

1 Neural networks from scratch

For this assignment, feel free to use a verbose iPython notebook that will also serve as a report. Before and after each step, feel free to insert copious comments to explain your approach and note down your observations. Additionally, comment EACH LINE OF CODE to make sure that it is clear to the evaluator what you have done.

Note: You can consult your classmates and the internet, but the final solution has to be written by you from scratch. Detection of copying will attract an F grade in the course without any warning.

1.1 Implementation of layers [7]

Implement the follow layers as classes using python and numpy:

- (i) Dense (fully-connected) layer that transforms an input vector of size N_{l-1} into an output vector of size N_l using weights and biases ($\mathbf{W}\mathbf{x} + \mathbf{b}$)
- (ii) ReLu and Softmax activation functions that operate point-wise on an input vector.
- (iii) Crossentropy loss.

The classes should have:

- (i) Flexibility to able to take variable sized inputs.
- (ii) A forward pass function that gives the desired output and a cache output that will be used in the backward pass
- (iii) A backward pass function that computes derivatives with respect to both weights/biases and inputs

1.2 Image classification [4]

Take any two classes from CIFAR 10 dataset, extract its features and design an artificial neural network using the layers implemented in part 1 to achieve the best possible accuracy. Repeat the same for 5 classes. Compute the number of parameters in the neural network and train it using stochastic (batch-wise) gradient descent until validation loss does not improve.

2 Using deep neural network libraries

Built a convolutional neural network with PyTorch or Tensorflow. [4]

- (i) Achieve similar performance using 50% of the parameters used in the ANN designed in task 1.
- (ii) Achieve similar performance using 25% of the parameters.