Week – V
Neural Networks a.k.a Deep Learning

ML Bootcamp 2021

GDG Cloud EDMONTON

AlbertaAI

Careera Analytics Lab

Slido: #gdg_ml_bootcamp_w5

Table of contents

Slido: #gdg_ml_bootcamp_w5

Neural Networks(NN)

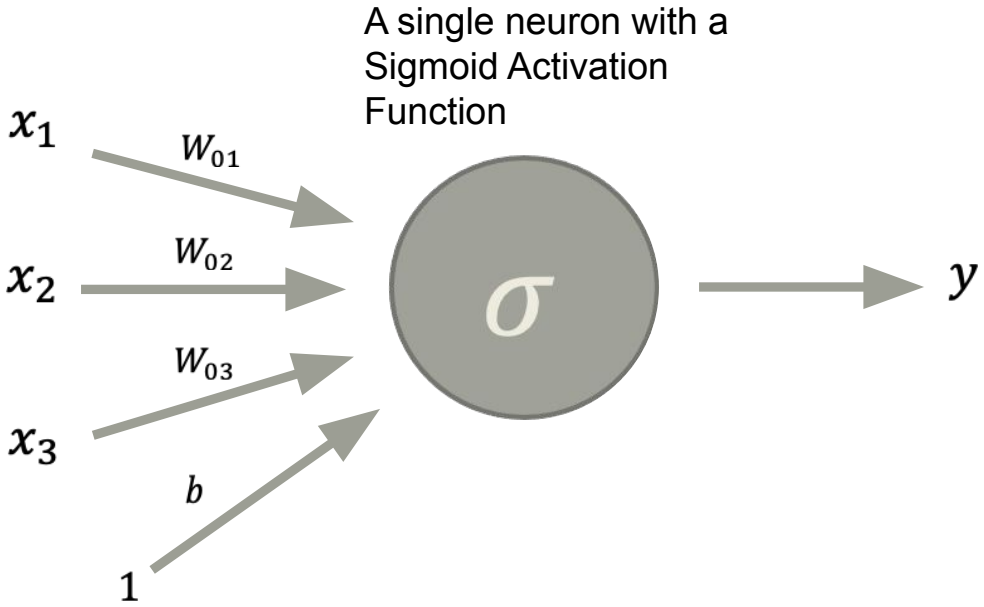
- They are a class of machine learning algorithms that have been inspired from biological system, specifically from the idea of mimicking the human brain, neurons and various connections
- The idea is still the same, learning from the input data to predict something.
- Highly complex computations are performed within a NN and it learns and corrects its **weights** as it learns to recognize the data pc

Some terminology first…

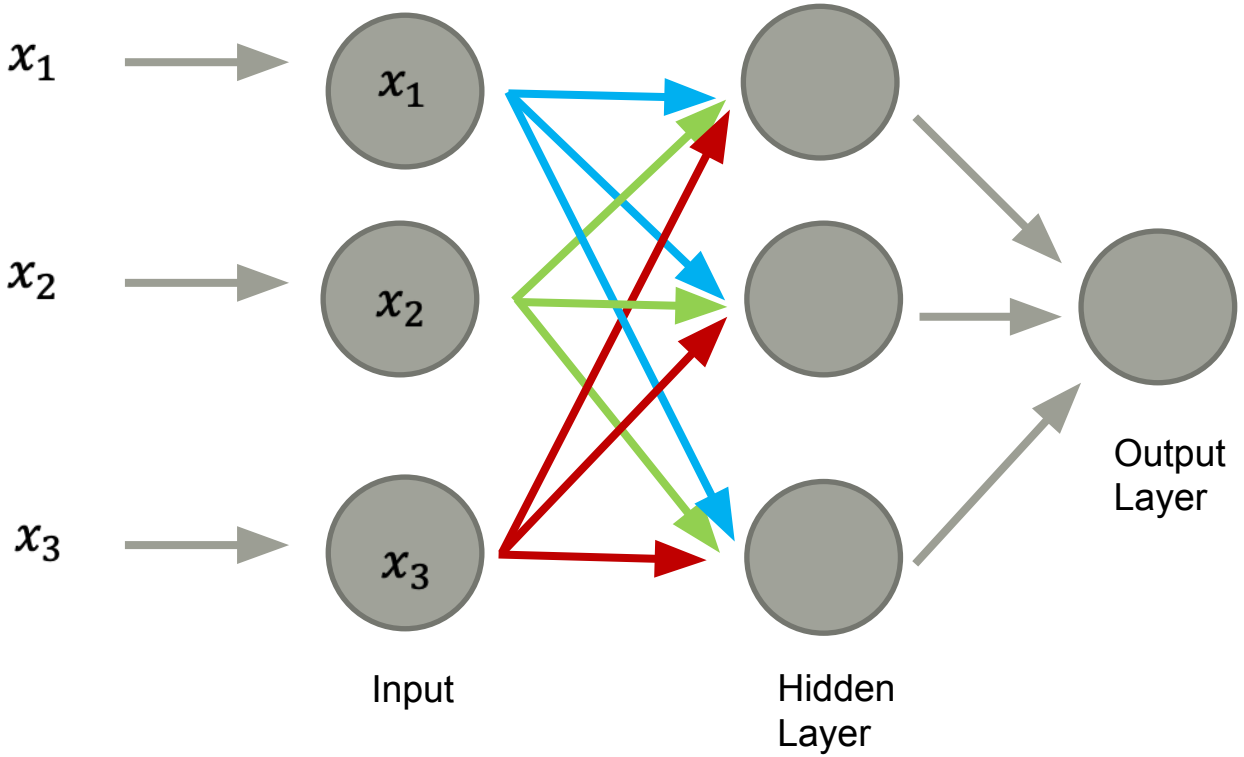
- **Neural Network:** A machine learning algorithm which learns the given input data by training its weights. Inspired from the human brain
- **Neuron:** A single unit of a NN that performs certain computations on the given input
- **Weights:** Like Linear regression the weights are the coefficients that the model learns in order to make its predictions. These weights define the model
- **Layers or Hidden Layers:** A stack of Neurons that performs computations on a given input. This is where all the learning happens
- **Input Layer:** This is the input data points. Not exactly considered a layer as there is no learning happening in this part, but some papers might still consider it a layer.
- **Activation Function**: A non-linear function that is part of the computations in order to provide the non-linearity to the model, otherwise it is just the same as a linear model
- **Epoch:** One epoch is equal to passing the entire dataset through the NN once
- **Batch size:** Training the NN happens in batches where the data is split into pieces and fed batch wise

Why do I need Neural Networks?

Computation inside a single unit/neuron



2-Layer Neural Network



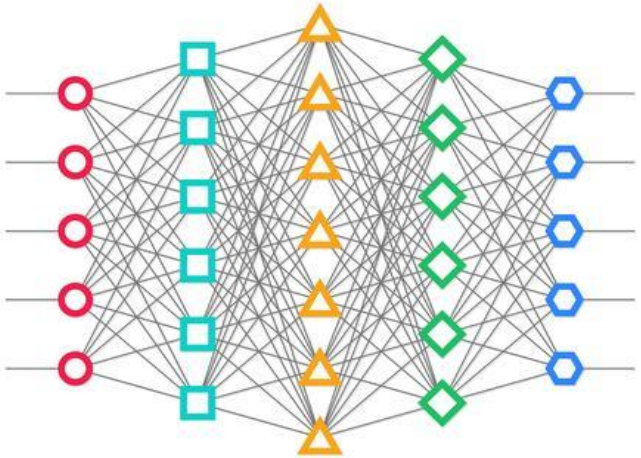
Training - Forward Propagation

Training- Back Propagation

Deep NN/Deep Learning and Vanishing gradients

- A Neural Network with multiple hidden layers is called a Deep Neural Network
- More layers will result in more complex functions being learned by the model
- The issue with having a deep network is that the gradients as they propagate backwards, will become very small and finally the weights will not be updated as expected. **This happens because of the sigmoid activation function**

- <u>Techniques to overcome this:</u>
  - Using Rectified Linear units
  - Batch Normalization
  - Max out units



Bias-Variance Trade-off

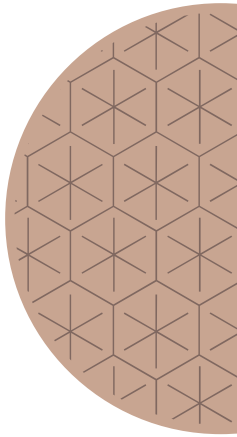
- After training your model it will either have a high/lower bias and high/low variance
- **Bias** is the intrinsic assumptions made by the model and **variance** represents the variability of the predictions of the data points
- Examples of **high bias models/low variance** include Linear/Logistic Regression which assume linearity in the data. **Low bias models/high variance** are decision trees, support vector machines etc.
- Need to locate the ideal point

Overfitting

- Neural Networks tend to overfit as there are a huge number of parameters and it tried to develop a very complex function in order to learn the data
- A common way to overcome overfitting is to use **Regularization** techniques. In case of NN one idea that worked out very well and is still preferred to be used by many is called **Dropout**
- **Idea is to randomly drop certain nodes/neurons during training. This would be, in a sense, ensemble learning as each time a different NN is being trained.**

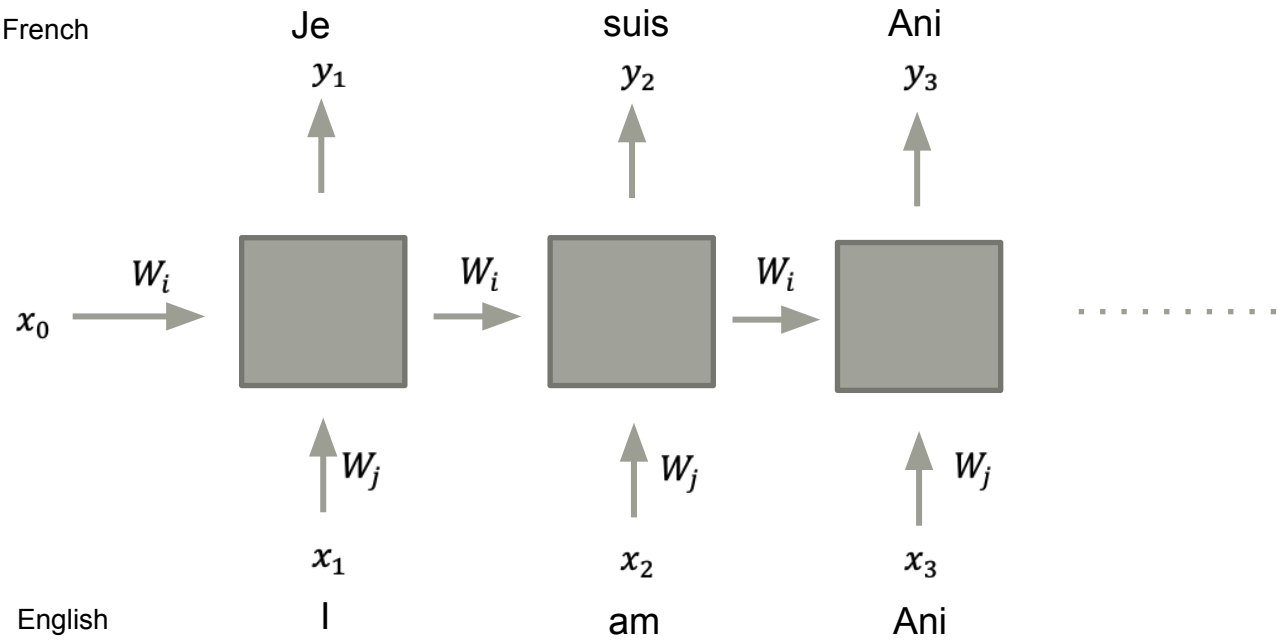
Various types of NN

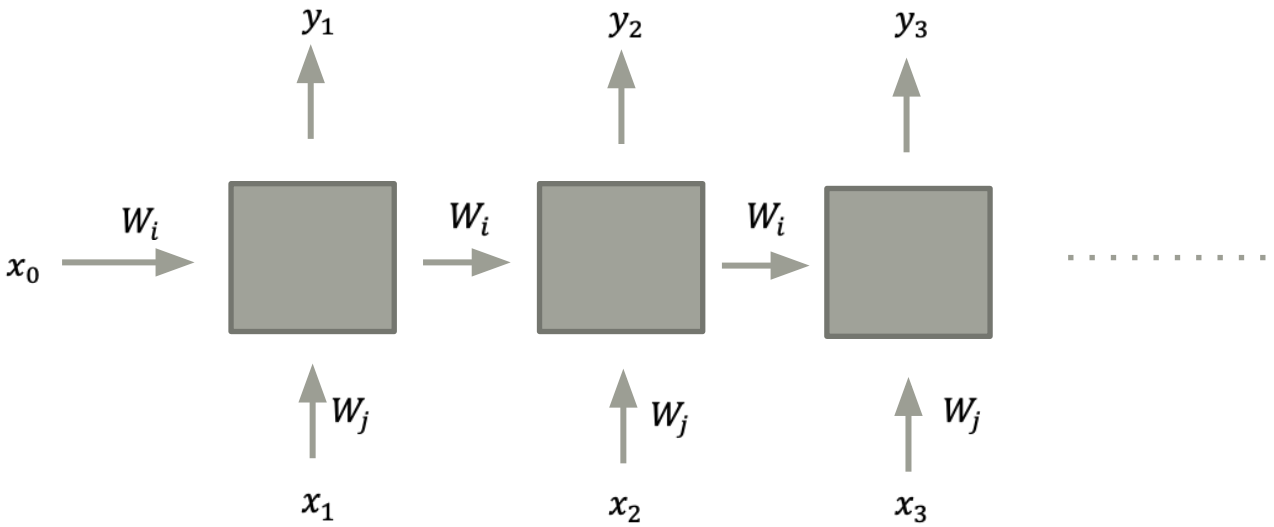
- Artificial Neural Networks
- **Recurrent Neural Networks**
- **Convolutional Neural Networks**
- Residual Neural Networks
- And more if you can think of various architectures….



Recurrent Neural Networks

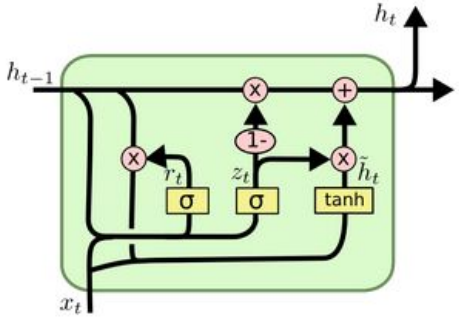
- These NN have been designed to learn sequential data points
- Examples of sequential data: language, timestamped data, audio signal etc.




y_1
y_2
y_3
W_i
W_i
W_i
x_0
W_j
W_j
W_j
x_1
x_2
x_3

Vanishing Gradients and intro to LSTMs

- If the sequence gets too long, then the gradients become small as shown in the previous slides
- A solution for this was proposed and a new and more powerful algorithm was created called Long Short Term Memory networks(LSTMs)



$$z_t = \sigma\left(W_z \cdot [h_{t-1}, x_t]\right)$$

$$r_t = \sigma\left(W_r \cdot [h_{t-1}, x_t]\right)$$

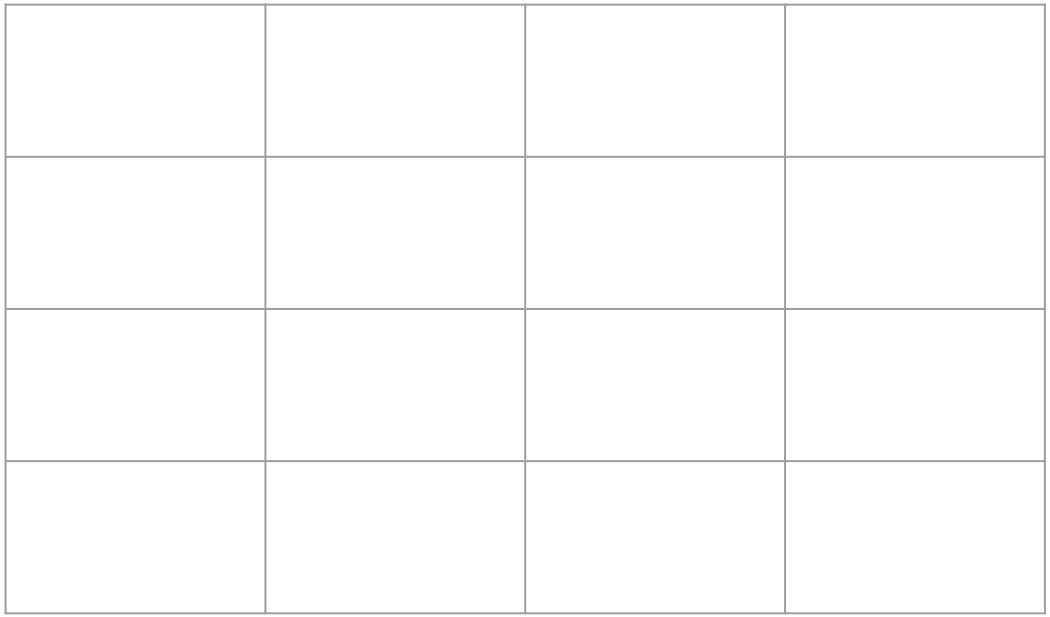
$$\tilde{h}_t = \tanh\left(W \cdot [r_t * h_{t-1}, x_t]\right)$$

$$h_t = (1 - z_t) * h_{t-1} + z_t * \tilde{h}_t$$

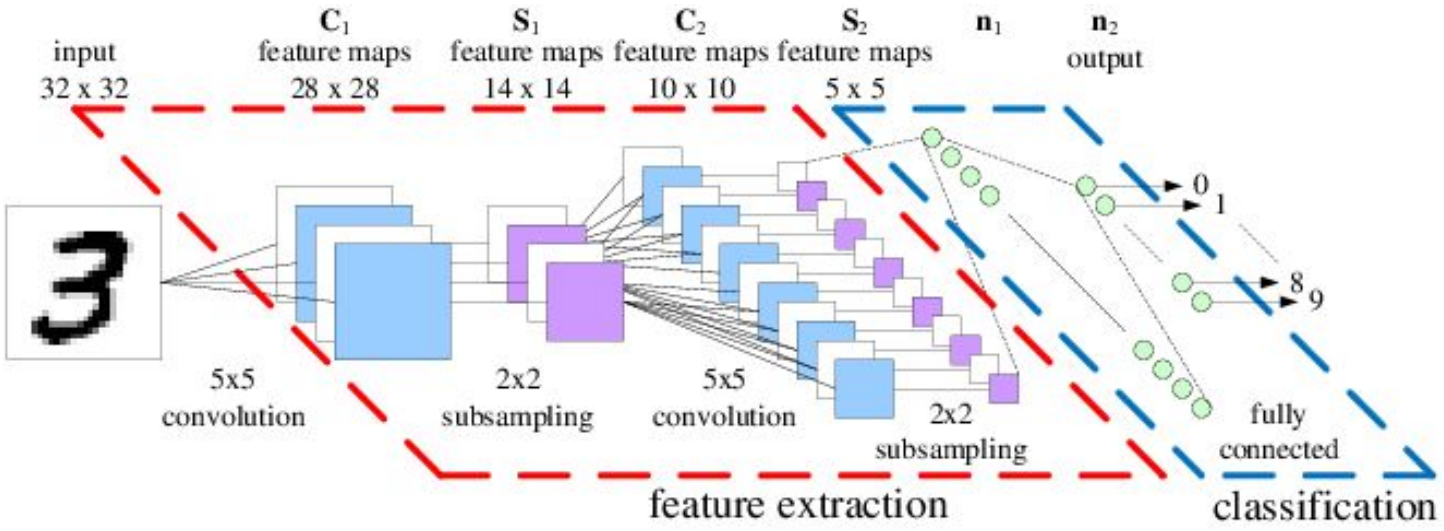
Convolutional Neural Networks

1. They have been used very commonly in Computer Vision/Image processing tasks
   - They act as very good feature extractors.
   - The word “Convolution” refers to the mathematical operation of convolution

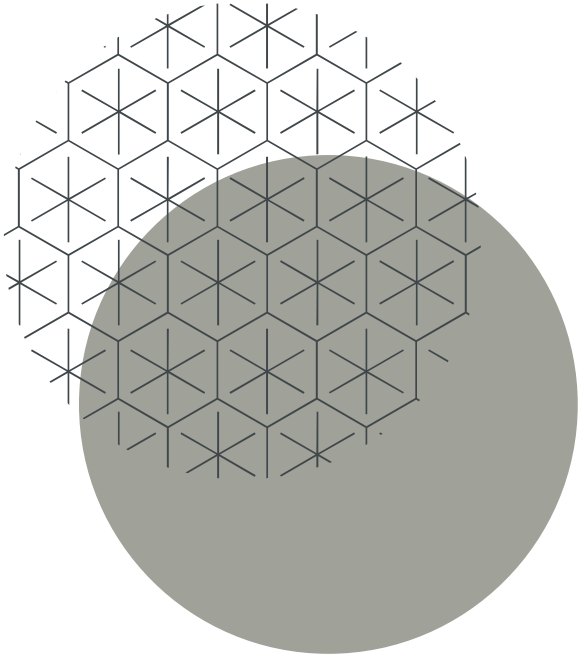
$$y(i) = \sum_{t=-inf}^{inf} x(t)w(i-t)$$

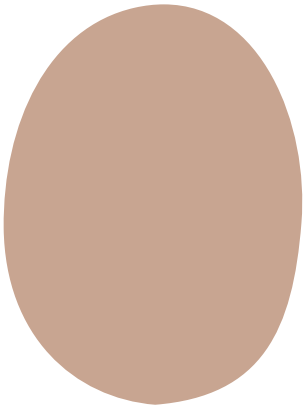


A CNN architecture for digit recognition



Lets start Practicing!





Slido: #gdg_ml_bootcamp_w3

Thanks

Do you have any questions?


slidesgo