

Data Science

Cheat Sheet - 10



Machine Learning

Part IV

ML Terminology and Concepts

Features: input data/variables used by the ML model

Feature Engineering: transforming input features to be more useful for the models. e.g. mapping categories to buckets, normalizing between -1 and 1, removing null

Train/Eval/Test: training is data used to optimize the model, evaluation is used to assess the model on new data during training, test is used to provide the final result

Classification/Regression: regression is prediction a number (e.g. housing price), classification is prediction from a set of categories (e.g. predicting red/blue/green)

Linear Regression: predicts an output by multiplying and summing input features with weights and biases

Logistic Regression: similar to linear regression but predicts a probability.

Overfitting: model performs great on the input data but poorly on the test data (combat by dropout, early stopping, or reduce # of nodes or layers)

Machine Learning

Part IV

Bias/Variance: how much output is determined by the features. more variance often can mean overfitting, more bias can mean a bad model

Regularization: variety of approaches to reduce overfitting, including adding the weights to the loss function, randomly dropping layers (dropout)

Ensemble Learning: training multiple models with different parameters to solve the same problem

A/B testing: statistical way of comparing 2+ techniques to determine which technique performs better and also if difference is statistically significant.

Baseline Model: simple model/heuristic used as reference point for comparing how well a model is performing

Bias: prejudice or favoritism towards some things, people, or groups over others that can affect collection/sampling and interpretation of data, the design of a system, and how users interact with a system.

Dynamic Model: model that is trained online in a continuously updating fashion.

Machine Learning

Part IV

Static Model: model that is trained offline

Normalization: process of converting an actual range of values into a standard range of values, typically -1 to +1

Independently and Identically Distributed (i.i.d):

data drawn from a distribution that doesn't change, and where each value drawn doesn't depend on previously drawn values; ideal but rarely found in real life

Hyperparameters: the "knobs" that you tweak during successive runs of training a model

Generalization: refers to a model's ability to make correct predictions on new, previously unseen data as opposed to the data used to train the model

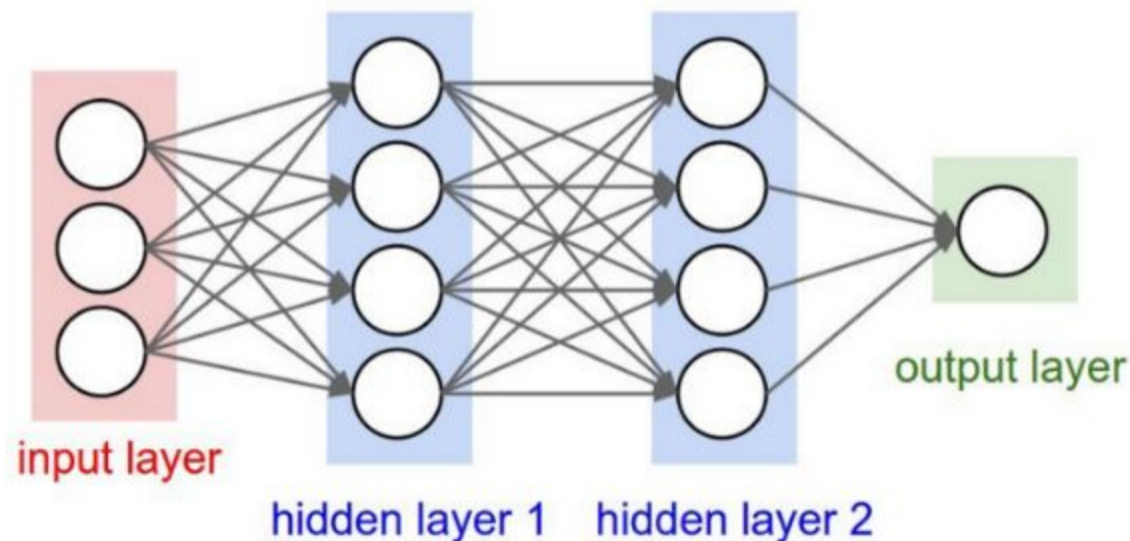
Cross-Entropy: quantifies the difference between two probability distributions

Deep Learning

Part I

What is Deep Learning?

Deep learning is a subset of machine learning. One popular DL technique is based on Neural Networks (NN), which loosely mimic the human brain and the code structures are arranged in layers. Each layer's input is the previous layer's output, which yields progressively higher-level features and defines a hierarchy. A Deep Neural Network is just a NN that has more than 1 hidden layer.



Recall that statistical learning is all about approximating $f(X)$. Neural networks are known as **universal approximators**, meaning no matter how complex a function is, there exists a NN that can (approximately) do the job. We can increase the approximation (or complexity) by adding more hidden layers and neurons.

Deep Learning

Part I

Popular Architectures

There are different kinds of NNs that are suitable for certain problems, which depend on the NN's architecture.

Linear Classifier: takes input features and combines them with weights and biases to predict output value

DNN: deep neural net, contains intermediate layers of nodes that represent "hidden features" and activation functions to represent non-linearity

CNN: convolutional NN, has a combination of convolutional, pooling, dense layers. popular for image classification.

Transfer Learning: use existing trained models as starting points and add additional layers for the specific use case. idea is that highly trained existing models know general features that serve as a good starting point for training a small network on specific examples

Deep Learning

Part I

RNN: recurrent NN, designed for handling a sequence of inputs that have "memory" of the sequence. LSTMs are a fancy version of RNNs, popular for NLP

GAN: general adversarial NN, one model creates fake examples, and another model is served both fake example and real examples and is asked to distinguish

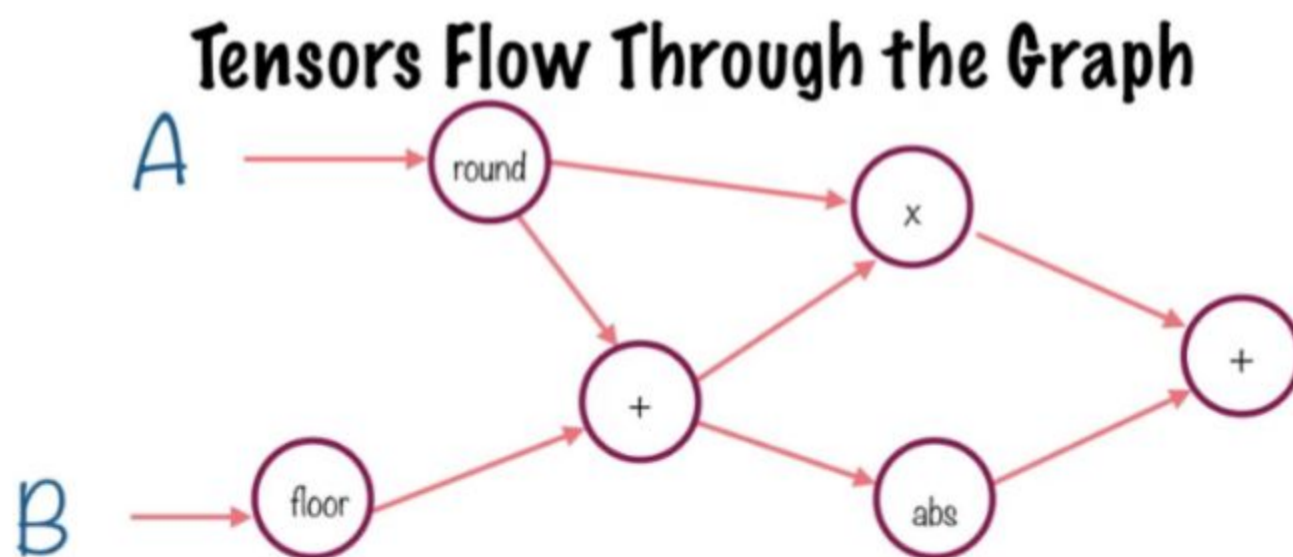
Wide and Deep: combines linear classifiers with deep neural net classifiers, "wide" linear parts represent memorizing specific examples and "deep" parts represent understanding high level features

Deep Learning

Part II

Tensorflow

Tensorflow is an open source software library for numerical computation using data flow graphs. Everything in TF is a graph, where nodes represent operations on data and edges represent the data. Phase 1 of TF is building up a computation graph and phase 2 is executing it. It is also distributed, meaning it can run on either a cluster of machines or just a single machine. TF is extremely popular/suitable for working with Neural Networks, since the way TF sets up the computational graph pretty much resembles a NN.



$$Y = (\text{round}(A) + \text{floor}(B)) * \text{round}(A) + \text{abs}(\text{round}(A) + \text{floor}(B))$$

Deep Learning

Part II

Tensors

In a graph, tensors are the edges and are multidimensional data arrays that flow through the graph. Central unit of data in TF and consists of a set of primitive values shaped into an array of any number of dimensions. A tensor is characterized by its rank (# dimensions in tensor), shape (# of dimensions and size of each dimension), data type (data type of each element in tensor).

Placeholders and Variable

Variables: best way to represent shared, persistent state manipulated by your program. These are the parameters of the ML model are altered/trained during the training process. Training variables.

Placeholders: way to specify inputs into a graph that hold the place for a Tensor that will be fed at runtime. They are assigned once, do not change after. Input nodes

DID YOU LIKE THIS POST

Follow **SKILLSLASH** for more such valuable content

