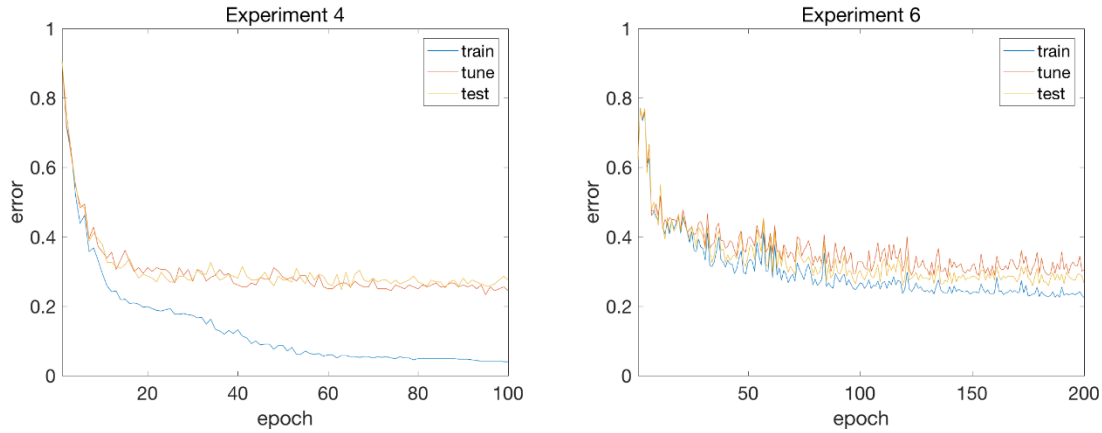
Learning Curve based on the number of training sample used

Note that due to the training speed, the experiment with 5972 instances only run 20 epochs, and the experiment with 9980 instances only run 18 epochs, so we do not know whether the models converge or not. However, in general, as more training samples, test error and tuning error dropped. The training error does not decrease consistently due to early stopping. As a result, with more samples, we are expecting even better results. This is reasonable since as the number of training set increases, the target concept is better represented by the training instances. Thus, the model can more accurately represent the target concept.

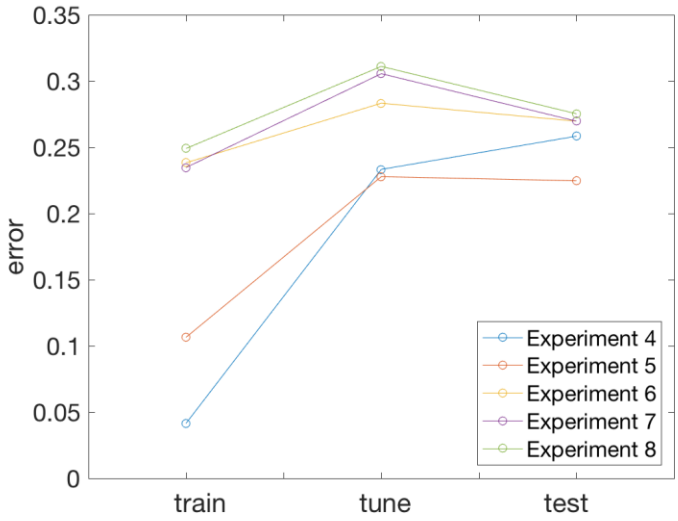


Experiments with dropout

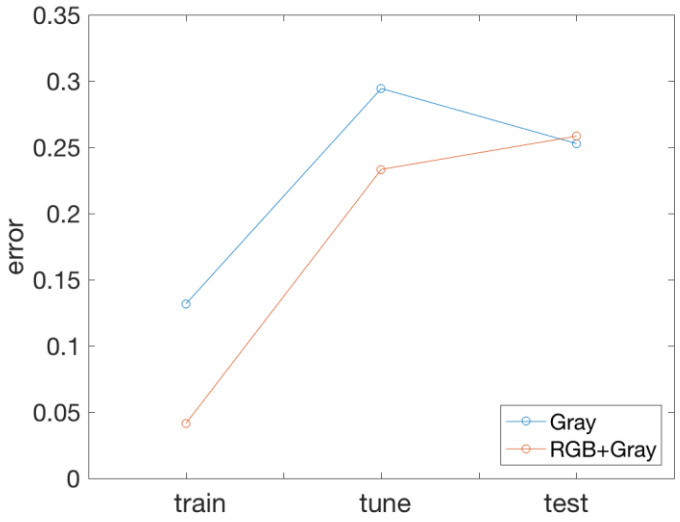
We tried couple different dropout configurations. Detailed configurations refer to list of deepnet configuration in appendix. As the two figures below shown, when dropout is applied, the learning curve is no longer smooth, but the decreasing trend is still there. However, the networks with dropout have slower convergence rate than network without dropout. Note that the left figure shows that after 40 epochs, the network is almost converged, but in the right figure, the error rate continues to decrease in 200 epochs. In general, dropout has higher error rate, but it provides a way to regularize the network and prevent overfitting. The dropout prevents models from heavily relying on certain features.



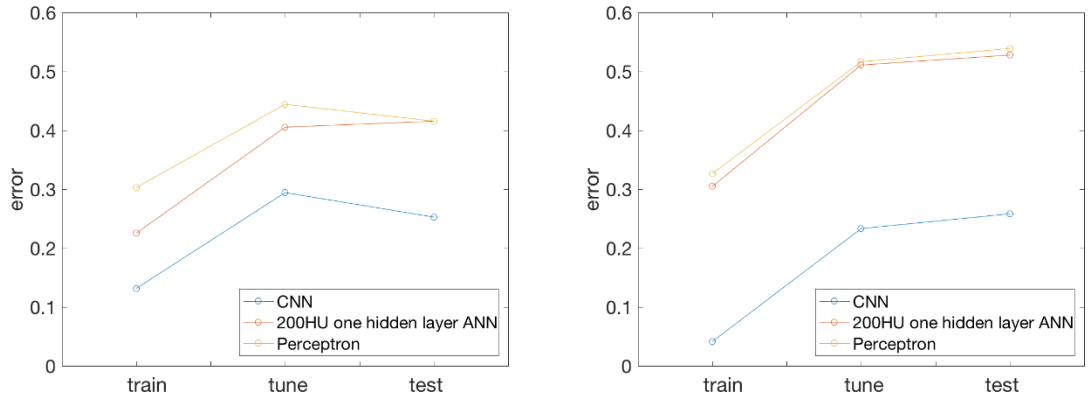
The following figure displays the performance of network with dropouts. Experiment 4 is the baseline (without drop out), the following experiments have increasing degree of dropout rate. By inspecting the learning rate plots shown in appendix, the convergence of network using dropout is slower. After 200 epochs, the error rate is still decreasing. Networks that employ dropout have greater generalization power. The power of dropout is capped at certain threshold. Experiment 5 has appropriate level of dropout rate and outperform baseline. However, excessive amount of dropout hurts the performance.



Experiments with input channels and different learning models



This figure compares the performance of CNN on Gray and RGB+Gray inputs. Even though color inputs have lower training and tuning error, they have similar testing error. This result counters our expectation that with color input we should perform better.



These two graph represents the performance of each model on gray images (left) and RGB+Gray images(right). With the increase of other color sources, ANN and perceptron's performance decreases while CNN increases, which also counter our expectation that the training error all models should be smaller. Possible explanation would be that colored images do not provide more information needed for ANN and perceptron training. In addition, note that as the error rate vs epoch figure in Appendix shown, the ANN training has not converged yet, so with more training epochs, the error rate may be smaller.

The comparison among three models shows that convolution network outperforms one hidden layer ANN and perceptron. The reason may be that for the specific task of processing images, CNN encode spatial information while perceptron and ANN do not.

Appendix

List of Deepnet Configuration

Experiment id	input dropout	first conv dropout	first pooling dropout	second conv dropout	second pooling dropout	fully connected dropout	Use RGB	num samples
1	0	0	0	0	0	0	yes	138
2	0	0	0	0	0	0	yes	227
3	0	0	0	0	0	0	yes	415
4	0	0	0	0	0	0	yes	554
5	0	0	0.2	0	0.2	0.5	yes	554
6	0	0	0.5	0	0.5	0.5	yes	554
7	0	0.2	0.5	0.2	0.5	0.5	yes	554
8	0.2	0.2	0.5	0.2	0.5	0.5	yes	554
9	0	0	0	0	0	0	no	554
10	0	0	0	0	0	0	yes	9980
11	0	0	0	0	0	0	yes	5972

List of Baseline

Experiment id	Hidden units used	Use RGB
12	200	no
13	0	no
14	200	yes
15	0	yes

List of Confusion Matrix for experiments

Experiment 1	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	36	0	1	1	0	3
butterfly(actual)	6	0	8	0	1	3
flower(actual)	0	2	31	2	0	2
piano(actual)	1	0	2	16	0	0
starfish(actual)	1	0	4	5	3	4
watch(actual)	4	0	10	1	0	31

Experiment 2	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	38	1	0	1	0	1
butterfly(actual)	7	2	6	0	0	3
flower(actual)	3	0	26	1	0	7
piano(actual)	1	0	2	14	0	2
starfish(actual)	1	0	4	3	0	9
watch(actual)	2	1	3	1	0	39

Experiment 3	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	39	0	0	0	0	2
butterfly(actual)	5	1	3	0	4	5
flower(actual)	1	1	29	1	3	2
piano(actual)	2	0	0	16	0	1
starfish(actual)	0	0	6	2	3	6
watch(actual)	3	0	2	1	1	39

Experiment 4	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	36	1	1	0	0	3
butterfly(actual)	2	9	3	0	3	1
flower(actual)	0	2	30	1	2	2
piano(actual)	1	1	1	13	2	1
starfish(actual)	1	3	3	1	5	4
watch(actual)	0	1	2	1	3	39

Experiment 5	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	39	0	1	0	0	1
butterfly(actual)	1	4	8	0	0	5
flower(actual)	0	0	36	0	0	1
piano(actual)	1	0	3	15	0	0
starfish(actual)	0	0	11	2	2	2
watch(actual)	0	0	4	0	0	42

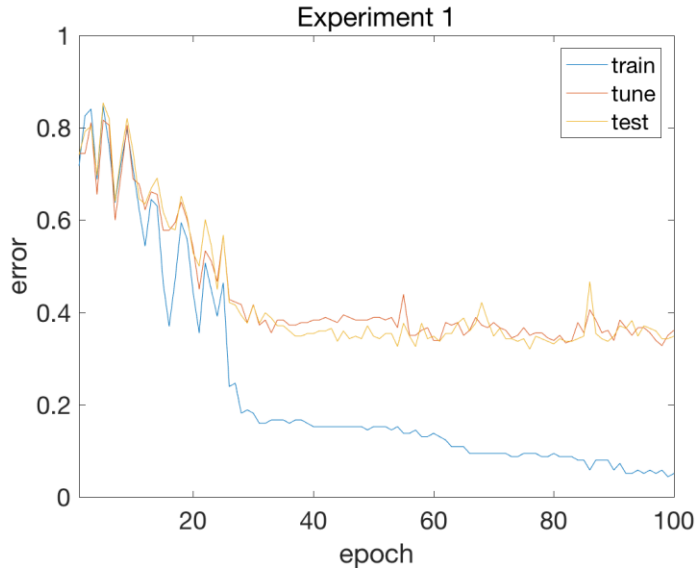
Experiment 6	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	41	0	0	0	0	0
butterfly(actual)	6	0	7	0	0	5
flower(actual)	1	0	36	0	0	0
piano(actual)	3	0	1	11	0	4
starfish(actual)	1	0	5	2	0	9
watch(actual)	0	0	4	0	0	42

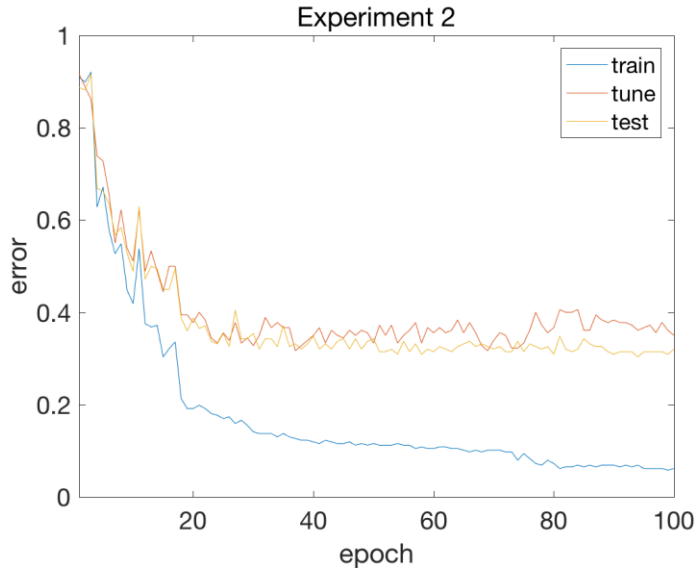
Experiment 7	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	40	0	1	0	0	0
butterfly(actual)	4	0	4	0	0	10
flower(actual)	0	0	36	0	0	1
piano(actual)	2	0	3	11	0	3
starfish(actual)	0	0	5	0	0	12
watch(actual)	1	0	2	0	0	43

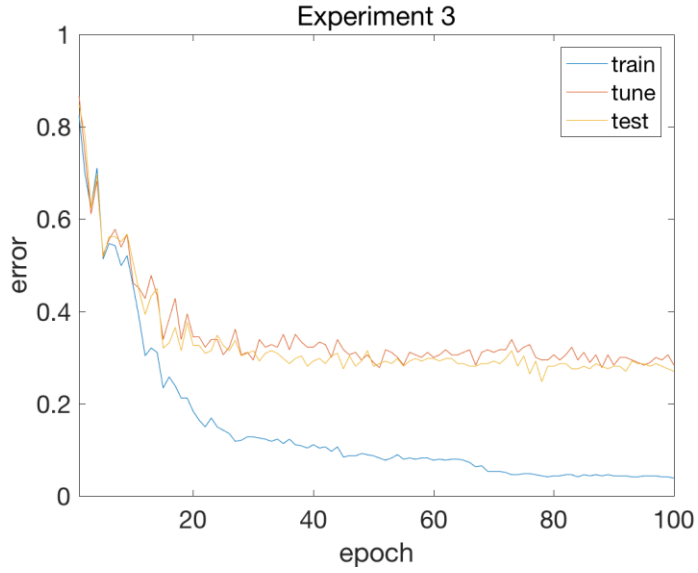
Experiment 8	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	41	0	0	0	0	0
butterfly(actual)	3	0	7	1	0	7
flower(actual)	0	0	36	0	0	1
piano(actual)	3	0	5	11	0	0
starfish(actual)	1	0	8	1	0	7
watch(actual)	2	0	3	0	0	41

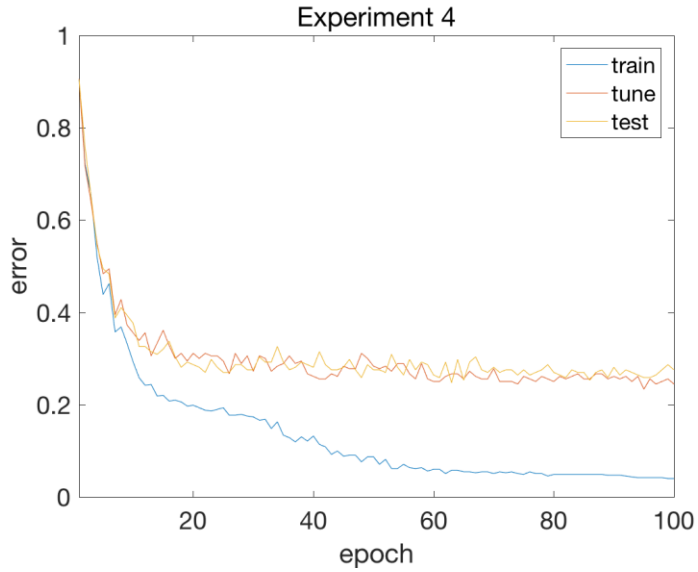
Experiment 9	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	37	0	2	0	0	2
butterfly(actual)	2	8	5	0	0	3
flower(actual)	2	1	32	0	1	1
piano(actual)	2	0	2	14	0	1
starfish(actual)	3	2	6	2	4	0
watch(actual)	1	3	3	0	1	38

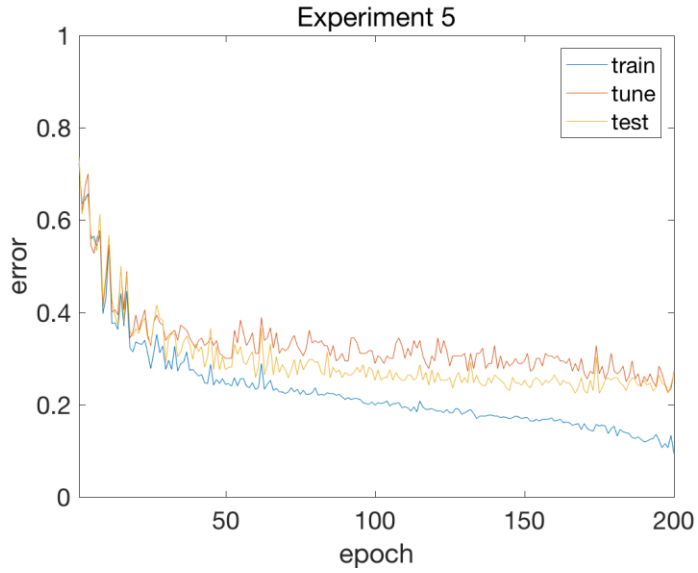
Experiment 10	Airplane(predict)	butterfly(predict)	flower(predict)	piano(predict)	starfish(predict)	watch(predict)
Airplane(actual)	38	0	0	1	0	0
butterfly(actual)	0	11	2	1	1	4
flower(actual)	0	1	27	0	1	0
piano(actual)	1	2	2	16	3	1
starfish(actual)	1	2	6	0	11	0
watch(actual)	1	2	0	1	1	41

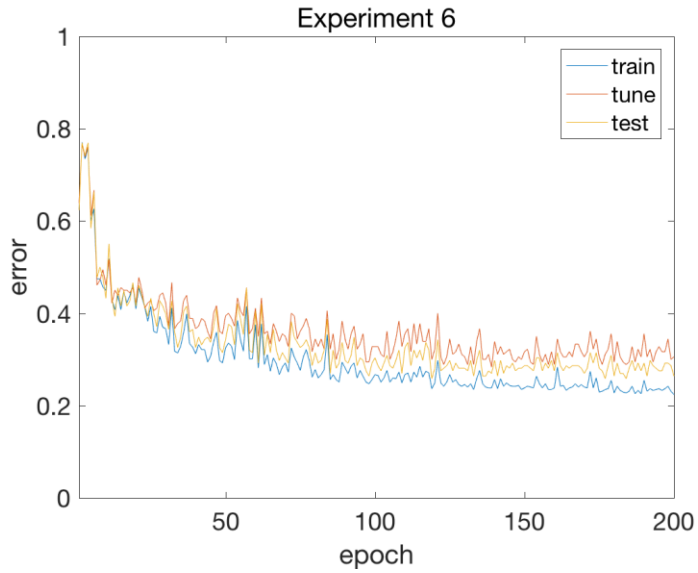

Experiment 1
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100

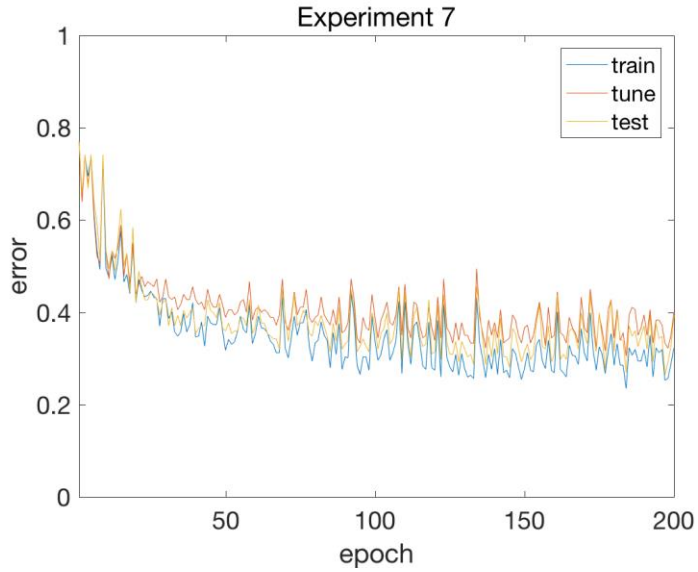

Experiment 2
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100

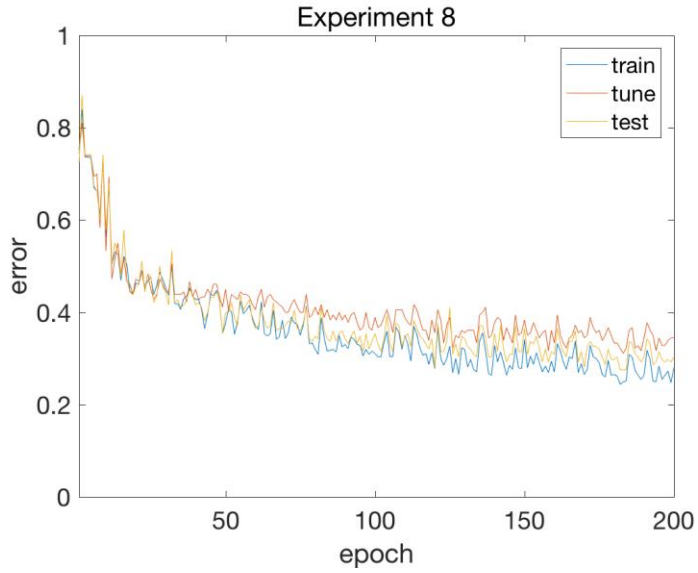

Experiment 3
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100

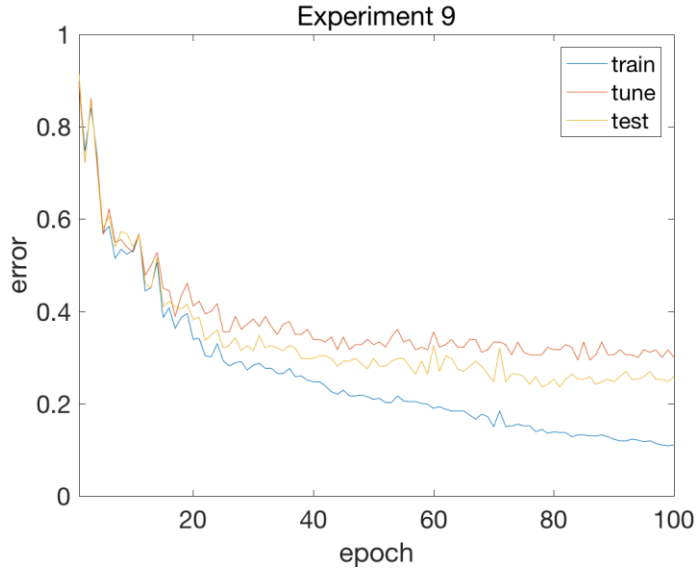

Experiment 4
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100

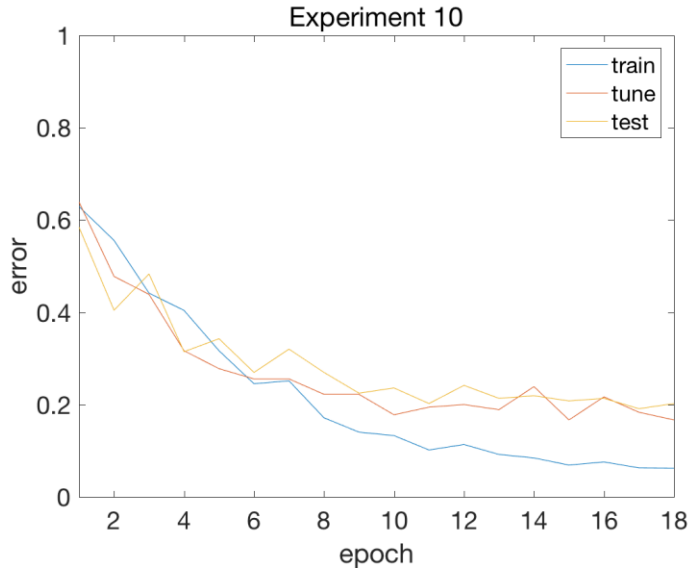

Experiment 5
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
50
100
150
200

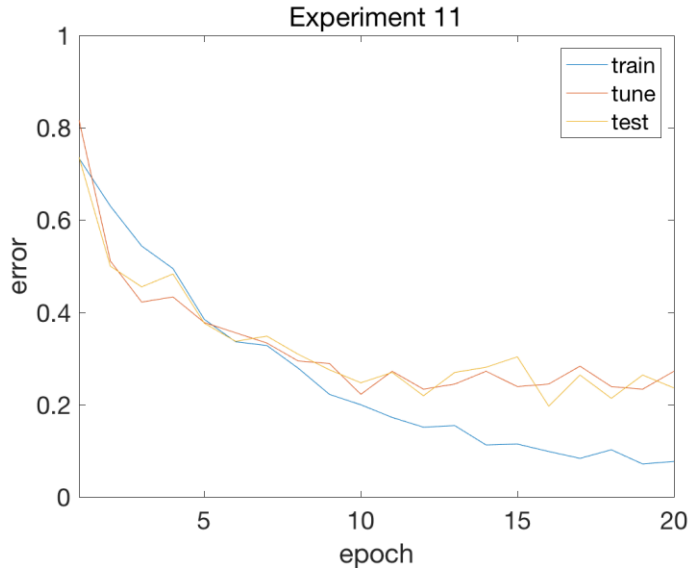

Experiment 6
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
50
100
150
200

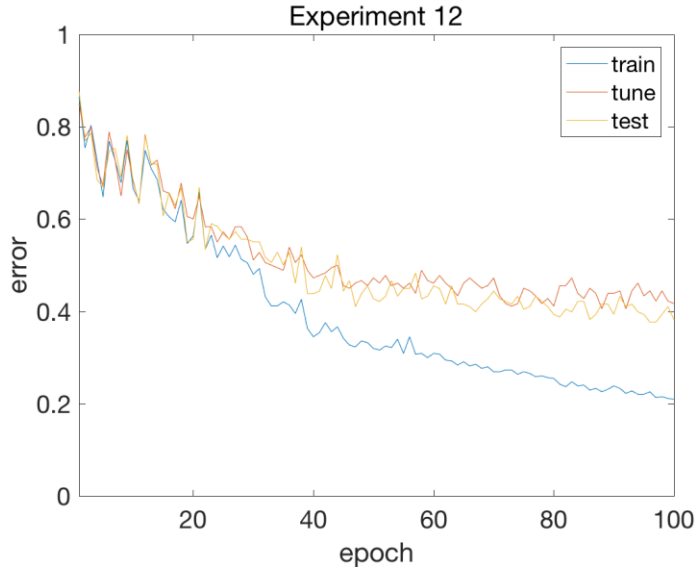

Experiment 7
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
50
100
150
200

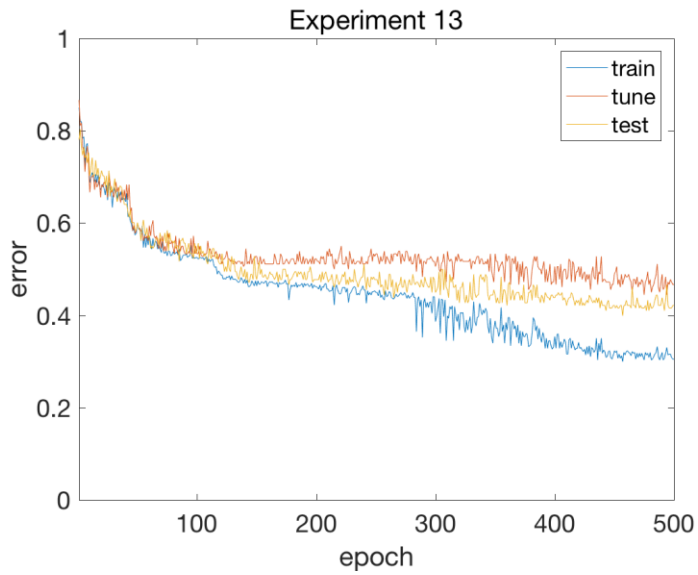

Experiment 8
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
50
100
150
200

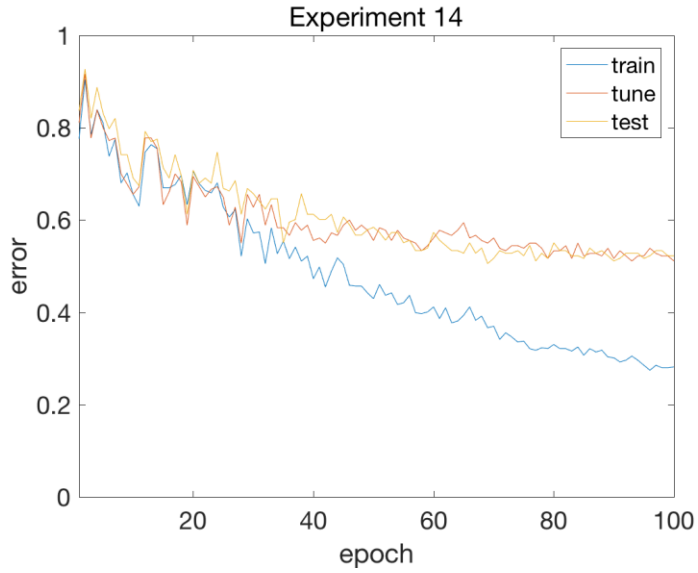

Experiment 9
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100

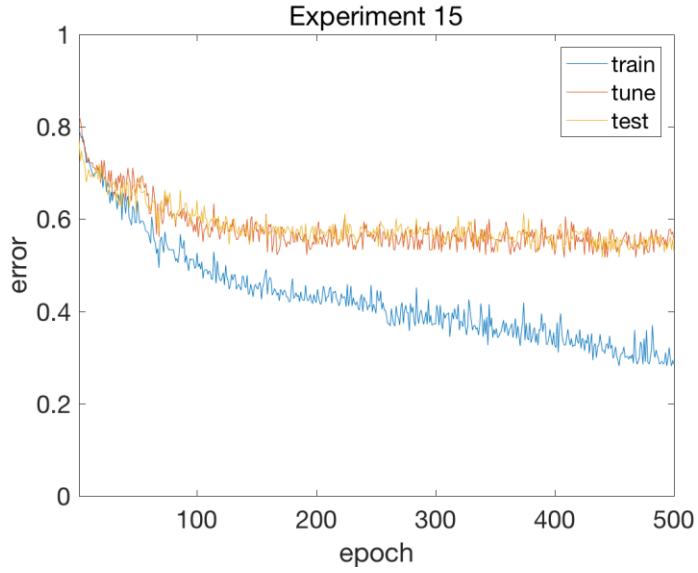

Experiment 10
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
2
4
6
8
10
12
14
16
18


Experiment 11
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
5
10
15
20


Experiment 12
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100


Experiment 13
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
100
200
300
400
500


Experiment 14
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
20
40
60
80
100


Experiment 15
train
tune
test
error
epoch
0
0.2
0.4
0.6
0.8
1
100
200
300
400
500