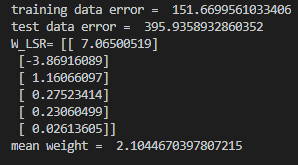
Abgabe Intelligente Adaptive Systeme Laborversuch 2

Malte Hoffmann

Tobias Schlauch

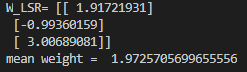
Steve Arnold Pouton Chedjou

V2A1

1. - fun\_true(): berechnet den Zielwert mit den vordefinierten optimalen Gewichten  
   - generateDataSet(): speichert N zufällige Werte zwischen xmin und xmax in X und  
   addiert gaußverteilten noise auf die Zielwertematrix drauf mit der Standardabweichung = sdnoise und Mittelpunkt = 0  
   - getDataError(): Berechnet das Fehlerquadrat von ausgerechneten Zielwert und echtem Zielwert und gibt die Hälfte der Summe der Fehlerquadrate  
   - phi\_polynomial(): berechnet alle Basisfunktionen von x[0], in unserem Fall ist x eindimensional und damit ist x[0] = x, und gibt diese in einem Spaltenvektor zurück  
   - von generateDataSet() werden die Datenmatrix X und die dazugehörige Zielwertmatrix T erstellt  
   - phi\_1(x) = 1, phi\_2(x) = x, phi\_3(x) = x^2, …, phi\_deg(x) = x^(deg)  
   - lmbda hat in diesem Beispiel keine Auswirkung, gibt jedoch üblicherweise die Bestrafung für Gewichte in der Fehlerfunktion an, sodass große Gewichte stärker bestraft werden als kleine und so die Gewichte allgemein klein bleiben  
   - X\_test und T\_test haben verschiedene Werte als X und T, da sie mit verschiedenen Zufallszahlen gefüllt werden. X\_test und T\_test werden zum Testen, X und T werden zum Trainieren verwendet   
   -die grünen Kreuze/Punkte stellen die Klassen der Datenvektoren da, die grüne Kurve die optimale Entscheidungskurve und die rote Kurve unsere aktuelle Entscheidungskurve dar
2. Screenshots einfügen mit Erklärung
3.    
     
   - Unser Netz ist overfitted, da wir sehr wenige Testpunkte haben und einen relativ hohen Polynomgrad. Dadurch passt sich die Entscheidungskurve gut an die Testpunkte an, osziliert aber zwischen diesen Testpunkten etwas, wodurch für neue Datenpunkte hohe Fehlerwerte entstehen.  
     
   Ermittelte Versuchsdaten für lmbda = 0 und N = 10  
     
   - Bei zu niedrigen Polynomgrad kann die Entscheidungskurve nicht optimal an das Problem angepasst werden. Dadurch entstehen hohe Lern- und Testfehler.   
   Bei zu hohen Polynomgrad wird die Entscheidungskurve zu stark an die Lerndatenpunkte angepasst und ist nicht mehr allgemein genug um die Testdatenpunkte richtig zu bestimmen, es kommt zum Overfitting



* Der Lerndatenfehler steigt, da die Entscheidungskurve bei vielen Datenpunkten nicht mehr so genau an jeden Lerndatenpunkt angepasst werden kann. Der Testdatenfehler sinkt, da die Entscheidungskurve nicht mehr so stark zwischen den Lerndatenpunkten osziliert.
* Mit deg = 2, N = 100000 und lmbda = 0 erreichen wir folgende Werte, welche die 10% Abweichung einhalten



1. - Durch die Regualisierung werden große Gewichte bestraft. Dadurch bleiben die Gewichte relativ klein und es kann ein Overfitting verhindert werden. Der Faktor lmbda gibt an wie strak die Regualisierung eingreifen soll, wobei ein großer Wert einem großen Einfluss entspricht.



* Wird Lambda zu klein gewählt kommt es zum Overfitting des Models und der Testdatenfehler ist zu groß. Bei zu großem Lambda wird die Flexibilität des Models zu stark eingeschränkt und die Entscheidungskurve kann nicht mehr genug an das Problem angepasst werden, wodurch die Testdatenfehler wieder steigen, allerdings nur relativ langsam.
* Bei lmbda = 20 haben wir den kleinsten Generalisierungsfehler.