# Deep Learning - H.W 3 - 236606

May 21, 2018

## 1 Submission Instruction

1. Submission deadline: 5/6/2018 23:59
2. TA in charge: Yonatan Geifman
3. Submit a zip file containing only your code and a report summarizing your work, in the webcourse site: http://webcourse.cs.technion.ac.il/236606/
4. Answers must be submitted as a PDF file, typed using any document editor (not a scan of handwritten answers)
5. Submission is in pairs only
6. No late submissions

## 2 Convolutional neural networks (50 points)

Plan an architecture and a training procedure for a convolutional network on CIFAR-10 dataset (you can read about the CIFAR-10 dataset here). Design and train your network so that it will satisfy the two following goals:

- The final accuracy on the test-set should be $> 85$
- The number of trainable parameters (weights and biases) should be as small as possible and at least $< 50$K

## 3 Transfer Learning

It is widely known (and was mentioned in class) that Convolution layers that learned on one dataset of images can provide useful representations (Features) for other datasets. In this question we will investigate some transfer learning approaches. Consider a VGG based architecture (VGG will be mentioned next class, you can also read about it by yourself) for the CIFAR-100 dataset, keras code can be downloaded from here. In this question we will build the best classifier as possible for CIFAR-10 while using a small sample of the training set

and a trained model for CIFAR-100. First you should create a smaller version of CIFAR-10 dataset by running:

```
from sklearn.model_selection import train_test_split
X_train_small, _, y_train_small, _ = train_test_split(X_train, y_train,
train_size=100, random_state=42,stratify=y_train)
```

This will sample a training set of 500 samples in a stratified fashion. We will use a trained model of CIFAR-100 and a sampled dataset and try to reach maximum accuracy by comparing 3 approaches.

### 3.1 Transfer Learning by Fine-Tuning (20 points)

For the first approach we will use a method named "fine-tuning". We take a trained model for CIFAR-100, replace the last fully-connected layer with a new initialized layer, where the output size is the number of classes in cifar-10 (10). We will then freeze all other layers and train the last layer only with the small (sample) available dataset for CIFAR-10. Repeat this procedure for datasets sampled from cifar-10 of the sizes 100, 1000, 10000. Make sure you use the random seed 42 so we can reproduce your results.

### 3.2 Transfer Learning with Embedding Nearest Neighbors (20 points)

Another approach for transfer learning in CNNs is by using the k-nearest-neighbors (KNN) algorithm on embedding activations. A DNN embedding is typically considered as the second last layer in the network. This layer is known to represent semantic relatedness where semantically similar images embedded closer to each other. For transfer learning we take the training set of the new dataset (CIFAR-10) and map it to the embedding domain of CIFAR-100. On inference we map a new instance to this domain and run the KNN algorithm with the training set. Repeat this procedure for samples from cifar-10 of the sizes 100,1000,10000.

In your report compare these two transfer learning methods (fine-tuning and KNN) and evaluate the results, try to motivate your result from a statistical learning theory perspective.

### 3.3 Transfer Learning - Your Own Solution (10 points)

Try to develop a different approach for transfer learning. In your solution consider effects as catastrophically forgetting (google it), over-fitting, bias-variance trade-off etc. In this section you can also use the Cifar-100 training set.