For testing this exercise, I have used 3 batch\_sizes: 32,64,120 with LR: 0.001,0.001,0.1 respectively.

Chart, line chart

Description automatically generated

Training loss and accuracy when the model is trained using different batch sizes(SGD).

It is known that simply increasing the learning rate does not fully compensate for large batch sizes. However, it is well known that too large of a batch size will lead to poor generalization. It is generally accepted that there is some “sweet spot” for batch size between 1 and the entire training dataset that will provide the best generalization. This “sweet spot” usually depends on the dataset. Here we are using small dataset- iris so it is quite difficult to conclude anything. Let’s see what results we got from above graphs:

1. When batch\_size is 32, blue line shows that risk is higher than other two. Initially blue line has same risk as other two but after few epochs it did not change much. Whereas for batch\_size 120, risk is same for almost 1/3rd of epochs and then it is decreasing for quite some time and then for last few epochs it is constant.
2. Accuracy is initially same for all 3 batch sizes and then for batch\_size 120 it is increasing for few epochs compared to other two and after that it in decreasing and then increasing till it became constant for last few epochs. For blue line, batch\_size is 32 and accuracy line is constant for quite many epochs and for few epochs in between it increased, became constant and at the end it increased and gave almost same accuracy as batch\_size=64. When batch\_size is 64 accuracy initially is constant and then it is increasing slowly til it became constant for last epochs.
3. When batch\_size = 32 and lr=0.001, optimizer=SGD we get accuracy from 20 percent to 100 percent whereas test accuracy is started giving numbers from 20 percent to 96 percent.
4. When batch\_size = 64 and lr=0.01, optimizer=SGD we get accuracy from 37 percent to 100 percent whereas test accuracy is started giving numbers from 86 percent to 100 percent.
5. When batch\_size = 120 and lr=0.1, optimizer=SGD we get accuracy from 20 percent to 98 percent whereas test accuracy is started giving numbers from 86 percent to 100 percent.

Let us now analyze ADAM optimizer outputs for same batch size and lr:

Chart

Description automatically generated

Training loss and accuracy when the model is trained using different batch sizes(ADM).

1. When batch\_size = 32 and lr=0.01, optimizer=ADM we get accuracy from 40 percent to 100 percent whereas test accuracy is started giving numbers from 50 percent to 100 percent.
2. When batch\_size = 64 and lr=0.01, optimizer=SGD we get accuracy from 58 percent to 100 percent whereas test accuracy is started giving numbers from 63 percent to 100 percent.
3. When batch\_size = 120 and lr=0.1, optimizer=SGD we get accuracy from 20 percent to 98 percent whereas test accuracy is started giving numbers from 63 percent to 100 percent.
4. Risk analysis graph shows that risk is gradually decreasing for all 3 batches and for lr 0.01 we get least loss compared to others. But what we can see is all 3 batches are almost giving same risk when compared to risk analysis of sgd optimizer.
5. Same thing is with accuracy. We are almost getting same accuracy for all 3 batches.

From above explanation, iris is an small dataset. But still we are able to compare results for different lr,batch sizes and two different optimizers. Learning rate doesn’t affect much to large batch\_sizes.