ECE521 Assignment3 Report

Shengjie Xu (1001175186) (40% Contribution)
Yu Cheng Gu (1001202067) (40% Contribution)
Shihan Zhang (1002055795) (20% Contribution)

August 25, 2020

For all following sections, we are using the following weight decay loss function for training:

$$\mathcal{L}_W = \lambda \left(\sum_{l=1}^{L} \text{Trace}(W^{(l)^T} W^{(l)}) \right)$$

For neural networks with $L-1$ hidden layers and one output layer. All evaluations are done at the end of epoch and weight decay loss is not included when calculating losses. All epoch indices start from zero. Unless otherwise specified, the following parameters are used:

- mini-batch size: 300
- learning rate: $1e-4$
- weight decay: $5e-4$

1 Feedforward fully connected neural networks

1.1 Layer-wise building block

Listing 1 is the function we use to build a layer of network. To implement weight decay, we also output $\text{Trace}(W^T W)$ in this function.

1.2 Learning

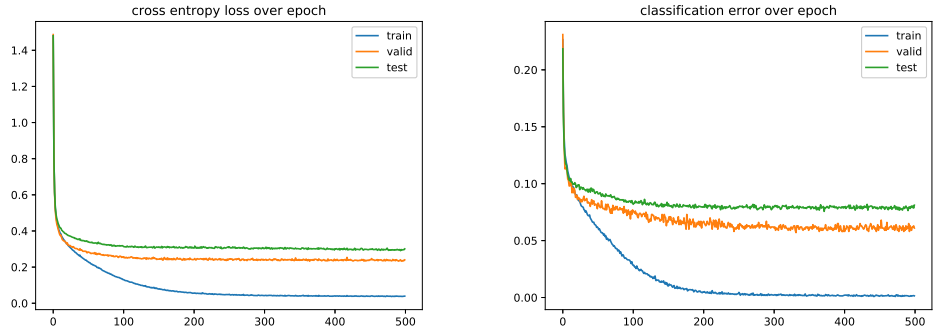
Figure 1 is the plot of cross entropy loss and classification error rate over three sets of data. This training runs 500 epochs in order to better capture the trend of training, after the stable state is reached and training stops doing useful work.

```
def build_layer(x,d):
  # x: activation from previous layer
  # d: number of output neurons in current layer

  width_input=x.get_shape().as_list()[1]
  w_init_val=tf.truncated_normal([width_input,d],
    mean=0.0,
    stddev=3.0/(width_input+d),
    dtype=tf.float32)

  w=tf.Variable(w_init_val,dtype=tf.float32)
  b=tf.Variable(tf.zeros([d],dtype=tf.float32),dtype=tf.float32)
  w_trace=tf.trace(tf.matmul(w,w,transpose_a=True))
  z=tf.matmul(x,w)+b
  return z,w,b,w_trace
```

Listing 1: Layer building function



(a) Cross entropy loss

(b) Classification error rate

Figure 1: Cross entropy loss and classification error for one layer 1000 hidden unit neural network

From the data in the plot, the training set cross entropy loss and classification errors keep dropping during the training. The loss and error in validation set and test set drops along with test set data during the first few tens of epochs, and then they start to drop much slower and converge. The classification errors for validation set and test set also starts to wiggle instead of monotonically decreasing.

1.3 Early stopping

Due to the small amount of weight decay used, the validation cross entropy loss and classification error start to drop considerably slower after about 10 epochs, and reach a plateau after around 100 epochs.

For early stopping point based on validation classification error, the first time it stops dropping for continuous two epochs happen after epoch 9, where:

- training classification error: $1585/15000 \approx 10.567\%$

- validation classification error: $98/1000 = 9.8\%$
- test classification error: $281/2724 \approx 10.316\%$

For early stopping point based on validation cross entropy loss, the first time it stops dropping for continuous two epochs happen after epoch 27. The early stopping points are different because the number of validation set data points are discrete and classification error is easy to wiggle, while the cross entropy loss changes more continuously. The early stopping point should be based on validation classification errors, because classification error is the most important metrics that we intend to optimize, and we want to stop when the classification error stops dropping.

2 Effect of hyperparameters

2.1 Number of hidden units

After training 100 epochs in lock step (using the same batches), the validation classification errors for three neural networks with one hidden layers are as follows:

- 100 hidden units: 8.4%
- 500 hidden units: 7.3%
- 1000 hidden units: 7.1%

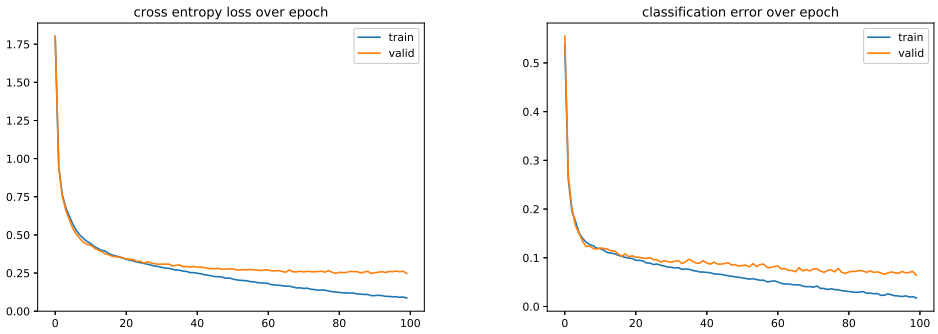
And the test set classification error is $231/2724 \approx 8.480\%$. This result shows that for single hidden layer neural networks, more hidden units results in lower classification error, although the improvement is limited when number of hidden units are sufficiently large.

2.2 Number of layers

Figure 2 is the loss and error for the neural network with two 500-unit layers over 100 training epochs. The cross entropy loss and classification error after training are:

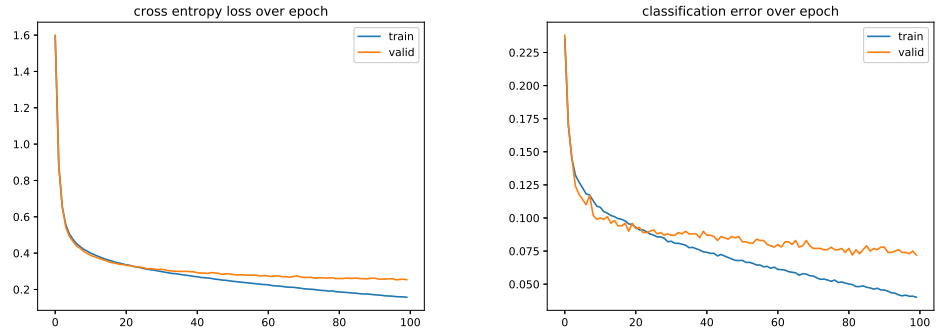
- validation cross entropy loss: 0.2470465898513794
- validation classification error: $64/1000 = 6.4\%$
- test set cross entropy loss: 0.30473977
- test set classification error: $215/2724 \approx 7.893\%$

The two layer neural network performs slightly better than single hidden layer neural network, but it also needs more epochs to train and it converges slower.



(a) Cross entropy loss (b) Classification error rate

Figure 2: Cross entropy loss and classification error for two layer $500 + 500$ hidden unit neural network



(a) Cross entropy loss (b) Classification error rate

Figure 3: Cross entropy loss and classification error for single layer 1000 hidden unit neural network with dropout

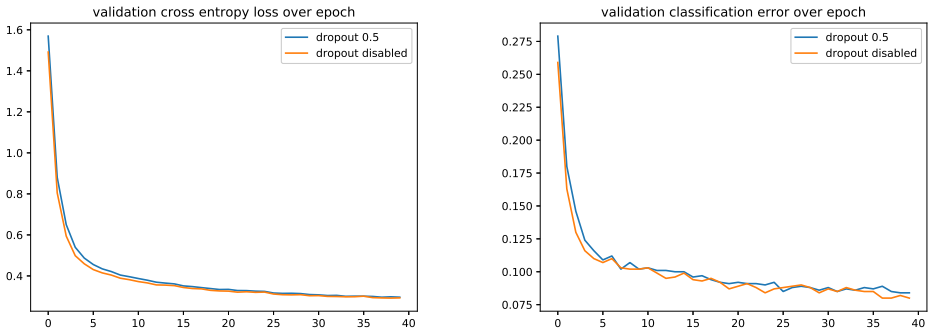
3 Regularization and visualization

3.1 Dropout

Figure 3 is the cross entropy loss and classification error for the neural network with 0.5 dropout probability. Comparing with the case without dropout, the validation classification error wiggles more often, and training errors also drops more slowly.

3.2 Visualization

Figure 4 is the validation cross entropy loss and classification error for neural network with and without dropout. Using the same criteria, the early stopping points for case with and without dropout are after epoch 7 and 12, respectively. For each hidden unit, the weights are normalized so that minimum is pure black and maximum is pure white, and other values have gray scale in between.



(a) Cross entropy loss (b) Classification error rate

Figure 4: Validation cross entropy loss and classification error for single layer 1000 hidden unit neural network with and without dropout

Table 1: Random hyperparameters and classification error

Hidden Layers	Learning rate	Weight decay	Dropout	validation	test
$[240, 377, 136, 264]$	3.695×10^{-3}	3.685×10^{-4}	No	90.7%	90.565%
$[117, 395, 279, 307, 178]$	1.598×10^{-3}	3.822×10^{-4}	No	90.7%	90.565%
$[448, 369, 348, 241, 193]$	7.997×10^{-4}	1.495×10^{-4}	No	90.7%	90.565%
$[221, 334]$	8.014×10^{-3}	1.241×10^{-4}	No	8.3%	8.443%
$[298, 150]$	5.760×10^{-4}	3.684×10^{-4}	No	6.9%	7.930%

The weights change quickly in the first few epochs, and then changes get slowed down. Minima and maxima changes with remaining weights in similar pace in the beginning, and after majority of weights finished changing, they continues to evolve while leaving other weights behind. This can be seen from the contrast of weight images; in the beginning many weights are close to minima and maxima, so the contrast is large, and later on they start to fall behind and becomes gray.

In our experiments the difference between the network with and without dropout is very small. Both stream of weight images evolve in similar pace, and similar number of hidden units do not have a recognizable image.

4 Exhaustive search for the best set of hyperparameters

4.1 Random search

We use hash of tuple of our student ids to seed python random, Numpy, and Tensorflow, and table 1 is the results after 100 epochs of training. Two shallow networks performs well after 100 epochs, but three deep networks performs badly; they continue to perform poorly even in 500 epochs of training.

Figure 5: Weights of hidden layer (no dropout) at end of epoch 3 (25%)

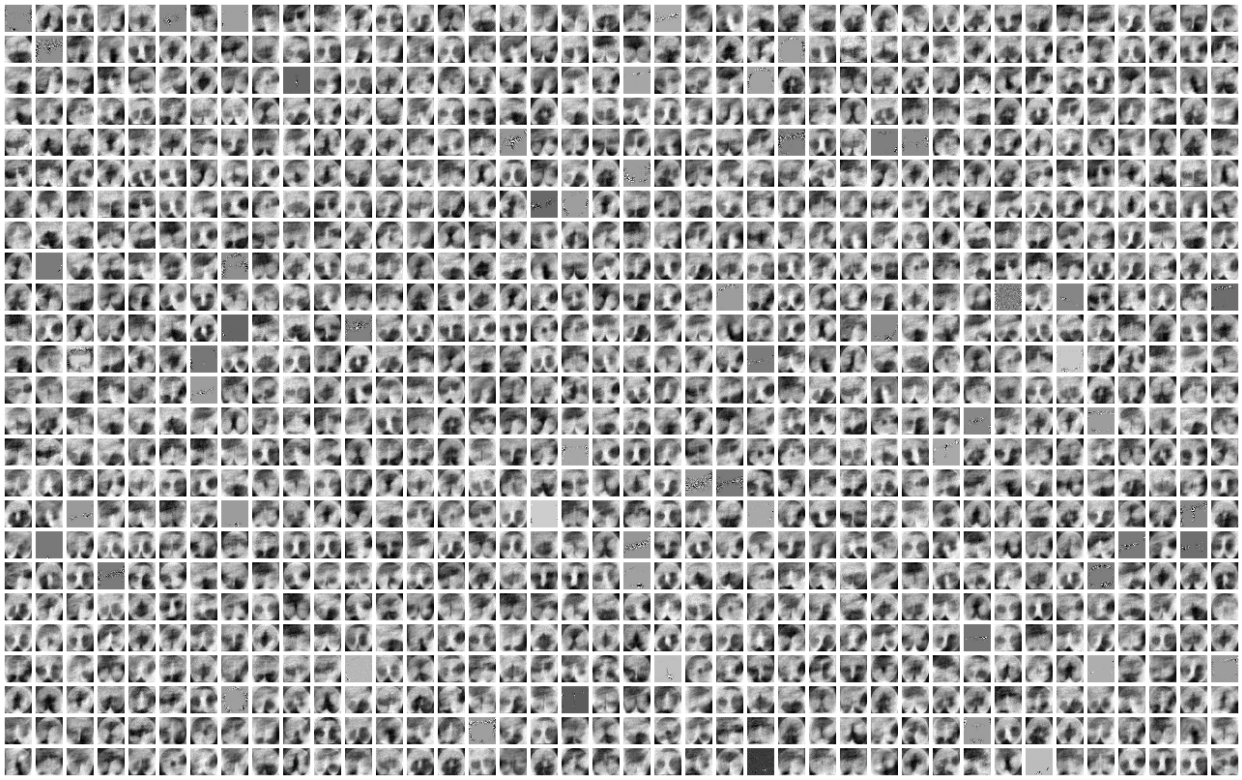


Figure 6: Weights of hidden layer (no dropout) at end of epoch 6 (50%)

Figure 7: Weights of hidden layer (no dropout) at end of epoch 9 (75%)

Figure 8: Weights of hidden layer (no dropout) at end of epoch 12 (100%)

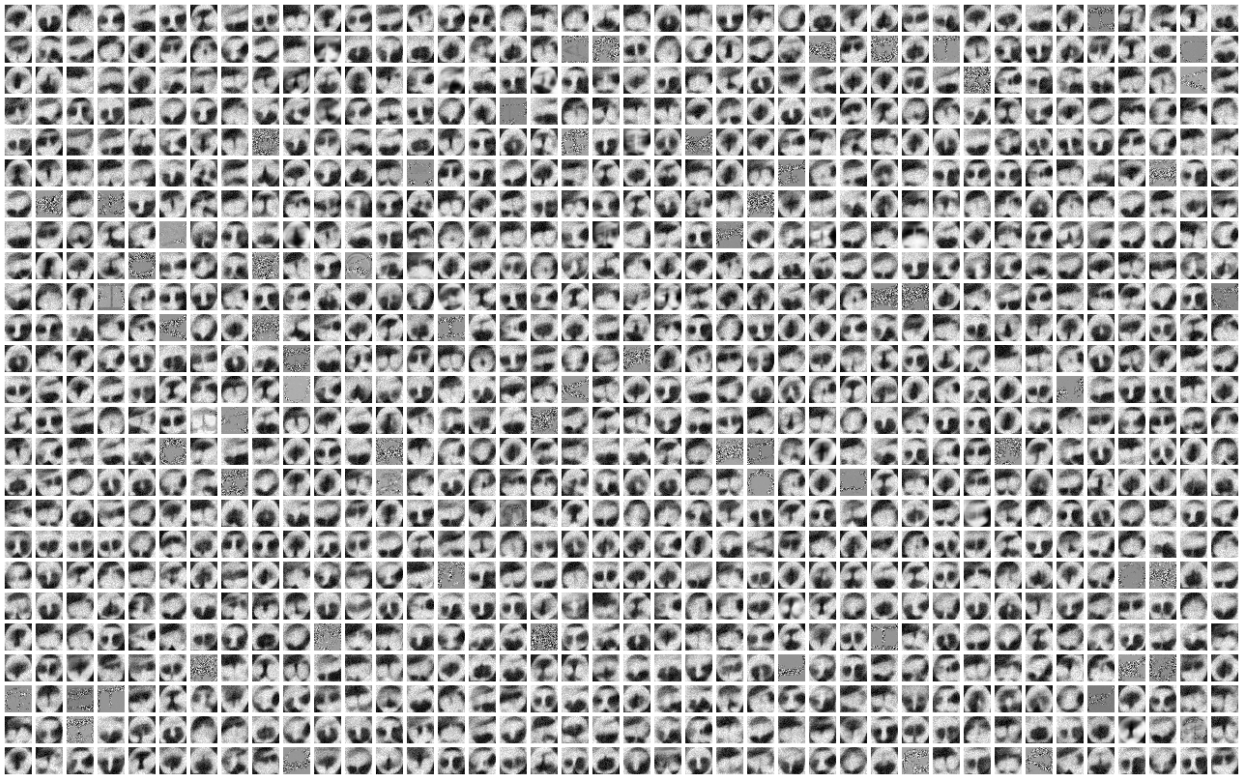


Figure 9: Weights of hidden layer (with dropout) at end of epoch 1 (25%)

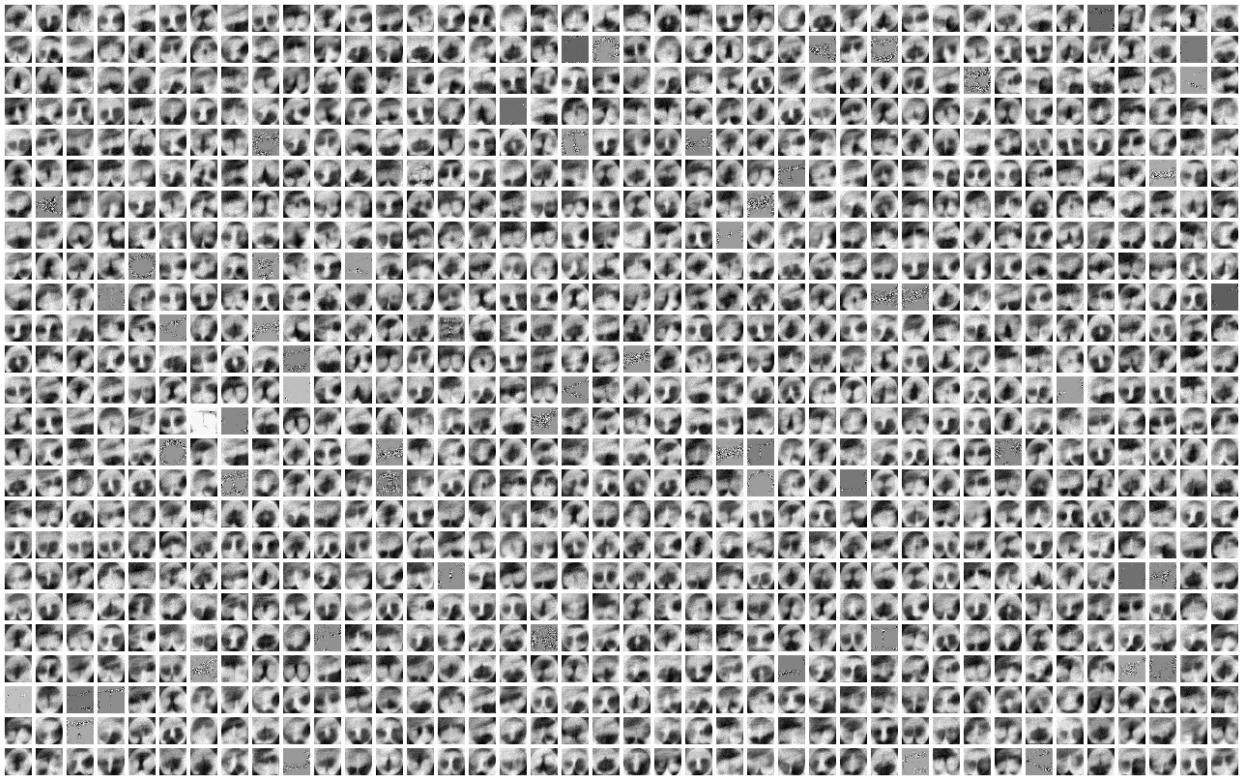


Figure 10: Weights of hidden layer (with dropout) at end of epoch 3 (50%)

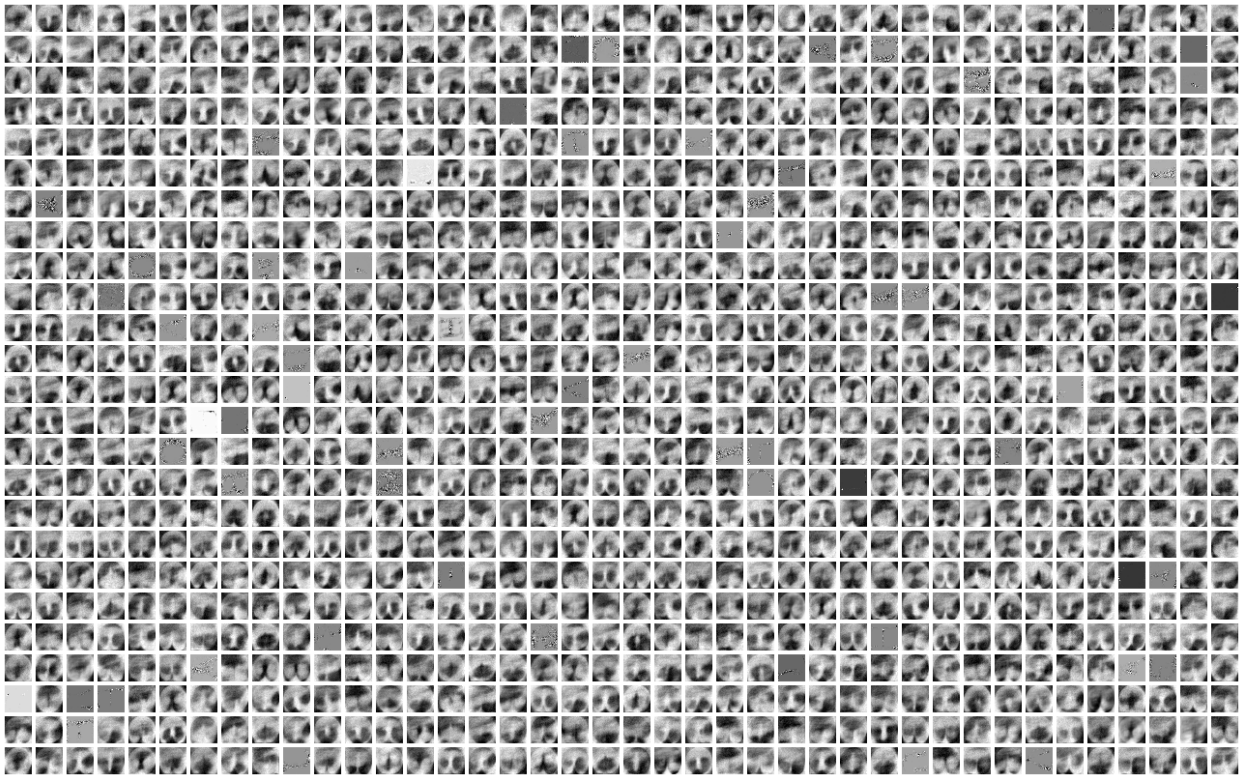


Figure 11: Weights of hidden layer (with dropout) at end of epoch 5 (75%)

Figure 12: Weights of hidden layer (with dropout) at end of epoch 7 (100%)

4.2 Exchange ideas among the groups

The following is the best set of hyperparameters so far:

- 5 hidden layers, 294 hidden units per layer
- Learning rate: 1.200×10^{-3}
- Weight decay: 1.836×10^{-4}
- Dropout: 0.5
- Validation and test set classification error: 6.20% and 6.94%