A Comparative Study of Neural Network Optimizers

Transcript

Hiba Khurshid

Slide 1:

Hi everyone, this is Hiba Khurshid and the topic of today's tutorial is a comparative study of neural network optimisers so in the study what we have done is we have taken a cifar-10 dataset and applied different optimisers and compare them with each other on the basis of different parameters so now let's get started

Slide 2:

So What are neural networks? Neural networks are basically inspired by our human brain just like in our brains we have neurons and they are all connected to each other and send signal similarly in artificial neural networks we have neurons that are connected and each connection has a weight. The neural network learns by adjusting these ways in order to minimise error in prediction. Optimisation place a very crucial role for accurate predictions. so one of the widely used technique for optimisation is gradient Descent along with its advanced variance like Adam RMSprop and others

Slide 3:

So now let's talk about the gradient decent optimisation the gradient descent optimisation works by iteratively updating the weights of a model to minimise the loss function. The loss function is the measure of how far the models prediction are from the actual values.

So now let's look at the formula here we are calculating the slope so in this region the slope is going to be negative and this negative slope will multiply with this negative and we will move in this direction in the positive direction but over here the slope is going to be positive and this positive will multiply with this negative and we will move in the opposite direction and by doing this we will eventually reach our goal which is the global minima

Slide 4:

So in gradient descent in one epoch it takes a complete dataset and then update the weight. if we have a very large data set the computational cost will increase whereas the mini batch gradient descent instead of taking a complete dataset random subset, the mini batches of the data is used

so we divide the data set into mini batches and for each batch we will compute the gradient of the loss function and then update the weights.

Slide 5:

If you look at the graph here we observe the zigzag pattern. This zigzag pattern is due to the randomness of the mini batches the advantages of mini batch gradient is that it reduces the memory requirement because it is working in mini batches and also it converges fast if you compare with the batch gradient decent .

The best case scenario to use mini batch gradient is where we have large data set and where the full batch update are computationally very expensive and it works very well for training the deep neutral networks

Slide 6:

In order to understand Adam optimiser and RMS prop first we need to look at the exponentially weighted moving average

Now look at this graph if you're moving forward in time we encounter more points the average is calculated at each point and it is calculated in such a way that the new points have high weight and very less weight is given to the old points so in this graph at the given timestamp, we give higher weights to these red points and lessweightage to these blue points

So now what happens when we take exponentially weighted moving average of these points what will happen is the average of these point in a vertical direction is going to be approximately zero and will be higher in the horizontal direction, so the net result will be that it moves very little in the vertical direction and it will be moving mostly in the horizontal direction

Slide 7:

Root mean Square propagation is an adaptive learning rate optimisation algorithm. It works by dividing the learning rate by a moving average of a squared gradient which helps to normalise the updates and stabilise training. So we have looked in the mini batch, that it behaves in this zig zag motion so why it is moving in this motion is because it is taking very high step in vertical direction and a very little step in a horizontal direction.So if this value is high, this value is going to be high and eventually the overall VT is going to be high and we are dividing this VT in this equation so the overall value of WT is going to be lower. So the movement in the vertical direction is going to be lower. Also, we are using this epsilon term to avoid overshooting

Slide 8:

So here how it works is that it maintains a moving average of a square gradients divide the learning rate by the root means square of these averages. and it helps to handle non-stationary objectives and gives a smooth convergence

Some advantages of RMS prop is that it adapt learning rate for each parameter dynamically Effective for non-stationary objectives, prevent oscillation in the optimisation process and requires minimal tuning of hyper parameters.

Slide 9:

moving on to the Adam optimiser the Adam optimiser combines the strength of both the momentum and the RMS prop to achieve adoptive learning rates for each parameter.Other advantages that it is very robust to sparse gradient and noisy data. works very well with the large data set and complex neural architectures and the bias correction improves early stage optimisation

Slide 10:

In other optimisation techniques, we use the fixed learning rate whereas adagrad works by scaling the learning rate for each parameter based on the sum of squares of its past gradient. larger parameters updates in the past gets smaller learning rate while those with smaller updates gets larger learning rate which helps in balance learning across all parameters

Slide 11:

For the comparison we use the cipher 10 dataset and it is composed of 60,000 images categorized into 10 classes for example cat dogs boats, cars airplanes et cetera each image is RGB colour and has 32×32 pixels this data is widely use for benchmarking image classification model

The matrices that we use in this study to compare different optimizers are, accuracy on test data their training time peak memory usage and their convergence behavior.

Slide 12:

So now let's look at the convergence behaviour of different optimisers if you look at the adam optimiser the training loss shows that it starts with a high value and then it decreases to the low value that shows a strong convergence and it converge smoothly downward to around 0.655 but if you look at the mini batch SGD the training loss shows steady decrease from 0.5 to 0.3 but if you look at the validation loss of the mini batch SGD it shows divergence. it is increasing which indicate a potential over fitting. In the RMS prop the training law is gradually decreasing whereas the validation laws shows instability with a peek around Epoch 2 to 3. Now, if you look at the adagrad training and validation loss remain relatively flat indicating the most stable convergence among all optimisers

Overall, the validation losses show more variation. Most significant learning occur occurs in the first 4 epochs and the best convergence pattern is shown by Adam optimiser with steady improvement and reasonable validation loss

Slide 13:
This graph helps visualise the trade of between computational efficiency and resource consumption, for different optimism so the mini batch SGD is efficient in terms of training time but slightly less memory efficient Adam optimiser has a high training time and memory usage indicating its complexity. RMS prop is moderate in both training time and memory usage offering a balanced trade off ada grad works well in terms of memory efficiency but has a slightly slower training time then min batch sgd

Slide 14:

Now, if you look at the accuracy result on the training data set at a great outperformed all the other optimise with the accuracy of 0.74 then we have Adam optimiser with the accuracy of 0.71 and then with a slightly less accuracy is RMS prop with 0.70 mini batch sgd performed relatively poor than other optimises with the accuracy of 0.68 not if you look at the accuracy result on the training