The code was written successfully, and the results produced from running the code closely resembled that which was shown in the assignment description.

To verify that the written code was indeed correct, the analytically calculated gradient was compared using the numerically computed gradient (using the slower, more precise methods supplied as a `ComputeGradsNumSlow.m`), using the method

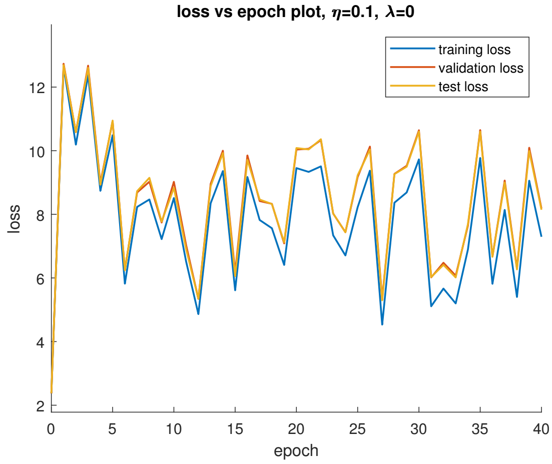
$$\text{alignment} = \frac{|g_a - g_b|}{\max 1e-6, |g_a| + |g_b|} \tag{1}$$

and the computed alignment was 4.88e-07 using the first full 100 samples in the `data_batch_1` dataset. The computed loss using this batch of data was 2.2827, which is an acceptable number, if compared to what was achieved in by the teachers in the miniGD.
Then, the network was trained on `data_batch_1`, validated on `data_batch_2`, and tested on `test_batch`. The final accuracy of the classsifer is shown in Table 1.

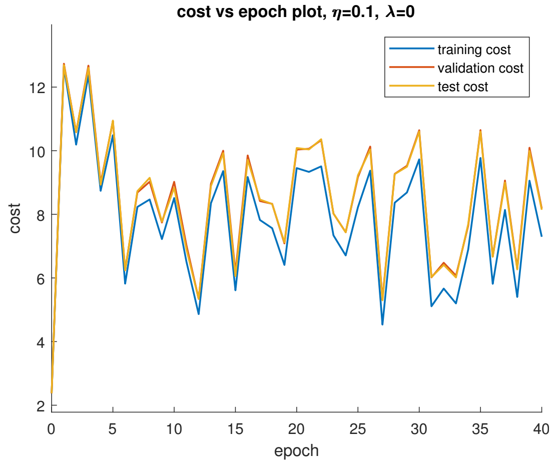
Table 1: Accuracy results for a single layer neural network, trained using mini-batch gradient descent, with sequential batches with size 100 and trained over 40 epochs.

	Training	Validation	Testing
$\lambda = 0, \ \eta = 0.1$	25.19%	21.83%	22.11%
$\lambda = 0, \ \eta = 0.01$	43.82%	37.18%	37.02%
$\lambda = 0.1, \eta = 0.1$	34.43%	32.20%	33.40%
$\lambda = 1, \ \eta = 0.01$	23.05%	21.79%	22.46%

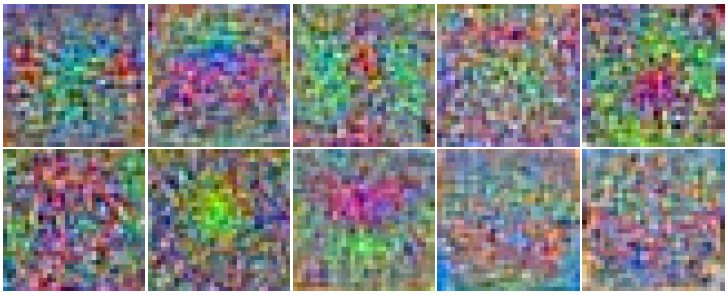
As we can see, the classifier performed the best without regulariser on the validation and test sets ($\lambda = 0$, $\eta = 0.01$). The regularised classifier with $\lambda = 0.1$ performed relatively well, which can also be seen in Figure 3, where shapes like a horse and a car are more distinguishable, if one compares against Figure 1 ($\eta = 0.1$) and Figure 4 ($\lambda = 1$).
The unregularised classifier with $\eta = 0.1$ (Figure 1) performed quite poorly too, which is more clearly visible in the cost/loss plots. The learning rate is too high, so the iterations bounce back and forth around values instead of converging as we want it to. Clearly the unregularised classifier $\eta = 0.01$ performs much better with a slower learning rate, and we also see in the plots of W in Figure 2 that shapes of horse and car are more distinguishable, albeit not as well as for the lightly regularised classifier.
Do note that the regularised weight vectors W are much smoother for the human eyes in the plots, ie objects are more visible and easy to see than their unregularised counterparts. Should one also compare the loss and cost plots of the regularised classifiers, the cost for the initial, randomized W matrix is much higher than for the non-regularised loss function. When the training algorithm has been running for just one epoch the cost seems to get much lower, and the cost function seems to be in general just a constant larger than the loss funtion for these particular hyperparameters.



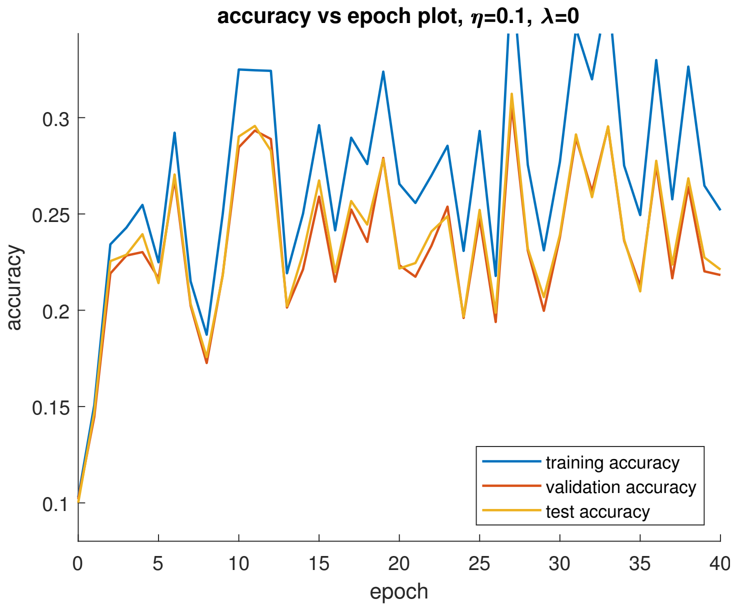
(a) Loss l as function of number of epochs for training.



(b) Cost J as function of number of epochs for training.

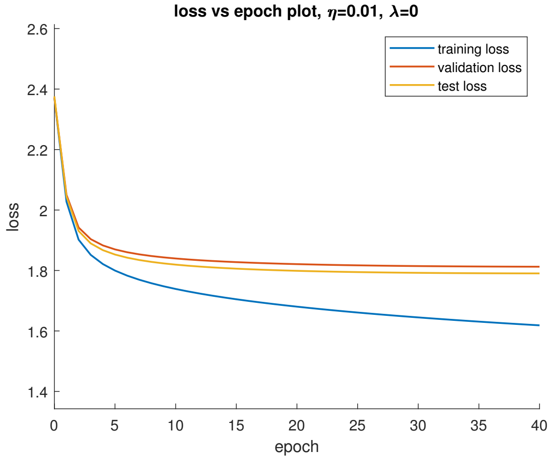


(c) Illustration of the trained W matrix.

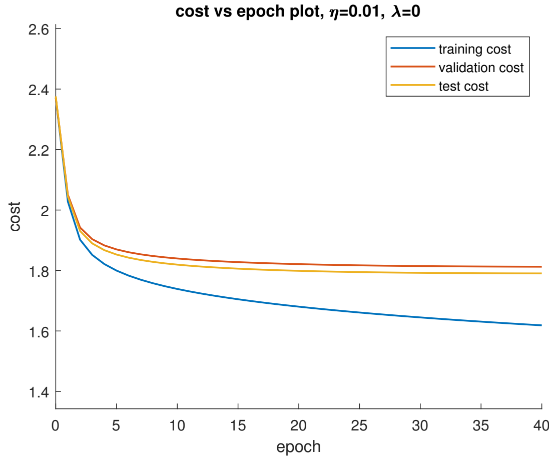


(d) Accuracy as function of number of epochs used for training.

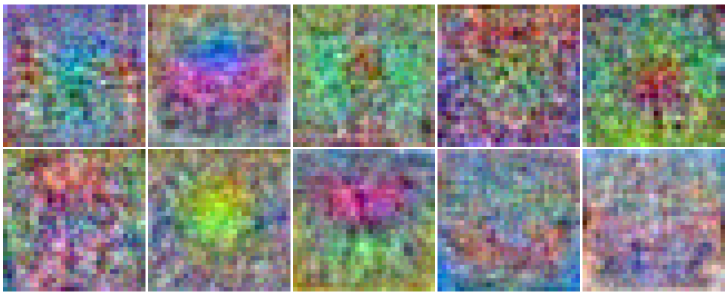
Figure 1: Plots of loss and cost function, as well as illustration of the trained W matrix for learning rate $\eta = 0.1$ and L_2 regularisation with $\lambda = 0$. Accuracy is also appended for good measure.



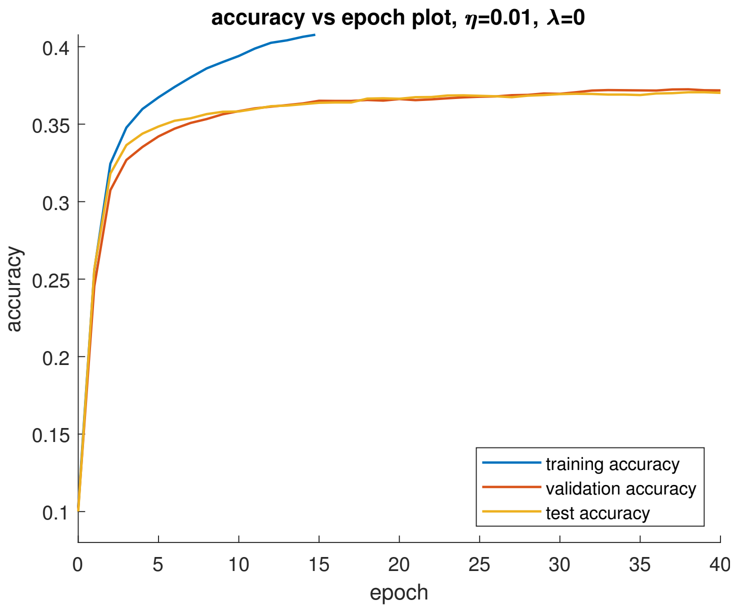
(a) Loss l as function of number of epochs for training.



(b) Cost J as function of number of epochs for training.

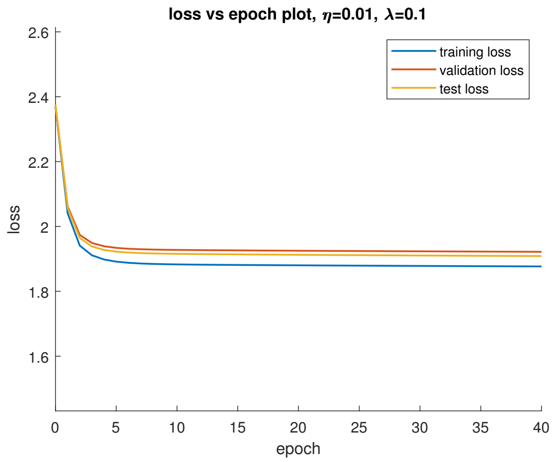


(c) Illustration of the trained W matrix.

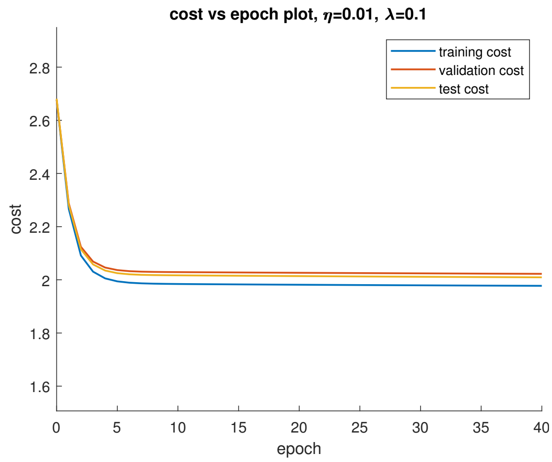


(d) Accuracy as function of number of epochs used for training.

Figure 2: Plots of loss and cost function, as well as illustration of the trained W matrix for learning rate $\eta = 0.01$ and L_2 regularisation with $\lambda = 0$.

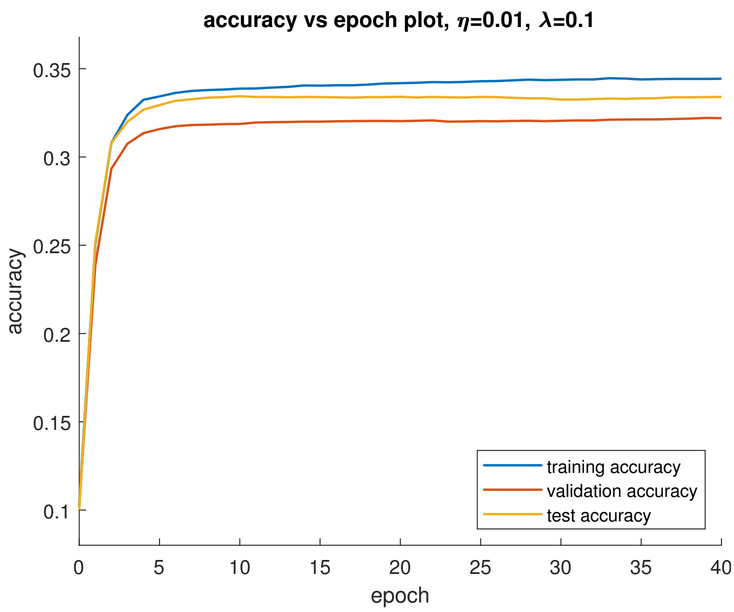


(a) Loss l as function of number of epochs for training.



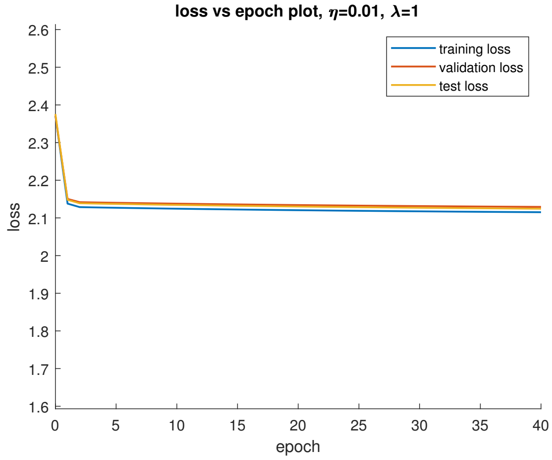
(b) Cost J as function of number of epochs for training.

(c) Illustration of the trained W matrix.

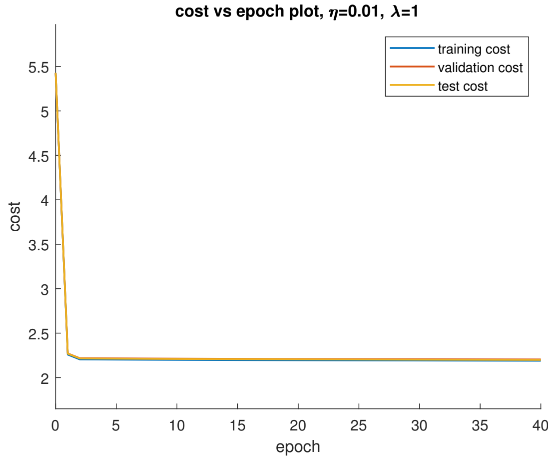


(d) Accuracy as function of number of epochs used for training.

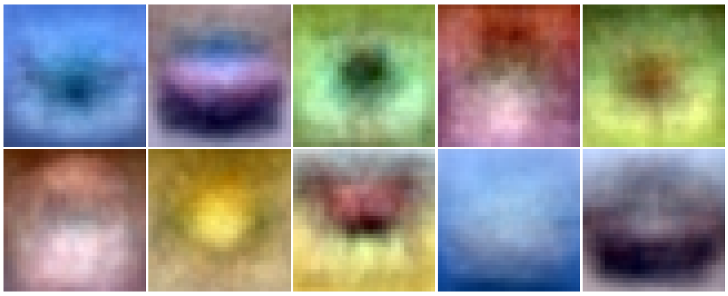
Figure 3: Plots of loss and cost function, as well as illustration of the trained W matrix for learning rate $\eta = 0.01$ and L_2 regularisation with $\lambda = 0.1$.



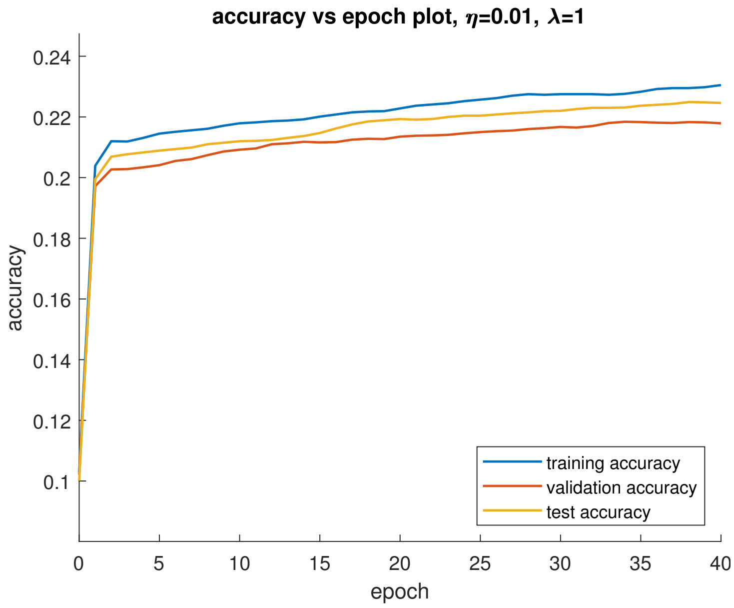
(a) Loss l as function of number of epochs for training.



(b) Cost J as function of number of epochs for training.



(c) Illustration of the trained W matrix.



(d) Accuracy as function of number of epochs used for training.

Figure 4: Plots of loss and cost function, as well as illustration of the trained W matrix for learning rate $\eta = 0.01$ and L_2 regularisation with $\lambda = 1$.