Deep Learning

Assignment 1

Author: Angeliki Mylonaki

Image Classification using Neural Networks

November 18, 2018

Overview

In this exercise we are using the Cifar-10 dataset as input to a Neural Network so as to classify images to categories based on what they show.

In this report, we are mainly reporting the steps followed to reach the final neural network and why each choice was made.

Data Preprocessing

In order to pre process our data we tried:

- L1,L2 regularization
- MinMax regularization
- Dividing each point with 255, since our values range between(0,255)

In our test Min Max scaler performed the best, so this is our choice moving forward.

Preprocessing is done in order to prevent our activation functions from producing extreme values really often which gradient problems.

Activation Function

In order to decide which activation function to use in our perceptrons, we tried several “must have” activation functions. Below we describe the reason behind our function choices, apart from them being popular in the neural network universe:

SIGMOID activation function is used in models that want to predict probability of an item belonging to a class, since it produces values between (0,1). Commonly used in binary classification problems

SOFTMAX activation function is used similarly to sigmoid, but is a more generalized activation function, used for multiclass cases.

TANH is a scaled sigmoid function, with stronger gradient.

RELU currently the most famous activation function, useful since it “deactivates” some neurons in the network, making it more efficient and sparse.

After experimenting, our Neural Network was found to behave better using Relu activation on its hidden layers and softmax on the output layer

Layer Architecture and Regularization

The number and size of hidden layers were chosen based on the network’s performance and accuracy.

We are also using Dropout Regularization method to avoid overfitting and strengthen our predictions. The performance of the network was significantly better when using dropout in all our hidden layers.

Model Selection

In order to decide on the final form of our Model, we tried several Optimization Algorithms using their default values upon a Neural Network with the same number of hidden layers and activation functions. We then kept the Optimizer that performed the best and tuned its parameters. In this report and also the corresponding jupiter notebook we present only the final

four optimizers that behaved the best. So optimizers like RMSprop, Adagrad. Adadelta etc are not discussed.

Below you can find the reason behind the optimizers being used, apart from being the best ones for our classification problem:

SGD it is the most popular Optimization method. Introduces wight updates to minimize loss function. Can be tuned to produce even better results via hyperparameters:

- Learning rate: how big of the steps to use to approach the global minimum
- Momentum: accelerating SGD to move faster towards the right direction. This means that it updates parameters only for some examples, so as to reduce unnecessary updates.
- Nesterov Momentum: Optimization of momentum. Tries to prevent the case where momentum while approaching the minima is too large causing as to miss it

Adam computes adaptive learning rates and converges really fast

Adamax similar to Adam, but better used when wanting sparse parameter updates. So parameter updates are influenced by less gradients and therefore are less susceptible to gradient noise.

Nadam is basically Adam with Nesterov's Momentum.

Model Tuning

Since we have concluded on our Model and Neural Network Architecture we are now selecting the best hyperparameters to tune the SGD optimizer.

In our case, the model seems to behave better with the values for the parameters as listed below:

Parameter	Best Value
Learning rate	0.01
decay	0.0

momentum	0.9
nesterov	False