Hackathon 2

Intro to Deep Learning CSCE 879

We have taken a '**MNIST** dataset that is already available with 'tensorflow_datasets' for Hackathon 2. We have defined several different **layers, activation functions, optimizers, epochs, splitting the dataset into separate training and validation sets** and compared their results to check on how well a model is being trained based on the usage of these different hyperparameters. The parameters like weights, logits that are being trained on training dataset and are then used to check how well the model prepared generalizes on the validation dataset. The term generalizes means on how well a trained model performs on unseen dataset. It also checks if model was overfitting on the training dataset or was being trained in real. All these hyperparameters are used for the **calculation of weights, outputs** from different layers, loss and the **process of backpropagation** by which weights are updated in a model **during training**. So, use of these **hyperparameters directly affect the rate at which model improves**. Detailed description of results obtained from these different factors are explained in next part.

I performed the experiment **12** times with **different hyperparameter values**. I have used different **activation functions** *(relu, sigmoid, tanh),* **optimizers** *(adam, adagrad, RMSprop, SGD),* **epochs** (*model trained for 5 and 10 epochs*), Layers -which in turn decide the number of parameters that would be used for training of the model.

1. **Parameters (Layers)** - We see from our results that whenever we train a model having **a greater number of parameters** it **generalizes well** on validation dataset as val_accuracy is more when we have used more layers or a greater number of parameters to train our model. We can see this difference if we compare the results from **(1st, 2 and 7th row)**.
2. **Activation functions** - On comparing different **activation functions**, we see that **relu and sigmoid** activation functions performs well on training dataset and generalizes well on validation dataset for the 'MNIST' dataset as compared to **tanh** activation function in which results are a bit lagging as compared to previous two.
3. **Epochs** -We see that a model **performs well** if it is trained for **a greater number of epochs**, can be compared from **row 1 & 11.**
4. **Optimizers - play an important role in training of the model** and selecting a correct optimizer plays an important role in training of a deep learning model. We see from the results we obtain that **adam optimizer** (generally due to its adaptive characteristic) **performs well** as compared to others to train the models (achieved perfect val_accuracy using this optimizer in two cases). Also, RMSprop and Adagrad optimizer perform well in training of the model, and we were able to get good results after using these optimizers. **SGD optimizer does not perform well** as the accuracy scores for both training and validation dataset are quite low as compared to when other optimizers were used.

Sr. No.	Layers	Activation function	Optimizer	No. of parameters	Epoch	Train accuracy
					Train, Val	Val Accuracy
1	(200,10)	relu	adam	159,010	5,1	0.921
						0.937
2	(50,10)	relu	adam	39,760	5,1	0.865
						0.9375
3	(200,10)	tanh	adagrad	159,010	5,1	0.81
						0.875
4	(50,10)	tanh	adagrad	39,760	5,1	0.64
						0.75
5	(200,10)	sigmoid	SGD	159,010	5,1	0.36
						0.5
6	(50,10)	sigmoid	SGD	39,760	5,1	0.4
						0.56
7	(200,50,10)	relu	adam	167,560	5,1	0.88
						1.0
8	(50,25,10)	sigmoid	adam	40,785	5,1	0.86
						0.875
9	(200,50,10)	relu	RMSprop	167,560	5,1	0.8
						0.875
10	(50,25,10)	sigmoid	RMSprop	40,785	5,1	0.886
						0.9375
11	(200,10)	relu	adam	159,060	10,1	0.93
						1.0
12	(200,10)	sigmoid	adam	39,760	10,1	0.916
						0.9375

Puranjit Singh

94849312