ML Assignment

Solomon Lamkhothang

21BDA29

Q. Have you come across Grid Search Cross Validation? Fit any two models covered in previous classes and optimize them using Grid search CV.

Cross validation works by splitting our dataset into random groups, holding one group out as the test, and training the model on the remaining groups. This process is repeated for each group being held as the test group, then the average of the models is used for the resulting model.

Q. What is Stride, Padding & Pooling? Explain with an example.

**Stride** is a component of convolutional neural networks, or neural networks tuned for the compression of images and video data. Stride is a parameter of the neural network's filter that modifies the amount of movement over the image or video. For example, if a neural network's stride is set to 1, the filter will move one pixel, or unit,  at a time. The size of the filter affects the encoded output volume, so stride is often set to a whole integer, rather than a fraction or decimal.

**Padding** is a term relevant to convolutional neural networks as it refers to the number of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, then every pixel value that is added will be of value zero. If, however, the zero padding is set to one, there will be a one-pixel border added to the image with a pixel value of zero.

**Pooling** is nothing other than down sampling of an image. The most common pooling layer filter is of size 2x2, which discards three forth of the activations. Role of pooling layer is to reduce the resolution of the feature map but retaining features of the map required for classification through translational and rotational invariants. In addition to spatial invariance robustness, pooling will reduce the computation cost by a great deal.

Q. What is overfitting? How to overcome overfitting in an ML model?

Overfitting occurs when you achieve a good fit of your model on the training data, while it does not generalize well on new, unseen data. In other words, the model learned patterns specific to the training data, which are irrelevant in other data.

We can try to do something about the overfitting. There are different options to do that:

* **Reduce the network’s capacity** by removing layers or reducing the number of elements in the hidden layers
* Apply **regularization**, which comes down to adding a cost to the loss function for large weights
* Use **Dropout layers**, which will randomly remove certain features by setting them to zero