Inoffizielle Ausarbeitung Neural Computation 2 Prof. Dorffner (IMKAI, MU), Stand: SS06

Ausgearbeitet von Murmel (Murmel.vienna@gmx.at)

Unsicherheit in der Neural Computation

1) **Wie ist ein MLP aufgebaut, was kann es, was ist ML?**
   Ein Multilayerperceptron besteht aus zwei oder mehreren Schichten, wobei eine Schicht die Input Units enthält, eine die Output Units (typisch linear) und mindestens eine Schicht Hidden Units (typisch sigmoid) existiert.

   Ein solches MLP kann durch die Überlagerung von gewichteten Sigmoiden beliebige Funktionen annähern. Es erhält seine Komplexität durch das Zusammenspiel vieler einfacher Elemente.

   Die Likelihood entspricht der Wahrscheinlichkeit, dass diese Daten beobachtet werden, wenn die Verteilung richtig ist. Maximum Likelihood versucht also, jenes Theta (also die Verteilungspparameter) zu finden, mit denen die Likelihood am Höchsten ist. Anstatt dies zu maximieren versucht man, die negative Log Likelihood (-log(L)) zu minimieren, da dies dem summierten quadratischen Fehler entspricht.

2) **Wie wird der Fehler bei normalverteiltem Rauschen angenommen, wie setzt er sich zusammen? Was ergibt sich daraus als Dilemma und Vorgehensweise?**
   Fehler = Bias² + Varianz + Rauschen
   Der Fehler setzt sich zusammen aus Bias² (systematischer Abweichung) Varianz (Unterschiede zwischen den Durchläufen) und Rauschen. Das Bias-Varianz-Dilemma besagt: Ein einfaches Modell hat hohen Bias und niedrige Varianz, ein komplexes Modell hat niedrigen Bias und hohe Varianz.

   Dies ist die formale Begründung dafür, warum immer mehrere Durchläufe durchgeführt werden sollten, denn ein komplexes Modell ist nur im Durchschnitt nah am wahren Modell. Die Varianz der Durchläufe liefert eine Schätzung der Varianz, die eine Unsicherheitsschätzung ermöglicht. (Unterschied zur Kreuzvalidierung: Varianz über mittleren Output statt mittleren Fehler)

3) **Wie wird die Varianz minimiert, wie der Bias?**
   Die Varianz kann man minimieren mittels:
   - Komitees (haben niedrige Varianz, aber höheren Bias -> Glättung) – man muss ihre Varianz nochmals schätzen
   - Regularisierung: Man minimiert die effektiven Freiheitsgrade, zB durch einen Strafterm
   - Dichteschätzung der Inputdaten (zB durch GMM), da die Varianz besonders dort hoch ist, wo wenige Daten vorliegen. Dies berücksichtigt jedoch nicht alls Aspekte der Varianz.

   Der Bias ist nicht direkt schätzbar. Man kann ihn minimieren durch:
   - Vergrößerung des Datensatzes (und Wahl eines sehr komplexen Modells)
   - Einbringen von Wissen, Vorannahmen in das Modell (zB Glattheit, Ordnung) -> Bias bekommt positive Bedeutung. Auch Regularisierung wäre so ein Vorwissen (zB Strafterm = Occams Razor, man geht davon aus dass das Modell nicht so komplex ist, oder soft Weight sharing nach Bishop: Gewichte in verschiedenen Gruppen sind gleich)

4) **Wieso kommt es eigentlich auch bei einem komplexen Modell kaum zu overfitting?**
Die Varianz ist viel kleiner als erwartet und es kommt kaum zu Overfitting, weil das Training Bias durch lokale Minima verursacht. Dies ist eine positive, aber unverlässliche Eigenschaft des MLP.

5) **Wie funktioniert heteroskedastische Modellierung? Was sind mixture density networks? Was sind ihre Probleme? Wie wird im heteroskedastischen Modell der Restfehler, wie der Bias bestimmt? Wie geht man in der Praxis vor?**
Bei der heteroskedastischen Modellierung wird die Abhängigkeit der Varianz vom Input ebenfalls durch ein MLP modelliert. (MLP mit Input als Input und als Output Mittelwert und Varianz) Es wird wieder mit Maximum Likelihood trainiert, die Fehlerfunktion wird komplexer.

Mixture Density Networks (MDNs) nähern beliebige Dichtefunktionen des Rauschens durch Gaussche Mixtures an. Sie sind MLPs mit 3k Outputs für k Gausskurven. Wenn das Rauschen nicht gaussverteilt ist (sondern zB nach oben stärker streut) kann das MDN bessere Konfidenzintervalle angeben.
Probleme von MDNs sind:

- Durch die hohe Anzahl an Freiheitsgraden werden viele Daten notwendig
- Die Likelihoodfunktion kann leicht gegen ∞ gehen, man braucht also eine gute Initialisierung
- Auch das MDN hat Bias und Varianz, die geschätzt werden müssen
- Die Parameter (μ, σ, π) sind eigentlich unabhängig voneinander, man muss also eigentlich für jeden Parameter getrennte Netze verwenden
- Mangelnde Identifizierbarkeit (Verteilung kann auf viele Arten modelliert werden)

Den Restfehler erhält man über eine Rückprojektion. Wenn das Modell fürs MDN optimal gewählt wurde, ist der Restfehler gaussverteilt.
Für ein perfektes Modell hat das normalisierte Rauschen Erwartungswert 0. Die Abweichung davon ergibt eine Schätzung des Bias. Aber diese Schätzung hat selbst wieder Bias und Varianz.

In der Praxis wählt man ein komplexes Modell, minimiert dessen Varianz durch ein Komitee, sieht sich das Restrauschen an (ob Gauss mit konstanter Varianz), und verwendet gegebenenfalls ein MDN mit Varianzminimierung. Hierfür schätzt man den Bias (aus Rauschverteilung), was auch für hochdimensionale Probleme möglich ist (sofern Rauschen auf Einzeloutputs unabhängig ist), schätzt die Varianz (durch Kreuzvalidierung) und hat schließlich den Konfidenzintervall Bias + Varianz + Rauschen

6) **Schreiben Sie das Bayessche Theorem auf und erklären Sie die Terme. Wie sieht es mit der Unsicherheit in der Klassifikation aus? Was ist zu Konfidenzintervallen und moderierten Wahrscheinlichkeiten zu sagen? Was sind die Probleme hierbei und wie löst man sie?**

$P(Ci|Xj)=(P(Xj|Ci)*P(Ci))/(P(Xj))$
Hierbei ist die linke die a-posteriori-Wkt, P(Ci) die a-priori-Wkt, P(Xj|Ci) die Likelihood und P(Xj) die Wahrscheinlichkeit für das Auftreten der Beobachtung (die Summe aller möglichen Fälle).

Auch in der Klassifikation existiert das Bias-Varianz-Dilemma. Die Varianz führt zu Unsicherheit in der Entscheidungsgrenze. Hierbei haben nicht die Wahrscheinlichkeiten, sondern die Performanz einen Konfidenzintervall. Die Unsicherheit durch niedrigere Wahrscheinlichkeiten flacht die Kurve ab, wie haben „moderierte Wahrscheinlichkeiten“.
Hier ist die Varianz nicht überall groß, wo es wenige Daten gibt (zB nicht in Randbereichen). Es ist zusätzlich noch eine Dichteschätzung nötig, das MLP ist aufgrund der direkten Anwendung von Bayes eigentlich nicht nötig.
Der Bias ist hier die systematische Abweichung vom Bayes-optimalen Klassifikator. Es ist eine Bias-Varianz-Dekomposition möglich, auch wenn diese schwierig ist. Der Bias selbst ist nicht direkt abschätzbar, aber er kann mit Benchmarks (nearest neighbour) verglichen werden.

Ensemble Methoden

7) Was ist PAC Lernen, wie wird es auf Neuronale Netze angewandt? Wieso funktioniert es besser als herkömmliche Methoden?

Probably Approximate Correct Lernen basiert auf einer Menge X von Instanzen (Beispielen), welche ein Zielkonzept c(X)=0 oder x(X)=1 besitzen. Außerdem gibt es eine Menge H von Hypothesen. Gesucht ist eine Hypothese, sodass h(X)=c(X). Die Hypothesen bilden also ein Komitee, ein Ensemble. Damit dieses etwas bringt, sollten folgende Dinge gegeben sein:

- Jede Einzelhypothese sollte zumindest besser als raten sein (Fehler <0.5)
- Hypothesen sollten unabhängig voneinander sein
- Hypothesen sollten eine große Varianz haben
- Fehler der Einzelhypothesen sollte jeweils ein Anderer sein

Die Vorteile dieser Methode sind:

- Statistisch: man hat Varianz aufgrund der Daten
- Komputational (lokale Minima): Varianz aufgrund des Trainings
- Repräsentational: Hypothesen alleine können Lösung nicht darstellen (Bias)

8) Worauf basieren Bagging, Boosting und AdaBoost? Was sind starke, was schwache Lerner, was die Effekte des Boosting? Wie kann man Perceptrons boosten?

Beim Bagging wählt man für das Komitee durch Bootstrapping (zufällige Auswahl mit Zurücklegen) das Trainingsset aus. Diese Lösung ist nicht besser als Kreuzvalidierung.

Beim Boosting werden häufig falsch klassifizierte Beispiele öfter ins Trainingsset genommen, indem die Beispiele nach Fehlern in vorangegangenen Trainingssets gewichtet werden. Dadurch entsteht ein sequentielles Komitee. AdaBoost ist ein Algorithmus hierzu. Boosting konvertiert das Ensemble von schwachen Lernern (kaum besser als Raten) zu starken Lernern (erreicht minimalen Fehler mit gewünschter Wahrscheinlichkeit), indem es eine Variation des schwachen Lerners erzeugt (erst dann Fehler unkorreliert). Die Gewichtung wirkt hierbei wie eine Und-Verknüpfung.

Man kann Boosting auch als Trainingsalgorithmus für MLPs betrachten, welcher sequentiell einem Perceptron Hidden Units hinzufügt und daraus ein MLP entstehen lässt.

9) Was sind MoEs, wie werden sie realisiert? Was sind die Bezüge zu MDNs?
Die Idee hinter Mixtures of Experts ist, dass mehrere Netze im Wettstreit um den richtigen Output stehen und ein „geting network“ die Outputs gewichtet und damit entscheidet, welcher „expert“ das meiste zu sagen hat.
Nach der Gewichtung eines Gating Netzes erhält man so GMMs. All dies ist sehr verwandt mit dem MDN, nur hier liegt der Fokus auf der Gewichtung der Teilmodelle.

Bayessche Inferenz

10) Was ist das Prinzip Bayesscher Inferenz? Zeigen Sie dies anhand einer Skizze am Beispiel der Gaussverteilung. Was ist Maximum Posterior und was die Rolle des Priors? Welche Inferenzverfahren unterscheidet man?
Bei Bayesscher Inferenz sucht man die Wahrscheinlichkeitsverteilung möglicher Modelle, gegeben die Daten. Je mehr Daten es gibt, desto eingeengter ist sind diese Modelle. Eine große Unsicherheit führt zu einer breiten Verteilung.
-> die Gauss-Kurve legt sich dorthin, wo am Meisten Daten sind. Je breiter, desto unsicherer.

Das Maximum der Posterior Verteilung liegt beim minimalem regularisierten Fehler. Die A-Priori-Annahme entspricht also der Regularisierung. Wenn n->∞ wird der Prior immer unwesentlicher, maximum posterior entspricht dann maximum likelihood.

Die Priors sind hierbei das a-priori Wissen, die Annahmen. Der Prior beeinflusst das Ergebnis, dies ist die Ursache für Kritik, ABER Wissen wird immer verwendet (s. Regularisierung), hier wird es explizit probabilistisch formuliert. In der Praxis besteht die Kunst in der sorgfältigen Auswahl der Priors.

Bei den Inferenzverfahren unterscheidet man:
- Gauss’sche Approximation
- Markov Chain Monte Carlo
- Variational Bayes

11) Was ist Gauss’sche Approximation? Worauf basiert das MCMC Verfahren? Welche Erweiterungen gibt es dafür?
Bei der Gausschen Approximation ist man nur an der Breite der Verteilung interessiert, nicht an deren genauer Form. Man approximiert durch eine Gaussverteilung und das Integral wird analytisch lösbar.

Bei MCMC sampelt man aus dem Posterior eine Verteilung, sodass die Netze und ihre Gewichtsvektoren gemäß des Posterior verteilt sind. Dies entspricht einem Komitee mit Gewichtung.
Die Grundlage ist der Random Walk (Zufallspfad). Man akzeptiert neue Punkte nur, wenn sie höheres Gewicht als die alten haben oder mit Wahrscheinlichkeit p(wnew)/p(wold) wenn sie niedrigeres Gewicht haben. Dies erzeugt Punkte proportional zur Datendichte.
Hierbei müssen die ersten paar Punkte gelöscht werden („burn in“) und für jeden Punkt wird eine regularisierte Likelihood benötigt (-> keine Optimierung). Es ist nur ein Quotient aus Dichten notwendig, also kein Normalisierungsfaktor. Ergebnis ist ein Komitee von Netzen, die proportional zu ihrem Posterior vertreten sind.

Erweiterungen wären:
- Zufallssprung wird aus Proposal Verteilung q gezogen
- Metropolis-Hastings: dieses q wird bei Akzeptanzentscheidungen mitbenutzt
- q kann auch dem Prior entsprechen
- q kann auch Gradienteninformationen benutzen („hybrider Monte Carlo“)

- Simulated Annealing oder Reversive Jump MCMC, damit MCMC nicht bei multimodalen Verteilungen hängen bleibt.

12) Wie löst man das Problem der Hyperparameter? Wie dehnt man den Ansatz auf die ganze Modellklasse aus?

Die Hyperparameter alpha (Priorbreite) und Beta (Rauschvarianz) wurden als fix angenommen, doch man kann sie „Ausintegrieren" (allgemeines Bayesianer-Prinzip). Hierdurch werden neue Annahmen nötig, deren Einfluss jedoch immer geringer wird.

Der Bayes-Ansatz kann auf die ganze Modellklasse ausgedehnt werden, indem mandie Modellevidenz berücksichtigt (hierarchischer Ansatz) – wenn alle p(Hi) gleich wird das Modell mit der höchsten Evidenz gewählt. Der Einfluss aller Parameter w wird hierbei ausintegriert und der ganze Datensatz kann verwendet werden (kein separater Validierungsdatensatz nötig!)

13) Wie wird die Modellkomplexität berücksichtigt? Was sind Occam- und Bayes-Faktor?

Die Bayes-Evidenz bestraft komplexe Modelle. Strafterm hierfür ist der Occam-Faktor (Unterschied im Gewicht des Posteriors / Unterschied im Gewicht des Priors) Er ist klein wenn das Modell a priori viel darstellen kann aber nur wenige Parameter die Daten gut beschreiben. Das beste Modell soll die Daten gut beschreiben aber nicht zu komplex sein.
Der Bayes Faktor $K=p(D|H0)/p(D|H1)$ dient dem Vergleich zweier Modelle (bei $K>1$ Modell H1 nicht besser, $K<1$ H1 nicht signifikant besser, $K<0.1$ H1 signifikant besser und $K<0.01$ hoch signifikant). Er ersetzt Signifikanztests bei der Kreuzvalidierung.

14) Wieso gibt es zwangsläufig mehrere Maxima und wie löst man dies? Was sind Nachteile von MCMC und wie heisst und funktioniert ein besseres Verfahren?

Gewichte bei Neuronalen Netzen haben keine eindeutige Bedeutung, es gibt daher viele äquivalente Netze. Man kann dies lösen indem man die Gewichte permutiert oder eine Reihenfolge erzwingt (zB nach Gewicht geordnet)

MCMC Methoden haben die Nachteile, dass sie sehr rechenintensiv sind und die Konvergenz nicht garantiert ist (beliebig langer „burn in").

Ein besseres Verfahren ist Variational Bayes – hier wählt man eine beliebige Verteilung q(w) so, dass das Integral, in welcher diese zur Berechnung vorkommt, berechenbar oder zumindest maximierbar wird. Die Differenz der unteren Schranke dieses Verfahrens zur tatsächlichen Modellevidenz wird auch die Kulback-Leibler Distanz genannt. Diese wird minimiert. Das Verfahren ist schneller als MCMC und liefert oft sehr brauchbare Lösungen (zB in der Modellselektion).

Kernel Methoden

15) Wie funktioniert die Perceptron Learning Rule, was sind seine Vor/Nachteile und wie kommt man davon zum neuronalen Netz? Was ist ein Kernel und was ist der Trick dahinter?

Die Perceptron Learning Rule basiert darauf, dass das Input abgezogen wird, wenn das Output falsch ist. Sie wird zur Klassifikation verwendet. Das Perceptron teilt damit praktisch den Raum mit einer Hyperebene, wobei die Gewichte die Richtung angeben. Das Lernen bewirkt eine Drehung, wenn ein Punkt auf der falschen Seite steht, und die Konvergenz ist garantiert, sofern das Problem linear trennbar (Nachteil!) Verwendet man statt einer linearen Funktion eine Sigmoide, erhält man ein MLP, verwendet man eine gausssche Funktion, erhält man ein RBFN.

Der Gedanke hinter Kernels ist, eine fixe Transformation zu haben, die ein Problem linear trennbar macht (evtl hochdimensional). Der Kernel ist ist eine Funktion, die als inneres Produkt von Funktionen Phi darstellbar ist. Der Trick ist: diese Funktionen Phi müssen nicht einmal bekannt sein! Durch diese Transformation wird das Problem linear trennbar. Dadurch kann das Problem noch so hochdimensional sein, die Berechnung erfolgt im niedrigdimensionalen Raum.

16) Was ist ein Gaussscher Kernel, was ein Large Margin Classifier und wie optimiert man sie?

Die Kernel Matrix eines Gausschen Kernels hat vollen Rang, die Dimension ist so groß wie das Traningsset. Phi ist hierfür nicht darstellbar, hat aber unendliche Dimension.

Im hochdimensionalen Raum ist overfitting leicht möglich. Die Lösung ist die Suche nach einer Enscheidungslinie mit größtmöglichem Abstand zu den Punkten. Das Quadratische Optimierungsproblem wird durch Lagrange-Multiplikanten gelöst. Die Daten stehen danach wieder als inneres Produkt im Term, der Kernel Trick kann wieder angewendet werden. Ein globales Minimum ist garantiert. Methoden heirfür wären:

- Chunking
- Decompositional Methods
- Sequential Minimal Optimization (SMO) für Variablenpaare

17) Was sind Support Vectors, wie werden sie auf Rauschen erweitert? Was sind Bedingungen für Kernels? Was versteht man unter Shatter, wie ist die VC-Dimension definiert?

Support Vectors sind Punkte am Rande des Margins, sie bestimmen alleine die Lösung. Alle anderen Punkte könnten weggelassen werden. Bei Rauschen führt man „slack variables“ ein, die den strengen Margin etwas aufweichen. Das duale Problem (Lagrange) bleibt jedoch gleich.

Eine Kernelfunktion muss positiv definit sein. Sind K1 und K2 Kernels, sind auch aK1 (a>0), K1+K2 und K1*K2 Kernels. Die Wahl des richtigen Kernels ist entscheidend, es braucht also eine Modellselektion.

„Shatter“ bedeutet, dass unter n Punkten alle 2^n Klassifikationen möglich sind.

Die VC-Dimension h ist das kleinste m von Punkten, für die der Lerner weniger als 2 ^m Klassifikationen schafft. Für komplexe Lerner kann oft nur die Schranke angegeben werden, die VC-Dim des Perceptrons wäre k+1 (k…Inputdimension)

18) Wie wird in der SVM Theorie structural risk minimization betrieben? Wie hängen SVMs mit Neuronalen Netzen zusammen? Nennen Sie andere Kernelverfahren.

Die Schranke für das Risiko lässt sich mittels einer Formel berechnen. Das Maximieren dieses Margins beschränkt die VC-Dimension.
Der Zusammenhand mit Neuronalen Netzen ist wie folgt: Verwendet man einen Gauss-Kernel, erhält man ein RBF, verwendet man einen Sigmoid-Kernel, ein MLP. Man hat so viele „Hidden Units“ wie Trainingsmuster, es handelt sich allerdings um eine andere Berechnung und der Raum ist unendlichdimensional. Es gibt einen formalen Zusammenhang zwischen SVMs und Boosting – die Punkte an der Entscheidungsgrenze bekommen größte Bedeutung.
Andere Kernelverfahren wären zB Kernel-PCA, Kernel-Fisher Diskriminante, Kernel Regression oder Gauss’sche Prozesse.

Independent Component Analysis

19) Was ist das Cocktail-Party Problem? Was sind die Ausgangspunkte der ICA? Was die Bedingungen?

Beim Cocktail-Party Problem gibt es zwei Tonquellen und zwei Mikrofone an verschiedenen Orten. Die Signale werden linear gemischt, man sucht die Originalsignale, aber die Mischmatrix ist unbekannt.

Dies ist Ausgangspunkt der ICA: Annahme ist x = As und s = Wx=A^-1 x
Lässt man den Zeitindex weg, so ist eine „blind source separation“ möglich bis auf multiplikative Faktoren (Amplitude der Quellen) und Reihenfolge (Permutation).

Die Bedingungen der ICA sind:

- Mindestens so viele Mischungen wie Quellen
- Quellen müssen unabhängig sein (p(a,b)=p(a)*p(b))
- Maximal eine der Quellen darf Gauss-verteilt sein
- Die Quellen sollten Mittelwert 0 und Varianz 1 haben (-> Vorverarbeitung)

20) Wie funktioniert der Ansatz der ICA? Welches Maß wird hierbei verwendet? Welche alternativen Maße gäbe es sonst?

Nach dem Grenzwertsatz tendiert die Summe von beliebigen Verteilungen zu einer Gaussverteilung. Man nimmt y als eine Linearkombination der Originalsignale, es ist mehr Gaussverteilt als jedes Originalsignal. Man versucht, y so „nicht-Gauss“ wie möglich zu machen, indem nur eine Komponente von z (=A^T w) nicht 0 ist.

Gesucht ist ein Maß, wie „nicht-Gauss“ eine Verteilung ist, um dieses Maß in einem Lernproblem zu Optimieren (zB mittels Gradientenverfahren).

Ein mögliches Maß ist die Kurtosis, für eine Gauss-Verteilung hat man kurt(y)=0, gilt kurt(y)<0 ist die Funktion supergauss, ist kurt(y)>0 ist sie subgauss.
Andere Mögliche Maße wären die Negentropy (Maß der Geordnetheit, für Gauss maximal), die Mutual information (berücksichtigt Signalbeziehungen) oder die Maximum Likelihood.

21) Was sind die Anwendungsmöglichkeiten der ICA? Wie hängt sie mit der PCA zusammen, wie mit Neuronalen Netzen? Was ist FastICA?

Die ICA wird bei Blind Source Separation von Tonsignalen, EEGs oder Finanzdaten verwendet,, oder auch bei Merkmalserkennung oder Dimensionsreduktion.

Bei der PCA müssen im Gegensatz zur ICA die Richtungen der größten Varianz (statt stärksten nicht-gauss) orthogonal aufeinander stehen. Ein MLP mit linearen Hidden Units führt eine PCA durch wenn Input=Output.

Mit speziellen nichtlinearen Regeln wird das Netz zur ICA, an nimmt eine Approximation der Negentropy und führt eine Gewichtsänderung durch, dies ist dem MLP (Hebbsches Lernen) verwandt.