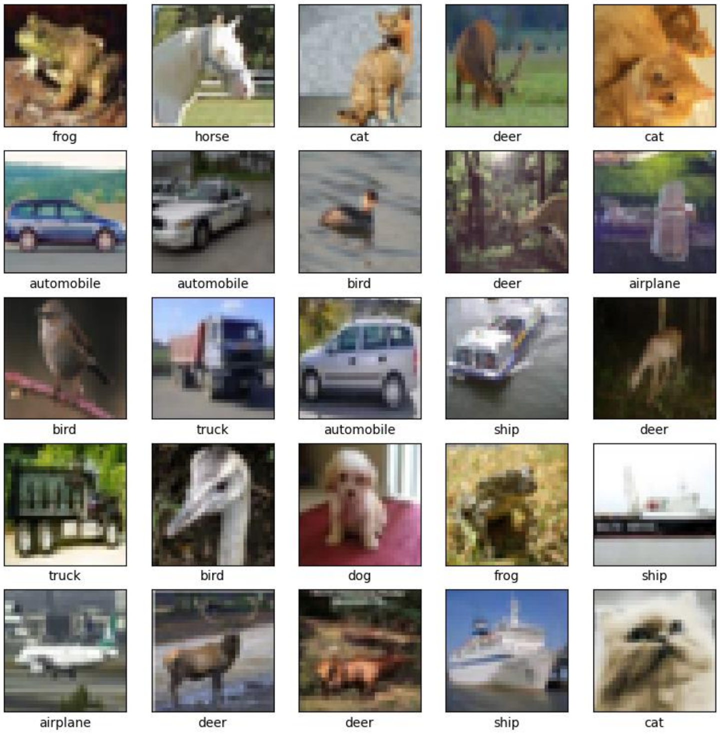
Description and analysis of the used data

The CIFAR-10 dataset comprises 60000 color images with a resolution of 32×32, separated into 10 classes (airplane, automobile, bird, cat, deer, dog, frog, horse, ship, truck), with 6000 images per class. It is already split into 6 batches with 10000 images each (~1000 from each class) that we will use in performing cross-validation. In order to be compatible with the softmax classifier, the images are vectorized into 32x32x3 = 3072 feature vectors.

The class distribution per batch is the following:

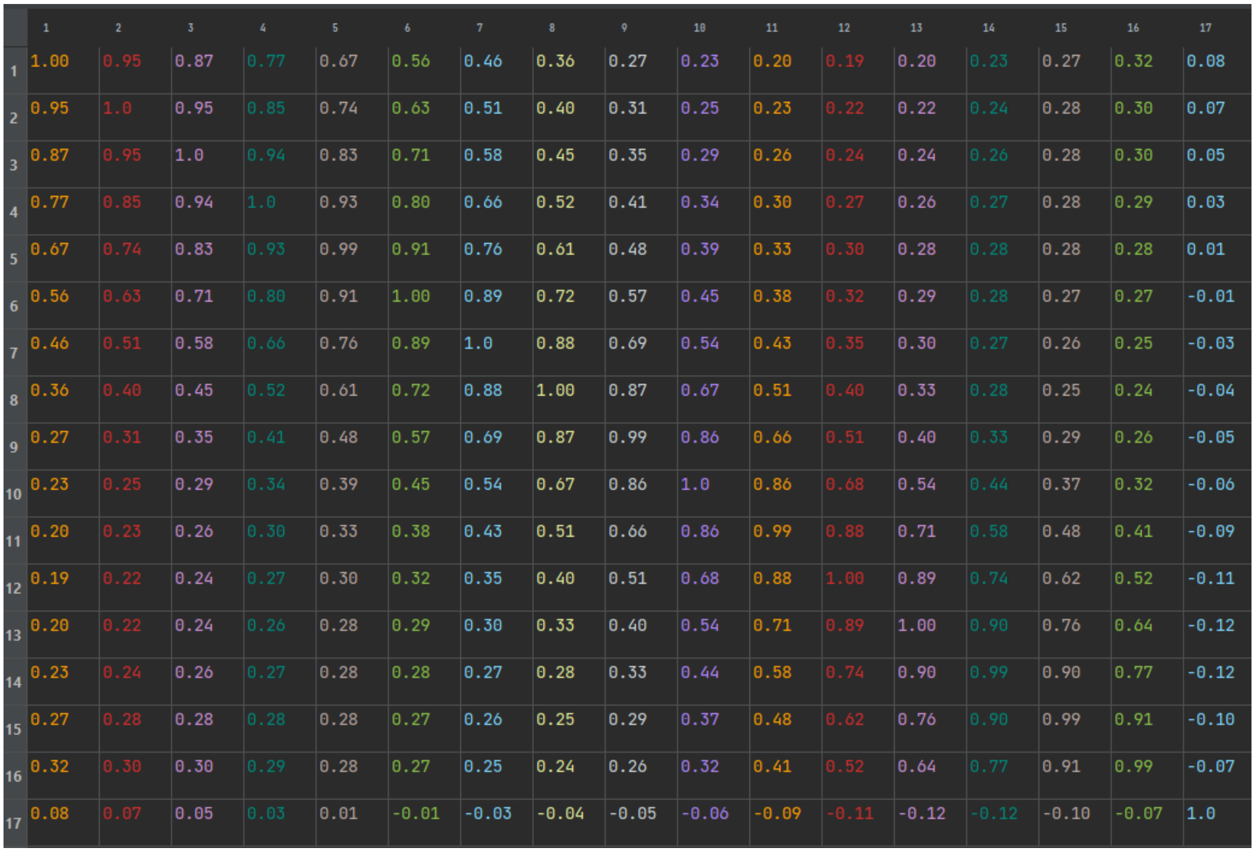
	Airplane	Automobile	Bird	Cat	Deer	Dog	Frog	Horse	Ship	Truck
Batch 1	1005	974	1032	1016	999	937	1039	1001	1025	981
Batch 2	984	1007	1010	995	1010	988	1008	1026	987	985
Batch 3	994	1042	965	997	990	1029	978	1015	961	1029
Batch 4	1003	963	1041	976	1004	1021	1004	981	1024	983
Batch 5	1014	1014	952	1016	997	1025	980	977	1003	1022
Batch 6	1000	1000	1000	1000	1000	1000	1000	1000	1000	1000

Example:



Description and analysis of the features used in learning

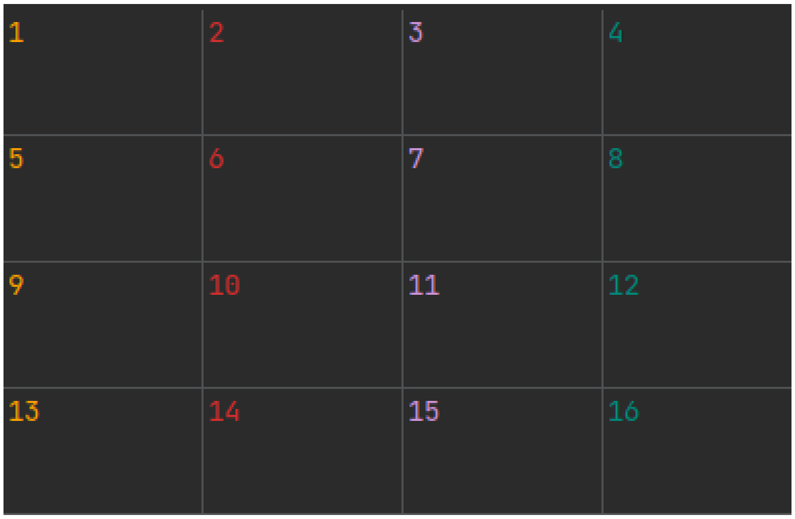
We perform a Pearson correlation on the features and the output. There are 3072 features, with 60k samples therefore there are too many computations. We reshape the 32x32x3 images to 4x4x192 volumes and perform a mean on the last axis in order to get a 4x4 volume (similar to a average pooling), from which we extract 16 features. Together with the output, we correlate 17 variables, hence the 17x17 matrix:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	1.00	0.95	0.87	0.77	0.67	0.56	0.46	0.36	0.27	0.23	0.20	0.19	0.20	0.23	0.27	0.32	0.08
2	0.95	1.0	0.95	0.85	0.74	0.63	0.51	0.40	0.31	0.25	0.23	0.22	0.22	0.24	0.28	0.30	0.07
3	0.87	0.95	1.0	0.94	0.83	0.71	0.58	0.45	0.35	0.29	0.26	0.24	0.24	0.26	0.28	0.30	0.05
4	0.77	0.85	0.94	1.0	0.93	0.80	0.66	0.52	0.41	0.34	0.30	0.27	0.26	0.27	0.28	0.29	0.03
5	0.67	0.74	0.83	0.93	0.99	0.91	0.76	0.61	0.48	0.39	0.33	0.30	0.28	0.28	0.28	0.28	0.01
6	0.56	0.63	0.71	0.80	0.91	1.00	0.89	0.72	0.57	0.45	0.38	0.32	0.29	0.28	0.27	0.27	-0.01
7	0.46	0.51	0.58	0.66	0.76	0.89	1.0	0.88	0.69	0.54	0.43	0.35	0.30	0.27	0.26	0.25	-0.03
8	0.36	0.40	0.45	0.52	0.61	0.72	0.88	1.00	0.87	0.67	0.51	0.40	0.33	0.28	0.25	0.24	-0.04
9	0.27	0.31	0.35	0.41	0.48	0.57	0.69	0.87	0.99	0.86	0.66	0.51	0.40	0.33	0.29	0.26	-0.05
10	0.23	0.25	0.29	0.34	0.39	0.45	0.54	0.67	0.86	1.0	0.86	0.68	0.54	0.44	0.37	0.32	-0.06
11	0.20	0.23	0.26	0.30	0.33	0.38	0.43	0.51	0.66	0.86	0.99	0.88	0.71	0.58	0.48	0.41	-0.09
12	0.19	0.22	0.24	0.27	0.30	0.32	0.35	0.40	0.51	0.68	0.88	1.00	0.89	0.74	0.62	0.52	-0.11
13	0.20	0.22	0.24	0.26	0.28	0.29	0.30	0.33	0.40	0.54	0.71	0.89	1.00	0.90	0.76	0.64	-0.12
14	0.23	0.24	0.26	0.27	0.28	0.28	0.27	0.28	0.33	0.44	0.58	0.74	0.90	0.99	0.90	0.77	-0.12
15	0.27	0.28	0.28	0.28	0.28	0.27	0.26	0.25	0.29	0.37	0.48	0.62	0.76	0.90	0.99	0.91	-0.10
16	0.32	0.30	0.30	0.29	0.28	0.27	0.25	0.24	0.26	0.32	0.41	0.52	0.64	0.77	0.91	0.99	-0.07
17	0.08	0.07	0.05	0.03	0.01	-0.01	-0.03	-0.04	-0.05	-0.06	-0.09	-0.11	-0.12	-0.12	-0.10	-0.07	1.0

We can notice a correlation between features coming from the same area. Also, there does not seem to be any correlation between the output and the features, meaning that probably all features are needed to predict the labels.

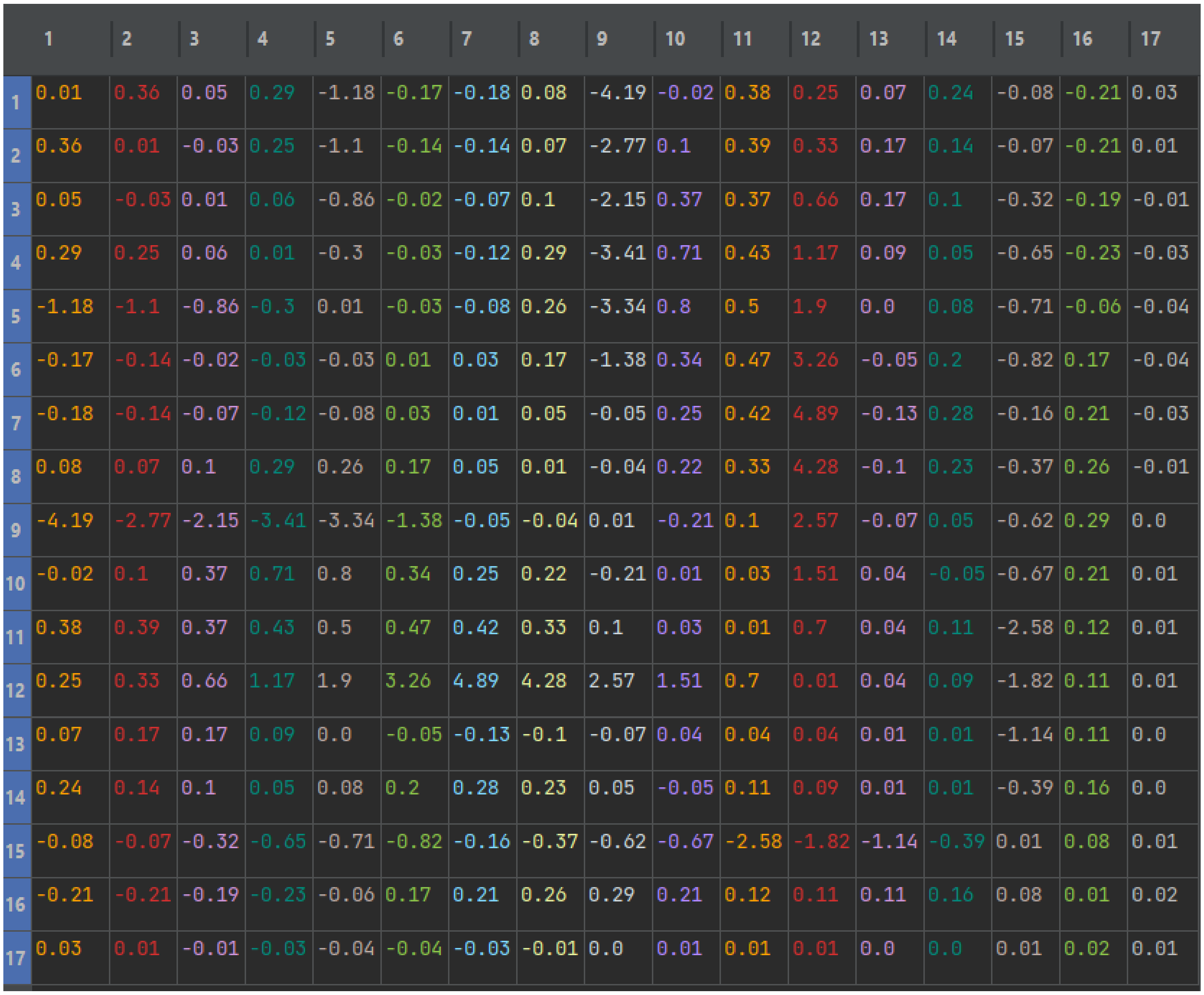
These are the features in 2 dimensions:



1	2	3	4
5	6	7	8
9	10	11	12
13	14	15	16

For independence values close to zero mean the null hypothesis, which is that the 2 variables are not related, other values mean that they are related.

The same for independence; we can notice that the input features are not related to the output:



	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17
1	0.01	0.36	0.05	0.29	-1.18	-0.17	-0.18	0.08	-4.19	-0.02	0.38	0.25	0.07	0.24	-0.08	-0.21	0.03
2	0.36	0.01	-0.03	0.25	-1.1	-0.14	-0.14	0.07	-2.77	0.1	0.39	0.33	0.17	0.14	-0.07	-0.21	0.01
3	0.05	-0.03	0.01	0.06	-0.86	-0.02	-0.07	0.1	-2.15	0.37	0.37	0.66	0.17	0.1	-0.32	-0.19	-0.01
4	0.29	0.25	0.06	0.01	-0.3	-0.03	-0.12	0.29	-3.41	0.71	0.43	1.17	0.09	0.05	-0.65	-0.23	-0.03
5	-1.18	-1.1	-0.86	-0.3	0.01	-0.03	-0.08	0.26	-3.34	0.8	0.5	1.9	0.0	0.08	-0.71	-0.06	-0.04
6	-0.17	-0.14	-0.02	-0.03	-0.03	0.01	0.03	0.17	-1.38	0.34	0.47	3.26	-0.05	0.2	-0.82	0.17	-0.04
7	-0.18	-0.14	-0.07	-0.12	-0.08	0.03	0.01	0.05	-0.05	0.25	0.42	4.89	-0.13	0.28	-0.16	0.21	-0.03
8	0.08	0.07	0.1	0.29	0.26	0.17	0.05	0.01	-0.04	0.22	0.33	4.28	-0.1	0.23	-0.37	0.26	-0.01
9	-4.19	-2.77	-2.15	-3.41	-3.34	-1.38	-0.05	-0.04	0.01	-0.21	0.1	2.57	-0.07	0.05	-0.62	0.29	0.0
10	-0.02	0.1	0.37	0.71	0.8	0.34	0.25	0.22	-0.21	0.01	0.03	1.51	0.04	-0.05	-0.67	0.21	0.01
11	0.38	0.39	0.37	0.43	0.5	0.47	0.42	0.33	0.1	0.03	0.01	0.7	0.04	0.11	-2.58	0.12	0.01
12	0.25	0.33	0.66	1.17	1.9	3.26	4.89	4.28	2.57	1.51	0.7	0.01	0.04	0.09	-1.82	0.11	0.01
13	0.07	0.17	0.17	0.09	0.0	-0.05	-0.13	-0.1	-0.07	0.04	0.04	0.04	0.01	0.01	-1.14	0.11	0.0
14	0.24	0.14	0.1	0.05	0.08	0.2	0.28	0.23	0.05	-0.05	0.11	0.09	0.01	0.01	-0.39	0.16	0.0
15	-0.08	-0.07	-0.32	-0.65	-0.71	-0.82	-0.16	-0.37	-0.62	-0.67	-2.58	-1.82	-1.14	-0.39	0.01	0.08	0.01
16	-0.21	-0.21	-0.19	-0.23	-0.06	0.17	0.21	0.26	0.29	0.21	0.12	0.11	0.11	0.16	0.08	0.01	0.02
17	0.03	0.01	-0.01	-0.03	-0.04	-0.04	-0.03	-0.01	0.0	0.01	0.01	0.01	0.0	0.0	0.01	0.02	0.01

Description and analysis of the proposed solution

The solution consists of 10 perceptrons stacked, one for each class. Their weights are updated using stochastic gradient descent on batches. Basically, the update is done after seeing a batch of examples. This is done for several passes through the dataset. After such a pass or epoch, we also compute the loss on the validation set. If this loss does not improve after 3 epochs, we halve the learning rate. The accuracy is also computed on both datasets, and the model is saved when the accuracy on the validation set improves.