ML ASSIGNMENT

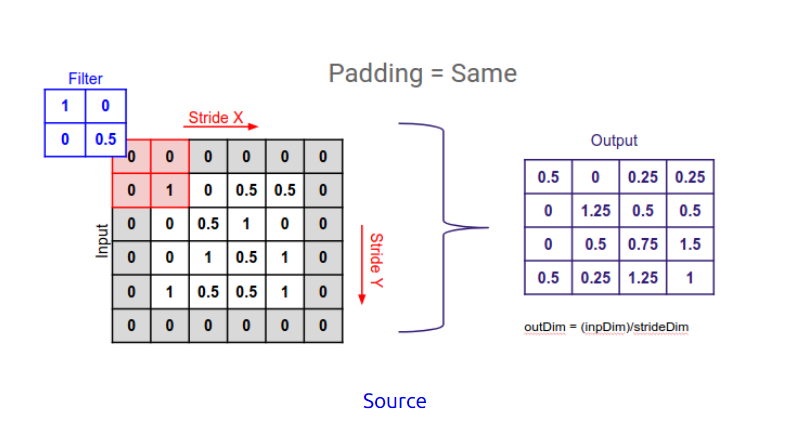
LAB 07

PRAKRUTHI B GOPAL

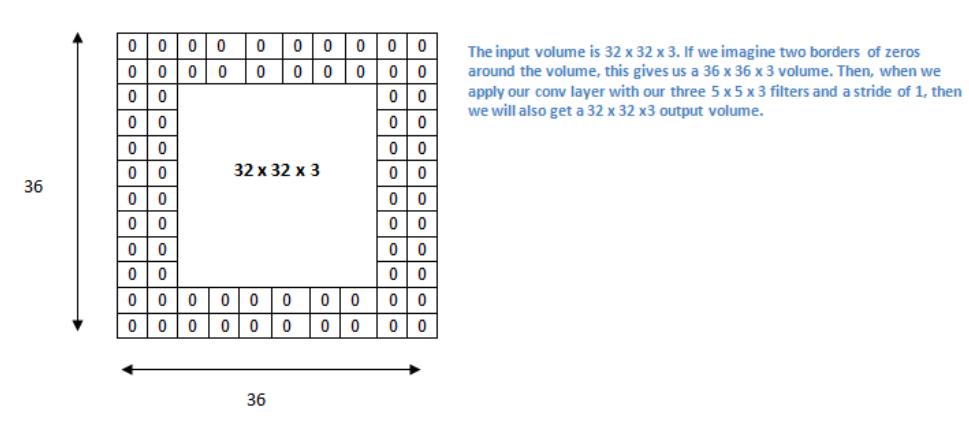
21BDA25

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

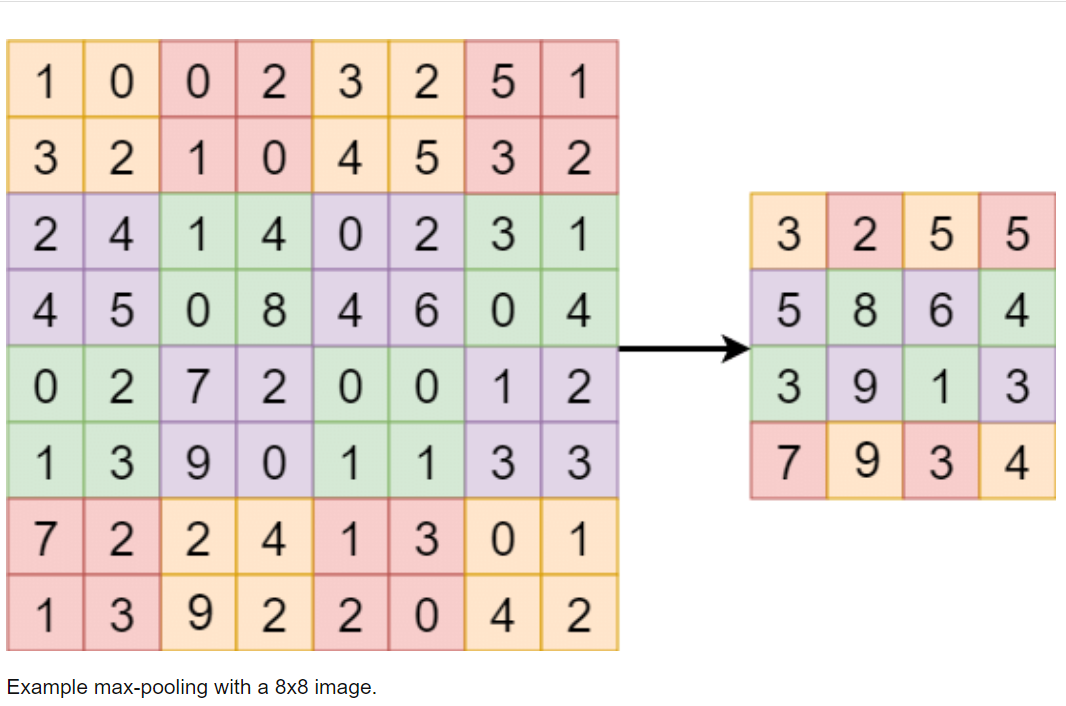
Padding is a term relevant to [convolutional neural networks](https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network) as it refers to the amount of pixels added to an image when it is being processed by the kernel of a CNN. For example, if the padding in a CNN is set to zero, 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.



Stride is a component of [convolutional neural networks](https://deepai.org/machine-learning-glossary-and-terms/convolutional-neural-network), or [neural networks](https://deepai.org/machine-learning-glossary-and-terms/neural-network) 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,  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.



The pooling operation involves sliding a two-dimensional filter over each channel of feature map and summarising the features lying within the region covered by the filter.



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

It is a common pitfall in [deep learning](https://www.v7labs.com/blog/deep-learning-guide) algorithms in which a model tries to fit the[training data](https://www.v7labs.com/blog/quality-training-data-for-machine-learning-guide)entirely and ends up memorizing the data patterns and the noise and random fluctuations.

These models fail to generalize and perform well in the case of unseen data scenarios, defeating the model's purpose.

**To avoid overfitting:**

**Train with more data**

With the increase in the training data, the crucial features to be extracted become prominent. The model can recognize the relationship between the input attributes and the output variable. The only assumption in this method is that the data to be fed into the model should be clean; otherwise, it would worsen the problem of overfitting.

**Feature selection**

Every model has several parameters or features depending upon the number of layers, number of neurons, etc.  The model can detect many redundant features or features determinable from other features leading to unnecessary complexity. We very well know that the more complex the model, the higher the chances of the model to overfit.

**Cross-validation**

Cross-validation is a robust measure to prevent overfitting. The complete dataset is split into parts. In standard K-fold cross-validation, we need to partition the data into k folds. Then, we iteratively train the algorithm on k-1 folds while using the remaining holdout fold as the test set. This method allows us to tune the hyperparameters of the neural network or machine learning model and test it using completely unseen data.

**Regularization**

If overfitting occurs when a model is too complex, reducing the number of features makes sense. Regularization methods like Lasso, L1 can be beneficial if we do not know which features to remove from our model. Regularization applies a "penalty" to the input parameters with the larger coefficients, which subsequently limits the model's variance.

**Ensembling**

It is a machine learning technique that combines several base models to produce one optimal predictive model. In Ensemble learning, the predictions are aggregated to identify the most popular result. Well-known ensemble methods include bagging and boosting, which prevents overfitting as an ensemble model is made from the aggregation of multiple models.

**Early stopping**

This method aims to pause the model's training before memorizing noise and random fluctuations from the data. There can be a risk that the model stops training too soon, leading to underfitting. One has to come to an optimum time/iterations the model should train.