

Neural Network & Optimization Basics

perceptron

Base neural model

Mainly focus on ~~FFNN~~ ^{Neural} network (other kinds later)

FFN network

[input layer hidden layer(s) output layer]

do linear transformation

takes help of activation fns.

boosts the model's representational power

different kinds → see on w act. are

also help with gradients computation (see later)

activations-introduce non-linearity

motivating theorem for using activations

Last thing to see here:

The learning process in a FF NN → BACKPROPAGATION

In NNs and learning models we try to learn the parameters to make them more "performant" in doing a task.

this is the fine-tuning process which involves training the model.

(we say fine-tune when model is already pretrained, & this has pretrained weights.)

how?

by gradually adjusting the parameters (wts & biases)

how?

by

- 1 - computing small changes (gradients)
- 2 - updating the parameters with the computed grads.
- 3 - repeat until reaching convergence (no more grad updates) (we reached an optimal solⁿ)

gradient descent

also referred to as optimizers

Momentum, Adam, AdamW, others
Batch Normalization, SGD, GD
In FFNN GD is back

GD is Linear

Regression vs Feed Forward NNs.

GD Failure Modes:

Vanishing gradients, Internal Covariate Shifts

Types of GD for FFNNs — the different strategies optimize the GD Process