

small filters over it, which are like tiny "feature detectors" that pick
es, textures', ↓

AI GPT can make mistakes. Check important info.

ENG
IN

09:28 PM
09-11-2024

① Explain Early Stopping, batch normalization & data augmentation (10).

See Early Stopping -

Form of regularization → prevent overfitting in ML & DL. Stops train process before model starts to overfit.

Monitor model's performance on a validation set during training process & stop training when performance starts to degrade, which is an indicator that model is beginning to overfit training data.

Working -

① Validation set:

A portion of training data set aside as validation set. This set used to evaluate model's performance at each epoch (iteration over the entire dataset).

② Performance monitoring -

Model's performance on validation set is monitored at each epoch. Common metrics for this purpose → accuracy, loss, or any other metric relevant to problem at hand.

③ Early Stopping Criterion -

If performance on validation set starts to degrade (i.e. eg: loss ↑ / accuracy ↓) it's an indicator that model is beginning to overfit to training data. At this pt, early stopping triggered process is halted.

④ Model Selectn -

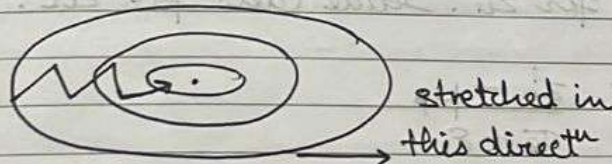
Since training is stopped before overfitting occurs, the model at the pt. of early stopping is best model, as it has not yet learned the noise in training data.

Batch Normalizatiⁿ -

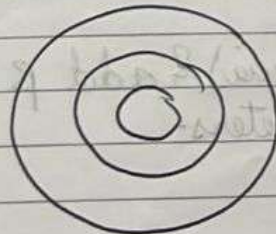
Algorithmic method that makes training of deep NN faster & more stable. Consists of normalizing activatⁿ vectors hidden from hidden layer using mean & variance of current batch. Applied rt. before / rt. after non-linear fn.

Data preprocessing tool used to bring numerical data to common scale without distorting its shape.

I/P should be on same scale due to which mean = 0 & std. dev. = 1 \forall I/P data. If data not normalized, plot \rightarrow



After data normalized, plot \rightarrow



Training faster & stable. So O/P of activatⁿ fn also should be normalized.

Data Augmentation -

By performing data augmentation can prevent NN from learning irrelevant patterns, essentially boosting overall performance.

Where do we augment data in ML pipeline?

Offline augmentation -

Preferred for relatively smaller datasets as you would end up \uparrow size of dataset by factor equal to no. of transforms you perform.

Online augmentation / Augmentation on the fly -

Larger datasets as you can't afford explosive \uparrow in size. Instead you would perform transforms on mini-batches that you would feed to the model.

Models -

Adversarial training / Adversarial ML

GAN

Neural style transfer

Reinforcement learning