Barbara Hammer, FB Mathematik/Informatik, Universität Osnabrück
and special guests: Kai Gersmann, Marc Strickert :)

Neuronale Netze (SS 2002)
9. Übungsblatt

Abgabe: Donnerstag, 20.6.02, $12^{\underline{00}}$ Uhr, Briefkasten ‚Neuronale Netze' im 4. Stock des AVZ

---

1. (5 Punkte) Beschreiben Sie umgangssprachlich worin sich die SVM vom Perzeptron unterscheidet? Was ist der Nutzen?

   Eine lineare SVM ohne Bias wird für die folgenden Punkte trainiert:

   $\{(1,0;0),(2,0;0),(3,0;0),(1,1;0),(2,1;0),(2,3;0),(1,4;0),$
   $(-1,0;1),(-1,1;1),(-2,0;1),(-2,-1;1)\}.$

   Wie sieht die sich ergebende SVM aus und welche Punkte können Support Vektoren werden?

2. (5 Punkte) Die SVM sieht eigentlich nicht nach Netz aus, sondern wurde als einfaches Perzeptron mit geeigneter Vorverarbeitung interpretiert. Man kann sie für geeignete Kerne aber als Netz schreiben. Angenommen, Sie haben eine SVM mit Polynomkern $k(\vec{x},\vec{y}) = (\vec{x}^t\vec{y}+1)^p$ trainiert und den Gewichtsvektor $\vec{w}$ erhalten. Wie kann man die sich ergebende SVM als feedforward Netz mit nur Polynomen als Aktivierungsfunktion in den hidden Neuronen und der Perzeptronaktivierung als Ausgabeaktivierung schreiben?

3. (5 Punkte) Was passiert bei LVQ im Mittel, wenn man lediglich mit einem Prototypen für die Daten einer einzigen Klasse lernt? (Klar, dann ist natürlich alles konstant auf die eine Klasse abgebildet – aber wo liegt der Prototyp?)

   LVQ mit nur zwei Prototypen legt einfach eine Gerade/Hyperebene in den Raum, entspricht also einem einfachen Perzeptron. Das Trainingsverhalten ist aber anders. Versuchen Sie, informell z.B. geometrisch zu argumentieren: findet LVQ mit zwei Prototypen und zwei Klassen immer eine Lösung, falls diese existiert? Oder legt LVQ auch bei zwei Prototypen und zwei Klassen noch die Prototypen im Mittel in die beiden Klassenschwerpunkte?

4. (5 Punkte) Trainieren Sie mit dem SNNS die Ziffern aus digits.pat mithilfe einer DLVQ-Architektur. Dokumentieren Sie das Training. Was kommt bei den Ziffern aus test.pat als Klassifikation heraus?

Mehr zum SNNS:
Dynamisches LVQ-Lernen/DLVQ: Das ist LVQ-Lernen, bei dem zunächst von der minimalen Anzahl benötigter Referenzvektoren (=Anzahl Klassen) ausgegangen wird und diese Anzahl sukzessive nach Bedarf erhöht wird. Das funktioniert im SNNS so: Die Pattern enthalten die Eingabe und zugeordnete Klasse. Die Netzarchitektur besteht zunächst aus entsprechend vielen Eingaben und genau einer Ausgabe (z.B. mit feedforward erzeugt), Initialisieren mit DLVQ_Weights (alle Hidden-Units werden gelöscht und die Referenzvektoren als Klassenmittelpunkte berechnet), Lernen mit Dynamic_LVQ, sofern ein Punkt falsch klassifiziert wurde, wird der nächste richtige Referenzvektor w um $\eta^+(x-w)$ geändert und der falsche Referenzvektor um $\eta^-(x-w)$. Das wird n-mal vorgenommen. (Parameter: η^+ z.B.$0.03, \eta^-$ z.B.$0.03, n$, Rest nicht benötigt). Danach werden evtl. aus den falsch klassifizierten Vektoren neue Referenzvektoren gewonnen (für jede Klasse maximal einer) und dem Netz hinzugefügt, das ganze maximal CYCLES-mal iteriert. Update-Funktion: Dynamic_LVQ.