Irene Volpe

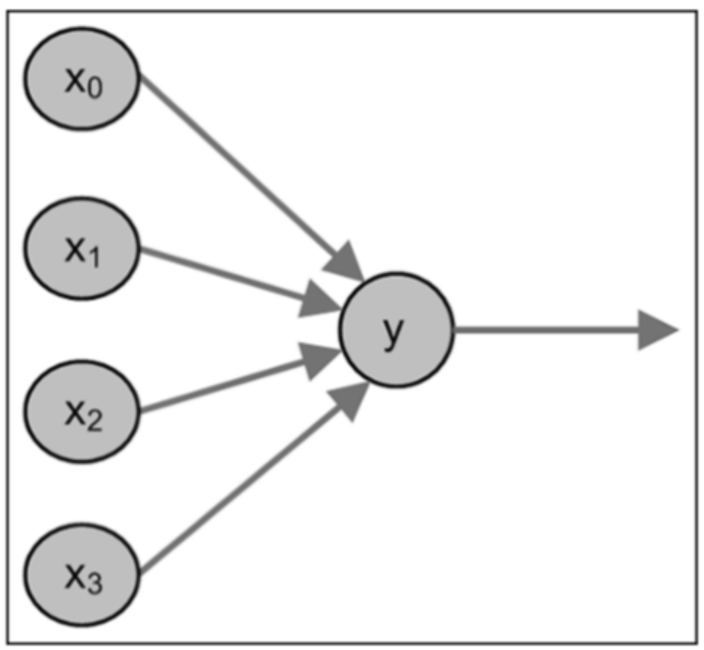
PML Presentation based on the Function Approximation

1. Number of hiddens and training performance

1. We were asked to write an MLP with
   - 1 input node,
   - 1 output node
   - 2 hidden layers, 50 hidden nodes each

and compare the training performance for the 2 layer MLP to that of the original 1 layer MLP.

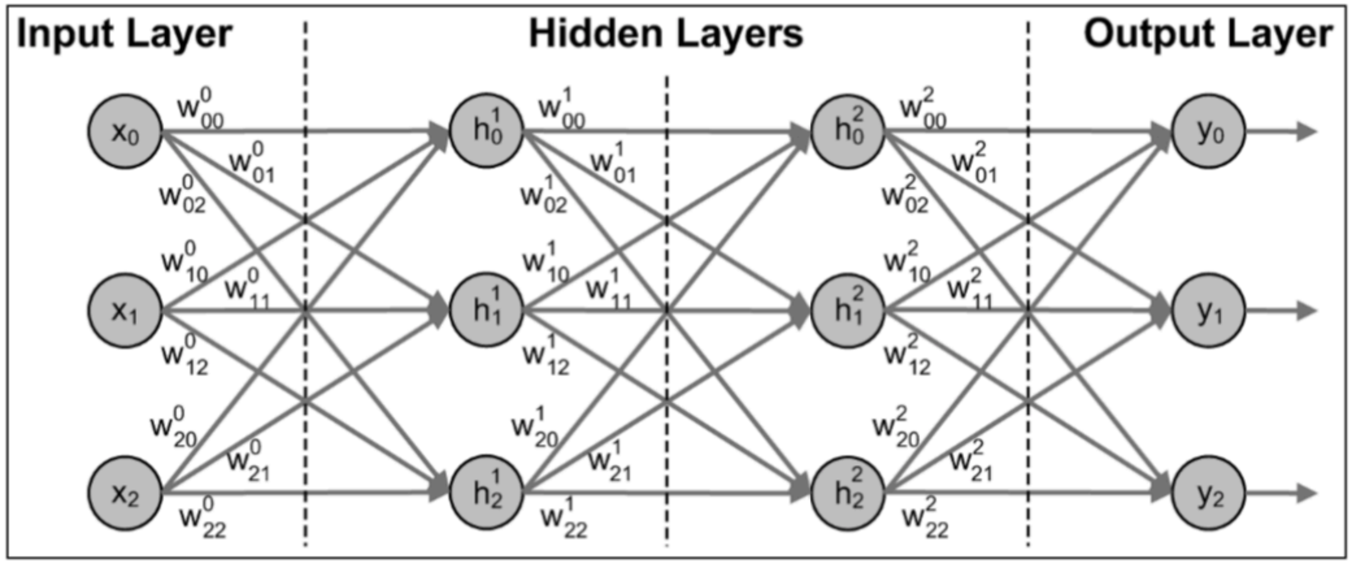
What is an MLP



A perceptron

Like biological neurons, perceptrons can be connected together so that the output of one perceptron serves as the input of another.

Columns of NNs are called *layers*, so NNs are frequently called multilayer perceptrons (MLPs).

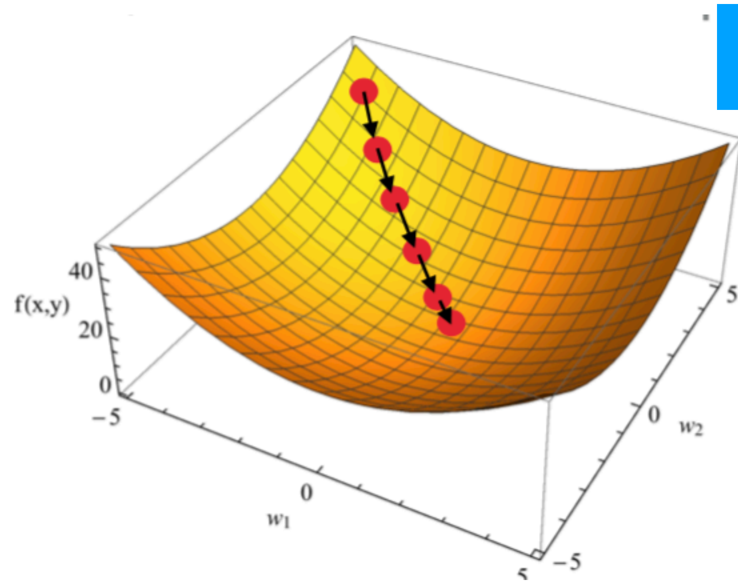


A MLP with 4 layers and 3 perceptrons.

How to compare training performance

Optimization purpose: select a point in hyperspace that optimises the performance using the cost function.

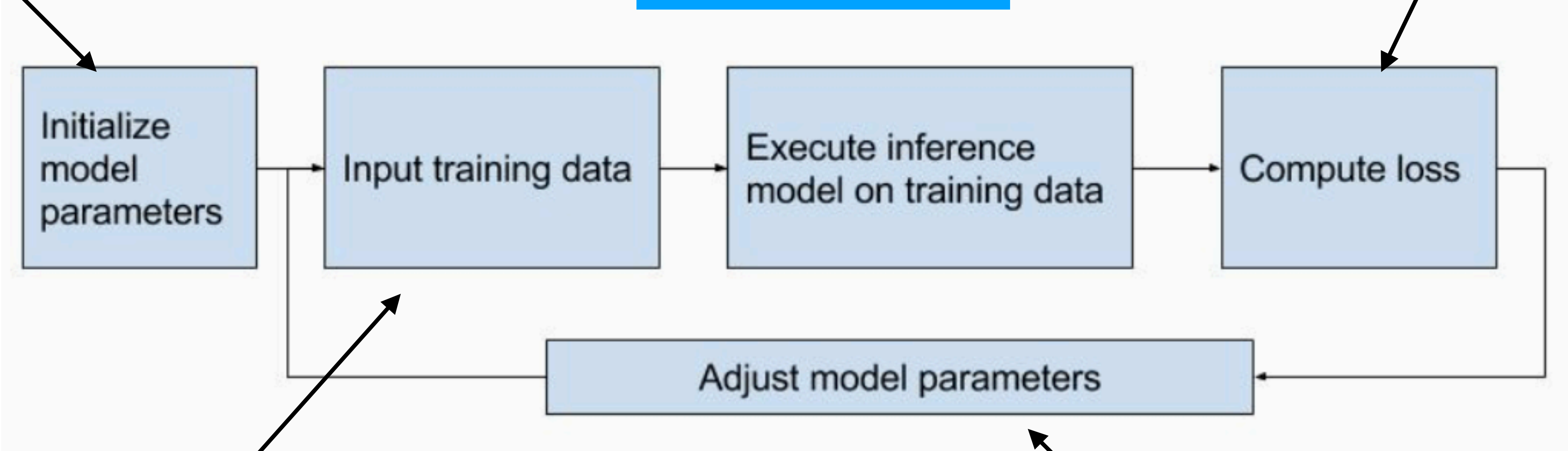
A Hyperspace



a value that will summarize how far the computed values in the last step with the expected output are.
we use L2 loss:
cost = tf.nn.l2_loss(output_layer - y)

Used to define a Hyperspace

A Training loop



Reads the training data and expected

We used the Adam optimiser algorithm.
optimizer =
tf.train.AdamOptimizer(learning_rate=learning_rate)
.minimize(cost)

Fig. 1 shows that:

- 3H MLPs would get lower cost values from the very first epochs

- 2H MLPs might would have take advantage of a further training over additional epochs.

- 1H delivers the worst accuracy.

Fig. 1
1,2,3Hs,
lr: 0.001

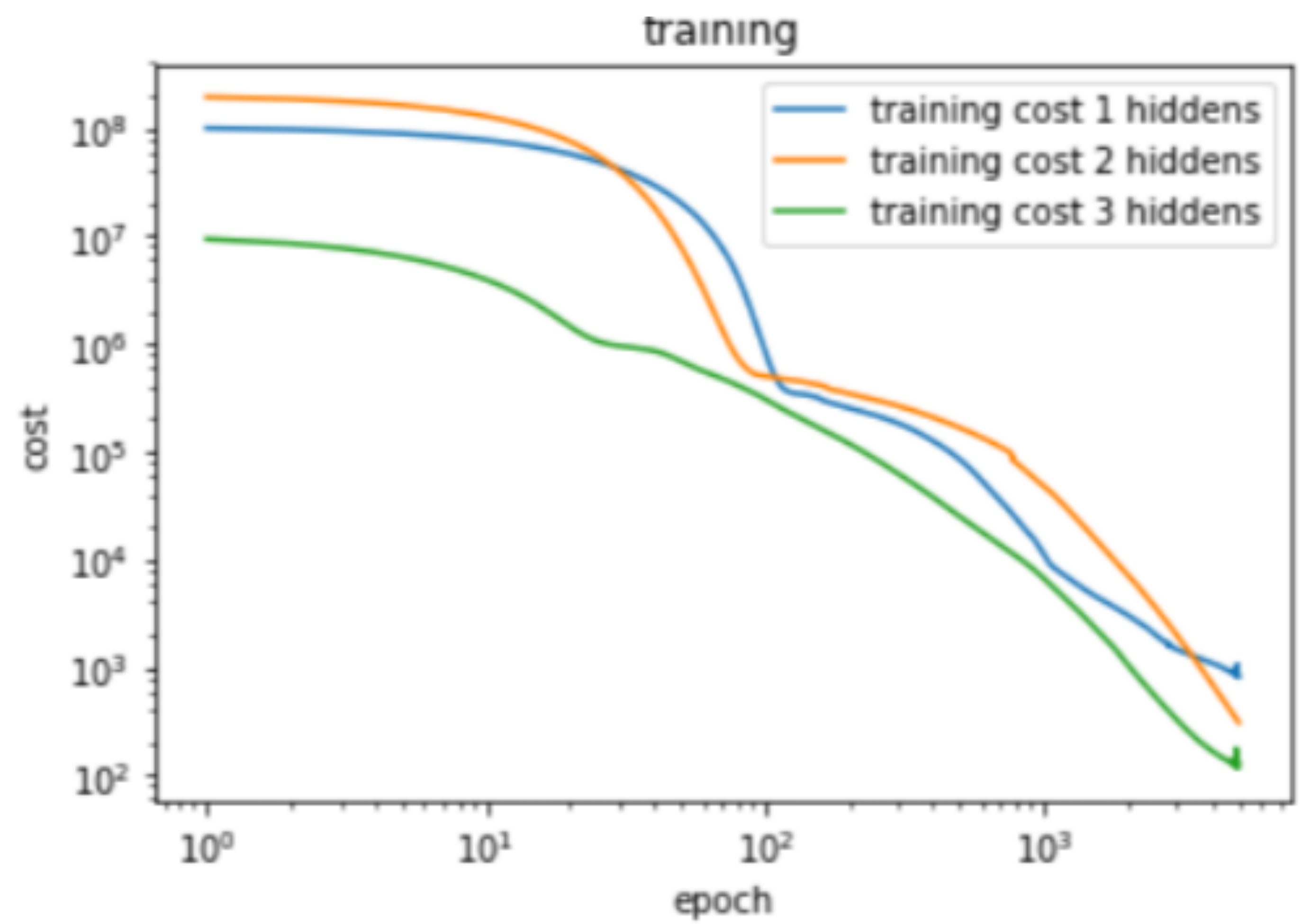
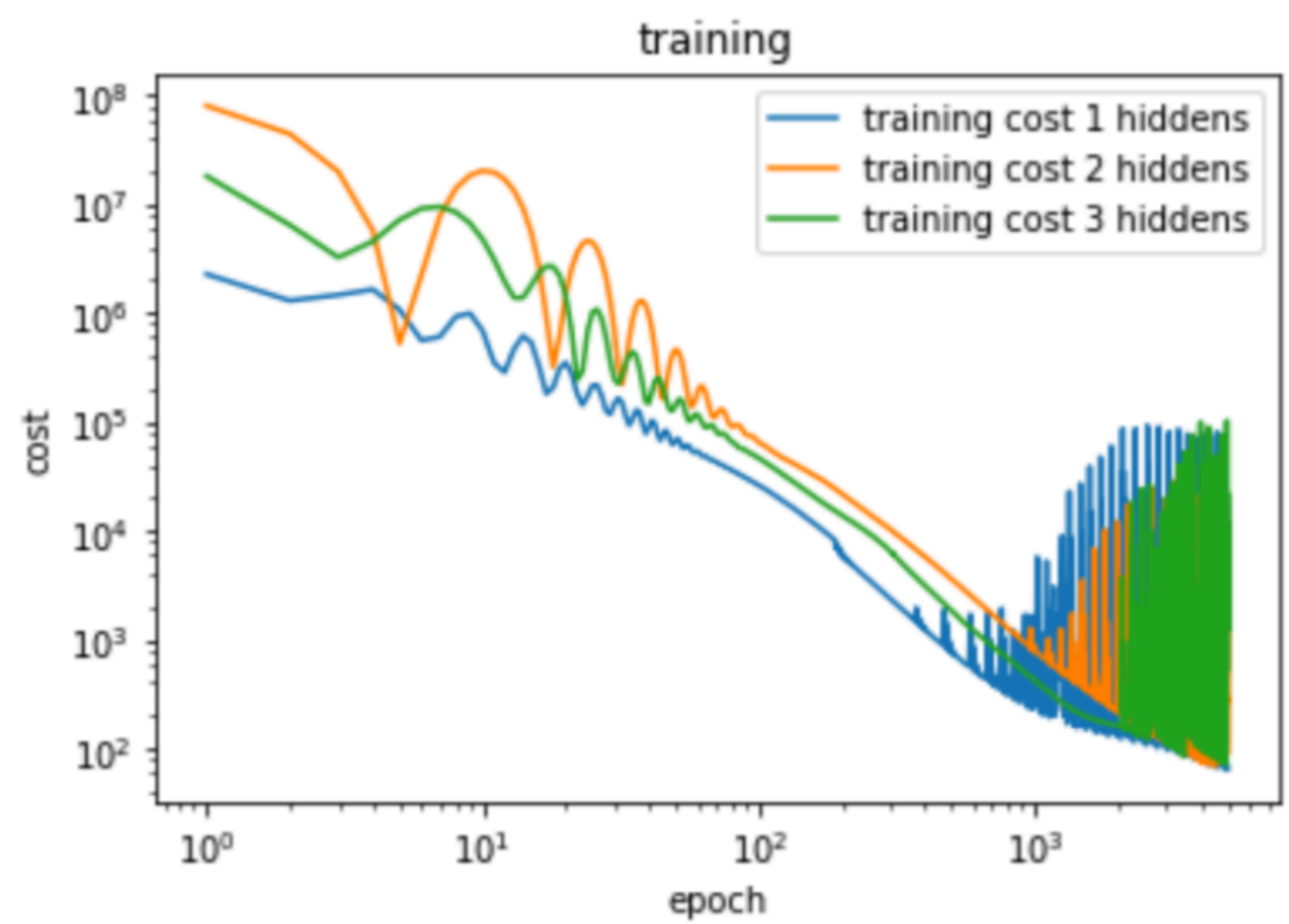


Fig 2 shows that increasing the learning rate to 0.01:

- we converge to minima less precisely.

- we affect the three models equally.

Fig. 2
1,2,3Hs,
lr: 0.01



2. Dropout and training performance

2. We were asked to make a plot to compare the cost evolution as a function of training epoch for (1) with / without the use of dropout for a keep probability of 50%.

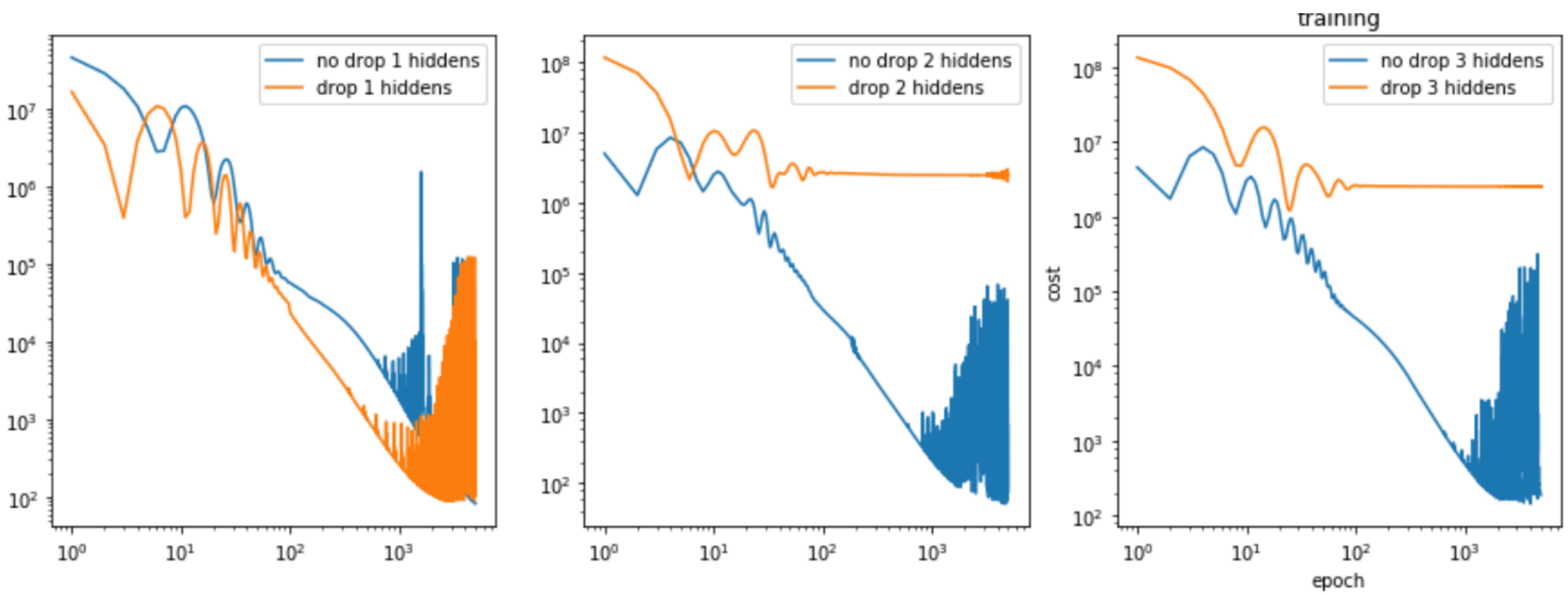


Fig 3: 0.5 drop vs no drop

Fig 3 shows that:

- dropout's performances are worse
- especially when having more than one hidden.

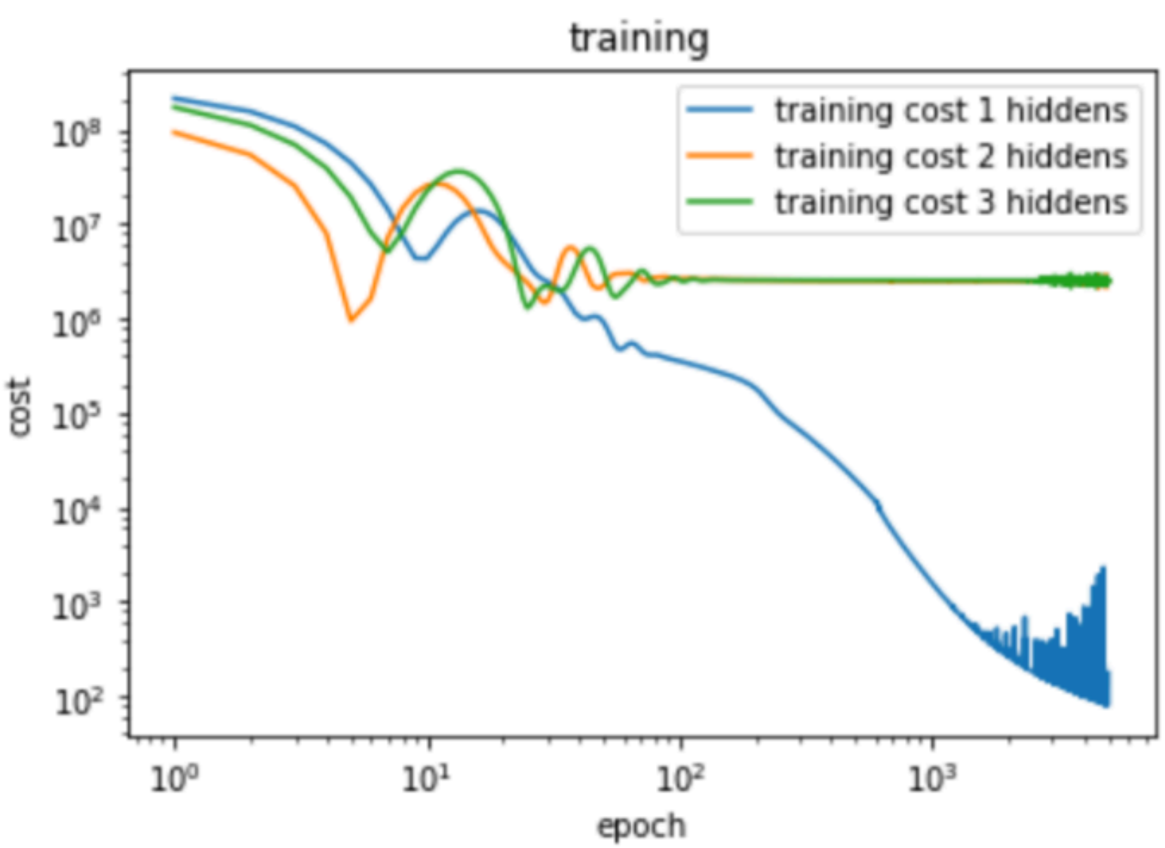


Fig 4: 0.5 drop and 1,2,3 H

Fig 4 shows that:

- after 1000 training epochs, 2H and 3H models with dropout reach a stable but bad accuracy,
- the 1 hidden layer model converges to better minima values.

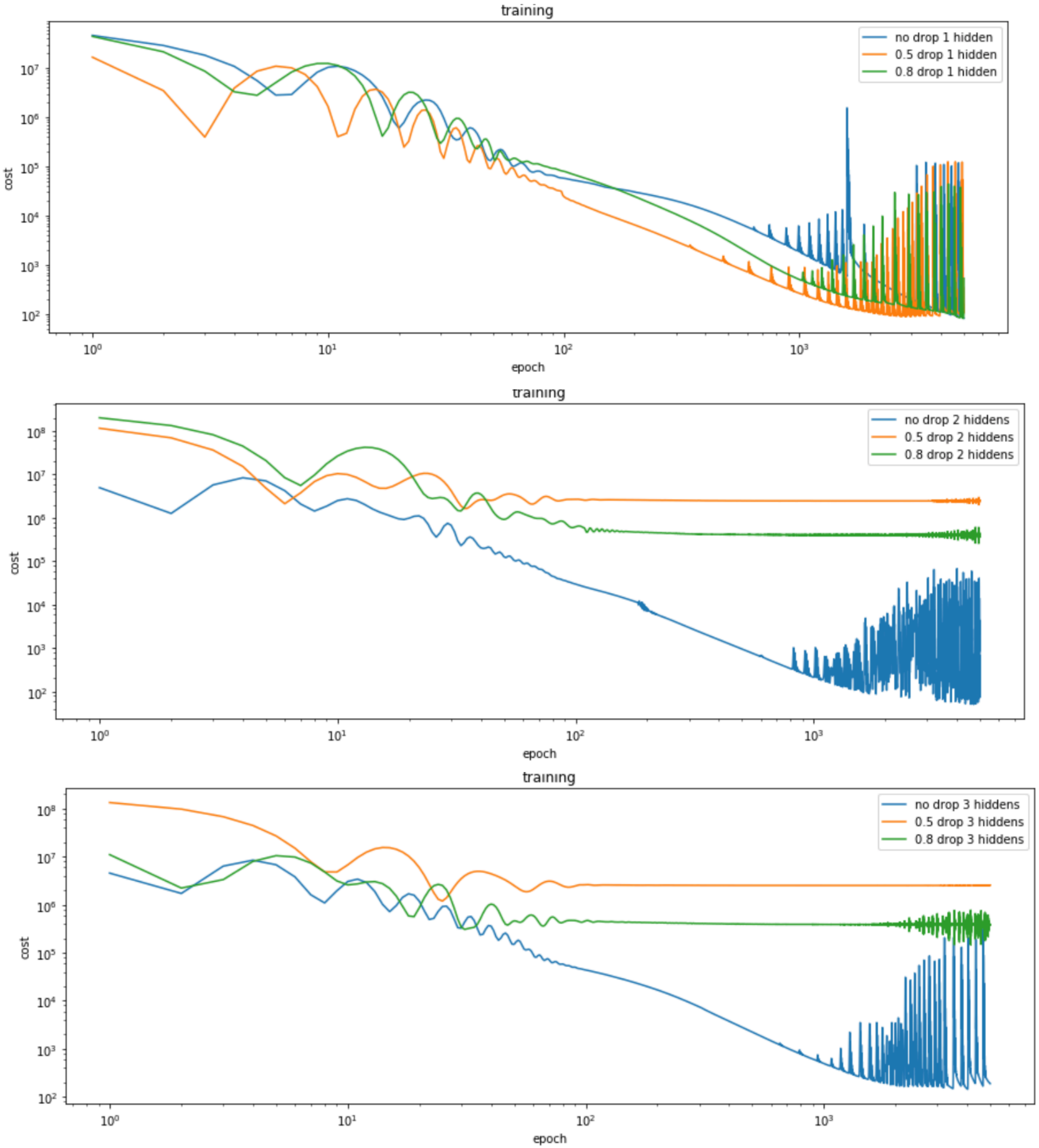


Fig 6:
1, 0.8 and 0.5 drops with 1,2,3 H

Fig 6 shows that, regardless the num of Hs:

- the best result is for keeping all the nodes
- the worst is only keeping half of them.

3. Batch learning and training performance

3. We were asked to make a plot to compare the cost evolution as a function of training epoch for batch training vs using all of the training events in one go.

Fig7:
4,11 and no batches

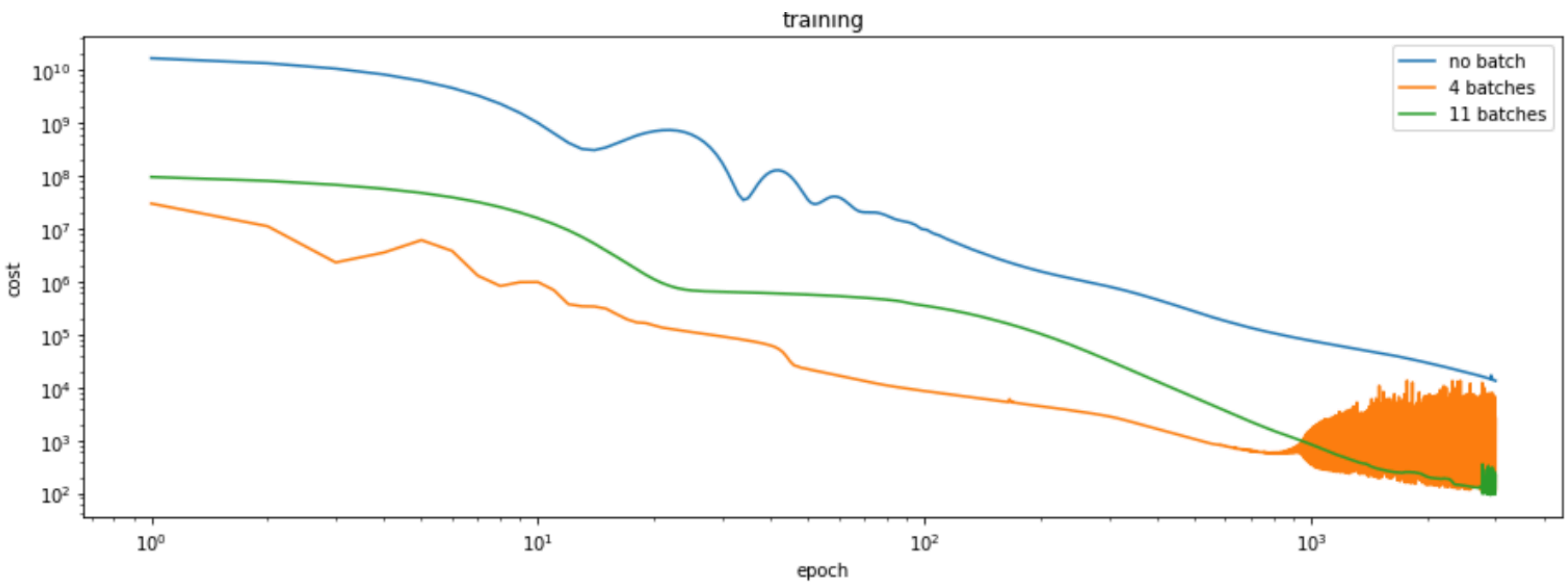


Fig 7 shows that:

- more batches lead to better minima.
- mid num of batches starts with the best values, and must be stopped earlier.
- no batch learning leads to poor minima (but the slope suggest it might improve with more epochs).

Batch learning, dropout and training performance

4. We were asked to make a plot to show the combined performance of using both batch training and dropout to explore the potential for using both methods.

Fig 8: 4 batches and 1,0.8,0.6 dropout values

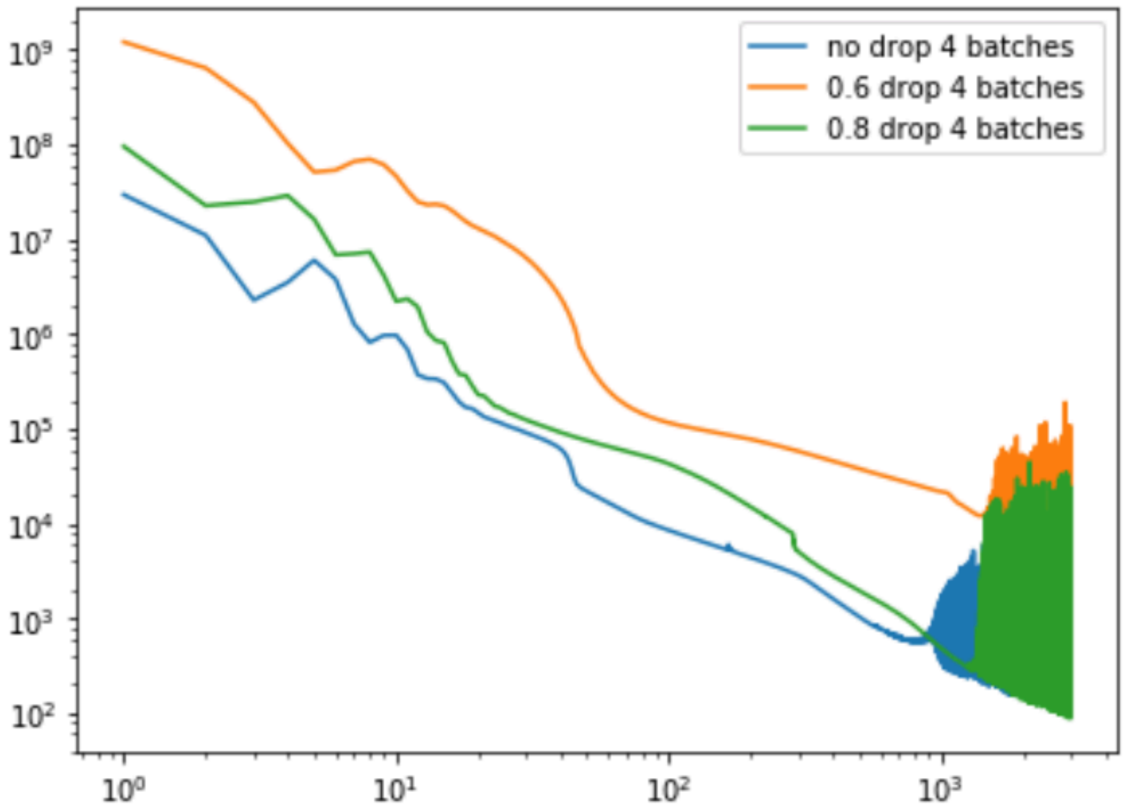


Fig 9 shows that:

- no drop must be stopped earlier
- highest drop model remains too bad.
- mid drop model reaches the best minima.

Fig 8: 11 batches and 1,0.8,0.6 dropout values

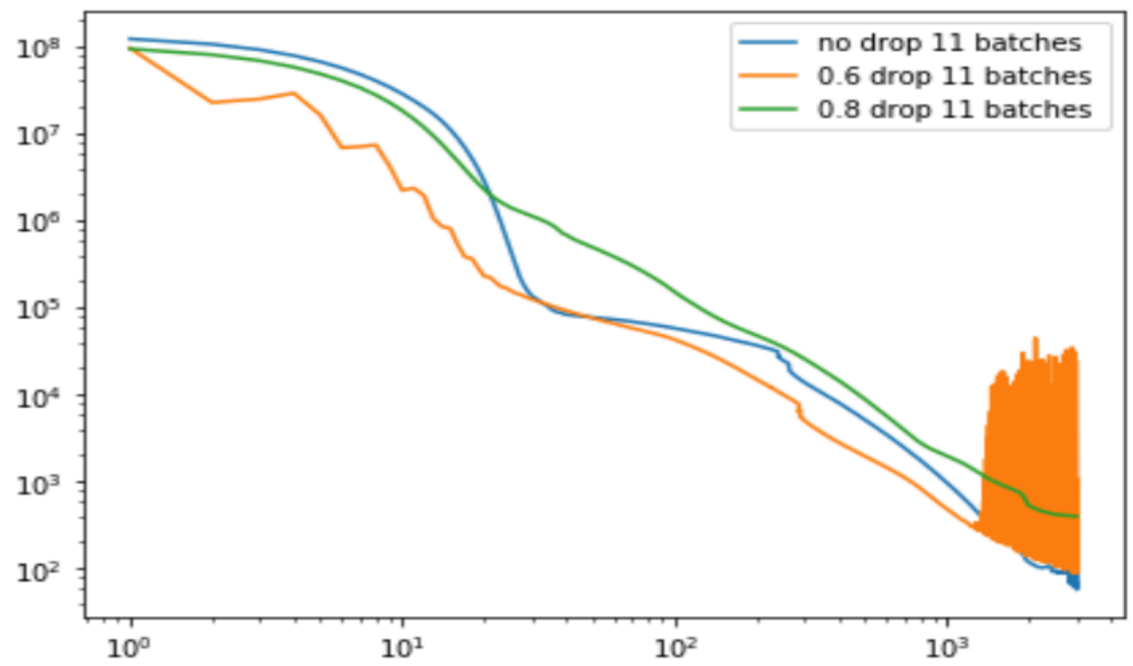


Fig 8 shows that:

- more batches reduced the drop out differences

Fig 9: 4 batches and:
1,0.8 dropout values;
1,0.6 dropout values

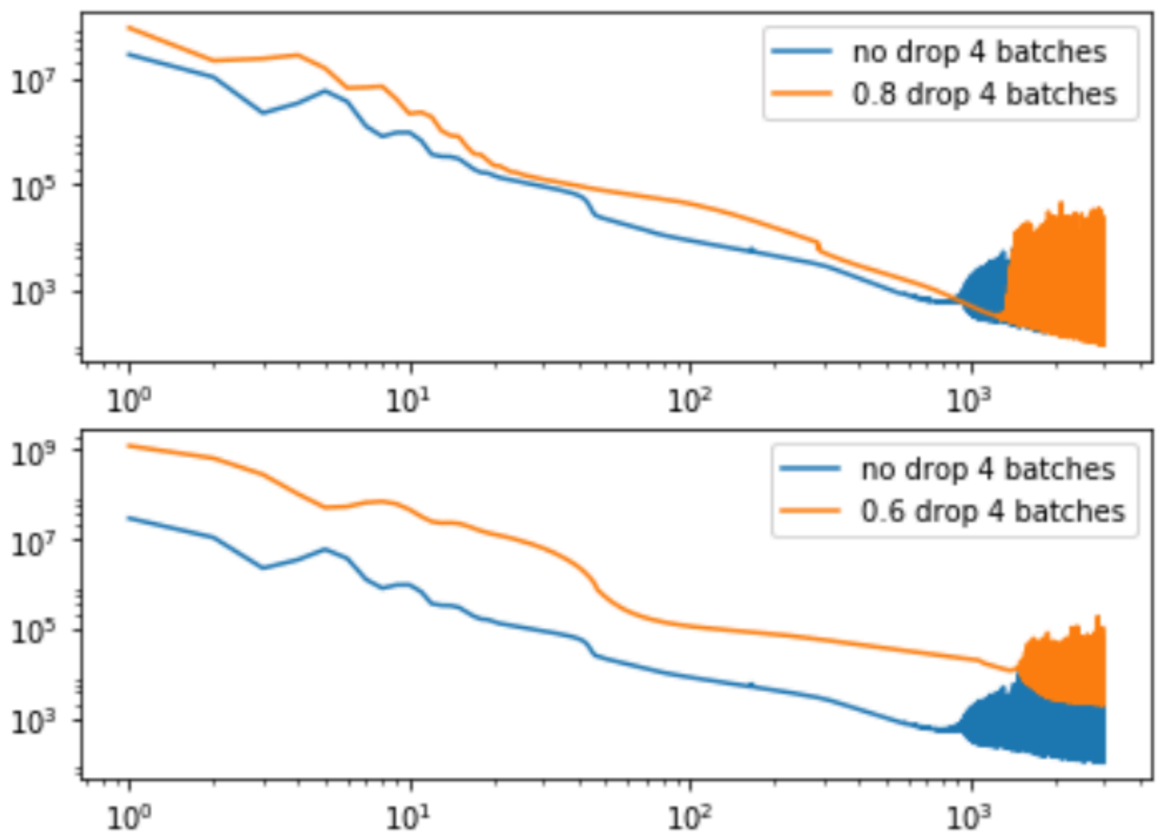


Fig 10: 11 batches and:
1,0.8 dropout values;
1,0.6 dropout values

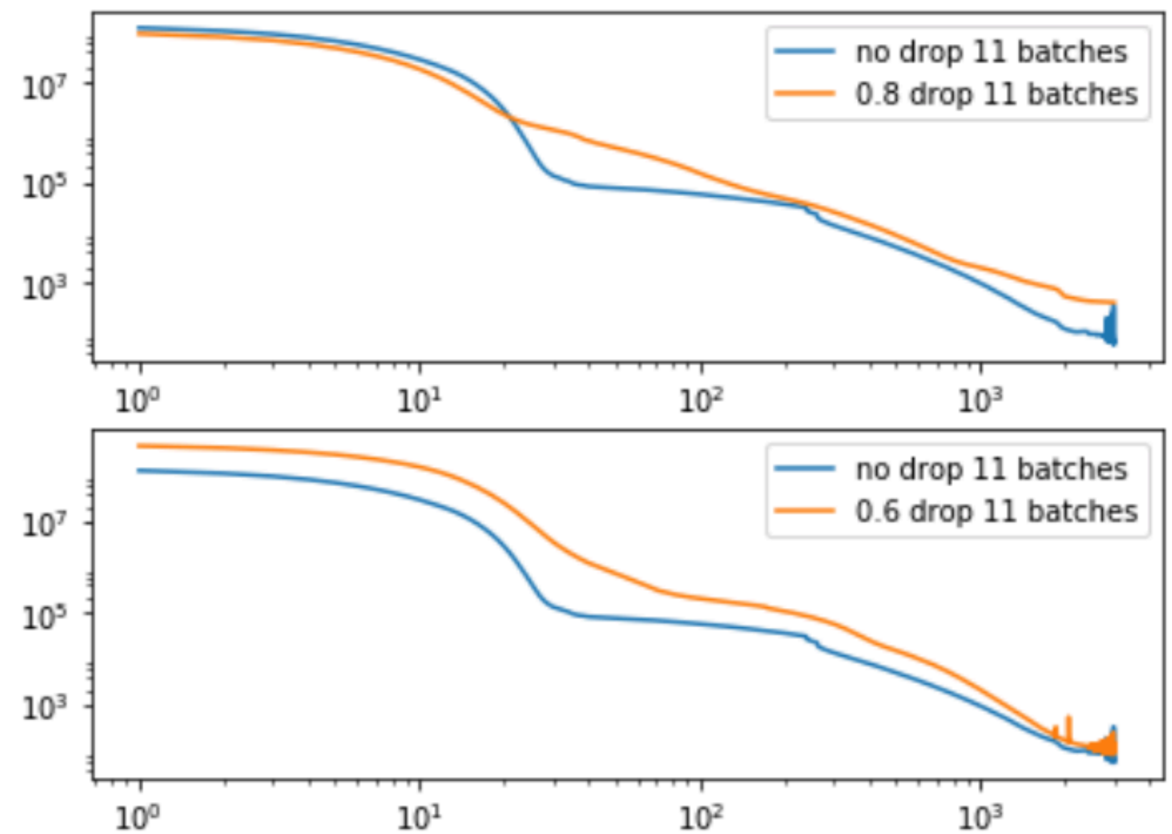


Fig 9,10 show that:
- 4 batch trained models must be stopped earlier
- batches were more effective than drops

Thank you!