Machine Learning- HW4

Esti Gitman 316413210

Orel Shahadi 206092090

PART 1: Fully connected layers

Change the activation functions to LeakyRelu or tanh or sigmoid. Name the new model `new_a_model`. Explain how it can affect the model.

The sigmoid activation function will change an input value into an output in the interval [0,1].

Advantage:

- Output normalization helps to maintain the output in a particular range when encountered with $(-\infty, +\infty)$ as in the linear function. That way, this helps getting the good efficiency and accuracy of the model.

Disadvantages:

- A Non zero Centered Activation Function
- When used in a neural network, because it transforms values larger than 1 to 1 and negative values to 0, It leads to saturation around 1 and 0 so the NN is very sensitive to changes. Therefore, using a sigmoid function makes it difficult for the network to adapt (it becomes harder when the number of hidden layers goes up).
- The sigmoid function can also cause neural networks to suffer from the vanishing gradient problem since error is backpropagated through the layers (when multiplying these gradients, the product of many values smaller than 1 goes to zero very quickly). The result is a decrease with each hidden layer.

ReLU takes an input and outputs the same input if positive and outputs 0 if negative.

Advantages:

- ReLU does not have a vanishing gradient because maximum threshold values are infinity. It enables high output prediction accuracy and good efficiency.
- The ReLU function is much less computationally taxing than sigmoid, because it does not activate all the neurons at the same time.

- Accelerates the convergence of stochastic gradient descent compared to the sigmoid function.
- Less computational taxing because it needs to pick max(0,x) instead of performing expensive exponential operations as in sigmoid.

Disadvantages:

- ReLU has a zero value region. On the one hand, this region enables the speed of the process, but on the other hand there is not much learning in this area (the weights are not updated during backpropagation), so there are neurons that never gets activated. It can reduce the learning rate.
- ReLU is not zero- centered. This fact damages the NN performance, causing sharp gradient fluctuations that affects the weights updates.

Task 3: ***Number of epochs.***

Train the new model using 25 and 40 epochs. What difference does it make in term of performance? Remember to save the compiled model for having initialized weights for every run as we did in tutorial 12.

In both cases, there were similar values of accuracy and loss. This is due to gradient descent limitations and/or a bad dataset.

Build the `model_relu` again and run it with a batch size of 32 instead of 64. What are the advantages of the mini-batch vs. SGD?

There is a trade-off between aaccuracy of parameter update and computational time to perform an update.

Batch gradient descent (BGD) can be very slow for large datasets. However, the Accuracy of parameter update is high.

Stochastic gradient descent (SGD) frequent updates with a high variance that cause the objective function to fluctuate heavily, resulting in a low accuracy. However, it has quick performance time.

Mini-batch gradient descent: a compromise between the two, allows both high accuracy of parameter update and Computational time to perform an update. Thus, being a better gradient descent than SGD.

Build the `new_a_model` again and add batch normalization layers. How does it impact your results?

Batch normalization refers to change in the distribution of inputs during training.

Advantages:

- It helps reduce the covariance shift- it has the effect of stabilizing the learning process and reducing the number of training epochs required to train deep networks.
- Provides some regularization, reducing generalization error.
- Reduces the vanishing gradients problem
- Less sensitive to the weight initialization
- Able to use much larger learning rates to speed up the learning process

Disadvantages:

- Slower predictions due to the extra computations at each layer
- Unstable when using small batch sizes
- Different results in training and inference
- Not good for online learning
- Not good for Recurrent Neural Networks

PART 2: Convolutional Neural Network (CNN)

Task 1: 2D CNN

Have a look at the model below and answer the following:

- How many layers does it have?
- How many filters in each layer?
- Would the number of parameters be similar to a fully connected NN?
- Is this specific NN performing regularization?

Answers:

- There are 24 layers.
- Conv2D_1 has 64 filters, Conv2D_2 has 128 filters, Conv2D_3 has 128 filters, Conv2D_4, Conv2D_5 have 256 filters each.
- We will use CNN when there are stationary signals, (their statistics do not change over time, so using the same filter is acceptable). This also applies to images, when we apply the same filter to different images.

The advantage of the method is the reduction of the parameters studied. (Compared to a fully connected network, where there are N weights per neuron).

- There are 2 regularization expressions: the first is regularization by the L2 norm using a decay rate of 1e-2, and the second appears in the network definition: there is a condition of "drop". For drop = True, a dropout is defined. Dropout is one of the regulatory methods as well. During training, we randomly "turn off" neurons: turn their weights values to zero randomly at each batch. In this case, we turned off 30% of the neurons randomly. This prevents weights from getting too high values while using regulation.

That's all folks ☺