

PORTLAND STATE UNIVERSITY

DEEP LEARNING: COMPUTATIONAL STRUCTURES
AND PROGRAMMING

WINTER 2021

Project 2

Author:

Hermann YEPDJIO

Professor:

Dr. Suresh SINGH

February 1, 2021



1 Construct a Simple CNN

1.1 Submit Plots

Table 1: Classification Accuracies

Loss(Activation) function/Learning Rate	MSE (Sig)	C_E (Sig)	MSE (Tanh)	C_E (Tanh)
0.1	10%	63%	8%	50%
0.01	10%	18%	10%	62%
0.001	10%	10%	10%	55%

- When looking at figures 1 and 2, we can see that most of the time, the training and testing loss curves decrease as the epoch number increases, which is the expected behavior because it shows that the network is learning.
- In most cases we can also see that the training and testing loss curves are very close to each other and some time even overlap. This is also an expected behavior as it shows that the network is not over-fitting on the training dataset.
- However, in some cases like in 2a, we can see that the training and testing loss curves decrease then start increasing as the epoch number increases. One reason to this behavior could be that we trained the model for too long and should have stopped when the losses started increasing.

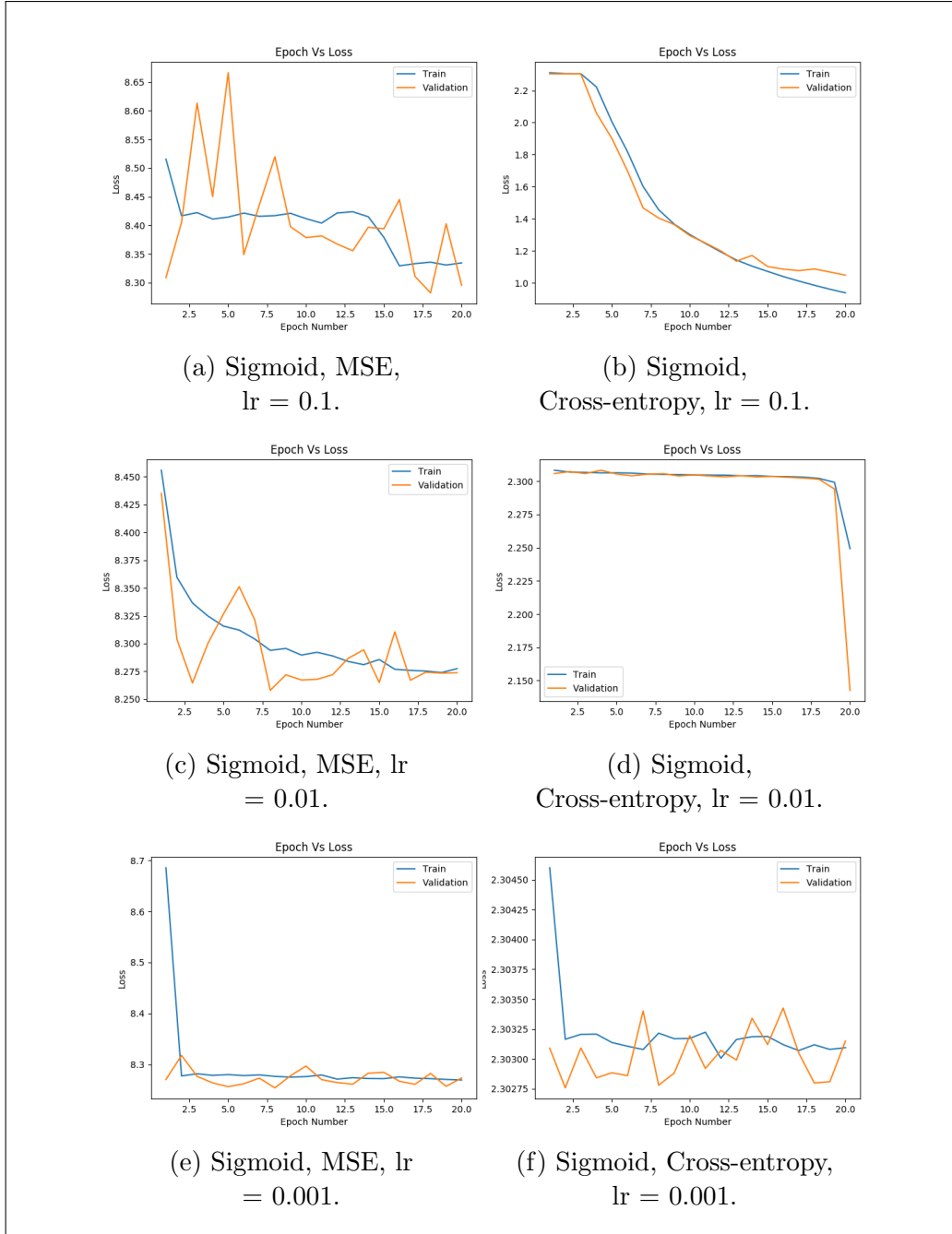
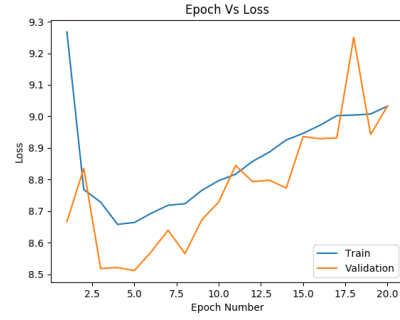
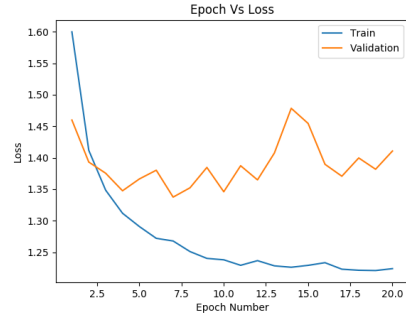


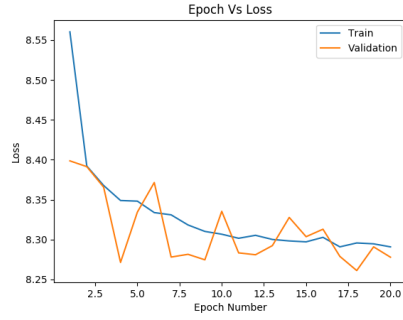
Figure 1: Experiments with the Sigmoid Activation Function



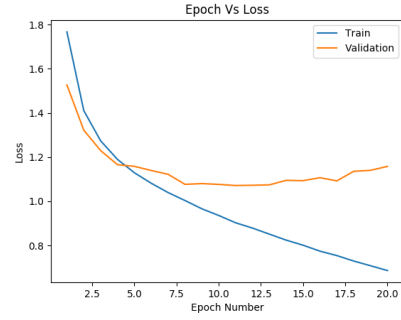
(a) Tanh, MSE, $lr = 0.1$.



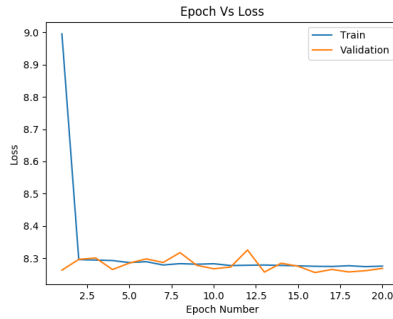
(b) Tanh, Cross-entropy, $lr = 0.1$.



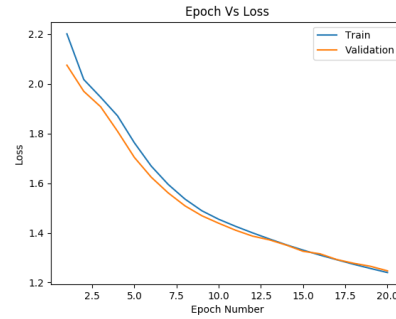
(c) Tanh, MSE, $lr = 0.01$.



(d) Tanh, Cross-entropy, $lr = 0.01$.



(e) Tanh, MSE, $lr = 0.001$.



(f) Tanh, Cross-entropy, $lr = 0.001$.

Figure 2: Experiments with the Tanh Activation Function

1.2 Feature Maps of 10 Images at the Last Convolution Layer

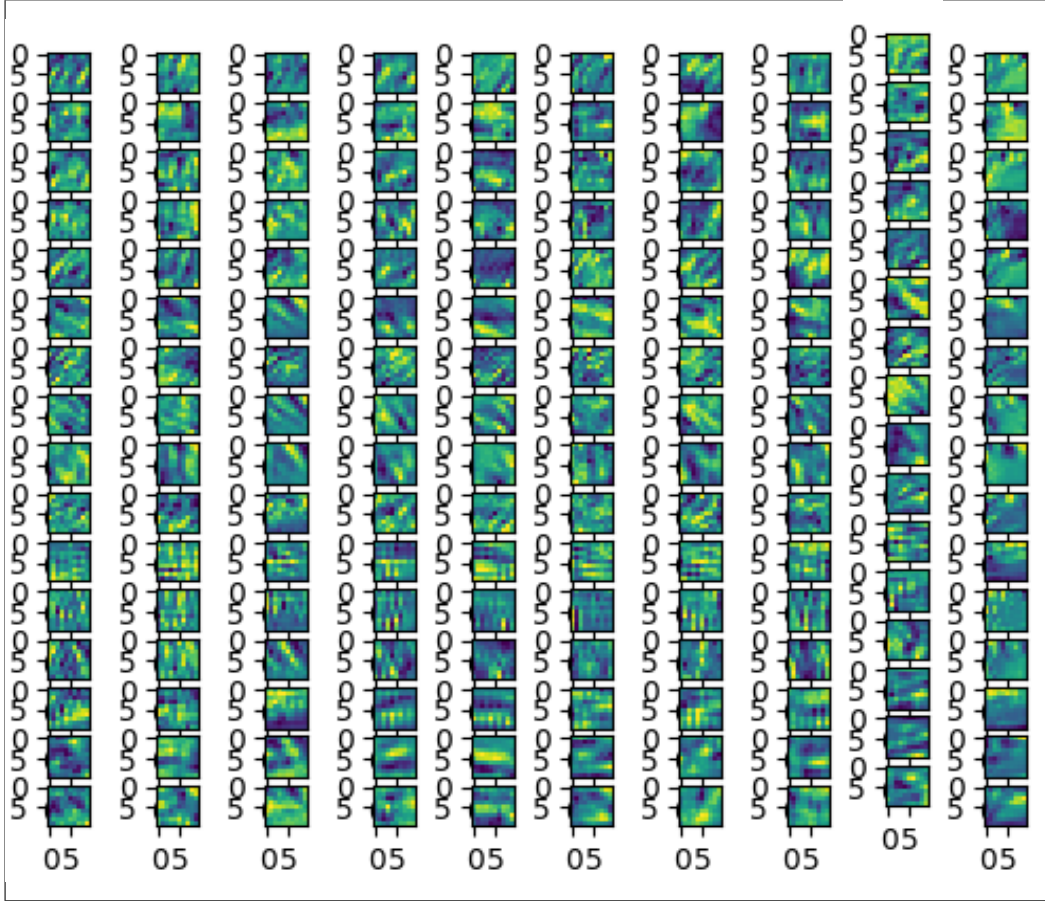


Figure 3: Feature maps of 10 images at the last convolution layer. Parameters = (b_size = 10, activation function = Tanh, loss function = cross-entropy, lr = 0.01).

From Figure 3, we can see that our network has already learned $16 \times 5 \times 5 = 400$ features for each image after the second convolutional layer. The network used corresponds to the one in figure 2d.

2 Select Relu, Learning Rate of 0.001, Cross-entropy Loss. Change the Network to Use 3x3 Kernels

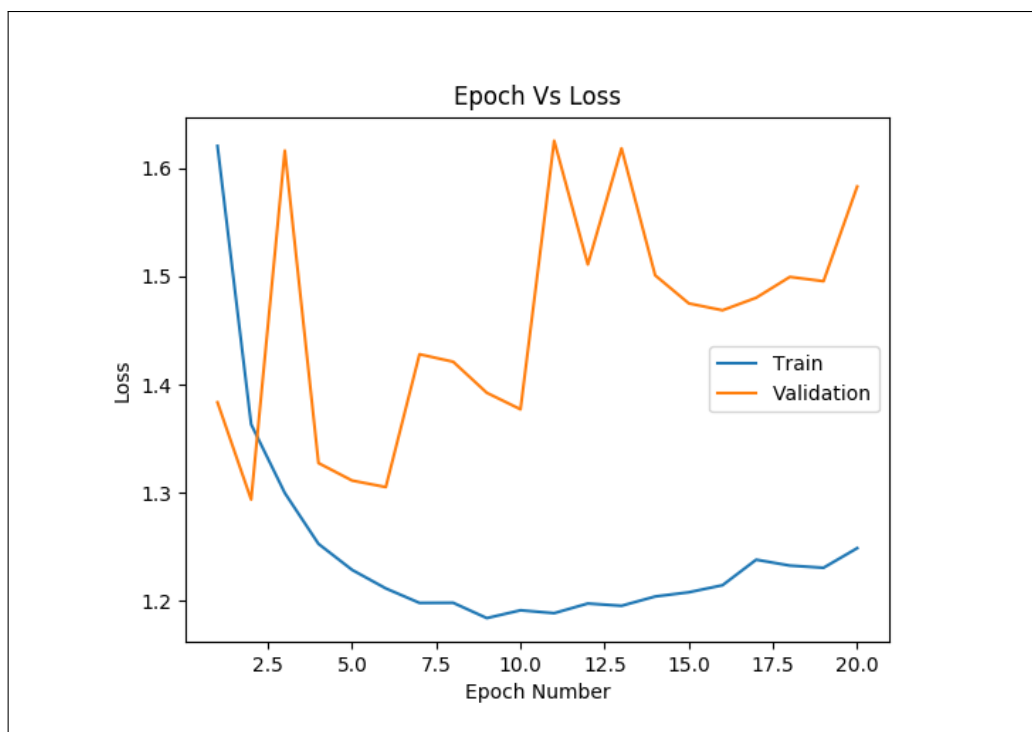


Figure 4: Relu, cross-entropy, lr = 0.001, 3x3 kernels.

In terms of classification accuracy, the previous network (figure 2d) achieved 62% while the current one (figure 4) achieved 51%. When looking at the loss curves, we can see that at epoch 20, the previous network is still learning while the current one seems to have stopped learning at epoch 7. This can explain why the previous network produced better accuracy results.

3 Build a CNN with 5 Convolution Layers with 3x3 Kernels and Corresponding 2x2 Average Pooling with Stride 1

3.1 How long did it take to train? Why?

It took 1h13 mins to train and the reason why it took so long is because by using padding and decreasing the kernel size and pooling stride, the size of the image does not change by much after each convolution block (convolution layer + pooling layer) which still leaves a large number of parameters to be trained after each block. Then by increasing the number of convolution blocks, the number of parameters to be trained exploded.

3.2 How does the Accuracy of this Network Compare Against the Previous Two? Explain

This network produced an accuracy of 61% compared to 62% and 51% respectively for the previous two networks (figures 2d and 4). However, when looking at the loss curves, we can see that for this network, the training loss gets close to 0 at epoch 50 while the test loss seems to be increasing. This can be a proof that the network has over-fitted the training dataset. One cause could be that the number of parameters trained is way larger than what is necessary to solve this kind of problem.

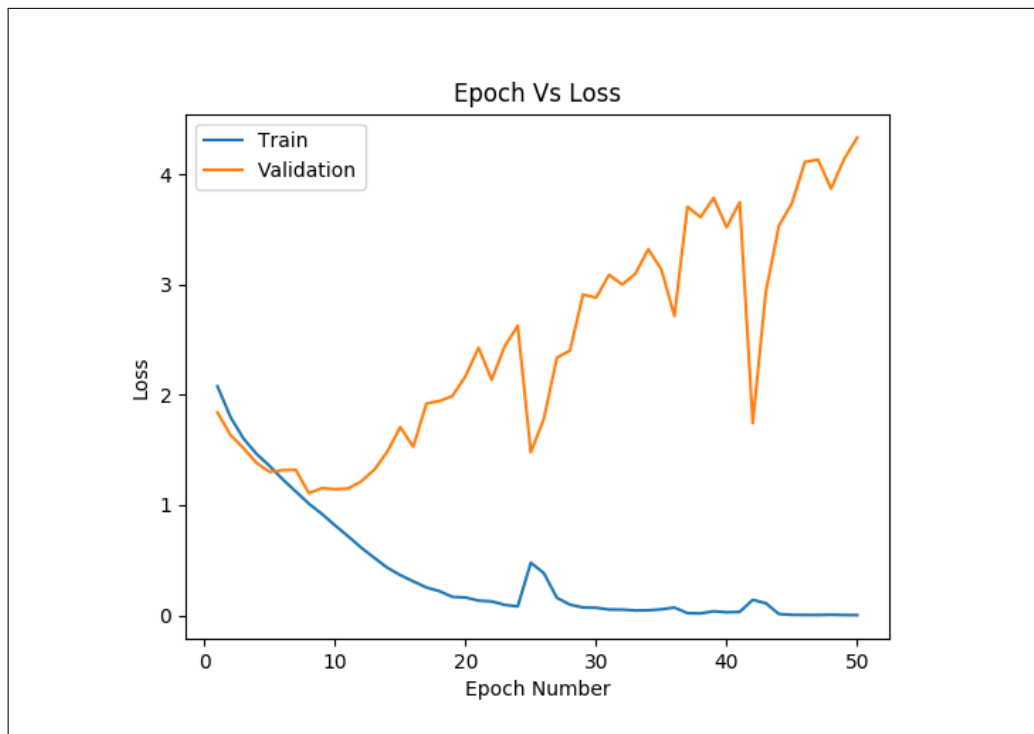


Figure 5: 5 Convolution Layers, Relu, cross-entropy, lr = 0.1, 3x3 kernels, 2x2 average pooling (stride = 1).