Deep Learning Quiz 1 Cheat Sheet Cheat Sheet

by Netsuiw via cheatography.com/200763/cs/42404/

Supervised Learning

Mapping from inputs to outputs	Need paired examples (x_i,y_i) to learn from
Examples are	Regression, Text Classification, Image Classification, etc.
Normally in the form of	Input -> Relate family of eqs to input -> Output prediction

Double descent & COD

Double descent is the phenomenon when	the test error increases while the training error is nearing zero and then decreases sharply and back to normal
The tendency of high-dimensional space to overwhelm the number of data points is called the curse of dimensionality	Two randomly sampled data points from normal are at right angles to each other with high likelihood

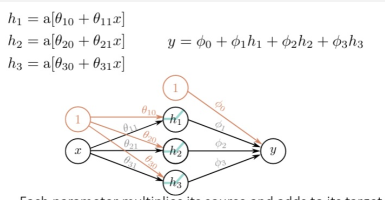
Double descent & COD (cont)

But distance from the origin of random samples is roughly constant and most of the volume of a high dimensional orange is in the peel not in the pulp	Volume of a diameter one hypersphere becomes zero and generate random points uniformly in hypercube, ratio of nearest to farthest becomes close to one

Loss/Cost function and Train/Test

Measurement of	how bad a model performs
Trains on pair of data	Find argmin of this loss function
Test on seperate set of data	Measure the loss there and see its generalizing power
Different loss functions are	Squared Loss, log liklihood, ramp loss, etc

Counting number of parameters



Each parameter multiplies its source and adds to its target

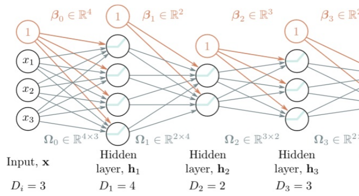
Initialization

If on initalization the variance is small or big	Then it can have floating point errors

Initialization (cont)

So, we want to set variance same in forward and backward pass	He Initalization does this by setting variance 2/D_h where variance of k or k+1 is same at layer k+1 or k

Counting number of parameters



Regularization techniques

Explicit regularization is adding of a regularizing term to the loss function	This is also known as the prior in the probabilistic view. Normally L2 regularization is used where the square weights are added and controlled by a regularization term

Regularization techniques (cont)

Implicit regualrization is the natural tendencies of optimization algorithms and other aspects of the training process	That even without explicitly adding regularization techniques, help improve the generalization performance of a model eg SGD due to batch sizes (cause of randomness)
Early stopping is the process of stopping training early to not overfit the weights since they start small	Ensembling is the collague of different models and is averaged then (by mean or median). Different subset of data resampled is bagging
Dropout is the technique of killing random units. Can eliminate kinks in function that are far from data and don't contribute to training loss	Adding noise can also improve generalization

Regularization techniques (cont)

Can also use baysian inference to provide more information (to priors)	Transfer learning, multi-task learning, self-supervised learning, and data augmentation can be used too to improve generalization

Bias Variance Tradeoff

Variance is the uncertainty in fitted model due to choice of training set	Bias is systematic deviation from the mean of the function we are modeling due to limitations in our model
Noise is inherent uncertainty in the true mapping from input to output	Can reduce variance by adding more datapoints
Can reduce bias by making model more complex	But doing one or the other increases since more complex model = overfitting = more variance

Momentum & Adam

Momentum is the weighted sum of the current gradient and previous gradients	We can think of momentum as a prediction on where we are stepping

Momentum & Adam (cont)

Normalizing the gradients can lead to being stuck if we don't land on the optimal point excatly	Adam prevents that by computing mean and pointwise squared gradients with momentum and moderating near start of the sequence

Schocastic Gradient Descent

Gradient descent might be slow	And not all gradients needed to find optimal point
Compute gradient based on only a subset of points – a mini-batch	Work through dataset sampling without replacement
One pass though the data is called an epoch	This can escape from local minima, but adds noise. Uses all data equally but

Backpropogation

Two passes are done, forward pass, and backwards pass	Forward pass deals with knowing the activations at each layer and how it affects the loss and calculating inbetween values

Backpropogation (cont)

We do not know gradients though so the loss cannot be modified (since units have a dependancy chain at update)	Backward pass calculates the gradients then of the loss function but in reverse
This is very efficient but is memory hungry	Also the problem is trying to split the computation proces apart (i.e. maybe parts exist in different computers)

Gradient Descent

Gradient descent finds optimal point (for convex function)	by step walking towards it, i.e., goes against the gradient calculated
So derivative of loss function wrt to parameters is calculated	And then params are updated by subtracting. A learning rate is applied to speed/slow it down

Deep neural networks

Simply neural networks with more than one hidden layer	Better than simply transposing the output of one shallow network to another (less params and regions)

Deep neural networks (cont)

Basically outputs from hidden units	Go into another hidden layer as inputs
Also obeys the universal approximation theorem	Difference from shallow network is more regions per parameters
The hyperparameters are K for width of network and D_i for number of units of the network at layer i	There exists problems where shallow networks would need way too many units to approximate

Convolutional networks

Parameters only look at local image patches and so share parameters across image	The convolutional operation averages together the inputs
Stride = shift by k positions for each output, Kernel size = weight a different number of inputs for each output, Dilated or atrous convolutions = intersperse kernel values with zeros	Stride decreases output size, Kernel size combines info from larger area, while the last one uses few params while combine info

Convolutional networks (cont)

But we want to lose information: done by apply several convolutions and stack them in channels (feature maps)	Receptive fields is the the region in the input space that a particular CNN's feature is affected by
Benifit of CNN is better inductive bias, forcing the network to process each location similarly, share info, search from small family of input/output maps, etc	Downsampling is the reducing of positions in data (max pooling most common ie take max), while upsampling is the increase

Reinforcement Learning

Create a set of states, actions, and rewards	Goal is to maximize reward by finding correct states
No data involved	Is recieved by the world build and explored
Examples are	Chess, Video games, etc

Reinforcement Learning (cont)

Flaws are that it is	Schocastic, temporial credit assignment, i.e., reward achieved by move or past moves, and Exploration-exploitation trade-off, i.e. when to explore and when to not

Shallow neural network

Use non convex (activation function)	to mold family of functions into dataset
Common activation functions are	ReLU, sigmoid/softmax (as final layer), tanh function (kinda like sigmoid), etc
Pass a set of linear func normally and activation function transforms it (known as hidden layer)	So that a specific weight is activated or not depending on that function

Shallow neural network (cont)

Called shallow since only one hidden layer	Universal approximation theorem states that enough hidden layers can approximate to any continuous function on a compact subset

Maximum likelihood

Points in a database can be from an underlying distribution	The main idea of using likelihood function is to estimate this distribution
Model predicts a conditional probability Pr(y|x)=Pr(y|θ)=Pr(y|f[x,ϕ])	Here the loss function aims to have correct outputs have high probability
So find argmax for ϕ (or argmin if we negative the objective function)	Product can be very small value so log is taken to make it a summation
Softmax is used in the case of multiclass categorization	It converts a vector of K real numbers into a probability distribution of K possible outcomes

Unsupervised Learning

Learning a dataset without any labels	So dataset is orgnaized in input only fashion
Examples are	Clustering, Outlier Finding, Generating examples, fill missing data
There are generative models	like generative adversal networks
Also probabilistic generative models	Who learn the dist over data. Examples are autoencoders, normalizing flows, and diffusion models