CS6462

Probabilistic and Explainable AI

Lesson 19

Bayesian Neural Networks

*****

Bayesian Inference

by Emil Vassev

March 10, 2025

Neural Networks

Definition:*

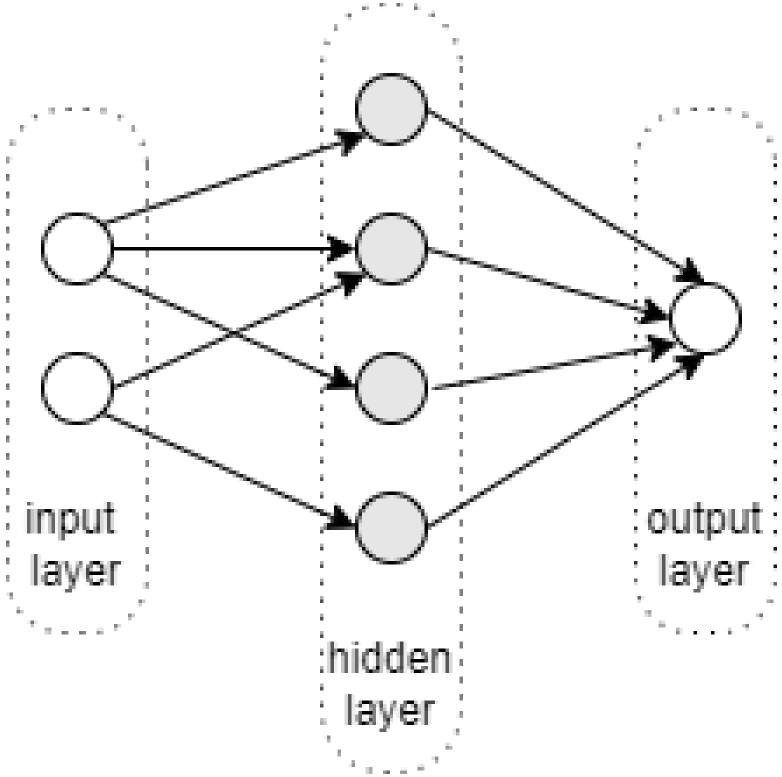
- A series of algorithms that endeavors to recognize underlying relationships in a set of data through a process that mimics the way the human brain operates. In this sense, neural networks refer to systems of neurons, either organic or artificial in nature.

Structure: multiple layers - tend to resemble the connections of neurons and synapses found in brain

- input layer – accepts input signals
- hidden layer(s) – hosts the algorithms
- output layer – delivers the result

Machine Learning with Neural Nets:

- set of algorithms designed to recognize patterns
- computer learns to perform tasks by analyzing training examples
- examples are labeled (added information – labels, used by ML)



**Investopedia*

Artificial Neurons

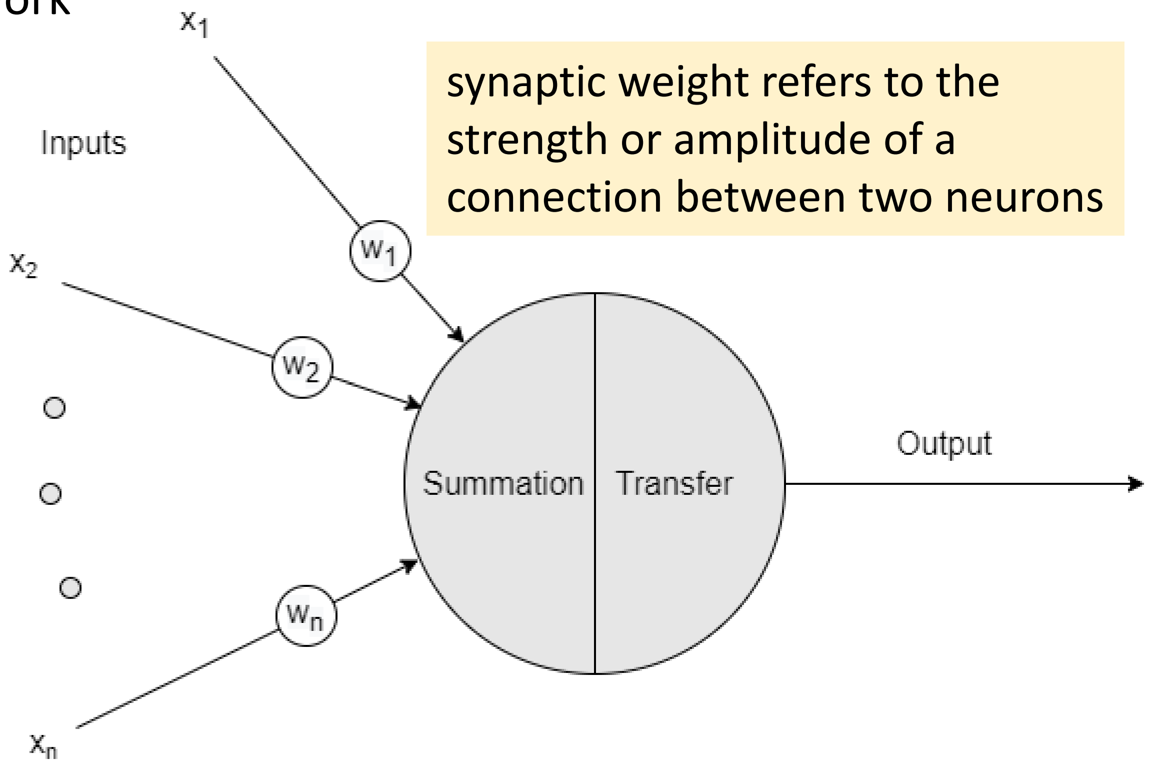
Specifics:

- the fundamental processing element of a neural network
- simulates the natural neuron
- inputs $X = \{x_1, x_2, \dots, x_n\}$
- weight factors $W = \{w_1, w_2, \dots, w_n\}$
- links connecting neurons have a weighting factor
- Summation and Transfer functions

Steps:

- every input is multiplied by its weighting factor
- modified inputs accepted by Summation function:
    - simplest form: sums up the input products
    - *Activation function*: enables Summation function to operate in a time-sensitive way
- the output of Summation function is sent to Transfer function:
    - turns the summation number into a real output via some algorithm
    - commonly supported: Sigmoid, Sine, Hyperbolic Tangent, Threshold, RelU

synaptic weight refers to the strength or amplitude of a connection between two neurons



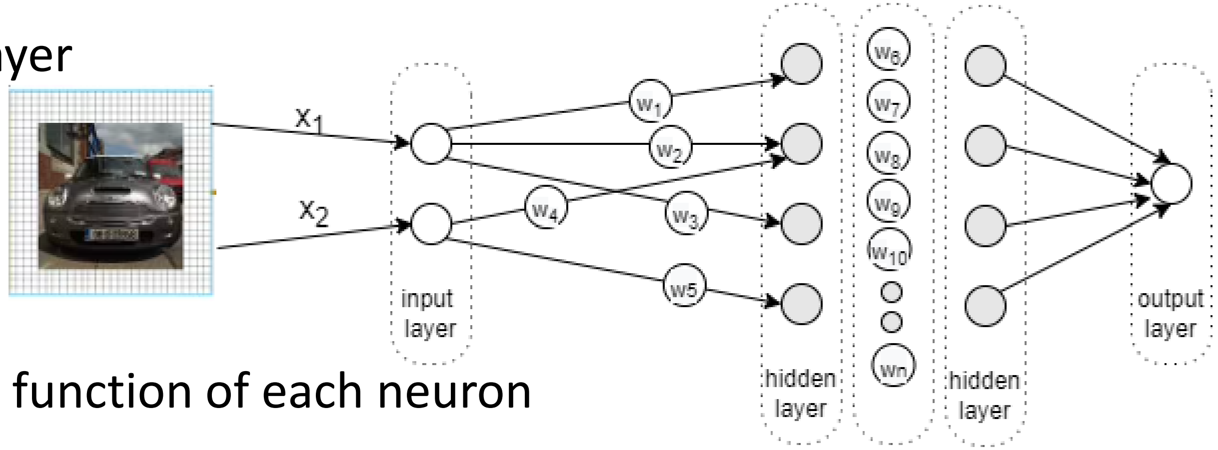
Neural Network Example

Description:

- real-life example of how traffic cameras identify license plates
- pictures are provided in dimension 28 x 28 pixels
- an image is fed as an input to identify the license plate

Neural network:

- each neuron has activation number - represents the grayscale value of the corresponding pixel [0..1]
- arrays of pixels (x_1 and x_2) are fed into the input layer
- inputs are passed to the first hidden layer:
- links are assigned with weights at random
- weights are multiplied with the input signal + bias
- weighted sum of the inputs is fed to the Activation function of each neuron
    - decides which neuron to process the input
    - if the neuron has processed the input, it sends it to neurons of the next layer
- the model predicts the outcome, applying a suitable application function to the output layer

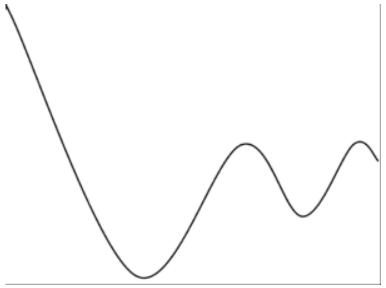


Training Neural Networks

Function fitting:

- the process of training a neural network on a set of inputs (training dataset) to produce associated target outputs
- requires an optimization algorithm:
  - searches through a space of possible values for the neural network to model weights
  - finds a set of weights that results in good performance on the training dataset

Optimization process:



- a search through a landscape for a candidate solution that is sufficiently satisfactory
- a point on the landscape is a specific set of weights for the model:
  - the elevation of that point is an evaluation of the set of weights
  - valleys represent good models with small values of loss
- common conceptualization of optimization problems - landscape is referred to as an error surface
- optimization algorithms:
  - iteratively walk through the landscape
  - update the weights and seek out good or low elevation areas

Bayesian Neural Networks

Specifics:

- able to quantify uncertainty in predictive output
- train the model weights as a distribution rather than searching for an optimal value
- generalize better with less overfitting

Probabilistic neural networks:

- provide outputs in the form of probability distributions
- standard Bayesian neural network outputs a single point estimate
  - if the network is run multiple times with the same inputs, this single point estimate will vary

Bayesian Deep Learning: Bayesian inference + neural networks

$$Posterior = \frac{Likelyhood\ * Prior}{Evidence}$$

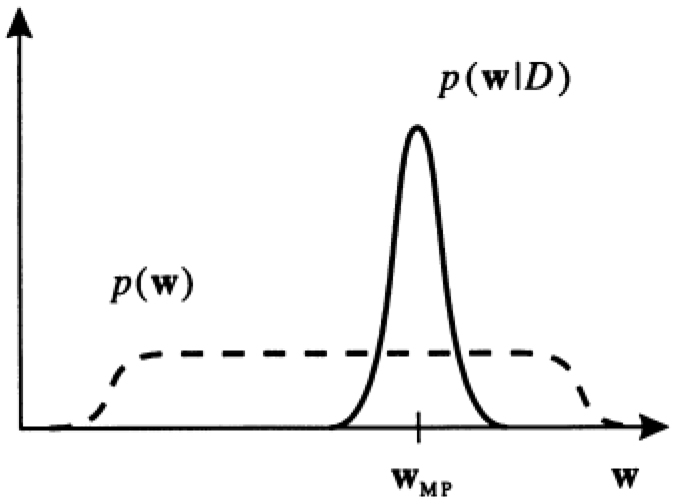
- Posterior Bayesian inference
  - estimates posterior probability of a hypothesis considering new evidence
  - starts with a prior probability distribution (the belief before any evidence)
  - uses the evidence to update this distribution
- Bayesian Neural Networks (BNN) – neural networks that use **Posterior Bayesian Inference** to come up with probability distribution over the network weights, given the training data

Bayesian Learning of Network Weights

Posterior inference over the neural network's weights:

- BNN runs posterior inference to find a posterior distribution over weights

$$p(w|D) = \frac{p(D|w) * p(w)}{p(D)}$$



$w = \{w_1, w_2, \dots, w_n\}$ – weights of the neural network

D – data, i.e., the result produced by the neural network

- Bayesian formalism of learning network weights:
  - changing our belief about the weights from the prior $p(w)$, to the posterior $p(w|D)$ as a consequence of considering the evidence $p(D)$

Bayesian Neural Networks – Pros and Cons

Advantages:*

- BNNs are more robust and able to generalize better than other neural networks
- BNNs can quantify the uncertainty in their predictive output
- BNNs can be used for many practical applications

Disadvantages:*

- BNNs can be more complicated to train than other neural networks
- BNNs require knowledge of the fields of probability and statistics
- BNNs can be slower to converge than other neural networks
- BNNs can require more training data

*Understanding a Bayesian Neural Network: A Tutorial – at NNART

Bayesian Neural Networks with Python

Libraries:

- *Keras* - a high-level neural networks library that provides a simplified interface for building neural networks; uses Tensorflow probability library and the Testsorflow Datasets library
- PyTorch-based:
  - *PyTorch-BayesianCNN* – Bayesian Convolutional Neural Network with variational inference
  - *blitz-bayesian-deep-learning* – an extensible library to create Bayesian Neural Network layers
  - *bayesian-neural-network-pytorch* – a PyTorch implementation of Bayesian Neural Networks
  - *bayesian-torch* - a library for Bayesian neural network layers and uncertainty estimation in Deep Learning extending the core of PyTorch

Summary

Bayesian Neural Networks – *Bayesian Inference*

Neural Networks

Artificial Neurons

Training Neural Networks

Bayesian Neural Networks

Optimization with Bayesian Neural Networks - Bayesian Learning of Network Weights

Pros and Cons of Bayesian Neural Networks

Python Libraries for Implementing Bayesian Neural Networks

Next Lesson:

- Bayesian Neural Networks - Bayesian Linear Regression

Thank You!

Questions?