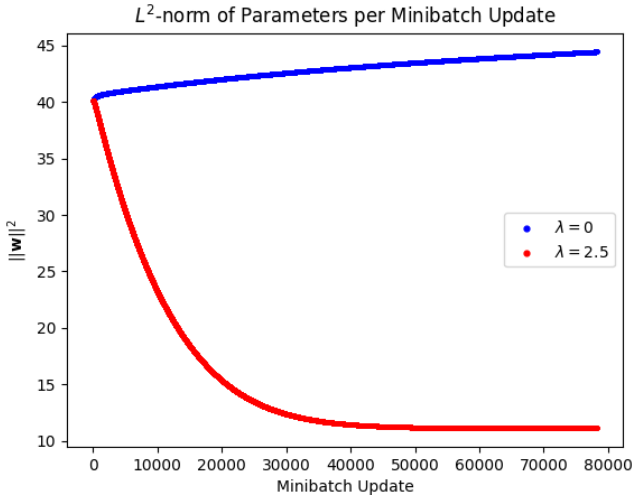
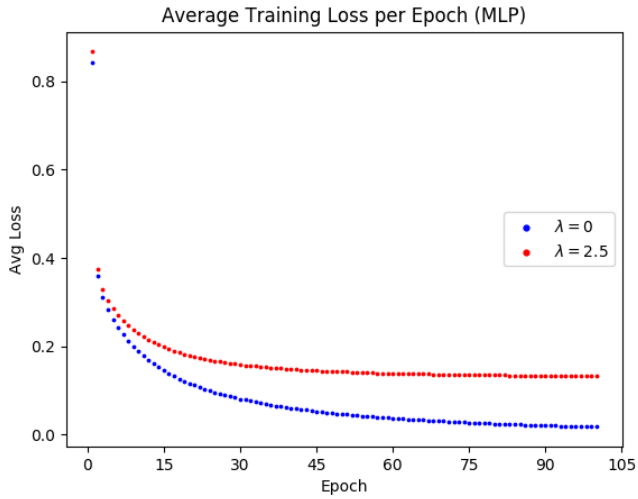
ASSIGNMENT 2 — PRACTICAL PART
CONVOLUTIONAL NEURAL NETWORKS

Samuel Laferrière* **& Joey Litalien†**
IFT6135 Representation Learning, Winter 2018
Université de Montréal
Prof. Aaron Courville
{samuel.laferriere.cyr,joey.litalien}@umontreal.ca

1 REGULARIZATION

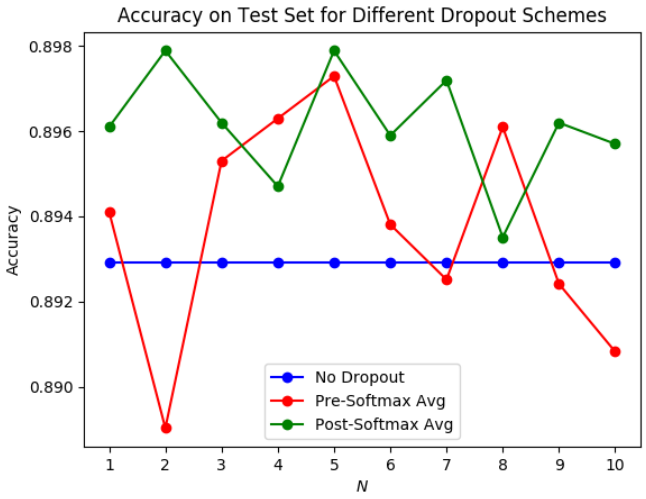
(a) *Early stopping and weight decay.* We plot the L^2-norm of all parameters $\mathbf{w}$ at each minibatch update for 100 epochs. To adapt the loss for minibatch SGD, we rescaled the regularization coefficient as $\lambda \leftarrow \lambda b/|\mathbf{X}|$, where b is the batch size and $\mathbf{X}$ is the entire training set. We also plot the average loss on the training set for both schemes.



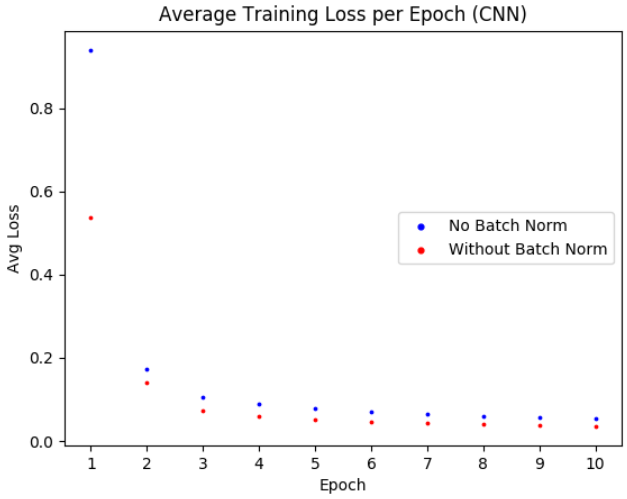


*Student ID P0988904
†Student ID P1195712

(b) *Dropout.* We plot the accuracy for all three dropout schemes. We linked the data points for a better visualization. Note the straight line for the first scheme, as expected.



(c) *Convolutional networks.* We plot the error at the end of each epoch for the model.



2 DOGS VS. CATS CLASSIFICATION

We have resized the images to $3 \times 64 \times 64$ pixels using the Python script provided and separated the dataset into training/valid/test sets as follows. The index ranges apply to both dogs and cats, making each subsets' class distributions equal.

Index range	[0, 7 499]	[7 500, 9 999]	[10 000, 12 499]
Dataset	TRAIN	VALID	TEST
Size	15 000	5 000	5 000

Table 1: Splitting the Dogs vs. Cats dataset.

For all our experiments, we used a learning rate of 0.001 with standard momentum 0.9 and a batch size of 128. Moreover, we used ReLU activations for the last fully connected layer. We used Adam as the optimizer as it seemed to give better results than SGD in general, and trained using binary cross entropy.

(a) *Architecture.* We tried different features for the same architecture inspired from VGG. Our model is described below.

Vanilla		
16	Conv	3×3
16	Conv	3×3
	MaxPool	2×2
32	Conv	3×3
32	Conv	3×3
	MaxPool	2×2
64	Conv	3×3
64	Conv	3×3
64	Conv	3×3
	MaxPool	2×2
128	Conv	3×3
128	Conv	3×3
128	Conv	3×3
	MaxPool	3×3
2048	Linear	512
	ReLU	
512	Linear	2

Augmented		
16	Conv	3×3
16	Conv	3×3
BatchNorm / WeightNorm		
	MaxPool	2×2
32	Conv	3×3
32	Conv	3×3
BatchNorm / WeightNorm		
	MaxPool	2×2
64	Conv	3×3
64	Conv	3×3
64	Conv	3×3
BatchNorm / WeightNorm		
	MaxPool	2×2
128	Conv	3×3
128	Conv	3×3
128	Conv	3×3
BatchNorm / WeightNorm		
	MaxPool2D	3×3
2048	Linear	512
	ReLU	
	Dropout	0.5
512	Linear	2

Table 2: Model architecture for Dogs vs. Cats.

(b) *Performance on test set.*

(c) *Visualization and possible improvements.*