Model Optimization

29.06.2021

Overfitting or underfitting:

1. model is ***underfitting*** the training data when the model performs poorly on the training data. This is because the model is unable to capture the relationship between the input examples (often called X) and the target values (often called Y).
2. model is ***overfitting*** your training data when you see that the model performs well on the training data but does not perform well on the evaluation data.This is because the model is memorizing the data it has seen and is unable to generalize to unseen examples.
3. Poor performance on the training data could be because the model is too simple (the input features are not expressive enough) to describe the target well. Performance can be improved by increasing model flexibility. To increase model flexibility, try the following:
    a. Add new domain-specific features and more feature Cartesian products, and change the types of feature processing used (e.g., increasing n-grams size).
    b. Decrease the amount of regularization used.
4. If a model is overfitting the training data, it makes sense to take actions that reduce model flexibility. To reduce model flexibility, try the following:
    a. Feature selection: consider using fewer feature combinations, decrease n-grams size, and decrease the number of numeric attribute bins.
    b. Increase the amount of regularization used.
5. Accuracy on training and test data could be poor because the learning algorithm did not have enough data to learn from. You could improve performance by doing the following:
    a. Increase the amount of training data examples.
    b. Increase the number of passes on the existing training data.

Dropout value:

1. A Simple Way to Prevent Neural Networks from Overfitting, 2014. Dropout simulates a sparse activation from a given layer, which interestingly, in turn,

encourages the network to actually learn a sparse representation as a side-effect.
2. The recommended values for the dropout rate are 0.1 for the input layer and between 0.5 and 0.8 for internal layers. Using dropout requires some adjustments to the hyperparameters. The more significant changes are: Increase the network size: dropout removes units during training, reducing the capacity of the network.
3. The main advantage of this method is that it prevents all neurons in a layer from synchronously optimizing their weights. This adaptation, made in random groups, prevents all the neurons from converging to the same goal, thus decorrelating the weights

Batch size and epochs:

1. The batch size is a number of samples processed before the model is updated.
2. The number of epochs is the number of complete passes through the training dataset.
3. They are both integer values and they are both hyperparameters for the learning algorithm, e.g. parameters for the learning process.
4. According to validation loss the if model is improving then the more epochs are better.
5. Less Number of batch size means more accuracy and it totally depends upon the size of the dataset.
6. Monitor the training during the epoch and validation loss after every epoch. The lowest training loss is the best fit model.
7. Validation set only once training epoch is completed

Learning rate scheduler and early stopping:

1. For progress in training, learning rate schedular or adaptive learning rate scheduler is used and it helps to improve model performance.
2. Learning rate schedules seek to adjust the learning rate during training by reducing learning rate according to predefined schedules. Common learning rate schedules are time decay, step decay and exponential decay.
3. The challenge of using learning rate schedules is that their hyperparameters have to be defined in advance and they depend heavily on the type of model and

problem. Another problem is that the same learning rate is applied to all parameter updates. If we have sparse data, we may want to update the parameters to different extent instead.

4. adaptive learning rate methods demonstrate better performance than learning rate schedules, and they require much less effort in hyperparameter settings.

While training very large and deep neural networks, the model might overfit very easily. This becomes a larger issue when the dataset is small and simple. We can easily know this when while training, the validation loss, and training loss gradually start to diverge. This means that the model is starting to overfit. Also, using a single and high-value learning rate might cause the model to miss the local optima altogether during the last phases of training.

In **early stopping**, when we see that the training and validation loss plots are starting to diverge, then we just terminate the training. This is usually done in these two cases:

- We are very sure that the model is starting to overfit.
- We are also sure that using a learning rate scheduler to reduce learning rate and more training will not help the model.

Trying to train a large neural network while using a single static learning rate is really difficult. This is where the learning rate scheduler helps. Using a learning rate scheduler, we can gradually decrease the learning rate value dynamically while training. There are many ways to do this. But the most commonly used method is when the validation loss does not improve for a few epochs.

Let’s say that we observe that the validation loss has not decreased for 5 consecutive epochs. Then there is a very high chance that the model is starting to overfit. In that case, we can start to decrease the learning rate, say, by a factor of 0.5.

Early stopping is another mechanism where we can prevent the neural network from overfitting on the data while training.

Generalization issue:

Generalization refers to a model's ability to adapt properly to new, previously unseen data, drawn from the same distribution as the one used to create the model.
Develop intuition about overfitting. Determine whether a model is good or not.