Unit 6 - Assignment 3

In [1]:
```
from IPython.display import Image
```

1. What is over fitting? Explain with respect to CNN.

Ans:- Overfitting happens when your model fits too well to the training set. It then becomes difficult for the model to generalize to new examples that were not in the training set. For example, your model recognizes specific images in your training set instead of general patterns. Your training accuracy will be higher than the accuracy on the validation/test set.

Steps for reducing overfitting:

- Add more data
- Use data augmentation
- Use architectures that generalize well
- Add regularization (mostly dropout, L1/L2 regularization are also possible)
- Reduce architecture complexity.

The first step is of course to collect more data. However, in most cases you will not be able to. Let's assume you have collected all the data. The next step is data augmentation: something that is always recommended to use. Data augmentation includes things like randomly rotating the image, zooming in, adding a color filter etc. Data augmentation only happens to the training set and not on the validation/test set. It can be useful to check if you are using too much data augmentation. For example, if you zoom in so much that features of a cat are not visible anymore, than the model is not going to get better from training on these images.

2. Explain the concept of computational load in detail.

Ans:- Computational complexity is a computer science concept that focuses on the amount of computing resources needed for particular kinds of tasks. In computational complexity theory, researchers assess the kinds of resources that will be needed for a given type or class of task in order to classify different kinds of tasks into various levels of complexity.

Although computational complexity is in some ways similar to the analysis of algorithms, it is essentially its own branch of mathematical theory. Some think of this approach as a measurement of how much work it would take to solve a particular problem or to achieve a particular task. Different kinds of analysis use computational complexity research to find which parts of a task may be most difficult for a computing system, or to figure out how to most efficiently complete some project. Although some developers might consider computational complexity to be irrelevant to their work, others have pointed out that successfully changing tasks or algorithms from a higher complexity class to a lower complexity class can make them work much better. Programmers and developers who use computational complexity theory on items like nested loops, logic trees or other kinds of rhythms can build more efficient systems with a better understanding of how to create less resource-hungry processes.

3. Explain dropout architecture of COVNET.

Ans:- . Deep learning neural networks are likely to quickly overfit a training dataset with few examples.

Ensembles of neural networks with different model configurations are known to reduce overfitting, but require the additional computational expense of training and maintaining multiple models.

A single model can be used to simulate having a large number of different network architectures by randomly dropping out nodes during training. This is called dropout and offers a very computationally cheap and remarkably effective regularization method to reduce overfitting and improve generalization error in deep neural networks of all kinds.

Dropout is implemented per-layer in a neural network. It can be used with most type of layers, such as dense fully connected layers, convolutional layers, and recurrent layers such as the long short-term memory network layer. Dropout may be implemented on any or all hidden layers in the network as well as the visible or input layer. It is not used on the output layer.

4. Define pooling layer? What's its importance?

Ans:- Pooling is a concept in deep learning visual object recognition that goes handin-hand with convolution. The idea is that a convolution (or a local neural network feature detector) maps a region of an image to a feature map. For example a 5x5 array of pixels could be mapped to oriented edge features.

Why to use Pooling Layers?

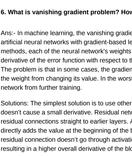
- Pooling layers are used to reduce the dimensions of the feature maps. Thus, it reduces the number of parameters to learn and the amount of computation performed in the network.
- The pooling layer summarises the features present in a region of the feature map generated by a convolution layer. So, further operations are performed on summarized features instead of precisely positioned features generated by the convolution layer. This makes the model more robust to variations in the position of the features in the input image.

5. Explain ReLu function with neat schematic.

Ans:- The Rectified Linear Unit is the most commonly used activation function in deep learning models. The function returns 0 if it receives any negative input, but for any positive value xx it returns that value back. So it can be written as f(x)=max(0,x)f(x)=max(0,x). Graphically it looks like this

In [2]:
```
Image("img3/relu.png")
```

Out[2]:



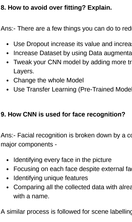
6. What is vanishing gradient problem? How it is resolved.

Ans:- In machine learning, the vanishing gradient problem is encountered when training artificial neural networks with gradient-based learning methods and backpropagation. In such methods, each of the neural network's weights receive an update proportional to the partial derivative of the error function with respect to the current weight in each iteration of training. The problem is that in some cases, the gradient will be vanishingly small, effectively preventing the weight from changing its value. In the worst case, this may completely stop the neural network from further training.

Solutions: The simplest solution is to use other activation functions, such as ReLU, which doesn't cause a small derivative. Residual networks are another solution, as they provide residual connections straight to earlier layers. As seen in Image 1, the residual connection directly adds the value at the beginning of the block, x, to the end of the block (F(x)+x). This residual connection doesn't go through activation functions that "squashes" the derivatives, resulting in a higher overall derivative of the block.

In [3]:
```
Image("img3/residual_block.png")
```

Out[3]:



Finally, batch normalization layers can also resolve the issue. As stated before, the problem arises when a large input space is mapped to a small one, causing the derivatives to disappear. Batch normalization reduces this problem by simply normalizing the input so |x| doesn't reach the outer edges of the sigmoid function.

7. What are the different applications of CNN?. Explain any two in detail.

Ans:-

1. **Decoding Facial Recognition**

Facial recognition is broken down by a convolutional neural network into the following major components -

Identifying every face in the picture

- Focusing on each face despite external factors, such as light, angle, pose, etc.
- Identifying unique features
- Comparing all the collected data with already existing data in the database to match a face with a name.

1. **Analysing Documents**

Convolutional neural networks can also be used for document analysis. This is not just useful for handwriting analysis, but also has a major stake in recognizers. For a machine to be able to scan an individual's writing, and then compare that to the wide database it has, it must execute almost a million commands a minute. It is said with the use of CNNs and newer models and algorithms, the error rate has been brought down to a minimum of 0.4% at a character level, and though it's complete testing is yet to be widely seen.

1. **Historic and Environmental Collections**

CNNs are also used for more complex purposes such as natural history collections. These collections act as key players in documenting major parts of history such as biodiversity, evolution, habitat loss, biological invasion, and climate change.

1. **Understanding Climate**

CNNs can be used to play a major role in the fight against climate change, especially in understanding the reasons why we see such drastic changes and how we could experiment in curbing the effect. It is said that the data in such natural history collections can also provide greater social and scientific insights, but this would require skilled human resources such as researchers who can physically visit these types of repositories. There is a need for more manpower to carry out deeper experiments in this field.

1. **Grey Areas**

Introduction of the grey area into CNNs is posed to provide a much more realistic picture of the real world. Currently, CNNs largely function exactly like a machine, seeing a true and false value for every question.

However, as humans, we understand that the real world plays out in a thousand shades of grey. Allowing the machine to understand and process fuzzier logic will help it understand the grey area us humans live in and strive to work against. This will help CNNs get a more holistic view of what human sees.

1. **Advertising**

CNNs have already brought in a world of difference to advertising with the introduction of programmatic buying and data-driven personalized advertising.

8. How to avoid over fitting? Explain.

Ans:- There are a few things you can do to reduce overfitting.

- Use Dropout increase its value and increase the number of training epochs
- Increase Dataset by using Data augmentation
- Tweak your CNN model by adding more training parameters. Reduce Fully Connected Layers.
- Change the whole Model
- Use Transfer Learning (Pre-Trained Models)

9. How CNN is used for face recognition?

Ans:- Facial recognition is broken down by a convolutional neural network into the following major components -

- Identifying every face in the picture
- Focusing on each face despite external factors, such as light, angle, pose, etc.
- Identifying unique features
- Comparing all the collected data with already existing data in the database to match a face with a name.

A similar process is followed for scene labelling as well.

10. Explain the concept of stride and padding.

Ans:- Reference https://towardsdatascience.com/covolutional-neural-network-cb0883dd6529

Stride denotes how many steps we are moving in each steps in convolution.By default it is one.

In [4]:
```
Image("img3/stride1.gif")
```

Out[4]:
```
<IPython.core.display.Image object>
```

We can observe that the size of output is smaller that input. To maintain the dimension of output as in input , we use padding. Padding is a process of adding zeros to the input matrix symmetrically. In the following example,the extra grey blocks denote the padding. It is used to make the dimension of output same as input.

In [5]:
```
Image("img3/stride_padding.gif")
```

Out[5]:
```
<IPython.core.display.Image object>
```

Let say 'p' is the padding Initially(without padding)

$$(NxN) * (FxF) = (N - F + 1)x(N - F + 1) - - - (1)$$

After applying padding

In [6]:
```
Image("img3/stride_padding.gif")
```

Out[6]:
```
<IPython.core.display.Image object>
```

If we apply filter F x F in (N+2p) x (N+2p) input matrix with padding, then we will get output matrix dimension (N+2p-F+1) x (N+2p-F+1). As we know that after applying padding we will get the same dimension as original input dimension (N x N). Hence we have,

$$(N + 2p - F + 1)x(N + 2p - F + 1)equivalentoNxN$$

$$N + 2p - F + 1 = N - - - (2)$$

$$p = (F - 1)/2 - - - (3)$$

The equation (3) clearly shows that Padding depends on the dimension of filter.

Author Name:- Hemant Ghuge
LinkedIn:- https://www.linkedin.com/in/hemantnthuge/
GitHub:- https://github.com/HemantGorakshGhuge