Convolution Neural Network (CNN)

By Sunil Ghimire- "**Stop talking! Show me your progress!!**"

So far, we have covered all about **Machine Learning,** we have written an AI program based on regression and classification programs, we have learned about MSE, MAE, RMSE, R^2, and Confusion Matrix. In our last tutorial, we learned about **Deep Learning** and concepts based on Deep Learning.

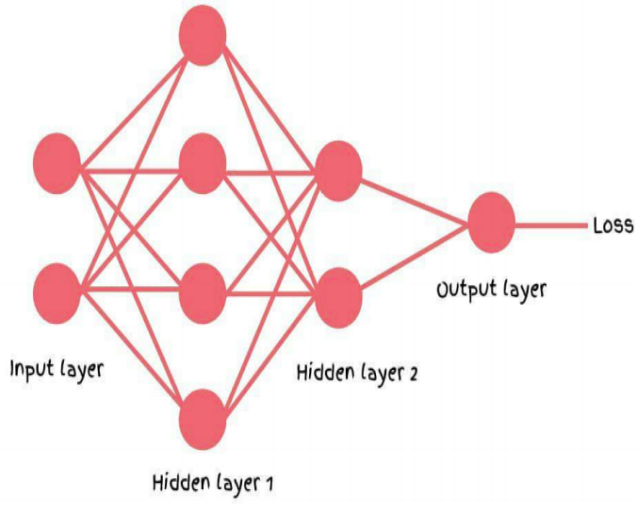
Now, it's time to check your knowledge. Please answer the below questions about Artificial Intelligence and Machine learning.

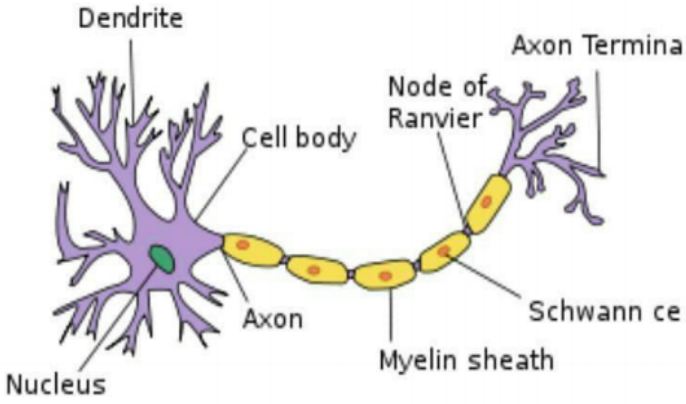
1. What is Machine Learning?
    a. The autonomous acquisition of knowledge through the use of manual programs
    b. The autonomous acquisition of knowledge through the use of computer programs
    c. The selective acquisition of knowledge through the use of manual programs
    d. The selective acquisition of knowledge through the use of computer programs
2. Which search method takes less memory?
    a. Depth-First Search
    b. Breadth-First Search
    c. Linear Search
    d. Optimal Search
3. Which is the best way to go for a Game solving problem?
    a. Random approach
    b. Heuristic approach (Some Knowledge is stored)
    c. An optimal approach
    d. Linear Approach
4. Agents behavior can be best described by _____
    a. Perception Sequence
    b. Agent Function
    c. Sensors and Actuators
    d. Environment in which the agent is performing

5. An agent is composed of _____
    a. Architecture
    b. Agent Function
    c. Perception Sequence
    d. Architecture and Program
6. Let's say, you are using activation function X in hidden layers of the neural network. At a particular neuron for any given input, you get the output as *-0.0001*. Which of the following activation function could X represent?
    a. RELU
    b. Tanh
    c. SIGMOID
    d. None of these
7. What kind of learning algorithm for "Future stock prices or currency exchange rates"?
    a. Recognizing Anomalies
    b. Prediction
    c. Generating Patterns
    d. Recognition Patterns
8. Fraud Detection, Image Classification, Diagnostic, and Customer Retention are applications in …
    a. Unsupervised Learning: Clustering
    b. Supervised Learning: Classification
    c. Reinforcement Learning
    d. Unsupervised Learning: Regression

In this article, we are going to learn about CNN which is an efficient recognition algorithm that is widely used in pattern recognition and image processing. We will learn about the CNN model summary which includes **Filter (KERNEL/ FEATURE DECODER), Stride, Convolution Layer, Pooling Layer and Flatten Layer.**

Now Let's start from the beginning, WHAT IS DEEP LEARNING?





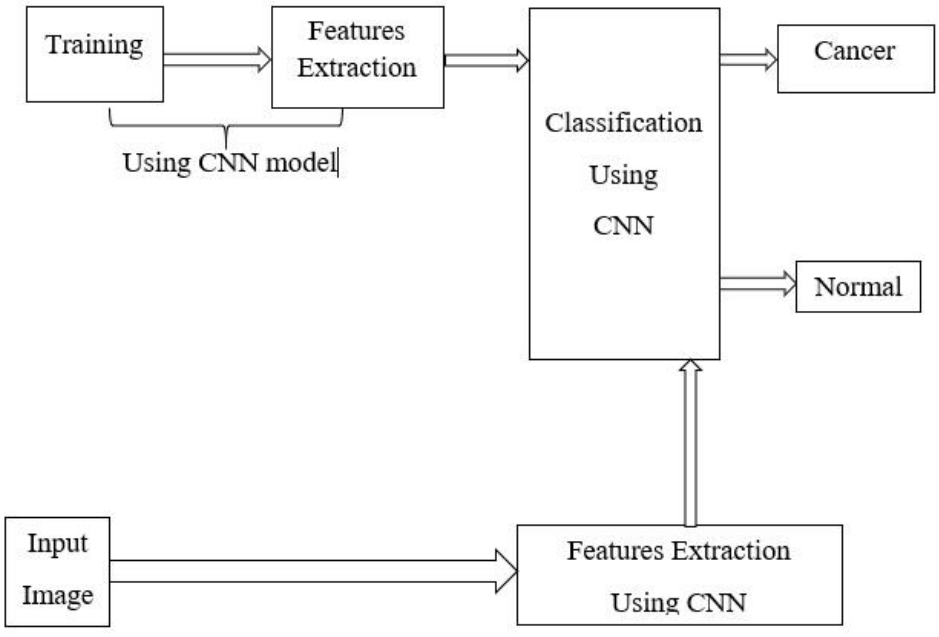
a. As we already know we can achieve Artificial Intelligence through machine learning
b. Deep Learning is a subset of machine learning
c. The Algorithms of Deep Learning try to imitate the working of the human brain in processing data and creating patterns for use in decision making
d. Deep Learning, uses the concept of artificial neural networks to carry out the process of machine learning
e. The artificial neural networks are built like a human brain, with neuron nodes connected together like a web.
f. In a human brain, there are about 200 billion neurons, where each neuron connects with other 100000 neighboring neurons.
g. In our brains, a neuron has a body, dendrites, and an axon. The signal from one neuron travels down the axon and transfers to the dendrites of the next neuron. That connection where the signal passes is called a synapse.
h. Neurons are the main idea behind deep learning algorithms. Let us consider the above structure where we get input and pass that to the hidden layers.
i. The output generated by the hidden layer-1 is passed as an input to the hidden layer-2. And this continuous if we have more hidden layers.
j. The output of the last hidden later is passed to the output layer and loss is calculated.
k. The biggest advantage of Deep Learning is automatic feature extraction.
l. It extracts lower-level features at starting hidden layers and higher features at ending layers.
m. Automatically learning features at multiple levels of abstraction allow a system to learn complex function

mapping the input to the output directly from data, without depending completely on human crafted features.

n. The above example of a deep learning model is the feedforward deep network or multilayer perceptron. (MLP)
o. The Deep in deep learning is a many-layered network
p. The learning here is Hierarchical Feature Learning, where every layer learns from previous layers.

HOW CNN TECHNIQUES IS USED BLOOD CANCER DETECTION PROJECT?

a. have proposed the segmentation method using color-based clustering to obtain the nucleus region and cytoplasm area from stained blood smear images. SVM classifiers are applied with relevant features and gain satisfactory results.
b. have proposed an automatic detection of white blood cells (WBCs) from peripheral blood images and classification of five types of WBCs: eosinophil, basophil, neutrophil, monocyte, and lymphocyte. Eosinophil and basophil from other WBCs are first classified by SVM with a granularity feature. The other three types are then recognized using convolutional neural networks to extract features, and random forest uses these features to classify those WBCs.



CNN OVER OTHER ALGORITHMS

There are a lot of algorithms that people used for image classification before CNN became popular. People used to create features from images and then feed those features into some classification algorithm like SVM. Some of the algorithms also used the pixel level values of images as a feature vector too. To give an example, you could train an SVM with 784 features where each feature is the pixel value for a 28×28 image.

CNN can be thought of as automatic feature extractors from the image. While if we use an algorithm with a pixel vector we lose a lot of spatial interaction between pixels, a CNN effectively uses adjacent pixel information to effectively downsample the image first by convolution and then uses a prediction layer at the end.

This concept was first presented by Yann le cun in 1998 for digit classification where he used a single convolution layer. It was later popularized by Alex net in 2012 which used multiple convolution layers to achieve state of art on the image net. Thus, making them an algorithm of choice for image classification challenges henceforth.

WORKING OF CNN (I)

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
conv2d_1 (Conv2D)            (None, 150, 150, 16)      448
_________________________________________________________________
conv2d_2 (Conv2D)            (None, 150, 150, 16)      2320
_________________________________________________________________
conv2d_3 (Conv2D)            (None, 150, 150, 32)      4640
_________________________________________________________________
conv2d_4 (Conv2D)            (None, 150, 150, 32)      9248
_________________________________________________________________
conv2d_5 (Conv2D)            (None, 150, 150, 64)      18496
_________________________________________________________________
conv2d_6 (Conv2D)            (None, 150, 150, 64)      36928
_________________________________________________________________
max_pooling2d_1 (MaxPooling2 (None, 75, 75, 64)        0
_________________________________________________________________
flatten_1 (Flatten)          (None, 360000)            0
_________________________________________________________________
dense_1 (Dense)              (None, 64)                23040064
_________________________________________________________________
dropout_1 (Dropout)          (None, 64)                0
_________________________________________________________________
dense_2 (Dense)              (None, 2)                 130
=================================================================
Total params: 23,112,274
Trainable params: 23,112,274
Non-trainable params: 0
```

The CNN consists layer of neurons and it is optimized for two-dimensional pattern recognition. CNN has three types of layers namely convolutional layer, pooling layer, and fully connected layer. Our network consists of **11 layers** excluding the input layer. The input layer takes in an RGB color image where each color channel is processed separately.

The first **6 layers** of the convolution network are convolution layers. The First 2 convolution layer applies 1**6 of 3*3 filters** to an image in the layer. The other two-layer applies **32 of 3*3 filters** to an image. And the last **2 layers** of convolution apply **64 of 3*3 filters** to an image. The nonlinear transformation sublayer employs the **ReLU activation function**. The max-pooling sublayer applies a **2*2 filter** to the image which results in reducing the image size to its half. At this point, the convolution network extracts **64 features**, each represented by a **32*32 array** for each color channel.

The eighth layer is the **flatten layer**. The flatten layer transforms a multidimensional array into a one-dimensional array by simply concatenating the entries of the multidimensional array together. The output of this flatten layer is a **one-dimensional array of size 4800**. The ninth layer is the fully connected ANN with the **ReLU activation** function that **maps 4800 input values to the 64 output values**. **The tenth layer is the dropout layer**. **50% of the input values** coming to the layer are dropped to zero to reduce the problem of overfitting. The el**eventh and the final layer is a fully connected** ANN with the sigmoid activation function that **maps 64 input values to 2 class labels**.

First, I train the convolution network using the data in the training set to find appropriate filters' weights in the three convolutional sublayers and the weights that yield minimum error in the two fully connected layers. Next, I evaluate the convolution network using the data in the validation set to obtain validation error and cross-entropy loss. I repeat the training of the convolution network in this same procedure until we complete 10 epochs. Last, I evaluate the performance of the convolution network using data in the test set.

CLASSIFICATION

Neural networks are used in the automatic detection of cancer in blood samples. A neural network is chosen as a classification tool due to its well-known technique as a successful classifier for many real applications. The training and validation processes are among the important steps in developing an accurate process model using CNNs. The dataset for training and validation processes consists of two parts; the training features set which are used to train the CNN model; whilst testing features sets are used to verify the accuracy of the trained using the feed-forward backpropagation network. In the training part, connection weights were always updated until they reached the defined iteration Number or suitable error. Neural networks are used in the automatic detection of cancer in blood samples. The neural network is chosen as a classification tool due to its well-known technique as a successful classifier for many real applications. The training and validation processes are among the important steps in developing an accurate process model using CNNs.

WORKING OF CNN (II)

Layer (type)	Output Shape	Param #
conv2d_1 (Conv2D)	(None, 32, 254, 254)	896
conv2d_2 (Conv2D)	(None, 32, 252, 252)	9248
max_pooling2d_1 (MaxPooling2	(None, 32, 126, 126)	0
dropout_1 (Dropout)	(None, 32, 126, 126)	0
conv2d_3 (Conv2D)	(None, 64, 124, 124)	18496
conv2d_4 (Conv2D)	(None, 64, 122, 122)	36928
max_pooling2d_2 (MaxPooling2	(None, 64, 61, 61)	0
dropout_2 (Dropout)	(None, 64, 61, 61)	0
flatten_1 (Flatten)	(None, 238144)	0
dropout_3 (Dropout)	(None, 238144)	0
dense_1 (Dense)	(None, 4)	952580

Total params: 1,018,148
Trainable params: 1,018,148
Non-trainable params: 0

FIRST, WHAT ARE PARAMETERS?

Parameters in general are weights that are learned during training. They are weight matrices that contribute to the model's predictive power, changed during the back-propagation process. Who governs the change? Well, the training algorithm you choose, particularly the optimization strategy makes them change their values.

Now that you know what "parameters" are, let's dive into calculating the number of parameters in the sample image we saw above. But, I'd want to include that image again here to avoid your scrolling effort and time.

a. Input Layer: The input layer has nothing to learn, at its core, what it does is just provide the input image's shape. So no learnable parameters here. Thus a number of **parameters = 0**.
b. CONV layer: This is where CNN learns, so certainly we'll have weight matrices. To calculate the learnable parameters here, all we have to do is just multiply the by the shape of width m, height n, previous layer's filters d and account for all such filters **k in the current layer**. Don't forget the bias term for each of the filters. A number of parameters in a CONV layer would be : **((m * n * d)+1)* k)**, added 1 because of the bias term for each filter. The same expression can be written as follows: ((**shape of width of the filter * shape of height of the filter * number of filters in the previous layer+1)*number of**

filters). Where the term "**filter**" refers to the number of filters in the current layer.

c. POOL layer: This has got no learnable parameters because all it does is calculate a specific number, no backdrop learning involved! Thus a number of **parameters = 0**.

d. Fully Connected Layer (FC): This certainly has learnable parameters, a matter of fact, in comparison to the other layers, this category of layers has the highest number of parameters, why? because every neuron is connected to every other neuron! So, how to calculate the number of parameters here? You probably know, it is the product of the number of neurons in the current layer c and the number of neurons on the previous layer p and as always, do not forget the bias term. Thus a number of parameters here are: **((current layer neurons c * previous layer neurons p)+1*c)**.

Now let's follow these pointers and calculate the number of parameters, shall we?

a. The first **input layer** has no parameters.

b. Parameters in the second **CONV1 (filter shape =3*3, stride=1) layer is: ((shape of width of filter*shape of height filter*number of filters in the previous layer+1)*number of filters) = (((3*3*3)+1)*32) = 896**.

c. Parameters in the fourth **CONV2 (filter shape =3*3, stride=1) layer is: ((shape of width of filter * shape of height filter * number of filters in the previous layer+1) * number of filters) = (((3*3*32)+1)*32) = 9248**.

d. The third **POOL1 layer** has no parameters.

e. Parameters in the fourth **CONV3 (filter shape =3*3, stride=1) layer is: ((shape of width of filter * shape of height filter * number of filters in the previous layer+1) * number of filters) = (((3*3*32)+1)*64) = 18496**.

f. Parameters in the fourth **CONV4 (filter shape =3*3, stride=1) layer is: ((shape of width of filter * shape of height filter * number of filters in the previous layer+1) * number of filters) = (((3*3*64)+1)*64) = 36928**.

g. The fifth **POOL2** layer has no parameters.

h. The **Softmax layer has ((current layer c*previous layer p)+1*c) parameters = 238144*4 + 1*4= 952580**.

RELATIONSHIP BETWEEN VALIDATION ACCURACY AND TRAINING ACCURACY

Validation Accuracy (val_accuracy) and Accuracy (Acc) are used to measured to evaluate model fitting. When there is a significant difference between these two, then your is overfitting. The validation accuracy (val_acc) should be equal to or slightly less than the training accuracy (acc) to be a better model.

When the training loss starts increasing (i.e. val_acc is decreasing) one should stop. But again if the overall accuracy is showing a significant difference you should think about some changes in the model.

☺ Thanks for your time ☺

What do you think of this "**Convolution Neural Network**"? (Appreciation, Suggestions, and Questions are highly appreciated).