1 Some Notes

1.1 On-policy vs. Off-policy

These two concepts are introduced in Chapter 5.

On-policy methods can be a special case of off-policy, where the target policy and behavior policy are the same.

on-policy	off-policy
simpler, considered first	greater variance, slower converge
Agent can pick actions	Agent can't pick actions
most obvious setup :)	learning with exploration playing without exploration
Agent always follows own policy	Learning from expert (expert is imperfect) Learning from sessions (recorded data)

	on-policy	off-policy
value based	Monte Carlo Learning	
	TD(0)	Q-Learning
	SARSA	DQN
	Expected SARSA	Double DQN
	n-Step TD/SARSA	Dueling DQN
	TD(\lamda)	
policy based	REINFORCE	
	REINFORCE with Advantage	
actor-critic	A3C	DDPG
	A2C	TD3
	TRPO	SAC
	PPO	IMPALA

1.2 argmax vs. max

$$\arg\max_x f(x) = f^{-1}\max_x f(x)$$

$$\max f(x) = f(\arg\max f(x))$$

$\arg\max f(x)$ is the function **argument** x at which the maximum of f occurs, and $\max f(x)$ is the **maximum** value of f.

For example, $f(x) = 100-(x-6)^2$, then $\max_x f(x) = 100$ and $\arg\max_x f(x) = 6$.

1.3 Gradient Descent vs. Gradient Ascent

The gradient of a continuous function f = the vector that contains the partial derivatives $\frac{\partial f(p)}{\partial x_i}$ computed at that point p. The gradient is finite and defined if and only if all partial derivatives are also defined and finite. The gradient formula:

$$\nabla f(x) = [\frac{\partial f(p)}{\partial x_1}, \frac{\partial f(p)}{\partial x_2}, \cdots, \frac{\partial f(p)}{\partial x_{|x|}}]^T$$

When using the gradient for optimization, we can either conduct gradient descent or gradient ascent.

Gradient Descent Gradient Descent is an iterative process through which we optimize the parameters of a ML model. It's particularly used in NN, but also in logistic regression and support vector machines (SVM). It is the most typical method for iterative minimization of a cost function. Its major limitation consists of its guaranteed convergence to a local, not necesarily global, minimum.

A hyperparameter (i.e. pre-defined parameter) α (learning rate), allows the fine-tuning of the process of decent. In particular, we may descent to a global

minimum. The gradient is calculated with respect to a vector of parameters for the model, typically the weight w. In NN, the process of applying gradient descent to the weight matrix is called backpropagation of the error.

Backpropagation uses the sign of the gradient to determine whether the weights should increase or decrease. The sign of the gradient allows us to direction of the closet minimum to the cost function. For a given α, we iteratively optimize the vector w by computing

$$w_{n+1} = w_n - \alpha \nabla_w f(w)$$

At step n, the weights of the NN are all modified by the product of the hyperparameter α times the gradient of the cost function, computed with those weights. If the gradient is positive, then we decrease the weights; if the gradient is negative, then we increase the weights.

Gradient Ascent Gradient ascent works in the same manner as gradient descent. The only difference is that gradient ascent maximize functions (instead of minimization).

$$w_{n+1} = w_n + \alpha \nabla_w f(w)$$

Gradient descent works on **convex functions**, while gradient ascent works on **concave functions**.

Summary

- The gradient is the vector containing all partial derivatives of a function in a point
- We can apply gradient descent on a convex function, and gradient ascent on a concave function
- Gradient descent finds the nearest minimum of a function, gradient ascent finds the nearest maximum
- We can use either form of optimization for the same problem if we can flip the objective function.

Gradient vs. Derivative

- A directional derivative = a slope in an arbitrary specified direction.
- A directional derivative is a rate of change of a function in any given direction.
- Gradient = a vector with slope of the function along each of the coordinate axes.
- Gradient indicates the direction of **greatest** change of a funciton of more than one variable.
- Gradient vector can be interpreted as the 'direction and rate of fastest increase'.

Differential vs. Derivative

- Differential is a subfield of calculus that refers to infinitesimal difference in some varying quantity
- Differential represents an equation that contains a function and one or more derivatives of that function
- The function which represents the relationship between the dependent and the independent variables is unknown
- The derivative of a function is the rate of change of the output value with respect to its input value
- Derivative represent the instantaneous change in the dependent variable with respect to its independent variable
- The function which represents the relationship between the variables is known

Loss/Cost/Objective function

- **Loss function** is usually a function defined on a data point, prediction and label, and measures the penalty. For example: square loss in linear regression, hinge loss in SVM, 01 loss in theoretical analysis
- **Cost function** is usually more general. It might be a sum of loss functions

over all training set plus some model complexity penalty. For example: MSE and SVM cost function

- **Objective function** is the most general term for any function that you optimize during training. For example, a probability of generating training set in maximum likelihood approach is a well defined objective function, but it is not a loss function nor cost function

We may say that, a loss function is a part of cost function which is a type of an objective function.

From wikipedia, in mathematical optimization and decision theory, a loss function or cost function (sometimes also called an error function) is a function that maps an event or values of one or more variables onto a real number intuitively representing some "cost" associated with the event. An optimization problem seeks to minimize a loss function. An objective function is either a loss function or its opposite (in specific domains, variously called a reward function, a profit function, a utility function, a fitness function, etc.), in which case it is to be maximized.

Surrogate vs. Approximation I think surrogate and approximation can be used interchangeably.

Surrogate loss function is used when the original loss function is inconvenient for calculation.